trackme Documentation

Release 2

TrackMe Limited, U.K.

Jul 14, 2025

QUICKSTART

1 Quickstart: 5

2 Use Cases Demo: 27

3 License & support: 119

4 Compatibility and download: 135

5 Requirements: 139

6 Installation: 143

7 Administration guide: 169

8 White papers: 559

9 User guide: 707

10 Troubleshoot & FAQ: 829

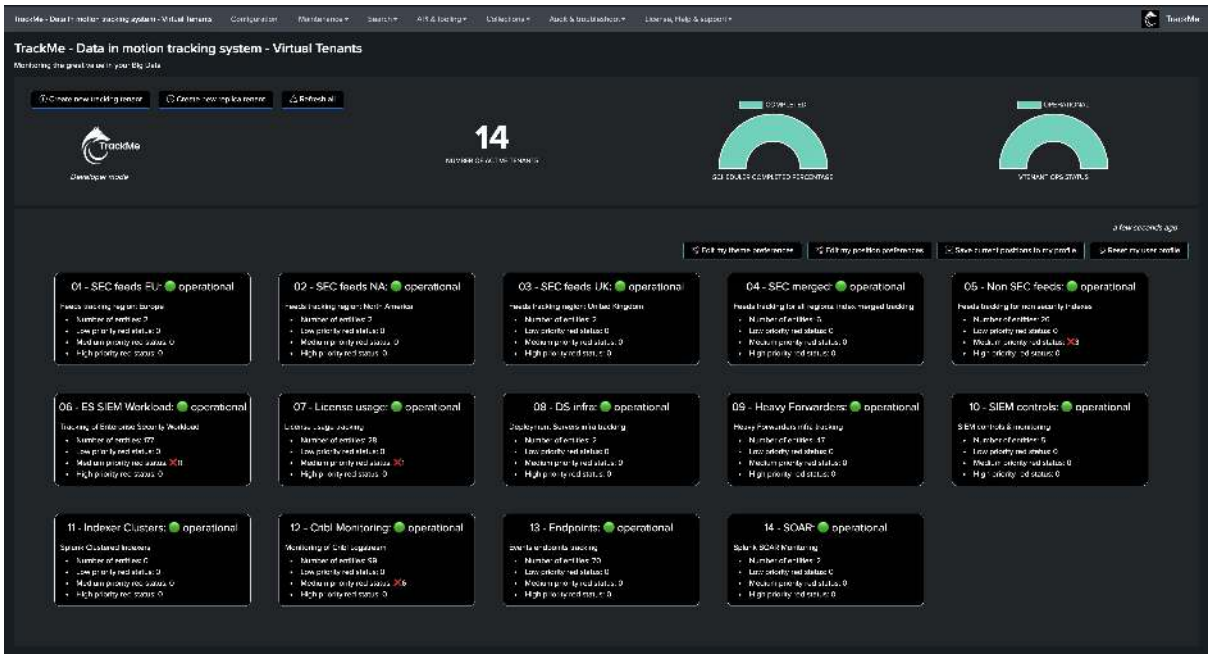
11 Versioning and build history: 839

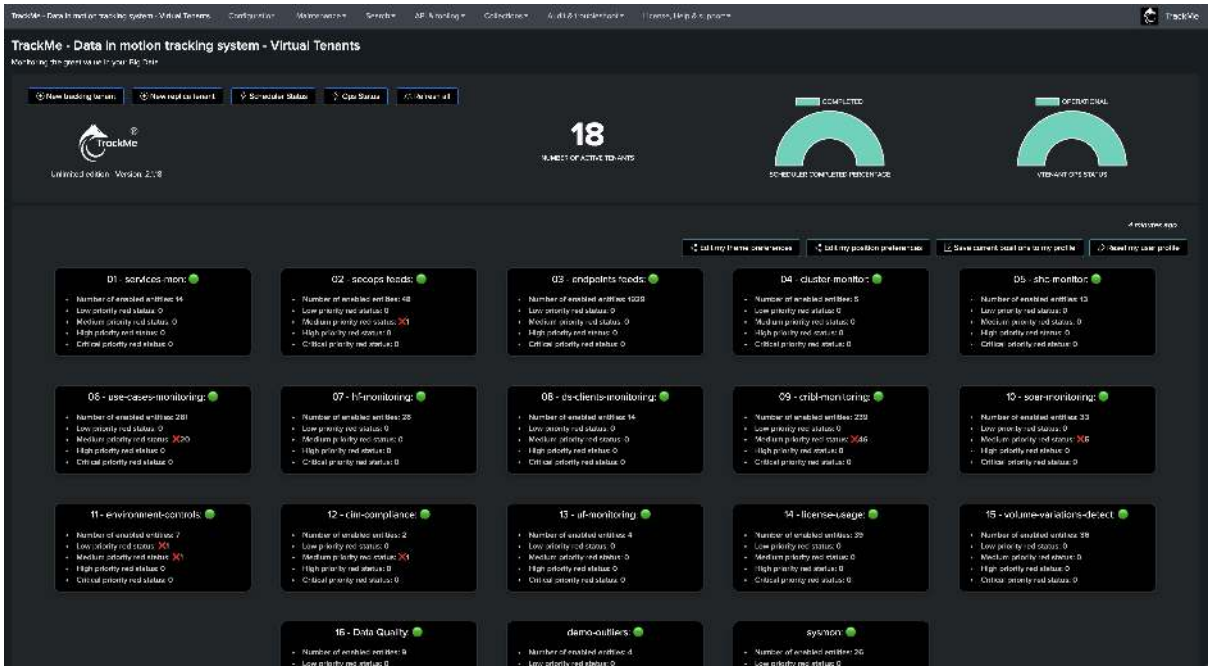
12 Various 943

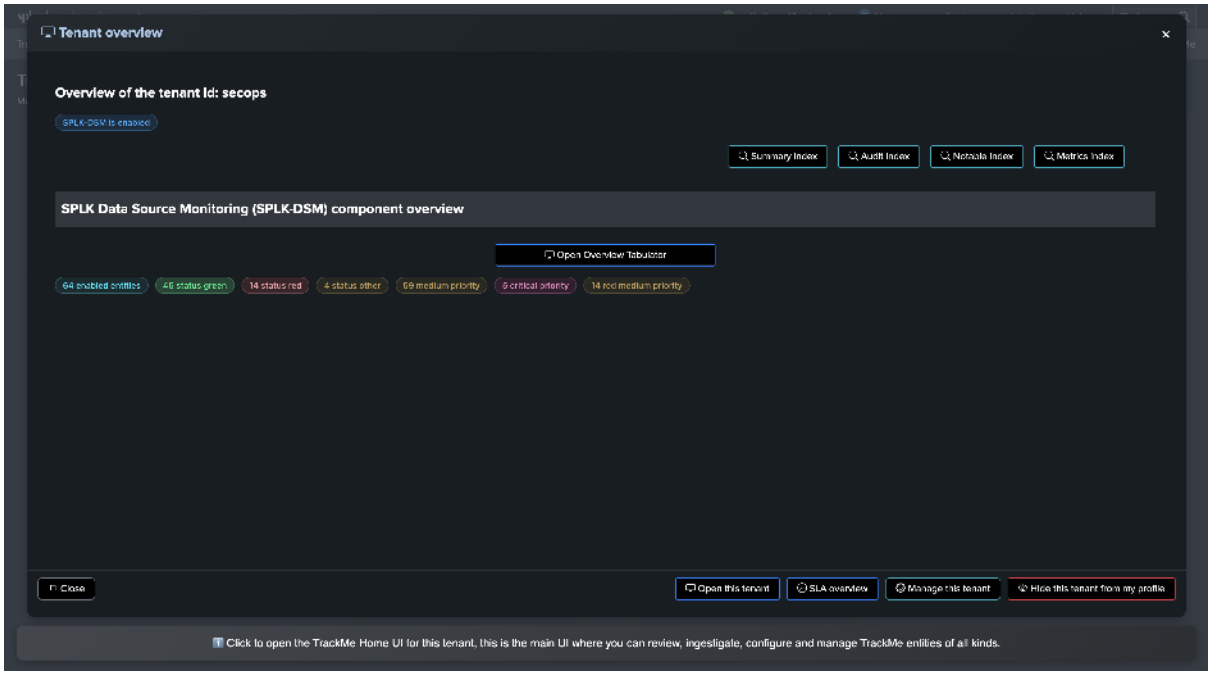
The PDF version of this documentation is available here.

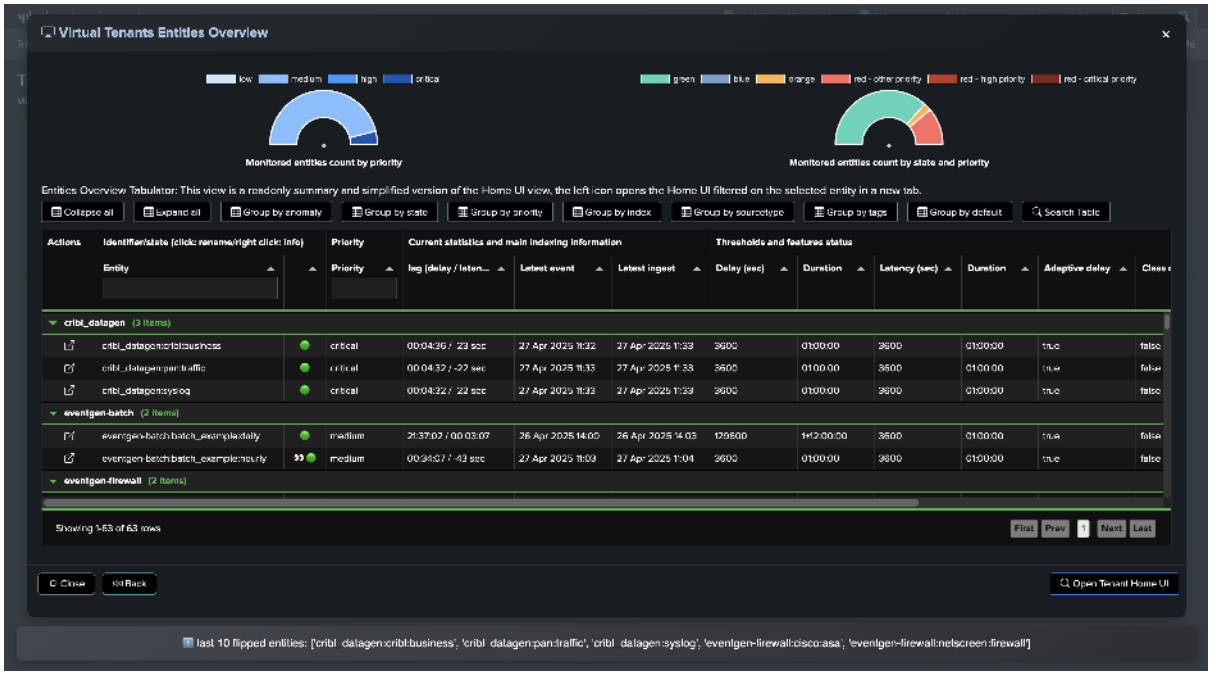
TrackMe for Splunk is the ideal companion for your Splunk deployment, no matter the size of your environment. Its unique capabilities help you on a daily basis to get the best value from your Splunk investments:

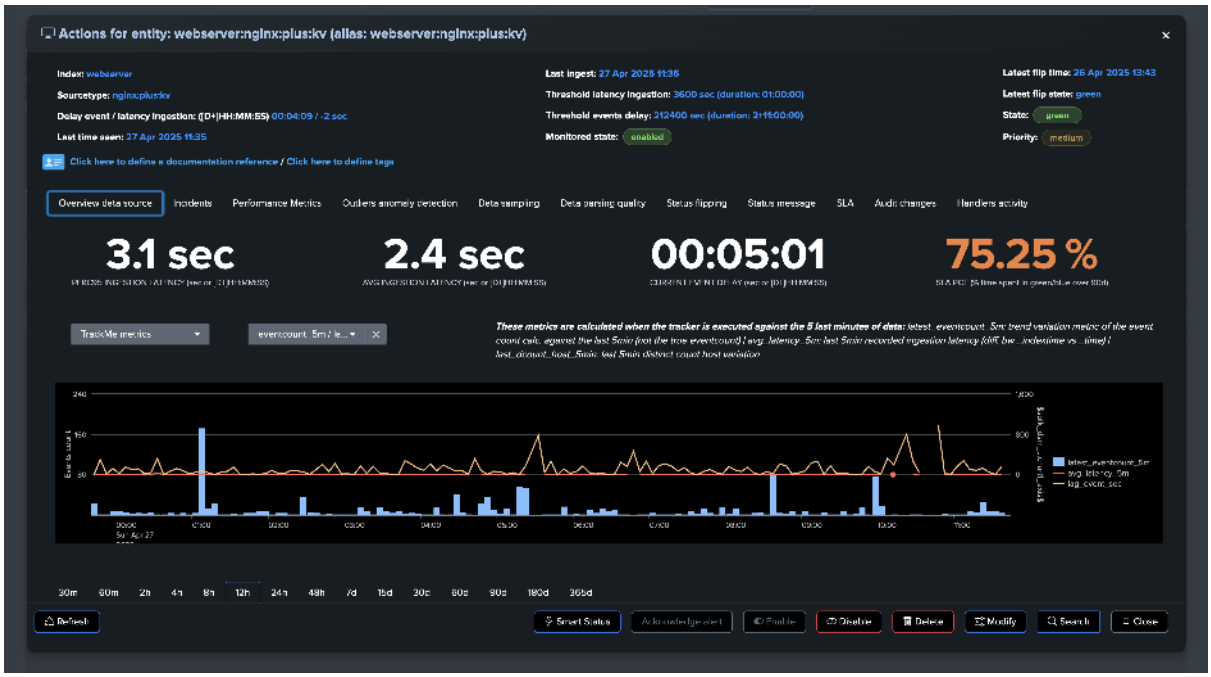
- Discover and maintain Splunk entities at scale, track availability and quality of any kind of data in Splunk
- Virtual tenancy is a key concept in TrackMe which allows creating on the fly knowledge objects in a repeatable way: create and scope tenants, experiment, destroy and restart as needed
- TrackMe allows tracking your local Splunk deployment, or transparently any number of Splunk remote deployments (subject to licensing restrictions)
- TrackMe's unique workflow combines best Splunk capabilities, from a comprehensive user interface to notable events generation, SLA tracking and many more
- Get the best from TrackMe components, using splk-feeds components provide deep Splunk feed tracking, splk-cim provides Common Information Model (CIM) compliance tracking, splk-flx (FLEX) adapts to any kind of Splunk magic query! (components are subject to license restrictions)
- Extend the visibility at any point in time with Hybrid and Elastic trackers, use Machine Learning outliers detection with deep and easy control, TrackMe is incredibly rich in features




TrackMe - Data in motion tracking system - Virtual Tenants
18
01 - services-mon:
02 - secops feeds:
03 - endpoints feeds:
04 - cluster monitor:
05 - shc monitor:
06 - use-cases-monitoring:
07 - hf-monitoring:
08 - ds-clients-monitoring:
09 - cribl-monitoring:
10 - soar-monitoring:
11 - environment-controls:
12 - cim-compliance:
13 - ufi-monitoring:
14 - license-usage:
15 - volume-variations-detect:
16 - Data Quality:
demo-outliers:
sysmon:


Tenant overview
Overview of the tenant id: secops
SPLK-DSM is enabled
Summary Index
Audit Index
Notable Index
Metrics Index
SPLK Data Source Monitoring (SPLK-DSM) component overview
Open Overview Tabulator
64 enabled entities
46 status green
14 status red
4 status other
59 medium priority
5 critical priority
14 red medium priority
Close
Open this tenant
SLA overview
Manage this tenant
Hide this tenant from my profile
Click to open the TrackMe Home UI for this tenant, this is the main UI where you can review, ingest/gate, configure and manage TrackMe entities of all kinds.


Virtual Tenants Entities Overview
Monitored entities count by priority
Monitored entities count by state and priority
Entities Overview Tabulator: This view is a readonly summary and simplified version of the Home UI view, the left icon opens the Home UI filtered on the selected entity in a new tab.
Collapse all
Expand all
Group by anomaly
Group by static
Group by priority
Group by index
Group by sourcetype
Group by tags
Group by default
Search table
Actions
Identifier/state (click: rename/right click: info)
Priority
Current statistics and main indexing information
Thresholds and features status
Entity
Priority
lag (delay / laten...
Latest event
Latest ingest
Delay (sec)
Duration
Latency (sec)
Duration
Adaptive delay
cribl_datagen (3 items)
cribl_datagen:criblbusiness
cribl_datagen:pantraffic
cribl_datagen:syslog
eventgen-batch (2 items)
eventgen-batch:batch_exemplardaily
eventgen-batch:batch_exemplarhourly
eventgen-firewall (2 items)
Showing 1-63 of 63 rows
First
Prev
Next
Last
Close
Refresh
Open Tenant Home UI
last 10 flipped entities: ['cribl_datagen:cribl:business', 'cribl_datagen:pan:traffic', 'cribl_datagen:syslog', 'eventgen-firewall:cisco:asa', 'eventgen-firewall:netscreen:firewall']


Actions for entity: webserver:nginx:plus:kv (alias: webserver:nginx:plus:kv)
Index: webserver
Sourcetype: nginx:plus:kv
Delay event / latency ingestion: (DD|HH:MM:SS) 00:04:09 / -2 sec
Last time seen: 27 Apr 2025 11:35
Last ingest: 27 Apr 2025 11:35
Threshold latency ingestion: 3600 sec (duration: 01:00:00)
Threshold events delay: 212400 sec (duration: 2+11:00:00)
Monitored state: enabled
Latest flip time: 26 Apr 2025 13:43
Latest flip state: green
State: green
Priority: medium
Click here to define a documentation reference / Click here to define tags
Overview data source
Incidents
Performance Metrics
Outliers anomaly detection
Data sampling
Data parsing quality
Status flipping
Status message
SLA
Audit changes
Handlers activity
3.1 sec
2.4 sec
00:05:01
75.25 %
TrackMe metrics
These metrics are calculated when the tracker is executed against the 5 last minutes of data: latest_eventcount_5m: trend variation metric of the event count calc. against the last 5min from the raw eventcount / avg_latency_5m: last 5min recorded ingestion latency (diff_hw - indextime vs _time) / last_dcount_host_5min: last 5min distinct count host variation
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
180d
365d
Refresh
Smart Status
Acknowledge alert
Enable
Disable
Delete
Modify
Search
Close

CHAPTER

ONE

QUICKSTART:

1.1 QUICK START - Starting with TrackMe: (feed tracking quick-start)

Starting with TrackMe!

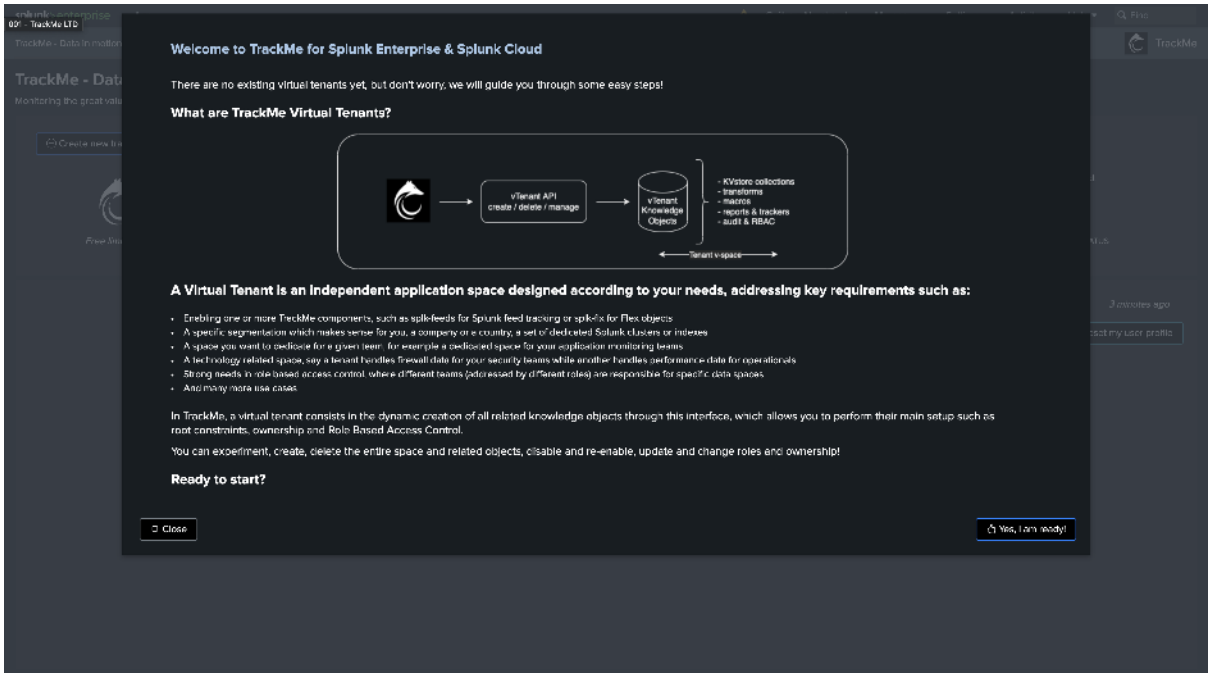
- This tutorial is a starting point for TrackMe new joiners!
- The objective is to help you get started with TrackMe, understand the basic concepts and focus on feeds Tracking for Splunk.
- In this tutorial, we will assume that you are starting entirely from scratch, and we will cover the essential steps to set up feed tracking in TrackMe, with main best practices.
- **This tutorial works in 3 main steps and can be used by all users, including Free community users:**
  - **Step 1:** Create a Virtual Tenant for data sources tracking
  - **Step 2:** Create a Virtual Tenant for data host sources
  - **Step 3:** Alerting and notifications
- This quickstart tutorial was last updated on the 25th of May 2025, to reflect the latest TrackMe releases and especially the new Stateful Alerting features.

1.1.1 TrackMe is installed, what now?

Once TrackMe is installed, the following wizard will guide you through the initial setup of a new Virtual Tenant:

About Virtual Tenants:

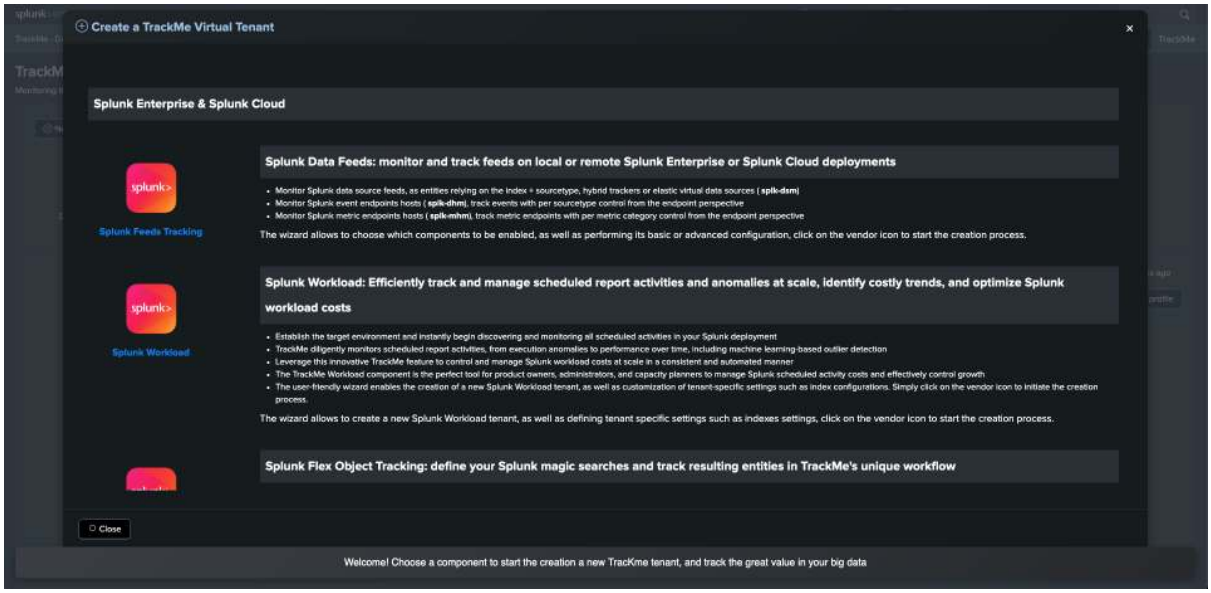
- Virtual Tenants are an essential core feature in TrackMe, this allows TrackMe to orchestrate and manage TrackMe-related knowledge objects and configuration, and address many powerful concepts such as multi-tenancy, data isolation, and more.
- You can also understand this concept as virtual instances of TrackMe within TrackMe, which you can create, destroy, and restart as needed.
- Virtual Tenants enable TrackMe components, which are designed to address use cases, such as **splk-dsm** which is the essential component for feed tracking.



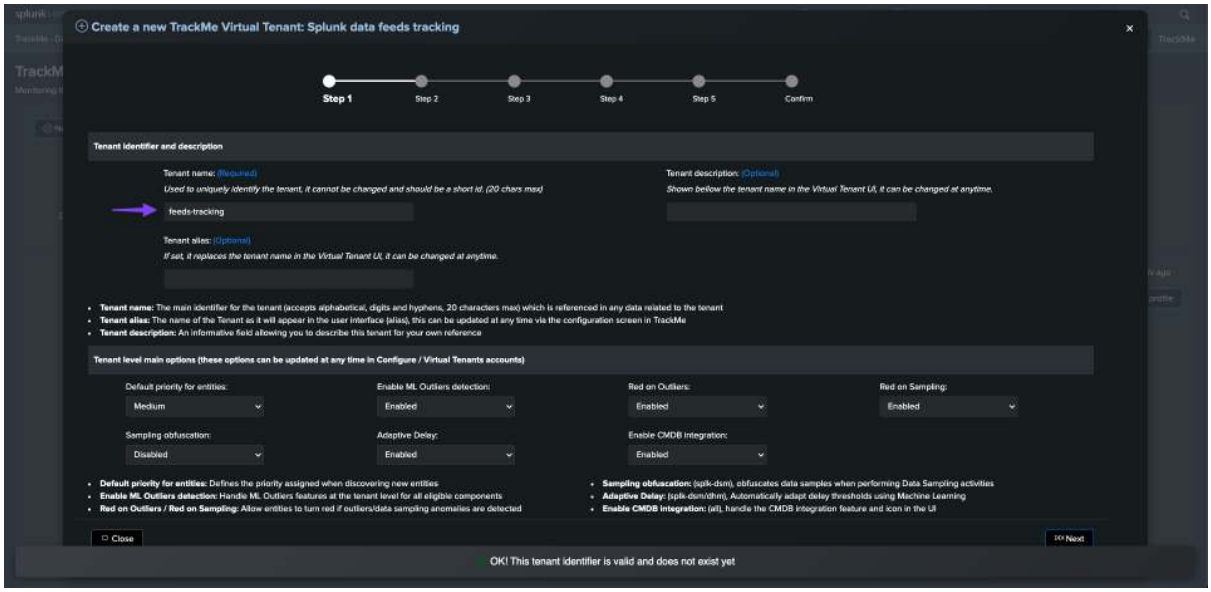
1.1.2 Step 1: Let's create a Virtual Tenant for data sources tracking

We will now create a Virtual Tenant to start tracking data sources:

- component: `Splunk Feeds Tracking`

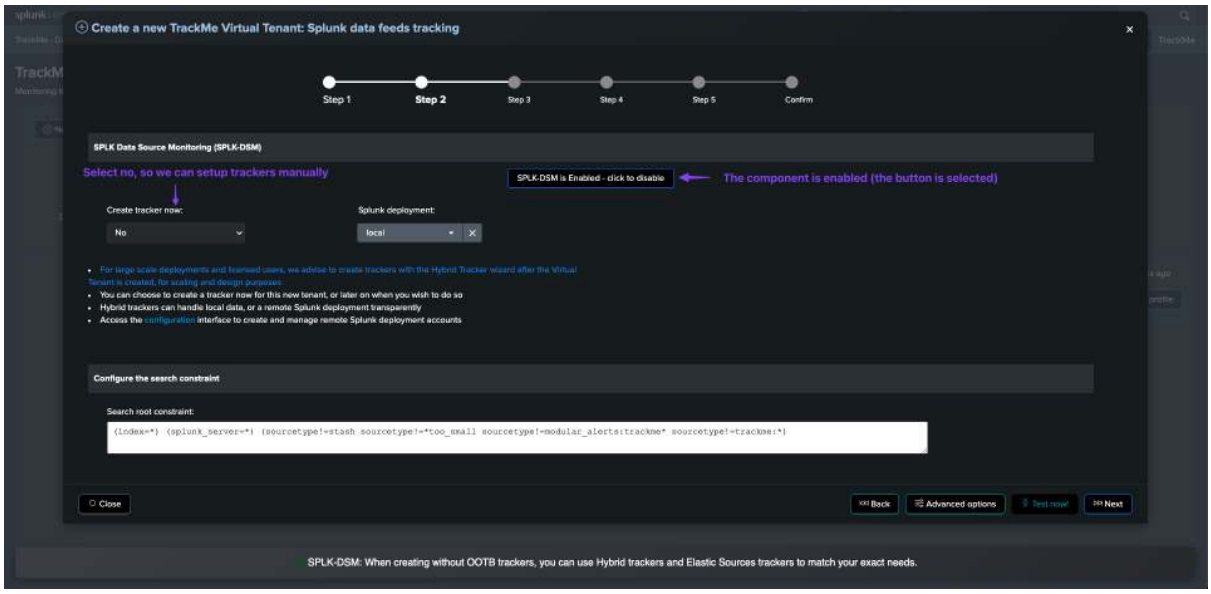


- tenant name: `feeds-tracking`
- tenant alias: This is optional and can be updated at any time. If set, the alias replaces the tenant identifier in the Virtual Tenants UI.
- tenant description: This is optional and can be updated at any time. If set, the description is displayed in the Virtual Tenants UI.

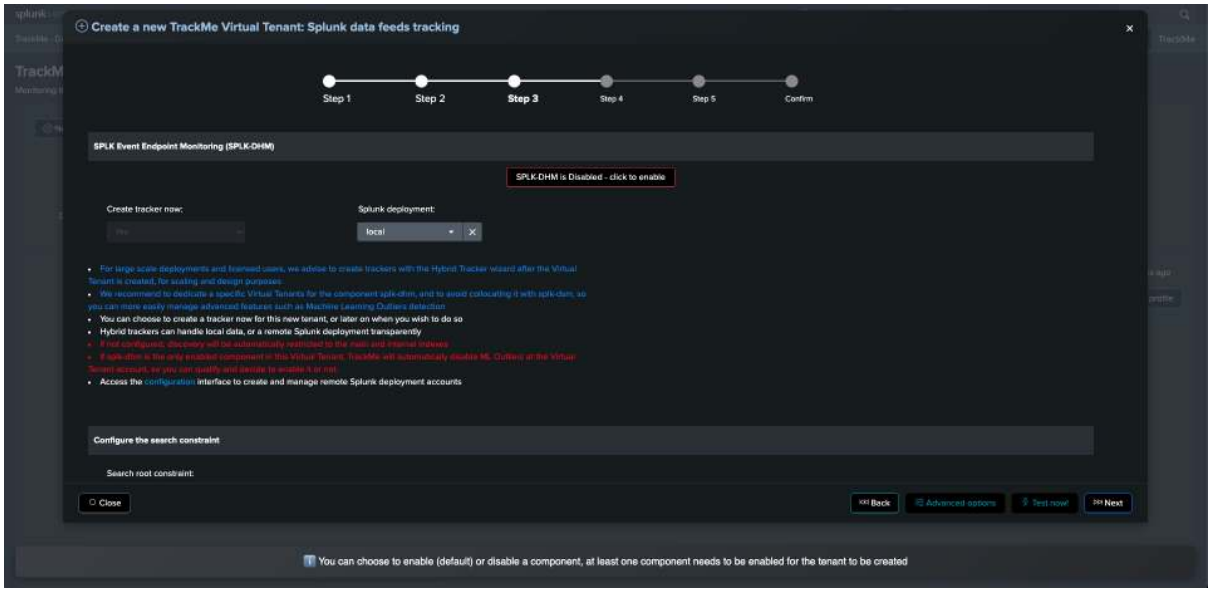


Creating tracker during the Virtual Tenant creation, or later on via the Hybrid Tracker management UI

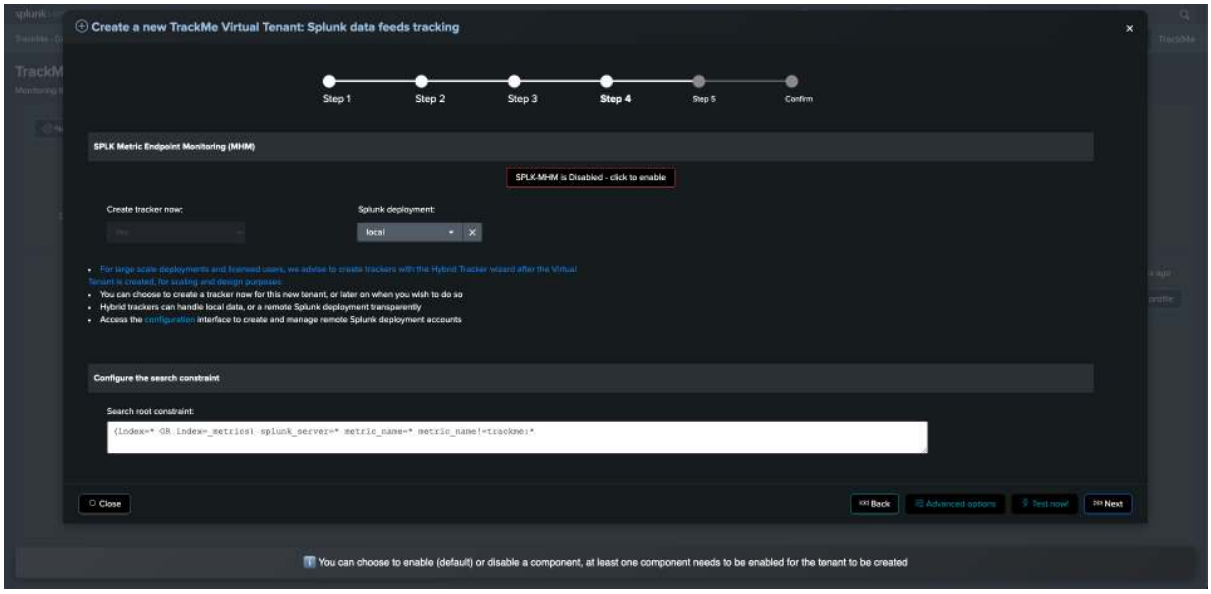
- The next 3 screens of the wizard for feed tracking are about the configuration of the TrackMe components (splk-dsm, splk-dhm, splk-mhm).
- By default, only the `splk-dsm` component is enabled.
- If your environment is small enough (less than 2/3 TB per day of ingestion), you can leave the default configuration and create the tracker during this stage (so leave "Yes" for Create Hybrid tracker).
- For larger environments, and up to any super large environments, we recommend disabling the creation of the Virtual Tenant and creating the tracker later on manually via the Hybrid Tracker management UI.



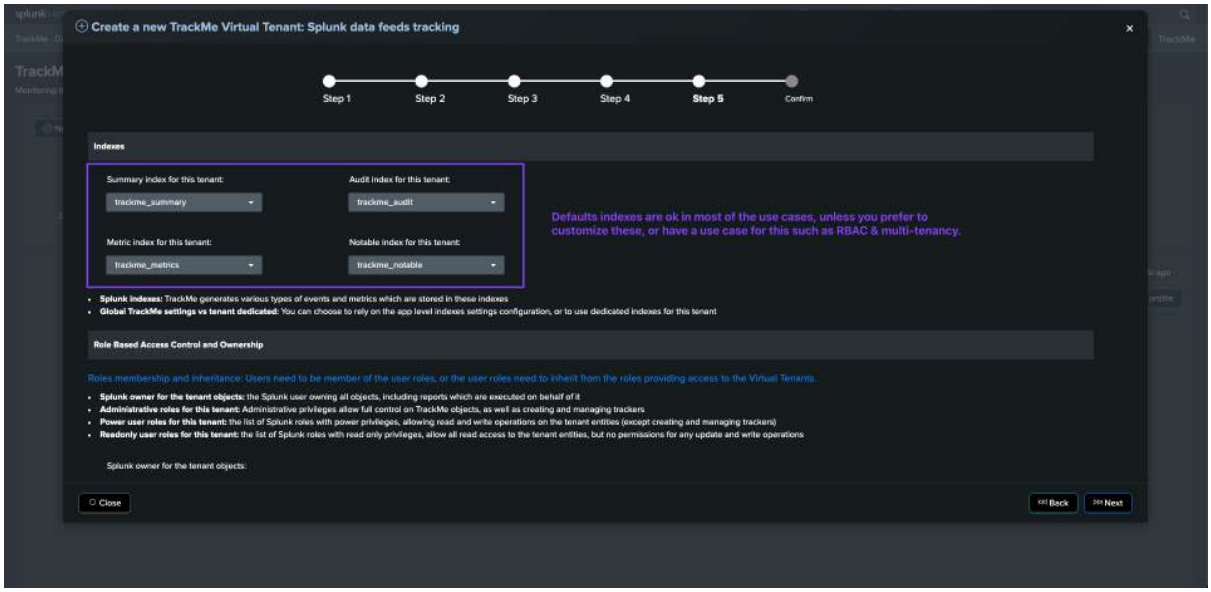
- In the next screen, the `splk-dhm` component is disabled by default, we will leave it disabled, this component is designed to track hosts and we will discuss it later in this tutorial.



- In the next screen, the `splk-mhm` component is disabled by default, we will leave it disabled, this component is designed to track metrics and we will discuss it later in this tutorial.



- The final configuration screen and its top section allows defining TrackMe indexes, in most cases, the default indexes are fine, and we can leave the default configuration.



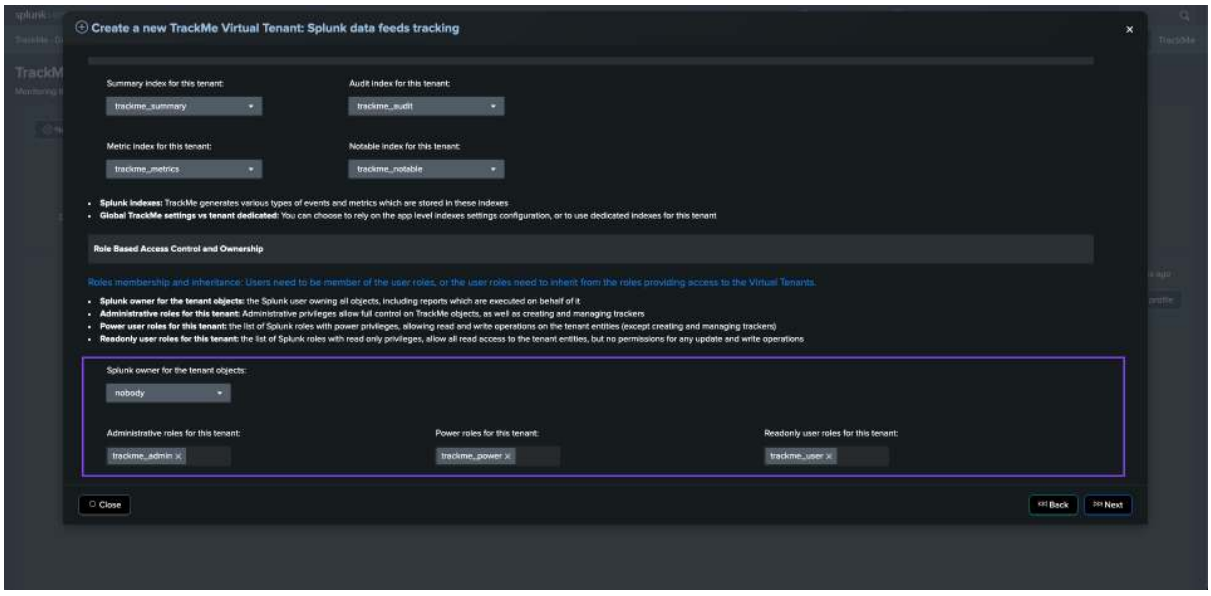
- The second part of that same screen allows defining ownership and permissions, all of these can be easily modified via TrackMe at any point in time, in short:

Owner:

- Default owner is nobody, which refers to the splunk-system-user, this means that TrackMe will automatically reassign the ownership of the knowledge objects to the splunk-system-user.
- This is fine for most cases and is a common practice in Splunk.
- On large or strict environments, creating a service account can be considered as good practice and notably allows tracking TrackMe's related activity easily, leveraging Splunk Workload Management, see: :ref: *trackme_admin_config* for more information.

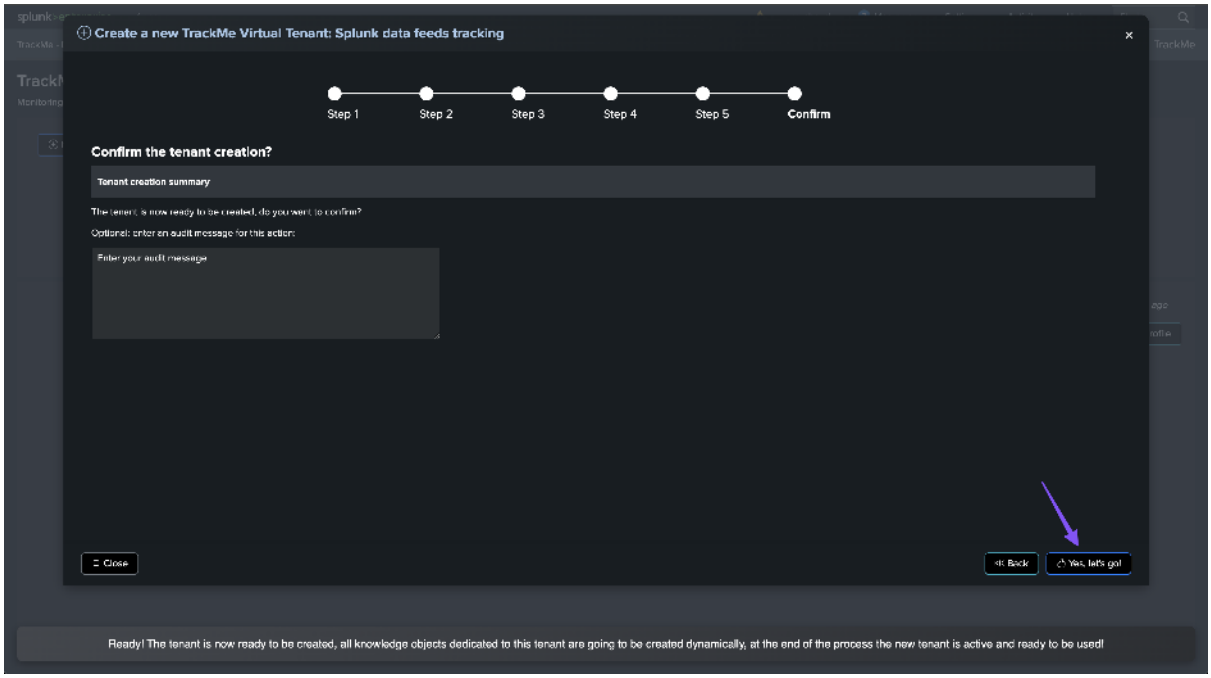
Permissions:

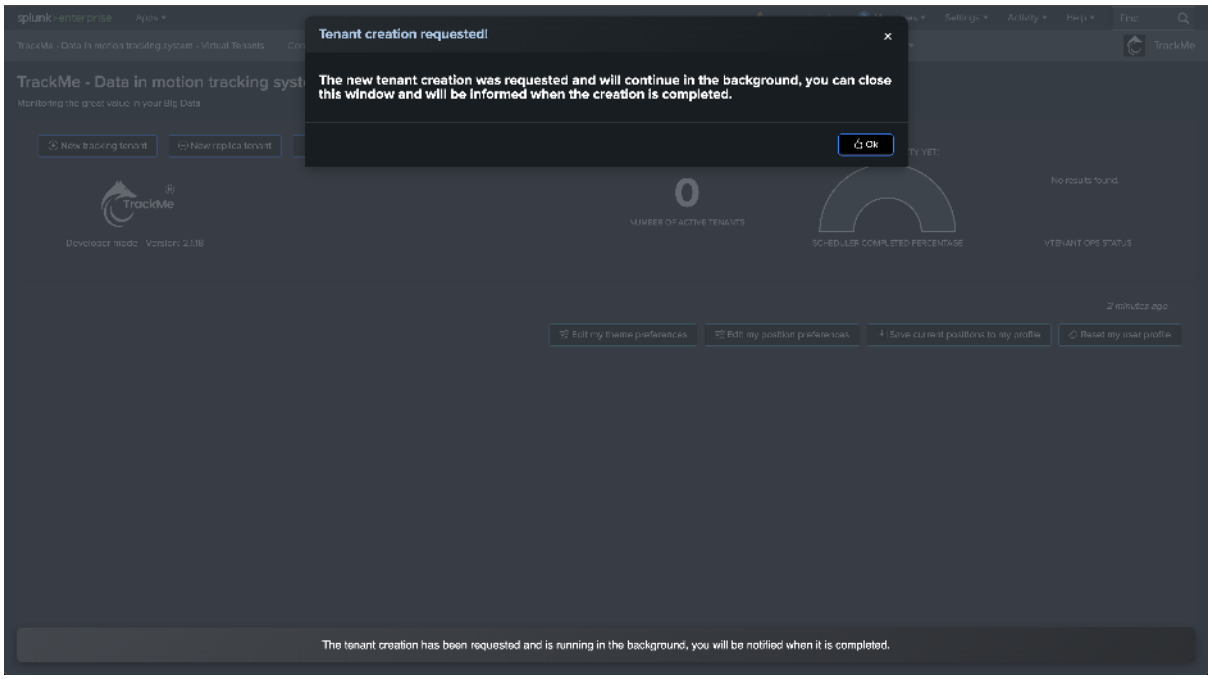
- The default permissions leverage a 3-dimensional model, from admin to power and finally read-only users.
- As a start, you generally can leave everything as default



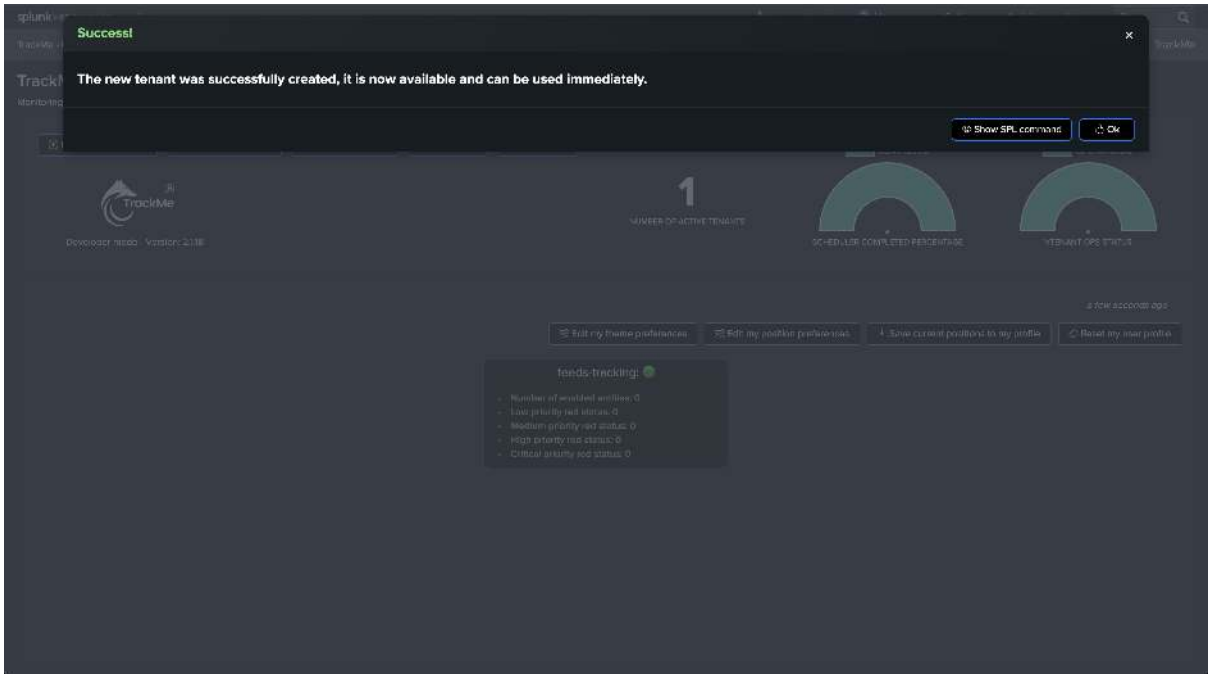
Validate the creation of the Virtual Tenant:

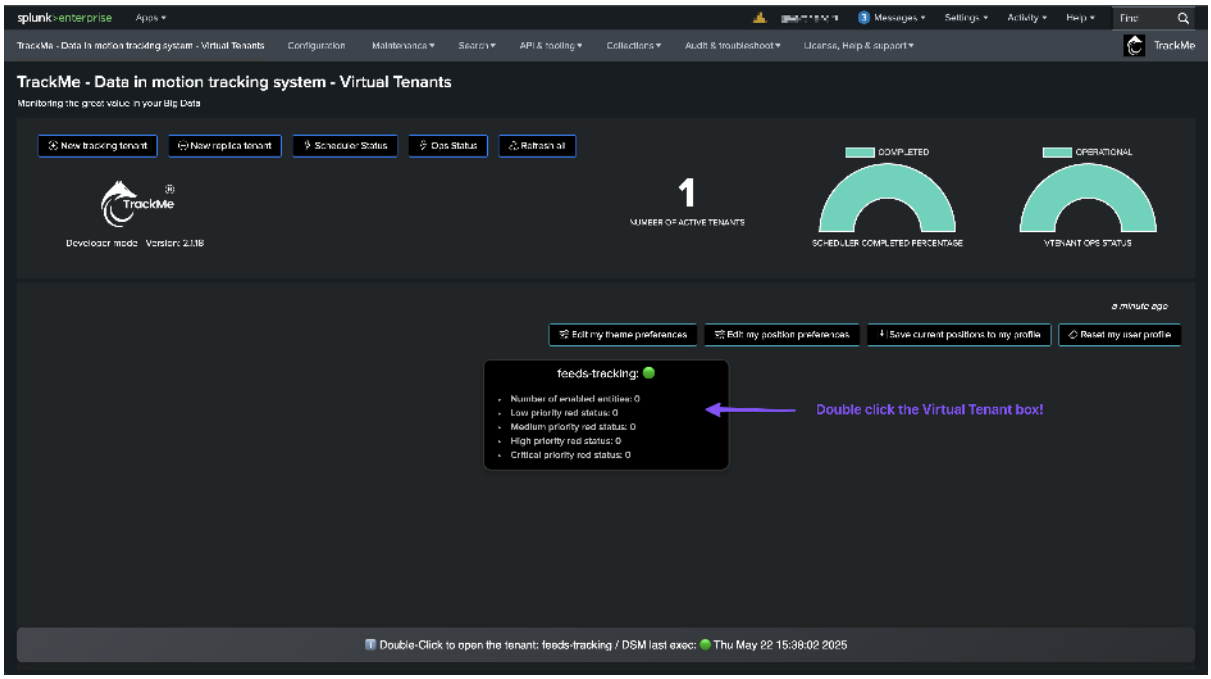
We can now validate the creation of the Virtual Tenant, after a short moment, the Virtual Tenant UI will be refreshed and a Virtual Tenant will now be available:

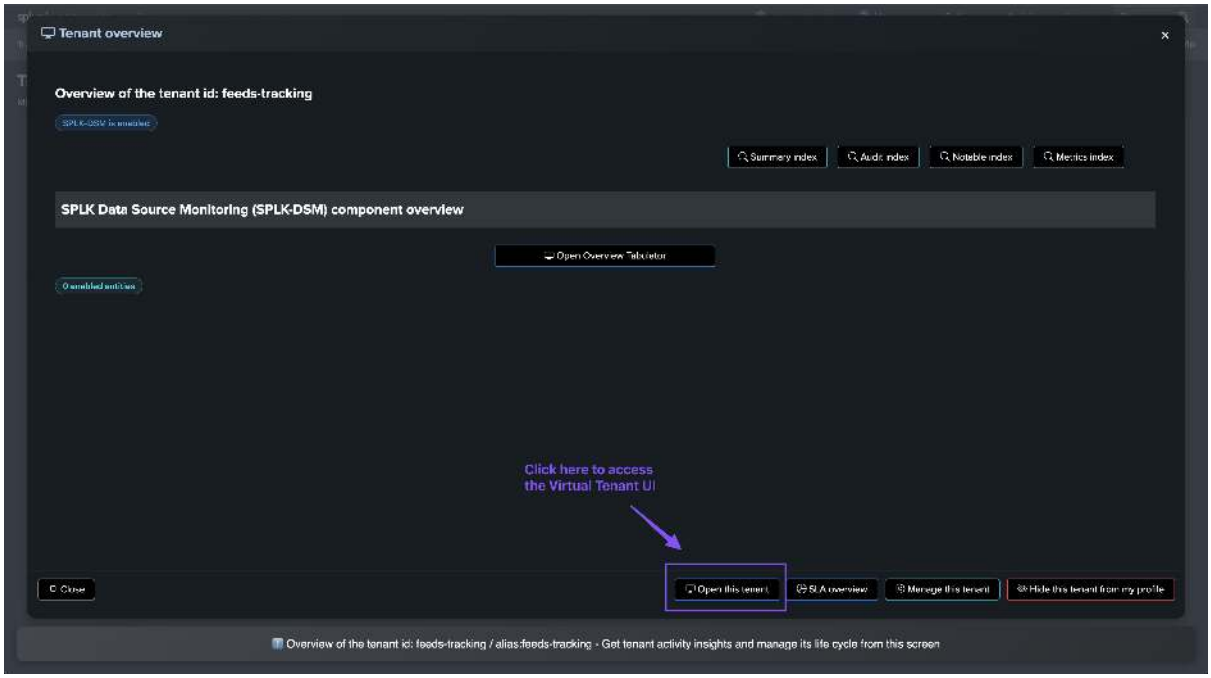




Double click on the Virtual Tenant box to open the following screen:

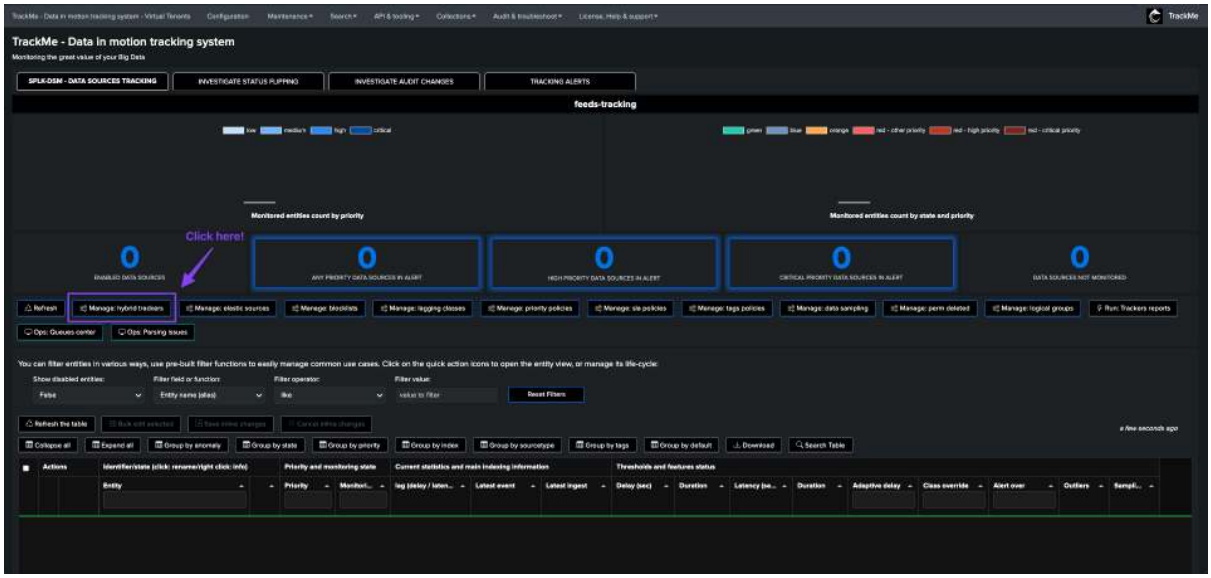

Success!
The new tenant was successfully created, it is now available and can be used immediately.
Show SPL command
Ok
1
feeds-tracking:

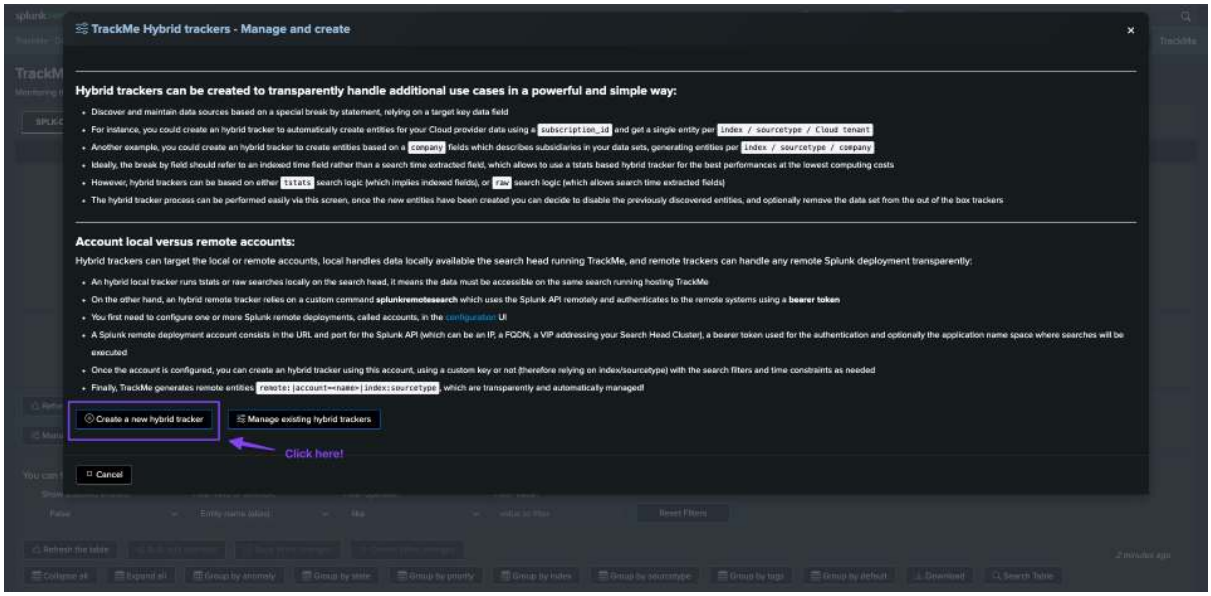

TrackMe - Data in motion tracking system - Virtual Tenants
Monitoring the great value in your Big Data
New tracking tenant
New replica tenant
Scheduler Status
Ops Status
Refresh all
1
NUMBER OF ACTIVE TENANTS
COMPLETED
SCHEDULER COMPLETED PERCENTAGE
OPERATIONAL
TENANT OPS STATUS
a minute ago
Edit my theme preferences
Edit my position preferences
Save current positions to my profile
Reset my user profile
feeds-tracking:
Number of enabled entities: 0
Low priority red status: 0
Medium priority red status: 0
High priority red status: 0
Critical priority red status: 0
Double click the Virtual Tenant box!
Double-Click to open the tenant: feeds-tracking / DSM last exec: Thu May 22 15:39:02 2025



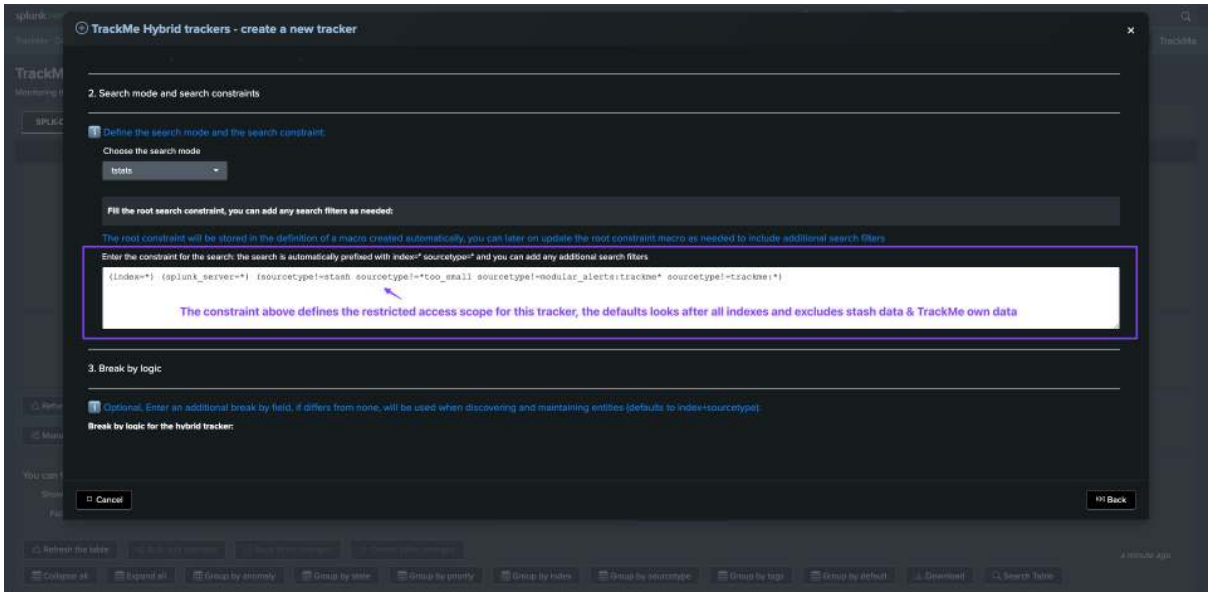
Create a hybrid tracker for splk-dsm

We will now create a hybrid tracker for the splk-dsm component, enter the freshly created Virtual Tenant:



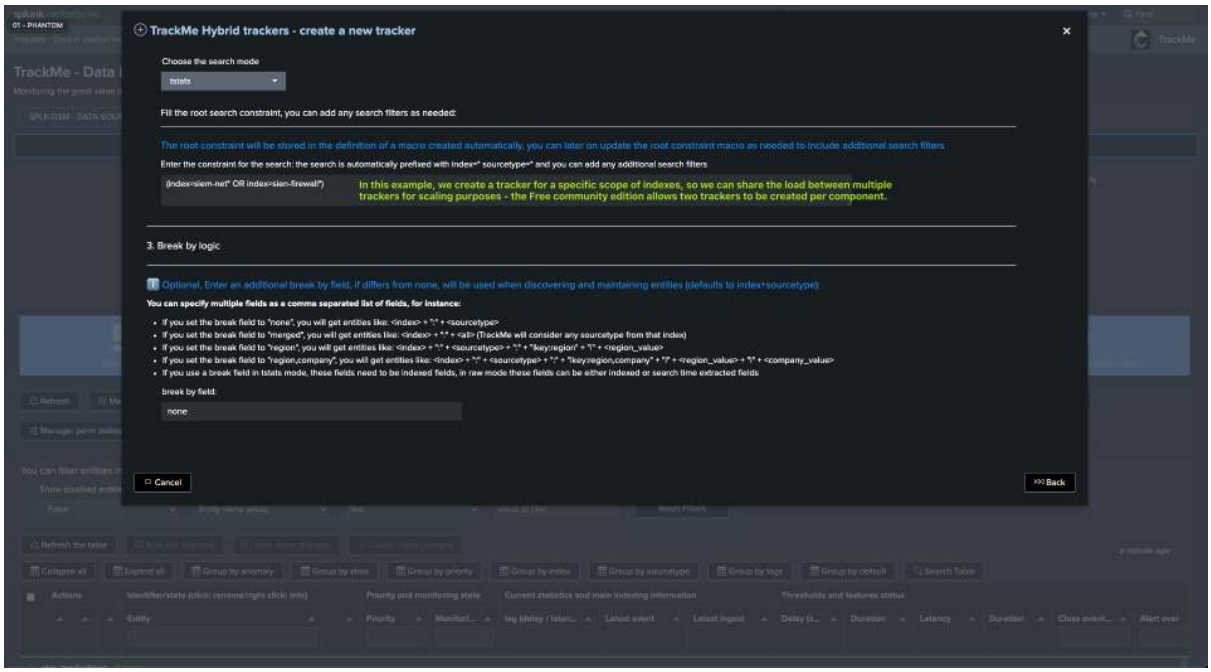


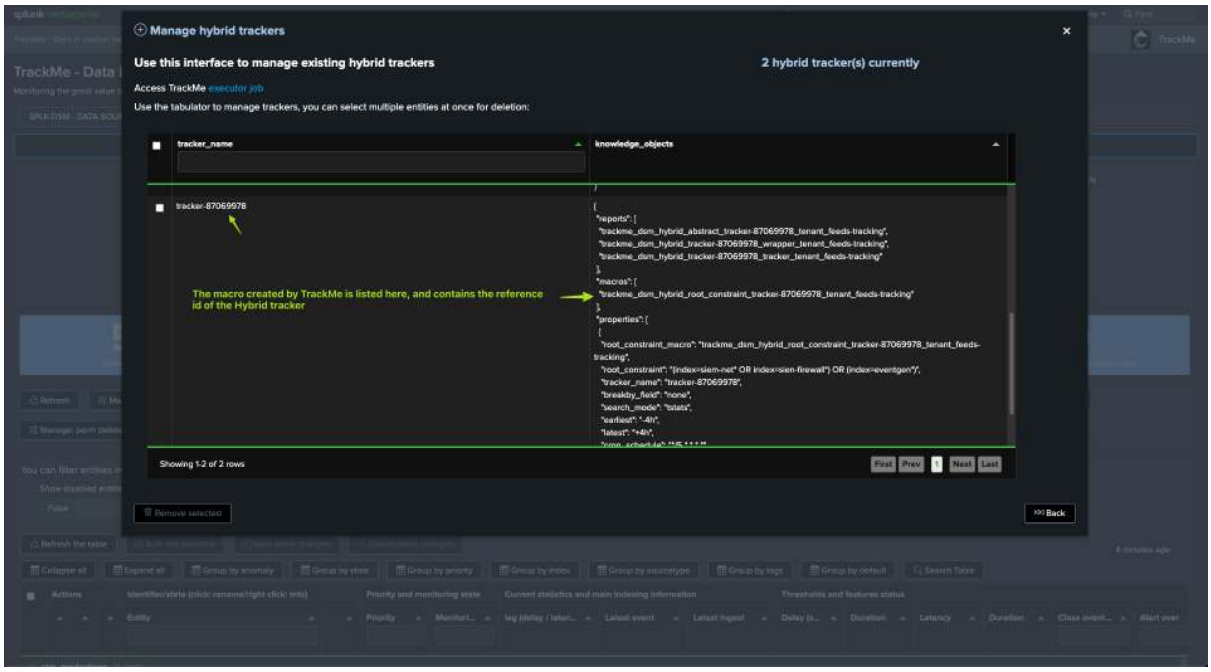
- Scroll down to the scope of the tracker root constraint:
  - The default root constraint tracks all indexers, and excludes summary data (stash) and TrackMe-related data.
  - On large environments, the scaling concept for TrackMe is to dedicate Hybrid Trackers to specific indexes scope, so trackers can share the load and be executed concurrently, allowing TrackMe to scale at any level.
  - On relatively small environments, the default configuration for a single tracker is in most cases able to cope with the load with no issues.
  - Finally, if you are a free community edition user, you can create up to two Hybrid Trackers per component.
  - For more information about scaling TrackMe for high-scale environments, please refer to the TrackMe documentation, notably: *Large Scale Environment and Best Practices Configuration Guide*



About Trackers scope:

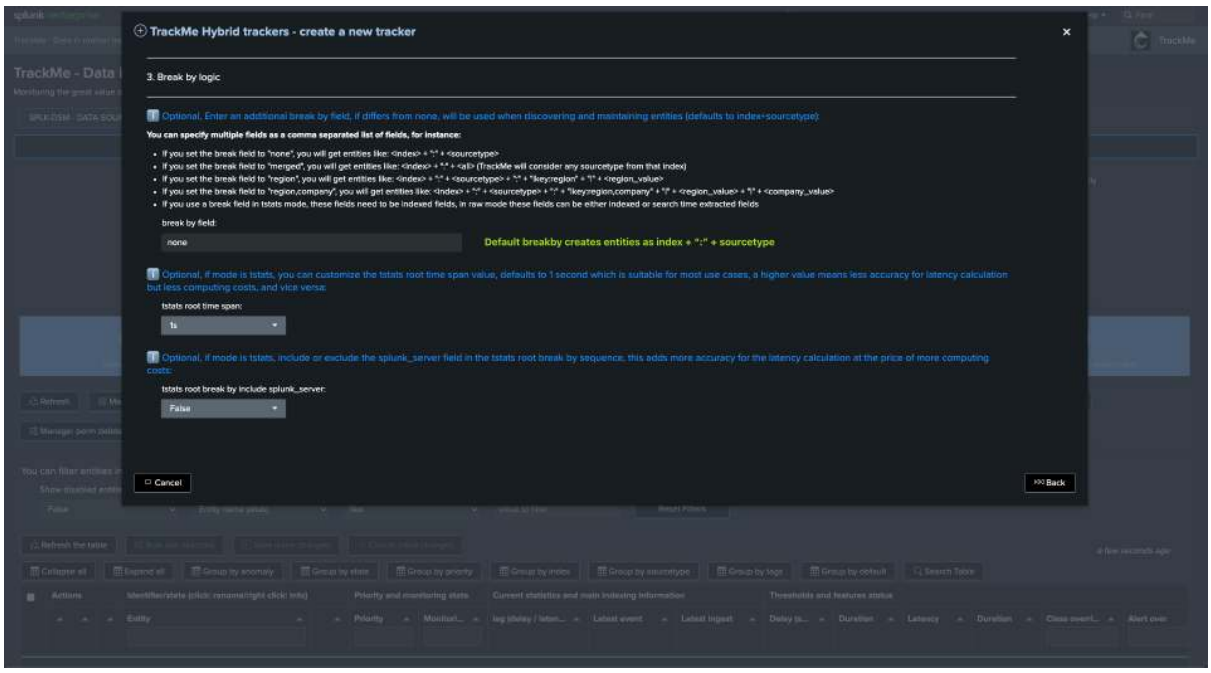
- When creating a Hybrid Tracker, TrackMe creates a dedicated Splunk macro which contains the scope of the tracker.
- You can therefore easily start with a restricted scope such as a list of index patterns, then modify this macro at any point in time to include additional contexts.
- To retrieve the macro name after the tracker is created, click on Manage Hybrid tracker from the home UI.



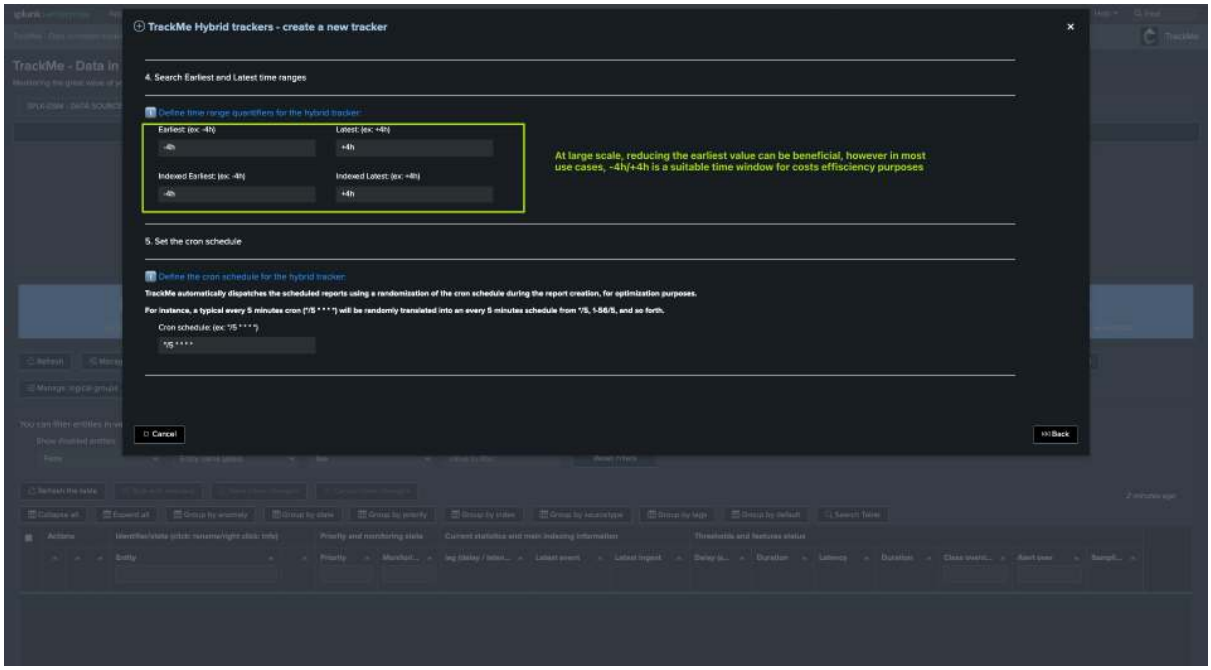


- Scroll down to the break by logic:
  - By default, the behavior is to create TrackMe entities as the combination of indexes and sourcetypes, therefore: `<index>:<sourcetype>`, in most cases, this is what you want.
  - An alternative, and depending on your requirements, is to use the `merge` mode, in this case TrackMe will create one entity per index: `<index>`.

  - Finally, you can also specify a custom list (comma-separated format) of fields as part of a custom break by, this is more sophisticated and is generally used for specific use cases such as leveraging custom indexed fields in large-scale environments.

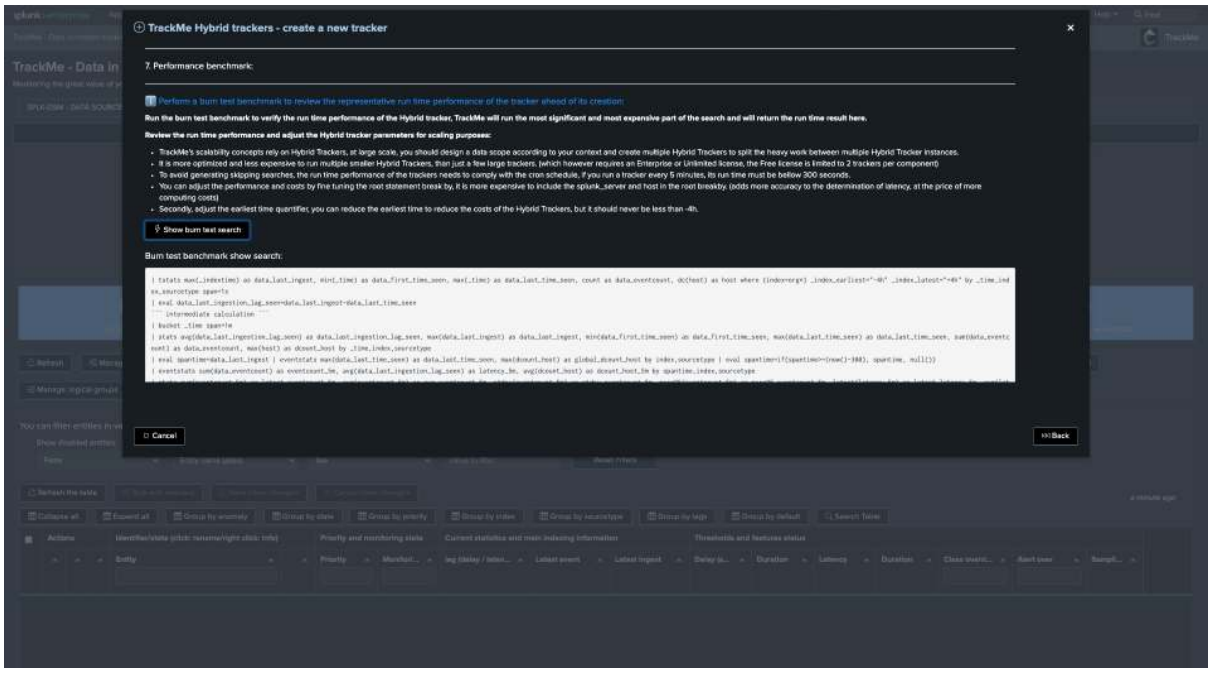


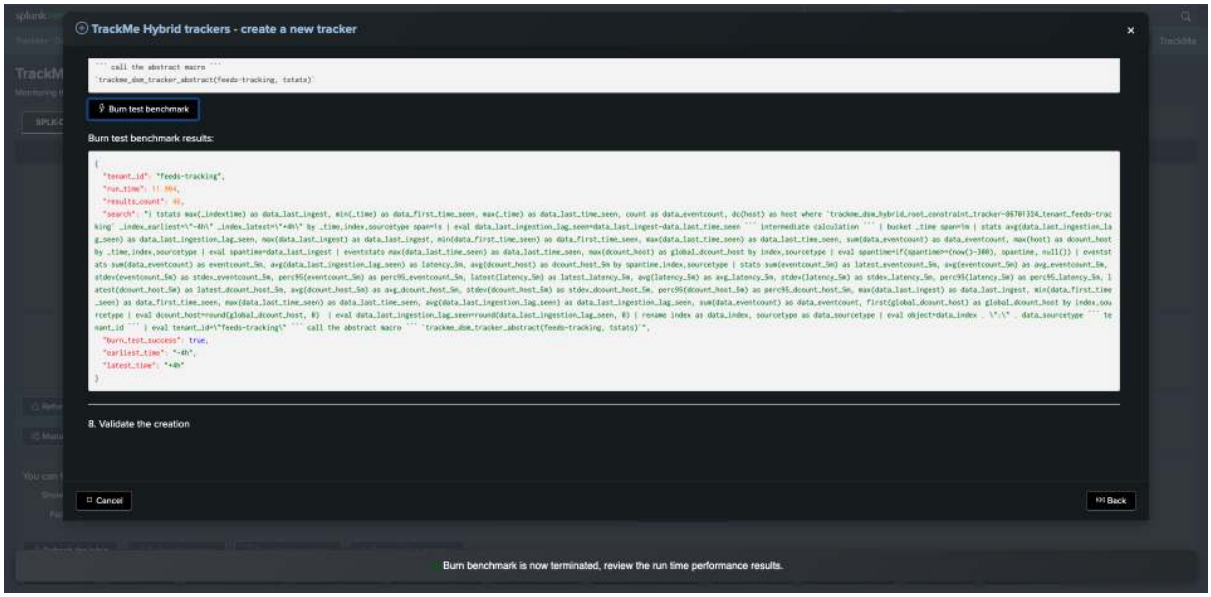
- Time window definition:
  - By default, TrackMe uses a -4h/+4h time range for the search logic, this is generally fine for most cases.
  - In some contexts such as very high-scale contexts, or limited capacity contexts, you can eventually reduce the earliest to reduce the associated costs and run time (most likely for splk-dhm than splk-dsm).



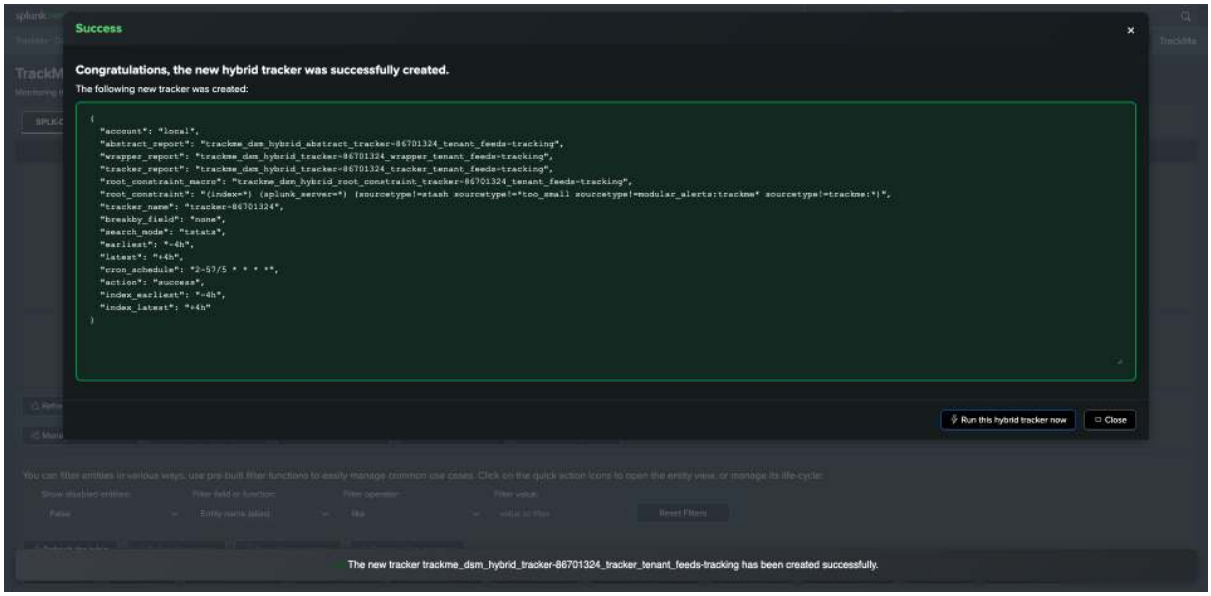
- Benchmarking the tracker:
  - TrackMe allows benchmarking the most expensive part of the search logic, this provides an accurate enough idea of how long the tracker will take to execute.

- The target is to execute the search every 5 minutes (while this is under your control, in most cases we want to get as close as possible to this frequency, especially for feeds tracking).
- This means that the search needs to be executed in less than 300 seconds to avoid generating skipping searches and missing executions.
- The scaling idea underneath is to create multiple trackers with a reduced scope, so they can be executed concurrently, and therefore reduce the execution time.

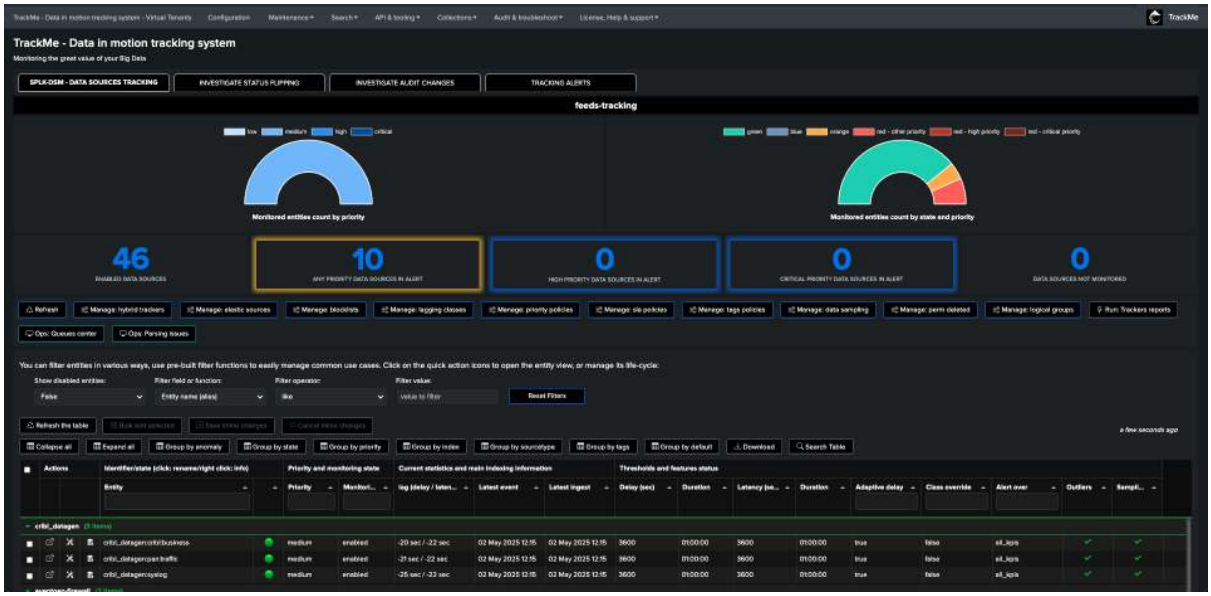




- Continue the process until the tracker is created.
- You can request its immediate execution:



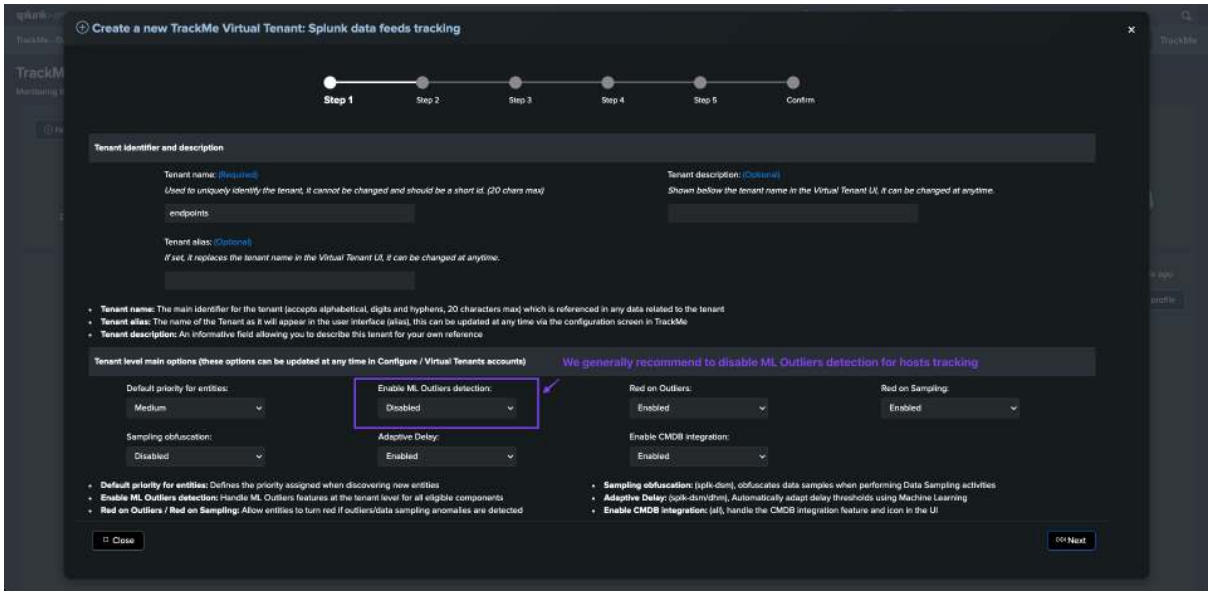
- After the tracker is executed, TrackMe will show up with entities created, and we can then continue our initial setup!

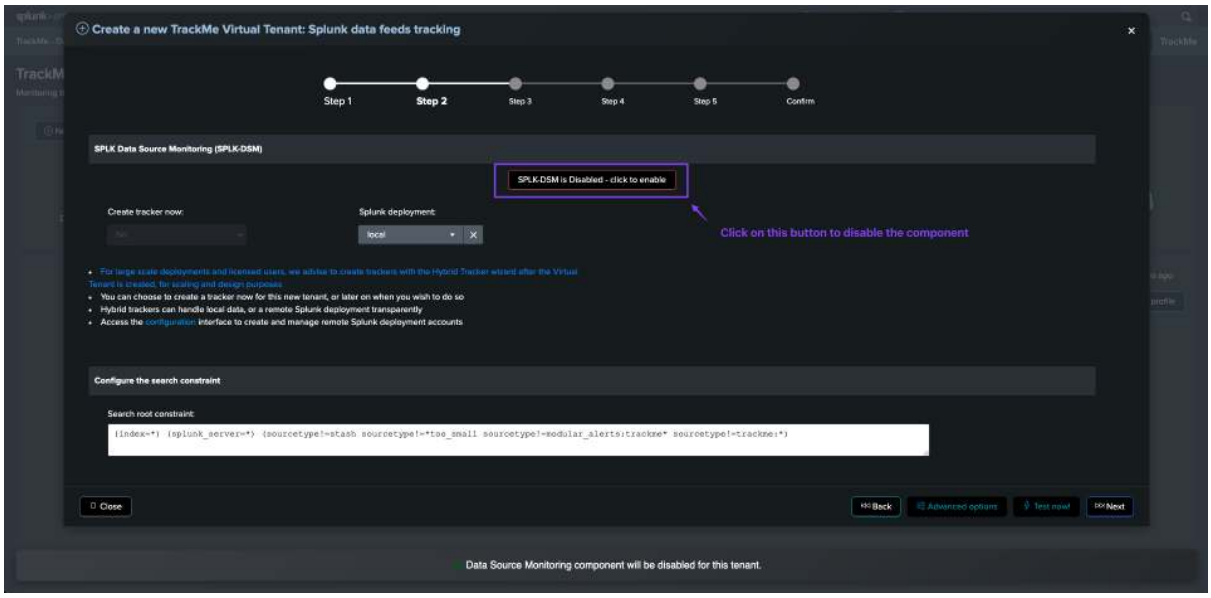


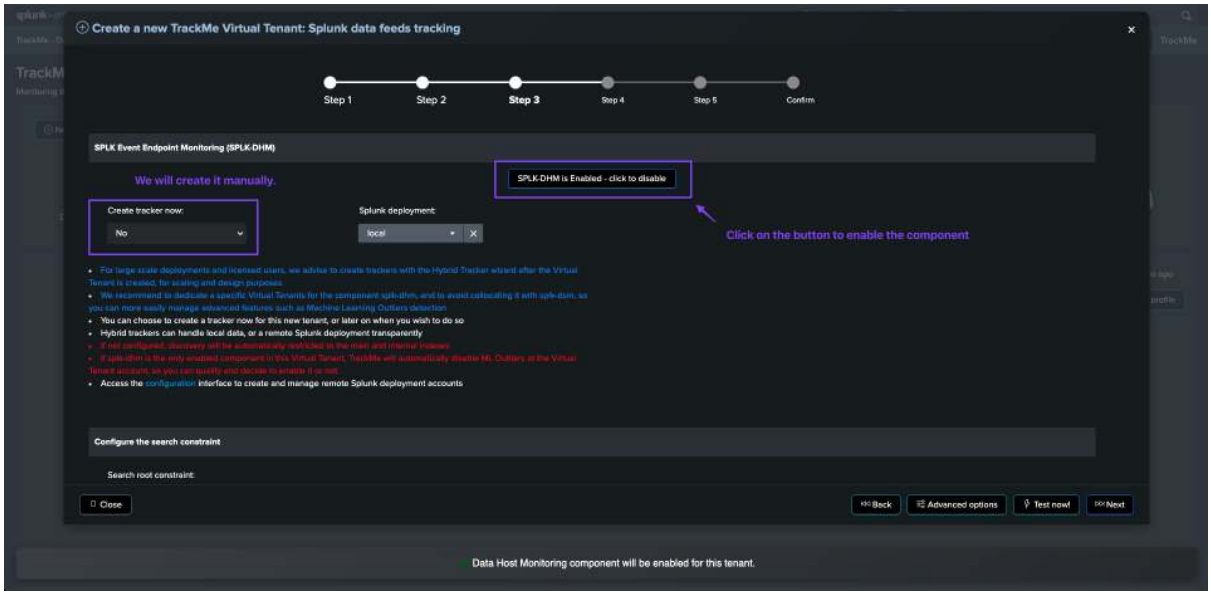
1.1.3 Step 2: Let's create a Virtual Tenant for data host sources

We will now create a Virtual Tenant to start tracking hosts, we recommend to setup hosts tracking in a dedicated tenant, so we can easily manage various features such as disabling ML Outliers:

- component: `splk-dhm`
- tenant name: `endpoints`

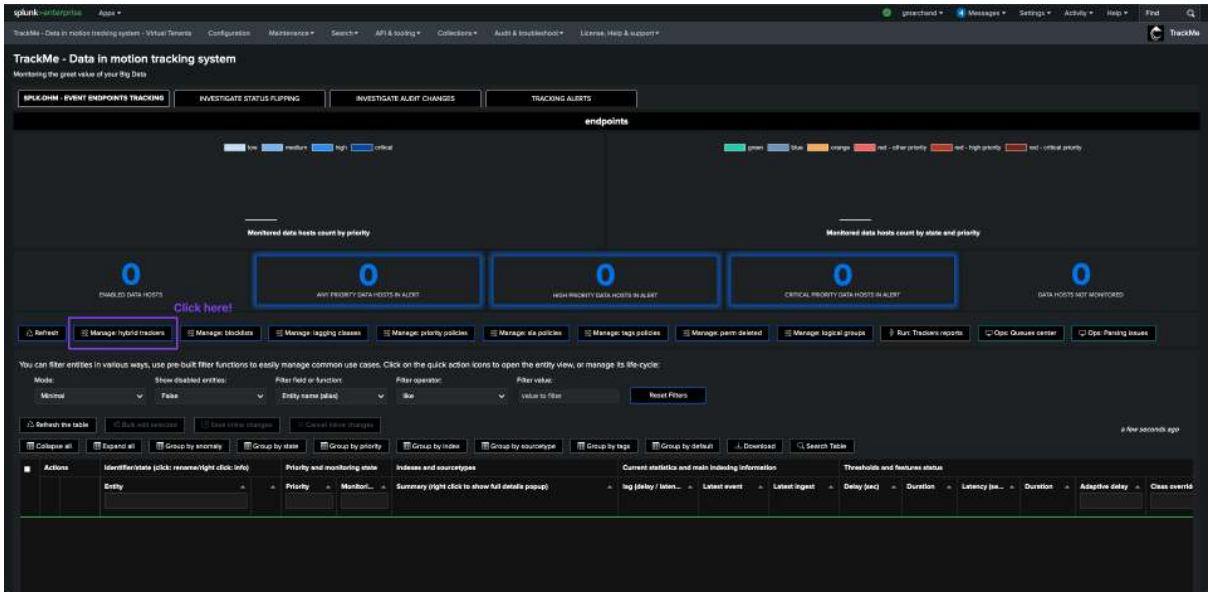




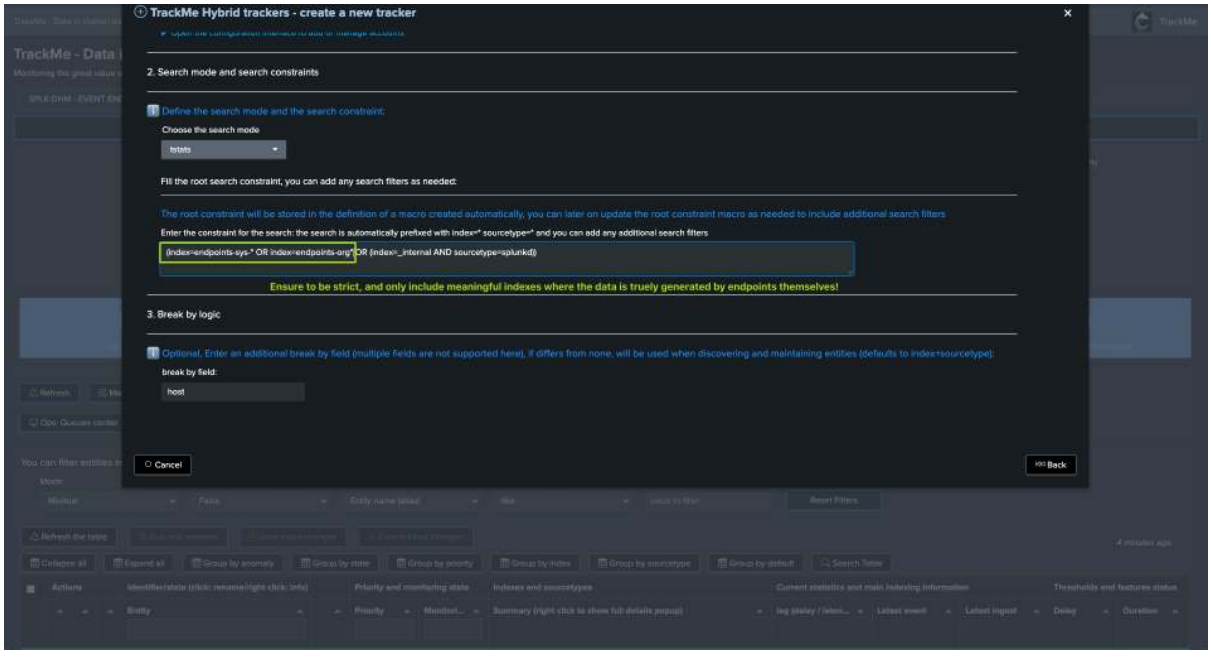


- Continue up to the creation of the Virtual Tenant, enter the new tenant and access the Hybrid

tracker wizard:



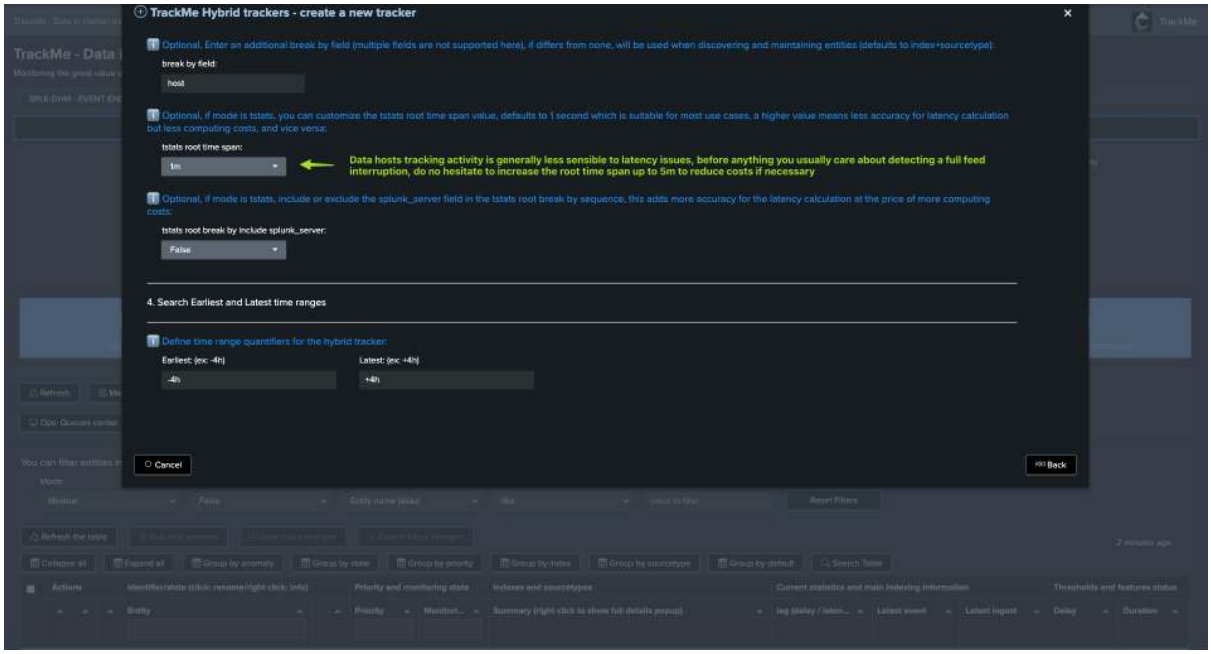
- We recommend being very strict with splk-dhm, notably we recommend restricting the scope as much as possible, and ensuring to allow only indexes that related to data generated by endpoints themselves.
- You totally can start with a certain scope, so you can first monitor the costs of the associated backends in TrackMe, then gradually increase the scope as needed.
- Finally you can also choose to include or not splunkd itself, if you wish to monitor the availability of Splunk UFs/UFs internals, which reflect a healthy connectivity and status of Splunk agents.



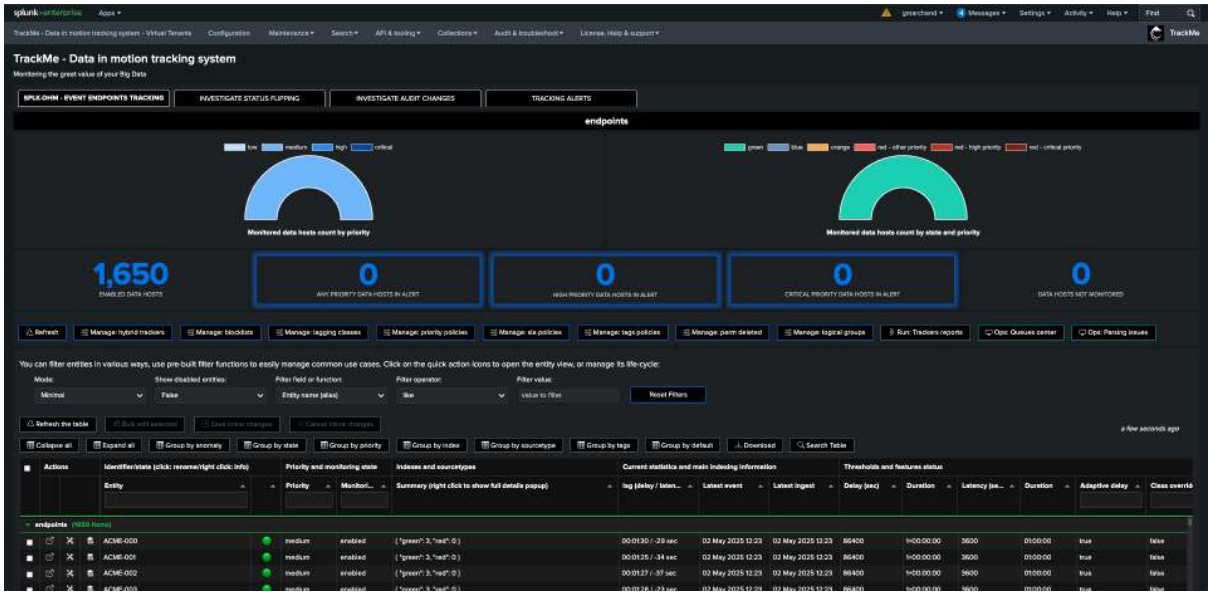
Create an hybrid tracker for splk-dhm

- You can increase the time span value up to 5m to reduce the associated computing costs, by increasing this value we reduce the strict accuracy of the latency calculation but this doesn't affect the delay calculation.
- Very often when tracking hosts, you essentially care about detecting hosts that have stopped emitting data for particular conditions, increasing this value reduces costs without altering this

capability at all.



- After the tracker is executed, TrackMe will show up with entities created, and we can then continue our initial setup!



1.1.4 Step 3: Alerting and notifications

Hint

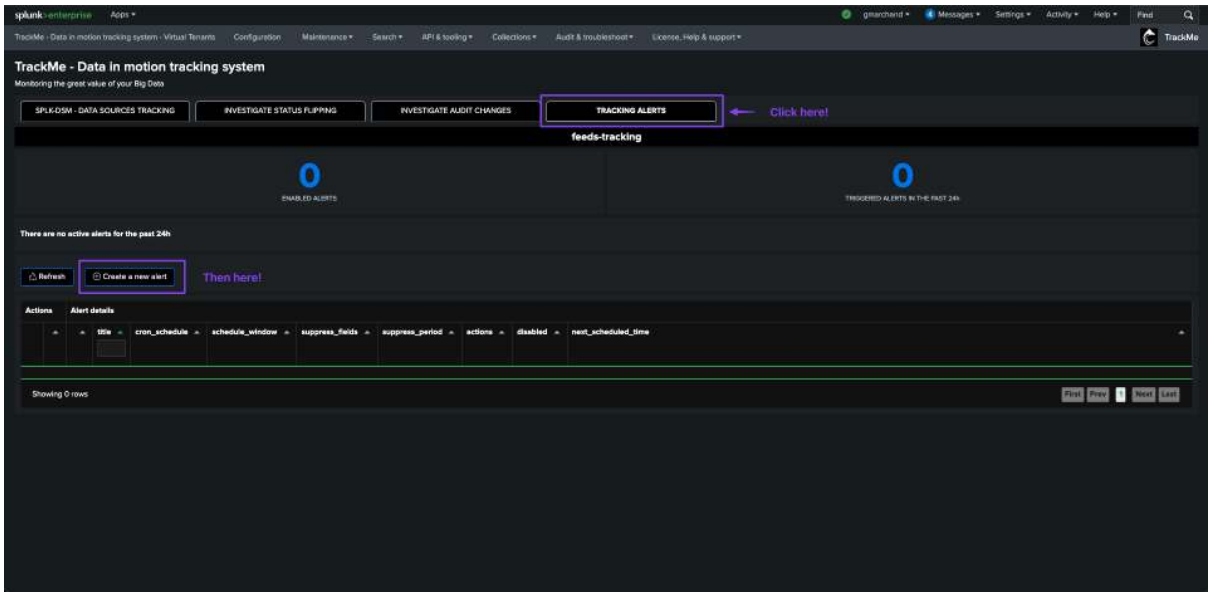
This quickstart was updated to use TrackMe's new StateFul Alerting features:

- Since TrackMe 2.1.11, a new concept of StateFul Alerting is available as the main alerting mechanism in TrackMe.
- These features provides a powerful, flexible and easy alerting capabilities as well as enhanced Email delivery notifications.
- For more information about this new concept, please refer to the *Alerting Architecture & Third-Party Integration* documentation.

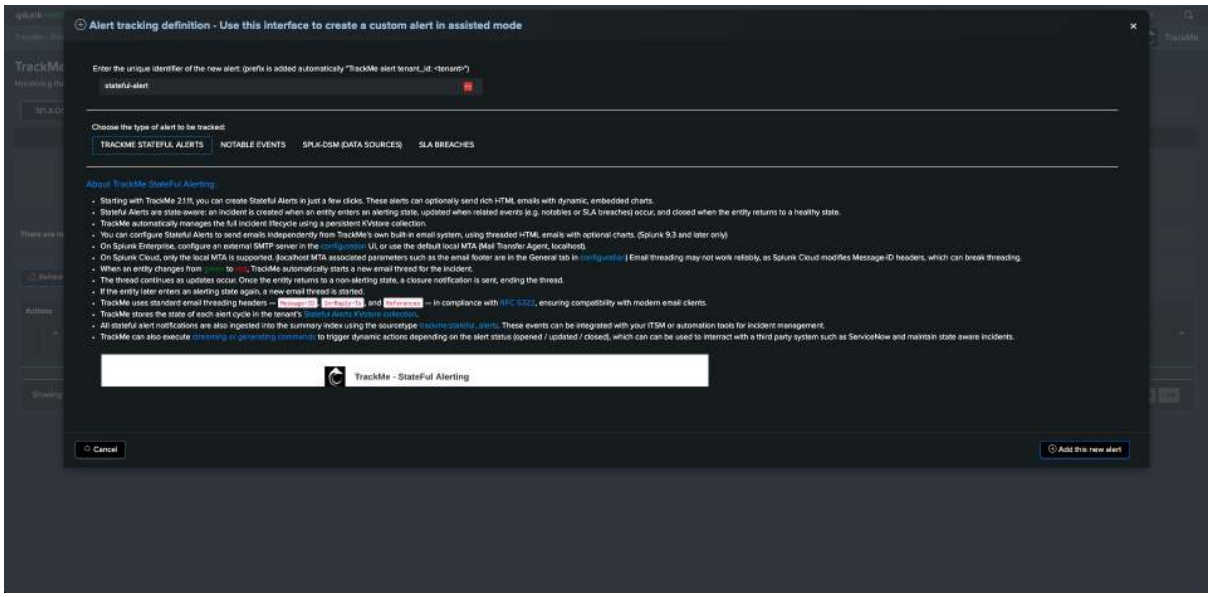
About Alerting recommendations:

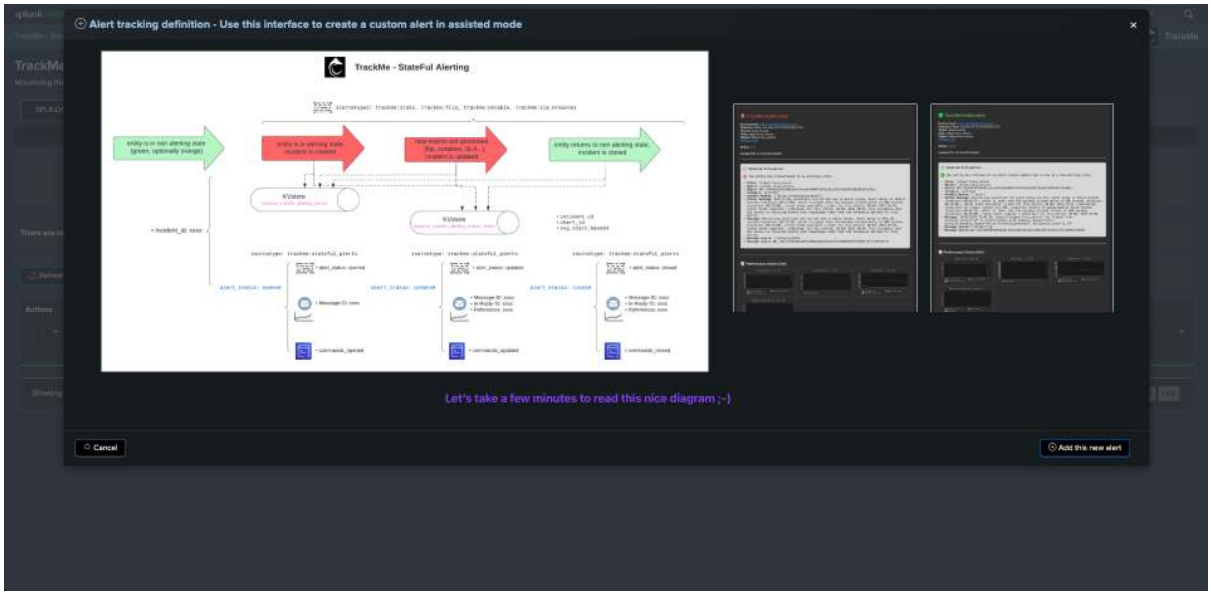
- In a Production context, a true challenge is the massive number of entities, from hundreds if not not many thousdands of sources, hosts and so forth, it can be overwhelming to manage.
- We recommend a selective approach which used in conjonction with TrackMe and Splunk capabilities allow you to tackle these challenges.
- What we recommend is notably to leverage the `priority` concepts, and alert only on properly qualified entities leveraging `high` and `critical` priorities.
- This approach allows you to focus on the most important entities, and to avoid alerting fatigue by progressively qualifying your alerting logic and maturity in TrackMe over time.

Access the Alerts creation wizard:

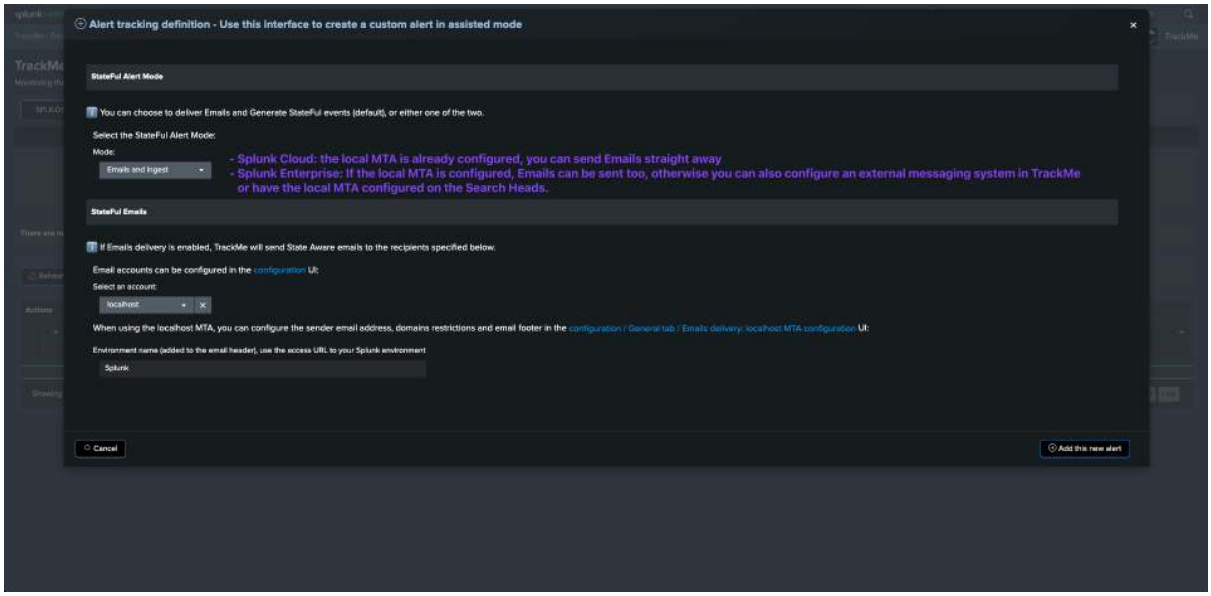


Welcoming screen for the alerts creation wizard:

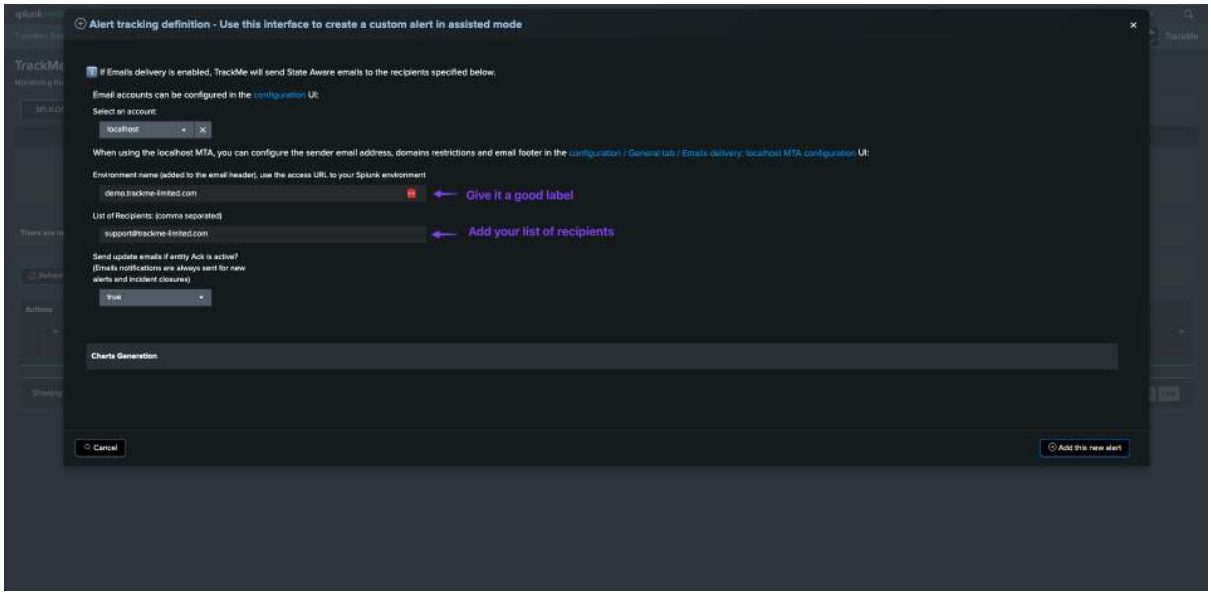




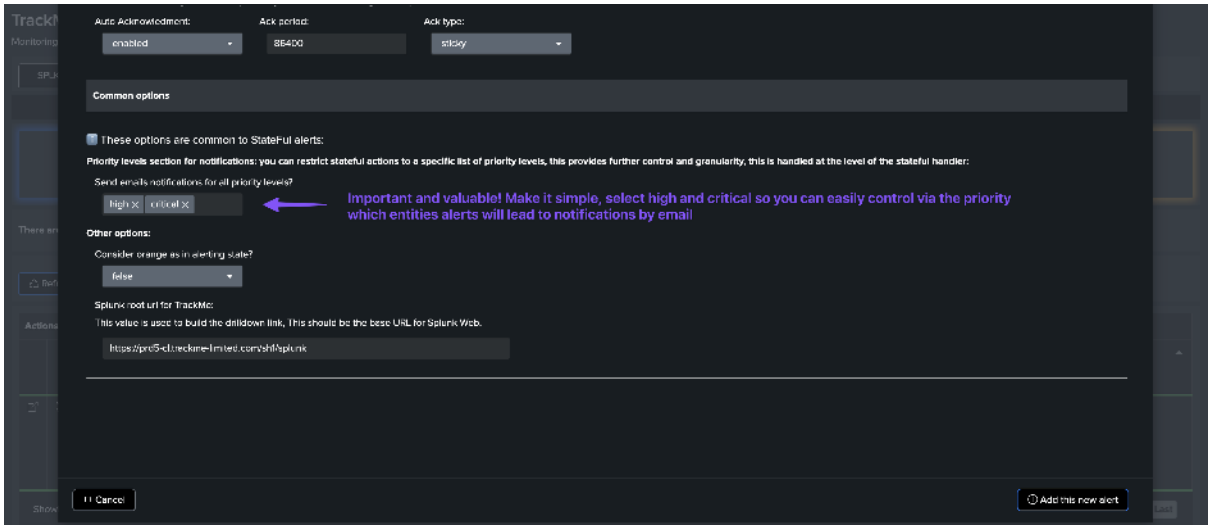
Choose the alerting mode, the defaults generates events and sends state aware Email notifications:



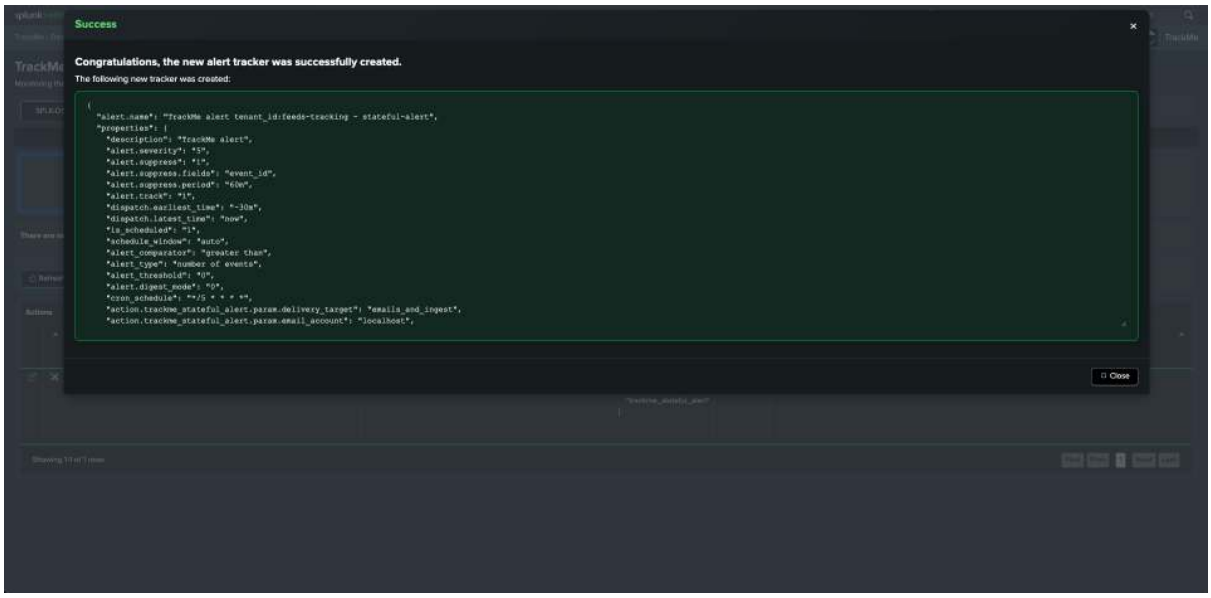
If sending Emails, configure Emails related settings:



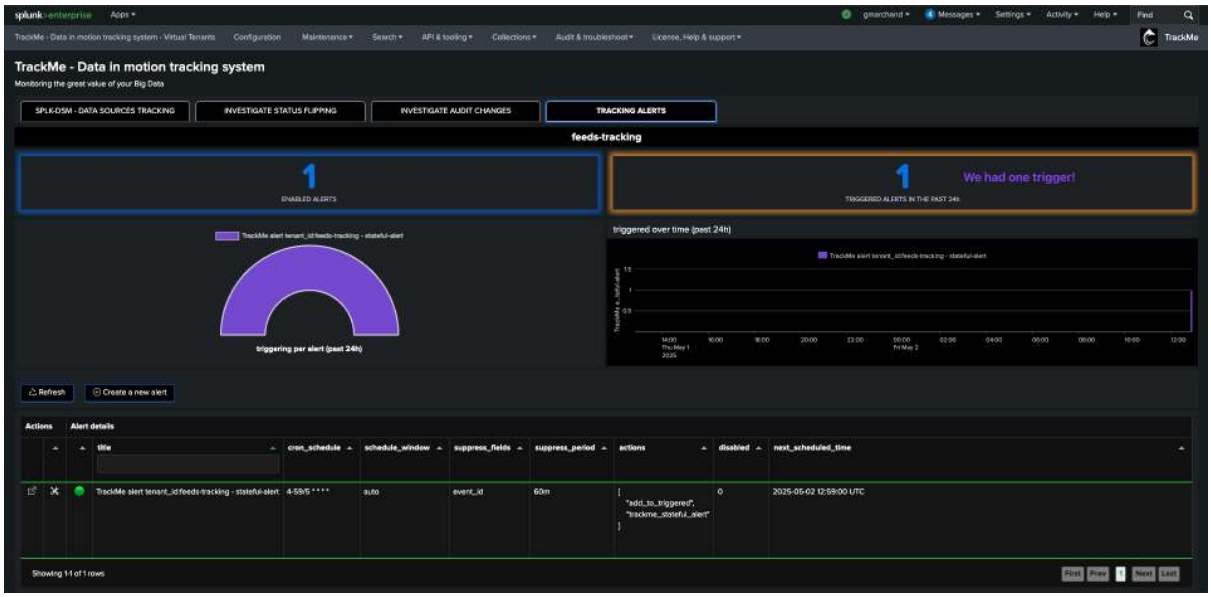
Define the proiority levels selection for email notifications:



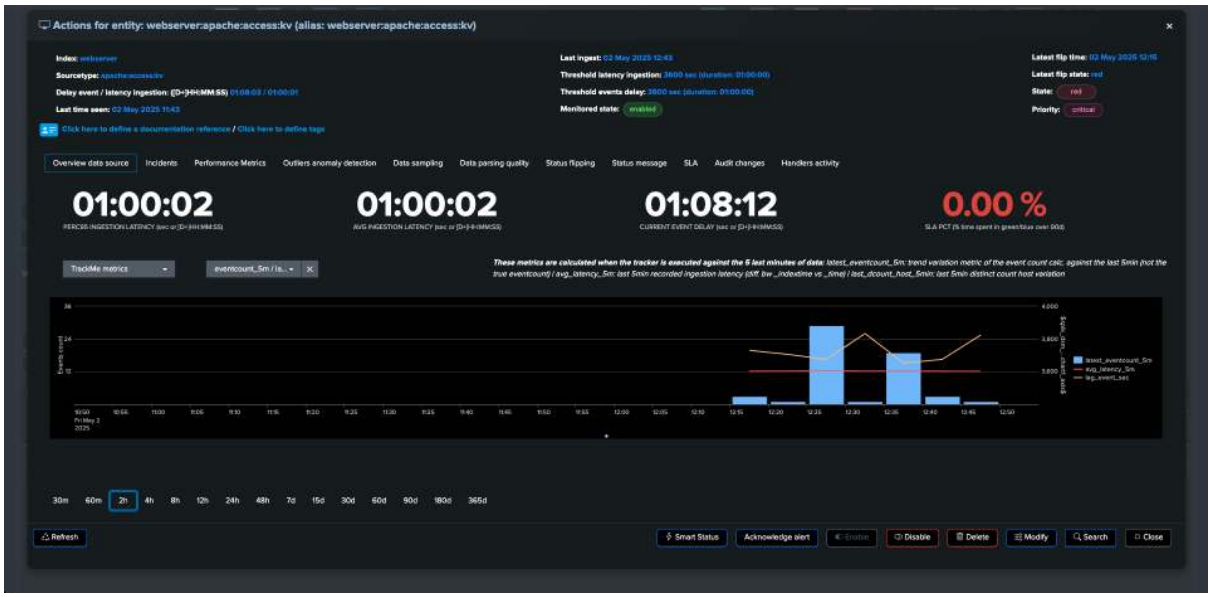
Validate the alerting creation:



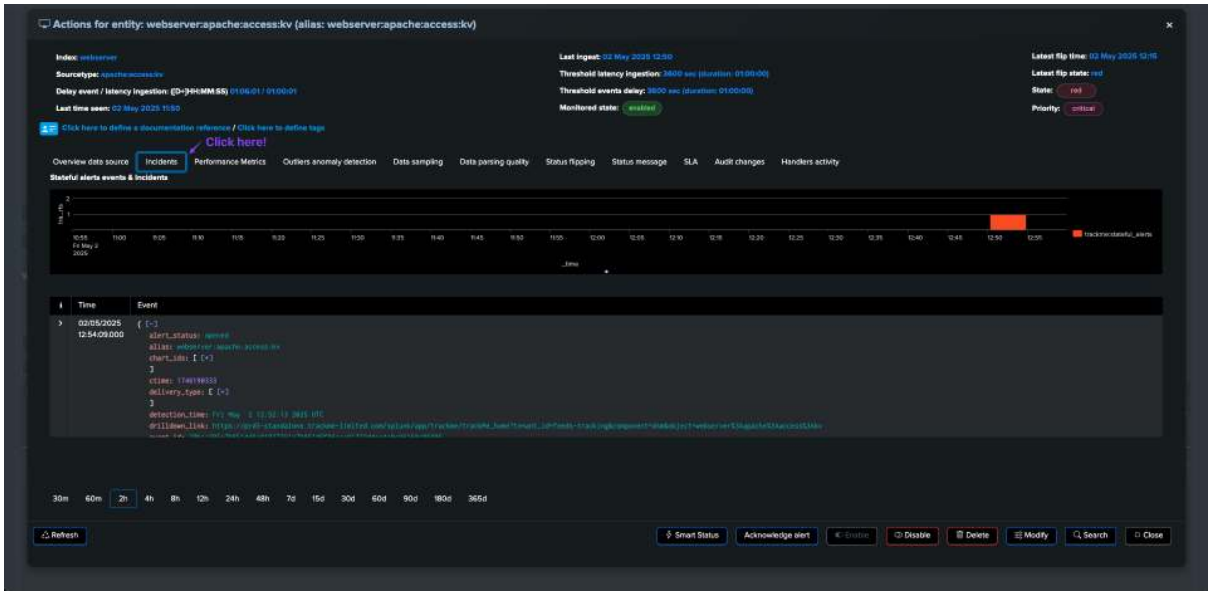
We are good to go now, if things are working as expected, any entity matching priority high or critical and currently in anomaly state, will trigger an alert!



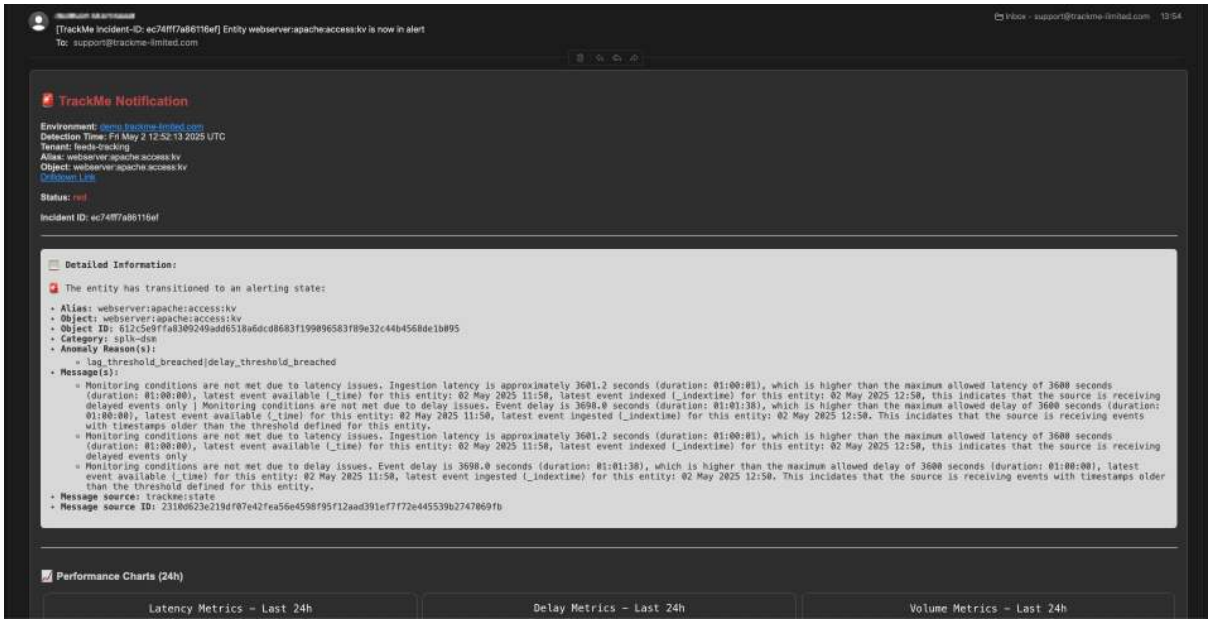
Let's access this entity in the UI:

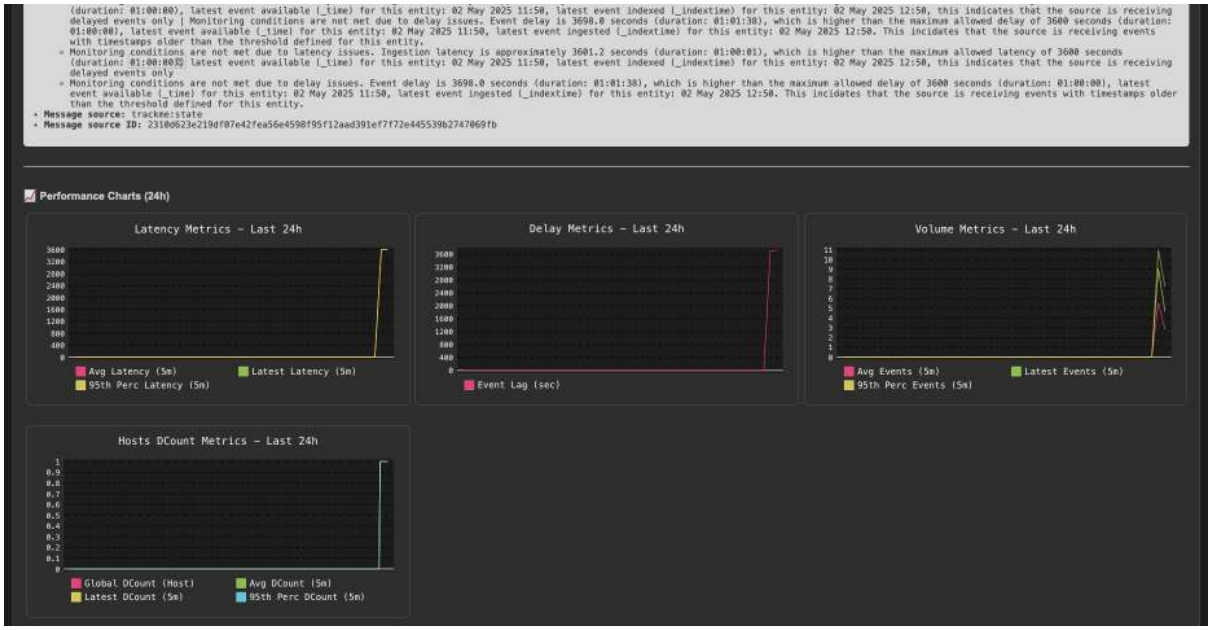


Access to the Incidents tab, and we can indeed see our incident created!



And if we did right, we also have an awesome Email notification!





Adapt to your preferences, repeat as needed, TrackMe is now Production ready!

CHAPTER

TWO

USE CASES DEMO:

2.1 Use Case Demo: This is All About Feeds

Use Case Demo: This is All About Feeds

- This white paper provides an overview of Splunk feeds tracking with TrackMe.
- It covers the key concepts and features around tracking data sources and hosts availability and indexing performance.
- Feeds tracking is a key feature of TrackMe, representing the root use case of TrackMe, and in many cases, the first reason why customers use TrackMe.
- This use case demo documentation explains and demonstrates the key features of TrackMe feeds tracking.

2.1.1 High Level Overview

At the source of TrackMe concepts stands the idea of tracking data sources and hosts availability and indexing performance.

The key features can be summarized as follows:

- `Discovery`: Discover and organize data sources automatically based on meaningful key metadata.
- `Persistence`: Persist entities and their key information over iterations, using various features from Splunk KVstore collections to events and metrics indexing.
- `Availability`: Track entities availability over time based on `delay` thresholds, to detect when a given entity has stopped forwarding events to Splunk.
- `Performance`: Track entities indexing performance over time based on `latency` thresholds, to detect when a given entity is suffering from indexing performance issues.
- `Quality`: Provide out-of-the-box and automated capabilities to detect meaningful lack of quality in the entities data.
- `Metrics`: Generate and store metrics in a high-performance way, using Splunk metric store capabilities.
- `Knowledge`: Accumulate knowledge over time to improve investigations and decision-making.
- `Machine Learning`: Implement meaningful and automated Machine Learning models to detect volume variations and potential anomalies on entities.
- `Flexibility`: Provide flexible and powerful capabilities to cover from basic tracking use cases to the most complex ones.
- `Alerting`: Provide out-of-the-box powerful and advanced alerting capabilities, from state-aware alerting to rich email deliveries, state-aware actions, and more.

- `Scalability`: Scale at any level, from few terabytes environments to hundreds of terabytes environments.
- `User Experience`: Provide a rich, easy, and fast user experience, with user interfaces and APIs that provide fast answers and browsing capabilities with TrackMe entities.

This use demo documentation covers four common and most valuable feeds related use cases:

- `Feed Interruption Detection`
- `Feed Indexing Performance Issues Detection`
- `Feed Volume Variation Detection`
- `Feed Quality Issues Detection`
- `Feed Hosts Distinct Count Anomaly Detection`

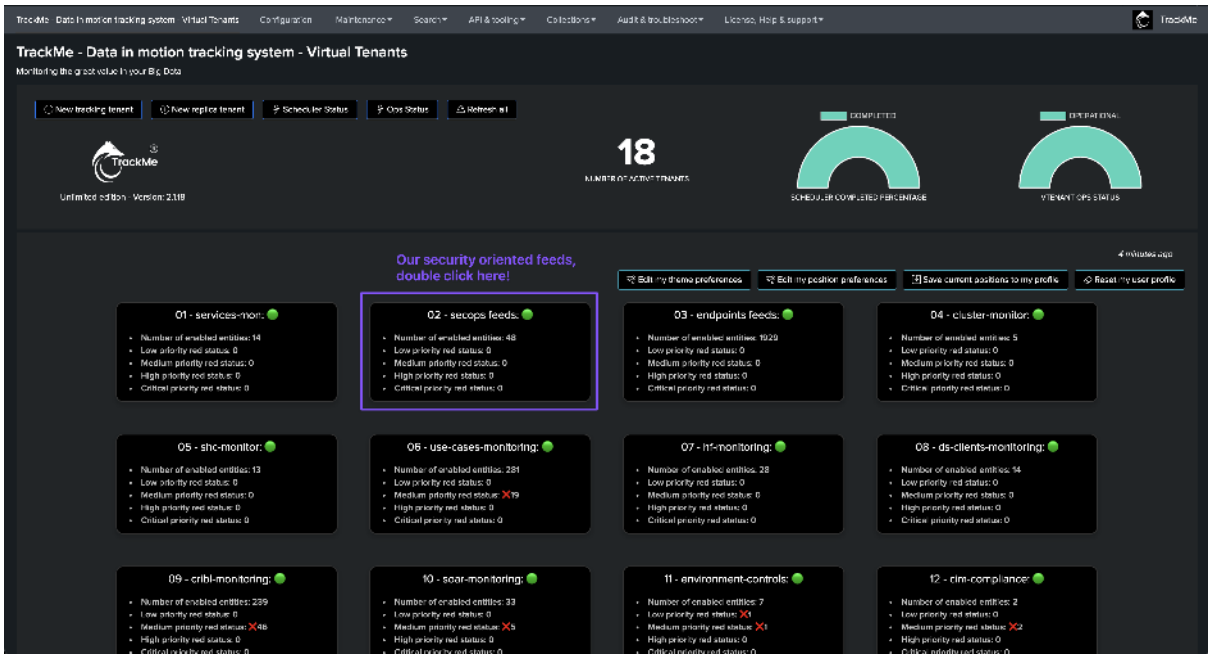
Many more use cases can be implemented with TrackMe, and through our Flex Trackers components, you can create custom monitoring solutions tailored to your specific needs.

2.1.2 Feeds Tracking with TrackMe Data Sources Monitoring (splk-dsm)

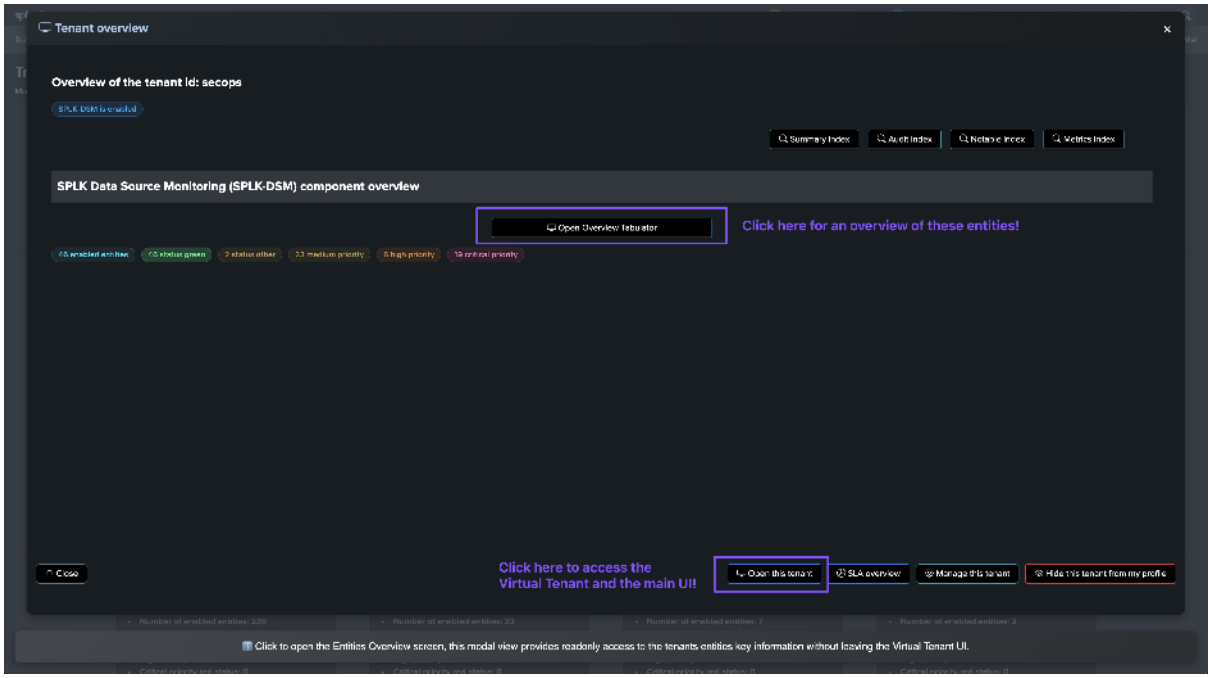
The Data Sources Monitoring TrackMe component (splk-dsm) is the key component to track data sources availability and indexing performance.

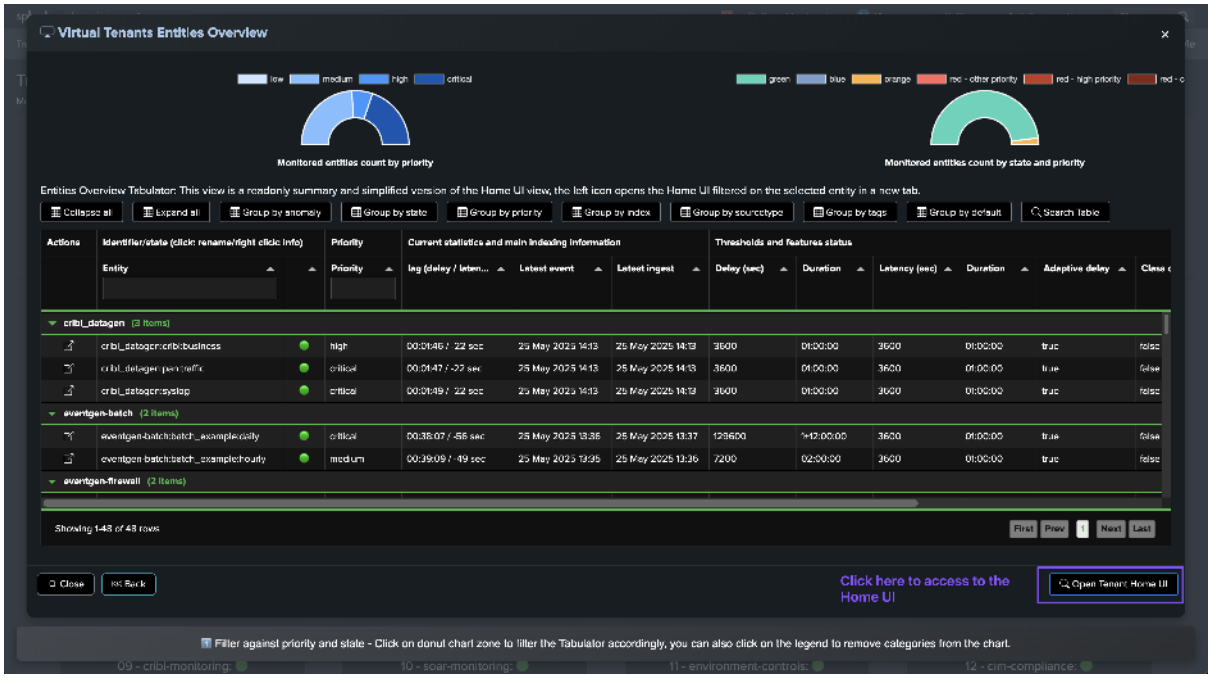
In its simplest form, it is tracking feeds, creating and organising entities based on `indexes` and `sourcetypes` metadata.

Virtual Tenants allow customers to segment use cases based on their needs, from geographically oriented use cases to use cases based on data sensitivity or perimeters:

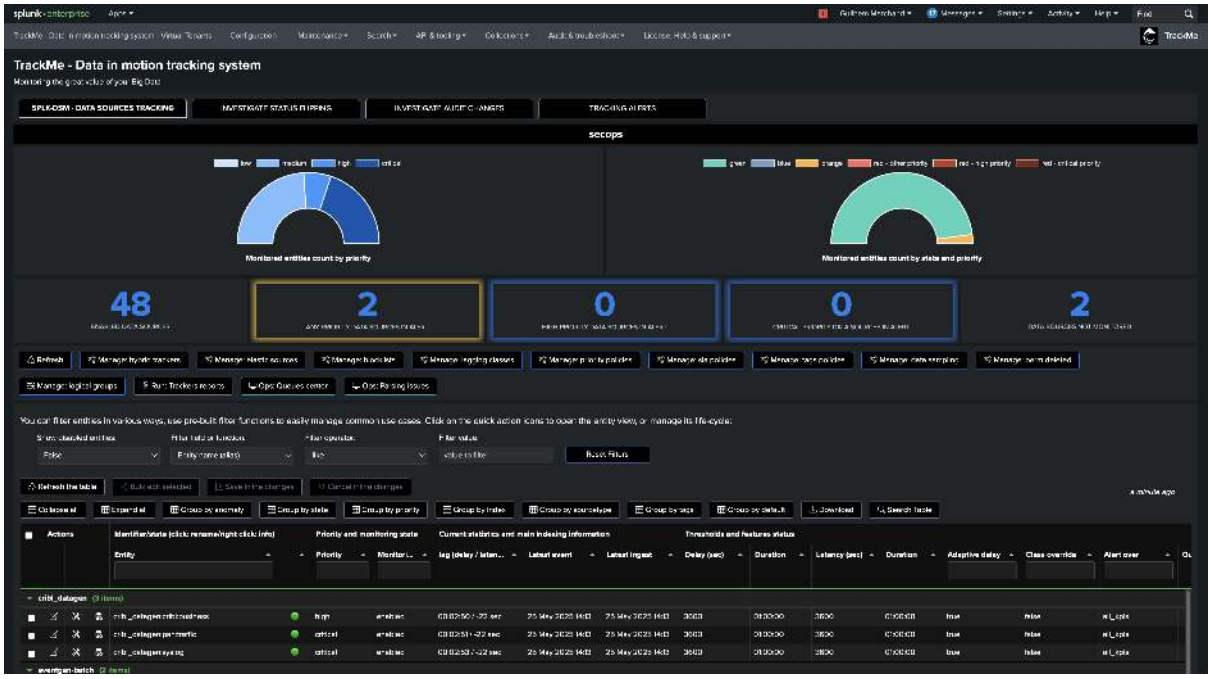


Virtual Tenant Entities Overview:

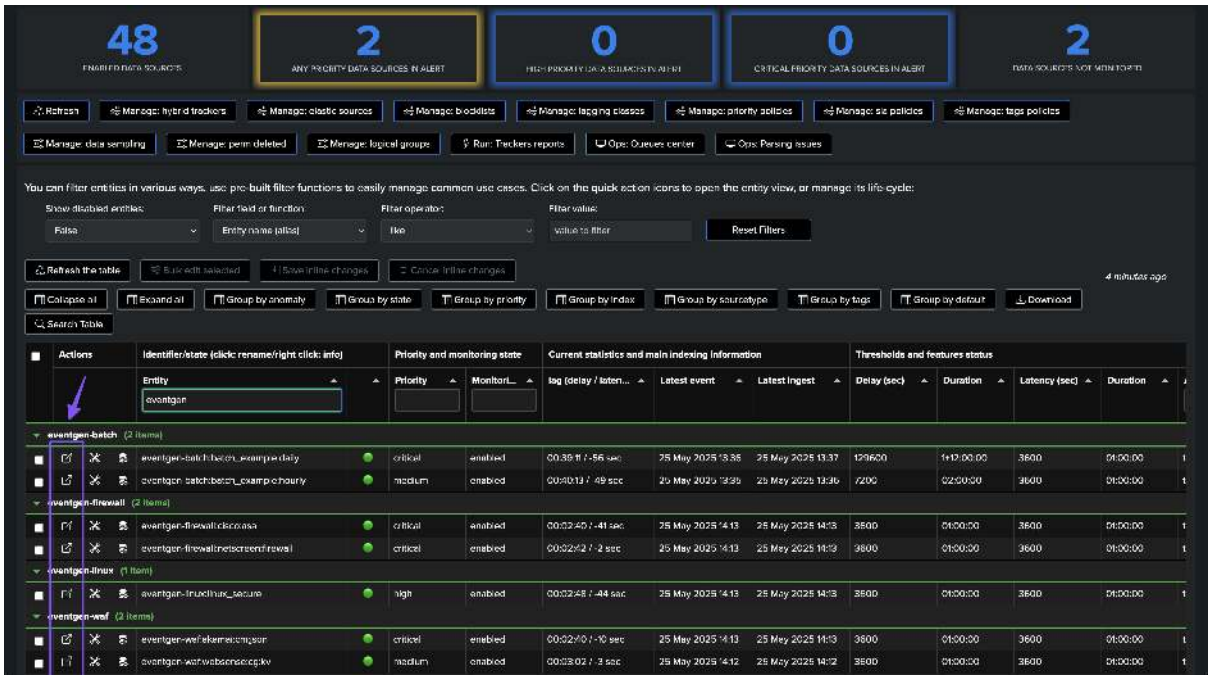




The Home UI is the main user interface to manage your TrackMe entities:



To Access to the status of a given entity, representing a Splunk feed (by default through its index and sourcetype metadata), click on the left hande side icon:

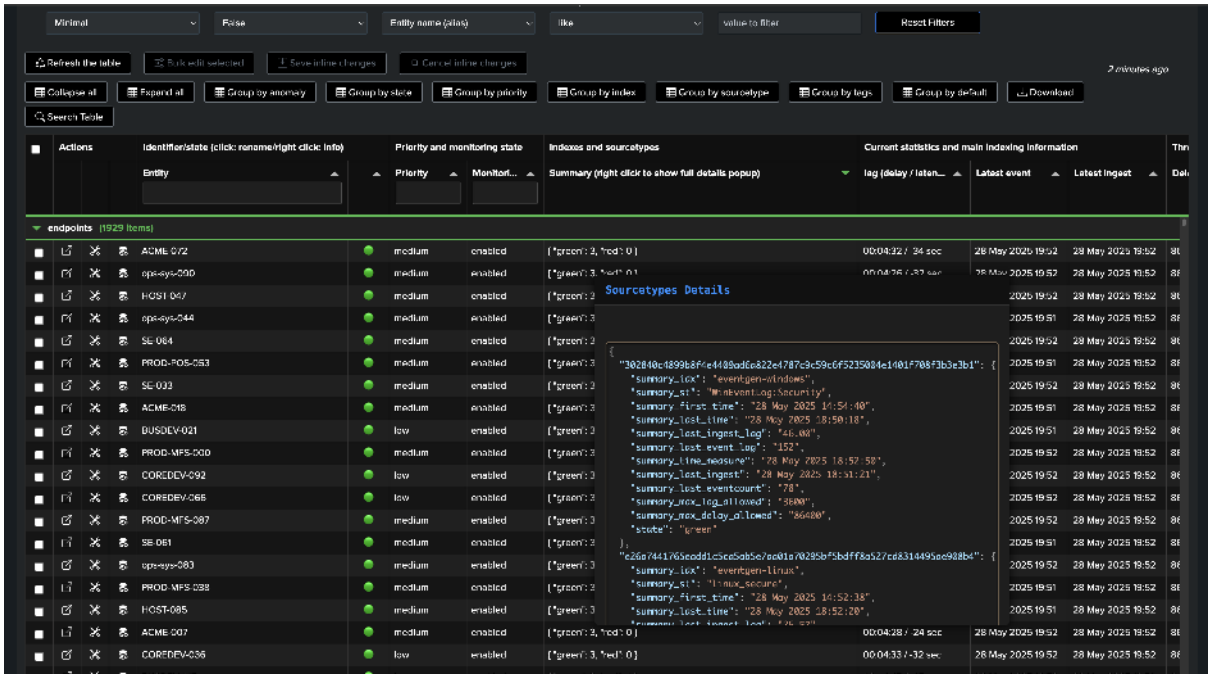


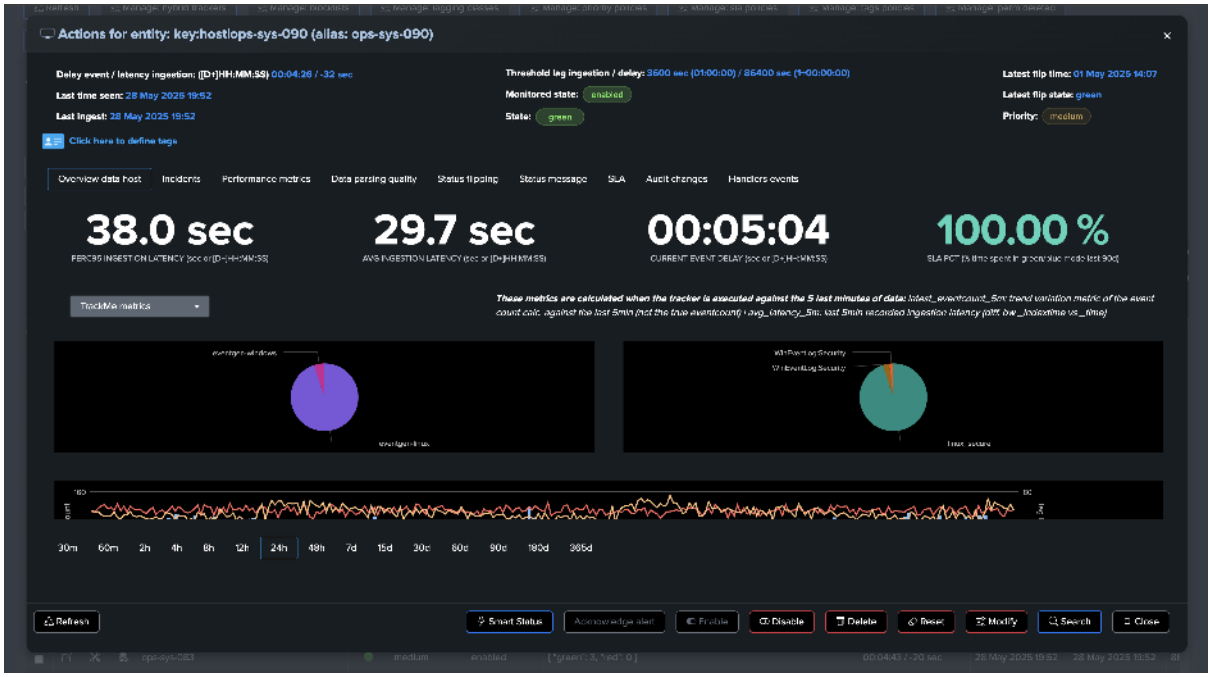
The entity details page provides a rich set of information about the entity, including its status, availability, indexing performance, metrics, knowledge, machine learning, flexibility, alerting, scalability, user experience, and more:

2.1.3 Hosts Tracking with TrackMe Hosts Sources Monitoring (splk-dhm)

The second main components for feeds tracking is the Hosts Sources Monitoring TrackMe component (splk-dhm).

It acts similarly to the Data Sources Monitoring TrackMe component (splk-dsm), at the difference that it is tracking data from host events producer perspective, which means sourcetypes associates with a given host depending on the tracker scope of data:





2.1.4 Primary Feeds Key Performance Indicators (KPIs)

TrackMe is all about Key Performance Indicators (KPIs) and high performing metrics, this means that whatever the use case, we turn data into metrics then act against these.

In all components, you will find a tab called `Performance Metrics` which lists available metrics for the given entity.

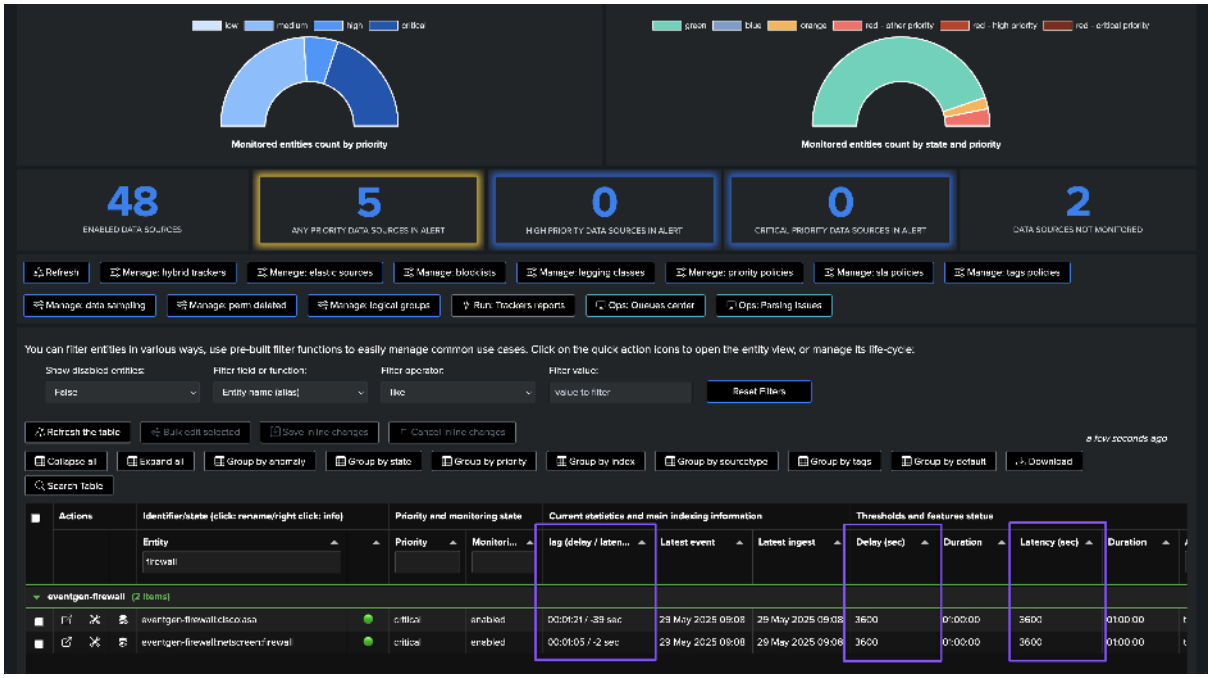
For Data Sources and Hosts, the primary KPIs are:

- `Latency`: The performance at the indexing level from the Splunk perspective, as the difference between the indexed time and the event time.
- `Delay`: How late the data is, calculated as the difference between now when the tracker is executed, and the very last event time.

Latency metrics:

Delay metric:

These two primary KPIs are those linked to the primary thresholds used to detect potential interruptions or performance issues for sources:

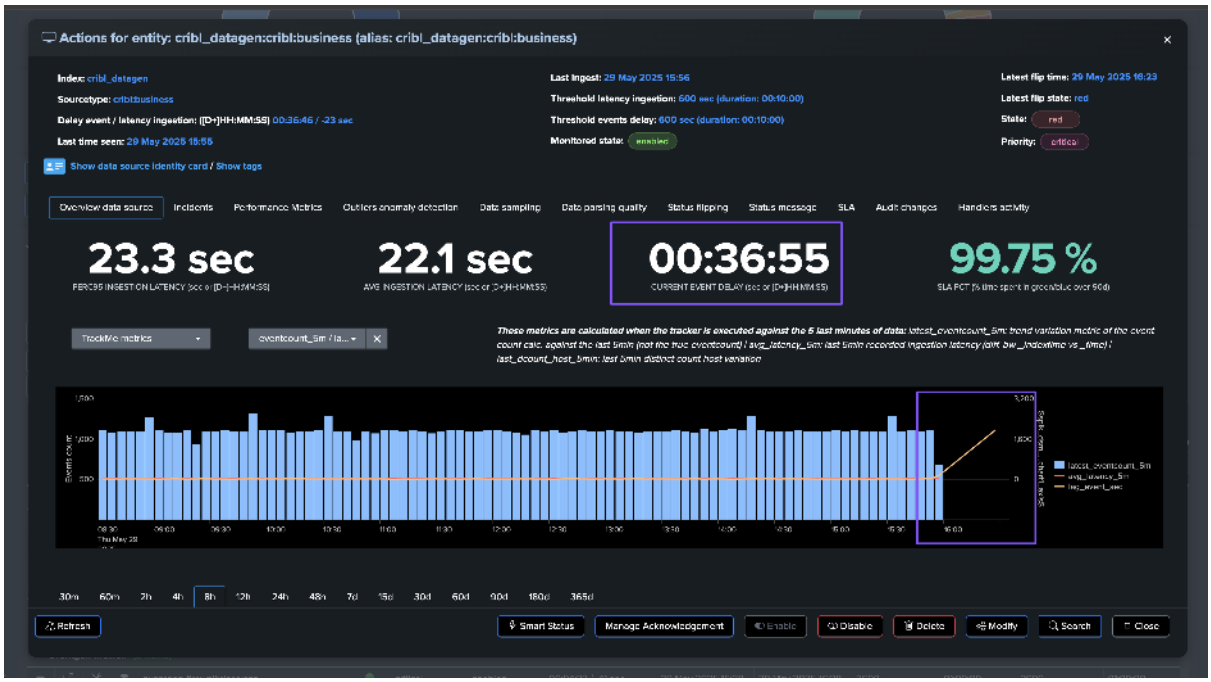


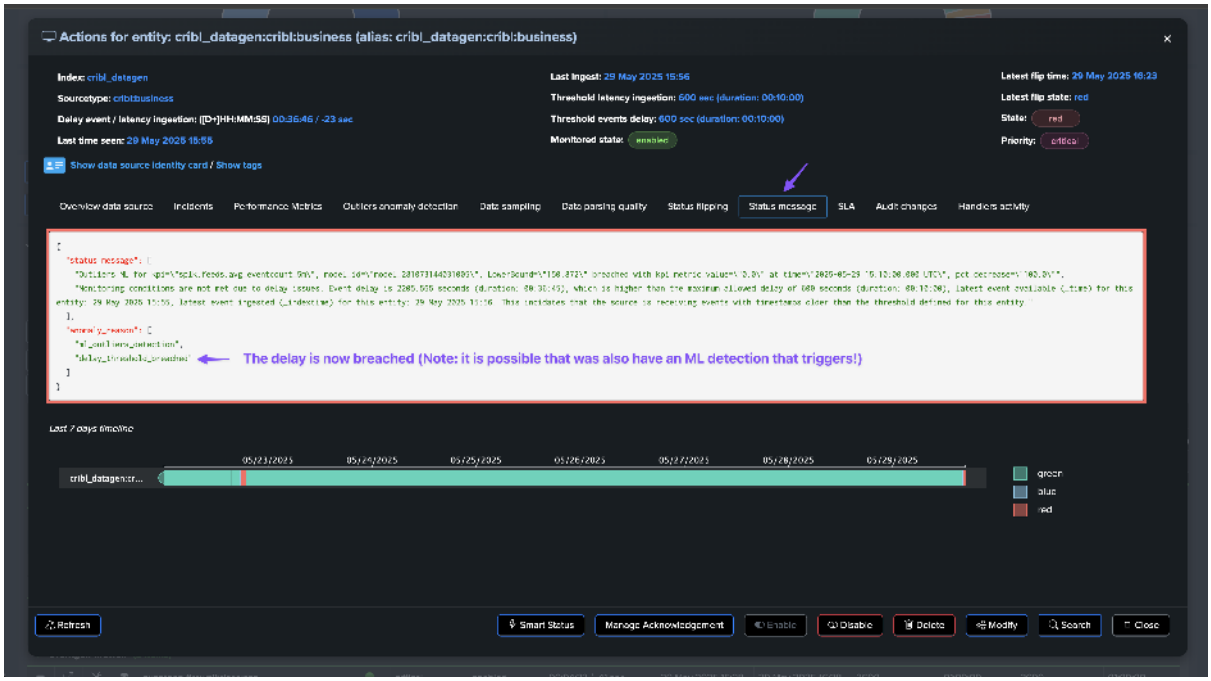
2.1.5 Use Case: Feed Interruption Detection

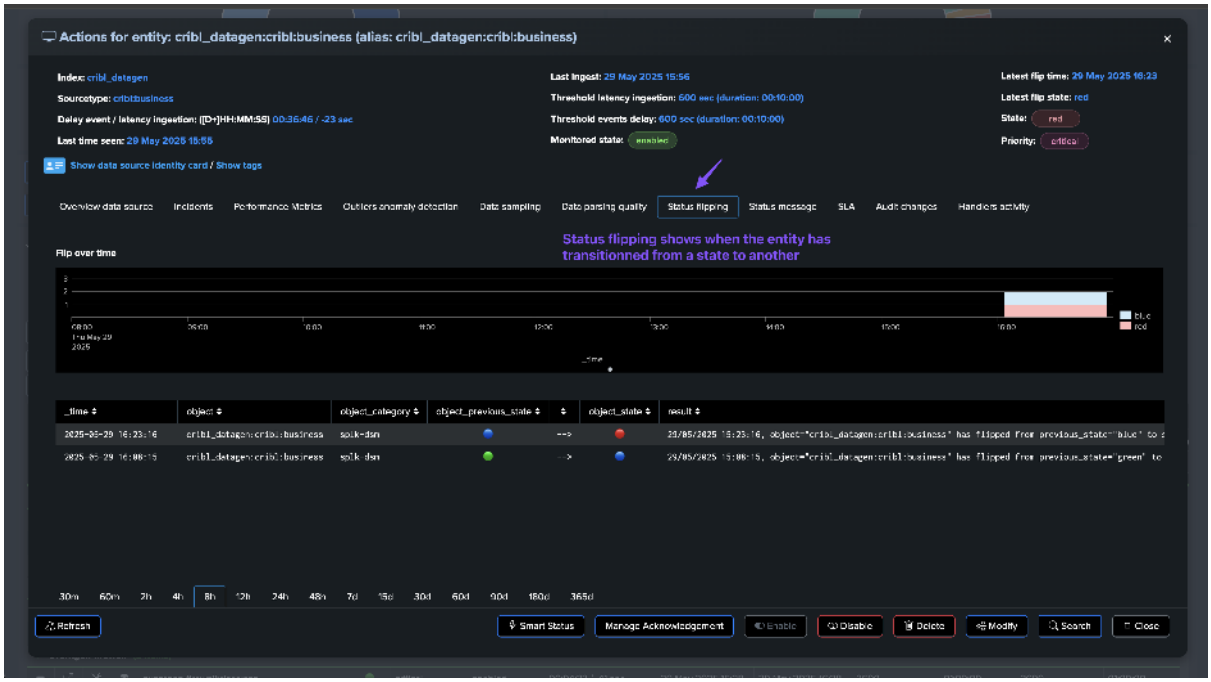
A first essential use case you want to cover when it comes to feeds availability monitoring is the detection of feed interruptions. In many use cases, you might expect the data to be flowing into Splunk in real-time or semi-real-time, though this isn't necessarily the case, and you need to have the capabilities to handle this on a case-by-case basis.

To cover this use case, TrackMe tracks the very latest event available on a per-entity basis, that is the event from the timestamp point of view (field _time). We calculate the difference between now, when the tracker is executed, and the last event, which gives you an indicator in seconds.

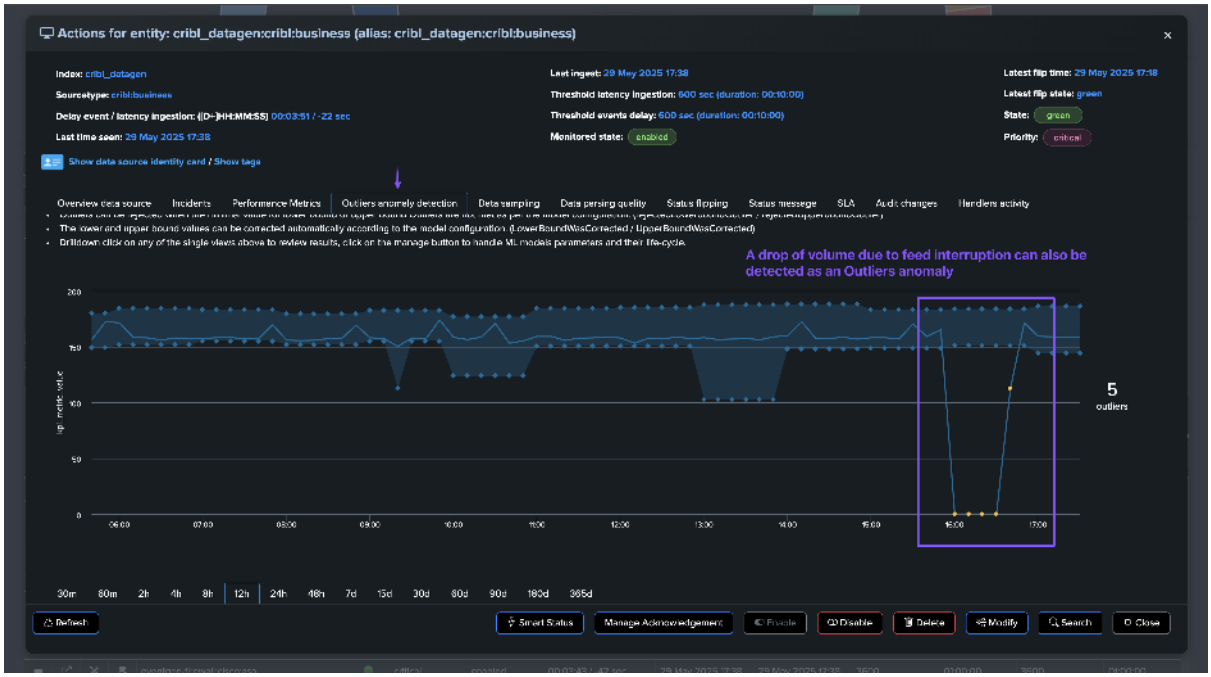
If this indicator goes beyond the established delay threshold, this is considered as a condition for anomaly, with anomaly_reason="delay_threshold_breached":



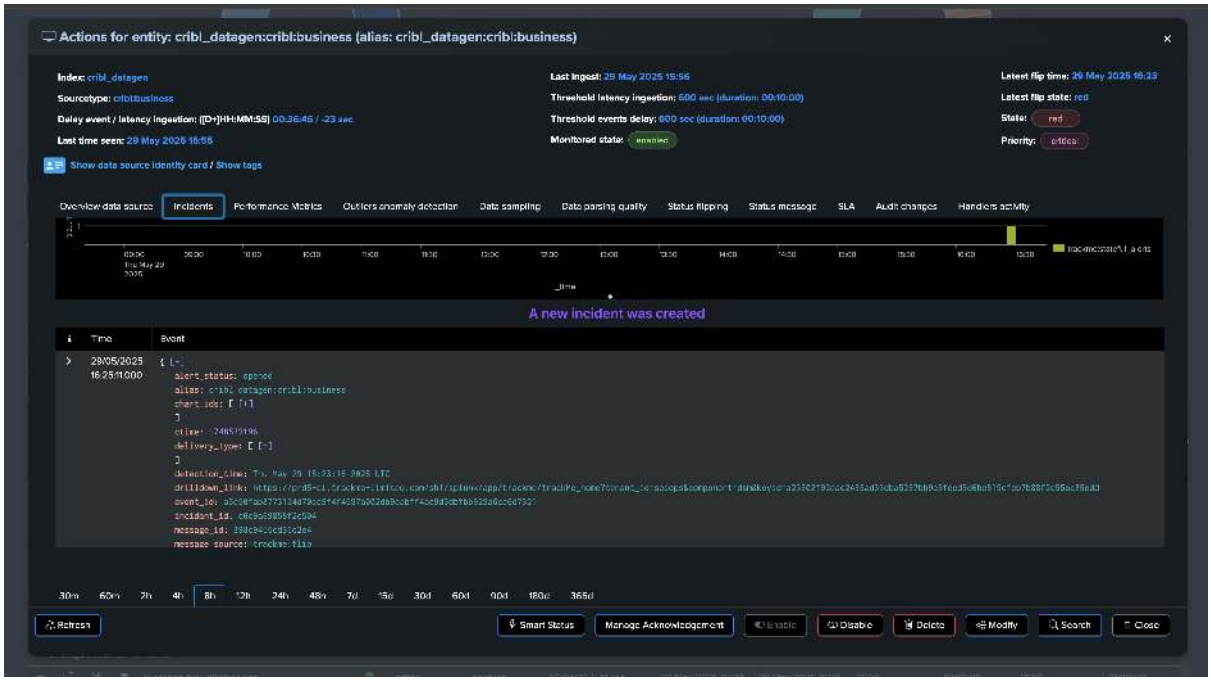

Actions for entity: cribl_datagen:cribl:business (alias: cribl_datagen:cribl:business)
Status message
The delay is now breached (Note: it is possible that was also have an ML detection that triggers!)


Actions for entity: cribl_datagen:cribl:business (alias: cribl_datagen:cribl:business)
Status flipping
Status flipping shows when the entity has transitionned from a state to another
Flip over time

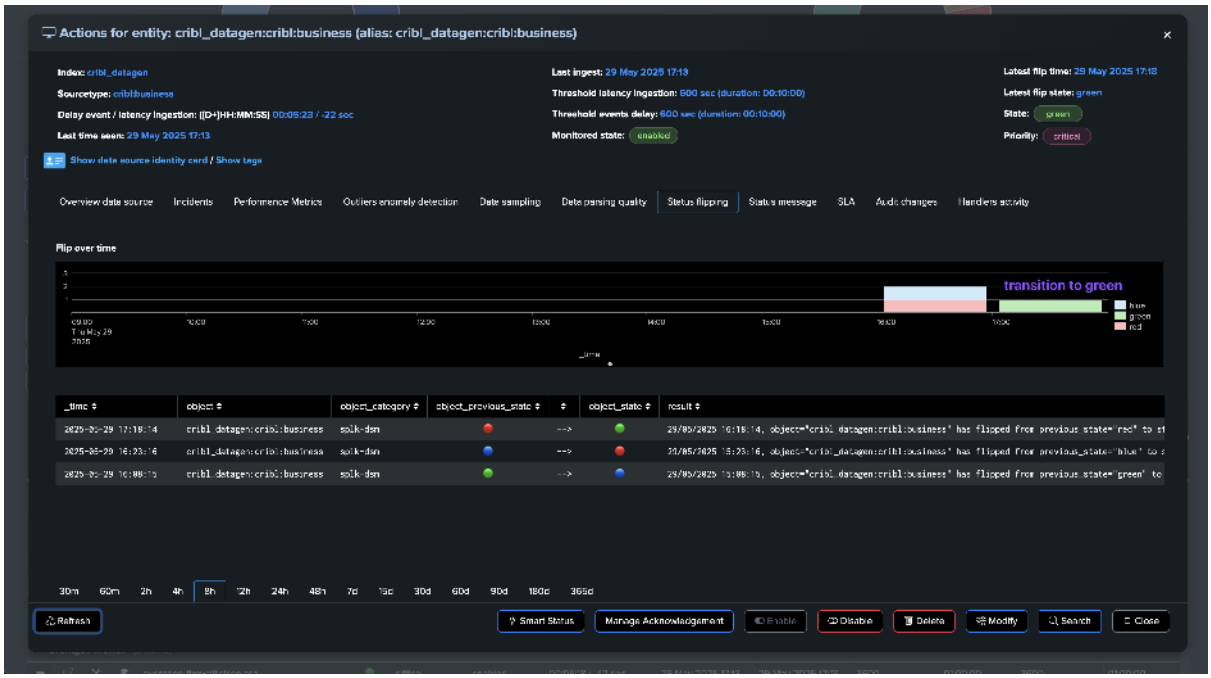
Depending on the conditions, an interruption in the data flow can also be generating an Outliers anomaly detection:



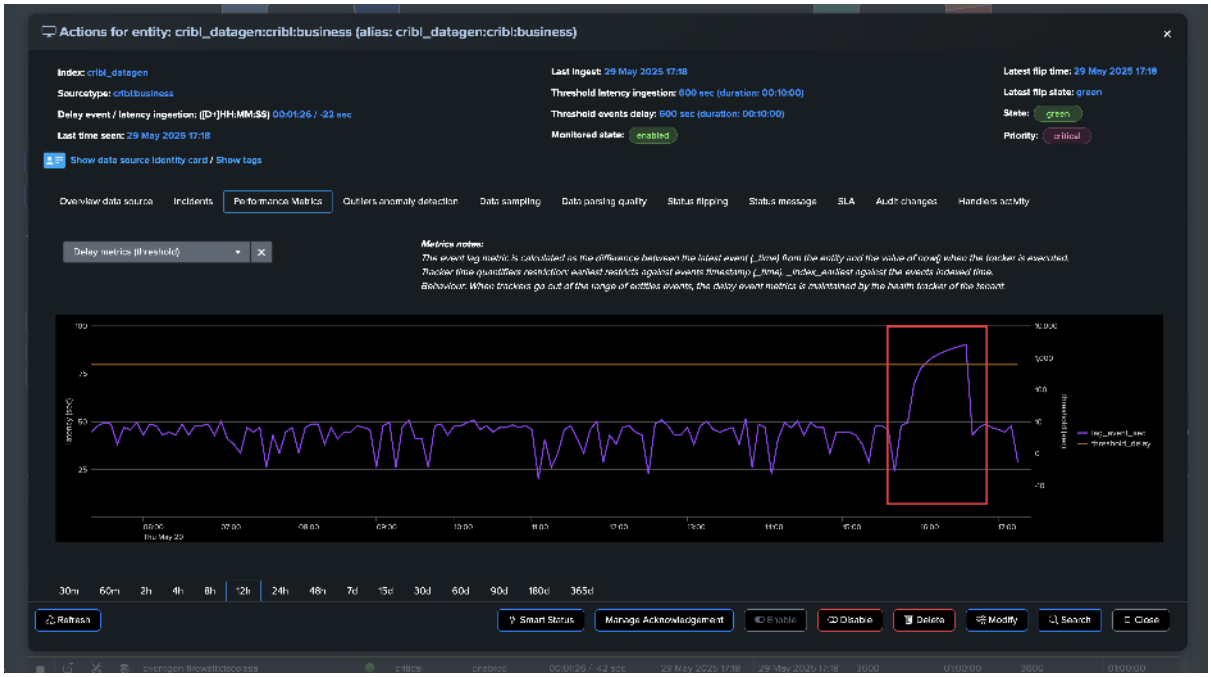
If alerting has been setup, TrackMe would generate an alert with this content:

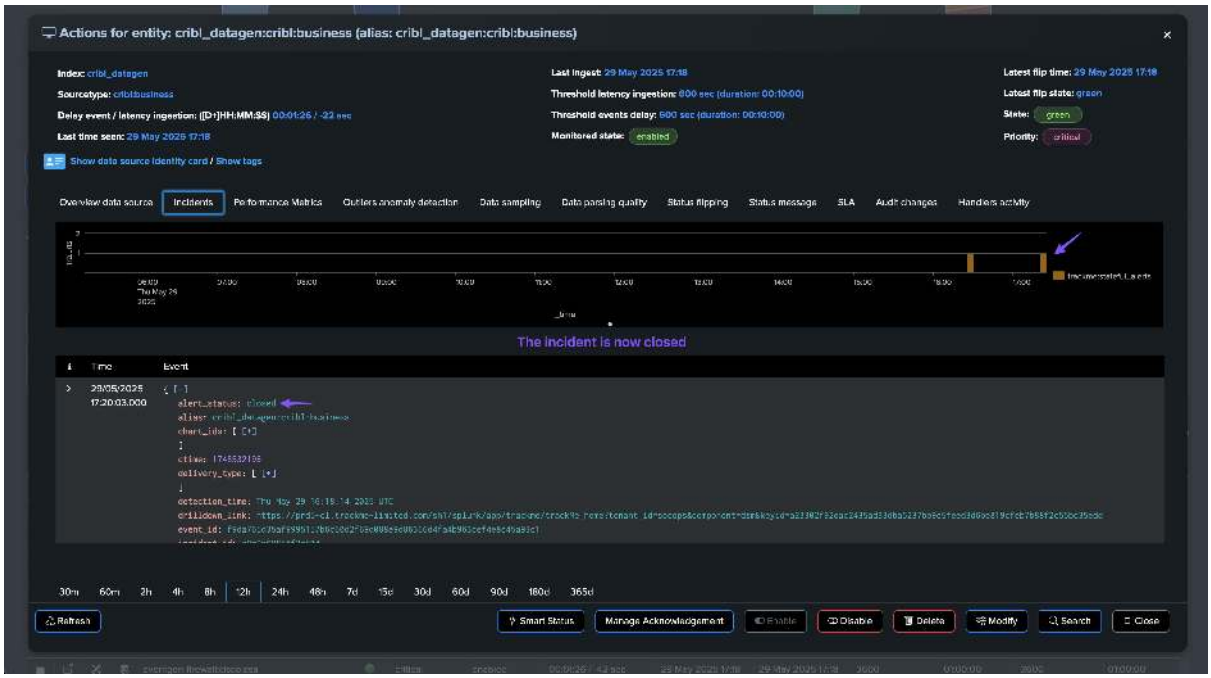


At some point, the issue was resolved and the data is back in Splunk:

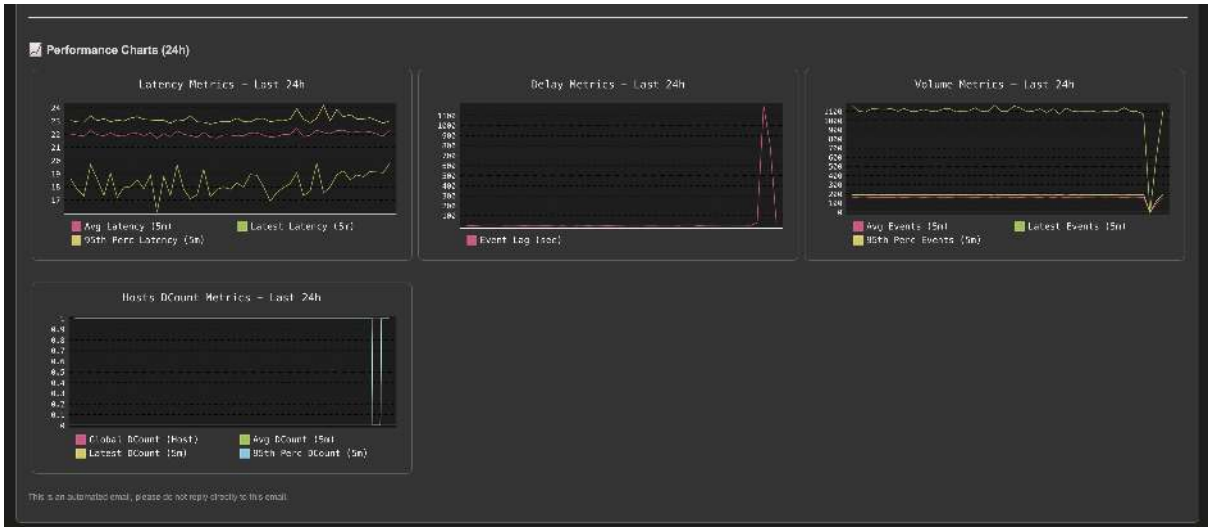


The alert is resolved and a closure notification is generated and sent:


Actions for entity: cribl_datagen:cribl:business (alias: cribl_datagen:cribl:business)
Overview data source
Incidents
Performance Metrics
Outliers anomaly detection
Data sampling
Data parsing quality
Status flipping
Status message
SLA
Audit changes
Handlers activity
Refresh
Smart Status
Manage Acknowledgement
Enable
Disable
Delete
Modify
Search
Close


Actions for entity: cribl_datagen:cribl:business (alias: cribl_datagen:cribl:business)
Overview data source
Incidents
Performance Metrics
Outliers anomaly detection
Data sampling
Data parsing quality
Status flipping
Status message
SLA
Audit changes
Handlers activity
The incident is now closed
Refresh
Smart Status
Manage Acknowledgement
Enable
Disable
Delete
Modify
Search
Close

TrackMe Notification
Environment: https://prd5-ci.trackme-limited.com
Detection Time: Thu May 29 16:18:14 2025 UTC
Alert Status: closed
Tenant: secops
Alias: cribl_datagen:cribl:business
Object: cribl_datagen:cribl:business
Drilldown Link
Status: green
Incident ID: c0c9a69859f2c504
Detailed Information:
The entity has received an incident closure update and is now in a non-alerting state:
Alias: cribl_datagen:cribl:business
Object: cribl_datagen:cribl:business
Object ID: a23302f92eac2435ad33dba5237bb9e5feed3d6be819cfeb7b88f2c55bc35edd
Category: splk-dsm
Anomaly Reason(s):
none
Message(s):
29/05/2025 16:18:14, object="cribl_datagen:cribl:business" has flipped from previous_state="red" to state="green" with anomaly_reason="none", previous_anomaly_reason="ml_outliers_detection", disruption_time="3297.98"
monitoring conditions for event delay are met. Event delay is -1.0 seconds (duration: -00:00:01), which is lower than the maximum allowed delay of 600 seconds (duration: 00:10:00), latest event available (_time) for this entity: 29 May 2025 16:18
monitoring conditions for ingest latency are met. Ingestion latency is approximately 22.037 seconds (duration: 00:00:22), which is lower than the maximum allowed latency of 600 seconds (duration: 00:10:00), latest event indexed (_indextime) for this entity: 29 May 2025 16:18
Message source: trackme:flip
Message source ID: 957804db6b688f16a3cb2252913390853a2082918f6c12337bd49e83eeaca360



About Adaptive Delay Thresholding

In addition with the threshold concept above, TrackMe also leverages Machine Learning to adapt the delay threshold based on the knowedge accumulated over time on a per entity basis, and depending on the conditions.

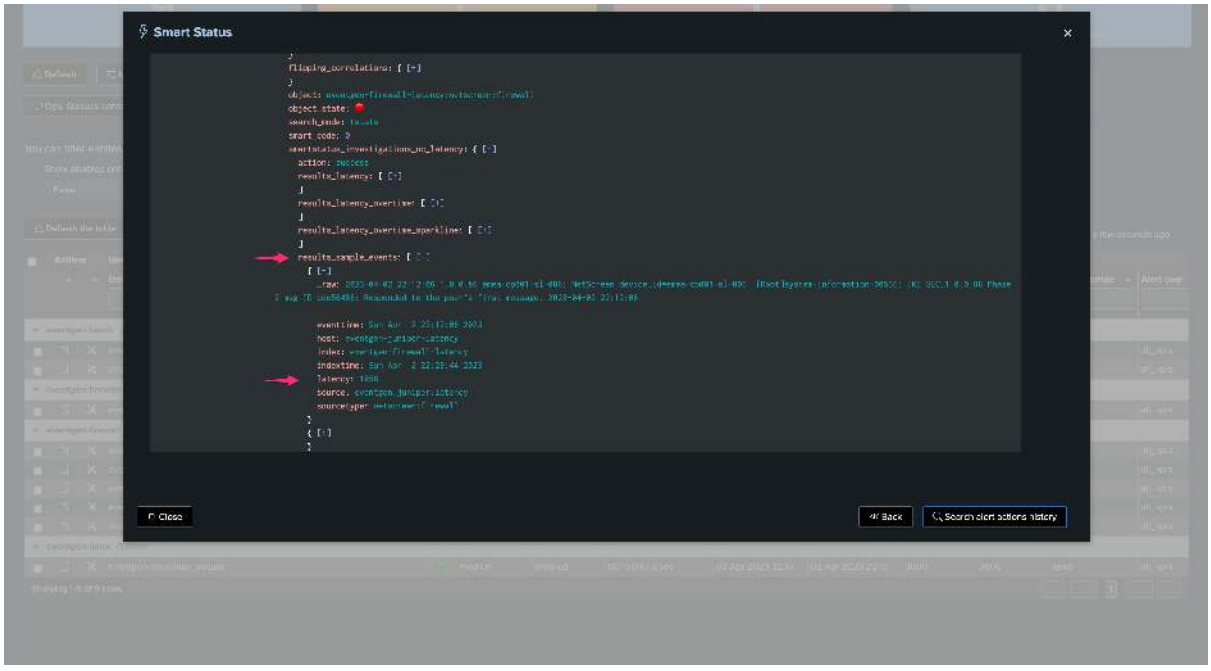
This is called "Adaptive Delay Thresholding" and and is widely documented here: *Splunk Feeds Delayed & Inactive Entities (splk-feeds)*.

This feature is very valuable to reduce the administration costs by automatically defining the best suitable values according to the entity behaviour, it can increase and decrease thesholds in function of the events and statuses.

2.1.6 Use Case: Feed Indexing Performance Issues Detection

Understanding Latency in Splunk

Latency is a critical metric that measures the time gap between when an event is generated and when it becomes available in Splunk. Think of it as the "travel time" for your data:



Why Latency Matters

When latency issues occur, they can cause significant problems:

- **Search Inconsistency**: Searches run at different times may return different results
- **Delayed Insights**: Critical information arrives later than expected
- **Resource Impact**: Can indicate underlying system problems

Common Causes of Latency

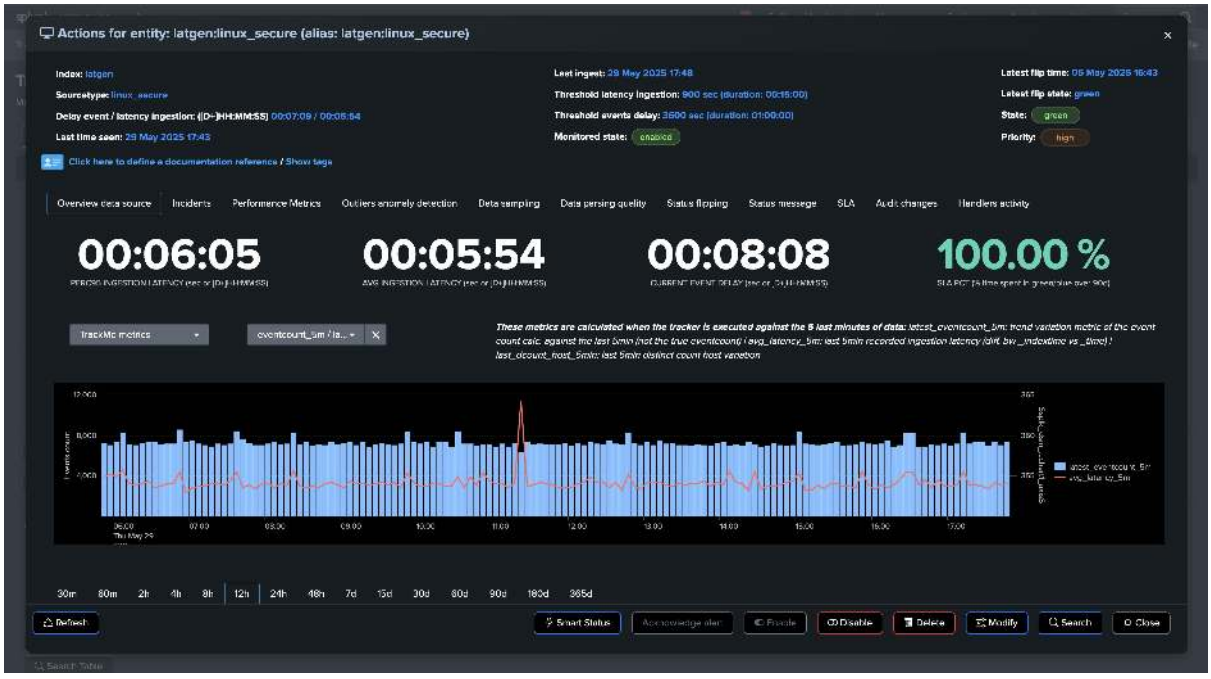
Latency issues can stem from various sources:

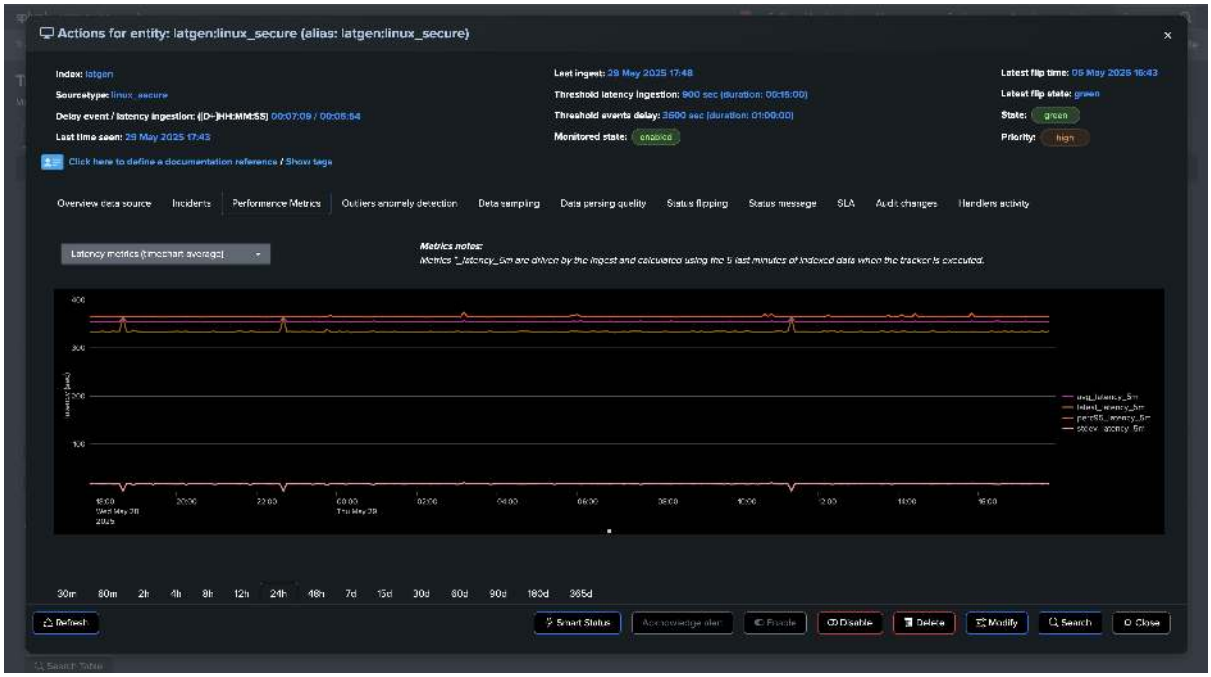
- Queue congestion
- Resource constraints
- Network bottlenecks
- System overload
- Configuration issues

How TrackMe Detects Latency Issues

TrackMe monitors latency by comparing the indexed time against the event time. When latency exceeds your configured threshold, TrackMe triggers an anomaly with `anomaly_reason="latency_threshold_breached"`.

Let's consider the following entity, currently the latency is very low and the entity is considered as healthy:



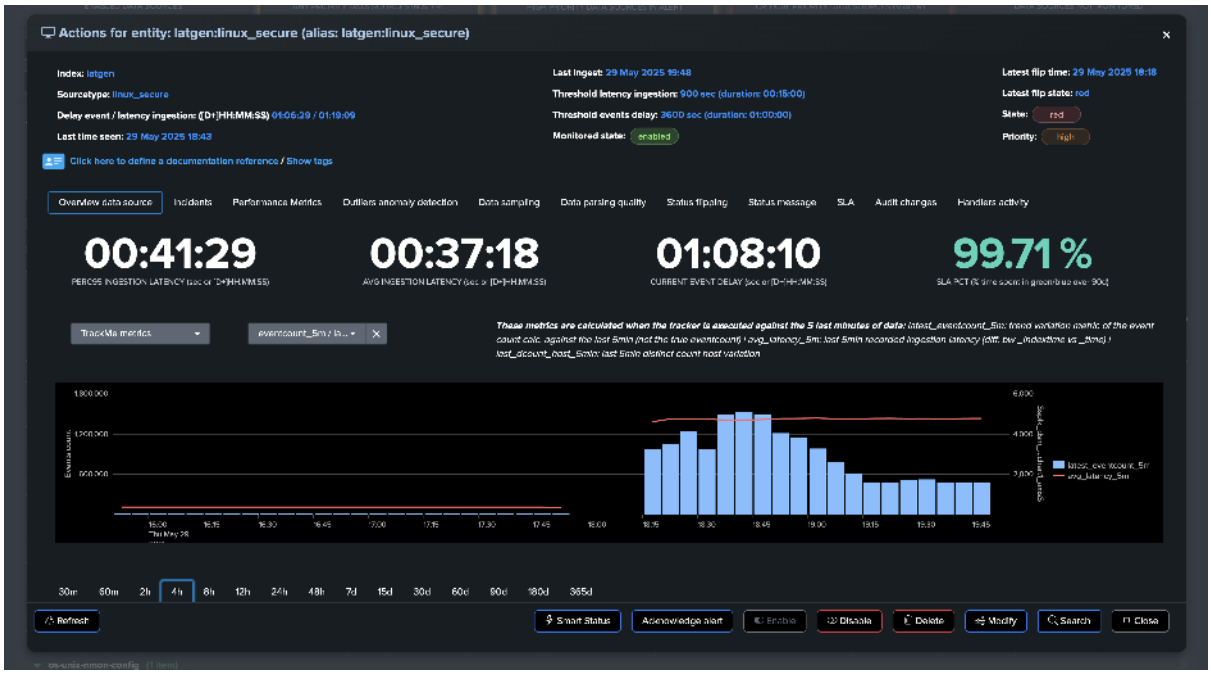


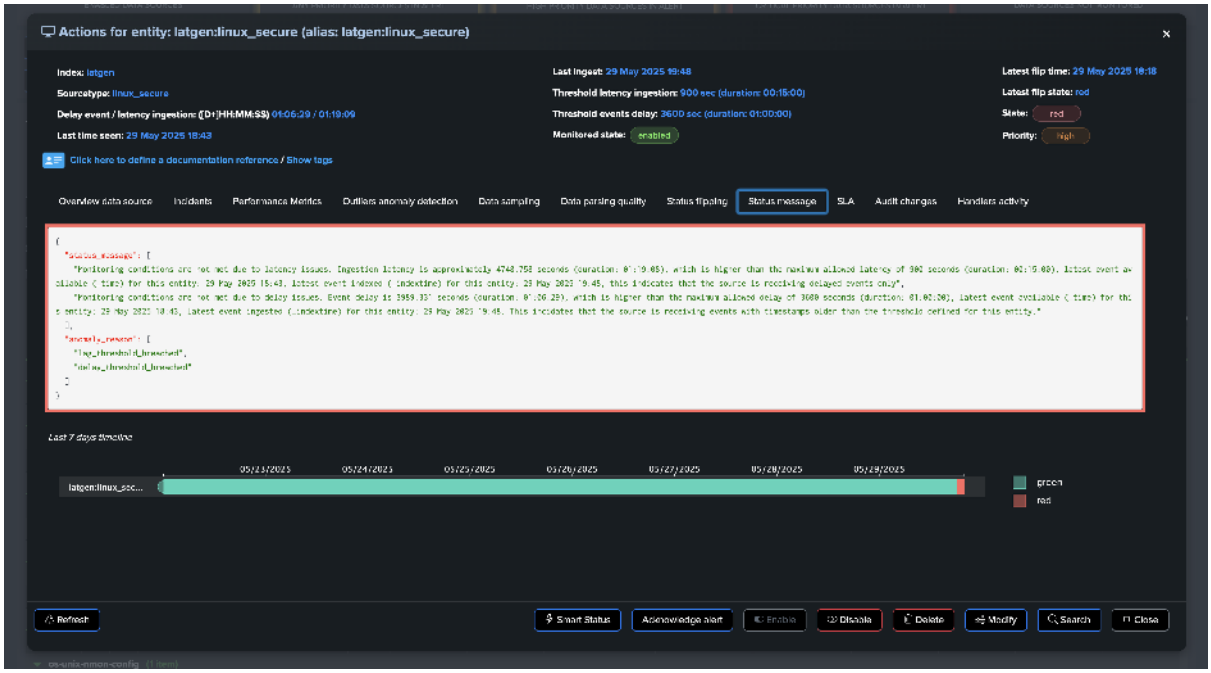
Suddenly, the entity starts experiencing latency issues. We begin receiving events with a significant time gap between their generation and indexing:

Some additional comments about latency:

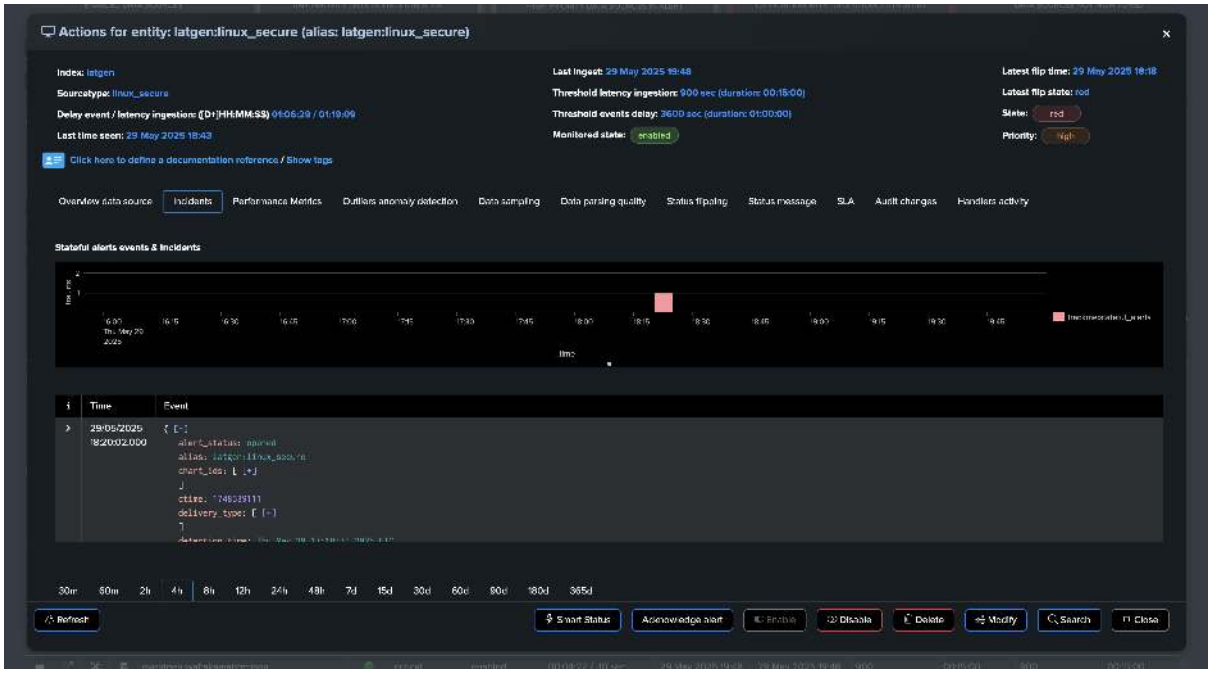
- It is likely that latency will also affect the delay, but this is not necessarily the case.
- You can well be receiving a mix of real-time or quasi-real-time events, while at the same time receiving events made available with a large latency.
- This can be the case for a variety of reasons, from data feeds catching up situations to simply having collectors suffering from performance issues while others sending to the same context do not.

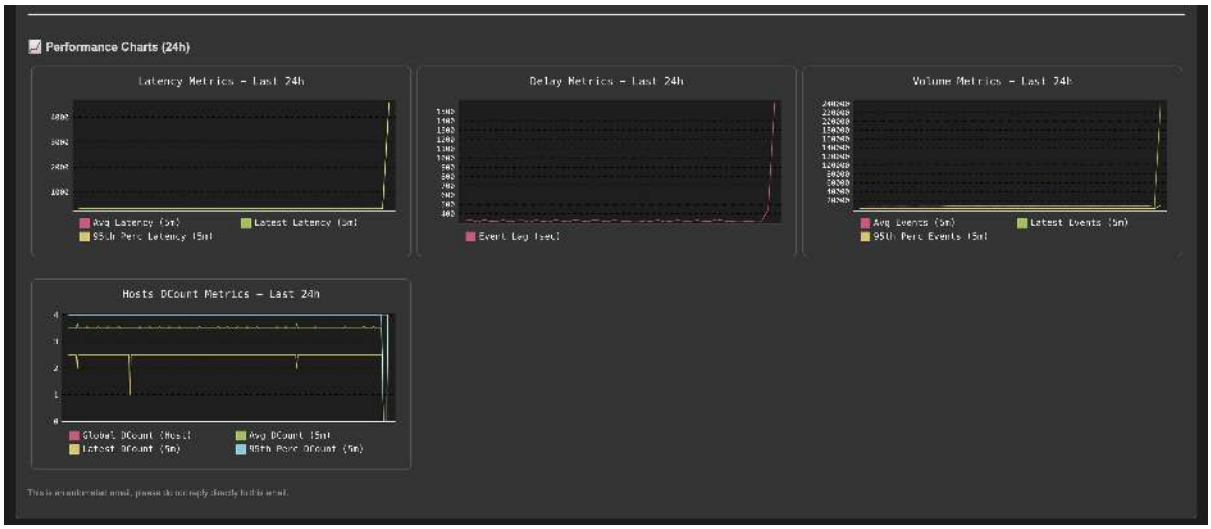
The situation escalates as the entity experiences high latency combined with a dramatic increase in event volume:



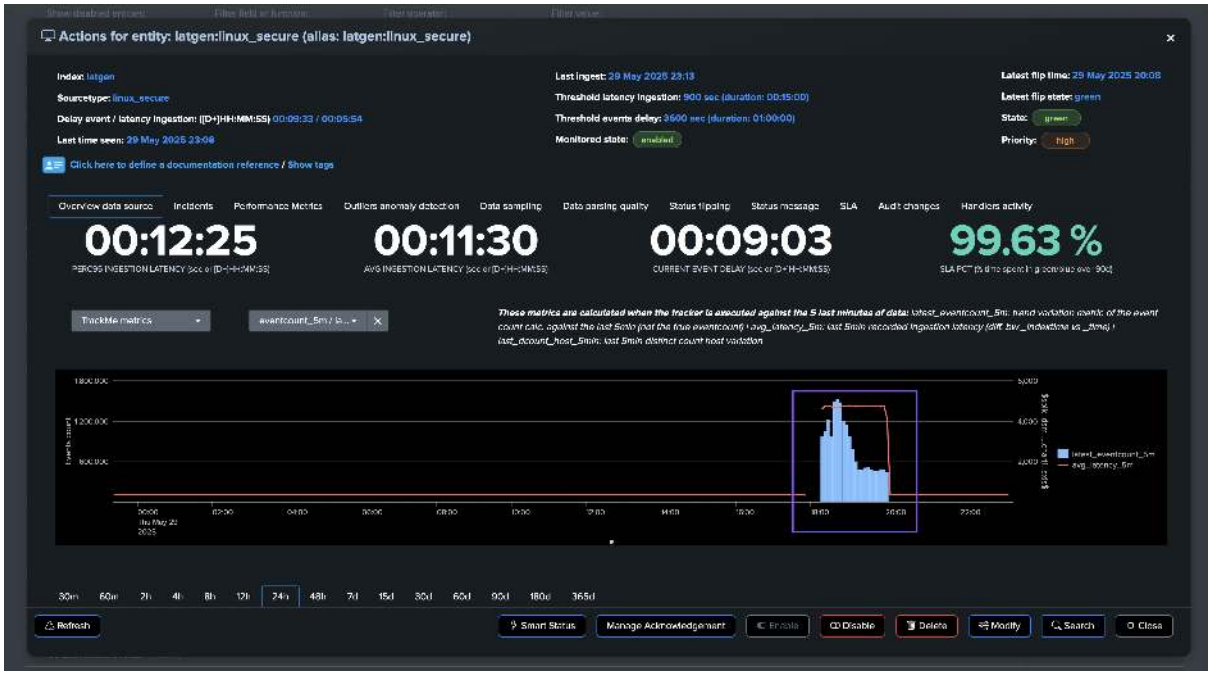


TrackMe immediately responds by opening an incident and sending a notification:

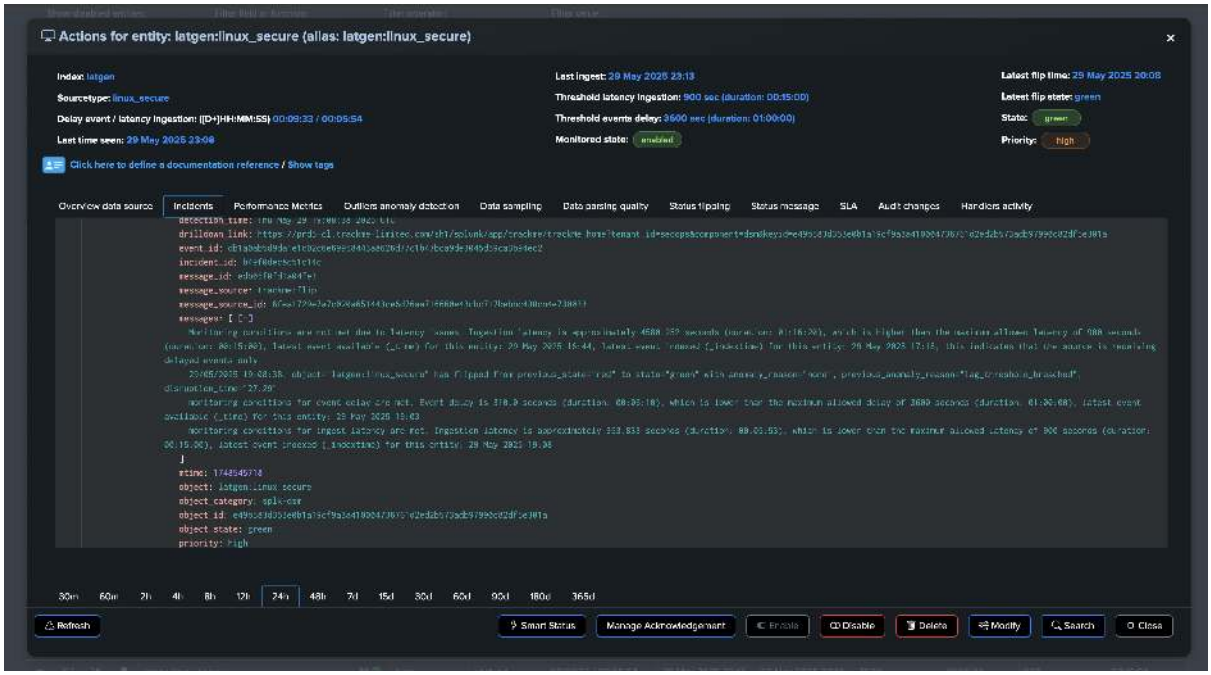




After the issue is resolved, the entity returns to a healthy state and the incident is automatically closed:


Actions for entity: latgen:linux_secure (alias: latgen:linux_secure)
Overview data source
Incidents
Performance Metrics
Outliers anomaly detection
Data sampling
Data parsing quality
Status flipping
Status message
SLA
Audit changes
Handlers activity
00:12:25
00:11:30
00:09:03
99.63 %
Refresh
Smart Status
Manage Acknowledgement
Disable
Delete
Modify
Search
Close

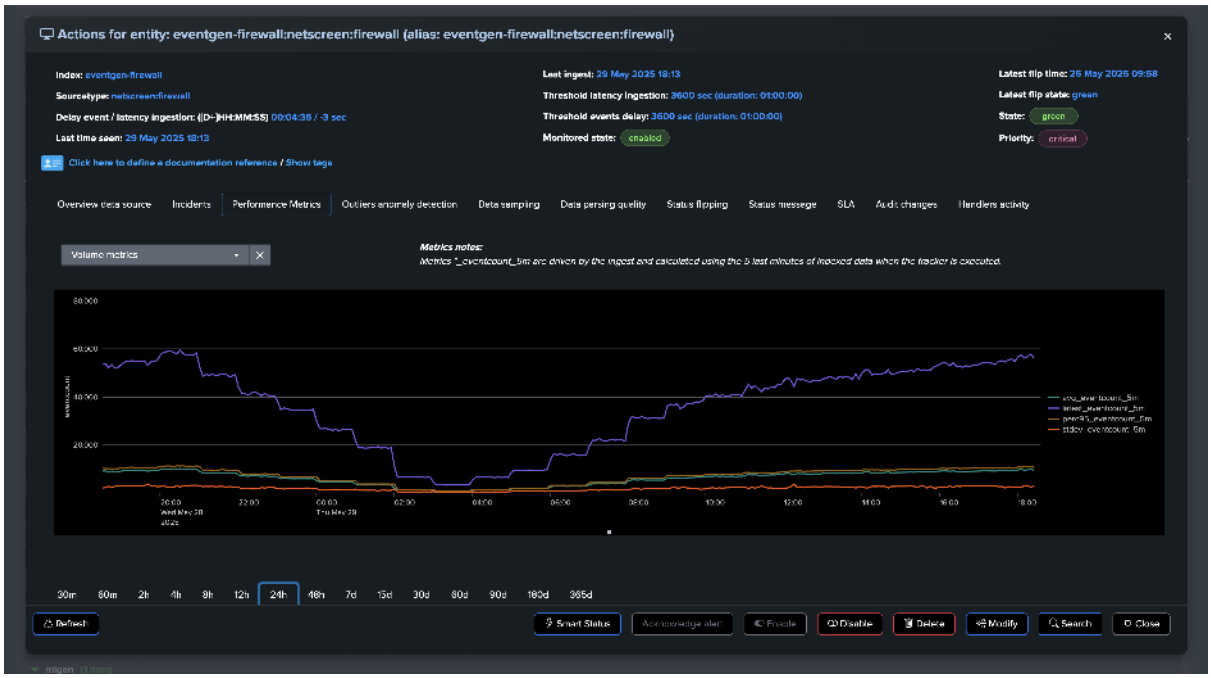
Actions for entity: latgen:linux_secure (alias: latgen:linux_secure)
Overview data source
Incidents
Performance Metrics
Outliers anomaly detection
Data sampling
Data parsing quality
Status flipping
Status message
SLA
Audit changes
Handlers activity
Refresh
Smart Status
Manage Acknowledgement
Disable
Delete
Modify
Search
Close

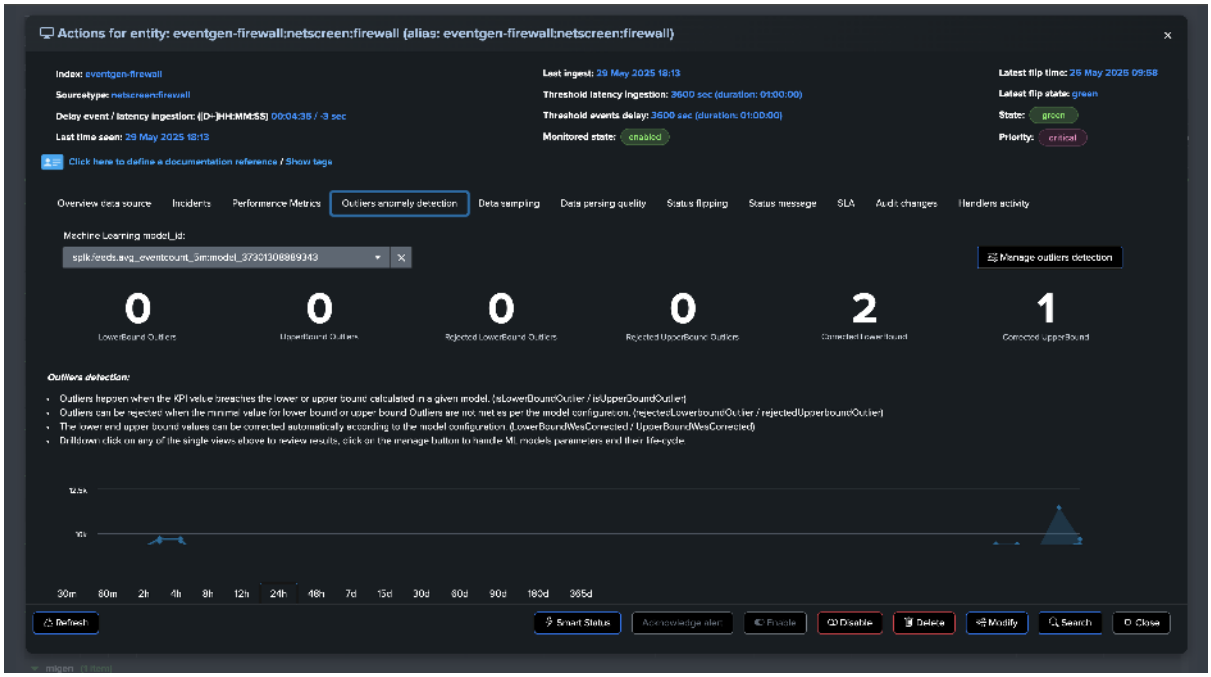


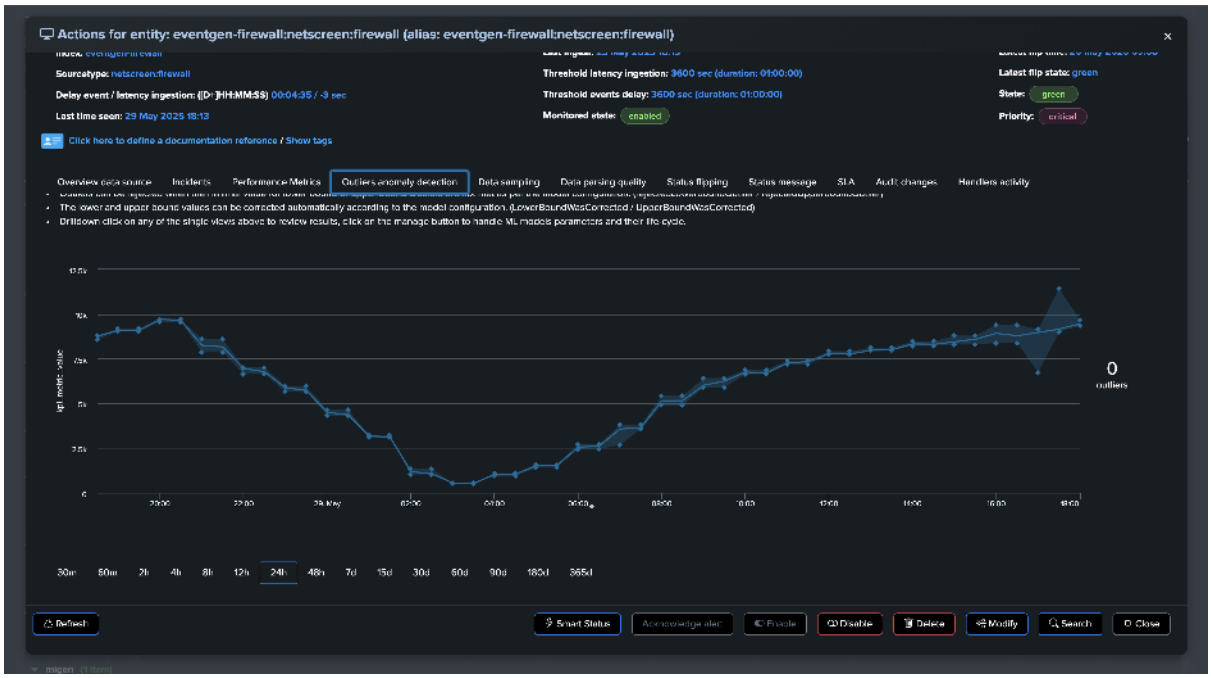
2.1.7 Use Case: Abnormally Low Volume Trend Detection (Outliers Detection)

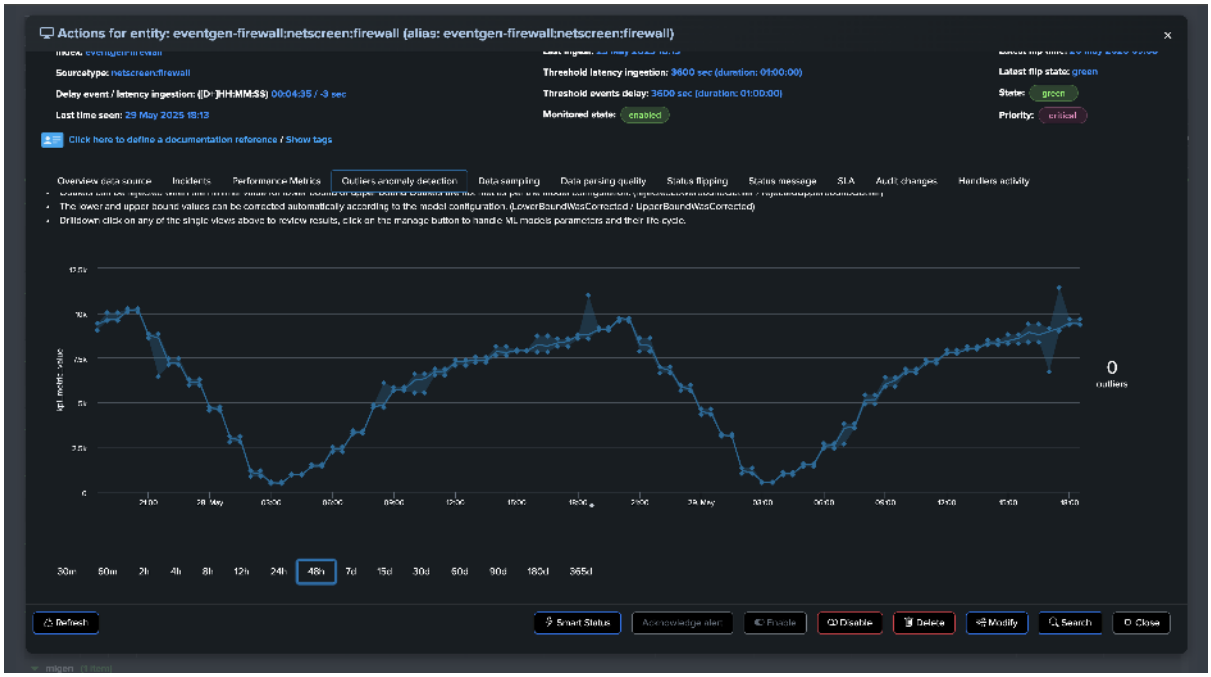
In the following example, our entity is healthy, we have data making it, not suffering from latency, and the volume is stable and expected:

Actions for entity: eventgen-firewall:netscreen:firewall (alias: eventgen-firewall:netscreen:firewall)
Overview data source
Incidents
Performance Metrics
Outliers anomaly detection
Data sampling
Data parsing quality
Status flipping
Status message
SLA
Audit changes
Handlers activity
3.0 sec
2.5 sec
00:01:37
99.66 %
trackMe metrics
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
180d
365d
Refresh
Smart Status
Disable
Delete
Modify
Search
Close

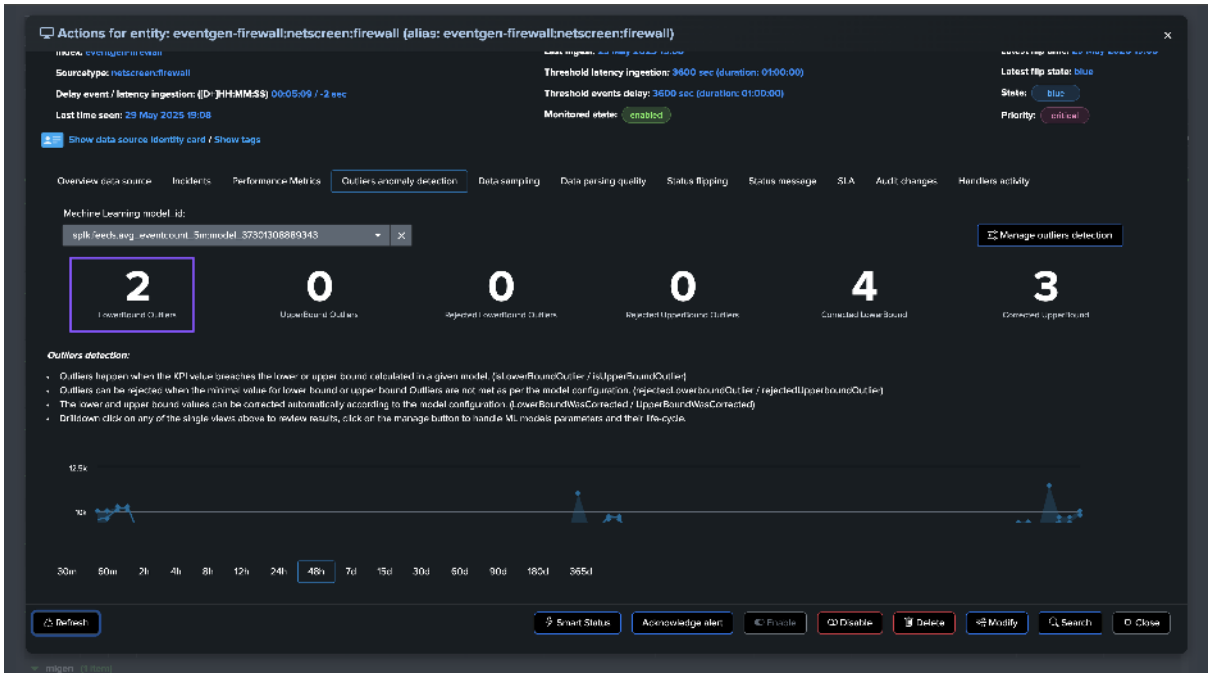

Actions for entity: eventgen-firewall:netscreen:firewall (alias: eventgen-firewall:netscreen:firewall)
Overview data source
Incidents
Performance Metrics
Outliers anomaly detection
Data sampling
Data parsing quality
Status flipping
Status message
SLA
Audit changes
Handlers activity
Volume metrics
Metrics notes:
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
180d
365d
Refresh
Smart Status
Disable
Delete
Modify
Search
Close

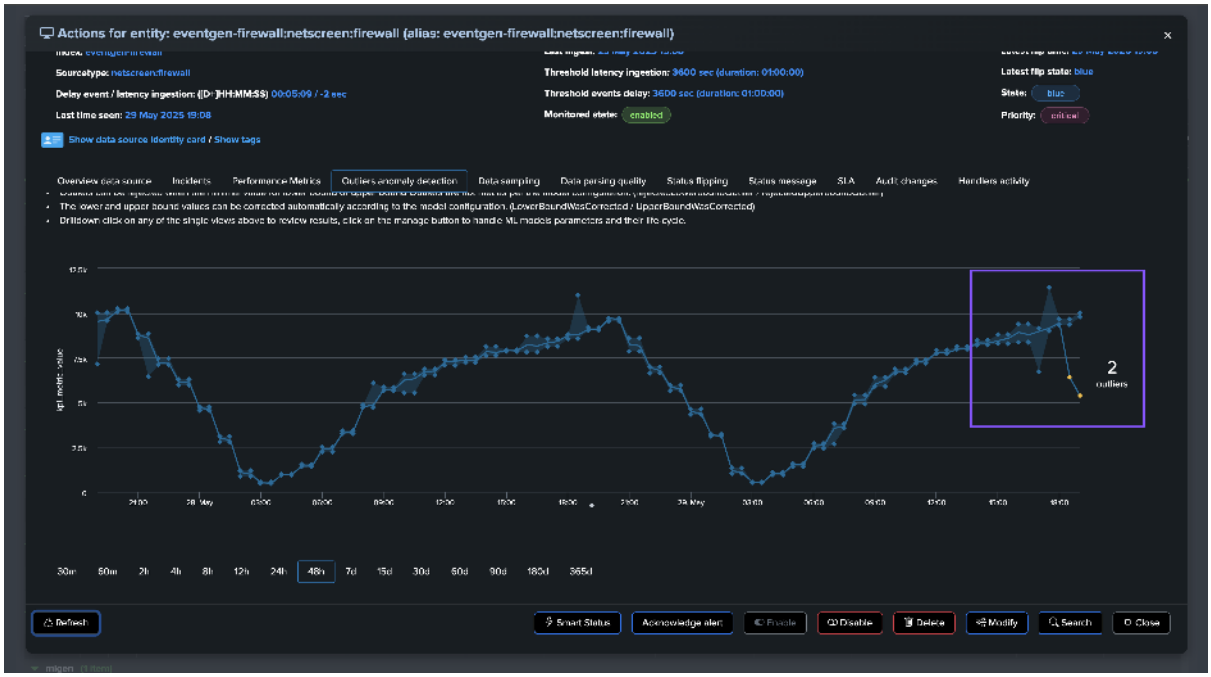

Actions for entity: eventgen-firewall:netscreen:firewall (alias: eventgen-firewall:netscreen:firewall)
Index: eventgen-firewall
Sourcetype: netscreen:firewall
Delay event / latency ingestion: ([D+]HH:MM:SS) 00:04:35 / -3 sec
Last time seen: 29 May 2025 18:13
Last ingest: 29 May 2025 18:13
Threshold latency ingestion: 3600 sec (duration: 01:00:00)
Threshold events delay: 3600 sec (duration: 01:00:00)
Monitored state: enabled
Latest flip time: 25 May 2025 09:58
Latest flip state: green
State: green
Priority: critical
Click here to define a documentation reference / Show tags
Overview data source
Incidents
Performance Metrics
Outliers anomaly detection
Data sampling
Data parsing quality
Status flipping
Status message
SLA
Audit changes
Handlers activity
Machine Learning model_id:
Manage outliers detection
0
0
0
0
2
1
Outliers detection:
Outliers happen when the KPI value breaches the lower or upper bound calculated in a given model. (isLowerBoundOutlier / isUpperBoundOutlier)
Outliers can be rejected when the minimal value for lower bound or upper bound Outliers are not met as per the model configuration. (rejectedLowerboundOutlier / rejectedUpperboundOutlier)
The lower and upper bound values can be corrected automatically according to the model configuration (LowerBoundWasCorrected / UpperBoundWasCorrected)
Drilldown: click on any of the single views above to review results, click on the manage button to handle ML models parameters and their life cycle
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
180d
365d
Refresh
Smart Status
Acknowledge alert
Enable
Disable
Delete
Modify
Search
Close


Actions for entity: eventgen-firewall:netscreen:firewall (alias: eventgen-firewall:netscreen:firewall)
Sourcetype: netscreen:firewall
Delay event / latency ingestion: ([D+]HH:MM:SS) 00:04:35 / -3 sec
Last time seen: 29 May 2025 18:13
Threshold latency ingestion: 3600 sec (duration: 01:00:00)
Threshold events delay: 3600 sec (duration: 01:00:00)
Monitored state: enabled
Latest flip state: green
State: green
Priority: critical
Click here to define a documentation reference / Show tags
Overview data source
Incidents
Performance Metrics
Outliers anomaly detection
Data sampling
Data parsing quality
Status flipping
Status message
SLA
Audit changes
Handlers activity
Drilldown: click on any of the single views above to review results, click on the manage button to handle ML models parameters and their life cycle
0
outliers
29 May
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
180d
365d
Refresh
Smart Status
Acknowledge alert
Enable
Disable
Delete
Modify
Search
Close

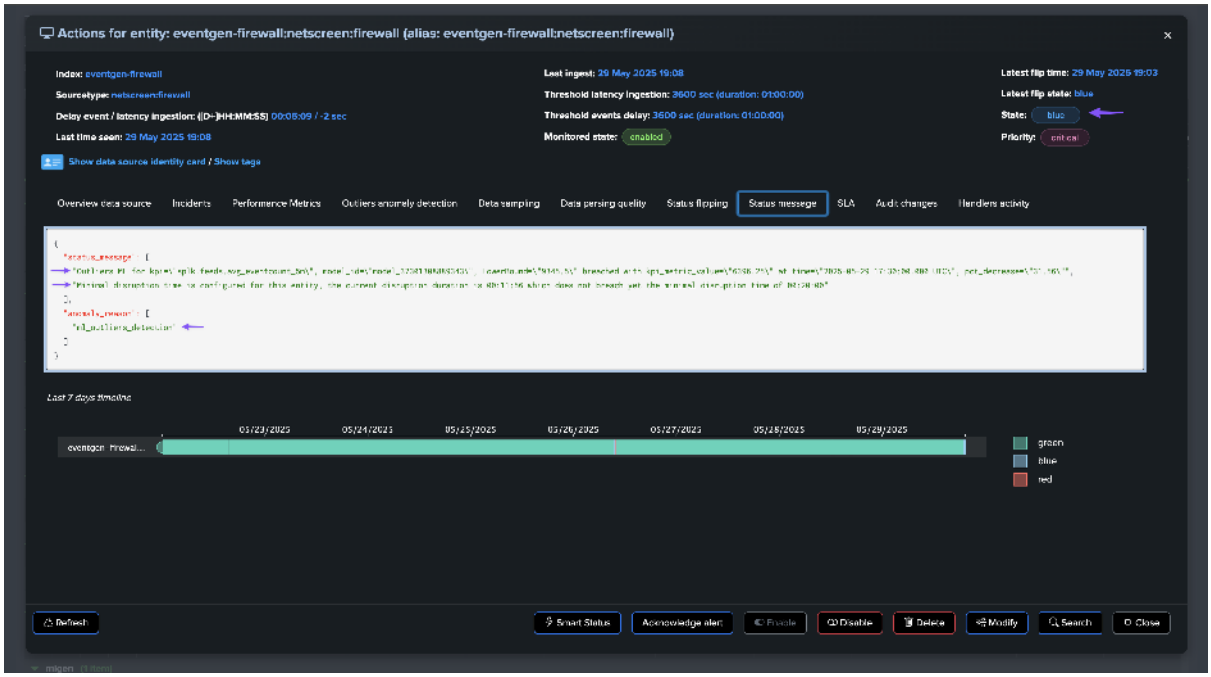


At some point, we start to see issues, we are still receiving events, but in a proportion that is not expected and indicates a potential issue:

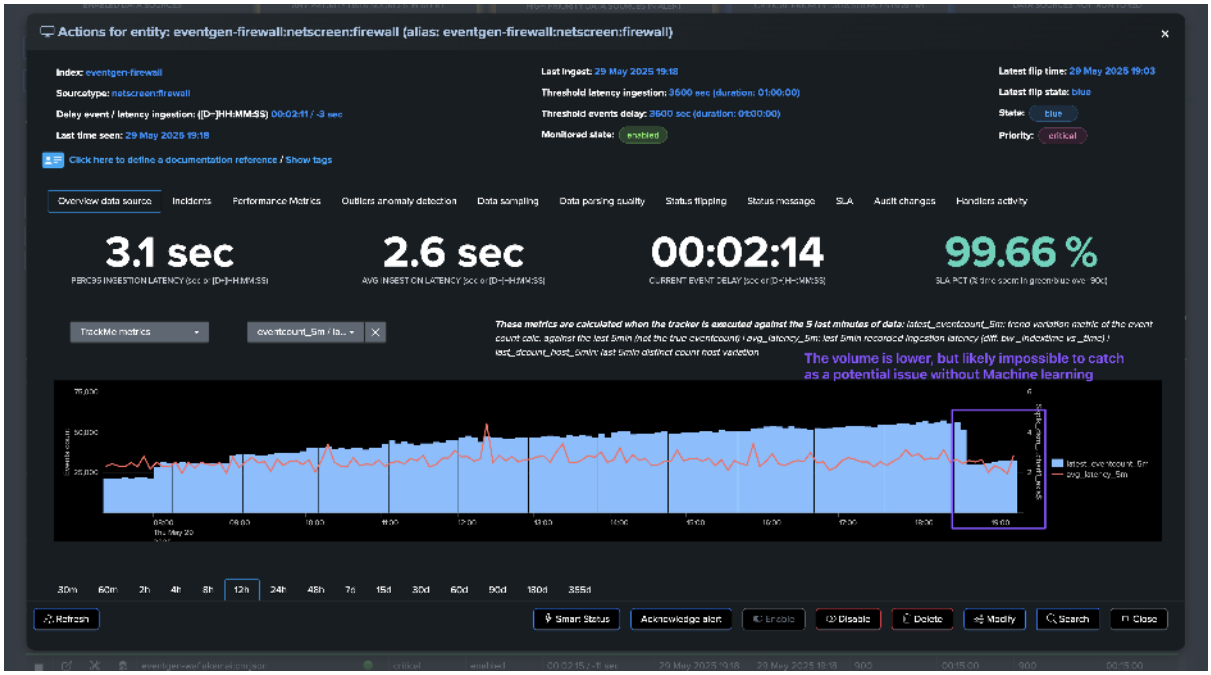


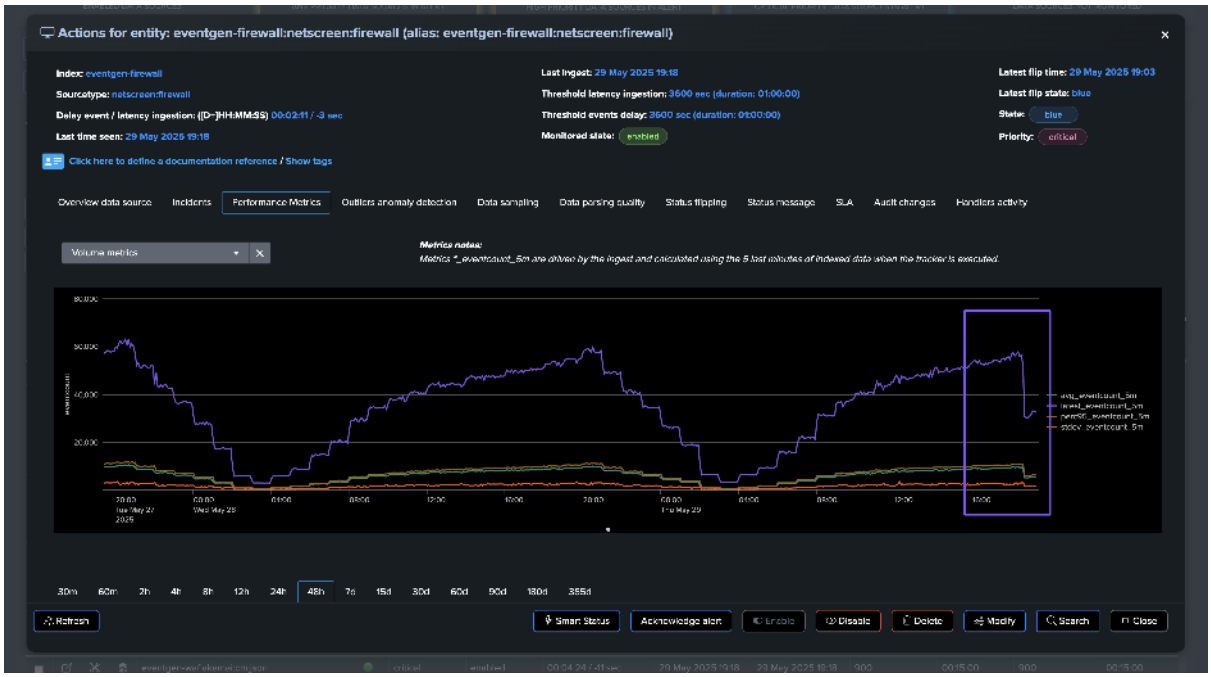


As this entity had a disruption queue configured, once the duration is breached, the entity transitioned to red and is now officially in alert:

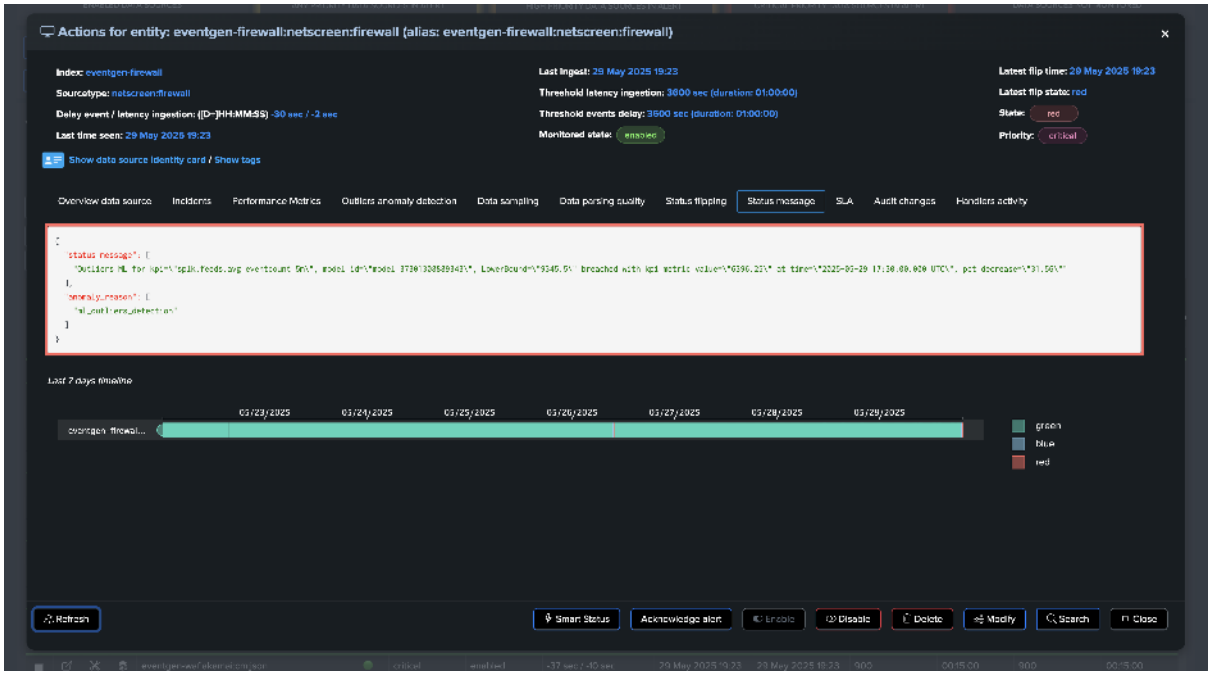


A significant amount of events is still making it to Splunk, without looking at this from the lens of Machine Learning and accumulated knowledge, we would have a hard time understanding that an abnormally low volume is happening.

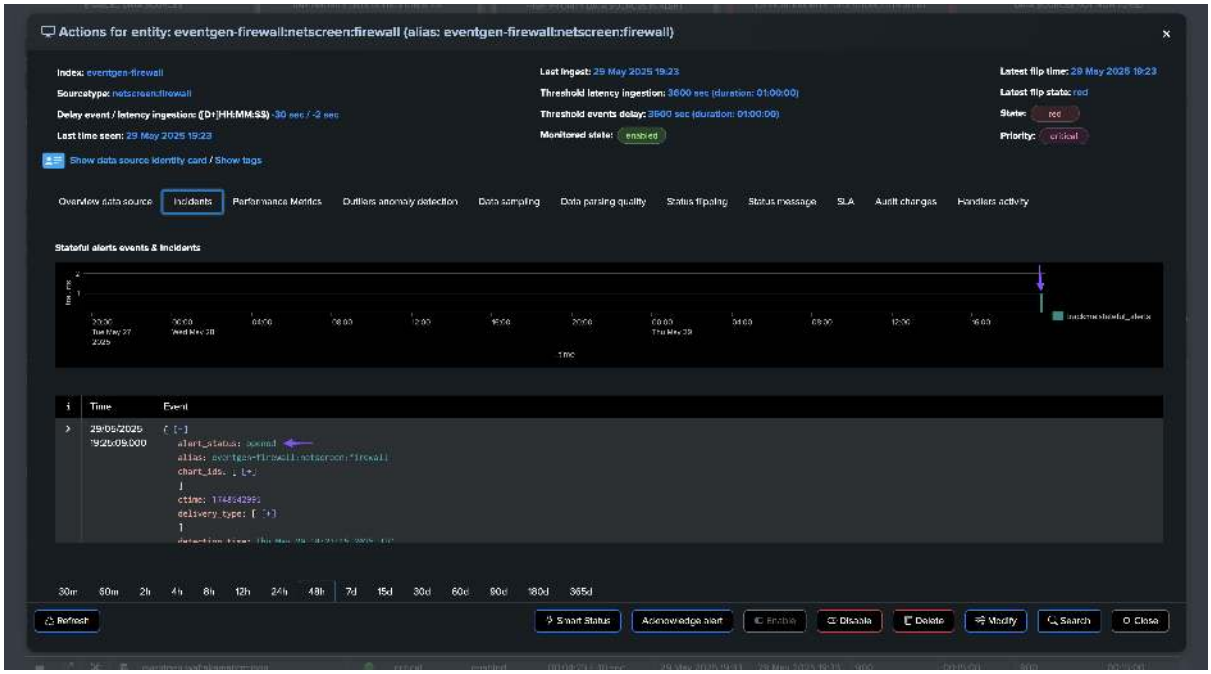


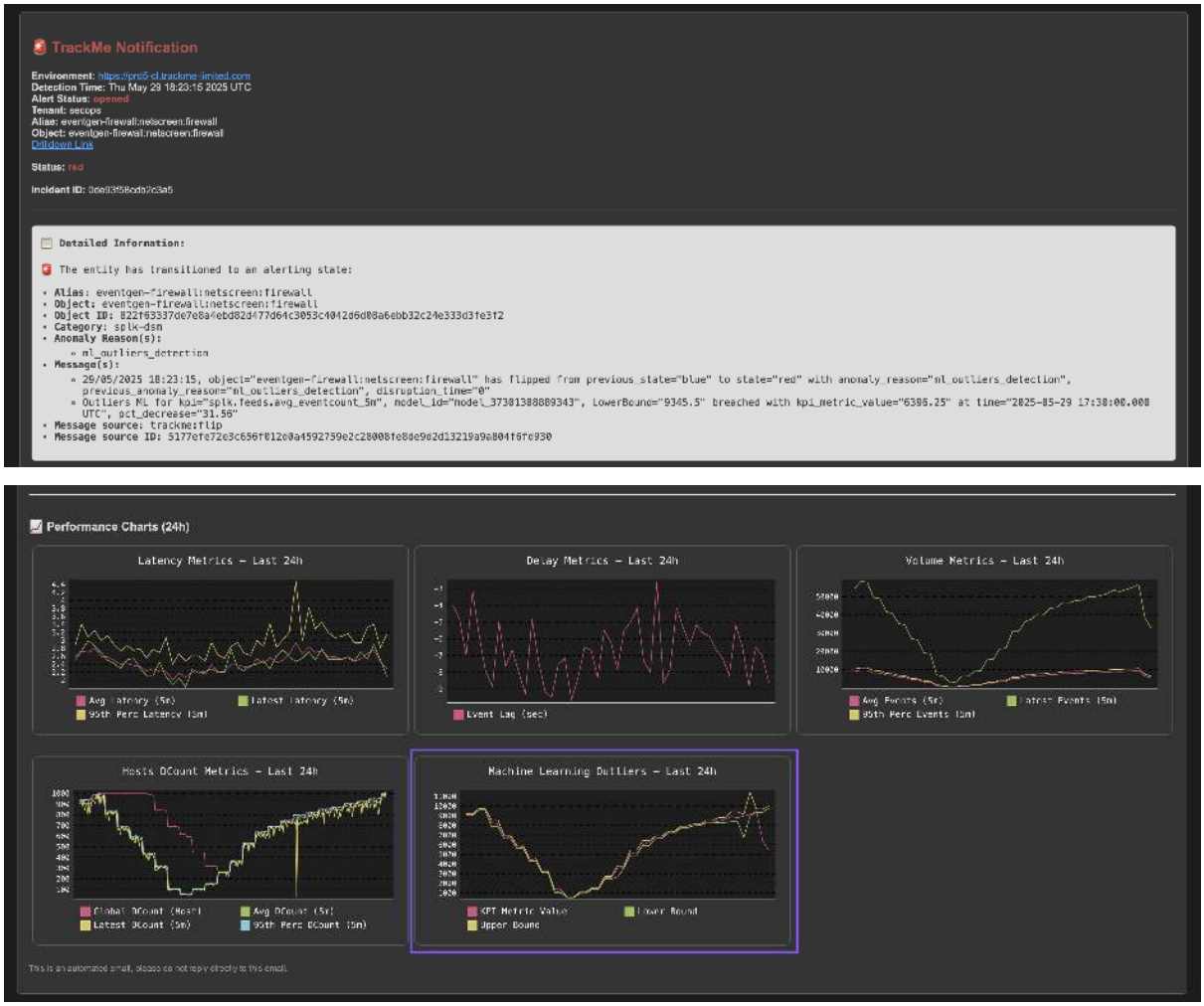


Once the disruption duration is over, the entity is now in alert:

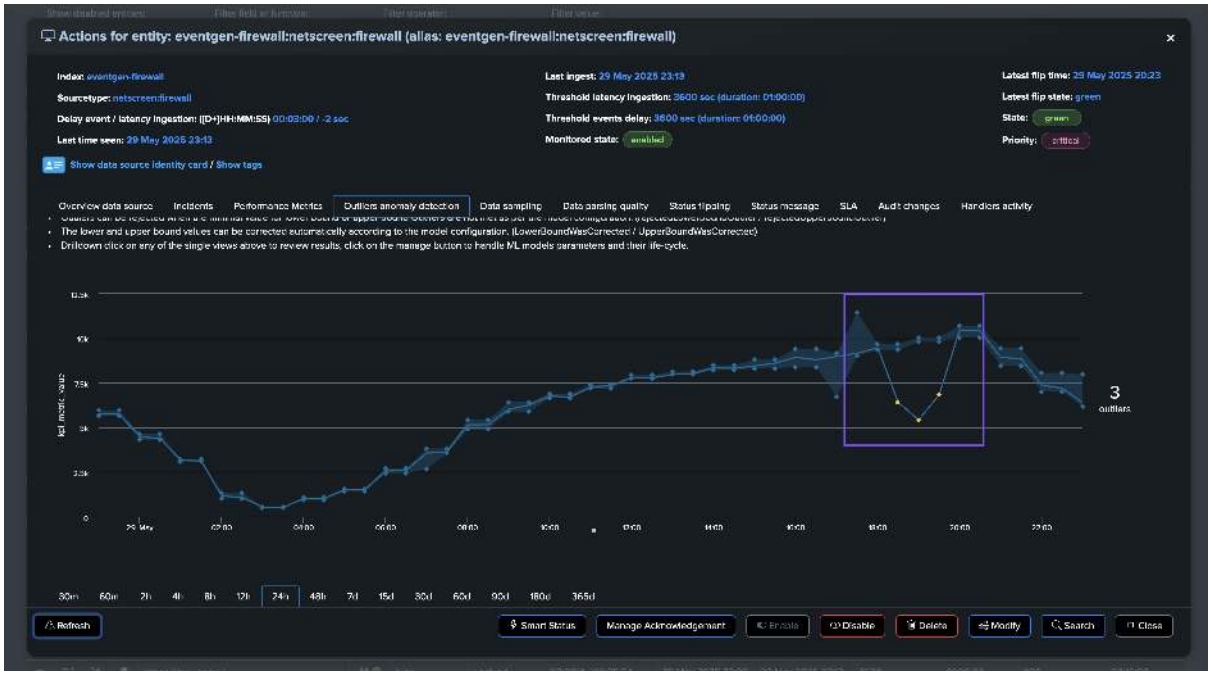


An incident is opened and a notification is sent:

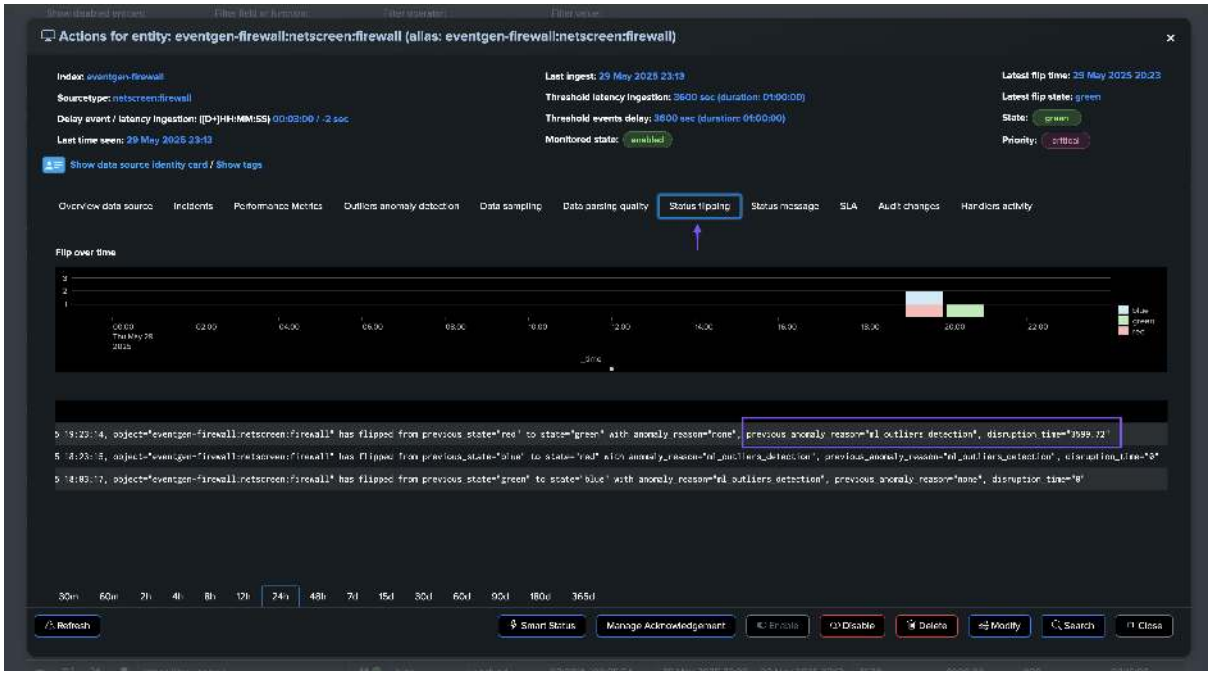


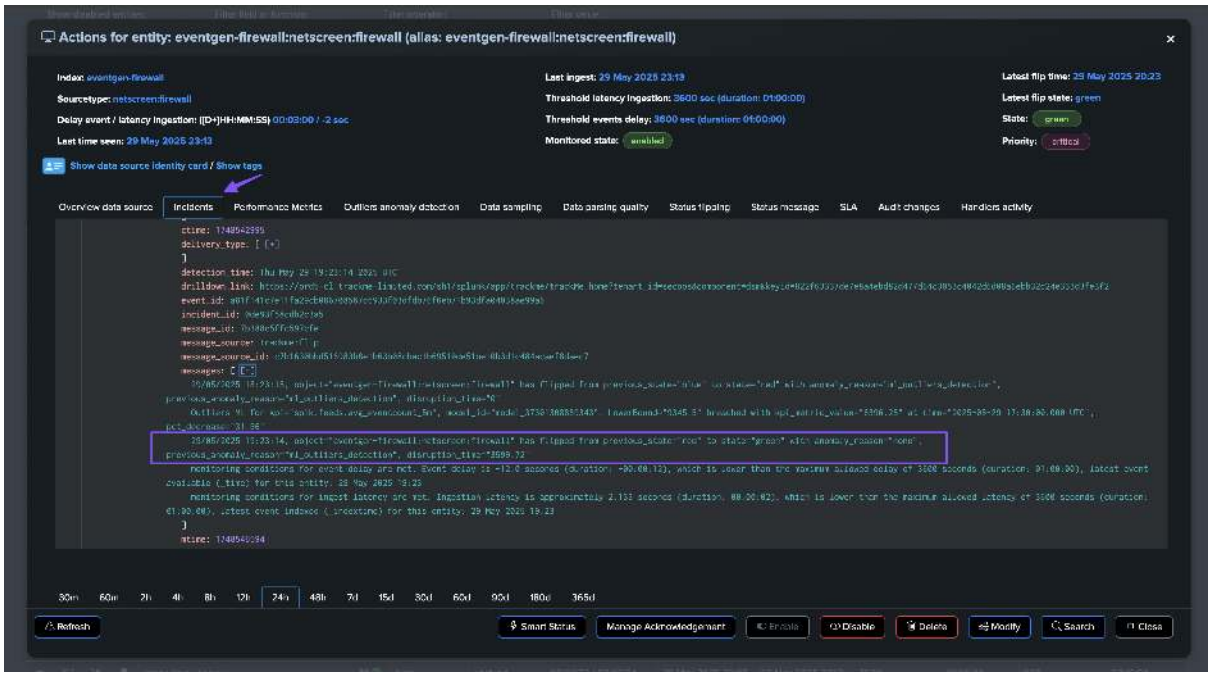


Once the issue is resolved, the entity will return to green after the Outliers anomaly condition is resolved and TrackMe has processed the ML rendering for this entity: (this can take some time depending on your settings and scale)



The incident is closed and a closure notification is sent:





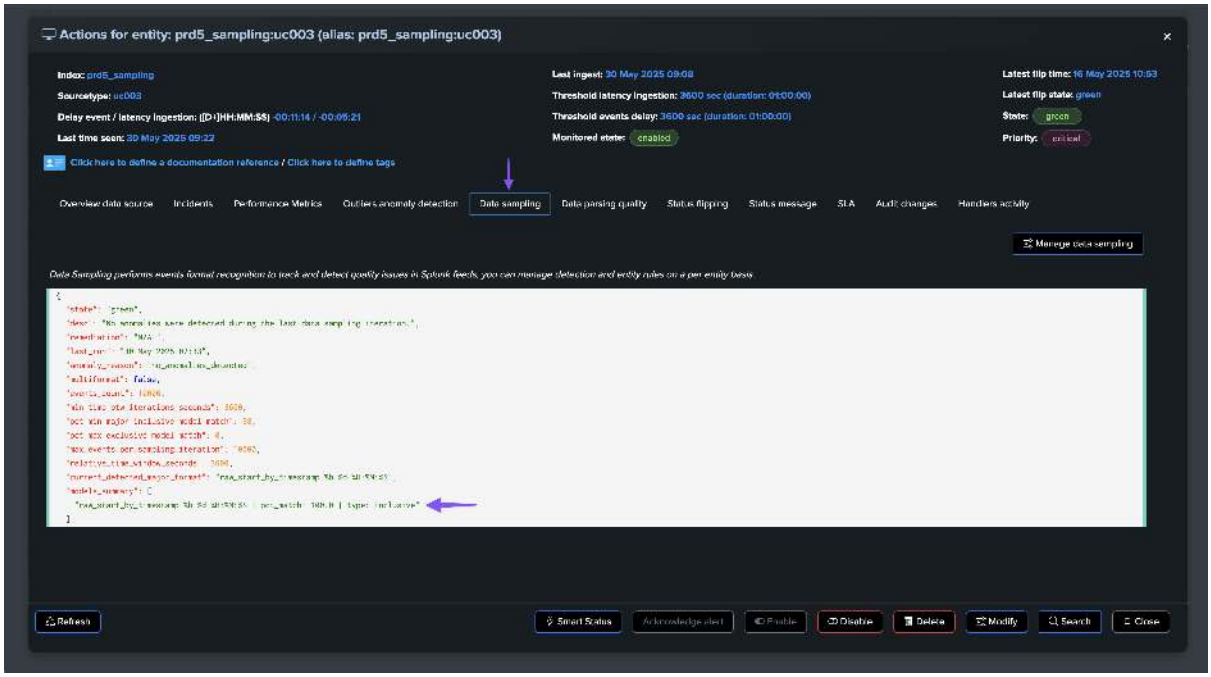
2.1.8 Use Case: Quality Issues Detection (Event Format Recognition)

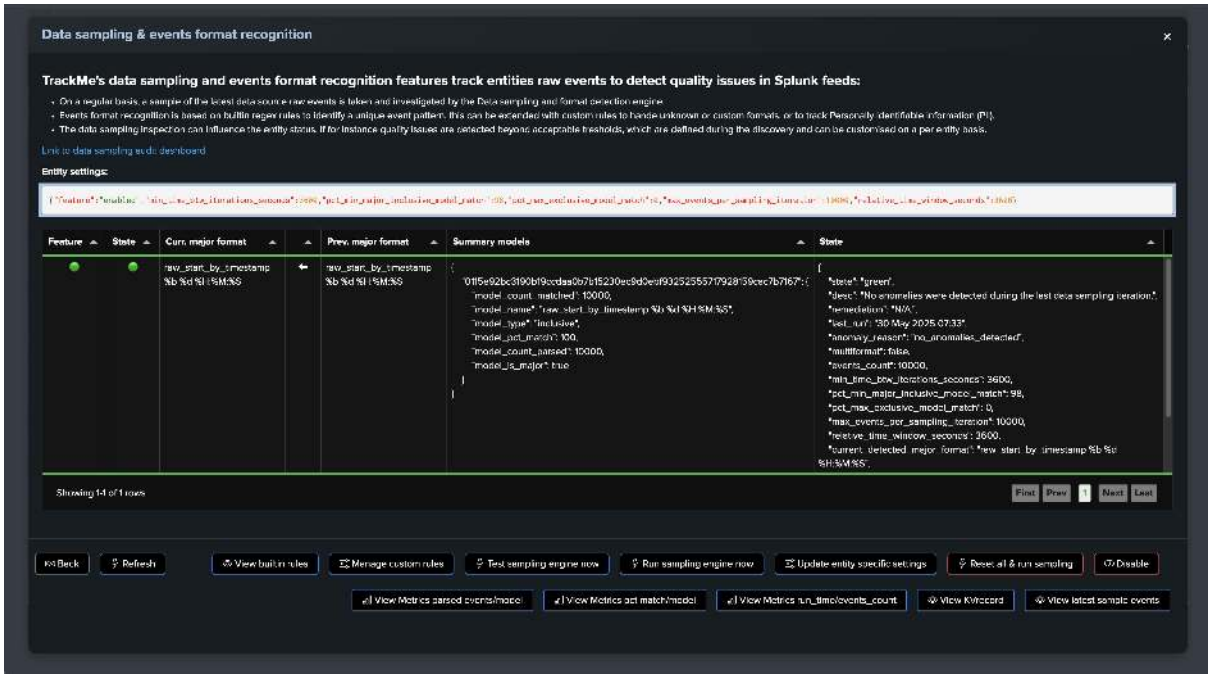
In Data Source monitoring (splk-dsm), TrackMe performs automated event format recognition using the events format recognition engine.

During this process, we will pick samples of events regularly, 10k events by default, which will be processed through the engine to detect the format of events, classified by types.

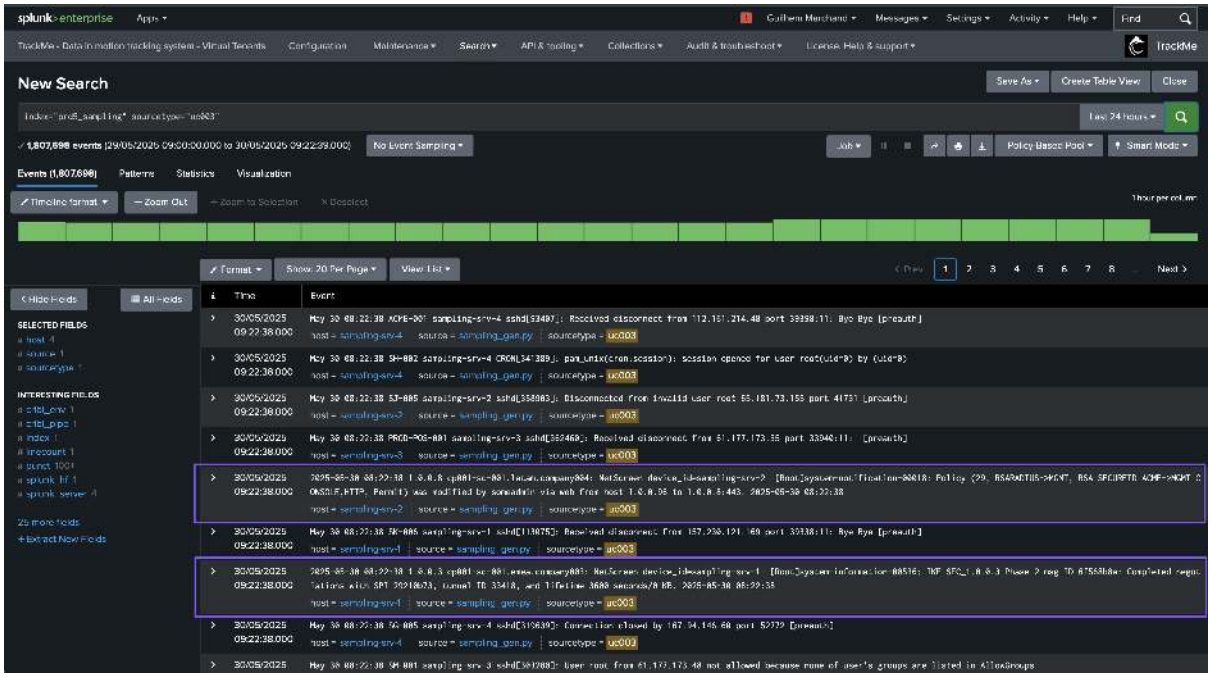
By default, TrackMe would alert against a lack of quality if it detects that the main format goes below 98% of the events being correctly classified.

Let's consider the following entity, the data sampling engine detected a single format and the entity is considered as healthy:



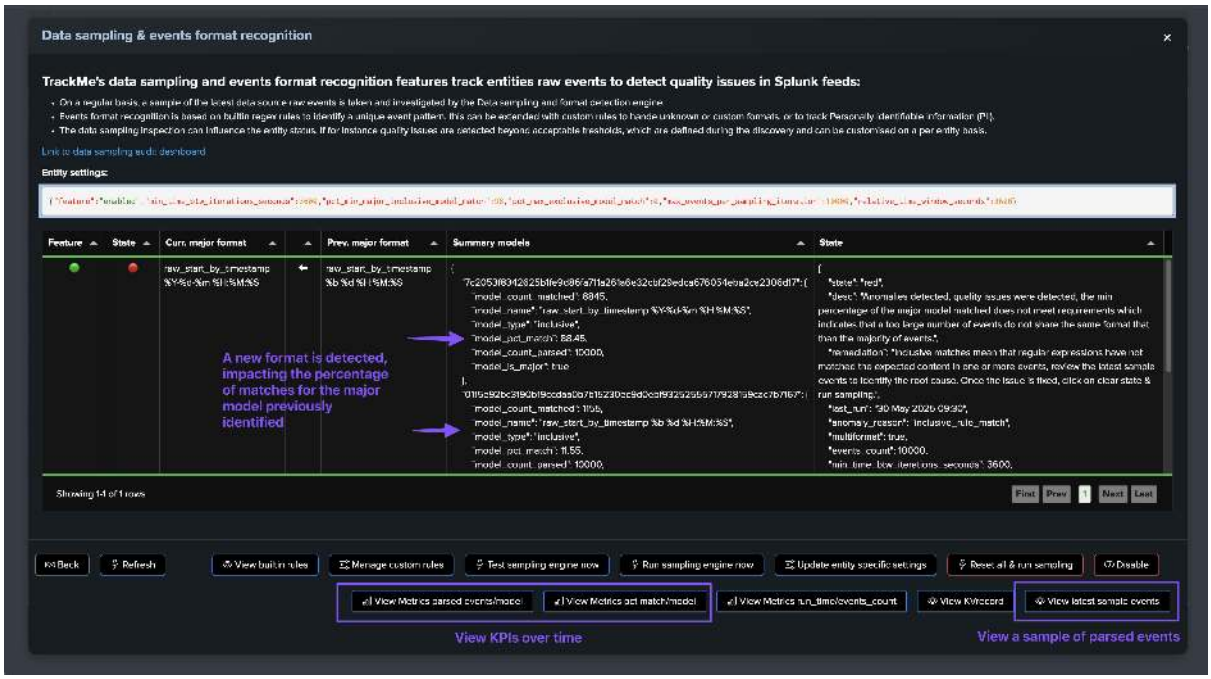


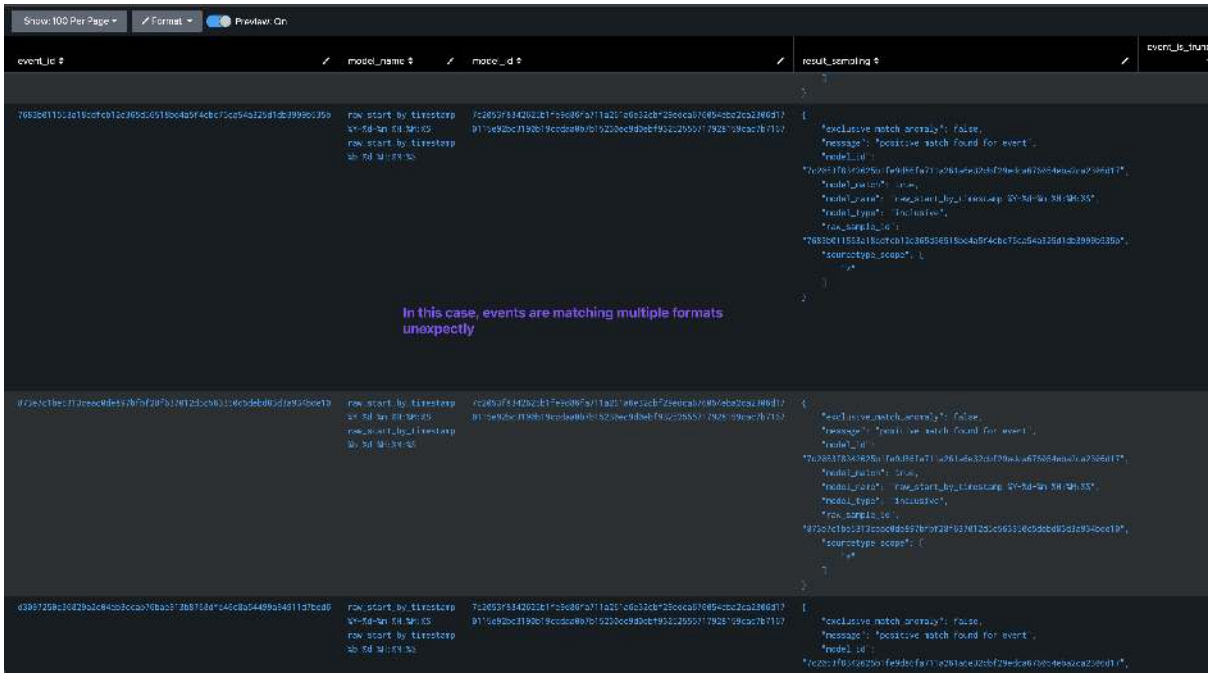
At some point, new sources are being on-boarded within the same sourcetype, however, mistakenly the expected format is not configured properly and events are making it with a different and non-expected structure:

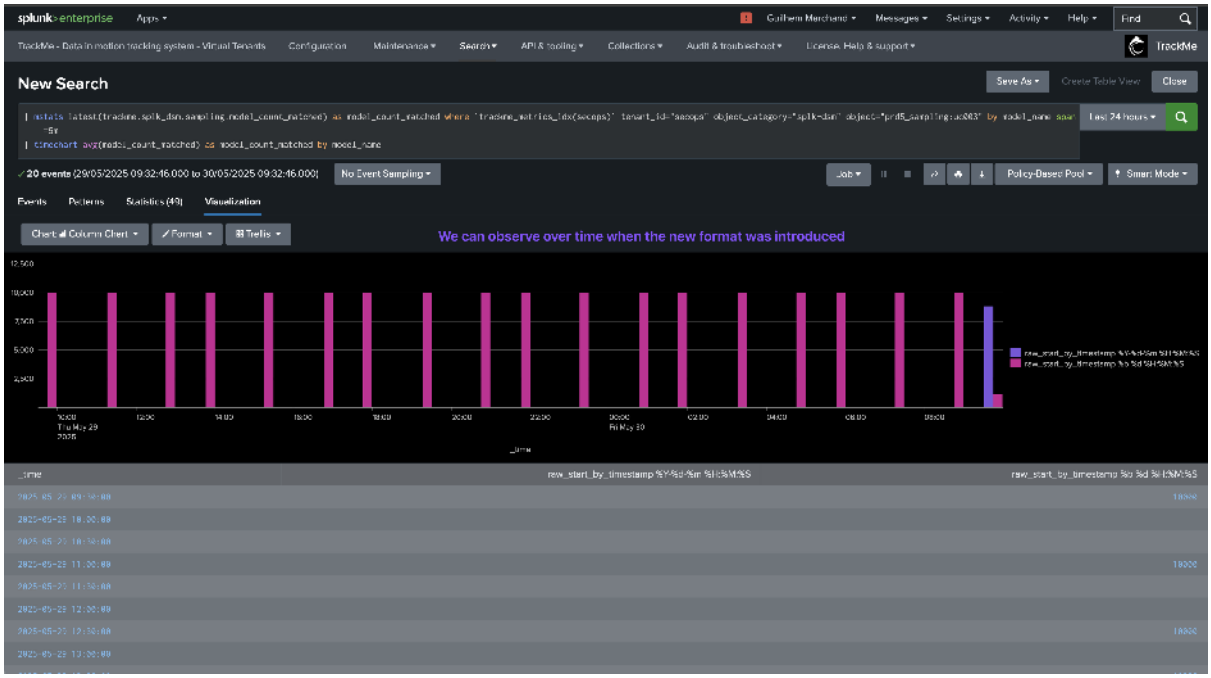


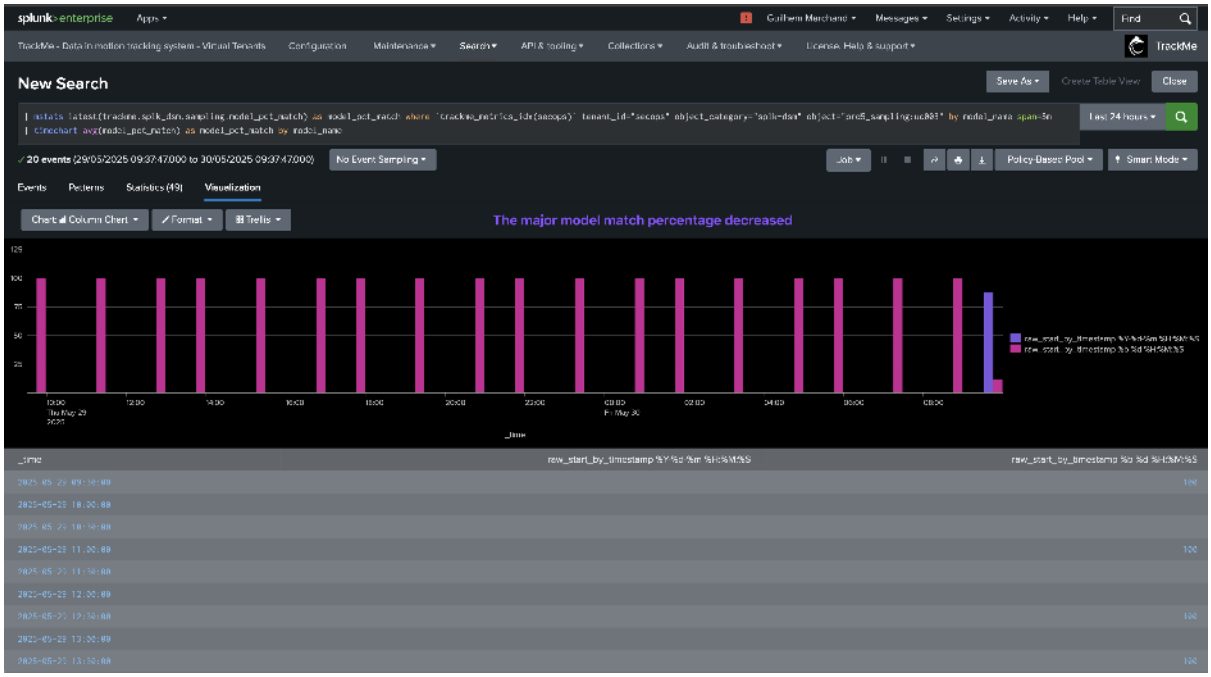
This may or may not be a strong issue, but in most use cases, receiving different types of events within the same sourcetype is not expected and leads to failures at the index time level, for instance, the timestamp extraction, or at search time with failures to extract fields for unstructured events.

The data sampling engine will detect that the format has changed and that the main format previously detected is not matching the required percentage of events, and that new event formats were introduced, leading to an alert for quality issues:

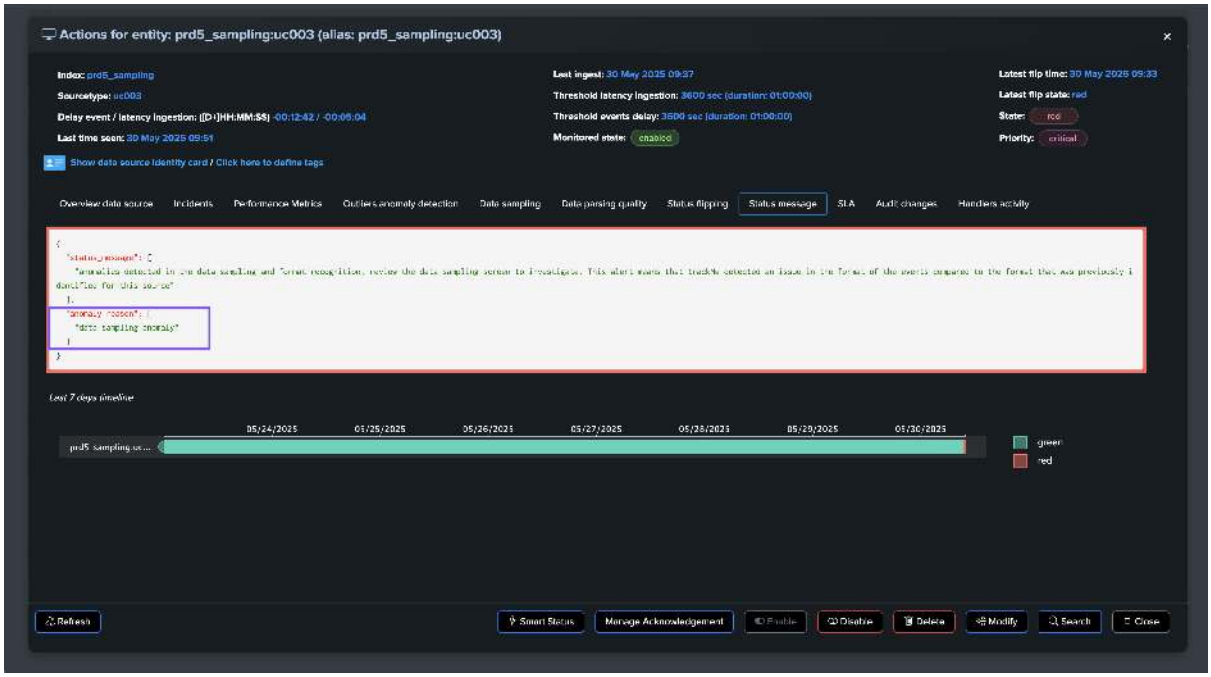



In this case, events are matching multiple formats unexpectly

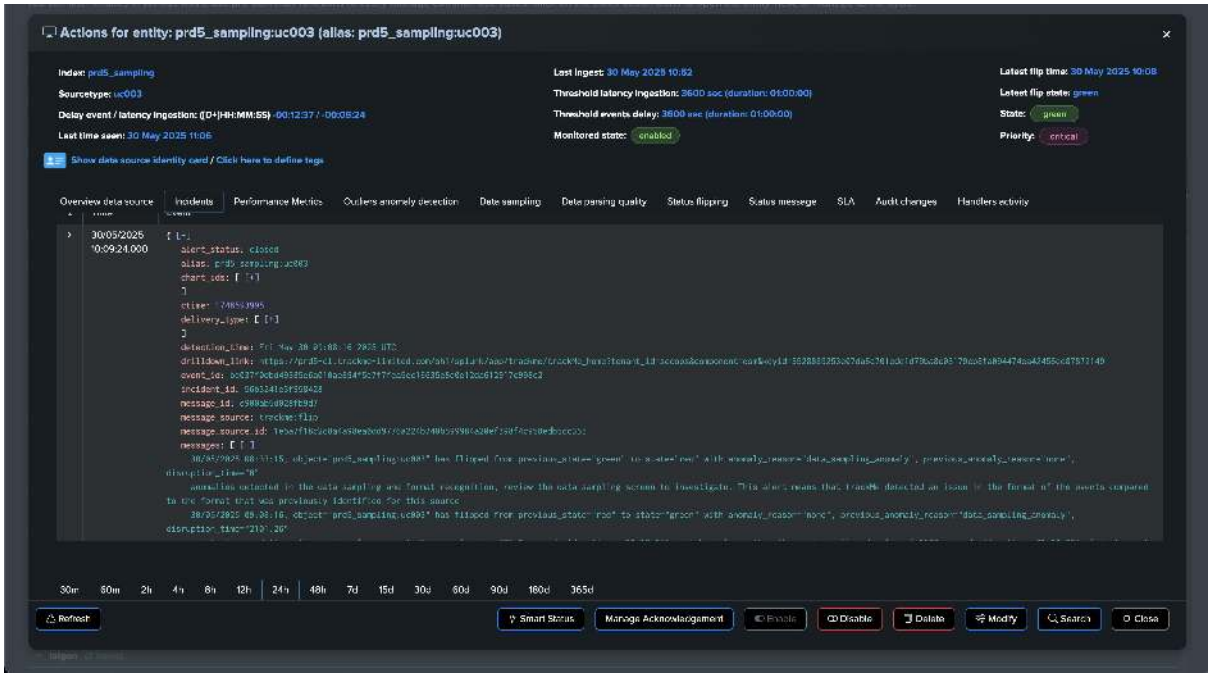

New Search
We can observe over time when the new format was introduced



The quality issue detected influences the entity status, and an anomaly_reason=”data_sampling_anomaly” is generated:



Once the root cause is identified and the issue is fixed, after some time the entity will return to green when the expected format is correctly detected again:



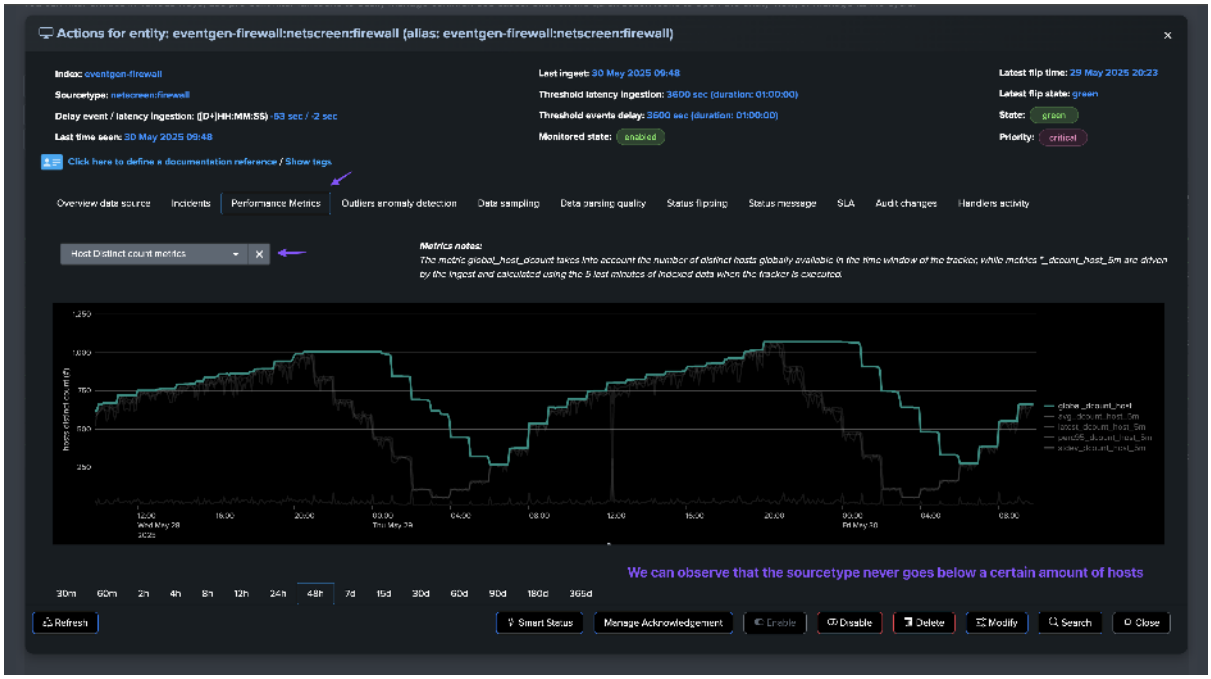
2.1.9 Use Case: Hosts Distinct Count Anomaly

Another use case is to detect anomalies in your feeds based on the amount of distinct hosts making it to Splunk.

This is a valuable use case especially at large scale that allows easily detecting issues, such as the potential loss of intermediate collectors or the potential loss of a major source.

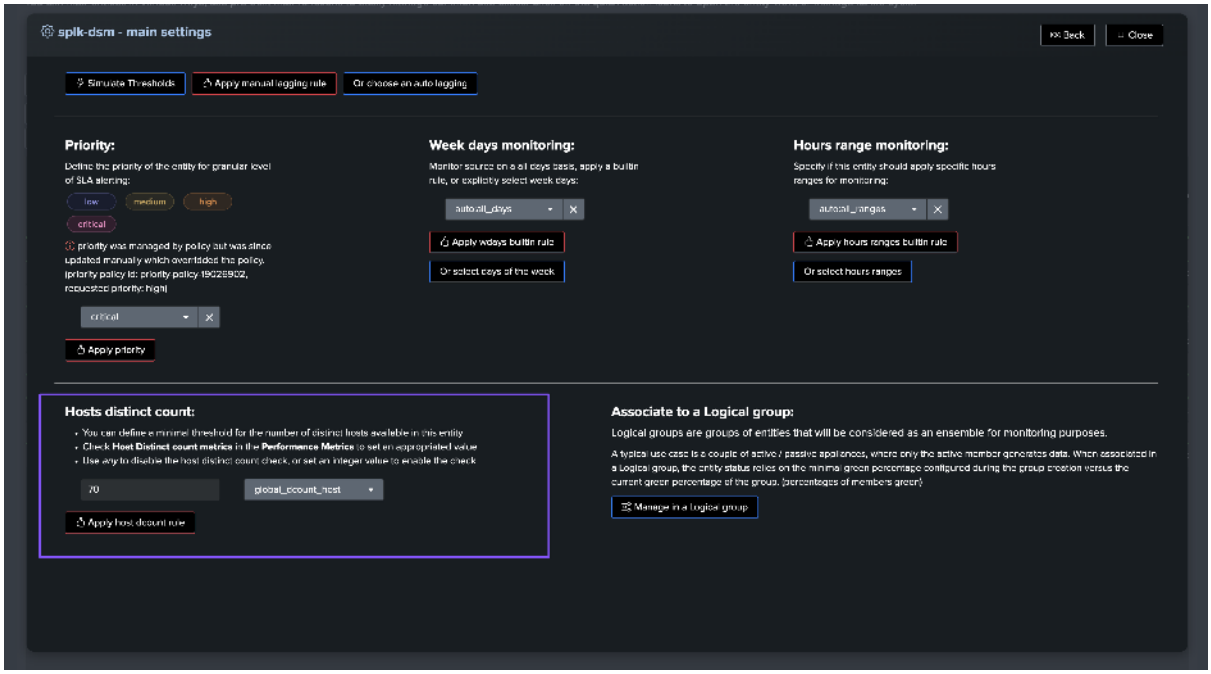
This activity can be performed using a static threshold against one of the distinct hosts metrics, as well as for more advanced use cases using Machine Learning Outliers detection.

The following example demonstrates the use of a static threshold to detect anomalies in the distinct hosts count, we can observe with the following entity that a certain amount of hosts are systematically available:

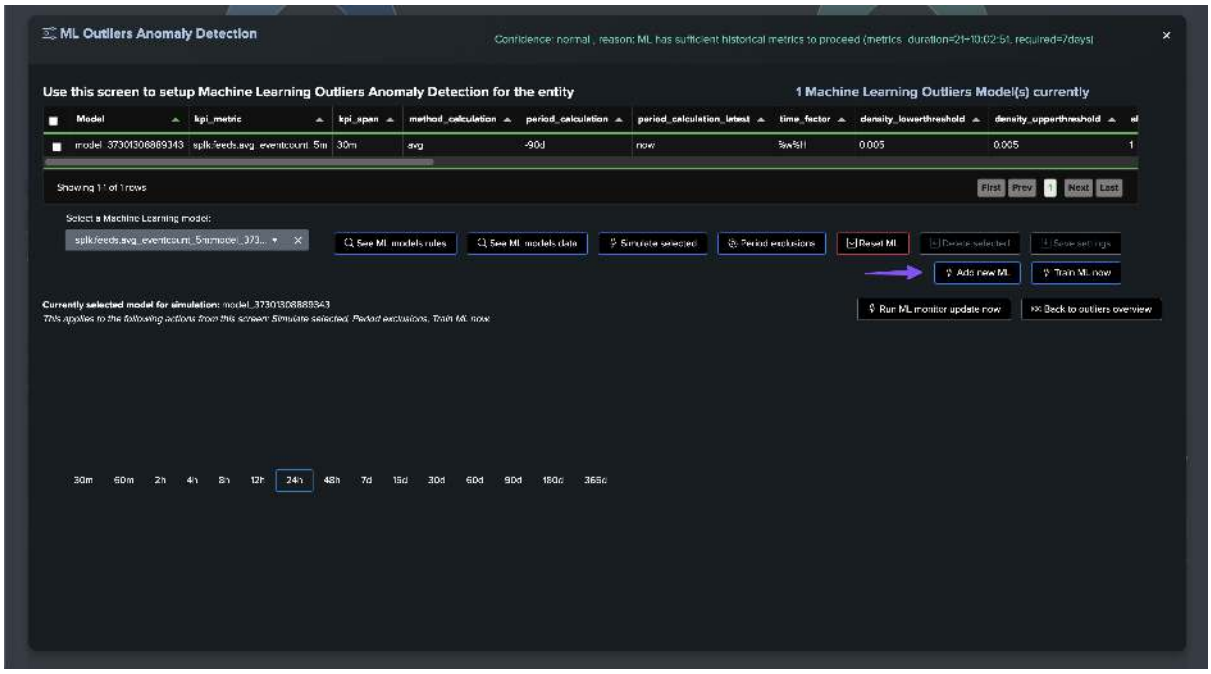


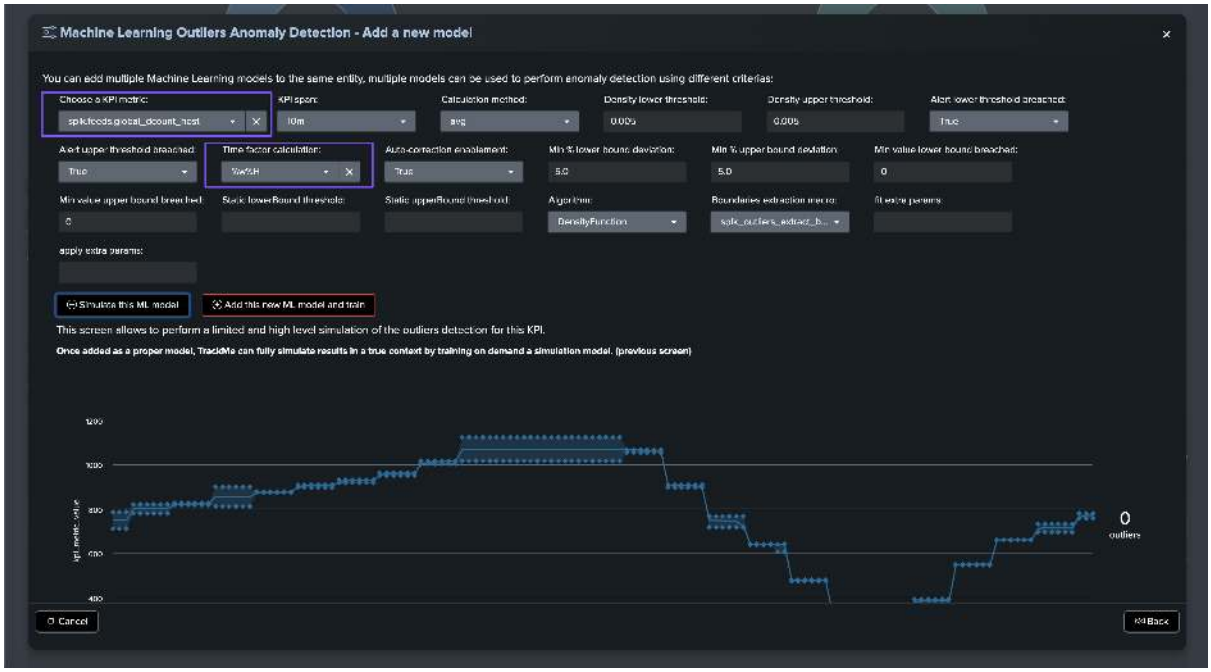
The big picture must be considered:

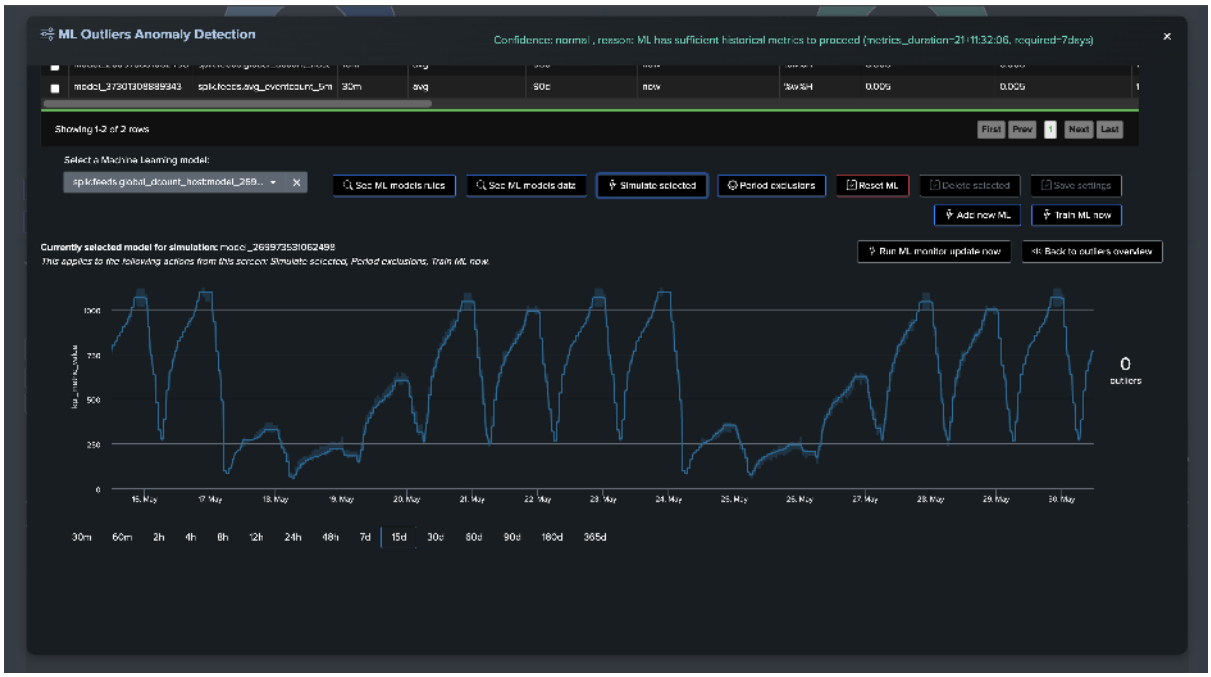
Based on this observation, we can configure a static minimal threshold for this entity, if the amount of distinct hosts goes below this threshold, the entity will be considered as in alert:



An alternative method or used in conjunction with the static threshold, is to use Machine Learning Outliers detection to detect anomalies in the distinct hosts count while taking into account time patterns:




Machine Learning Outliers Anomaly Detection - Add a new model
You can add multiple Machine Learning models to the same entity, multiple models can be used to perform anomaly detection using different criterias:
Choose a KPI metric:
KPI span:
Calculation method:
Density lower threshold:
Density upper threshold:
Alert lower threshold breached:
Alert upper threshold breached:
Time factor calculation:
Auto-correction enablement:
Min % lower bound deviation:
Min % upper bound deviation:
Min value lower bound breached:
Min value upper bound breached:
Static lowerBound threshold:
Static upperBound threshold:
Algorithm:
Boundaries extraction macro:
fit extra params:
apply extra params:
Simulate this ML model
Add this new ML model and train
This screen allows to perform a limited and high level simulation of the outliers detection for this KPI.
Once added as a proper model, TrackMe can fully simulate results in a true context by training on demand a simulation model. (previous screen)
0
outliers
Cancel
Back

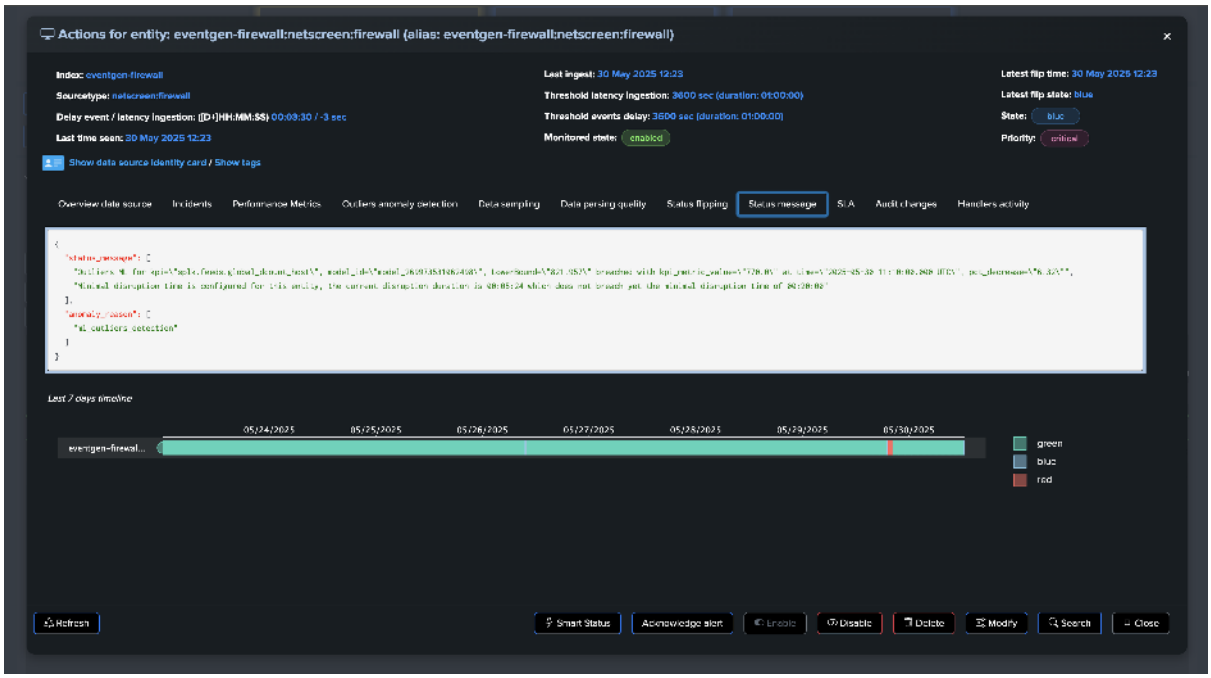

ML Outliers Anomaly Detection
Confidence: normal , reason: ML has sufficient historical metrics to proceed (metrics_duration=21:11:32:06, required=7days)
Showing 1-2 of 2 rows
First
Prev
Next
Last
Select a Machine Learning model:
See ML models rules
See ML models data
Simulate selected
Period exclusions
Reset ML
Delete selected
Save settings
Add new ML
Train ML now
Currently selected model for simulation:
This applies to the following actions from this screen: Simulate selected, Period exclusions, Train ML now
Run ML monitor update now
Back to outliers overview
0
outliers
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
180d
365d

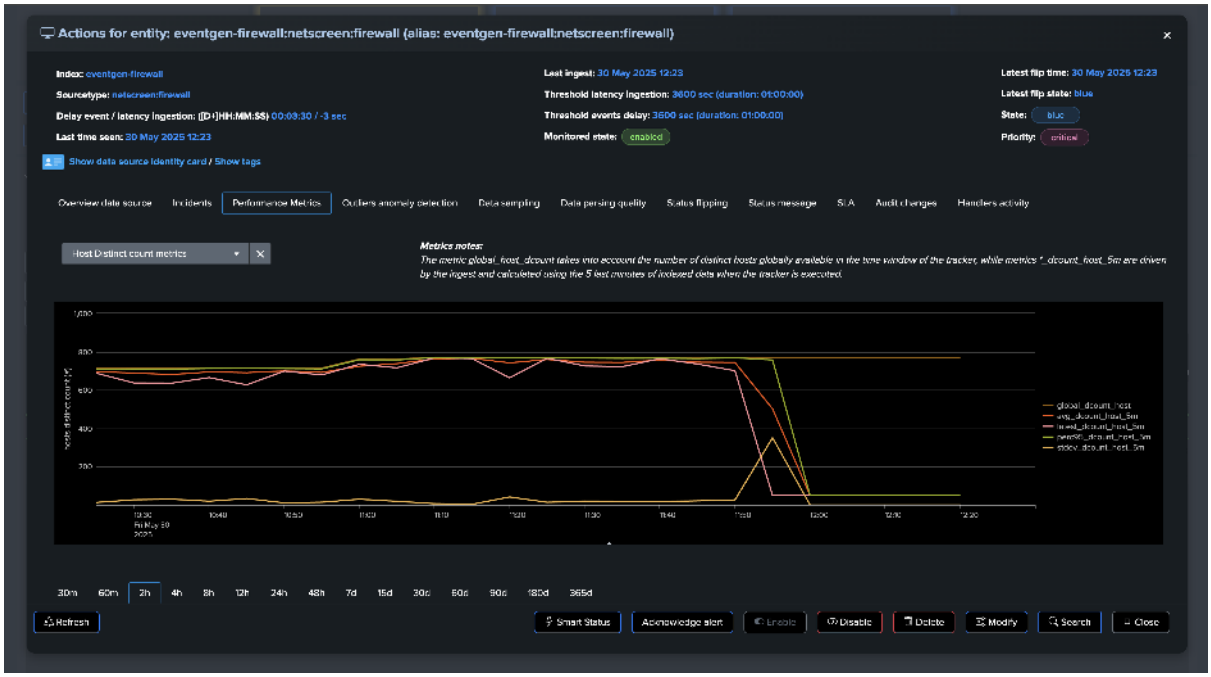
If at some point, the amount of distinct hosts goes below the threshold, the entity will be considered as in anomaly, and an incident will be opened:

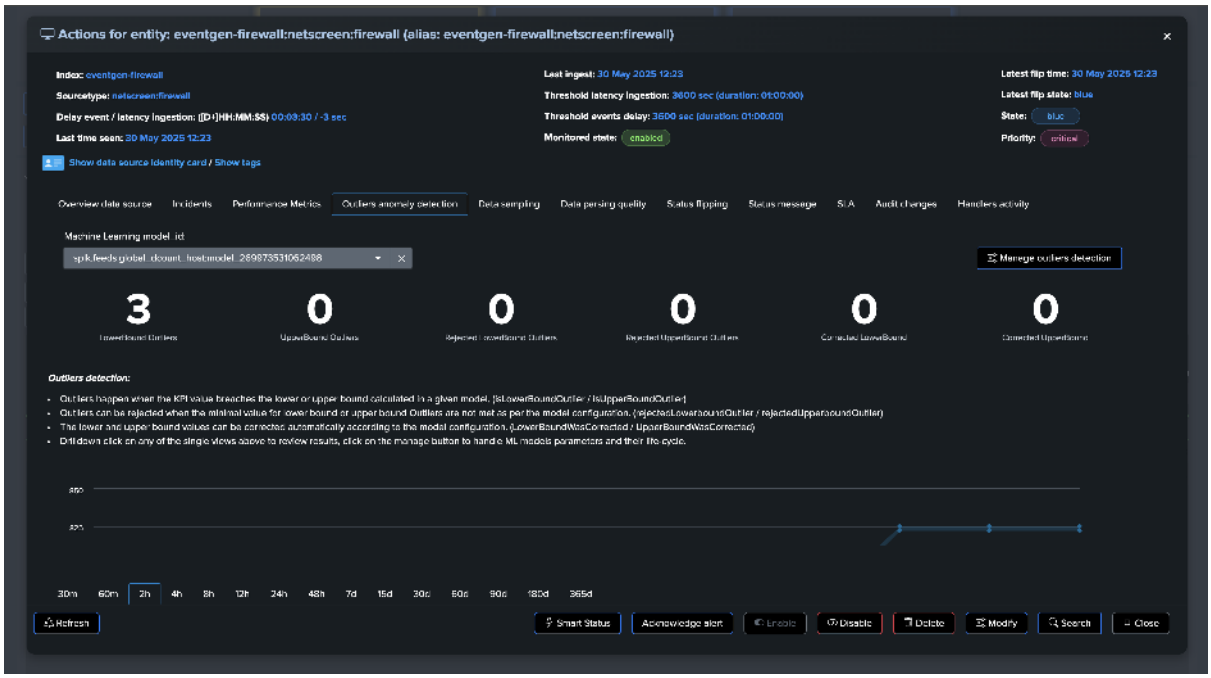
Notes:

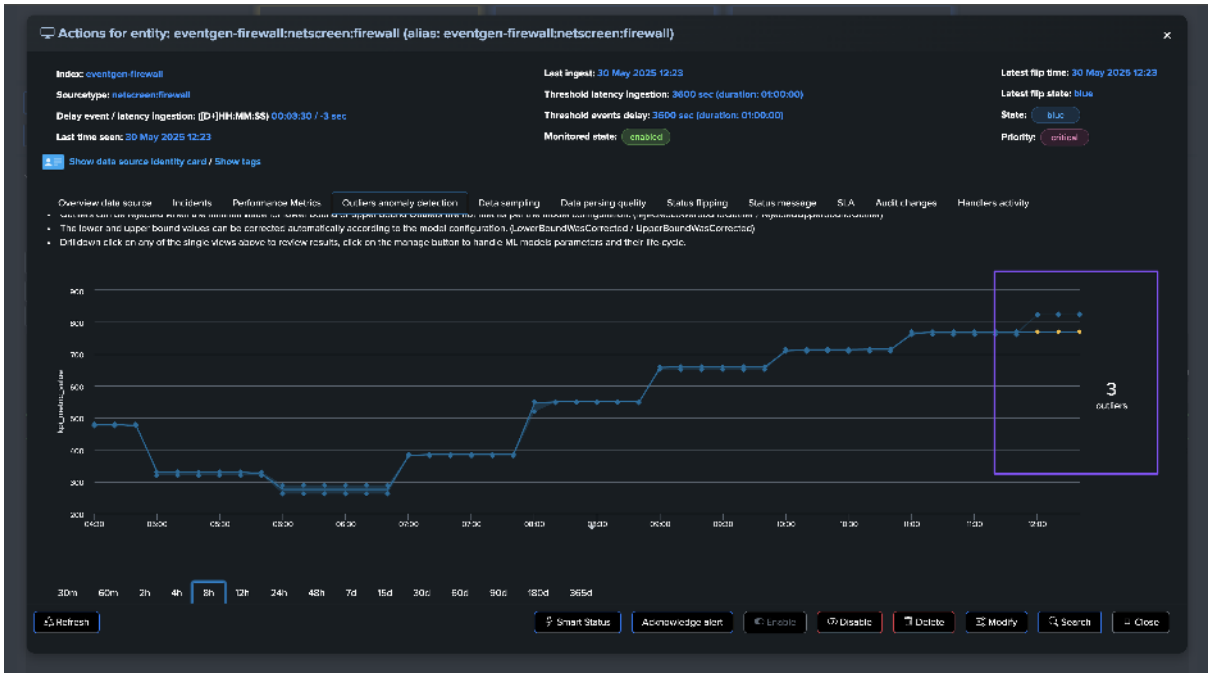
- The "global" distinct count KPI measures the number of unique hosts across the entire tracker period. Due to this global aggregation, there may be a delay in detecting anomalies as the system needs to accumulate enough data to establish a clear pattern.
- A decrease in the number of distinct hosts often correlates with a decrease in event volume. In such cases, TrackMe's Machine Learning Outliers detection may trigger alerts for both volume-based and distinct hosts models simultaneously, providing multiple indicators of the same underlying issue.

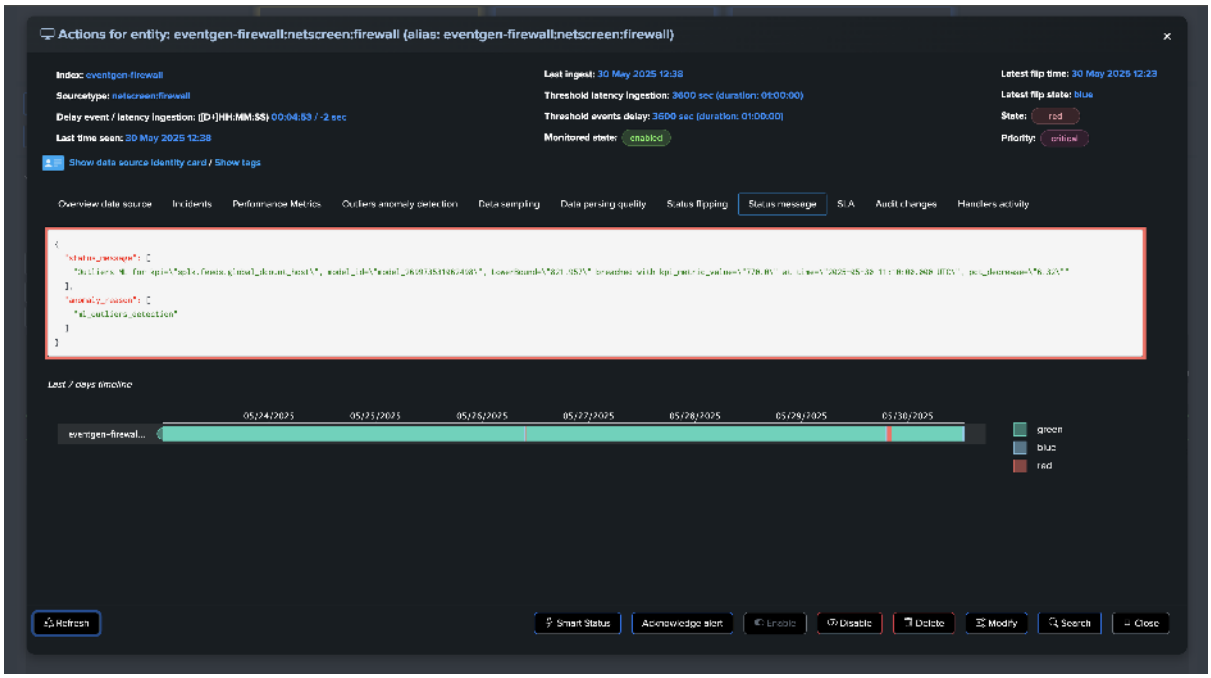
For the purposes of this demo, we have set both the static threshold and Machine Learning Outliers detection. After some time, both detection methods trigger:

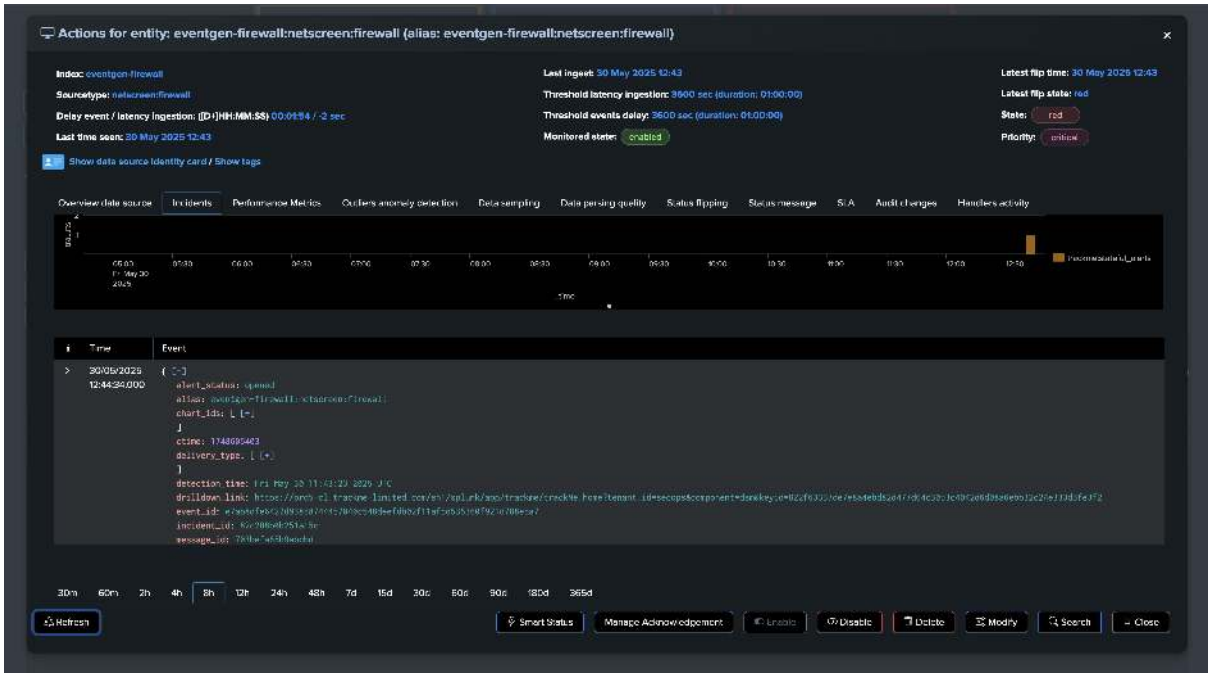



Actions for entity: eventgen-firewall:netscreen:firewall (alias: eventgen-firewall:netscreen:firewall)
Performance Metrics
Host Distinct count metrics
Metrics notes:
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
180d
365d
Refresh
Smart Status
Acknowledge alert
Enable
Disable
Delete
Modify
Search
Close


Actions for entity: eventgen-firewall:netscreen:firewall (alias: eventgen-firewall:netscreen:firewall)
Outliers anomaly detection
Machine Learning model id
Manage outliers detection
3
0
0
0
0
0
Outliers detection:
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
180d
365d
Refresh
Smart Status
Acknowledge alert
Enable
Disable
Delete
Modify
Search
Close

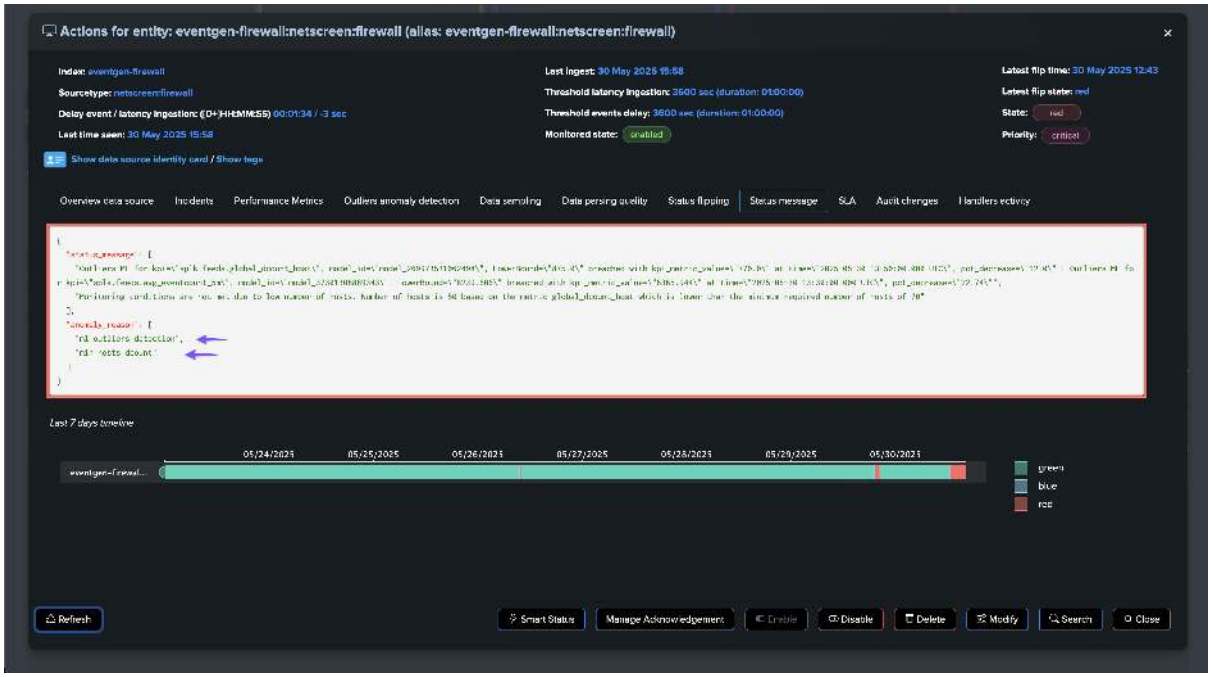

Actions for entity: eventgen-firewall:netscreen:firewall (alias: eventgen-firewall:netscreen:firewall)
Index: eventgen-firewall
Sourcetype: netscreen:firewall
Delay event / latency ingestion: [[D+]HH:MM:SS] 00:00:30 / -3 sec
Last time seen: 30 May 2025 12:23
Last ingest: 30 May 2025 12:23
Threshold latency ingestion: 3600 sec (duration: 01:00:00)
Threshold events delay: 3600 sec (duration: 01:00:00)
Monitored state: enabled
Latest flip time: 30 May 2025 12:23
Latest flip state: blue
State: blue
Priority: critical
Show data source identity card / Show tags
Overview data source
Incidents
Performance Metrics
Outliers anomaly detection
Data sampling
Data parsing quality
Status flipping
Status message
SLA
Audit changes
Handlers activity
The lower and upper bound values can be corrected automatically according to the model configuration. (LowerBoundWasCorrected / UpperBoundWasCorrected)
Drilldown: click on any of the single views above to review results, click on the manage button to handle ML models parameters and their life cycle.
3
outliers
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
180d
365d
Refresh
Smart Status
Acknowledge alert
Enable
Disable
Delete
Modify
Search
Close


Actions for entity: eventgen-firewall:netscreen:firewall (alias: eventgen-firewall:netscreen:firewall)
Index: eventgen-firewall
Sourcetype: netscreen:firewall
Delay event / latency ingestion: [[D+]HH:MM:SS] 00:04:53 / -2 sec
Last time seen: 30 May 2025 12:38
Last ingest: 30 May 2025 12:38
Threshold latency ingestion: 3600 sec (duration: 01:00:00)
Threshold events delay: 3600 sec (duration: 01:00:00)
Monitored state: enabled
Latest flip time: 30 May 2025 12:23
Latest flip state: blue
State: red
Priority: critical
Show data source identity card / Show tags
Overview data source
Incidents
Performance Metrics
Outliers anomaly detection
Data sampling
Data parsing quality
Status flipping
Status message
SLA
Audit changes
Handlers activity
"status_message": [
"anomaly_reason": [
"ml_outliers_detection"
Last 7 days timeline
05/24/2025
05/25/2025
05/26/2025
05/27/2025
05/28/2025
05/29/2025
05/30/2025
eventgen-firewall...
green
blue
red
Refresh
Smart Status
Acknowledge alert
Enable
Disable
Delete
Modify
Search
Close

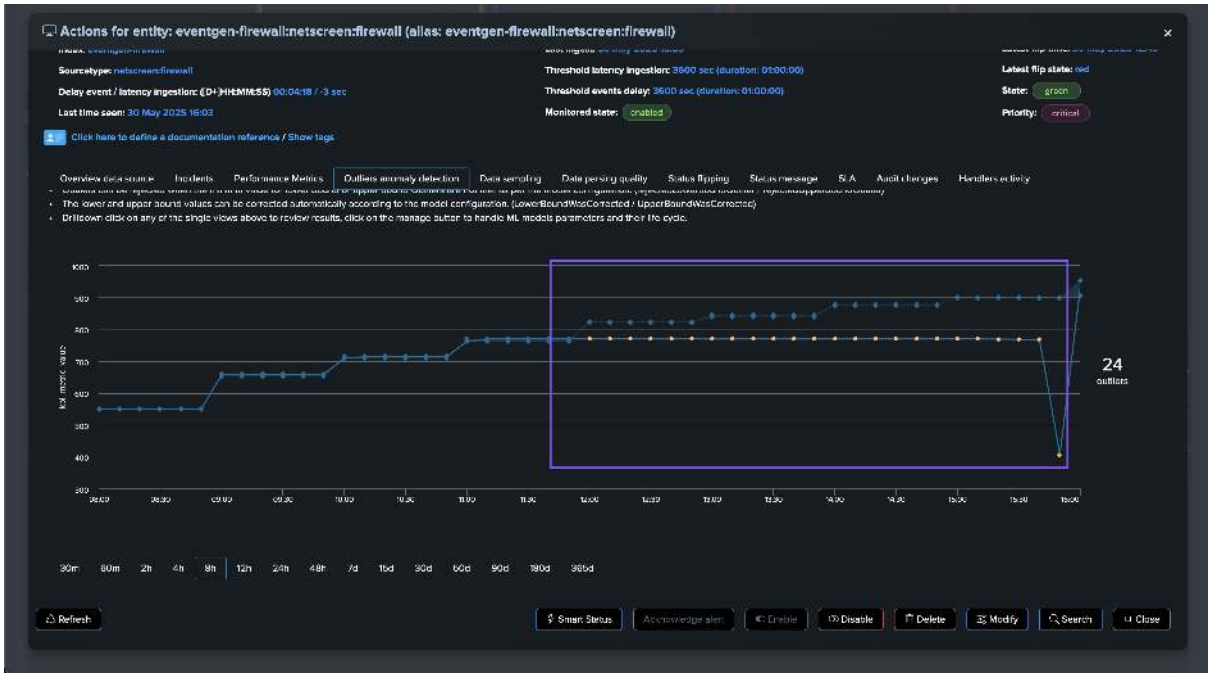

Actions for entity: eventgen-firewall:netscreen:firewall (alias: eventgen-firewall:netscreen:firewall)
Index: eventgen-firewall
Sourcetype: netscreen:firewall
Delay event / latency ingestion: [[D+]HH:MM:SS] 00:01:14 / -2 sec
Last time seen: 30 May 2025 12:43
Show data source identity card / Show tags
Last ingest: 30 May 2025 12:43
Threshold latency ingestion: 3600 sec (duration: 01:00:00)
Threshold events delay: 3600 sec (duration: 01:00:00)
Monitored state: enabled
Latest flip time: 30 May 2025 12:43
Latest flip state: red
State: red
Priority:
Overview data source
Incidents
Performance Metrics
Outliers anomaly detection
Data sampling
Data parsing quality
Status flipping
Status message
SLA
Audit changes
Handlers activity
Time
Event
30/05/2025 12:44:34.000
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
180d
365d
Refresh
Smart Status
Manage Acknowledgement
Enable
Disable
Delete
Modify
Search
Close

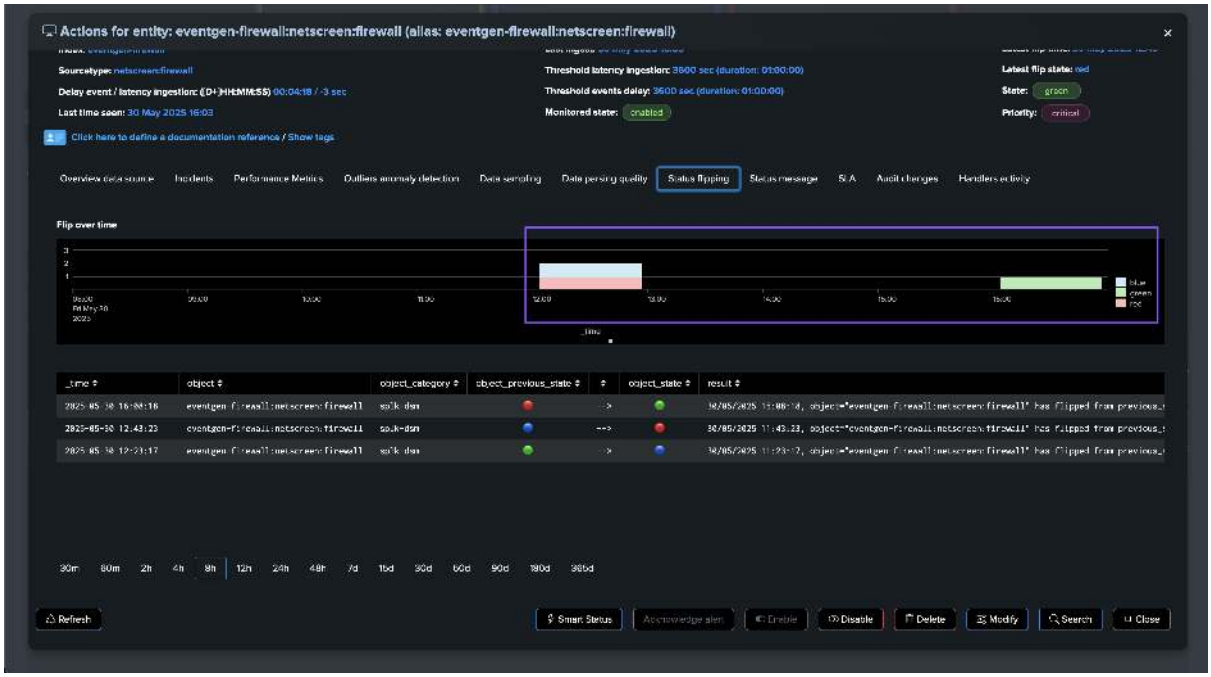
TrackMe Notification
Environment: https://prd5-cl.trackme-limited.com
Detection Time: Fri May 30 11:43:23 2025 UTC
Alert Status: opened
Tenant: secops
Alias: eventgen-firewall:netscreen:firewall
Object: eventgen-firewall:netscreen:firewall
Drilldown Link
Status: red
Incident ID: 82c208b0b251af5c
Detailed Information:
The entity has transitioned to an alerting state:
Alias: eventgen-firewall:netscreen:firewall
Object: eventgen-firewall:netscreen:firewall
Object ID: 822f63337de7e8a4ebd82d477d64c3053c4042d6e88a6ebb32c24e333d3fe3f2
Category: splk-dsm
Anomaly Reason(s):
ml_outliers_detection
Message(s):
30/05/2025 11:43:23, object="eventgen-firewall:netscreen:firewall" has "flipped from previous_state="blue" to state="red" with anomaly_reason="ml_outliers_detection", previous_anomaly_reason="ml_outliers_detection", disruption_time="0"
Outliers ML for kpi="splk.feeds.global_dcount_host", model_id="model_2899735310624 98", LowerBound="821.957" breached with kpi_metric_value="770.0" at time="2025-05-30 11:10:00.000 UTC", pct_decrease="6.32"
Message source: trackme:flip
Message source ID: bccef7332b01c646b0d5ef4ca9433f2a2abf8b3e3eaf86a1d269975a8e402237

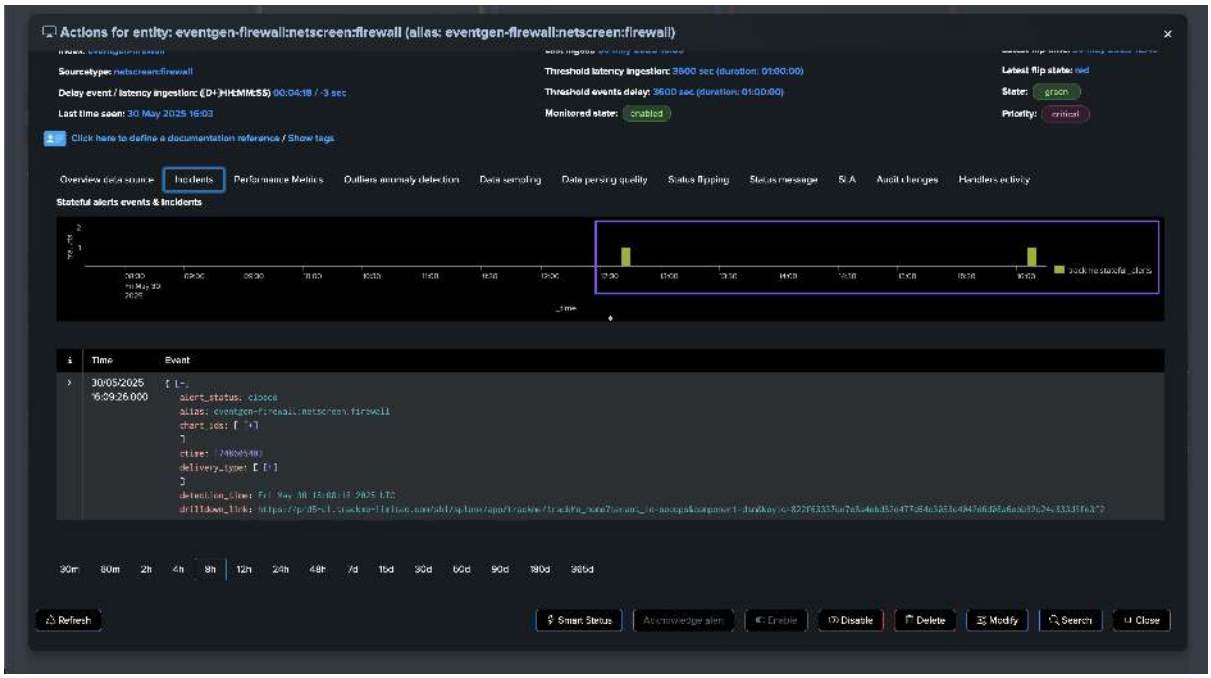
Performance Charts (24h)
Latency Metrics - Last 24h
Delay Metrics - Last 24h
Volume Metrics - Last 24h
Hosts DCount Metrics - Last 24h
Machine Learning Outliers - Last 24h
This is an automated email, please do not reply directly to this email.

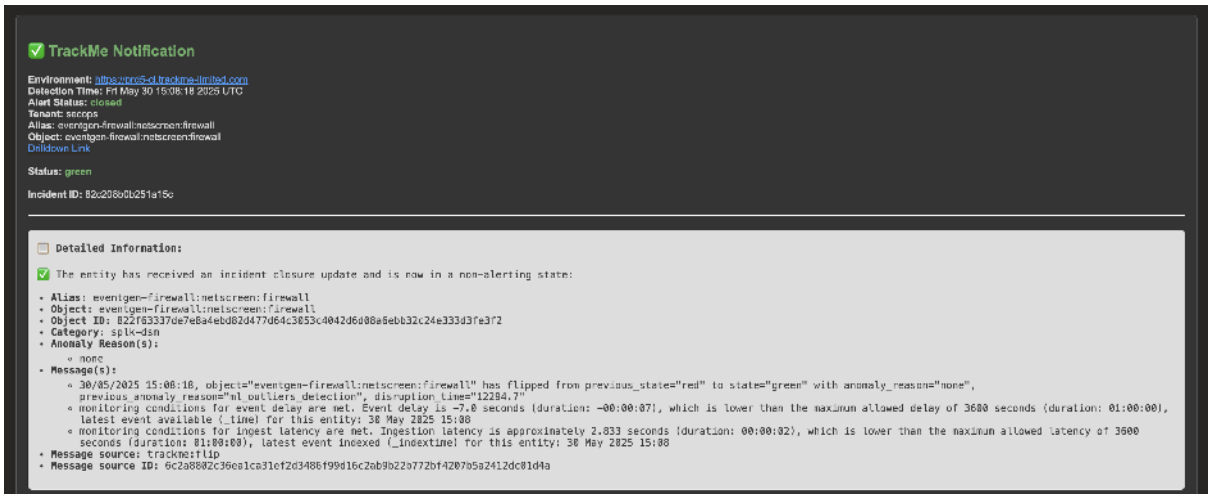


As usual, once the issue is resolved, the entity returns to green after the Outliers anomaly condition is resolved and TrackMe has processed the ML rendering for this entity:




Actions for entity: eventgen-firewall:netscreen:firewall (alias: eventgen-firewall:netscreen:firewall)
Sourcetype: netscreen:firewall
Delay event / latency ingestion: (D+)HH:MM:SS) 00:04:18 / -3 sec
Last time seen: 30 May 2025 16:03
Threshold latency ingestion: 3600 sec (duration: 01:00:00)
Threshold events delay: 3600 sec (duration: 01:00:00)
Monitored state: enabled
Latest flip state: red
State: green
Priority: critical
Click here to define a documentation reference / Show tags
Overview data source
Incidents
Performance Metrics
Outliers anomaly detection
Data sampling
Data parsing quality
Status flipping
Status message
SLA
Audit changes
Handlers activity
Flip over time
_time
object_category
object_previous_state
object_state
result
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
180d
365d
Refresh
Smart Status
Acknowledge alert
Enable
Disable
Delete
Modify
Search
Close


Actions for entity: eventgen-firewall:netscreen:firewall (alias: eventgen-firewall:netscreen:firewall)
Sourcetype: netscreen:firewall
Delay event / latency ingestion: (D+)HH:MM:SS) 00:04:18 / -3 sec
Last time seen: 30 May 2025 16:03
Threshold latency ingestion: 3600 sec (duration: 01:00:00)
Threshold events delay: 3600 sec (duration: 01:00:00)
Monitored state: enabled
Latest flip state: red
State: green
Priority: critical
Click here to define a documentation reference / Show tags
Overview data source
Incidents
Performance Metrics
Outliers anomaly detection
Data sampling
Data parsing quality
Status flipping
Status message
SLA
Audit changes
Handlers activity
Stateful alerts events & incidents
Time
Event
30/05/2025 16:09:26.000
alert_status: closed
alias: eventgen-firewall:netscreen:firewall
chart_ids: [1]
ctime: 1748620140
delivery_type: [0]
detection_time: Fri May 30 15:08:18 2025 UTC
drilldown_link:
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
180d
365d
Refresh
Smart Status
Acknowledge alert
Enable
Disable
Delete
Modify
Search
Close


TrackMe Notification
Environment: https://prd5-cl.trackme-limited.com
Detection Time: Fri May 30 15:08:18 2025 UTC
Alert Status: closed
Tenant: secops
Alias: eventgen-firewall:netscreen:firewall
Object: eventgen-firewall:netscreen:firewall
Drilldown Link
Status: green
Incident ID: 82c208b0b251a15c
Detailed Information:
The entity has received an incident closure update and is now in a non-alerting state:
Alias: eventgen-firewall:netscreen:firewall
Object: eventgen-firewall:netscreen:firewall
Object ID: 822f63337de7e8a4ebd82d477d64c3053c4042d6d08a8ebb32c24e333d3fe3f2
Category: splk-dsn
Anomaly Reason(s):
none
Message(s):
30/05/2025 15:00:10, object="eventgen-firewall:netscreen:firewall" has flipped from previous_state="red" to state="green" with anomaly_reason="none", previous anomaly_reason="ml_outliers_detection", disruption_time="12284.7"
monitoring conditions for event delay are met. Event delay is -7.0 seconds (duration: -00:00:07), which is lower than the maximum allowed delay of 3600 seconds (duration: 01:00:00), latest event available (_time) for this entity: 30 May 2025 15:08
monitoring conditions for ingest latency are met. Ingestion latency is approximately 2.833 seconds (duration: 00:00:02), which is lower than the maximum allowed latency of 3600 seconds (duration: 01:00:00), latest event indexed (_indextime) for this entity: 30 May 2025 15:08
Message source: trackmeflip
Message source ID: 6c2a8882c36ea1ca31ef2d3486f99d16c2ab9b22a772bf4207b5a2412dc01d4a

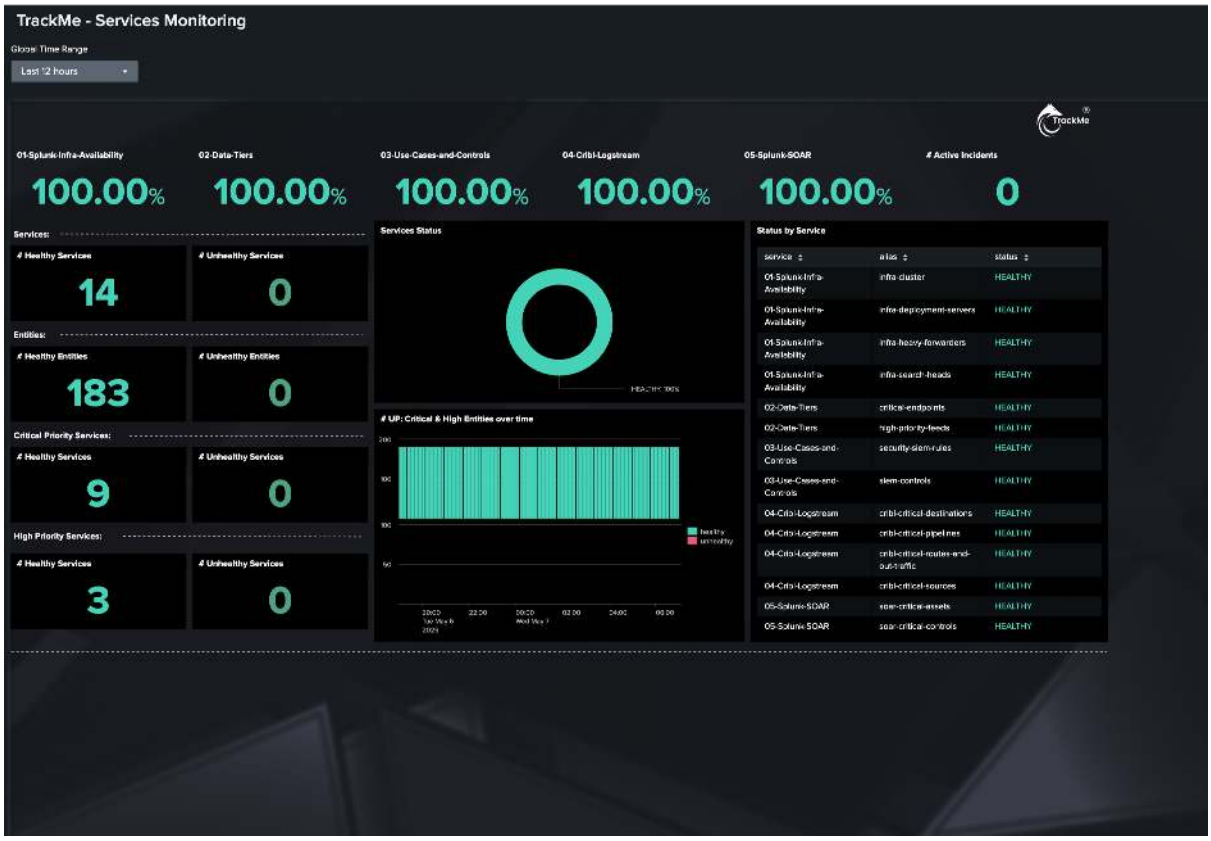
2.2 Use Case Demo: 360 Services Monitoring with TrackMe

Use Case Demo: 360 Services Monitoring with TrackMe

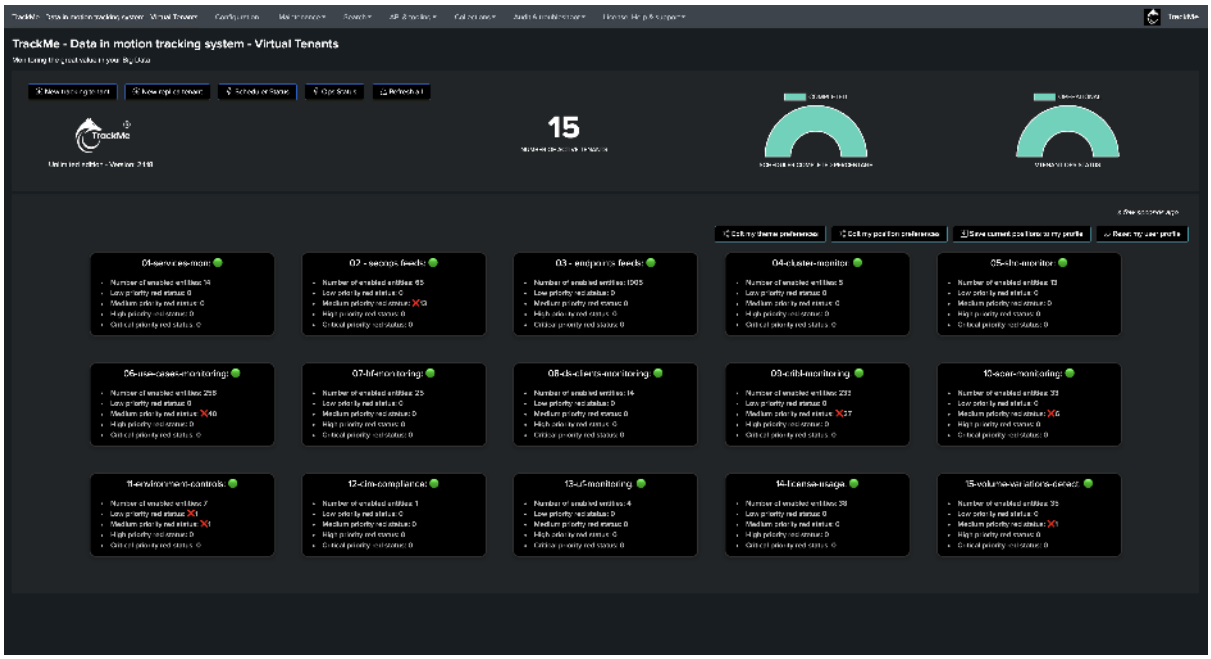
- This use case demo demonstrates how TrackMe can be used to perform a **360 degree monitoring** of the different services that are commonly composing Splunk environments, with addition of third parties notably Cribl Logstream.
- **The purpose of this demo is to show step by step how to design and implement TrackMe concepts and features, notably:**
  - `Data Tiers:` Monitoring high priority Splunk Feeds availability and performance using TrackMe component `splk-dsm`.
  - `Data Tiers:` Monitoring high priority endpoints availability (Think about Active Directory domain controllers, Checkpoint firewalls, etc.) using TrackMe component `splk-dsm`.
  - `Data Tiers:` Monitoring abnormal volume variations in Splunk indexes and Splunk license usage using TrackMe component `splk-flx`.
  - `Splunk Tiers:` Monitoring key aspects and metrics of Splunk Indexers Cluster using TrackMe component `splk-flx`.
  - `Splunk Tiers:` Monitoring key aspects and metrics of Splunk Search Head Cluster or Search Heads using TrackMe component `splk-flx`.
  - `Splunk Tiers:` Monitoring key aspects of Splunk Heavy Forwarder tiers using TrackMe component `splk-flx`.
  - `Splunk Tiers:` Monitoring Splunk deployment servers and clients using TrackMe component `splk-flx`.
  - `Use Cases & Controls:` Monitoring Splunk core & Splunk Enterprise Security use cases using TrackMe Workload component `splk-wlk`.
  - `Use Cases & Controls:` Monitoring various environments control points using TrackMe component `splk-flx`.
  - `Cribl Logstream:` Monitoring Cribl Logstream availability and performance using TrackMe component `splk-flx`.
  - `Splunk SOAR tier:` Monitoring Splunk SOAR platforms using TrackMe component `splk-flx`.
- Some of the components leveraged in this demo are restricted features available in TrackMe Enterprise Edition & Unlimited Edition.
- **This demo documentation is currently a work in progress** and will be updated in the future to reflect the latest features and capabilities of TrackMe.

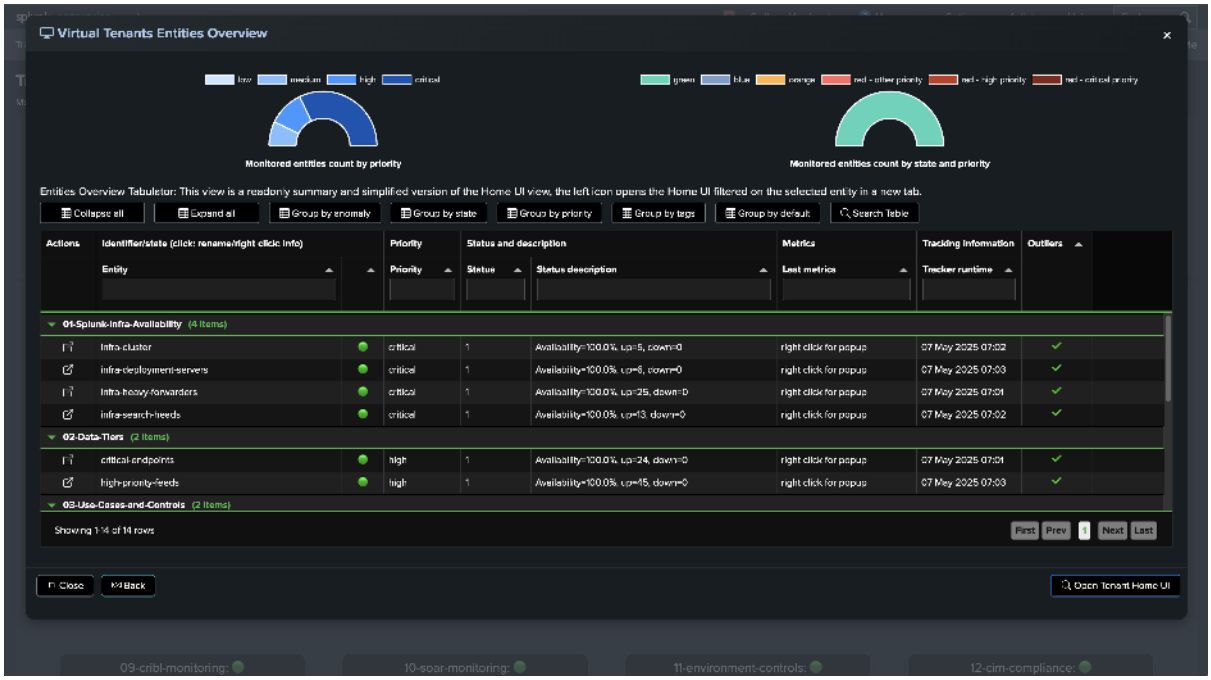
2.2.1 Pictures Gallery

The following image shows a template Splunk dashboard which calls TrackMe Flex converging entities, which transparently correlate the status of TrackMe entities to form the representation of the different tiers in the environment: (find this template in the API & Tooling menu, from TrackMe 2.1.18)

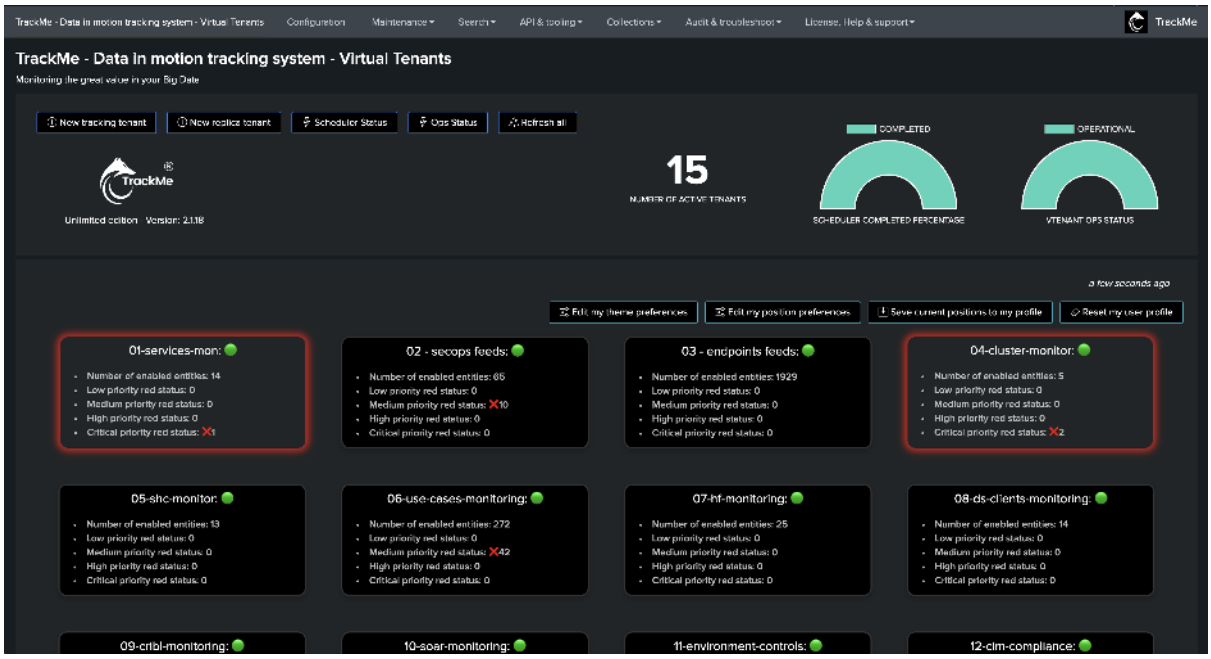


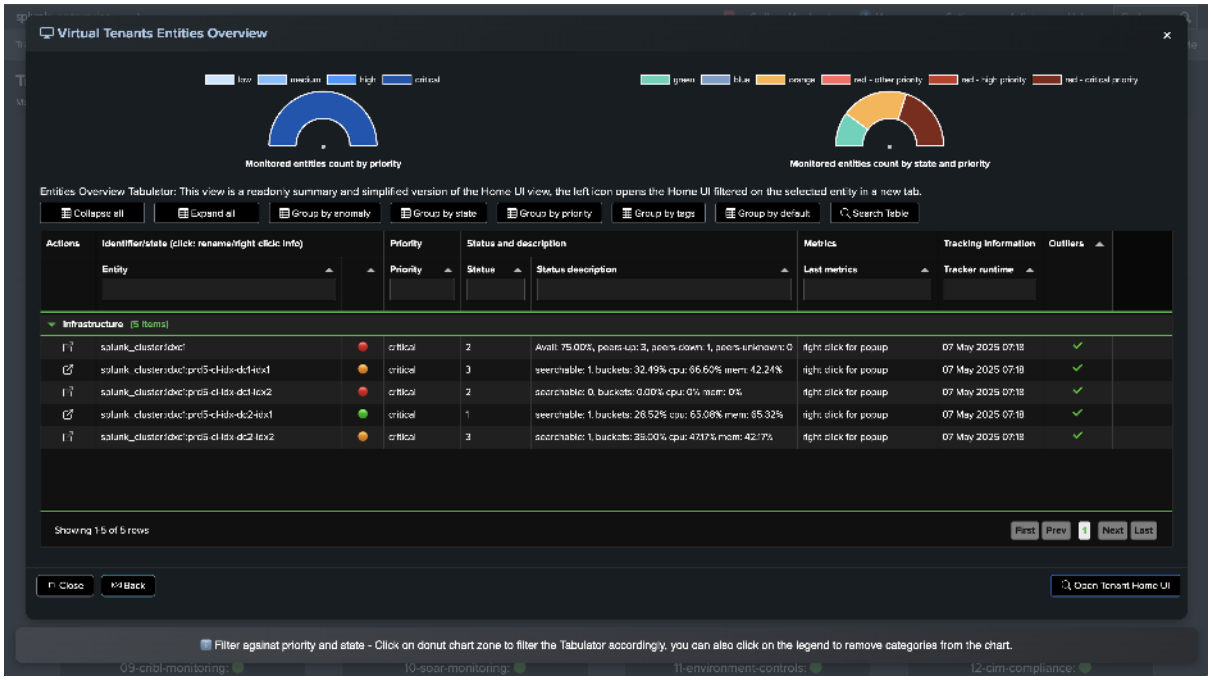
The following images show TrackMe Virtual Tenants Home page:



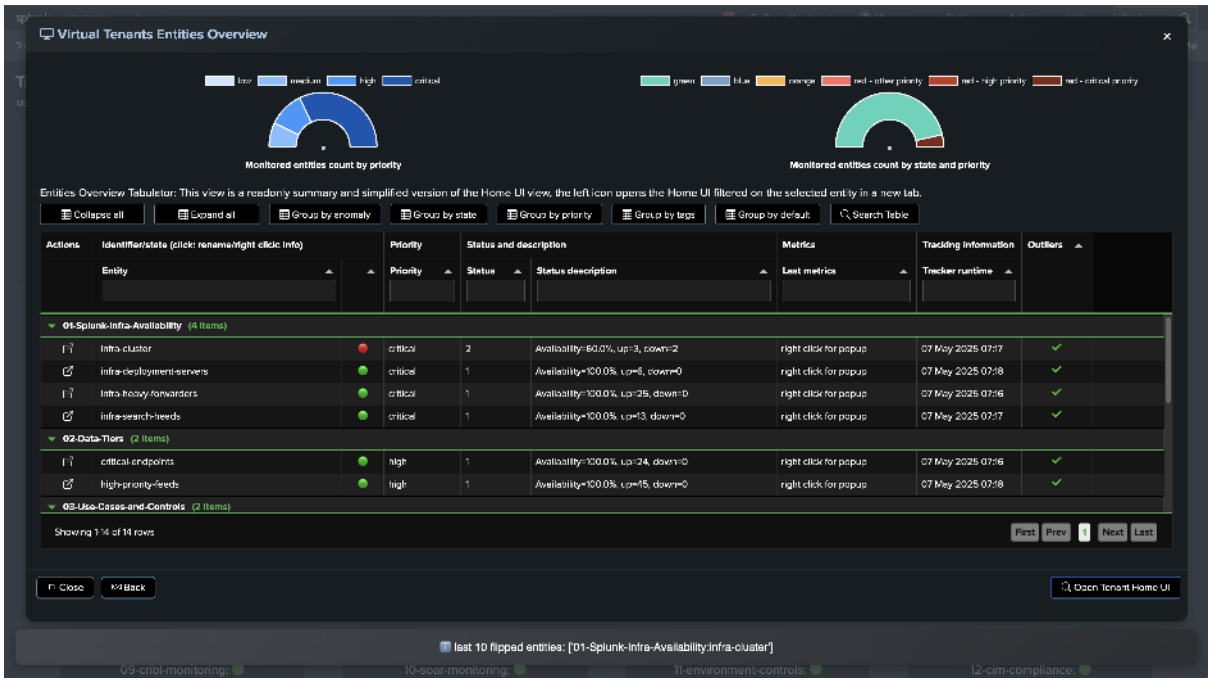


An incident is affecting the Splunk Indexers Cluster tier:

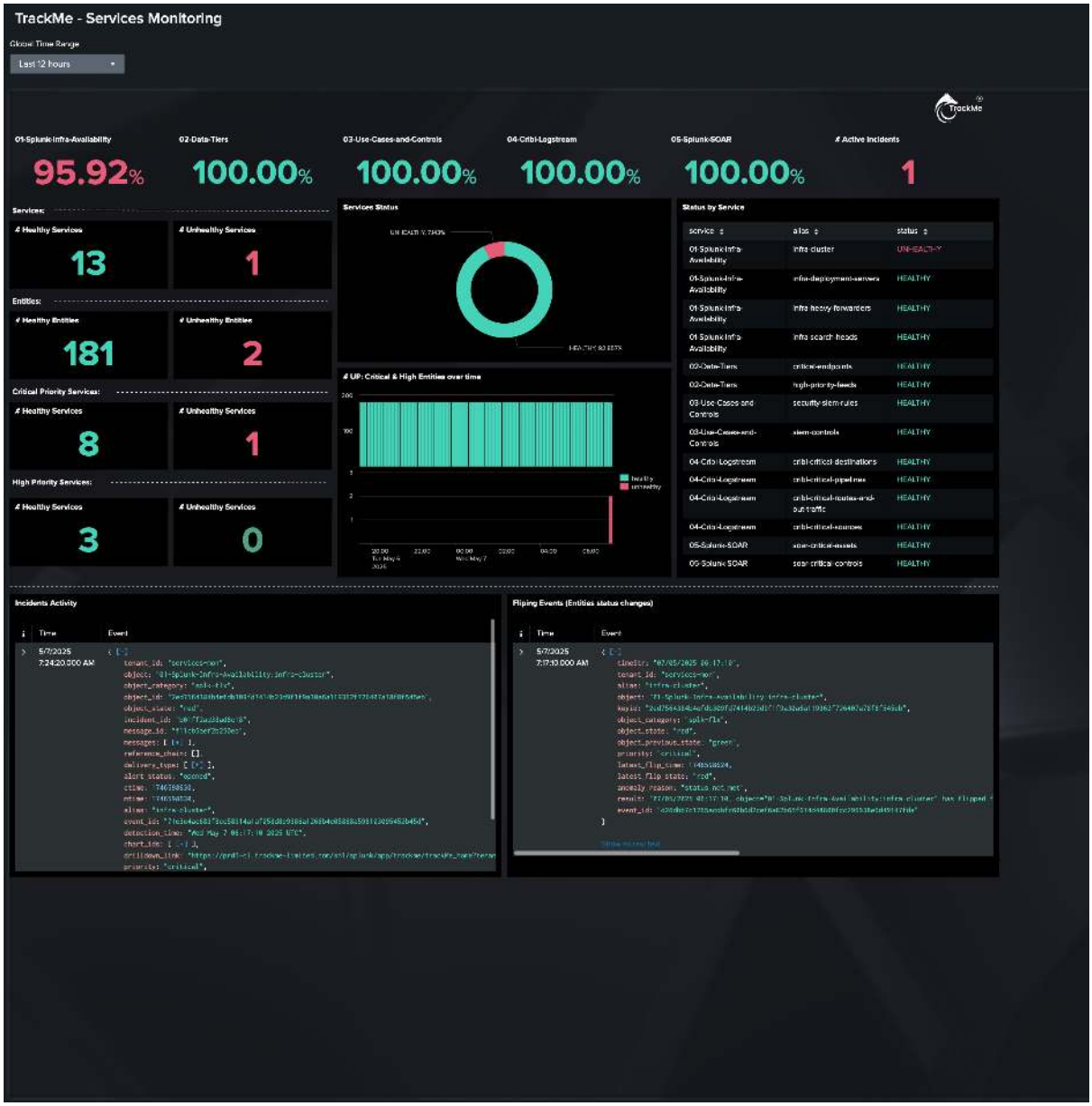




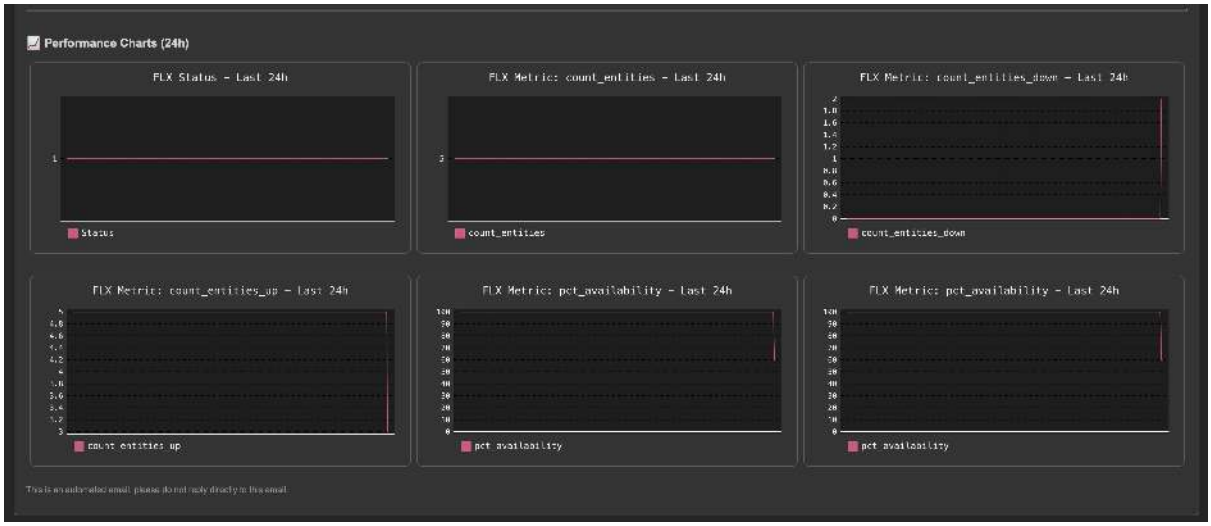
The Services Monitoring Virtual Tenant view:



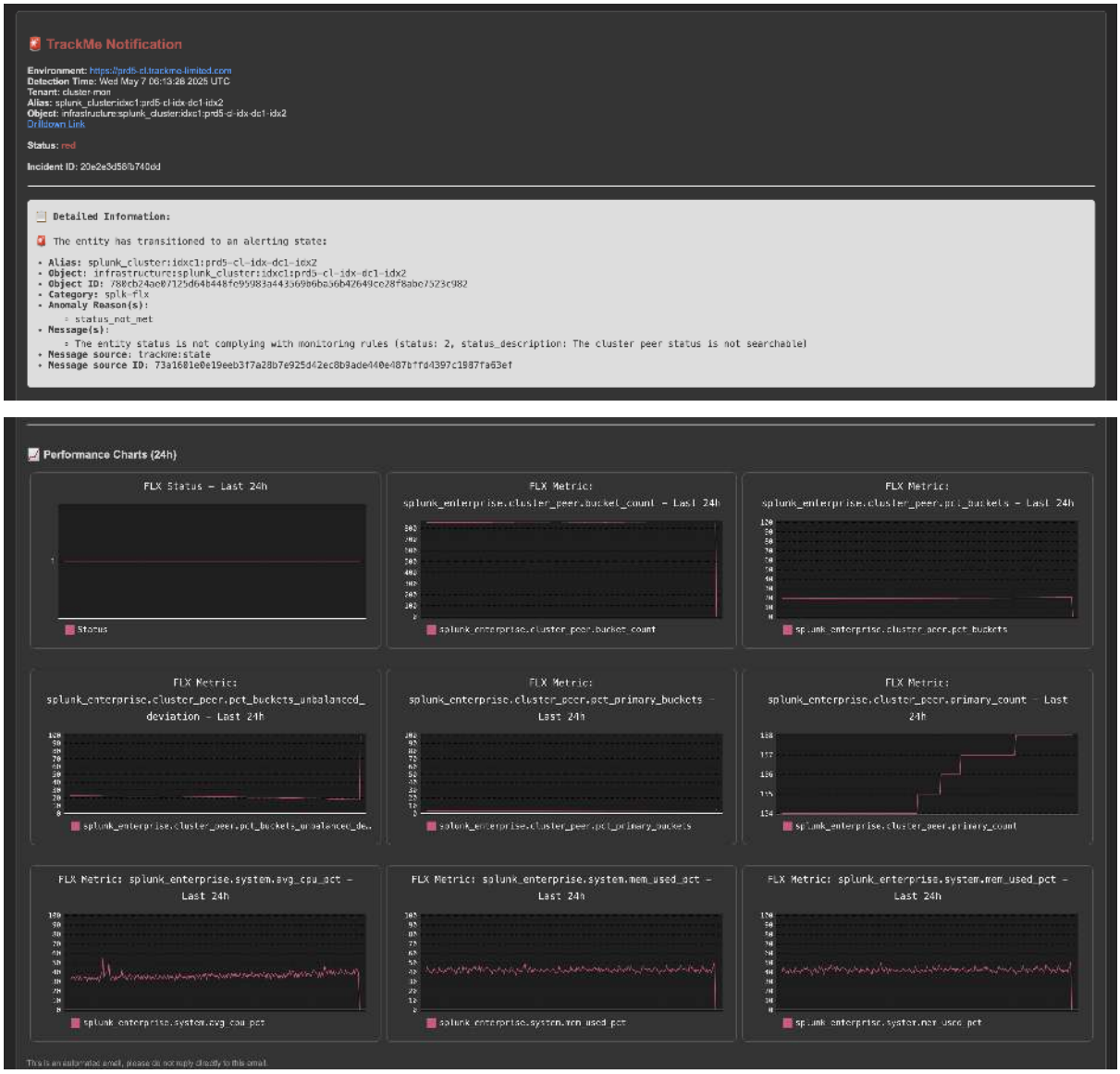
The Services Monitoring Dashboard view:



The StateFul Opening incident Email notification:

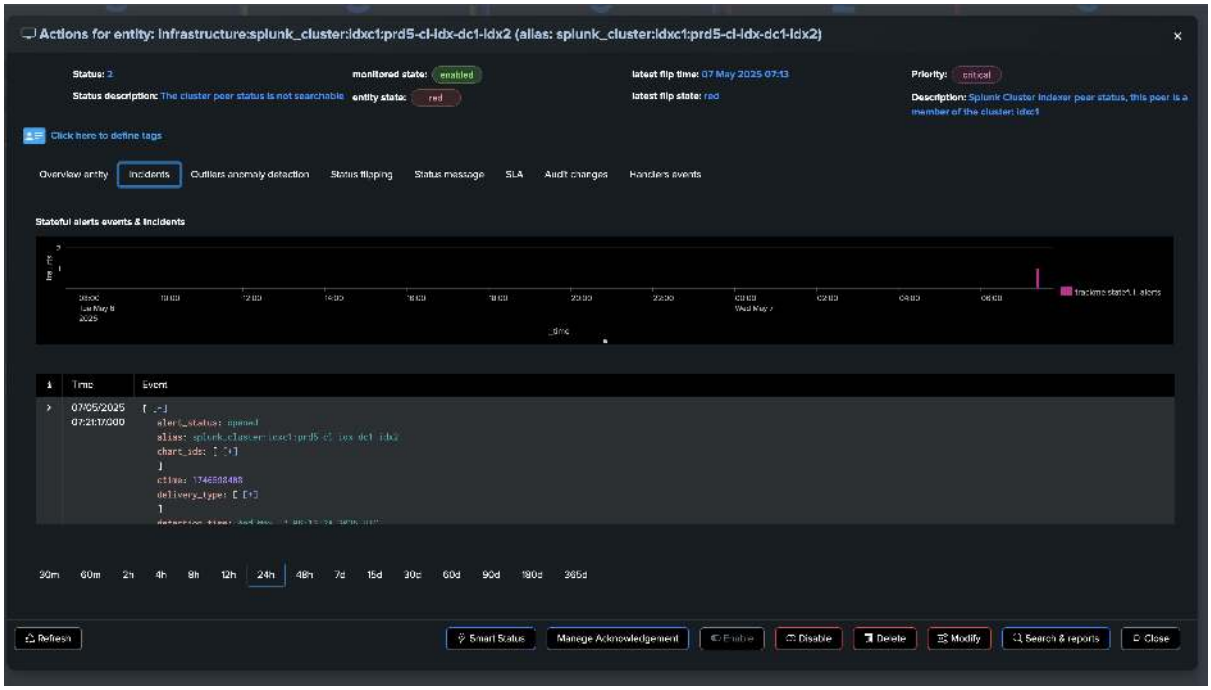


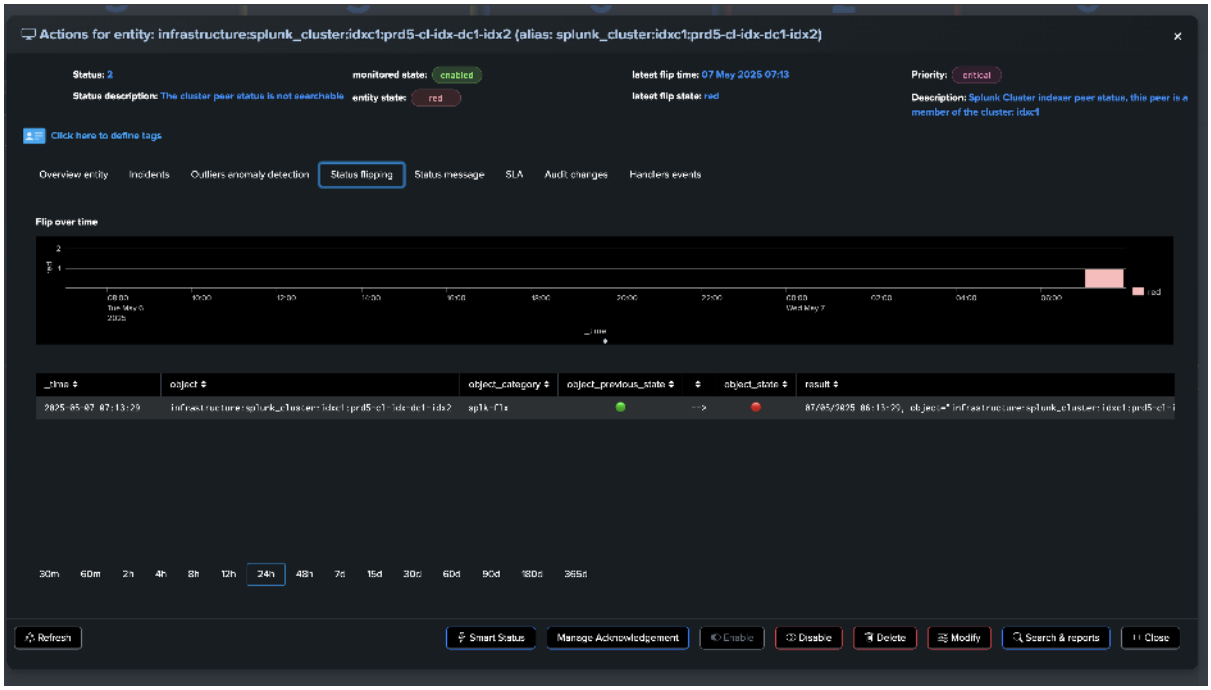
The StateFul Opening incident notification from the Splunk Indexer tier tenant:

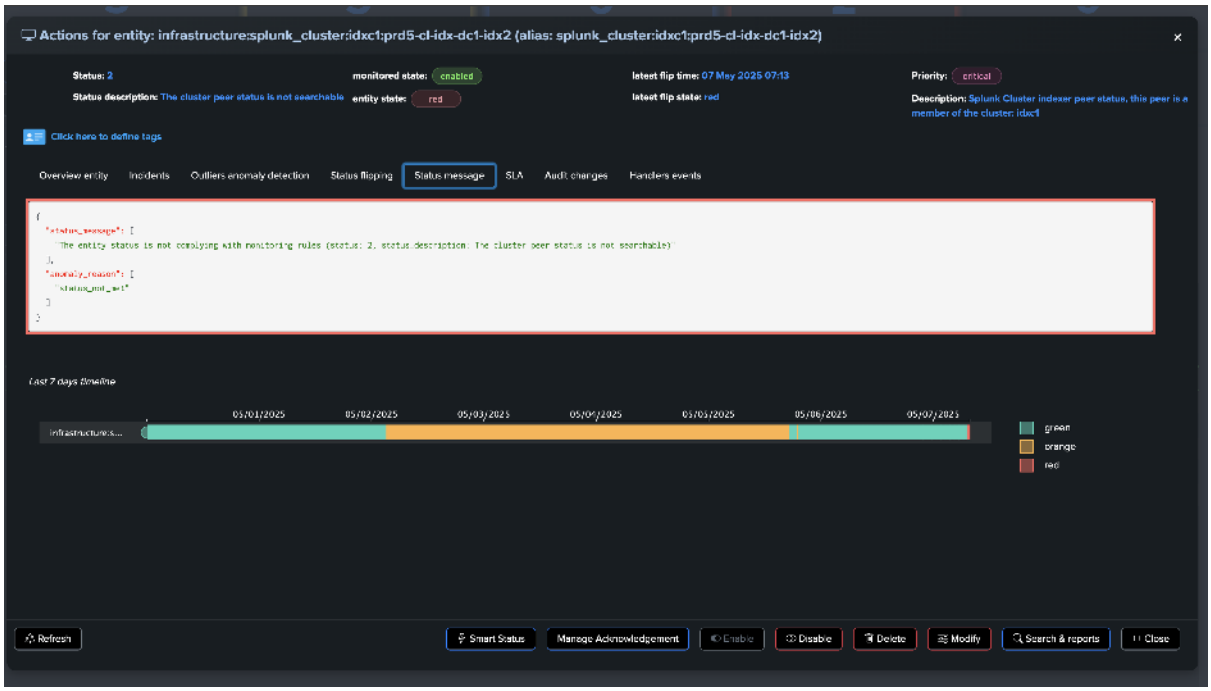


Several views of the faulty Splunk indexer:

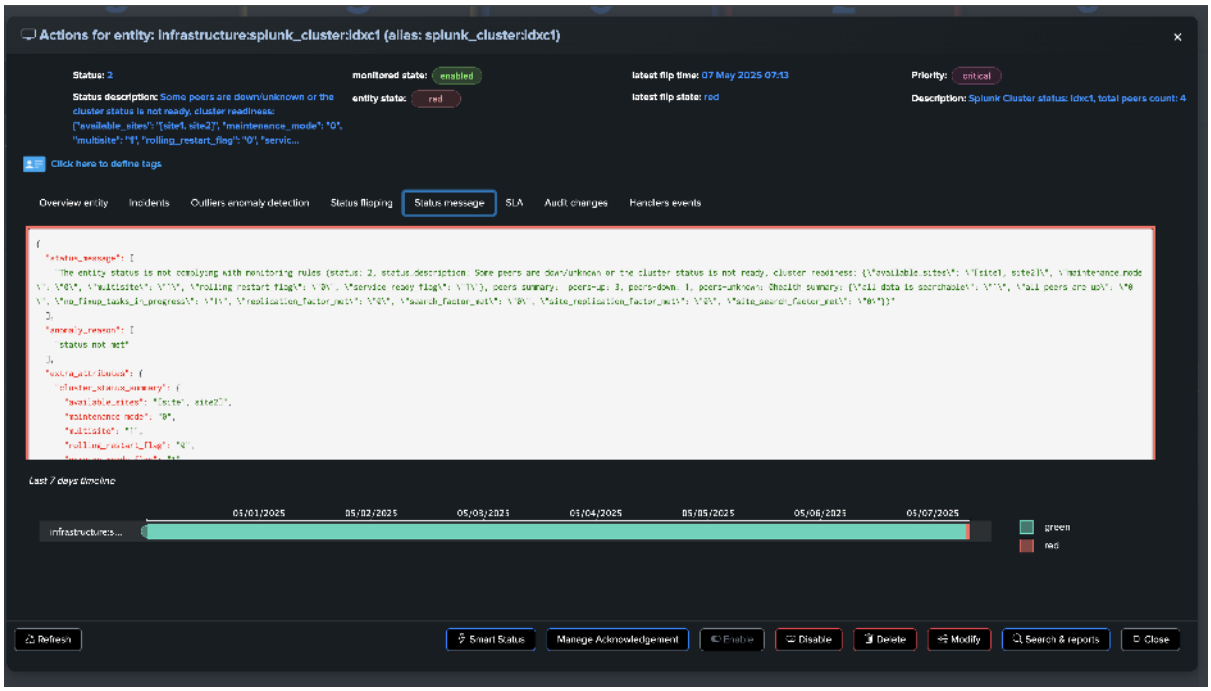
Actions for entity: infrastructure:splunk_cluster:idxc1:prd5-cl-idx-dc1-idx2 (alias: splunk_cluster:idxc1:prd5-cl-idx-dc1-idx2)
Status: 2
Status description: The cluster peer status is not searchable
monitored state: enabled
entity state: red
latest flip time: 07 May 2025 07:13
latest flip state: red
Priority: critical
Description: Splunk Cluster Indexer peer status, this peer is a member of the cluster: idxc1
Click here to define tags
Overview entity
Incidents
Outliers anomaly detection
Status flipping
Status message
SLA
Audit changes
Handlers events
Metrics for this entity:
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
180d
365d
Refresh
Smart Status
Manage Acknowledgement
Enable
Disable
Delete
Modify
Search & reports
Close


Actions for entity: infrastructure:splunk_cluster:idxc1:prd5-cl-idx-dc1-idx2 (alias: splunk_cluster:idxc1:prd5-cl-idx-dc1-idx2)
Status: 2
Status description: The cluster peer status is not searchable
monitored state: enabled
entity state: red
latest flip time: 07 May 2025 07:13
latest flip state: red
Priority: critical
Description: Splunk Cluster Indexer peer status, this peer is a member of the cluster: idxc1
Click here to define tags
Overview entity
Incidents
Outliers anomaly detection
Status flipping
Status message
SLA
Audit changes
Handlers events
Stateful alerts events & Incidents
Time
Event
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
180d
365d
Refresh
Smart Status
Manage Acknowledgement
Enable
Disable
Delete
Modify
Search & reports
Close

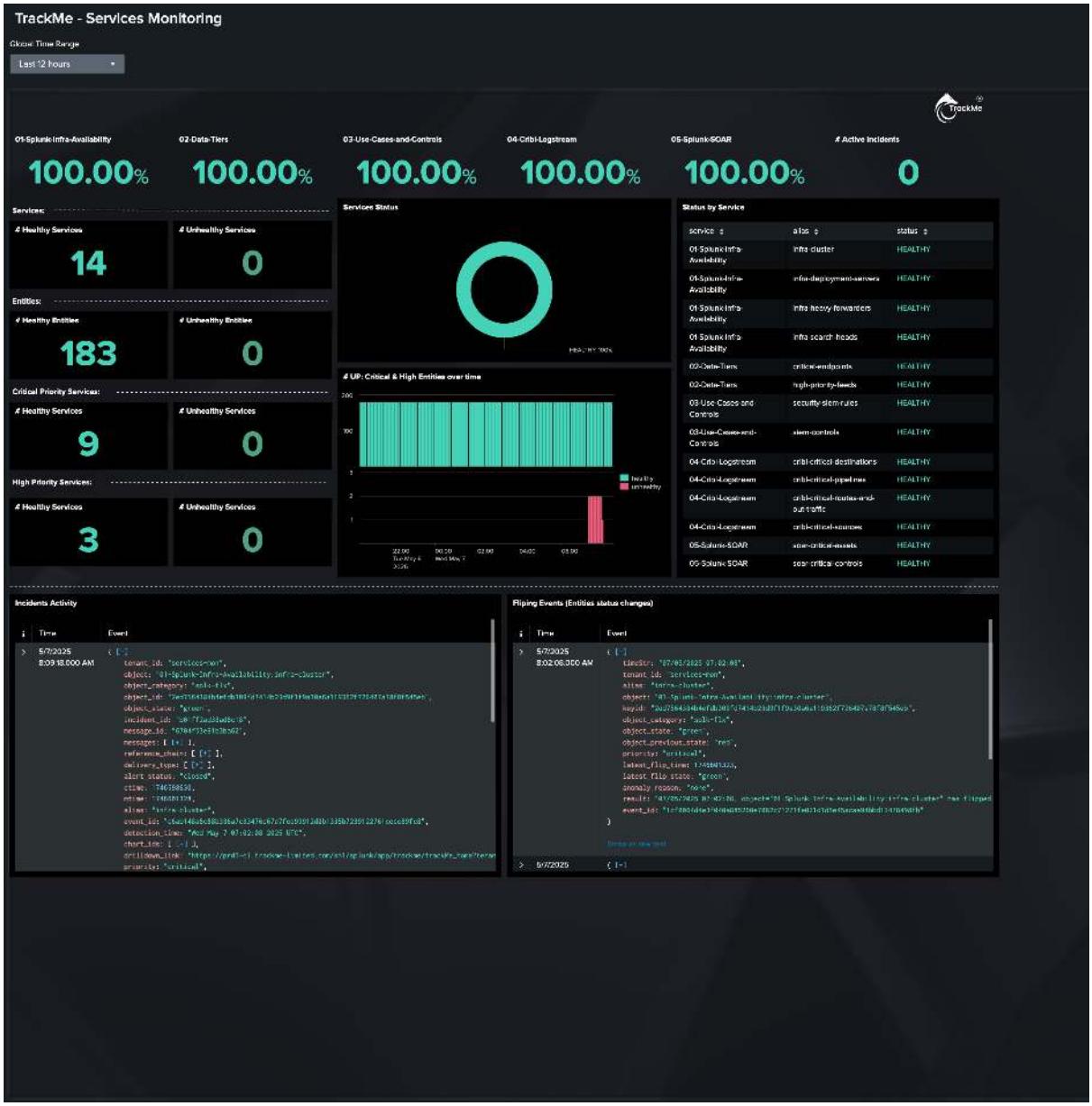




A few pictures from the global Splunk cluster entity view:

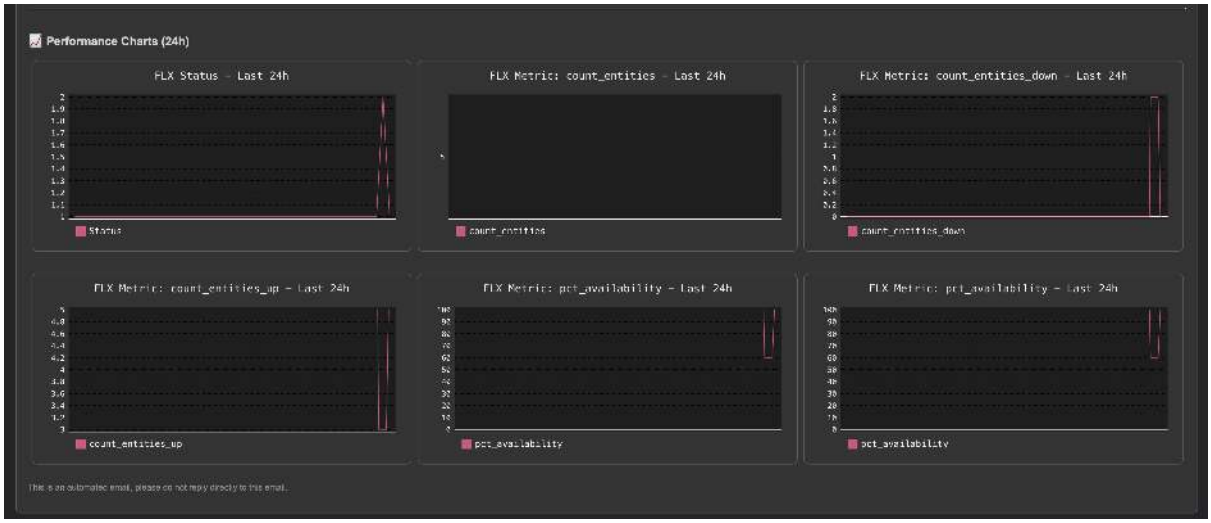


After some time, the issue is resolved, the faulty indexer is back in service, the incident is closed and the dashboard shows the updated situation for our Splunk tiers:



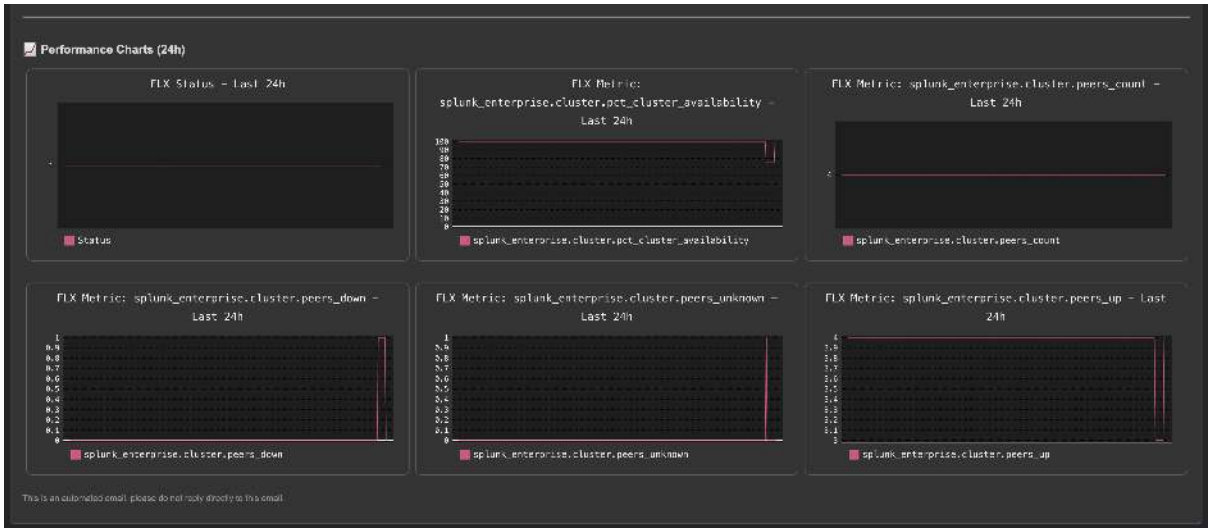
Closure incident Email notifications were sent to the team:

Notes: for the documentation purposes, we show notification for the service as well as the entities, but you may choose to only send notifications for the tiers services.


Performance Charts (24h)
FLX Status - Last 24h
FLX Metric: count_entities - Last 24h
FLX Metric: count_entities_down - Last 24h
FLX Metric: count_entities_up - Last 24h
FLX Metric: pct_availability - Last 24h
FLX Metric: pct_availability - Last 24h
Status
count_entities
count_entities_down
count_entities_up
pct_availability
pct_availability
This is an automated email, please do not reply directly to this email.

TrackMe Notification
Environment: https://prd5-cl.trackme-limited.com
Detection Time: Wed May 7 08:53:23 2025 UTC
Tenant: cluster-mon
Alias: splunk_cluster:idxc1:prd5-cl-idx-dc1-idx2
Object: infrastructure:splunk_cluster:idxc1:prd5-cl-idx-dc1-idx2
Drilldown Link
Status: green
Incident ID: 20e2e3d58fb740dd
Detailed Information:
The entity has received an incident closure update and is now in a non-alerting state:
Alias: splunk_cluster:idxc1:prd5-cl-idx-dc1-idx2
Object: infrastructure:splunk_cluster:idxc1:prd5-cl-idx-dc1-idx2
Object ID: 780cb24ae07125d64b448fe95803a443569b5ba56b42649ce28f8a3e7523c982
Category: splk-flx
Anomaly Reason(s):
none
Message(s):
07/05/2025 06:53:23, object="infrastructure:splunk_cluster:idxc1:prd5-cl-idx-dc1-idx2" has flipped from previous_state="red" to state="green" with anomaly_reason="none", previous_anomaly_reason="status_not_met", disruption_time="12.31"
The entity status is complying with monitoring rules (status: 1, status_description: The cluster peer status is searchable and properly balanced)
Message source: trackme:flip
Message source ID: 1e7125b5e5c7b73137c3b4758a424d38ff6b3799a3567ee0cd1f6e2238bd78cd

TrackMe Notification
Environment: https://prd5-cl.trackme-limited.com
Detection Time: Wed May 7 08:58:28 2025 UTC
Tenant: cluster-mon
Alias: splunk_cluster:idxc1
Object: infrastructure:splunk_cluster:idxc1
Drilldown Link
Status: green
Incident ID: 9bed851dd3584c9b
Detailed Information:
The entity has received an incident closure update and is now in a non-alerting state:
Alias: splunk_cluster:idxc1
Object: infrastructure:splunk_cluster:idxc1
Object ID: aa3aaa9e155c2bd5124a2aa0b0602c9ad66a0e6a7e7e266eb4507660cde2e26f
Category: splk-flx
Anomaly Reason(s):
none
Message(s):
07/05/2025 06:58:28, object="infrastructure:splunk_cluster:idxc1" has flipped from previous_state="red" to state="green" with anomaly_reason="none", previous_anomaly_reason="status_not_met", disruption_time="8.13"
The entity status is complying with monitoring rules (status: 1, status_description: All peers are up and running, peers summary: up:4, down: 0, unknown: 0 / cluster readiness: rolling_restart_flag: 0, service_ready_flag: 1, health_summary: {"all_data_is_searchable": "1", "all_peers_are_up": "1", "no_fixup_tasks_in_progress": "1", "replication_factor_met": "1", "search_factor_met": "1", "site_replication_factor_met": "1", "site_search_factor_met": "1"})
Message source: trackme:flip
Message source ID: 8167b2e84fab1d24dc5c2efa43b898af5b4ec8cc18c0d496aaae3758f7361edb



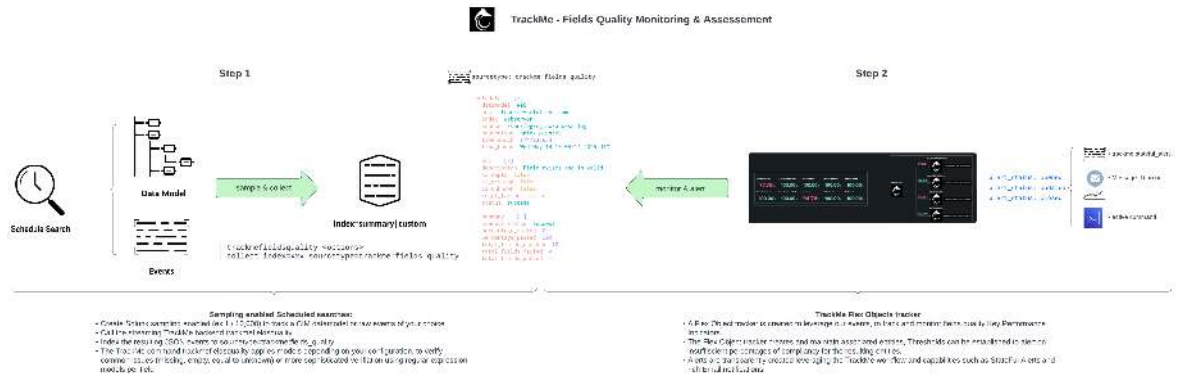
2.3 Use Case Demo: Fields Quality (CIM and non-CIM)

Use Case Demo: Fields Quality

- This white paper describes a new concept for performing continuous fields quality assessment on Splunk data.
- This leverages different scalable Splunk techniques and TrackMe components to perform this task.
- Fields quality assessment is a crucial aspect of Splunk monitoring, as it helps ensure that the data is accurate and consistent, ready to serve use cases.
- By implementing this, you will be able to build a solid, robust, and scalable solution to monitor the quality of fields parsing for your various sourcetypes in Splunk, as well as getting automated alerts when fields quality issues are detected.
- These concepts are applicable to both Splunk Common Information Model (CIM) and non-CIM data.
- Finally, we leverage a TrackMe restricted component called Flex Objects (splk-flx) to perform the continuous fields quality assessment, although parts of this logic are available to TrackMe Community Edition users.
- **TrackMe version 2.1.18 and later** is required to leverage this feature.
- **TrackMe version 2.1.19** added several utilities to support this use case and ease the implementation. (trackmefieldsqualityextract, trackmefieldsqualitygendict, trackmefieldsqualitygensummary)
- This work was made possible thanks to the support and deep collaboration of a major fellow TrackMe customer, thank you!

2.3.1 High level workflow and diagram

The following diagram shows the high-level workflow for fields quality assessment:



From a high-level perspective, the workflow is as follows:

Step 1: Collect

- The user defines a set of Splunk scheduled searches that leverage Splunk Sampling to sample data from the CIM data models or events of their choice.
- These searches call the streaming TrackMe backend `trackmefieldsquality` which performs the assessment of the fields quality: - Using one of the different methods supported by the command, define the fields of interest to monitor. - The command verifies for common issues: missing, empty or null, equal to unknown - The command can also check the content of the fields using a submitted regular expression as part of a model provided in input - The command generates a JSON object with the results of the assessment, as well as the global summary of the assessment for the sampled event - Metadata are stored in the JSON object (index, sourcetype, etc.) which can also be extended as per the user needs
- The search finally calls the Splunk `collect` command to index the JSON results using the TrackMe sourcetype `trackme:fields_quality`

Step 2: Monitor & Alert

- A TrackMe Virtual Tenant is created and enables the TrackMe Flex Objects (splk-flx) component.
- A Flex Object tracker is created which consumes the resulting JSON events, defines the associated entities and track the quality of the sampling over time.
- Thresholds can be defined for each entity using TrackMe capabilities, to finally generate automated alerts when the percentages of compliance go beyond the defined thresholds.

Hint

Common Information Knowledge (CIM) context versus raw events context

- This concepts apply equally to both the Common Information Model to any raw events in Splunk.
- CIM parsing quality is generally a critical topic when use cases heavily rely on CIM, but there can be main use cases where you need to ensure of the parsing quality out of a CIM context.
- These concepts are applicable to both CIM and non-CIM data, and you totally can simply use the same approach for both.

2.3.2 Phase 1: Collect

The primary step is to define what needs to be monitored depending on your needs and objectives.

The collect phase is highly flexible and scalable, in short the concept is the following:

- Create a set of scheduled searches that you will execute on a regular basis, for instance once per day during night off-peak hours.

- Each search will use the Splunk sampling feature, which allows randomly selecting a certain number of subset of events from the scope of the search.
- It then calls the `trackmefieldsquality` command with different parameters, which performs the assessment of the fields quality and generates a JSON object per event.
- Finally, the search will call the `collect` command to index the JSON results using the TrackMe sourcetype `trackme:fields_quality`.

About the Common Information Model (CIM)

- If your objective is to monitor the quality of your CIM parsing, from the lens of the CIM data models, you will likely want to have at least one search per CIM data model and node.
- Example: - 1 search for the `Web` data model - 1 search for the `Network_Traffic` data model - 1 search for the `Malware` data model - 1 search for the `Endpoints.Process` data model - etc.

Hint

About TrackMe system level sharing

- By default, TrackMe shares its content, including the command `trackmefieldsquality` at the application level only.
- This means that you cannot execute this command outside of the TrackMe application unless you share TrackMe at the system level.
- You can update your system configuration to change this behavior. Go to Splunk Web, navigate to `Manage Apps`, and update `permissions` on TrackMe so that `Apply selected role permissions to` is set to `All apps (system)`.

CIM: generate the JSON dictionary models

Hint

About the command trackmefieldsqualitygendict

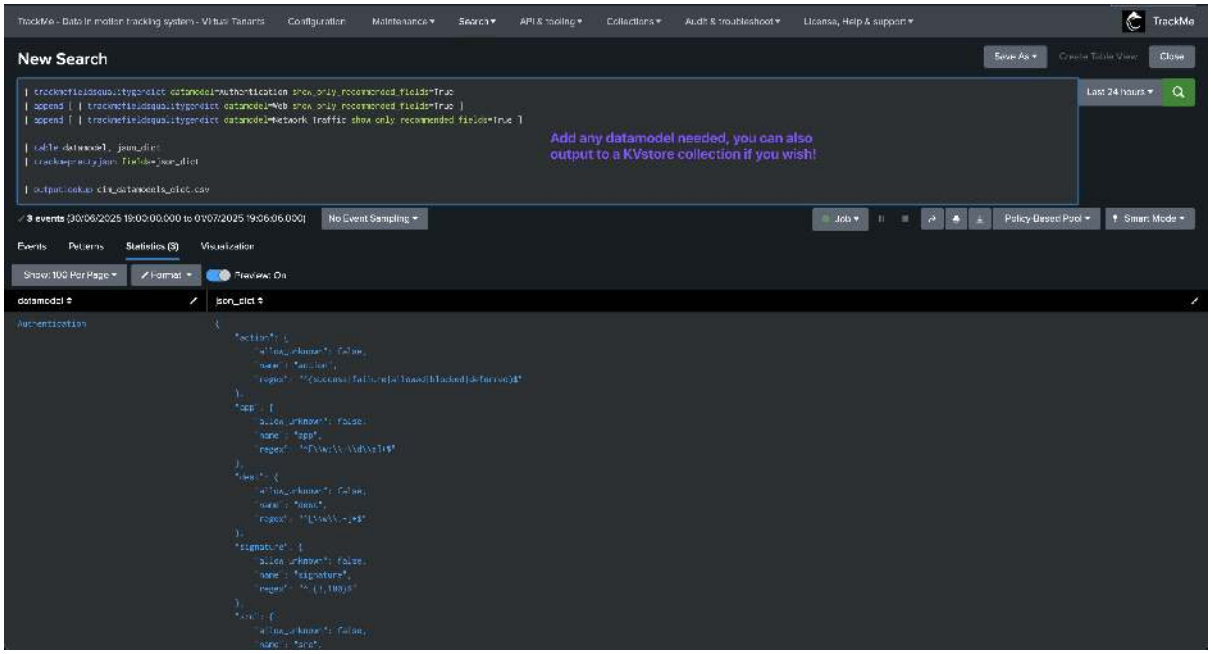
- This command is available in TrackMe version 2.1.19 and later.
- It allows generating the JSON dictionary model for a given CIM data model.
- It behaves similarly to the CIM Vladitor application and can restricted to only recommended fields per data model.

In this example, we will use the utility `trackmefieldsqualitygendict` to generate the JSON dictionary model for the Authentication, Web and Network_Traffic CIM data models, we will store these into a CSV lookup table so these can be modified as needed.

```
| trackmefieldsqualitygendict datamodel=Authentication show_only_recommended_
↪fields=True
| append [ | trackmefieldsqualitygendict datamodel=Web show_only_recommended_
↪fields=True ]
| append [ | trackmefieldsqualitygendict datamodel=Network_Traffic show_only_
↪recommended_fields=True ]

| table datamodel, json_dict
| trackmeprettyjson fields=json_dict

| outputlookup cim_datamodels_dict.csv
```



CIM: define the collect scheduled search

Hint

About the collect scheduled searches

- You can choose between two main approaches, running searches using the `datamodel` and `raw` searches.
- Both approaches are supported and can generate performing searches, even at large scale, especially leveraging the `Splunk sampling` feature.
- Alternatively to the `Sampling` approach, you could also simply use a "| head" approach, which is however much less meaningful in comparison. (less event distribution, less representative of the events)
- A great design would be to generate one search per context, such as one search per datamodel/sourcetype.

Raw search example

Hint

You can choose to create specific searches per context (for instance per datamodel/sourcetype), or more global searches per datamodel

- This command is available in TrackMe version 2.1.19 and later.
- It allows extracting the list of fields to check for a given CIM data model.
- It behaves similarly to the CIM Vladiator application and can restricted to only recommended fields per data model.

This is pretty much the approach that would use with CIM Vladiator application while reviewing your CIM compliance on a one shot basis.

Example for the Authentication datamodel:

```
(`cim_Authentication_indexes`) tag=authentication NOT (action=success user=*$)

``` custom metadata to identify the datamodel ```
| eval datamodel="Authentication", nodename="Authentication"

``` call the backend ```
| lookup cim_datamodels_dict.csv datamodel OUTPUT json_dict as quality_dict
| trackmefieldsquality fields_to_check_dict_fieldname="quality_dict" output_mode=json␣
↪metadata_fields="datamodel,nodename" include_field_values=True

``` call collect ```
| collect index=summary sourcetype=trackme:fields_quality
```

Example for the Network_Traffic datamodel:

```
(`cim_Network_Traffic_indexes`) tag=network tag=communicate

``` custom metadata to identify the datamodel ```
| eval datamodel="Network_Traffic", nodename="All_Traffic"

``` call the backend ```
| lookup cim_datamodels_dict.csv datamodel OUTPUT json_dict as quality_dict
| trackmefieldsquality fields_to_check_dict_fieldname="quality_dict" output_mode=json␣
↪metadata_fields="datamodel,nodename" include_field_values=True

``` call collect ```
| collect index=summary sourcetype=trackme:fields_quality
```

Example for the Web datamodel:

```
(`cim_Web_indexes`) tag=web

``` custom metadata to identify the datamodel ```
| eval datamodel="Web", nodename="Web"

``` call the backend ```
| lookup cim_datamodels_dict.csv datamodel OUTPUT json_dict as quality_dict
| trackmefieldsquality fields_to_check_dict_fieldname="quality_dict" output_mode=json␣
↪metadata_fields="datamodel,nodename" include_field_values=True

``` call collect ```
| collect index=summary sourcetype=trackme:fields_quality
```

Datamodel search sampling example

Example for the Authentication datamodel:

```
| datamodel Authentication Authentication flat strict_fields=false summariesonly=t

``` custom metadata to identify the datamodel ```
| eval datamodel="Authentication", nodename="Authentication"

``` call the backend ```
| lookup cim_datamodels_dict.csv datamodel OUTPUT json_dict as quality_dict
| trackmefieldsquality fields_to_check_dict_fieldname="quality_dict" output_mode=json␣
↪metadata_fields="datamodel,nodename" include_field_values=True
```

(continues on next page)
```

(continued from previous page)

```
``` call collect ```
| collect index=summary sourcetype=trackme:fields_quality
```

Example for the Network_Traffic datamodel:

```
| datamodel Network_Traffic All_Traffic flat strict_fields=false summariesonly=t

``` custom metadata to identify the datamodel ```
| eval datamodel="Network_Traffic", nodename="All_Traffic"

``` call the backend ```
| lookup cim_datamodels_dict.csv datamodel OUTPUT json_dict as quality_dict
| trackmefieldsquality fields_to_check_dict_fieldname="quality_dict" output_mode=json␣
↪metadata_fields="datamodel,nodename" include_field_values=True

``` call collect ```
| collect index=summary sourcetype=trackme:fields_quality
```

Example for the Web datamodel:

```
| datamodel Web Web flat strict_fields=false summariesonly=t

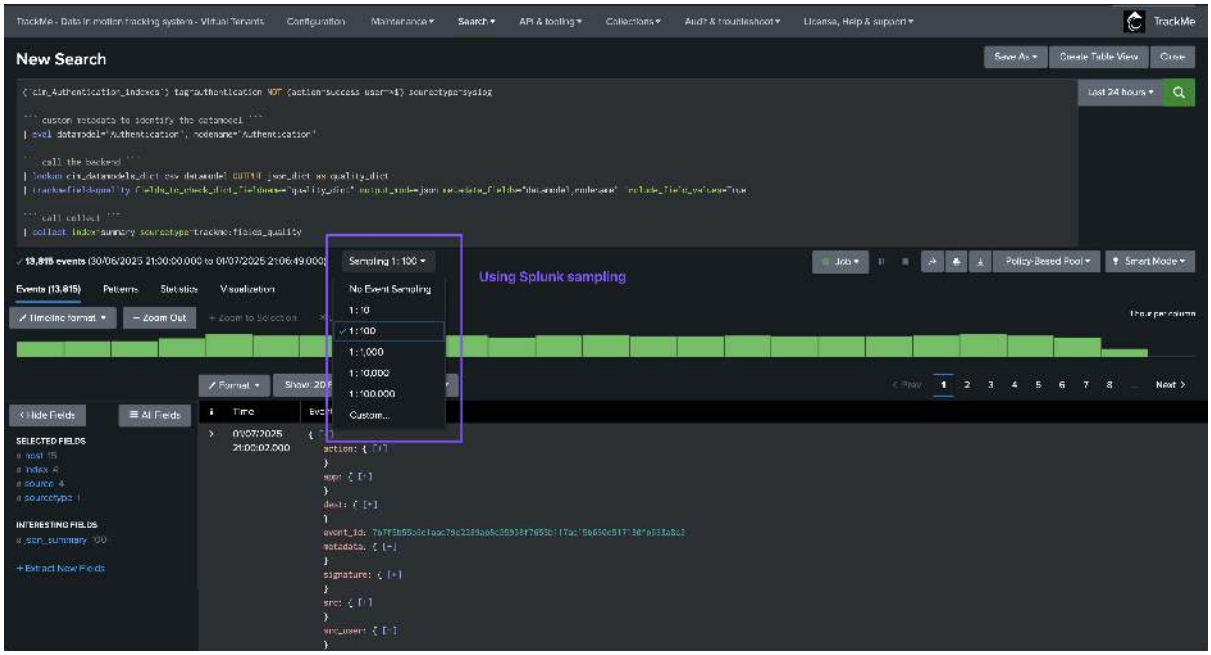
``` custom metadata to identify the datamodel ```
| eval datamodel="Web", nodename="Web"

``` call the backend ```
| lookup cim_datamodels_dict.csv datamodel OUTPUT json_dict as quality_dict
| trackmefieldsquality fields_to_check_dict_fieldname="quality_dict" output_mode=json␣
↪metadata_fields="datamodel,nodename" include_field_values=True

``` call collect ```
| collect index=summary sourcetype=trackme:fields_quality
```

Using Splunk sampling

In both cases, the ideal approach is to leverage the Splunk sampling feature to ensure a representative sample of the data, in a efficient way:
```

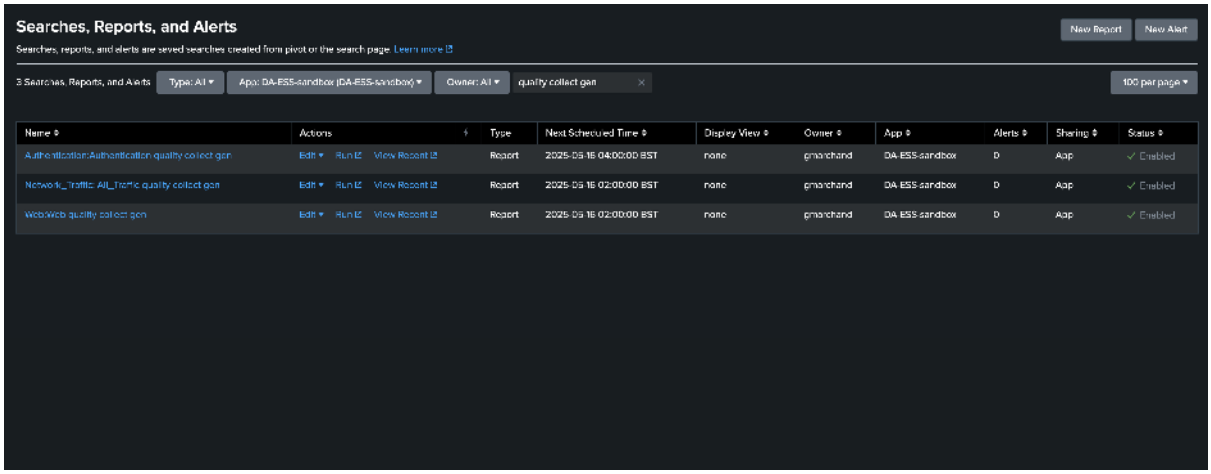


Final collect example

In this example, we monitor the fields quality of the following data models:

- `Authentication`
- `Web`
- `Network_Traffic`

This looks like the following:



2.3.3 Phase 2: Monitor & Alert

This is the simplest part of the work!

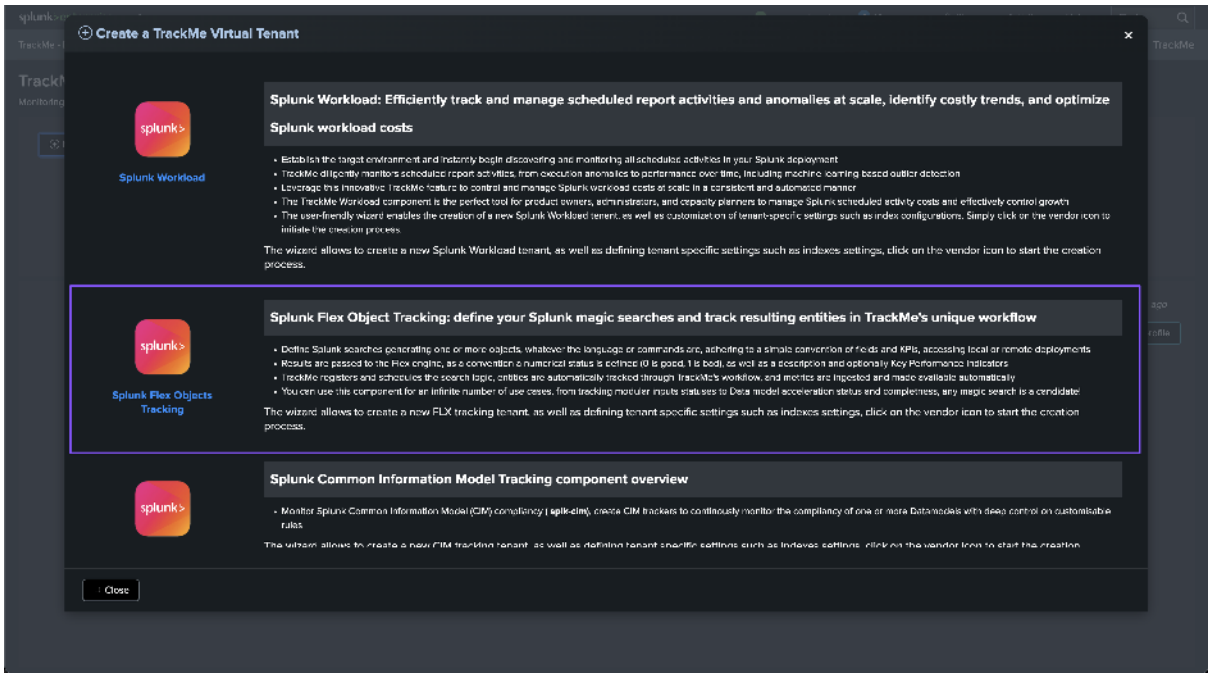
In short, we will:

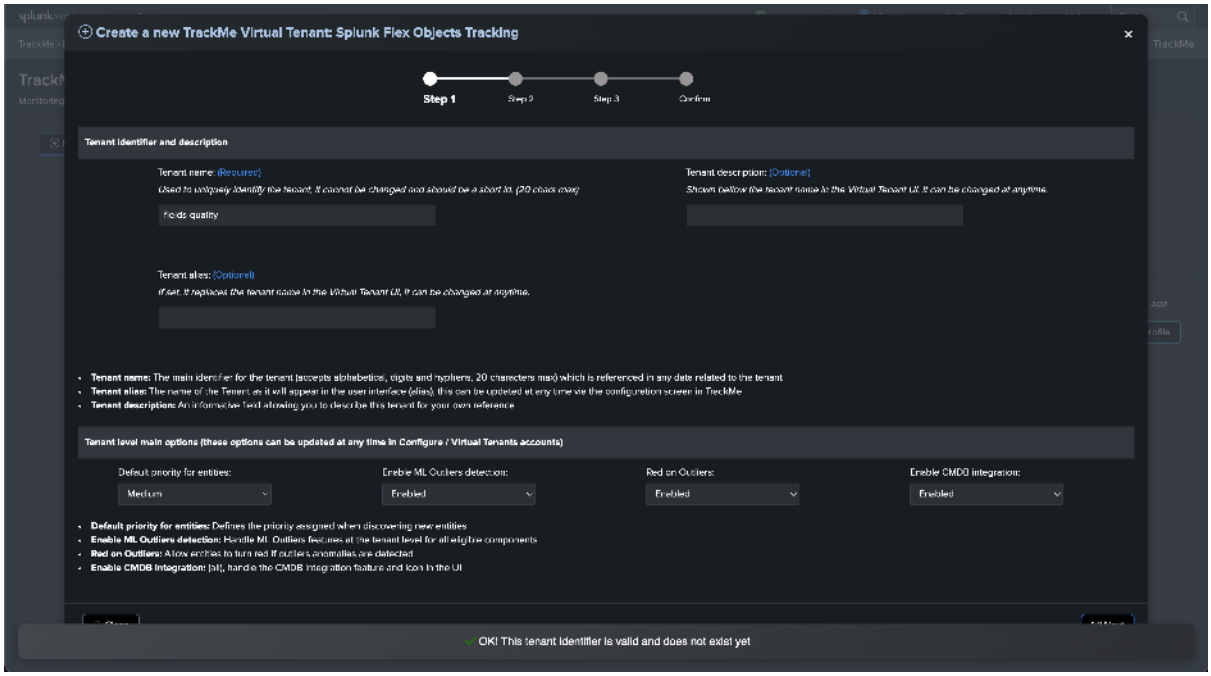
- Create a new TrackMe Virtual Tenant dedicated to the purposes of monitoring the fields quality.
- The Virtual Tenant enables the TrackMe Flex Objects (splk-flx) component.
- We will create a Flex Object tracker using the out-of-the-box use case called `splk_splunk_fields_quality`.

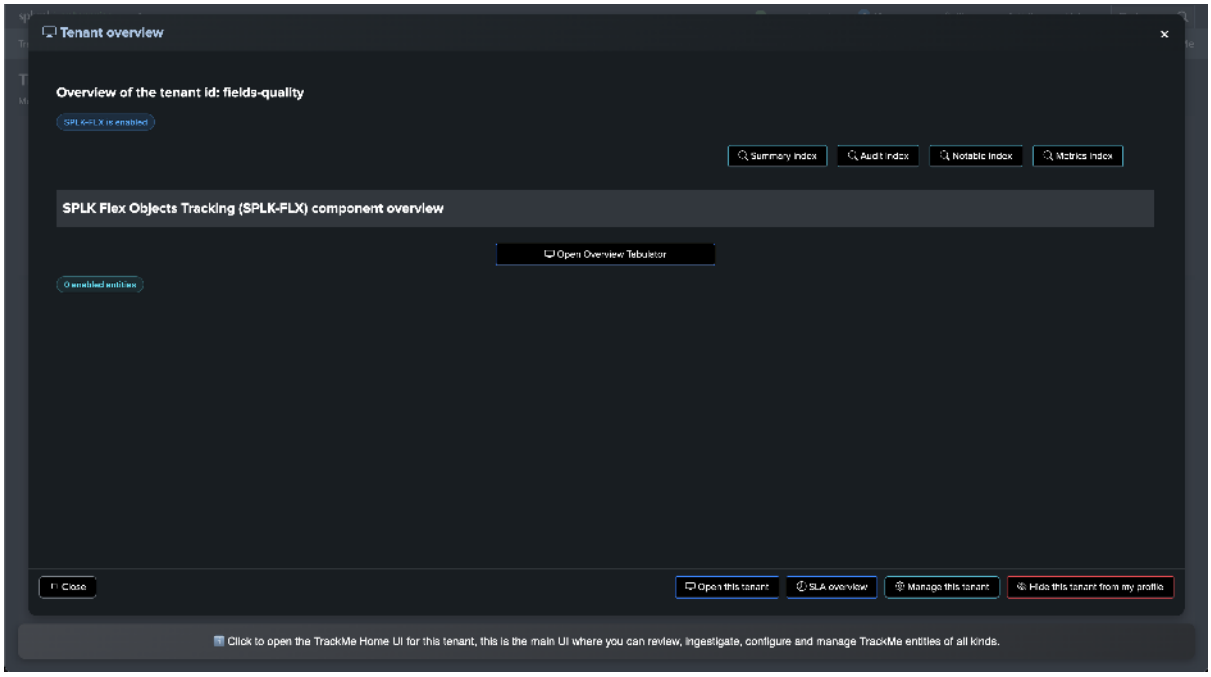
- The Flex Object tracker breaks our Metadata convention, automatically identifying the entities and tracking the quality of the sampling over time.

Creating the Virtual Tenant

We create a new Virtual Tenant called `fields-quality`:

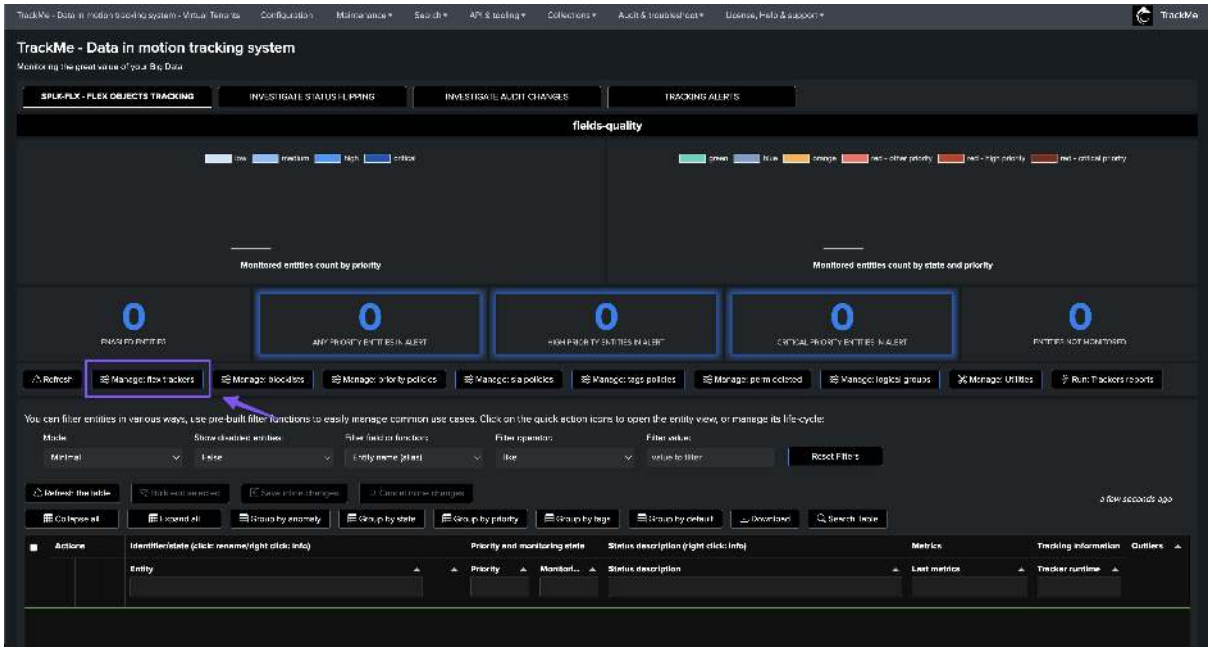




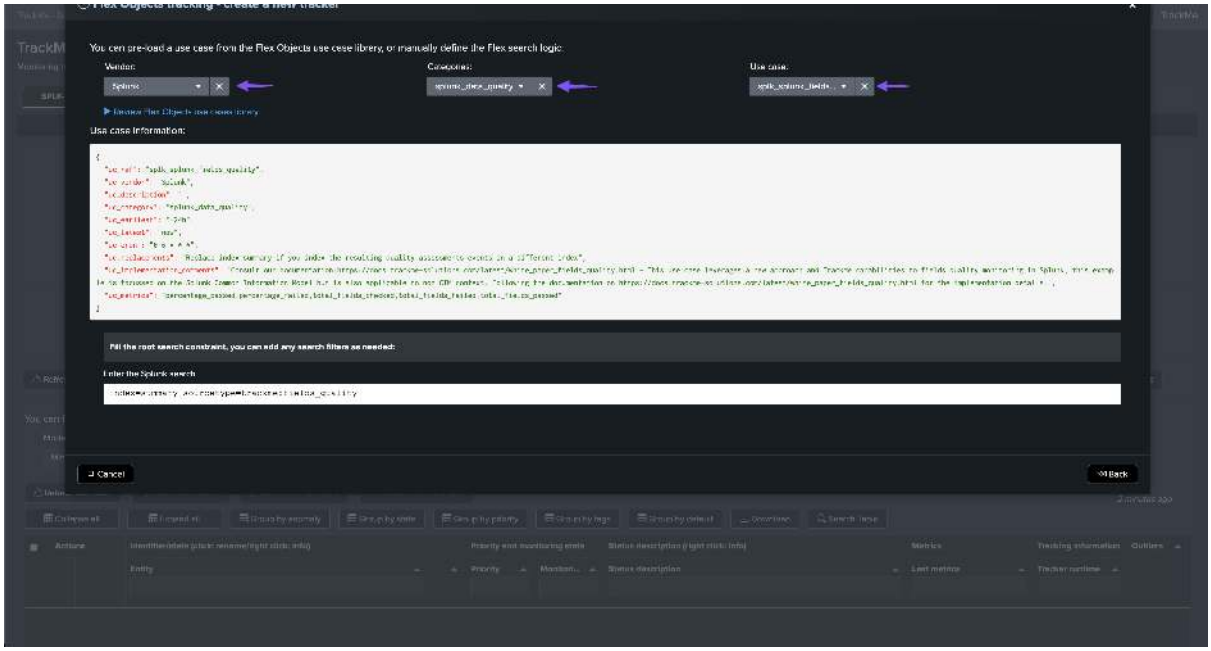


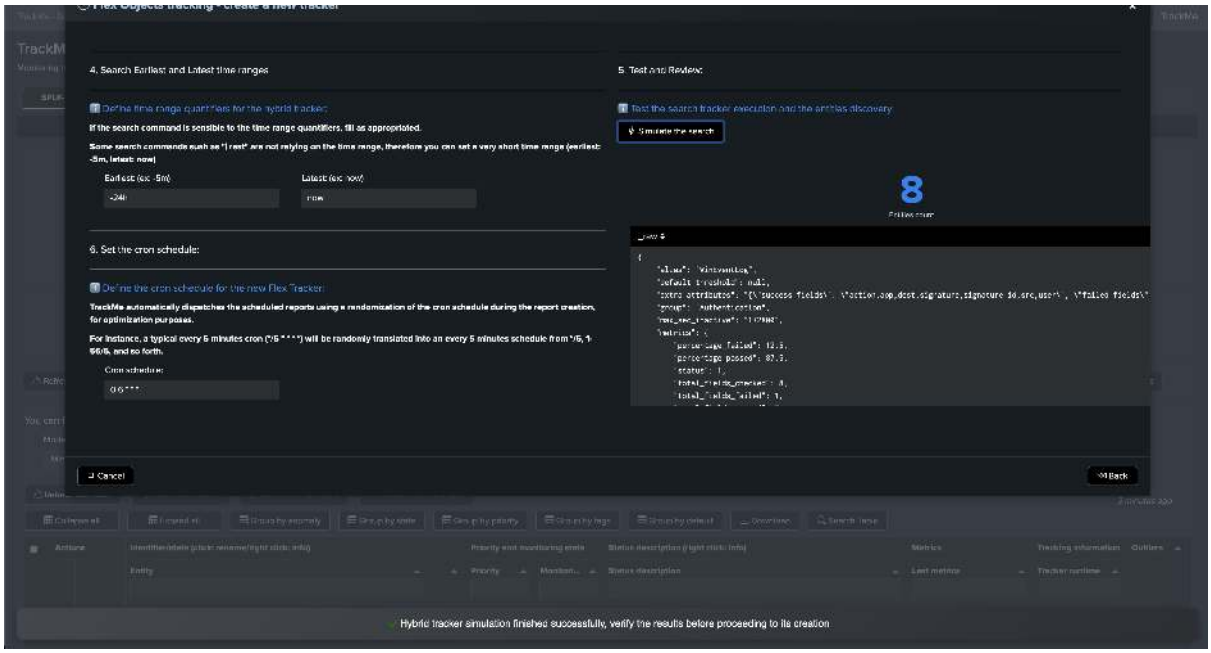
Creating the Flex Object tracker

Once we have a Virtual Tenant, we can create a Flex Object tracker using the out-of-the-box use case called `splk_splunk_fields_quality`:

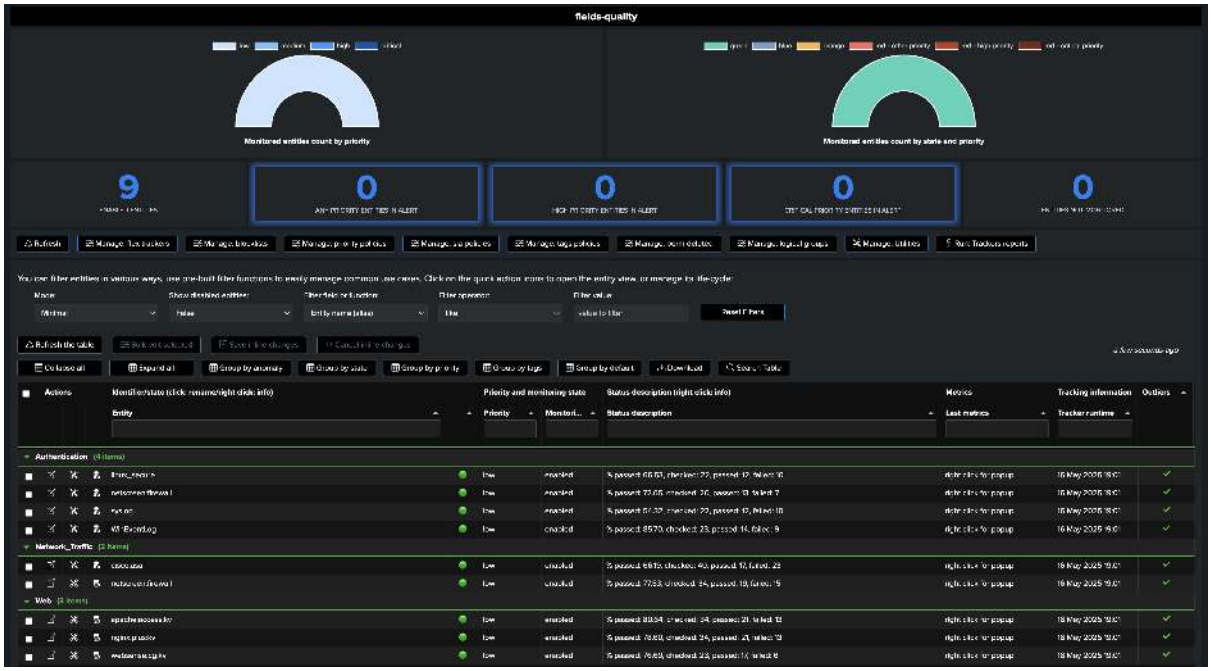


We create a new Flex Object tracker using the out-of-the-box use case called `splk_splunk_fields_quality`:

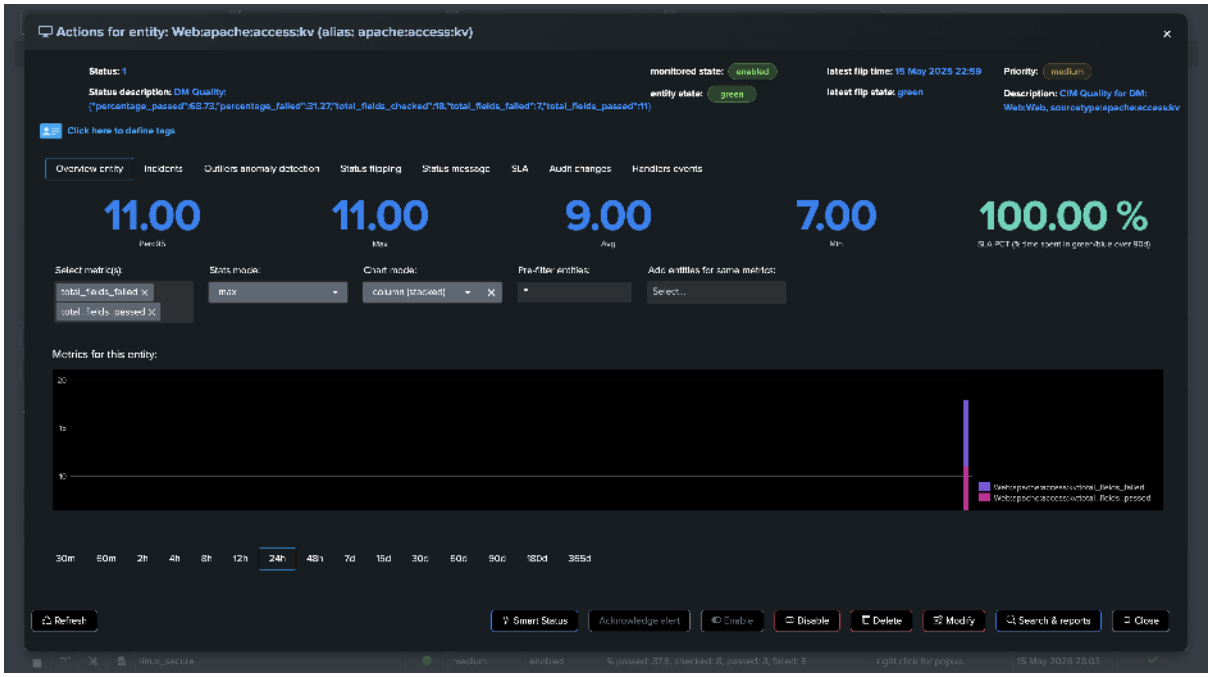


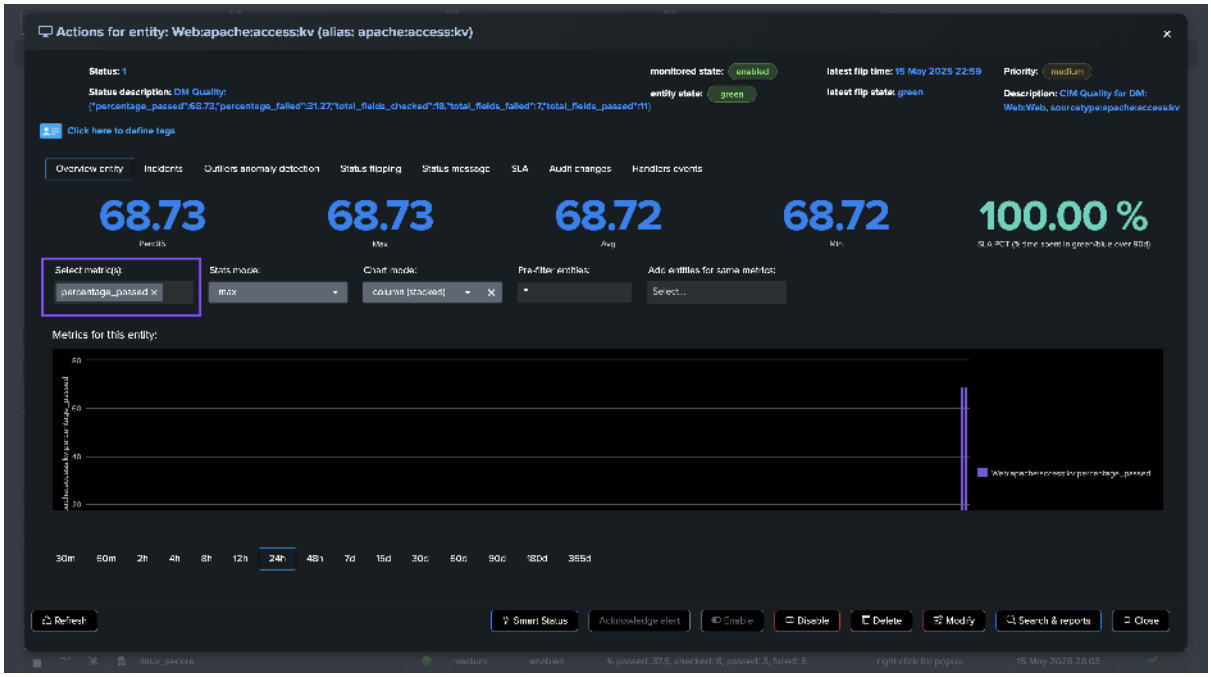


After a first execution, entities are created and ordered by Data Model:

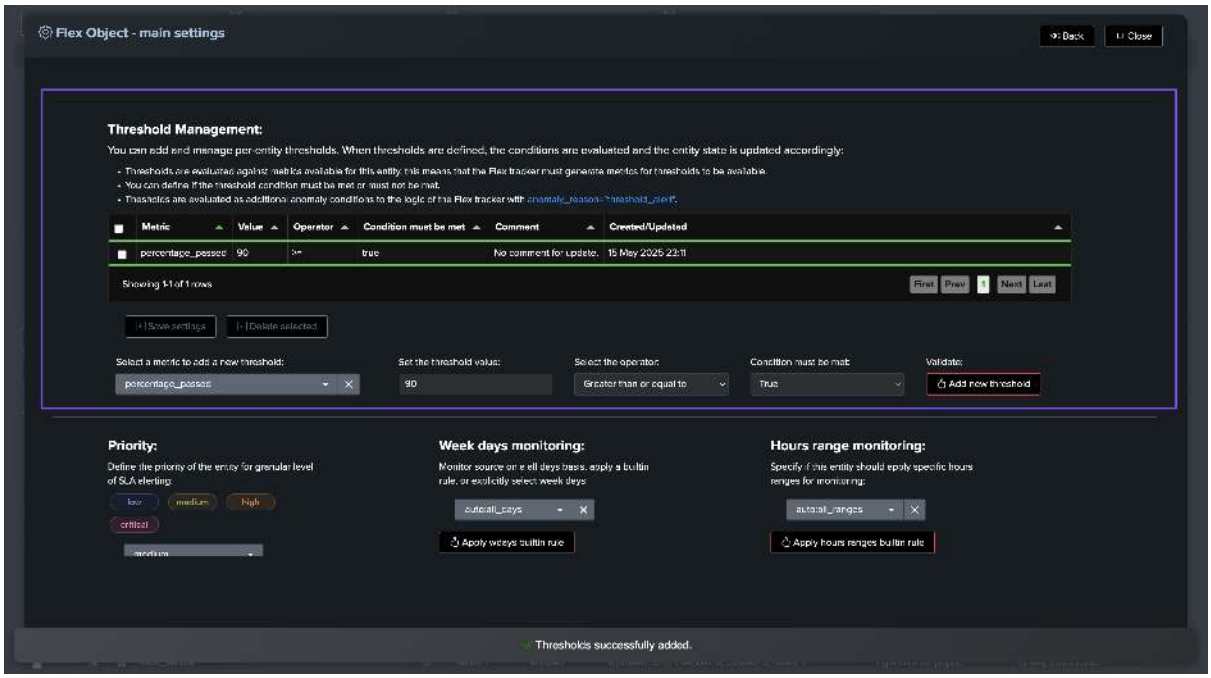


Key Performance Indicators start to be collected:

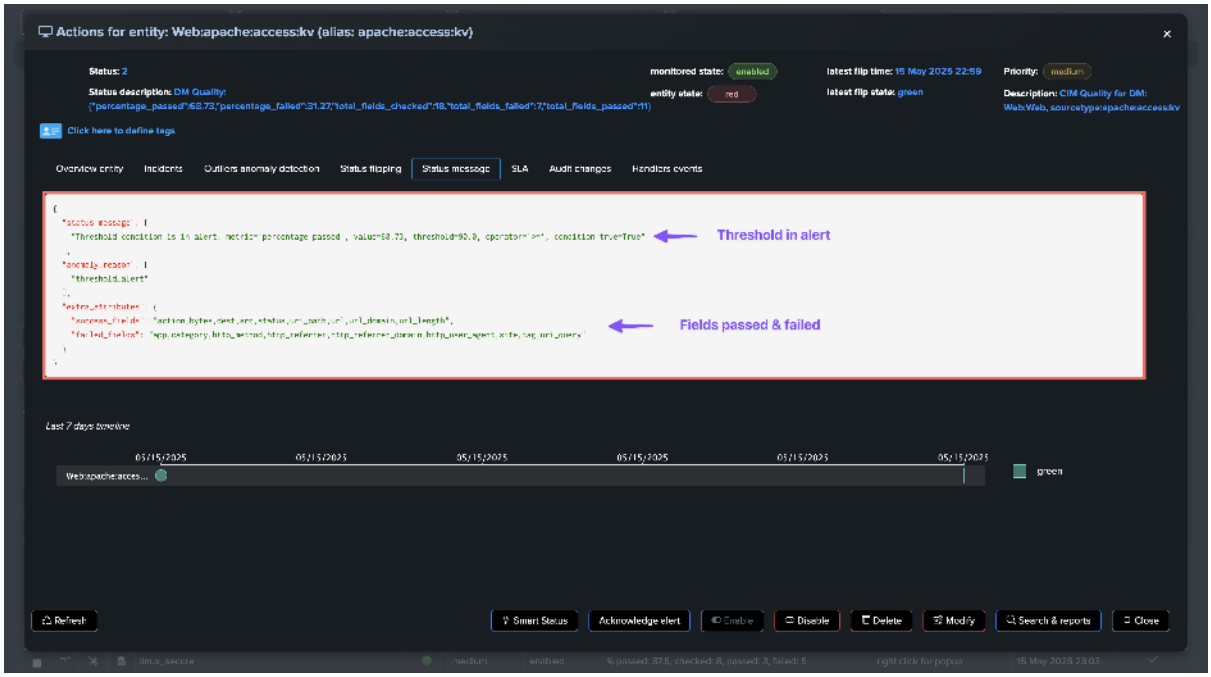




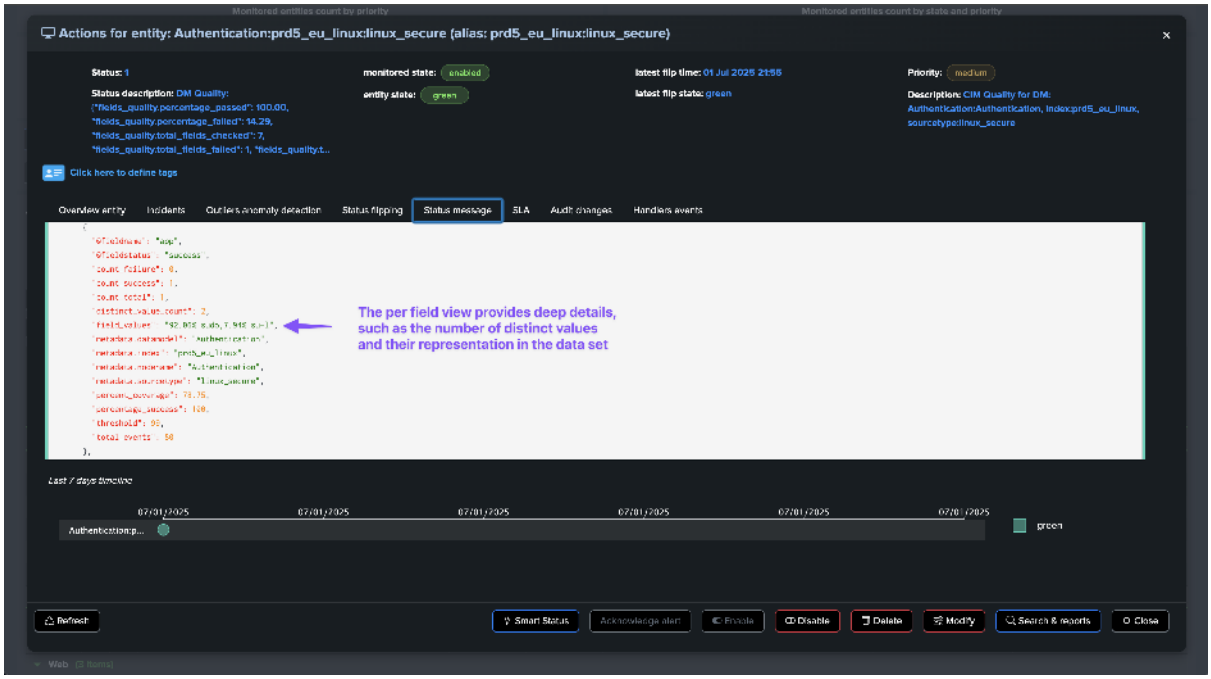
The default behavior will let you enabling and defining a threshold according to your needs:



Which would lead to turning this entity to red if the threshold is exceeded:



The Status message contains the details of the fields, including figures and most representitive values. (maxvals=15, as in the CIM Vladiator application)



We now have a complete and flexible solution to monitor the quality of the fields of your data models over time!

The details of the fields is stored in the KVstore in a JSON structure within the field `extra_attributes`, you can very easily access the fields and values:

```
| trackmegetcoll tenant_id=fields_quality component=flx
| table object, alias, extra_attributes
| trackmeprettyjson fields=extra_attributes
```

We also have access to sampled events so that we can easily build additional dashboards for investigation and analytic purposes.

2.3.4 Annex: usage and options for the command trackmefieldsquality

The command `trackmefieldsquality` is used in the collect phase to parse and validate the fields compliancy, this is a powerful and flexible Python backend that provides various options.

The command accepts the following parameters:

argument	description	default	example or valid values
`fields_to_check_list`	The list of fields to verify, provided as a comma-separated list	None	"action,app,bytes,url"
`fields_to_check_fiel`	Name of the field containing the list of fields to check (comma-separated)	None	"fields_list"
`fields_to_check_dict`	JSON string containing a dictionary of fields to check with optional regex patterns	None	'{"field1": {"name": "field1", "regex": "^[A-Z]+$"}, "field2": {"name": "field2"}}'
`fields_to_check_dict_`	Path to a JSON file containing a dictionary of fields to check with optional regex patterns	None	"/opt/splunk/etc/apps/myapp/mydir/we
`fields_to_check_dict_`	Name of the field containing a JSON string with a dictionary of fields to check	None	fields_dict
`fields_to_check_searc`	The search command to use to generate the dictionary of fields to check	None	"| trackmefieldsqualitygendict datamodel=Web show_only_recommended_fields=True"
`include_field_values`	Boolean option to include field values in the JSON summary	False	True/False
`pretty_print_json`	Boolean option to pretty print the JSON summary	True	True/False
`output_mode`	The mode to output the results (json or raw)	json	json
`metadata_fields`	CSV list of metadata fields to include in the metadata section of the JSON	index,sourcetype,host,sour	"datamodel"
`summary_fieldname`	Defines the name of the summary field	summary	"summary"
`metadata_fieldname`	Defines the name of the metadata field added to the summary JSON	metadata	"metadata"

The first 6 options are **mutually exclusive**, only one of these options can be used at a time:

- `fields_to_check_list`
- `fields_to_check_fieldname`
- `fields_to_check_dict`
- `fields_to_check_dict_path`

- `fields_to_check_dict_fieldname`
- `fields_to_check_search_command`

These options exist so that we can cover all use cases, this provides all levels of flexibility to use different Splunk techniques such as subsearches or simply storing the list of fields or models, dynamic generation in SPL, etc.

Let's take an example over each of these options:

Argument: fields_to_check_list

This is the most simple use case:

- Provide the list of fields to be checked as a comma-separated list
- For each field, we will check the following: - Missing - Empty - Null - Equal to unknown

If the field passes all these checks, it is declared as valid with a status of `success`, and `failure` otherwise.

The command will store with the JSON object a section for the field, which includes `flags` for each check, with a boolean value True or False:

- `is_missing`
- `is_empty`
- `is_unknown`
- `regex_failure`

In addition, the command will account the field and its status in the `summary` section of the JSON object:

- `overall_status`: the overall status of the field checks, either `success` or `failure`
- `total_fields_checked`: the total number of fields checked
- `total_fields_failed`: the total number of fields that failed
- `total_fields_passed`: the total number of fields that passed
- `percentage_failed`: the percentage of fields that failed
- `percentage_passed`: the percentage of fields that passed

Note: this argument does NOT process a regular expression to check for the content, therefore the `regex_failure` flag will always be `False`. (see next options for this)

Argument: fields_to_check_fieldname

This does exactly the same as `fields_to_check_list`, but instead of providing the list of fields to check as a comma-separated list, we provide the name of a field that contains the list of fields to check.

You would therefore call the command as follows:

Example:

```
| eval fields_list="action,app,bytes,url"
| trackmefieldsquality fields_to_check_fieldname="fields_list"
```

The point of having this option is that could for instance use a Splunk subsearch to generate the list of fields dynamically, for instance by accessing a lookup table where you store the fields depending on your criteria, or any other solution of your choice.

Hint

Providing a JSON dictionary model

- The next 3 options allow to provide a JSON dictionary model that models the fields to check, as well as optional parameters for each field.
- Especially, you can define a regular expression with the field `regex` to be apply against the value, allowing to valid the content of the field according to any needs.
- You can also define the field `allow_unknown` to be `True` or `False`, which can be used to disable the check for the field `is_unknown`.

Argument: fields_to_check_dict

This option is more sophisticated and allows to define a dictionnary that models the fields to check, and as well as an option regular expression to be apply against the value.

For instance, the following distionnary would verify the fields `bytes` including the fact that this should be a numerical value using a regular expression, the field `action` that for instance would accept only `success` or `failure`, and finally the field `http_referrer` where we would only perform the basic checks without verifying that its value matches certain criteria.

```
| trackmefieldsquality fields_to_check_dict="{\"bytes\": {\"name\": \"bytes\", \
↪"regex\": \"^\\\d*\"}, \"action\": {\"name\": \"action\", \"regex\": \"^
↪(success|failure)$\"}, \"http_referrer\": {\"name\": \"http_referrer\"}}" pretty_
↪print_json=False output_mode=json metadata_fields="datamodel" include_field_
↪values=True
```

In this case, if a regrex expression is provided, the `regex_failure` flag will be set to `True` if the value does not match the regular expression, and `False` otherwise, which accounts for status of the field in addition to the other checks.

Example of JSON output:


```
{
    "time": 1747261746,
    "bytes": {
        "status": "success",
        "description": "Field exists and is valid.",
        "is_missing": false,
        "is_empty": false,
        "is_unknown": false,
        "regex_failure": false,
        "value": "309"
    },
    "action": {
        "status": "failure",
        "description": "Field exists but value does not match the required pattern.",
        "is_missing": false,
        "is_empty": false,
        "is_unknown": false,
        "regex_failure": true,
        "value": "Bad Request"
    },
    "http_referrer": {
        "status": "failure",
        "description": "Field is 'unknown'.",
        "is_missing": false,
        "is_empty": false,
        "is_unknown": true,
        "regex_failure": false,
        "value": "unknown"
```


(continues on next page)

(continued from previous page)


```
    },
    "summary": {
        "overall_status": "failure",
        "total_fields_checked": 3,
        "total_fields_failed": 2,
        "total_fields_passed": 1,
        "percentage_failed": 66.67,
        "percentage_passed": 33.33
    },
    "metadata": {
        "time_epoch": 1747261746,
        "time_human": "Wed May 14 22:29:06 2025 UTC",
        "index": "webserver",
        "sourcetype": "nginx:plus:kv",
        "host": "trackme-solutions.com",
        "source": "/var/log/nginx/access.log",
        "datamodel": "Web"
    },
    "event_id": "f5ed1437ee8486a3782ebaea846dad37c52d47b825d1913c1ae7d085ba01f943"
}
```


Hint

Escaping backslashes and special characters

- The tricky part is that you need to pay attention to the JSON provided as input to the command
- Especially, double quotes within the JSON string need to be escaped.
- The regular expression also needs to be escaped, for instance `^\\\d*` which would otherwise be `\d*` in normal circumstances.

Argument: fields_to_check_dict_fieldname

Similarly to `fields_to_check_fieldname`, this option allows to provide the name of a field that contains the dictionary of fields to check as in the previous example.

This allows to use a Splunk subsearch to generate the dictionary dynamically, for instance by accessing a lookup table where you store the fields depending on your criteria, or any other solution of your choice.

Example:

```
| eval fields_dict="{\"bytes\": {\"name\": \"bytes\", \"regex\": \"^\\\d*\"}, \
↪"action\": {\"name\": \"action\", \"regex\": \"^(success|failure)$\"}, \"http_
↪referrer\": {\"name\": \"http_referrer\"}}"
| trackmefieldsquality fields_to_check_dict_fieldname="fields_dict" pretty_print_
↪json=False output_mode=json metadata_fields="datamodel" include_field_values=True
```

Argument: fields_to_check_search_command

This option allows to provide a search command that will be executed to generate the dictionary of fields to check dynamically. This is particularly useful when you want to generate the dictionary on-the-fly using commands like `trackmefieldsqualitygendict` or any other search command that produces the required JSON dictionary format.

The search command will be executed and its output will be used as the dictionary for fields quality assessment. This provides maximum flexibility for dynamic dictionary generation.

Example:

```
| trackmefieldsquality fields_to_check_search_command="| trackmefieldsqualitygendict␣
↪datamodel=Web show_only_recommended_fields=True" pretty_print_json=False output_
↪mode=json metadata_fields="datamodel" include_field_values=True
```

Example with custom search command:

```
| trackmefieldsquality fields_to_check_search_command="| inputlookup my_data_dict_
↪lookup | where somefilter=\"somevalue\"" pretty_print_json=False output_mode=json␣
↪metadata_fields="datamodel" include_field_values=True
```

Argument: fields_to_check_dict_path

This option is similar to `fields_to_check_dict_fieldname`, but instead of providing the dictionary as a JSON string, we provide the path to a JSON file that contains the dictionary.

The file must exist on the file system of the Splunk instance, and the path must be provided as a string.

Our JSON file would look like this:

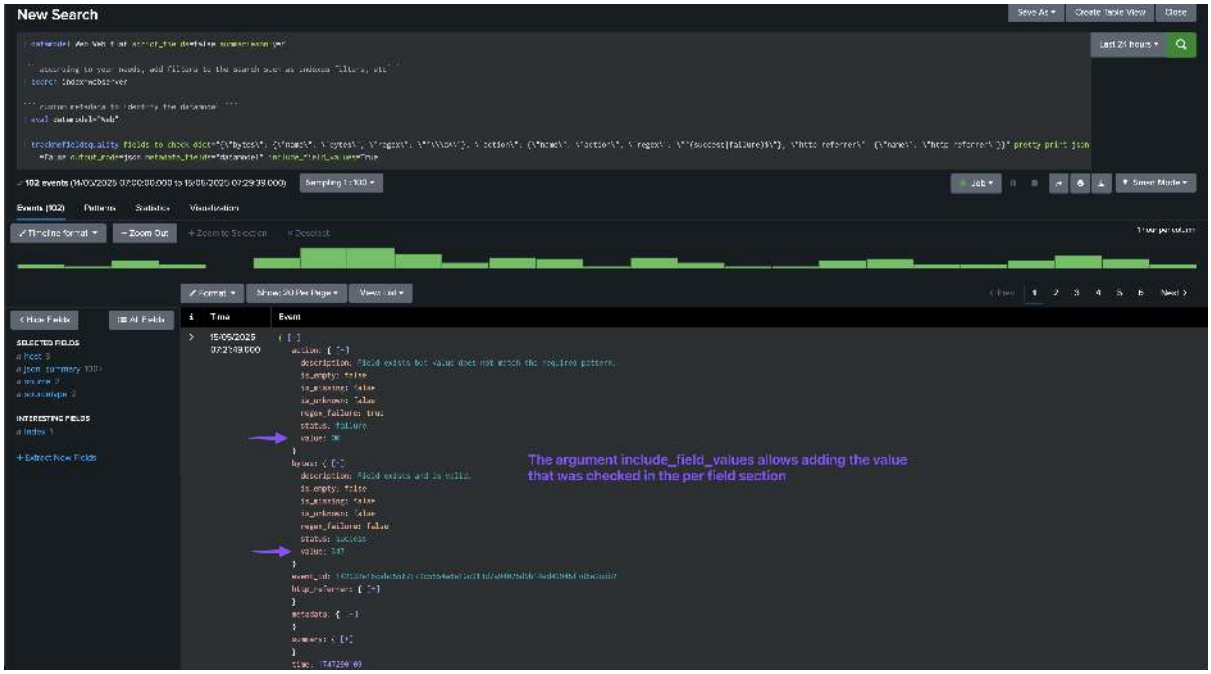
```
{
    "action": {
        "name": "action",
        "regex": "^(success|failure)$"
    },
    "bytes": {
        "name": "bytes",
        "regex": "^\\d*"
    },
    "http_referrer": {
        "name": "http_referrer"
    }
}
```

Example:

```
| trackmefieldsquality fields_to_check_dict_path="/opt/splunk/etc/apps/myapp/mydir/
↪web_datamodel.json" pretty_print_json=False output_mode=json metadata_fields=
↪"datamodel" include_field_values=True
```

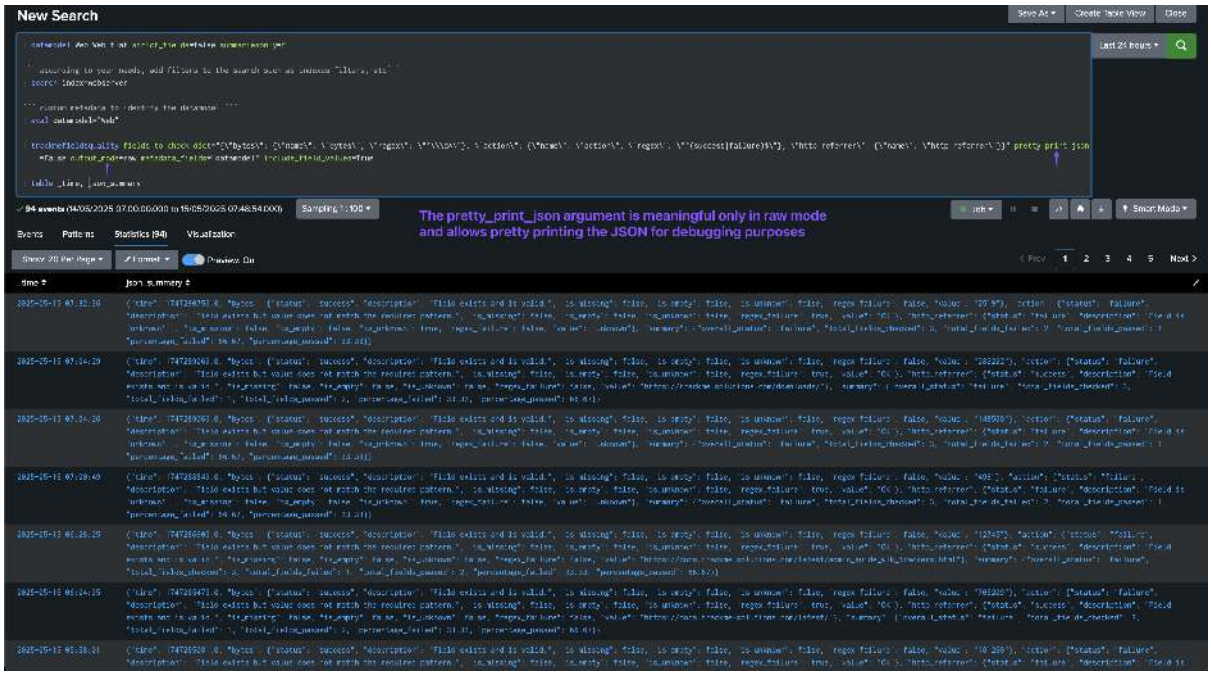
Argument: include_field_values

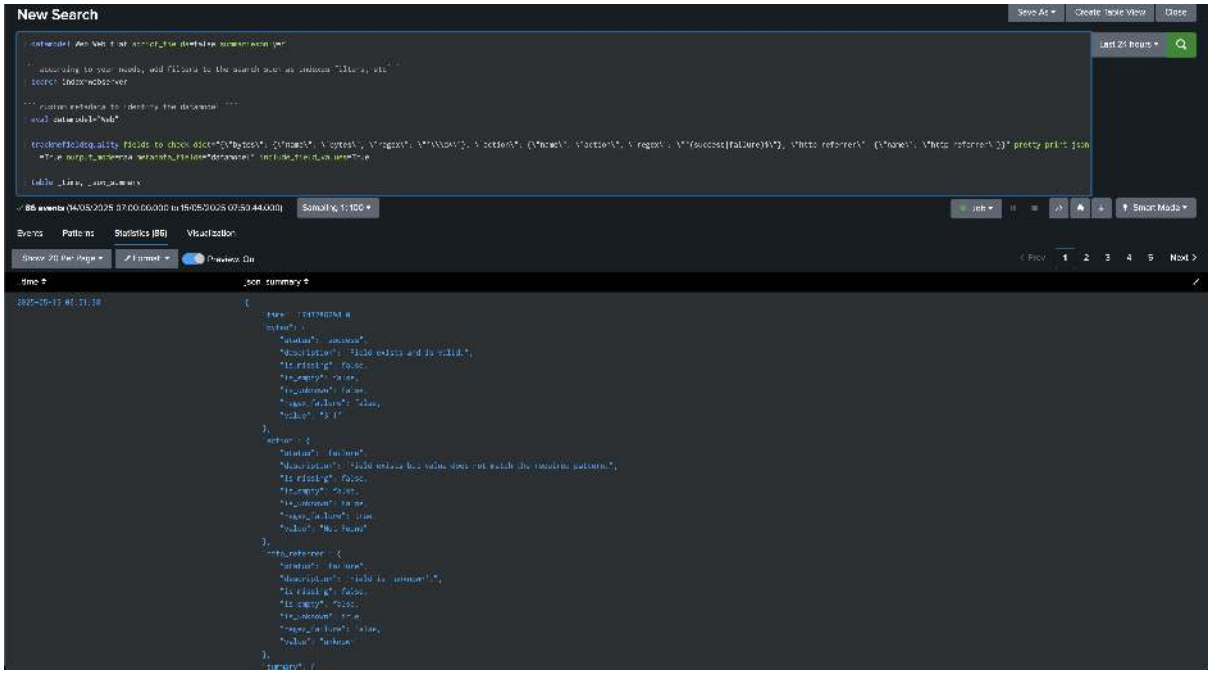
This option allows to include the field values in the JSON output, which can useful for analytics and reporting purposes.



Argument: pretty_print_json

This option allows to pretty print the JSON output, which can be useful for debugging purposes.





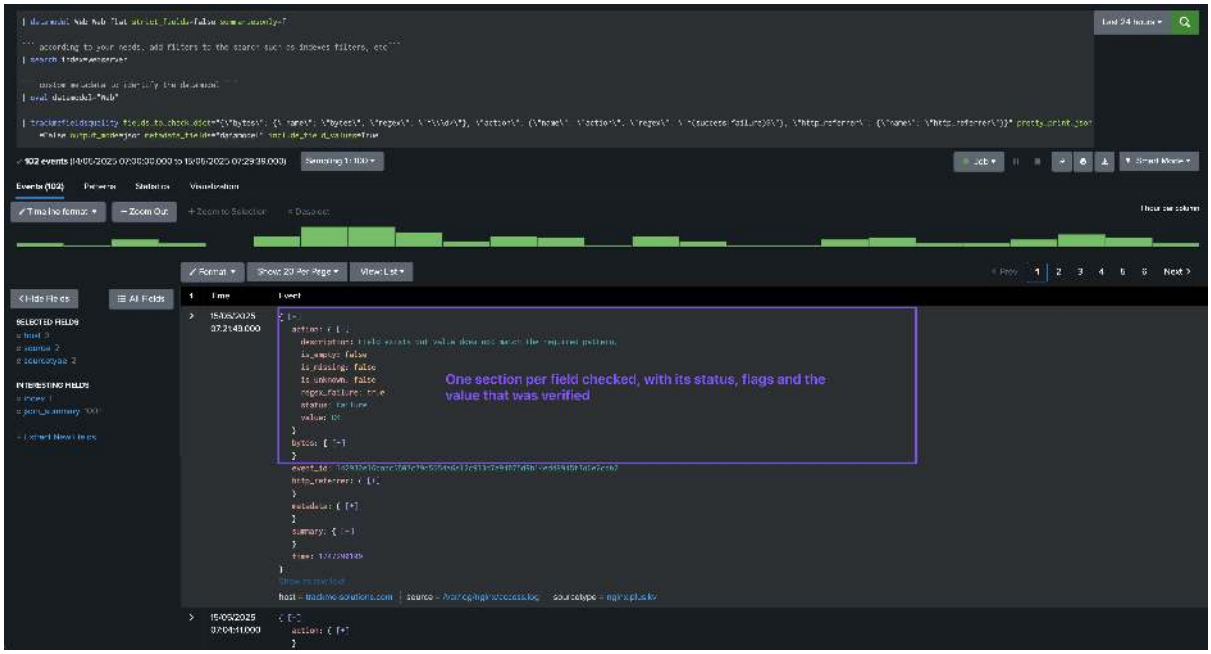
Argument: output_mode

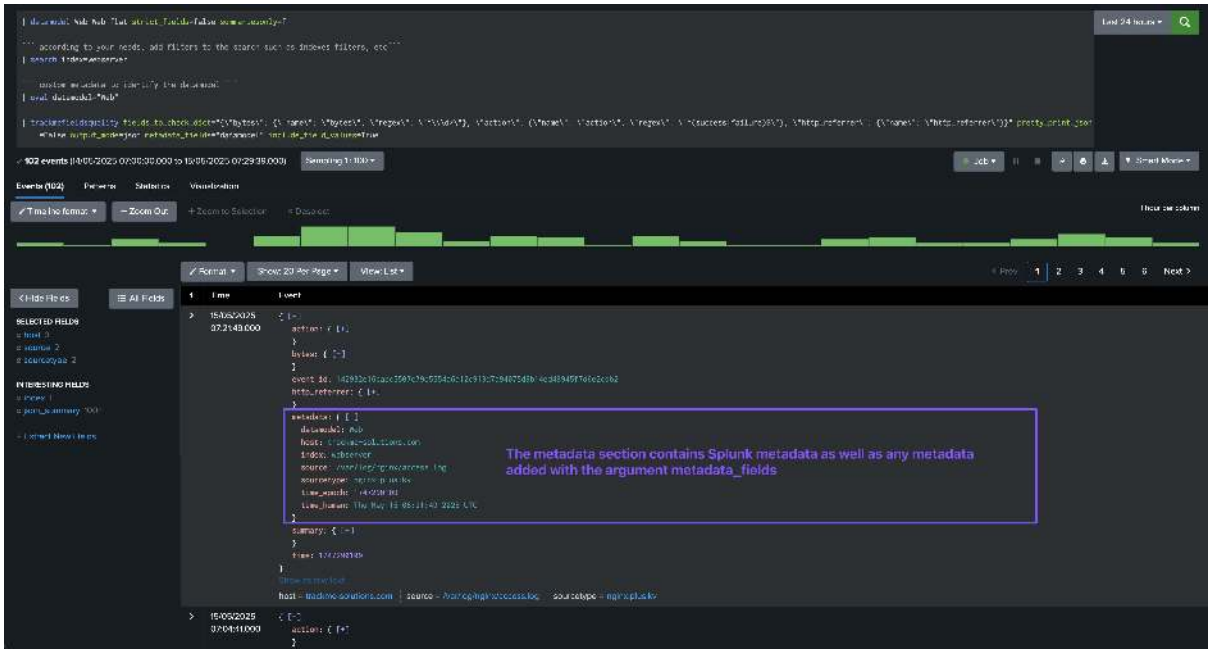
This option allows to specify the output mode, which can be `json` or `raw`.

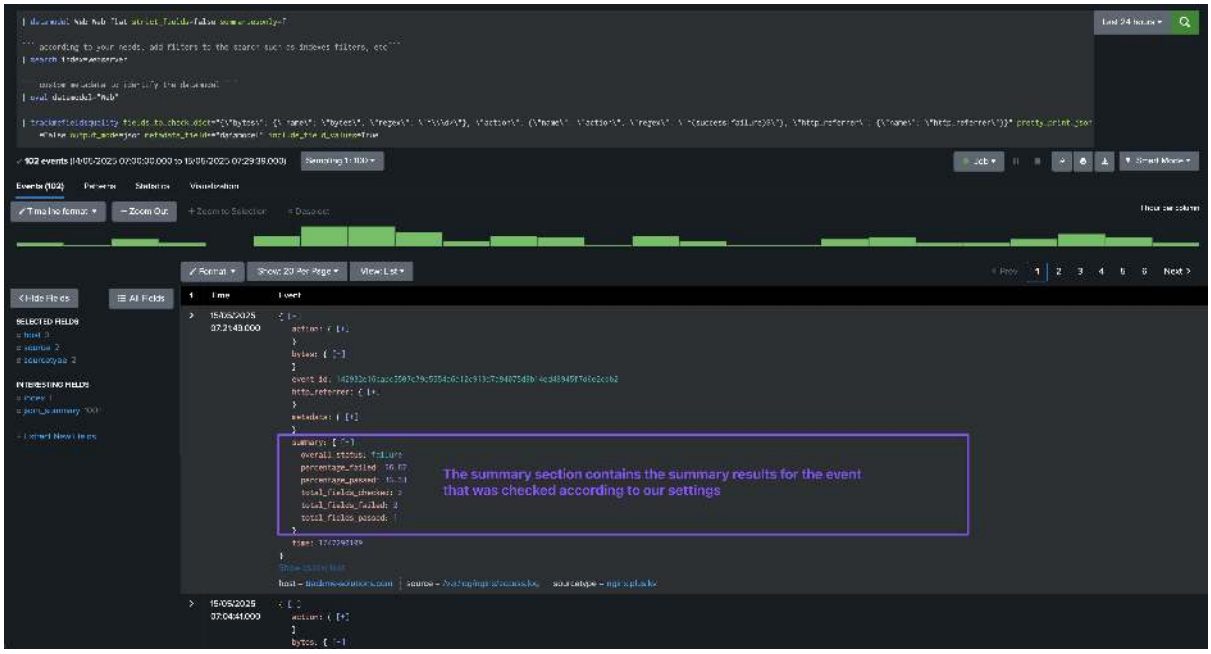
We recommend using the `json` mode for most use cases, especially this document is focused on the JSON output which is leveraged and indexed using the TrackMe sourcetype `trackme:fields_quality`.

output_mode=json

In JSON mode, the command verifies the fields and generates a JSON object per event:

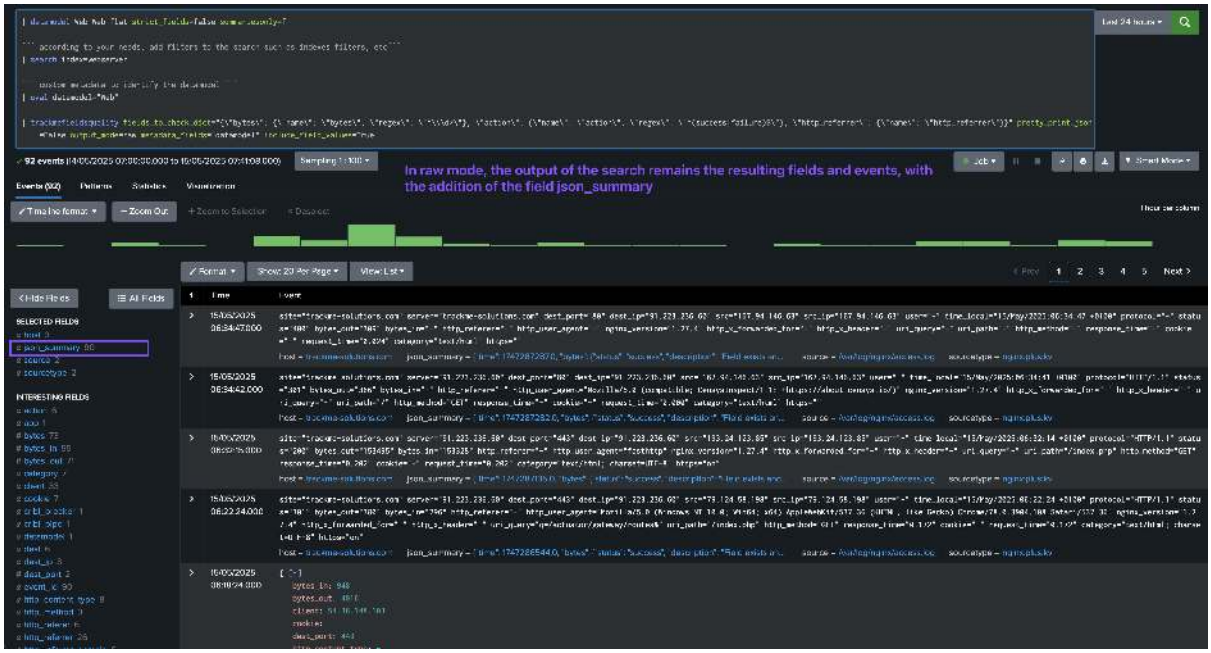


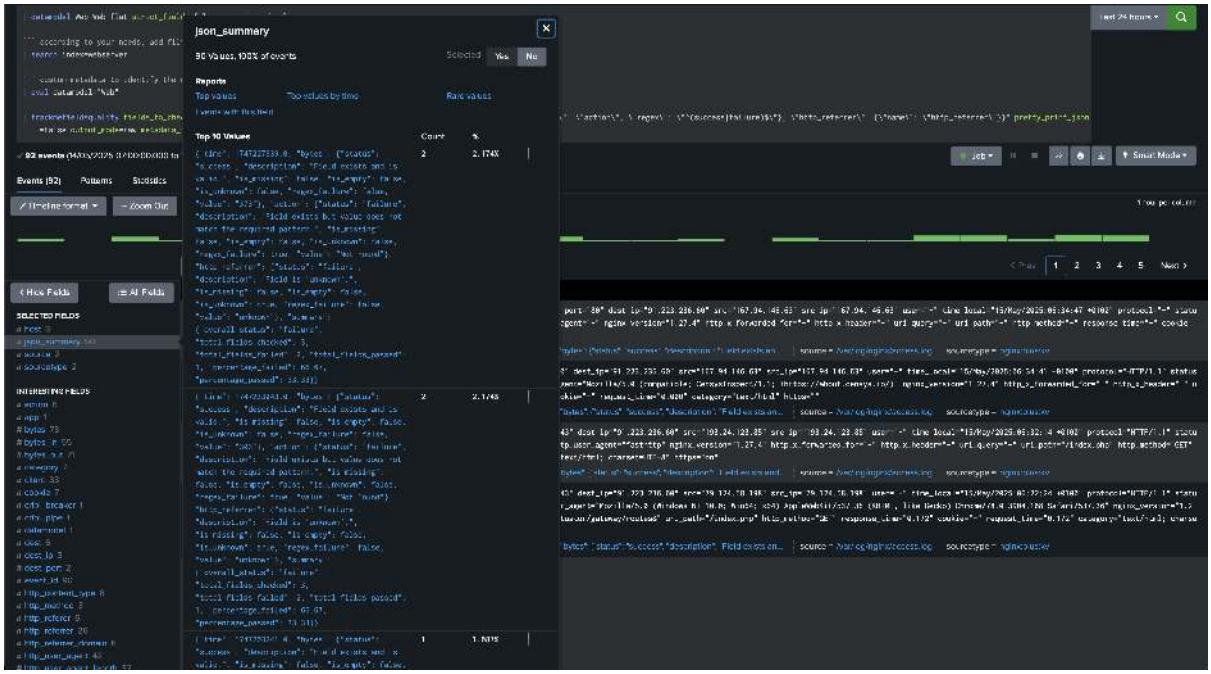




output_mode=raw

In raw mode, the command generates the events as they are, and adds a field called `json_summary` which contains our JSON object:





Argument: metadata_fields

This option allows to specify the metadata fields as a comma-separated list of fieldsto include in the JSON output.

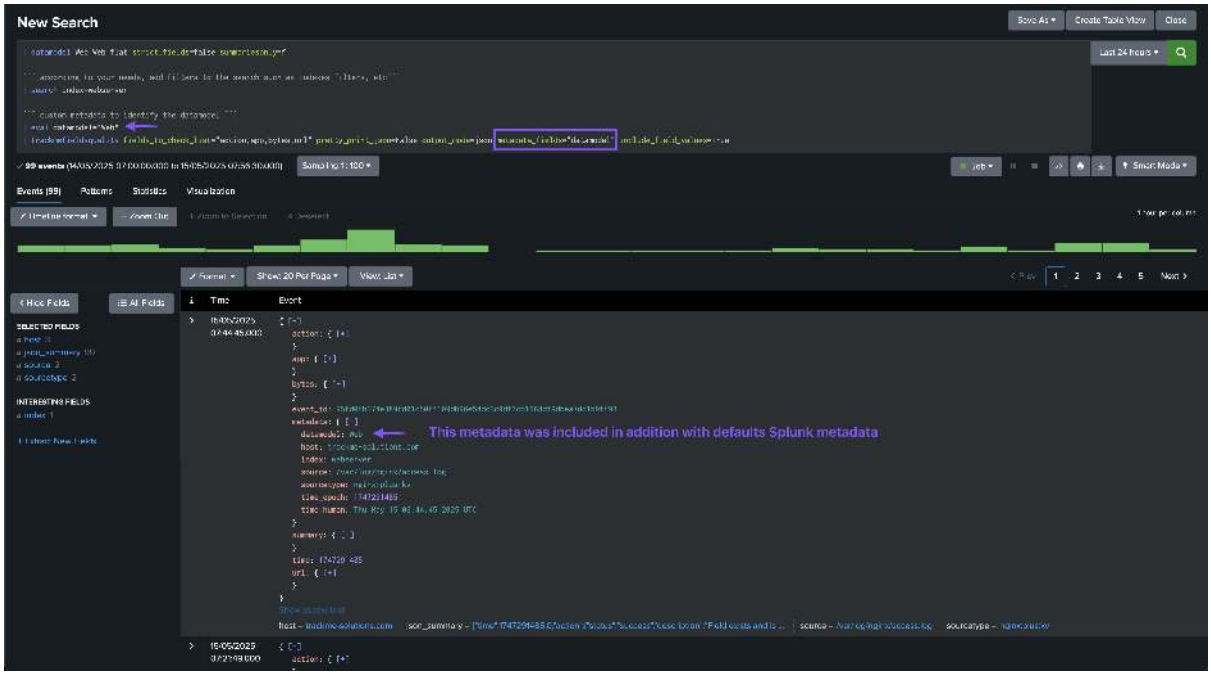
The metadata fields always include the following:

- `index`: the index of the event
- `sourcetype`: the sourcetype of the event
- `host`: the host of the event
- `source`: the source of the event

By defining the `metadata_fields` parameter, you can add additional fields to the JSON output, for instance the `datamodel` field which in our implementation is used to identify the data model of the event.

In our example, we are defining a field called `datamodel` using an eval which will be added to the JSON output:

```
| eval datamodel="Web"
| trackmefieldsquality fields_to_check_list="action,app,bytes,url" pretty_print_
↪json=False output_mode=json metadata_fields="datamodel" include_field_values=True
```

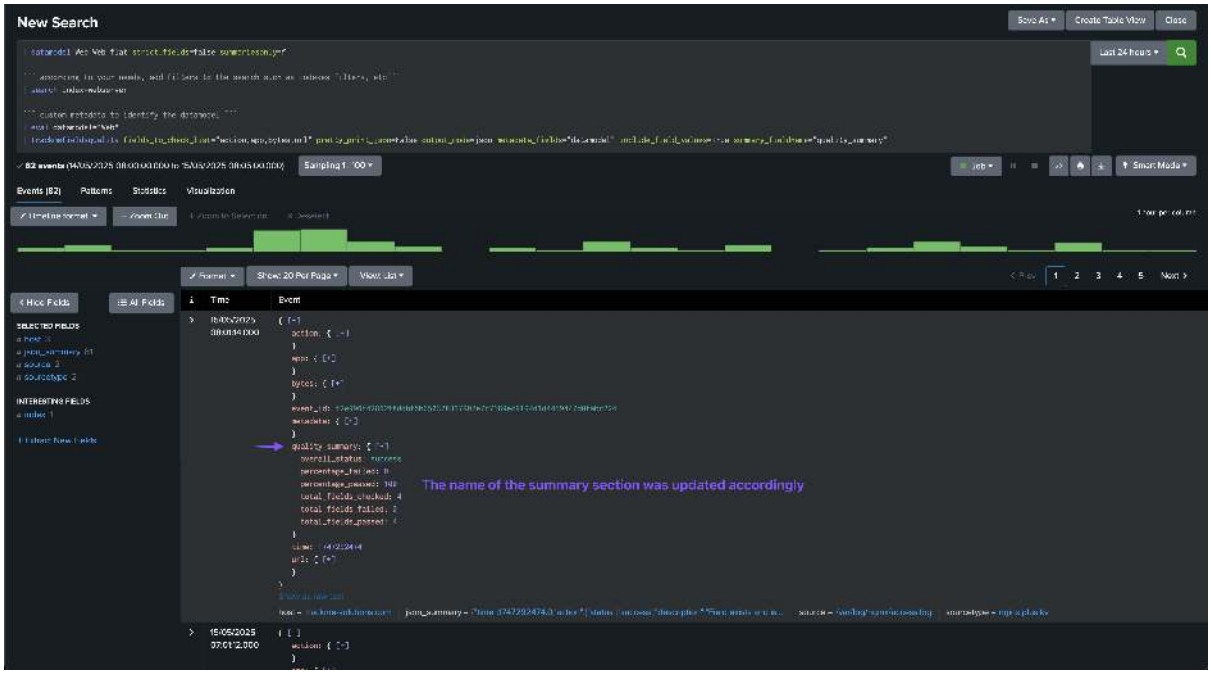


Argument: summary_fieldname

This option allows to specify the name of the summary field in the JSON output, this defaults to `summary` but can be customised if needed, for instance if there is a conflict with a field from the data model or events.

In this example, instead of `summary`, we use `quality_summary`:

```
| trackmefieldsquality fields_to_check_list="action,app,bytes,url" pretty_print_
↪json=False output_mode=json metadata_fields="datamodel" include_field_values=True␣
↪summary_fieldname="quality_summary"
```

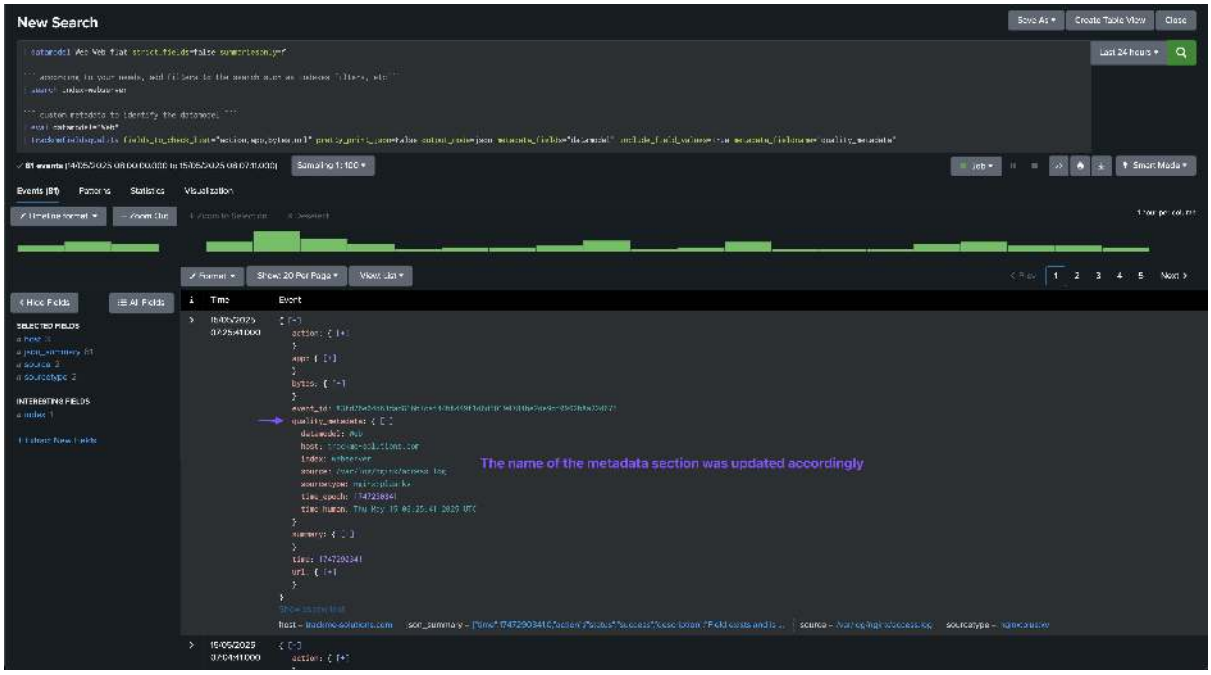


Argument: metadata_fieldname

This option allows to specify the name of the metadata field in the JSON output, this defaults to `metadata` but can be customised if needed, for instance if there is a conflict with a field from the data model or events.

In this example, instead of `metadata`, we use `quality_metadata`:

```
| trackmefieldsquality fields_to_check_list="action,app,bytes,url" pretty_print_
↪json=False output_mode=json metadata_fields="datamodel" include_field_values=True␣
↪metadata_fieldname="quality_metadata"
```



2.3.5 Annex: usage and options for the command trackmefieldsqualitygendict

The command `trackmefieldsqualitygendict` is used to generate JSON dictionary models for CIM data models, which can then be used with the `trackmefieldsquality` command for fields quality assessment.

The command accepts the following parameters:

argument	description	default	example or valid values
`datamodel`	The name of the CIM data model to generate the dictionary for	None (required)	"Authentication", "Web", "Network_Traffic"
`show_only_recomm`	Boolean option to only include recommended fields from the data model	False	True/False
`allow_unknown`	Boolean option to allow unknown field values	False	True/False
`allow_empty_or_m`	Boolean option to allow empty or missing field values	False	True/False

Usage:

```
| trackmefieldsqualitygendict datamodel=<datamodel name> show_only_recommended_fields=
↪<boolean> allow_unknown=<boolean> allow_empty_or_missing=<boolean>
```

Examples:

Generate dictionary for Authentication data model with only recommended fields:

```
| trackmefieldsqualitygendict datamodel="Authentication" show_only_recommended_
↪fields=true
```

Generate dictionary for Web data model with all fields and allow unknown values:

```
| trackmefieldsqualitygendict datamodel="Web" show_only_recommended_fields=false␣
↪allow_unknown=true
```

Generate dictionary for Network_Traffic data model with only recommended fields and allow empty/missing values:

```
| trackmefieldsqualitygendict datamodel="Network_Traffic" show_only_recommended_
↪fields=true allow_empty_or_missing=true
```

Generate dictionary for Authentication data model with all options enabled:

```
| trackmefieldsqualitygendict datamodel="Authentication" show_only_recommended_
↪fields=true allow_unknown=true allow_empty_or_missing=true
```

Output:

The command generates a JSON dictionary that can be used with the `trackmefieldsquality` command. The output includes:

- Field names as keys
- Field configuration objects as values, including: - `name`: The field name - `regex`: Optional regular expression pattern for validation - `allow_unknown`: Boolean flag to allow "unknown" values (can be overridden by the command parameter) - `allow_empty_or_missing`: Boolean flag to allow empty or missing values (can be overridden by the command parameter)

Example output for Web data model:


```
{
    "action": {
        "name": "action",
        "regex": "^(success|failure|allowed|blocked|deferred)$",
        "allow_unknown": false,
        "allow_empty_or_missing": false
    },
    "app": {
        "name": "app",
        "regex": "^[\\w:\\-\\d\\s]+$",
        "allow_unknown": false,
        "allow_empty_or_missing": false
    },
    "dest": {
        "name": "dest",
        "regex": "^[\\w\\.-]+$",
        "allow_unknown": false,
        "allow_empty_or_missing": false
    },
    "signature": {
        "name": "signature",
        "regex": "^.{3,100}$",
        "allow_unknown": false,
        "allow_empty_or_missing": false
    },
    "src": {
        "name": "src",
        "regex": "^[\\w\\.-]+$",
        "allow_unknown": false,
        "allow_empty_or_missing": false
    },
    "src_user": {
        "name": "src_user",
        "regex": "^[\\w\/\\\\\\-\\.$]{1,30}$",
        "allow_unknown": false,
        "allow_empty_or_missing": false
    },
    "user": {
        "name": "user",
        "regex": "^[\\w\/\\\\\\-\\.$]{1,30}$",
        "allow_unknown": false,
        "allow_empty_or_missing": false
    }
}
```


Integration with fields quality workflow:

This command is typically used in the first phase of the fields quality workflow to generate the dictionary models that will be stored in lookup tables and used by the collect scheduled searches:

```
| trackmefieldsqualitygendict datamodel=Authentication show_only_recommended_
↪fields=True
| append [ | trackmefieldsqualitygendict datamodel=Web show_only_recommended_
↪fields=True ]
| append [ | trackmefieldsqualitygendict datamodel=Network_Traffic show_only_
↪recommended_fields=True ]
```

(continues on next page)

(continued from previous page)

```
| table datamodel, json_dict
| trackmeprettyjson fields=json_dict
| outputlookup cim_datamodels_dict.csv
```

The generated lookup table can then be used in the collect scheduled searches:

```
| lookup cim_datamodels_dict.csv datamodel OUTPUT json_dict as quality_dict
| trackmefieldsquality fields_to_check_dict_fieldname="quality_dict" output_mode=json␣
↪metadata_fields="datamodel,nodename" include_field_values=True
```

2.3.6 Annex: usage and options for the command trackmefieldsqualitygensummary

The command `trackmefieldsqualitygensummary` is used to generate a summary of the quality of fields in records that have been processed by the `trackmefieldsquality` command. This command is typically used in the monitoring phase to aggregate and summarize field quality data for reporting and analysis purposes.

The command accepts the following parameters:

argument	description	default	example or valid values
`maxvals`	Max number of distinct values to report in field_values	15	10, 20, 50
`fieldvalues_format`	Format of field_values, either list or csv	csv	list, csv
`groupby_metadata_fiel`	Comma-separated list of metadata fields to group by in addition to fieldname	"" (empty)	"metadata.datamodel,metadata.nodename,meta

Usage:

```
| trackmefieldsqualitygensummary maxvals=<max number of distinct values to report>␣
↪fieldvalues_format=<format of field_values, either list or csv> groupby_metadata_
↪fields=<comma separated list of metadata fields to group by in addition to␣
↪fieldname>
```

Examples:

Generate summary with default settings:

```
| trackmefieldsqualitygensummary
```

Generate summary with custom maxvals and list format:

```
| trackmefieldsqualitygensummary maxvals=20 fieldvalues_format=list
```

Generate summary with grouping by metadata fields:

```
| trackmefieldsqualitygensummary maxvals=15 fieldvalues_format=csv groupby_metadata_
↪fields="metadata.datamodel,metadata.nodename,metadata.index,metadata.sourcetype"
```

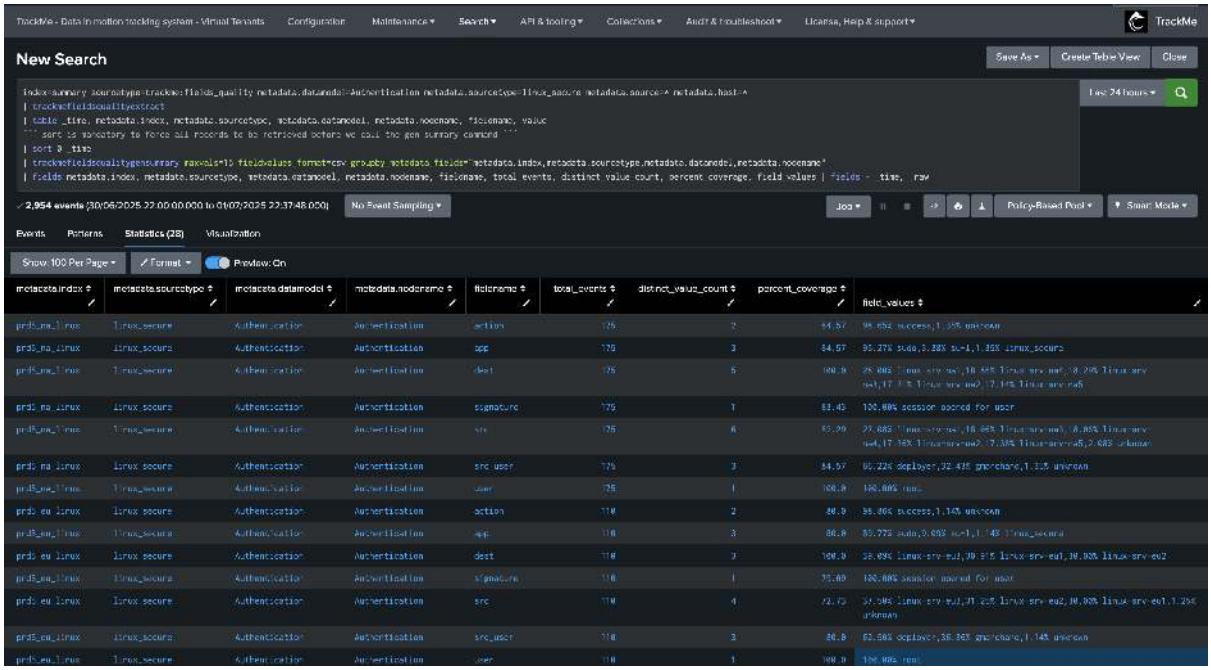
Output:

The command generates a summary that includes:

- **fieldname**: The name of the field being analyzed
- **total_events**: Total number of events processed for this field

- **distinct_value_count**: Number of distinct values found for this field
- **percent_coverage**: Percentage of events where this field has a value
- **field_values**: Summary of the most common values for this field (limited by maxvals)

Example output:



Example of field_values output:

The field_values column contains a summary of the most representative values for each field. For example:

- **CSV format**: "98.65% success,1.35% unknown"

Note: the list format renders the same information but as a proper list, which will appear in Splunk as a multi-value field.

Integration with fields quality workflow:

This command is typically used in the monitoring phase, specifically within the Flex Object tracker SPL logic to generate summaries of field quality data:

```
| search index=summary sourcetype=trackme:fields_quality metadata.sourcetype=*␣
↪metadata.source=* metadata.host=*
| trackmefieldsqualityextract
| table _time, metadata.index, metadata.sourcetype, metadata.datamodel, metadata.
↪nodename, fieldname, value
| sort 0 _time
| trackmefieldsqualitygensummary maxvals=15 fieldvalues_format=csv groupby_metadata_
↪fields="metadata.index,metadata.sourcetype,metadata.datamodel,metadata.nodename"
```

The output from this command is then used to populate the `extra_attributes` field in the Flex Object tracker, providing detailed field quality information for each entity.

Argument: maxvals

This option controls the maximum number of distinct values to report in the field_values column. This is useful for limiting the output size while still providing meaningful insights into the most common values for each field.

Example:

```
| trackmefieldsqualitygensummary maxvals=10
```

This would limit the field_values output to show only the top 10 most common values for each field.

Argument: fieldvalues_format

This option controls the format of the field_values output. Two formats are supported:

- **csv**: Comma-separated values format (default)
- **list**: List format with percentages in parentheses

CSV format example: "98.65% success,1.35% unknown"

List format example: "success (98.65%), unknown (1.35%)"

Example:

```
| trackmefieldsqualitygensummary fieldvalues_format=list
```

Argument: groupby_metadata_fields

This option allows grouping the summary by additional metadata fields beyond the default fieldname grouping. This is particularly useful when you want to analyze field quality across different dimensions such as data models, nodes, indexes, or sourcetypes.

Example:

```
| trackmefieldsqualitygensummary groupby_metadata_fields="metadata.datamodel,metadata.
↪nodename,metadata.index,metadata.sourcetype"
```

This would generate separate summaries for each combination of datamodel, nodename, index, and sourcetype, allowing for more granular analysis of field quality across different contexts.

Integration example in Flex Object tracker:

In the Flex Object tracker source code, this command is used to generate detailed field summaries that are stored in the entity's extra_attributes:

```
| join type=outer metadata.index, metadata.sourcetype, metadata.datamodel, metadata.
↪nodename, fieldname [ search index=summary sourcetype=trackme:fields_quality␣
↪metadata.sourcetype=* metadata.source=* metadata.host=*
| trackmefieldsqualityextract
| table _time, metadata.index, metadata.sourcetype, metadata.datamodel, metadata.
↪nodename, fieldname, value
| sort 0 _time
| trackmefieldsqualitygensummary maxvals=15 fieldvalues_format=csv groupby_metadata_
↪fields="metadata.index,metadata.sourcetype,metadata.datamodel,metadata.nodename"
| fields metadata.index, metadata.sourcetype, metadata.datamodel, metadata.nodename,␣
↪fieldname, total_events, distinct_value_count, percent_coverage, field_values |␣
↪fields - _time, _raw
]

``` generate the per field json for reporting purposes, we also rename fieldname to␣
↪@fieldname so it appears first in the JSON ```
| rename fieldname as @fieldname, fieldstatus as @fieldstatus
| tojson
| rename _raw as per_field_json

``` calculate ```
| stats values(eval(if('@fieldstatus'=="success", '@fieldname', null()))) as list_
```

(continues on next page)

(continued from previous page)

```
↪fields_passed, values(eval(if('@fieldstatus'=="failure", '@fieldname', null()))) as␣
↪list_fields_failed, values(per_field_json) as per_field_json, max(total_events) as␣
↪total_events_parsed by metadata.index, metadata.sourcetype, metadata.datamodel,␣
↪metadata.nodename

``` format the per_field_json ```
| eval per_field_json = "[" . mvjoin(per_field_json, ", ") . "]"

``` build the list of fields that passed and failed ```
| eval all_fields = mvappend(list_fields_failed, list_fields_passed)
| eval all_fields = mvdedup(all_fields)
| eval final_state = mvmap(
all_fields,
if(
mvfind(list_fields_failed, "^" . all_fields . "$") >= 0,
all_fields . "|failed",
all_fields . "|success"
)
)
| eval success_fields = mvfilter(match(final_state, "\|success$"))
| eval failed_fields = mvfilter(match(final_state, "\|failed$"))
| eval success_fields = mvmap(success_fields, mvindex(split(success_fields, "|"), 0))
| eval failed_fields  = mvmap(failed_fields,  mvindex(split(failed_fields, "|"), 0))
| fields - final_state
| eval success_fields=if(isnull(success_fields), "", success_fields), failed_
↪fields=if(isnull(failed_fields), "", failed_fields)
| fields - list_fields_passed, list_fields_failed

``` calculate ```
| eventstats dc(all_fields) as total_fields_checked, dc(success_fields) as total_
↪fields_passed, dc(failed_fields) as total_fields_failed by metadata.index, metadata.
↪sourcetype, metadata.datamodel, metadata.nodename
| eval percentage_passed=round(total_fields_passed/total_fields_checked*100, 2),␣
↪percentage_failed=round(total_fields_failed/total_fields_checked*100, 2)

``` rename ```
| rename metadata.* as "*"

``` save this as parts of extra attributes ```
| eval extra_attributes = "{" . "\"success_fields\": \"" . mvjoin(success_fields, ",
↪") . "\", \"failed_fields\": \"" . mvjoin(failed_fields, ",") . "\", " . "\"fields\
↪":" . per_field_json . "}"
| fields - per_field_json

``` set principal metadata for the flex entity ```
| eval group = datamodel
| eval object = nodename . ":" . index . ":" . sourcetype, alias=index . ":" .␣
↪sourcetype
| eval object_description = "CIM Quality for DM: " . datamodel . ":" . nodename . ",␣
↪index:" . index . ", sourcetype:" . sourcetype

``` gen metrics ```
| eval metrics = "{" .
"\"fields_quality.percentage_passed\": " . if(isnum(percentage_passed), percentage_
↪passed, 0) . ", " .
"\"fields_quality.percentage_failed\": " . if(isnum(percentage_failed), percentage_
```

(continues on next page)
```

(continued from previous page)

```
↪failed, 0) . ", " .
"\"fields_quality.total_fields_checked\": " . if(isnum(total_fields_checked), total_
↪fields_checked, 0) . ", " .
"\"fields_quality.total_fields_failed\": " . if(isnum(total_fields_failed), total_
↪fields_failed, 0) . ", " .
"\"fields_quality.total_fields_passed\": " . if(isnum(total_fields_passed), total_
↪fields_passed, 0) . ", " .
"\"fields_quality.total_events_parsed\": " . if(isnum(total_events_parsed), total_
↪events_parsed, 0) . "}"

``` no outliers for now ```
| eval outliers_metrics="{}"

``` basic status, thresholds can be defined on a per entity basis ```
| eval status=1
| eval status_description="DM Quality: " . metrics
| eval status_description_short="% passed: " . percentage_passed . ", checked: " .␣
↪total_fields_checked . ", passed: " . total_fields_passed . ", failed: " . total_
↪fields_failed

``` this sets a default threshold, which can then be overriden on a per entity basis␣
↪via the TrackMe UI ```
| eval default_threshold = "{'metric_name': 'fields_quality.percentage_passed',
↪'operator': '>=', 'value': 95, 'condition_true': 1}"

```alert if inactive for more than 2 days```
| eval max_sec_inactive=86400*2
```

2.3.7 Annex: Understanding & troubleshooting the percentage of success

The percentage of success is calculated by the main Flex tracker SPL logic, it represents the percentage of times the field is passing the quality check, a field is considered as passing the quality check if:

- `is_empty: false`
- `is_null: false`
- `is_unknown: false` (depending on if a dictionary is used and if in the dictionary and for this field, unknown is allowed or not)
- `regex_failure: false` (depending on if a dictionary is used and if there is a regex expression for this field)

```
index=summary sourcetype=trackme:fields_quality metadata.sourcetype=* metadata.
↪source=* metadata.host=*
| table _time, _raw
| trackmefieldsqualityextract

| stats count as count_total, count(eval(status=="success")) as count_success,␣
↪count(eval(status=="failure")) as count_failure by fieldname, metadata.index,␣
↪metadata.sourcetype, metadata.datamodel, metadata.nodename
| eval percentage_success=round(count_success/count_total*100, 2)
```

In this example, we are looking at the percentage of success for the field "action" in the datamodel Network_Traffic and nodename All_Traffic, for the sourcetype cisco:asa:

```
index=summary sourcetype=trackme:fields_quality metadata.datamodel=Network_Traffic␣
↪metadata.nodename=All_Traffic metadata.sourcetype=cisco:asa metadata.source=*␣
↪metadata.host=*
| table _time, _raw
| trackmefieldsqualityextract

| stats values(value) as values, values(regex_expression) as regex_expression, count␣
↪as count_total, count(eval(status=="success")) as count_success, count(eval(status==
↪"failure")) as count_failure by fieldname, metadata.index, metadata.sourcetype,␣
↪metadata.datamodel, metadata.nodename
| eval percentage_success=round(count_success/count_total*100, 2)

``` set the threshold per field, this defines if the field is considered as passed or␣
↪failed globally, this threshold has to be part of the SPL logic ```
| eval threshold=99

``` flag field ```
| eval fieldstatus=if(percentage_success>=threshold, "success", "failure")

| where fieldname="action"
```

2.3.8 Annex: Understanding and troubleshooting the percentage of coverage and others summary information per field

The percentage of coverage is reported by the `trackmefieldsqualitygensummary` command, it represents the percentage of events where the field has a value, that is not null and not empty. (it is for instance present equally in the CIM Vladiator table)

In the Flex tracker, this is the join part:

```
search index=summary sourcetype=trackme:fields_quality metadata.sourcetype=* metadata.
↪source=* metadata.host=*
| trackmefieldsqualityextract
| table _time, metadata.index, metadata.sourcetype, metadata.datamodel, metadata.
↪nodename, fieldname, value, regex_expression
``` sort is mandatory to force all records to be retrieved before we call the gen␣
↪summary command ```
| sort 0 _time
| trackmefieldsqualitygensummary maxvals=15 fieldvalues_format=csv groupby_metadata_
↪fields="metadata.index,metadata.sourcetype,metadata.datamodel,metadata.nodename"
| fields metadata.index, metadata.sourcetype, metadata.datamodel, metadata.nodename,␣
↪fieldname, total_events, distinct_value_count, percent_coverage, field_values,␣
↪regex_expression | fields - _time, _raw
```

You can for example run this search to get the percentage of coverage for the field "action" in the datamodel Network_Traffic and nodename All_Traffic, for the sourcetype cisco:asa:

```
index=summary sourcetype=trackme:fields_quality metadata.datamodel=Network_Traffic␣
↪metadata.nodename=All_Traffic metadata.sourcetype=cisco:asa metadata.source=*␣
↪metadata.host=*
| table _time, _raw
| trackmefieldsqualityextract

| table _time, metadata.index, metadata.sourcetype, metadata.datamodel, metadata.
↪nodename, fieldname, value, regex_expression
``` sort is mandatory to force all records to be retrieved before we call the gen␣
↪summary command ```
```

(continues on next page)

(continued from previous page)

```
| sort 0 _time
| trackmefieldsqualitygensummary maxvals=15 fieldvalues_format=csv groupby_metadata_
↪fields="metadata.index,metadata.sourcetype,metadata.datamodel,metadata.nodename"
| fields metadata.index, metadata.sourcetype, metadata.datamodel, metadata.nodename,␣
↪fieldname, total_events, distinct_value_count, percent_coverage, field_values,␣
↪regex_expression | fields - _time, _raw

| where fieldname="action"
```

This would be equivalent to run the following search:

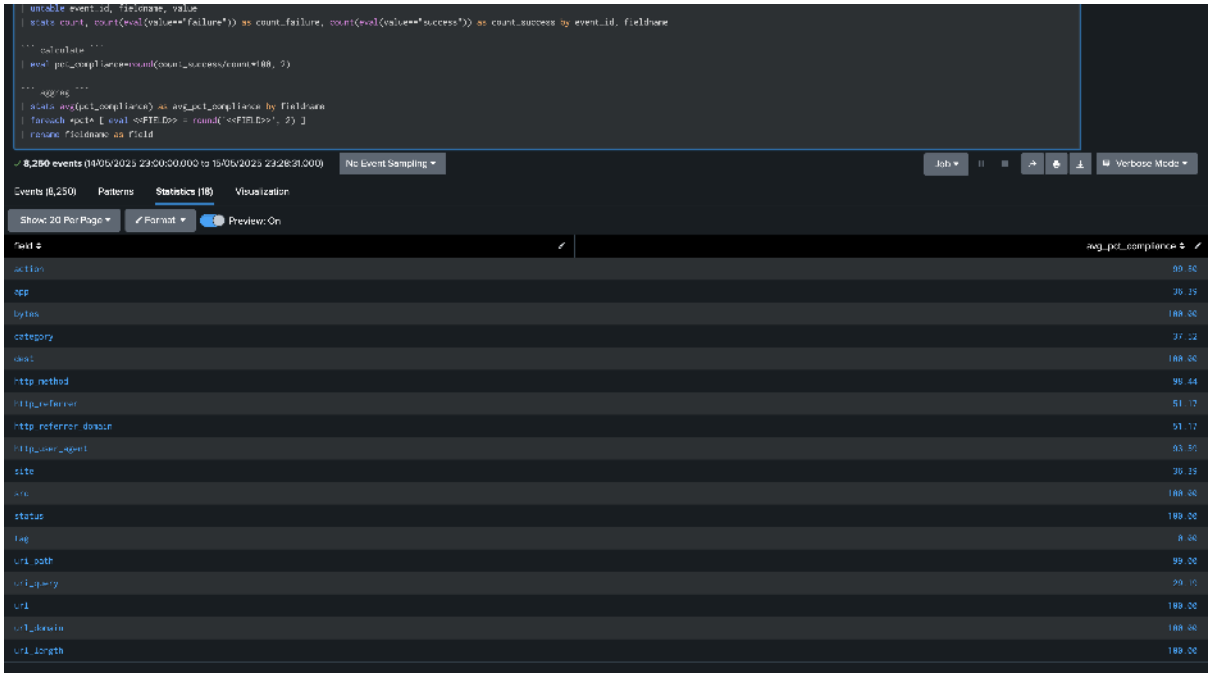
```
index=summary sourcetype=trackme:fields_quality metadata.datamodel=Network_Traffic␣
↪metadata.nodename=All_Traffic metadata.sourcetype=cisco:asa metadata.source=*␣
↪metadata.host=* | head 100
| table _time, _raw
| trackmefieldsqualityextract
| table _time, metadata.index, metadata.sourcetype, metadata.datamodel, metadata.
↪nodename, fieldname, value, is_empty, is_missing

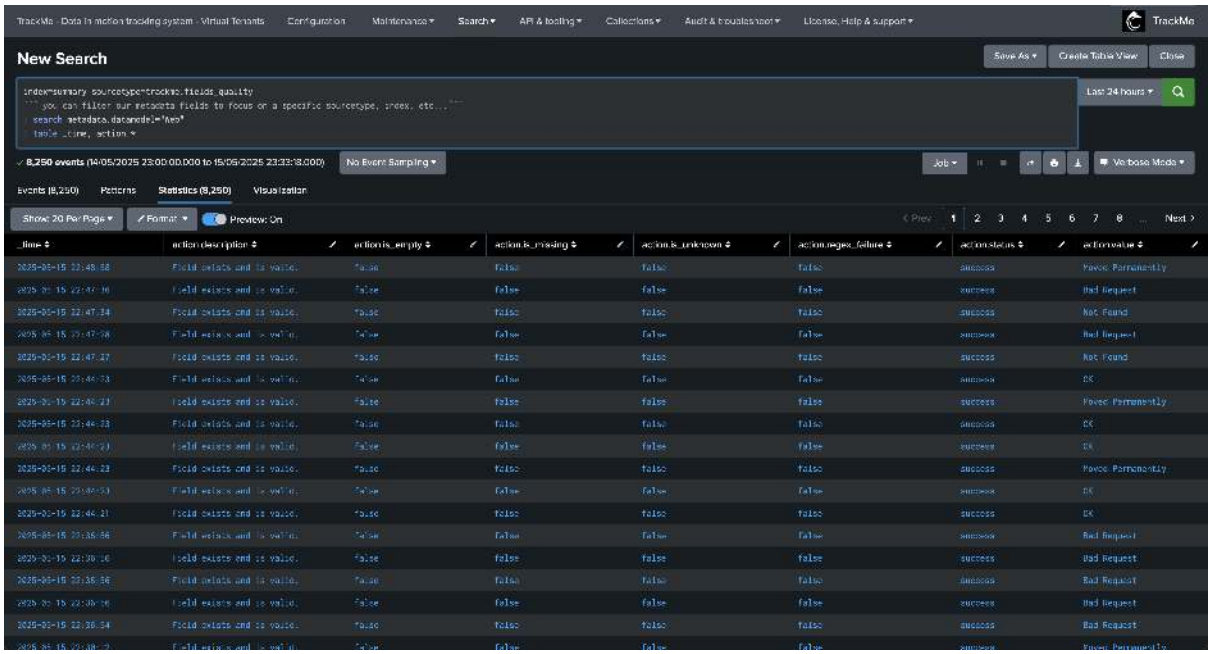
| where fieldname="action"

| stats count, count(eval(is_empty=0 AND is_missing=0)) as count_covered,␣
↪count(eval(is_empty=1 OR is_missing=1)) as count_not_covered
| eval percentage_covered=count_covered/count*100
```

2.3.9 Annex: Per field table statistics

The following search example shows the statistics per field:

```
index=summary sourcetype=trackme:fields_quality
``` you can filter our metadata fields to focus on a specific sourcetype, index, etc..
↪.```
| search metadata.datamodel="Web"

``` stats ```
| fields - summary.* metadata.*
| stats first(*status) as "*status" by event_id
| rename "*.status" as "*"

``` untable ```
| untable event_id, fieldname, value
| stats count, count(eval(value=="failure")) as count_failure, count(eval(value==
↪"success")) as count_success by event_id, fieldname

``` calculate ```
| eval pct_compliance=round(count_success/count*100, 2)

``` aggreg ```
| stats avg(pct_compliance) as avg_pct_compliance by fieldname
| foreach *pct* [ eval <<FIELD>> = round('<<FIELD>>', 2) ]
| rename fieldname as field
```
```



2.3.10 Annex: Looking after a specific field

The following search example shows the statistics per field:

```
index=summary sourcetype=trackme:fields_quality
``` you can filter our metadata fields to focus on a specific sourcetype, index, etc..
↪.```
| search metadata.datamodel="Web"
| table _time, action.*
```


```

2.3.11 Annex: Flex Object tracker SPL logic

The following search example shows the Flex Object tracker SPL logic:

```
index=summary sourcetype=trackme:fields_quality metadata.sourcetype=* metadata.
↪source=* metadata.host=*
| table _time, _raw
| trackmefieldsqualityextract

| stats count as count_total, count(eval(status=="success")) as count_success,␣
↪count(eval(status=="failure")) as count_failure by fieldname, metadata.index,␣
↪metadata.sourcetype, metadata.datamodel, metadata.nodename
| eval percentage_success=round(count_success/count_total*100, 2)

``` set the threshold per field, this defines if the field is considered as passed or␣
↪failed globally, this threshold has to be part of the SPL logic ```
| eval threshold=99

``` flag field ```
| eval fieldstatus=if(percentage_success>=threshold, "success", "failure")

``` join field summary ```
| join type=outer metadata.index, metadata.sourcetype, metadata.datamodel, metadata.
↪nodename, fieldname [ search index=summary sourcetype=trackme:fields_quality␣
↪metadata.sourcetype=* metadata.source=* metadata.host=*
| trackmefieldsqualityextract
| table _time, metadata.index, metadata.sourcetype, metadata.datamodel, metadata.
↪nodename, fieldname, value, regex_expression
``` sort is mandatory to force all records to be retrieved before we call the gen␣
↪summary command ```
| sort 0 _time
| trackmefieldsqualitygensummary maxvals=15 fieldvalues_format=csv groupby_metadata_
↪fields="metadata.index,metadata.sourcetype,metadata.datamodel,metadata.nodename"
| fields metadata.index, metadata.sourcetype, metadata.datamodel, metadata.nodename,␣
↪fieldname, total_events, distinct_value_count, percent_coverage, field_values,␣
↪regex_expression | fields - _time, _raw
]

``` generate the per field json for reporting purposes, we also rename fieldname to␣
↪@fieldname so it appears first in the JSON ```
| rename fieldname as @fieldname, fieldstatus as @fieldstatus
| tojson
| rename _raw as per_field_json

``` calculate ```
| stats values(eval(if('@fieldstatus'=="success", '@fieldname', null()))) as list_
↪fields_passed, values(eval(if('@fieldstatus'=="failure", '@fieldname', null()))) as␣
↪list_fields_failed, values(per_field_json) as per_field_json, max(total_events) as␣
↪total_events_parsed by metadata.index, metadata.sourcetype, metadata.datamodel,␣
↪metadata.nodename

``` format the per_field_json ```
| eval per_field_json = "[" . mvjoin(per_field_json, ", ") . "]"

``` build the list of fields that passed and failed ```
| eval all_fields = mvappend(list_fields_failed, list_fields_passed)
| eval all_fields = mvdedup(all_fields)
| eval final_state = mvmap(
```

(continues on next page)

(continued from previous page)

```
all_fields,
if(
mvfind(list_fields_failed, "^" . all_fields . "$") >= 0,
all_fields . "|failed",
all_fields . "|success"
)
)
| eval success_fields = mvfilter(match(final_state, "\|success$"))
| eval failed_fields = mvfilter(match(final_state, "\|failed$"))
| eval success_fields = mvmap(success_fields, mvindex(split(success_fields, "|"), 0))
| eval failed_fields  = mvmap(failed_fields,  mvindex(split(failed_fields, "|"), 0))
| fields - final_state
| eval success_fields=if(isnull(success_fields), "", success_fields), failed_
↪fields=if(isnull(failed_fields), "", failed_fields)
| fields - list_fields_passed, list_fields_failed

``` calculate ```
| eventstats dc(all_fields) as total_fields_checked, dc(success_fields) as total_
↪fields_passed, dc(failed_fields) as total_fields_failed by metadata.index, metadata.
↪sourcetype, metadata.datamodel, metadata.nodename
| eval percentage_passed=round(total_fields_passed/total_fields_checked*100, 2),␣
↪percentage_failed=round(total_fields_failed/total_fields_checked*100, 2)

``` rename ```
| rename metadata.* as "*"

``` save this as parts of extra attributes ```
| eval extra_attributes = "{" . "\"success_fields\": \"" . mvjoin(success_fields, ",
↪") . "\", \"failed_fields\": \"" . mvjoin(failed_fields, ",") . "\", " . "\"fields\
↪":" . per_field_json . "}"
| fields - per_field_json

``` set principal metadata for the flex entity ```
| eval group = datamodel
| eval object = nodename . ":" . index . ":" . sourcetype, alias=index . ":" .␣
↪sourcetype
| eval object_description = "CIM Quality for DM: " . datamodel . ":" . nodename . ",␣
↪index:" . index . ", sourcetype:" . sourcetype

``` gen metrics ```
| eval metrics = "{" .
"\"fields_quality.percentage_passed\": " . if(isnum(percentage_passed), percentage_
↪passed, 0) . ", " .
"\"fields_quality.percentage_failed\": " . if(isnum(percentage_failed), percentage_
↪failed, 0) . ", " .
"\"fields_quality.total_fields_checked\": " . if(isnum(total_fields_checked), total_
↪fields_checked, 0) . ", " .
"\"fields_quality.total_fields_failed\": " . if(isnum(total_fields_failed), total_
↪fields_failed, 0) . ", " .
"\"fields_quality.total_fields_passed\": " . if(isnum(total_fields_passed), total_
↪fields_passed, 0) . ", " .
"\"fields_quality.total_events_parsed\": " . if(isnum(total_events_parsed), total_
↪events_parsed, 0) . "}"

``` no outliers for now ```
| eval outliers_metrics="{}"
```

(continues on next page)

(continued from previous page)

```
``` basic status, thresholds can be defined on a per entity basis ```
| eval status=1
| eval status_description="DM Quality: " . metrics
| eval status_description_short="% passed: " . percentage_passed . ", checked: " .␣
↪total_fields_checked . ", passed: " . total_fields_passed . ", failed: " . total_
↪fields_failed

``` this sets a default threshold, which can then be overriden on a per entity basis␣
↪via the TrackMe UI ```
| eval default_threshold = "{'metric_name': 'fields_quality.percentage_passed',
↪'operator': '>=', 'value': 95, 'condition_true': 1}"

```alert if inactive for more than 2 days```
| eval max_sec_inactive=86400*2
```
```

CHAPTER

THREE

LICENSE & SUPPORT:

3.1 EULA License

3.1.1 TrackMe Limited (TrackMe) EULA

End-User License Agreement ("Agreement")

Our EULA was last updated on 10 February 2025.

We provide the Application, also known as TrackMe. In this Agreement, you means the person or entity using the Application. If you are agreeing to this Agreement not as an individual but on behalf of your company, government, or other entity for which you are acting (for example, as an employee or governmental official), then you means your entity and you are binding your entity to this Agreement.

Please read this End-User License Agreement carefully before clicking the "I Agree" button, downloading or using TrackMe, Data Tracking system for Splunk.

Interpretation and Definitions

Interpretation

The words of which the initial letter is capitalized have meanings defined under the following conditions. The following definitions shall have the same meaning regardless of whether they appear in singular or in plural.

Definitions

For the purposes of this End-User License Agreement:

- "Agreement" means this End-User License Agreement that forms the entire agreement between you and the Company regarding the use of the Application.
- "Application" means the software program provided by the Company downloaded by You through an Application Store's account to a device, named TrackMe, Data Tracking system for Splunk.
- "Application Store" or "Splunk Base" means the digital distribution service operated and developed by Splunk Inc. ("Splunk") by which the Application can be downloaded to your Splunk Enterprise or Splunk Cloud environment.
- "Company" (referred to as either "the Company", "We", "Us" or "Our" in this Agreement) refers to TrackMe Limited, U.K.
- "Consequential Loss" includes any consequential loss, indirect loss, real or anticipated loss of profit, loss of benefit, loss of revenue, loss of business, loss of goodwill, loss of opportunity, loss of savings, loss of reputation, loss of use and/or loss or corruption of data, whether under statute, contract, equity, tort (including negligence), indemnity or otherwise.
- "Content" refers to content such as text, images, or other information that can be posted, uploaded, linked to or otherwise made available by You, regardless of the form of that content.
- "Country" refers to: United Kingdom.

- "Third-Party Services" means any services or content (including data, information, applications and other products services) provided by a third-party that may be displayed, included or made available by the Application.

Acknowledgment

By clicking on "Agree to Download" button in SplunkBase Website, or downloading or using the Application, you are agreeing to be bound by the terms and conditions of this Agreement. If you do not agree to the terms of this Agreement, do not click on the "Agree to Download" button in SplunkBase Website, do not download or do not use the Application.

You must be at least 16 years old to use the Application.

This Agreement is a legal document between you and the Company and it governs your use of the Application made available to you by the Company.

This Agreement is between you and the Company only and not with the Application Store. Therefore, the Company is solely responsible for the Application and its content.

The Application is licensed, not sold, to you by the Company for use strictly in accordance with the terms of this Agreement.

We may amend this Agreement at any time, by providing written notice to you, unless otherwise stated in this Agreement. By clicking "I accept" or continuing to use the Application after the notice or 30 days after notification (whichever date is earlier), you agree to the amended Agreement. If you do not agree to the amendment, you may cease accessing the Application.

License

Scope of License

Subject to your compliance with this Agreement, the Company grants you a revocable, non-exclusive, non-transferable, royalty free, limited license to download, install and use the Application strictly in accordance with the terms of this Agreement. All other uses are prohibited without our prior written consent.

You may only use the Application on a device that you own or control and as permitted by the Application Store's terms and conditions.

License Restrictions

When using the Application, you agree not to, and you will not permit others to:

- do or attempt to do anything that is unlawful or inappropriate;
- anything that would constitute a breach of an individual's privacy (including uploading private or personal information without an individual's consent) or any other legal rights;
- use the Application to defame, harass, threaten, menace or offend any person, including using the Application to send unsolicited electronic messages;
- tamper with or modifying the Application (including by transmitting viruses and using trojan horses);
- use data mining, robots, screen scraping or similar data gathering and extraction tools on the Application;
- License, sell, rent, lease, assign, distribute, transmit, host, outsource, disclose or otherwise commercially exploit the Application or make the Application available to any third party; or
- Remove, alter or obscure any proprietary notice (including any notice of copyright or trademark) of the Company or its affiliates, partners, suppliers or the licensors of the Application.

Intellectual Property

The Application, including without limitation all copyrights, patents, trademarks, trade secrets and other intellectual property rights (**Our Intellectual Property**) are, and shall remain, the sole and exclusive property of the Company.

We authorise you to use Our Intellectual Property solely for your limited commercial use. You must not exploit Our Intellectual Property for any other purpose, nor allow, aid or facilitate such use by any third party.

You must not, without our prior written consent:

- *(a)* copy, in whole or in part, any of Our Intellectual Property;
- *(b)* reproduce, retransmit, distribute, disseminate, sell, publish, broadcast or circulate any of Our Intellectual Property to any third party; or
- *(c)* breach any intellectual property rights connected with the Application, including (without limitation) altering or modifying any of Our Intellectual Property, causing any of Our Intellectual Property to be framed or embedded in another website, or creating derivative works from any of Our Intellectual Property.

Modifications to the Application

The Company reserves the right to modify, suspend or discontinue, temporarily or permanently, the Application or any service to which it connects, with notice and without liability to you.

Updates to the Application

The Company may from time to time provide enhancements or improvements to the features/functionality of the Application, which may include patches, bug fixes, updates, upgrades and other modifications.

Updates may modify or delete certain features and/or functionalities of the Application. You agree that the Company has no obligation to (i) continue to provide or enable any particular features and/or functionalities of the Application to you.

You further agree that all updates or any other modifications will be (i) deemed to constitute an integral part of the Application, and (ii) subject to the terms and conditions of this Agreement.

Maintenance and Support

The Company provides maintenance and support for licensed customers, customers using the Free community edition are not entitled to any maintenance or support, To the extent that any maintenance or support is required by applicable law, the Company, shall be obligated to furnish any such maintenance or support.

Third-Party Services

The Application may display, include or make available third-party content (including data, information, applications and other products services) or provide links to third-party websites or services.

You acknowledge and agree that the Company shall not be responsible for any Third-party Services. To the maximum extent permitted by law, we shall have no liability for any Third-Party Services, or any unavailability of the Application due to a failure of the Third-Party Services.

Our solution use a number of third party components, which are documented at the following address: https://docs.trackme-solutions.com/latest/third_party_credits.html

By using our product, you also agree to the terms and conditions of these third party components.

Indemnification

Neither party may benefit from the limitations and exclusions set out in this clause in respect of any liability arising from its deliberate default.

The restrictions on liability in this Indemnification clause apply to every liability arising under or in connection with this Agreement including liability in statute, contract, equity, tort (including negligence), misrepresentation, restitution, indemnity or otherwise.

Nothing in this Agreement limits any liability which cannot legally be limited, including liability for:

- *(a)* death or personal injury caused by negligence;
- *(b)* fraud or fraudulent misrepresentation;

- *(c)* breach of the terms implied by section 2 of the Supply of Goods and Services Act 1982 (title and quiet possession); and
- *(d)* defective products under the Consumer Protection Act 1987.

To the maximum extent permitted by law:

- *(a)* you agree to indemnify us for any liability we incur due to your breach of the Acknowledgement and/or Licence clauses and the Intellectual Property clause of this Agreement;
- *(b)* neither party will be liable for Consequential Loss;
- *(c)* each party's liability for any liability under this Agreement will be reduced proportionately to the extent the relevant liability was caused or contributed to by the acts or omissions of the other party or any of that party's personnel, including any failure by that Party to mitigate its losses; and
- *(d)* our aggregate liability for any liability arising from or in connection with this Agreement will be limited to us resupplying the Application to you.

No Warranties

2.1 You represent, warrant and agree that:

- *(a)* you will not use our Application, including Our Intellectual Property, in any way that competes with our business;
- *(b)* if you are agreeing to this Agreement not as an individual but on behalf of your company, government, or other entity for which you are acting (for example, as an employee or governmental official), then you access and use the Application on behalf of that entity;
- *(c)* there are no legal restrictions preventing you from entering into this Agreement;
- *(d)* all information and documentation that you provide to us in connection with this Agreement is true, correct and complete; and
- *(e)* you have not relied on any representations or warranties made by us in relation to the Application (including as to whether the Application is or will be fit or suitable for your particular purposes), unless expressly stipulated in this Agreement.

Severability

If any provision of this Agreement is held to be unenforceable or invalid, such provision will be changed and interpreted to accomplish the objectives of such provision to the greatest extent possible under applicable law and the remaining provisions will continue in full force and effect.

Waiver

Except as provided herein, the failure to exercise a right or to require performance of an obligation under this Agreement shall not affect a party's ability to exercise such right or require such performance at any time thereafter nor shall the waiver of a breach constitute a waiver of any subsequent breach.

Access, Term and Termination

This Agreement shall remain in effect until terminated by you or the Company. The Company may, in its sole discretion, at any time and for any or no reason, suspend or terminate this Agreement with or without prior notice.

We may revoke access to the Application at any time by giving 30 days' written notice to you.

Should we suspect that you are in breach of this Agreement, we may suspend your access to the Application while we investigate the suspected breach. Should we determine that you are in breach of this Agreement, your access to the Application will be terminated immediately.

You may also terminate this Agreement by deleting the Application and all copies thereof from your device or from your computer.

Upon termination of this Agreement, you shall cease all use of the Application and delete all copies of the Application from your device.

Termination of this Agreement will not limit any of the Company's rights or remedies at law or in equity in case of breach by you (during the term of this Agreement) of any of your obligations under the present Agreement

Termination and Exit Strategy

TrackMe Limited provides customers with clear termination rights and an exit strategy to ensure a seamless transition in case of service discontinuation.

Termination Rights:

- Customers may terminate their agreement with TrackMe by providing a written notice of 30 days, to be sent by email to contact@trackme-solutions.com.
- TrackMe reserves the right to terminate the agreement if the customer breaches its terms, fails to comply with applicable regulations, or engages in unauthorized use of the services.

Exit Strategy and Data Retrieval:

- Upon termination, customers will have 30 days to download or request a copy of their data before deletion.
- TrackMe will provide reasonable assistance in transitioning data to another service provider if requested.
- After the grace period, all customer data will be securely deleted, except where legal or regulatory retention requirements apply.

These provisions ensure that customers maintain control over their data and can transition away from TrackMe services with minimal disruption.

Cooperation with Regulatory Authorities

TrackMe Limited is committed to compliance with applicable regulatory frameworks, including the Digital Operational Resilience Act (DORA). As part of this commitment, TrackMe agrees to:

- **Regulatory Requests**: Cooperate with financial regulators, supervisory authorities, and other relevant bodies by providing necessary information related to operational resilience, security, and risk management.
- **Inspections and Assessments**: Allow authorized regulators to conduct inspections and assessments of TrackMe's systems and processes related to ICT risk management.
- **Incident Reporting Assistance**: Provide timely support to financial entities in fulfilling their regulatory obligations, including the reporting of ICT-related incidents.
- **Compliance Documentation**: Maintain detailed records of security, performance, and business continuity measures, making them available to regulators upon request.

These commitments ensure transparency and regulatory compliance while upholding the security and confidentiality of TrackMe's services.

Incident Reporting and Notification

TrackMe Limited prioritizes transparency and accountability in managing ICT-related incidents. In accordance with regulatory requirements, including the Digital Operational Resilience Act (DORA), TrackMe follows a structured incident reporting framework.

Incident Notification Commitments:

- **Significant Incidents**: TrackMe will notify affected customers within 24 hours of detecting a security breach, data compromise, or major service disruption.
- **Regulatory Assistance**: TrackMe will provide customers with relevant details needed to fulfill their regulatory incident reporting obligations.
- **Communication Channels**: Incident notifications will be sent via TrackMe's support helpdesk.
- **Remediation Updates**: TrackMe will provide ongoing updates about mitigation efforts and expected resolution timelines for critical incidents.

TrackMe ensures that all incidents are logged, investigated, and remediated promptly, with continuous improvement to prevent recurrence.

Information Security and Compliance

TrackMe Limited is committed to maintaining high security standards to protect customer data and ensure operational resilience. TrackMe adheres to industry-recognized security frameworks and best practices, including:

- **ISO 27001 Compliance**: TrackMe follows the principles of ISO 27001 for information security management.
- **SOC 2 Controls**: TrackMe implements security, availability, and confidentiality controls aligned with SOC 2 standards.
- **Regular Security Audits**: TrackMe conducts periodic internal and external security audits to validate compliance with regulatory and security requirements.
- **Secure Development Practices**: TrackMe employs secure coding practices, vulnerability testing, and continuous monitoring to safeguard its applications.
- **Customer Assurance**: Upon request, TrackMe can provide security documentation detailing compliance measures and risk management processes.

By adhering to these standards, TrackMe ensures the security, integrity, and confidentiality of its services, in compliance with DORA and other applicable regulations.

Access and Audit Rights

TrackMe Limited acknowledges the importance of regulatory compliance and operational resilience. To ensure adherence to applicable financial regulations, including the Digital Operational Resilience Act (DORA), TrackMe grants authorized financial entities and regulatory authorities the right to conduct audits and assessments of TrackMe's services related to operational resilience, cybersecurity, and incident response.

Such audits shall be conducted with reasonable notice and during normal business hours. TrackMe shall provide all necessary documentation, access to relevant systems, and support to facilitate these audits, ensuring transparency and compliance with applicable laws.

Financial entities using TrackMe's services may also conduct internal assessments, subject to confidentiality and security agreements.

Service Performance Standards

TrackMe Limited is committed to maintaining a high standard of service performance to support operational resilience and regulatory compliance. As part of this commitment, TrackMe ensures the following service levels:

- **Availability**: TrackMe aims to maintain an uptime of 99.5% measured on a monthly basis, excluding planned maintenance periods.
- **Response Time**: TrackMe's support team will respond to critical service issues within twenty fours (24) business hours.
- **Maintenance and Updates**: Regular maintenance and security updates will be communicated in advance, with minimal disruption to service.
- **Data Processing Performance**: TrackMe ensures that all data processing activities meet industry standards for accuracy and efficiency.

Any deviations from these service levels will be investigated and remediated promptly. TrackMe Limited will also provide customers with performance reports upon request.

Service Locations and Data Processing

TrackMe Limited operates its services in secure data centers located in the United Kingdom and EU. Customer data may be processed and stored in these locations based on applicable regulatory and compliance requirements.

- **Primary Hosting Locations**: TrackMe Limited uses cloud service providers and data centers located in [List countries or regions].
- **Data Transfers**: Any data transfers between regions comply with applicable data protection laws, including the General Data Protection Regulation (GDPR) where applicable.
- **Subcontractors**: TrackMe may engage third-party cloud providers or partners for infrastructure services. A list of such subcontractors is available upon request.
- **Customer Control**: Customers may request information on where their specific instance of TrackMe services is hosted.

TrackMe Limited ensures that all service locations meet high security and operational resilience standards.

Data Protection and Confidentiality

TrackMe Limited is committed to protecting the confidentiality, integrity, and security of customer data. All data collected, processed, and stored within TrackMe services is handled in compliance with applicable data protection laws, including but not limited to the General Data Protection Regulation (GDPR).

Key Data Protection Measures: - **Data Encryption**: Customer data is encrypted at rest and in transit using industry-standard encryption protocols. - **Access Controls**: Strict access controls ensure that only authorized personnel can access sensitive data. - **Data Minimization**: TrackMe collects and processes only the minimum necessary customer data required for service operation. - **Confidentiality Commitments**: TrackMe employees, contractors, and third-party service providers are bound by confidentiality obligations. - **Customer Rights**: Customers have the right to request information on how their data is processed and may request data deletion or correction.

TrackMe Limited continuously evaluates and improves its security measures to maintain compliance with evolving regulatory requirements.

Business Continuity and Disaster Recovery

TrackMe Limited is committed to ensuring the resilience and availability of its services, even in the event of disruptions. To maintain operational stability, TrackMe has implemented comprehensive business continuity and disaster recovery plans.

Key Resilience Measures: - **Redundant Infrastructure**: TrackMe operates on a highly available infrastructure with redundant components to minimize downtime. - **Regular Backups**: Customer data is backed up at regular intervals to prevent data loss and ensure quick recovery in case of incidents. - **Incident Response Procedures**: TrackMe maintains a structured incident response framework to detect, investigate, and mitigate potential threats. - **Disaster Recovery Plan**: A formal disaster recovery plan is in place, ensuring timely restoration of services in the event of system failures. - **Testing and Reviews**: Continuity plans and recovery strategies are regularly tested and updated to adapt to evolving risks.

In case of a major disruption, TrackMe will promptly inform customers and provide updates on recovery efforts.

Governing Law

The laws of the Country, excluding its conflicts of law rules, shall govern this Agreement and your use of the Application. Your use of the Application may also be subject to other local, state, national, or international laws.

Entire Agreement

The Agreement constitutes the entire agreement between you and the Company regarding your use of the Application and supersedes all prior and contemporaneous written or oral agreements between you and the Company.

You may be subject to additional terms and conditions that apply when you use or purchase other Company's services, which the Company will provide to you at the time of such use or purchase.

Assignment

You must not assign or deal with the whole or any part of your rights or obligations under this Agreement without our prior written consent.

Contracts (Rights of Third Parties) Act 1999

Notwithstanding any other provision of this Agreement, nothing in this Agreement confers or is intended to confer any right to enforce any of its terms on any person who is not a party to it.

Contact Us

If you have any questions about this Agreement, you can contact Us:

- By visiting this page on our website: www.trackme-solutions.com
- By sending us an email: contact@trackme-solutions.com

3.2 Terms & Conditions

Our Terms & Conditions were last updated on 9 April 2024.

3.2.1 TrackMe Limited IT Software Terms and Conditions

In these terms and conditions (Terms), when we say you or your, we mean both you and, if applicable, any entity you are authorised to represent (such as your employer). When we say we, us, or our, we mean TrackMe Limited, a company registered in England and Wales with company number 13817409.

Please read these Terms carefully before using our downloadable platform ("TrackMe") for Splunk Enterprise, Splunk Cloud software or before accessing TrackMe through our direct website.

These Terms form our contract with you, and set out our obligations as a service provider and your obligations as a customer. You cannot use TrackMe unless you agree to these Terms. By using our software, you agree to be bound by these Terms.

1. License Types & Restrictions

You and Each Authorised User must be at least 16 years old to use TrackMe. If you are using TrackMe on behalf of your employer or a business entity, you, in your individual capacity, represent and warrant that you are authorised to act on behalf of your employer or the business entity and to bind the entity and the entity's personnel to these Terms.

You represent, warrant and agree that:

- *(a)* you will not use TrackMe, including Our Intellectual Property, in any way that competes with our business;
- *(b)* there are no legal restrictions preventing you from entering into these Terms;
- *(c)* all information and documentation that you provide to us in connection with these Terms is true, correct and complete; and
- *(d)* you have not relied on any representations or warranties made by us in relation to TrackMe (including as to whether TrackMe is or will be fit or suitable for your particular purposes), unless expressly stipulated in these Terms.

You must not:

- *(a)* access or use TrackMe in any way that is improper or breaches any laws, infringes any person's rights (for example, intellectual property rights and privacy rights), or gives rise to any civil or criminal liability;
- *(b)* interfere with or interrupt the supply of TrackMe, or any other person's access to or use of TrackMe;
- *(c)* introduce any viruses or other malicious software code into TrackMe;
- *(d)* use any unauthorised or modified version of TrackMe, including but not limited to for the purpose of building similar or competitive software or for the purpose of obtaining unauthorised access to TrackMe;

- *(e)* attempt to access any data or log into any server or account that you are not expressly authorised to access;
- *(f)* use TrackMe in any way that involves service bureau use, outsourcing, renting, reselling, sublicensing, concurrent use of a single user login, or time-sharing;
- *(g)* circumvent user authentication or security of any of our networks, accounts or hosts or those of any third party; or
- *(h)* access or use TrackMe to transmit, publish or communicate material that is, defamatory, offensive, abusive, indecent, menacing, harassing or unwanted.

License Types:

While you have an account with Splunk or with us (Account), we grant you a right to use TrackMe (which may be suspended or revoked in accordance with these Terms). This right cannot be passed on or transferred to any other person. The types of licences that we offer will be set out on our website or on the Splunk Enterprise or Splunk Cloud software (as applicable).

2. Support

We offer support for features and behaviors which relate directly to the usage of TrackMe for Splunk Cloud and Splunk Enterprise. Issues related to Splunk products or any other third-party applications are not covered by our support agreement. Prioritized support is included for customers with a valid TrackMe Enterprise or TrackMe Unlimited. Support is available by email. We will use reasonable endeavours to respond to all email enquiries within 24 hours.

Contacting TrackMe support:

- Customers with a valid TrackMe Enterprise or TrackMe Unlimited can contact us at support@trackme-solutions.com
- If the support case requires it, we will propose a virtual support meeting through our Zoom meeting services, or your own meeting tools depending on your preferences

Prioritized developments:

- If you have specific needs or requirements that are not covered by our product, we will prioritize developments if you are a customer under active subscription.
- The company remains the sole owner of the developments and the intellectual property of the developments made for the product.
- We also reserve the right not to implement a feature if it is not in line with our product roadmap or if it is not technically feasible, although we will always do our best to meet your requirements.

Unless we agree otherwise, the support we may agree to provide cannot be used to support any other products or services, and does not include training, installation of software or hardware, software development or the modification, deletion or recovery of data or any on-site services.

Additional Services: If you require additional services, we may, in our sole discretion, provide such additional services (to be scoped and priced in a separate contract provided by us).

Third Party Products or Services: Where you engage third parties to operate alongside our services (for example, any third-party software systems you wish to integrate with TrackMe), those third parties are independent of us and you are responsible for (meaning we will not be liable for) the goods or services they provide, unless we expressly agree otherwise.

3. License Subscription Renewal and Expiration

If you wish not to renew your TrackMe Enterprise or TrackMe Unlimited license, TrackMe will automatically return to the free Community Edition mode, and components out of the scope of the Free Limited Edition mode will automatically be disabled. If you renew your subscription, your license will be extended accordingly. You are responsible for paying any levies or taxes associated with your use of TrackMe, for example sales taxes, value-added taxes or withholding taxes (unless we are required by law to collect these on your behalf.

4. Changes to TrackMe

While we strive to always make TrackMe available to you, we do not make any guarantees that it will be available 100% of the time. Our services may be disrupted during certain periods, including, for example, as a result of scheduled or emergency maintenance. Our services (including TrackMe) may interact with, or be reliant on, products or services provided by third parties, such as cloud hosting service providers. To the maximum extent permitted by law, we are not liable for disruptions or downtime caused or contributed to by these third parties. We will try to provide you with reasonable notice, where possible, of any disruptions to your access to TrackMe.

Minor changes to TrackMe: We may change TrackMe:

- *(a)* to reflect changes in relevant laws and regulatory requirements; and
- *(b)* to implement minor technical adjustments, improvements, and to keep up-to-date with technological advancements. These changes will not substantially affect your use of TrackMe.

More significant changes to TrackMe: We will try to avoid making any significant changes to TrackMe which are likely to materially disadvantage your use of TrackMe. However, where we intend to make a change to TrackMe which may materially disadvantage your use of TrackMe, we will notify you in advance of making any changes, and you may then contact us to terminate these Terms and receive a full refund for any unused services (if applicable) before the changes take effect.

5. Changes to the Terms

We reserve the right to modify, amend, or update these Terms at any time and for any reason without prior notice. Such modifications, amendments, or updates will be effective immediately upon posting on our website.

To ensure transparency and keep our customers fully informed of any changes that may affect their use of the TrackMe software for Splunk Enterprise and Splunk Cloud, we commit to sending an automated notification of any such changes to the designated point of contact for each customer. This notification will be sent to the email address on file with us for such communications.

Customers are encouraged to periodically review the most current version of the Terms, which will always be available on our website. Continued use of the TrackMe software after any changes to the Terms signifies acceptance of those changes. It is the responsibility of the customer to update their designated point of contact and corresponding email address as necessary to ensure that notifications of changes are received in a timely manner.

If you disagree with the revised Terms, you have the right to terminate your use of the TrackMe software and we agree to refund you for any prepaid unused Fees on a pro-rata basis.

6. Payments

The license can be activated in your Splunk environment at any time, with or without online verification through Cryptolens Licensing API.

For more details and technical information about the Cryptolens API service and our registration process, please consult the following documentation: https://docs.trackme-solutions.com/latest/license_registration.html

We provide a license key which requires online verification through Cryptolens Licensing API, and a license file which allows offline registration for Splunk environments with no outside communications capabilities.

Once you have chosen the licence that you require, you agree to pay the fees set out on TrackMe (Fees) by the date specified on TrackMe to access certain features of TrackMe. You agree that the Fees will be payable in advance of you receiving access to TrackMe and your chosen licences (if licence Fees are payable). We accept payments by bank transfer once we issue a tax invoice (U.K. national and international IBAN, payments can be processed in USD, GBP AND EUR currencies).

7. Delivery

After you agree to our Terms and accept our quotation, we will issue an invoice through our billing system. Once the invoice is generated and the Fees are paid as set out above, we will provide you with the TrackMe license key and license file via our licensing services.

The TrackMe Enterprise or TrackMe Unlimited license key and license key file which allows offline registration is provided by electronic means, we will also deliver the license through our DocuSign service to electronically sign the delivery.

8. Authorised Users

You may be permitted to invite a number of users to TrackMe, who will be permitted to download and use TrackMe under your license (**Authorised Users**).

You must ensure that each Authorised User complies with these Terms and our End User Licence Agreement. You are responsible and liable for the acts or omissions of your Authorised Users.

Each Authorised User must agree to our End User Licence Agreement (as available at [insert URL]) in order to download and use TrackMe.

9. Legal Compliance and Jurisdiction

We are committed to ensuring that any information we store is secure and required for the purposes of providing our services to you.

As a Software which runs on your environment, whenever you are a Splunk Enterprise or Splunk Cloud customer, we do not store any personal data, and we do not have access to your data.

Information stored by us is restricted to the minimal required information for support and billing purposes notably the end customer name and contact details and the TrackMe Enterprise or TrackMe Unlimited license details.

10. Termination and Refund Policy

Clause 10 is designed to ensure clear communication and procedures for the termination of your TrackMe subscription. This policy outlines the conditions under which subscriptions may be terminated and clarifies our stance on refunds.

Termination by the User:

Subscribers may decide not to renew their subscription at any point without the need for formal termination, given that TrackMe does not automatically renew subscriptions. Should you wish to discontinue your use of TrackMe before the end of your current subscription period, we request that you notify us in writing at contact@trackme-solutions.com. While early termination does not entitle the subscriber to a refund for the remaining portion of the subscription, it ensures that our records are up-to-date, and we can cease any renewal reminders.

No Refund Policy:

In alignment with our commitment to transparency and the digital nature of our software licenses, TrackMe maintains a no refund policy. Once a license is issued and activated, we cannot offer refunds. We encourage customers to thoroughly evaluate our software through available trials or developer modes before making a purchase. Our team is available to address any questions or concerns you might have prior to purchasing.

Termination by us:

We reserve the right to terminate a subscription if there are grounds such as non-compliance with our Terms, misuse of the software, or failure to comply with payment terms. Such termination will be communicated in writing, and the subscription will be deactivated immediately. In cases of termination by us for cause, no refund will be provided.

Subscription Expiration:

If you choose not to renew your subscription, it will expire at the end of the current billing cycle. Upon expiration, access to the Full Registered Edition features will be suspended, and the software will revert to the Free Limited Edition mode, where applicable.

Policy Changes:

We reserve the right to modify Clause 10 at any time. Customers will be notified of significant changes through direct communication or via updates on our website. We value our relationship with our customers and aim to provide clear and fair policies. If you have any questions or need further clarification

regarding our Subscription Termination Policy, please do not hesitate to contact us at contact@trackme-solutions.com."

This revised section ensures consistency with the operational practices mentioned earlier, emphasizing user control over subscription management and the company's no refund policy. It provides clear instructions for both users and the company regarding subscription termination and outlines the conditions under which these actions can take place.

11. Intellectual Property

You acknowledge and agree that any Intellectual Property or content (including copyright and trademarks) available on TrackMe, TrackMe itself (including how it looks and functions), any algorithms or machine learning models used on TrackMe, as well as our copyrighted works, trademarks, inventions, designs and other intellectual property (**Our Intellectual Property**) will at all times vest, or remain vested, in us.

We authorise you to use and download Our Intellectual Property solely for your limited commercial use. You must not exploit Our Intellectual Property for any other purpose, nor allow, aid or facilitate such use by any third party. Use must be limited to Authorised Users on devices that are controlled or approved by you.

You must not, without our prior written consent:

- *(a)* copy, in whole or in part, any of Our Intellectual Property;
- *(b)* reproduce, retransmit, distribute, disseminate, sell, publish, broadcast or circulate any of Our Intellectual Property to any third party; or
- *(c)* breach any intellectual property rights connected with TrackMe, including (without limitation) altering or modifying any of Our Intellectual Property, causing any of Our Intellectual Property to be framed or embedded in another website, or creating derivative works from any of Our Intellectual Property.

This Clause 11 will survive the termination or expiry of these Terms.

12. Liability

Neither party may benefit from the limitations and exclusions set out in this clause in respect of any liability arising from its deliberate default.

The restrictions on liability in this 'Liability' clause apply to every liability arising under or in connection with these Terms including liability in statute, contract, equity, tort (including negligence), misrepresentation, restitution, indemnity or otherwise.

Nothing in these Terms limits any Liability which cannot legally be limited, including Liability for:

- *(a)* death or personal injury caused by negligence;
- *(b)* fraud or fraudulent misrepresentation; and
- *(c)* defective products under the Consumer Protection Act 1987.

To the maximum extent permitted by law, we shall have no Liability for any third party products or services, or any unavailability of TrackMe due to a failure of the third party products or services.

To the maximum extent permitted by law:

- *(a)* you agree to indemnify us for any Liability we incur due to your breach of these Terms;
- *(b)* neither party will be liable for any Consequential Loss; and
- *(c)* a party's liability for any Liability under these Terms will be reduced proportionately to the extent the relevant liability was caused or contributed to by the acts or omissions of the other party, including any failure by that other party to mitigate its loss; and
- *(d)* our aggregate liability for any Liability arising from or in connection with these Terms will be limited to us resupplying the services to you or, in our sole discretion, to us repaying you the amount of the fees paid by you to us in respect of the supply of the relevant services to which the Liability relates.

The terms implied by sections 3, 4 and 5 of the Supply of Goods and Services Act 1982 are, to the maximum extent permitted by law, excluded from these Terms.

This 'Liability' clause will survive the termination or expiry of these Terms.

13. General

Assignment: Subject to the below clause, a party must not assign or deal with the whole or any part of its rights or obligations under these Terms without the prior written consent of the other party (such consent is not to be unreasonably withheld).

Assignment of Debt: You agree that we may assign or transfer any debt owed by you to us, arising under or in connection with these Terms, to a debt collector, debt collection agency, or other third party.

Contracts (Rights of Third Parties) Act 1999: Notwithstanding any other provision of these Terms, nothing in these Terms confers or is intended to confer any right to enforce any of its terms on any person who is not a party to it.

Disputes: Neither we or you may commence court proceedings relating to any dispute, controversy or claim arising from, or in connection with, these Terms (including any question regarding its existence, validity or termination) (Dispute) unless we and you first meet (in good faith) to resolve the Dispute. Nothing in this clause will operate to prevent us or you from seeking urgent injunctive or equitable relief from a court of appropriate jurisdiction. If the Dispute is not resolved at that initial meeting:

- *(a)* where you are resident or incorporated in England and Wales, refer the matter to mediation, administered by The Centre for Effective Dispute Resolution; or
- *(b)* where you are not resident or incorporated in England and Wales, refer the matter to arbitration administered by the London Court of International Arbitration (LCIA), with such arbitration to be conducted in London, before one arbitrator, in English and in accordance with the LCIA Arbitration Rules.

Force Majeure: To the maximum extent permitted by law, we shall have no Liability for any event or circumstance outside of our reasonable control.

Marketing: You agree that we may send you electronic communications about our products and services. You may opt-out at any time by using the unsubscribe function in our electronic communications

Governing law: These Terms are governed by the laws of England and Wales. Each party irrevocably and unconditionally submits to the exclusive jurisdiction of the courts operating in England and Wales and any courts entitled to hear appeals from those courts and waives any right to object to proceedings being brought in those courts. TrackMe may be accessed in the UK and overseas. We make no representation that TrackMe complies with the laws (including intellectual property laws) of any country outside of the UK. If you access TrackMe from outside the UK, you do so at your own risk and are responsible for complying with the laws in the place you access TrackMe. The United Nations Convention of Contracts for the International Sale of Goods is expressly excluded from these Terms.

Privacy: All personal data you and your Authorised Users provide to us will be treated in accordance with our privacy policy. You can find our privacy policy at [insert link].

Severance: If a provision of these Terms is held to be void, invalid, illegal or unenforceable, that provision is to be read down as narrowly as necessary to allow it to be valid or enforceable, failing which, that provision (or that part of that provision) will be severed from these Terms without affecting the validity or enforceability of the remainder of that provision or the other provisions in these Terms.

Third party sites: TrackMe may contain links to websites operated by third parties. Unless we tell you otherwise, we do not control, endorse or approve, and are not responsible for, the content on those websites. We recommend that you make your own investigations with respect to the suitability of those websites. If you purchase goods or services from a third party website linked from TrackMe, such third party provides the goods and services to you, not us. We may receive a benefit (which may include a referral fee or a commission) should you visit certain third-party websites via a link on TrackMe (Affiliate Link) or for featuring certain products or services on TrackMe. We will make it clear by notice to you which (if any) products or services we receive a benefit to feature on TrackMe, or which (if any) third party links are Affiliate Links.

14. Definitions

Consequential Loss includes any consequential loss, indirect loss, real or anticipated loss of profit, loss of benefit, loss of revenue, loss of business, loss of goodwill, loss of opportunity, loss of savings, loss of reputation, loss of use and/or loss or corruption of data, whether under statute, contract, equity, tort (including negligence), indemnity or otherwise.

Intellectual Property means any copyright, registered or unregistered designs, patents or trade marks, business names, get-up, goodwill, domain names, know-how, inventions, processes, trade secrets or confidential information, circuit layouts, software, computer programs, databases or source codes, including any application for registration of, and any improvements, enhancements or modifications of, the foregoing, and any right to apply for and be granted, renewals or extensions of, and rights to claim priority from, such rights and all similar or equivalent rights or forms of protection which subsist or will subsist now or in the future, including in respect of the foregoing.

Liability means any expense, cost, liability, loss, damage, claim, notice, entitlement, investigation, demand, proceeding or judgment (whether under statute, contract, equity, tort (including negligence), misrepresentation, restitution, indemnity or otherwise), howsoever arising, whether direct or indirect and/or whether present, unascertained, future or contingent and whether involving a third party or a party to these Terms or otherwise.

3.3 Support

3.3.1 TrackMe Enterprise & Unlimited Edition

As a registered user of TrackMe Enterprise or Unlimited edition, one or more points of contacts will be identified and personal accesses will be provided to our support portal:

- https://support.trackme-solutions.com

Once connected to the support portal, you will be able to:

- Open a new support request
- Review and manage the status of your opened support requests

If your company does not have any registered user yet, please send us an email and we will create an account for you:

- support@trackme-solutions.com

Once we have received and processed your request, an activation email will be sent to you with your personal access to the support portal.

To submit a support case, you can also contact us by email at the same address; a support request will be created automatically and associated with your user account.

SLA, Response Time and Support Coverage for Registered Customers

- We offer support for features and behaviors which relate directly to the usage of TrackMe for Splunk Cloud and Splunk Enterprise.
- Issues related to Splunk products, or any other third-party applications are not covered by our support agreement.
- Prioritized support is included for customers with a valid Full Registered Edition subscription.
- Support is available by email with a Service Level Agreement (SLA) of 24 hours for a support request to be taken in charge, from Monday to Friday, 8 am to 8 pm UK time.
- If the support case requires it, we will propose a virtual support meeting through our Zoom meeting services, or your own meeting tools if required.
- If you have specific needs or requirements that are not covered by our product, we will prioritize developments if you are a customer under active subscription.

3.3.2 TrackMe Free Community Edition

If you are a user of the Free limited edition of TrackMe (aka unregistered), we provide best effort support without any warranty.

3.3.3 Report a Bug or Submit a Request for Enhancement

You can report issues and submit enhancement requests in our public GitHub repository:

- https://github.com/trackme-limited/trackme-report-issues

This repository also references current and past issues linked to every change performed in TrackMe over time.

You can also contact us by email:

- support@trackme-solutions.com

3.3.4 Splunk Community Slack

Contact us on Splunk Community Slack, and even better, ask the community!

- https://splunk-usergroups.slack.com

3.3.5 Splunk Community (ex. Splunk Answers)

Open a question in Splunk Community:

- https://community.splunk.com

CHAPTER

FOUR

COMPATIBILITY AND DOWNLOAD:

4.1 Compatibility

4.1.1 Splunk core compatibility

Splunk core

- TrackMe is compatible with Splunk 8.1.x up to 9.0.x until the TrackMe release 2.0.99
- TrackMe is compatible with Splunk 9.1 and later with TrackMe release 2.0.x and 2.1.x
- TrackMe is developed, tested and qualified against latest generation of Splunk Enterprise and Splunk Cloud Victoria in priority.

4.1.2 Splunk Enterprise and Splunk Cloud compatibility

Splunk Enterprise compatibility

- TrackMe is compatible with Splunk Enterprise from Splunk 8.1.x

Operating system compatibility for Splunk Enterprise customers

We strongly recommend Linux as the Operating System for Splunk Enterprise customers, however TrackMe is supported on Splunk Enterprise supported Operating Systems:

- Linux (strongly recommend, and in fact used by the vast majority of customers)
- Windows (not recommended, we do not expect any good performances in Splunk in general)
- Mac OS (for development and testing purposes only)

We only test and qualify TrackMe releases and behaviours under Splunk Enterprise on Linux, and on Splunk Cloud Victoria experience.

Note

If you are using TrackMe hosted in Windows or Mac OS, we will provide best effort support only.

Splunk Cloud compatibility

- TrackMe is compatible with Splunk Cloud Victoria and Splunk Cloud Classic
- On Splunk Cloud Classic, TrackMe Version 2.0.60 and later is required to address some issues due to Splunk SLIM partitioning

Appinspect and vetting

We carefully comply with Splunk development best practices to ensure that we provide the highest level of application quality.

Therefore, TrackMe releases are reviewed and validated with Splunk Appinspect:

- https://dev.splunk.com/enterprise/reference/appinspect

When a new release of TrackMe is published, our automation also processes the release through Appinspect automation and the Appinspect HTML report is available in the release directory of our download Website:

- https://downloads.trackme-solutions.com/

TrackMe is compatible with Splunk Enterprise and Splunk Cloud products.

4.1.3 Python compatibility

Python 3 compatibility

- TrackMe supports **Python 3** only.

4.1.4 FIPS Mode compatibility

TrackMe and FIPS mode

- **Since TrackMe version 2.0.99, TrackMe is officially compatible with Splunk FIPS mode, and continuously validated on FIPS.**
- About FIPS: https://docs.splunk.com/Documentation/Splunk/latest/Security/SecuringSplunkEnterprisewithFIPs
- In Splunk 9.3.0, several changes were made and some aspects of FIPS are correctly applied, this requires TrackMe 2.0.99 and evolutions we have made to rely on sha256 instead of md5 crypto in various search and Python logics.

4.1.5 Web Browser compatibility

TrackMe is supported on Splunk supported Web Browsers:

See: https://docs.splunk.com/Documentation/Splunk/latest/Installation/Systemrequirements

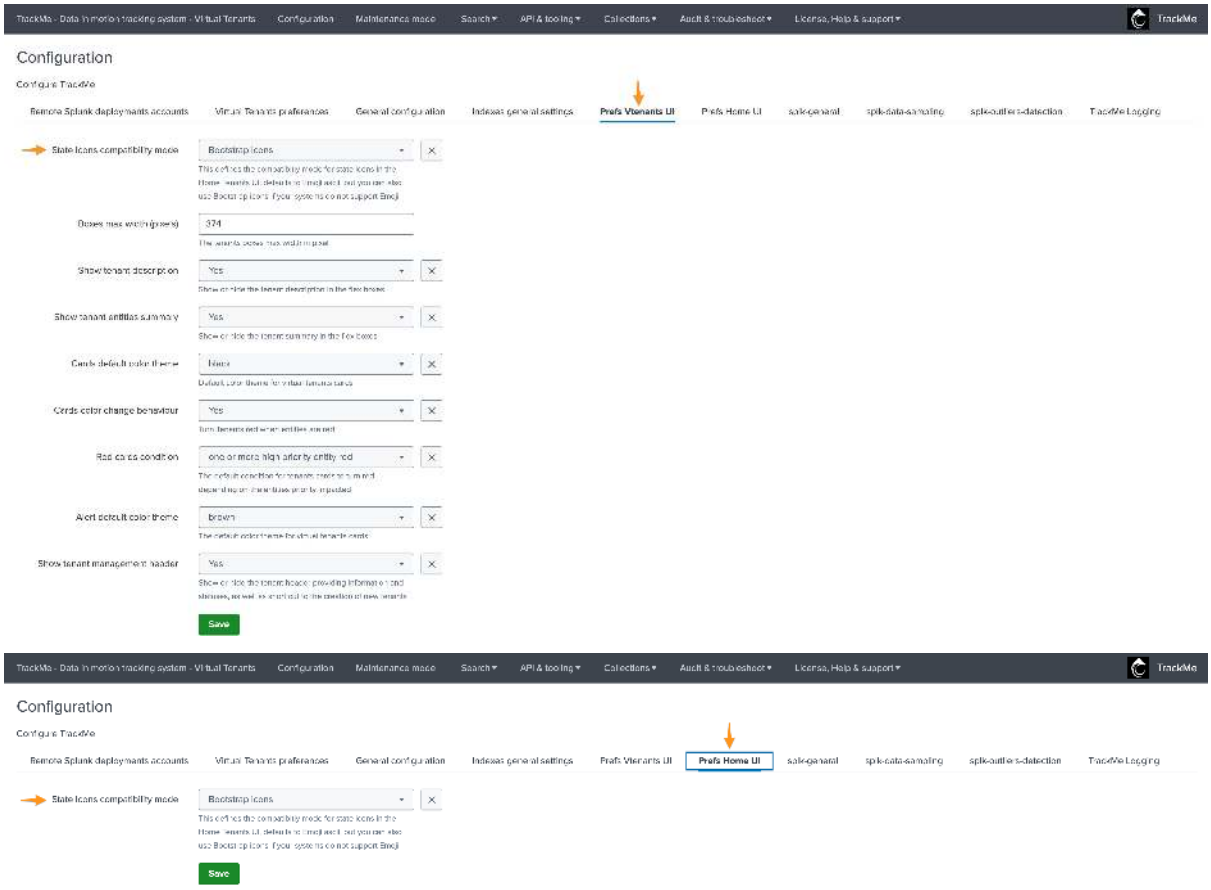
Supported Web Browsers:

- Google Chrome
- Mozilla Firefox
- Apple Safari

Microsoft Edge is not an officially supported Web browser by Splunk, however TrackMe is also compatible and no major issues were reported for TrackMe.

Charset support:

- Some Windows versions of Web Browsers do not support Emoji charset, and therefore some icons may not be displayed properly by default



4.2 Download

4.2.1 TrackMe for Splunk Enterprise & Splunk Cloud

TrackMe for Splunk Enterprise & Splunk Cloud can be downloaded from Splunk Base, as well as from our release Website:

- https://splunkbase.splunk.com/app/4621
- https://downloads.trackme-solutions.com

4.2.2 TrackMe App on SOAR

We provide a TrackMe application for Splunk SOAR (Phantom), this application for SOAR allows powerful actions to automate and interact with TrackMe straight from Splunk SOAR:

- https://downloads.trackme-solutions.com/trackme-app-on-soar

You can also access the source code and the releases from our GitHub repository:

- https://github.com/trackme-limited/trackme-app-on-soar

4.2.3 TrackMe Configuration Manager (TCM)

The TrackMe Configuration Manager (TCM) is a Splunk Application which can be used for CI/CD and configuration management purposes of TrackMe, it acts as a utility which can receive and replay configuration transactions, to promote changes between environments and also to restore configuration steps by replaying these, it can be downloaded from our release Website.

For more information about the TCM, see: *TrackMe CI/CD management (TCM)*

- https://downloads.trackme-solutions.com/TA-trackme-conf-manager

You can also access the source code and the releases from our GitHub repository:

- https://github.com/trackme-limited/TA-trackme-conf-manager

4.2.4 TA-trackme-cribl

This Splunk application provides API integration between Splunk and Cribl API, it acts as an SPL interface to the Cribl API so you can even more interactions between the platforms, leverage real time requests from Splunk dashboard and also leverage it in TrackMe such as custom Flex Object trackers, it can be downloaded from:

- https://downloads.trackme-solutions.com/TA-trackme-cribl

You can also access the source code and the releases from our GitHub repository:

- https://github.com/trackme-limited/TA-trackme-cribl

CHAPTER

FIVE

REQUIREMENTS:

5.1 Requirements for TrackMe

5.1.1 Target for deployments

For Splunk Enterprise customers

For Splunk on-premise customers, we recommend to deploy TrackMe on:

- A dedicated Search Head or Search Head Cluster, especially if you intend to monitor a large number of environments or if you are a large scale Splunk customer
- Alternatively, you can deploy TrackMe on the instance used for the purposes of the Splunk Monitoring Console

Splunk premium applications

- We do not support colocating TrackMe with any of the Splunk premium applications (Splunk Enterprise Security, Splunk ITSI)
- The reason is that both TrackMe and premium applications can have heavy requirements in term of workload, and TrackMe could therefore slightly impact these products

For Splunk Cloud customers

From Splunk Cloud Victoria, applications are deployed through self-services equally on all Search Heads, including Premium applications Search Head:

- When deploying TrackMe on Splunk Cloud, the application will be deployed automatically on all Search Heads, including your Premium application Search Heads, if any
- However, a virgin deployment of TrackMe has no workload as long as there are no Virtual Tenants created
- TrackMe should be configured and used from the first Search Head tier of your Splunk Cloud stack, commonly called the Ad-hoc Search Head tier

Splunk premium applications

- Similarly to on-premise deployments and while TrackMe is deployed on the premium applications Search Heads, we do not support running TrackMe on these Search Heads
- The reason is that both TrackMe and premium applications can have heavy requirements in term of workload, and TrackMe could therefore slightly impact these products

5.1.2 System requirements for the use of TrackMe

Supported Operating Systems

TrackMe is supported on any flavour of:

- Linux
- MacOS

Currently, running TrackMe on Windows Operating Systems is not supported, although this might be the case in the future.

Containerized computing platforms

Similarly to Splunk Enterprise, running TrackMe on containerized environment is supported:

- https://docs.splunk.com/Documentation/Splunk/latest/Installation/Systemrequirements

Resources requirement

The requirements for TrackMe in terms of CPU slightly relies on its usage and configuration, the main factors are:

- If the Splunk instance hosting TrackMe is itself more than a Search Head (if the instance is as well an indexer for instance in the case of all in one Splunk instance, which is not recommended)
- The number of enabled TrackMe Virtual Tenants
- The number of enabled Hybrid, Flex and Elastic Trackers
- The number of entities monitored, which influence factors like the number of Machine Learning models to be created and maintained for instance
- The repartitions between local tracking and Splunk Remote Searches tracking, which influence how much the load is shared between the local instance and the remote Splunk deployments, if any
- The number of active users, the more users are actively using TrackMe, the less search slots are available for the back-end purposes

Splunk typical requirements:

- TrackMe relies on Splunk and shares its typical requirements
- We recommend to honour the minimal specifications for Splunk: https://docs.splunk.com/Documentation/Splunk/latest/Installation/Systemrequirements

Minimal CPU requirements:

- We recommend a minimal setup of **16 vCPUs** for running TrackMe in a Production context.
- This highly depends on the number of trackers and use cases that are implemented, the nature of the use cases, as well as the number and activity of users in TrackMe.

Recommended requirements:

- **32 vCPUs** (or more) for running TrackMe in a Production context with a large scale deployment, and a large number of use cases.
- **32 GB** of memory is recommended for running TrackMe in a Production context.

Applications dependencies

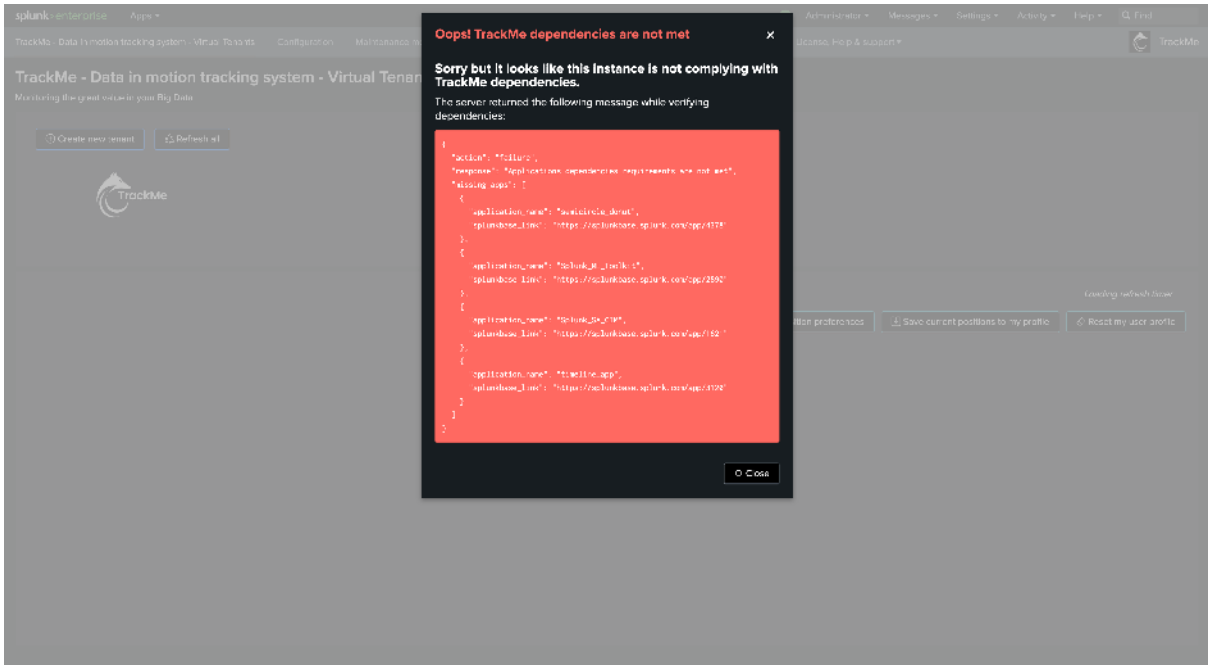
TrackMe requires the following applications to be deployed and available on the instance hosting TrackMe:

- Splunk_SA_CIM: https://splunkbase.splunk.com/app/1621
- Splunk_ML_Toolkit: https://splunkbase.splunk.com/app/2890

- semicircle_donut: https://splunkbase.splunk.com/app/4378

note: The Splunk Machine Learning toolkit itself requires the Python Scientific package, TrackMe relies on the ML Toolkit for the management of Machine Learning models.

Should any, or all of the dependencies not be met, TrackMe will refuse to load the Virtual Tenant User Interface and show the list of missing applications, such as:



CHAPTER

SIX

INSTALLATION:

6.1 Installation of TrackMe

6.1.1 Installation Target

TrackMe is a Search Head component only; therefore, it must be deployed only on a target Search Head layer:

Splunk roles	Required
Search head	yes (*)
Monitoring console	yes (*)
Indexer tiers	no
Heavy Forwarders	no

- You can deploy TrackMe on a dedicated Search Head or Search Head Cluster, or alternatively on the machine hosting the Splunk Monitoring Console.
- If you choose to install TrackMe on your Search Head layer in a Search Head Cluster (SHC), TrackMe must be deployed through the SHC deployer node.

6.1.2 Indexes Definition

Default Indexes

TrackMe requires at least 4 indexes to be defined. In a distributed context, this usually means defining indexes in your manager node:

Default index name	Purpose
trackme_summary	TrackMe entities activity such as the state, flipping events, and Smart Status
trackme_audit	TrackMe modifications and application audit activity
trackme_notable	Notable events are generated by the TrackMe notable alert action
trackme_metrics	Various metrics are generated depending on the components

Indexes are defined by default in the application package:

```
trackme/default/indexes.conf
```

Typically:

- If TrackMe is running on an "all-in-one" instance, you do not need to define the indexes as the application defines these already (mostly for testing and development purposes).
- It is good practice to use Splunk volumes rather than the default SPLUNK_DB variable to define the location of the buckets (which is required when using SmartStore).

Therefore, a typical definition would be:

```
[trackme_notable]
coldPath = volume:primary/trackme_notable/colddb
homePath = volume:primary/trackme_notable/db
thawedPath = $SPLUNK_DB/trackme_notable/thaweddb

[trackme_summary]
coldPath = volume:primary/trackme_summary/colddb
homePath = volume:primary/trackme_summary/db
thawedPath = $SPLUNK_DB/trackme_summary/thaweddb

[trackme_audit]
coldPath = volume:primary/trackme_audit/colddb
homePath = volume:primary/trackme_audit/db
thawedPath = $SPLUNK_DB/trackme_audit/thaweddb

[trackme_metrics]
coldPath = volume:primary/trackme_metrics/colddb
homePath = volume:primary/trackme_metrics/db
thawedPath = $SPLUNK_DB/trackme_metrics/thaweddb
datatype = metric
```

Adapt with the volume definitions in your context.

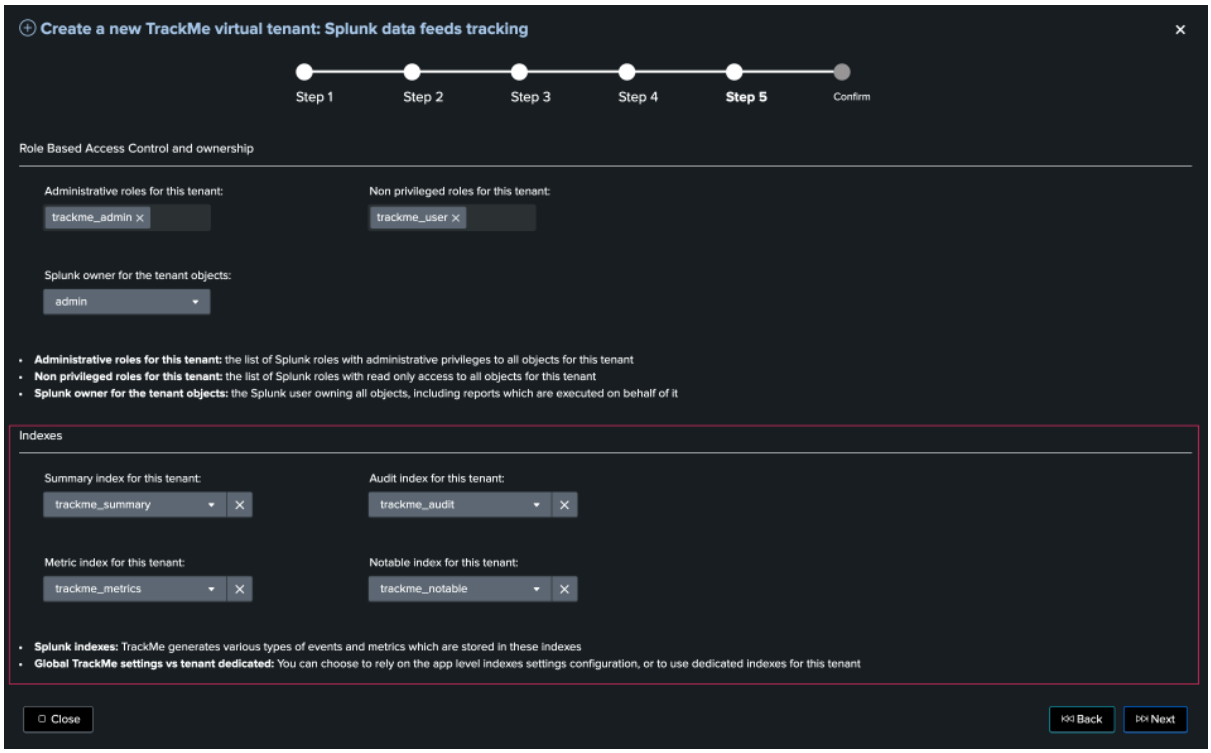
Virtual Tenants Specific Indexes

A core concept of TrackMe is called Virtual Tenants, which provides many powerful features. Part of this concept provides the capability to define a per-tenant specific set of indexes.

Therefore, you can choose to define specific indexes for one or more tenants, and a different set of indexes for other tenants.

This allows TrackMe to comply with any Role-Based Access Control (RBAC) requirements, for instance, to allow a population of users to access specific tenants while another population can access others.

Example: Virtual Tenant indexes definition configuration screen:



6.1.3 Installing TrackMe

Splunk Enterprise

Installing TrackMe on Splunk Enterprise on-premises deployments depends on the type of deployment.

- For standalone instances, refer to: https://docs.splunk.com/Documentation/AddOns/released/Overview/Singleserverinstall
- For distributed deployments, refer to: https://docs.splunk.com/Documentation/AddOns/released/Overview/Distributedinstall

Notes for Older Generations of Splunk (prior to Splunk 9.x)

TrackMe is developed for the latest generation of Splunk software; therefore, some built-in parameters are primarily targeting Splunk 9.x compatible configurations.

If you are running TrackMe on a version prior to Splunk 9.x, you should apply the following customizations:

Create a local/distsearch.conf:

```
# distsearch.conf

# Avoid the replication to the indexers of the KVstore backup tarball compressed␣
↪files
[replicationBlacklist]
trackme_backup_tgz = apps/trackme/backup/*.tgz
trackme_backup_dirs = apps/trackme/backup/...

# These lookups do not need to be replicated
trackme_cim_regex = apps/trackme/lookups/trackme_cim_regex.csv

# Machine Learning models: Anomaly detection will generate various ML models files,␣
↪these are not needed on the indexer layers
trackme_mlmodels = apps/trackme/lookups/__mlspl_*.mlmodel
```

Notes:

- In Splunk 9.x, biased language has been addressed; this stanza became replicationDenylist.
- In TrackMe, this stanza prevents ML models files from being unnecessarily replicated to the indexers and being part of the knowledge bundle.

Splunk Cloud

Installing TrackMe on Splunk Cloud relies on Cloud self-services. Refer to:

- https://docs.splunk.com/Documentation/SplunkCloud/latest/Admin/SelfServiceAppInstall

6.1.4 Upgrading TrackMe

In summary, upgrading TrackMe follows the same process as installation:

- In Splunk Enterprise, you will download the updated release, extract the new version, restart the instance if in a standalone Search Head, or apply the SHC bundle if running in an SHC.
- In Splunk Cloud, when a new release has been published and vetted, the new version release number appears as upgradable through the application management interface; you will therefore follow the self-services process.

TrackMe implements an automated concept to perform required application-level upgrade procedures, called `schema version`. See *Upgrading TrackMe*.

6.2 Upgrading TrackMe

6.2.1 1. Introduction

Upgrading TrackMe is a straightforward process which is similar to the installation.

However, TrackMe also implements a sophisticated and automated upgrade process called `schema version` which is used to dynamically run specific actions depending on the origin version and the state of TrackMe Virtual tenants.

The actions can involve the deletion, creation, or update of TrackMe knowledge objects depending on the context.

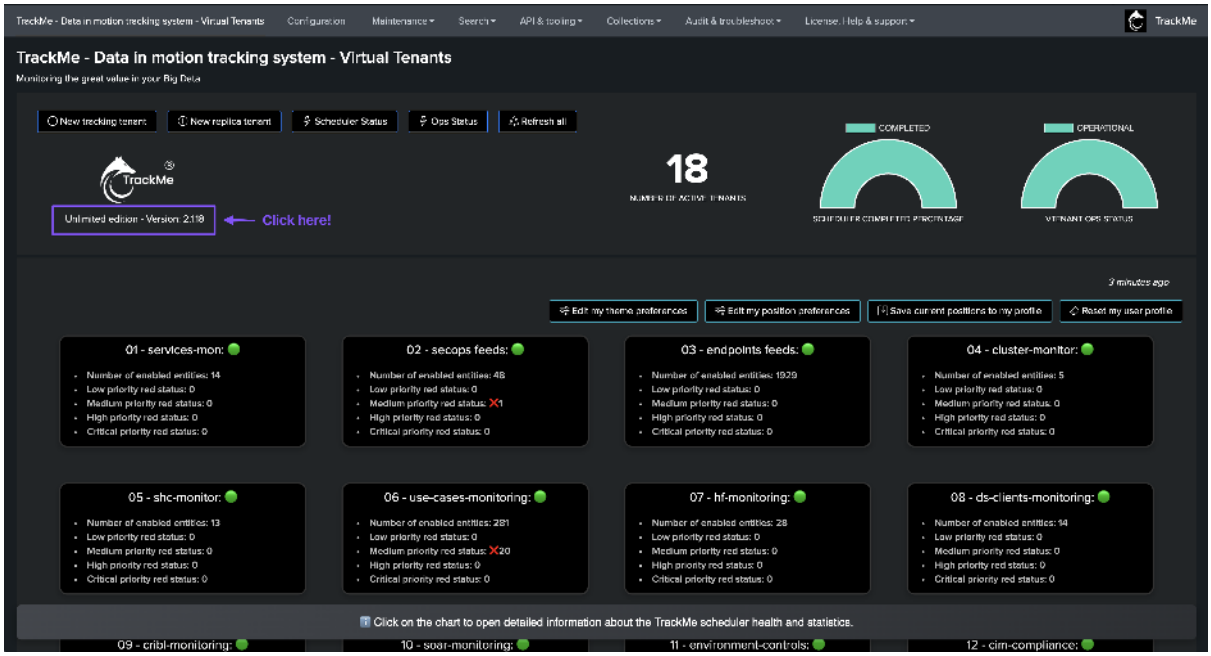
The purpose of this documentation is to thoroughly describe how the `schema version` upgrade works.

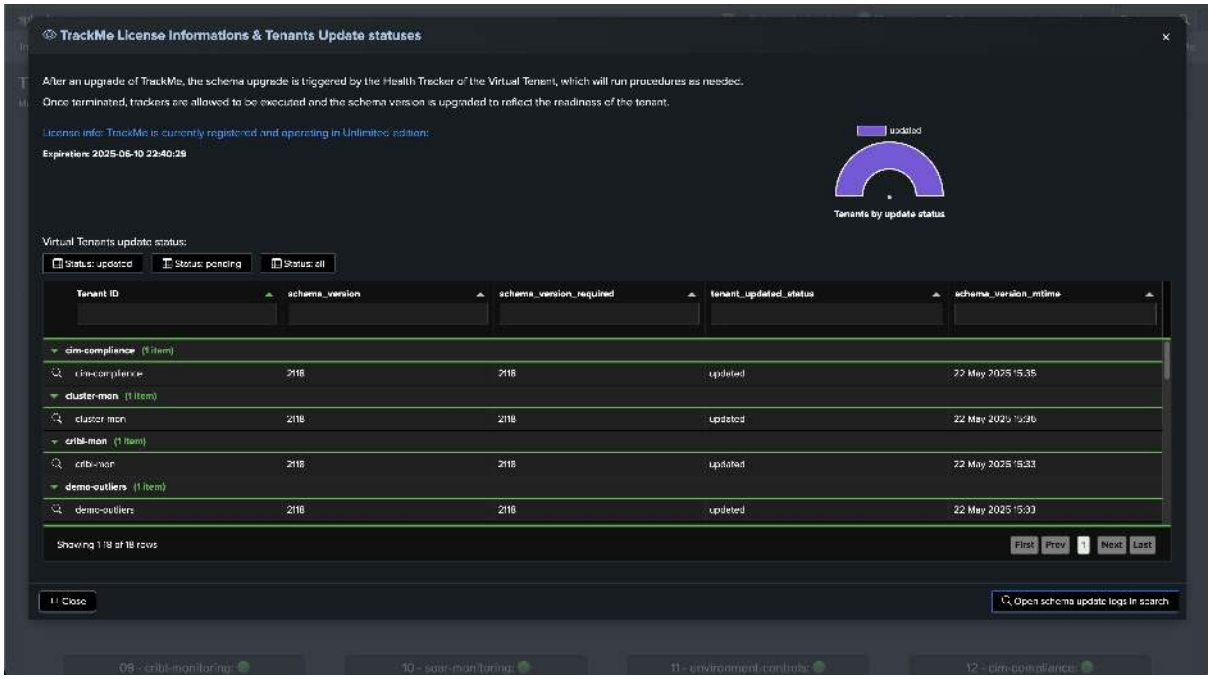
6.2.2 2. Schema version

Once you have upgraded TrackMe using the Splunk standard upgrade process depending on your context (Splunk Enterprise versus Cloud, SHC versus standalone), TrackMe will automatically verify and perform additional upgrade procedures as needed.

The `schema version` is versioning information which is stored in the KVstore record of the Virtual Tenant. You can verify the current schema version value in the central KVstore:

```
| inputlookup trackme_virtual_tenants | eval keyid=_key
| fields keyid, tenant_id, schema_version
```





The schema version corresponds to the TrackMe version minus the "dots". If you are running version 2.1.0, the schema version is 2100 (yes, we added a zero here, so we follow a four-digit convention).

Hint

TrackMe 2.1.0 automated backup improvements:

- Since TrackMe 2.1.0, the schema upgrade process was slightly improved and notably a backup process procedure was added to ensure that the upgrade process is more secure and reliable.
- In short, if TrackMe detects that the schema version needs to be upgraded, it will automatically execute a backup job before starting the upgrade process.

2.1 Schema version verification

The `schema version` verification is an automated process handled by the Health tracker of the Virtual Tenant.

When a Virtual Tenant is created, TrackMe also creates a dedicated Health tracker for this tenant. The Health tracker is responsible for various administration tasks such as detecting any issues encountered by the tenant's trackers, as well as running the `schema version` verification.

Health tracker name in Splunk:

- trackme_health_tracker_tenant_<tenant_id>

2.2 Logs

The logs of the Health tracker and any upgrade-related events are available via the navigation bar shortcut **"Audit & Troubleshoot / Logs - TrackMe custom commands / common to components / trackmetrackerhealth"** and match the following index/sourcetype:

Log structure changes since TrackMe 2.1.0:

- Since TrackMe 2.1.0, changes were made to improve the usability of the logs. Notably, a concept of task_instance_id and task_name was introduced to slightly improve tracking the activity of the tasks in the Health tracker.

- The field **context** has been replaced by a consistent new field **task**

```
index=_internal sourcetype=trackme:custom_commands:trackmetrackerhealth task="schema_
↪upgrade"
```

You can track the run time of every task handled by the Health tracker using the following search example:

```
index=_internal sourcetype=trackme:custom_commands:trackmetrackerhealth instance_id=*␣
↪task_instance_id=* task=* run_time=* tenant_id=*
| table _time tenant_id instance_id task task_instance_id run_time _raw
| sort 0 - _time
```

2.3 Schema version upgrade & traces

When the Health Tracker runs (every 5 minutes), it verifies the schema version:

```
2024-09-11 16:46:18,938 INFO trackmetrackerhealth.py generate 488 tenant_id="feeds-
↪tracking", instance_id=879a5021-3abc-47f0-a6f9-e21714240537, task="schema_upgrade",␣
↪task_instance_id=80a9366d-177e-465a-b065-573dfaf400b0, starting task.
2024-09-11 16:46:18,953 INFO trackme_libs_schema.py trackme_schema_get_version 127␣
↪task="schema_upgrade", task_instance_id=80a9366d-177e-465a-b065-573dfaf400b0,␣
↪tenant_id="feeds-tracking", trackme_schema_get_version, current schema_version="2097
↪"
```

If an upgrade is required, the following message is logged:

```
2024-09-11 16:46:18,953 INFO trackmetrackerhealth.py generate 592 tenant_id="feeds-
↪tracking", instance_id=879a5021-3abc-47f0-a6f9-e21714240537, task="schema_upgrade",␣
↪task_instance_id=80a9366d-177e-465a-b065-573dfaf400b0, detected migration required␣
↪for schema version 2098, schema_version="2097", schema_version_required="2100",␣
↪processing now.
```

Required procedures are then performed automatically. TrackMe starts by executing a backup job:

```
2024-09-11 16:46:31,698 INFO trackmetrackerhealth.py generate 613 tenant_id="feeds-
↪tracking", instance_id=879a5021-3abc-47f0-a6f9-e21714240537, task="schema_upgrade",␣
↪task_instance_id=80a9366d-177e-465a-b065-573dfaf400b0, backup post call executed␣
↪successfully
```

It will then execute the required upgrade procedures:

```
2024-09-11 16:46:31,755 INFO trackme_libs_schema.py trackme_schema_upgrade_2098 8596␣
↪task="schema_upgrade", task_instance_id=80a9366d-177e-465a-b065-573dfaf400b0,␣
↪Starting function trackme_schema_upgrade_2098, tenant_id="feeds-tracking"
2024-09-11 16:46:32,902 INFO trackme_libs_schema.py trackme_schema_upgrade_2098 8724␣
↪task="schema_upgrade", task_instance_id=80a9366d-177e-465a-b065-573dfaf400b0,␣
↪schema migration 2098, tenant_id="feeds-tracking", successfully deleted transform␣
↪definition, transform="trackme_dsm_tenant_feeds-tracking"
```

Once all required upgrade operations have been completed, the following message is logged:

```
2024-09-11 16:46:45,134 INFO trackme_libs_schema.py trackme_schema_update_version 225␣
↪task="schema_upgrade", task_instance_id=80a9366d-177e-465a-b065-573dfaf400b0,␣
↪tenant_id="feeds-tracking", schema version upgraded successfully, new schema_
↪version="2100"
```

(continues on next page)

(continued from previous page)

```
2024-09-11 16:46:45,135 INFO trackmetrackerhealth.py generate 664 tenant_id="feeds-
↪tracking", instance_id=879a5021-3abc-47f0-a6f9-e21714240537, task="schema_upgrade",␣
↪task_instance_id=80a9366d-177e-465a-b065-573dfaf400b0, run_time="26.2", task has␣
↪terminated.
```

Disabled Virtual Tenants

- Disabled Virtual Tenants are not migrated as long as they are disabled (since the Health Tracker will not run)
- If the Virtual Tenant is re-enabled, its upgrade will be processed automatically within the next 5 minutes

6.3 Migrate from TrackMe V1 to TrackMe V2

6.3.1 Introduction

The purpose of this page is to provide a path for existing users of TrackMe V1 to migrate to TrackMe V2, as easily as possible.

Hint

About configuration losses, interruption of service and efforts

- We appreciate many fellow TrackMe V1 users invested time and efforts in fine tuning the many various configuration items in TrackMe V1
- In many cases, TrackMe became a central tool, providing deep visibility and efficient monitoring capabilities
- Although the migration requires manual operations, there is a limited amount of efforts needed for TrackMe V2 to be operation and replacing your previous deployment
- However, during this phase, existing monitoring operations will stop, until your new TrackMe V2 deployment is operational
- Review this migration manual, performing planning and backups, and upgrade peacefully
- If you missed something, you can always retrieve these information from summary events, TrackMe backups or file backups

Top information

- You **cannot** run both TrackMe V1 and V2 on the same Splunk instance / SHC
- When the application is upgraded, it replaces V1 entirely, all configuration, settings are permanently lost (apart from TrackMe and third party backups)
- **Prior to the migration** you must follow this documentation if you wish to migrate existing settings and rules
- **If for some reasons** you didn't read this documentation **before** (sic), you still can recover information from TrackMe's summary events and backups
- The documentation cannot cover every subtitle of potential customizations of TrackMe, you may need to adapt some of these to your own context

TrackMe V2 is deeply different from TrackMe V1 in all means, TrackMe V2 notably introduces a brand

new key concept of Virtual Tenants which slightly influences the migration path.

Due to the highly flexible nature of TrackMe V2, there are various design questions that could be taken into account, the present document describes the general migration path and options.

Migrating or restarting from scratch?

Depending on your context, both approaches can make sense, in short:

- Starting from scratch has the advantage of starting fully clean, without taking any risks of leverage obsolete rules or settings which you could improve today
- You can directly leverage the many powerful advantages of TrackMe V2, including designing Virtual Tenants for instance, for a better experience and value
- On the opposite, you may have spent time and efforts in qualifying topics like entities priorities, it is fully understandable that you would want to migrate these settings

The migration path is in reality doing a little bit of both Worlds, you can decide at any step what makes the most sense in your context, and selectively import settings from your TrackMe V1 deployment.

6.3.2 Planning the migration

Not all users are equal in term when it comes to the level of access, Splunk Enterprise users have full backend accesses, while users in Splunk Cloud have much more limitations.

In short, the migration from an existing deployment of TrackMe V1 can be summarised as:

- Performing backups prior to the migration (KVstore backups and file-system level backups)
- Auditing local knowledge objects and customisations
- Allowing a maintenance window for the operation, if you are relying on TrackMe V1 in a Production context, no alerts will be available as long as the migration and configuration is not terminated
- Performing the application upgrade from V1 to V2
- Creating at least one TrackMe V2 Virtual Tenant to replace the migrated TrackMe V1 deployment
- Creating Hybrid Trackers, discover and perform the basis setup
- Import target key configuration items, such as entities priority or custom threshold policies
- Creating additional Knowledge Objects, such as TrackMe alerts or Elastic Sources, etc

6.3.3 Prior to the migration: Performing backups and exports

file-system level backup (For Splunk Enterprise)

Splunk Cloud users

- If you are a Splunk Cloud user, you can ignore this step
- Performing backup here is only for Splunk on-premise users, and a simple additional safety before proceding to an upgrade as a generic recommendation

It is recommended to perform a full backup of the TrackMe application root directory prior to the upgrade.

In a basic way, this means creating a compressed archive of the existing root directory:

```
cd /opt/splunk/etc/apps
tar -czf <target_path>/trackme_v1_backup_DDMMYYYY.tgz trackme
```

TrackMe KVstores backup

TrackMe backups

- While you wouldn't be able to restore directly the V1 backups into TrackMe V2, these provides means to retrieve and access any useful settings of TrackMe
- For instance, if you later on realise that you have created some specific content such as Elastic Sources, block lists, tags and so forth, these information can be retrieved from the JSON files contained in the backup archive

TrackMe V1 has a concept of a backup job which targets its own KVstore collections, this job is by default executed once per day and stores backup archives locally on the search head.

You can ensure that this process is functional, and run an immediate backup job prior to the migration.

In TrackMe V1, this scheduled report is named:

`TrackMe - Backup KVstore collections and purge older backup files`.

You can manually run this job to ensure a backup will be taken prior to the upgrade.

Auditing TrackMe V1 objects

TrackMe V1 has 3 main types of components:

- Data sources tracking
- Data hosts tracking
- Metric hosts tracking

As part of its normal activity, TrackMe V1 generates summary events which contain the essential entity information, such as the priority of their monitor status.

There are various additional TrackMe V1 features which you may have used, mostly:

- Elastic Sources, either shared or dedicated
- Allow and block lists, which are specific collections for each component
- Data sampling custom rules
- Tag policies
- TrackMe alerts

Use the following search to detect which KVstore collections have actual records, you can ignore empty collections:

Notes:

- run this search **prior** to the migration, if TrackMe was upgraded already, this is too late so you can ignore these steps
- ignore empty collections
- some of the KVstore collections have generated and/or dynamic content, this is not relevant for the purpose of the migration
- for each of the collection resulting from the following search, you can manually copy the collection in a CSV file for the ease of re-creation and/or re-applying some of steps

```
| rest splunk_server=local /services/server/introspection/kvstore/collectionstats
| mvexpand data
| spath input=data
```

(continues on next page)

(continued from previous page)

```
| rex field=ns "(?<app>.*)\.(?<collection>.*)"
| stats first(count) as count by app, collection
| where app="trackme" AND count>0
| where NOT collection in ("kv_trackme_alerts_ack", "kv_trackme_audit_changes", "kv_
↪trackme_backup_archives_info", "kv_trackme_data_sampling", "kv_trackme_maintenance_
↪mode", "kv_trackme_objects_summary", "kv_trackme_summary_investigator_volume_
↪outliers")
| rex field=collection "^kv_(?<transforms_definition>.*)"
| fields app collection transforms_definition count
```

TrackMe Collections CSV export

BEFORE TrackMe is upgraded, the easiest and simplest procedure is to manually export the principal collections into CSV lookups, which will be used to restore main configuration items.

These steps need to be taken prior to the migration, once upgraded the original TrackMe V1 collections will not exist anymore and therefore their content wilk has been deleted automatically.

Allow lists & block lists

- TrackMe V2 handles concepts of allow lists & block lists in a fundamentally different fashion, as of now, these cannot be migrated and need to be replayed if necessary

Splunk Cloud

- Splunk Cloud will normally preserve CSV lookup files created prior to the migration (unless for some technical reasons, the app is uninstalled at some points)
- As an additional measure of safety, you can as well download the CSV lookups out of Splunk when performing the following steps

Hint

Empty collections:

- If any of the following collection is empty, you can safety ignore its export and associated restoration

Proceed as follows:

Export Data Sources collection

```
| inputlookup trackme_data_source_monitoring | eval keyid=_key
| eval object=coalesce(data_name, data_host, metric_host)
| outputlookup trackme_v1_trackme_data_source_monitoring.csv
```

Export Data Hosts collection

```
| inputlookup trackme_host_monitoring | eval keyid=_key
| eval object=coalesce(data_name, data_host, metric_host)
| eval object="key:host|" . lower(object)
| outputlookup trackme_v1_trackme_host_monitoring.csv
```

Export Metric Hosts collection

```
| inputlookup trackme_metric_host_monitoring | eval keyid=_key
| eval object=coalesce(data_name, data_host, metric_host)
| eval object="key:host|" . lower(object)
| outputlookup trackme_v1_trackme_metric_host_monitoring.csv
```

Export Shared Elastic Sources collection

Elastic Shared

- In TrackMe V1, Elastic Shared sources were sharing the time range period with the scheduled period of the Elastic tracker
- In TrackMe V2, the period is properly retrieved and applied by the backend responsible from spawning the searches, the following sets to a default of -4h/+4h to allow the transition
- Review the Elastic records, adapt the earliest/latest if necessary, and make sure the index and sourcetype fields are set to useful values (which do not have to be real indexes/sourcetypes)
- Several fields have been renamed, make sure to properly execute the following export statement

```
| inputlookup trackme_elastic_sources
| rename data_name as object, elastic_data_index as elastic_index, elastic_data_
↪sourcetype as elastic_sourcetype
| eval earliest="-4h", latest="+4h"
| outputlookup trackme_v1_trackme_elastic_sources_shared.csv
```

Export Dedicated Elastic Sources collection

Elastic Dedicated

- Elastic Dedicated cannot be migrated from V1 to V2, the trackers have to be re-created from TrackMe V2
- However, you can export the V1 collection records to support the re-creation in the further steps

```
| inputlookup trackme_elastic_sources_dedicated | eval keyid=_key
| outputlookup trackme_v1_trackme_elastic_sources_dedicated.csv
```

Export Data Sampling custom rules

```
| inputlookup trackme_data_sampling_custom_models | eval keyid=_key
| outputlookup trackme_v1_trackme_data_sampling_custom_models.csv
```

Export Lagging policies for Data Sources & Hosts

```
| inputlookup trackme_custom_lagging_definition | eval keyid=_key
| outputlookup trackme_v1_trackme_custom_lagging_definition.csv
```

Export Lagging policies for Metric Hosts

```
| inputlookup trackme_metric_lagging_definition | eval keyid=_key
| outputlookup trackme_v1_trackme_metric_lagging_definition.csv
```

Export tag policies

```
| inputlookup trackme_tags_policies | eval keyid=_key
| outputlookup trackme_v1_trackme_tags_policies.csv
```

Export Logical groups

```
| inputlookup trackme_logical_group | eval keyid=_key
| eval object_group_members=lower(mvjoin(object_group_members, "|"))
| outputlookup trackme_v1_trackme_logical_group.csv
```

6.3.4 Upgrading to TrackMe V2

Upgrade the application

Once you are ready to migrate, perform the application upgrade as usually:

- If running Splunk Enterprise in a standalone instance, you can either upgrade through Splunk Web, or manually by extracting the compressed tarball archive, in both cases, make sure to restart Splunk post-upgrade
- If running Splunk Enterprise in a Search Head Cluster, extract TrackMe V2 to the SHC deployer, publish the bundle and wait for the Rolling Restart to complete
- If you are a Splunk Cloud Victoria user, you can upgrade to TrackMe V2 through Splunk Cloud self-services
- If you are a Splunk Cloud user on any other flavour, request the application upgrade to Cloud Operations

What will happen once TrackMe was upgraded?

- TrackMe V2 will be mostly fully virgin, the Virtual Tenants wizard will propose you to guide you through the creation of your first tenant
- Previously defined KVstore do not exist anymore, their content is now lost, this will be the case for almost all KVstore collections
- There are no more entities monitored yet
- There are no more active alerts

Create your first Virtual Tenant

Designing Virtual Tenants

Free Limited Edition

- The Free Limited Edition mode is restricted to two Virtual Tenants
- The Full Registered Edition allows to create an unlimited number of tenants

A new key concept in TrackMe V2 consists in the Virtual Tenants feature, depending on your context, Virtual Tenants allow you to:

- Dedicate entire virtual instances of TrackMe depending on your needs (think about departments, technologies, companies, teams, etc. . .)
- Scale by scoping Virtual Tenants to reasonable volumes of entities (customers rarely have 10,000 or 20,000 forwarders without any form of convention, context or regions!)
- And many more

For the purposes of the migration, this present documentation assumes the creation of a single Virtual Tenant which replaces your previous existing TrackMe V1 deployment.

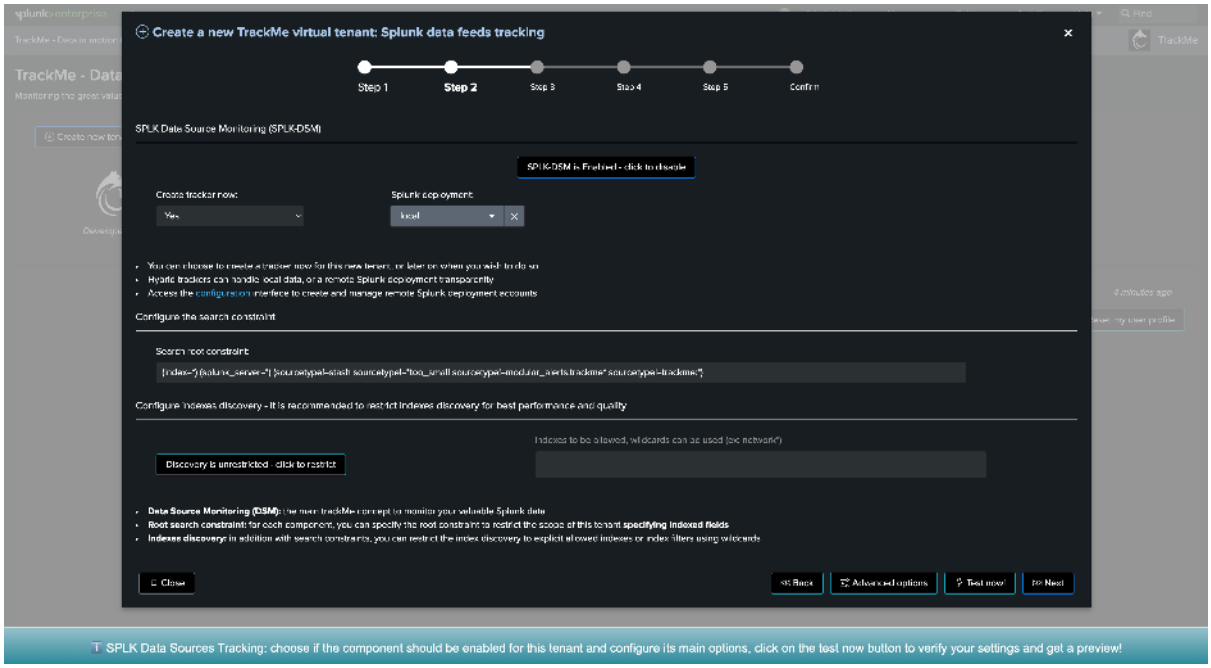
However, keep in mind that Virtual Tenants provide a key concept to address many various use cases with minimal levels of design.

For more details about the creation of Virtual Tenants for Splunk feeds tracking, see *Creating Virtual Tenants*

Use the wizard to create your new Virtual Tenant

Following the wizard and creation your first Virtual Tenant, you can define various options at the wizard phase, you can for instance:

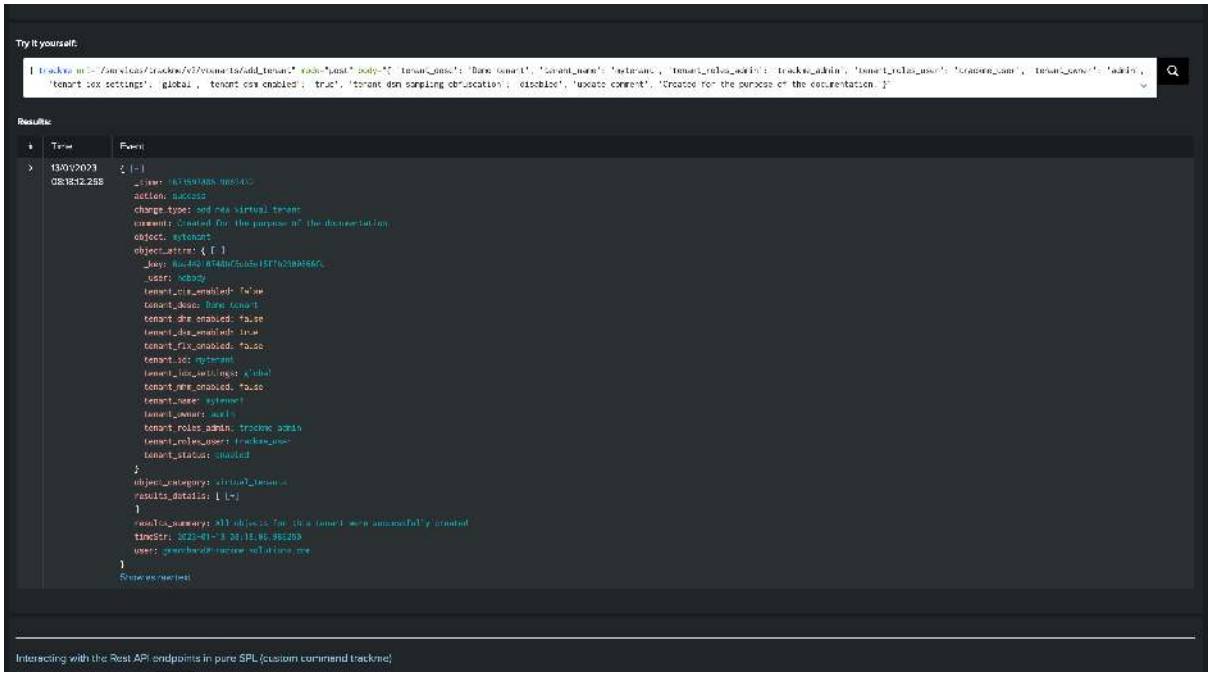
- Create a tenant with no trackers at all, then create your own Hybrid Trackers after through the wizard
- Create a tenant with trackers to be created at the stage of the Virtual tenant creation
- Choose settings such as target for indexers, ownership and RBAC



Create a new Virtual Tenant in command line, with no trackers

In TrackMe, go in a Search, and run the following command:

```
| trackme url="/services/trackme/v2/vtenants/add_tenant" mode="post" body="{\"tenant_
↪desc\":\"Splunk Feeds tracking\",\"tenant_name\":\"splk-feeds-tracking\",\"tenant_
↪roles_admin\":[\"trackme_admin\"],\"tenant_roles_user\":[\"trackme_user\"],\"tenant_
↪owner\":\"admin\",\"tenant_idx_settings\":\"global\",\"tenant_dsm_enabled\":true,\
↪"tenant_dsm_sampling_obfuscation\":\"disabled\",\"tenant_dhm_enabled\":true,\
↪"tenant_dhm_alerting_policy\":\"track_per_host\",\"tenant_mhm_enabled\":true}"
```



There are many more options available through the REST API, for more information see: *TrackMe REST API*

Create Hybrid Trackers

BEFORE restoring any parameters from TrackMe V1, ensure that you have create relevant Hybrid Trackers and started to discover and maintain entities in your brand new TrackMe V2 deployment.

For more information about how to create Hybrid Trackers, see: *splk-feeds - Creating and Managing Hybrid Trackers*

Restore priorities, monitored state and other main settings

In most of the use cases, there are two first pieces of information that you will want to restore regarding previously known entities.

Component splk-dsm (previously called Data Source Monitoring)

For the purpose of the documentation, and if you have not exported the collection prior to the upgrade, you can still use the summary data to generate the latest known information:

Ignore this if you performed exports prior to the migration

```
index=trackme_summary sourcetype=stash source="current_state_tracking:data_source"
| stats latest(priority) as priority, latest(data_monitored_state) as monitored_state␣
↪by key, object
| outputlookup trackme_v1_trackme_data_source_monitoring.csv
```

Apply:

notes:

- replace `splk-feeds-tracking` with the identifier of the tenant previously created

```
| inputlookup trackme_dsm_tenant_splk-feeds-tracking | eval keyid=_key
| lookup trackme_v1_trackme_data_source_monitoring.csv object OUTPUT data_max_lag_
↪allowed as previous_data_max_lag_allowed, priority as previous_priority, data_
↪monitored_state as previous_monitored_state, data_monitoring_wdays as previous_data_
```

(continues on next page)

(continued from previous page)

```
↪monitoring_wdays, data_override_lagging_class as previous_data_override_lagging_
↪class, min_dcount_host as previous_min_dcount_host
| eval data_max_delay_allowed=coalesce(previous_data_max_lag_allowed, data_max_delay_
↪allowed)
| eval data_max_lag_allowed=coalesce(previous_data_max_lag_allowed, data_max_lag_
↪allowed)
| eval priority=coalesce(previous_priority, priority), monitored_
↪state=coalesce(previous_monitored_state, monitored_state)
| eval data_monitoring_wdays=coalesce(previous_data_monitoring_wdays, data_monitoring_
↪wdays), data_override_lagging_class=coalesce(previous_data_override_lagging_class,␣
↪data_override_lagging_class), min_dcount_host=coalesce(previous_min_dcount_host,␣
↪min_dcount_host)
| outputlookup trackme_dsm_tenant_splk-feeds-tracking append=t key_field=keyid
```

Component splk-dhm (previously called Data Host Monitoring)

For the purpose of the documentation, and if you have not exported the collection prior to the upgrade, you can still use the summary data to generate the latest known information:

Ignore this if you performed exports prior to the migration

```
index=trackme_summary sourcetype=stash source="current_state_tracking:data_host"
| stats latest(priority) as priority, latest(data_monitored_state) as monitored_state␣
↪by key, object
| eval object=lower(object)
| eval object="key:host|" . lower(object)
| outputlookup trackme_v1_trackme_host_monitoring.csv
```

Apply:

notes:

- replace `splk-feeds-tracking` with the identifier of the tenant previously created

```
| inputlookup trackme_dhm_tenant_splk-feeds-tracking | eval keyid=_key
| lookup trackme_v1_trackme_host_monitoring.csv object OUTPUT data_max_lag_allowed as␣
↪previous_data_max_lag_allowed, priority as previous_priority, data_monitored_state␣
↪as previous_monitored_state, data_monitoring_wdays as previous_data_monitoring_
↪wdays, data_override_lagging_class as previous_data_override_lagging_class, data_
↪host_alerting_policy as previous_splk_dhm_alerting_policy
| eval data_max_delay_allowed=coalesce(previous_data_max_lag_allowed, data_max_delay_
↪allowed)
| eval data_max_lag_allowed=coalesce(previous_data_max_lag_allowed, data_max_lag_
↪allowed)
| eval priority=coalesce(previous_priority, priority), monitored_
↪state=coalesce(previous_monitored_state, monitored_state)
| eval data_monitoring_wdays=coalesce(previous_data_monitoring_wdays, data_monitoring_
↪wdays), data_override_lagging_class=coalesce(previous_data_override_lagging_class,␣
↪data_override_lagging_class)
| eval splk_dhm_alerting_policy=coalesce(previous_splk_dhm_alerting_policy, splk_dhm_
↪alerting_policy)
| outputlookup trackme_dhm_tenant_splk-feeds-tracking append=t key_field=keyid
```

Push expected hosts

- You can push expected hosts from a third party lookup (such as a CMDB or the export of TrackMe V1)

- This way, if you have hosts which are not active yet, TrackMe will automatically take these into account, and update their status as soon as they are active
- See: *Pushing Expected Sources to TrackMe (Tracking Expected Sources or Hosts in splk-dsm/splk-dhm)*

Component splk-mhm (previously called Metric Host Monitoring)

For the purpose of the documentation, and if you have not exported the collection prior to the upgrade, you can still use the summary data to generate the latest known information:

Ignore this if you performed exports prior to the migration

```
index=trackme_summary sourcetype=stash source="current_state_tracking:metric_host"
| stats latest(priority) as priority, latest(metric_monitored_state) as monitored_
↪state by key, object
| eval object="key:host|" . lower(object)
| outputlookup trackme_v1_trackme_metric_host_monitoring.csv
```

Apply:

notes:

- replace `splk-feeds-tracking` with the identifier of the tenant previously created

```
| inputlookup trackme_mhm_tenant_splk-feeds-tracking | eval keyid=_key
| lookup trackme_v1_trackme_metric_host_monitoring.csv object OUTPUT priority as␣
↪previous_priority, metric_monitored_state as previous_monitored_state
| eval priority=coalesce(previous_priority, priority), monitored_
↪state=coalesce(previous_monitored_state, monitored_state)
| outputlookup trackme_mhm_tenant_splk-feeds-tracking append=t key_field=keyid
```

Restore lagging policies for splk-dsm/splk-dhm

If you had created lagging policies in TrackMe V1 for splk-dsm/splk-dhm, these records were stored in the following collection:

- collection: kv_trackme_custom_lagging_definition, transforms: trackme_custom_lagging_definition

You can restore the policies from an export made previously:

notes:

- replace `splk-feeds-tracking` with the identifier of the tenant previously created
- TrackMe V2 has an additional concept in the lagging policies where delay and lag are stored independently
- For the purpose of the migration, we will apply the V1 value in both cases
- You can later modify these values if needed in TrackMe V2

Apply:

```
| inputlookup trackme_v1_trackme_custom_lagging_definition.csv
| eval value_delay=value, value_lag=value
| outputlookup trackme_common_lagging_classes_tenant_splk-feeds-tracking
```

Restore lagging policies for splk-mhm

If you had created lagging policies in TrackMe V1 for splk-mhm, these records were stored in the following collection:

- collection: kv_trackme_custom_lagging_definition, transforms: trackme_custom_lagging_definition

You can restore the policies from an export made previously:

notes:

- replace `splk-feeds-tracking` with the identifier of the tenant previously created

Apply:

```
| inputlookup trackme_v1_trackme_metric_lagging_definition.csv
| outputlookup trackme_mhm_lagging_classes_tenant_splk-feeds-tracking
```

Restore logical groups association

If you had created logical groups association in TrackMe V1 for splk-dhm/splk-mhm, these records were stored in the following collection:

- collection: kv_trackme_logical_group, transforms: trackme_logical_group

You can restore the policies from an export made previously:

notes:

- replace `splk-feeds-tracking` with the identifier of the tenant previously created
- Make sure you carefully exported and respect the SPL syntax, the object_group_members field is expected as a pipe separated field form the export, and transformed back to an mvfield

Apply:

```
| inputlookup trackme_v1_trackme_logical_group.csv
| makemv delim="|" object_group_members
| outputlookup trackme_common_logical_group_tenant_splk-feeds-tracking
```

Restore tag policies

If you had created tag policies in TrackMe V1, these records were stored in the following collection:

- collection: kv_trackme_tags_policies, transforms: trackme_tags_policies

You can restore the policies from an export made previously:

notes:

- replace `splk-feeds-tracking` with the identifier of the tenant previously created

```
| inputlookup trackme_v1_trackme_tags_policies.csv
| outputlookup trackme_common_tag_policies_tenant_splk-feeds-tracking
```

Restore Shared Elastic Sources

You can restore Elastic Sources quite easily, if you had created any, their records were stored in the following collection:

- collection: kv_trackme_elastic_sources, transforms: trackme_elastic_sources

notes:

- replace `splk-feeds-tracking` with the identifier of the tenant previously created

```
| inputlookup trackme_v1_trackme_elastic_sources_shared.csv
| outputlookup trackme_dsm_elastic_shared_tenant_splk-feeds-tracking
```

Re-create Dedicated Shared Elastic Sources

TrackMe V1 dedicated Elastic Trackers cannot be re-created by simply exporting / importing the KVstore collections.

Actual knowledge objects need to be properly created, and their syntax and behaviours fundamentally changed in TrackMe V2.

As well, in various use cases, Elastic Dedicated trackers can advantageously be replaced by Hybrid Trackers which are much more scalable, and allow to manage from a few to many entities resulting from the search logic.

For more information about how to create Hybrid Trackers, see: *splk-feeds - Creating and Managing Hybrid Trackers*

Dedicated Trackers

- Rely on the previously exported KVstore collection to review the context of Trackers that had been created in TrackMe V1
- Use the Elastic tracker wizard as needed to re-create brand new entities from TrackMe V2

Restore Data Sampling custom rules

If you had created tag policies in TrackMe V1, these records were stored in the following collection:

- collection: kv_trackme_data_sampling_custom_models, transforms: trackme_data_sampling_custom_models

notes:

- replace `splk-feeds-tracking` with the identifier of the tenant previously created

```
| inputlookup trackme_v1_trackme_data_sampling_custom_models.csv
| outputlookup trackme_dsm_data_sampling_custom_models_tenant_splk-feeds-tracking
```

Re-create TrackMe alerts

You **cannot** migrate as such TrackMe V1 alerts, nor modify the existing alerts to match TrackMe V2 requirements.

You must disable these alerts in the first phase, then create new alerts up to your needs in TrackMe V2, and finally later on, delete the TrackMe V1 alerts once you are up and running.

To identity existing active alerts:

```
| rest splunk_server=local /servicesNS/nobody/trackme/saved/searches | search eai:acl.
↪app="trackme" alert.track=1
| fields title, cron_schedule, schedule_window, alert.suppress.fields, alert.suppress.
↪period, disabled, next_scheduled_time, id, actions
```

For more details about the creation of TrackMe alerts, see *Alerting Architecture & Third-Party Integration*

End of migration

You are now done and should have a fully operational fancy TrackMe V2 deployment, well done!

Review TrackMe V2 carefully, for more information about the massively enhanced troubleshooting and administration experience in TrackNe V2:

- See *Troubleshooting TrackMe*
- See *Manage Virtual Tenants*
- See *Scheduling Virtual Tenants*

You can now get some rest and enjoy well deserved pizza, with or without pepperoni depending on your preferences.

6.4 Migrating and Cloning TrackMe

Since TrackMe version 2.1.5, you can easily migrate TrackMe from a deployment to another, or cloning a deployment, using its powerful built-in backup and restore capabilities:

- Consult the white paper: *Backing up and Restoring TrackMe*.

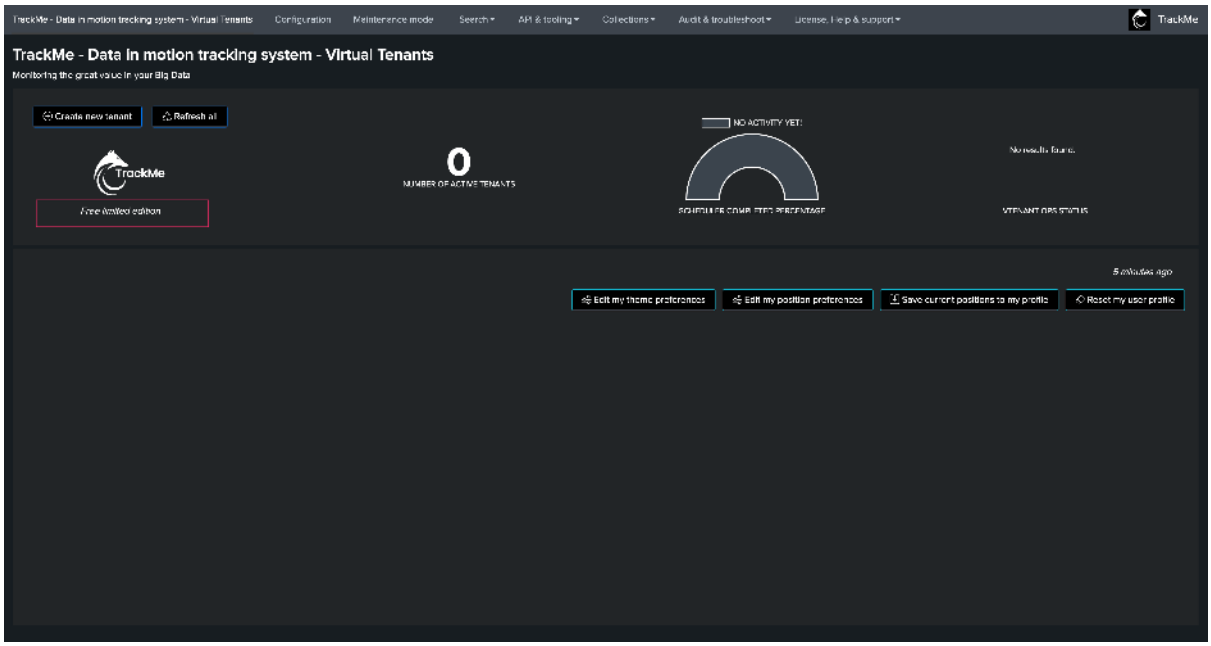
6.5 License registration

6.5.1 TrackMe licensing modes

TrackMe has different licensing modes available:

- Community Limited mode edition (default)
- Unlimited edition / Enterprise Edition / Trial registered edition
- Developer mode

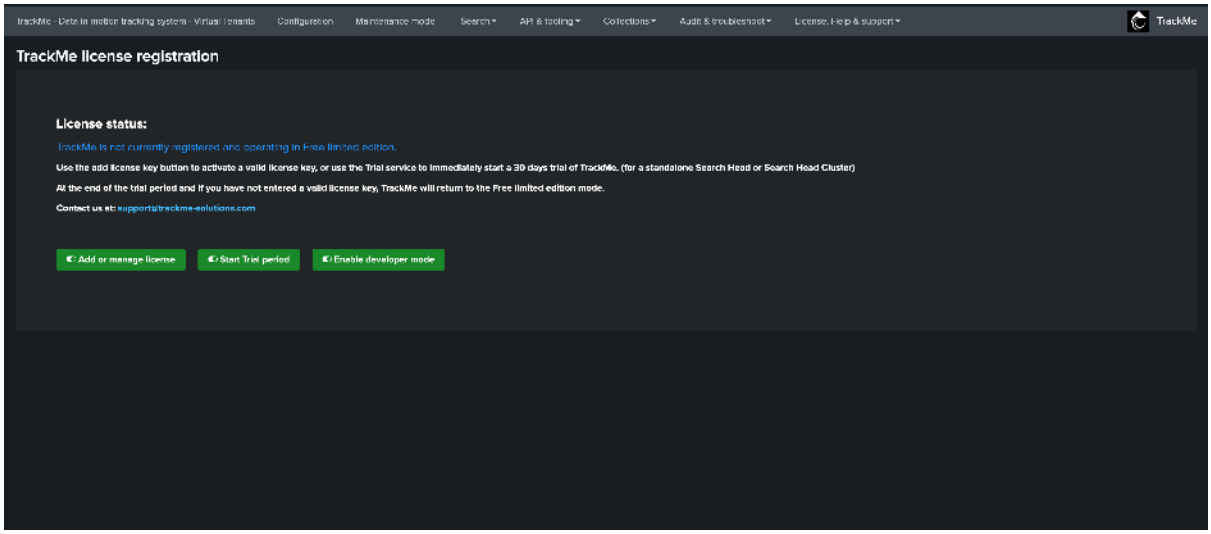
Once TrackMe has been installed, the **Community Limited mode edition** is enabled. You can observe the current registration mode from the TrackMe welcome user interface:



6.5.2 Registering and managing TrackMe licensing

To register and manage the license, go to:

- Navigation bar / License, Help & support / Register the license or start a trial



TrackMe Licensing API services

TrackMe for the management of licensing relies on the services of Cryptolens:

- https://help.cryptolens.io

Under the following circumstances, an external call to the Cryptolens API service is made:

- Registering for a Trial license within TrackMe
- Registering the license key for Enterprise and Unlimited customers (online activation)

An external call to the Cryptolens API service is made to verify the validity of the license or generate a new Trial license key if requesting a Trial:

- https://help.cryptolens.io/security/api-ip

The traffic is outgoing:

- https://app.cryptolens.io:443

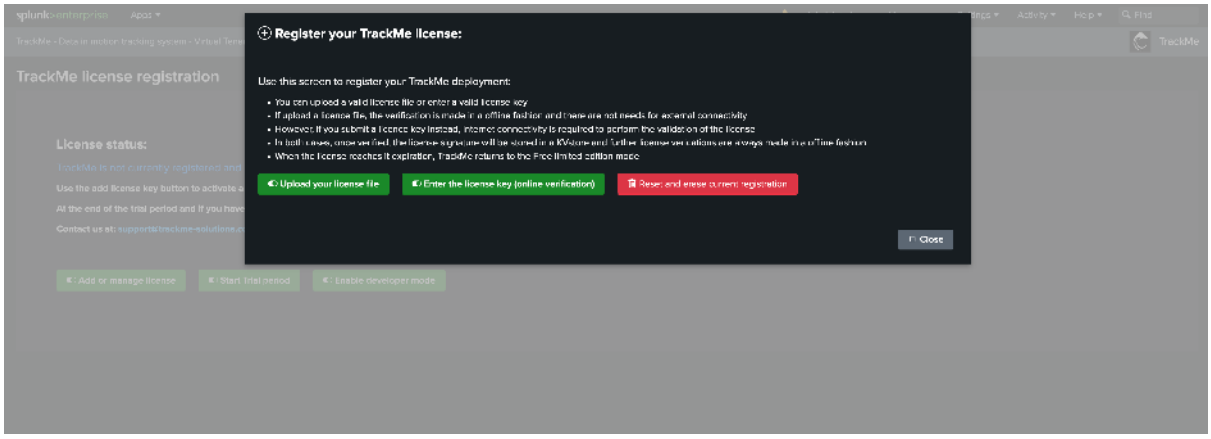
If you do not want this external connection, or if your Search Head cannot access this service, you can use the offline registration process using the license file which we will have provided.

Register your TrackMe license

If you are a licensed customer, you will have received your licensing information:

- A unique license key
- A license file containing the unique signature corresponding to your license

To register your license, click on the button “Add or manage license”:

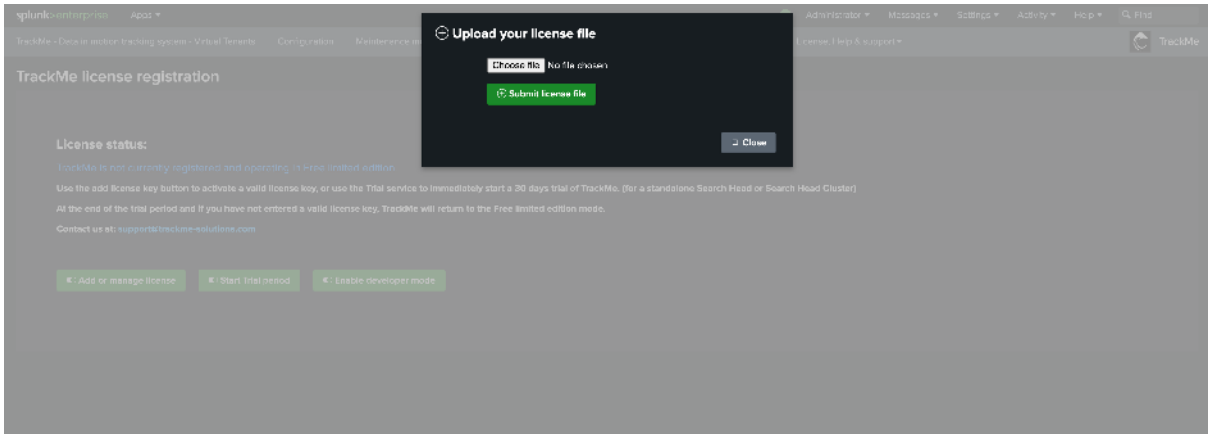


You can choose to:

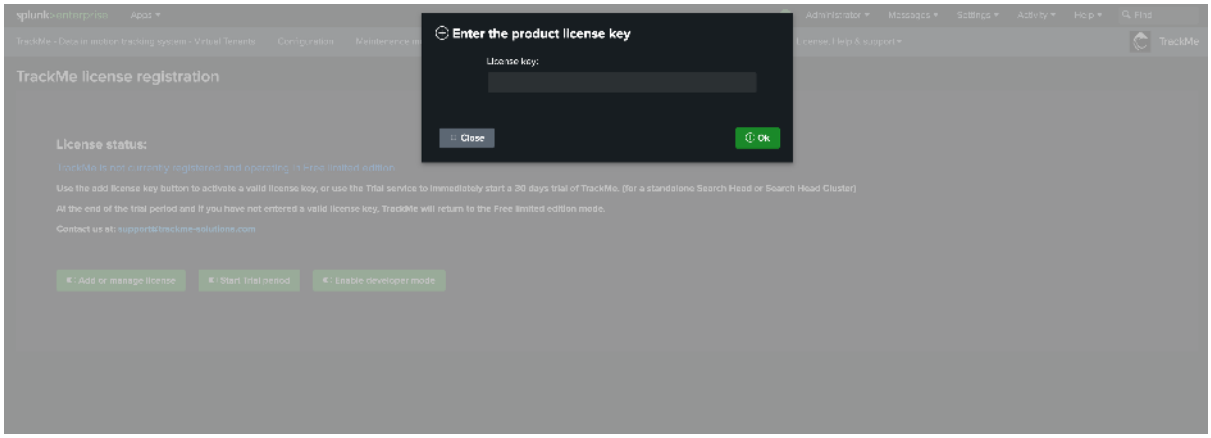
- Upload your license file containing the license signature, which does not require connectivity to our licensing API services
- Submit your license key, TrackMe will contact our licensing API services, verify and validate the license, and if successful will cache the license signature

Once successfully submitted, the license signature is cached in a KVstore, and all further license verifications are systematically made in an offline fashion without the need to contact our licensing API services.

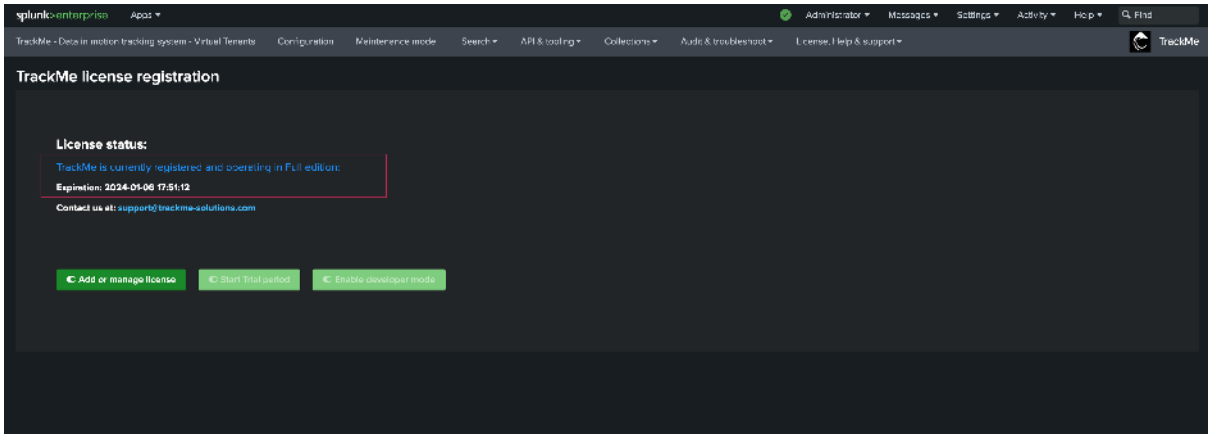
Uploading your license file activation:

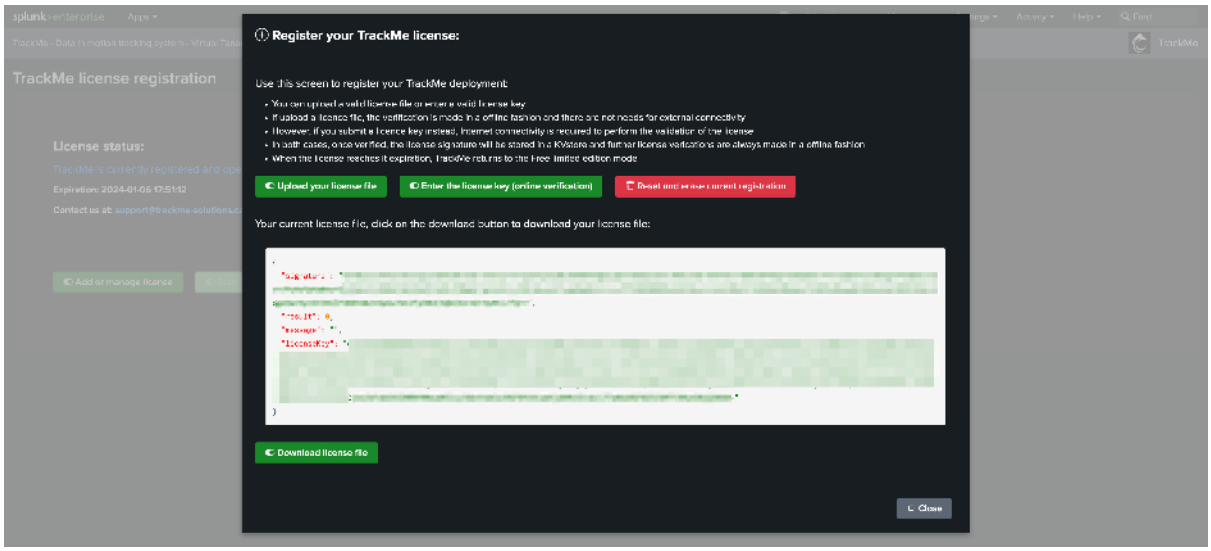


Submitting the license key for verification:



Once registered, the user interface shows the status and the expiration date, you can click on the "Add or manage license" button to review and download the license signature:



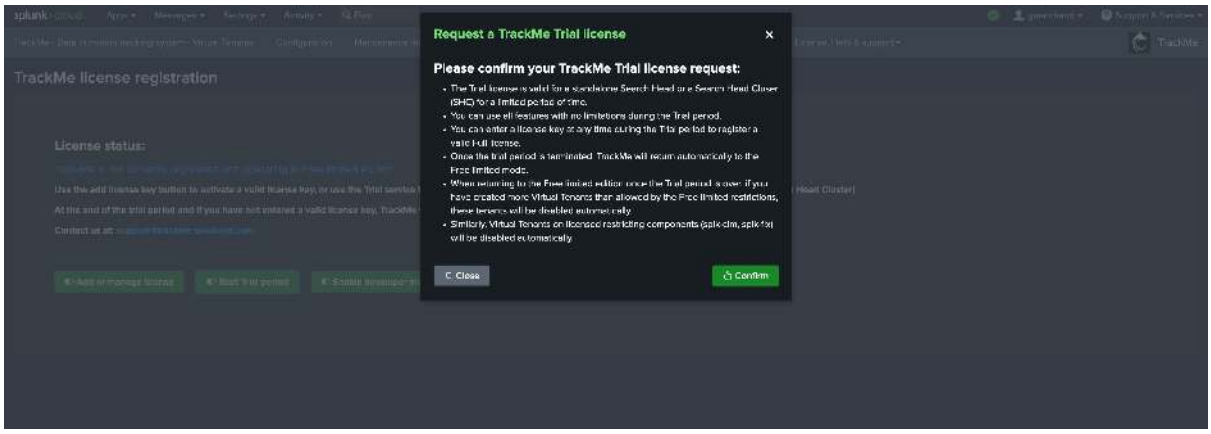


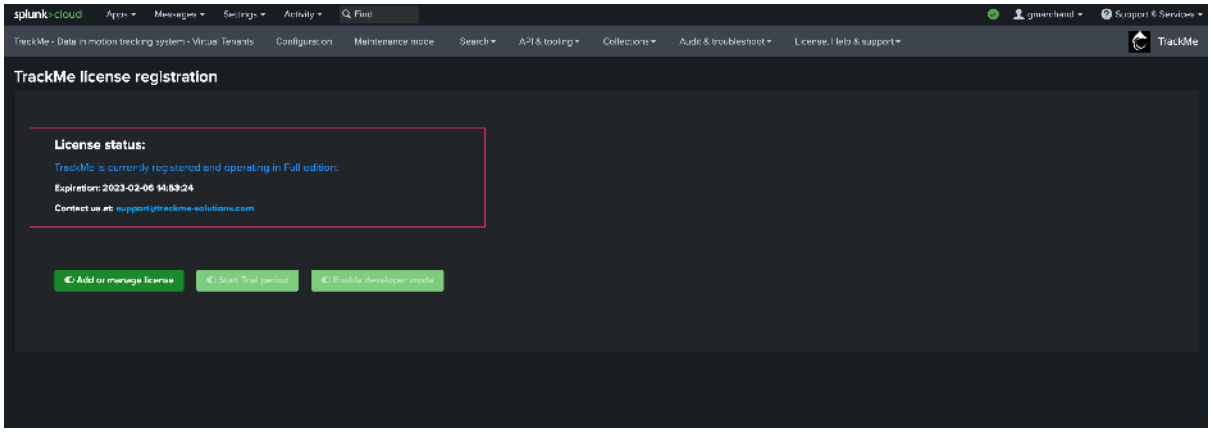
Start a Trial

You can request and start a Trial period immediately in TrackMe. A Trial period allows you to test all features with no restrictions for a limited period of time.

The Trial license can be generated directly from the application, although it requires that the Search Head has Internet connectivity to our licensing API.

If you need a Trial activation file for an environment with no connectivity, or wish to have a longer Trial period, please contact us at: support@trackme-solutions.com





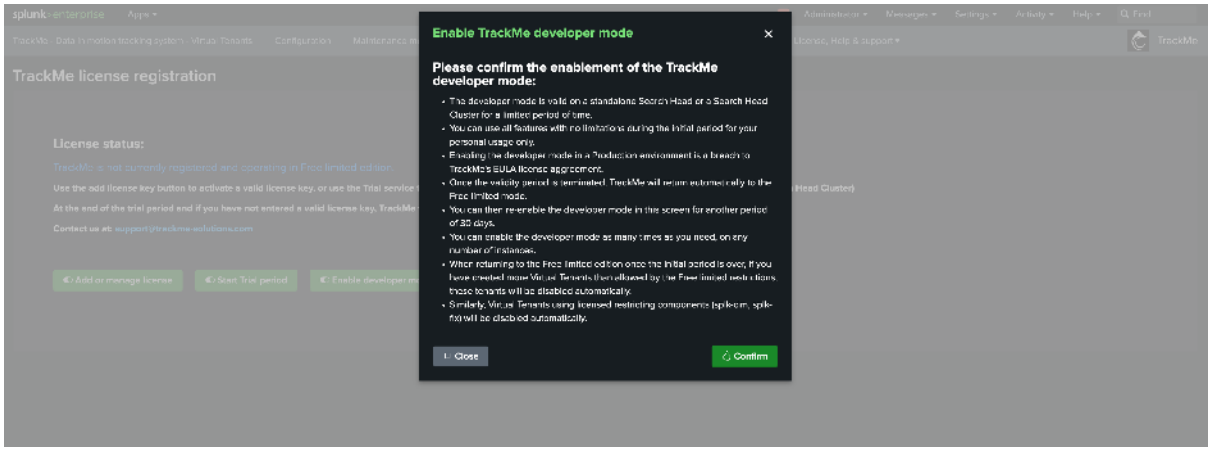
Trial additional information:

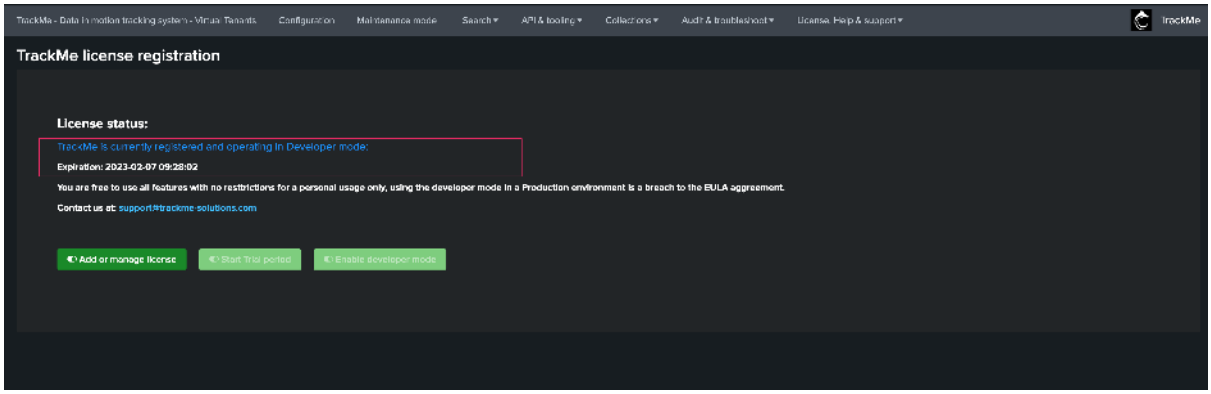
- The Trial license is valid for a standalone Search Head or a Search Head Cluster (SHC) for a limited period of time.
- You can use all features with no limitations during the Trial period.
- You can enter a license key at any time during the Trial period to register a valid Full license.
- Once the trial period is terminated, TrackMe will automatically return to the Free limited mode.
- When returning to the Free limited edition once the Trial period is over, if you have created more Virtual Tenants than allowed by the Free limited restrictions, these tenants will be automatically disabled.
- Similarly, Virtual Tenants using licensed restricting components (splk-cim, splk-flx) will be automatically disabled.

Enable the developer mode

You can enable the developer mode in non-Production contexts, which allows you to use all features with no restrictions during a period of 30 days, which can be repeated as many times as you wish:

- The developer mode is valid on a standalone Search Head or a Search Head Cluster for a limited period of time.
- You can use all features with no limitations during the initial period for your development, testing, and qualification usage only.
- Enabling the developer mode in a Production environment is a breach of TrackMe's EULA license agreement.
- Once the validity period is terminated, TrackMe will automatically return to the Free limited mode.
- You can then re-enable the developer mode in this screen for another period of 30 days.
- You can enable the developer mode as many times as you need, on any number of instances.
- When returning to the Free limited edition once the initial period is over, if you have created more Virtual Tenants than allowed by the Free limited restrictions, these tenants will be automatically disabled.
- Similarly, Virtual Tenants using licensed restricting components (splk-cim, splk-flx) will be automatically disabled.





Free Limited edition mode

You can use TrackMe in its default Free Limited edition mode with no restrictions in terms of periods and number of instances. However, there are restrictions in terms of features and especially the number of hybrid trackers:

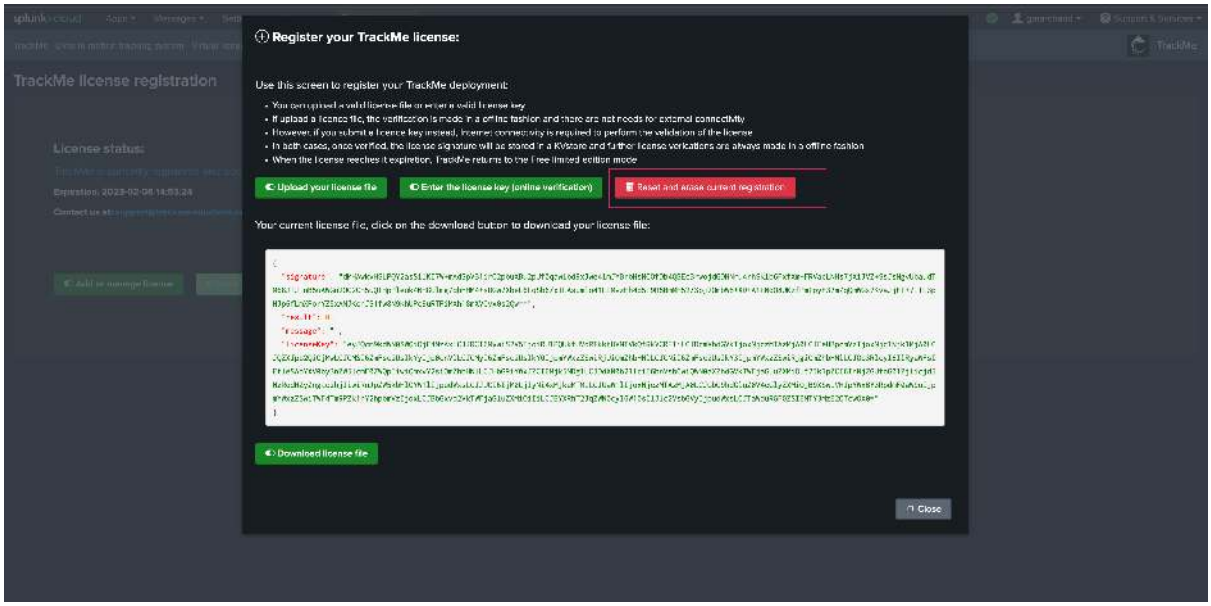
Consult our pricing page on TrackMe's Website for more details about the restrictions and plans:

- https://trackme-solutions.com/pricing

If you need more information or if you have any questions, contact us at:

- contact@trackme-solutions.com

To return to the Free Limited edition mode from Full or Trial, or if you enabled the developer mode, use the reset registration:


Register your TrackMe license:
TrackMe license registration
Use this screen to register your TrackMe deployment:
You can upload a valid license file or enter a valid license key
If upload a licence file, the verification is made in a offline fashion and there are not needs for external connectivity
However, if you submit a licence key instead, internet connectivity is required to perform the validation of the license
In both cases, once verified, the license signature will be stored in a KVstore and further license verications are always made in a offline fashion
When the license reaches it expiration, TrackMe returns to the Free limited edition mode
Upload your license file
Enter the license key (online verification)
Reset and erase current registration
Your current license file, click on the download button to download your license file:
Download license file
Close

CHAPTER

SEVEN

ADMINISTRATION GUIDE:

7.1 Configuration

Hint

Looking for a Quickstart?

- If you are looking for a quickstart, you can refer to the *QUICK START - Starting with TrackMe: (feed tracking quickstart)*
- The quickstart guide provides a step-by-step guide to get started with TrackMe, focusing on feed tracking, the easy and simple way

Using a service account for TrackMe is not mandatory

- It is not mandatory to use a service account for TrackMe, as described in the next steps, but it can be considered a good practice which provides several advantages in terms of management and monitoring
- However, this is not required as such and not doing so will not prevent TrackMe from operating normally
- By default, TrackMe will create knowledge objects and run searches as the "nobody" user (Splunk system account), which does not require any setup

Hint

Distinguishing permissions requirements between service accounts, TrackMe administrators, Power and read-only users

- A **service account** is a Splunk user (internal or SAML) that is used by TrackMe to perform scheduled activities, such as the creation of knowledge objects, the execution of scheduled searches, the creation of Virtual Tenants, etc.
- TrackMe users can have different levels of permissions. TrackMe comes with 3 built-in concepts for users: **administrators**, **power** users, and **read-only** users
- TrackMe leverages sophisticated techniques to ensure that you can define minimal permissions, for both the service account and the TrackMe users
- This allows for TrackMe users (TrackMe admins, power and read-only users) to avoid having to provide potentially dangerous capabilities such as list_settings, list_storage_passwords, etc.
- TrackMe comes with 3 built-in roles, **trackme_admin**, **trackme_power** and **trackme_user**

- **trackme_user** inherits from Splunk built-in **user** role, **trackme_power** inherits from **trackme_user** and **trackme_admin** inherits from **trackme_power**
- Each TrackMe built-in role enables the associated TrackMe capability, for instance **trackme_admin** enables the **trackmeadminoperations** capability

7.1.1 Summary Requirements for the TrackMe Service Account

The following requirements are the minimal requirements for the TrackMe service account:

Indexes Access

The service account should be able to search all `non-internal indexes` and all `internal indexes` (or at least the indexes containing data to be monitored, as well as the _internal index)

Capabilities

The service account should have the following capabilities at the minimum:

Capability	Note
get_metadata	
get_typeahead	
list_accelerate_search	
list_all_objects	
list_inputs	
list_metrics_catalog	
output_file	
pattern_detect	
request_remote_tok	
rest_access_server_endpoints	
rest_apps_view	
rest_properties_get	
rest_properties_set	
run_collect	
run_custom_command	
run_dump	
run_mcollect	
schedule_search	Required for running scheduled searches
search	
trackmeadminoperations	Can be inherited from the trackme_admin role
trackmepoweroperations	Can be inherited from the trackme_power role
trackmeuseroperations	Can be inherited from the trackme_user role

7.1.2 Summary Requirements for TrackMe Administrators

Essentially, TrackMe administrators need to have the following capability:

Capability	Note
trackmeadminoperations	Can be inherited from the trackme_admin role

In addition, and to be able to access and update the configuration menu items (Menu Configuration), administrators need:

- These are required capabilities by the Splunk UCC Framework, which is used by TrackMe for the purposes of the configuration level backend

- This is in fact optional; however, lacking these capabilities will not allow using the configuration UI to create remote service accounts for instance
- These capabilities would be required for users in charge of the highest level of administration of TrackMe; these are not required for the service accounts
- These capabilities are required only for the configuration UI; these are not required for the creation and/or management of TrackMe knowledge objects (virtual tenants, trackers, ...)

Capability	Note
list_settings	Allows accessing the configuration UI
list_storage_passwords	Allows accessing the configuration UI
admin_all_objects	Allows updating the configuration items

7.1.3 Service Account and Permissions

To operate, TrackMe allows and recommends defining a Splunk user that has the ownership of any knowledge objects created by TrackMe as part of the Virtual Tenant lifecycle:

- Knowledge Objects (such as reports, alerts...) will be assigned to the user tagged as the owner of the Virtual Tenant
- Scheduled activities will run on behalf of the service account owner

By default, TrackMe assigns the user "admin" as the default owner of the Virtual Tenant; it is best practice to create your own service account owner. The following minimal permissions and capabilities are required:

- The service account needs to be a member of the built-in role `trackme_admin` as this provides the `trackmeadminoperations` capability, or this capability needs to be granted explicitly
- The service account needs to be able to search all `non-internal indexes` and all `internal indexes`
- The service account needs to be able to run scheduled searches; typically you can use the Splunk built-in `power` role

TrackMe implements a strict least privileges approach; consult *Role Based Access Control and ownership*

Note

Local service account user or SAML service account

- You can set up the service account user as a local user or a SAML user on the TrackMe Search Heads tier
- For other Search Head tiers, TrackMe can interact with the Splunk API for various powerful use cases such as TrackMe Flex Object trackers or the Workload component
- This requires a service account on the target Search Heads tier and a bearer token to be created
- If you want to create a SAML service account for TrackMe's remote search capabilities, you need to have the SAML AQR setup, and an Identity Provider (IDP) supported by Splunk
- Reference: https://docs.splunk.com/Documentation/Splunk/latest/Security/Setupauthenticationwithtokens
- Reference: https://docs.splunk.com/Documentation/SplunkCloud/latest/Security/SAMLConfigJWT

7.1.4 Creating a Service Account for TrackMe with Minimal Permissions

Note

Version 2.0.48 and later required for minimal permissions

- TrackMe version 2.0.48 and later is required for the following procedure allowing a strict minimalist service account
- Before this version, the service account needs to have extended capabilities such as list_settings and list_storage_passwords capabilities; therefore, the recommendation was for the service account to be a member of admin/sc_admin
- Some advanced use cases such as Flex Object trackers dealing with the Splunk `| rest` command or SOAR-related use cases may need additional capabilities to be granted to the service account user

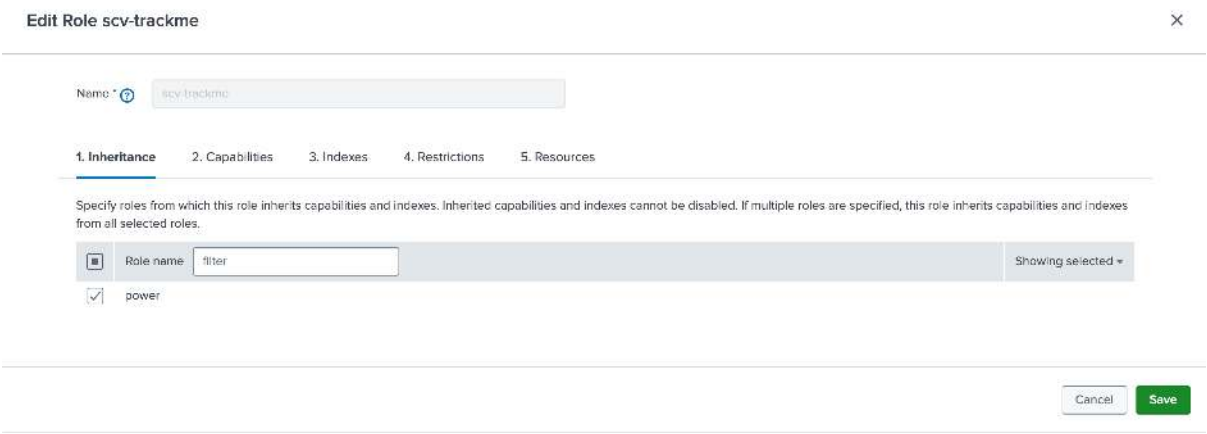
One option is to create a specific role for the TrackMe service account with:

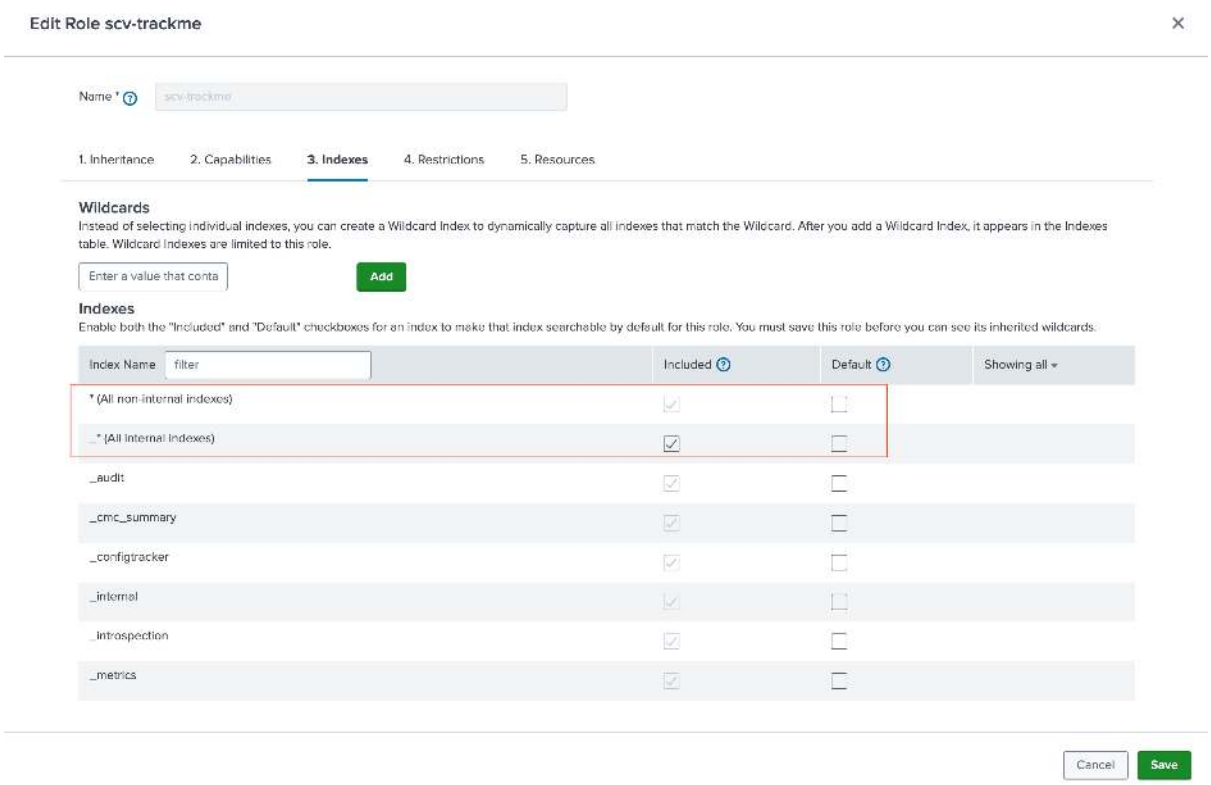
- Inheritance roles: `power`
- Role membership: `trackme_admin`
- Indexes: `all non-internal` and `all internal` indexes
- Resources: While TrackMe is optimized to distribute scheduled searches, it should be capable of running sufficient concurrent searches and it requires a large file quota to avoid issues

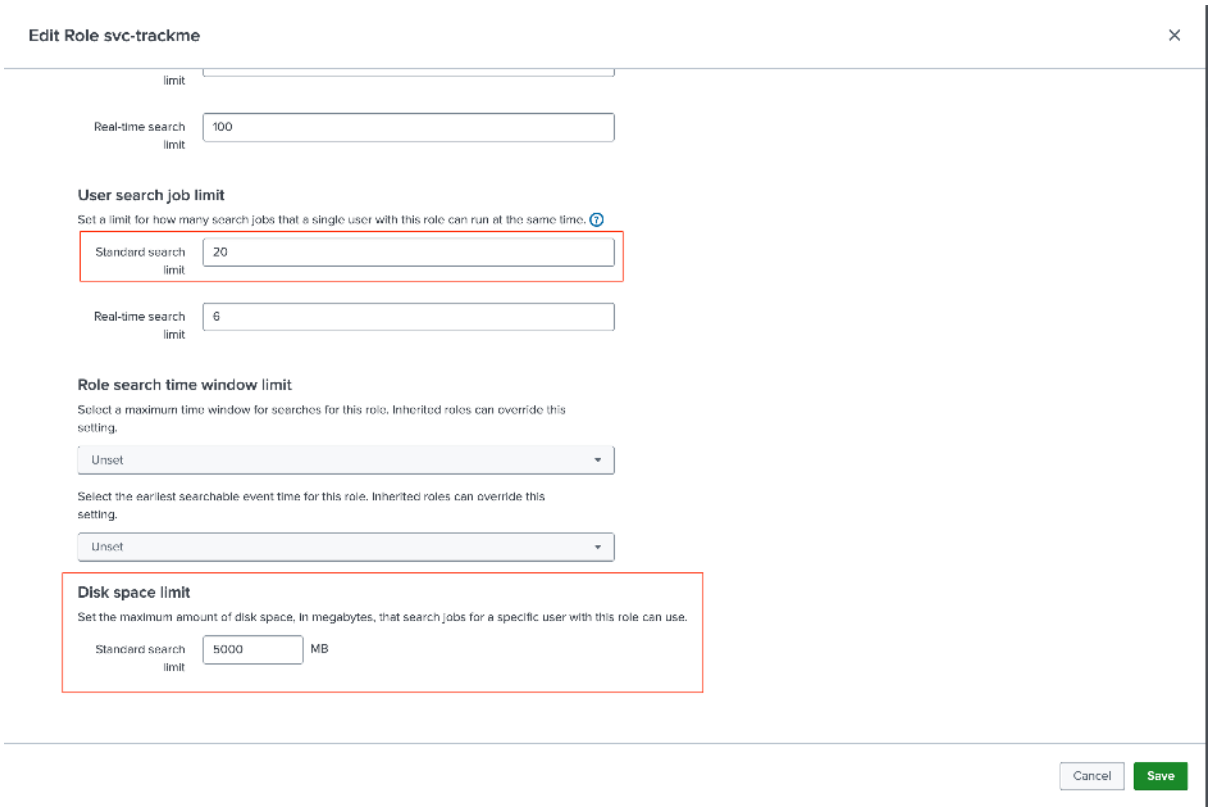
Hint

trackme_admin membership for the service account

- Before version 2.0.61, the service account needs to be an explicit member of the `trackme_admin` role (or the admin role in the tenants); this is needed because TrackMe requires explicit role membership (opposed to inheritance) to grant access to the Virtual Tenants
- From version 2.0.61, all RBAC dimensions in TrackMe support inheritance transparently

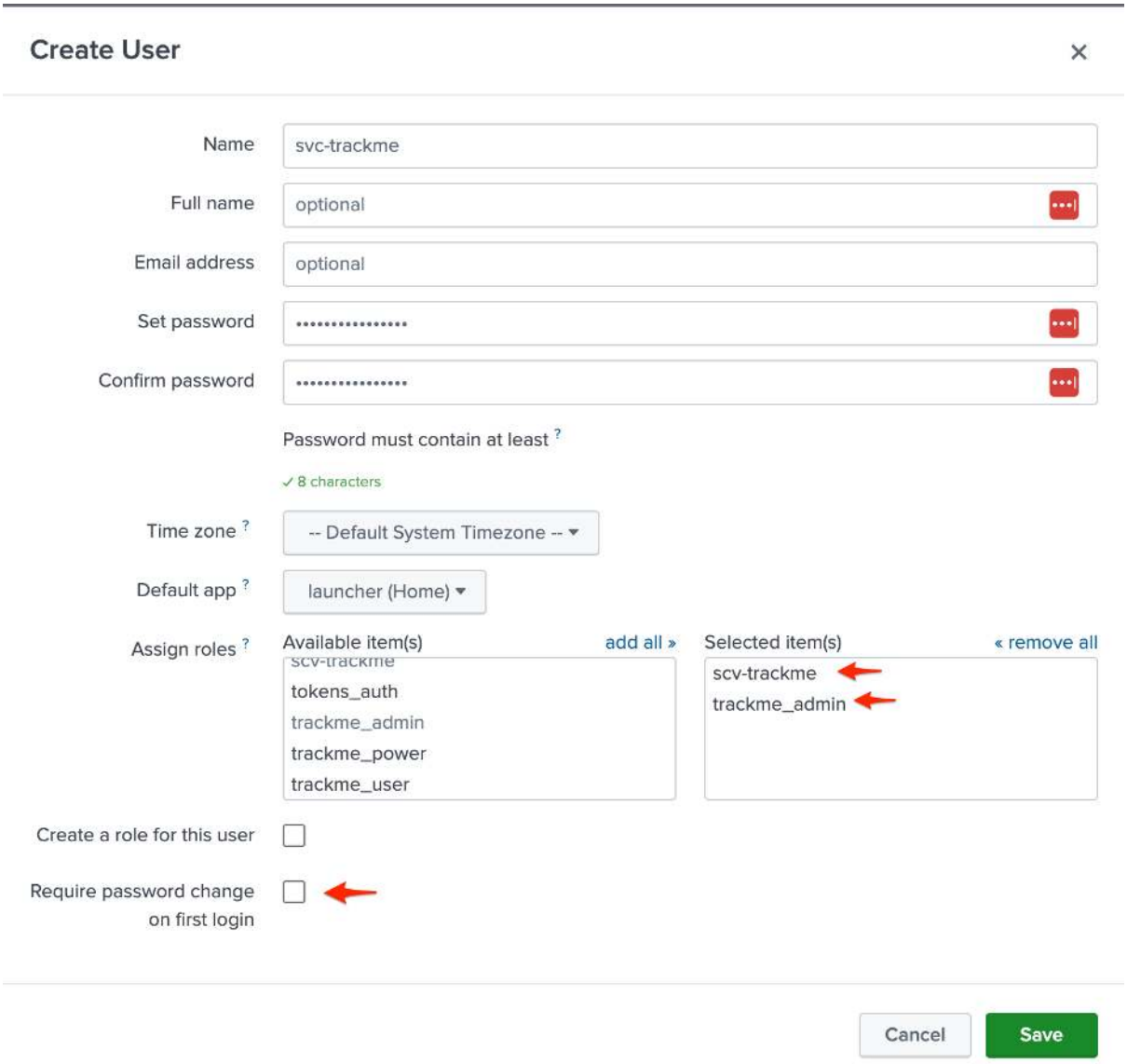




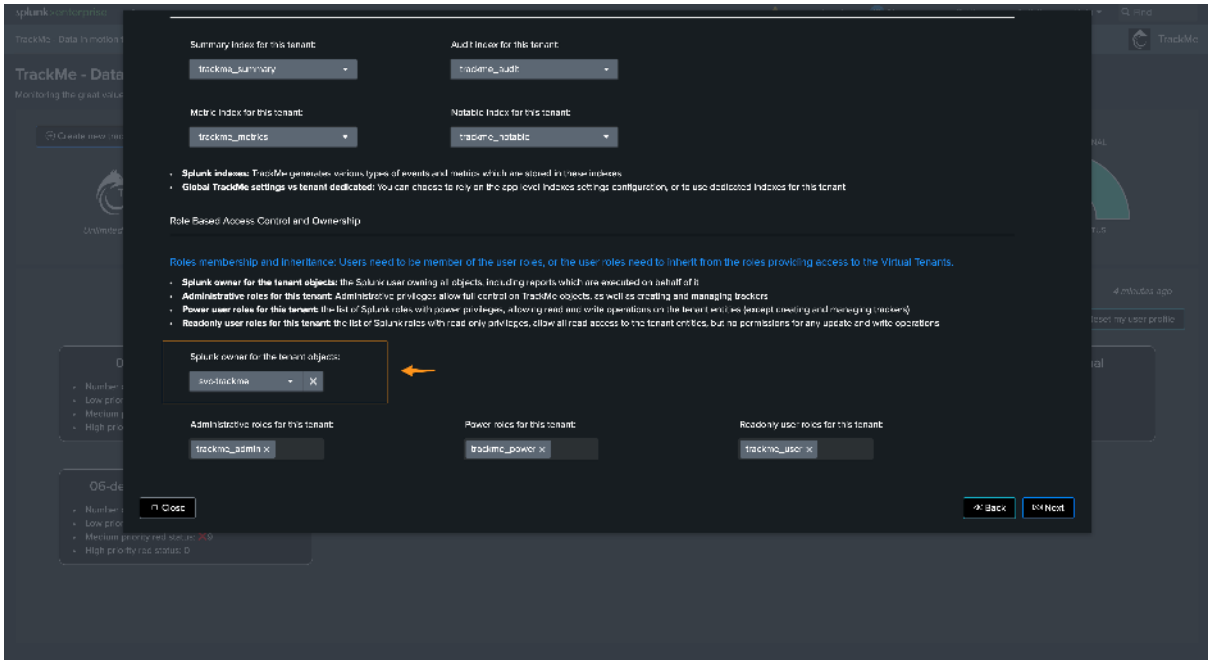


You can can then create the service account itself, example:

- The user is a member of the `svc-trackme` role
- As mentioned above, it is also a member of `trackme_admin` to be granted access to the Virtual Tenants
- Uncheck the box "Require password change on first login"



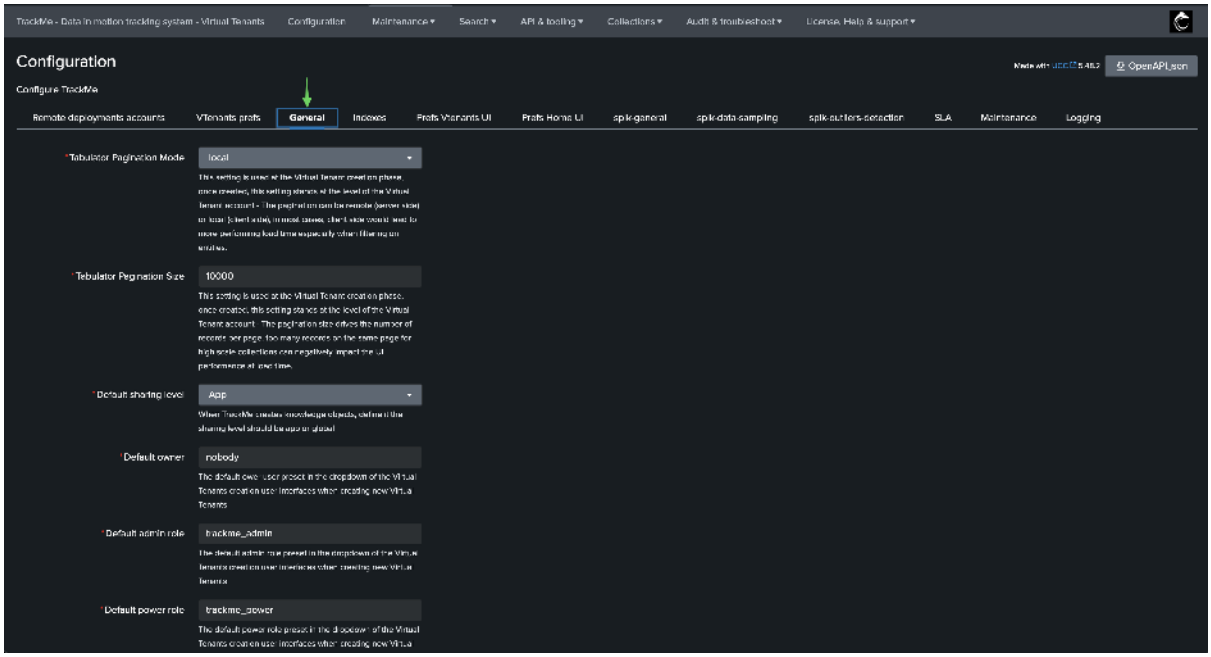
When you create a Virtual Tenant, you will specify the service account as the owner of the Virtual Tenant:

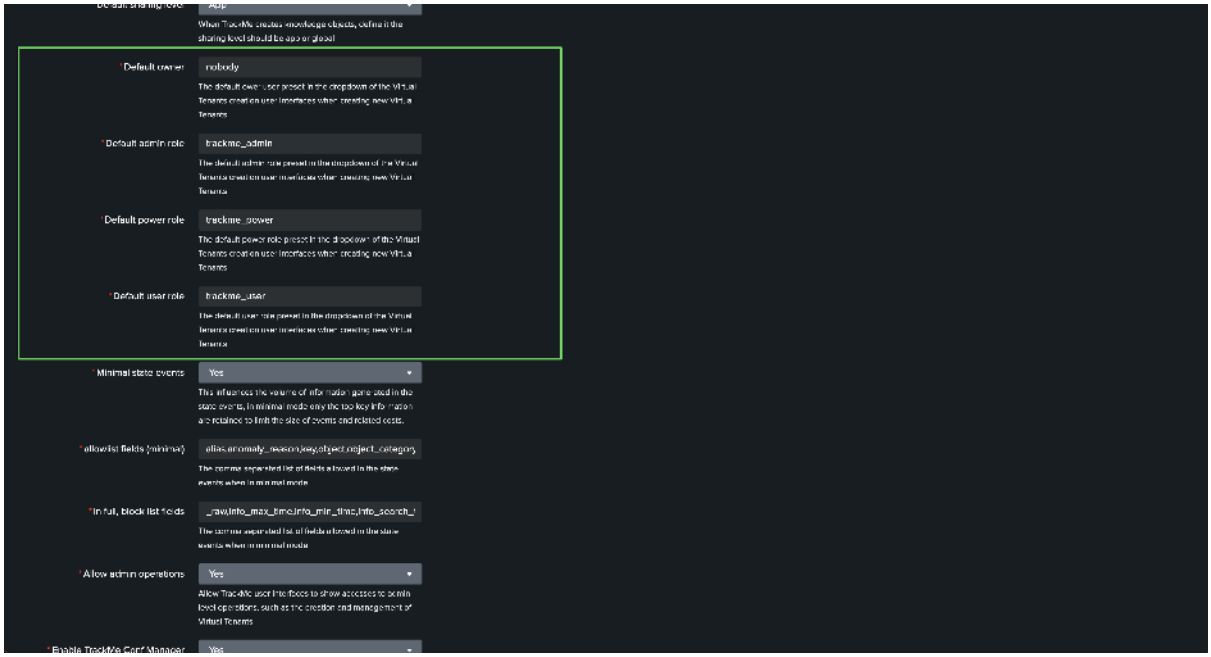


Hint

preset RBAC for the tenant creation UI

- Since the **version 2.0.52**, you can preset values for the owner and roles when creating a new Virtual Tenant from the UI
- Go in the Configuration then General Configuration





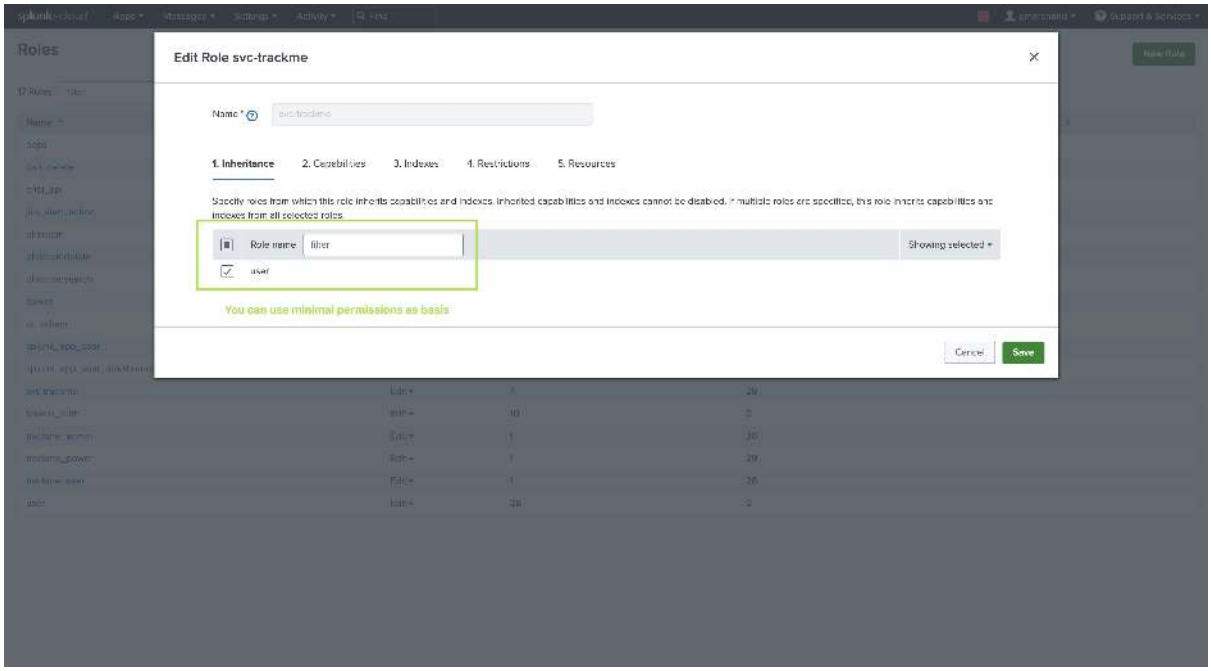
7.1.5 Minimal capabilities and resources for Remote Accounts and the user associated with the bearer token

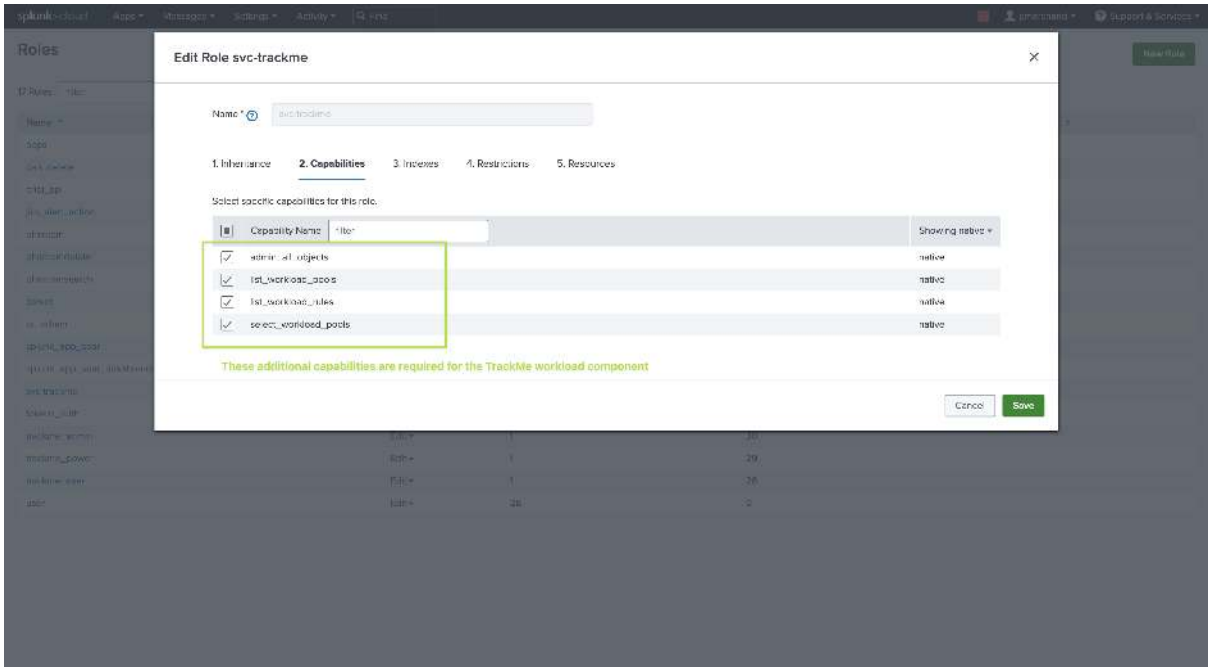
TrackMe remote capabilities rely on a Splunk bearer token authentication, this token is associated with a Splunk user on the remote side which itself is associated with specific roles, capabilities, permissions and resources restrictions:

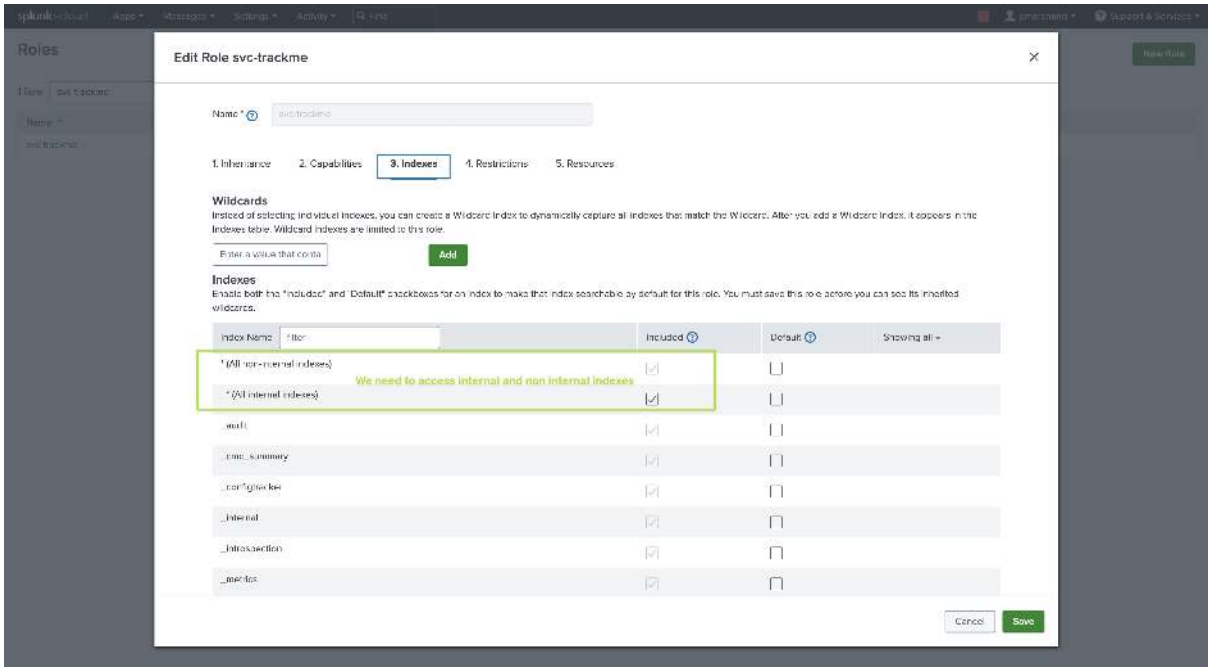
- **Roles and capabilities:** The user can be created with minimal permissions using the Splunk `user` role out of the box role. (you can inherit from user or a role providing the same capabilities than power)
- **Indexes:** Make sure the user can access to both normal and `internal` indexes.
- **Restrictions:** The user for TrackMe should not have any time limits restrictions, there are use cases which require long term searches.
- **Resources:** It is recommended to give to this user `enough concurrent searches` (unlike very basic or minimal user) as well as a `sufficient quota`. (5GB or 10Gb for instance)
- **Additional capability required:** Finally, for the purposes of the Workload, this user also needs to have the following capabilities granted `admin_all_objects`, `select_workload_pool`, `list_workload_pools` and `list_workload_rules` which are required for TrackMe's backend to access to all objects of all applications in a remote manner. (for the Metadata in the Workload component)

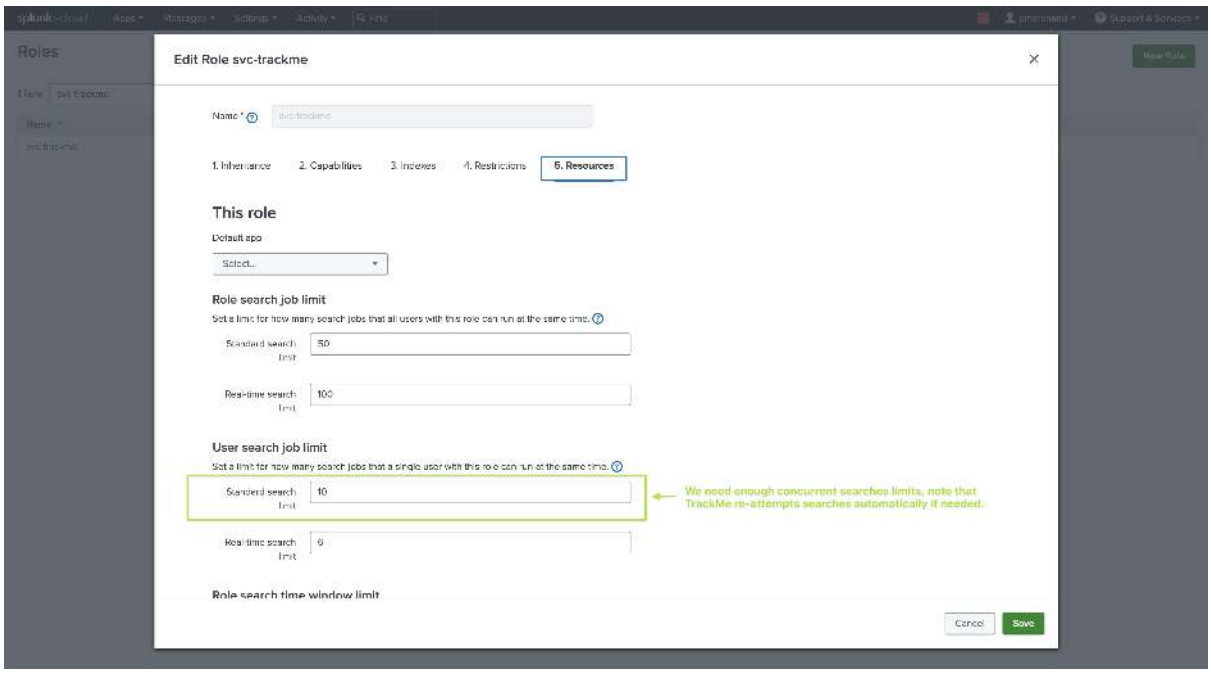
Table 1: In addition with the basic capabilities provided by the user role, these capabilities must be granted

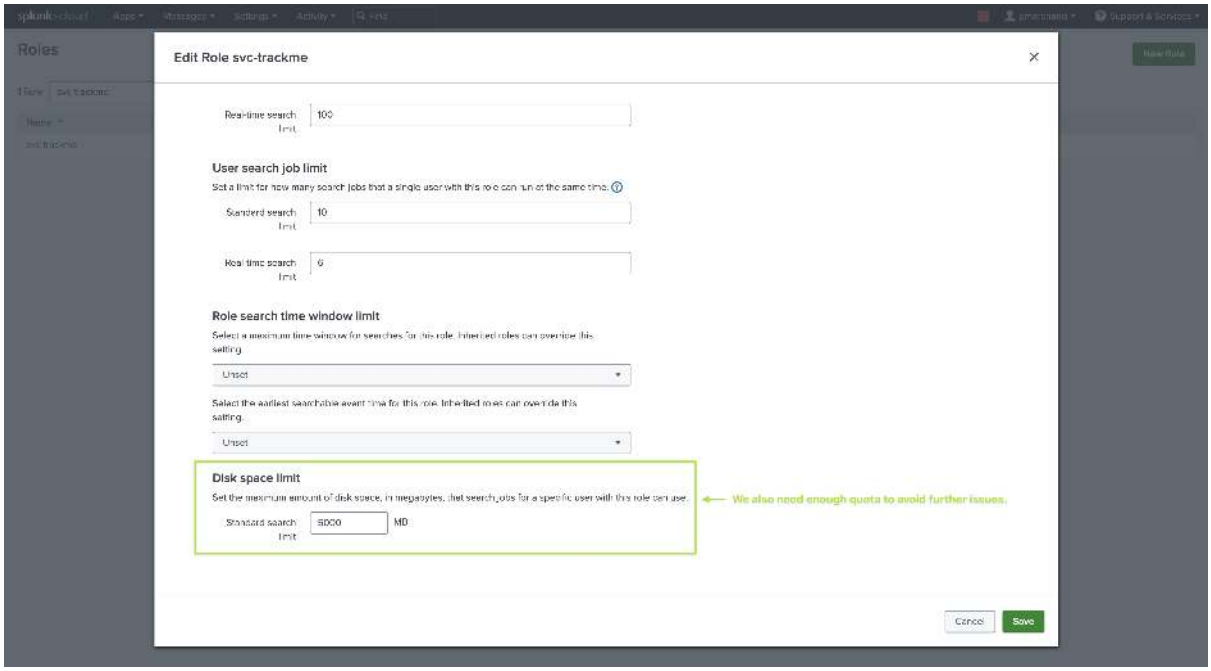
Capability	Comment
admin_all_objects	Required
select_workload_pool	Required
list_workload_pools	Required
list_workload_rules	Required


Roles
Edit Role svc-trackme
Name *
1. Inheritance
2. Capabilities
3. Indexes
4. Restrictions
5. Resources
Role name
filter
Showing selected
user
You can use minimal permissions as basis
Cancel
Save


Roles
Edit Role svc-trackme
Name *
1. Inheritance
2. Capabilities
3. Indexes
4. Restrictions
5. Resources
Select specific capabilities for this role.
Capability Name
filter
Showing native
admin_all_objects
native
list_workload_pools
native
list_workload_rules
native
select_workload_pools
native
These additional capabilities are required for the TrackMe workload component
Cancel
Save


Edit Role svc-trackme
1. Inheritance
2. Capabilities
3. Indexes
4. Restrictions
5. Resources
Wildcards
Add
Indexes
We need to access internal and non internal indexes
Cancel
Save


Edit Role svc-trackme
1. Inheritance
2. Capabilities
3. Indexes
4. Restrictions
5. Resources
This role
Default app
Role search job limit
Standard search limit
50
Real-time search limit
100
User search job limit
Standard search limit
10
We need enough concurrent searches limits, note that TrackMe re-attempts searches automatically if needed.
Real-time search limit
0
Role search time window limit
Cancel
Save



7.1.6 Users and roles

TrackMe is deeply RBAC capable, consult the following documentation to configure users accesses for TrackMe:

- *Role Based Access Control and ownership*

7.1.7 Web Browsers and system compatibility

TrackMe should work fine with most Web Browsers and systems; however, if you experience icon issues due to the lack of support of ASCII emojis, you can enable the Bootstrap compatibility mode:

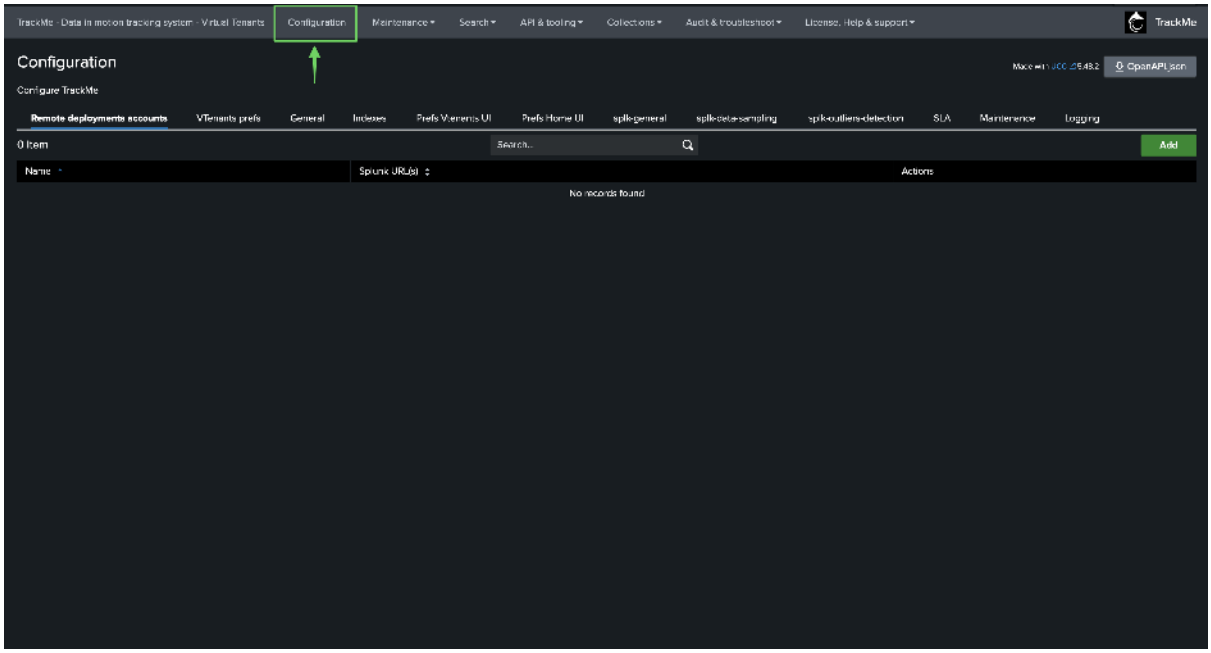
- *Web Browser compatibility*

7.1.8 Accessing TrackMe Configuration

TrackMe relies on the Splunk UCC Framework for the purposes of the configuration level backend:

- https://splunk.github.io/addonfactory-ucc-generator

The Splunk UCC framework provides various powerful features which are leveraged notably for the purposes of handling the application-level configuration; for these purposes, a configuration user interface is available:



Default configurations are located in the following configuration file:

```
trackme/default/trackme_settings.conf
```

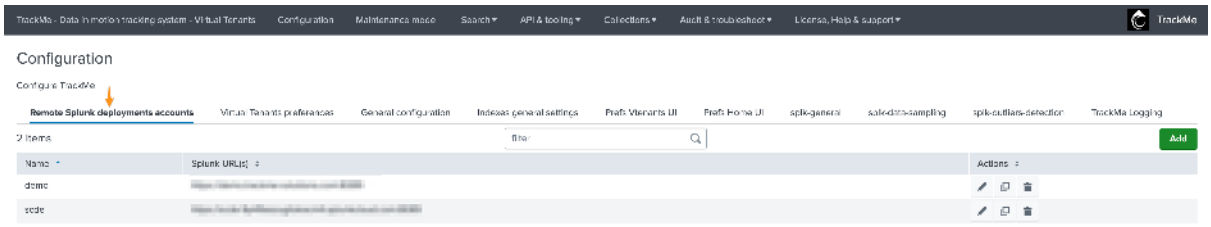
The configuration can therefore be performed via:

- The configuration user interface: this creates a local/trackme_settings which is automatically replicated among the members when running in a Search Head Cluster
- By deploying a local/trackme_settings.conf accordingly (if running in Search Head Cluster, this file would be located in shcluster/apps/trackme/local/trackme_settings)

However, the recommended method as a basis is to configure TrackMe through the intended configuration user interface.

7.1.9 Remote Splunk deployments accounts

The Splunk remote deployments accounts tab is where you will configure any remote Splunk environment you will monitor with TrackMe, if any,



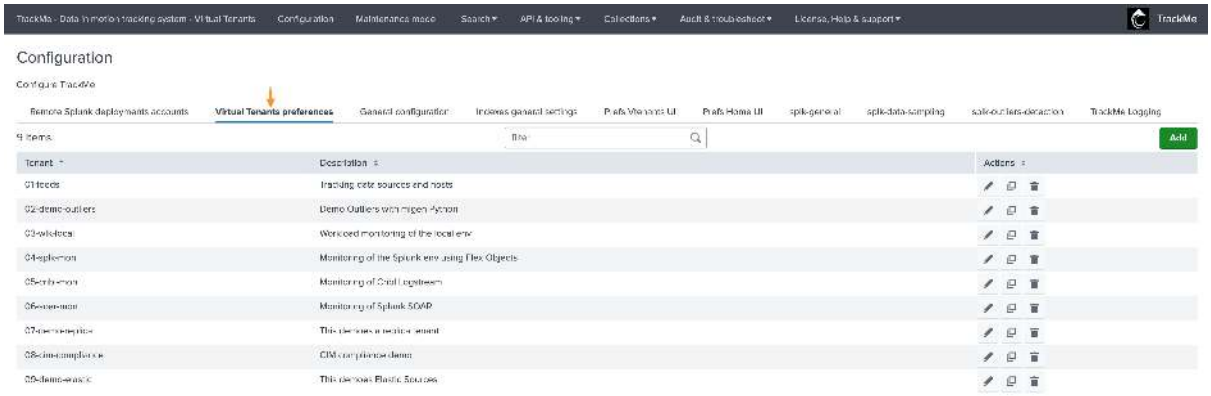
Splunk remote deployment accounts are documented here: *Splunk Remote Deployments (splunkremotesearch)*

Table 2: Remote Splunk Deployments Accounts

Field	Default Value	Description
name		Enter a unique name for this Splunk remote environment. Must be 1 to 50 characters, begin with a letter, and consist of lower case alphanumeric characters and underscores.
splunk_url		A list of comma-separated targets, e.g., https://splunk1:8089,https://splunk2:8089 (SSL is enforced and URLs will be prefixed with https:// if not set). The URL can be based on IP or FQDN.
bearer_token		Set the bearer token used for remote access to the KVstore instance (client instances only).
app_namespace	search	The Splunk application namespace on the remote system where searches will be executed. Defaults to the "search" app.
rbac_roles	admin,sc_admin,trackme_	A comma-separated list of Splunk roles that are allowed to access this account, either by direct membership or inheritance.
timeout_connect_check	15	The maximal timeout value in seconds for the health check connection test. If the health check fails, the next target in the pool is used (if applicable). Defaults to 15 seconds.
timeout_search_check	300	The maximal timeout value in seconds for the remote search connection. Increase this value if the target responds slowly. Defaults to 300 seconds.

7.1.10 Virtual Tenants Accounts

Virtual Tenants Accounts are created and deleted automatically by TrackMe when managing Virtual Tenants through the Web or REST API; you can update the tenant-level configuration in this screen:



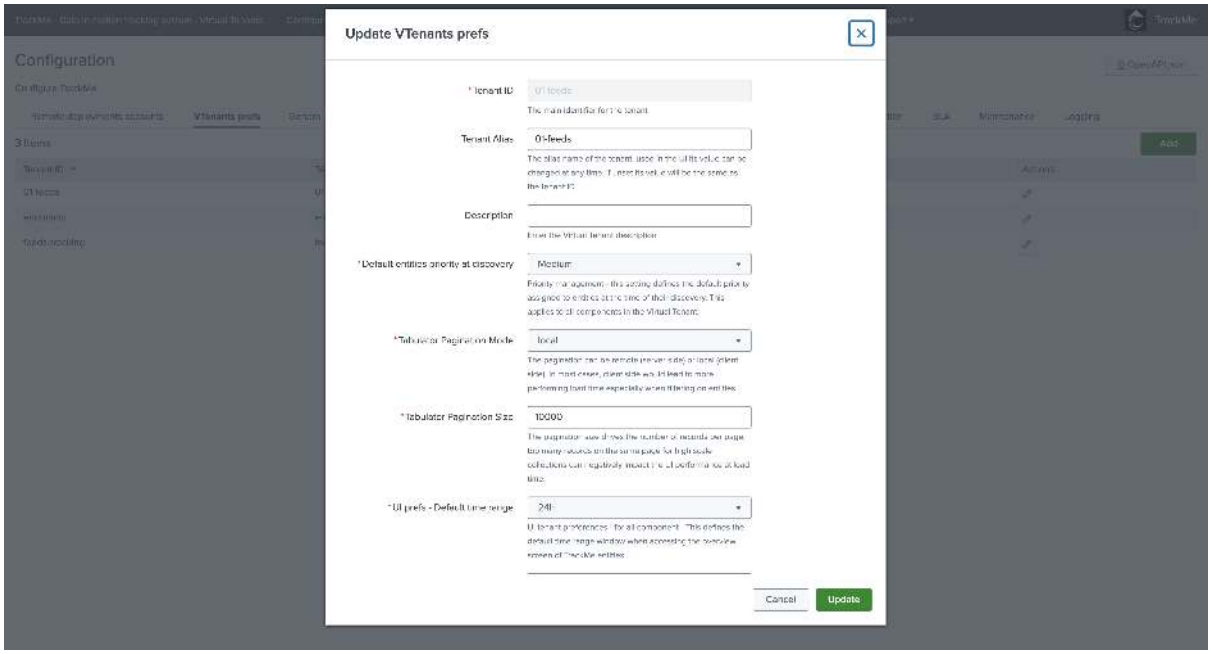

Update VTenants prefs
Tenant Alias
Description
Default entities priority at discovery
Medium
Tabulator Pagination Mode
local
Tabulator Pagination Size
10000
UI prefs - Default time range
24h
Cancel
Update

Table 3: Virtual Tenants Preferences

Field	Label	Default Value	Description
name	Tenant ID		The main identifier for the tenant (Tenant ID). Must only contain alphanumeric characters, hyphens, and underscores.
alias	Tenant Alias		The alias name of the tenant, used in the UI. Its value can be changed at any time; if unset, it will default to the tenant ID.
description	Tenant Description		A description of the virtual tenant, must be between 1 and 100 characters long.
default_priority	Default entities priority at discovery	medium	Defines the default priority assigned to entities at the time of their discovery (Critical, High, Medium, Low).
pagination_mode	Tabulator Pagination Mode	local	Defines pagination as either remote (server side) or local (client side). In most cases, client-side pagination improves performance.
pagination_size	Tabulator Pagination Size	10000	Determines the number of records per page. Too many records per page may negatively impact UI performance.
ui_default_timerange	UI prefs - Default time range	24h	Defines the default time range for the tenant's overview screen (options include 30m, 60m, 2h, etc.).
ui_min_object_width	UI prefs - Object min width	300	Minimum width (in pixels) for the object field in the UI, to accommodate longer entity names.
ui_expand_metrics	UI prefs - expand metrics	0	Whether to expand metrics information by default (Yes = 1, No = 0).
ui_home_tabs_order	UI prefs - Home tabs visibility and order	dsm,cim,flx,dhm,mhm,w	List of tabs to be displayed in the Home UI in a comma-separated list. The order of the tabs is defined by the order in the list.
outliers_set_state	Red on Outliers	1	Allows outliers to influence entity status (Yes = 1, No = 0).
data_sampling_set_sta	Red on sampling	1	For splk-dsm only: allows data sampling to influence entity status (Yes = 1, No = 0).

7.1.11 General

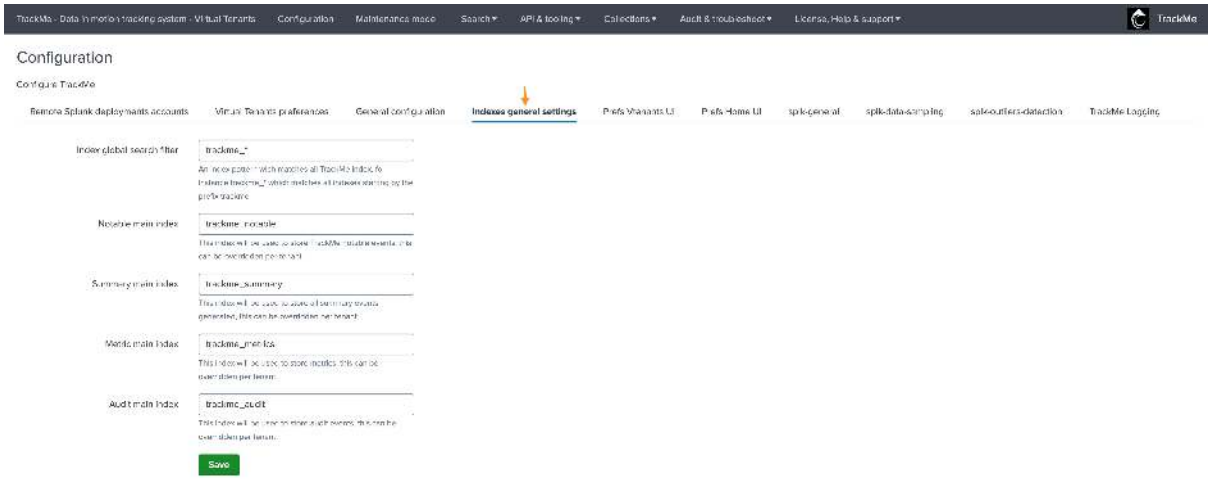
This tab defines various general configuration:

Table 4: TrackMe General Configuration

Field	Label	Default Value	Description
pagination_mode	Tabulator Pagination Mode	local	This setting is used at the Virtual Tenant creation phase. Once created, this setting stands at the level of the Virtual Tenant account. Pagination can be remote (server-side) or local (client-side), with client-side offering better performance in most cases.
pagination_size	Tabulator Pagination Size	10000	Drives the number of records per page. Too many records can negatively impact the UI performance.
trackme_defa	Default sharing level	app	Defines whether knowledge objects should be shared at the app or global level when created.
trackme_owne	Default owner	admin	The default owner user preset in the dropdown of the Virtual Tenants creation UI.
trackme_adm	Default admin role	trackme_adm	The default admin role preset in the dropdown of the Virtual Tenants creation UI.
trackme_powe	Default power role	trackme_powe	The default power role preset in the dropdown of the Virtual Tenants creation UI.
trackme_user_	Default user role	trackme_user	The default user role preset in the dropdown of the Virtual Tenants creation UI.
state_events_	Minimal state events	1	Influences the volume of information in state events. In minimal mode, only key information is retained to limit size and costs.
state_events_	Allowlist fields (minimal)	alias,anomaly_	Comma-separated list of fields allowed in state events when in minimal mode.
state_events_	Blocklist fields (full mode)	_raw,info_ma	Comma-separated list of fields blocked in state events when in full mode.
allow_admin_o	Allow admin operations	1	Allows TrackMe UI to show accesses to admin-level operations such as creating and managing Virtual Tenants.
enable_conf_ma	Enable TrackMe Conf Manager	0	Enables the TrackMe conf manager receiver, allowing admin-level operations to be sent to the receiver for replay in the target environment.
trackme_ack_	Default Ack duration	86400	Default duration (in seconds) for the acknowledgment action for entities in alert.
trackme_ack_	Expire Ack on anomaly reason change	1	Automatically removes an Ack when the anomaly reason changes.
trackme_ack_	Expire Ack on anomaly reason change min time	3600	Minimum time in seconds between the creation of an Ack and its expiration due to an anomaly reason change.
trackme_ack_	Expire Ack on anomaly reason change (auto Ack only)	1	Restricts Ack expiration due to anomaly changes to automatic Acks only. User Acks are not impacted.
trackme_ack_	Remove Ack on green state	1	Automatically removes an Ack when the entity returns to green state, unless sticky Ack is enabled.

7.1.12 Indexes general settings

This tab defines the indexes by default for Virtual Tenants:



If you intend to create Virtual Tenants specific indexes, we strongly recommend using a prefix pattern as a strict convention, for instance:

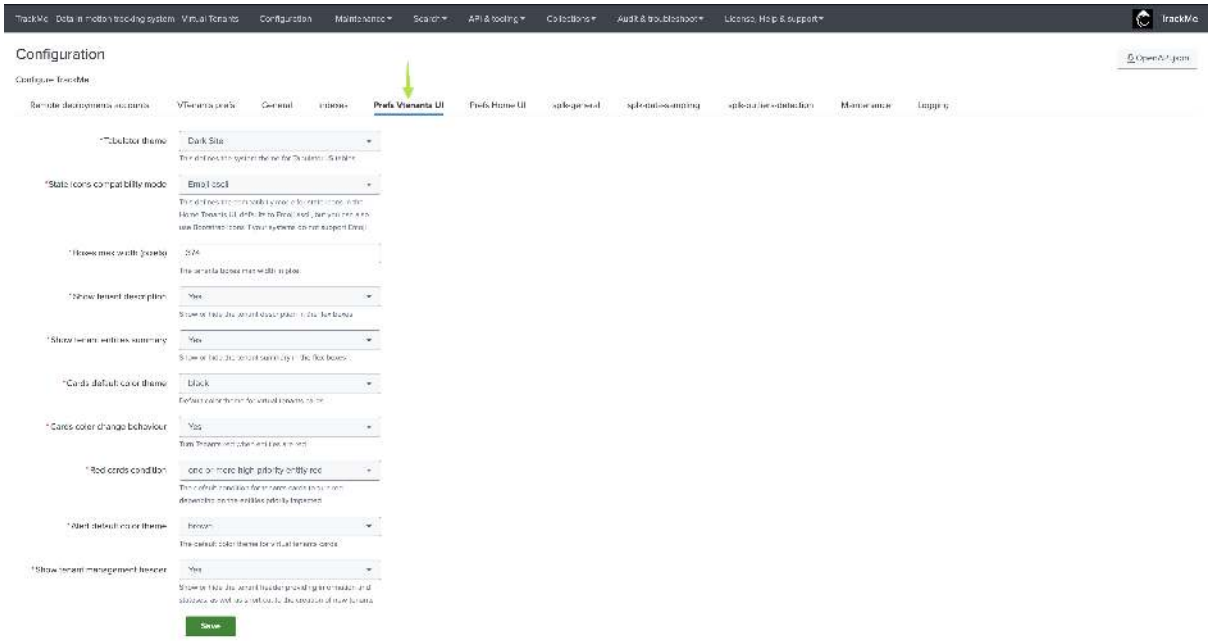
- trackme_<context>_<index name>

Table 5: Indexes General Settings

Field	Default Value	Description
trackme_idx_search_fil	trackme_*	An index pattern which matches all TrackMe index, for instance `trackme_*` which matches all indexes starting by the prefix trackme
trackme_notable_idx	trackme_notable	This index will be used to store TrackMe notable events, this can be overridden per tenant
trackme_summary_idx	trackme_summary	This index will be used to store all summary events generated, this can be overridden per tenant
trackme_metric_idx	trackme_metrics	This index will be used to store metrics, this can be overridden per tenant
trackme_audit_idx	trackme_audit	This index will be used to store audit events, this can be overridden per tenant

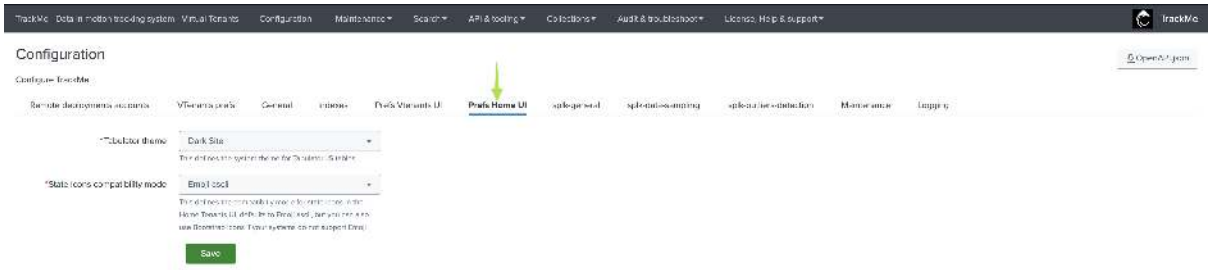
7.1.13 Prefs Vtenants UI

You can define default theme preferences for the Virtual Tenant user interface, users can update these preferences for their own profile too:



7.1.14 Prefs Home UI

You can define default theme preferences for the Home user interface:



7.1.15 splk-general

This tab defines various options specific to Splunk:

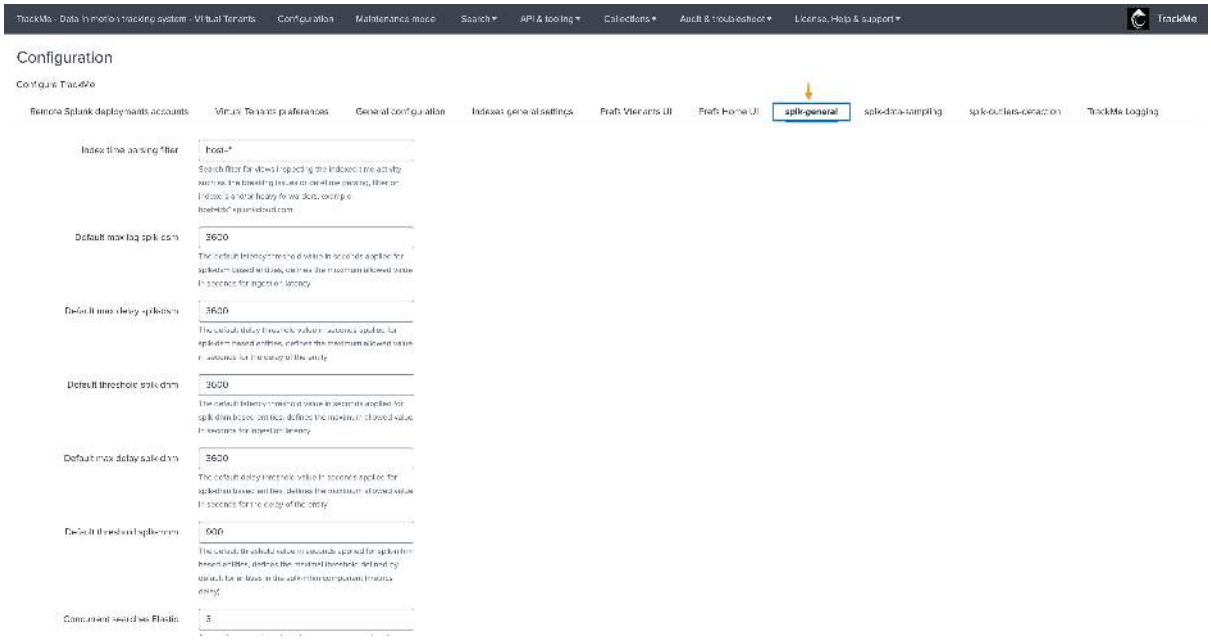

TrackMe - Data in motion tracking system - Virtual Tenants
Configuration
Maintenance mode
Search
API & tooling
Collections
Audit & troubleshoot
License, Help & support
TrackMe
Configuration
Configure TrackMe
Remote Splunk deployments accounts
Virtual Tenants preferences
General configuration
Indexes general settings
Pref: Vtenants UI
Pref: Home UI
splk-general
splk-dma-sampling
splk-outliers-detection
TrackMe Logging
Index time parsing filter
host=*
Default max lag splk-dsm
3600
Default max delay splk-dsm
3600
Default threshold splk-dhm
3600
Default max delay splk-dhm
3600
Default threshold splk-mhm
900
Concurrent searches Elastic
5

Table 6: splk-general Configuration

Field	Label	Default Value	Description
splk_general_idx_	Index time parsing filter	host=*	Search filter for views inspecting indexed time activity such as line breaking issues or datetime parsing. Filter on indexers and/or heavy forwarders, example: host=idx*.splunkcloud.com.
splk_general_dsm_	Latency default (splk-dsm)	3600	The default latency threshold value in seconds applied for splk-dsm based entities. Defines the maximum allowed value for ingestion latency.
splk_general_dsm_	Delay default (splk-dsm)	3600	The default delay threshold value in seconds for splk-dsm based entities. Defines the maximum allowed delay for entities.
splk_general_dhm_	Latency default (splk-dhm)	3600	The default latency threshold value in seconds applied for splk-dhm based entities. Defines the maximum allowed value for ingestion latency.
splk_general_dhm_	Delay default (splk-dhm)	3600	The default delay threshold value in seconds for splk-dhm based entities. Defines the maximum allowed delay for entities.
splk_general_mhm	Delay default (splk-mhm)	900	The default threshold value in seconds for splk-mhm based entities, defining the maximum metrics delay.
splk_general_feeds	Future indexing tolerance (splk-dsm/splk-dhm)	-600	For splk-dsm/splk-dhm only, defines the negative amount of time (in seconds) used for tolerance before data is assumed to be indexed in the future.
splk_general_feeds	Auto disablement period (splk-dsm/splk-dhm/splk-mhm)	60d	For splk-dsm/splk-dhm/splk-mhm only, defines the period in days after which inactive entities get automatically disabled. Set to 0d to disable this feature.
splk_general_elast	Concurrent searches Elastic	3	Defines the number of parallel concurrent searches for Shared Elastic sources at the system level. This can be overridden per tenant using *max_concurrent_searches*.
splk_general_dsm_	CMDB lookup search splk-dsm	inputlookup my_cmdb where (index="$data_index$ AND sourcetype="$data_sourc	Defines the CMDB lookup search for splk-dsm. You can use tokens for any field maintained in the KVstore collection.
splk_general_dhm_	CMDB lookup search splk-dhm	inputlookup my_cmdb where (host="$alias$")	Defines the CMDB lookup search for splk-dhm. You can use tokens for any field maintained in the KVstore collection.
splk_general_mhm	CMDB lookup search splk-mhm	inputlookup my_cmdb where (host="$alias$")	Defines the CMDB lookup search for splk-mhm. Tokens like *$tenant_id$* can make the search tenant-specific.
splk_general_cim_	CMDB lookup search splk-cim	inputlookup	Defines the CMDB lookup search for splk-cim. Tokens like *$tenant_id$* can make the search tenant-specific.

7.1.16 splk-data-sampling

This tab defines various options specific to the Data Sampling feature for splk-dsm (splk-feeds):

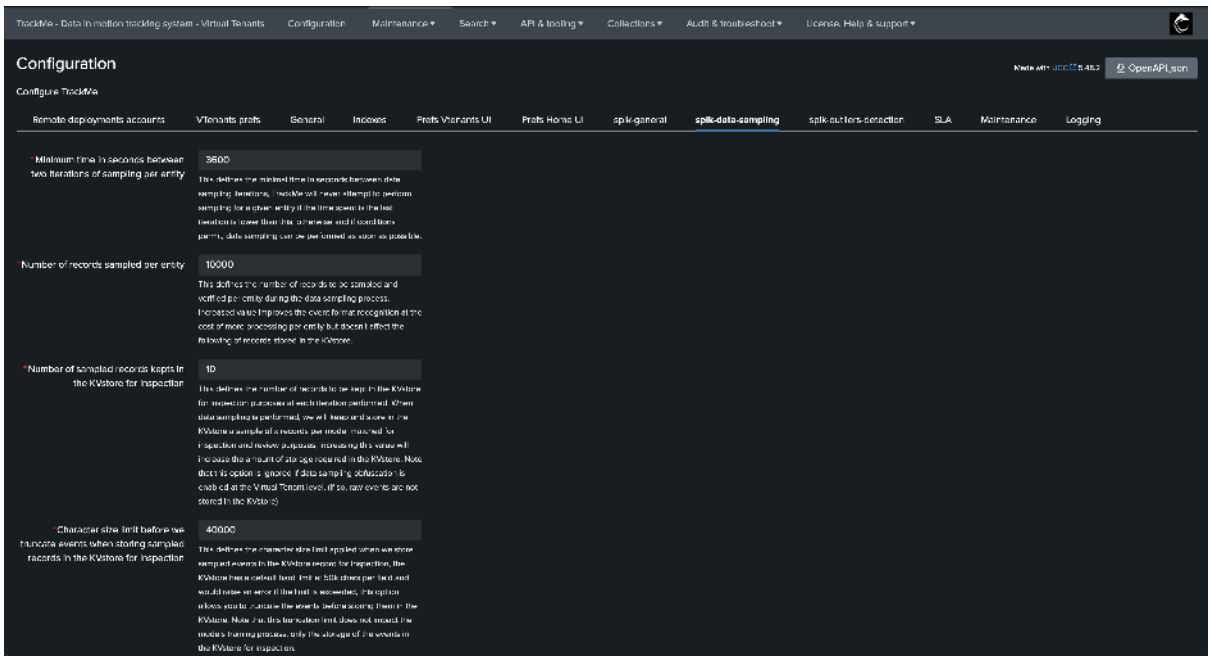


Table 7: splk-data-sampling Configuration

Field	Default Value	Description
splk_data_sampling_m	3600	Defines the minimal time in seconds between two iterations of sampling per entity. TrackMe will not attempt to perform sampling for a given entity if the time since the last iteration is lower than this value.
splk_data_sampling_nc	10000	Number of records to be sampled and verified per entity during the data sampling process. Increased value improves event format recognition but requires more processing.
splk_data_sampling_nc	10	Defines the number of records to be stored in the KVstore for inspection purposes at each iteration. This value can be increased if more storage space is available.
splk_data_sampling_re	40000	Character size limit before truncating events when storing sampled records in the KVstore for inspection. This truncation does not impact the model training process, only storage.
splk_data_sampling_pc	98	Minimum percentage of events that must match the major inclusive model. If the main model has less than this percentage of events matching, the entity's state will be impacted.
splk_data_sampling_pc	95	Maximum percentage of events matching an exclusive model that can be accepted. By default, no events matching an exclusive model are accepted, but this value can be increased.
splk_data_sampling_re	3600	Size of the time window for the sampling operation, in seconds, relative to the latest event time known for the entity. This window is used to calculate the earliest time for sampling searches.

7.1.17 splk-outliers-detection

This tab defines various options specific to the Machine Outliers detection features:

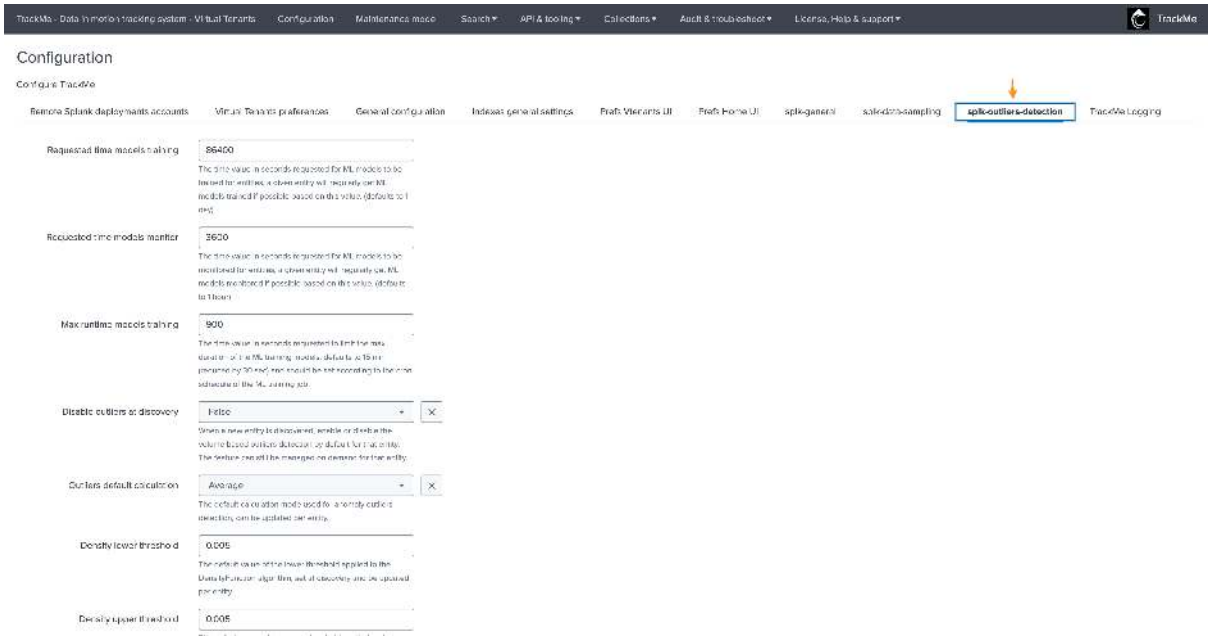


Table 8: splk-outliers-detection Configuration

Field	Label	Default Value	Description
splk_outliers_min_	Min days historical metrics for confidence	7	The minimal number of days of historical metrics required to compute the confidence level of the outliers detection, defaults to 7 days.
splk_outliers_time	Requested time models training	604800	The time value in seconds requested for ML models to be trained for entities; a given entity will regularly get ML models trained if possible based on this value. (defaults to 7 days)
splk_outliers_time	Requested time models monitor	3600	The time value in seconds requested for ML models to be monitored for entities; a given entity will regularly get ML models monitored if possible based on this value. (defaults to 1 hour)
splk_outliers_max	Max runtime models training	900	The time value in seconds requested to limit the max duration of the ML training models, defaults to 15 min (reduced by 30 sec) and should be set according to the cron schedule of the ML training job.
splk_outliers_max	Max time since last training	15	When executing a rendering operation, TrackMe verifies the last time this model was trained. If this time exceeds the value set here, the model will be retrained automatically before rendering. (defaults to 15 days)
splk_outliers_dete	Disable outliers at discovery	0	When a new entity is discovered, enable or disable the volume-based outliers detection by default for that entity. The feature can still be managed on demand for that entity.
splk_outliers_calcu	Outliers default calculation	stdev	The default calculation mode used for anomaly outliers detection, can be updated per entity.
splk_outliers_dens	Density lower threshold	0.005	The default value of the lower threshold applied to the DensityFunction algorithm, set at discovery and can be updated per entity.
splk_outliers_dens	Density upper threshold	0.005	The default value of the upper threshold applied to the DensityFunction algorithm, set at discovery and can be updated per entity.
splk_outliers_alert	Volume lower breached	1	Alert when the lower bound threshold is breached for volume-based KPIs.
splk_outliers_alert	Volume upper breached	0	Alert when the upper bound threshold is breached for volume-based KPIs.
splk_outliers_alert	Latency lower breached	0	Alert when the lower bound threshold is breached for latency-based KPIs.
splk_outliers_alert	Latency upper breached	1	Alert when the upper bound threshold is breached for latency-based KPIs.
splk_outliers_dete	Default period for calculation	-30d	The relative period used by default for outliers calculations, applied during entity discovery and can be updated per entity.
splk_outliers_dete	Default latest time quantifier for calculation	now	The relative time quantifier for the latest time used by default for outliers calculations, applied during entity discovery and can be updated per entity. Defaults to now and can accept Splunk relative time quantifiers such as -1h@h.
splk_outliers_dete	Default outliers	-30d	The default time factor applied for

7.1.18 SLA configuration

This tab defines various options specific to the SLA feature:

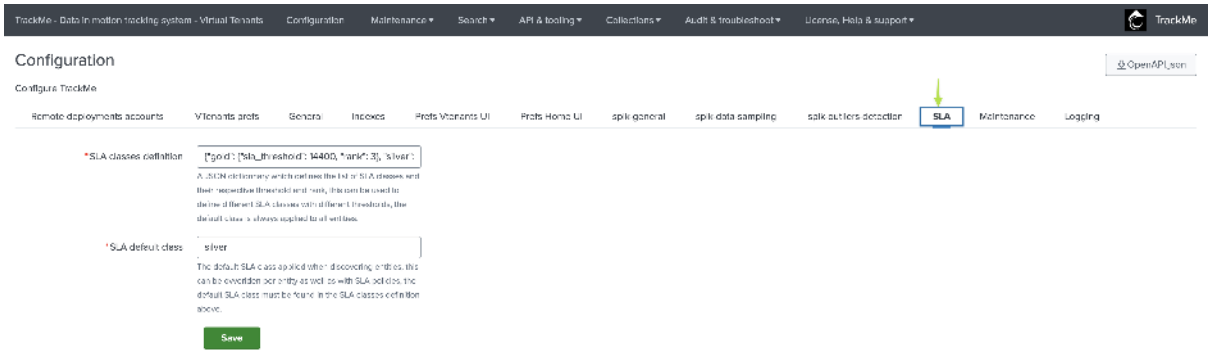


Table 9: SLA

Field	Label	Default Value	Description
sla_classes	SLA classes definition	{“gold”: {“sla_threshold”: 14400, “rank”: 3}, “silver”: {“sla_threshold”: 86400, “rank”: 2}, “platinum”: {“sla_threshold”: 172800, “rank”: 1}}	A JSON dictionary which defines the list of SLA classes and their respective threshold and rank; this can be used to define different SLA classes with different thresholds, the default class is always applied to all entities.
sla_default_class	SLA default class	silver	Defines if SLA exclusions should exclude planned only, unplanned only, or both planned and unplanned events.

7.1.19 Maintenance

This tab defines various options specific to the Maintenance feature:

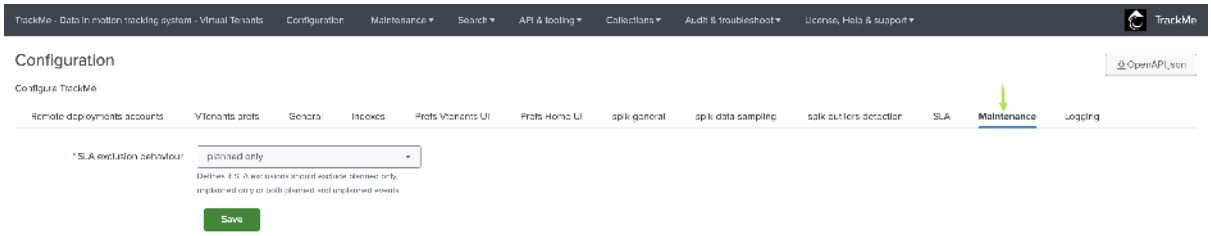
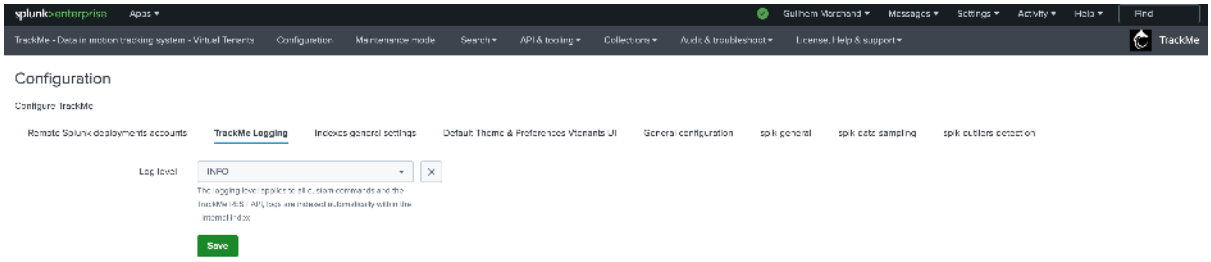


Table 10: SLA

Field	Label	Default Value	Description
maintenance_kdb_exclusion_b	SLA exclusion behaviour	planned	Defines if SLA exclusions should exclude planned only, unplanned only, or both planned and unplanned events.

7.1.20 TrackMe Logging

This tab defines the logging level for TrackMe, all custom commands, REST endpoints, and any other TrackMe components rely on this setting to define the level of logging:



It is not recommended in a Production context to set TrackMe in DEBUG mode in normal circumstances as TrackMe will be extremely chatty in debug.

A typical logging message will look like: (INFO mode in this example)

```
2023-01-10 17:22:04,520 INFO trackmesplkflxparse.py stream 366 tenant_id="flx-demo-dma
↪", context="live", TrackMeSplkFlxParse has terminated successfully, turn debug mode␣
↪on for more details, results_count="2"
```

The logging level is extracted at search time, via props.conf settings, example:

```
# catch all sourcetype
[(?::){0}trackme:custom_commands:*]
EXTRACT-log_level = \d{4}-\d{2}-\d{2}\s\d{2}:\d{2}:\d{2}\,\d*\s(?<log_level>\w*)\s
```

Therefore, you can review errors for instance with the following SPL search which would review both REST API endpoints errors and the custom commands:

```
(index=_internal sourcetype=trackme:rest_api log_level=ERROR) OR (index=_internal␣
↪sourcetype=trackme:custom_commands:* log_level=ERROR)
```

We strongly believe that the truth stands in the logs; therefore, we take great care at making sure logging in TrackMe is giving you the greatest level of quality and reliability!

See the following documentation for more about logging & troubleshooting in TrackMe:

- *Troubleshooting TrackMe*

7.2 TrackMe theme for Tabulator

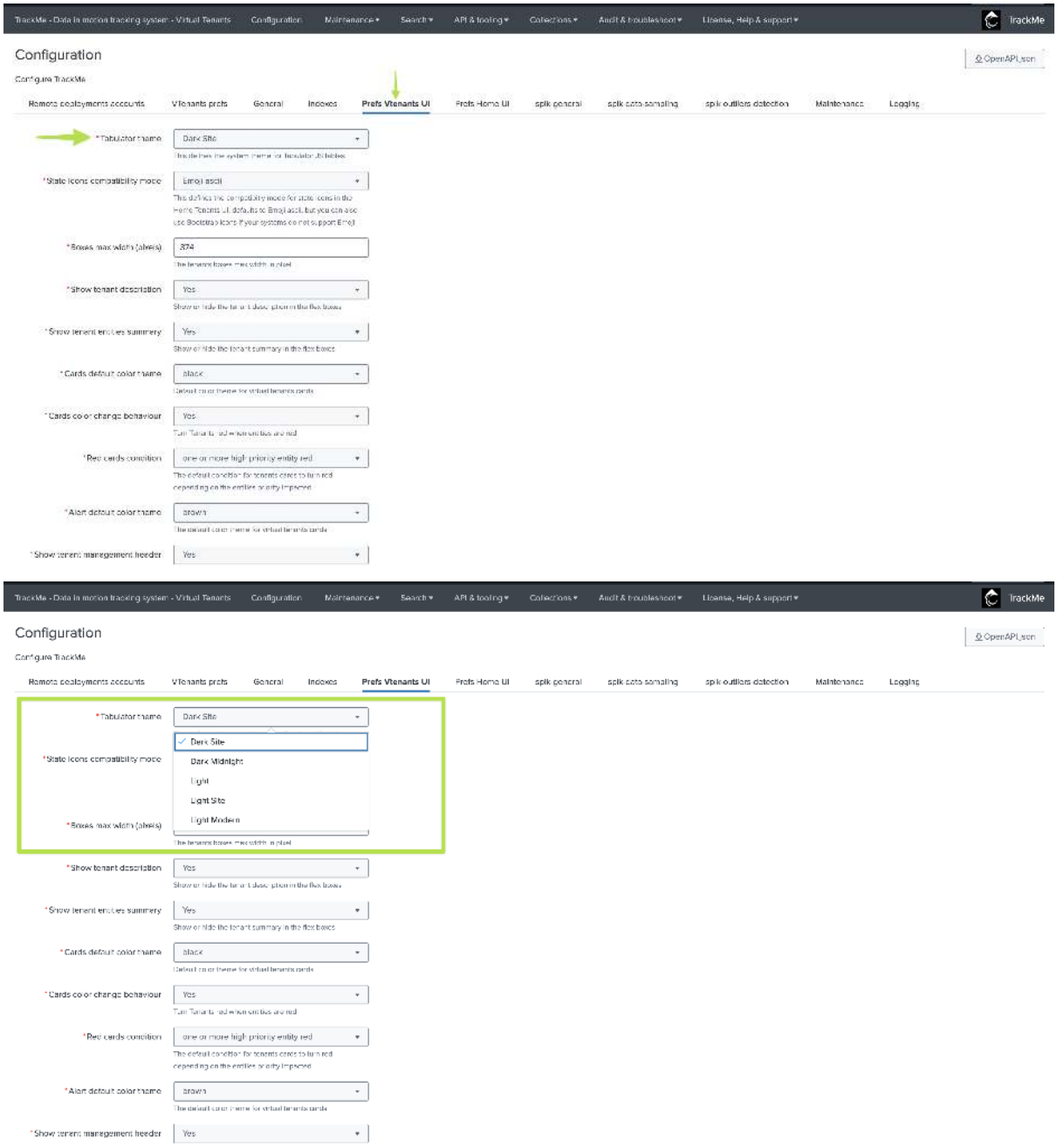
Theme for TrackMe main table component (Tabulator JS)

- TrackMe heavily uses a key component called Tabulator JS.
- This component is used to display data in many forms, allowing various powerful and efficient user interactions in TrackMe.
- Since TrackMe 2.0.87, we have added capabilities to customise the look and feel of the Tabulator component.
- As an administrator, you can choose the **default system level Tabulator theme** for both TrackMe's Virtual Tenant UI and the home tenants UI.
- These preferences can be used on a **per user level**, if set the user level preferences take precedence over the system wide settings.

7.2.1 Configuring the Tabulator theme and Icons theme preferences for Virtual Tenant UI and Home Tenant UI

In TrackMe's configuration UI, you can choose the Tabulator theme:

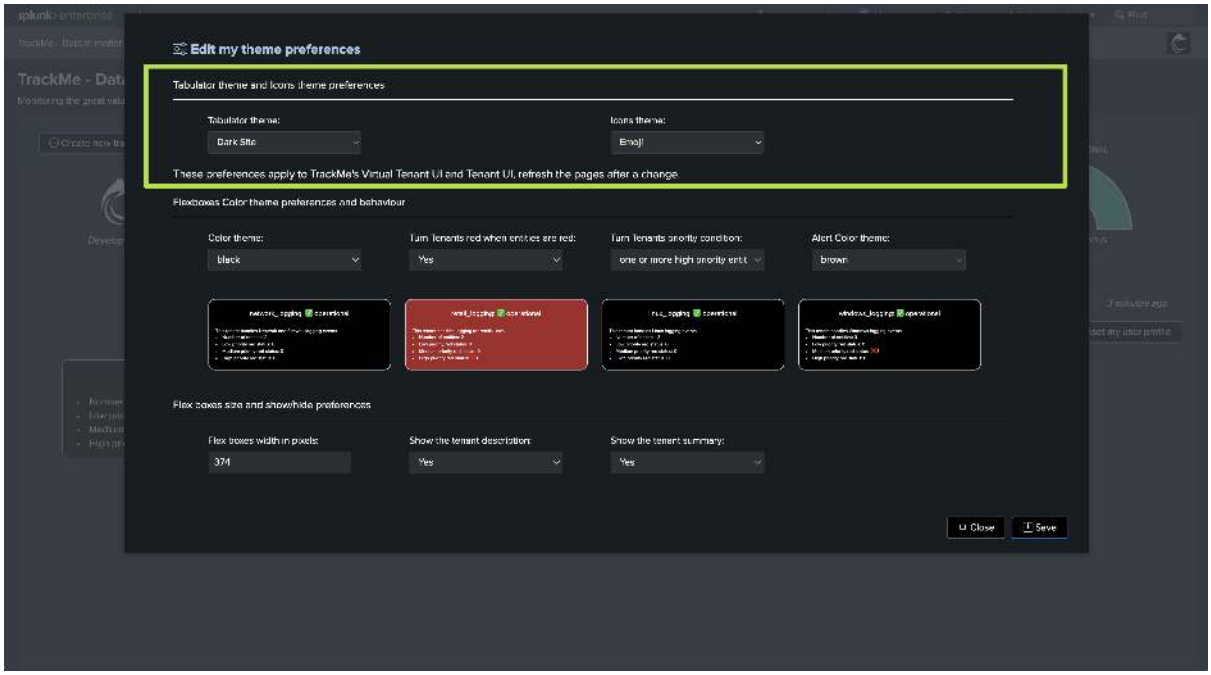
Example for the Virtual Tenant UI:



Note: these are system wide settings, and will be effective for all TrackMe users.

7.2.2 User level configuration for Tabulator theme & Icons theme preferences

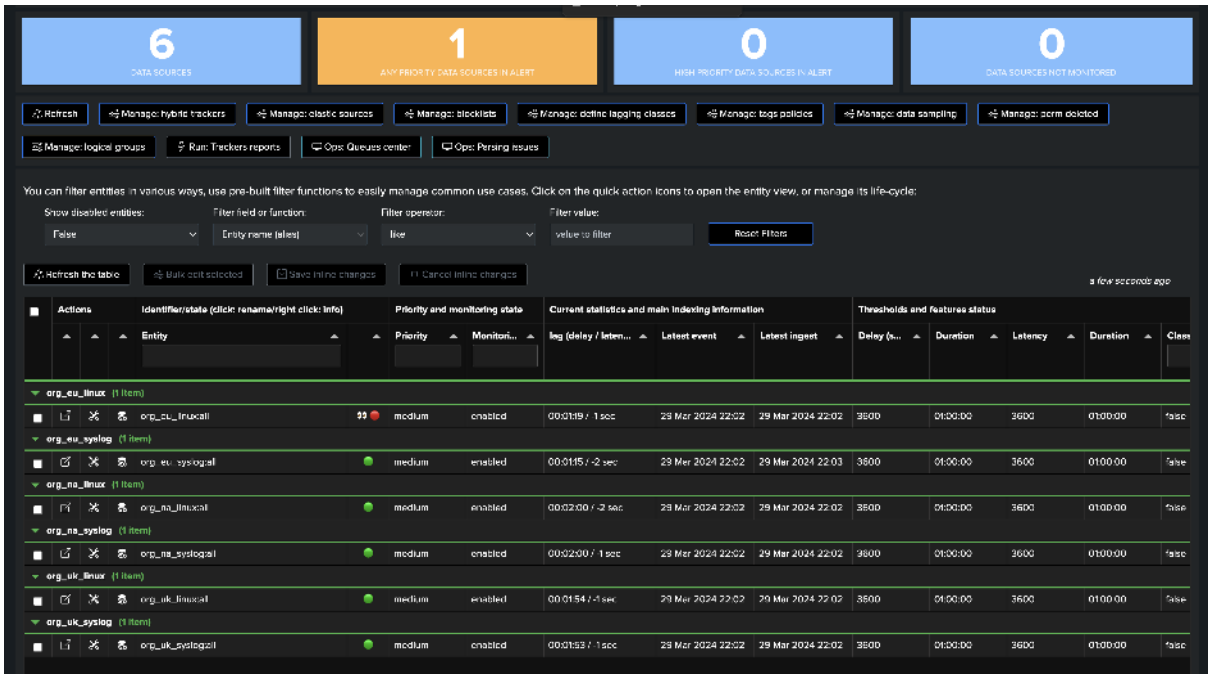
From the TrackMe Virtual Tenant user UI, any user can define its own set of preferences:



If a user has no preferences set, the system wide settings will be used.

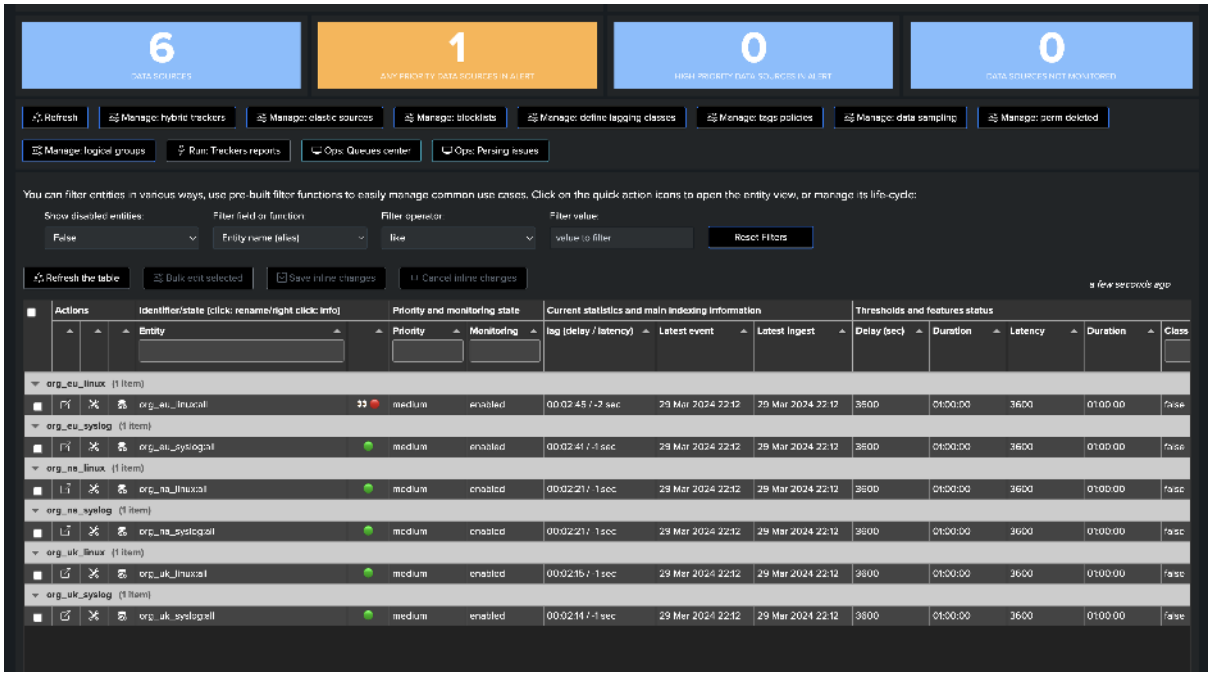
7.2.3 Dark Site theme

This is the default Tabulator theme, starting from TrackMe 2.0.87:



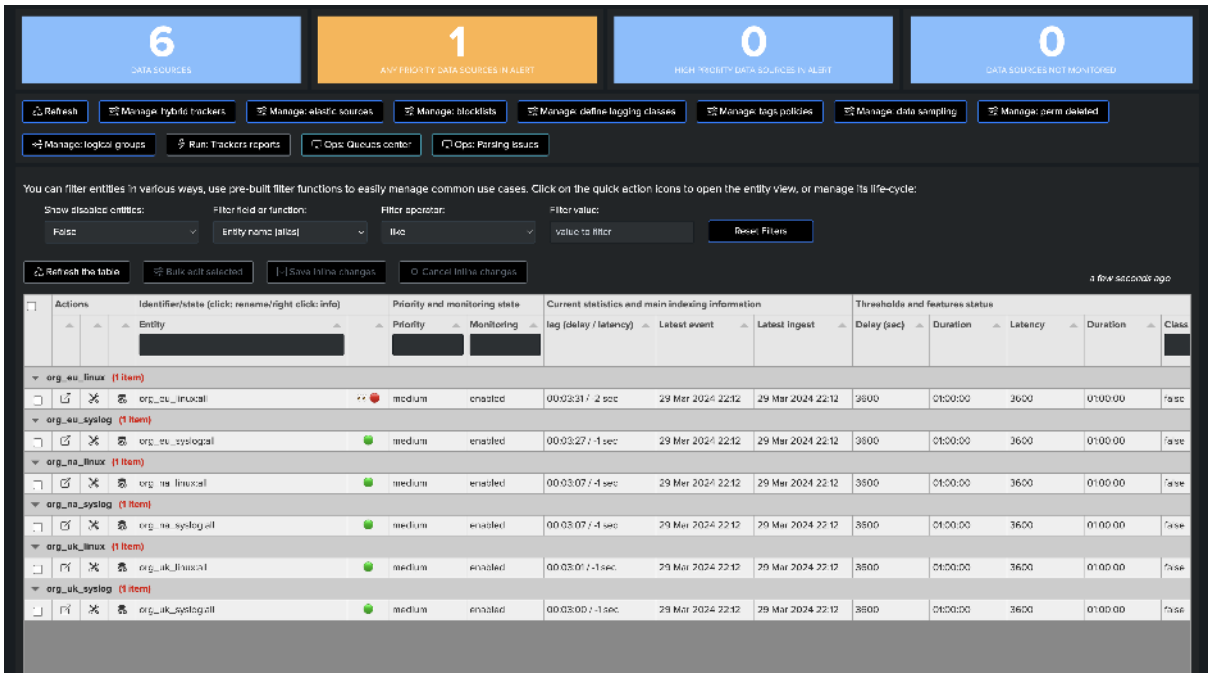
7.2.4 Dark Midnight theme

The Dark Midnight theme was the default Tabulator theme since TrackMe 2.0.0, and before TrackMe 2.0.87:



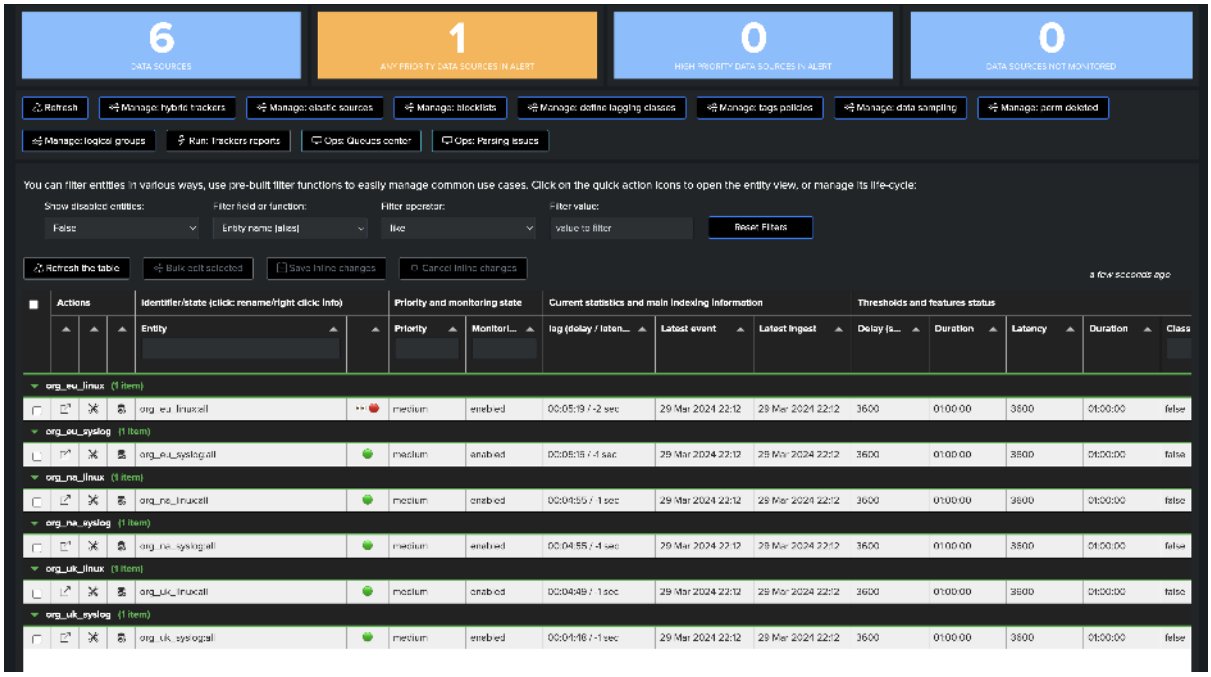
7.2.5 Light theme

A light theme:



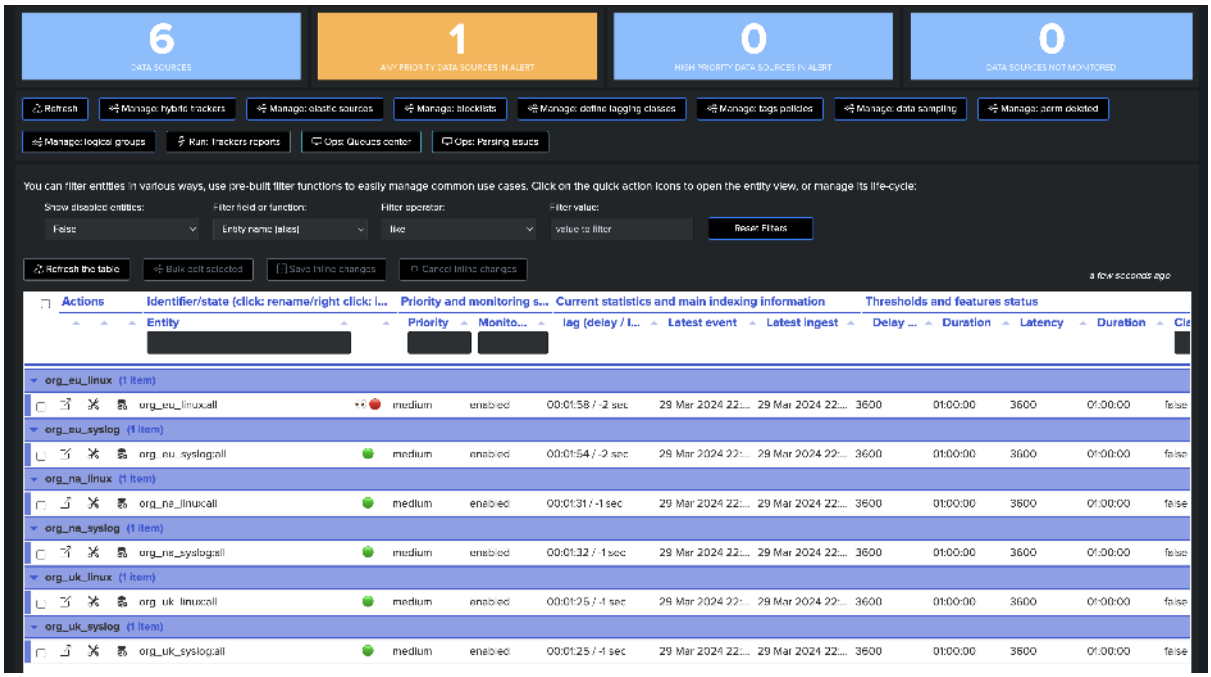
7.2.6 Light Site theme

A light alternative theme:



7.2.7 Light Modern theme

Another light alternative theme:



7.3 Large Scale Environment and Best Practices Configuration Guide

7.3.1 Introduction to High Scale in TrackMe

This documentation aims to provide best practice configuration steps for TrackMe, particularly in large-scale environments.

TrackMe is highly configurable, flexible, and designed for performance and efficiency at scale.

It scales from daily ingests in TB to dozens and much more, with limited and optimized computing costs when respecting good design practices.

However, adhering to best practices and considering specific design aspects are crucial for optimal results.

7.3.2 Scaling Concepts with TrackMe

TrackMe scaling concepts depend on various aspects of the solution, with some key points highlighted below.

TrackMe Virtual Tenants

Virtual Tenants are a core concept in TrackMe V2. In summary:

- Virtual Tenants act as virtual instances of TrackMe within the application.
- By using Virtual Tenants, you can define the scope and design of your TrackMe implementation.
- A Virtual Tenant orchestrates the lifecycle of all TrackMe knowledge objects, from KVstore collections to TrackMe trackers.
- Virtual Tenants are created on-demand by TrackMe administrators and can be disabled or deleted later.
- This feature allows scoping of Splunk data and entities created by TrackMe, using technical or functional concepts tailored to your preferences and context.
- Virtual Tenants support Role-Based Access Control (RBAC) and cater to a wide range of functional and business requirements.

Virtual Tenants offer capabilities beyond the "MSP" (Managed Service Provider) requirements, making TrackMe a unique Splunk application.

TrackMe Trackers

TrackMe implements a concept of Trackers, these are scheduled backend jobs created:

- There are different types of Trackers, depending on the components
- For Feeds tracking, for instance, you would create multiple Hybrid trackers to address the required data scope, with no overlap between trackers

Dedicated TrackMe Search Head Tiers

At very large scale, it is obviously beneficial to consider a dedicated Search Head tier for TrackMe, either standalone or dedicated:

- Keep in mind that TrackMe can use its remote search capabilities; therefore, it technically can deal with any remote deployment, on-premise or Cloud
- On-Premise Splunk customers at large scale should consider dedicating a Search Head tier for TrackMe
- Cloud customers will leverage the ad-hoc Search Head tier; some customers can also have multiple ad-hoc Search Head tiers and therefore dedicate a specific Search Head tier to TrackMe

In any case, keep in mind that TrackMe should never be actively tracking in the same Search Head tier as a Splunk Premium application, and that we do **not** support this scenario (Enterprise Security, ITSI).

7.3.3 Requirements for High Scale

TrackMe Splunk Service Account

It is a best practice to use a service account; this allows, for instance, to easily identify related search workloads, investigate the costs related to TrackMe, and so forth.

Using a service account rather than the default `admin` user when assigning searches also facilitates the implementation of Splunk Workload Management (WLM).

TrackMe implements a least privileges approach, consult:

- *Configuration*
- *Role Based Access Control and ownership*

TrackMe Remote Accounts

With its unique capabilities, TrackMe allows transparent execution of searches against remote Splunk deployments or standalone instances, from Search Head Clusters to Heavy Forwarders, or utility nodes such as Deployment Servers, License Managers, etc.

Depending on the components and your context, you may need as a prerequisite to get service accounts created on remote systems, a bearer token to be created, and remote accounts to be configured in TrackMe.

Consult:

- *Splunk Remote Deployments (splunkremotesearch)*

7.3.4 Feeds Tracking (splk-feeds components family)

Feeds tracking is the original core concept of TrackMe, this is covered by 3 components:

- `splk-dsm` which is the main and most valuable component; we recommend focusing on this component first
- `splk-dhm` which tracks data by the event host; this is also a valuable component which can however be more expensive from a compute costs perspective, and requires more maturity
- `splk-mhm` which tracks metrics from the metric endpoint perspective; this component is more specific to IT-Ops related use cases

Hint

Recommendations

- Focus first on the Data Source tracking (splk-dsm)
- Be specific and address in priority the most critical indexes in Splunk
- For large scale environments, we recommend some preparation work to be performed; the indexes naming convention should be known and documented, as well as top priority indexes and perimeters
- Create Tenants according to your needs taking into account perimeters, teams and permissions, etc.
- Create empty tenant and proceed to the Hybrid Trackers configuration manually
- Create concurrent Hybrid Trackers to address specific scope of your environment; it is **more cost efficient** to have multiple concurrent Hybrid Trackers addressing restricted scope of indexes than just a few dealing with a huge amount of data

Create an Empty Tenant

Open the tenant wizard creation, and create the first tenant:

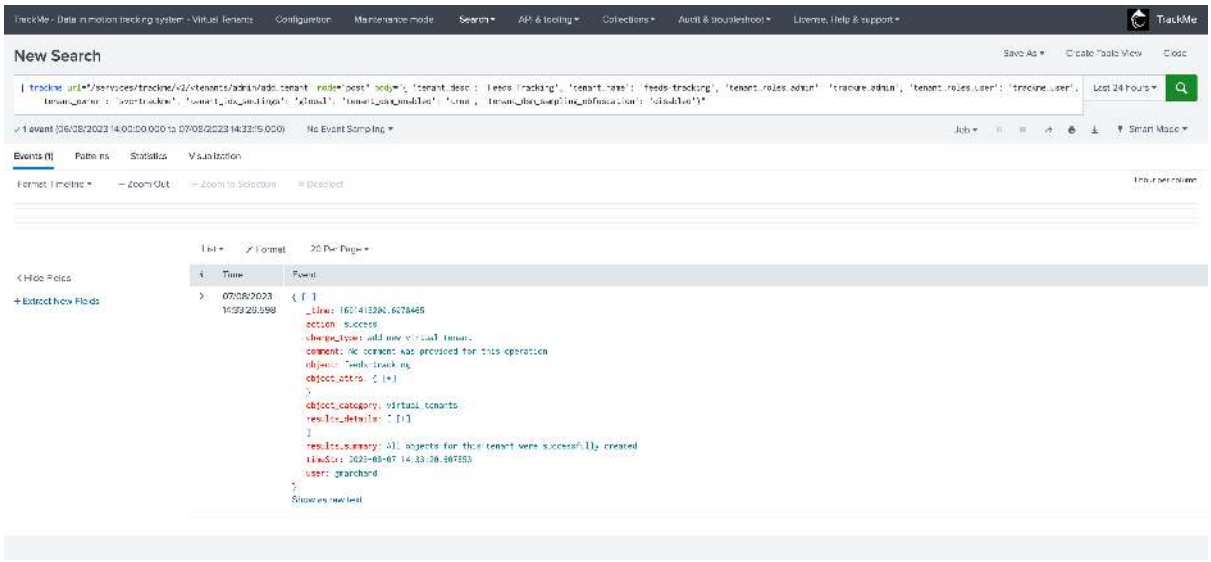
- Only enable the Data Source tracking component (splk-dsm)
- Do not create Hybrid trackers at the phase of the Virtual Tenant creation; instead, create trackers progressively processing benchmarks in the same time

You can create a brand new virgin tenant in a single line of SPL:

Replace the name of the tenant, the service account and roles as needed

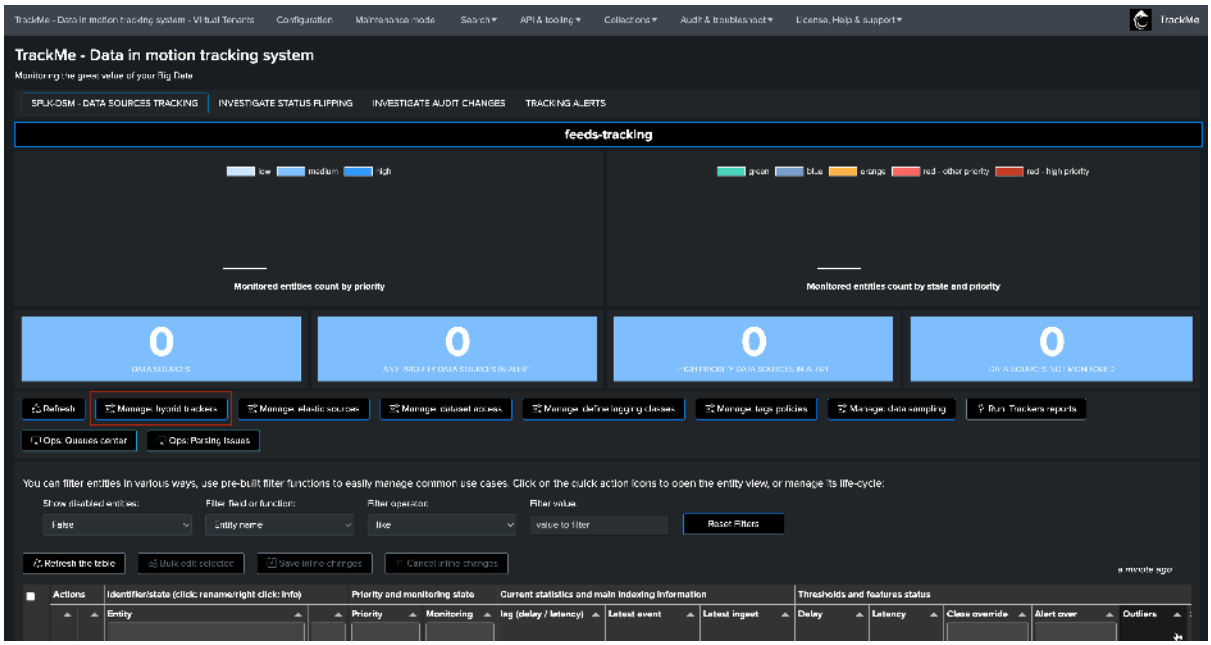
```
| trackme url="/services/trackme/v2/vtenants/admin/add_tenant" mode="post" body="{
↪'tenant_desc': 'Feeds Tracking', 'tenant_name': 'feeds-tracking', 'tenant_roles_
↪admin': 'trackme_admin', 'tenant_roles_user': 'trackme_user', 'tenant_owner': 'svc-
↪trackme', 'tenant_idx_settings': 'global', 'tenant_dsm_enabled': 'true', 'tenant_
↪dsm_sampling_obfuscation': 'disabled'}"
```

Notes: this creates a brand new tenant only with the splk-dsm component; components can be later on added and removed from existing tenants without having to delete and re-create the tenant.

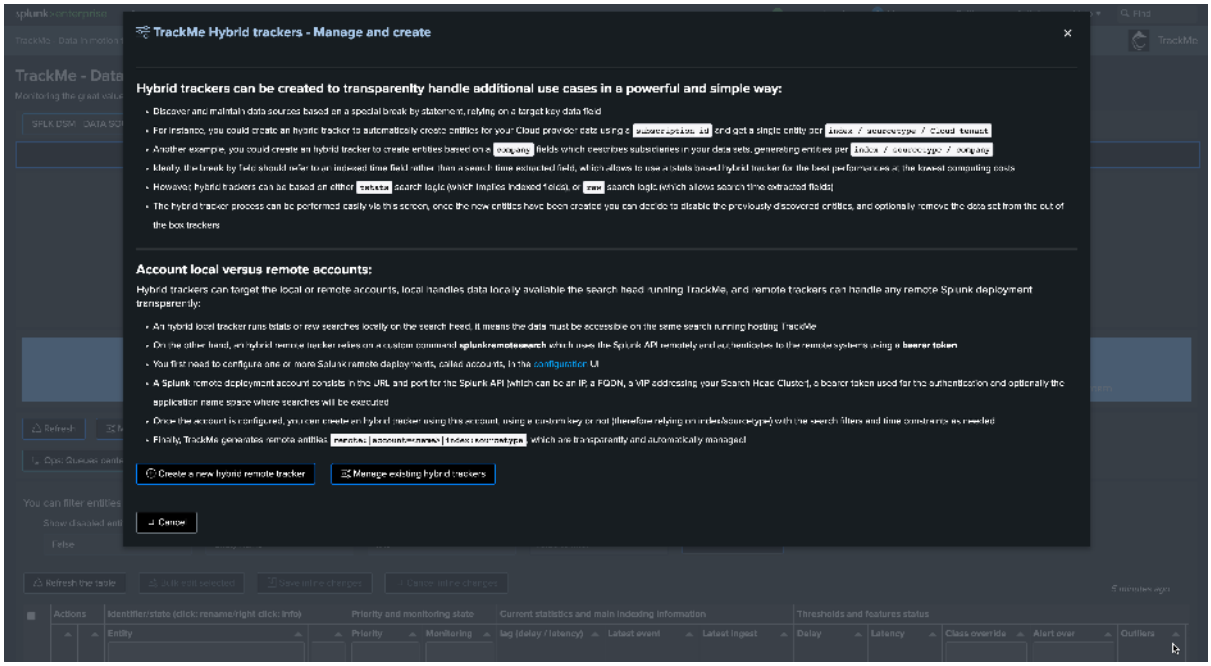


Enter the new tenant

Once the new Virtual Tenant is created, enter the tenant and click on Manage Hybrid Trackers:



Then click on Create new hybrid trackers:



This is where Hybrid Trackers will be defined, tested, benchmarked and finally created.

Open a Search tab in parallel

Have a search tab handy, and let's run the following simple Splunk search:

indexes convention

- In large scale environments, it is a best and right practice to have strong naming conventions
- If for instance, your goal is to track security related data, perhaps all these indexes start by `sec_` or `siem_`
- Then, make sure to include these concepts in the searches thereafter

```
| tstats count where index=* earliest=-7d latest=now by index
| eventstats sum(count) as total_count
| eval percent=round(count/total_count*100, 2)
| fields - total_count
| sort - limit=0 percent
```

The point of this basic search is to identify the volume of events in each Splunk index, then calculate the percentage against the total number of events ingested in the past 7 days.

This in addition with the knowledge of your context and requirements should drive the creation of Hybrid Trackers.

For instance, let's assume that majority of events is represented by

- Firewall related data (Palo Alto, etc)
- System logs related data (Windows eventlogs, etc)
- Cloud security logs (such as AWS, GCP, etc)
- Anything else

In a large scale environment, a likely scenario could be create 4 hybrid trackers, this could for instance results in basically 4 indexes constraints:

Table 11: Hybrid Tracker roles and indexes constraints

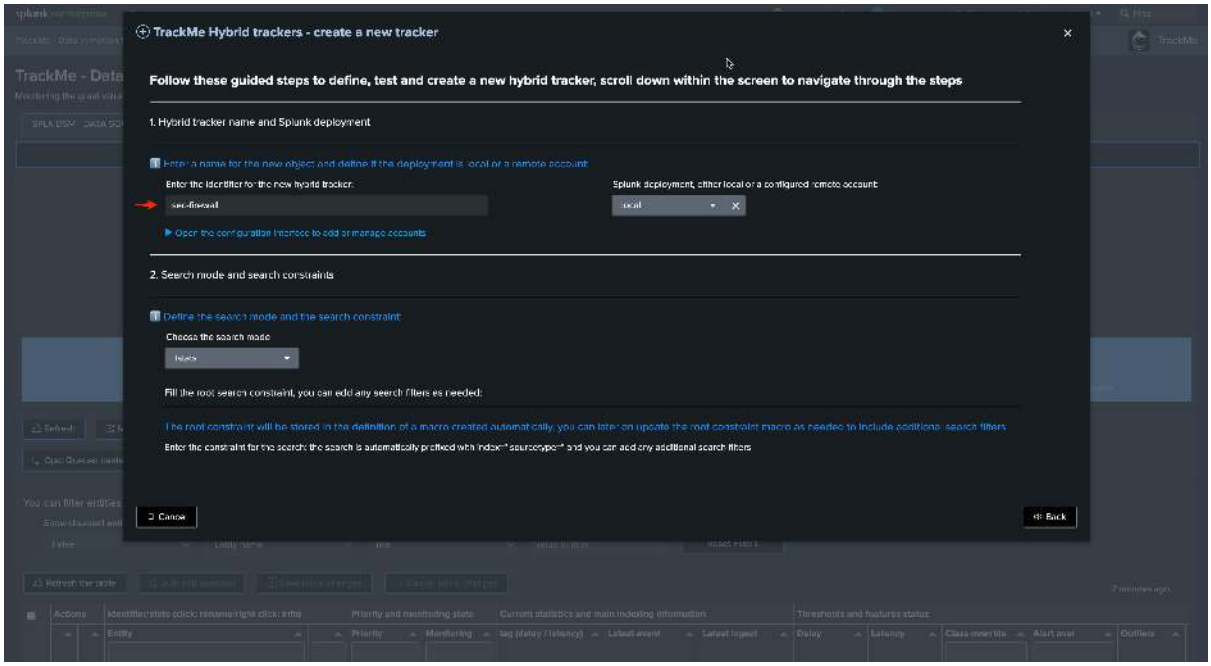
Role	Indexes constraint
Firewall	`(index=sec_firewall_*)`
System logs	`(index=sec_oswin_* OR index=sec_osnix_*)`
Cloud	`(index=sec_cloud_*)`
Other data	`index=sec_* NOT (index=sec_firewall_* OR index=sec_oswin_* OR index=sec_osnix_* OR index=sec_cloud_*)`

This might well be a first shot that you can review and update while you are running into the configuration, this however provides a clear and easy picture of what to do.

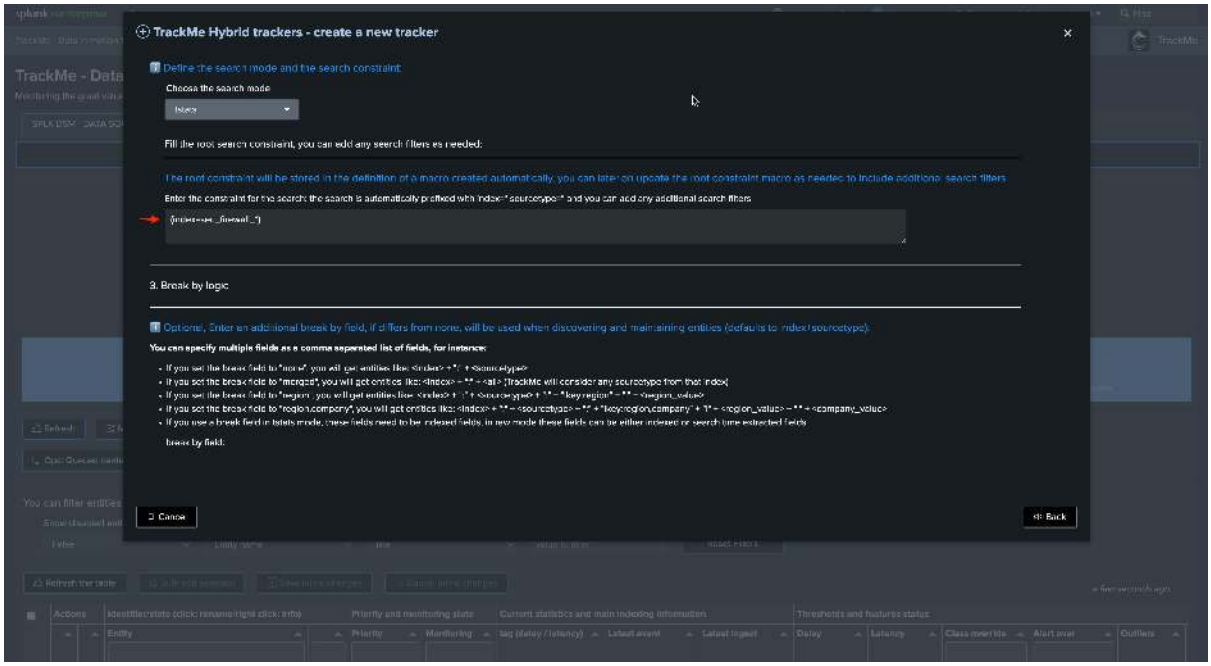
Define, test, benchmark and validate Trackers

Back in the Hybrid Tracker creation user wizard, let's define our first Hybrid Tracker, test and benchmark:

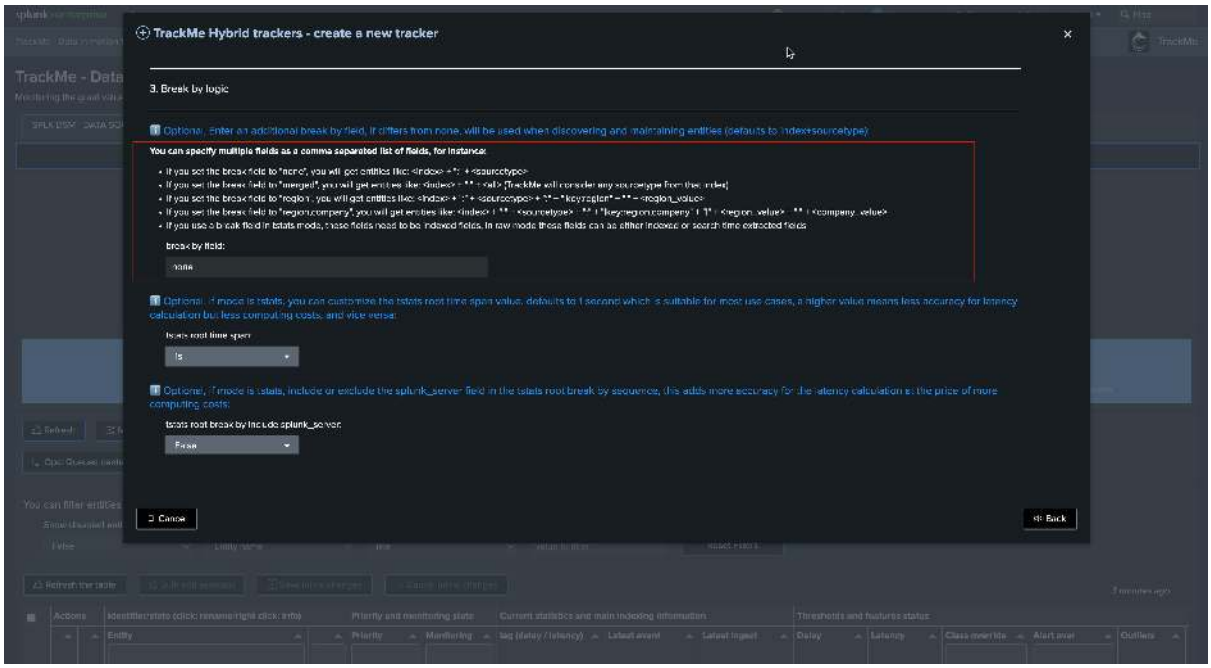
Provide a meaningful name, this will ease the management over time:



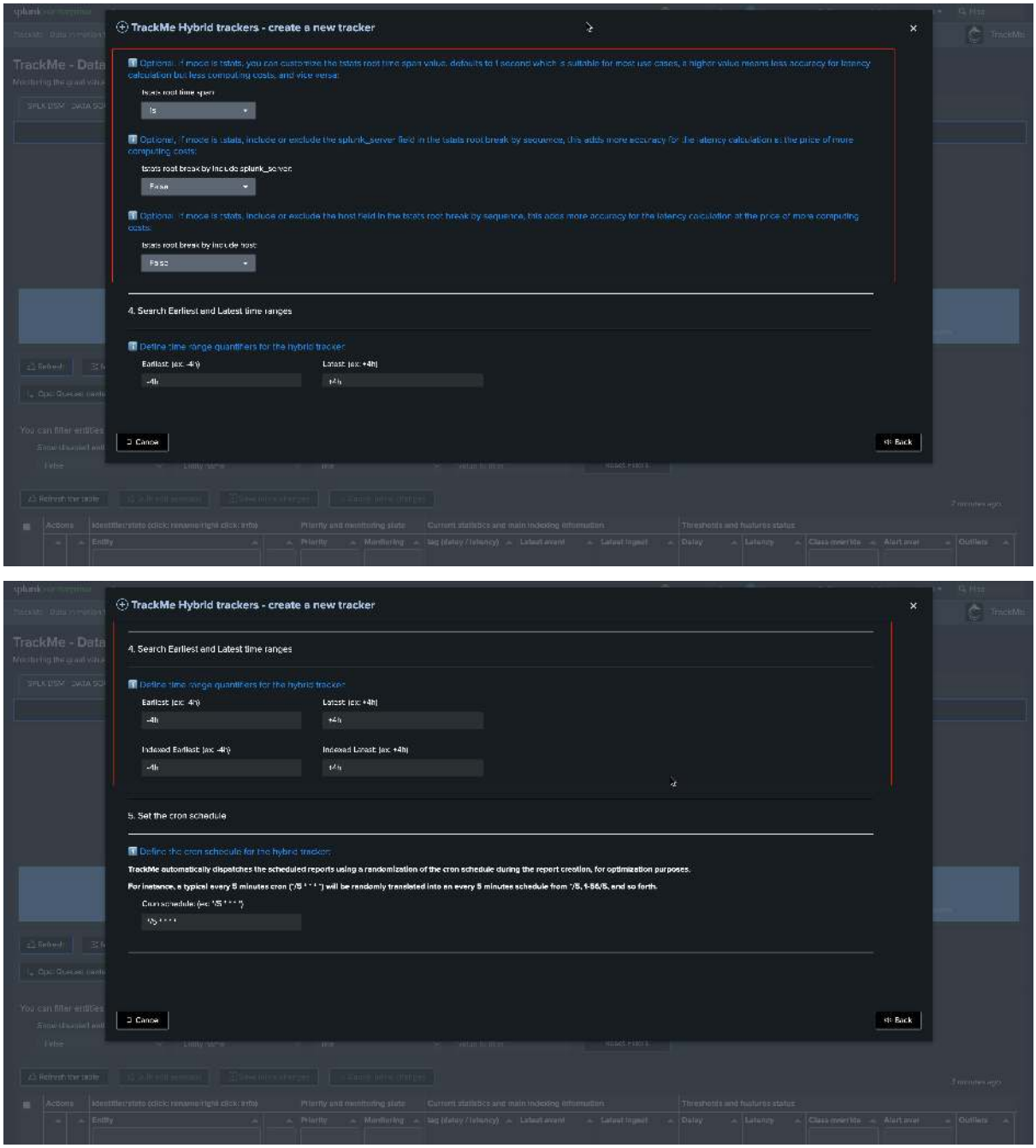
*Replace the entire search constraint, TrackMe will store this into a macro associated with the Hybrid Tracker which can be updated as needed later on:**



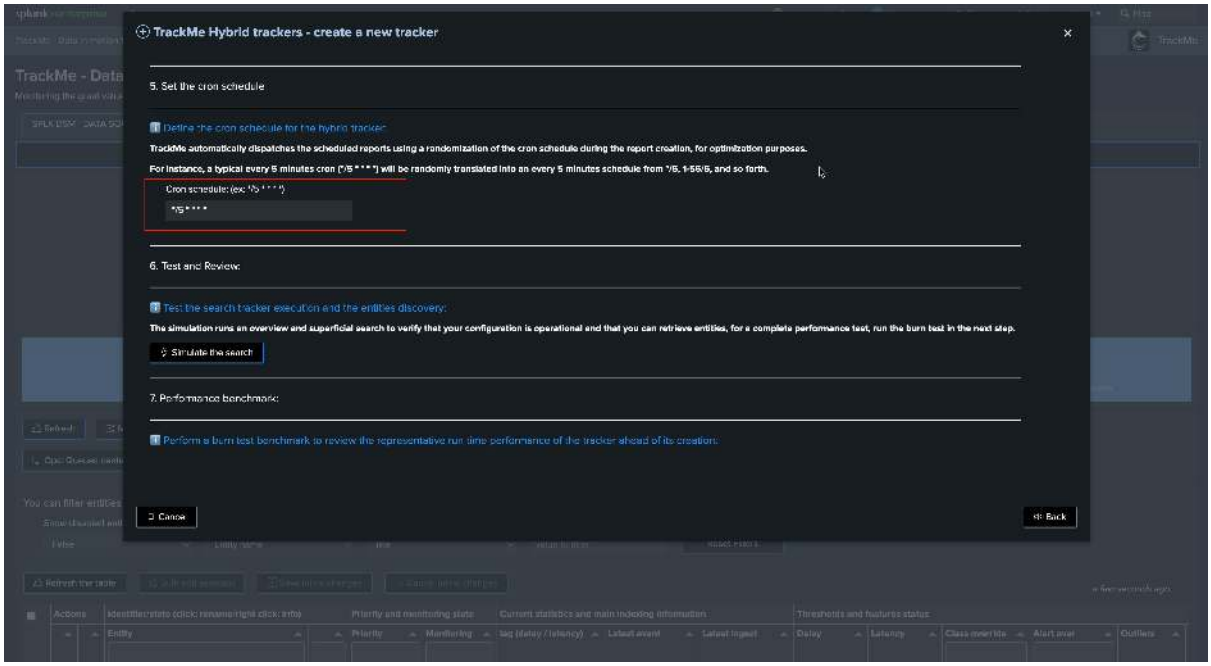
Review the break by statement, this all depends on your requirements, the default means index + ":" + sourcetype:



Review additional break by, earliest and latest time quantifiers, for high scale TrackMe licensed customers, these are generally the best cost efficient parameters, users using the Free Limited licence are limited to two Hybrid Trackers and can restrict these furthers to reduce the run time of the trackers:



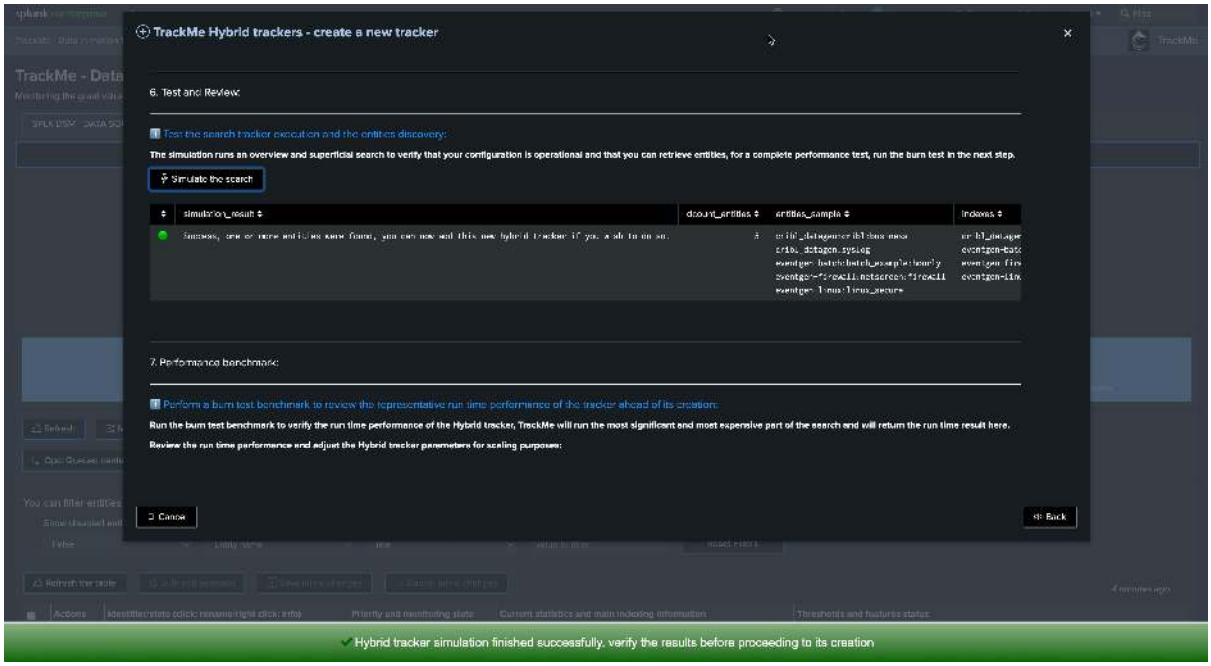
The cron by default is set to every 5 minutes, this is the recommended configuration also for high scale environments as long as the Hybrid Trackers are properly designed:



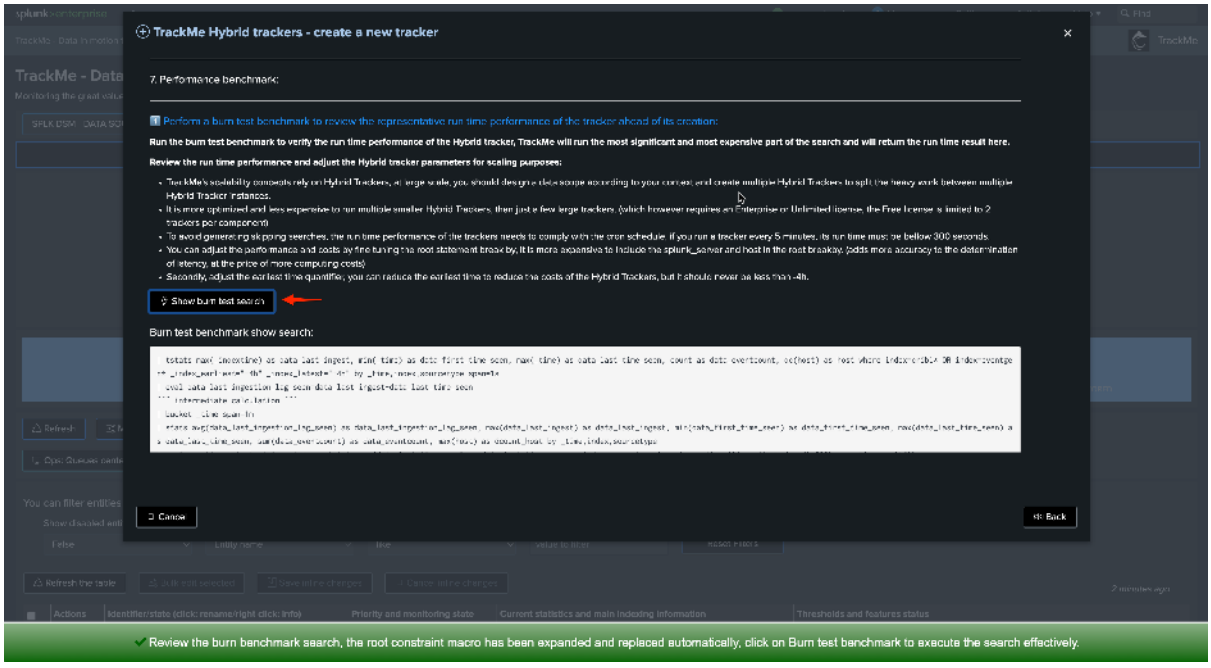
Notes:

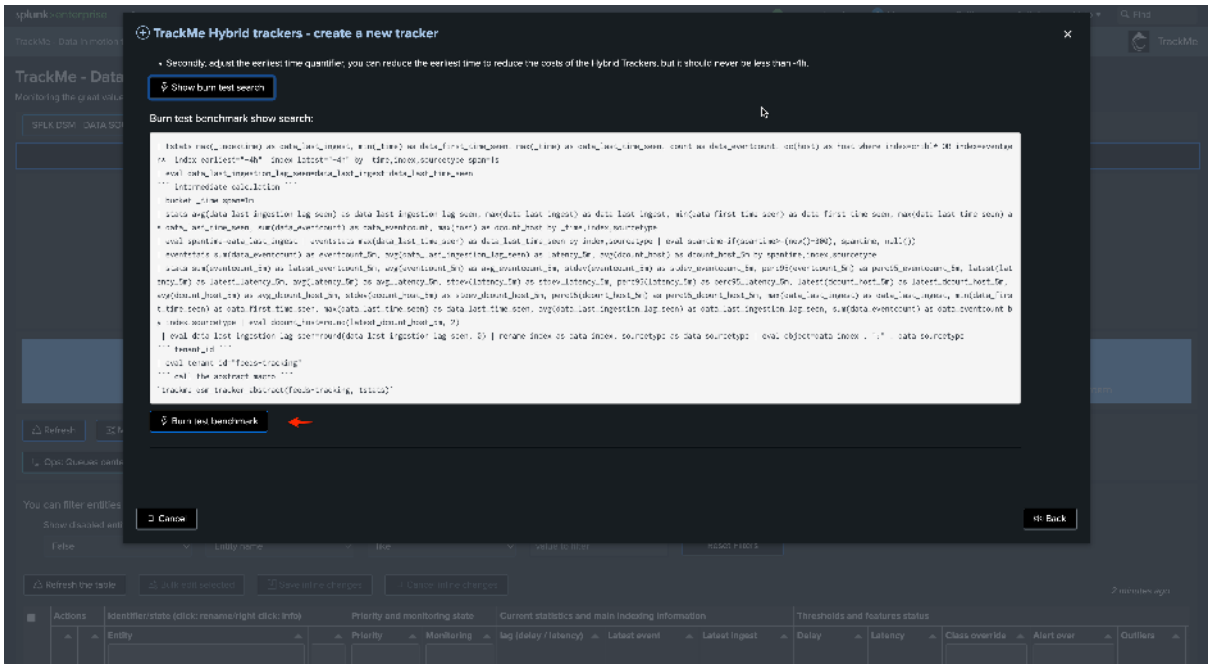
- A crontab of 5 minutes means we have 300 seconds for the tracker to be executed before starting to generate skipping searches
- Skipping searches is the indicator that the configuration is not optimal and that the Hybrid Tracker has too much to be done for a single search instance

Finally, we can test the search, this test only returns a subset of entities found and gives us a first indication of the performance as well as the content that will be retrieved:

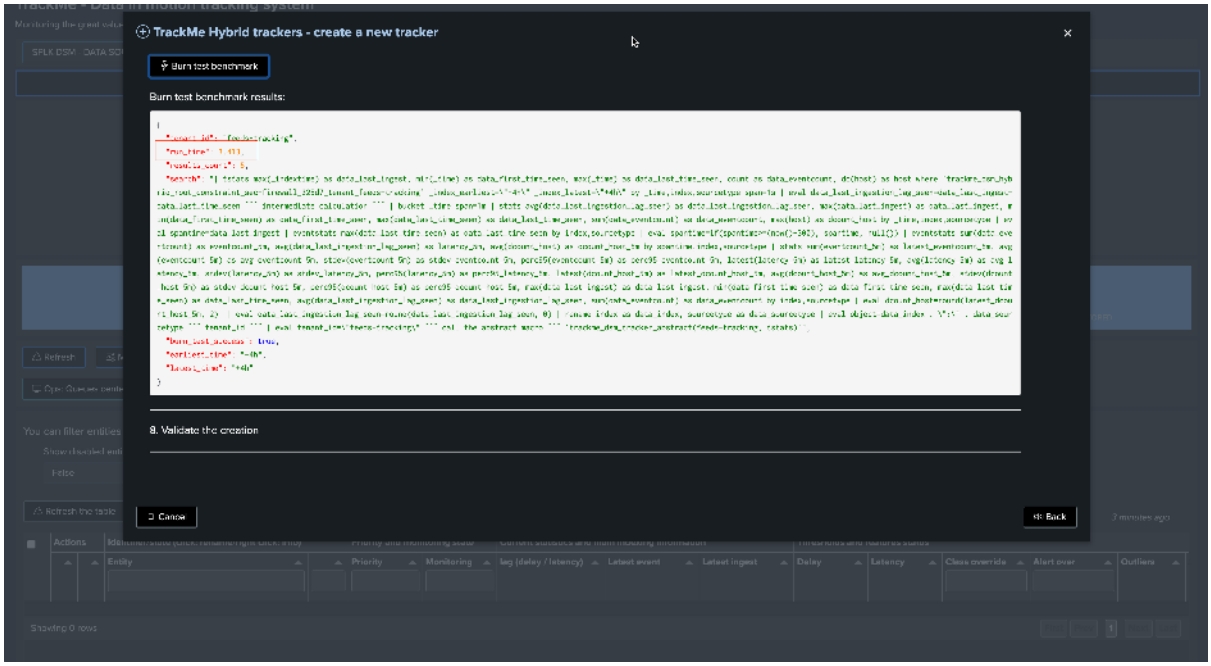


Now, let's return the benchmark search and run the benchmark test, our search needs to complete in well less than 300 seconds according to our cron time, ideally max approximately 200 seconds:





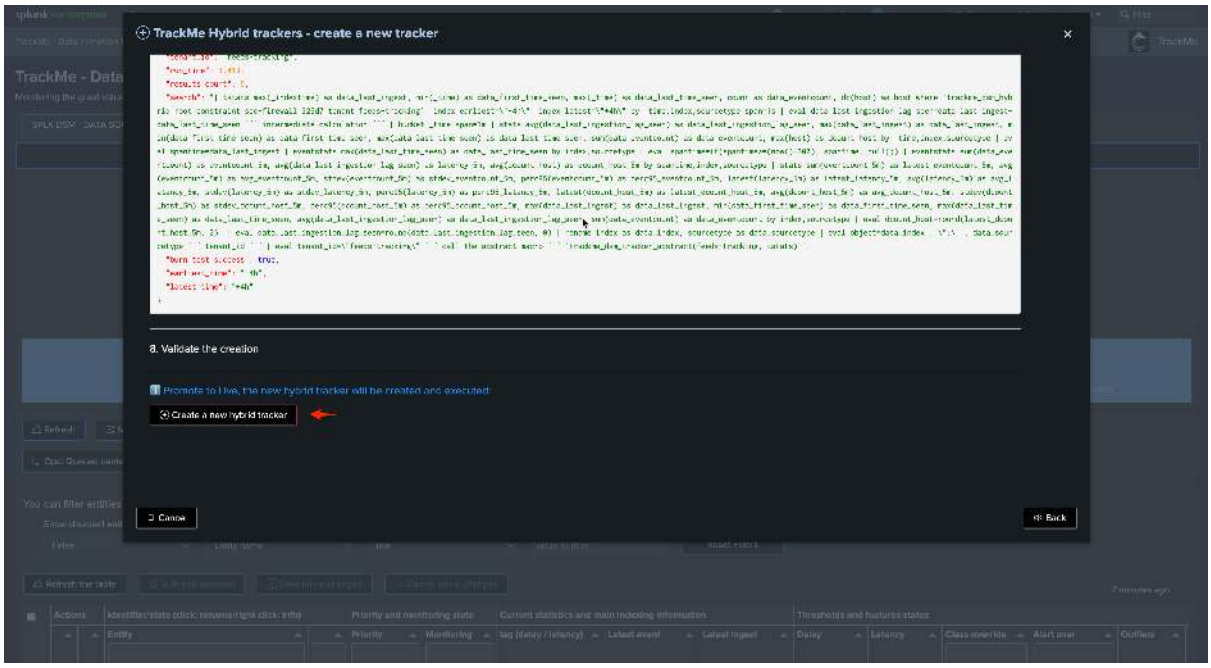
*At this stage, if the search run time super fast, this means that the index scope can safety be increased so you can reduce the total number of Hybrid Trackers, but remember that is much more cost efficient to run concurrent smaller Hybrid trackers than just a few very large:**



Notes:

- In some circumstances and in Splunk Cloud, the benchmark search can reach the Nginx timeout (which is picky picky)
- In that case, you can simply take the benchmark search out and run it in in a search with associated earliest and latest quantifiers
- The Benchmark search represents the most expensive part of the search logic that TrackMe will orchestrate

Once we are happy with the design and performance, we can validate the creation of the Hybrid Tracker:

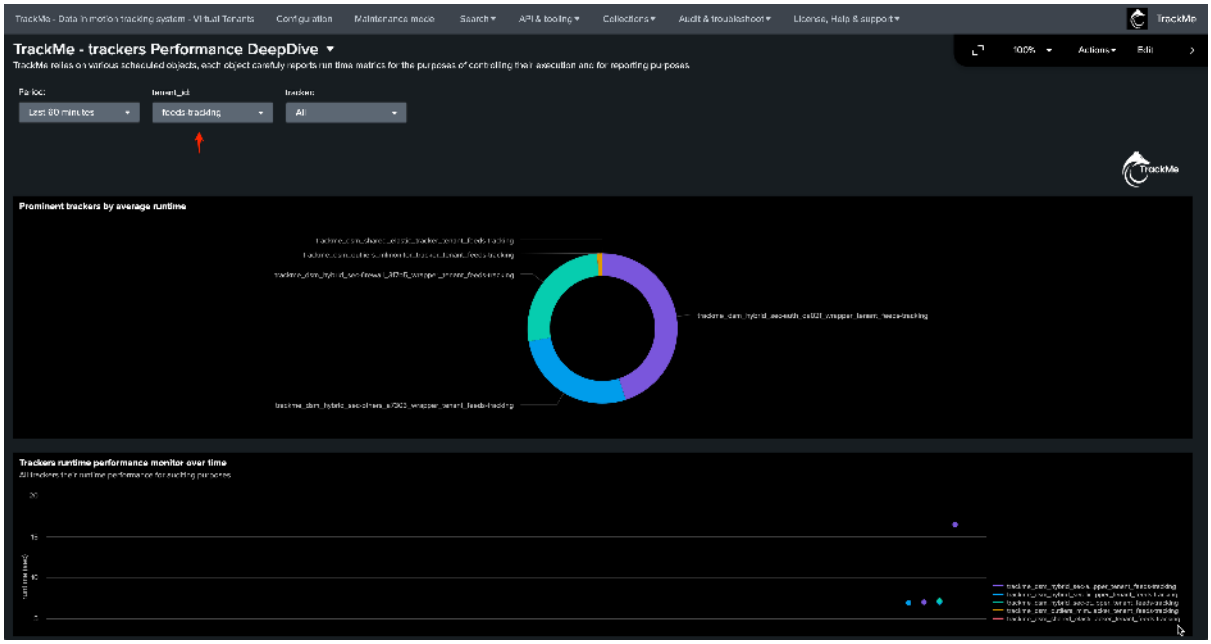


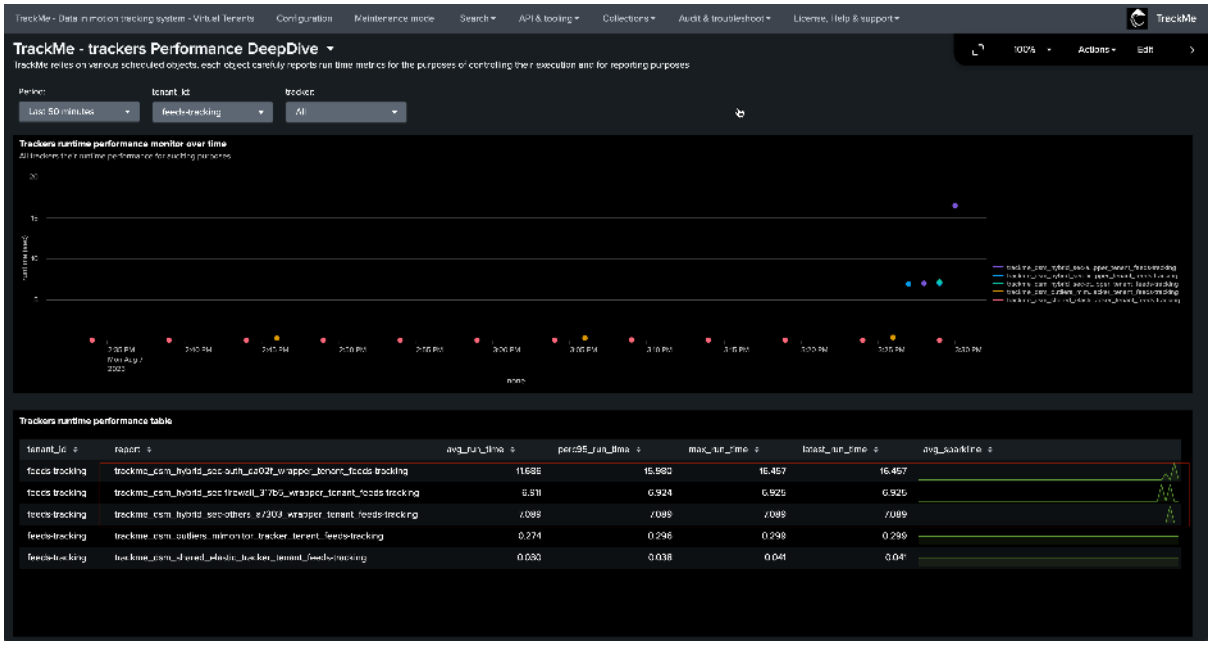
Review Hybrid Trackers Performance

After all Hybrid Trackers were created, and after a few executions, consult the Performance deepdive dashboard: (menu Audit & Troubleshoot)

Types of trackers

- TrackMe has various types of trackers depending on the components and their mission
- Not all share the same type of schedule; some trackers like the ML related trackers are designed to run over long periods of time on a less frequent schedule, dealing with TrackMe own metrics and automatically stopping as needed
- You want to focus on the Hybrid Trackers





7.3.5 Other components and additional considerations

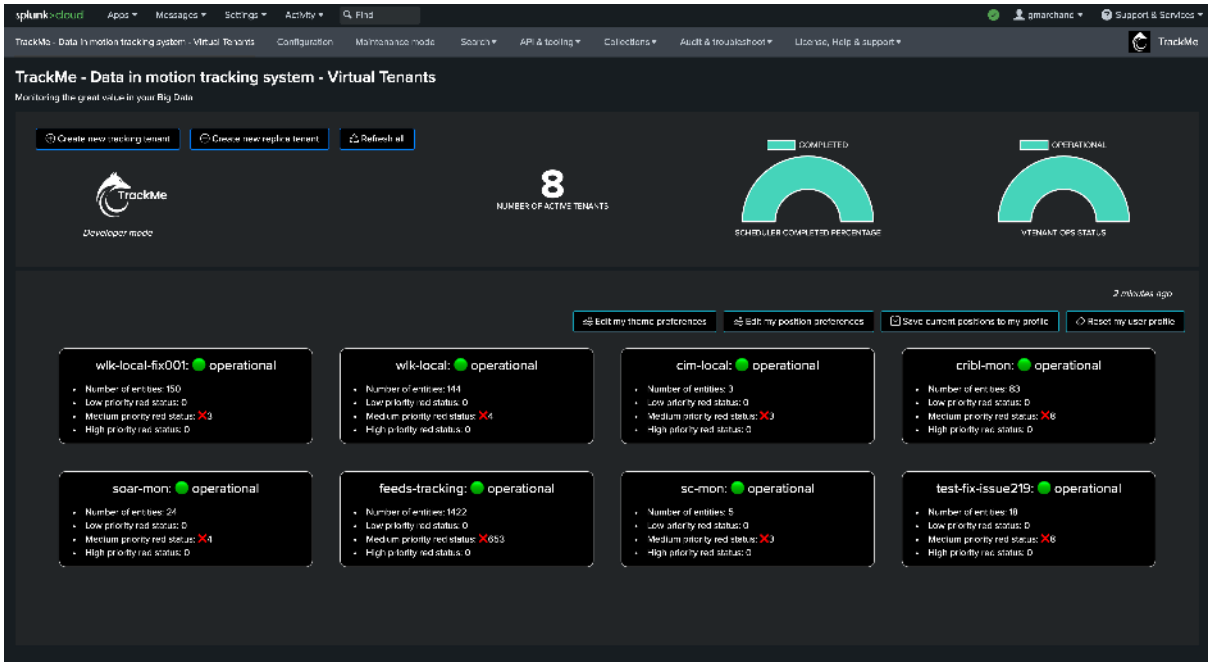
General Tracker Design

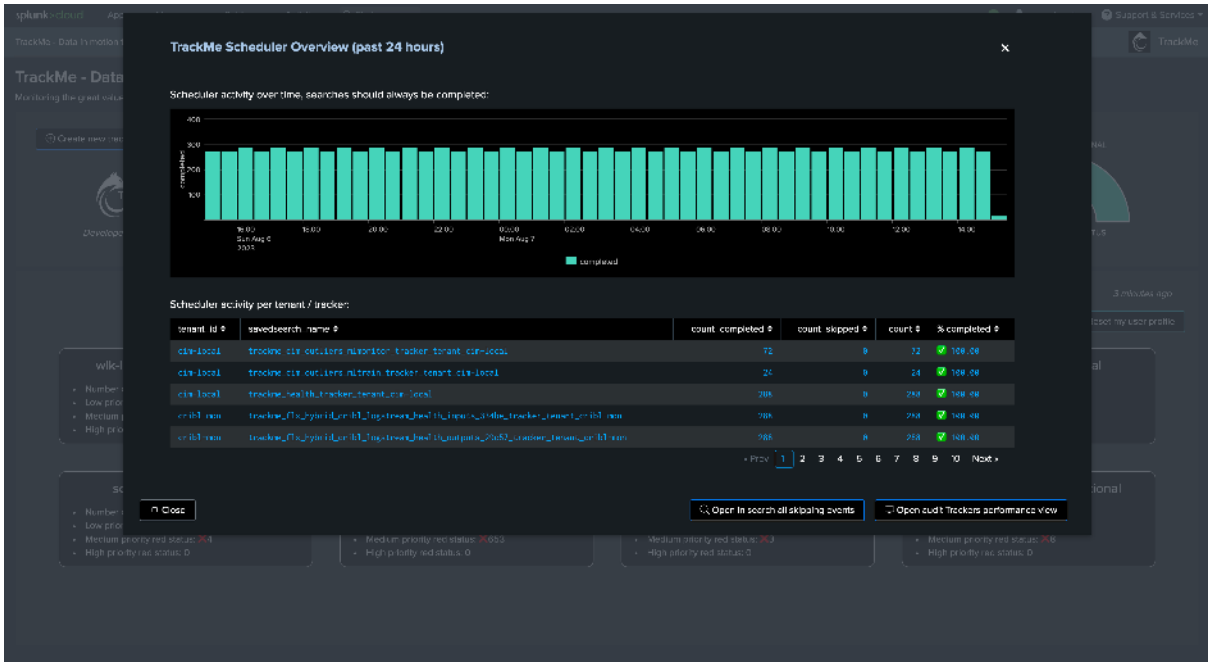
As a general rule, all TrackMe components and underneath associated trackers obey the same design:

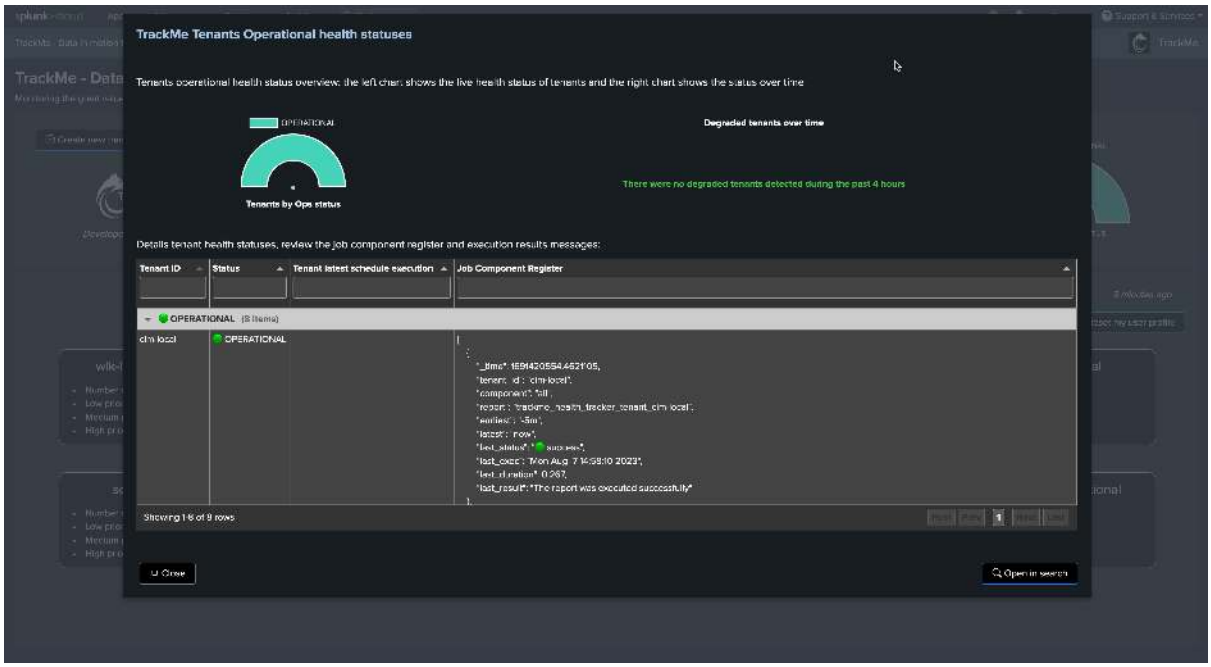
- Trackers should complete in a runtime that is consistent with their cron schedule
- Trackers should not overlap; this means that for instance with feeds tracking, trackers should look at specific set of indexes but unless using a different break by, there are no reasons why these would share the same indexes
- Trackers systematically report their runtime performance and report failures to the component register
- Trackers are component specific; the activity of feeds tracking is not the same as Workload or Flex Trackers, but the philosophy remains similar

Review and Monitor Virtual Tenants Activity

TrackMe carefully reports any issue affecting the Virtual Tenants and trackers; for instance, concurrent execution being reached on the instance would lead to failures easily visible:

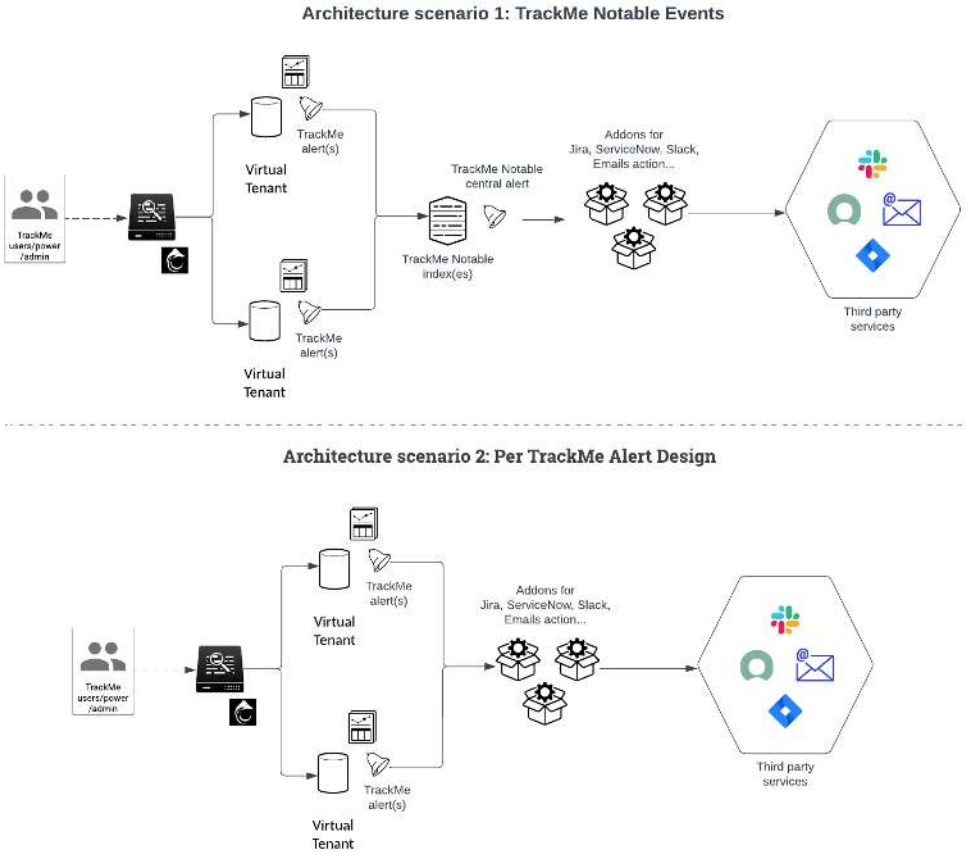






7.3.6 Alerting Architecture in TrackMe

We recommend defining an Alerting Architecture based on TrackMe Notable events which provides several advantages, especially at large scale:



However, both options are valid design options, which can be implemented in TrackMe in a straightforward and flexible way.

See *Alerting Architecture & Third-Party Integration*

7.4 Creating Virtual Tenants

7.4.1 What is a TrackMe Virtual Tenant

About TrackMe Virtual Tenants

- In TrackMe, Virtual Tenants are basically similar to a Virtual instance of TrackMe which handles the application life cycle from A to Z.
- A virtual tenant is an independent application space that can be dedicated according to your needs, addressing any of your requirements in terms of scoping and segmentation.
- When you access TrackMe after the initial deployment, the application comes with no Virtual Tenants created yet.

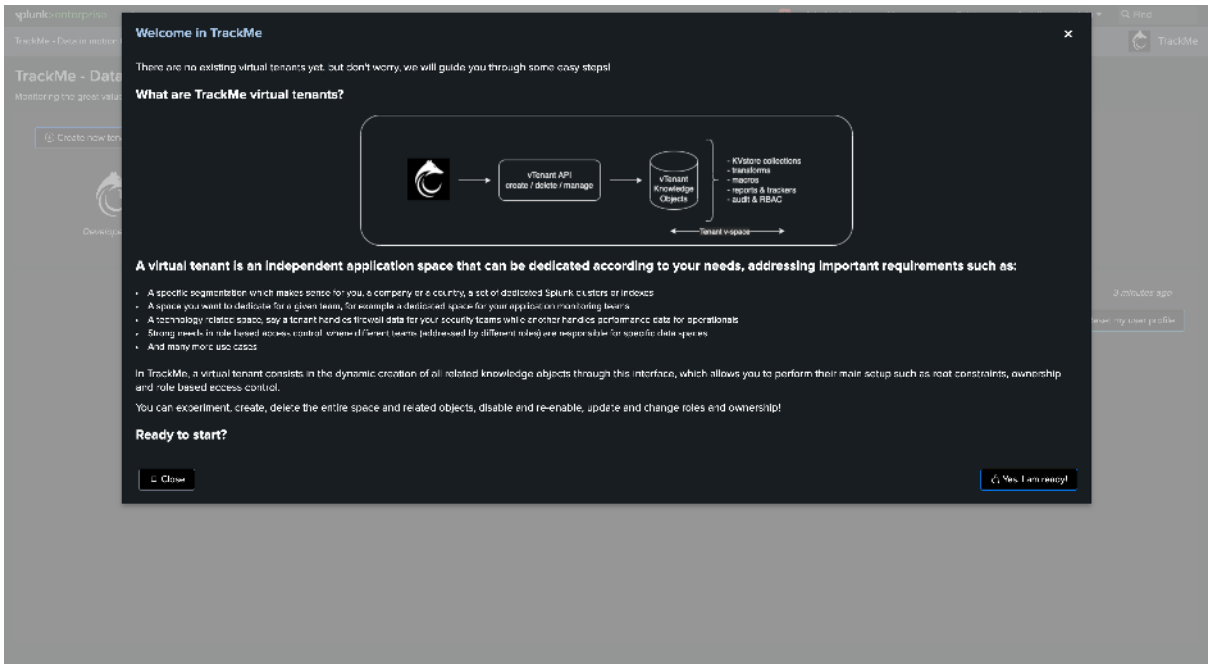
Purposes of Virtual Tenants:

- A specific segmentation which makes sense for you, a company or a country, a set of dedicated Splunk clusters or indexes
- Specific TrackMe components
- A space you want to dedicate for a given team, for example a dedicated space for your application monitoring teams

- A technology related space, say a tenant handles firewall data for your security teams while another handles performance data for operational teams
- Strong needs in role based access control, where different teams (addressed by different roles) are responsible for specific data spaces
- And many more use cases

In TrackMe, a virtual tenant consists in the dynamic creation of all related knowledge objects through this interface, which allows you to perform their main setup such as root constraints, ownership and role based access control.

You can experiment, create, delete the entire space and related objects, disable and re-enable, update and change roles and ownership!



7.4.2 Types of Virtual Tenants

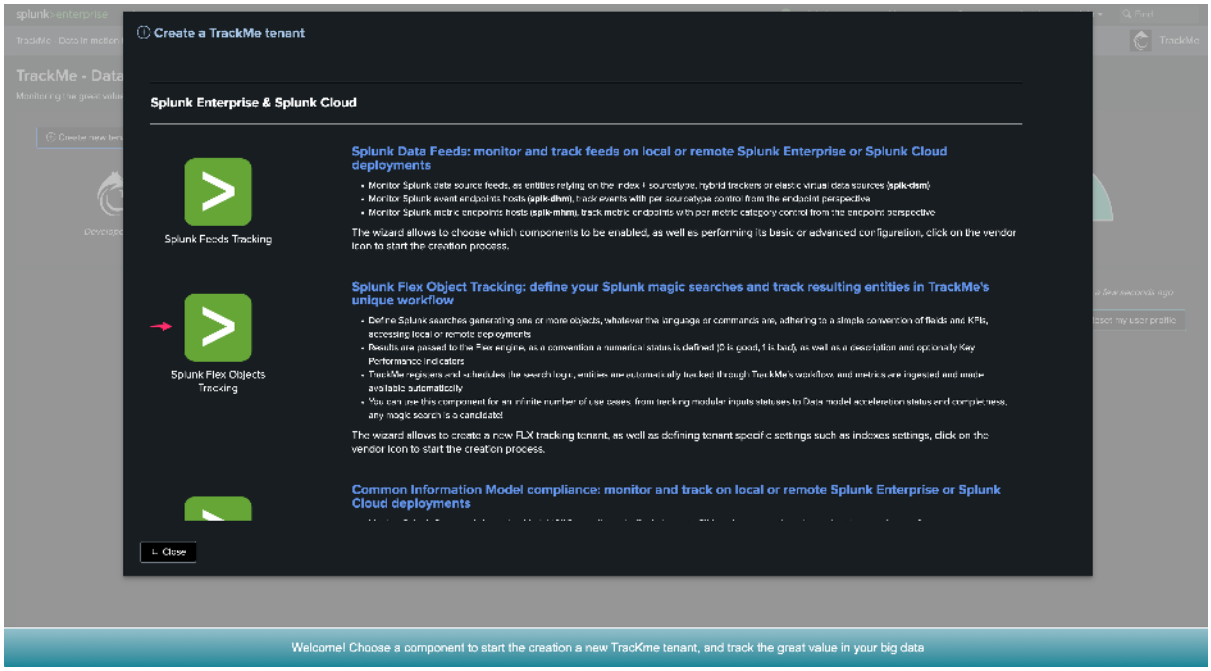
In TrackMe, Virtual Tenants are linked to TrackMe components, the following types of Virtual Tenants are available currently:

- splk-feeds Virtual Tenants, which include the following components (and can be enabled independently): splk-dsm / splk-dhm / splk-mhm
- splk-wlk Virtual Tenants, which stands for the TrackMe Workload component
- splk-flx Virtual Tenants, which stands for TrackMe Flex Magic components
- splk-cim Virtual Tenants, which stands for TrackMe Common Information Model compliance tracking

7.4.3 Creating a Splunk Feeds (splk-feeds) Virtual Tenant

splk-feeds wizard

A wizard guides through the creation of a Splunk feeds tracking Virtual Tenant, these guides steps allow you to:



- define the name (tenant_id) and description of the tenant
- choose which splk-feeds components will be enabled in the tenant (splk-dsm / splk-dhm / splk-mhm)
- for each component, define its main options such as the target Splunk environment (local or remote), data discovery scope, custom break by, etc
- define the Role Based Access Control policies (RBAC, administration and user roles, knowledge objects owner)
- define the Virtual Tenant indexes

Tenant identifier (tenant_id), Tenant Alias (tenant_alias) and Description (tenant_desc)

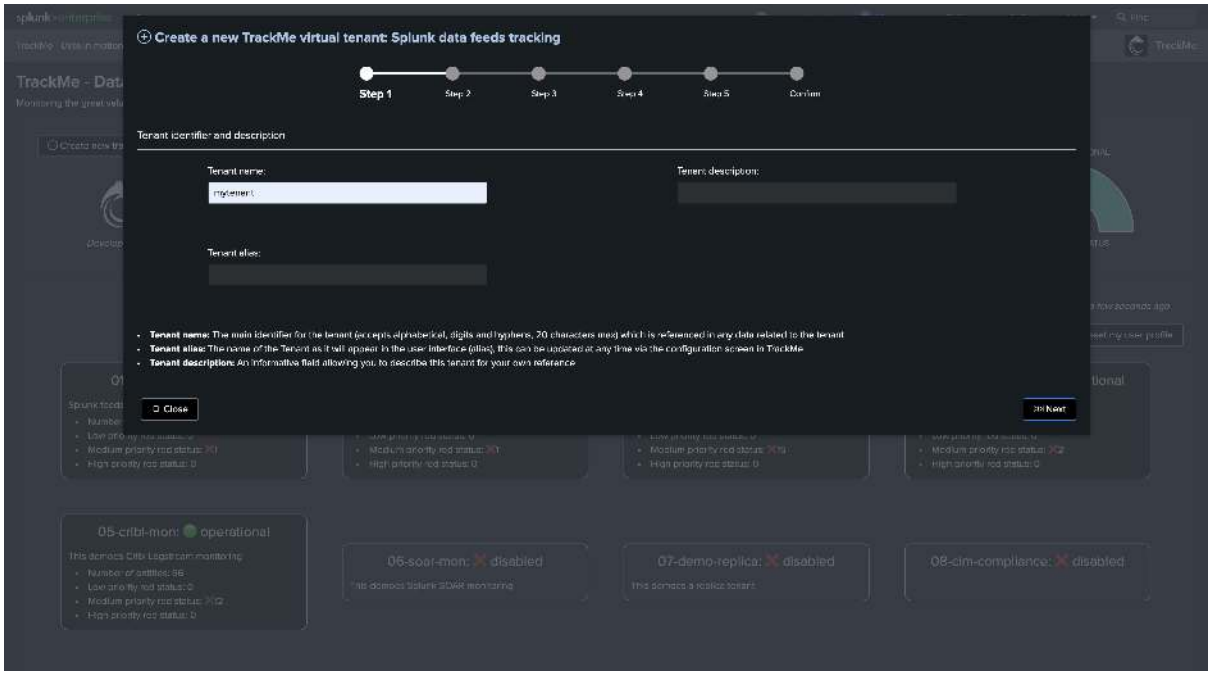
Hint

Tenant Alias since TrackMe 2.0.83

- Since TrackMe 2.0.83, you can define and update at any time the tenant alias
- Unlike the tenant_id which is immutable, the tenant_alias can be updated at any time
- The tenant alias is the name of the tenant as shown in the Virtual Tenant UI, it is also used to order the list of tenants in the UI
- The alias is optionnally defined during the Virtual Tenant creation, and can later on be updated in Configure / Virtual Tenants account

In this step, you define the unique identifier for the tenant (tenant_id) and optionally and alias and its description:

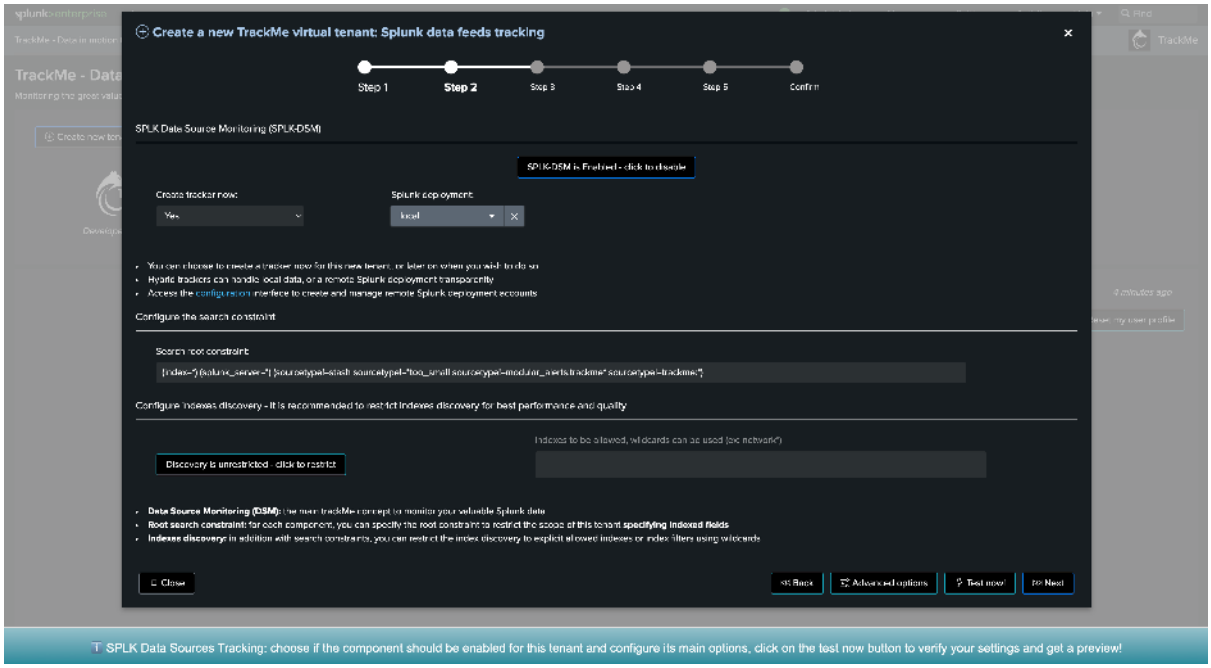
The tenant_id is to be unique amongst all the created tenants and is **immutable**, this field name is used in every piece of data (events and metrics) generated by TrackMe. (Note: the tenant_id is an indexed field)



splk-dsm

The component splk-dsm stands for Splunk Data Source Monitoring, it consists in tracking Splunk data with various powerful features, In a nutshell:

- Tracking Splunk feeds from the lens of the index / sourcetype
- Optionally, add a concept of custom break by to add a custom indexed or search type available field
- TrackMe will then generate and maintain entities accordingly, generate Key Performance Indicators, track outliers behaviour (Machine Learning), Data quality, etc



The main options to consider in the wizard:

- **Create tracker now:** you can choose to create a tracker within the wizard and during the Virtual Tenant creation time, which creates an Hybrid Tracker. (Hybrid trackers can as well be created at

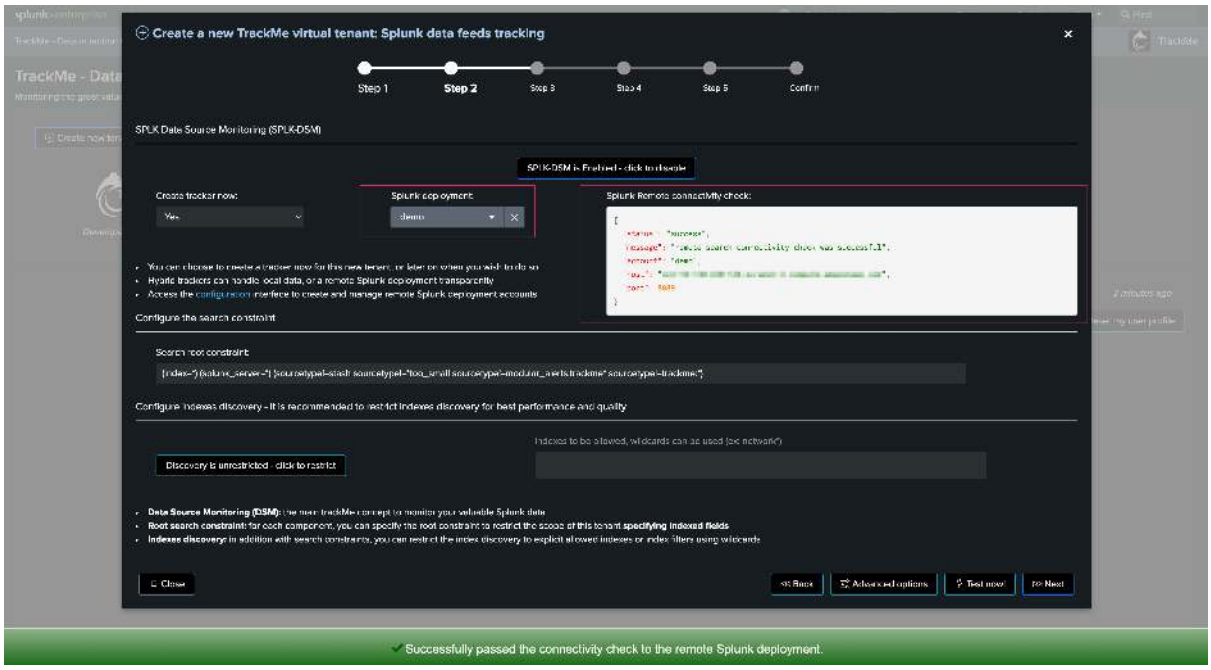
any time in the Virtual Tenant)

- **Splunk deployment:** you can set if the data is locally available, or if the target is a Splunk remote deployment
- **Splunk root search constraint:** defines the root search constraint that applies for the discovery and management of feeds entities
- **Restrict indexes discovery:** in addition with the root search constraint, you can define the indexes discovery using explicit and wildcard based index patterns
- **Advanced options:** define settings for the Hybrid tracker, if enabled, and other component specific options if any (custom break by, etc)
- **Test now:** allows you to test in preview the execution and entities discovery

local versus remote deployment:

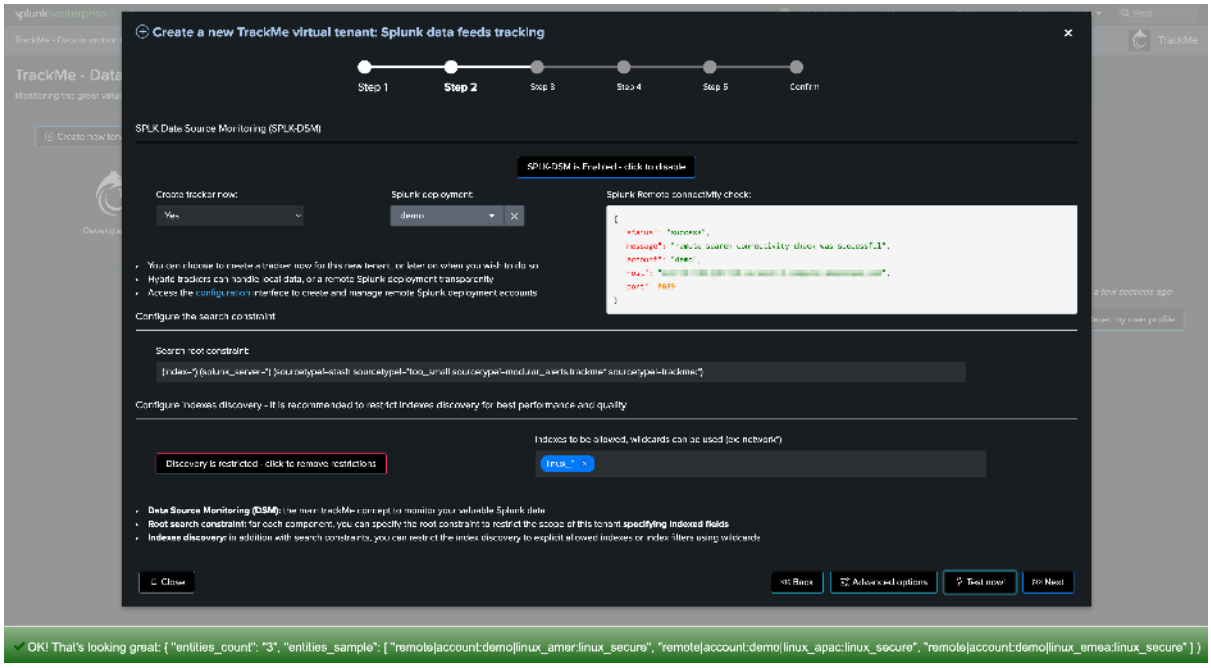
In TrackMe, when defining Virtual Tenants and trackers, you can always choose between the local deployment, and a Splunk remote deployment if you have configured any,

When switching to a remote target, TrackMe will perform a connectivity and authentication test, and show the result:



Testing the Hybrid tracker:

You can test your configuration at anytime using the "Test now" button, this performs a preview search accordingly to your target and settings:



A notification will appear at the bottom of the screen depending on if there are entities that could be found or not.

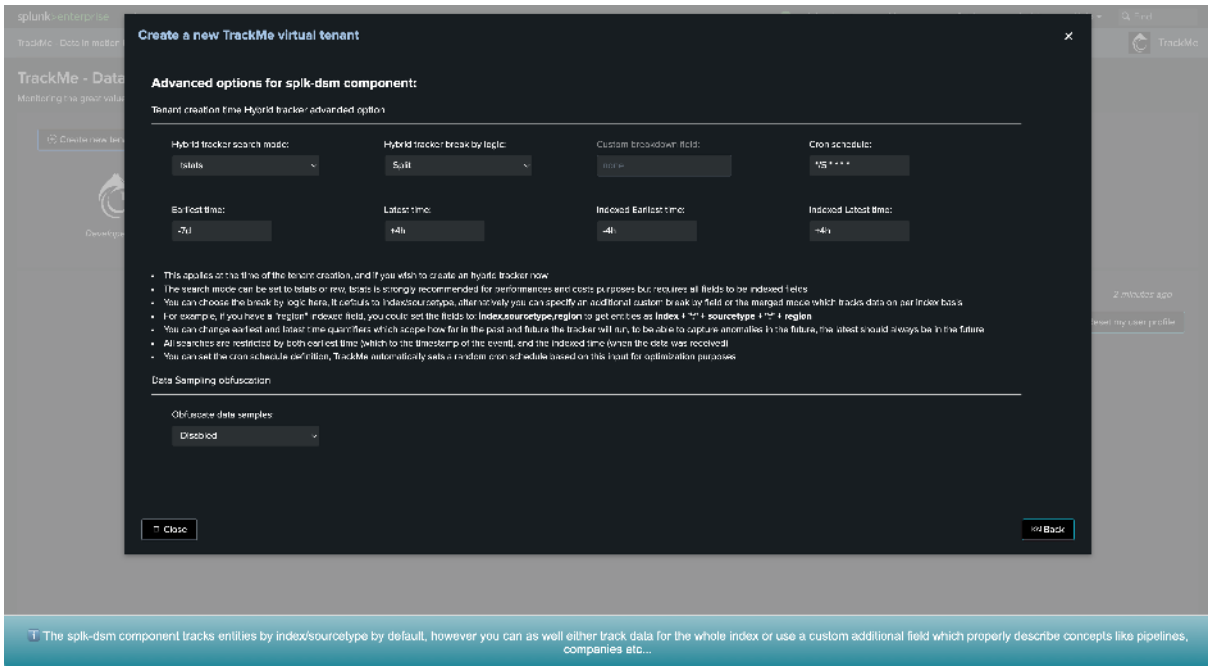
Restricting indexes discovery:

You can restrict the scope of the Virtual Tenant, either by customising the root search constraint, or specifying indexes patterns:

Both the search constraint and indexes discovery configuration can be updated later on in the Virtual Tenant configuration.

Advanced options:

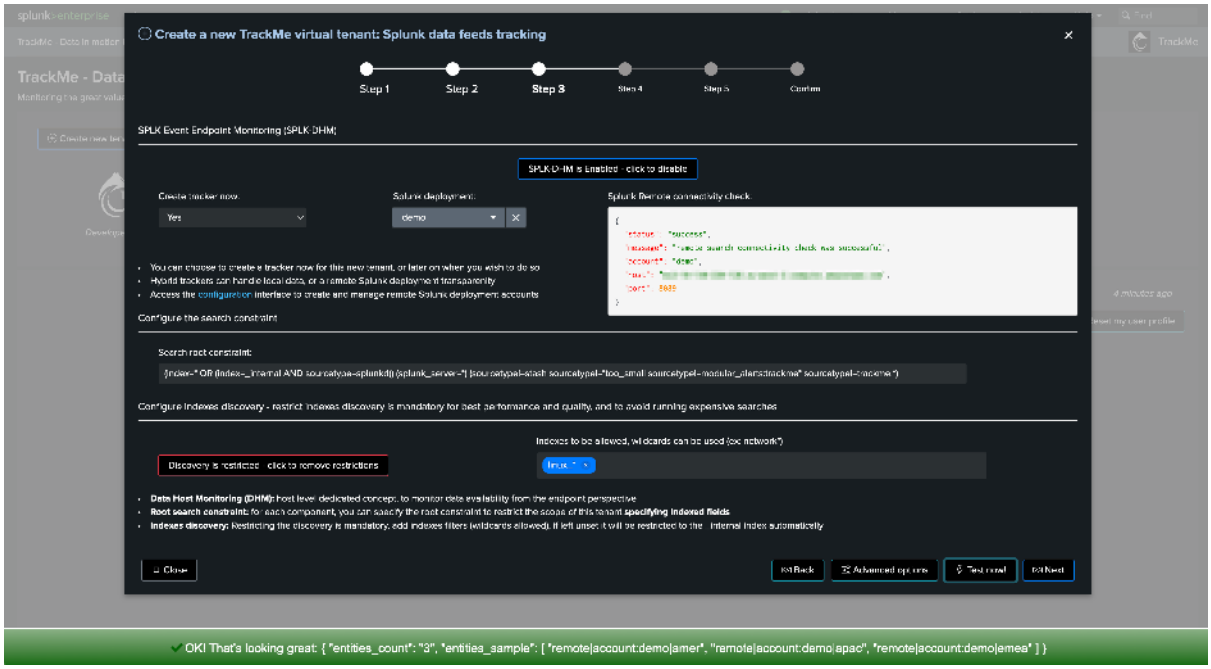
This screen allows you to customise the Hybrid tenant creation, as well as defining additional options specifics to the component.



splk-dhm

The component splk-dhm stands for Splunk Data Host Monitoring, it consists in tracking Splunk data from the lens of an endpoint concept:

- Track sourcetypes activity per endpoint
- Define the concept of endpoint (default to the host Metadata, but can be updated to any custom field)
- Monitor independently or as a whole the data availability per endpoint, apply policies, etc!



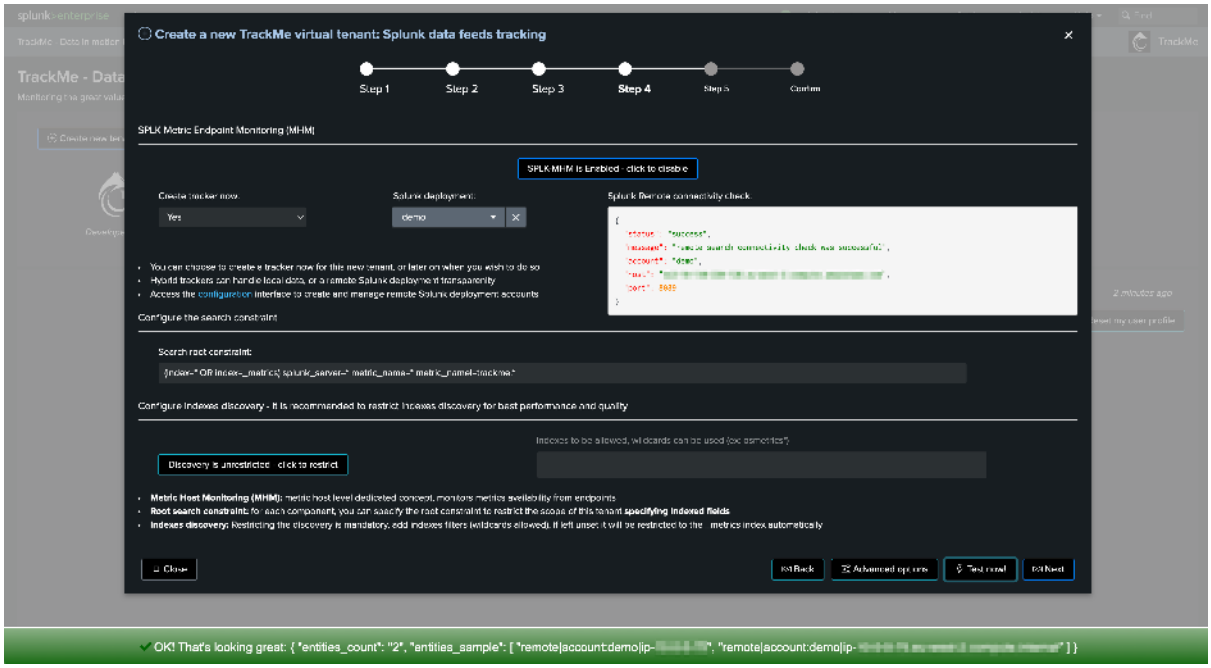
The main options to consider in the wizard:

- **Create tracker now:** you can choose to create a tracker within the wizard and during the Virtual Tenant creation time, which creates an Hybrid Tracker. (Hybrid trackers can as well be created at any time in the Virtual Tenant)
- **Splunk deployment:** you can set if the data is locally available, or if the target is a Splunk remote deployment
- **Splunk root search constraint:** defines the root search constraint that applies for the discovery and management of feeds entities
- **Restrict indexes discovery:** in addition with the root search constraint, you can define the indexes discovery using explicit and wildcard based index patterns
- **Advanced options:** define settings for the Hybrid tracker, if enabled, and other component specific options if any (custom break by, etc)
- **Test now:** allows you to test in preview the execution and entities discovery

splk-mhm

The component splk-mhm stands for Splunk Metric Host Monitoring, it consists in tracking Splunk metrics from the lens of an endpoint concept:

- Track metric categories availability per endpoint
- Define the concept of endpoint (default to the host Metadata, but can be updated to any metric dimension)
- Apply policies, etc

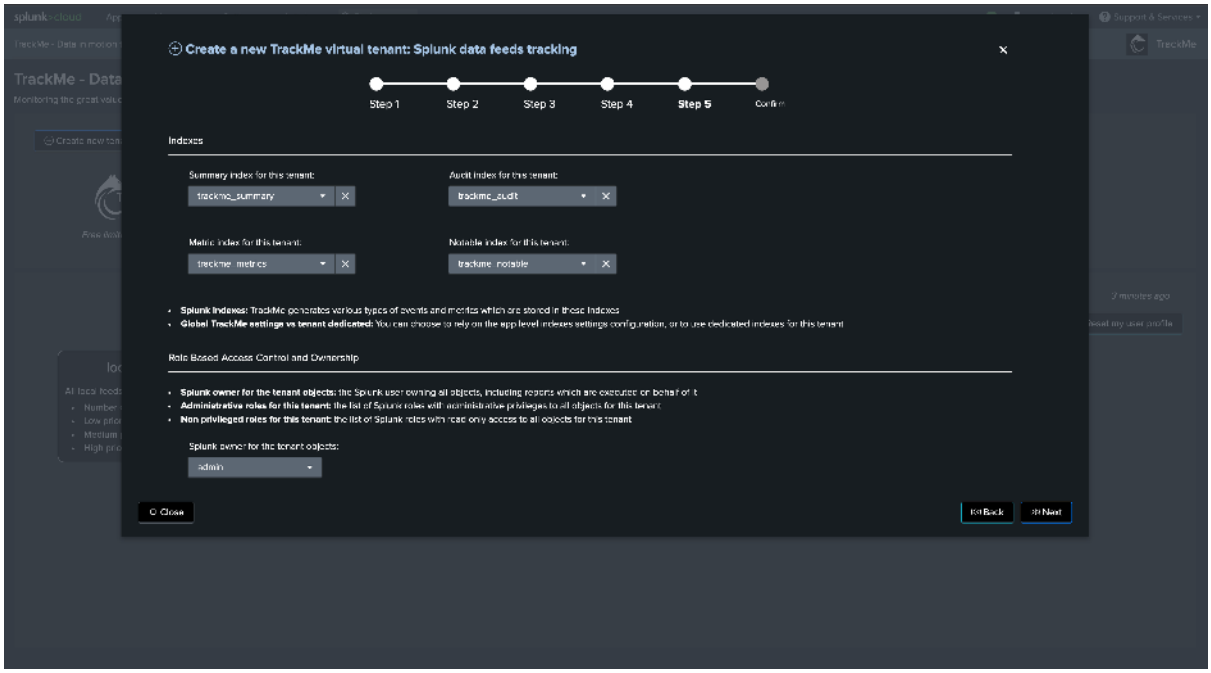


The main options to consider in the wizard:

- **Create tracker now:** you can choose to create a tracker within the wizard and during the Virtual Tenant creation time, which creates an Hybrid Tracker. (Hybrid trackers can as well be created at any time in the Virtual Tenant)
- **Splunk deployment:** you can set if the data is locally available, or if the target is a Splunk remote deployment
- **Splunk root search constraint:** defines the root search constraint that applies for the discovery and management of feeds entities
- **Restrict indexes discovery:** in addition with the root search constraint, you can define the indexes discovery using explicit and wildcard based index patterns
- **Advanced options:** define settings for the Hybrid tracker, if enabled, and other component specific options if any (custom break by, etc)
- **Test now:** allows you to test in preview the execution and entities discovery

RBAC, ownership and indexes

Common to all Virtual Tenants, the final step allows you to define your RBAC policy, the knowledge object owner and the indexes that will be used in the scope of this Virtual Tenant:



RBAC:

- define the user roles required for the administration of the Tenant, users members of these roles can access and administrate the tenant
- define the user roles required for the usage of the Tenant without modification privileges, users members of these roles can access the tenant but cannot perform any kind of modifications

Owner:

- defines the Splunk user owning all the tenant related knowledge objects
- executions of the Tracker for instance will be executed on behalf of this user
- any further knowledge object, such as a new Hybrid tracker, that you would create later on will be automatically assigned to this user

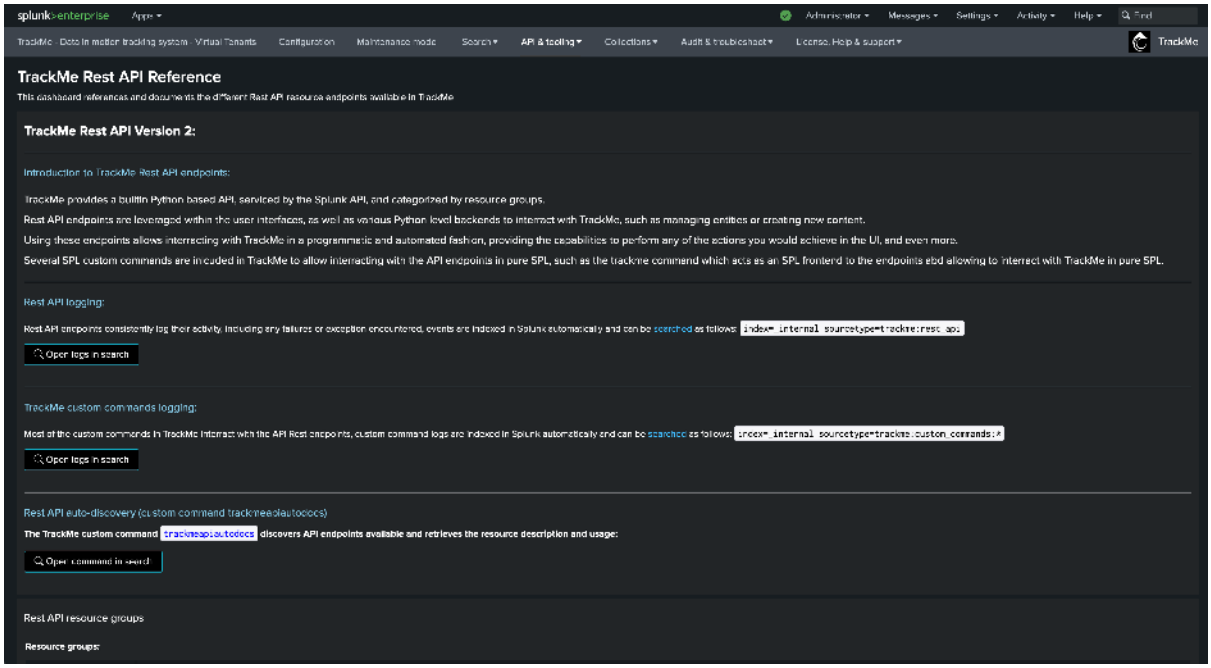
Indexes:

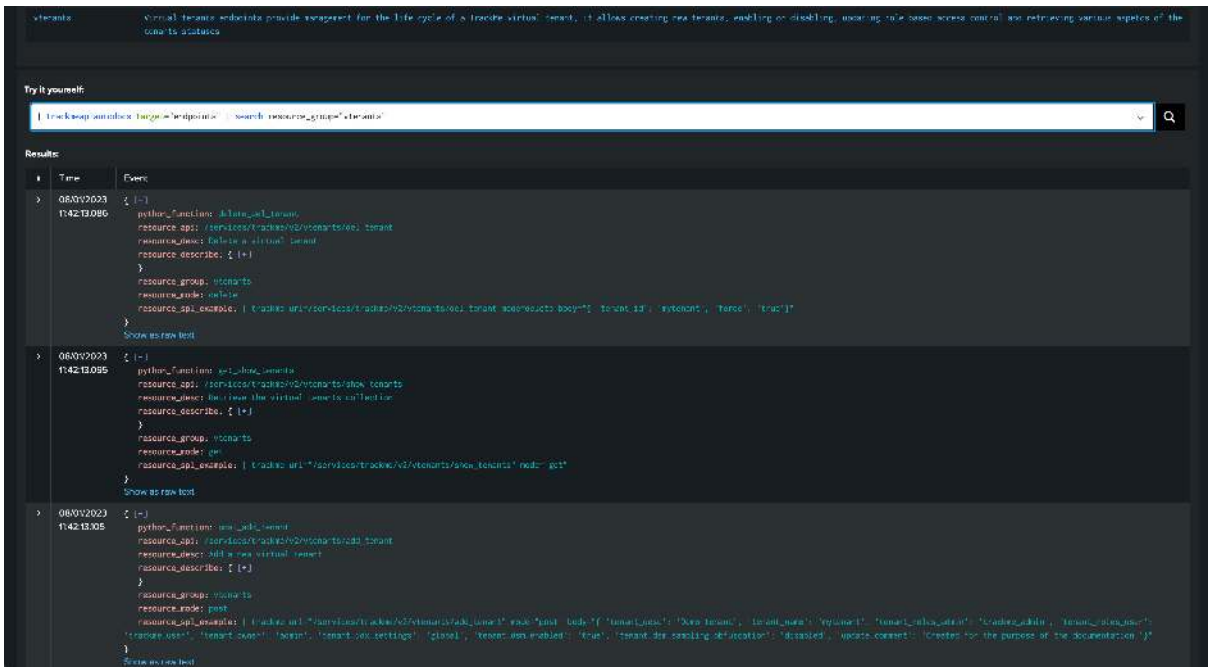
- defines the Splunk indexes for this tenant
- the indexes need to have been defined prior to this step

splk-feeds REST

TrackMe provides a deep REST API for every action that is available in the application, for a full list of endpoints and options, consult the REST API reference user interface:

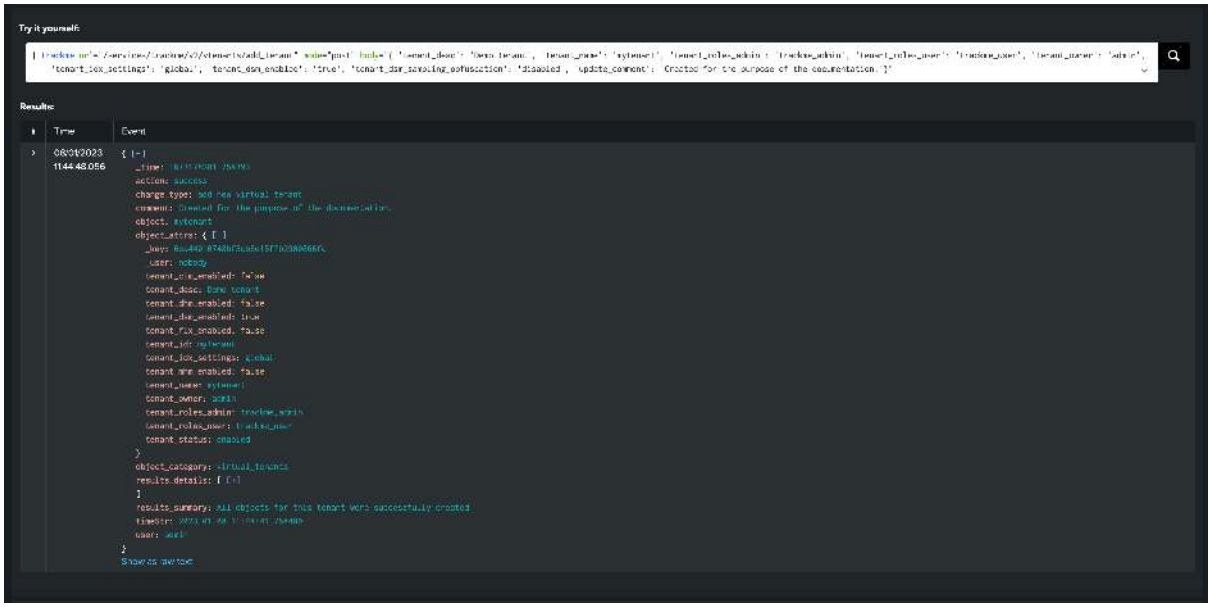
- Navigation bar / API & Tooling / TrackMe REST API Reference





Example, you can create a new Virtual Tenant for splk-dsm with the following SPL command:

```
| trackme url="/services/trackme/v2/vtenants/admin/add_tenant" mode="post" body="{
↪'tenant_desc': 'Demo tenant', 'tenant_name': 'mytenant', 'tenant_roles_admin':
↪'trackme_admin', 'tenant_roles_user': 'trackme_user', 'tenant_owner': 'admin',
↪'tenant_idx_settings': 'global', 'tenant_dsm_enabled': 'true', 'tenant_dsm_sampling_
↪obfuscation': 'disabled', 'update_comment': 'Created for the purpose of the␣
↪documentation.'}"
```

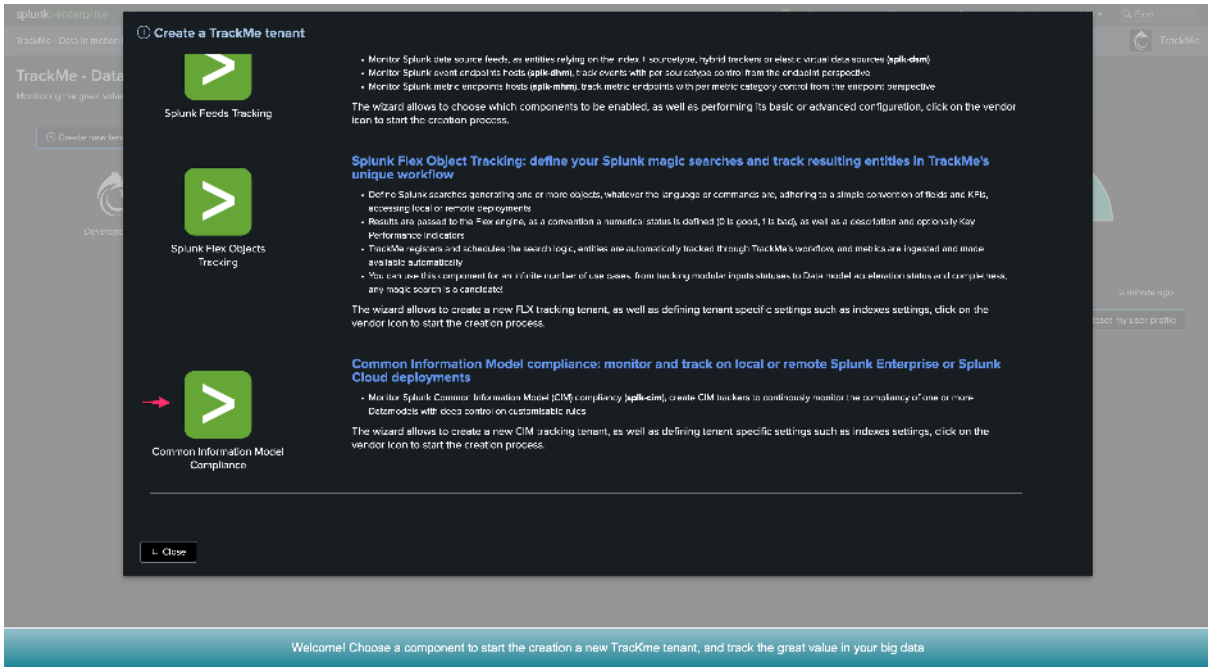


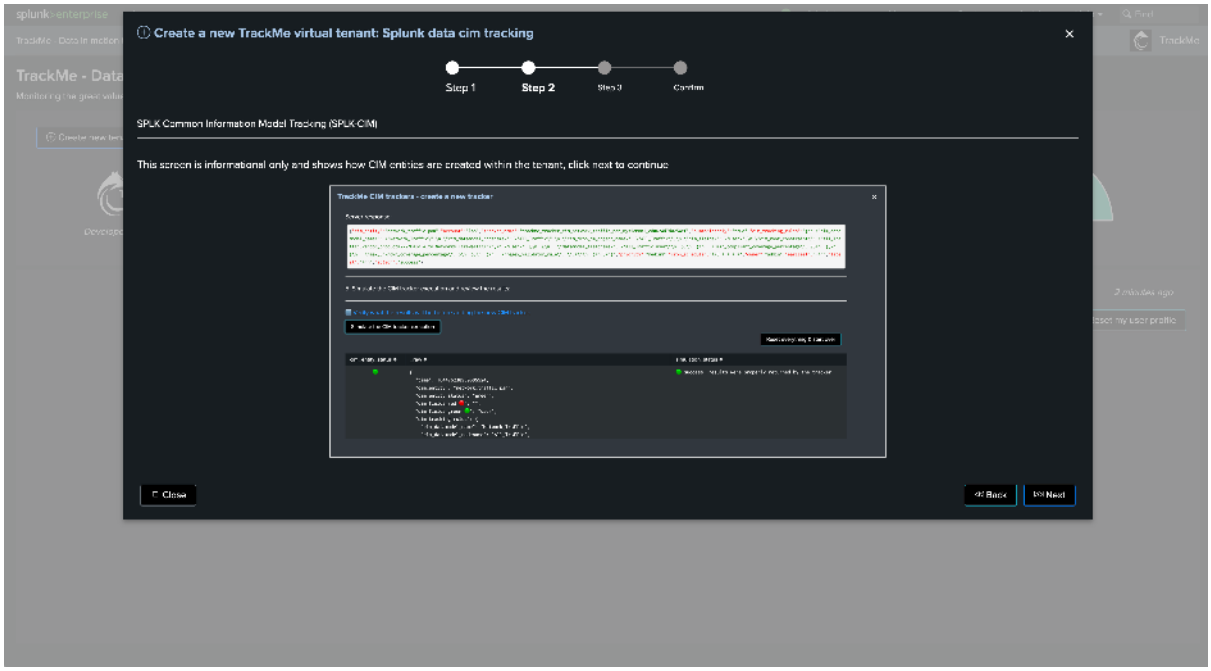
7.4.4 Creating a CIM compliance (splk-cim) Virtual Tenant

splk-cim wizard

When creating a new splk-cim Virtual Tenant, you only need to specify the tenant identifier and description, as well as the RBAC, ownership and indexes policies.

The configuration process of entities is handled once the tenant has been created within the Tenant user interface:





splk-cim REST

You can create a new splk-cim Virtual Tenant using the following SPL command:

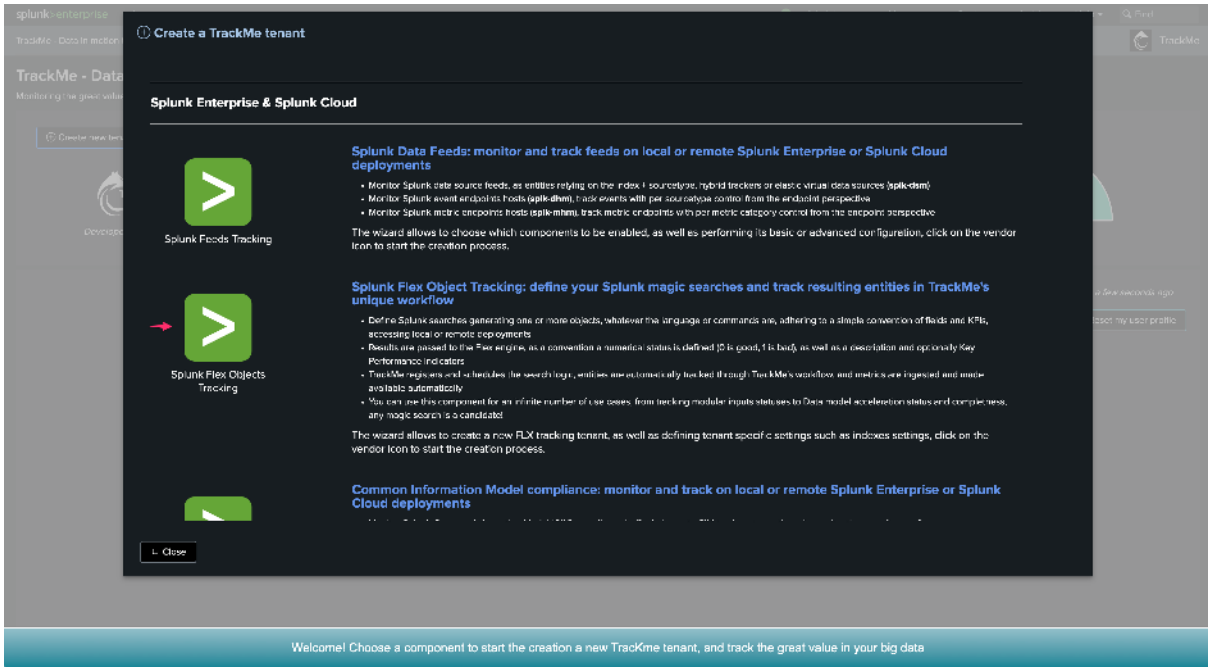
```
| trackme url="/services/trackme/v2/vtenants/admin/add_tenant" mode=post body="{
↪'tenant_desc': 'SIEM', 'tenant_name': 'mytenant', 'tenant_roles_admin': 'trackme_
↪admin', 'tenant_roles_user': 'trackme_user', 'tenant_owner': 'admin', 'tenant_idx_
↪settings': 'global', 'tenant_cim_enabled': 'true'}"
```

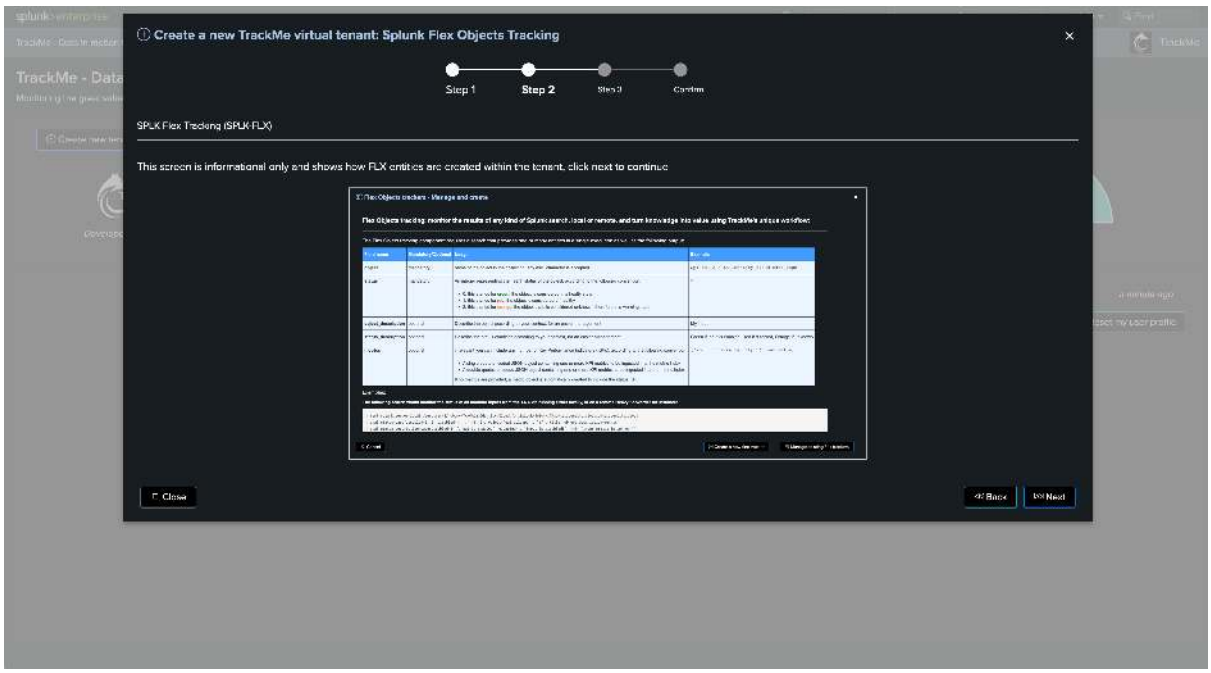
7.4.5 Creating an Splunk Flex Object (splk-flx) Virtual Tenant

splk-flx wizard

When creating a new splk-flx Virtual Tenant, you only need to specify the tenant identifier and description, as well as the RBAC, ownership and indexes policies.

The configuration process of entities is handled once the tenant has been created within the Tenant user interface:





splk-flx REST

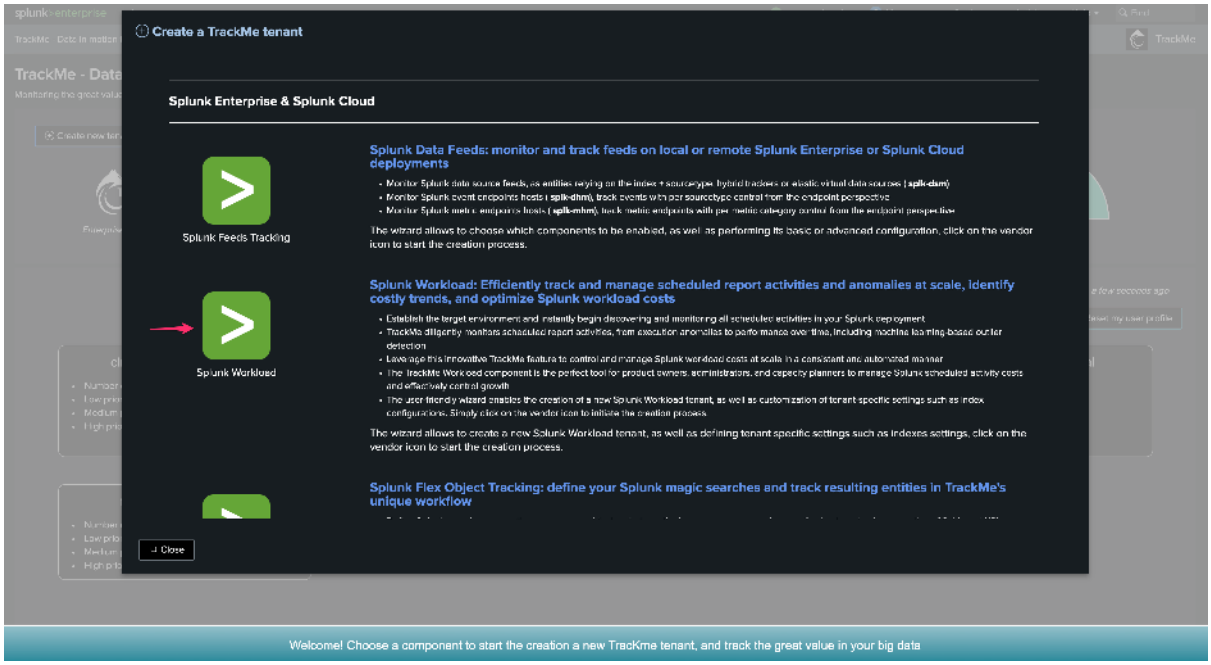
You can create a new splk-wlk Virtual Tenant using the following SPL command:

```
| trackme url="/services/trackme/v2/vtenants/admin/add_tenant" mode=post body="{
↪'tenant_desc': 'SIEM', 'tenant_name': 'mytenant', 'tenant_roles_admin': 'trackme_
↪admin', 'tenant_roles_user': 'trackme_user', 'tenant_owner': 'admin', 'tenant_idx_
↪settings': 'global', 'tenant_flx_enabled': 'true'}"
```

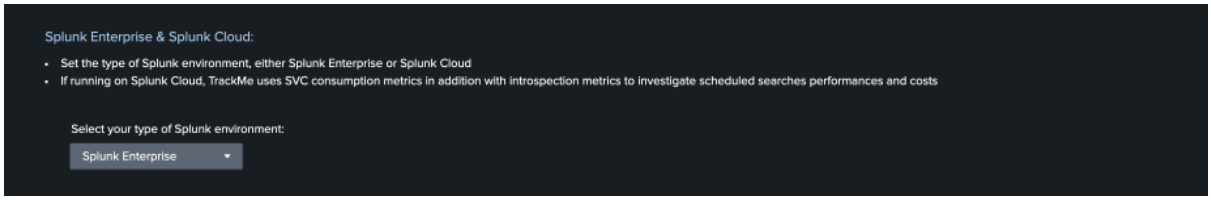
7.4.6 Creating a Splunk Workload Virtual Tenant

splk-wlk wizard

When creating a Splunk Workload Virtual Tenant, you will specify various options as the tracker configuration is handled during the Virtual Tenant creation phase:



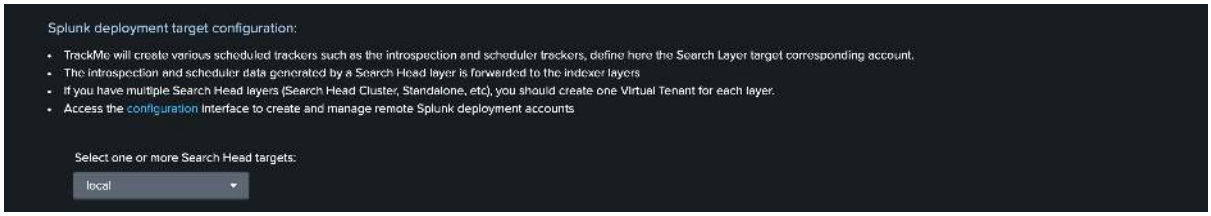
Splunk deployment type



Define the type of deployment:

- If you select Splunk Cloud, a tracker will be created to monitor the Splunk SVC consumption summary metrics.

Splunk deployment target



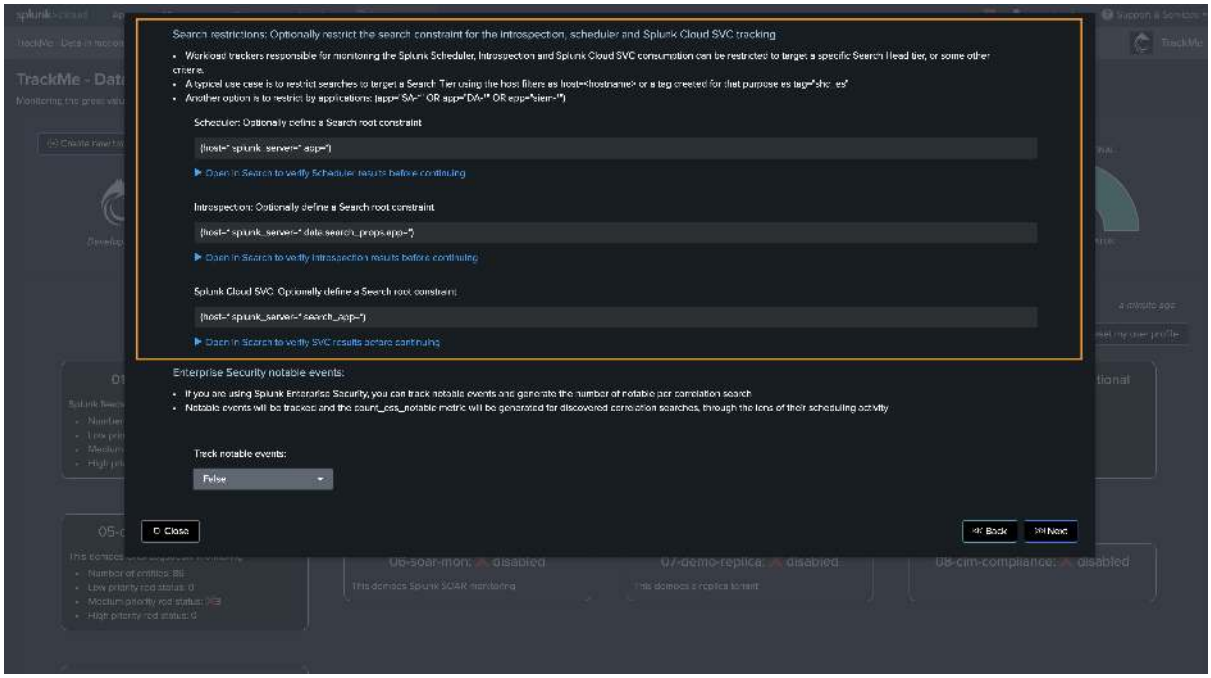
Define the target:

- If local, the searches are going to be performed locally, which the introspection, scheduler and other types of searches are running against data that can be searched on the Search Head hosting TrackMe
- You can as well set a remote deployment account, which can target one or more Splunk REST API endpoints
- TrackMe will adapt transparently searches as needed to use the splunkremotesearch command with the appropriate account

Root search constraint

Multiple Search Head Tiers

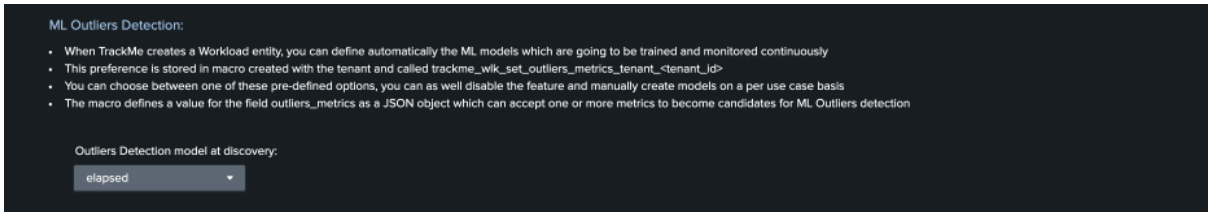
- When you have multiple logical Search Head tiers (for instance a Search Head Cluster and one or more Standalone Search Heads), it is very important to restrict the root constraint and target only these Search Head members
- To do so, ensure to use the **host** Metadata, either explicitly (host=myserver1 OR host=server2) or any equalivent technique of your choice (subsearch, lookups, etc)
- You can for instance dedicate a Tenant per Search Head tier which is the easiest solution, alternatively you can also use the Grouping option and manual definition of the Workload trackers for advanced setups with multiple Search Head tiers within the same tenant



Search constraint:

- You can optionally define additional search filters to be used for the introspection, scheduler and Splunk Cloud SVC metrics
- This can be useful to define the scope of the Workload tenant, filtering on Splunk applications or Splunk host related metadata (host, splunk_server)

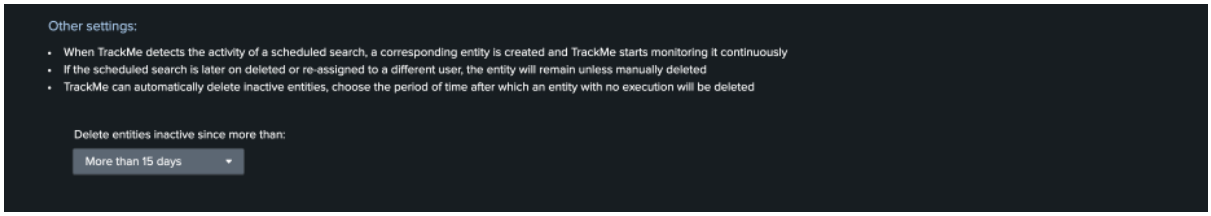
ML Outliers



Define ML outliers models at the entity discovery phase:

- When TrackMe will discover entities, it can automatically create and train ML models
- The default behaviour is to train ML models against the elapsed metric (the search run time from the introspection perspective)

Workload Inactive entities



Automatically purge inactive scheduled entities after a given period of time:

- This settings influences the behaviour of the "inactive_entities" Workload tracker
- When the tracker runs, it inspects entities which have not been active for a period of time, and depending on this value, it will automatically removes these entities from the KVstore collections

splk-wlk REST

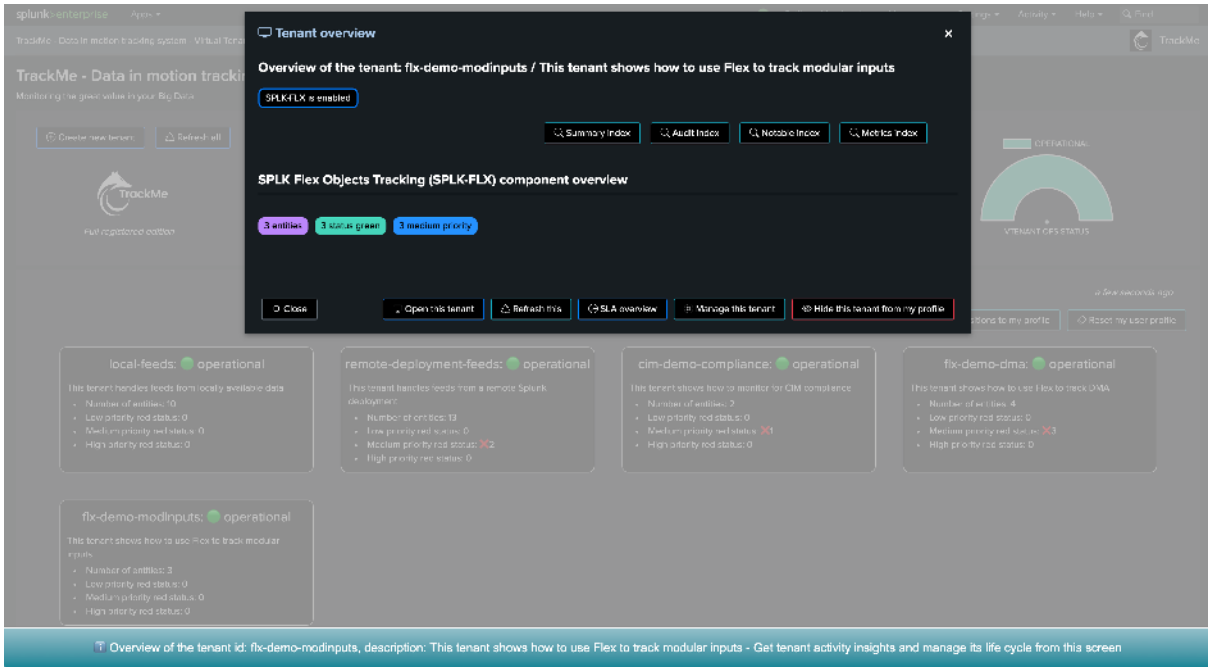
You can create a new splk-wlk Virtual Tenant using the following SPL command:

```
| trackme url="/services/trackme/v2/vtenants/admin/add_tenant" mode="post" body="{
↪'tenant_desc': 'SIEM', 'tenant_name': 'mytenant', 'tenant_roles_admin': 'trackme_
↪admin', 'tenant_roles_user': 'trackme_user', 'tenant_owner': 'admin', 'tenant_idx_
↪settings': 'global', 'tenant_wlk_enabled': 'true'}"
```

7.5 Manage Virtual Tenants

7.5.1 Access a Virtual Tenant

To manage the life cycle of a Virtual Tenant, double click on the tenant from the Virtual Tenants user interface:



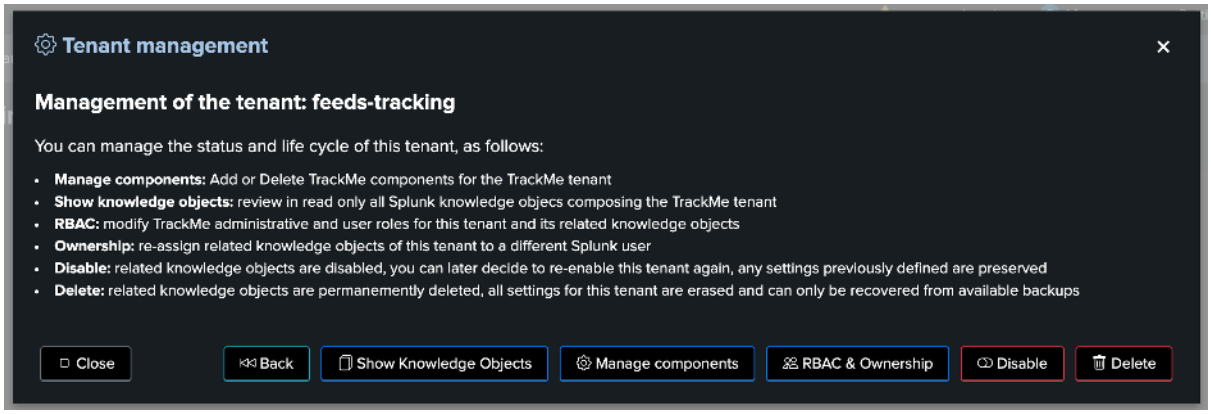
From this screen, you can:

- Open the Virtual Tenant user interface
- Refresh the Virtual Tenant entities information
- Access the SLA dashboard for this Virtual Tenant

- Access to the Virtual Tenant management options
- Hide this Virtual Tenant from your personal profile
- Access to any of the Virtual Tenant indexes easily

7.5.2 Managing a Virtual Tenant

When accessing to the Virtual Tenant management:

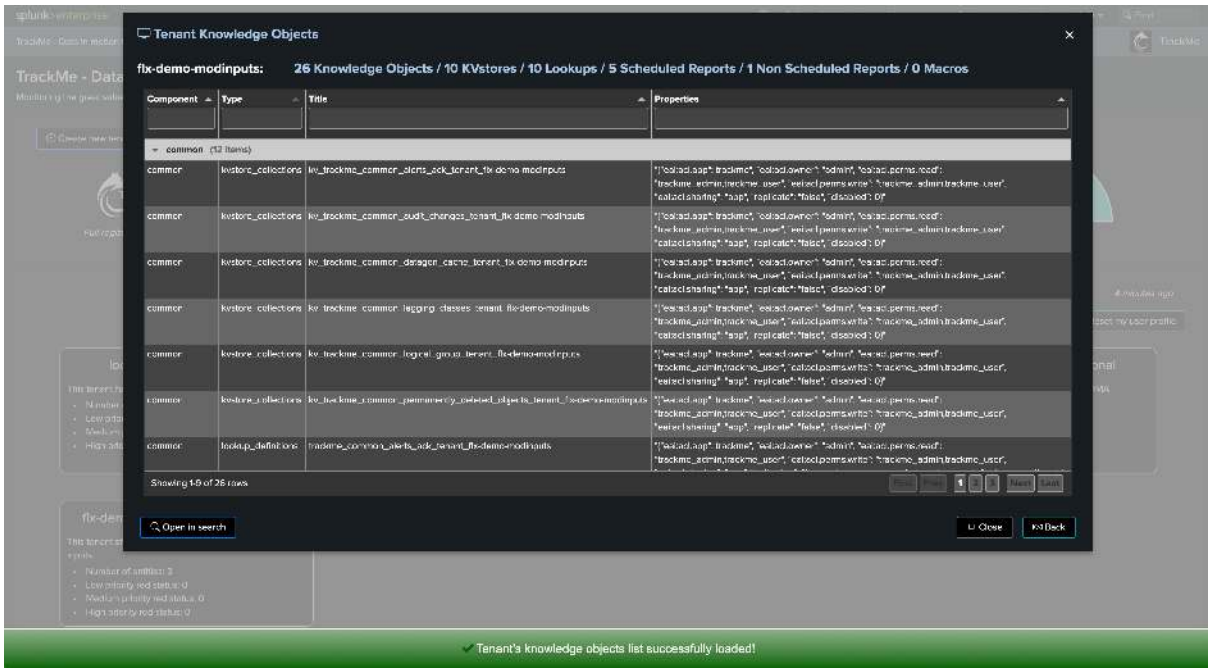


From this screen, you can:

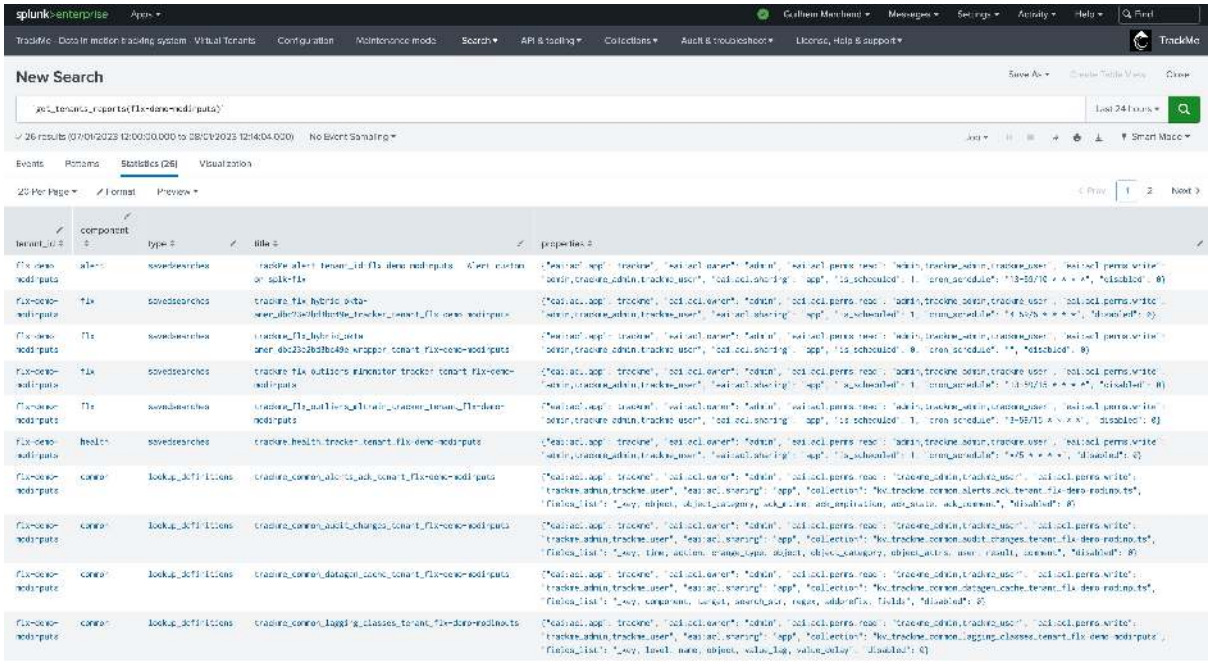
- Access to the Knowledge Objects explorer for this tenant
- Modify components enabled in the tenant (add / delete tenants)
- Update the RBAC policies and ownership
- Disable this tenant and all of its related Knowledge Objects
- Permanently delete this tenant and all of its related Knowledge Objects

Knowledge Objects Explorer

This screen allows you to easily get the full list of Knowledge Objects and their types for that specific tenant:

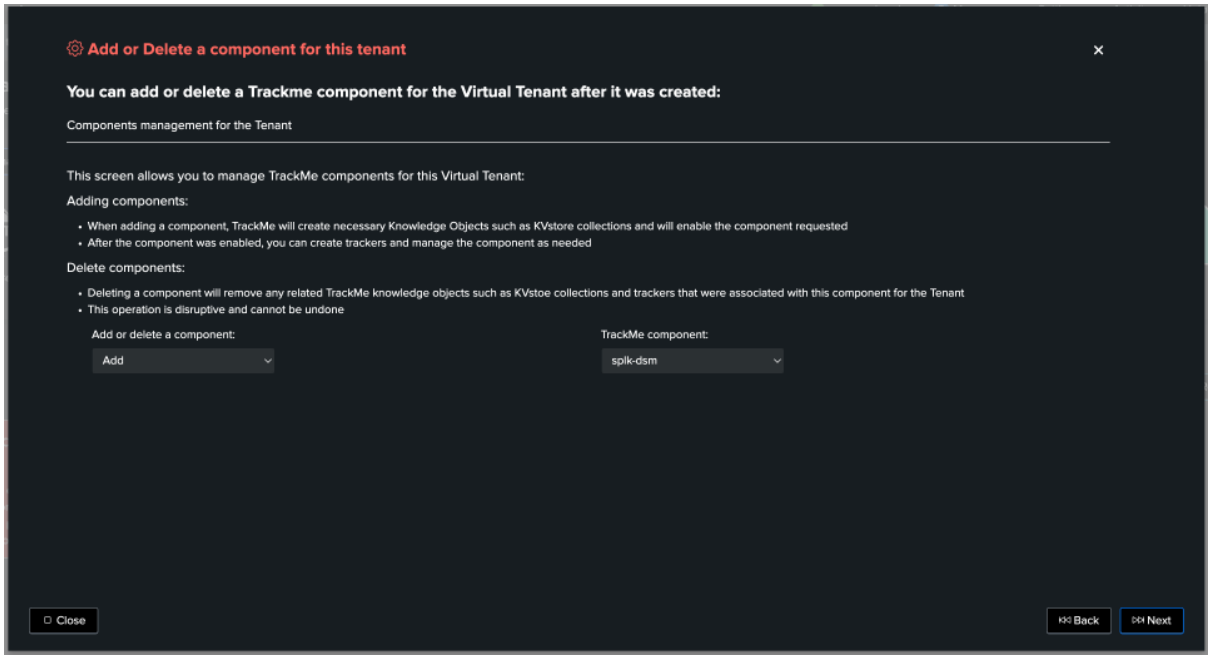


Click on Open in search to access through the Splunk Search user interface:



Modify (Add/Delete) TrackMe components

This screen allows to add or delete a component to an existing Virtual Tenant:



Splunk Workload (splk-wlk)

- Currently, the splk-wlk component cannot be added to or deleted from an existing Virtual Tenant
- However, a Workload Virtual Tenant can be modified to add or delete other components

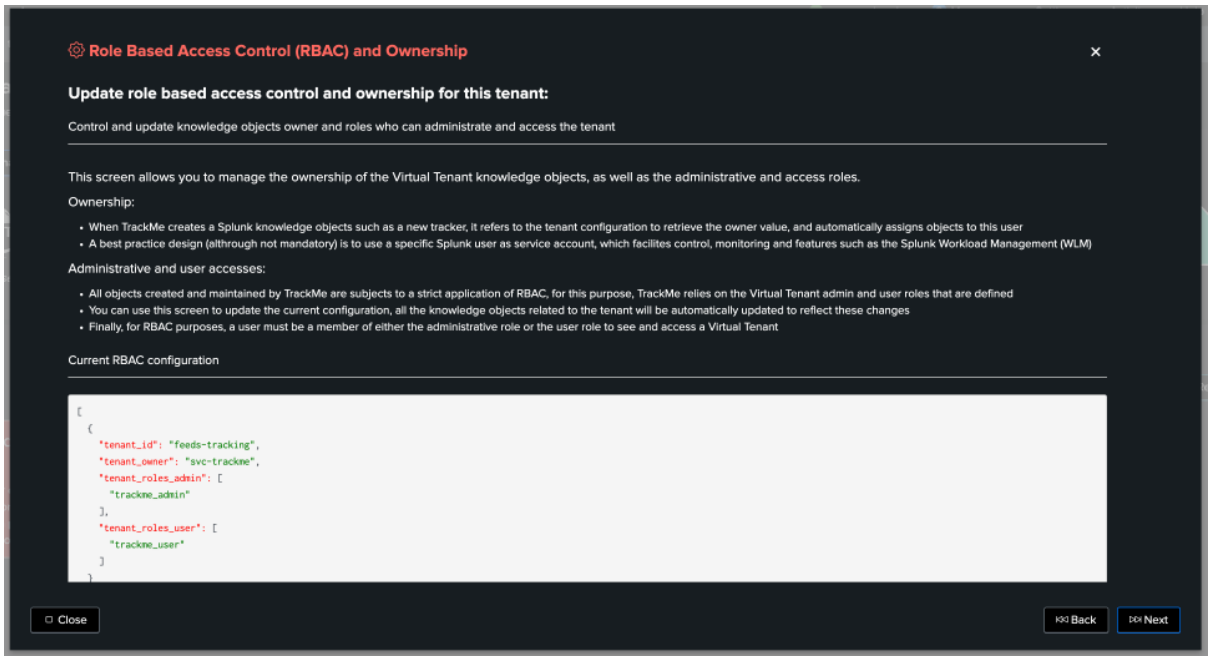
Attention

Deleting a component

- When deleting a component from a Virtual Tenant, all TrackMe related knowledge objects are permanently deleted
- Related KVstores content are as well permanently purged, this operation cannot be undone

Updating RBAC and ownership

The screen allows you to modify the current RBAC policies and the Splunk ownership for this tenant:

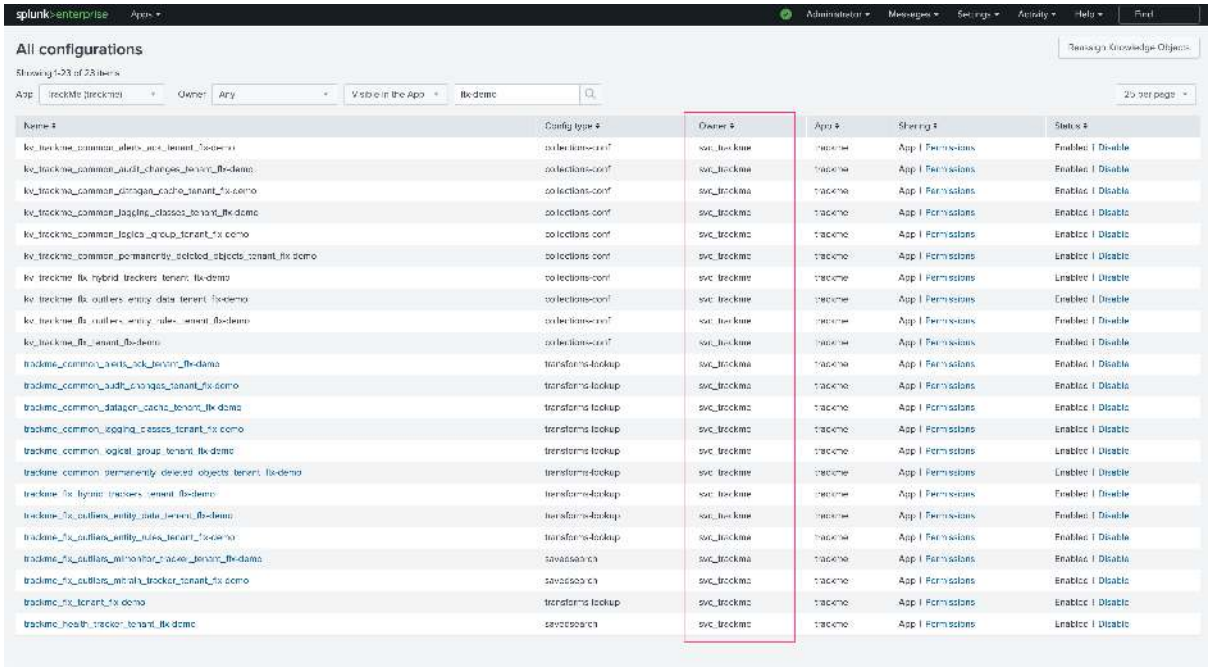


When updating the policies or ownership:

- All Knowledge Objects will be updated accordingly
- If the owner is changed, all Knowledge Objects will automatically be reassigned to the target Splunk user
- Future TrackMe objects will be created and assigned to the owner user defined for the Virtual Tenant

This can as well be performed via the REST endpoint and the following SPL command:

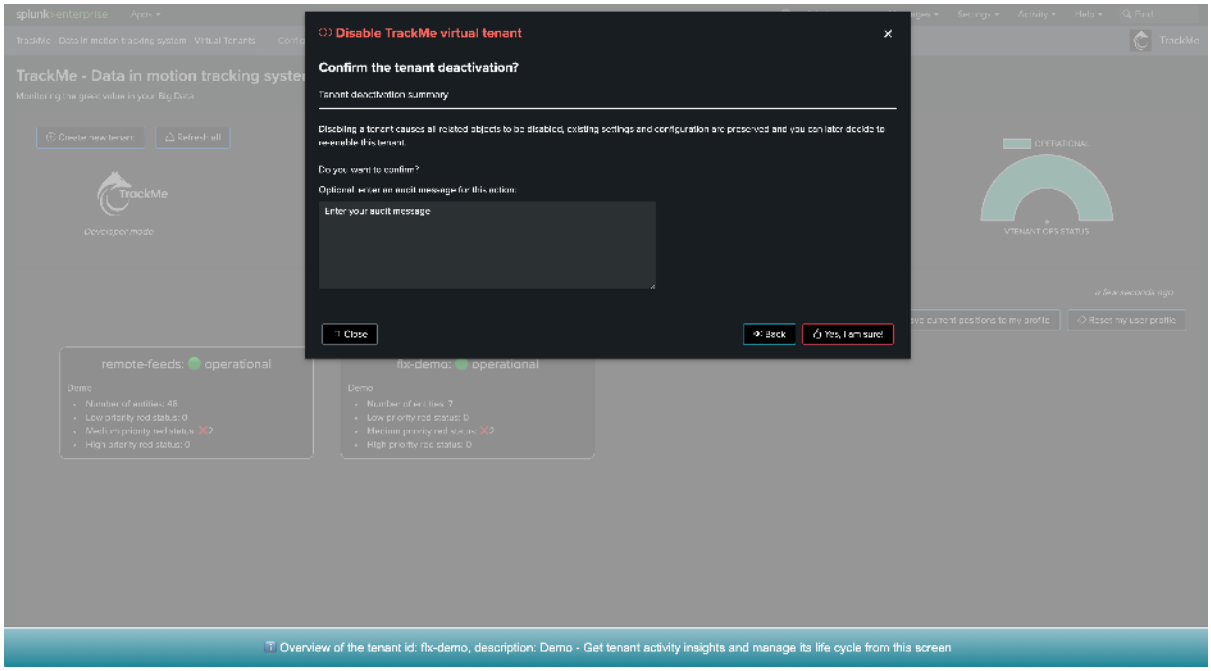
```
| trackme url=/services/trackme/v2/vtenants/admin/update_tenant_rbac mode=post body="
↪{ 'tenant_id': 'flx-demo', 'tenant_roles_admin': 'trackme_amer_admin', 'tenant_
↪roles_user': 'trackme_amer_user', 'tenant_owner': 'svc_trackme'}"
```

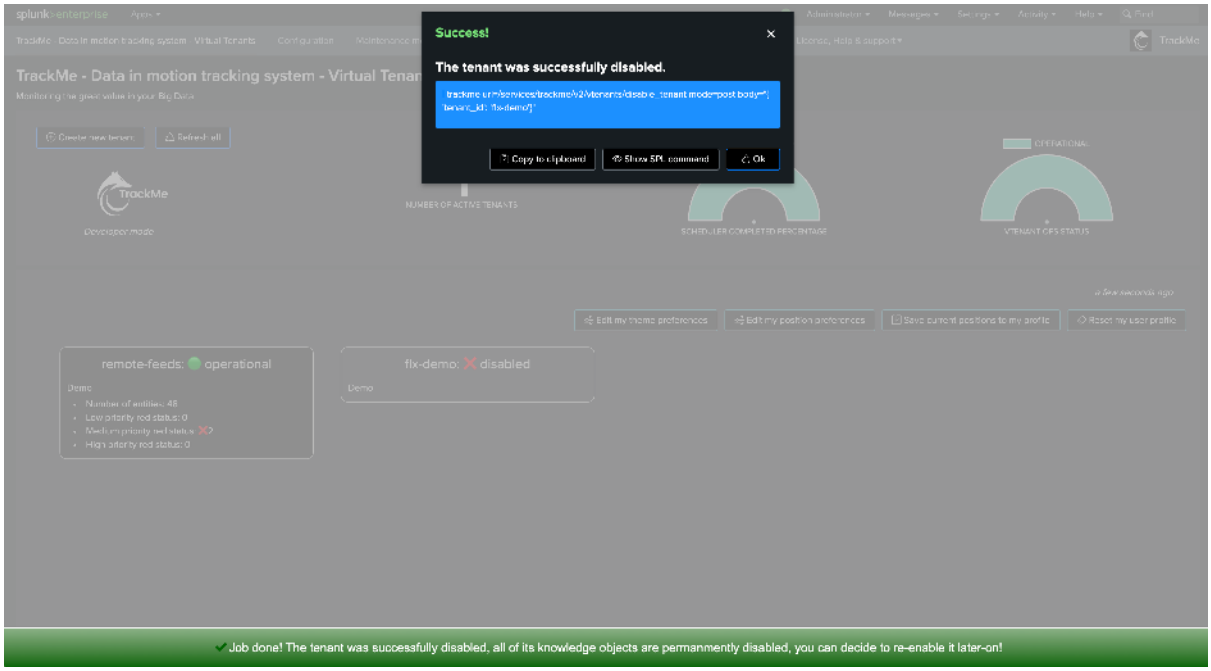


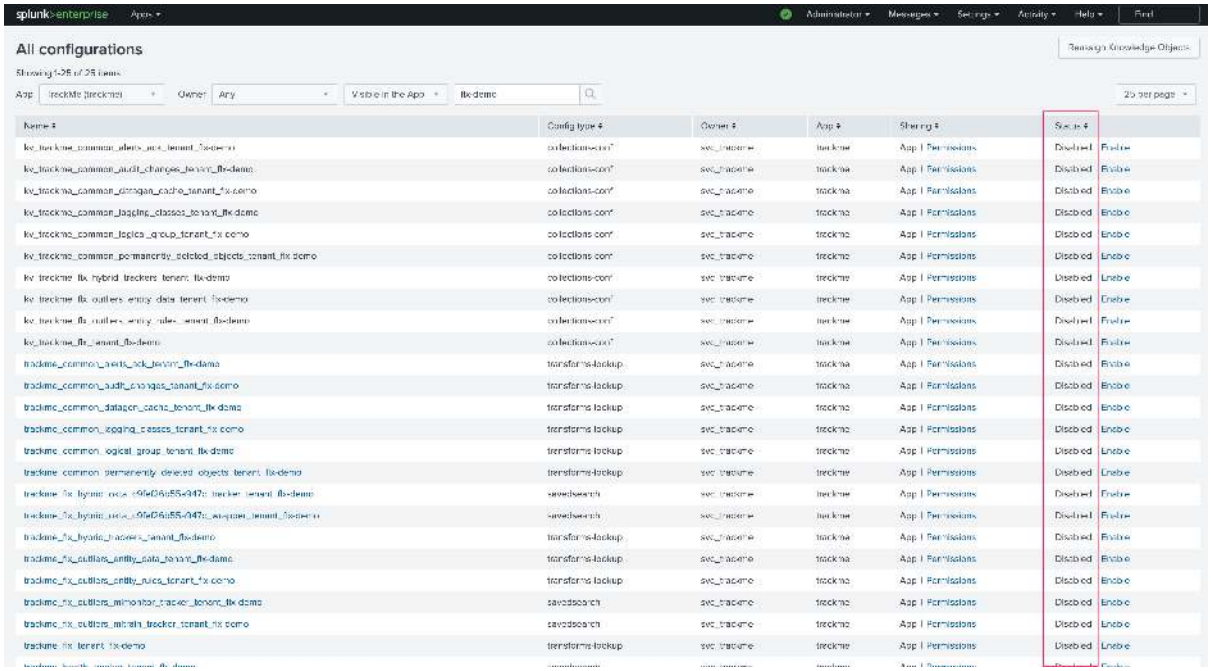
Enabling / Disabling a tenant

You can enable / disable a tenant at any time, when disabling a tenant:

- All Knowledge Objects are disabled in Splunk
- None of the executions related to that tenant will be performed as long as the tenant is disabled
- No data is removed, but all entities are inactive and not maintained anymore
- A disabled tenant can be re-enabled by an admin at any time

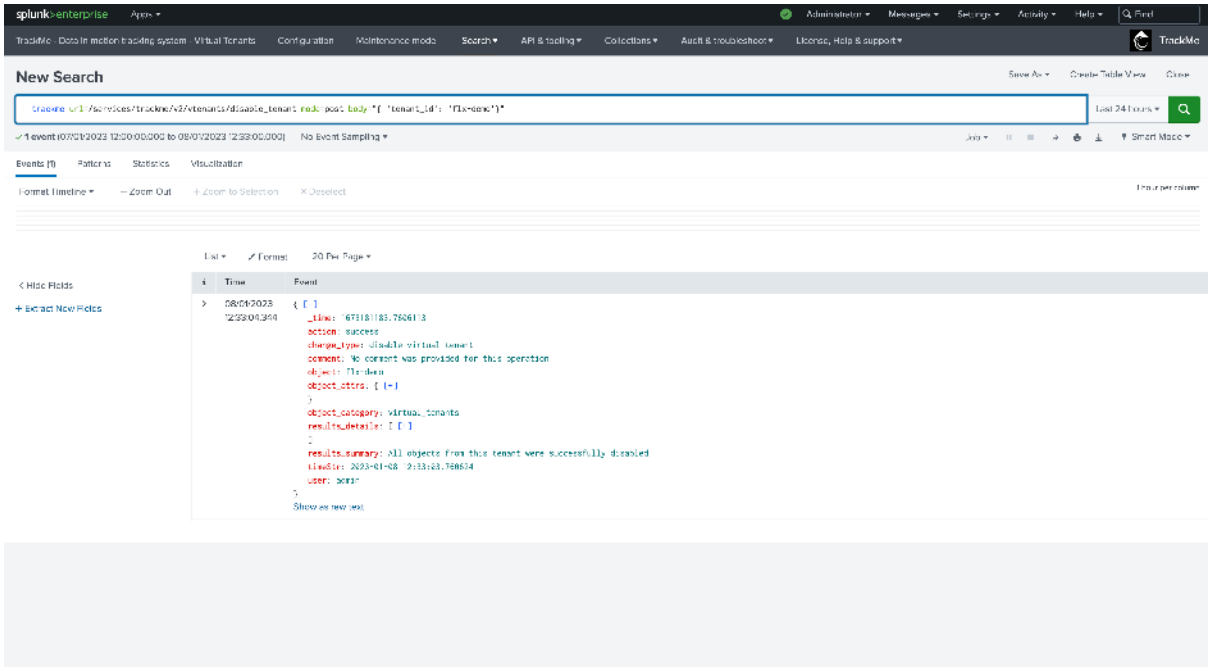






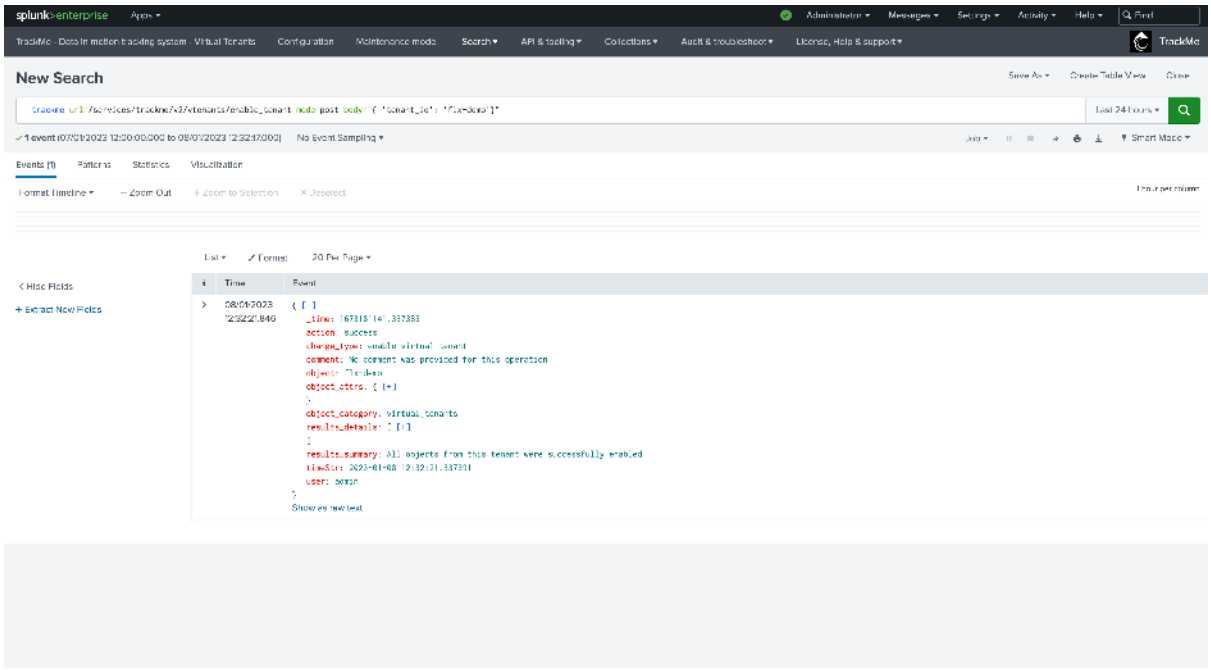
You can disable a tenant using the REST endpoint with the following SPL command:

```
| trackme url=/services/trackme/v2/vtenants/admin/disable_tenant mode=post body="{
↪'tenant_id': 'flx-demo'}"
```



You can enable a tenant using the REST endpoint with the following SPL command:

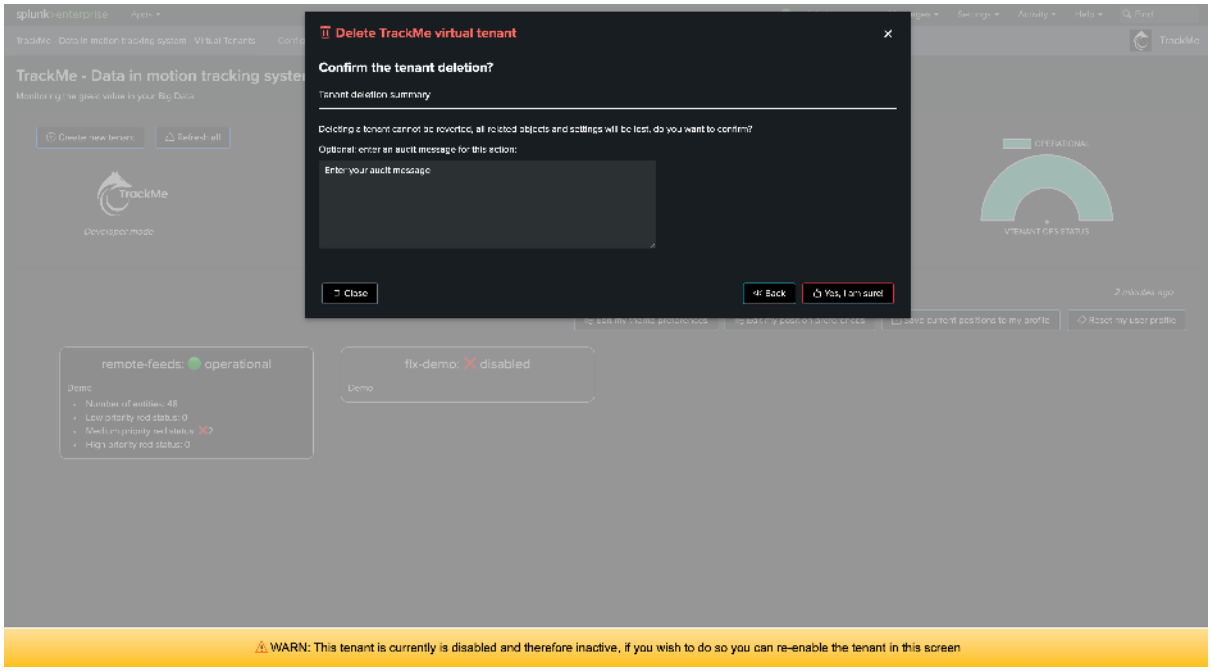
```
| trackme url=/services/trackme/v2/vtenants/admin/enable_tenant mode=post body="{
↪'tenant_id': 'flx-demo'}"
```

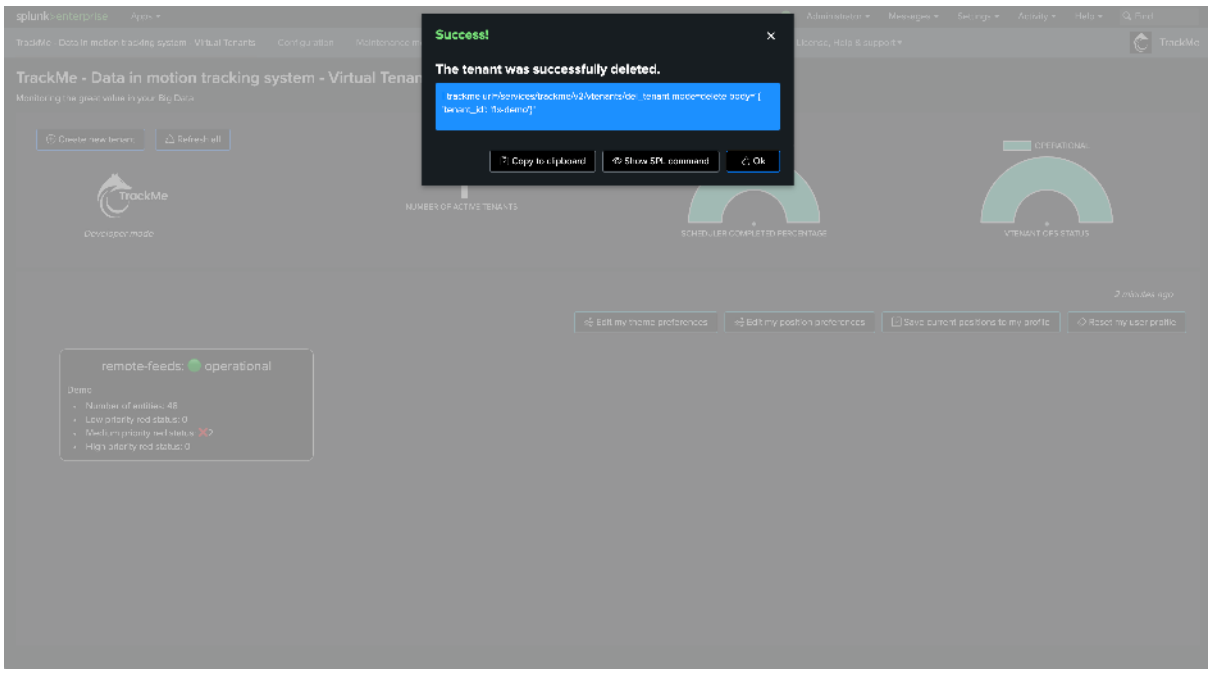


Deleting a tenant

When deleting a tenant:

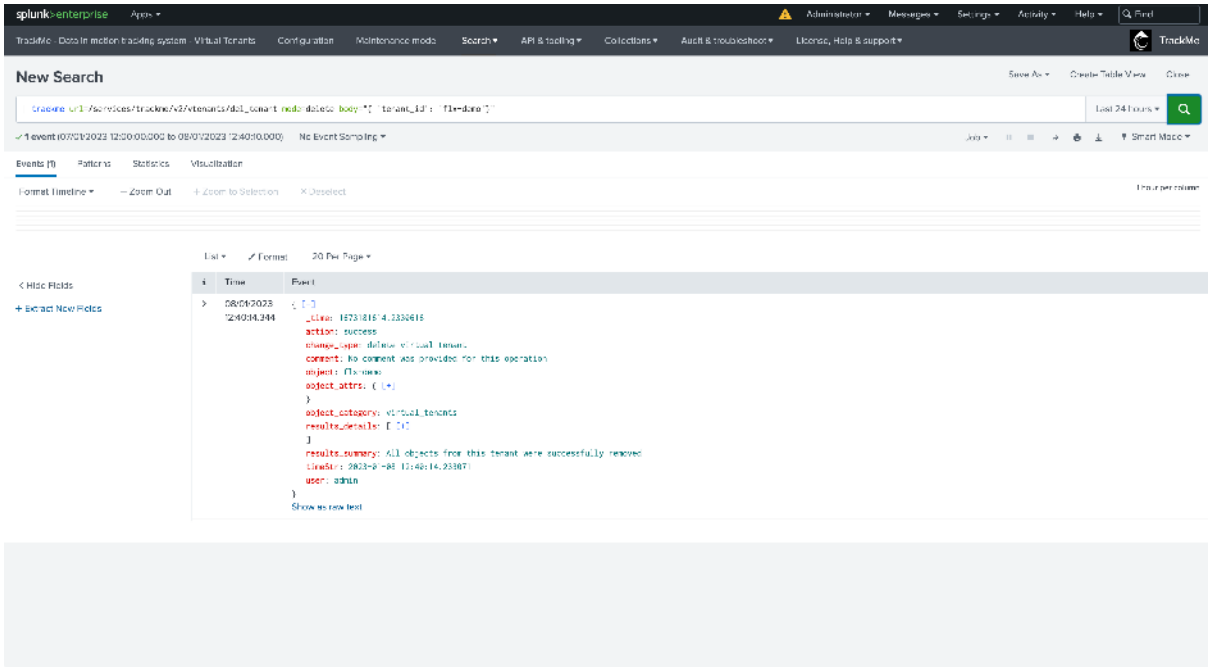
- All of its related objects are permanently deleted
- This operation cannot be undone, KVstore collection are flushed and could only be restored from backups, objects such as Hybrid trackers would need to be re-created
- Once a tenant has been deleted, there are no traces of the tenant left apart from the previously generated events and metrics which are left untouched





You can delete a tenant using the REST endpoint with the following SPL command:

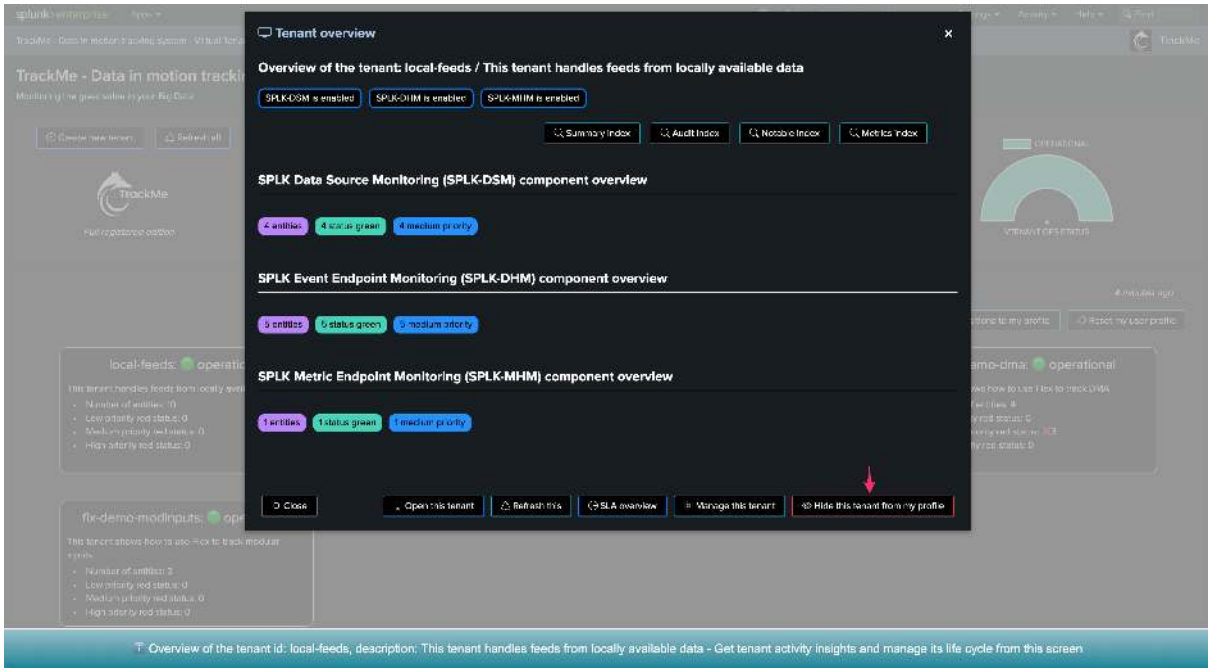
```
| trackme url=/services/trackme/v2/vtenants/admin/del_tenant mode=delete body="{
↪'tenant_id': 'flx-demo'}"
```

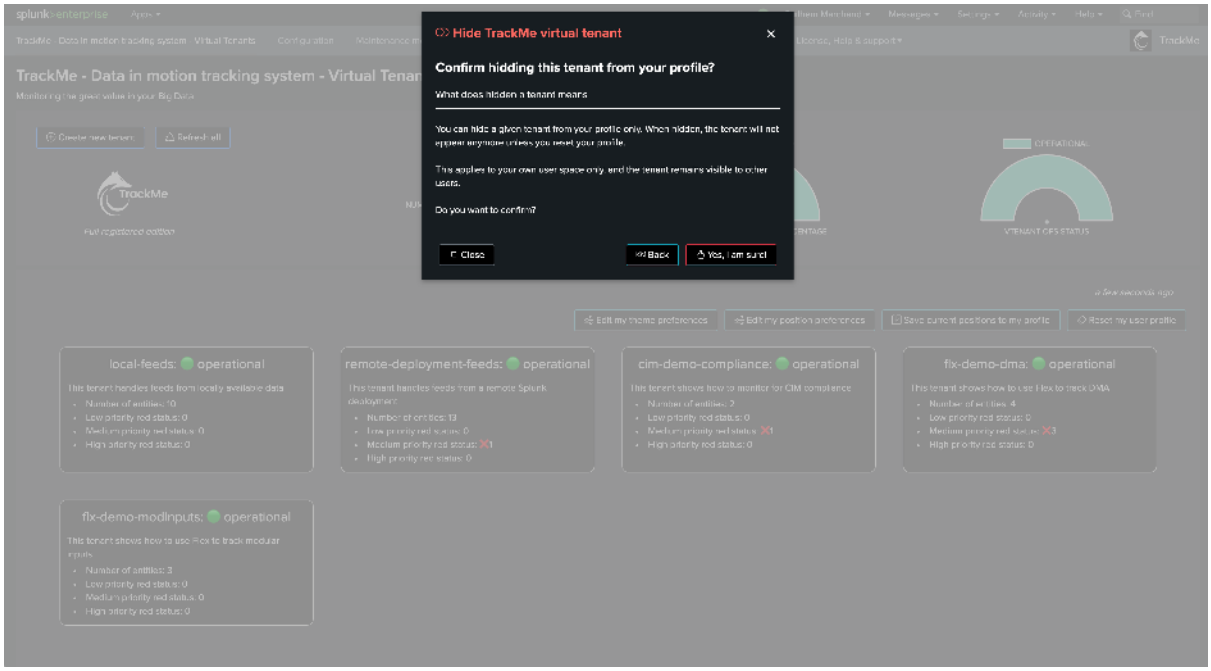


7.5.3 Hiding a tenant from your personal profile

You can use this option to permanently hide a given tenant from your personal profile:

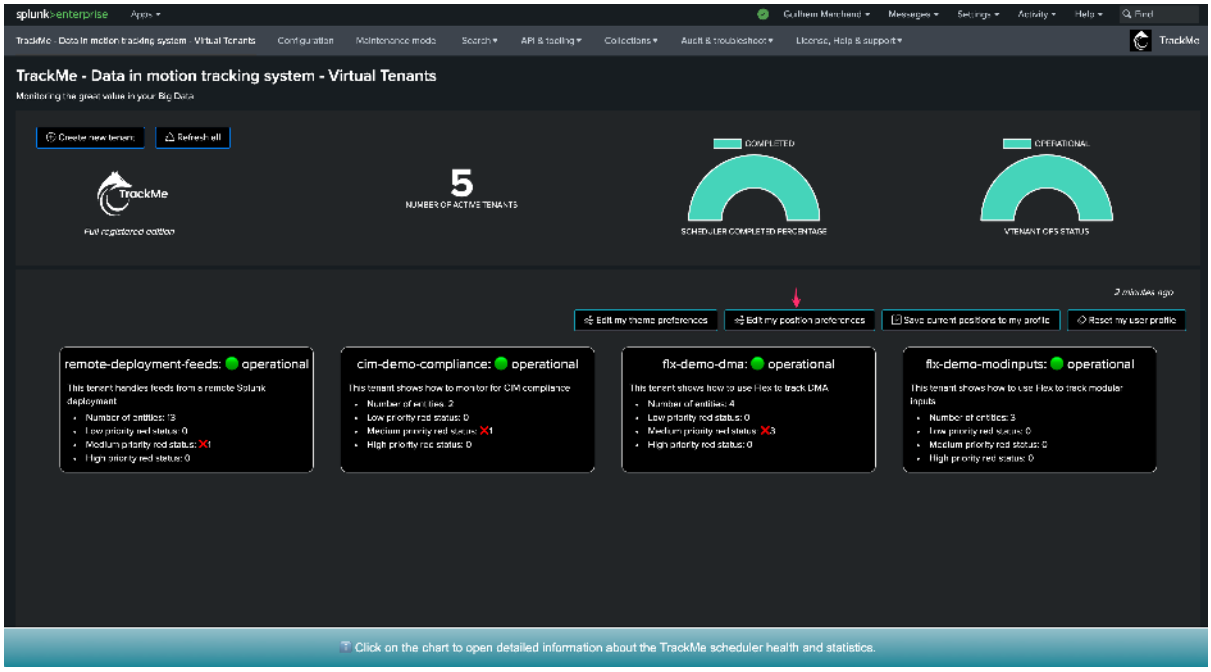
- Other TrackMe users are unaffected
- One or more tenants can be hidden
- You can modify this at any time, and refer to the default configuration if wanted

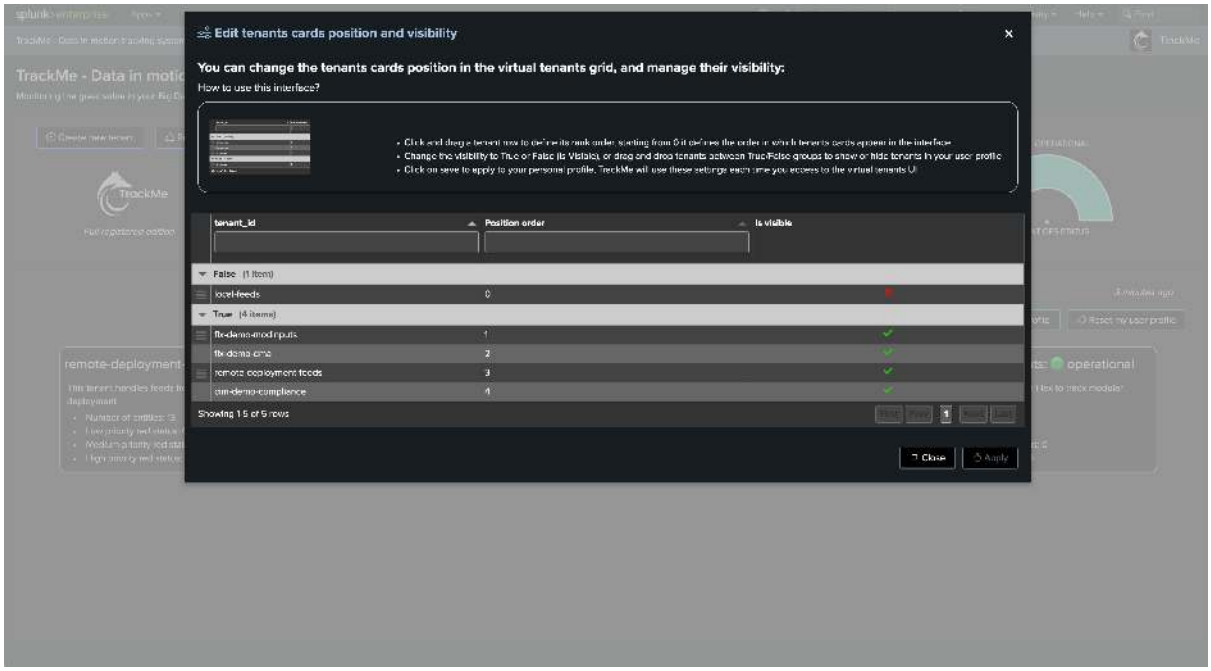




Once a tenant has been hidden, it won't appear any longer from the Virtual Tenants user interface for your own profile, the tenant life cycle itself is obviously left untouched.

To access your personal tenant setting preferences, click on the button "Edit my position preferences" from the Virtual Tenants user interface:

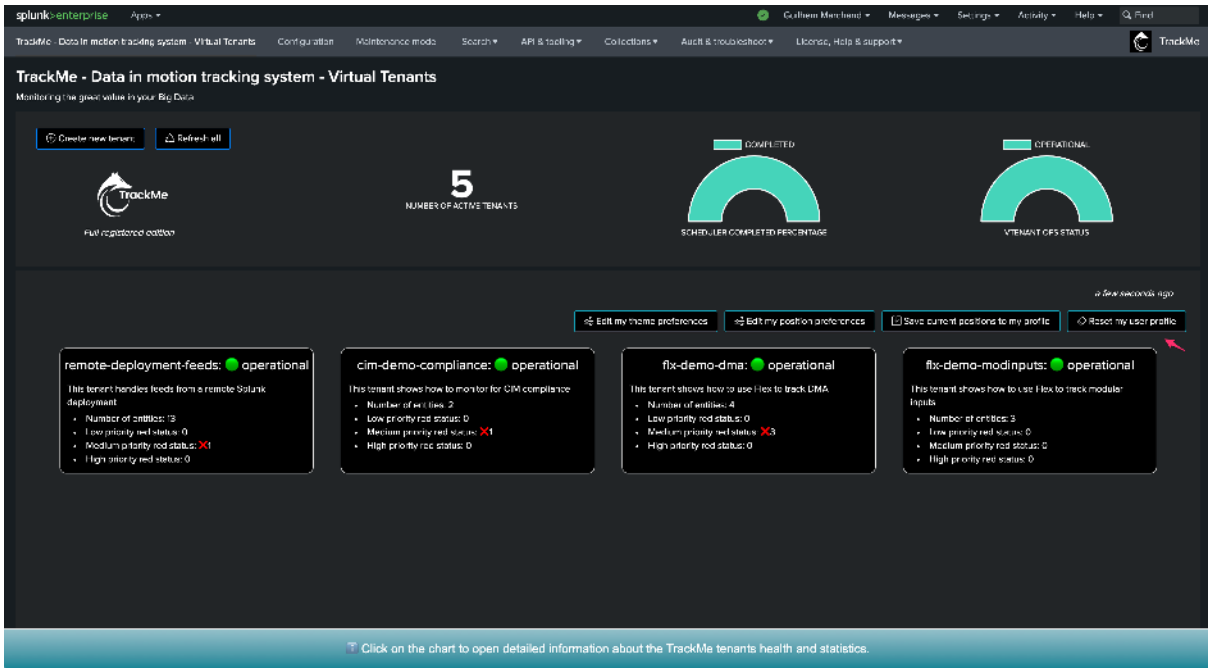




From this screen, you can:

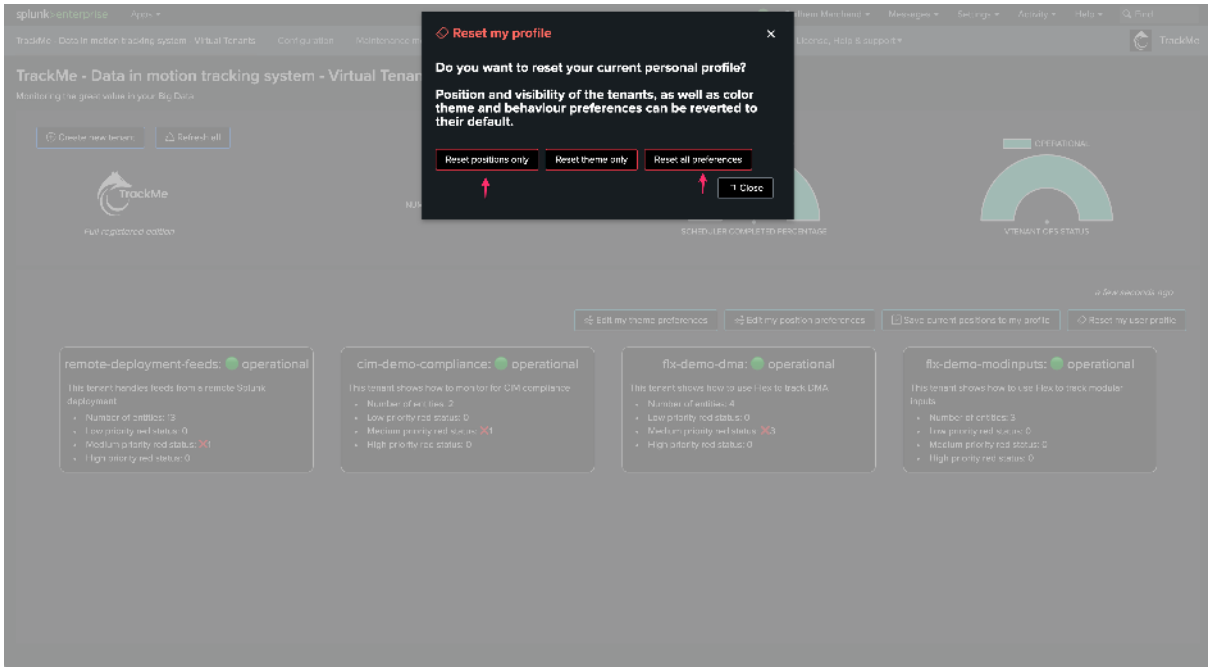
- re-order the tenants flex boxes upon your preferences
- show or hide any available tenants accordingly

If you wish to reset your tenant preferences, click on the button "Reset my user profile" from the Virtual Tenants user interface:



You can then choose to:

- Reset only your Tenants Flex boxes preferences
- Reset only your theme preferences
- Reset all preferences

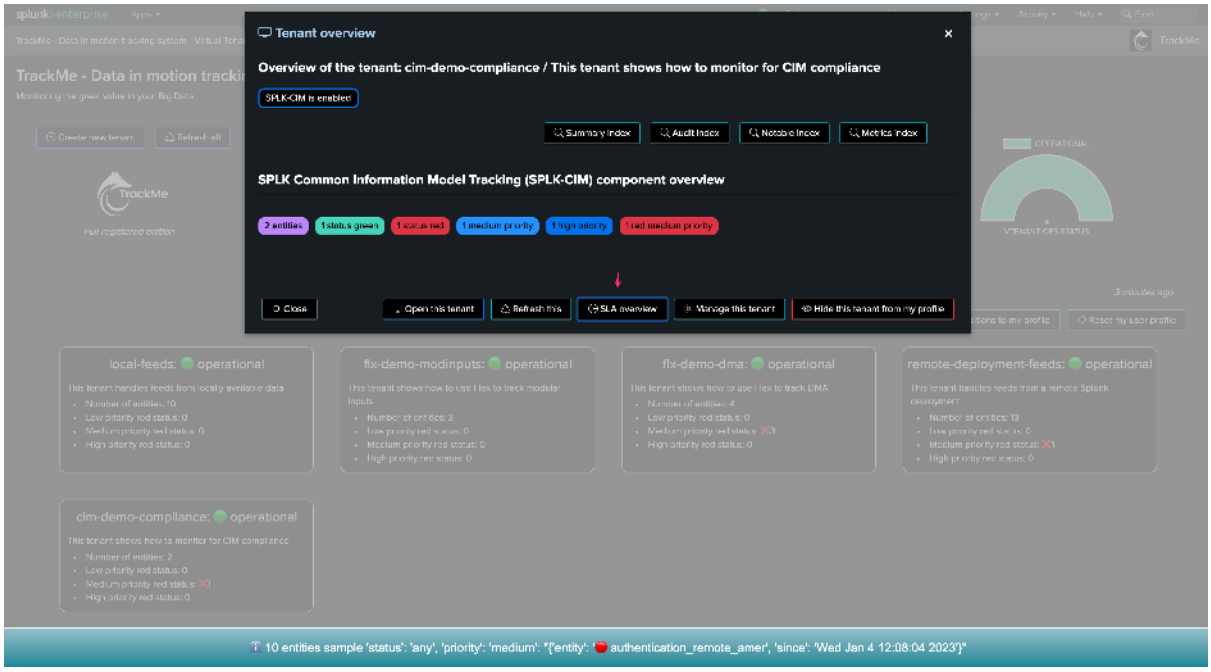


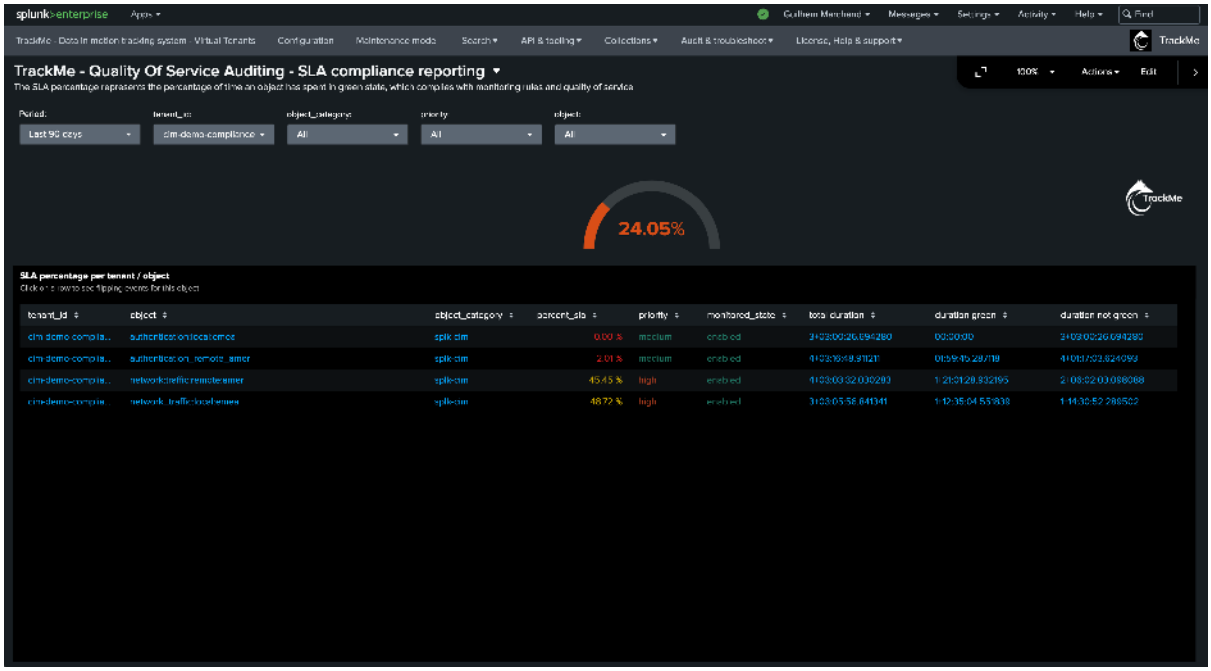
7.5.4 SLA compliance

In TrackMe, the SLA compliance purpose is to track the level of quality from the lens of the percentage of time spent in green state for a given entity.

This Key Performance Indicator (KPI) is a concept that you can then rely on to report the level of quality of the Splunk data availability according to your context.

To access the SLA dashboard from a Virtual Tenant, click on the "SLA overview" button from the tenant welcoming screen:





7.6 Operational Status Virtual Tenants

7.6.1 What is the Ops Status for Virtual Tenants

Hint

TrackMe Virtual Tenants are themselves continuously monitored to report any kind of failures that would be encountered by any of the application trackers.

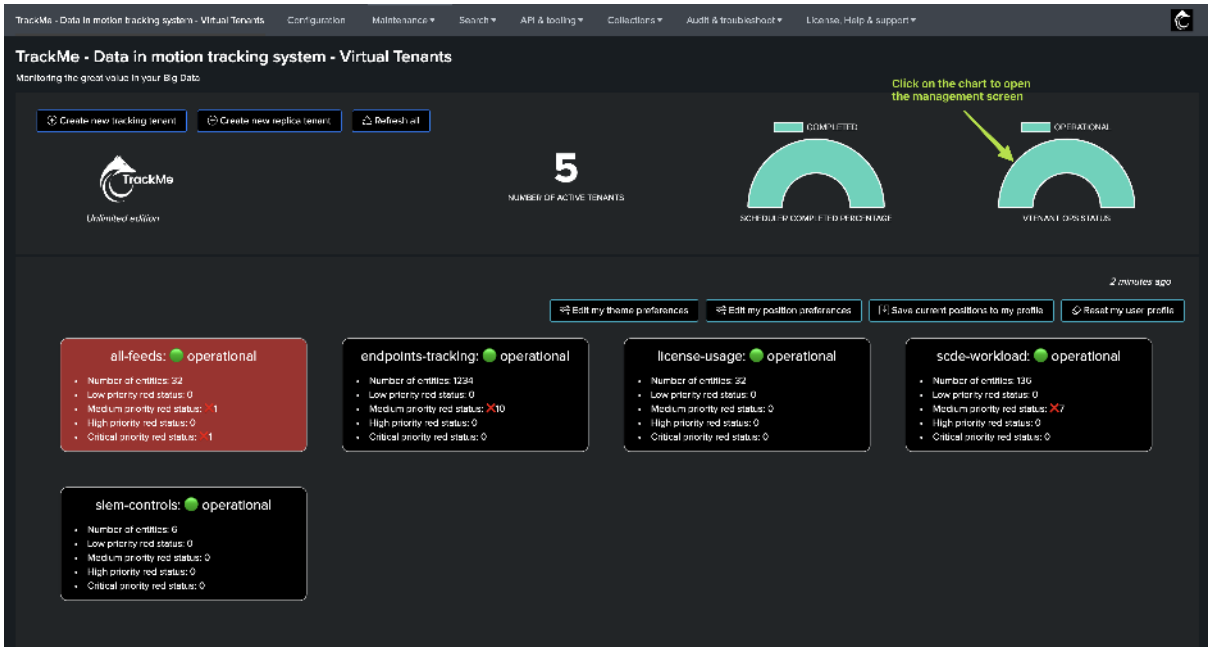
TrackMe trackers can fail when technical issues are encountered, this can happen for various reasons, such as:

- Overloaded environments, where there are not search slots available (although TrackMe has a concept of automated re-attempts)
- Search Head is in detention (overused file-systems, etc)
- Network connectivity issues, credentials issues
- Corrupted configuration (ex: missing Knowledge Objects or incorrect permissions)
- And many more

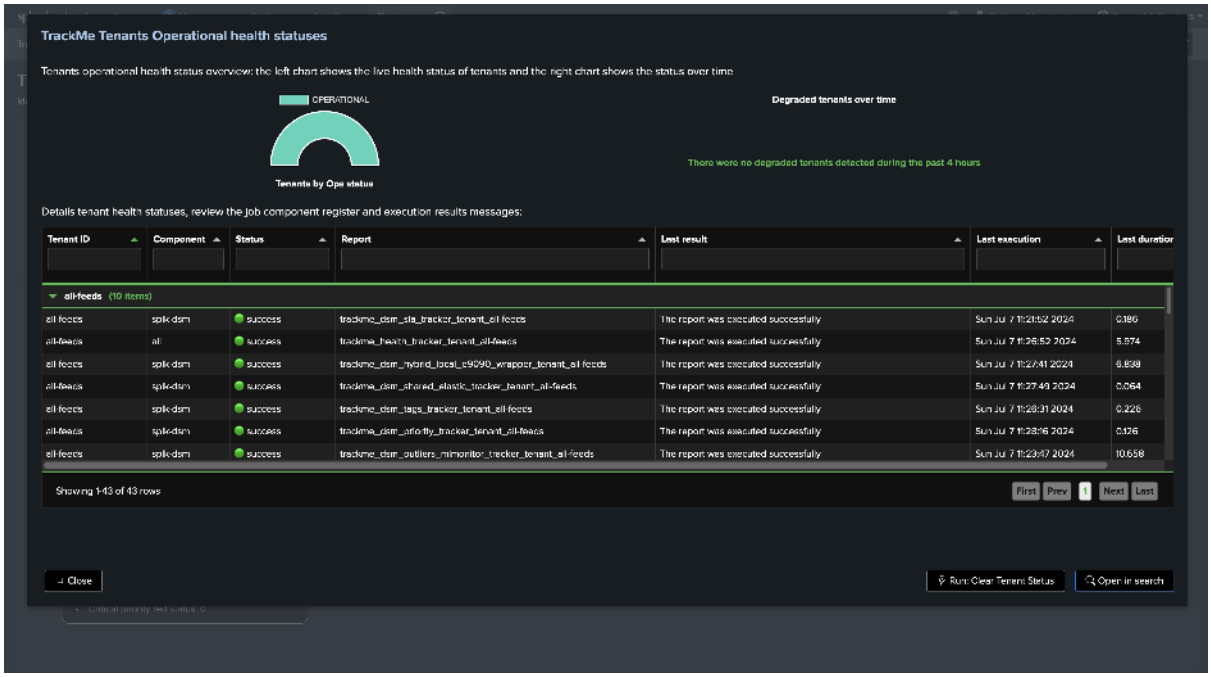
With the goal of providing the highest visibility and quality, every Tracker and related custom commands carefully report failures to a central component called the components register.

In addition, when a Virtual Tenant is created, a Health Tracker is created which investigates the component register results, reports and generates Health events continuously.

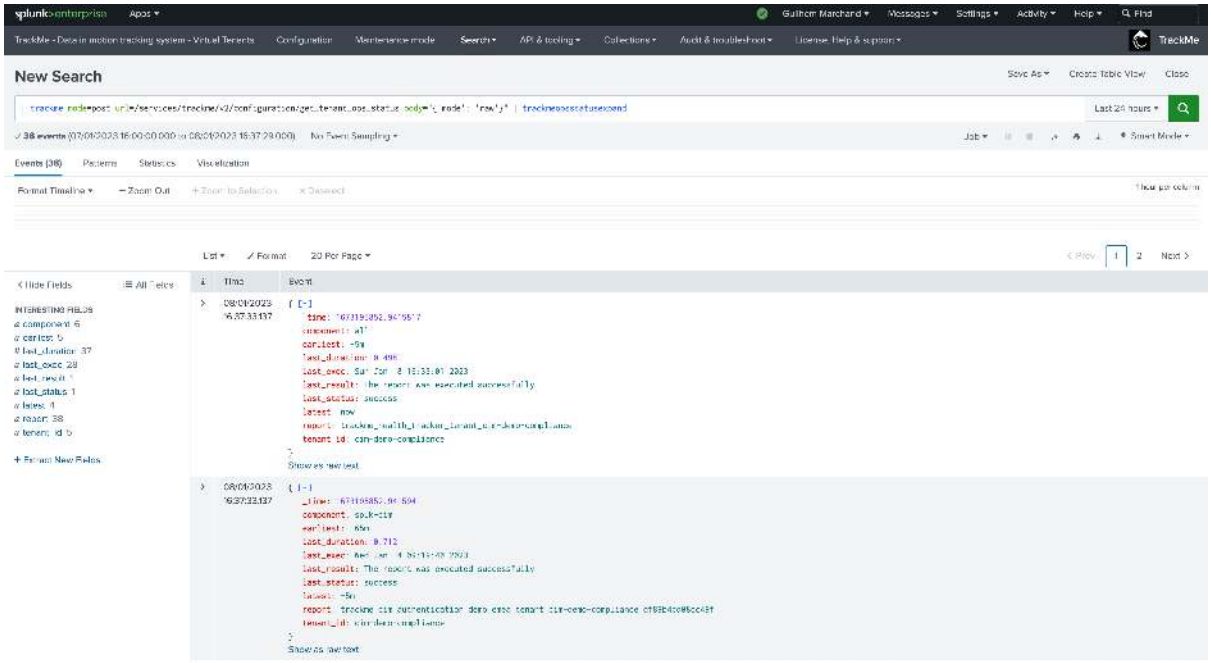
You can easily observe the operation status of TrackMe Virtual Tenants in the Virtual Tenants user interface:



Click on the chart to open the detailed view:

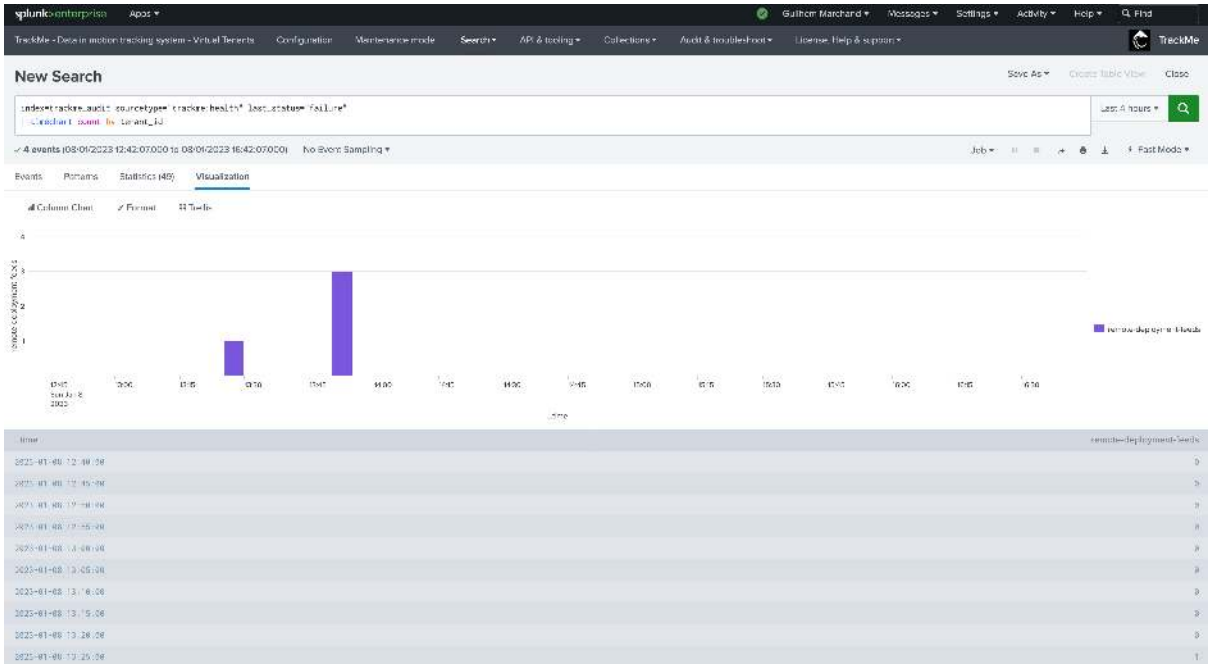


Click on "Open in search" to access detailed per tracker results in the Splunk Search user interface:



Health events are stored in the audit index(es) of the Virtual Tenant(s), you can search for Health failures events easily, such as:

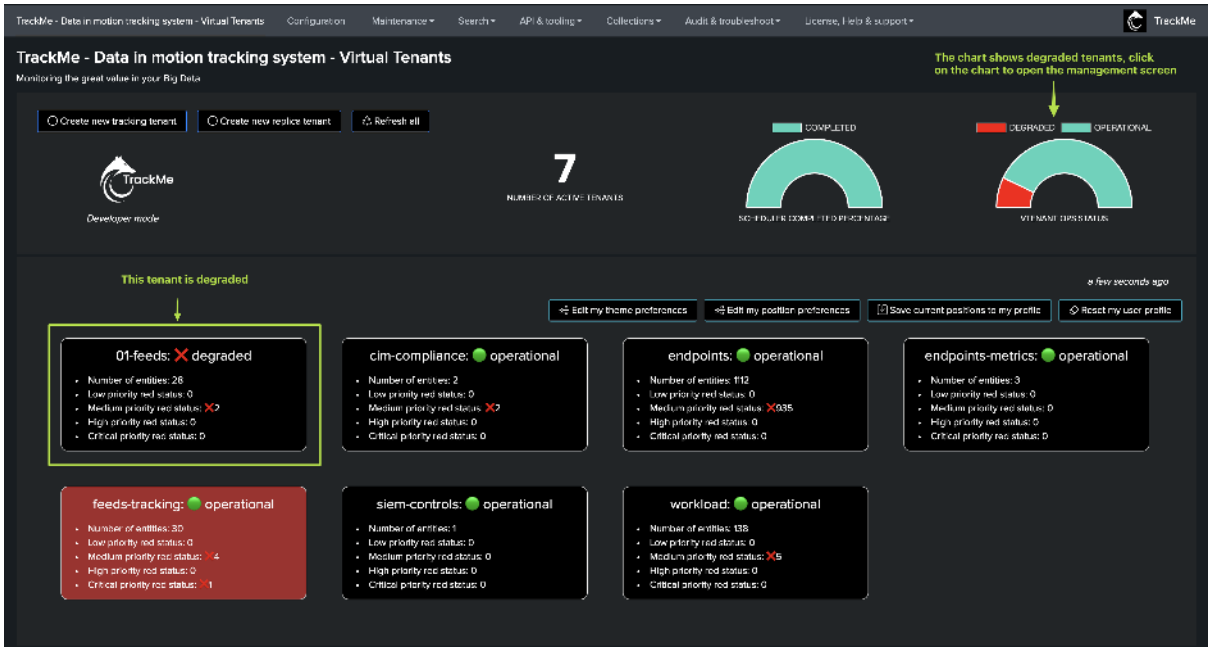
```
index=trackme_audit sourcetype="trackme:health" last_status="failure"
| timechart count by tenant_id
```



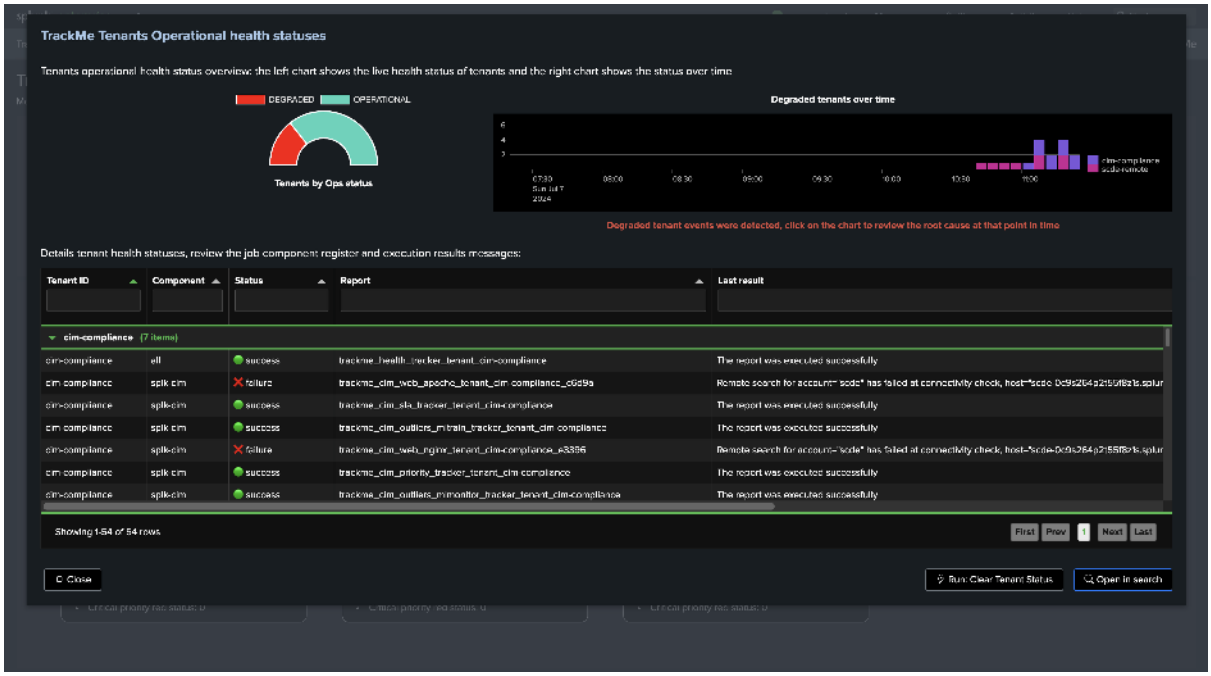
7.6.2 Example of Failure Detection

In this example, a remote Splunk Deployment is encountering issues due to a file-system being filled over the accepted limit by Splunk, therefore this instance does not accept searches any longer.

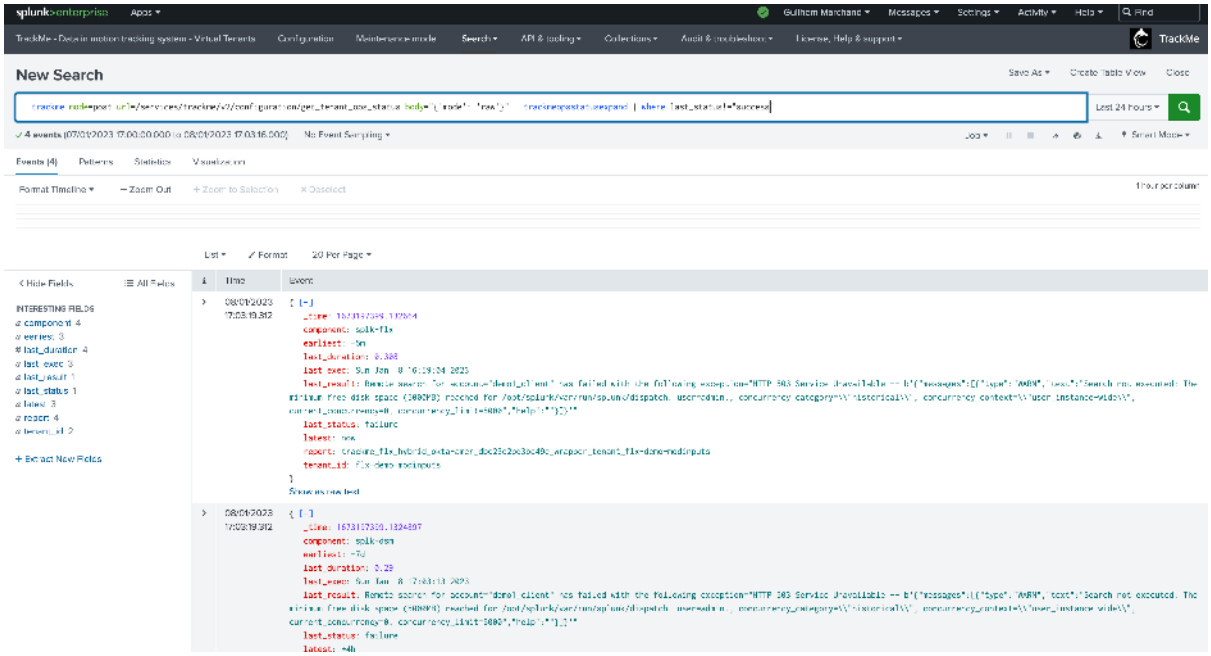
We can observe the Vtenants Ops status reporting issues currently, as well as which Virtual Tenants are affected currently:



Accessing the detailed Ops status view quickly shows the affected tenants, related accounts and technical root cause:



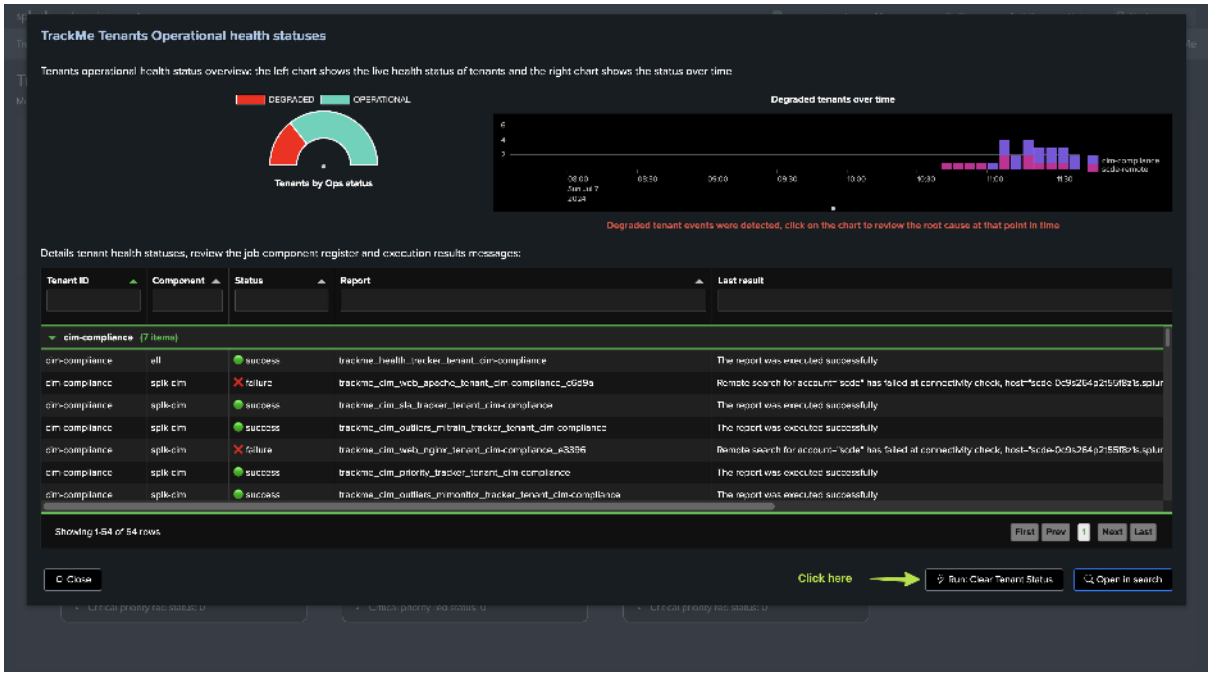
We can access the live status from REST:



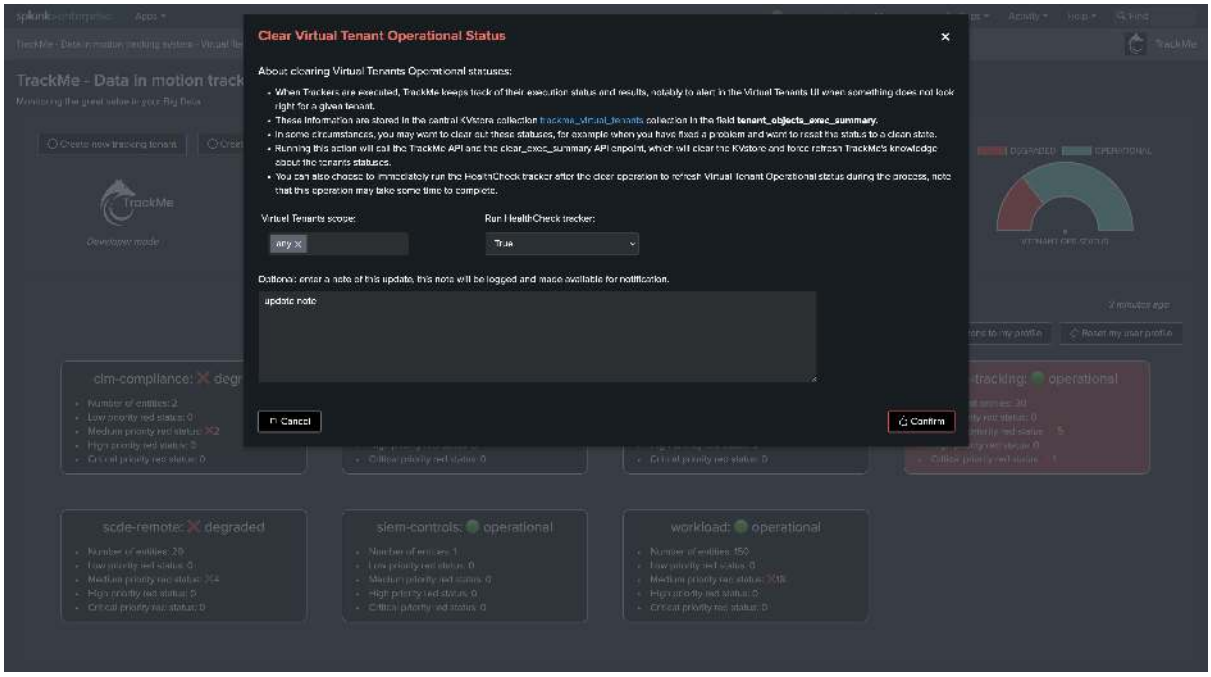
7.6.3 Clearing the Virtual Tenants Operational Status

From TrackMe 2.0.97, you can clear the Virtual Tenants Operational Status via the Virtual Tenants UI and from the REST API:

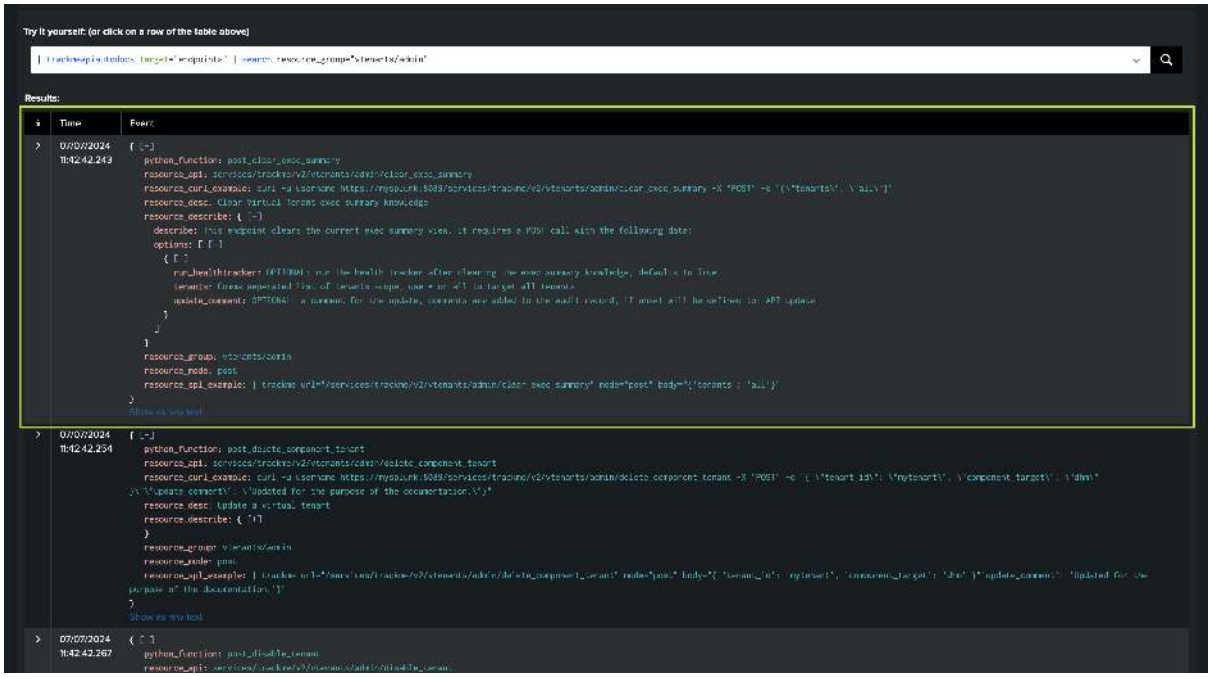
Access to the screen TrackMe Tenants Operational health statuses:



You can then request to clear for all tenants, or a selection of tenants and you can also choose to execute or not the health tracker during the clear process:



Underneath, the UI calls the REST endpoint to clear the Virtual Tenants Operational Status:



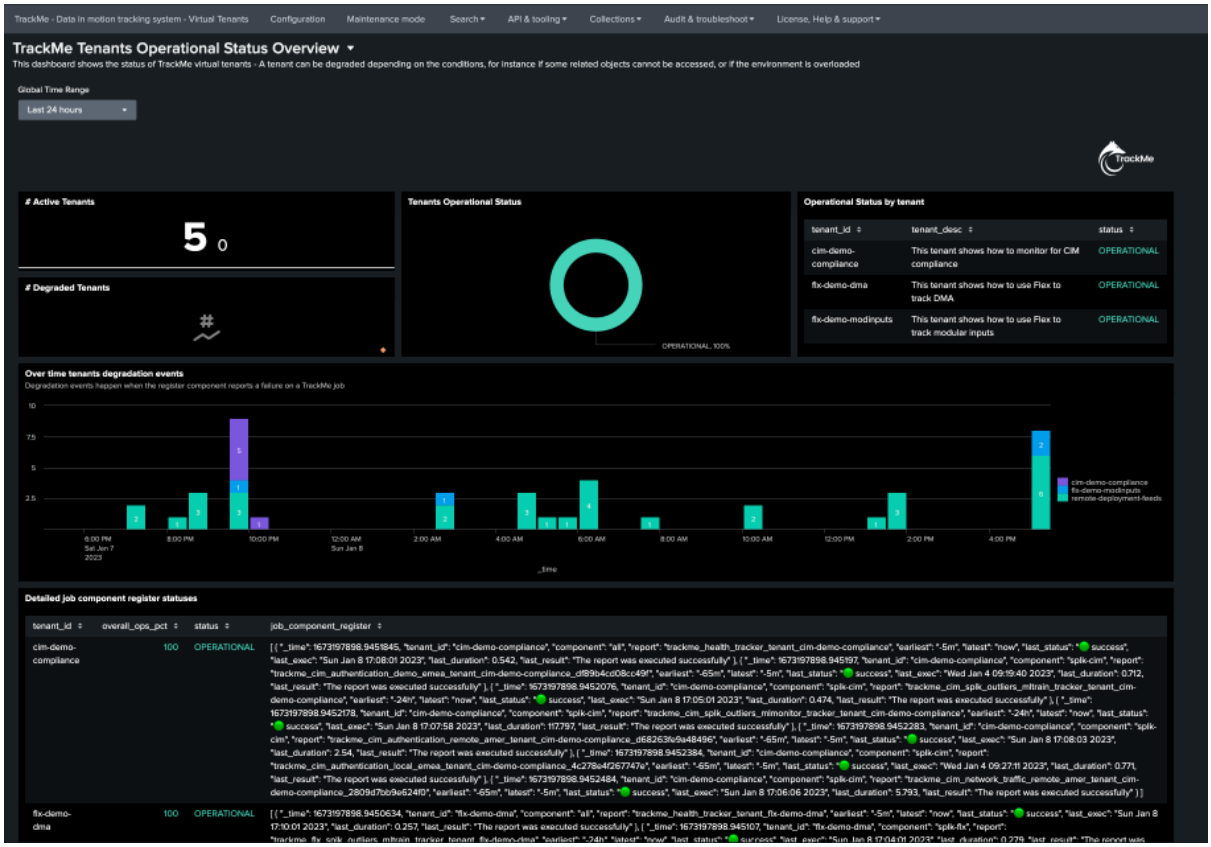
Example:

```
| trackme url="/services/trackme/v2/vtenants/admin/clear_exec_summary" mode="post"␣
↪body="{'tenants': 'all'}"
```

7.6.4 Additional resources

An additional (UFS) dashboard is available at:

- Navigation Bar / Audit & Troubleshoot / Audit - Operational Statuses



7.6.5 Virtual Tenants Ops status technical overview

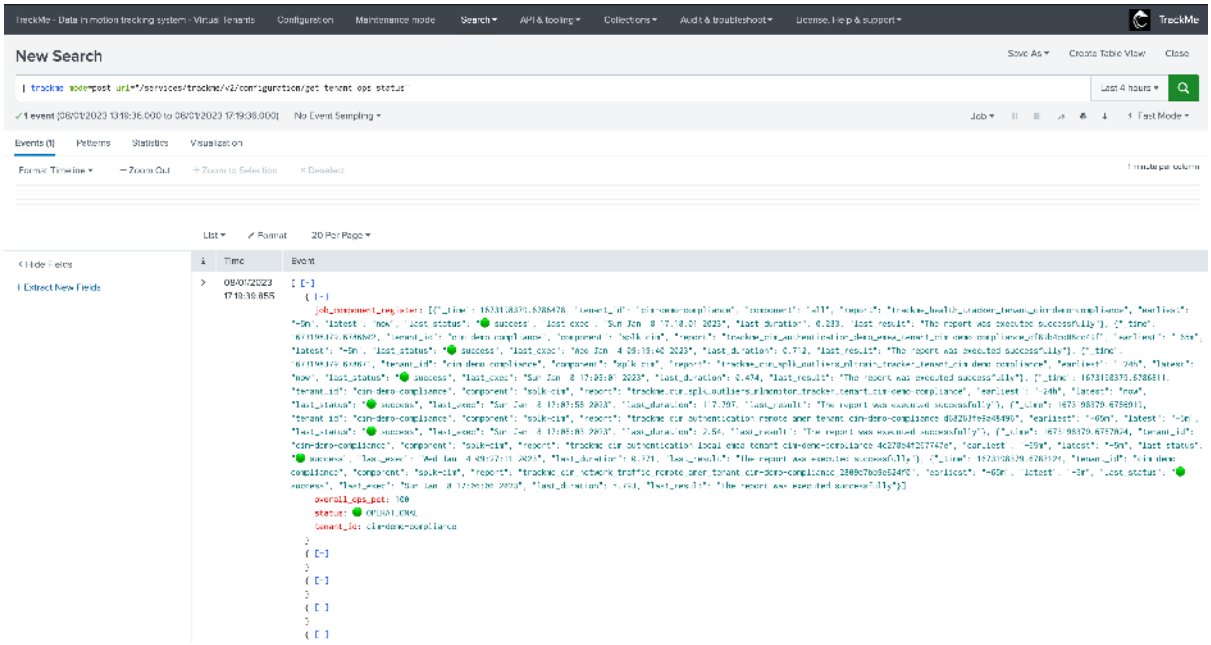
When a Virtual Tenant is created, a Health tracker is created and executed every 5 minutes:

```
trackme_health_tracker_tenant_<tenant_id>
```

This tracker executes the following custom command:

```
| trackmetrackerhealth tenant_id="<tenant_id>"
```

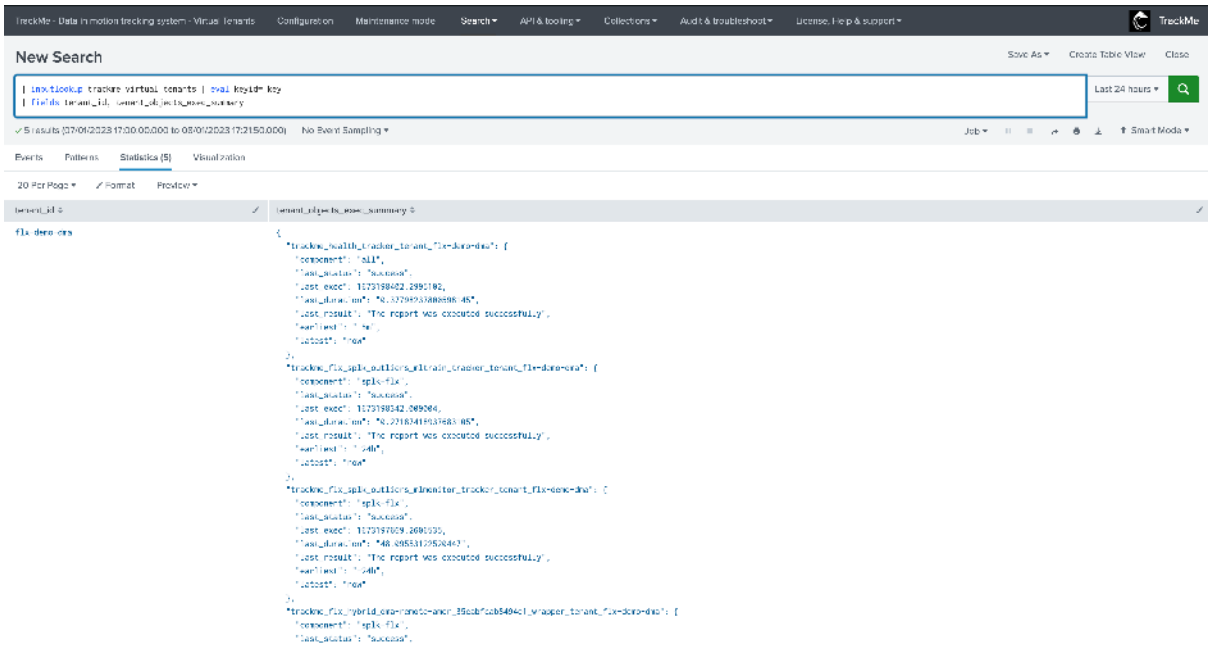
This custom command performs various operations and relies on the REST endpoint:



```
| trackme mode=post url="/services/trackme/v2/configuration/get_tenant_ops_status"
```

TrackMe components, such as Hybrid Trackers and their related custom commands call the components register at the backend level, and these states are stored in a Python dictionanry in the central store KVstore collection:

```
| inputlookup trackme_virtual_tenants | eval keyid=_key
| fields tenant_id, tenant_objects_exec_summary
```

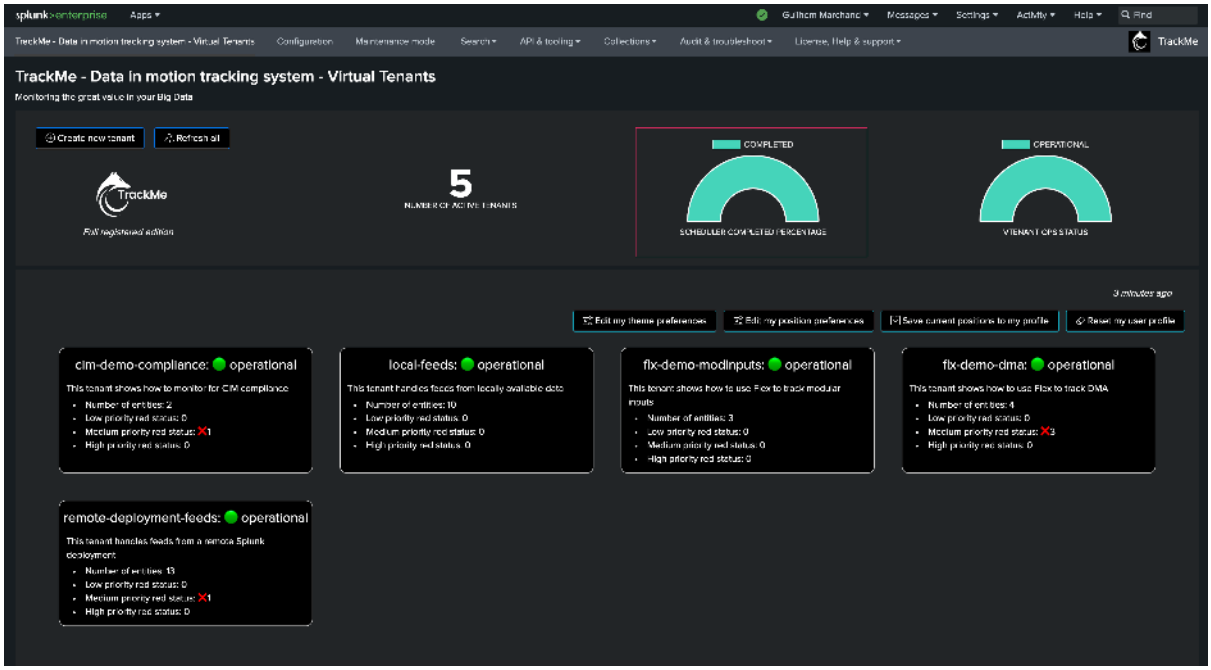


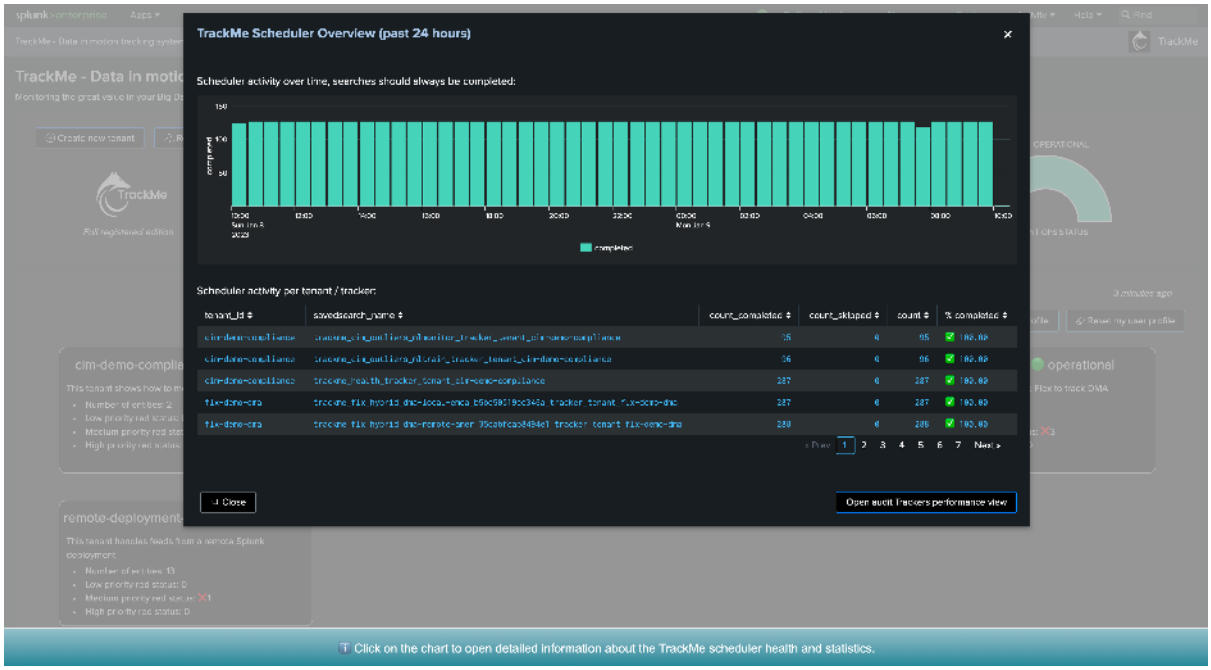
7.7 Scheduling Virtual Tenants

7.7.1 Monitoring the Virtual Tenant scheduling activity

TrackMe heavily relies on the proper scheduling and execution of its related scheduled jobs, called Trackers in the context of TrackMe.

You can easily monitor the scheduling activity from the Virtual Tenants user interface:



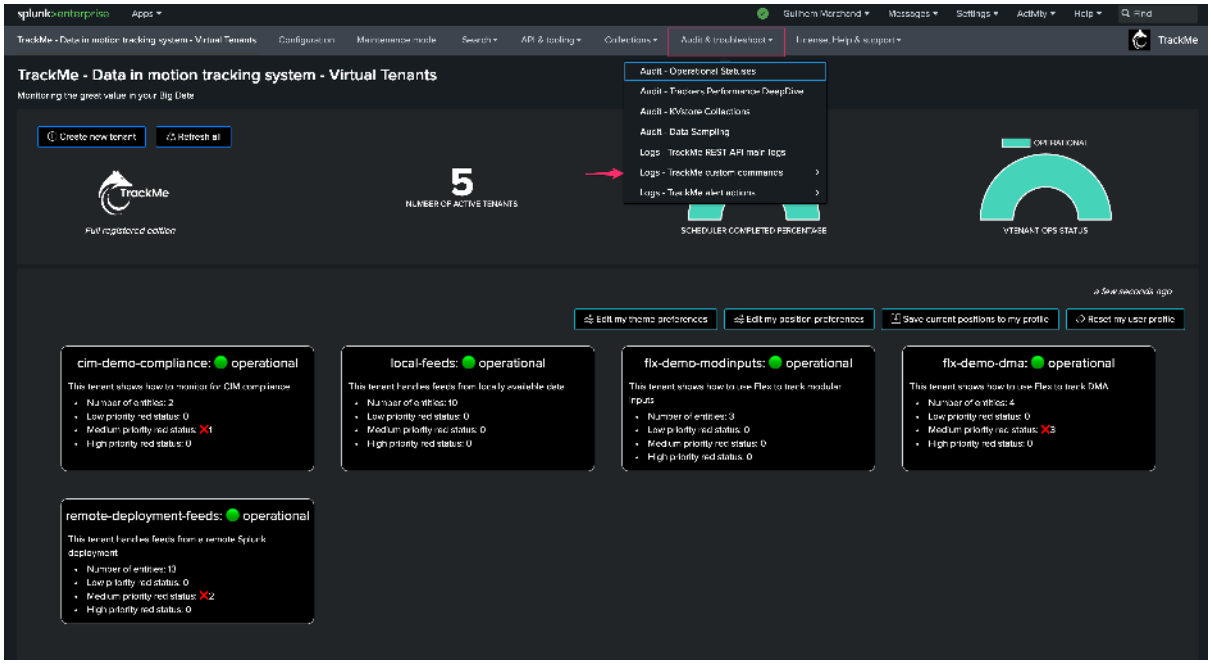


There should ideally be no skipping searches in the TrackMe deployment. TrackMe trackers carefully log their own run time performance which is leveraged in the "TrackMe Performance Deepdive" dashboard:

- Skipping searches can happen if the deployment is slightly loaded, and/or the environment is lacking compute resources (CPUs).
- If you have skipping searches, this generally indicates that you need to review the quality and the performance of the searches, or increase the compute resources for the TrackMe deployments.

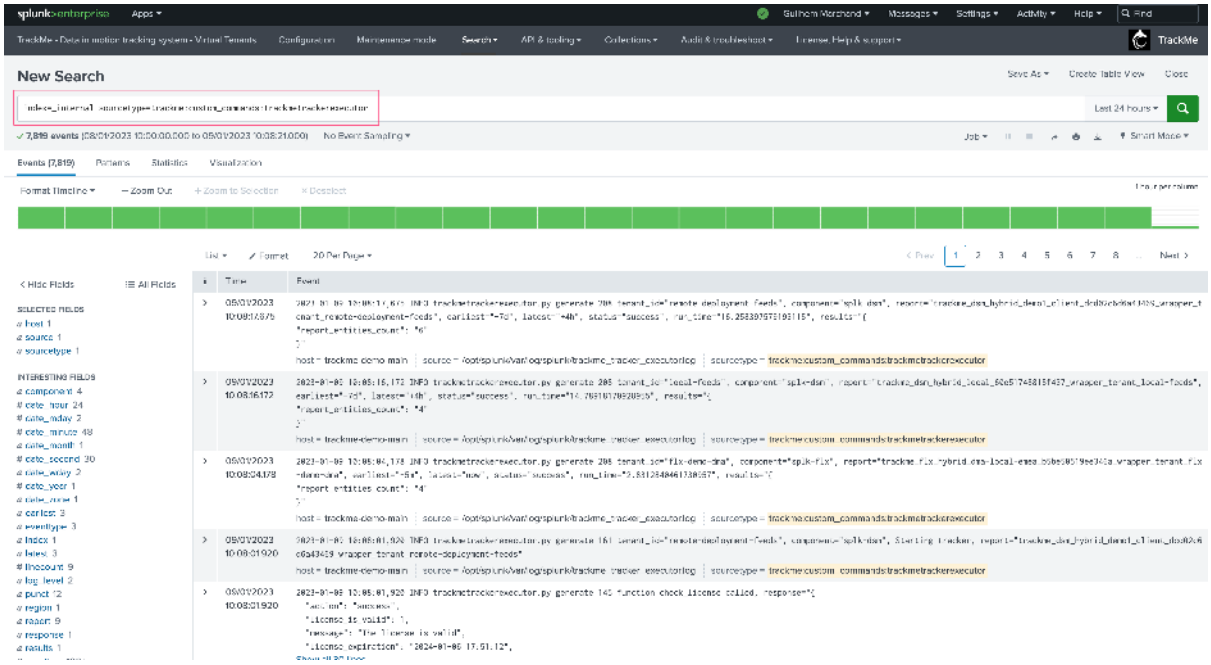
7.7.2 Trackers logging and troubleshooting

All TrackMe trackers log their own activity carefully, including their run time performance Key Performance Indicators. You can access the TrackMe internal logs through the navigation bar:

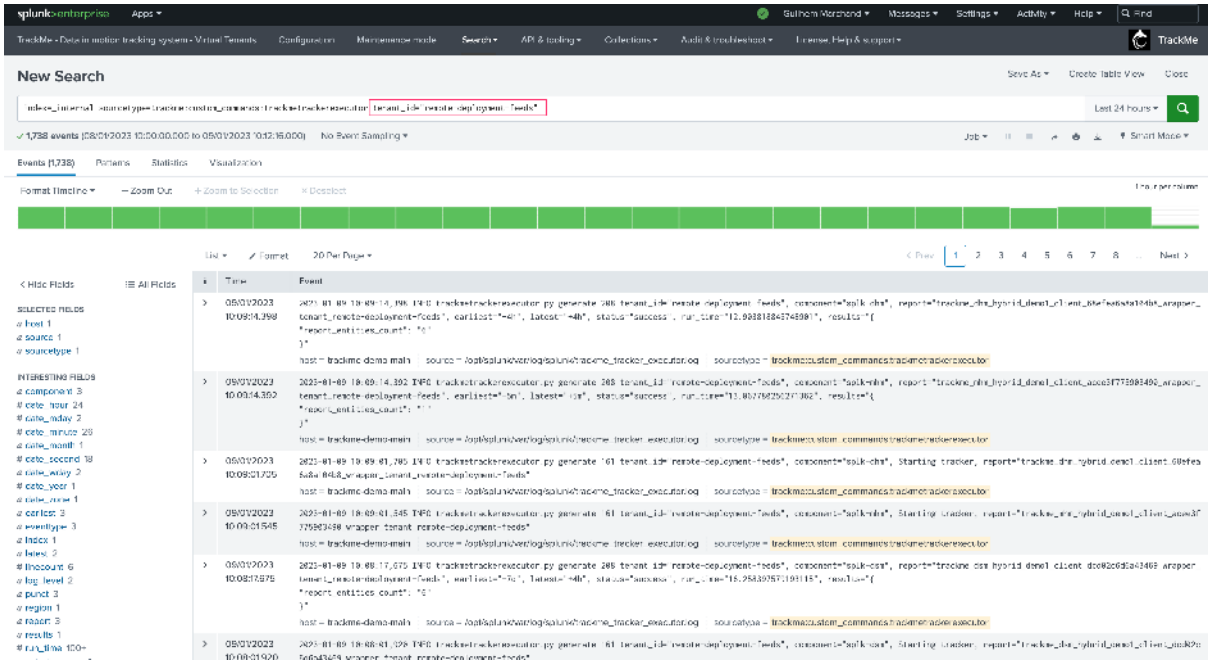


For instance, splk-feeds trackers are orchestrated by a TrackMe custom command called `trackmetrackerexecutor`:

```
index=_internal sourcetype=trackme:custom_commands:trackmetrackerexecutor
```

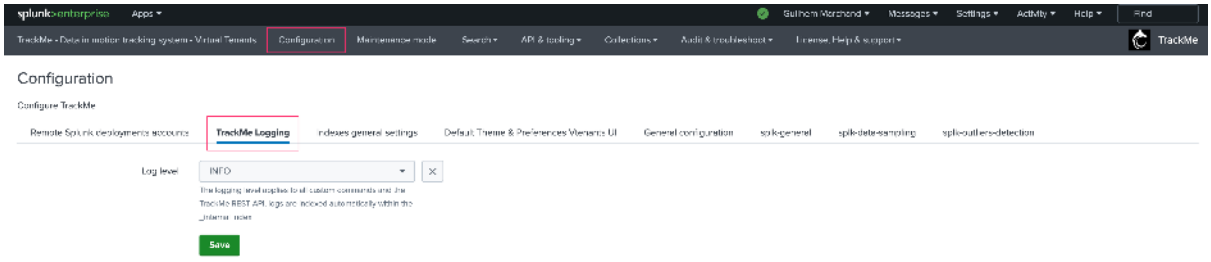


In logging INFO level, the custom command logs the tenant_id as well as its related scheduling configuration, and a result summary:



Note: In DEBUG mode, TrackMe will become very verbose. Therefore, the DEBUG mode should be activated only temporarily for debugging purposes.

You can control the logging level in the Configuration user interface:

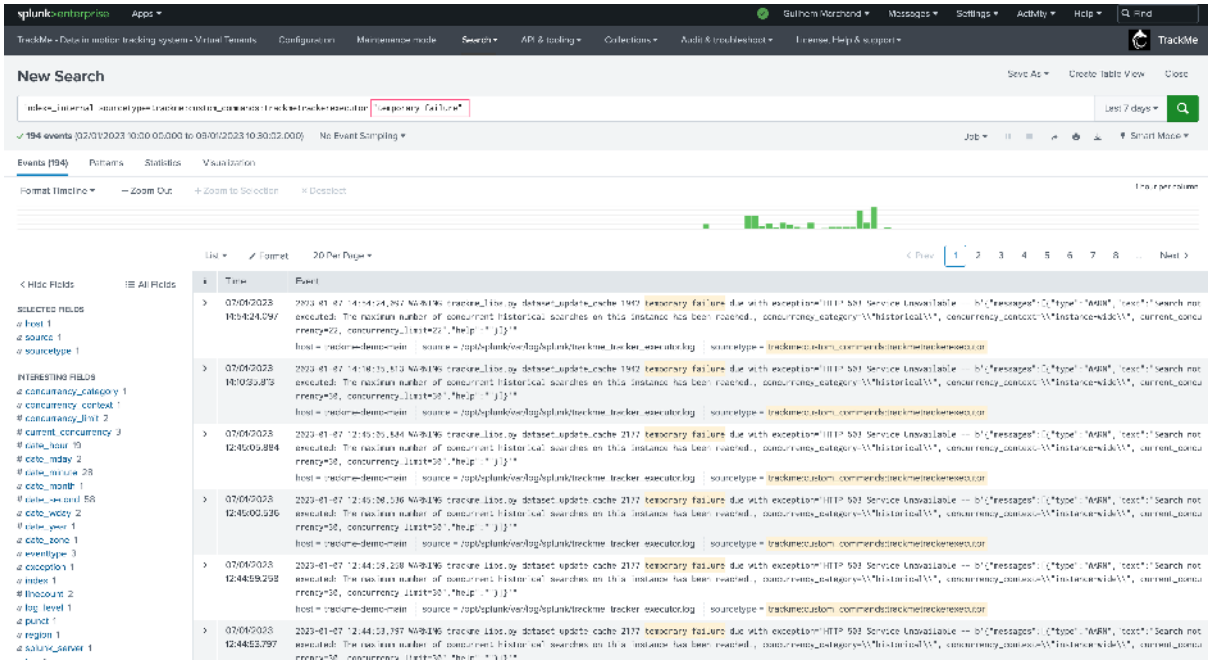


7.7.3 Automated scheduling re-attempting concept

TrackMe has a concept of automated re-attempts when it detects that a given execution cannot be performed due to max search concurrency being reached.

This can be observed in the custom command logs such as:

```
index=_internal sourcetype=trackme:custom_commands:* "temporary failure"
```



If the maximum number of re-attempts is reached, its execution will be canceled and it will log this as a permanent failure in error log level.

7.8 Personal user profile for Virtual Tenants

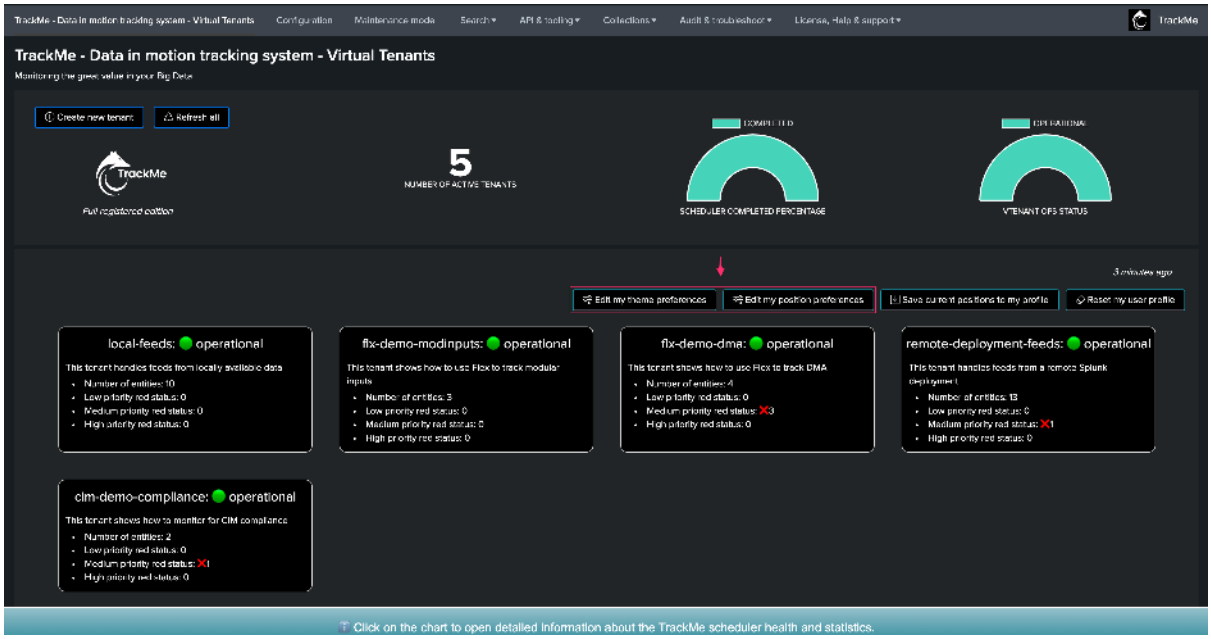
7.8.1 Per user preferences

There are two types of profiles that can be set per user, these concern:

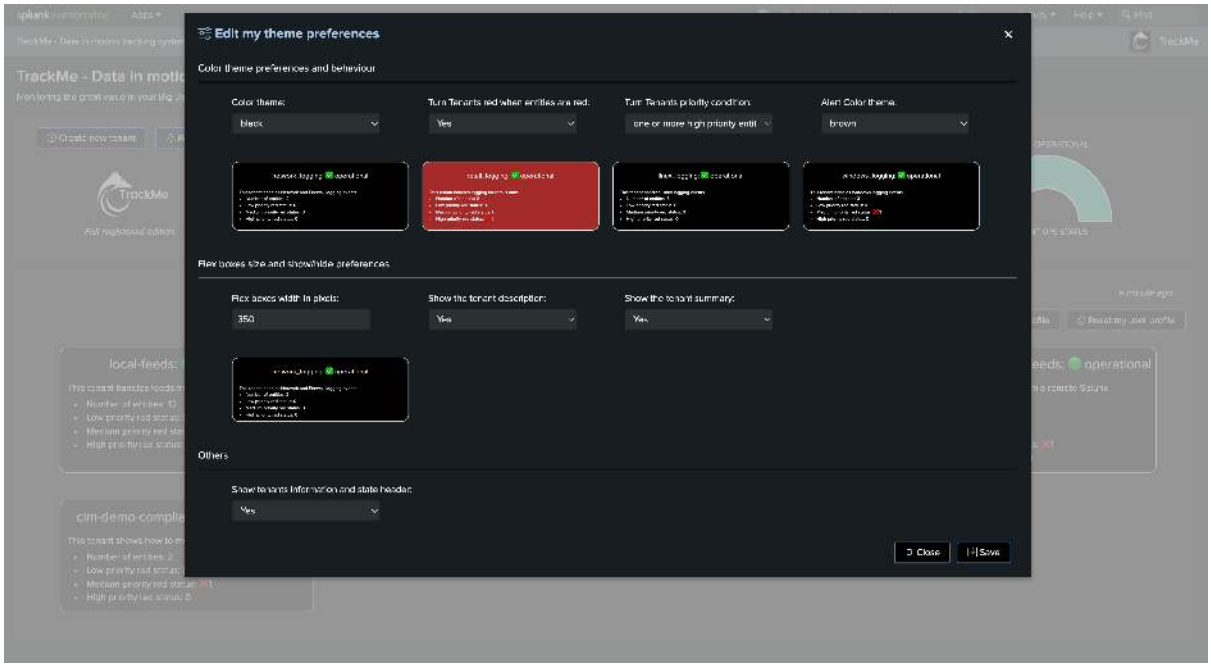
- The Virtual Tenants theme preferences
- The Virtual Tenants Flex boxes positions and visibility

Each user can define their own preferences, without affecting other users:

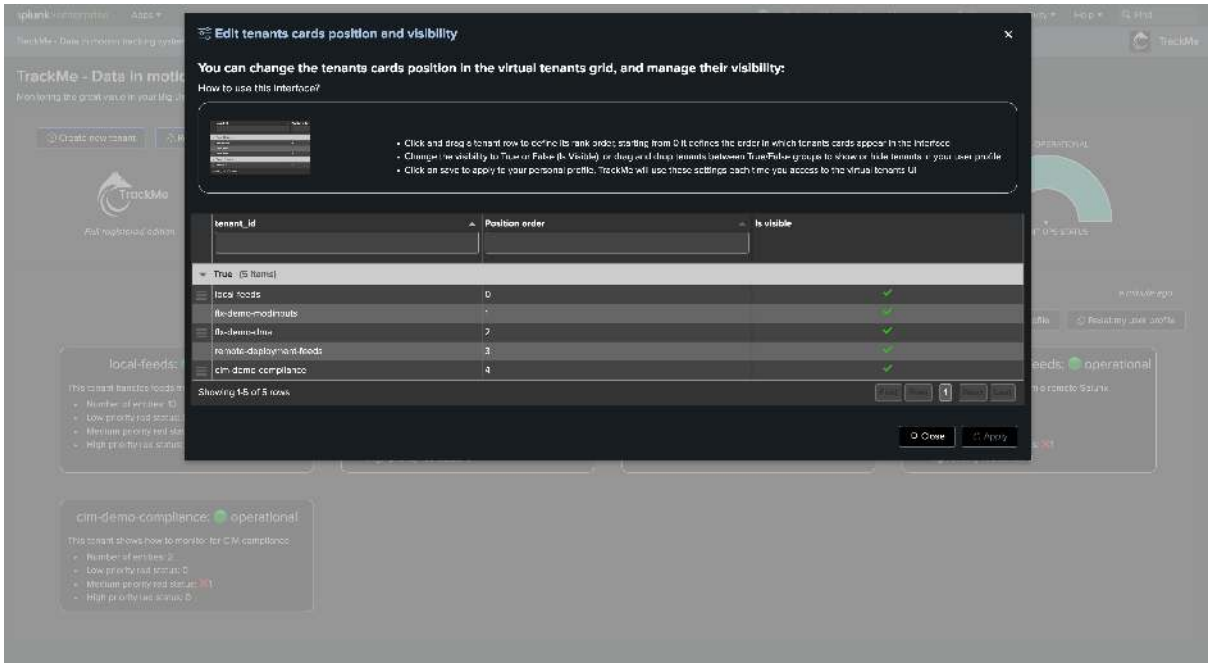
Accessing preferences setup as a user:



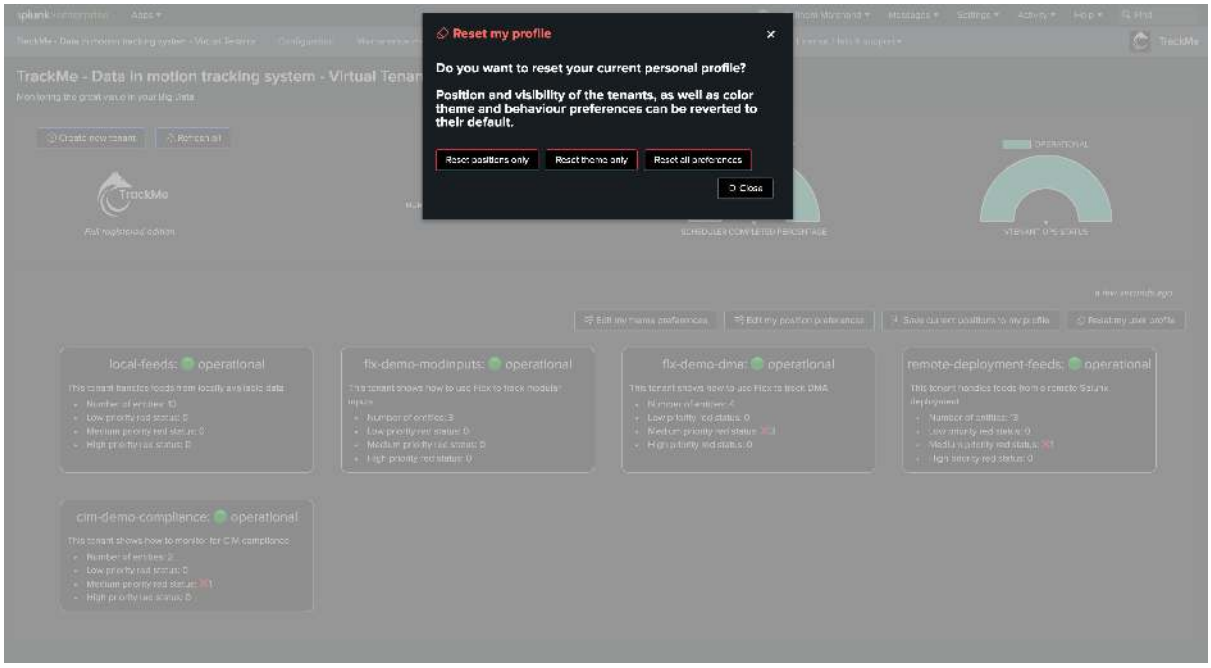
Theme preferences:



Virtual tenant position and visibility preferences:



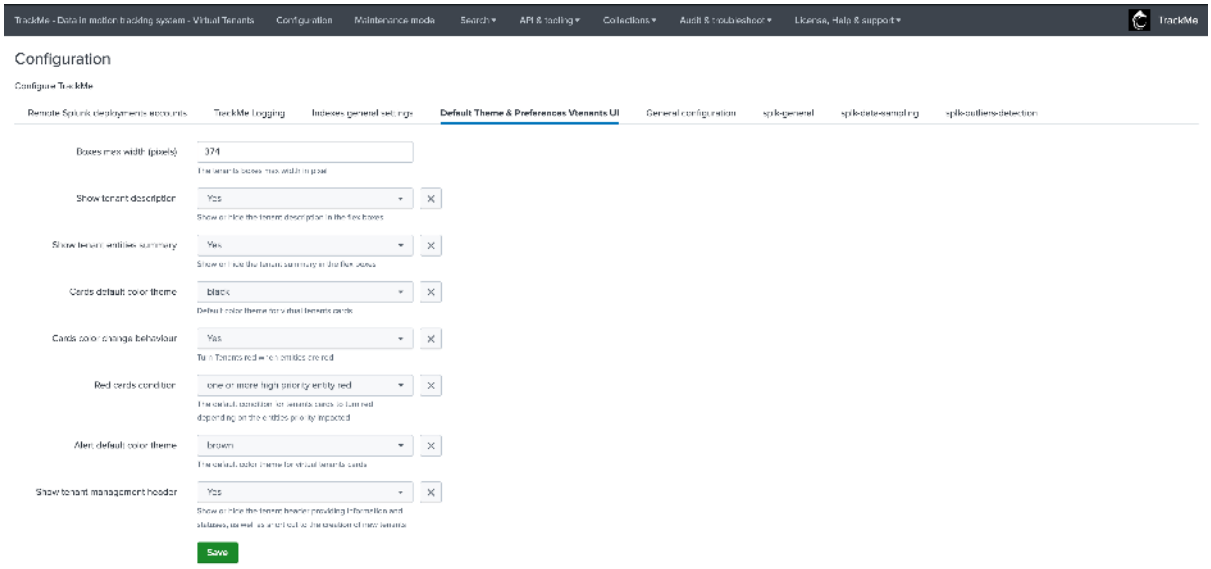
As a user, you can reset your preferences, partially or totally, if you wish to do so:



7.8.2 Application level default preferences

Default theme preferences for all users are set at the application level, and can be customized by the administrators in the TrackMe configuration user interface:

Go to Navigation bar / Configuration:



These preferences are set by default in the following configuration file and stanza:

- trackme/default/trackme_settings.conf
- configuration stanza: “trackme_theme_default”

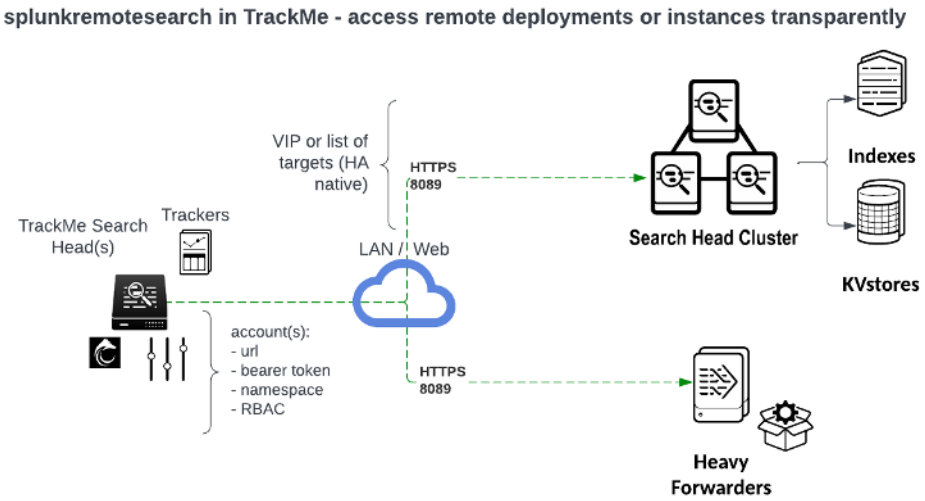
7.9 Splunk Remote Deployments (splunkremotesearch)

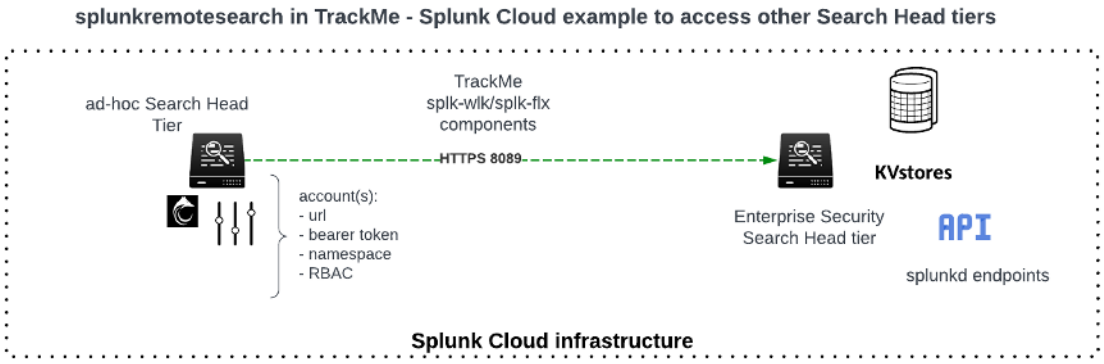
7.9.1 Overview of Splunk Remote Search Capabilities in TrackMe

TrackMe can manage locally available data, as well as data from remote Splunk deployments (Splunk Enterprise, Splunk Cloud) and remote independent instances such as utility nodes (Deployment Servers, Cluster Managers, License Manager, etc.) or Heavy Forwarders.

This is a key capability provided by TrackMe, which is used transparently when you configure trackers, allowing you to target either the local environment where TrackMe is hosted, or any other Splunk remote deployment.

There are plenty of use cases where this capability of TrackMe is game-changing, enabling TrackMe to become the single pane of glass for your monitoring and operations.





The Splunk remote search feature in TrackMe relies on multiple aspects:

- The `splunkremotesearch` TrackMe generating custom command, which relies on the Splunk Python SDK to perform searches remotely at scale
- The concept of **accounts** which allows TrackMe to store and access pre-configured configuration defining the Splunk remote environment
- Various backend-level features in TrackMe designed to identify and handle the remote entities accordingly
- The Remote deployment concept can be used for various purposes, from feeds tracking on remote deployments, to Flex objects, CIM compliance tracking, or Workload between different Search Head tiers

Note

Local service account user or SAML service account

- On the remote Search Heads tier counterpart, you can use a local user or a SAML user associated with the bearer token
- However, to use SAML, your SAML setup needs to support AQR and it needs to be configured
- Reference: https://docs.splunk.com/Documentation/Splunk/latest/Security/Setupauthenticationwithtokens
- Reference: https://docs.splunk.com/Documentation/SplunkCloud/latest/Security/SAMLConfigJWT

Hint

TrackMe supports multiple REST endpoints per account, with High Availability and Disaster Recovery capabilities: (from version 2.0.22)

- For each account, you can specify a comma-separated list of REST endpoints per account
- TrackMe will automatically verify the connectivity and randomly choose a REST endpoint target among any reachable endpoints
- Therefore, you can specify multiple endpoints to support high availability and distribution of the searches against multiple endpoints automatically
- This capability can be useful, for example, if you deal with a Search Head Cluster and cannot use a load balancer or VIP to access the cluster

Hint

TrackMe supports and requires Role-Based Access Control per Remote account defined: (from version 2.0.34)

- For each account, you need to specify a comma-separated list of Splunk roles that are allowed to use this Remote account
- For retro-compatibility purposes, if the roles for an existing account have not been set up yet, TrackMe will use built-in roles in addition to typical admin roles (admin, sc_admin, trackme_user, trackme_power, trackme_admin)
- If the user calling the `splunkremotesearch` command is not a member of the account-specified roles, access will be refused
- From version 2.0.61, both **direct membership and inheritance** are supported; users need to be a member of any of the provided roles, or a role which inherits from any of these roles

Hint

TrackMe automatically performs the rotation of Bearer tokens: (from version 2.1.8)

- Since TrackMe 2.1.8, we support automated rotation of the Splunk bearer tokens for remote accounts
- For every Splunk remote account, TrackMe will automatically try to rotate the bearer token based on the remote account's configured retention (every 7 days by default)
- This requires the Splunk service account on the remote side to own the capabilities

`list_tokens_all` and `edit_tokens_own`, so the token can be created and the previous token revoked

- The token renewal and revocation process is orchestrated by the general health tracker, which is executed once per day
- Note that the initial token you have manually created will not be revoked by TrackMe, and would be disabled automatically depending on your settings

Hint

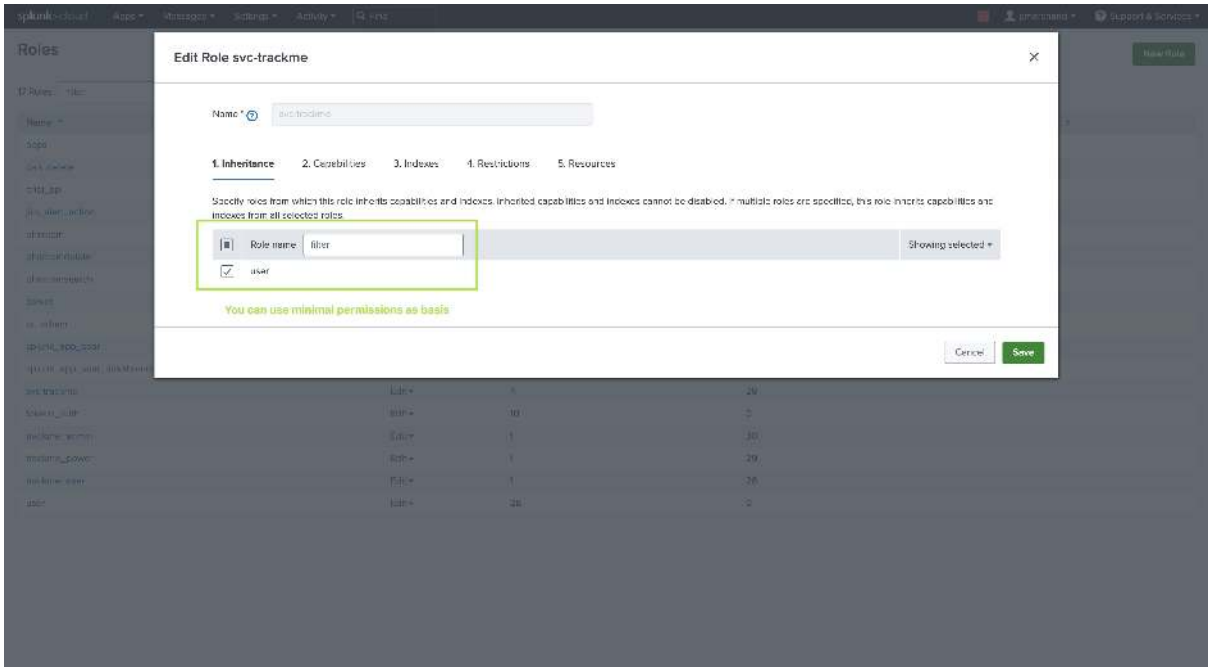
TrackMe Audit dashboard: Splunk Remote Account Overview (from version 2.1.16)

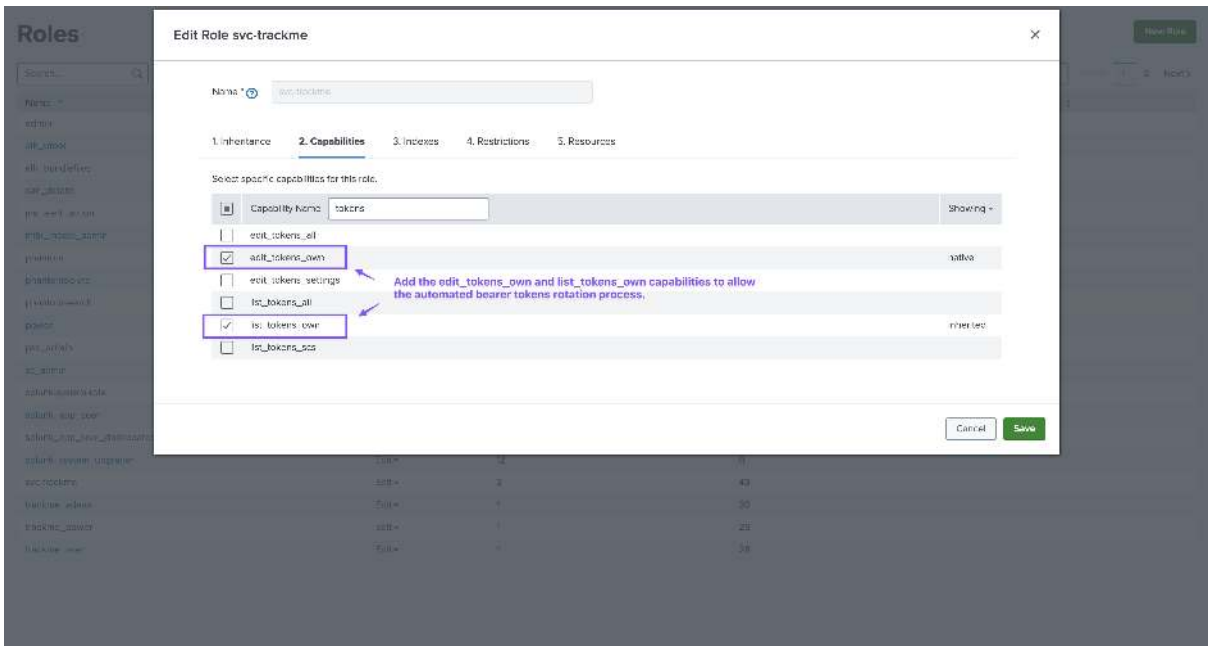
- TrackMe now includes an Audit dashboard to monitor the Splunk remote accounts
- This dashboard is available from the menu `Audit & Troubleshooting -> Audit - Splunk Remote Accounts Overview`
- This dashboard provides a comprehensive overview of the Splunk remote accounts, including the bearer token rotation process

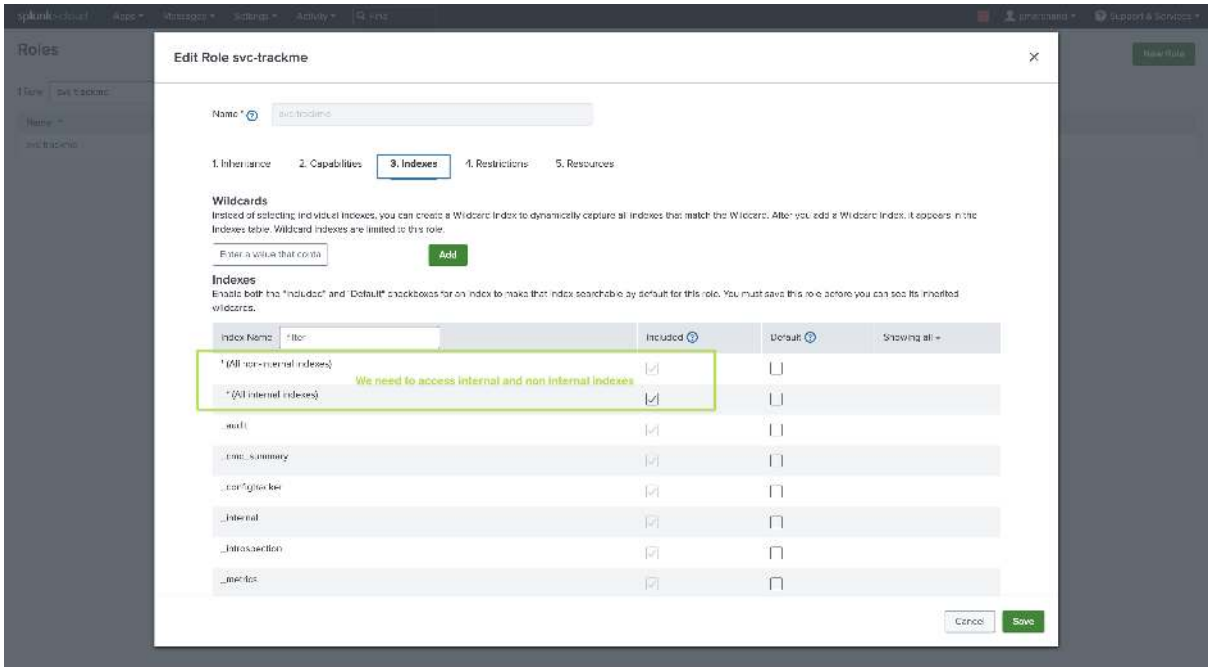
7.9.2 Minimal RBAC Requirements for the Remote User Account

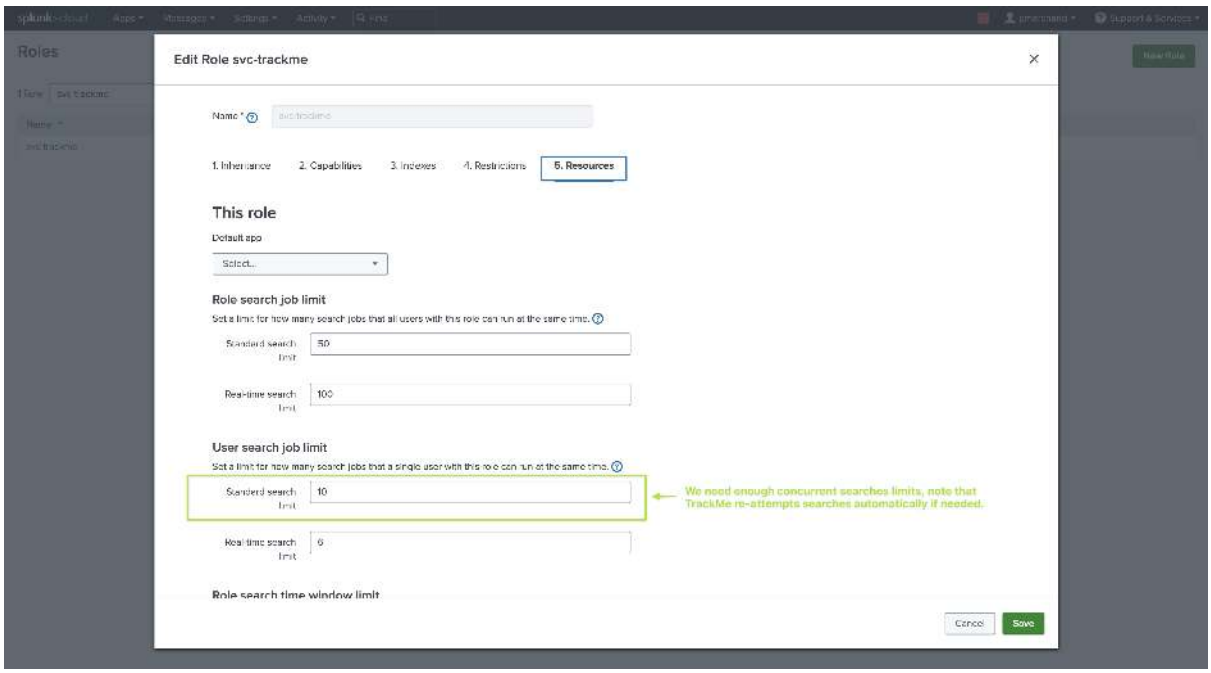
TrackMe remote capabilities rely on a Splunk bearer token authentication; this token is associated with a Splunk user on the remote side which itself is associated with specific roles, capabilities, permissions, and resource restrictions:

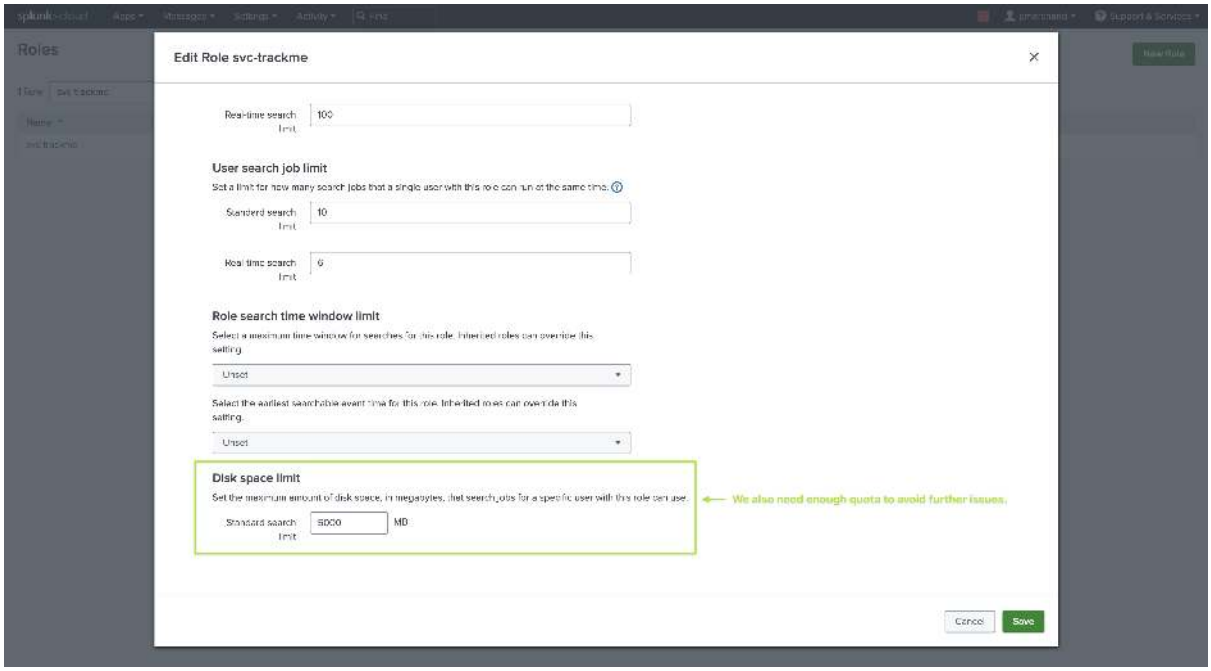
- **Roles and capabilities:** The user can be created with minimal permissions using the Splunk `user` role out of the box (You can inherit from user or a role providing the same capabilities as power)
- **Additional capabilities for tokens rotation:** To allow the automated bearer tokens rotation, the user needs to have the capability `list_tokens_own` and `edit_tokens_own`
- **Indexes:** Make sure the user can access both normal and `internal` indexes
- **Restrictions:** The user for TrackMe should not have any time limits restrictions, as there are use cases which require long-term searches
- **Resources:** It is recommended to give this user **enough concurrent searches** (unlike a very basic or minimal user) as well as a **sufficient quota** (5GB or 10GB for instance)


Roles
Edit Role svc-trackme
Name *
1. Inheritance
2. Capabilities
3. Indexes
4. Restrictions
5. Resources
Role name
filter
user
You can use minimal permissions as basis
Cancel
Save


Roles
Edit Role svc-trackme
Name *
1. Inheritance
2. Capabilities
3. Indexes
4. Restrictions
5. Resources
Select specific capabilities for this role.
Capability Name
tokens
edit_tokens_all
edit_tokens_own
edit_tokens_settings
list_tokens_all
list_tokens_own
list_tokens_scs
Add the edit_tokens_own and list_tokens_own capabilities to allow the automated bearer tokens rotation process.
Cancel
Save


Edit Role svc-trackme
1. Inheritance
2. Capabilities
3. Indexes
4. Restrictions
5. Resources
Wildcards
Add
Indexes
We need to access internal and non internal indexes
Cancel
Save


Edit Role svc-trackme
1. Inheritance
2. Capabilities
3. Indexes
4. Restrictions
5. Resources
This role
Default app
Role search job limit
Standard search limit
50
Real-time search limit
100
User search job limit
10
We need enough concurrent searches limits, note that TrackMe re-attempts searches automatically if needed.
Role search time window limit
Cancel
Save

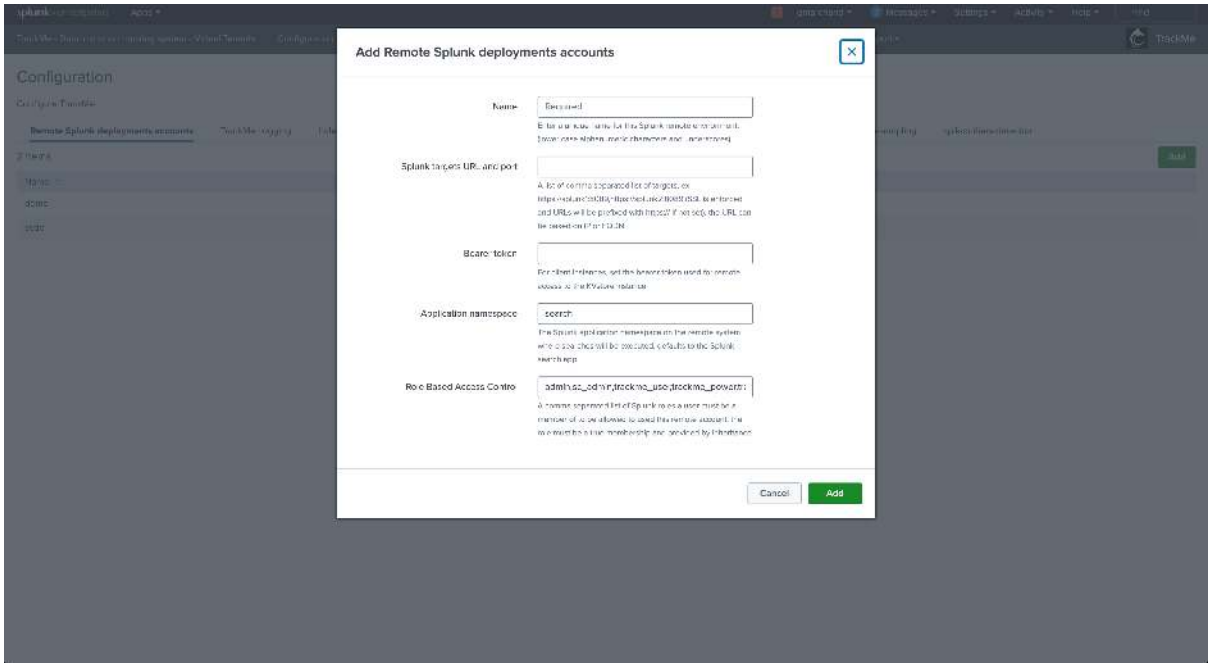


7.9.3 Configuring a New Remote Account

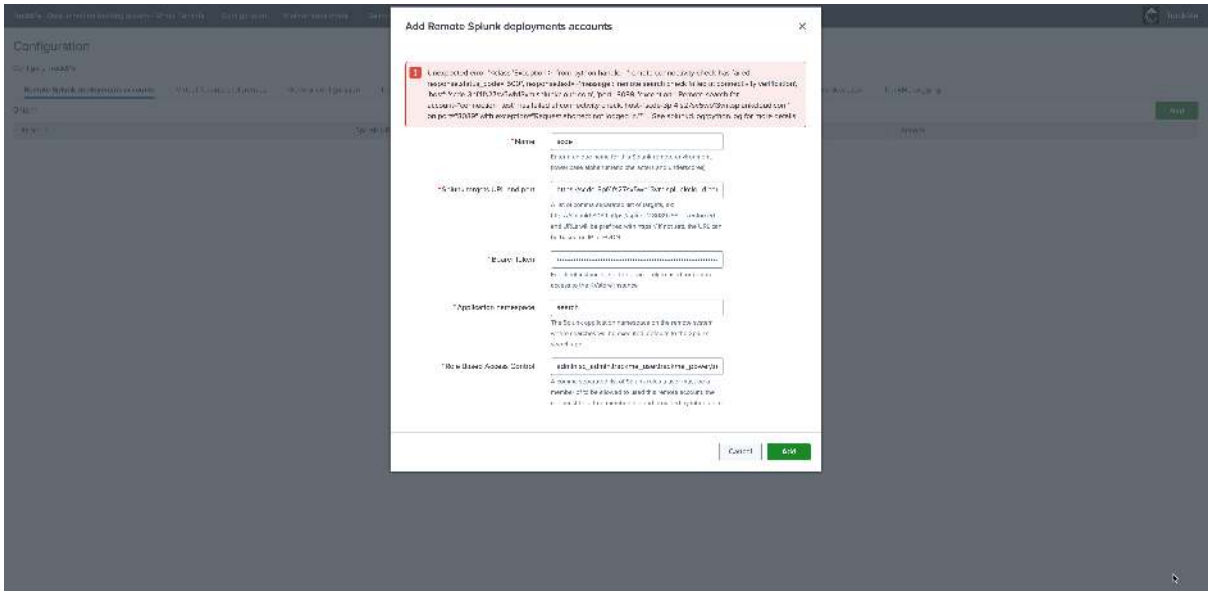
When configuring a Splunk Remote deployment, you will provide:

- A name which uniquely identifies the account (the account ID)
- The URL of the Splunk Search Head REST API, this can be a Fully Qualified Domain Name, a hostname, or an IP address (which itself can be a load balancer)
- Multiple REST endpoints can be submitted as a comma-separated list of targets
- The bearer token value, which is the secured credential to access the environment, is securely stored in the Splunk credential store
- The application namespace where remote searches are to be executed, defaults to the Search application on the remote target
- A comma-separated list of Splunk roles allowed to use this account (membership or inheritance)
- At the time of creation or later on if you edit the account, a connectivity check is automatically performed which validates the network connectivity and authentication. If the check fails, the interface will refuse the requested action

To configure a new remote account, access the Configuration user interface from the navigation bar:

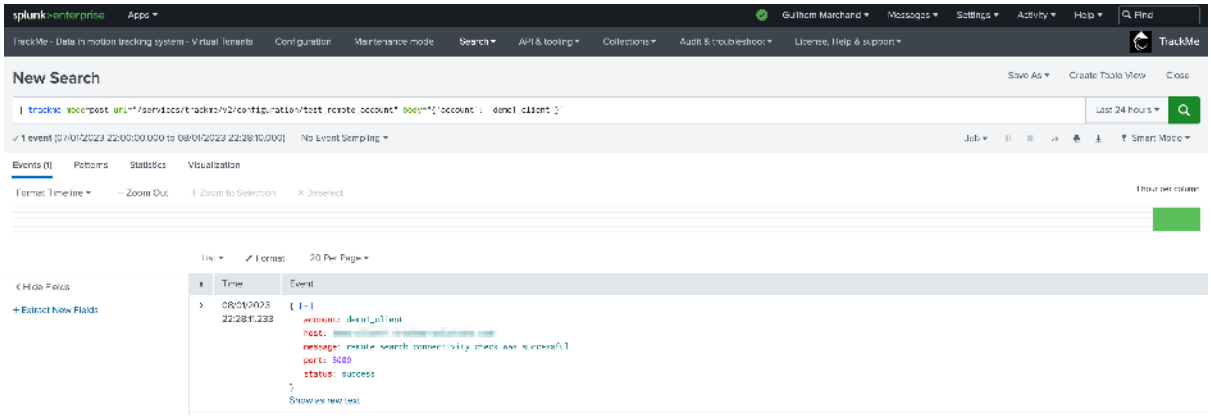


If the connectivity check fails at the creation step, the configuration UI will raise an exception which indicates the root cause of the failure:



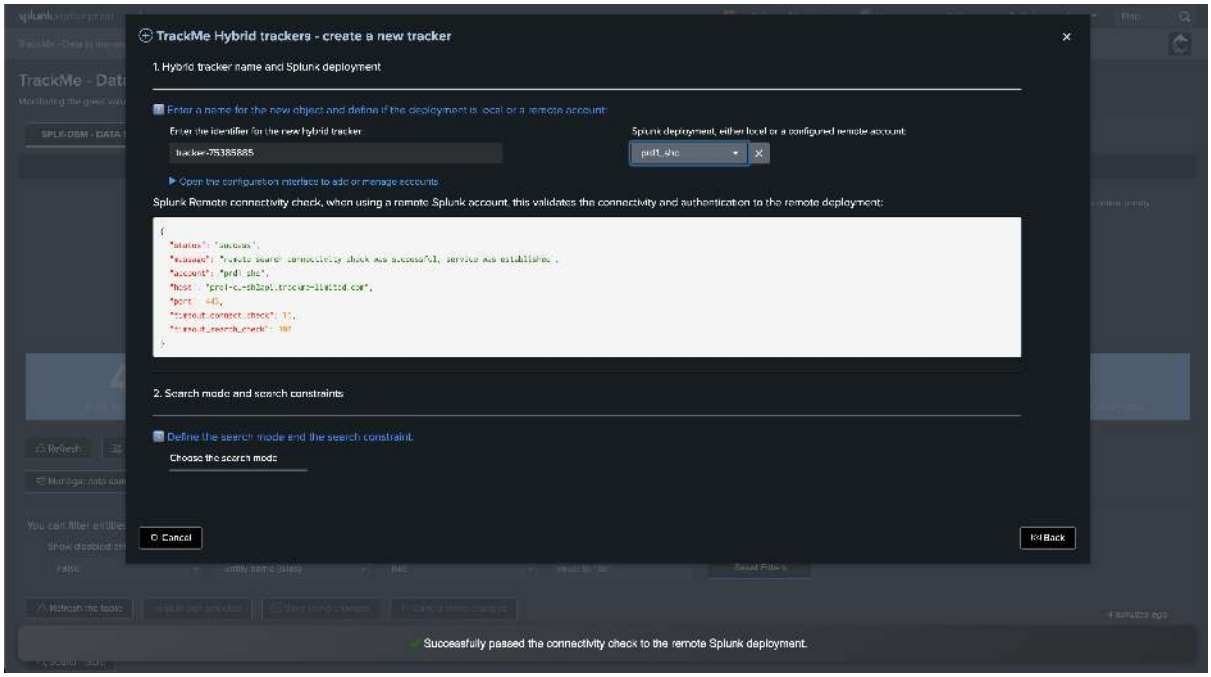
Once the account exists, you can test the connectivity easily using the following TrackMe REST endpoint in SPL:

```
| trackme mode=post url="/services/trackme/v2/configuration/test_remote_account" body=
↪"{'account': '<name of the account>'}"
```



The endpoint verifies that TrackMe can connect and authenticate successfully to the Splunk remote deployment.

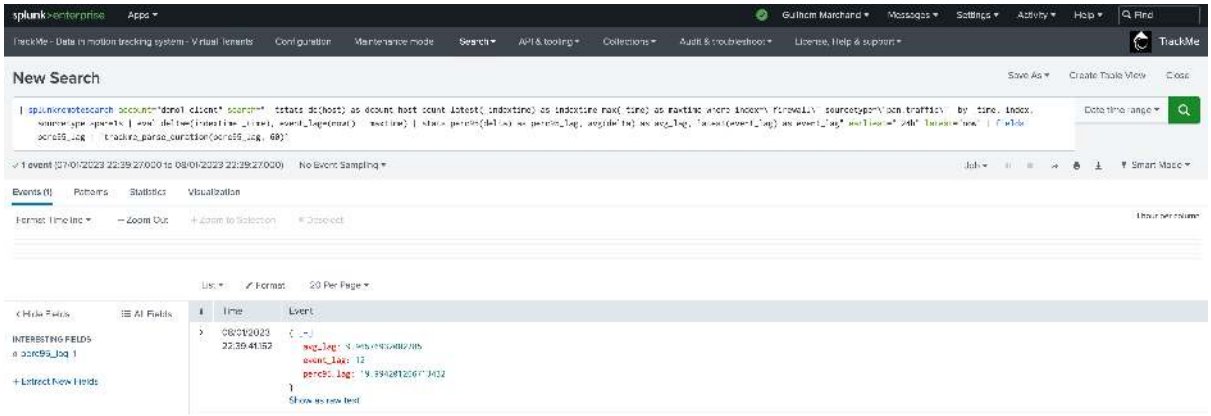
When creating a new Virtual Tenant or a new Hybrid object, TrackMe will perform the same connectivity verification automatically:



7.9.4 Example of a Splunk Remote Search Performed by TrackMe

When performing Remote Splunk searches, TrackMe automatically defines an optimized search to split the search logic between the part executed remotely, and the search logic handled locally.

Example:

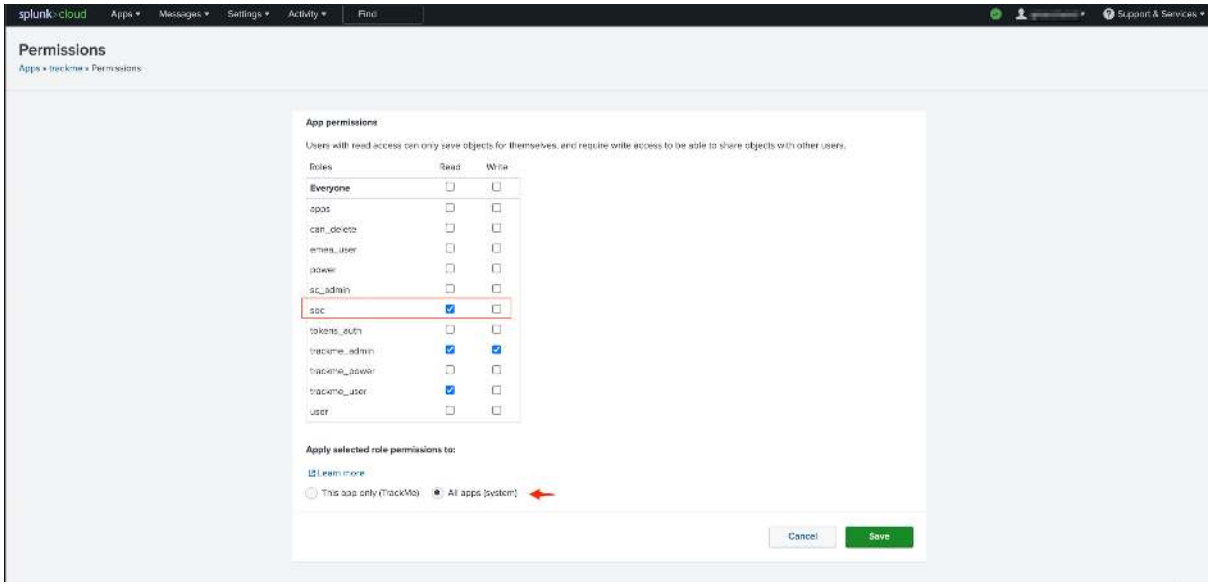


7.9.5 Using splunkremotesearch in a Different Application Namespace

To use the command `splunkremotesearch` from a different application than TrackMe, such as Search & Reporting, you need to perform a few steps as follows.

Share TrackMe at System Level

TrackMe by default is shared at the application level. You need to share at the **system** level as well as provide read access at least for your user roles ("soc" in our example):



Manage Capabilities

TrackMe requires capabilities to be able to access the command `splunkremotesearch`; the user roles need to have the capability:

- trackmeuseroperations

You can choose to inherit from the `trackme_user` role, or you can add the capability to your user roles as needed.

*Note: Access is granted by any of the 3 built-in trackme_* user roles**

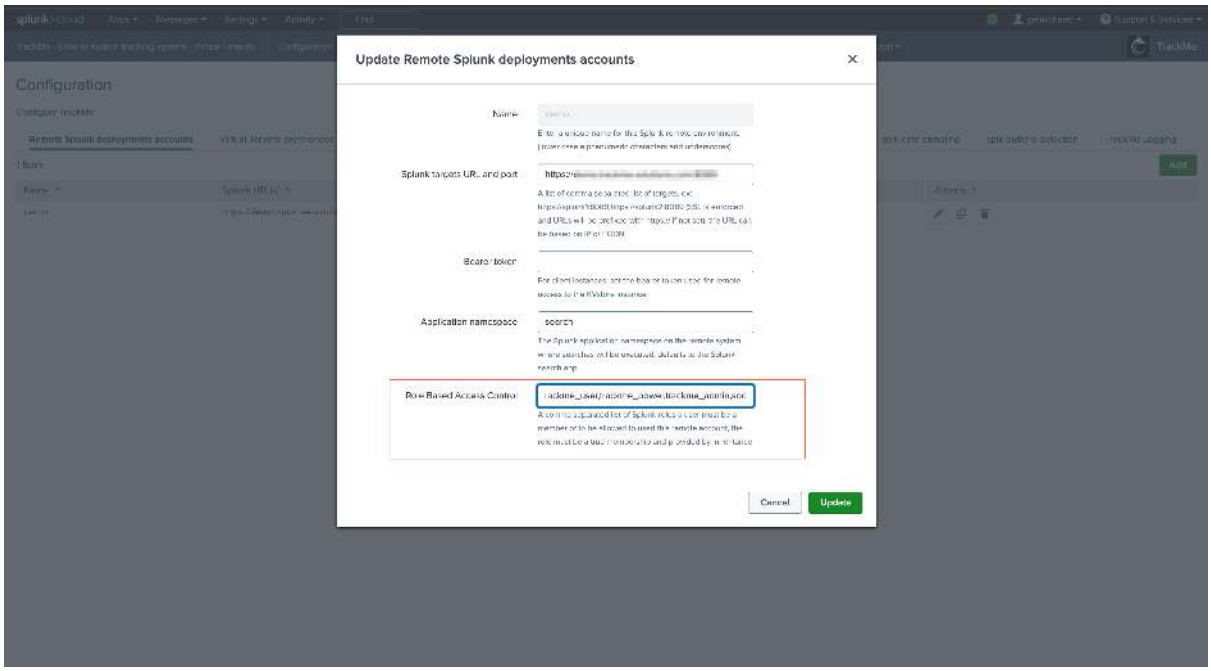
See: *Role Based Access Control and ownership* for more information about capabilities in TrackMe.

Manage Role-Based Access Control on the Account

User roles need to be listed in the account; by default when creating an account, the following roles are listed:

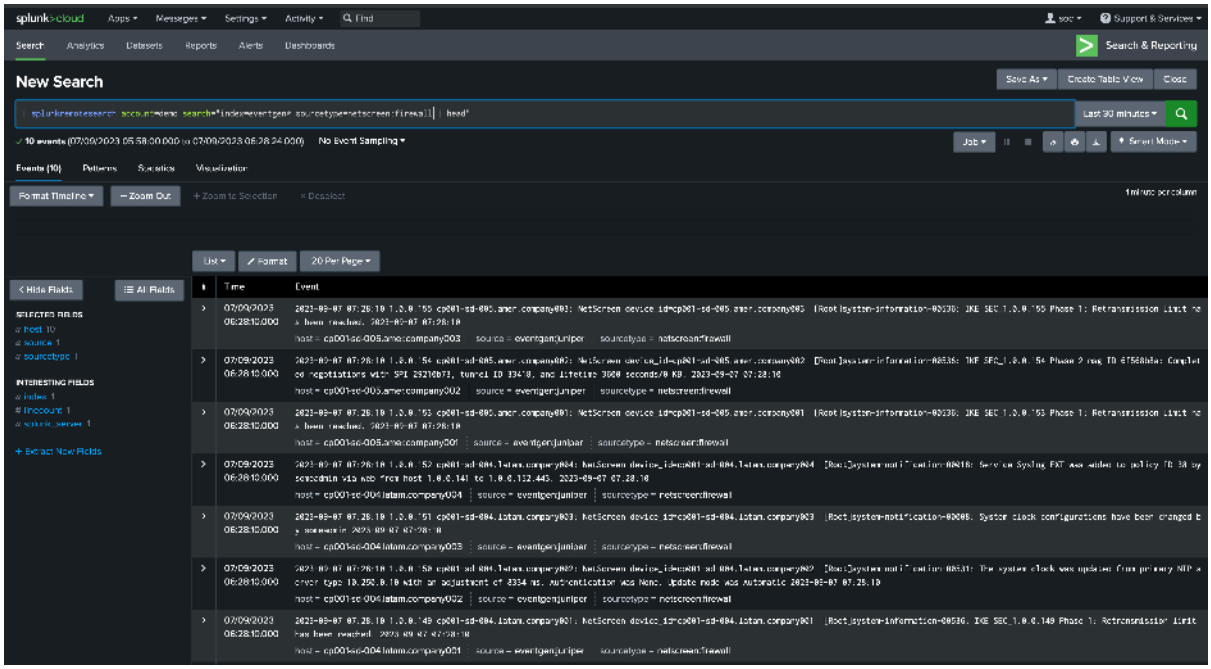
- admin, sc_admin, trackme_user, trackme_power, trackme_admin

This is an explicit membership requirement; in our example, we grant the user role "soc" to allow a member of this role to use the `splunkremotesearch` command against this account:



Use the Command

In our example, the "soc" user is a member of a role with the same name, and can now use the `splunkremotesearch` command as needed to perform searches against the remote deployment, from any other application namespace:



Accesses on the remote deployment rely on the service account which is associated with the bearer token configured in the account in TrackMe.

7.9.6 Troubleshooting Failure to Create or Update a Remote Account

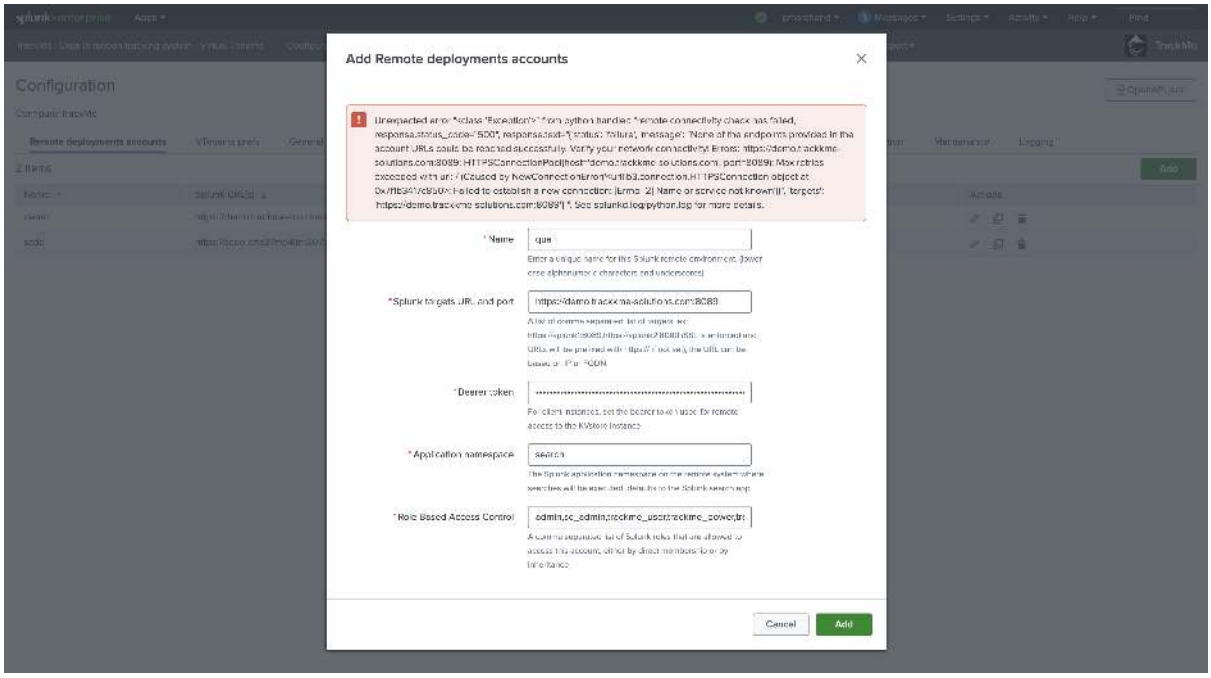
TrackMe Remote Search capabilities are an API-based integration with Splunk API relying on Bearer token authentication; you can refer to:

- https://docs.splunk.com/Documentation/Splunk/latest/Security/TroubleshootTokenAuth

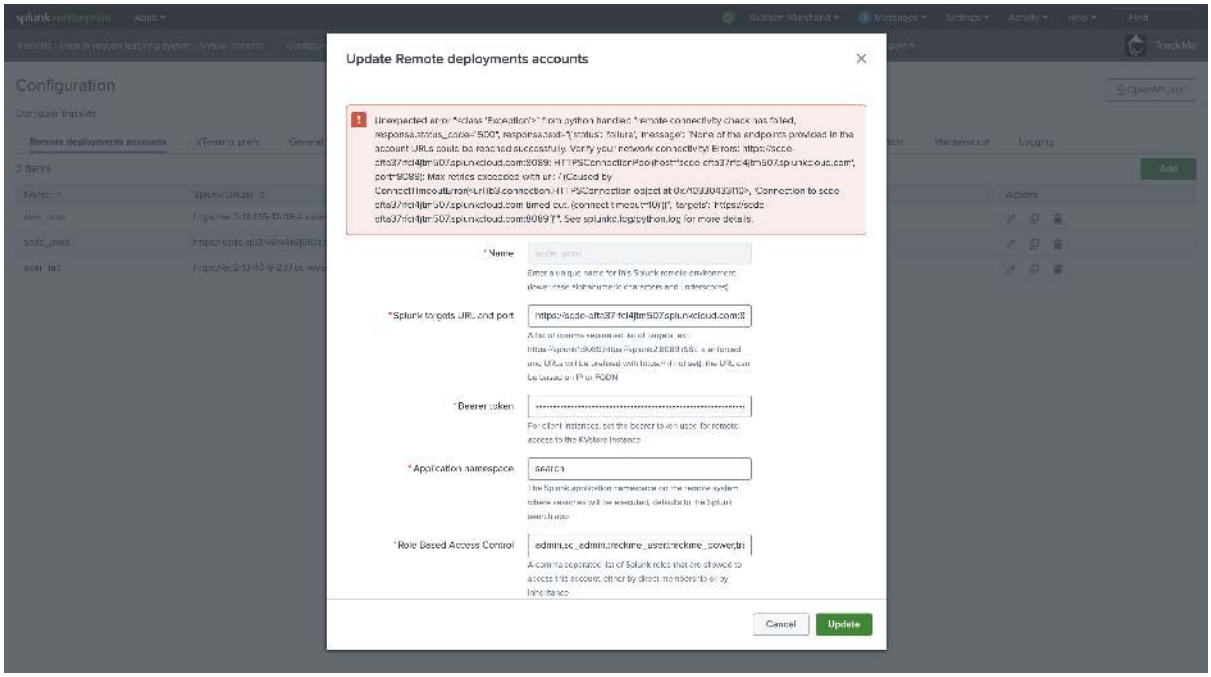
When attempting to configure a new remote account, or update an existing account, TrackMe will attempt to validate the connectivity, both from a network and authentication perspective.

If this step fails, the UI will clearly raise and show the exception encountered, for instance:

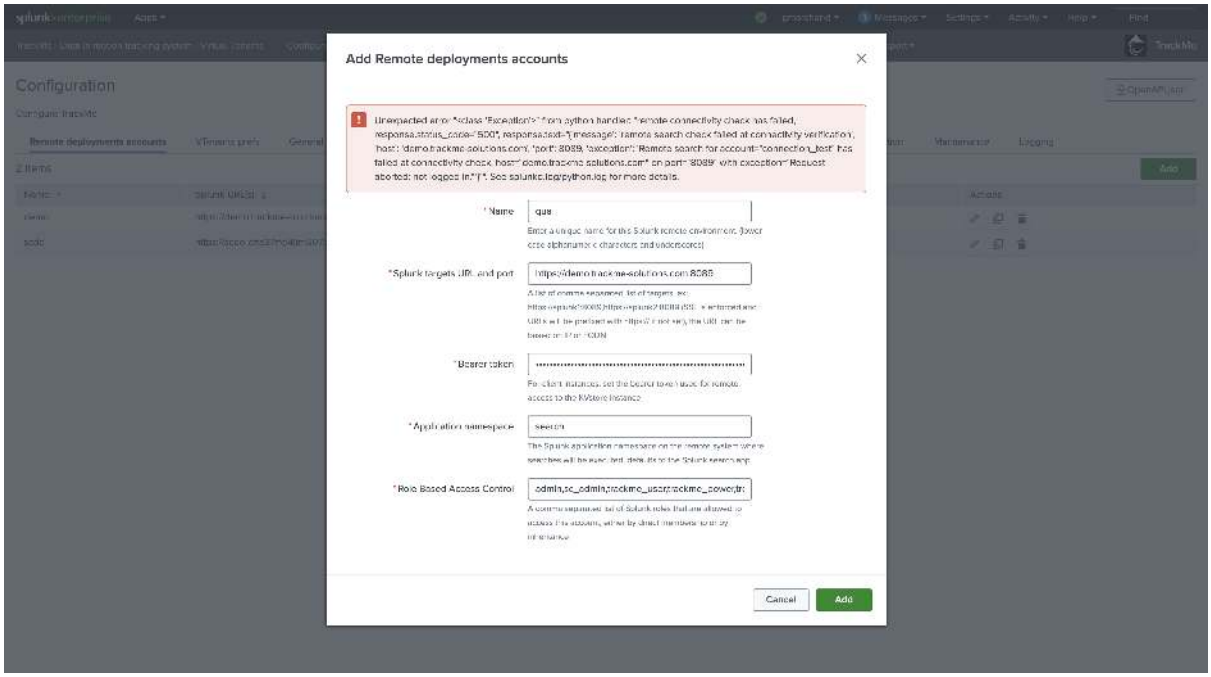
Example of a network connectivity failure: (wrong host)



Example of a network connectivity failure: (timeout)



Example of a refused authentication by the remote counterpart:



Note: You can also test the connectivity in pure SPL via the following TrackMe REST API endpoint:

```
| trackme mode=post url="/services/trackme/v2/configuration/test_remote_connectivity"␣
↪body="{'target_endpoints': 'https://mysplunk.mydomain.com:8089', 'bearer_token':
↪'xxxx', 'app_namespace': 'search'}"
```

Inspect splunkd.log logs on the remote counterpart:

If the connectivity fails for authentication related reasons, such as a wrong token or lack of capabilities, verify the splunkd.log on the remote counterpart, you can use the following Splunk search:

```
index=_internal source=*/splunkd.log JsonWebToken
```

Messages will be logged by Splunk, eventually indicating the root cause of the failure.

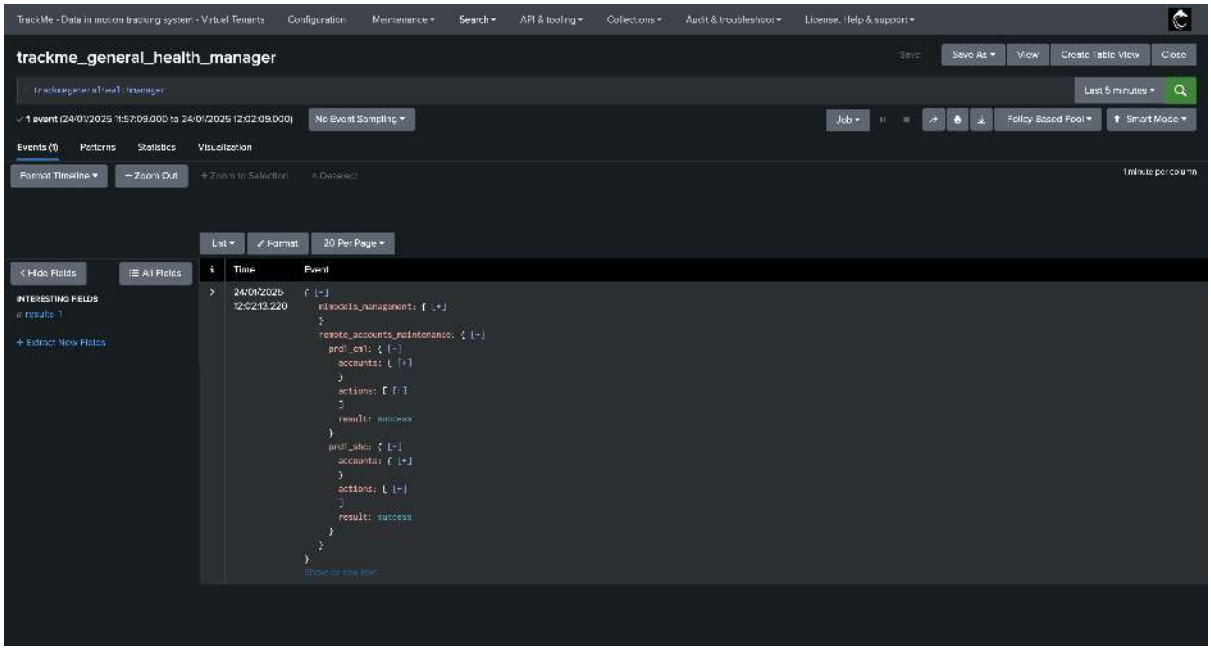
Example of a refused token:

```
02-05-2024 17:30:44.226 +0000 ERROR JsonWebToken [2843158 TcpChannelThread] -␣
↪JsonWebToken validation failed because: Token xxxx-xxxx has an invalid signature.
```

7.9.7 Bearer tokens automated rotation and revoking

Since TrackMe 2.1.8, we support automated rotation of the Splunk bearer tokens for remote accounts, the process works as follows:

- Once per day, a scheduled job called `trackme_general_health_manager` (General Health Manager tracker) is executed.
- **This job calls the endpoint `/services/trackme/v2/configuration/admin/maintain_remote_account` for each existing remote account, which does the following actions:**
  - Verify that the Remote account is up to date regarding available options, and update if necessary.
  - Verify if the Remote account has metadata available, and create/update if necessary. (stored in the KVstore `trackme_remote_account_token_expiration`)
  - If the Remote account bearer token has reached the rotation frequency, attempt to create a new bearer token, update the account and counters and finally revoke the previously generated token.



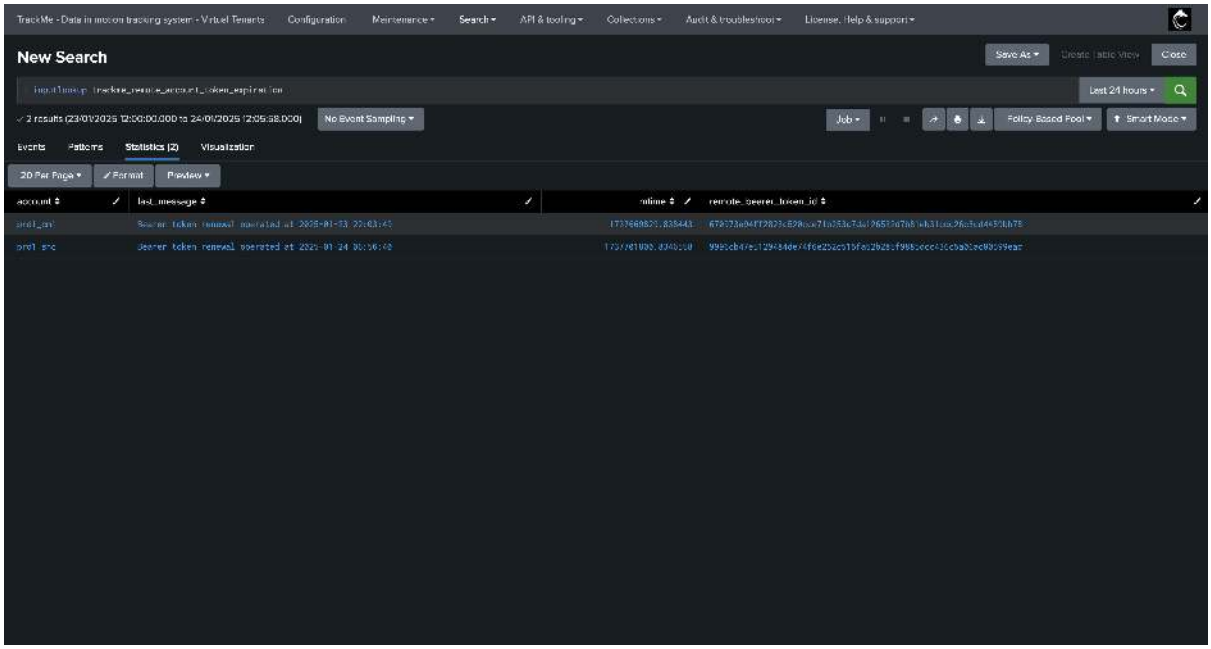
Notes about the initial token and rotation process:

- The initial token you have manually created will NOT be revoked by TrackMe, and would be disabled automatically depending on your settings.
- Once at least one rotation has been performed, and if it was successful, the initial token is not used anymore.
- From the stage of the first rotation, the previously generated and used token is automatically deleted by TrackMe, and the new token is automatically configured in the Splunk remote account.

Where to find bearer tokens rotation metadata?

TrackMe stores the metadata related to the bearer tokens rotation in the KVstore collection `trackme_remote_account_token_expiration`, you can access this collection using the following SPL search:

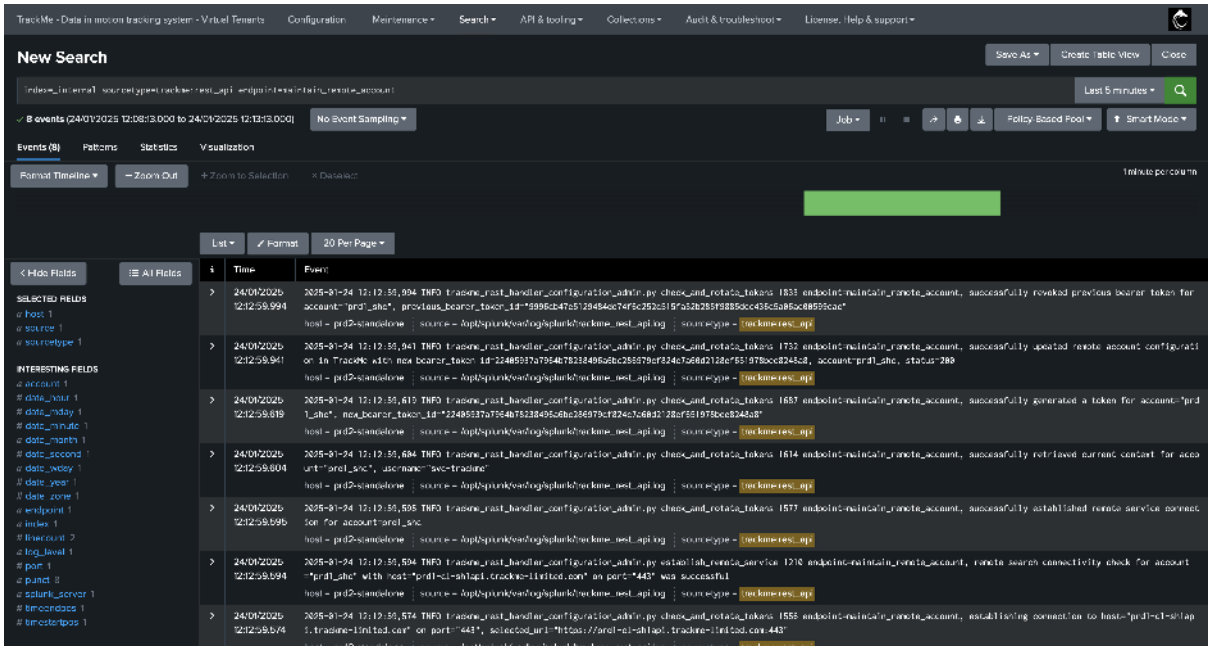
```
| inputlookup trackme_remote_account_token_expiration
```



How to access detailed logs about the tokens rotation process?

Logs are part of the TrackMe REST API logs:

```
index=_internal sourcetype=trackme:rest_api endpoint=maintain_remote_account
```

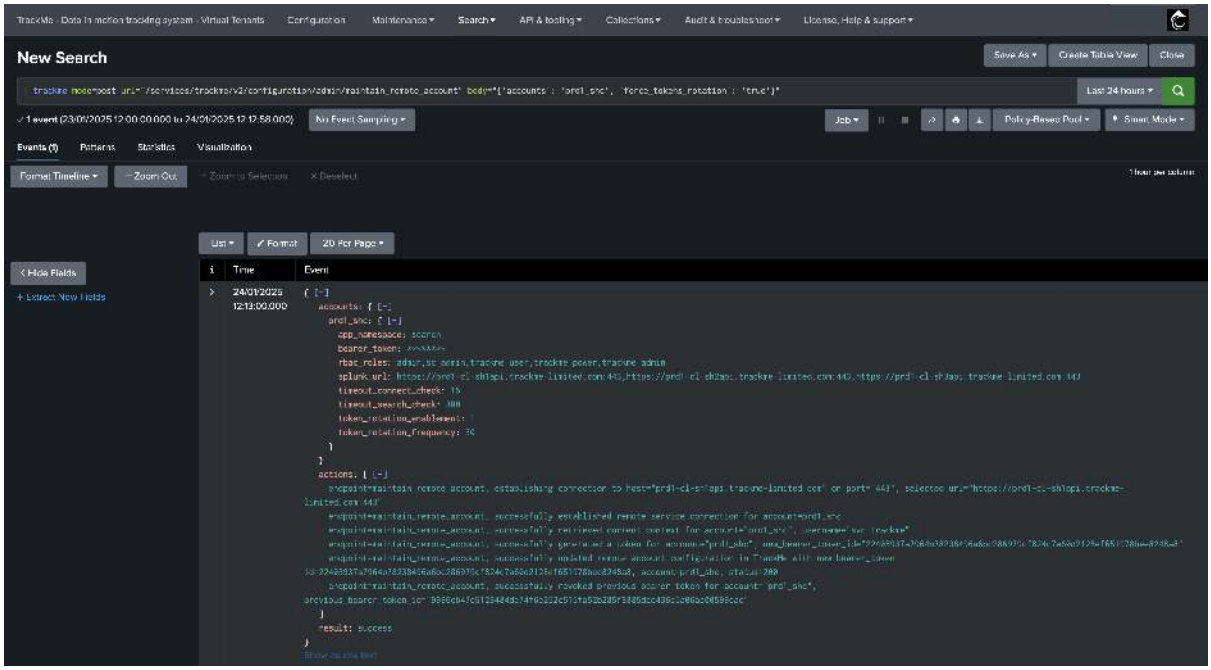


How can I force the rotation of a bearer token?

The rotation of the bearer token can be manually requested and performed by calling the TrackMe REST API endpoint, you can use the following SPL search:

```
| trackme mode=post url="/services/trackme/v2/configuration/admin/maintain_remote_
↪account" body="{'accounts': '<account name>', 'force_tokens_rotation': 'true'}"
```

*Note: replace <account_name> with the actual name of the account, or use * to rotate all accounts.*



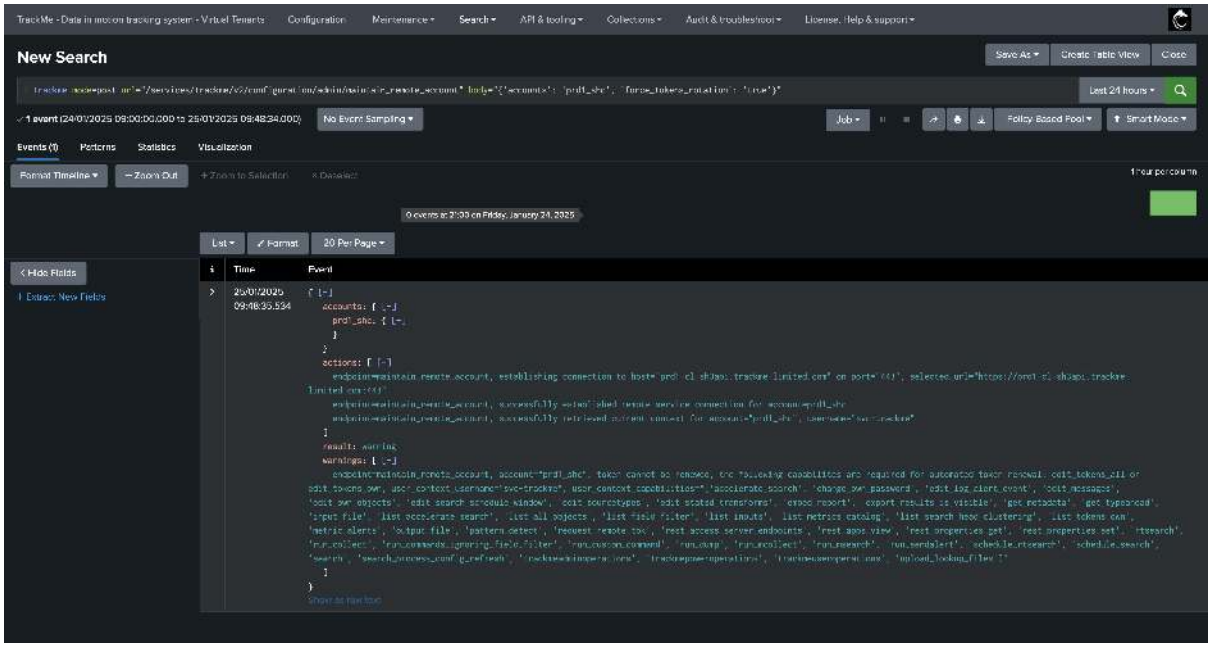
What if the token cannot be renewed?

If for some reasons, the token cannot be renewed, the process will be logged in the TrackMe REST API logs, and the token will not be revoked, the previous token will still be used:

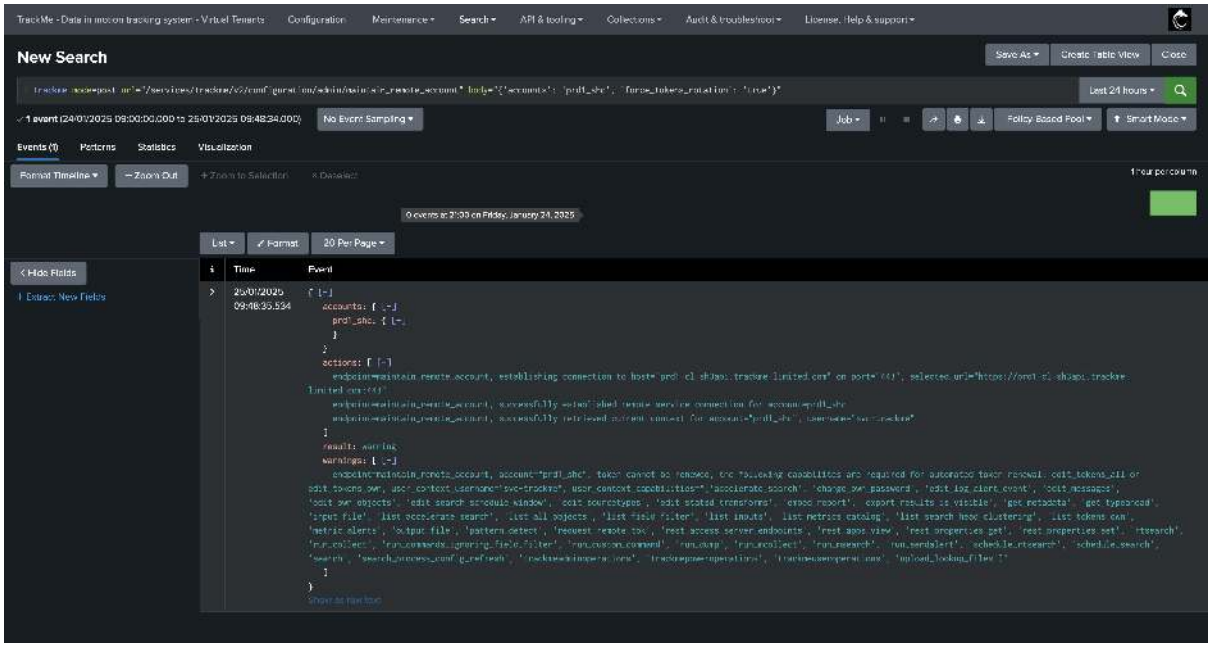
The following search will show errors or warnings related to the tokens rotation process:

```
index=_internal sourcetype=trackme:rest_api endpoint=maintain_remote_account (log_
↪level=ERROR OR log_level=WARN)
```

forced rotation is requested but failed:

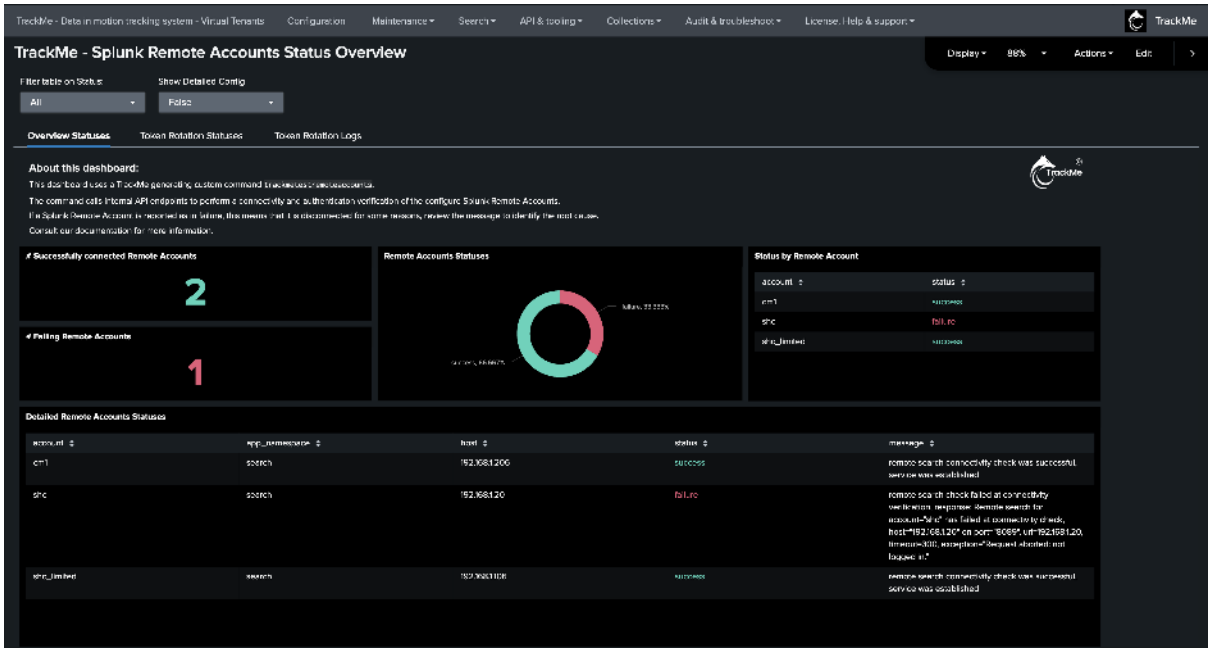


In this example, the remote user account lacks the capability to create a new token, the rotation process is aborted.



7.9.8 Audit Dashboard: Splunk Remote Accounts Overview

Access the Audit dashboard from the menu `Audit & Troubleshooting` -> `Audit - Splunk Remote Accounts Overview`.



7.9.9 Remote Accounts Commands Examples and References

Get the list of all configured remote accounts

This provides a list of all configured remote accounts, excluding the local account:

```
| trackme mode=get url="/services/trackme/v2/configuration/list_accounts"
| spath
| rename "accounts{}" as accounts
| table accounts
| mvexpand accounts
| where accounts!="local"
```

Test the Connectivity of all configured remote accounts

This command will test the connectivity of all configured remote accounts:

```
| trackme mode=get url="/services/trackme/v2/configuration/list_accounts" | spath
| rename "accounts{}" as accounts
| table accounts
| mvexpand accounts
| where accounts!="local"

| map maxsearches=100 search="| trackme mode=post url=/services/trackme/v2/
↪configuration/test_remote_account body=\"{'account': '$accounts$'}\""
| trackmeprettyjson fields=_raw
| spath
```

Get the detailed including the bearer token for all configured remote accounts

This command will provide the detailed information about all configured remote accounts, including the bearer token:

```
| trackme mode=get url="/services/trackme/v2/configuration/list_accounts" | spath
| rename "accounts{}" as accounts
| table accounts
| mvexpand accounts
```

(continues on next page)

(continued from previous page)

```
| where accounts!="local"

| map maxsearches=100 search="| trackme mode=post url=/services/trackme/v2/
↪configuration/get_remote_account body=\"{'account': '$accounts$'}\""
| trackmeprettyjson fields=_raw
| spath
```

Force the rotation of the bearer token for all configured remote accounts

This command will force the rotation of the bearer token for all configured remote accounts in a single command:

```
| trackme mode=get url="/services/trackme/v2/configuration/list_accounts" | spath
| rename "accounts{}" as accounts
| table accounts | mvexpand accounts
| where accounts!="local"

| map maxsearches=100 search="| trackme mode=post url=/services/trackme/v2/
↪configuration/admin/maintain_remote_account body=\"{'accounts': '$accounts$',
↪'force_tokens_rotation': 'true'}\""
| trackmeprettyjson fields=_raw
```

Get the token expiration metadata for all configured remote accounts and the time since the last rotation

This command will provide the token expiration metadata for all configured remote accounts and the time since the last rotation:

```
| inputlookup trackme_remote_account_token_expiration | eval keyid=_key
| eval _time=mtime
| eval time_since=tostring(round(now()-mtime), "duration")
```

7.10 Role Based Access Control and ownership

Least privileges approach

- Since the release **version 2.0.34**, TrackMe uses a strict **least privileges** approach which avoids any requirements for dangerous or problematic Splunk capabilities, such as list_settings or list_storage_passwords
- TrackMe uses a 3 RBAC levels schema, with the following built-in roles: trackme_user, trackme_power, trackme_admin
- Each of these roles provides a built-in capability which allows access to the associated REST API endpoints
- Depending on the level of privileges, users will have access to different parts of TrackMe user interfaces; for instance, a read-only user will not have visibility of any administration-related shortcuts or entity edition buttons
- When migrating from a prior version, TrackMe will automatically assign the same permissions for existing tenants for the trackme_admin roles of the tenants to the trackme_power roles
- You can use TrackMe's built-in RBAC re-assignment to update a tenant configuration according to your needs
- RBAC is also supported and required at the level of Splunk remote account

- Roles **inheritance** for TrackMe Virtual Tenants and Remote accounts is supported from version 2.0.61

About role inheritance

- Roles **inheritance** for TrackMe Virtual Tenants and Remote accounts is supported from version 2.0.61
- From this version, you can use direct membership or inheritance for all RBAC dimensions in TrackMe

7.10.1 Introduction to RBAC in TrackMe

Role Based Access Control and Ownership management are supported and taken in charge natively, relying on a strict implementation of Splunk capabilities.

In short:

- When a Virtual Tenant is created, TrackMe administrators can define target indexes for every type of data generated by TrackMe
- Access to these indexes can be granted via Splunk Role Based Access Control
- The Virtual Tenant knowledge objects permissions are configured during creation, allowing two types of access: administrative access and user access
- When new knowledge objects are created for the tenant via TrackMe, the permission rules defined for the Virtual Tenants are applied automatically to these new objects
- At any time, a TrackMe administrator can update the Role Based Access Control permissions for the Virtual Tenant; the tenant configuration will be updated and all existing knowledge objects for the tenant will be updated automatically

TrackMe built-in roles capability matrix

The following table summarizes TrackMe's built-in roles, associated capabilities, endpoints access, and privileges categories:

Table 12: TrackMe built-in Roles and Capabilities

Role	Capability	Endpoints root	Description
trackme_user	trackmeuseroperations	all but write and admin	allows read-only access in TrackMe
trackme_power	trackmepoweroperations	*/write	allows management of TrackMe entities but not the creation of new content (such as trackers)
trackme_admin	trackmeadminoperations	*/admin	allows content management such as the creation of tenants and trackers

As a summary:

- When defining your own roles integration, you can choose to have users being members of TrackMe built-in roles, or you can choose to inherit these roles into your own roles to grant TrackMe capabilities accordingly

- A member of a role providing the trackmeadminoperations capability can create tenants and scheduled logics (trackers) even if the user's Splunk permissions do not normally allow the creation of scheduled knowledge objects; this is made possible only in TrackMe for TrackMe context via TrackMe's REST API and through elevation of privileges according to the user's capabilities
- If a user who does not own the trackmepoweroperations capability attempts to perform a write operation through an associated REST endpoint, Splunk will refuse access to this endpoint
- Similarly, if a user who does not own the trackmeadminoperations attempts to reach any of the admin-related REST endpoints, Splunk will refuse access to the requested endpoint

7.10.2 Definition of RBAC and ownership at the creation phase of the Virtual Tenant

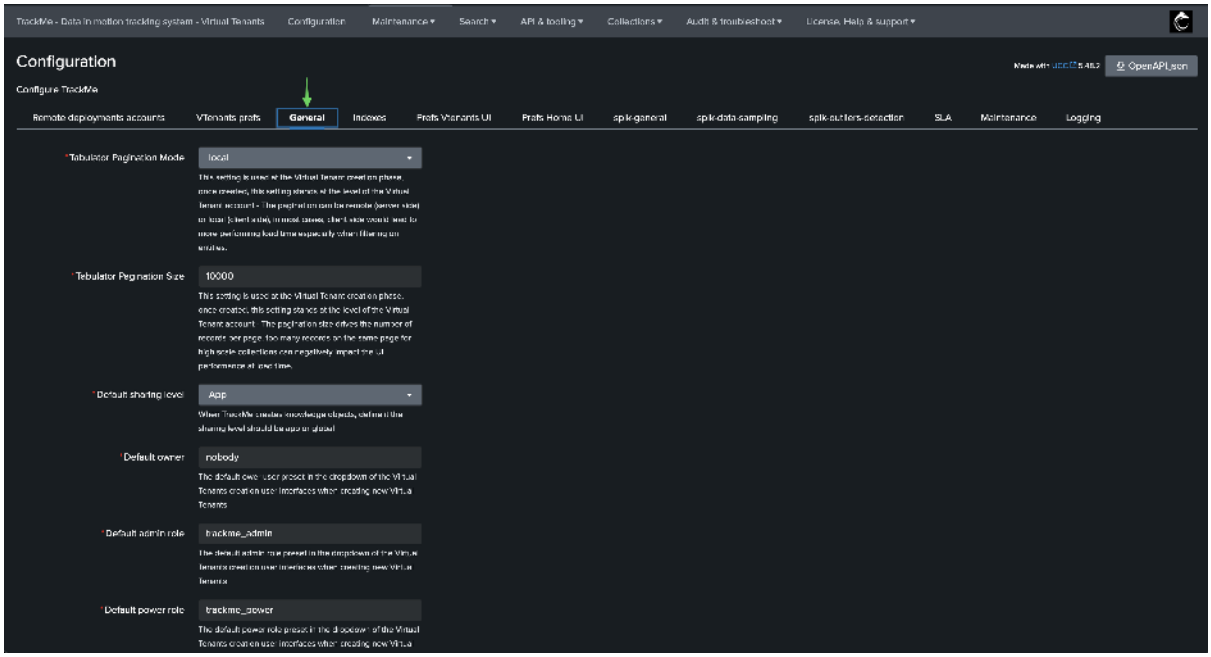
Consider the following example:

- A Virtual Tenant is created for the purpose of the SIEM quality control operations for our ACME EMEA department
- The tenant is created with TrackMe indexes dedicated to this department
- The tenant is created with specific roles allowing its administration, and roles allowing non-privileged access to the tenant
- Access to these indexes as well as the administration and user access are provided by two Splunk roles: `emea_siem_admin` for admin and `emea_quality_control` for normal user access
- Although it is not mandatory, the ownership is set to a service account user `srv-trackme` to facilitate further control for Splunk administrators

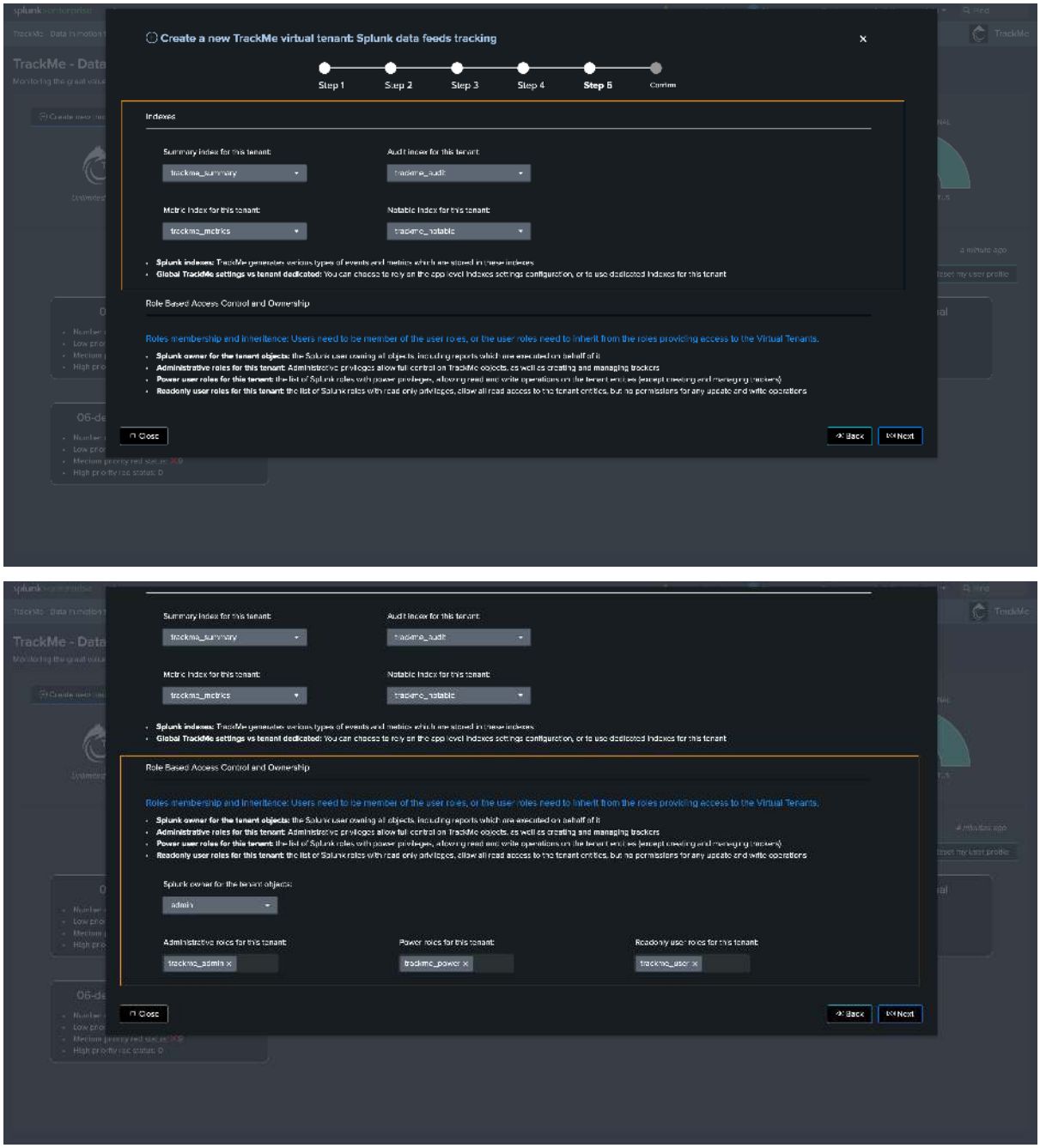
Hint

Preset RBAC for the tenant creation UI

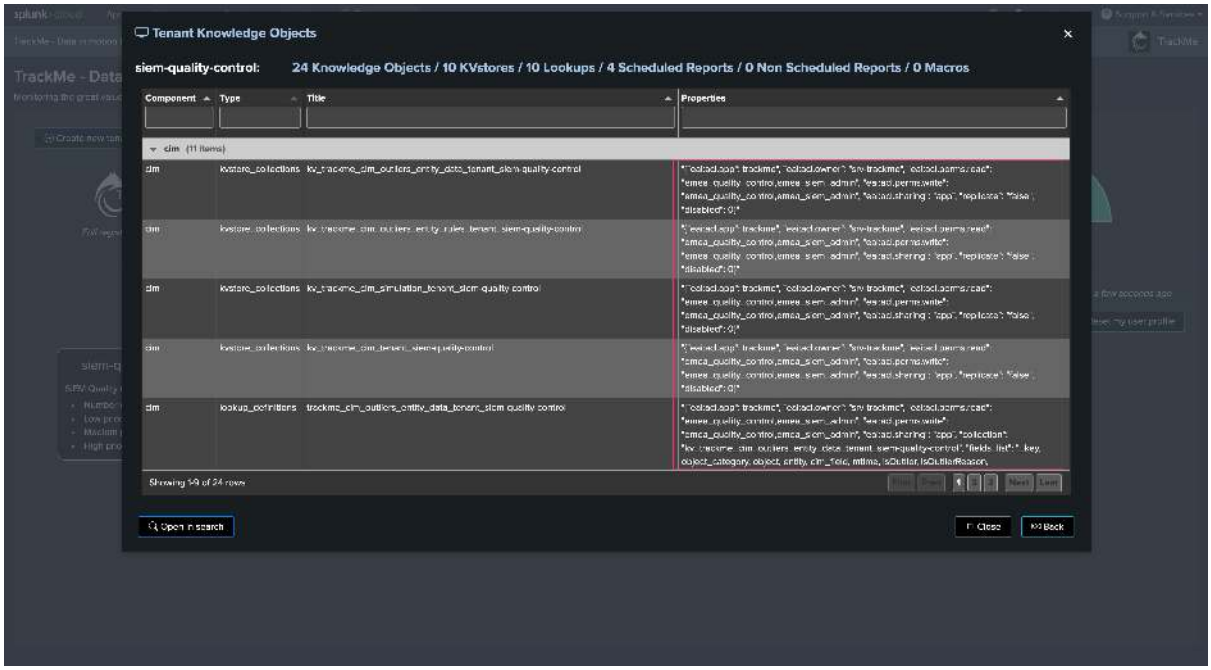
- Since **version 2.0.52**, you can preset values for the owner and roles when creating a new Virtual Tenant from the UI
- Go to Configuration then General Configuration



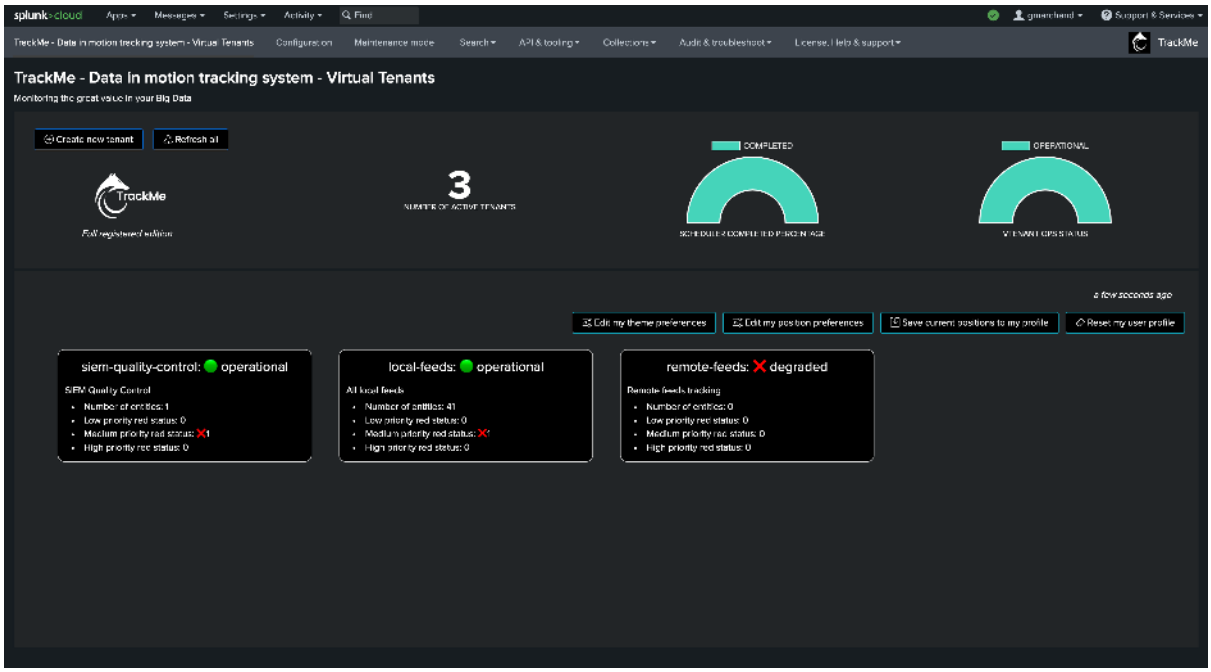
Defining RBAC when creating new Virtual Tenants:



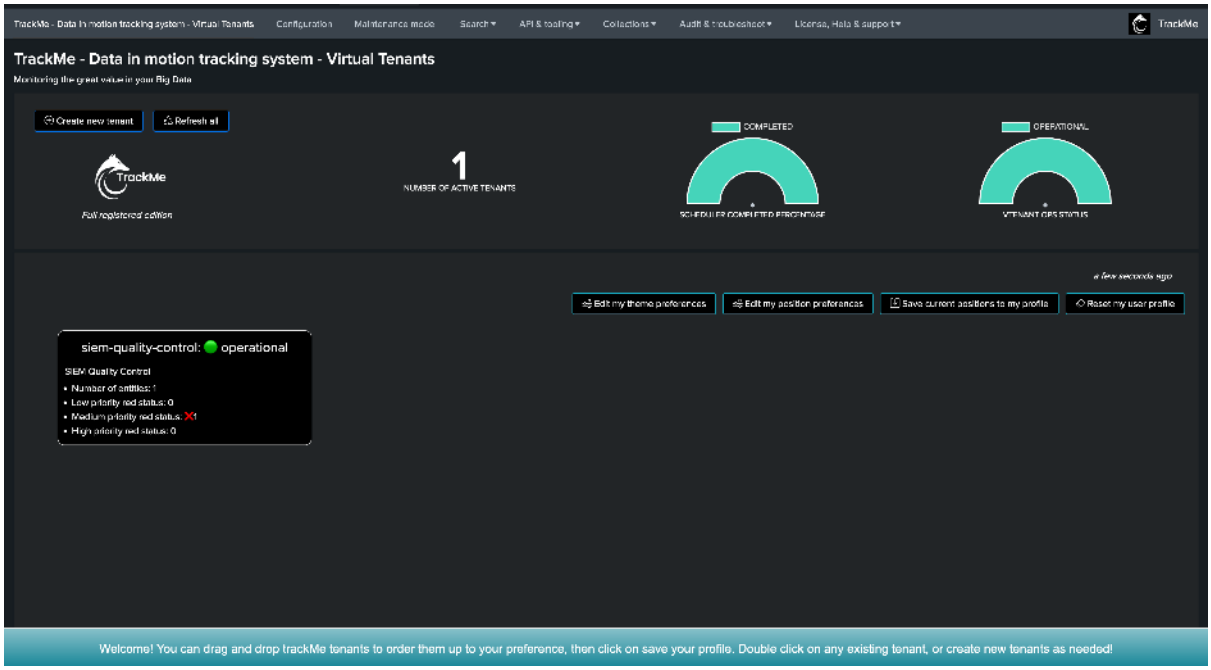
Once the tenant has been created, we can observe the permissions and ownership carefully defined by TrackMe:



Althrough we may have several tenants in the environmment, TrackMe will provide access to the tenants according to the RBAC policies, for instance a full admin access shows all existing tenants:



A user member of our `emea_quality_control` role will only see and be able to access to list of granted Virtual Tenants:

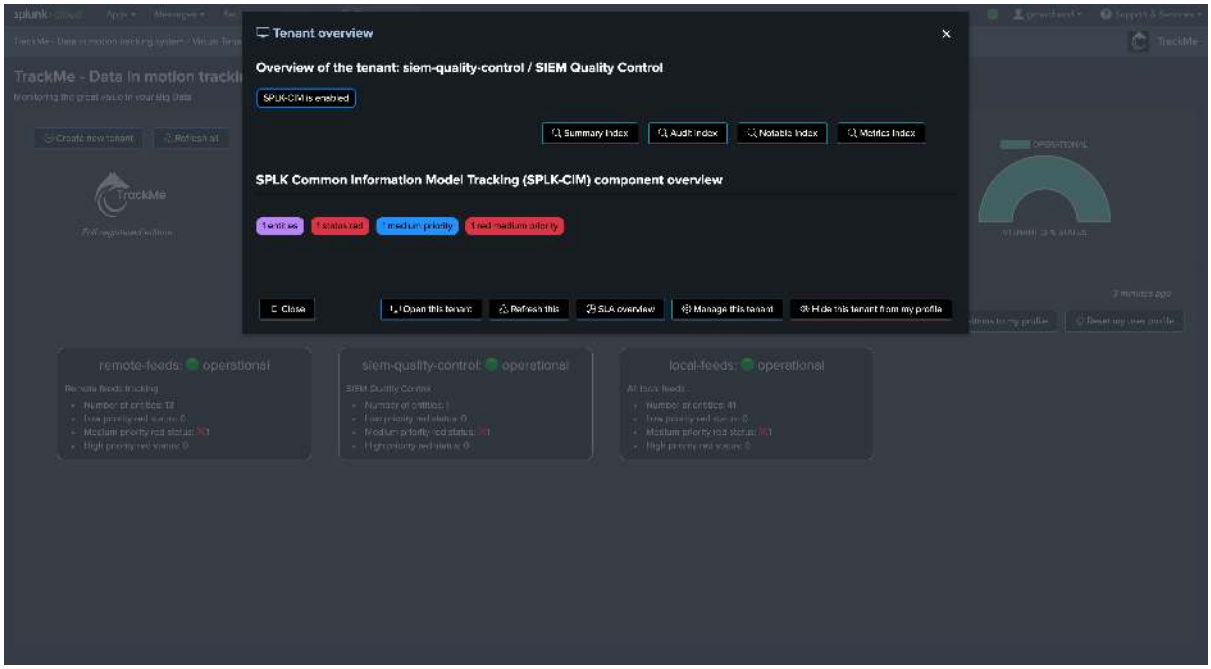


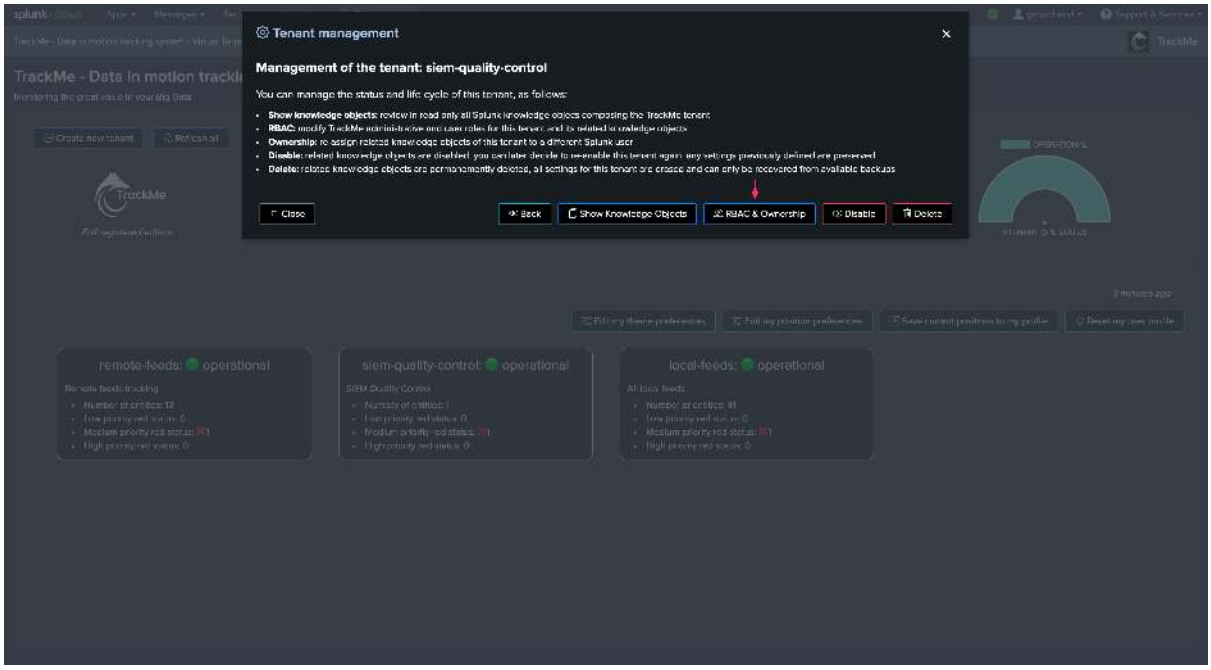
7.10.3 Updating RBAC policies and ownership assignment for an existing Virtual Tenant

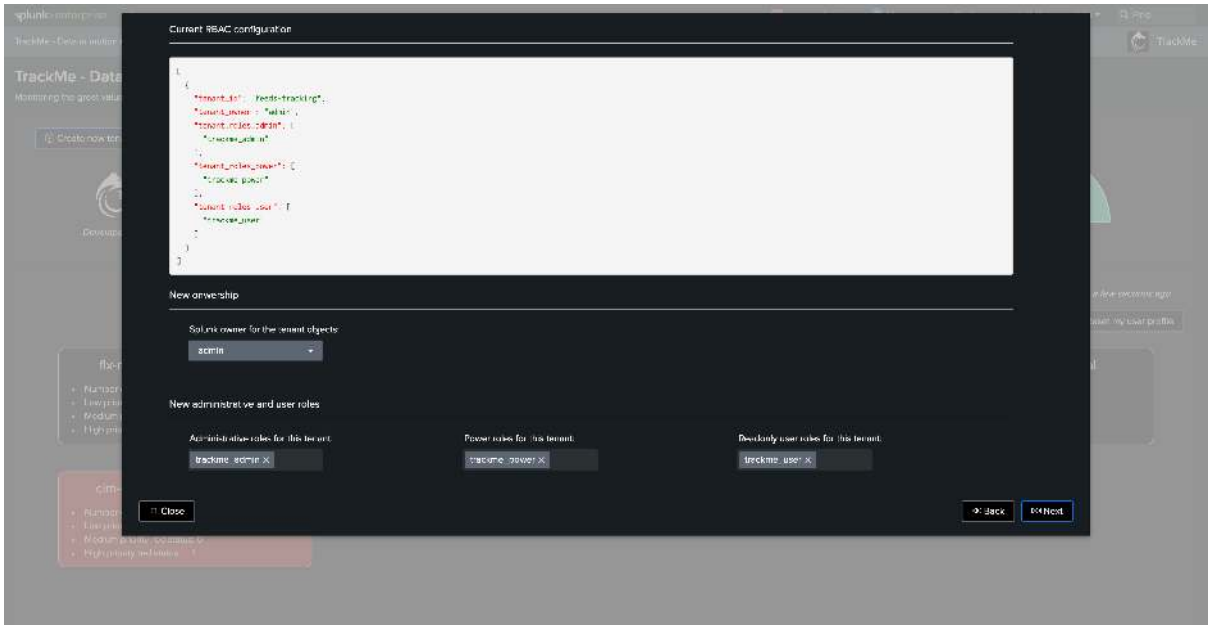
TrackMe allows updating the Role Based Access Control policies as well as the ownership of an existing Virtual Tenant.

Updating RBAC through the Virtual Tenant user interface

Access the Virtual Tenant user interface, double-click on the tenant and access its administration screen:







Carefully update the ownership and/or RBAC policies; TrackMe will update the Virtual Tenant configuration, all of its knowledge objects will be re-assigned if necessary, and their permissions updated accordingly.

Updating RBAC through the REST API

You can equally update the RBAC policies and ownership through the REST API endpoint:

For instance, say we want to add an additional user role access `siem_users` to our tenant `siem-quality-control`, we could run:

```
| trackme url=/services/trackme/v2/vtenants/admin/update_tenant_rbac mode=post body="
↪{ 'tenant_id': 'siem-quality-control', 'tenant_roles_admin': 'emea_siem_admin',
↪'tenant_roles_power': 'emea_siem_power', 'tenant_roles_user': 'emea_quality_control,
↪siem_users', 'tenant_owner': 'srv-trackme'}"
```

7.11 Alerting Architecture & Third-Party Integration

7.11.1 Introduction to Stateful Alerting in TrackMe

Hint

TrackMe Stateful alerting is a major enhancement in TrackMe alerting concepts and management, with **state-aware alerts**, advanced **Email thread alert notifications**, and **active command** execution.

- With Stateful alerting, TrackMe automatically maintains cycles of incidents in a persistent and state-aware manner, used in conjunction with sophisticated and powerful capabilities.
- Notably, TrackMe can deliver rich email notifications, in HTML format and with automated and embedded metrics charts.
- It also generates stateful events, which can be used to trigger advanced third-party notifications, starting with the incident creation, incident updates, and incident closure when the entity returns to a normal state.
- These features allow TrackMe to provide a complete, powerful, and flexible state-aware alerting architecture.
- **Email charts are not available in Splunk 9.1/9.2 environments due to Python 3.8.x and later requirements.**
- Since TrackMe 2.1.16, TrackMe can also **execute active commands (generating or streaming commands)** depending on whether the incident is opened, updated, or closed, allowing flexible and powerful state-aware actions such as **incident creation, updating, and closing.**
- Since TrackMe 2.1.18, Stateful Alerts can be configured to automatically acknowledge entities when the incident is **opened**.
- TrackMe **2.1.19** has addressed all compatibility issues with emails clients and notably embedded charts, this is the case for instance for **Outlook** and **Web Gmail**.
- TrackMe **2.1.19** also simplifed the configuration and added granular **priority levels selections** management of emails and commands notifications as the level of the handler, allowing to simply

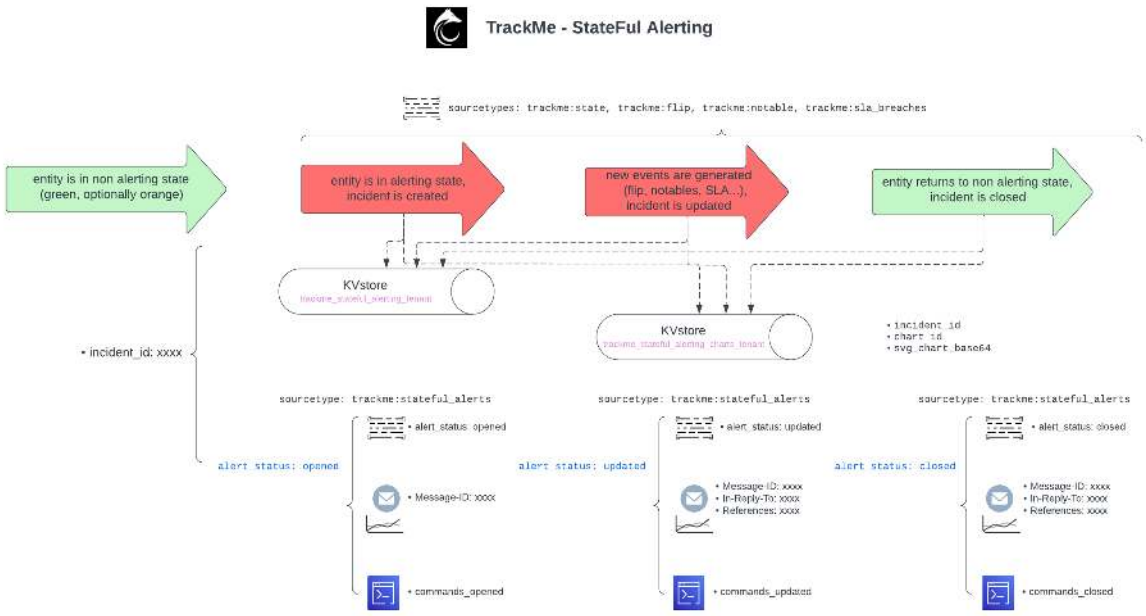
create the stateful alert which handles incidents for all entities, and selecting which entities would lead to the generation of emails and commands notifications.

- Since TrackMe **2.1.20** the charts time range window is configurable at the alert level, with a default of 24 hours.

7.11.2 Sophisticated Alerting Made Easy - TrackMe Alerting in a Nutshell

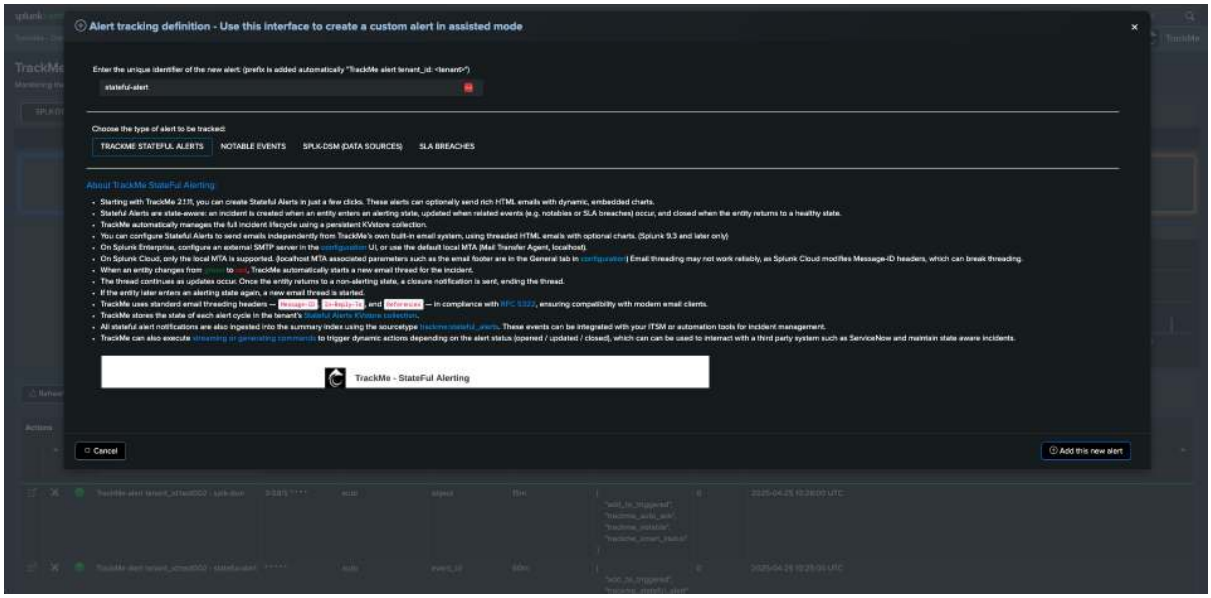
In short:

- Once enabled, when a given entity turns into an alerting state, a new incident is created.
- When new events are detected, such as Notable events or SLA breaches, the incident is updated.
- Eventually the entity returns to non alerting state, the incident is closed.
- Rich HTML emails notifications can be delivered by TrackMe, including automated and embedded metrics charts.
- Events are also generated, in the `sourcetype=trackme:stateful_alerts`.
- Therefore, TrackMe can generate **Opening / Updating / Closing events**, and/or **rich Emails notifications**.
- **Active commands (generating or streaming commands)** can also be executed, depending on the incident status.



7.11.3 Creating a Stateful Alert in a Few Easy Steps

In the TrackMe Virtual Tenant, go to the "TRACKING ALERTS" tab, and use the wizard to create a new alert:



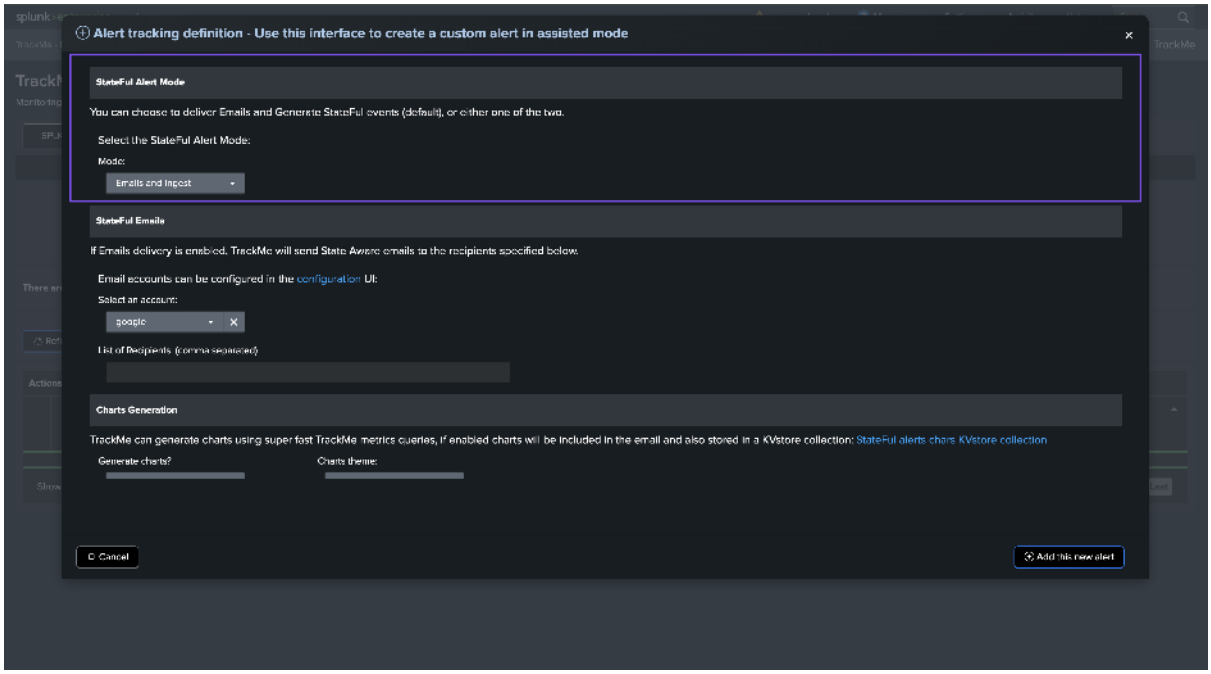
Step 1: Select the Stateful Alert Mode

The first step is to select the alert mode, available options are:

- `Emails and Ingest`: generate both emails deliveries and ingest events
- `Emails, Commands and Ingest`: generate emails deliveries, active commands and ingest events
- `Commands and Ingest`: generate active commands and ingest events
- `Commands and Emails`: generate active commands and emails deliveries
- `Emails only`: generate emails deliveries only
- `Ingest only`: generate ingest events only
- `Commands only`: generate active commands only

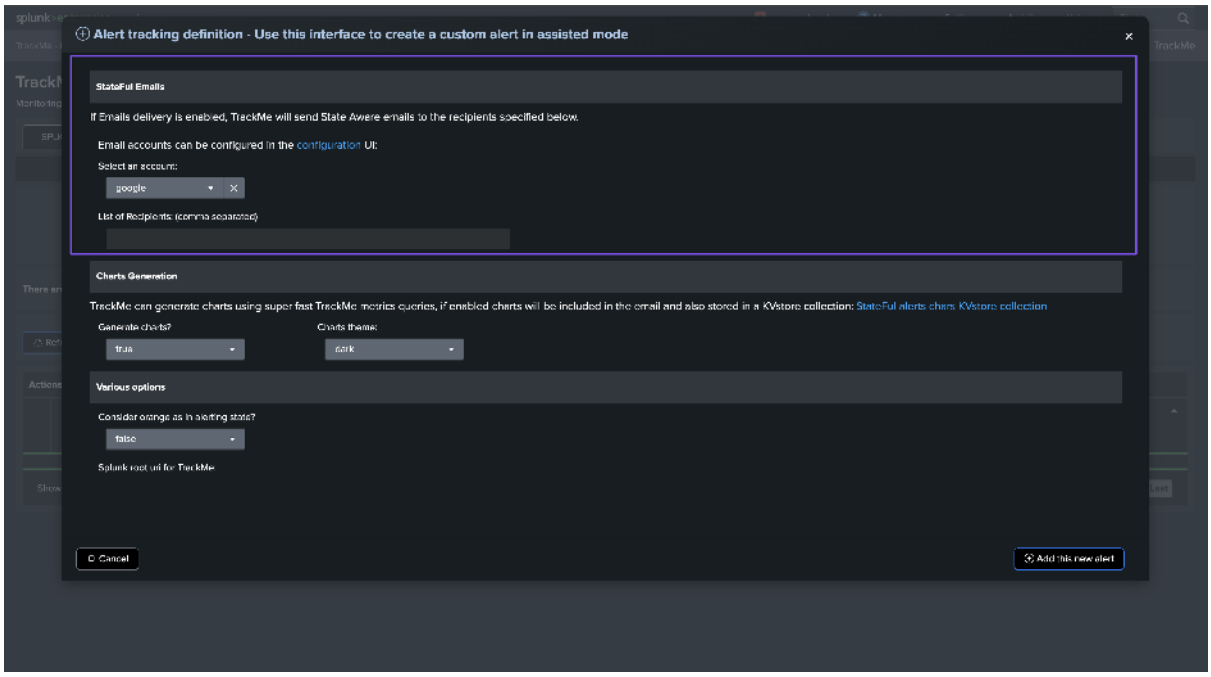
Ingest actions

- We recommend to always have the ingest actions enabled, unless you have a specific use case with multiple alerts.



Step 2: Configure emails delivery, if wanted

If you selected the Emails and Ingest mode, you can configure the emails delivery options:



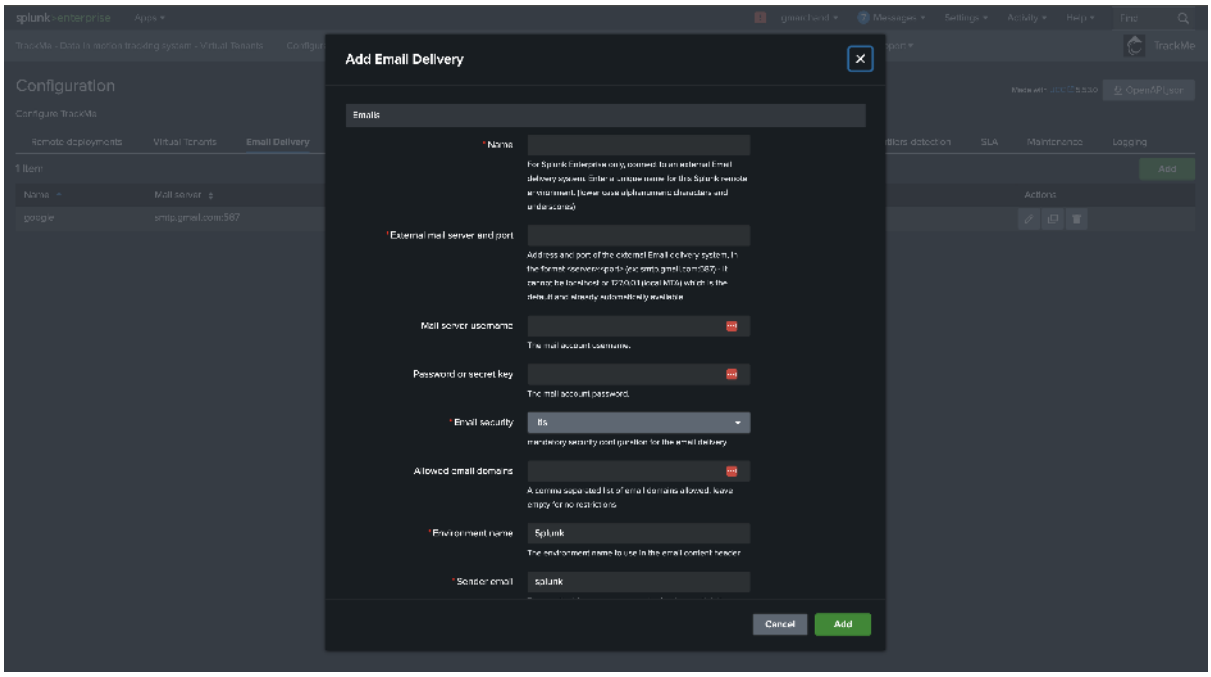
For Splunk Cloud environments:

- You cannot configure an external SMTP destination in Splunk Cloud; the local MTA is mandatory.
- Therefore, for Splunk Cloud, you do not need any further configuration; TrackMe uses the local MTA as a default.

For Splunk Enterprise environments:

- You can choose to use the local MTA or configure an external SMTP destination.
- You can configure an external SMTP destination in Splunk Enterprise, and TrackMe will use it to deliver emails.

In the Configuration UI, go to the Emails tab and configure the SMTP destination, note that using SSL or TLS is mandatory:

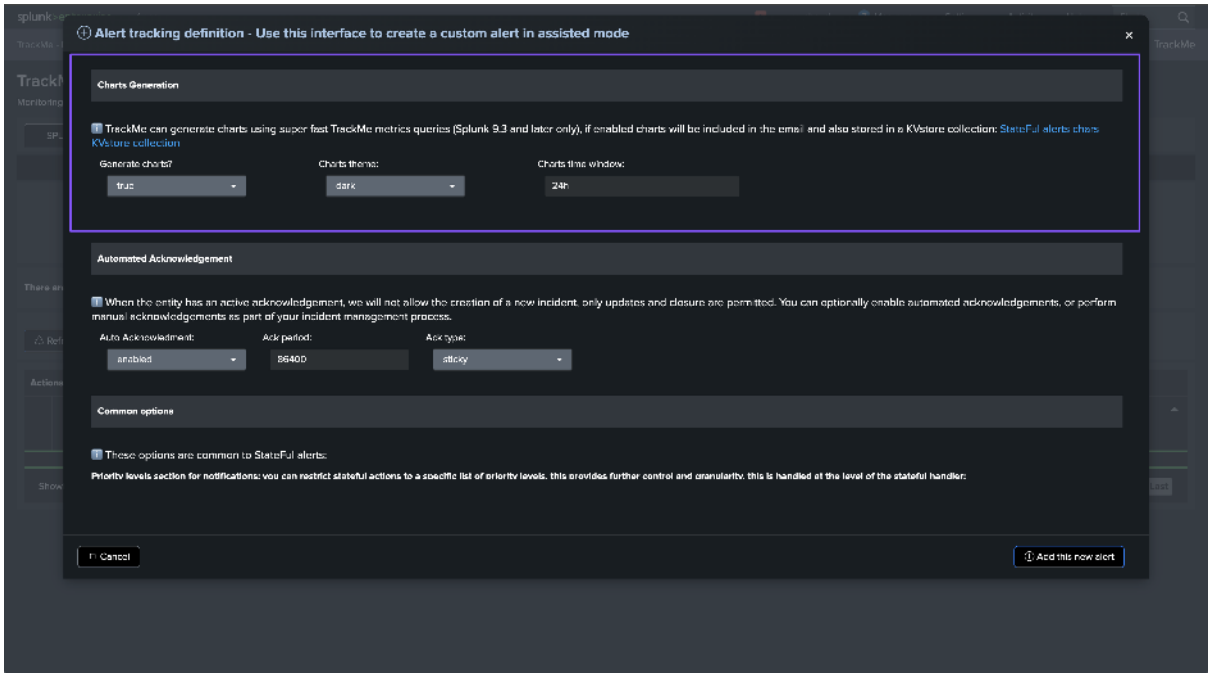


Step 3: Charts generation

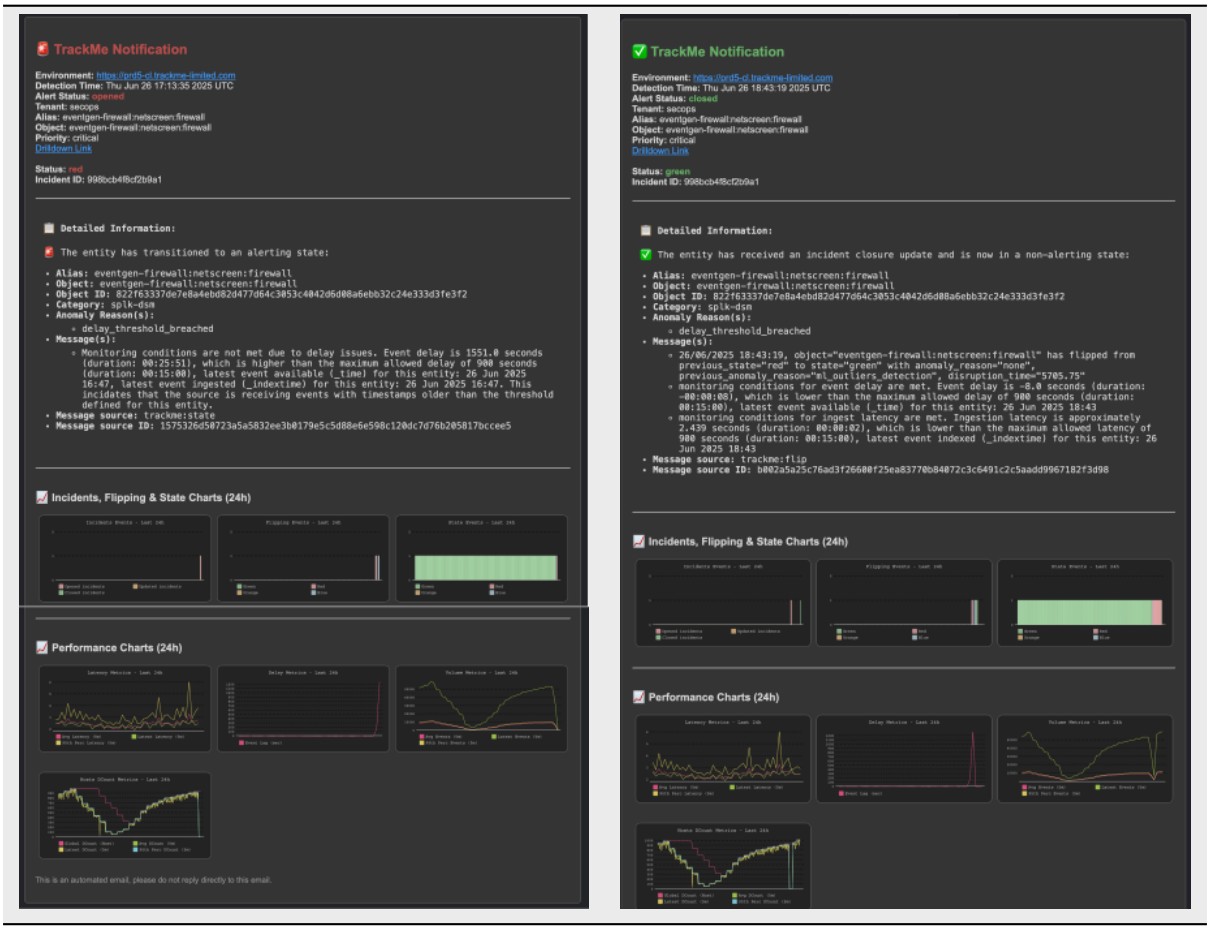
About Emails clients compatibility

- Emails charts use SVG images format, some limited email clients such as Web Gmail do not support this format and charts are not displayed.

By default, TrackMe generates dynamic charts using super fast TrackMe metrics queries and indexes, and embeds them in the emails notifications:



What does this look like? Very cool!:



TrackMe also stores the charts in base64 format in the persistent KVstore collection, so you can access them later on if needed notably for third party integrations.

Replace mytenant with the name of your Virtual Tenant:

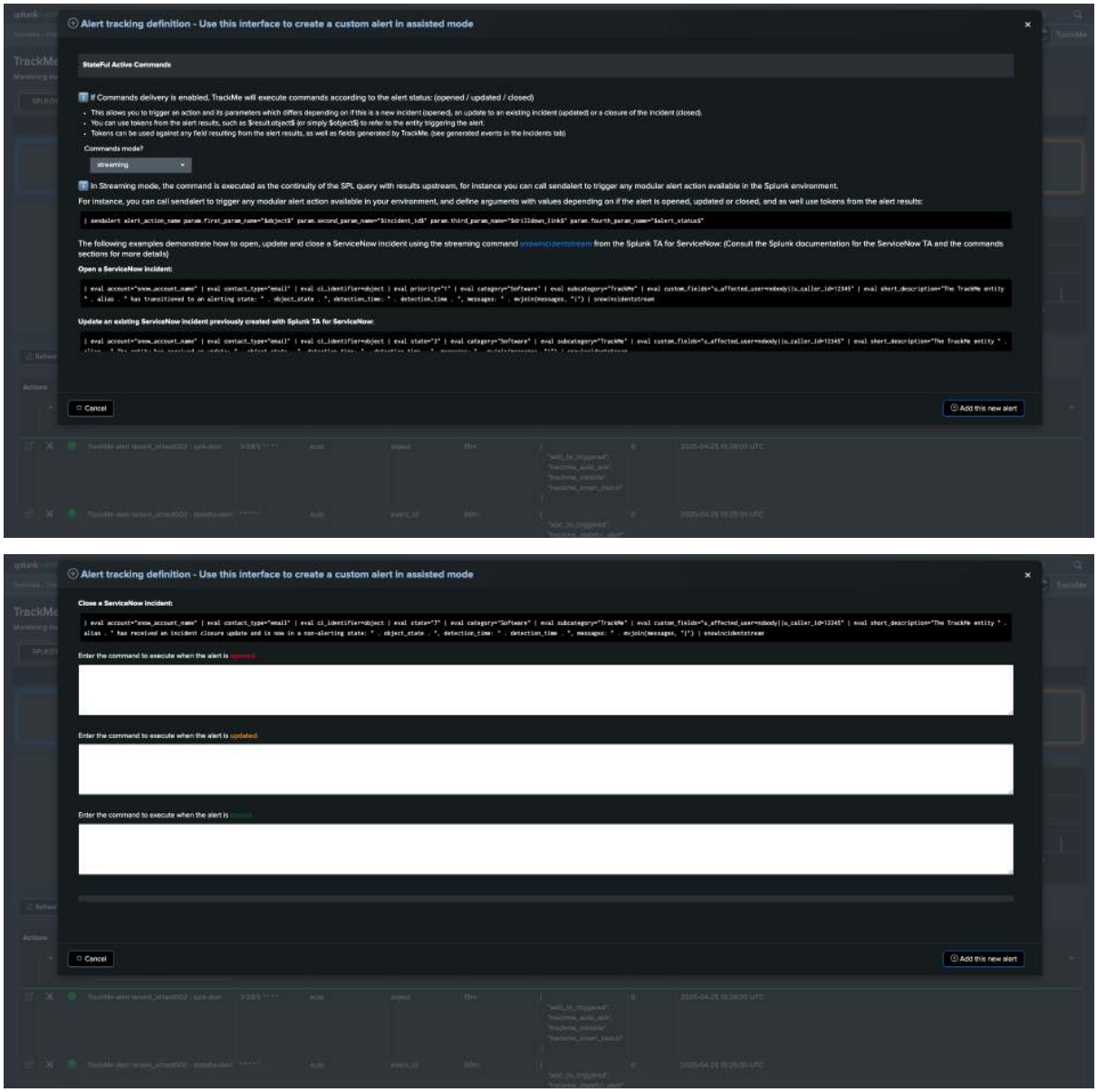
```
| inputlookup trackme_stateful_alerting_charts_tenant_mytenant | eval key=_key, _
↪time=mtime
```

You could link stateful alerts events to the charts in the KVstore collection, in a Splunk search:

```
index=trackme_summary sourcetype=trackme:stateful_alerts tenant_id=mytenant
| lookup trackme_stateful_alerting_charts_tenant_mytenant incident_id OUTPUT chart_
↪svg_base64
```

Step 4: Active commands

In TrackMe, active commands can be generated, depending on the incident status.

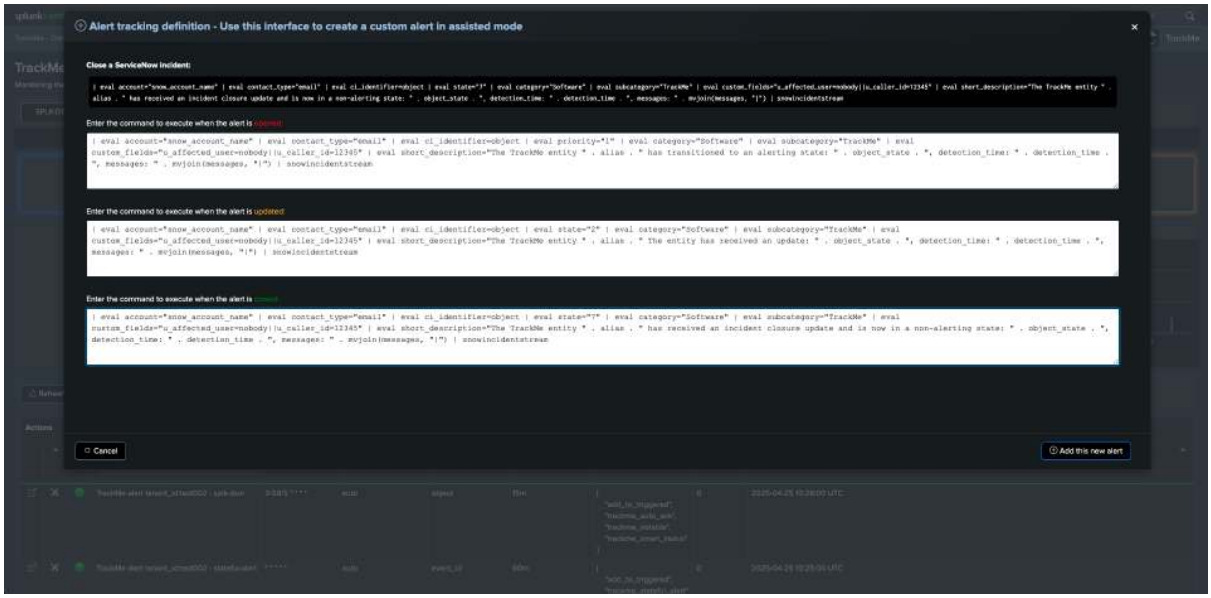


Streaming commands example: Creating / Updating / Closing incidents in ServiceNow

This implementation example uses the Splunk TA for ServiceNow application and its built-in command to create, update, and close incidents.

For more details about the command usage, consult the ServiceNow TA documentation.

When creating the TrackMe alert, setup the active commands for each state of the alert: (opened, updated, closed)



These are examples of commands that need to be adapted to your ServiceNow context:

commands_created:

```
| eval account="snow_account_name" | eval contact_type="email" | eval ci_
↪identifier=object | eval priority="1" | eval category="Software" | eval subcategory=
↪"TrackMe" | eval custom_fields="u_affected_user=nobody||u_caller_id=12345" | eval␣
↪short_description="The TrackMe entity " . alias . " has transitioned to an alerting␣
↪state: " . object_state . ", detection_time: " . detection_time . ", messages: " .␣
↪mvjoin(messages, "|") | snowincidentstream
```

commands_updated:

```
| eval account="snow_account_name" | eval contact_type="email" | eval ci_
↪identifier=object | eval state="2" | eval category="Software" | eval subcategory=
↪"TrackMe" | eval custom_fields="u_affected_user=nobody||u_caller_id=12345" | eval␣
↪short_description="The TrackMe entity " . alias . " The entity has received an␣
↪update: " . object_state . ", detection_time: " . detection_time . ", messages: " .␣
↪mvjoin(messages, "|") | snowincidentstream
```

commands_closed:

```
| eval account="snow_account_name" | eval contact_type="email" | eval ci_
↪identifier=object | eval state="7" | eval category="Software" | eval subcategory=
↪"TrackMe" | eval custom_fields="u_affected_user=nobody||u_caller_id=12345" | eval␣
↪short_description="The TrackMe entity " . alias . " has received an incident␣
↪closure update and is now in a non-alerting state: " . object_state . ", detection_
↪time: " . detection_time . ", messages: " . mvjoin(messages, "|") |␣
↪snowincidentstream
```

You can call tokens at any stage of the command, using the format:

- `field_name="$result.field_name$"`

Or:

- `field_name="$field_name$"`

Tokens are applicable against any field resulting from the alert, as well as fields added by TrackMe, you can consult an example of an incident event in the `Incidents` tab, see the anatomy of StateFul Incidents Events section for more details.

Generating commands example

TrackMe can also call a generating command instead, which you can leverage to tackle your specific use case.

The following example shows a simple usage for demonstration purposes using the Splunk `makeresults` command, where we would add a subset of the content to a lookup file, which could then be consumed by a third-party logic.

commands_created:

```
| makeresults | eval request_type="new", ci="$object$", ci_id="$object_id$", alert_
↪status="$alert_status$", event_id="$event_id$",description="The TrackMe entity
↪$alias$ has transitioned to an alerting state: $object_state$, detection_time:
↪$detection_time$" | outputlookup append=t trackme_incidents.csv
```

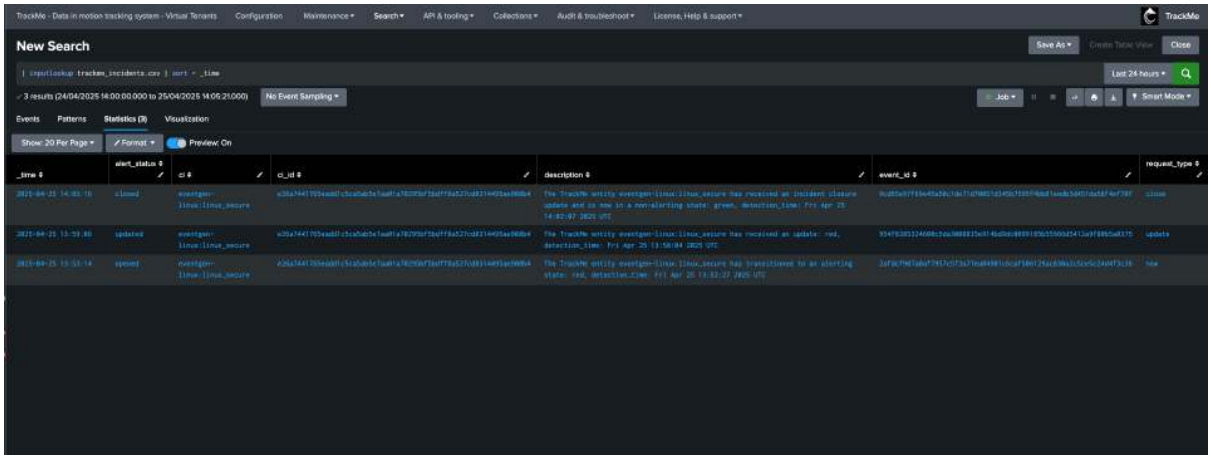
commands_updated:

```
| makeresults | eval request_type="update", ci="$object$", ci_id="$object_id$", alert_
↪status="$alert_status$", event_id="$event_id$", description="The TrackMe entity
↪$alias$ has received an update: $object_state$, detection_time: $detection_time$" |␣
↪outputlookup append=t trackme_incidents.csv
```

commands_closed:

```
| makeresults | eval request_type="close", ci="$object$", ci_id="$object_id$", alert_
↪status="$alert_status$", event_id="$event_id$", description="The TrackMe entity
↪$alias$ has received an incident closure update and is now in a non-alerting state:
↪$object_state$, detection_time: $detection_time$" | outputlookup append=t trackme_
↪incidents.csv
```

Example of a cycle of incidents for a same entity:



Troubleshooting active commands execution

Hint

About the user context

- It is important to remember that the command is executed in the context of the owner of the alert.
- When it comes to TrackMe, the owner of the alert is handled automatically by TrackMe to correspond to the owner of the Virtual Tenant.

- By default, the owner when creating Virtual Tenants in TrackMe is nobody. (splunk-system-user)
- Therefore, keep this in mind while troubleshooting the command execution, as a third party command might potentially require specific permissions and capabilities to be available for the user context.

Consult the logs of the Stateful alert action to review the execution of the active commands:

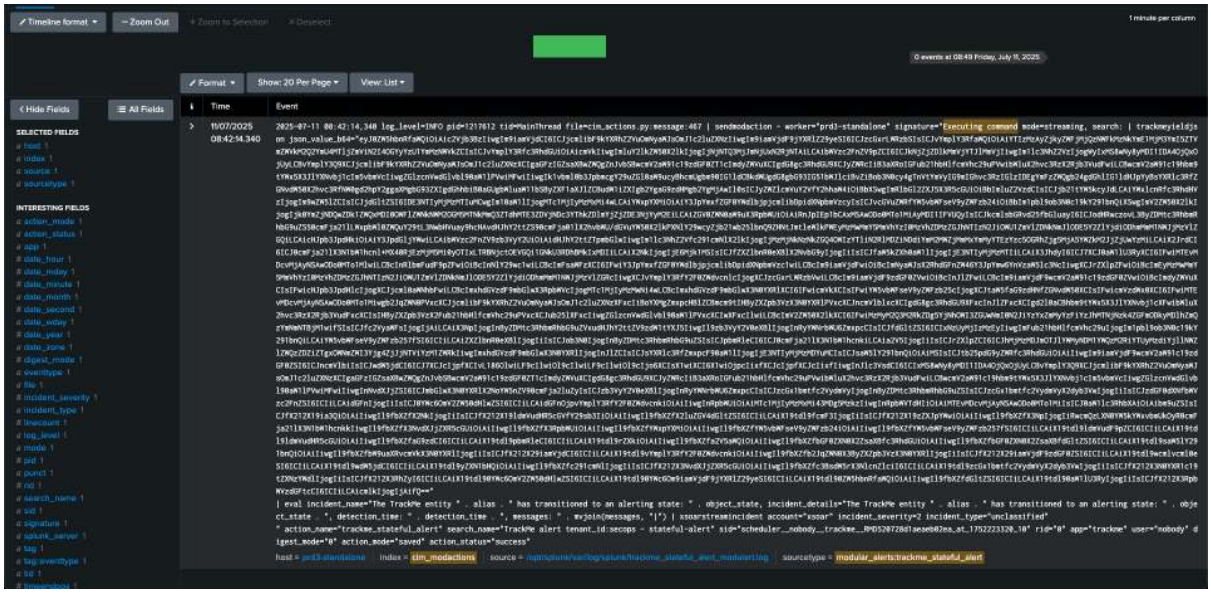
```
index=cim_modactions sourcetype=modular_alerts:trackme_stateful_alert "Command"
```

Especially, the command execution will be logged as:

```
index=cim_modactions sourcetype=modular_alerts:trackme_stateful_alert "Executing␣
↪command"
```

If the active command is executed, but does not return any results, the logs will show an error message, otherwise an informational message will be shown.

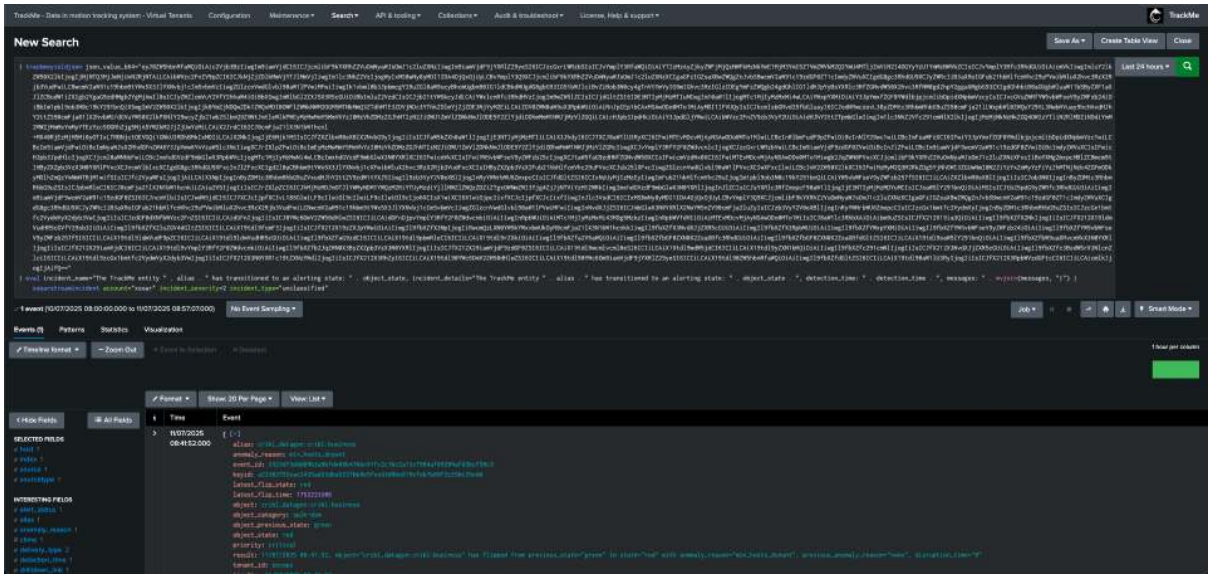
TrackMe processes active commands by converting the upstream results in a base64 encoded string, then executing the trackMe command trackmeyieldjson followed by the requested search logic, such as:



This is a very powerful and flexible approach, you can easily re-execute the command at anytime with 100% of the original context and easily troubleshoot any issue:

```
| trackmeyieldjson json_value_b64="<encoded_base64_string"
```

example:



Step 5: Various options

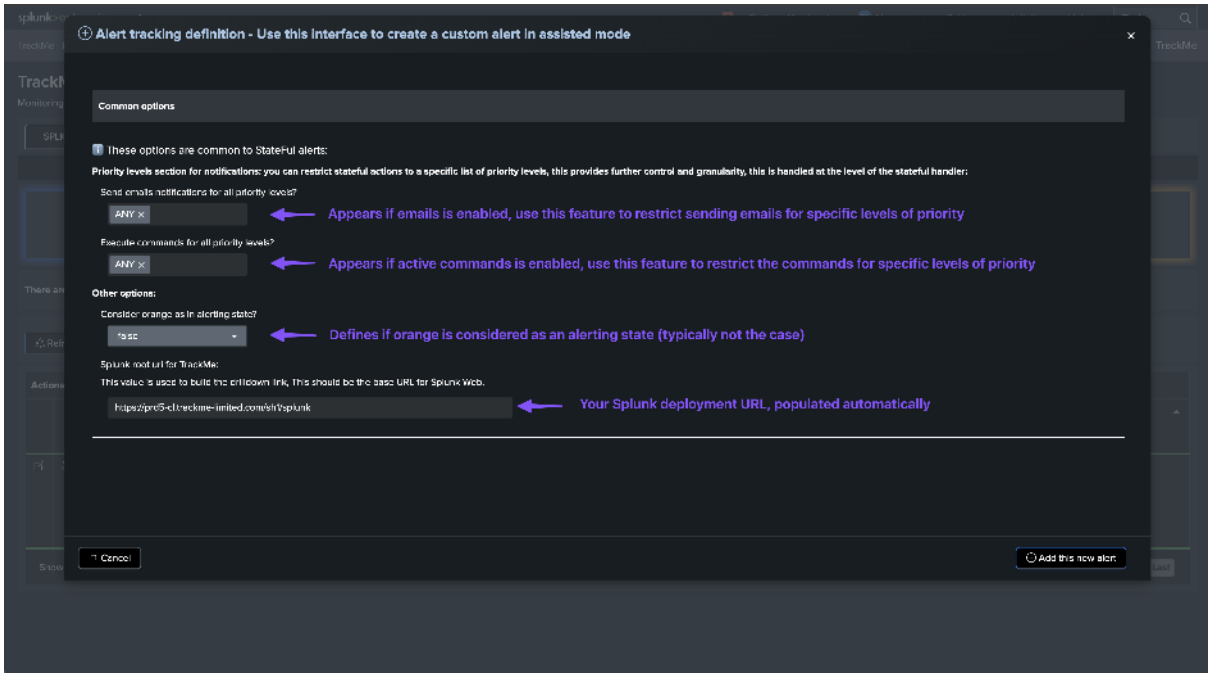
Hint

About the cron schedule: the cron schedule is not shown as configurable anymore, this is made on purpose because the stateful alerts is meant to be execute on a frequent basis of ideally every 5 minutes:

- The stateful handlers tracks changes in the entities from different angles, and then decides if the notification means the beginning of an incident, an update or a closure.
- For this to work as expected, the alert must be execute as soon as possible and it then handles the incident life cycle.
- Therefore, it is **not recommeded to change the default cron schedule which should be set to every 5 minutes.**

Finally, there are various options that you can configure for the alert, such as:

- `Consider orange as an alerting state`: In TrackMe, orange is typically used to indicate a warning state; you can configure TrackMe to consider this as an alerting state.
- `Drilldown URI`: The drilldown URI is used to build the drilldown link in the email notifications.
- `Priority`: The priority of entities for triggering the alert; you may, for instance, want to trigger the alert only on high or critical priority entities.



Additional information

TrackMe implements various dynamic actions, depending on the circumstances or the type of entities it is dealing with, to adapt automatically the behavior of Stateful alerts.

- **Time detection, deduping and backfilling:** TrackMe automatically identifies if events should be skipped according to the entity cycle, this means that it will skip events if necessary and will never send duplicated alerts, whatever the triggers of the alerts are. (so you can have an overlapped search approach safely and TrackMe does the hard work!)
- **Dynamic charting:** TrackMe automatically chooses which metrics should be charted, and how, based on the entity type and metrics available.
- For instance, for the splk-dsm component, it will automatically add component specific charts such as data sampling charts. (quality issues)
- For components like splk-flx and splk-wlk, TrackMe will automatically verify metrics availability for the entity, and will generate charts accordingly.
- **Status flexibility:** TrackMe has different statuses, you can influence for instance if orange should be considered as an alerting state or not. On the other hand, "blue" which is a special status (notably for logical grouping) is automatically considered as a non alerting state, eligible for incident closures.

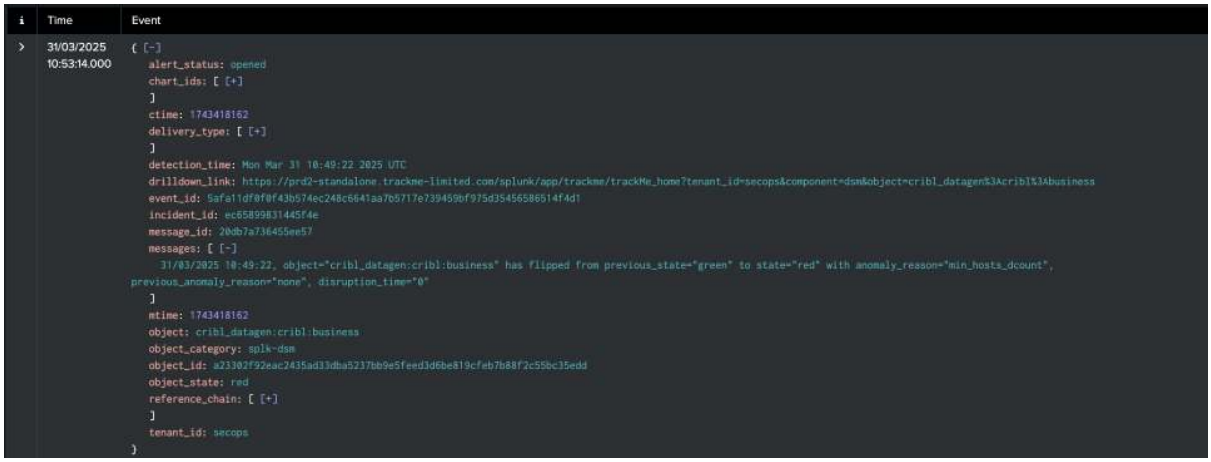
Anatomy of Stateful Incidents Events

In addition with rich email notifications, TrackMe also generates incident events, which can easily be used to trigger third party integrations.

- alert_status: `opened` - This is the start of the incident
- alert_status: `updated` - This is an update of the incident
- alert_status: `closed` - This is the closure of the incident

Example of an opening incident event:

```
index=trackme_summary sourcetype="trackme:stateful_alerts" alert_status=opened tenant_
↪id=* object_category=* object=*
```



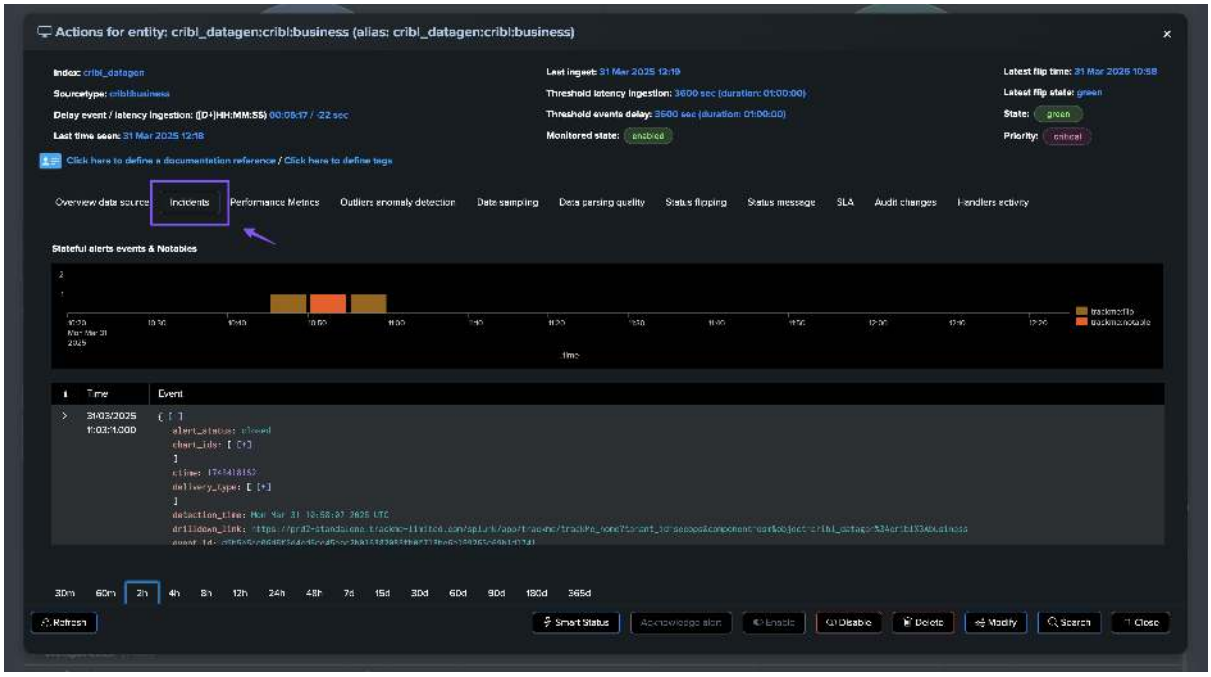
Example of an updating incident event:

```
index=trackme_summary sourcetype="trackme:stateful_alerts" alert_status=updated␣
↪tenant_id=* object_category=* object=*
```

Example of a closing incident event:

```
index=trackme_summary sourcetype="trackme:stateful_alerts" alert_status=closed tenant_
↪id=* object_category=* object=*
```

In TrackMe, incidents events can be accessed through the UI and the Incidents tab (which was renamed from Notables to Incidents in this release):



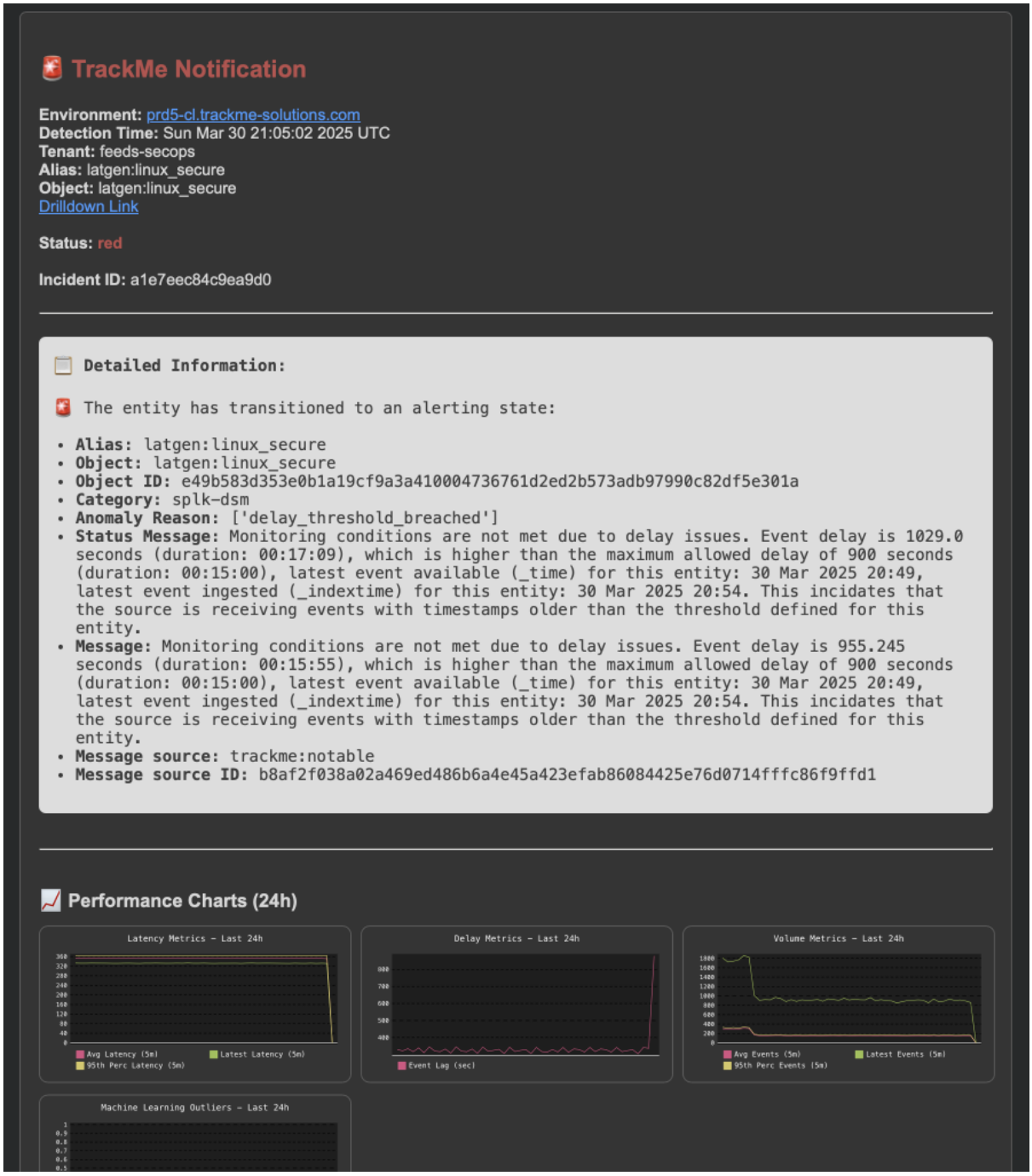
TrackMe Stateful alerts events are `JSON` formatted events, these contain the following key information:

Field	Description
alert_statu	The status of the alert, this can be `opened`, `updated` or `closed`
chart_ids	The list of chart ids associated with the incident, these can be used to retrieve the charts from the KVstore collection
ctime	The creation time of the incident (epoch)
delivery_type	The list of delivery types that were called for this incidents, such as `email` and `ingest`
detection_time	The time of the detection of the alert in human readable format (strftime %c %Z)
drilldown_link	A link to the TrackMe Home user interface which filters on the tenant, component and entity, and automatically opens the entity overview screen (from TrackMe 2.0.88)
event_id	The unique identifier for this notable event, as the sha-256 hash of the notable event content (from TrackMe 2.1.0)
incident_id	The unique identifier for this incident, this is shorter identifier which is common to events and objects that related to the same incident
message_id	A unique identifier for this message, this is principally used for emails
messages	The list of messages from TrackMe that occur during the incident life cycle, this related to the TrackMe status_message field
mtime	The last modification time of the incident (epoch)
object	The object name for this entity, this basically is the name of entities in TrackMe
object_catego	A TrackMe identifier for each component, this describes which type of TrackMe component this notable is related to.
object_id	The unique identifier for this object in the tenant / component central KVstore collection
object_state	The current state of the object (blue / green / orange / red)
reference_chain	The list of message_id that relate to the same incident, this is used for emails threading purposes
tenant_id	The tenant identifier this Notable event is related to.
priority	The priority definition for this object (low / medium / high / critical)
source_mes	The TrackMe sourcetype at the origin of the incident, such as trackme:flip
source_mes	The unique identifier (event_id) of the original event that triggered the incident

Example: Feed tracking entity (splk-dsm)

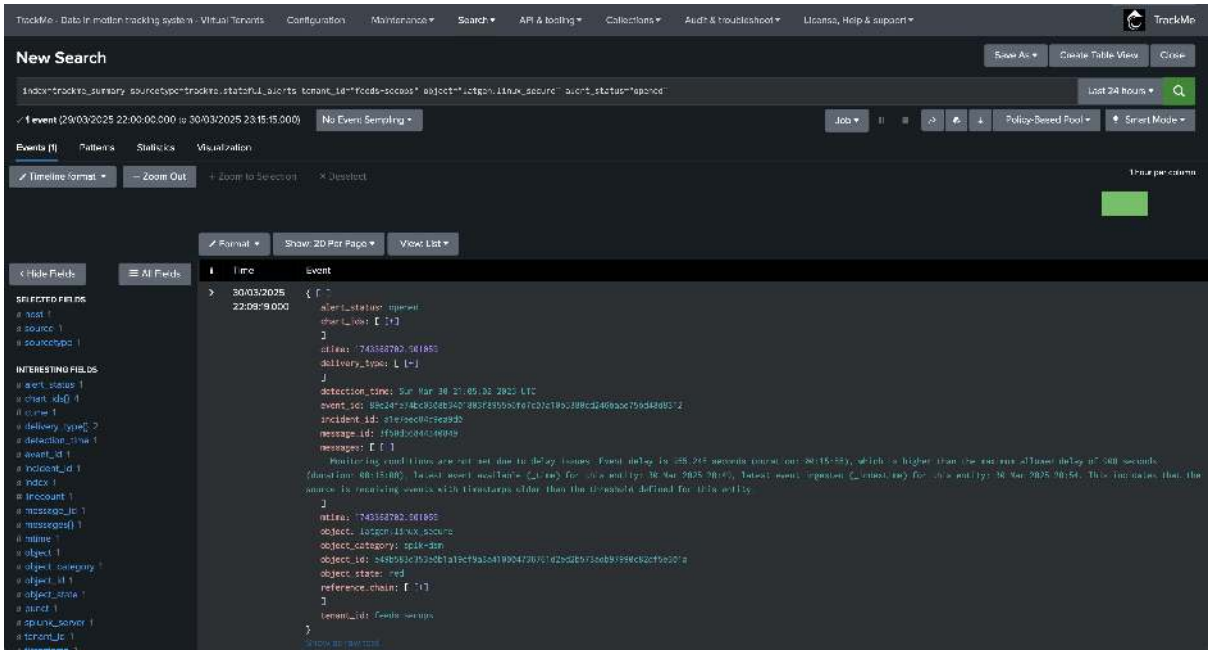
In this example, a sourcetype is affected by high latency and/or delay, a first notification is generated:

A first notification is generated, either through a notable event of a flip event:

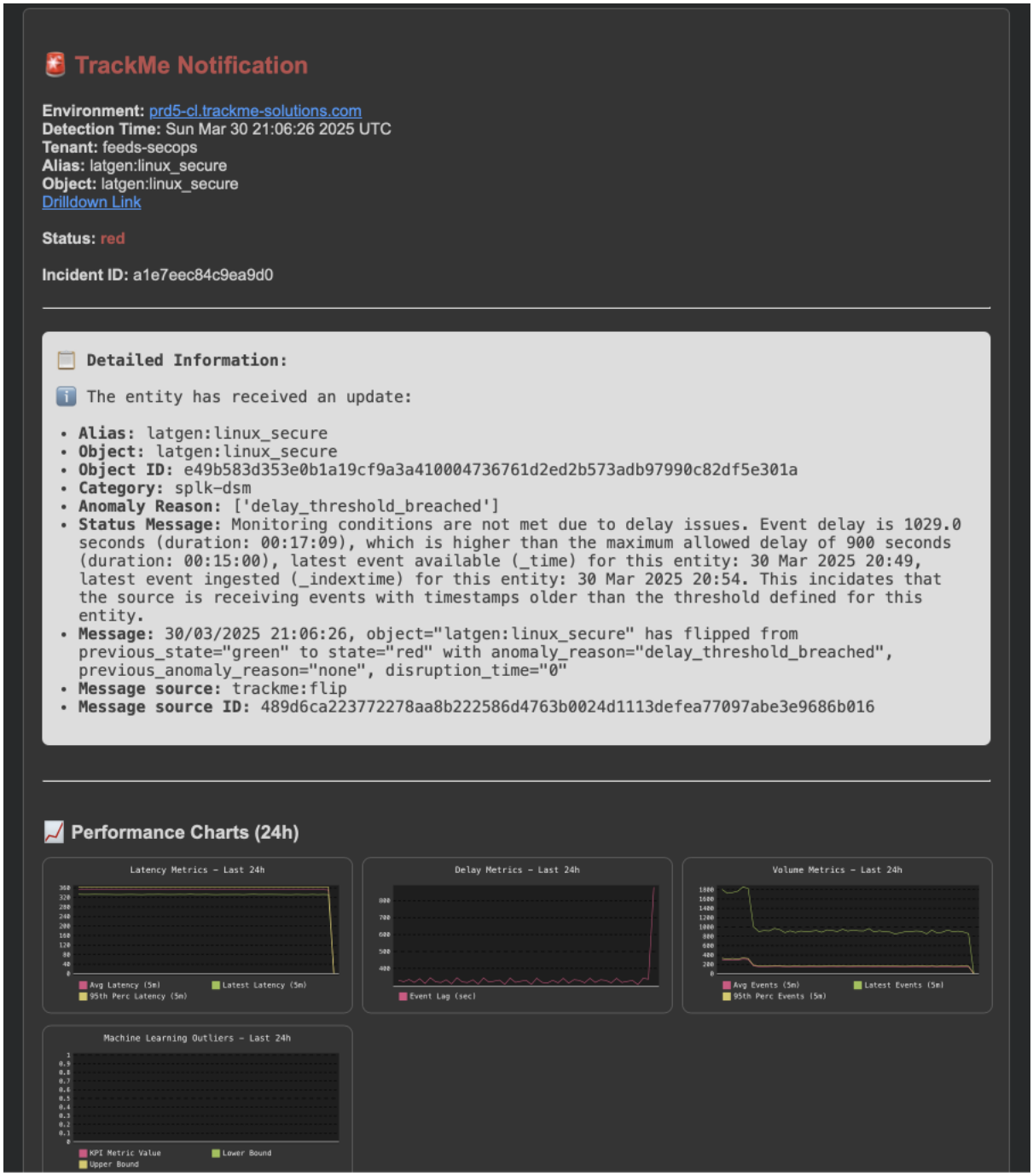


An opening incident event is created:

```
index=trackme_summary sourcetype=trackme:stateful_alerts tenant_id="mytenant" object=
↪"myobject" alert_status="opened"
```

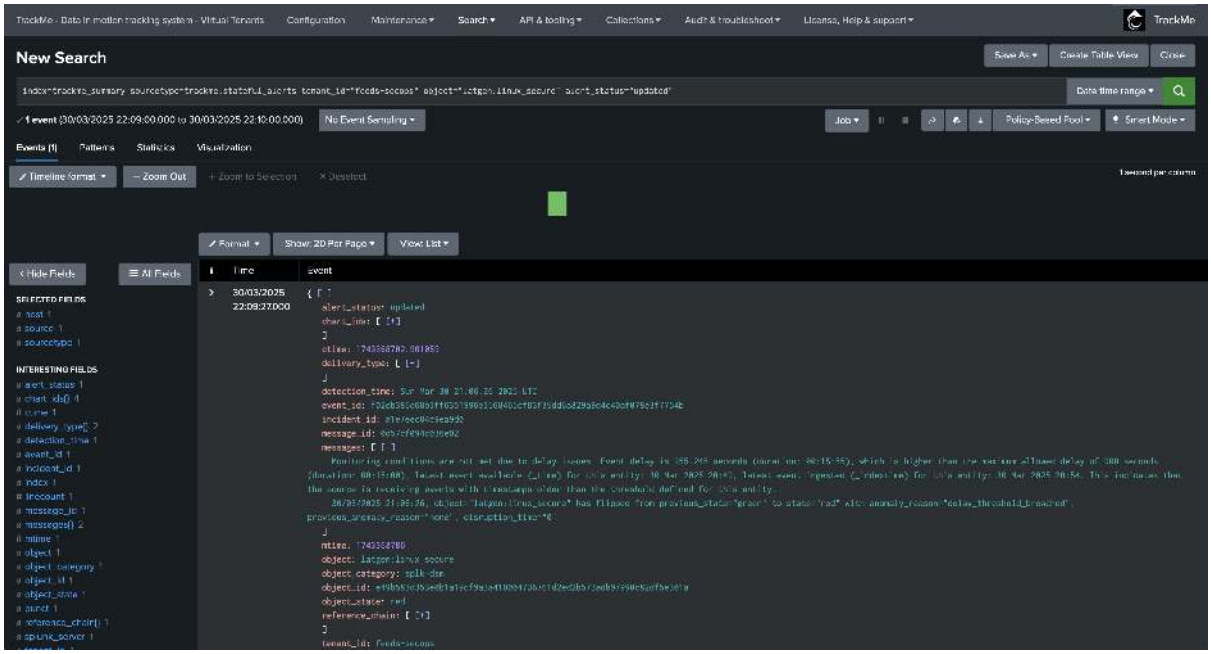


Eventually, an update will be issued if a new event is generated:

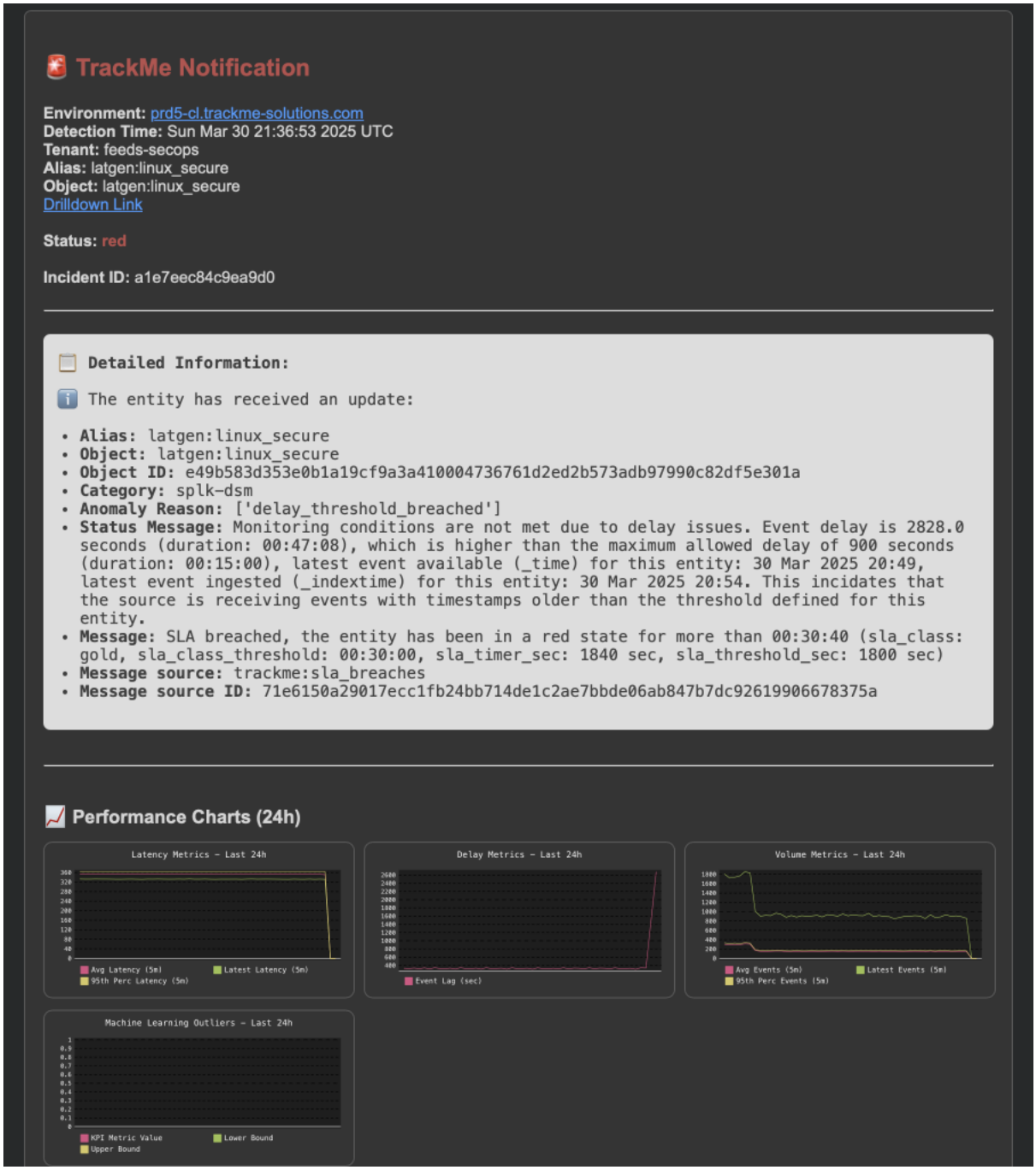


An update incident event is created:

```
index=trackme_summary sourcetype=trackme:stateful_alerts tenant_id="mytenant" object=
↪"myobject" alert_status="updated"
```

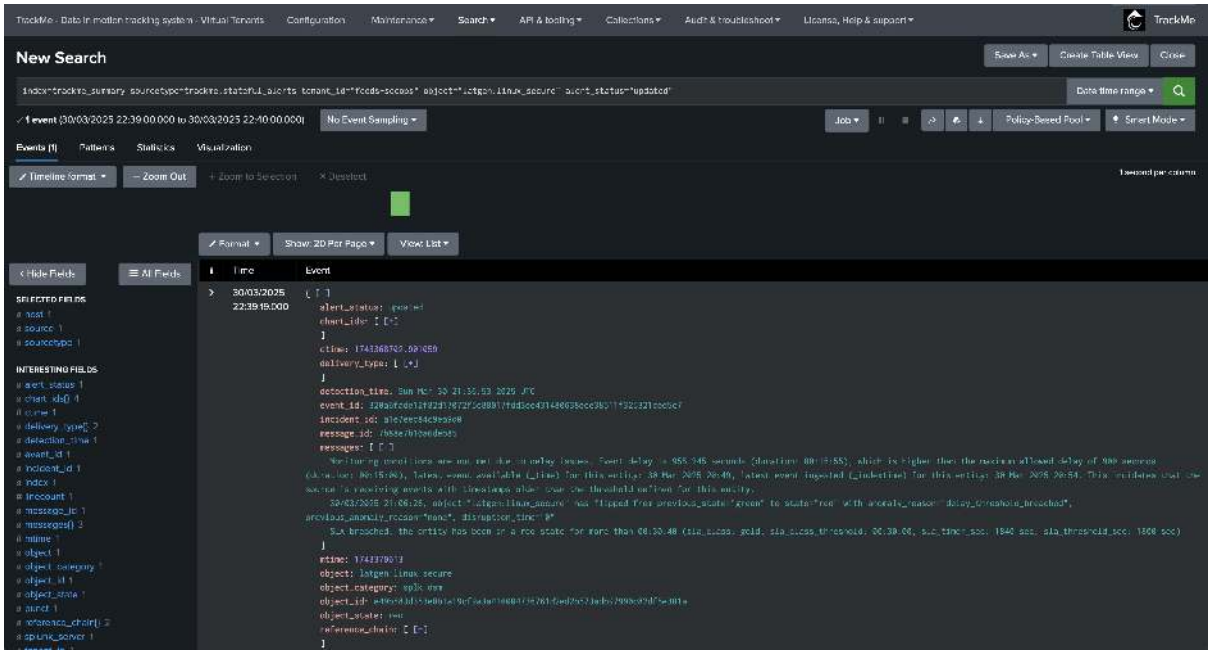


Later on, the SLA is breached, a new notification is generated:

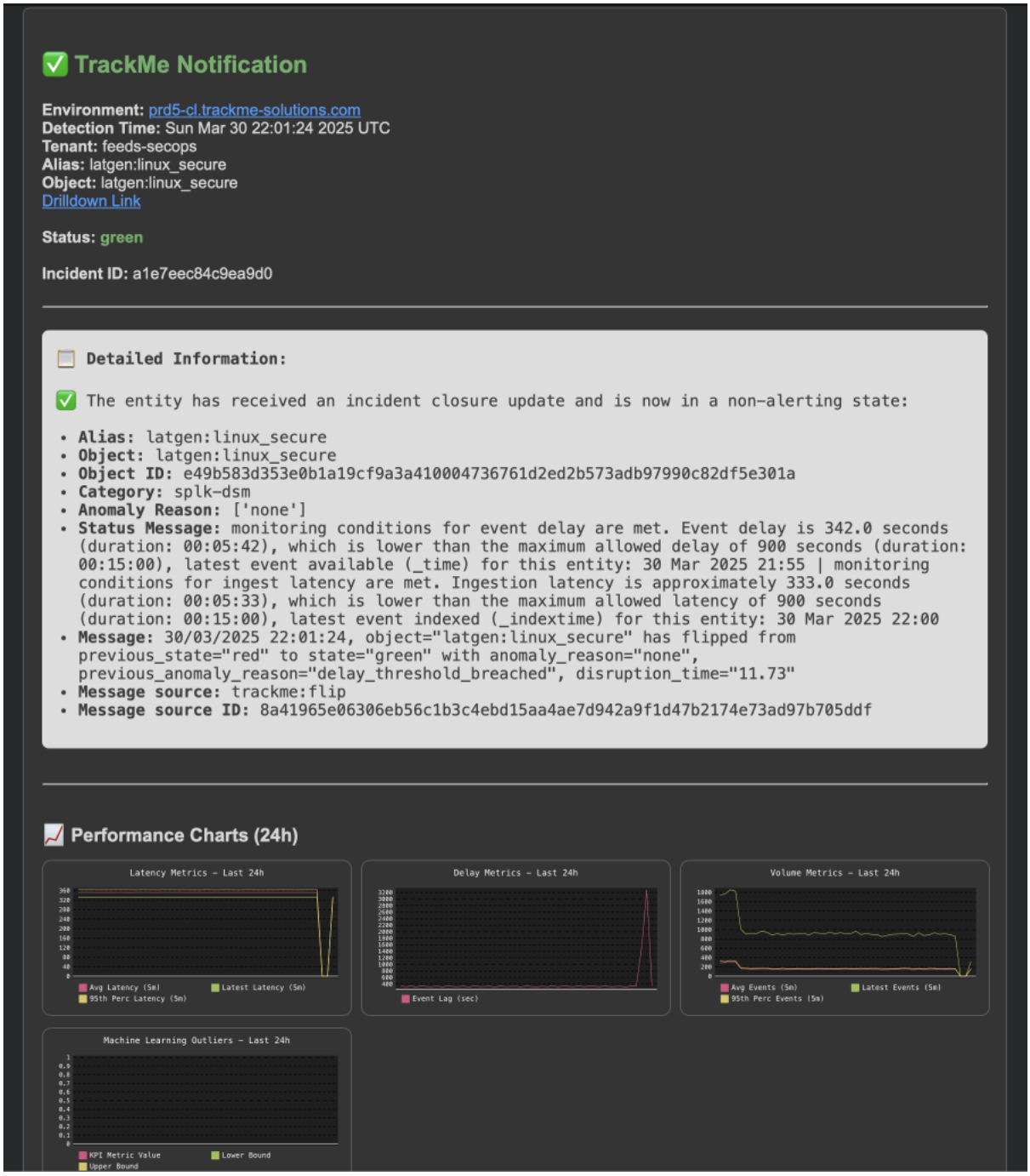


Another update incident event is created:

```
index=trackme_summary sourcetype=trackme:stateful_alerts tenant_id="mytenant" object=
↪"myobject" alert_status="updated"
```

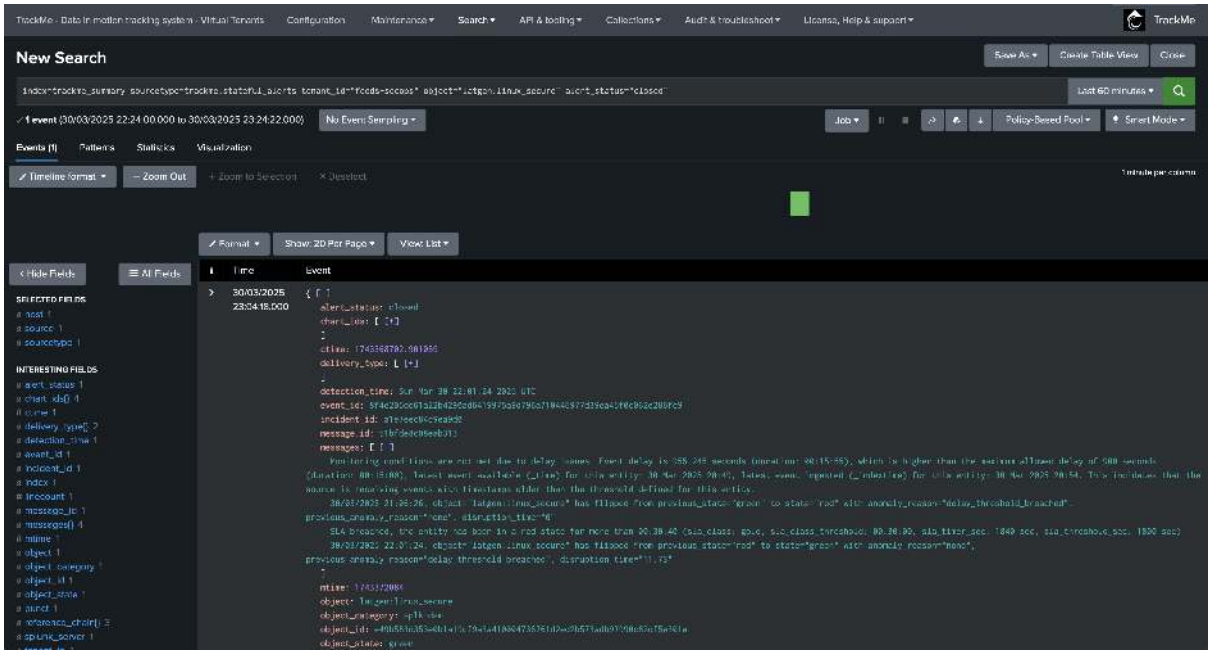


Finally, the issue is resolved, an incident closure notification is generated:



An incident closure event is created:

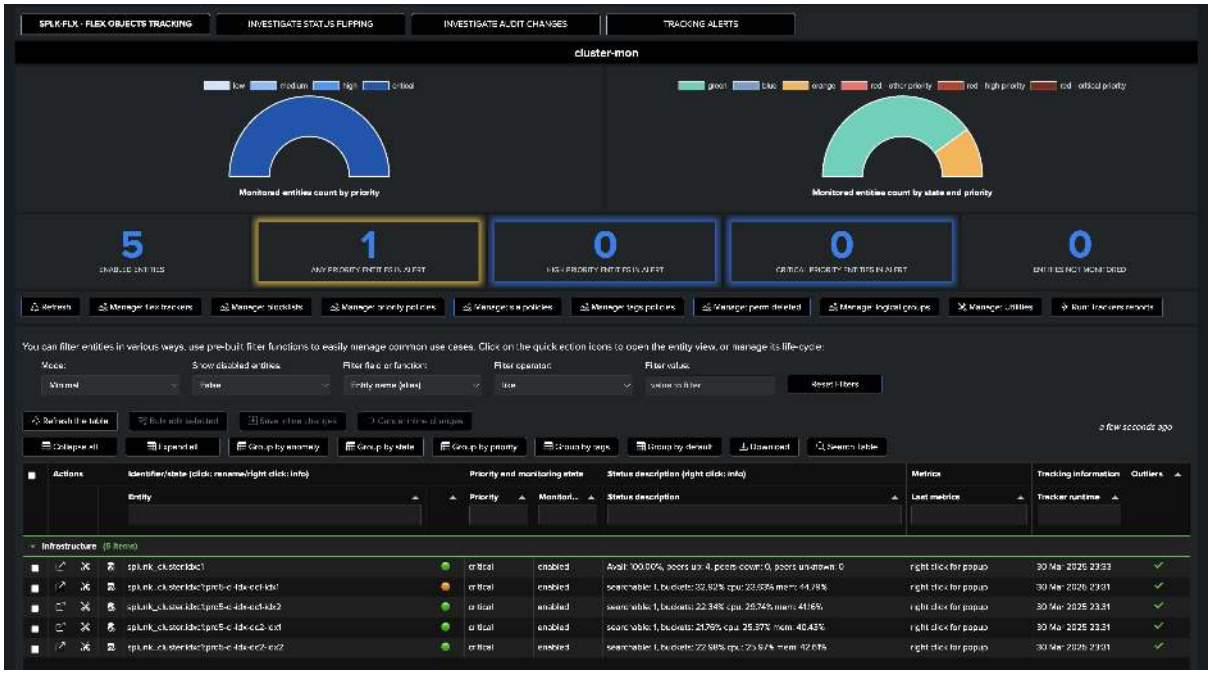
```
index=trackme_summary sourcetype=trackme:stateful_alerts tenant_id="mytenant" object=
↪"myobject" alert_status="closed"
```



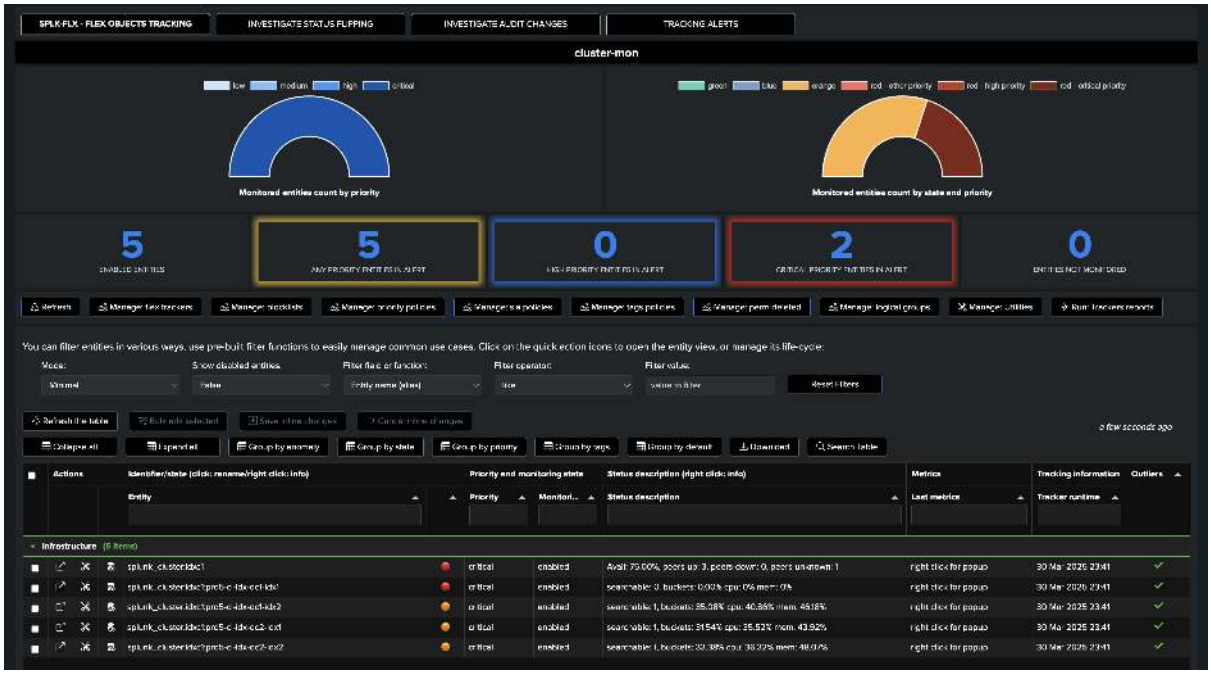
Example: Splunk Cluster Monitoring with splk-flx

In this example, we monitor a Splunk cluster and Splunk indexers through the Flex object component.

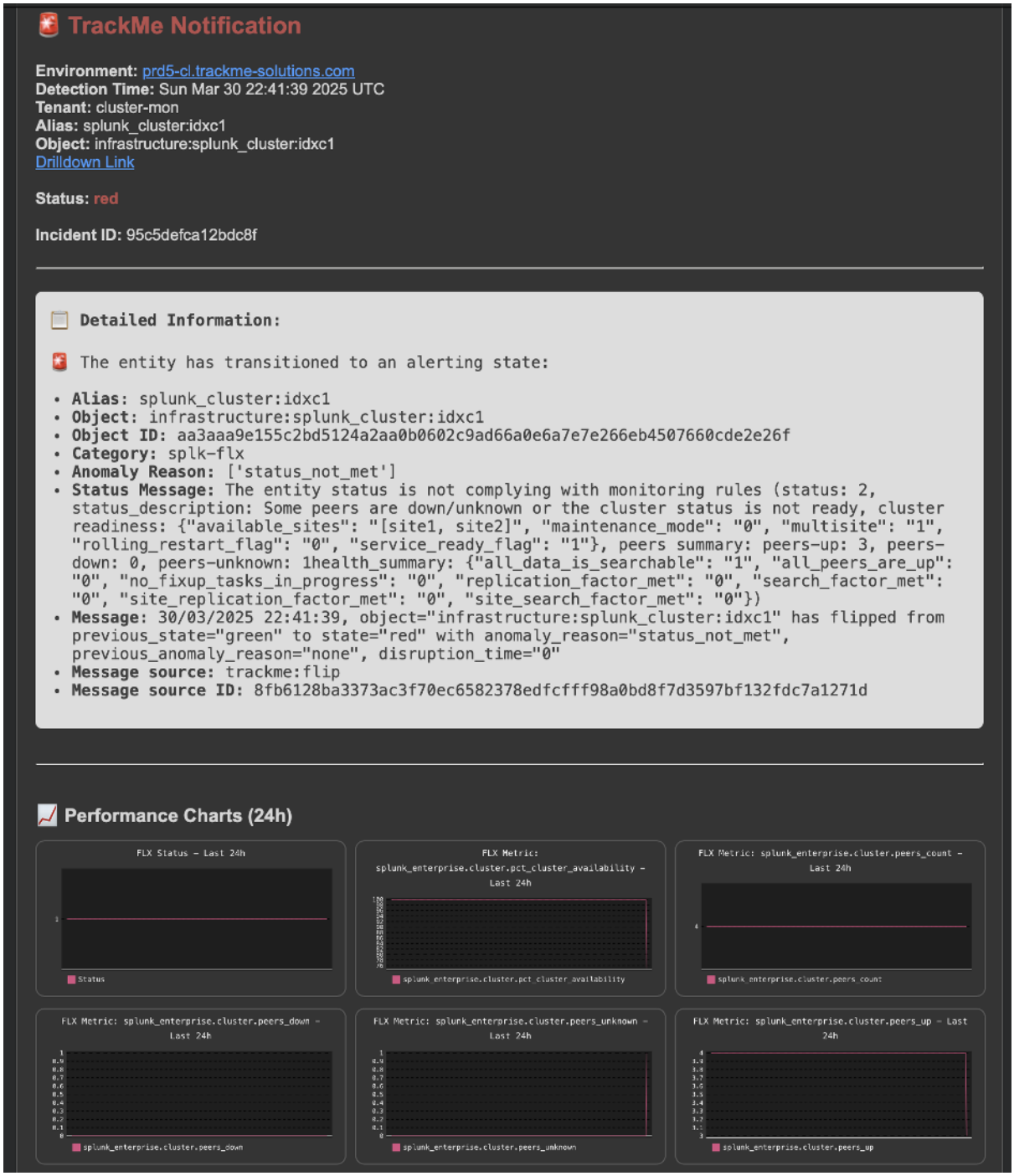
Initial status: The cluster global status is healthy, all peers are up and running: (although one peer is reported as orange due to bucket imbalance)



An incident affects an indexer, the global cluster status as well as the indexer turn into alerting state:

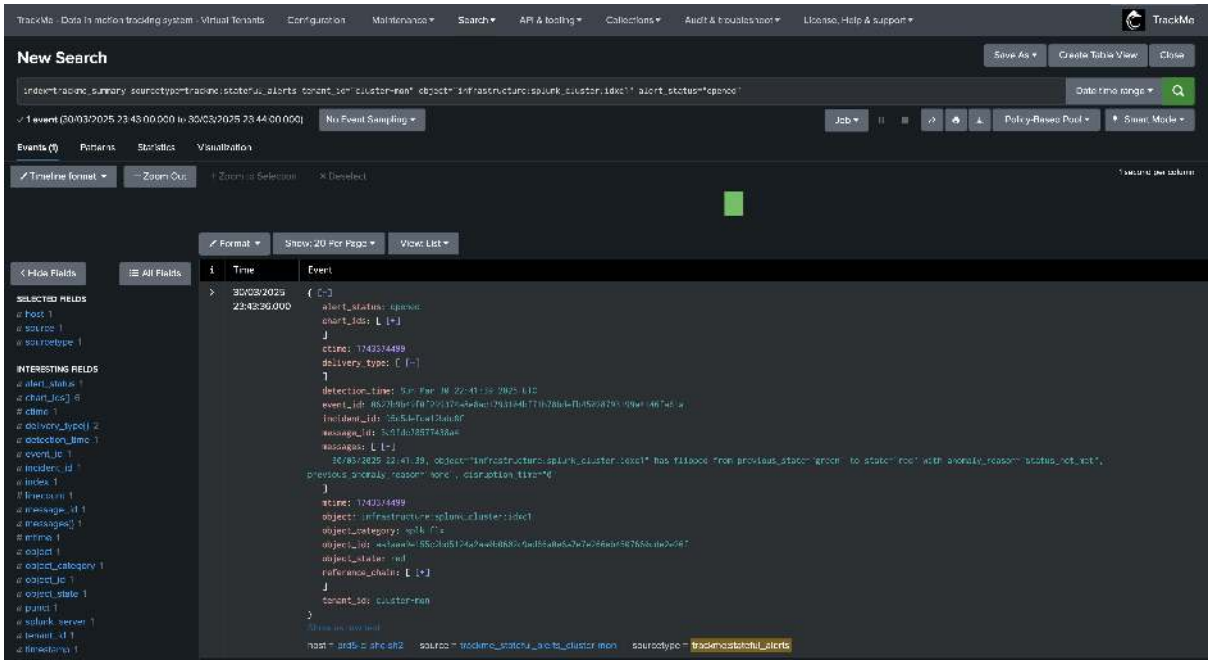


A first notification is generated for the cluster status:

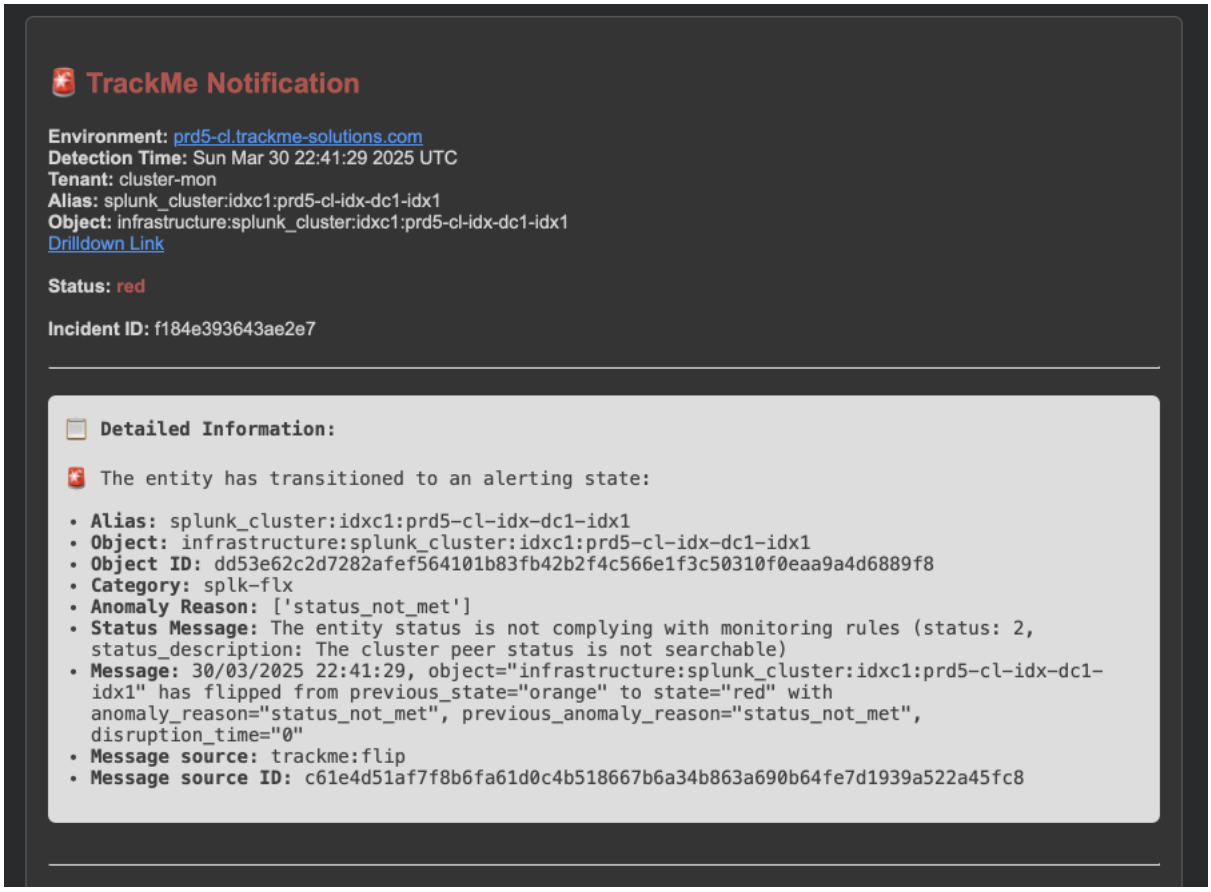


An opening incident event is created:

```
index=trackme_summary sourcetype=trackme:stateful_alerts tenant_id="cluster-mon"␣
↪object="infrastructure:splunk_cluster:idxc1" alert_status="opened"
```

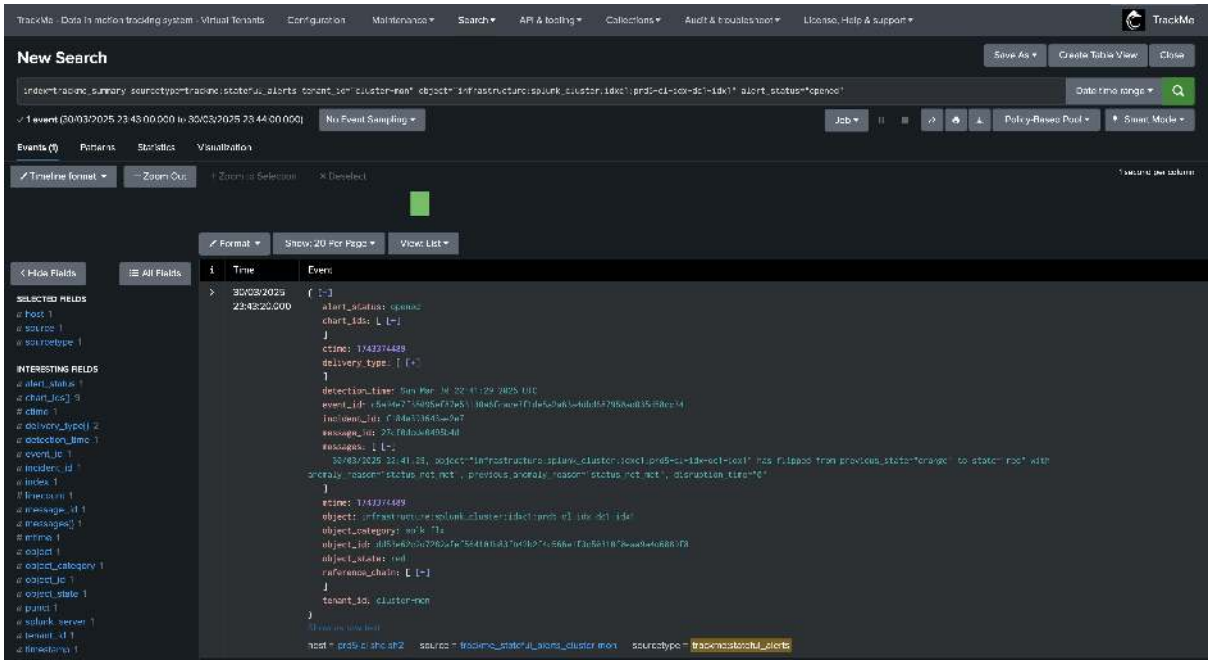


Notifications are also generated for the indexer:

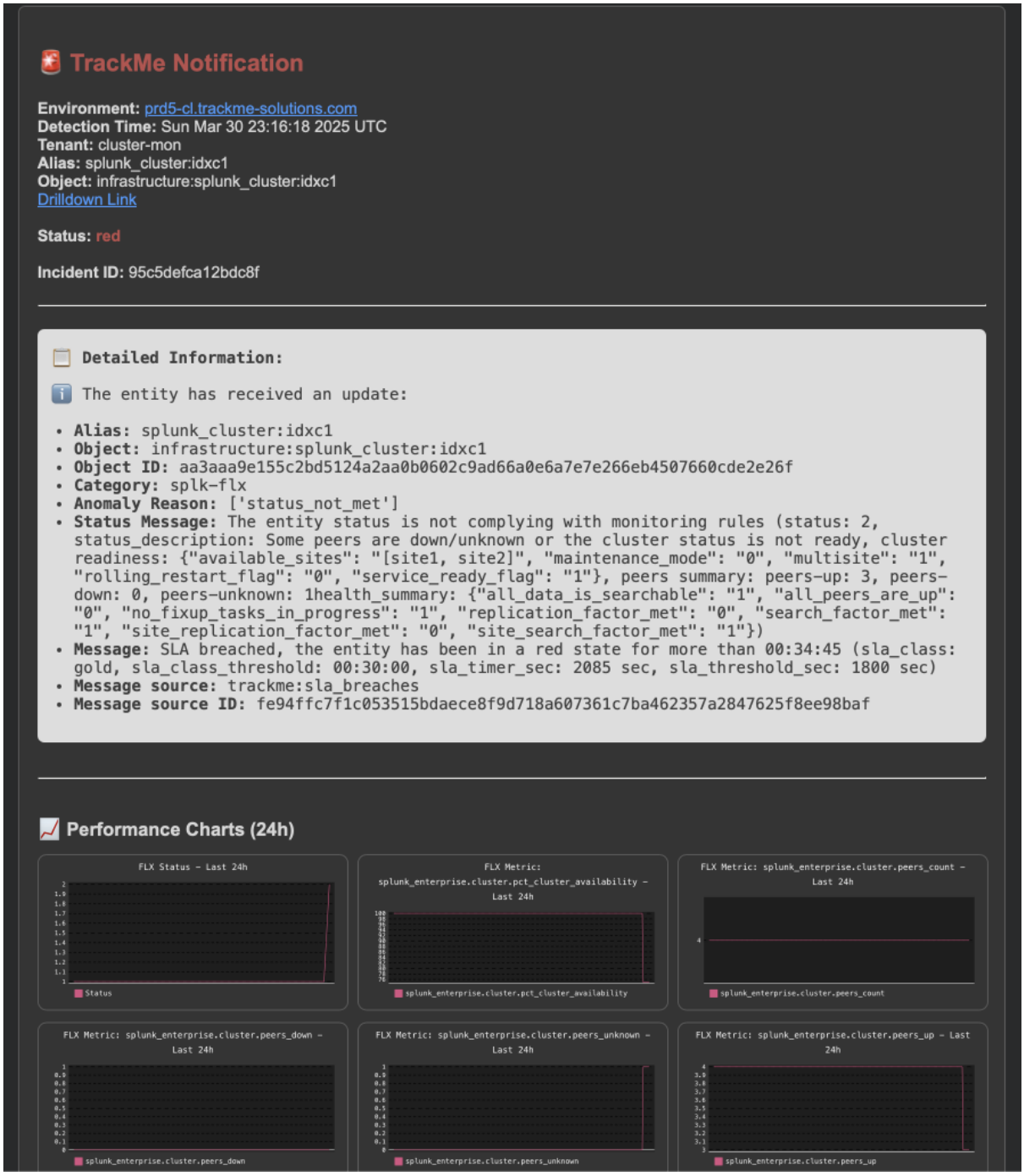


An update incident event is created:

```
index=trackme_summary sourcetype=trackme:stateful_alerts tenant_id="cluster-mon"␣
↪object="infrastructure:splunk_cluster:idxc1" alert_status="updated"
```

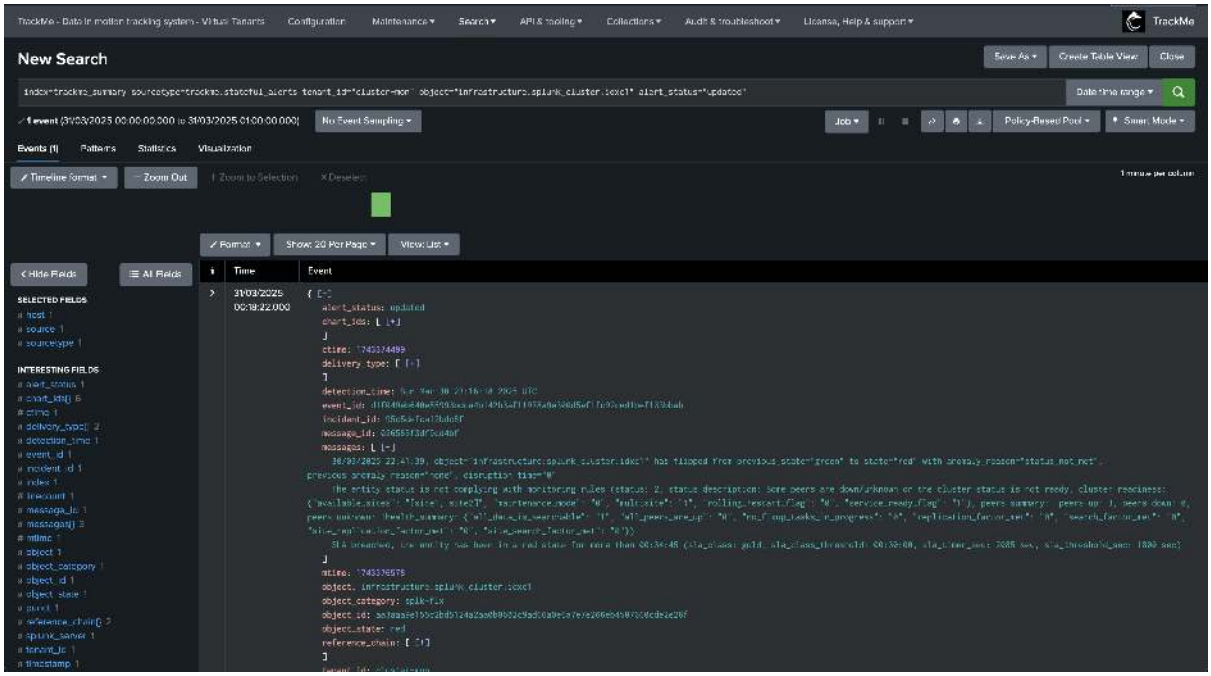


As the incident continues, several notifications can be generated, for instance when the SLA is breached:

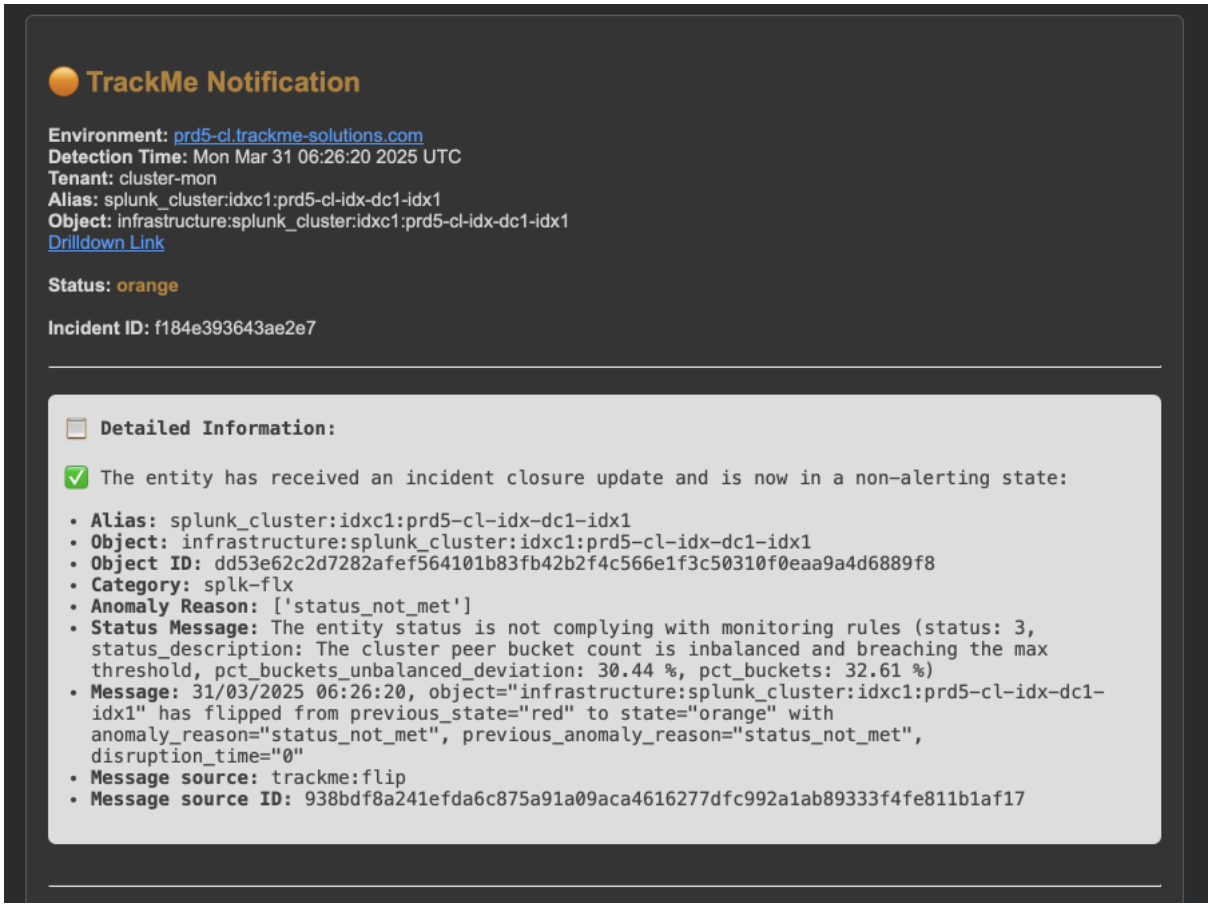


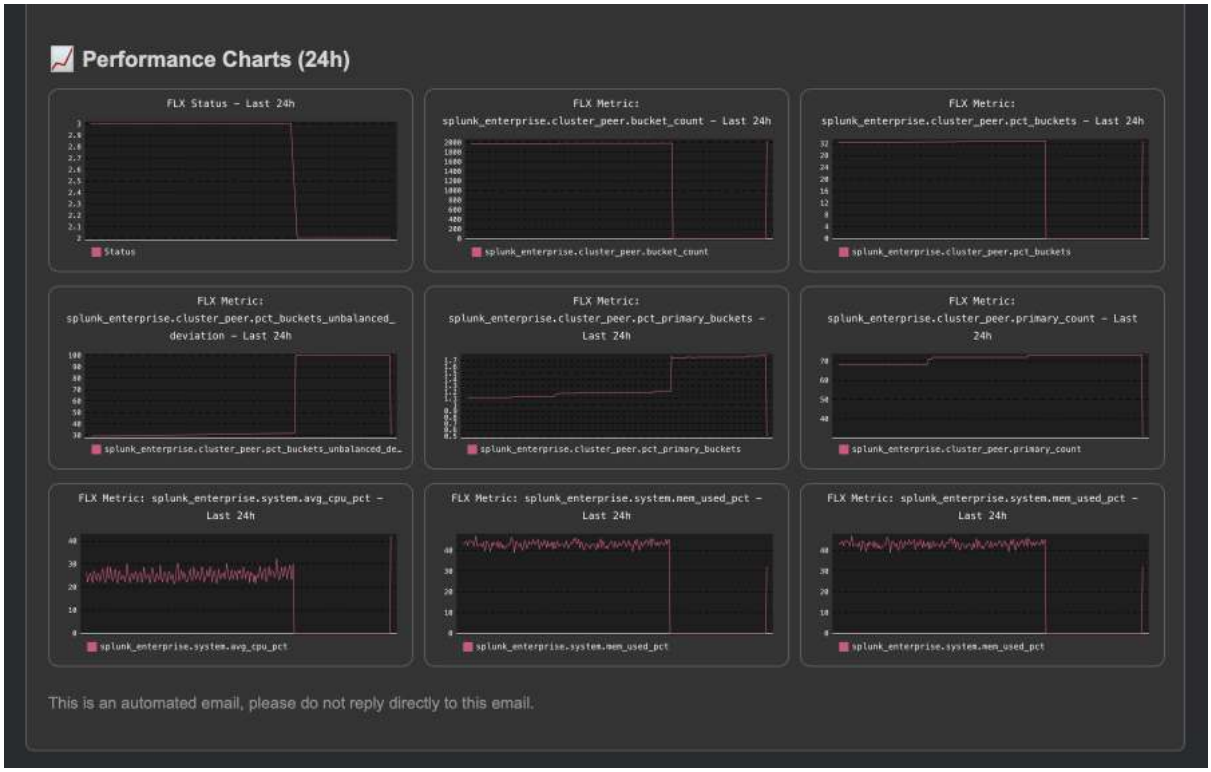
An update incident event is created:

```
index=trackme_summary sourcetype=trackme:stateful_alerts tenant_id="cluster-mon"␣
↪object="infrastructure:splunk_cluster:idxc1" alert_status="updated"
```



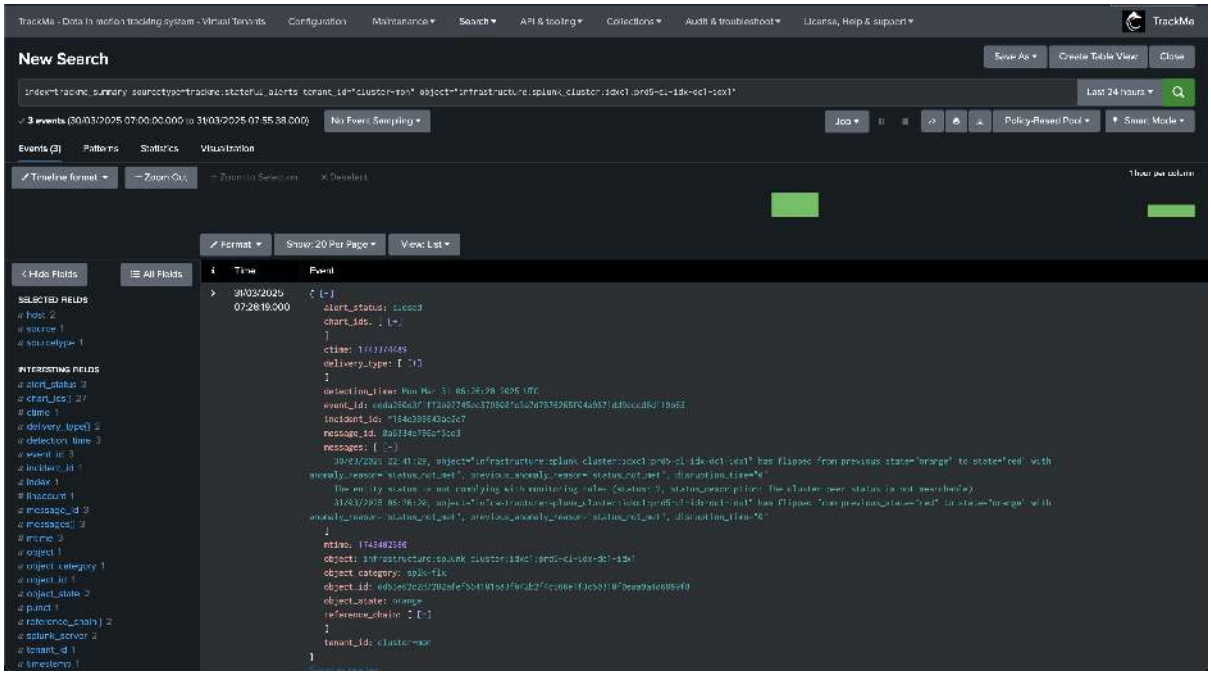
At some point, the issue on our indexer is resolved, the entity returned to a non alerting state (here orange due to bucket imbalance), the incident is closed:



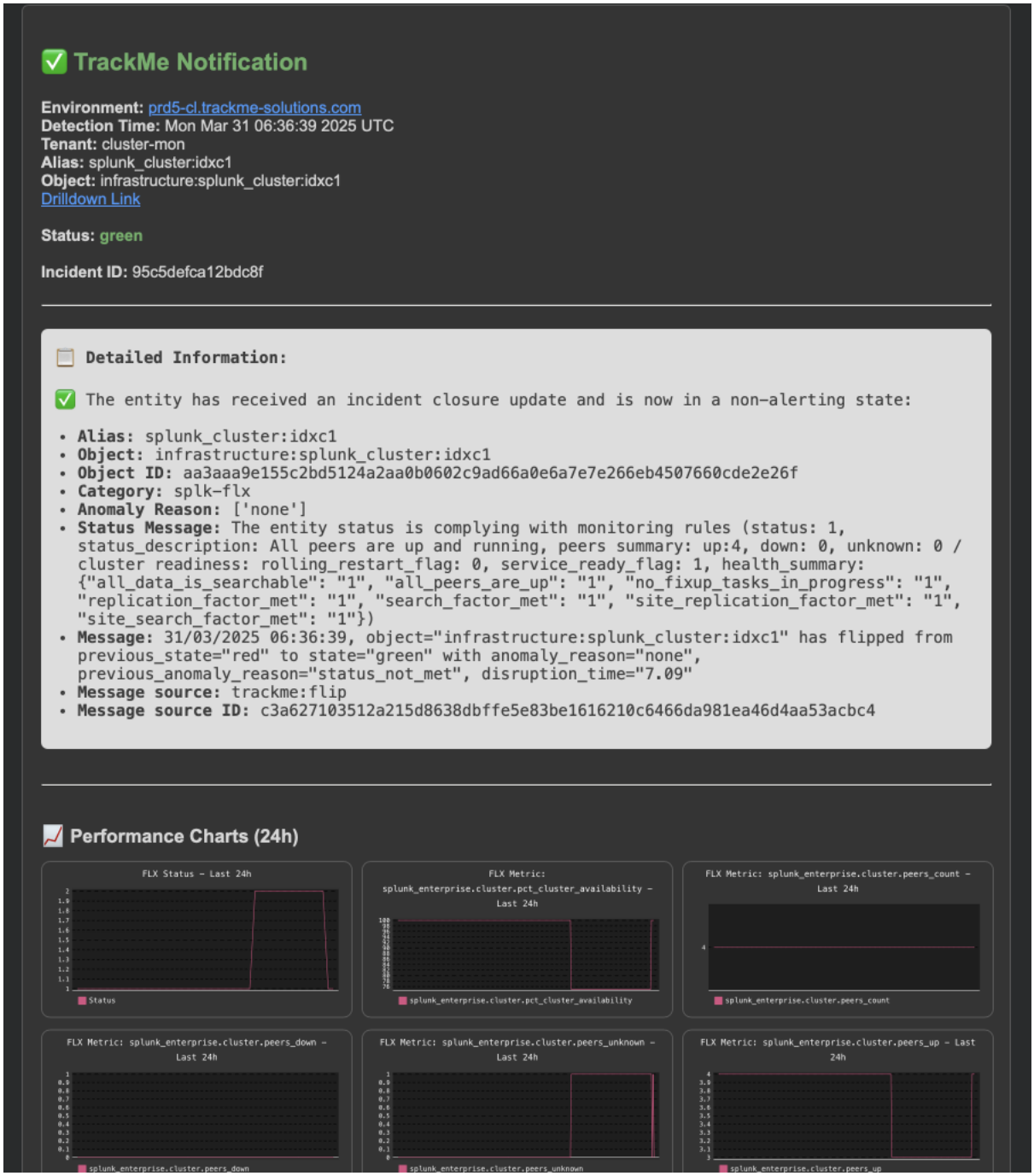


An incident closure event is created:

```
index=trackme_summary sourcetype=trackme:stateful_alerts tenant_id="cluster-mon"␣
↪object="infrastructure:splunk_cluster:idxc1:prd5-cl-idx-dc1-idx1" alert_status=
↪"closed"
```

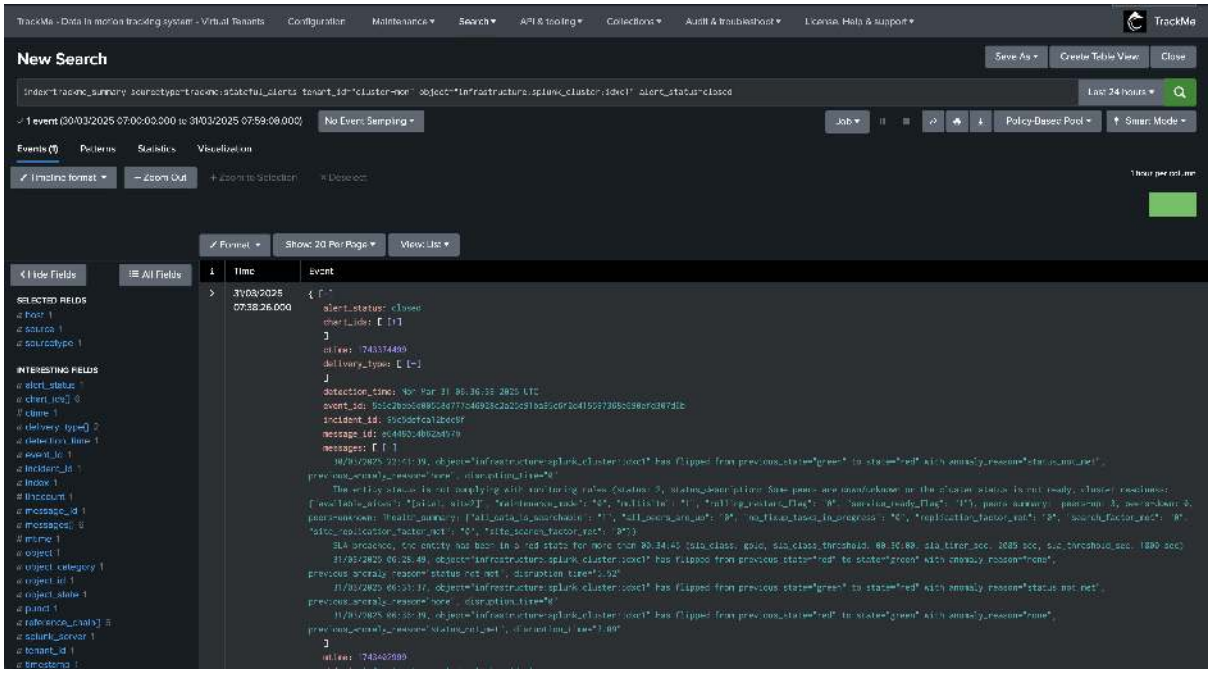


Finally, after some time, our indexer cluster catches up and the entity returns to a healthy state, the incident is closed:

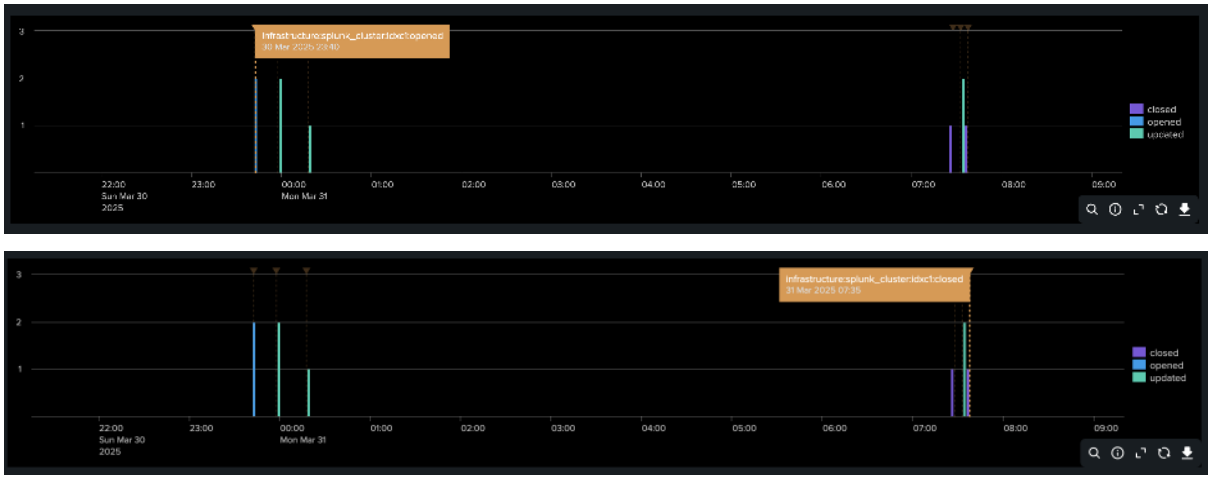


A final incident closure event is created:

```
index=trackme_summary sourcetype=trackme:stateful_alerts tenant_id="cluster-mon"␣
↪object="infrastructure:splunk_cluster:idxc1" alert_status=closed
```



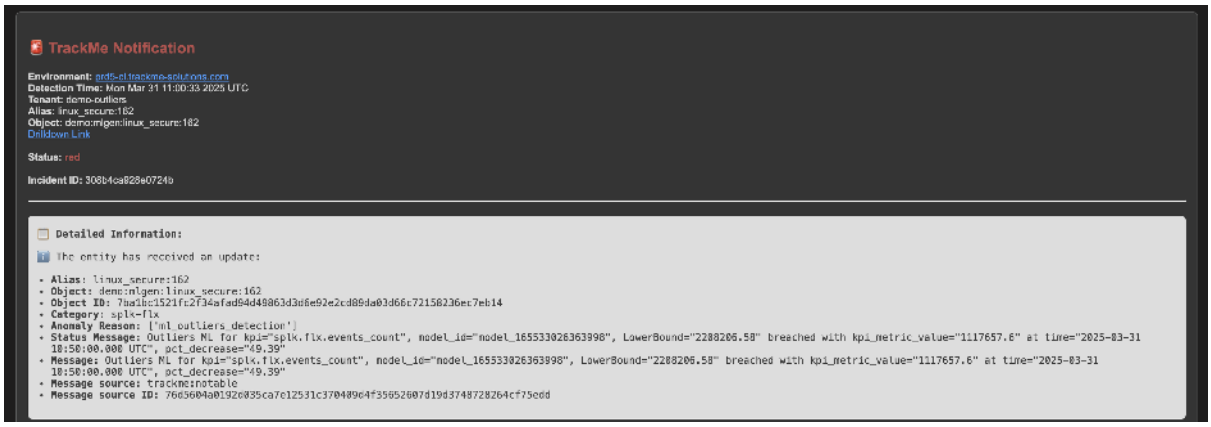
Visualizing the incident life cycle in a Splunk chart with annotations:

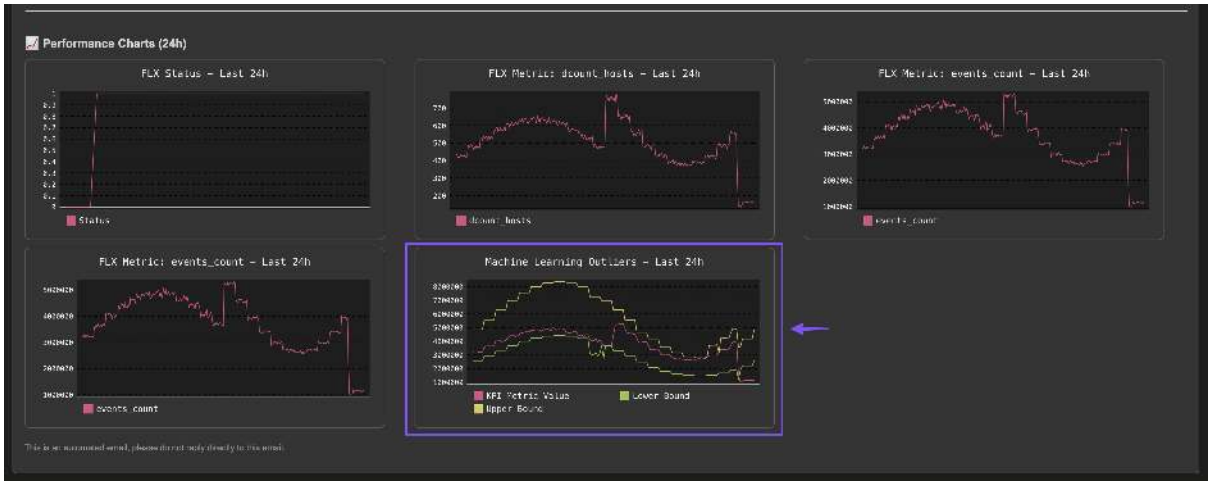


Example of a notification due to Machine Learning Outliers detection:

Below is an example of a notification due to Machine Learning Outliers detection:

Note: TrackMe will automatically add a data sampling related chart in the notification, to help you visualise the issue.





Example of a notification due to a feed data quality issue (data sampling):

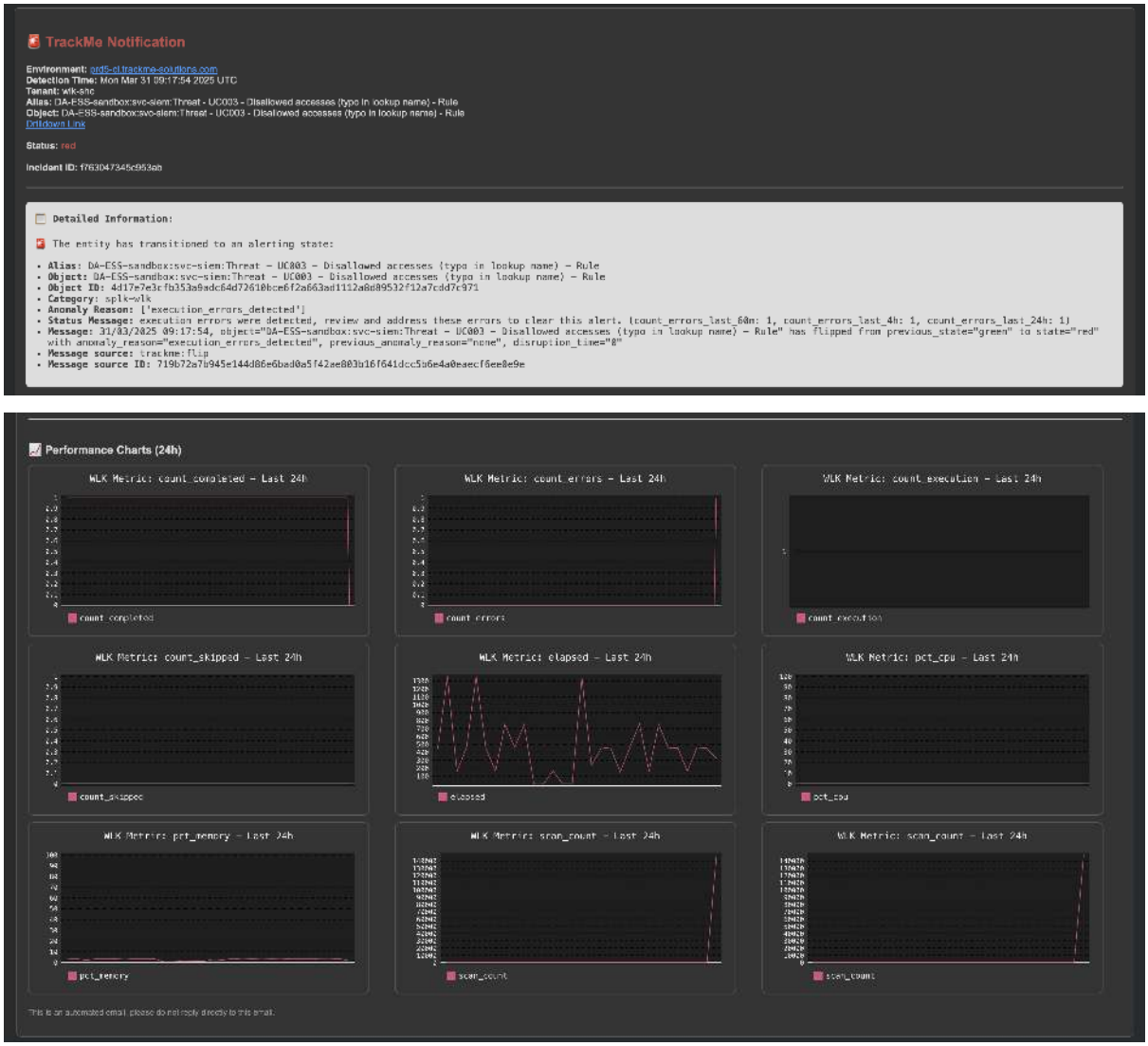
Below is an example of a notification due to a feed data quality issue (data sampling):

Note: TrackMe will automatically add a data sampling related chart in the notification, to help you visualise the issue.

Example of a notification for the Workload component (Splunk scheduled searches and alerts monitoring):

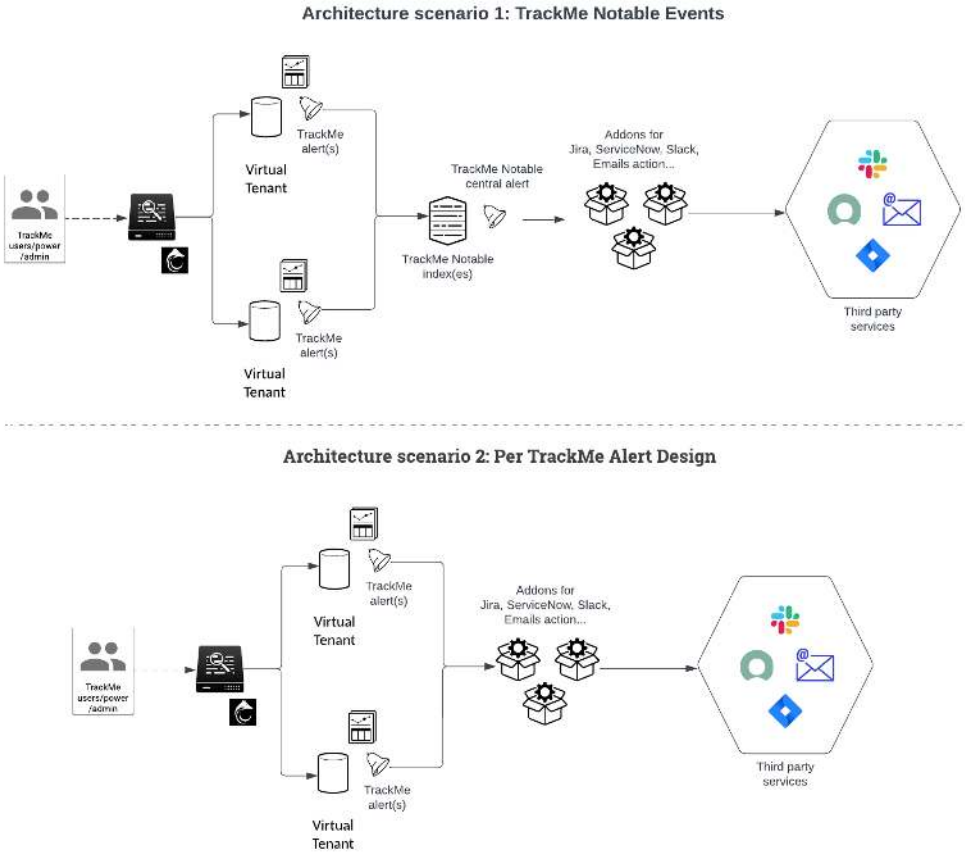
Below is an example of a notification for the Workload component (splk-wlk):

Note: For this component, TrackMe will automatically generate charts per metrics, depending on metrics available in the entity.



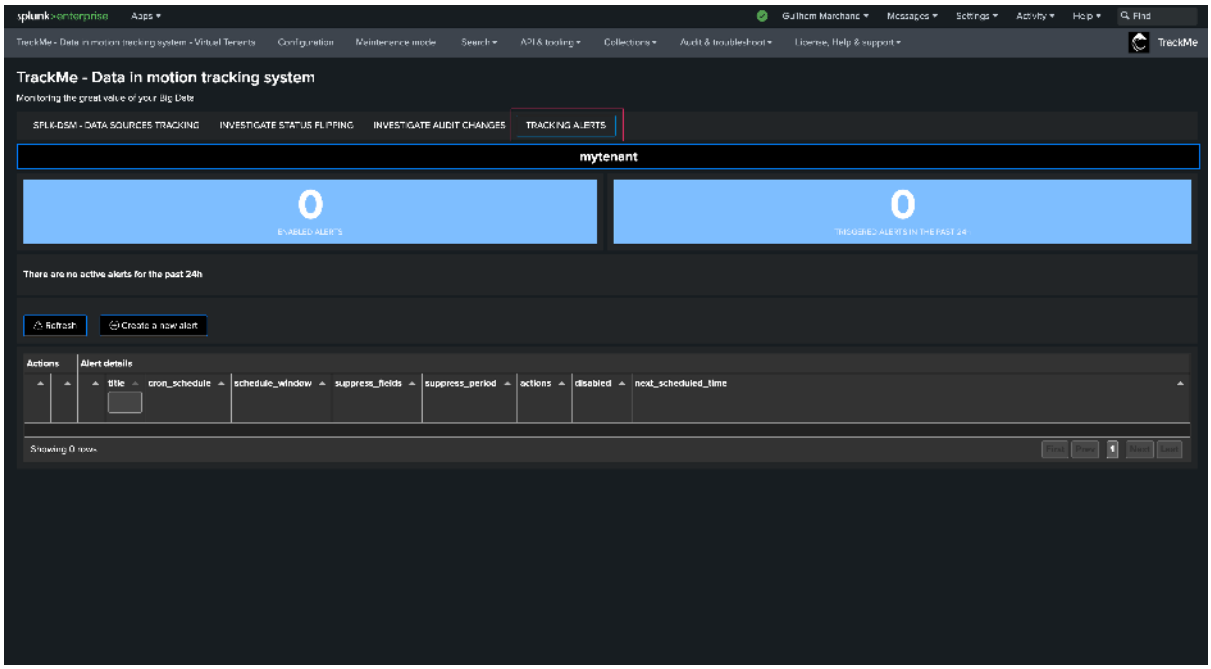
7.11.4 Legacy Alerting with Notables or direct alerts

Legacy Alerting is based on Notables or direct alerts:

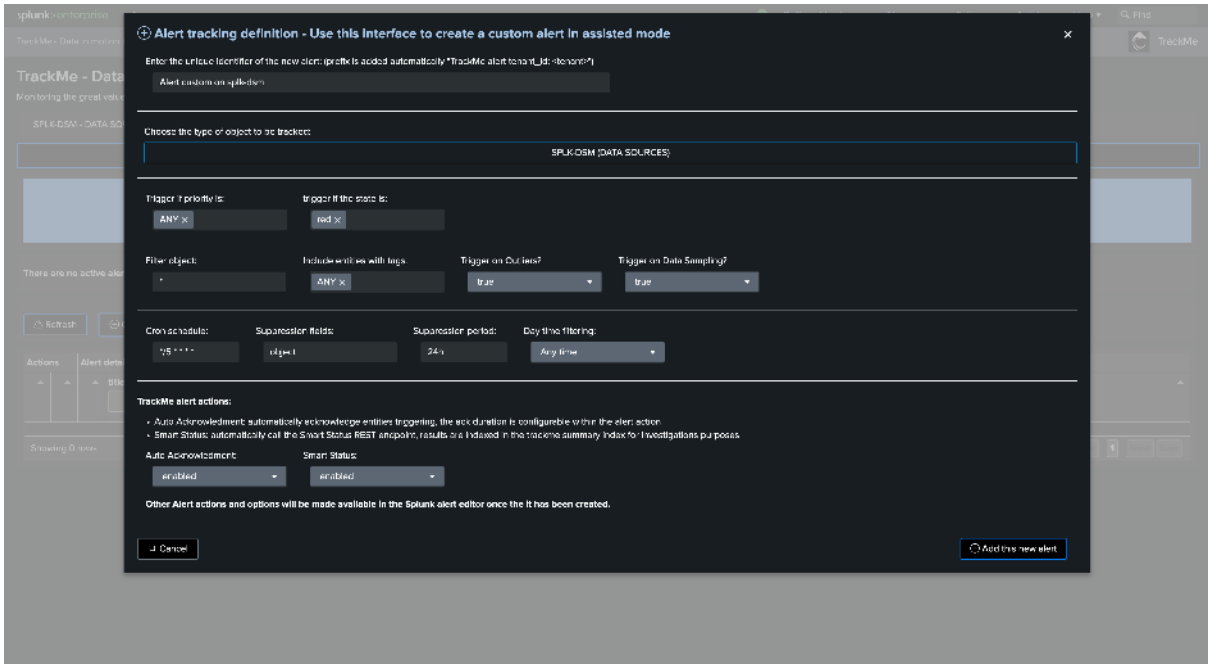


7.11.5 Creating an alert in TrackMe

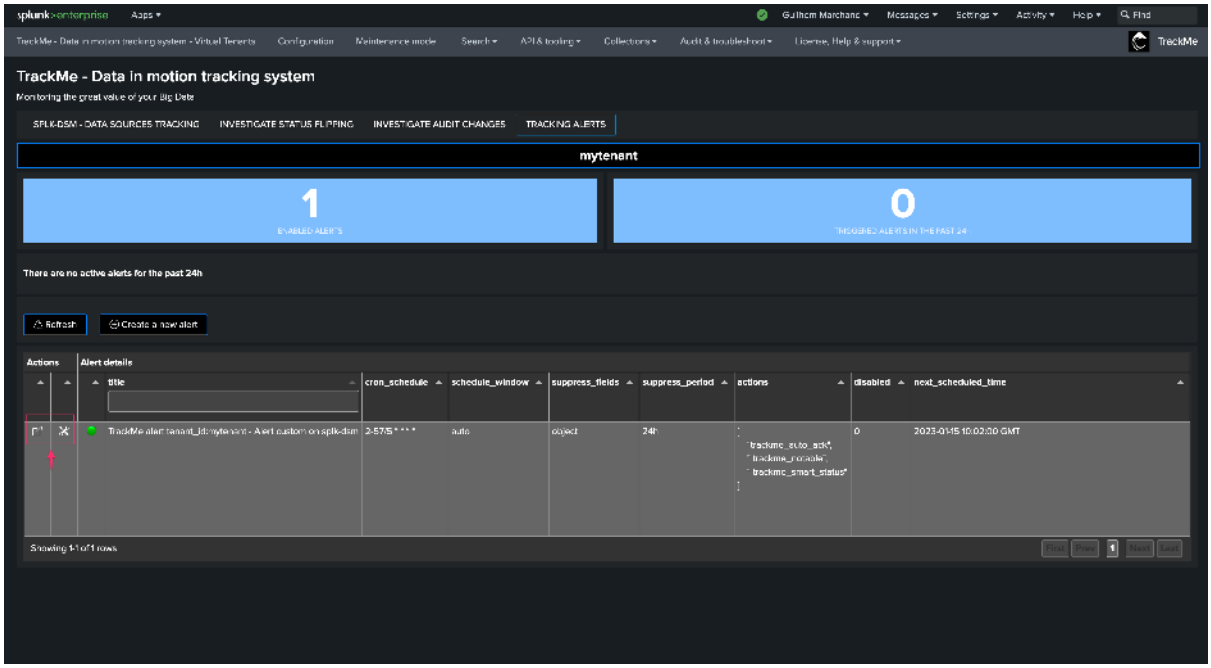
You can create a new alert in the scope of any Virtual Tenant:

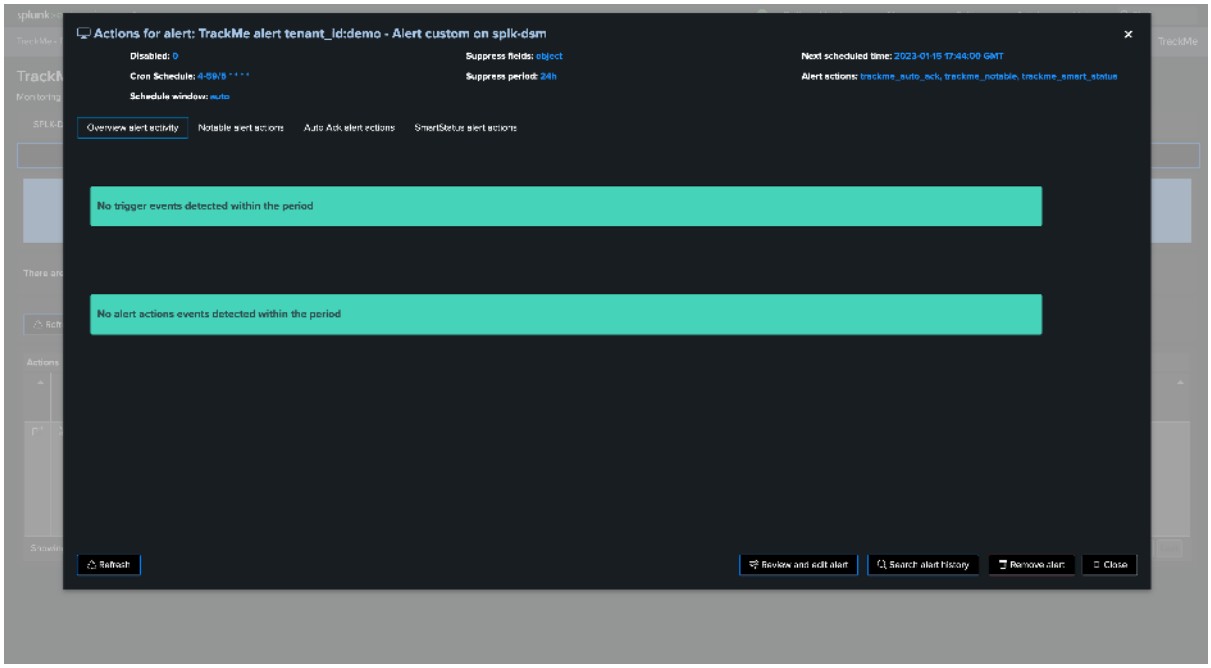


Available option may differ per TrackMe component, for instance with splk-dsm:



Once created, the alert reports in the alert user interface, and can be managed from this screen:

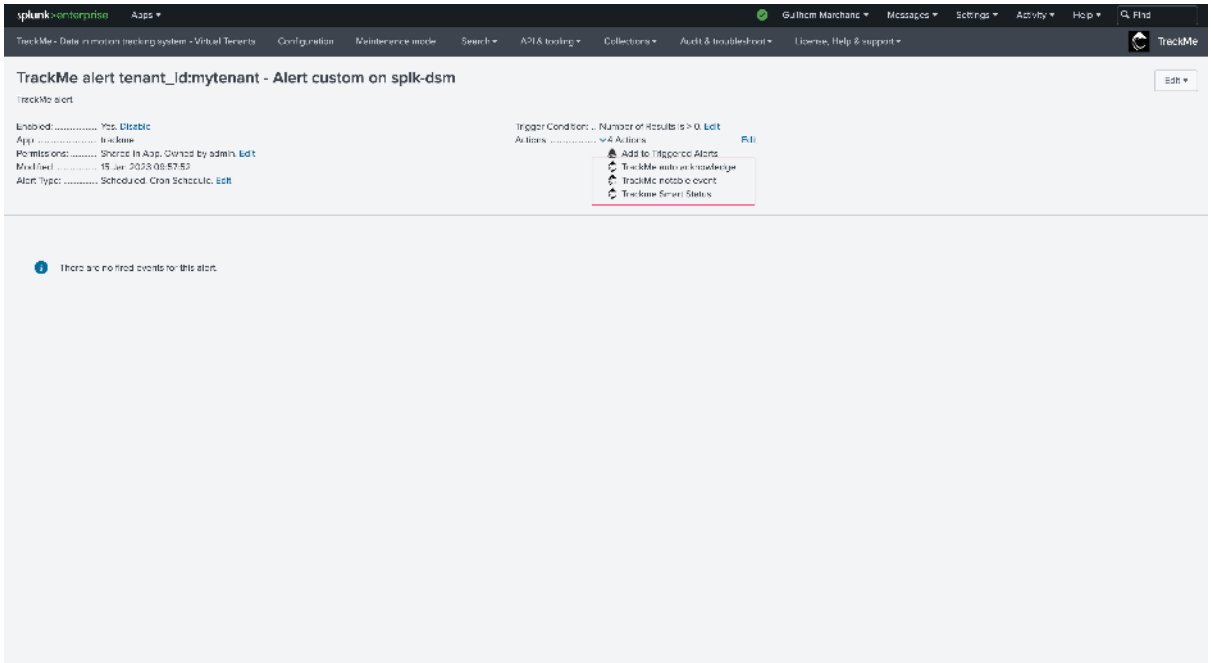


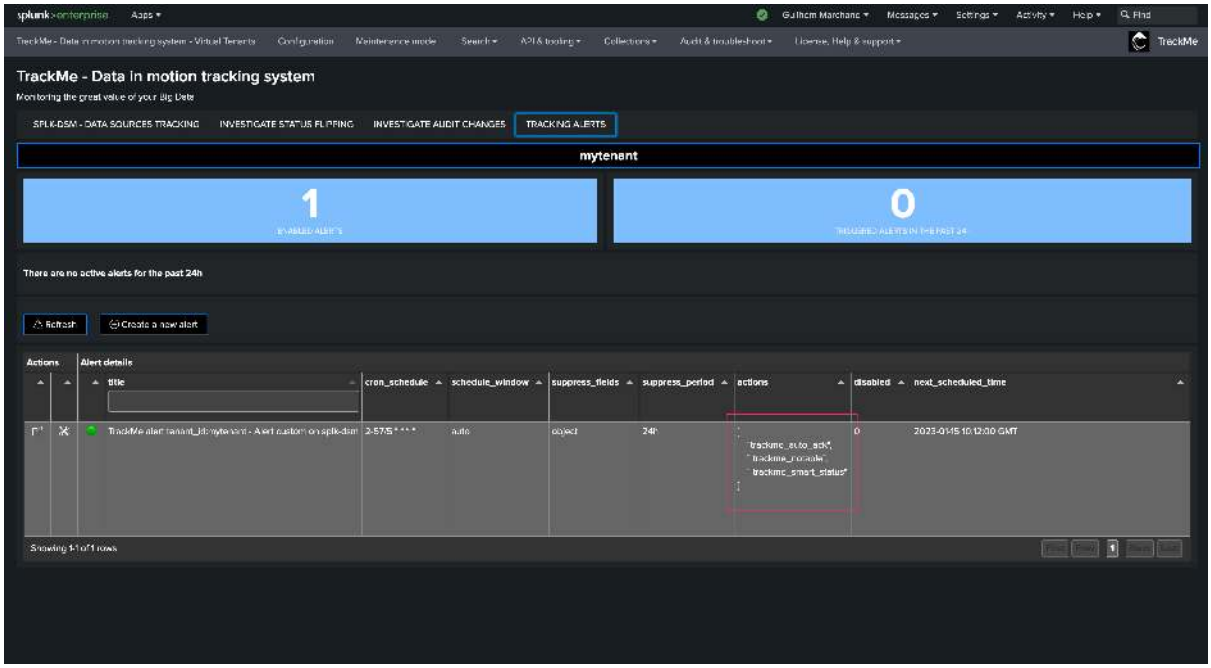


TrackMe modular alert actions

TrackMe has several alert actions that are leveraged when creating alerts:

- **TrackMe Notable event:** This alert action generates a JSON event in the Virtual Tenant notable index summarizing the status of an entity when the alert triggers
- **TrackMe Auto Acknowledgment:** This alert action automatically acknowledges the entity in TrackMe's workflow for a given period of time
- **TrackMe SmartStatus:** This alert performs automated investigations when an alert triggers for one or more entities, to improve and speed up root cause identification and add context to an issue detected by TrackMe

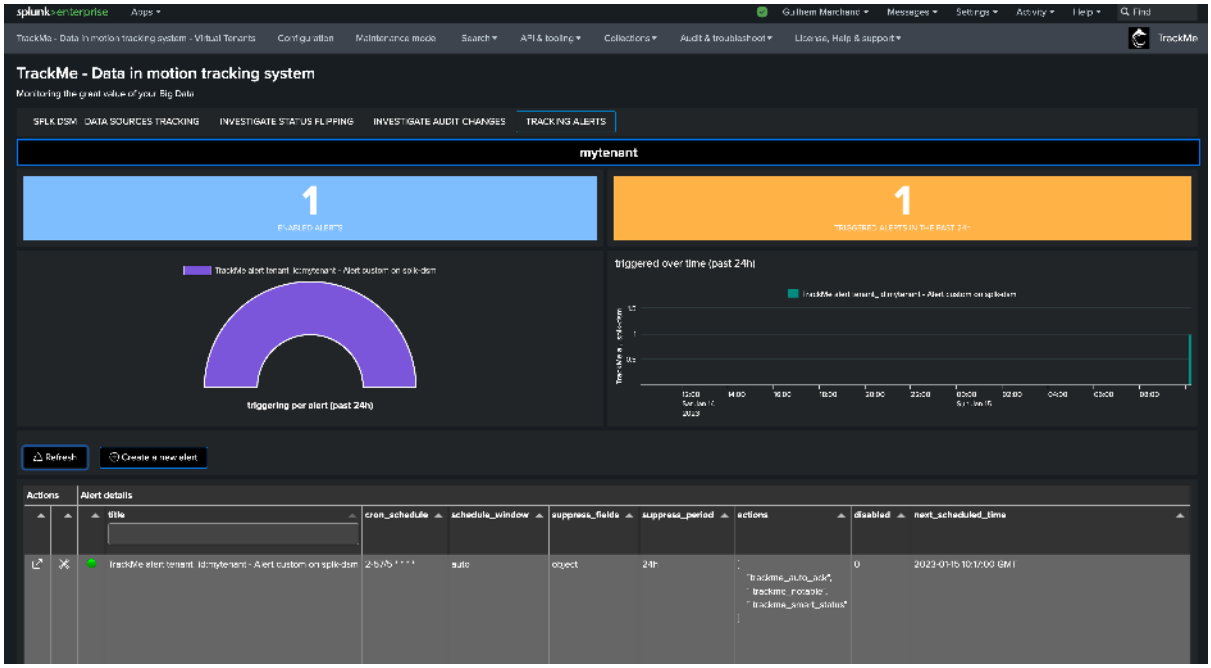


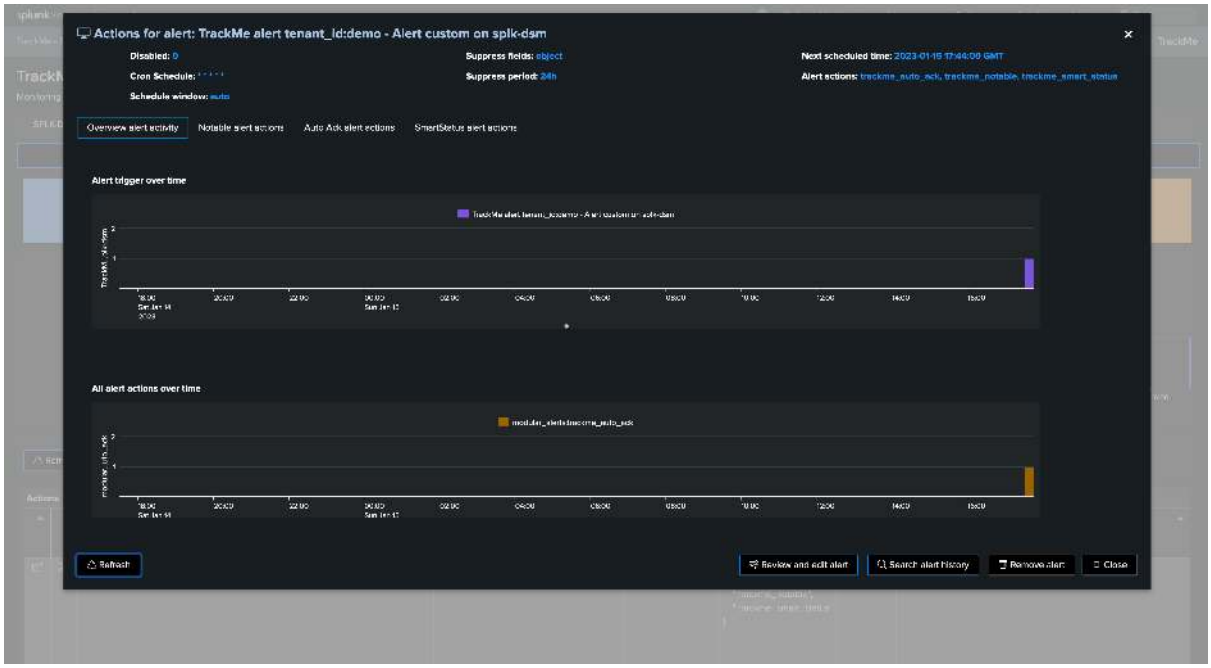


The TrackMe Notable event is a mandatory alert action, others can be disabled during the creation of the alert if you wish to do so.

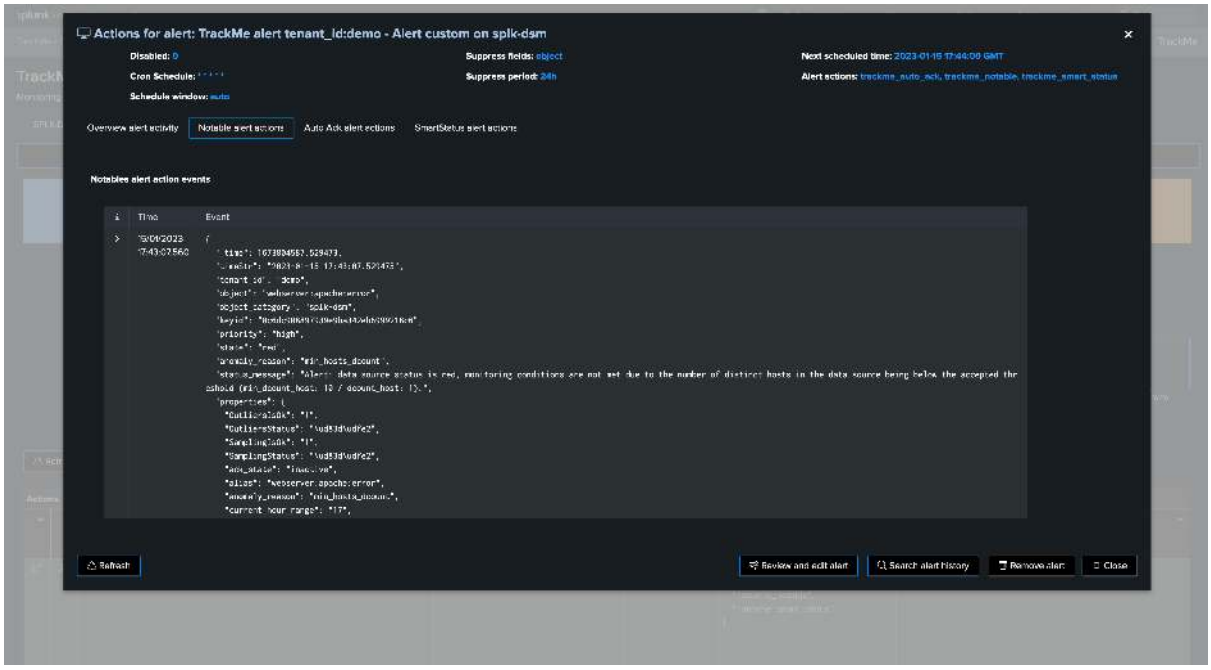
Alerts Triggering

When the alert fires and triggers results, the user interface shows a summary of the triggers and the actions involved:

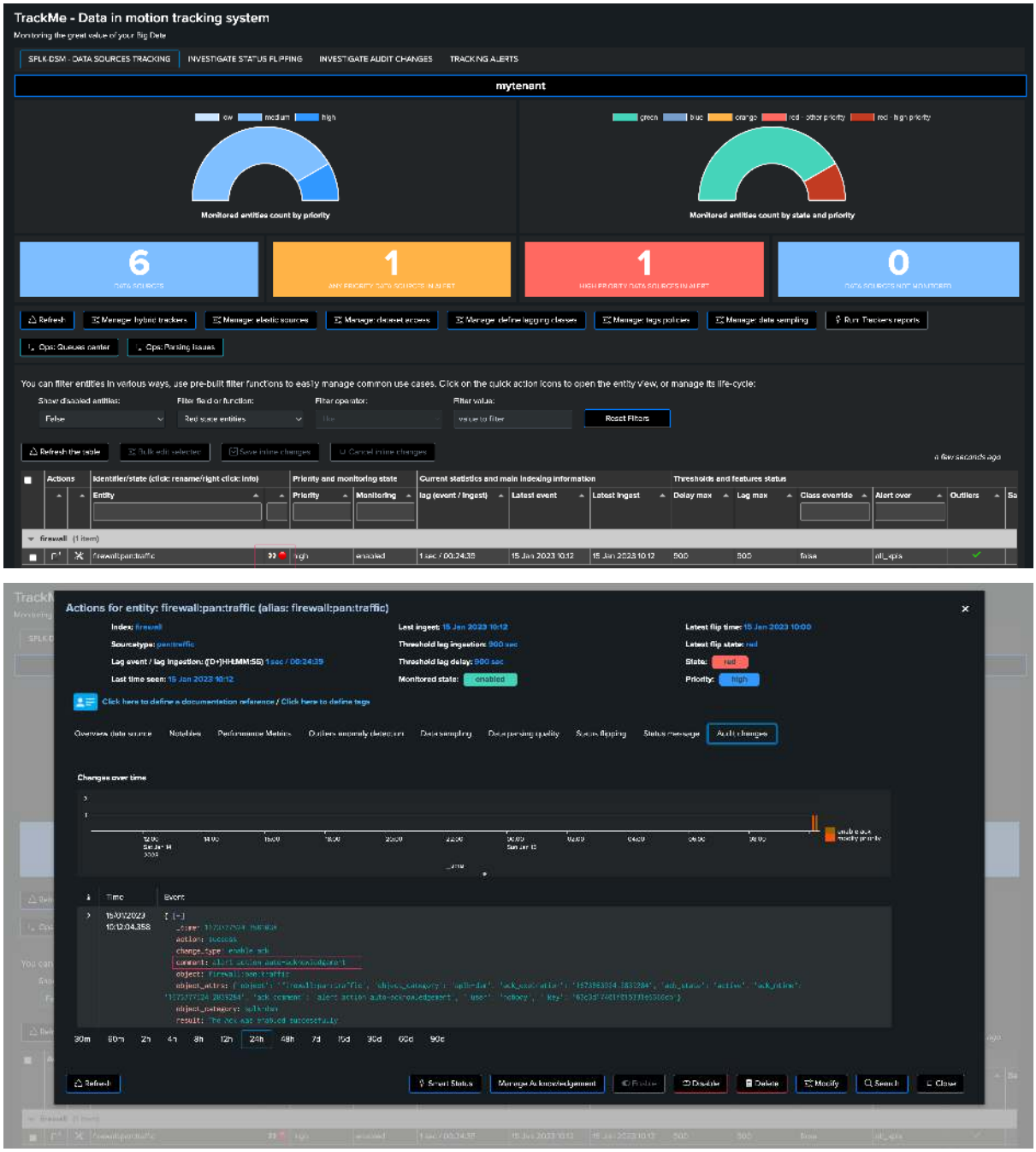




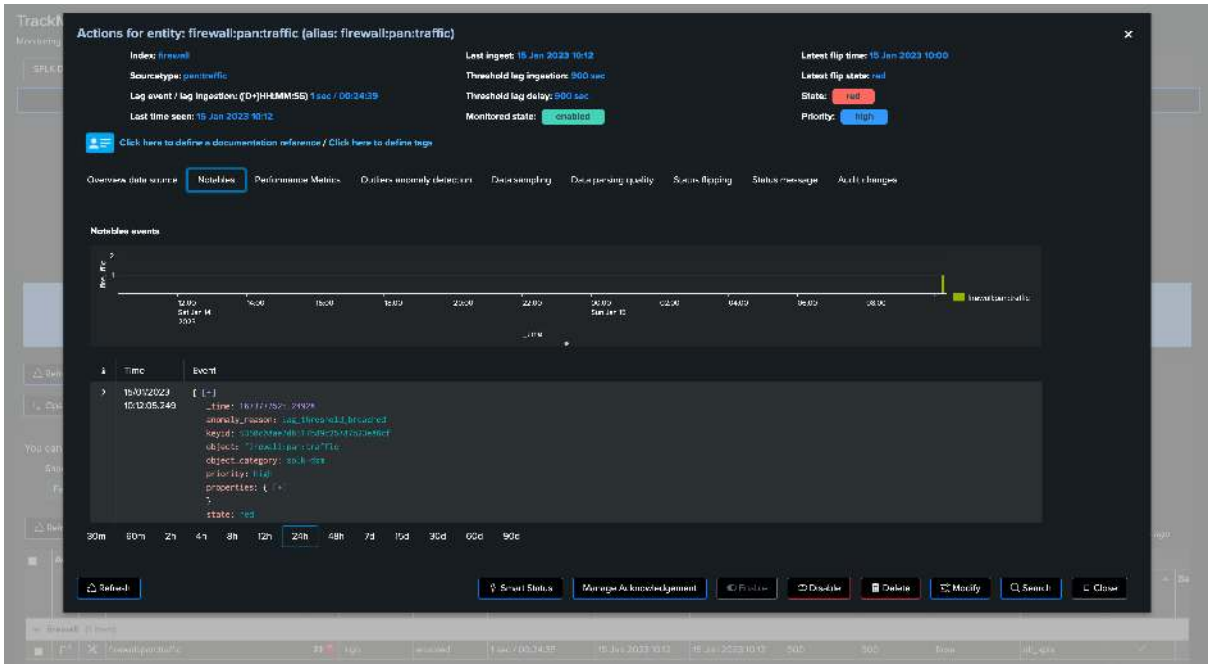
The alert user interface allows to investigate alert actions traces:



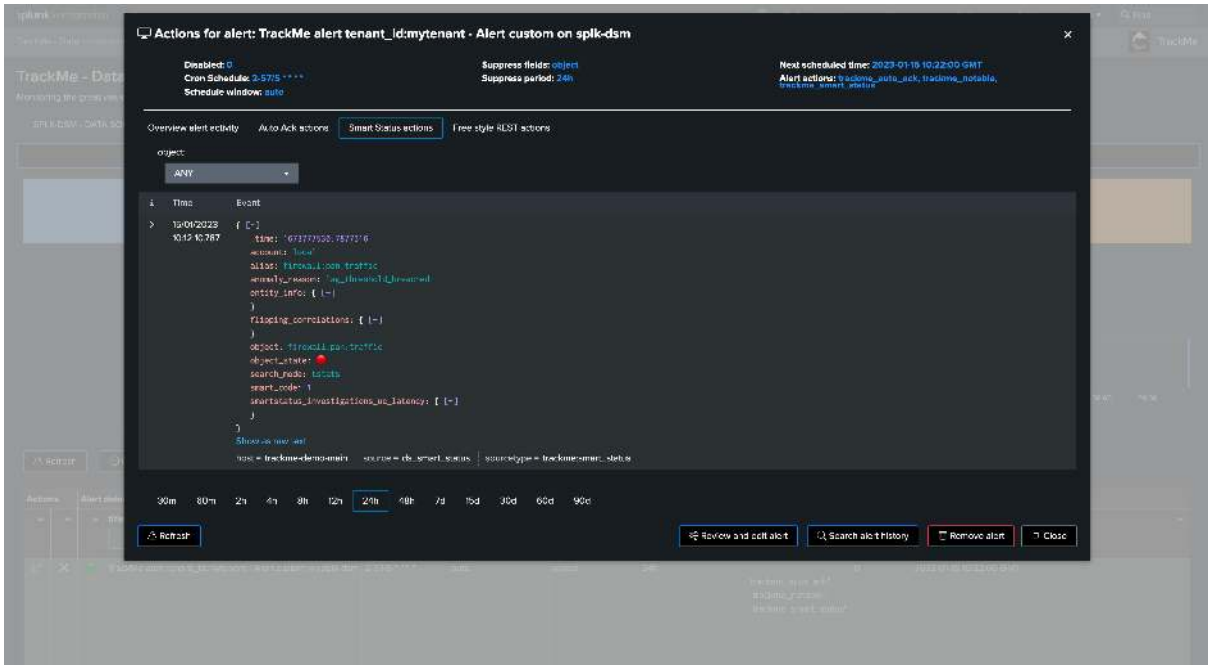
From the entities perspective, we can for instance observe that the entity alert was acknowledged automatically:



A notable event was created:



As well as a TrackMe SmartStatus event:



7.11.6 Third Parties Alert Integration

Introduction to Third Parties Alerting Notification

Any TrackMe alert can be extended to send notifications to a third party, typically:

- Sending emails to a recipient, such as the owner of the entities/service, or TrackMe administrators and analysts
- Opening an incident to your ITSM such as **ServiceNow** or **Atlassian Jira**
- Creating a message in a **Slack** or **Microsoft Teams channel**

Note

Two main alerting architecture design can be implemented:

- **Architecture scenario 1: Notable based design** TrackMe alerts generate notable events, a centralised Splunk alert monitors for TrackMe notables and run alert actions for the third party
- **Architecture scenario 2: Per alert design** TrackMe alerts are configured to run the third party alert action and perform the associated action directly

While the two options are possible, the Architecture scenario 1 based on Notables has multiple advantages:

- Each TrackMe alert handles all entities, ensuring the creation of TrackMe notable events and other features such as the acknowledgment
- Allows centralising the definition of what should lead to the creation of an incident, sending an email or sharing channel message
- Notable events are normalised and provide all necessary information in a consistent fashion
- For instance, you can have a single "TrackMe forward notable" alert which restricts to `high` priority entities and any additional logic of your own, easily in SPL
- This logic can easily be maintained over time, without having to modify multiple alerts and provides a deeper and more consistent control

This documentation covers both approaches in the next sections.

Architecture scenario 1: TrackMe Notable Events

Introduction: Anatomy of TrackMe Notable Events

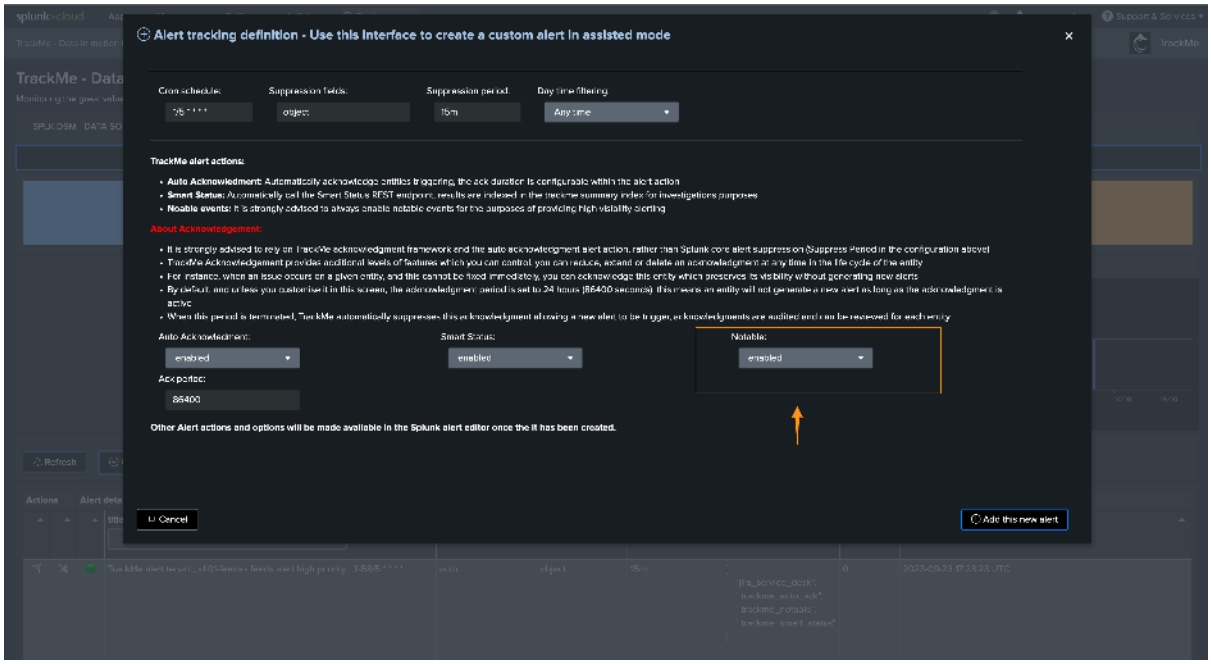
Hint

Forwarding TrackMe Notable to Splunk SOAR

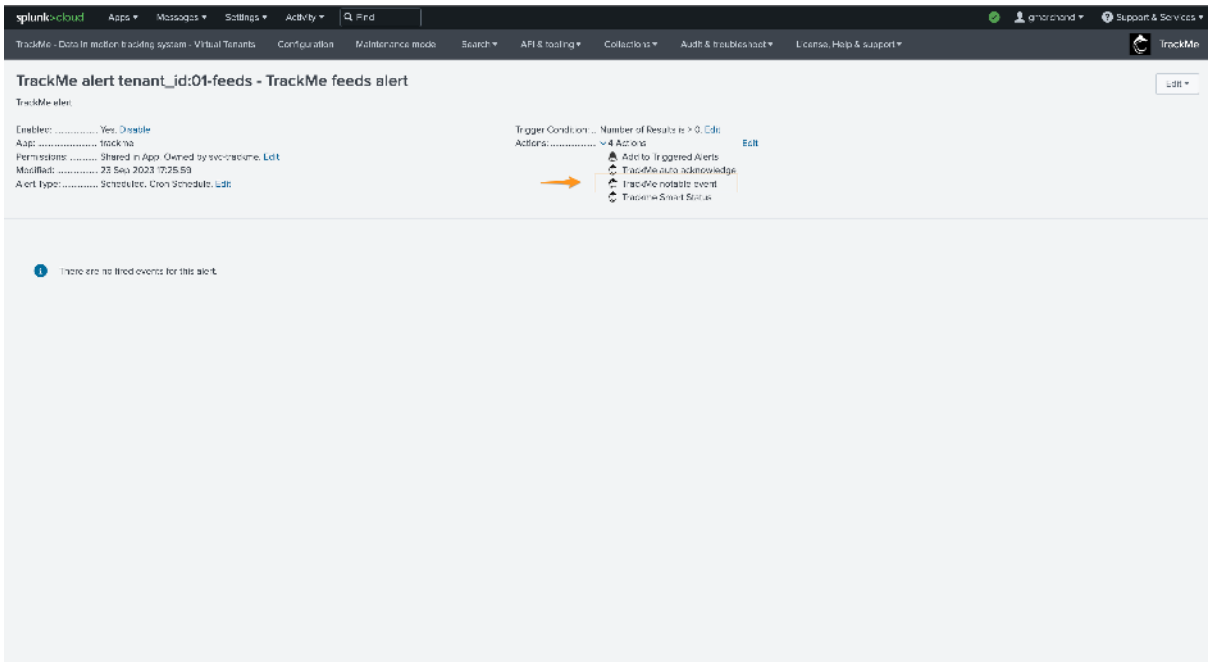
- If you are looking to forward TrackMe Notable Events to Splunk SOAR, consult the following: :*Forwarding TrackMe Notable Events to SOAR*

TrackMe Notable Events creation

TrackMe notable events are generated when a TrackMe alert triggers against one or more entities, relying on the built in `TrackMe Notable` modular alert action.



This alert action configuration is also visible when editing a TrackMe alert:



Note

Notable events are ONLY created when a TrackMe alert is configured and triggers; in this scenario, all eligible entities should therefore be covered by an alert per tenant and component.

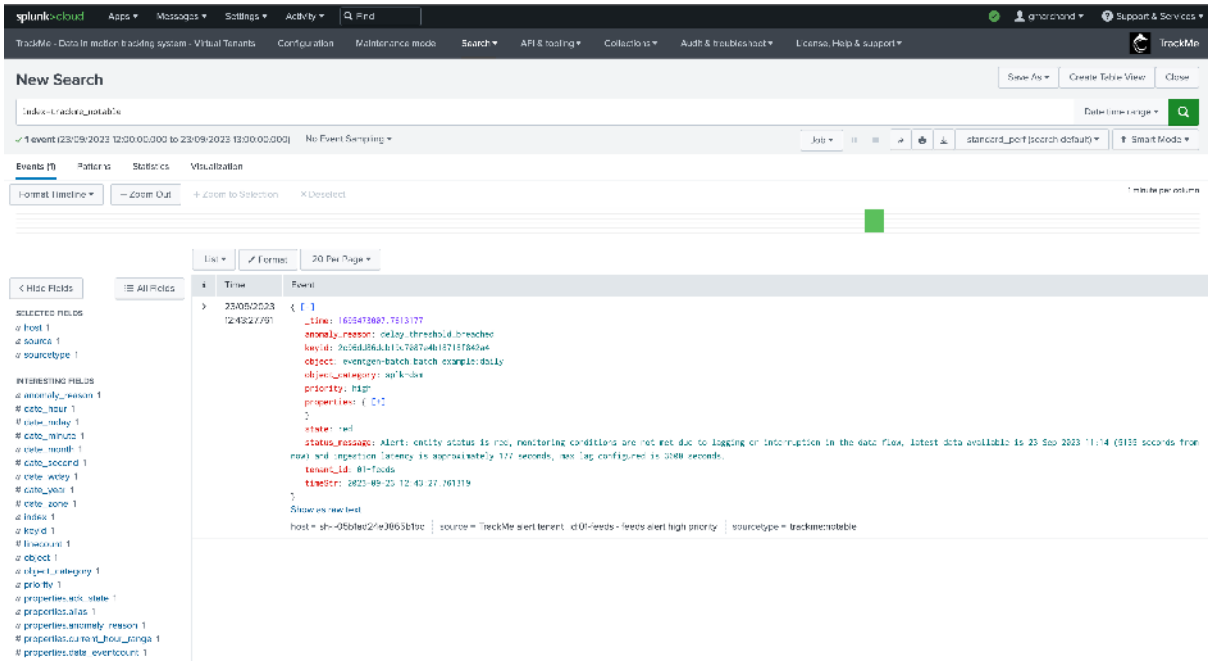
TrackMe Notable Events structure

TrackMe Notable events are `JSON` formatted events, these contain the following key information:

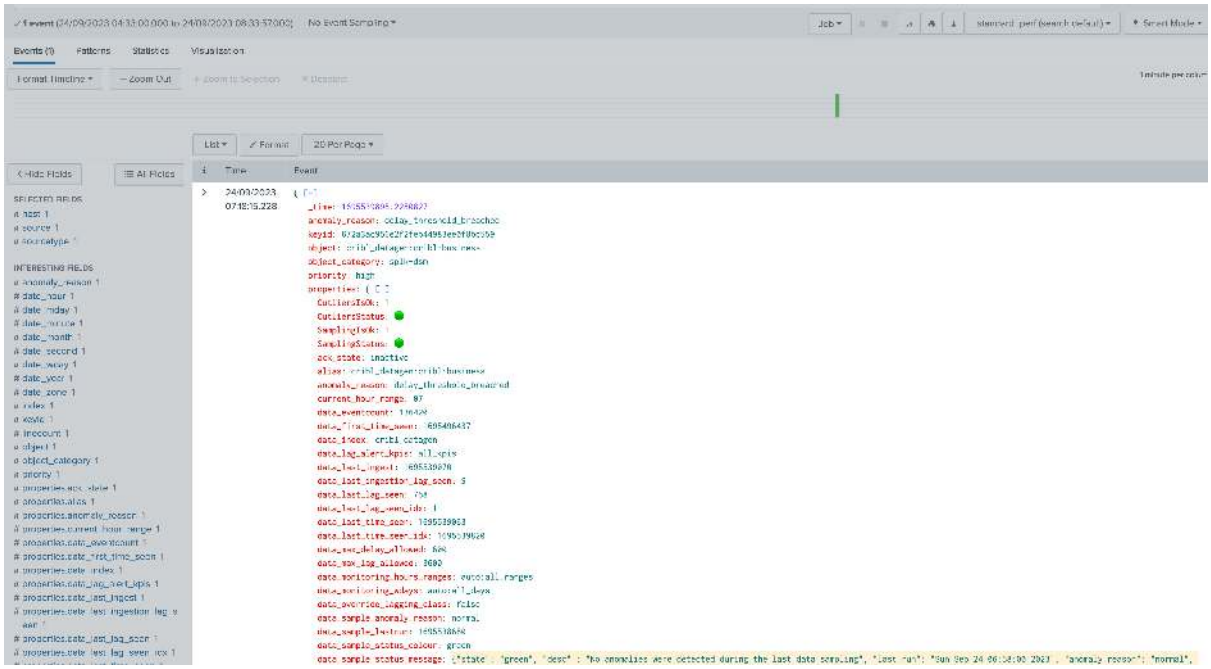
Field	Description
tenant_id	The tenant identifier this Notable event is related to.
object	The object name for this entity, this basically is the name of entities in TrackMe
object_categ	A TrackMe identifier for each component, this describes which type of TrackMe component this notable is related to.
keyid	The unique identifier for this object in the tenant / component central KVstore collection
priority	The priority definition for this object (low / medium / high / critical)
state	The status of the object (blue / green / orange / red)
anomaly_	A normalised field describing the reason behind the status of a given entity
status_messa	A human readable detailed description of the reason why TrackMe generated an alert for this entity
timeStr	The time of the event in human readable format
drilldown_linl	A link to the TrackMe Home user interface which filters on the tenant, component and entity, and automatically opens the entity overview screen (from TrackMe 2.0.88)
event_id	The unique identifier for this notable event, as the sha-256 hash of the notable event content (from TrackMe 2.1.0)
properties	The complete TrackMe object record in a nested JSON structure, as it was when the alert triggered

Therefore, TrackMe Notable events contain all necessary information in a consistent and normalized manner, to interact with any third party.

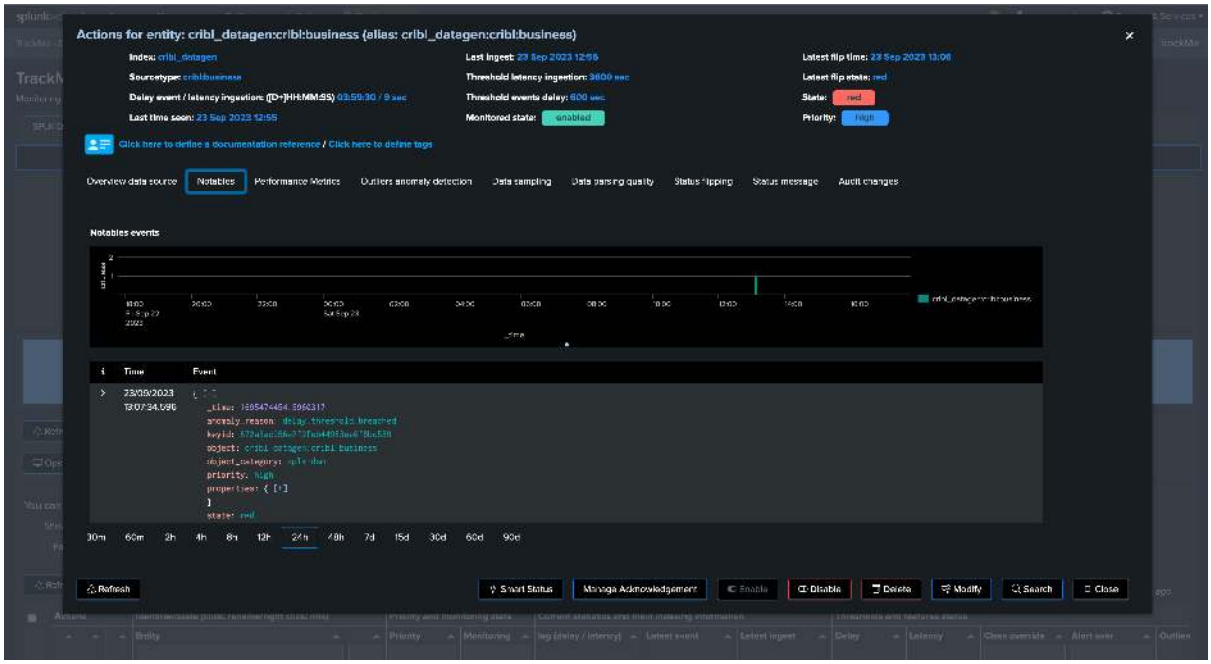
Example of a TrackMe Notable event:



The properties field contains the entire entity record as nested JSON; it is also extracted automatically so you can use fields as needed:



TrackMe Notable events are easy to access and review in TrackMe's user interface:

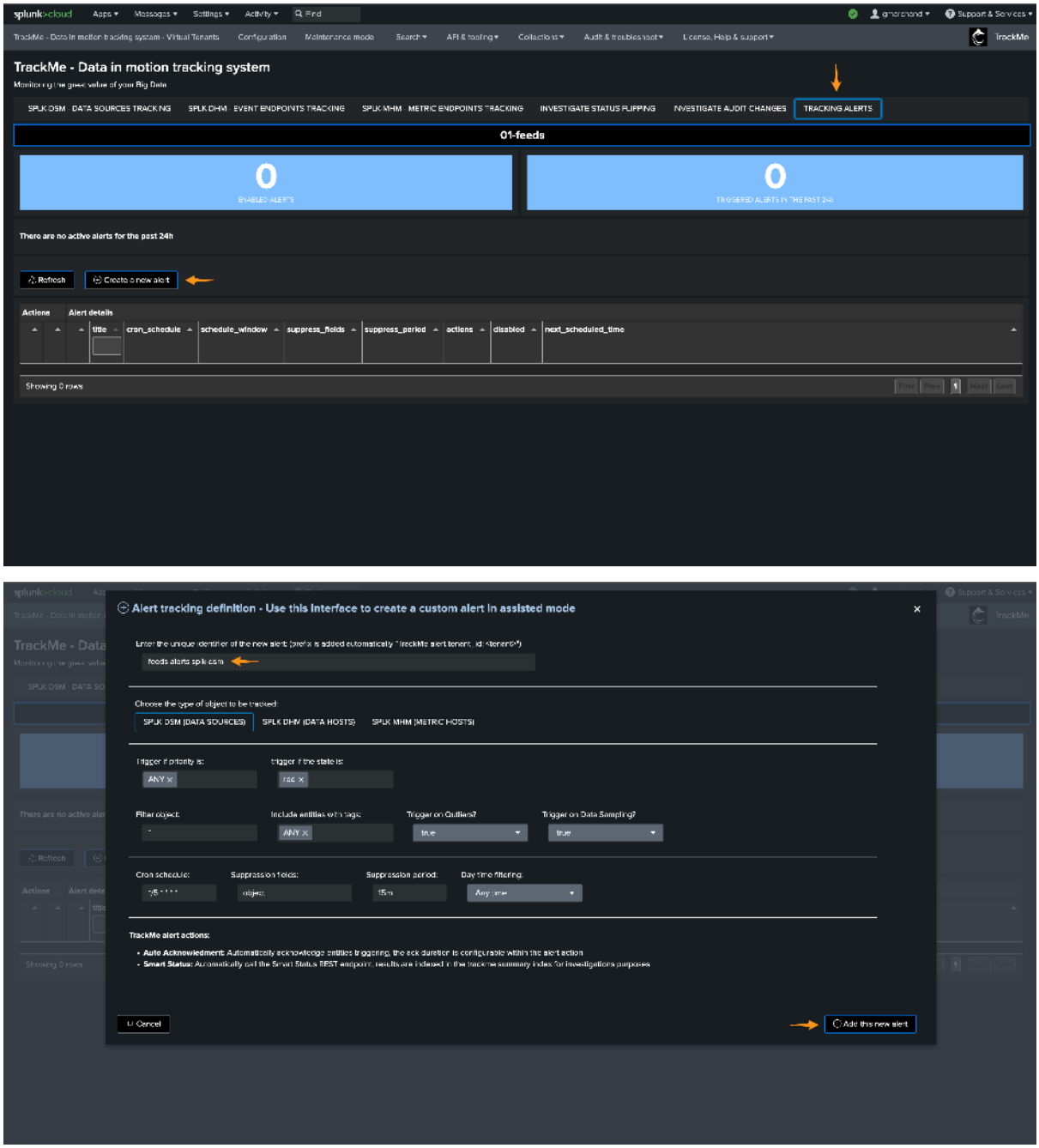


Step 1: Create TrackMe alerts for tenants and components in used

The first step is to create a TrackMe alert for every tenant and component that is in use.

It is not necessary at this stage to filter on concepts like the priority because filtering will be made while monitoring the TrackMe notable events.

For each tenant, access the **TRACKING ALERTS** tab, and use the wizard to create an alert, you can leave all options to their default:



Step 2: Create a custom alert monitoring TrackMe notable

The second step is to create a custom Splunk alert which searches for TrackMe notables and enables third-party interactions, for instance:

As a basis:

```
index=trackme_notable tenant_id=* priority=*
| table tenant_id, keyid, event_id, object, object_category, state, priority, anomaly_
↪reason, status_message, drilldown_link
```

For instance, a typical use case would be to create incidents only for high priority entities in alert:

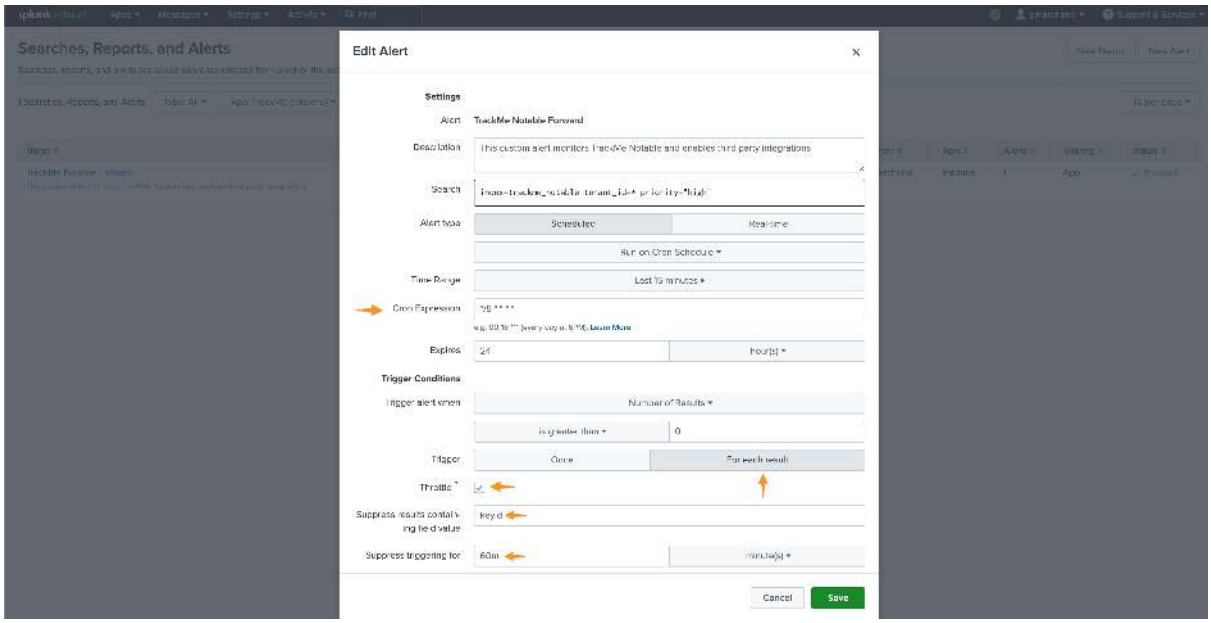
```
index=trackme_notable tenant_id=* priority="high"
| table tenant_id, keyid, event_id, object, object_category, state, priority, anomaly_
↪reason, status_message, drilldown_link
```

Note

In TrackMe, you can define different notable indexes per tenant, if you did so you will want to adapt your search.

Create the Splunk alert:

- Cron every 5 minutes: */5 * * * *
- Earliest time: -15m
- Trigger for each result
- Throttle against the keyid for 60m



Then enable the third party alert action, for example we will enable creating incident in JIRA, the following shows an example of content for the alert which calls tokens replacements:

```
TrackMe alert for a Splunk high priority entity:

- tenant: $result.tenant_id$
- object: $result.object$
- object_category: $result.object_category$
- state: $result.state$
- priority: $result.priority$
- anomaly_reason: $result.anomaly_reason$
- status_message: $result.status_message$
- drilldown_link: $result.drilldown_link$
```

Hint

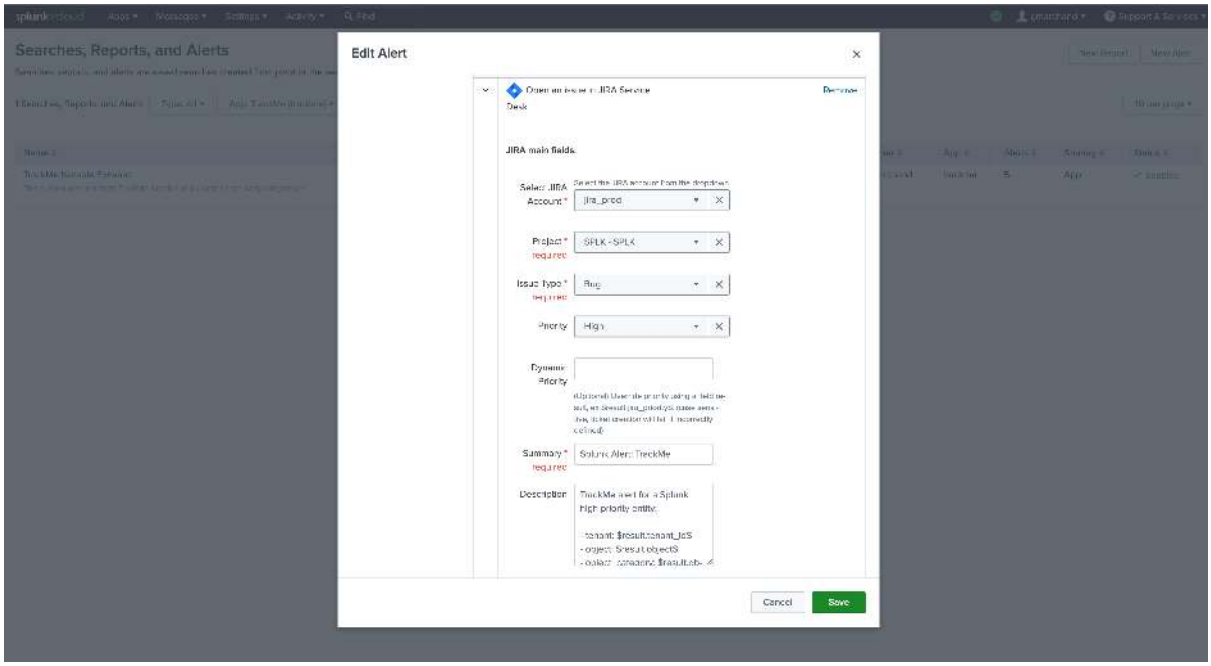
Re-assigning to a service account

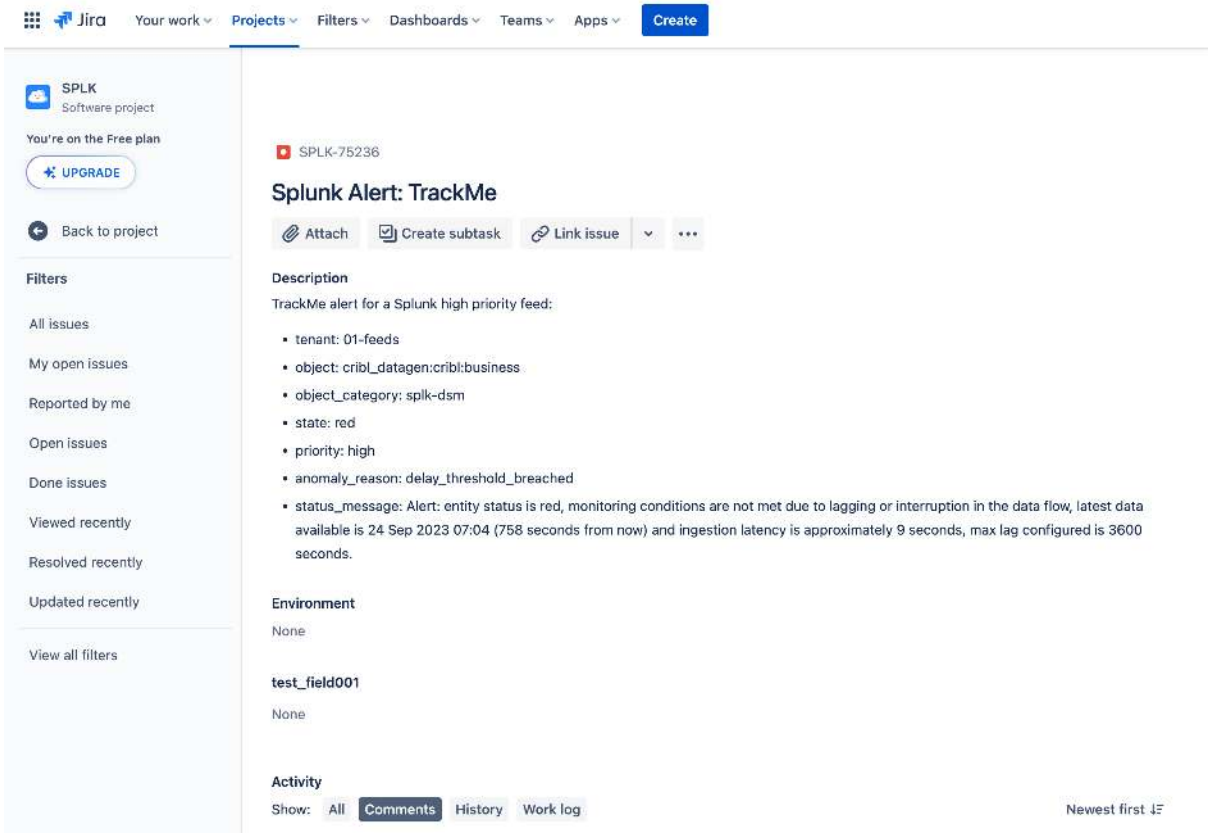
- As a good practice, you should re-assign the created Splunk alert to the TrackMe service account user, or nobody
- You should avoid running alerts on behalf of normal users

Adding third parties actions:

Atlasian Jira example:

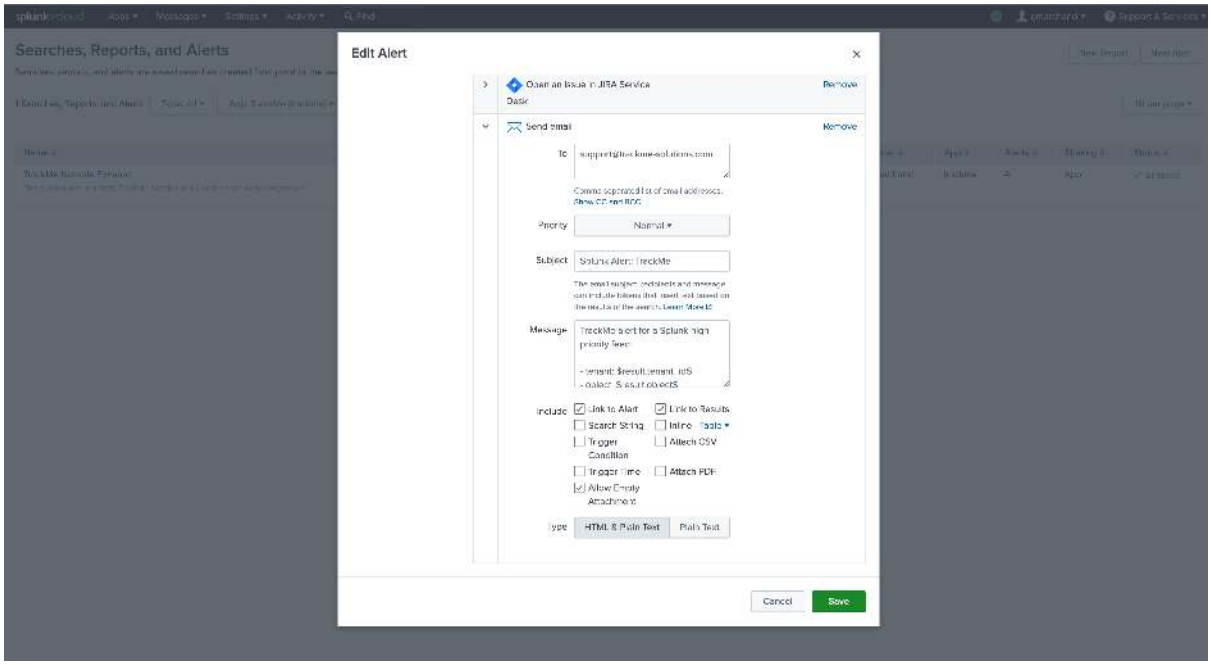
Note: this integration relies on the following Add-on for Jira: https://splunkbase.splunk.com/app/4958

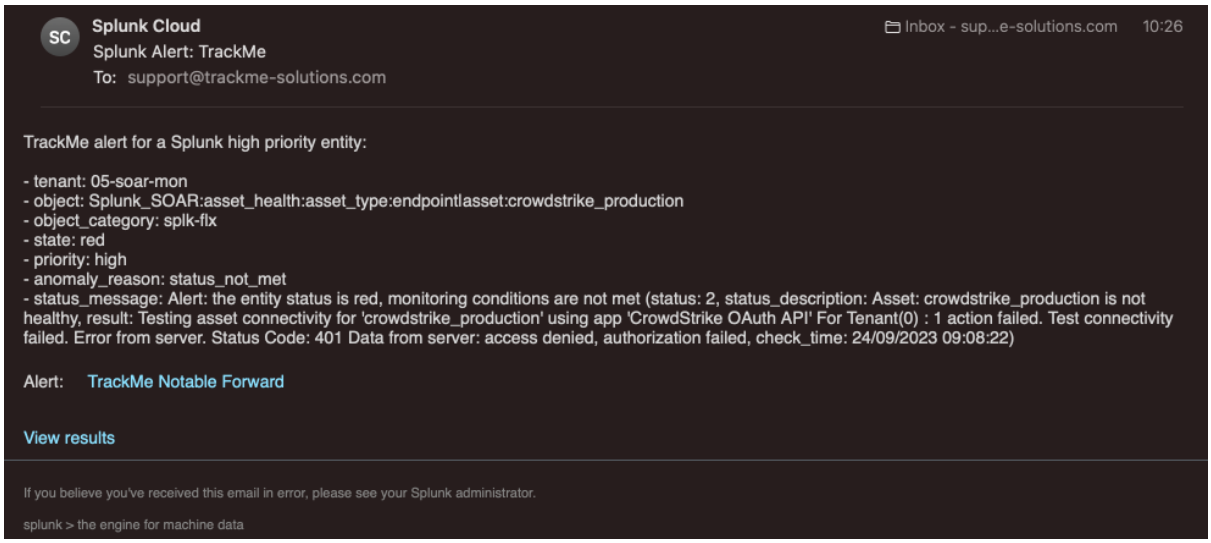




Email example:

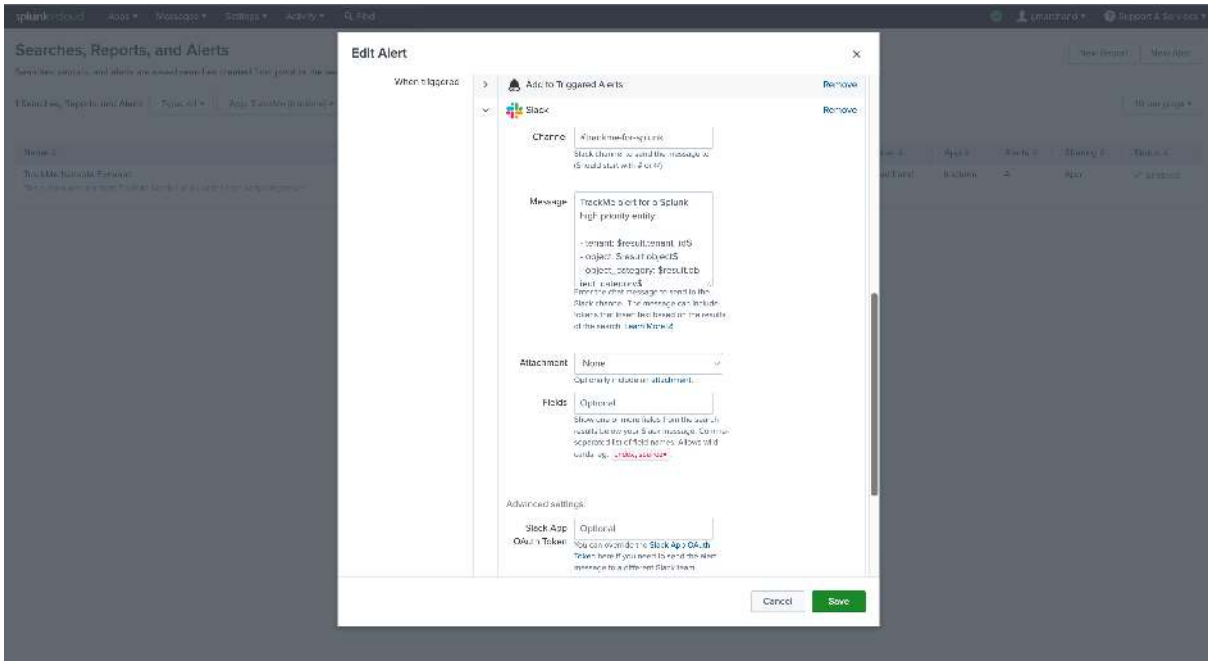
Note: Sending emails is a builtin alert action in Splunk





Slack example:

Note: this integration relies on the following Add-on for Slack: https://apps.splunk.com/app/2878



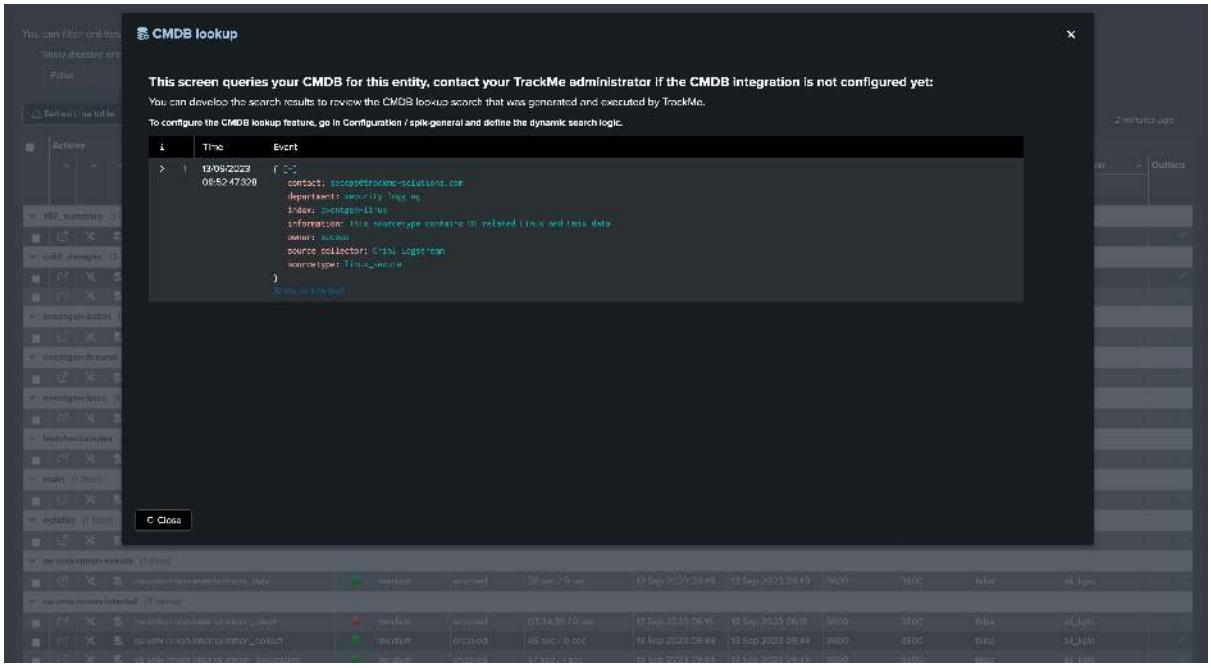
Optional: Notable events enrichment

When using Notable events as the central means for third party alerting, further logic can easily be implemented such as custom normalisation and enrichment.

With enrichment for instance, you may want to leverage your CMDB knowledge to add context information such as the owner or contact address email for the notification, there are plenty of options and designs possible.

TrackMe allows to lookup easily in your CMDB to provide context information handy in the user interface, you can leverage the same content from the central Notable alert, for more information about the CMDB Lookup integrator see: *CMDB Lookup Integration*

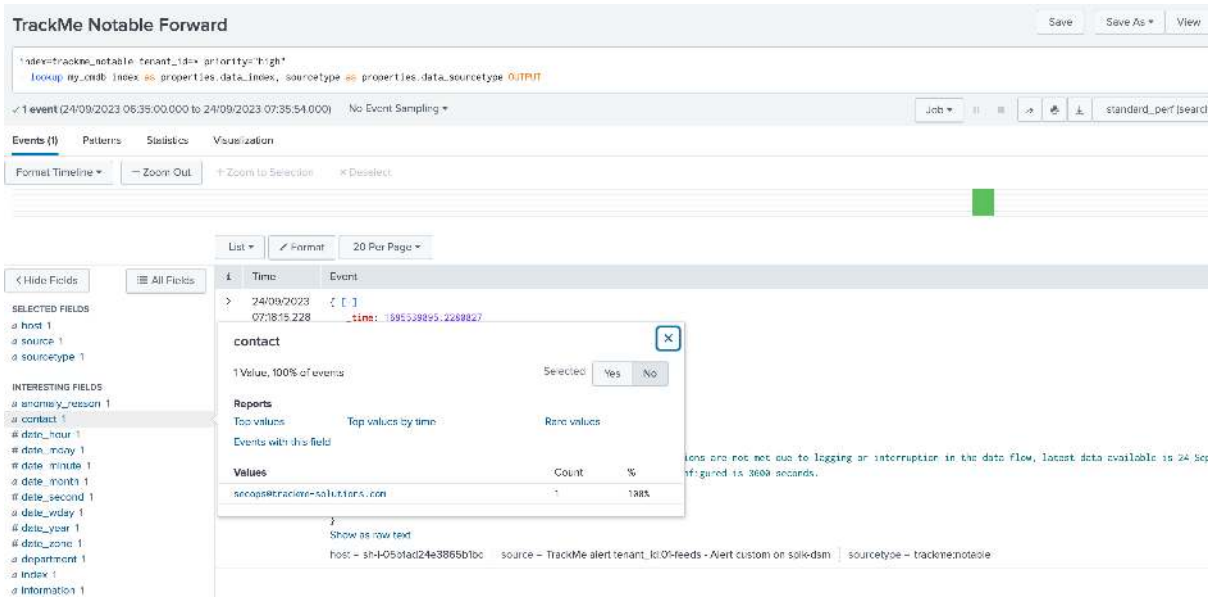
CMDB integration in the user interface:



For instance, we could update our Notable event alert:

```
index=trackme_notable tenant_id=* priority="high"
| lookup my_cmdb index as properties.data_index, sourcetype as properties.data_
↪sourcetype OUTPUT
| table tenant_id, keyid, event_id, object, object_category, state, priority, anomaly_
↪reason, status_message, drilldown_link
```

Enrichment information are automatically available:



Any of these fields can be used equally in your alert configuration using tokens, for instance we could update the recipient email address and add some logic to handle unknown entities:

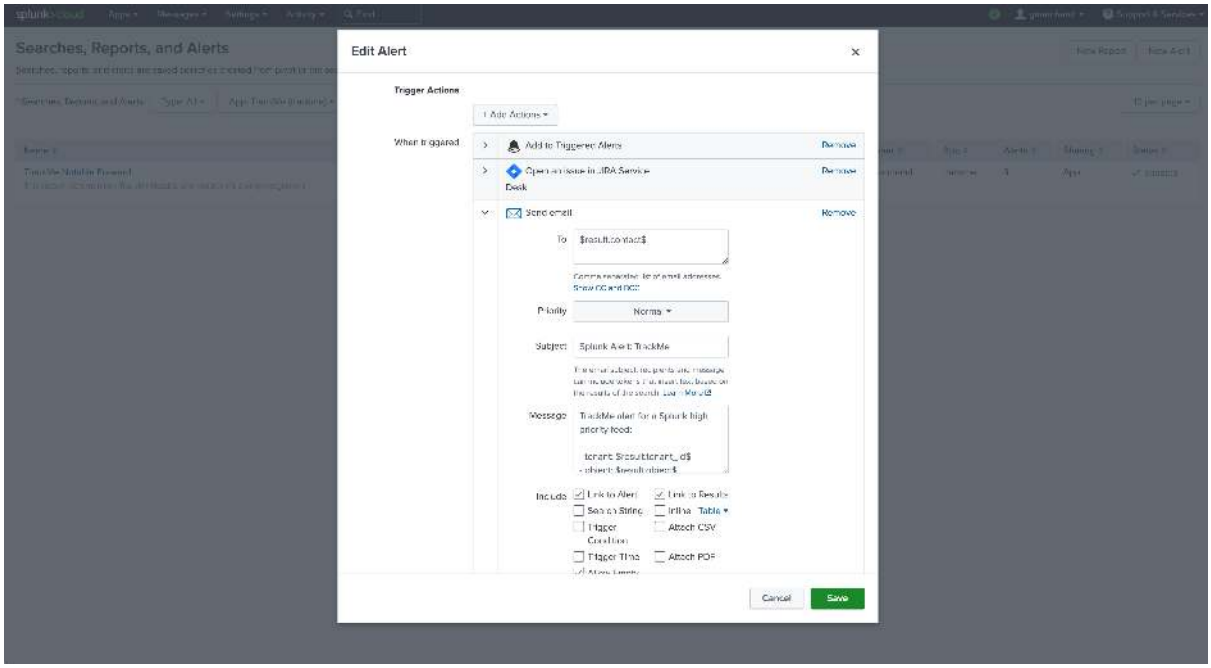
```
index=trackme_notable tenant_id=* priority="high"
| lookup my_cmdb index as properties.data_index, sourcetype as properties.data_
↪sourcetype OUTPUT
| eval contact=if(isnull(contact), "support@trackme-solutions.com", contact)
```

(continues on next page)

(continued from previous page)

```
| table tenant_id, keyid, event_id, object, object_category, state, priority, anomaly_
↪reason, status_message, drilldown_link, contact
```

And recycle this token instead:



Architecture scenario 2: Per TrackMe Alert Design

Another design choice is to configure third party alert actions on a per TrackMe alert instead of relying on the Notable events.

Note

This integration can also be considered as a valid choice, eventually easier and more logical, however it does not present some of the advantages in comparison with a TrackMe Notable based design:

- Each alert needs to be configured individually, which can be more challenging to maintain over time
- Entities that are not covered by any TrackMe alert will not lead to TrackMe alert actions being executed (Notable, Ack, SmartStatus), for instance if you restrict alerts using entities priority or tags and some entities are out of the scope
- Normalisation, enrichment or the introduction of custom SPL logic is decentralised opposed to the central alert based on Notables
- Finally, you may need more alerts to be created (for instance to cover entities by priority and have alert actions executed for all entities)

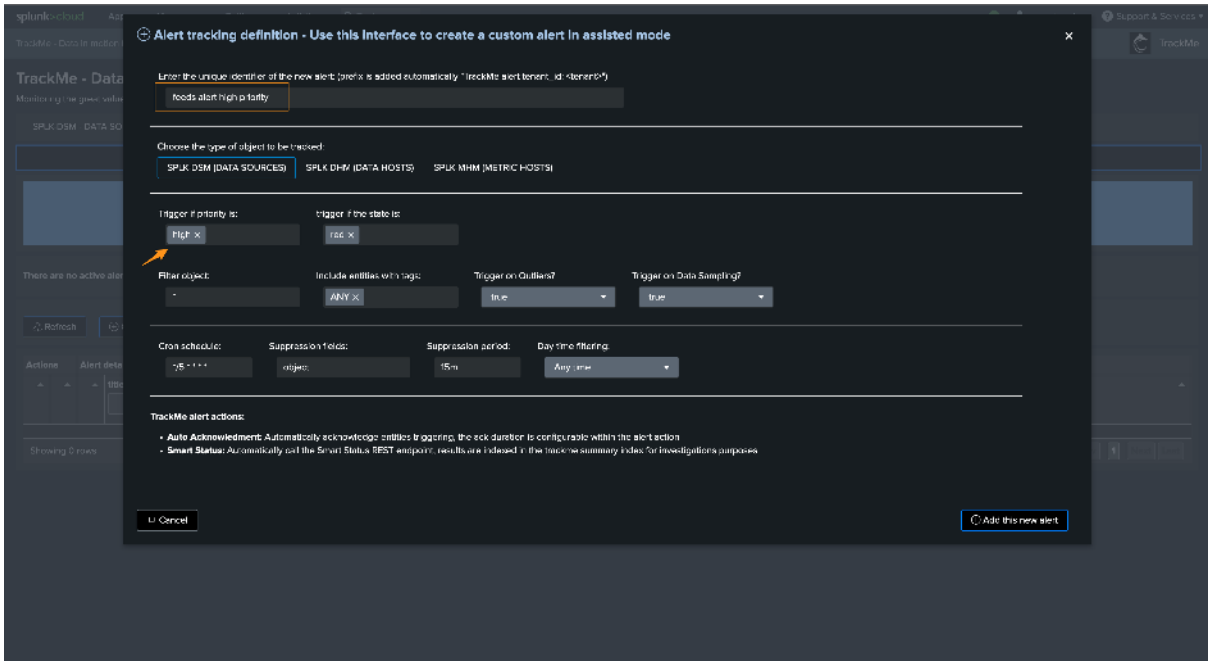
For the purposes of this documentation, the following scenario will be implemented:

- A first alert is created which targets `high` priority entities, when triggering a notification is sent using a Splunk alert action to open an incident in our ITSM tool (we will use Jira but this is applicable to any Splunk supported third party)
- A second alert is created which targets `medium` priority entities, this time an email will be sent when the alert triggers

- Entities set as `low` are not subject to a third party integration, however an alert should also be active for notable events and other TrackMe alert actions to be triggered

Step 1: Create a TrackMe alert for high priority entities

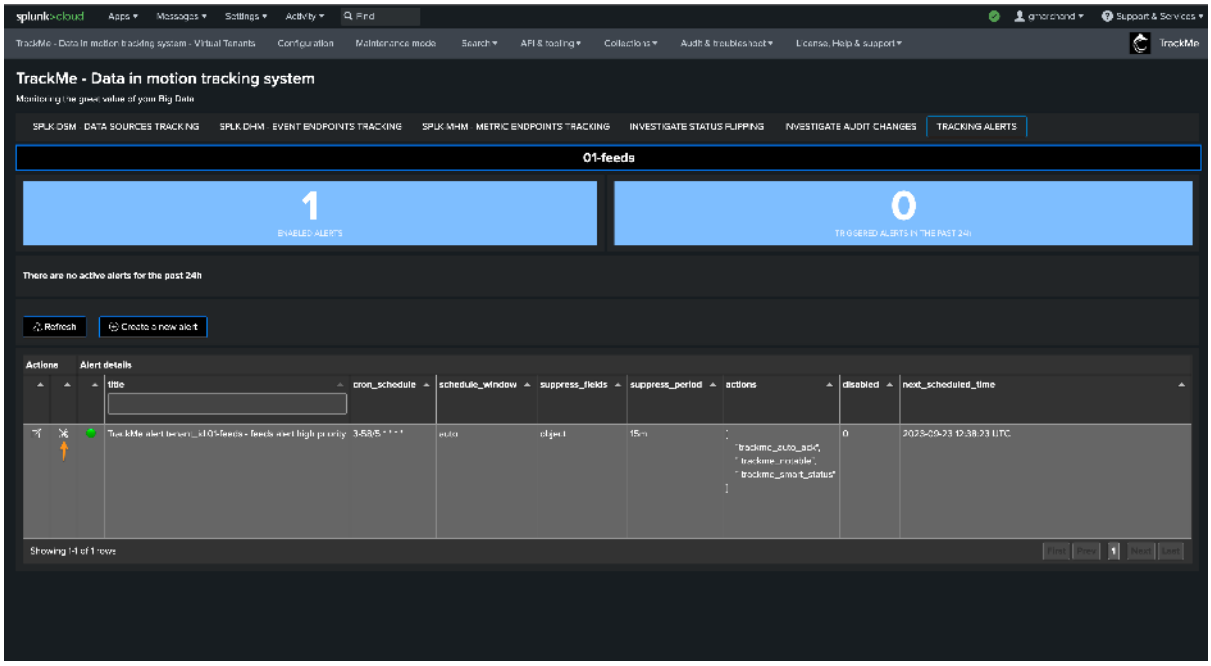
We start by creating an alert in TrackMe for high priority entities:



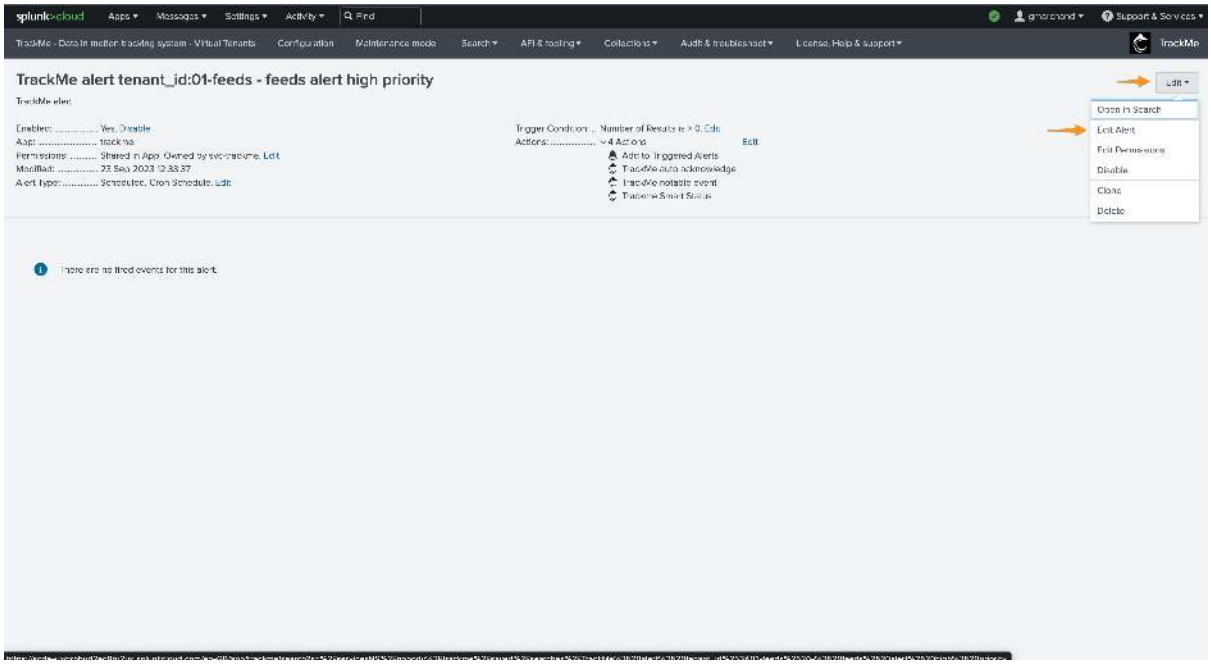
Repeat the same process for other priority levels, all entities should be covered by alerts, in this example we would therefore need 3 alerts in total.

Step 2: Enable alert actions per alert

The next step is to update TrackMe alerts to include the alert actions to be enabled:

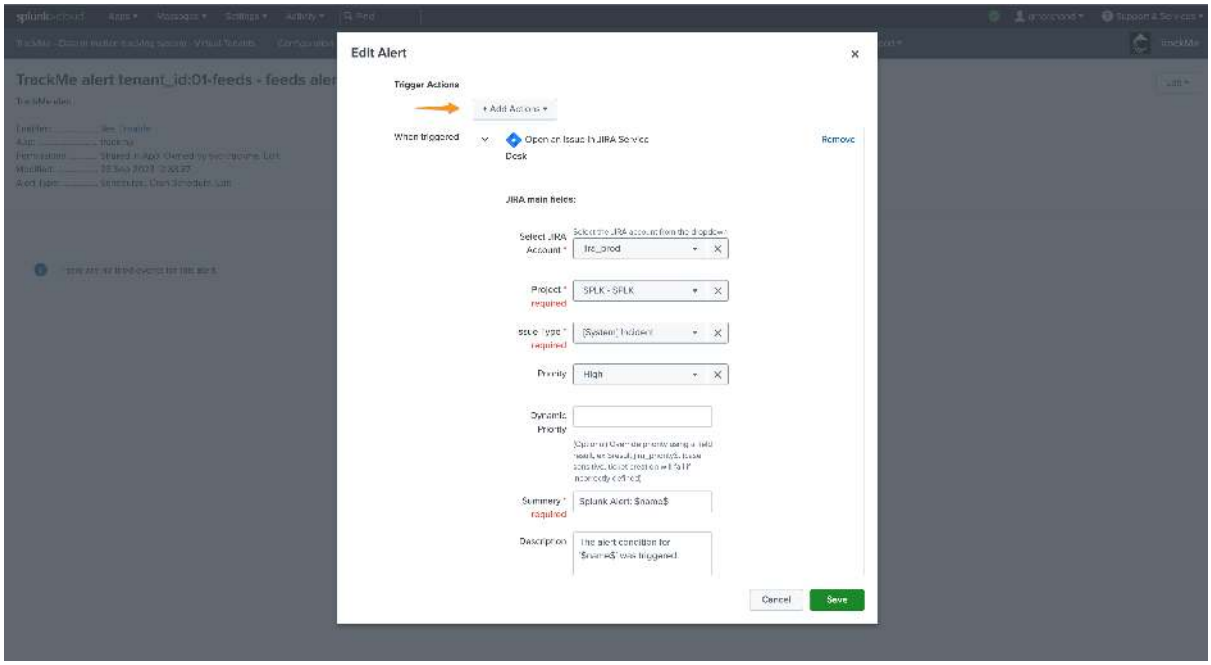


This opens the Splunk alert edition management screen, we will add a new alert action:



We add for instance add the Jira alert action and define some of the most use fields returned by TrackMe:

These steps are the same as in the architecture scenario 1 Notable based, consult the previous section for more insights.



Context body example:

```
TrackMe alert for a Splunk high priority entity:

- tenant: $result.tenant_id$
- object: $result.object$
- object_category: $result.object_category$
- state: $result.state$
- priority: $result.priority$
- anomaly_reason: $result.anomaly_reason$
```

(continues on next page)

(continued from previous page)

```
- status_message: $result.status_message$
- drilldown_link: $result.drilldown_link$
```

The same process needs to be achieved for every alert that should monitor TrackMe entities, per tenant and component.

When alerts trigger, Splunk automatically configured the alert actions according to your settings.

7.11.7 Deleting TrackMe alerts

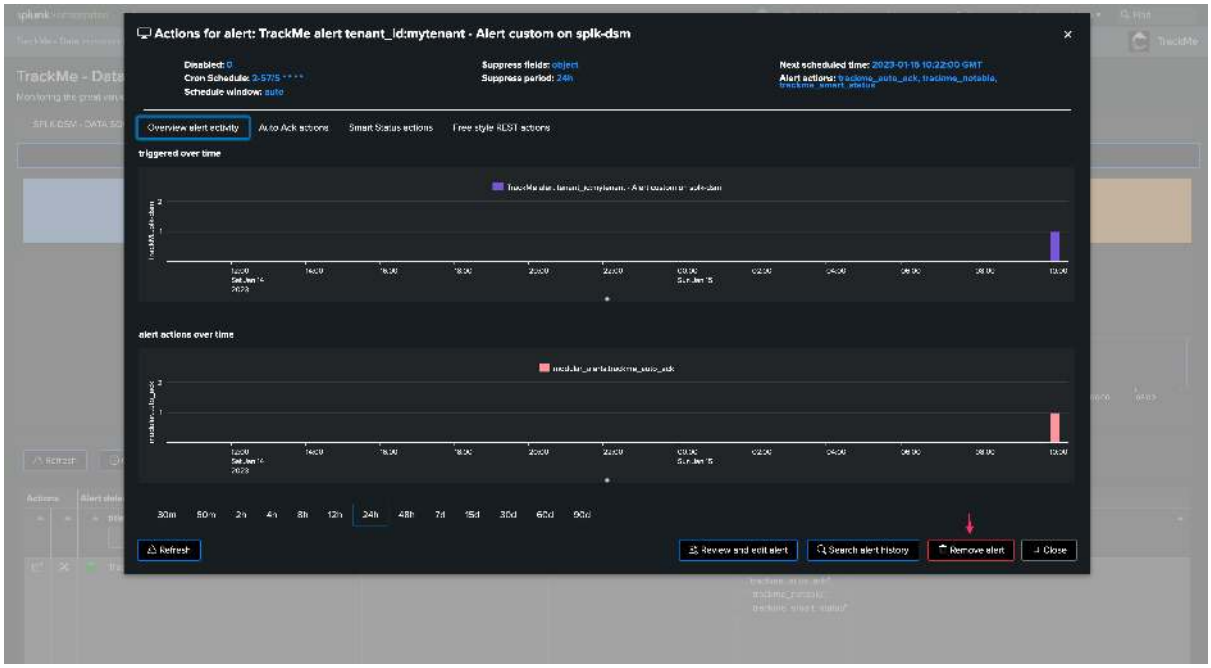
Warning

TrackMe alerts are orchestrated by TrackMe and should be deleted in TrackMe

- When creating a TrackMe alert, it will preserve knowledge of it
- This allows automatically managing assignment and Roles Based Access Control settings, according to the Virtual Tenant setup
- Therefore, you **must** delete a TrackMe alert via TrackMe, and not directly via Splunk

Deleting an alert from the user interface

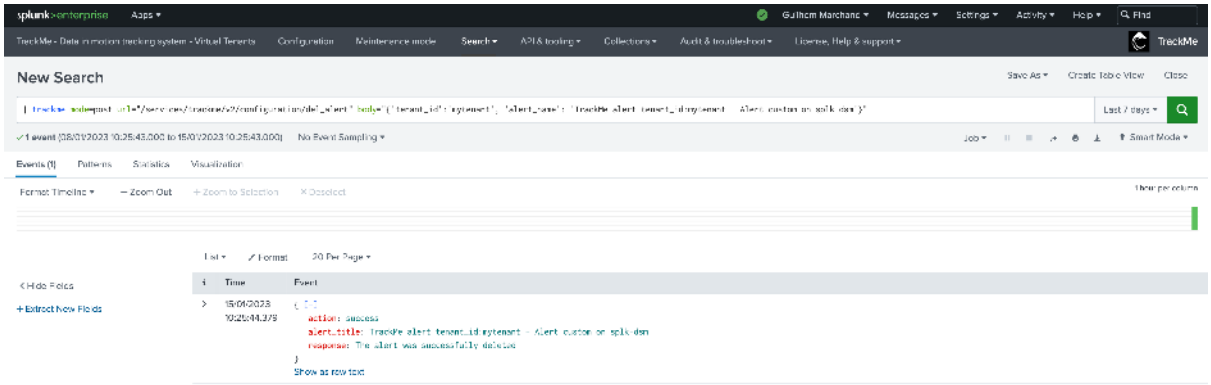
You can delete TrackMe alerts from the user interface, because TrackMe maintains the knowledge of objects associated with a given Virtual Tenant, alerts should be deleted through the user interface or the REST API:



Deleting an alert from the REST API

You can as well delete an alert using the REST API and the following endpoint:

```
| trackme mode=post url="/services/trackme/v2/alerting/admin/del_alert" body="{
↪'tenant_id':'mytenant', 'alert_name': 'TrackMe alert tenant_id:mytenant - Alert␣
↪custom on splk-dsm'}"
```



7.12 Priority Management

7.12.1 About priority levels in TrackMe

Hint

Priority levels are an important concept in TrackMe which can easily be leveraged to categorize the importance of your different Splunk feeds, and for instance redirect notifications depending on an entity's priority level.

- **TrackMe supports the following priority levels:**
  - **critical**
  - **high**
  - **medium**
  - **low**
  - **pending**
- The priority level of an entity is defined by default when **discovered**, and can be updated **per entity**, defined by **policy**, or managed **externally**.
- Entities are usually filtered when forwarding alerts to a third party, such as email or a ticketing system, based on their priority level.
- Priority is a valuable concept that can be leveraged to reduce noise and alert fatigue. A frequent and recommended practice is to focus on high and critical priorities, while other priorities can be considered over time.

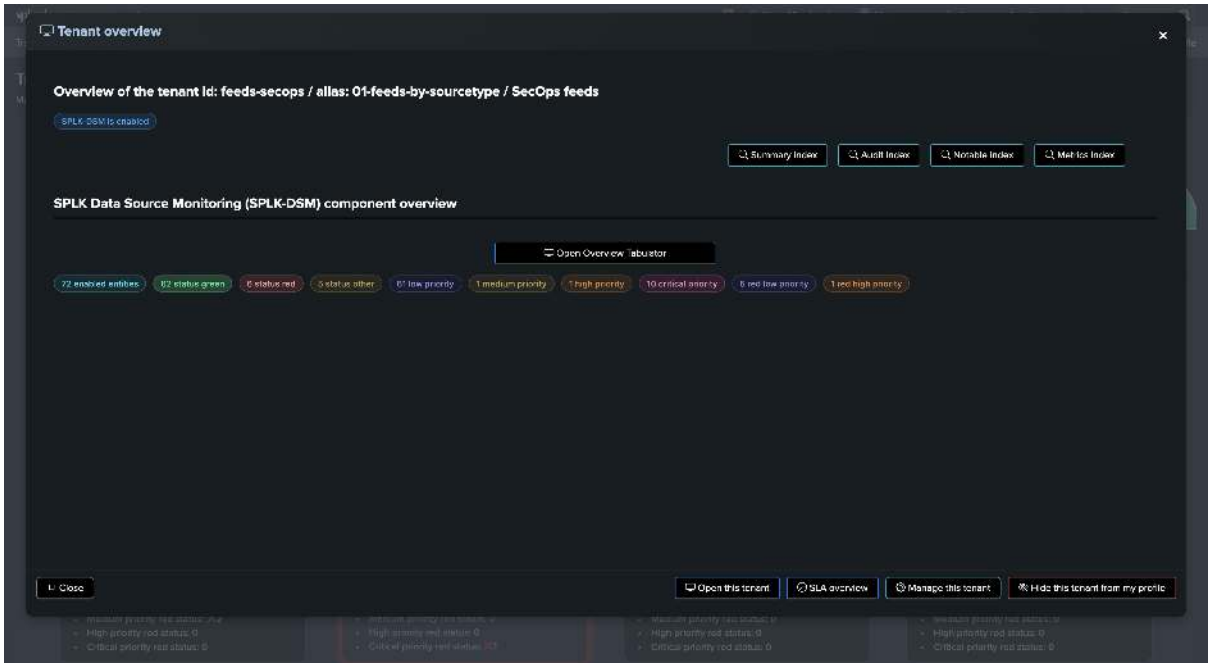
About the pending priority level

- The **pending** priority is a special priority level added in TrackMe 2.1.21, which can be used to qualify newly discovered entities before allowing them to be parts of the alerting framework.
- Especially, when creating alerts, TrackMe calls the following macro:

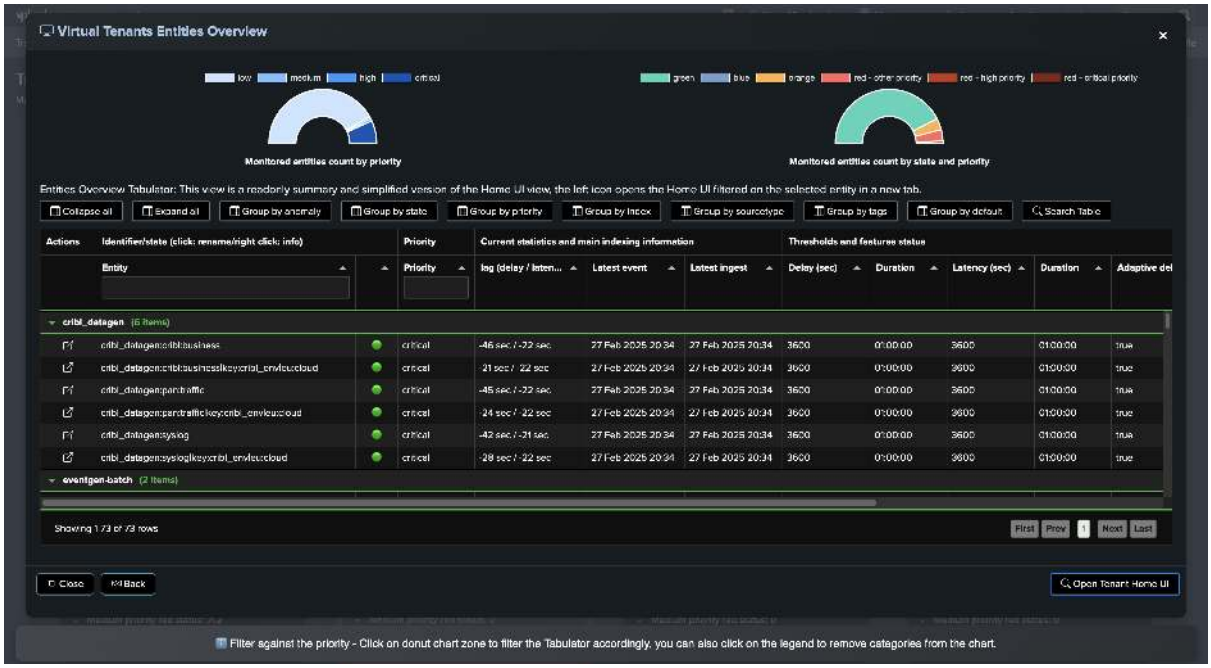
trackme_apply_maintenance_mode_v2 which contains the following exclusion:

```
``` exclude entities with priority pending ```
| where NOT (priority="pending")
```

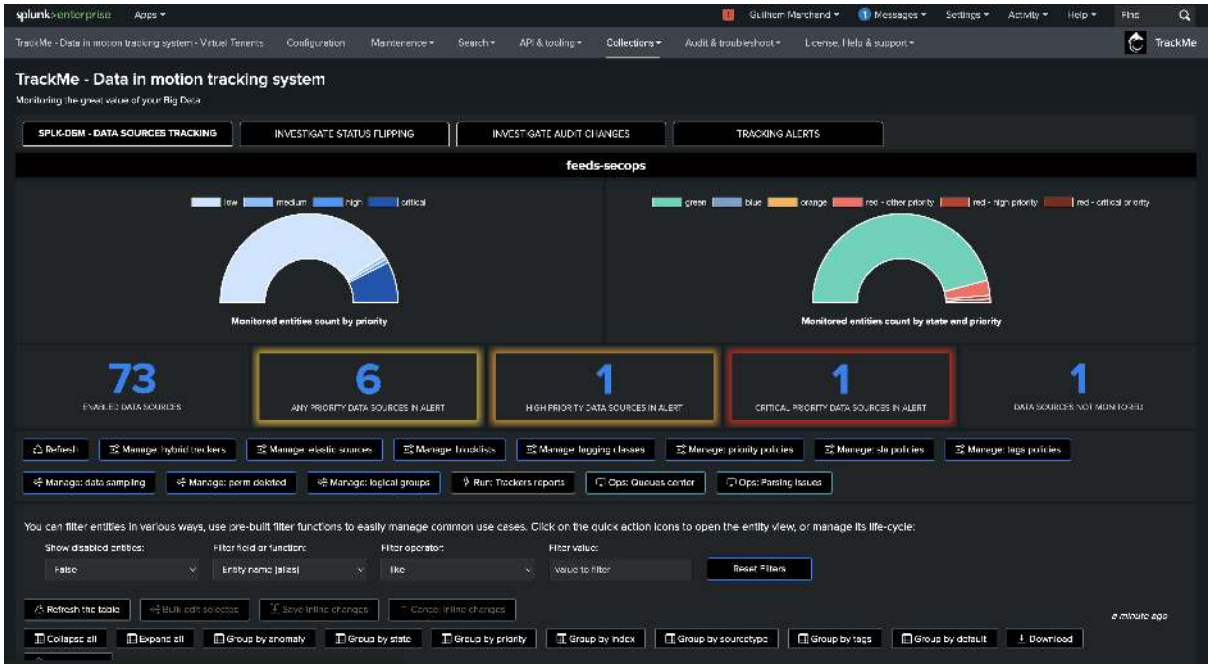
View example from a Virtual Tenant:



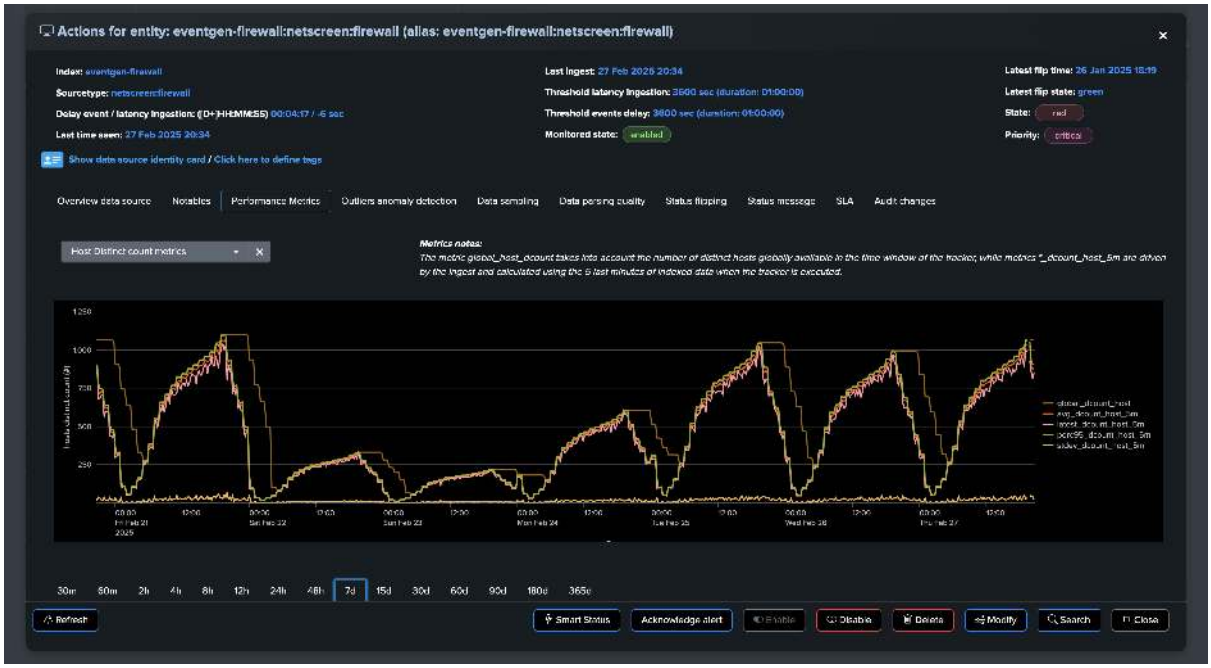
Preview in Virtual Tenant:



View in Home UI:
```



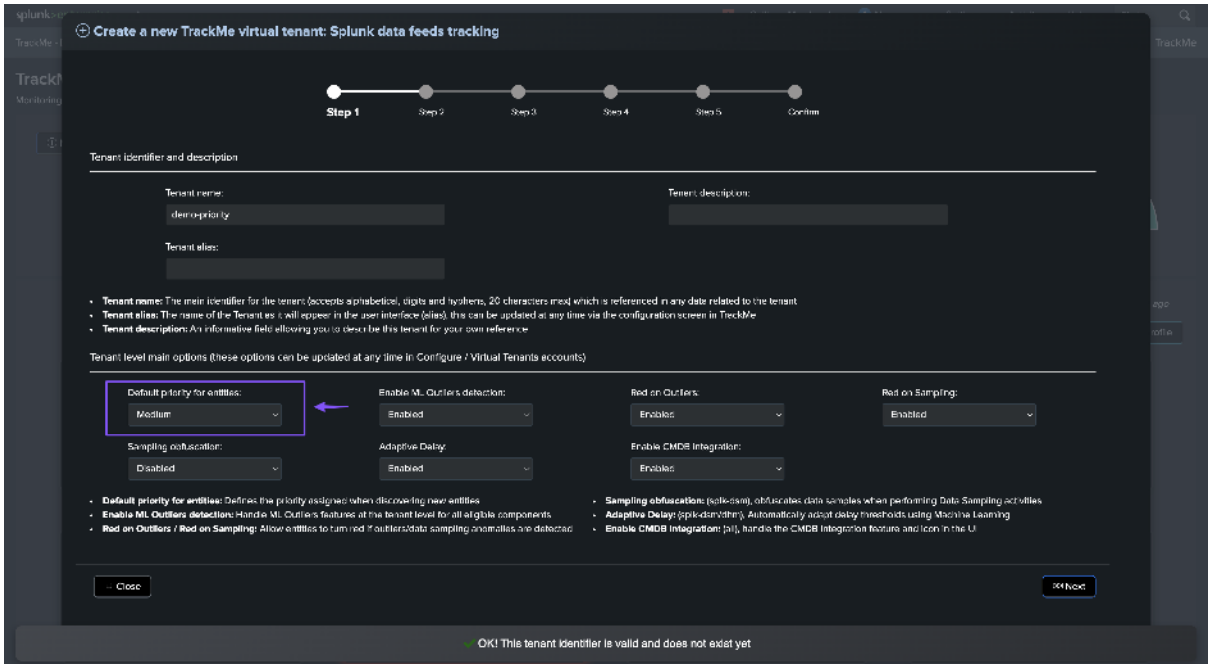
View of an entity:



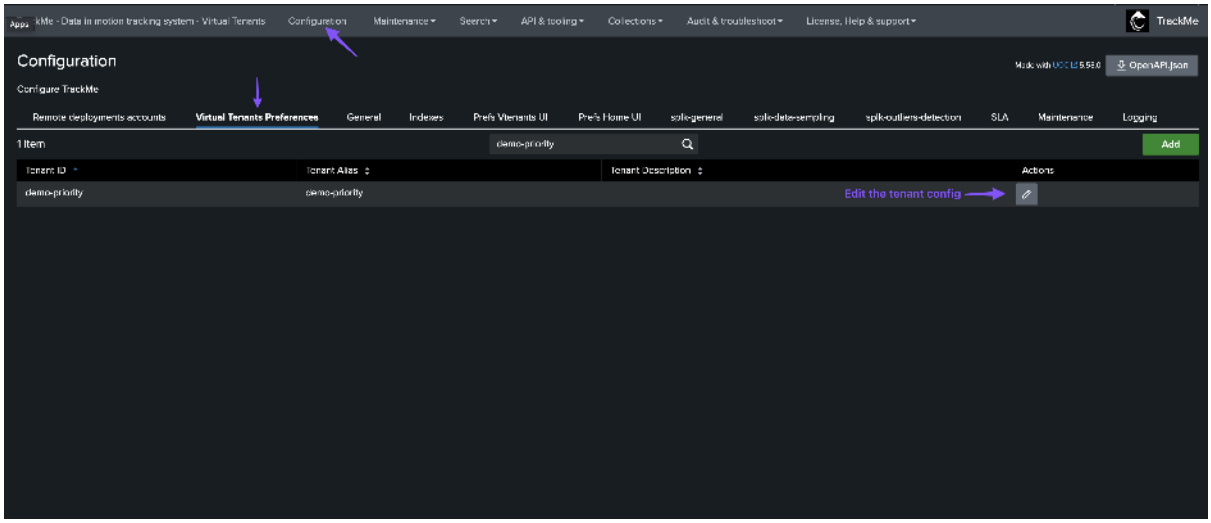
7.12.2 Priority at discovery time

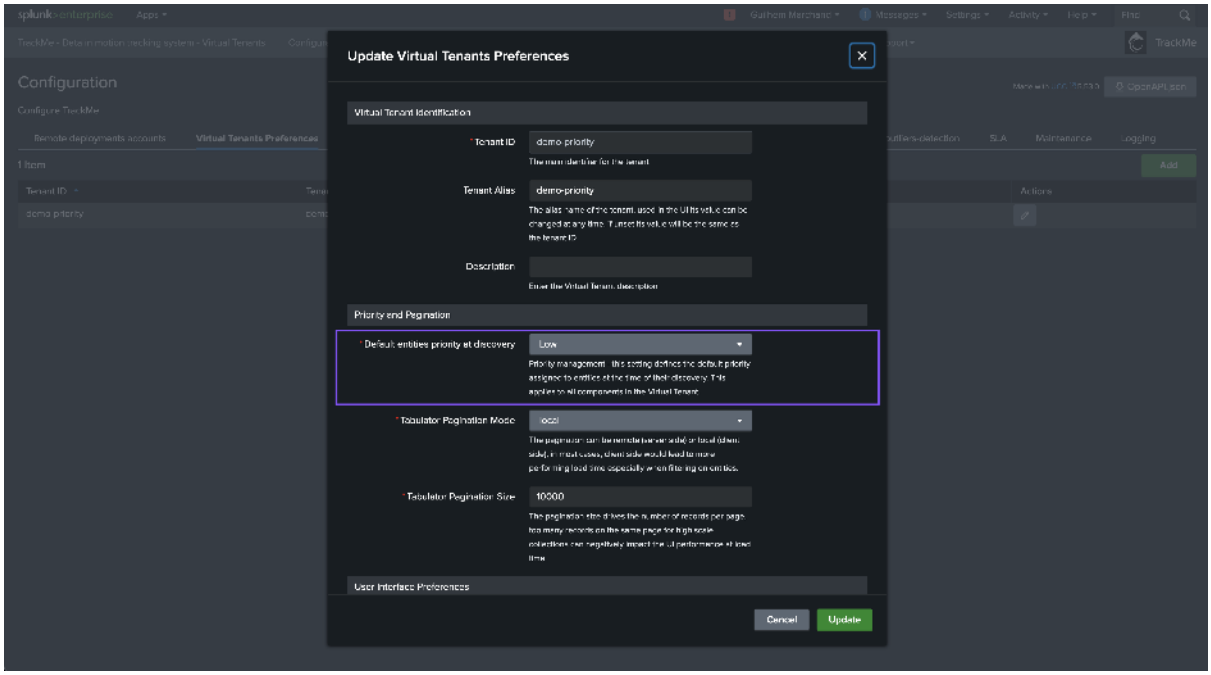
TrackMe applies the priority level at discovery time, which is configurable per Virtual Tenant:

When a Virtual Tenant is created, you can define the default priority that will be applied to entities as they are discovered:



Once the Virtual Tenant is created, you can update the default priority level in the Virtual Tenant's settings:

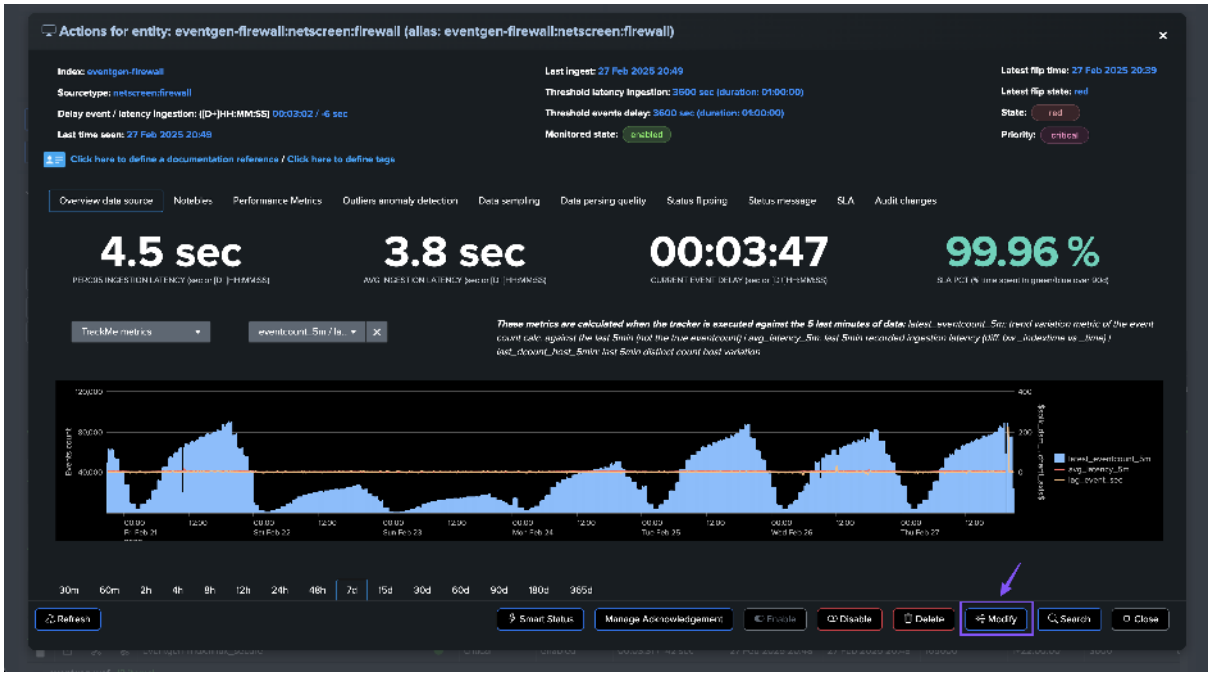


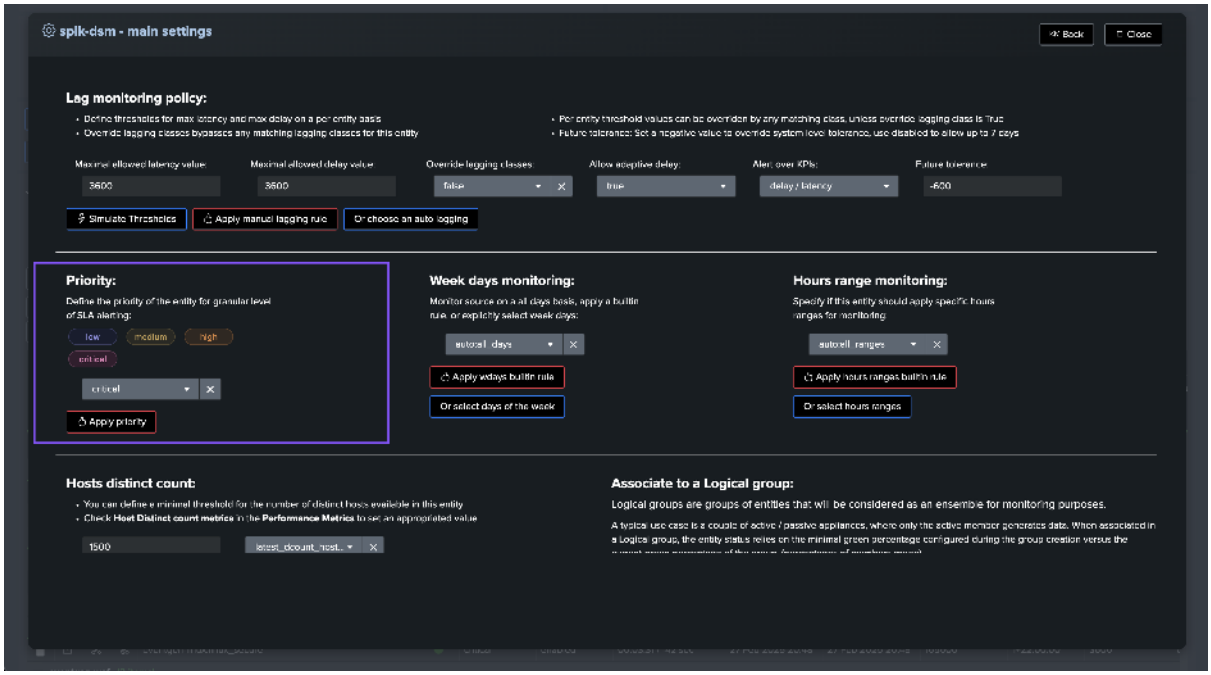


7.12.3 Managing priority at the entity level

Updating the entity priority level in the modification screen

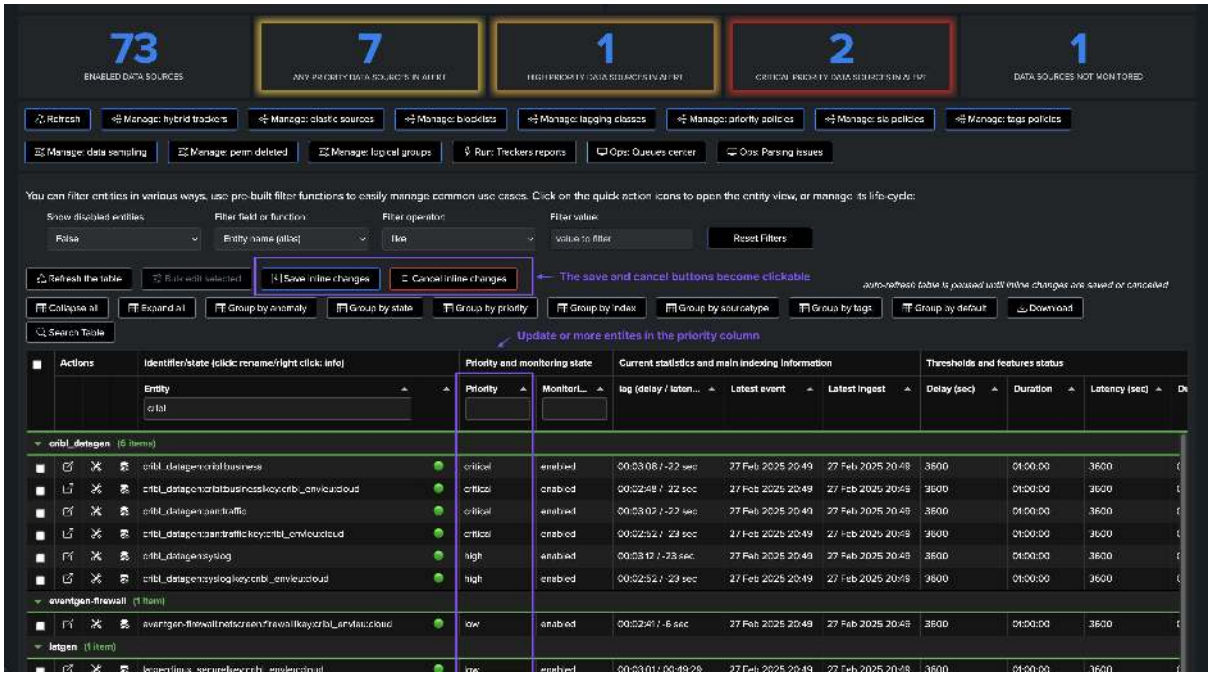
In the main entity screen, you can update the priority from the modification screen:





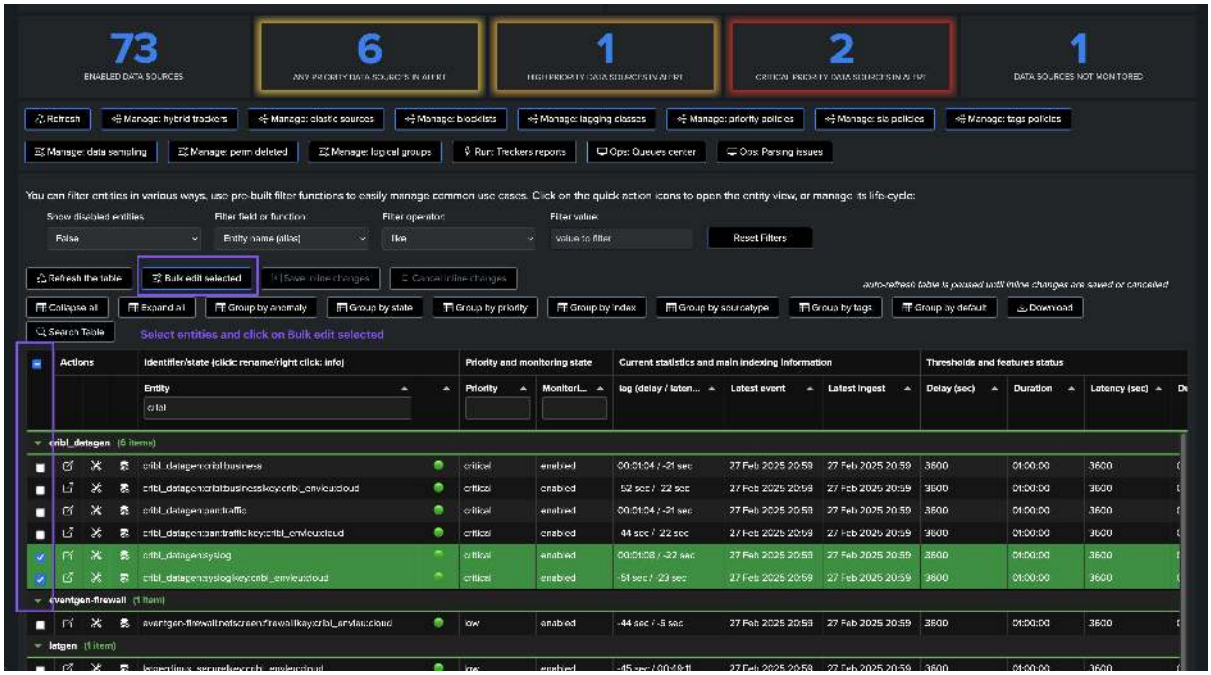
Updating entities in bulk via the table view

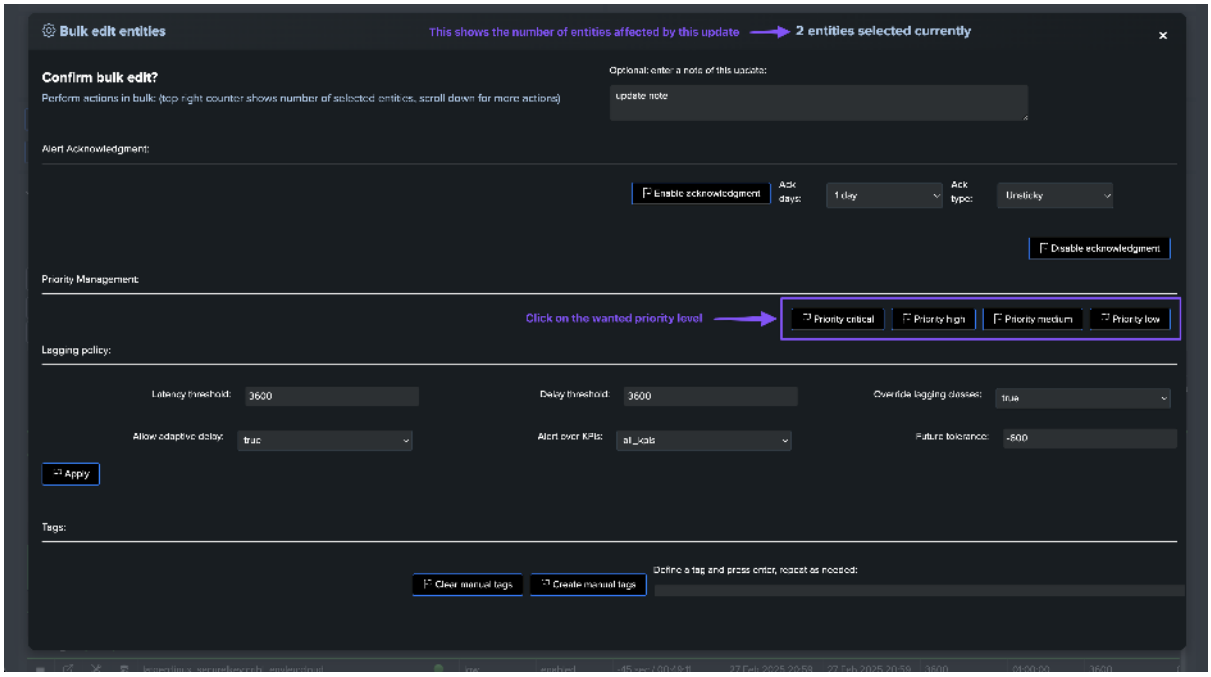
In the Home view, you can update one or multiple entities at once within the table:



Updating entities in bulk selection

You can also select one or multiple entities in the table and update their priority level:



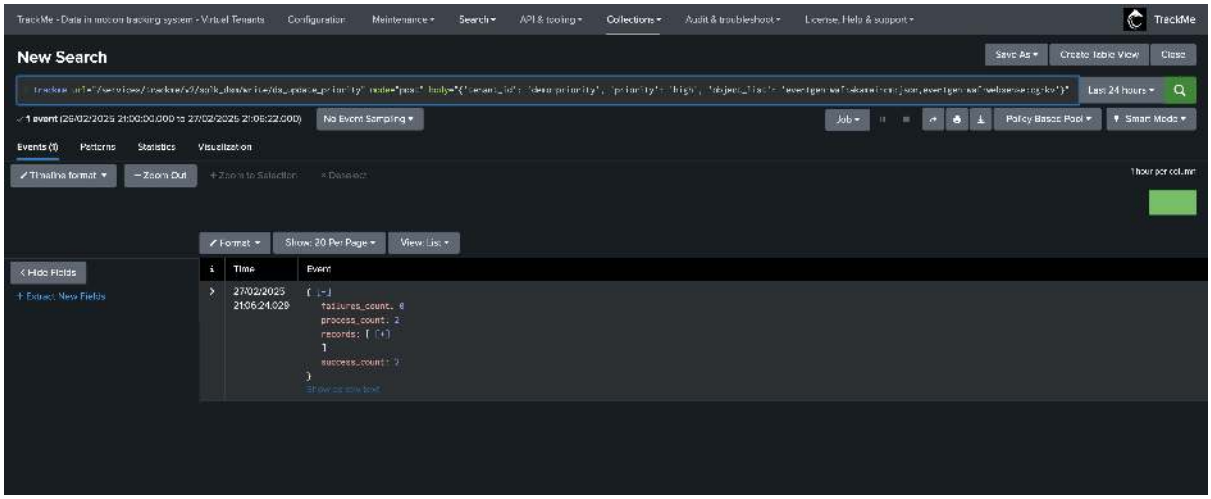


Updating entities in SPL and REST API

You can also update the priority level of entities using SPL and the REST API, or by any other means that can interact with the REST API:

Example of SPL:

```
| trackme url="/services/trackme/v2/splk_dsm/write/ds_update_priority" mode="post"␣
↪body="{'tenant_id': 'demo-priority', 'priority': 'high', 'object_list': 'eventgen-
↪waf:akamai:cm:json,eventgen-waf:websense:cg:kv'}"
```

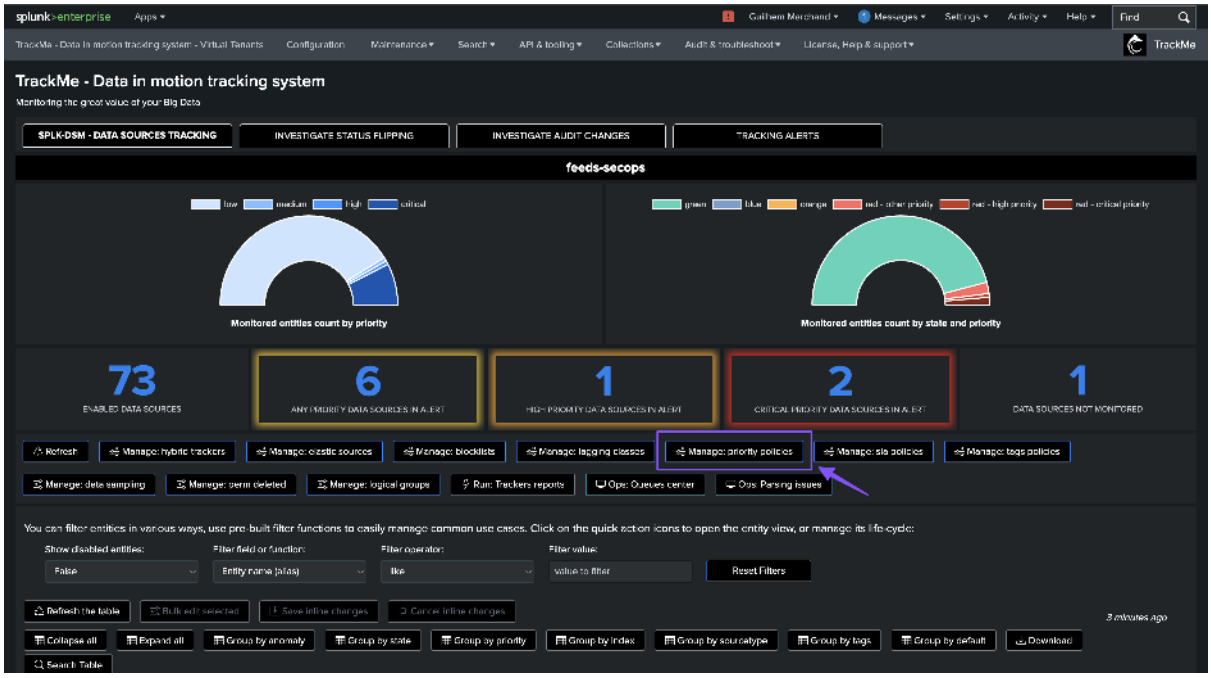


7.12.4 Managing priority via policy

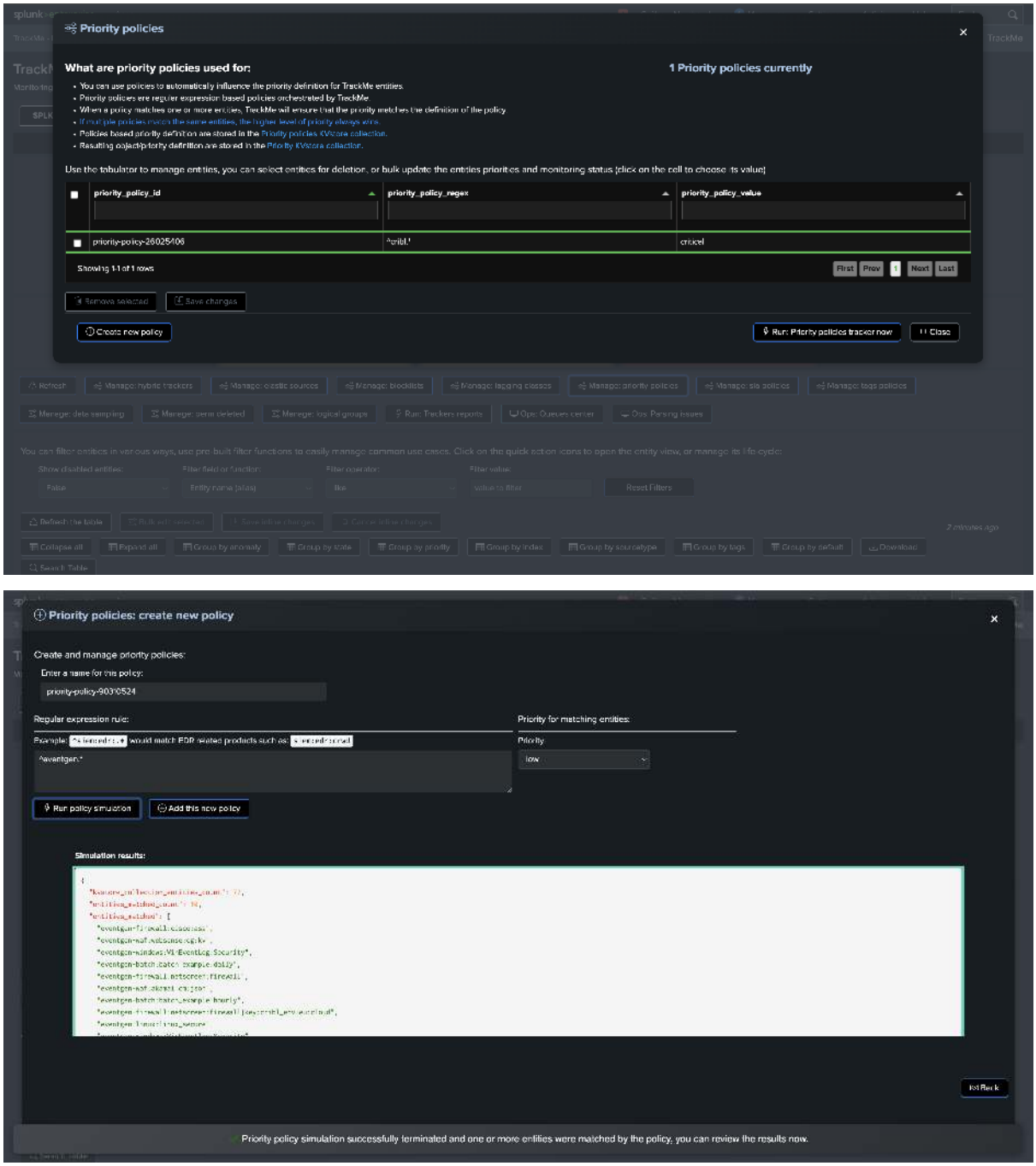
TrackMe supports the management of priority levels via policy, which can be defined per Virtual Tenant:

- Policies are regex-based expressions that are orchestrated by TrackMe automatically.
- Matching entities are automatically updated with the priority level defined in the policy.
- If multiple policies match a given entity, the highest priority level takes precedence.
- **Since TrackMe 2.1.10**, an entity managed by policies can still be updated manually, and the policy will not override the manual update.
- TrackMe will show an informational message in the entity screen, displaying the policy managing the entity, the requested priority level, and the effective priority level.

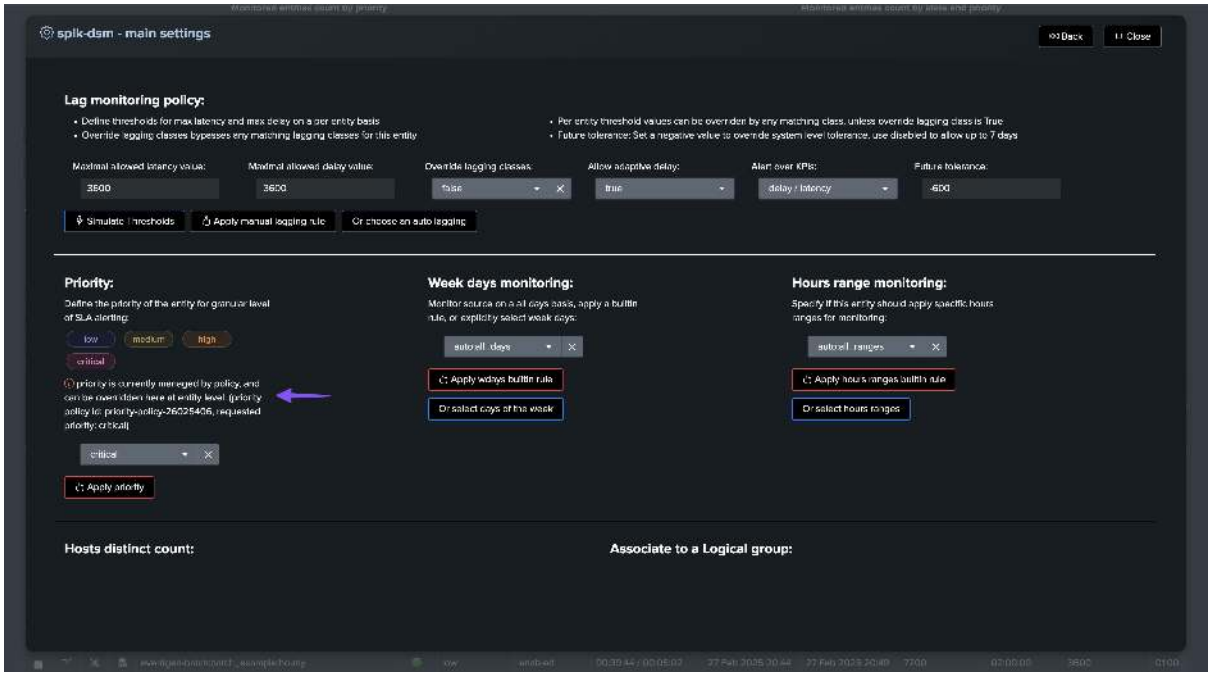
Accessing the policy management screen:



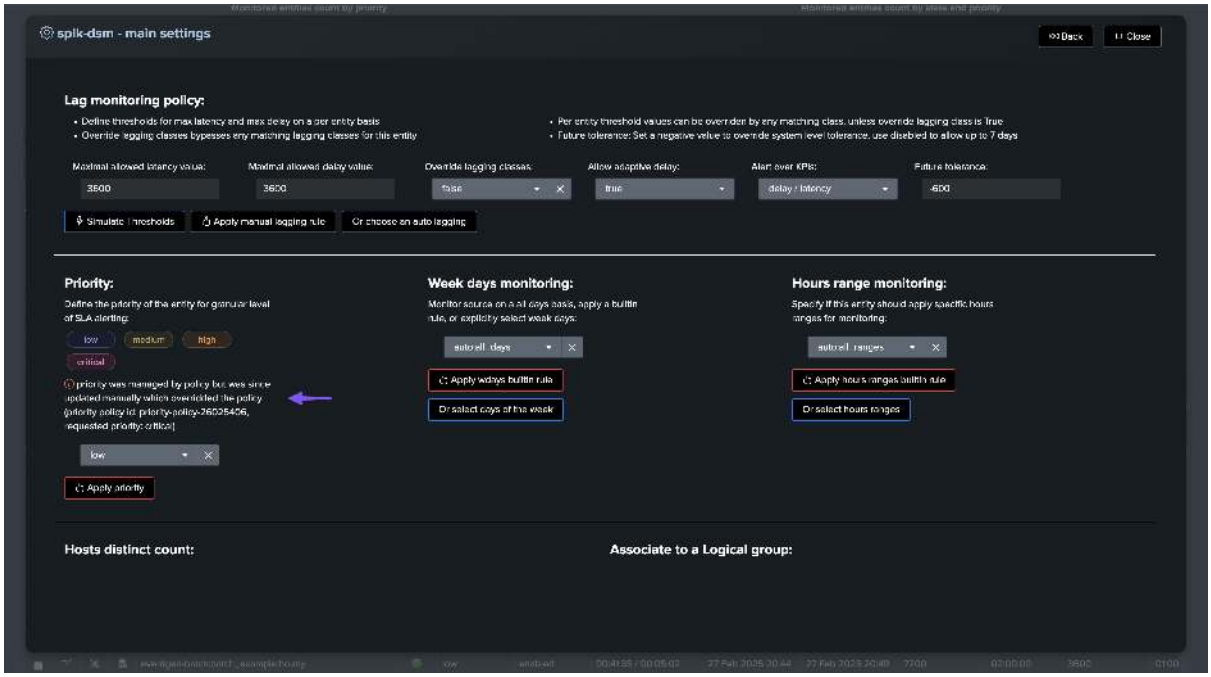
Defining a policy:



Modification screen when an entity is managed by a policy:



Modification screen when an entity is managed by a policy and manually updated:

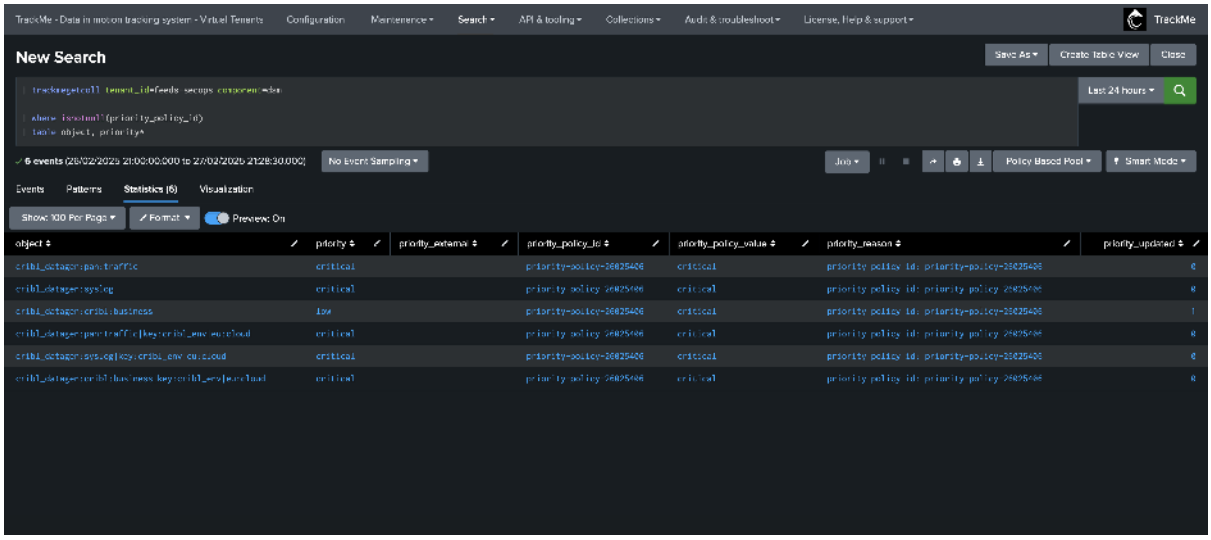


Accessing entities managed by policies using trackmegetcoll:

- Update the tenant_id
- The following SPL can be accessed via the "Search table" button in TrackMe's UI; it leverages the real-time decision maker and TrackMe REST API

```
| trackmegetcoll tenant_id=feeds-secops component=dsm

| where isnotnull(priority_policy_id)
| table object, priority*
```



7.12.5 Managing priority externally

TrackMe supports the management of priority levels externally:

- External management allows you to update the priority level of entities using Splunk and any logic of your own.
- Priority policies take precedence over external management.
- If an entity is managed externally, the priority level can still be updated manually, and external management will not override the manual update.
- TrackMe will show an informational message in the entity screen, displaying the external management managing the entity, the requested priority level, and the effective priority level.
- These instructions require **TrackMe 2.1.10** or later.

Example of SPL:

- Update the tenant_id
- Update the search to match your needs, lookup files and logic

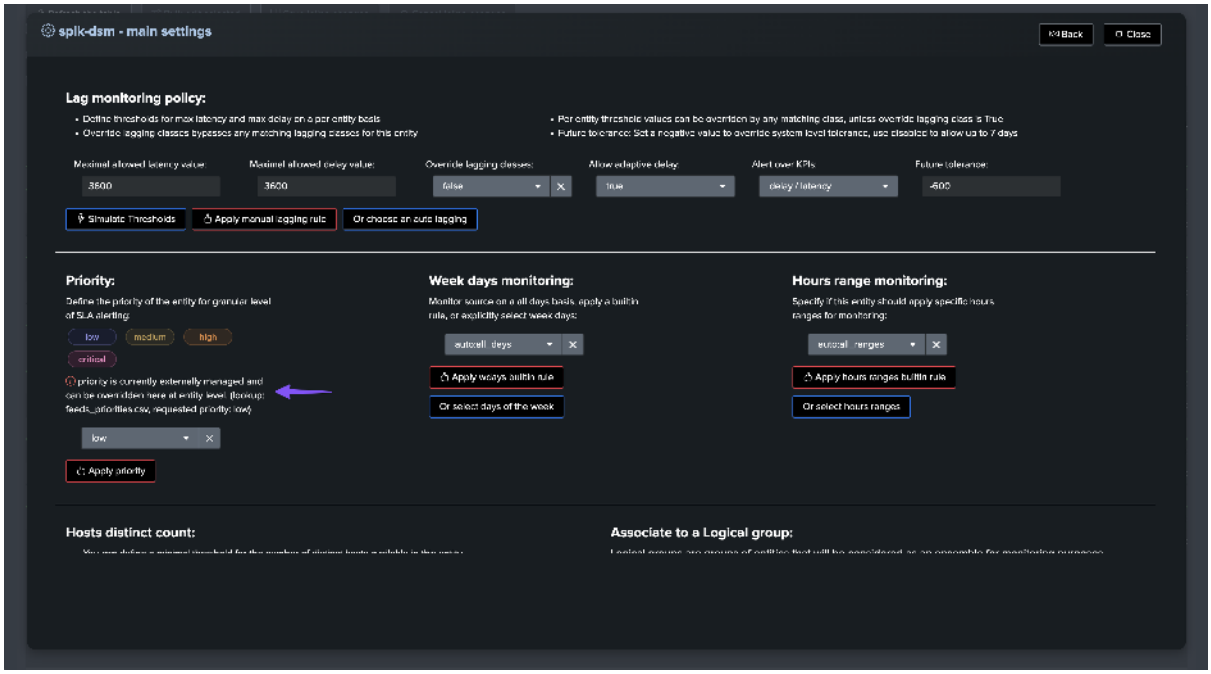
```
| inputlookup trackme_dsm_tenant_feeds-secops | eval keyid=_key

``` In this example, we leverage a Splunk lookup file referencing indexes and used to␣
↪define the field priority_external ```
| lookup feeds_priorities.csv index as data_index OUTPUT priority as priority_external
| where isnotnull(priority_external)

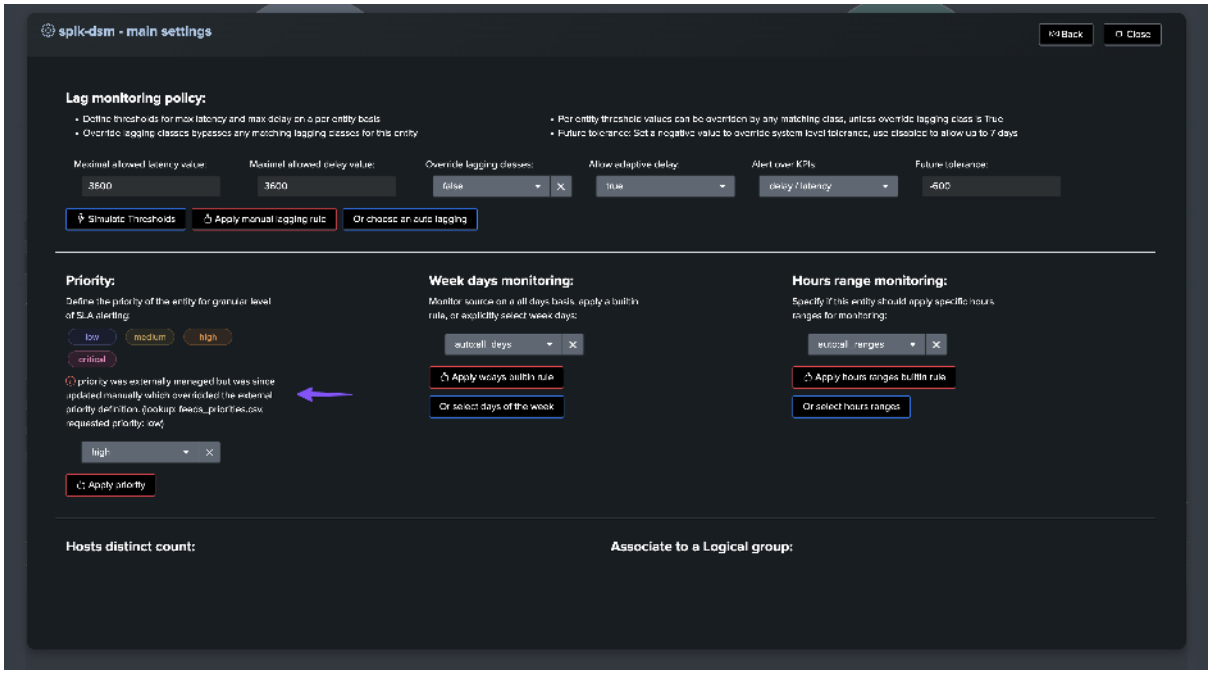
``` The field priority_reason will be used by TrackMe to display an informational␣
↪message ```
| eval priority_reason="lookup: feeds_priorities.csv"

``` Finally, the KVstore records will be updated; schedule this search so that any␣
↪newly discovered entity will retrieve the expected externally managed priority ```
| outputlookup append=t key_field=keyid trackme_dsm_tenant_feeds-secops
```

Modification screen when an entity is managed externally:
```



Modification screen when an entity is managed externally and manually updated:

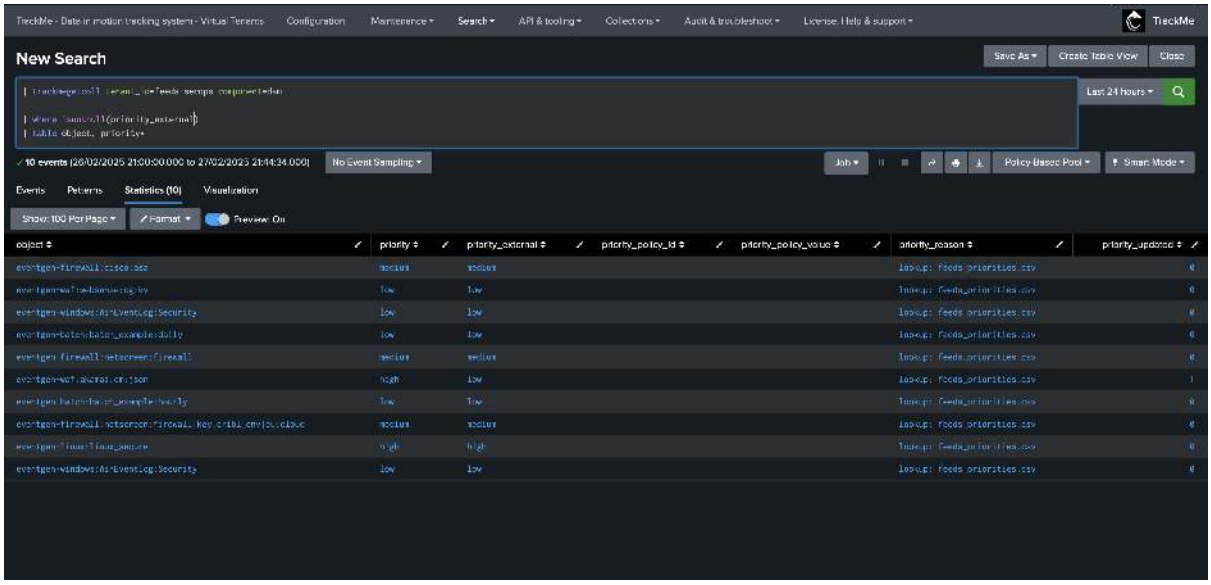


Accessing entities managed by policies using trackmegetcoll:

- Update the tenant_id
- The following SPL can be accessed via the "Search table" button in TrackMe's UI; it leverages the real-time decision maker and TrackMe REST API

```
| trackmegetcoll tenant_id=feeds-secops component=dsm

| where isnotnull(priority_external)
| table object, priority*
```



7.13 Outliers Anomaly Detection

7.13.1 Machine Learning Outliers Anomaly Detection in TrackMe

TrackMe implements Machine Learning Outliers Anomaly detection in every component, from the feeds tracking to the monitoring of scheduled activity in Splunk, based on the following concepts:

- TrackMe relies on the **Splunk Machine Learning Toolkit** and the **Python Scientific Packages** with its own custom logic and workflow which orchestrates the lifecycle of anomaly detection in the product
- We use the `apply` and `fit` commands from the toolkit and orchestrate their usage when entities are discovered and maintained; the `density function` is used for anomaly detection calculation purposes
- TrackMe orchestrates the Anomaly Detection workflow in two essential steps: the ML models generation and training (`mltrain`), and the ML models rendering phase (`mlmonitor`) where TrackMe verifies the Anomaly detection status for a given entity
- Depending on the TrackMe `component`, ML models are generated automatically using metrics that are relevant for the component activity. Models can be created, deleted, and customized easily to change the model behaviors if necessary
- See the following white paper for a great use case around Machine Learning: *Use TrackMe to detect abnormal events count drop in Splunk feeds*

Hint

ML models learn over time from historical data

- ML detection in TrackMe requires historical data and gets more accurate over time
- Depending on the settings, this can require up to several weeks or months of historical data with the most granular parameters
- Historical data translates to the metrics stored in TrackMe's metric store indexes, not the raw data itself
- Metrics start to get generated as soon as TrackMe discovers and starts to maintain an entity
- While TrackMe learns about the data by training ML models, it applies various safeguards to avoid generating false positive alerts

Hint

ML confidence (new in TrackMe 2.0.72)

- Since TrackMe 2.0.72, ML models have a confidence level which is calculated based on the number of days of historical data available for the models training
- By default, TrackMe defines a minimal requirement of 7 days before the confidence level is set to `normal`, otherwise it is set to `low`
- The confidence level and confidence_reason are stored in the rules KVstore collection, used while rendering models and also displayed in the **Manage Outliers screen**
- The minimal number of days of historical metrics to define the confidence level to `normal` is driven by a system-wide configuration option called `splk_outliers_min_days_history`

Hint

Time factor "none" for non-seasonality driven outliers (new in TrackMe 2.0.72)

- Since TrackMe 2.0.72, ML models can be set with the `time_factor` defined to `none`
- This enables TrackMe's outlier calculations to exclude seasonal concepts; in some cases, this can better address KPIs which are not driven by seasonality
- This can be set on a per-entity/model basis, or also chosen as the default model setting when TrackMe initiates ML models (option: `splk_outliers_detection_timefactor_default`)

Hint

TrackMe 2.0.84 Outliers evolutions and enhancements

- In TrackMe 2.0.84, we have released major enhancements to the Outliers engine; the following notes describe the most important changes
- **splunk-system-user and private ownership:** ML Models are now owned by the splunk-system-user and private; this prevents having the models showing up in Splunk Web or the Splunk App for lookup editing, as well as increasing Splunk API response time when loading a large number of lookups
- **schema-upgrade:** TrackMe via its schema upgrade process will automatically reassign existing ML models within 5 minutes of the upgrade
- **Improved Outliers endpoints and custom commands:** TrackMe's Outliers-related endpoints have been improved with better, more sophisticated, smarter, and more efficient code! Training and rendering Outliers TrackMe commands are notably improved with enhanced outputs and behaviors
- **Minimal thresholds for LowerBound and UpperBound Outliers:** You can now, on a per-model basis, define minimal threshold values for LowerBound and UpperBound outliers; values not respecting these thresholds are rejected automatically
- **Outliers investigations and management user interfaces:** Outliers-related user interfaces have been improved to provide more visibility. This includes single views showing the distinction between LowerBound and UpperBound outliers, rejected Outliers, and corrected Outliers for each category
- **True context Simulation model training:** TrackMe's Outliers simulation now runs in `true`

`context`; this means that when you run a simulation, TrackMe trains a simulation model which 100% respects the live model behaviors. This provides a true context and prevents any deviation or inconsistency compared with results in the live Outliers view

- **Automated ML training at the backend level when TrackMe calls rendering functions:** TrackMe now automatically trains an ML model when it calls the rendering phase if it detects that the model is out of date and has not been recently updated. The maximum number of days since last training is controlled by a system-wide parameter `splk_outliers_max_days_since_last_train_default` (15 days by default)
- **Orphan ML records and ML models cleanup:** TrackMe now automatically cleans up orphan ML records, as well as orphan ML models. This is controlled by a new health global tracker called `trackme_general_health_manager` which runs daily. This job handles global health-related tasks for all TrackMe tenants
- **Bulk actions:** Various key actions for Outliers management can be performed in bulk via the user interface: reset Outliers status, enable/disable, run ML train, and ML monitor operations

Hint

TrackMe 2.0.89 Using custom algorithms, fit and apply extra parameters, additional selectable time periods and customizable boundaries extraction

- In TrackMe 2.0.89, we have released various additional capabilities, notably for customers with advanced Machine Learning requirements or practice
- **Custom algorithms:** You can define custom algorithms at the global configuration level (Configuration UI). These algorithms become available for selection when creating or updating ML models, and you can also define the default algorithm to use when TrackMe initiates ML models
- **boundaries extraction macro:** TrackMe now refers to a Splunk macro for the extraction of the boundaries. To address any requirements, you can also add custom boundaries extraction macros and influence the default macro used when creating or updating ML models
- **fit extra parameters:** You can define extra parameters for the `fit` command, defined by default and/or modified per ML model. With this feature, you can define extra parameters allowed by the Splunk MLTK. For more information, see: https://docs.splunk.com/Documentation/MLApp/latest/User/Algorithms (An example of usage: you can define the extra parameters to `exclude_dist="beta"` to exclude the Beta distributions in the density function)
- **apply extra parameters:** Similarly, you can define extra parameters for the `apply` command, defined by default and/or modified per ML model
- **additional time periods:** More period options have been added, so you can extend the training of models for long time ranges beyond 90 days

Hint

TrackMe 2.1.10 Machine Learning Outliers selective enablement and UI charting fix

- In TrackMe 2.1.10, we have released the capability to selectively enable or disable Machine Learning Outliers on a per-tenant AND component basis
- This relies on the `mloutliers` and `mloutliers_allowlist` parameters; the allowlist parameter permits restricting the list of components for which Machine Outliers enablement applies
- As well, we have fixed a remaining UI issue that was leading to charting not loading under some circumstances

7.13.2 Data Seasonality and Behaviors

In most cases, data have typical patterns which we can eventually recognize when running investigations.

The situation is, however, more complex when it comes to automating this recognition. Machine Learning and current major progress in AI are leading the way through new powerful ways to tackle these challenges.

Note

Generating samples for outliers detection

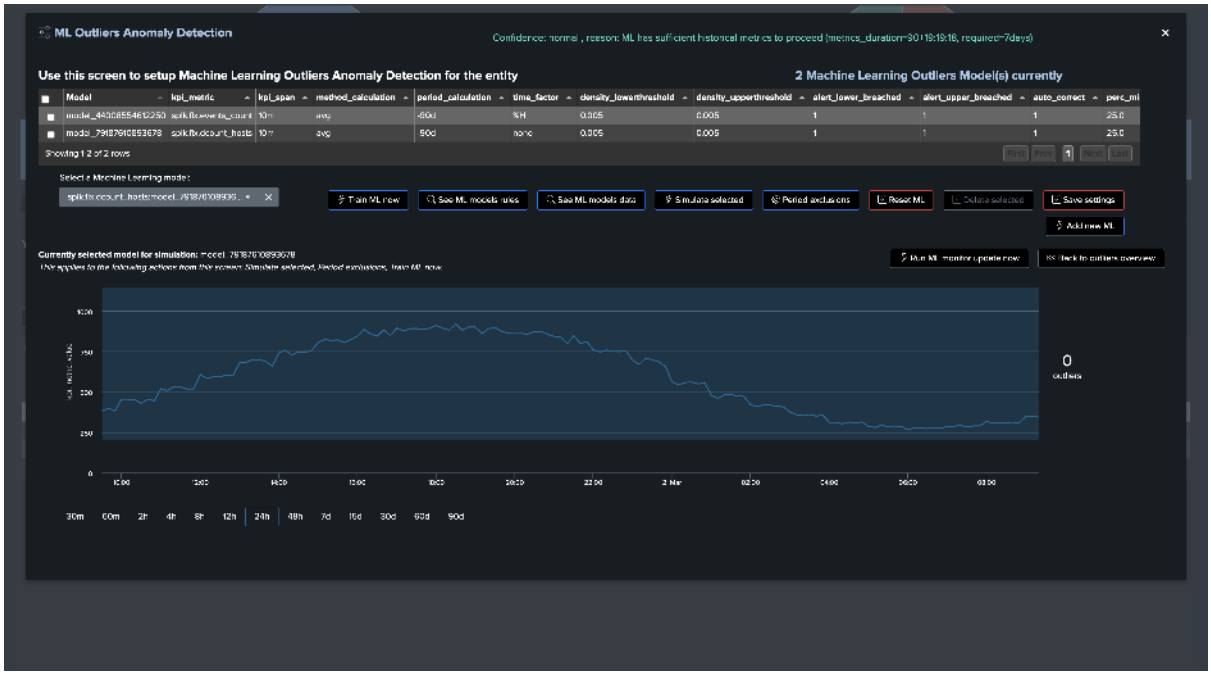
- You can find, download and use with no restrictions the following content: https://github.com/trackme-limited/mlgen-python
- We use this content to generate samples with seasonality concepts for the purposes of development, qualification, and documentation

Sample pattern over past 30 days, seasonality by weekdays with higher activity during working hours:

7.13.3 Data Not Driven by Seasonality

In some cases, applying seasonal concepts to the data may not be the best approach. KPIs with no variation depending on weekdays or hours are good examples.

Since TrackMe 2.0.72, ML models can be set with the `time_factor` defined to `none`; this enables TrackMe's outlier calculations to exclude seasonal concepts:

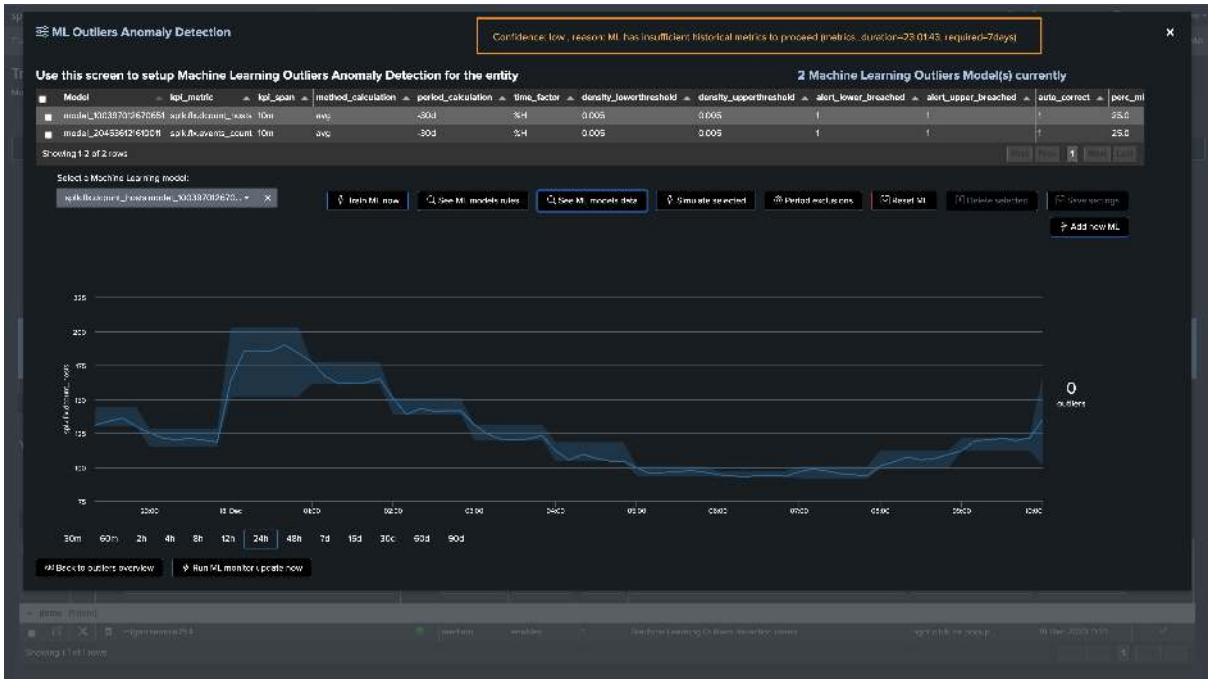


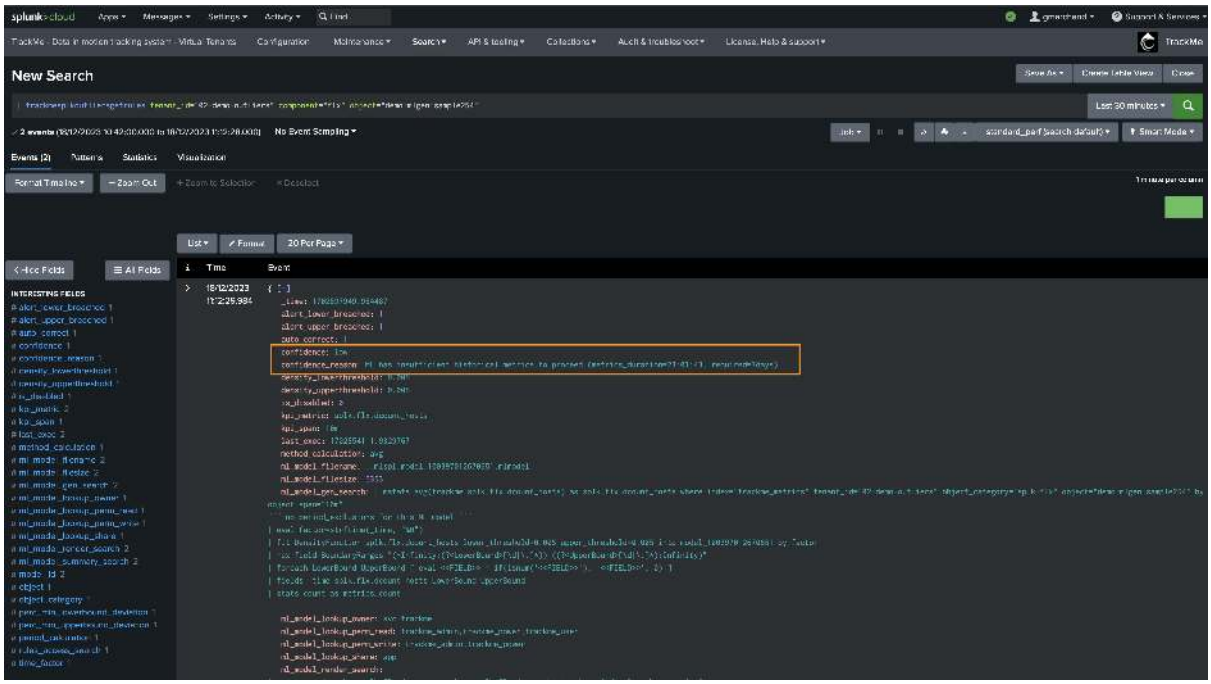
7.13.4 Confidence Level

Since TrackMe 2.0.72, TrackMe establishes a confidence level when training ML models; this confidence level can be:

- `low`: TrackMe maintains ML Outliers for training and rendering purposes, but the Outliers status will not influence the entity status
- `normal`: TrackMe maintains ML Outliers for training and rendering purposes, and the Outliers status will influence the entity status

The value of confidence as well as confidence_reason are stored in the rules KVstore collection; it can be easily viewed in the **Manage Outliers screen** as well as by accessing the rules:





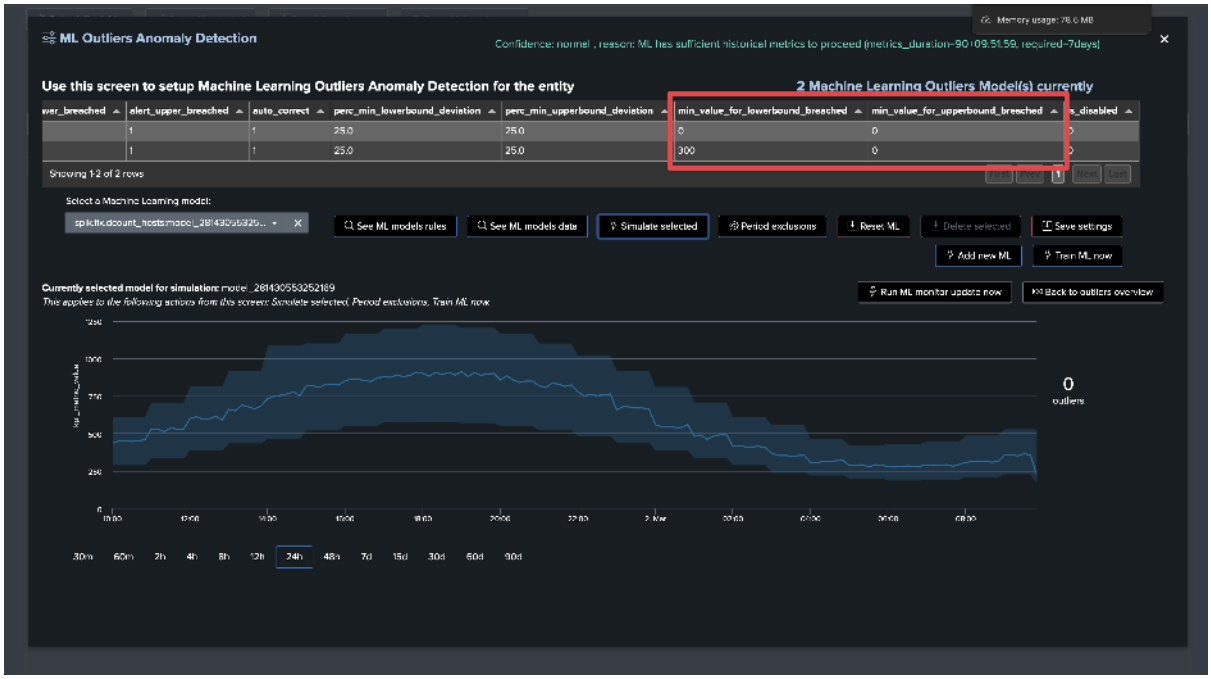
7.13.5 Minimal and Maximum Thresholds for LowerBound and UpperBound Outliers Breaches

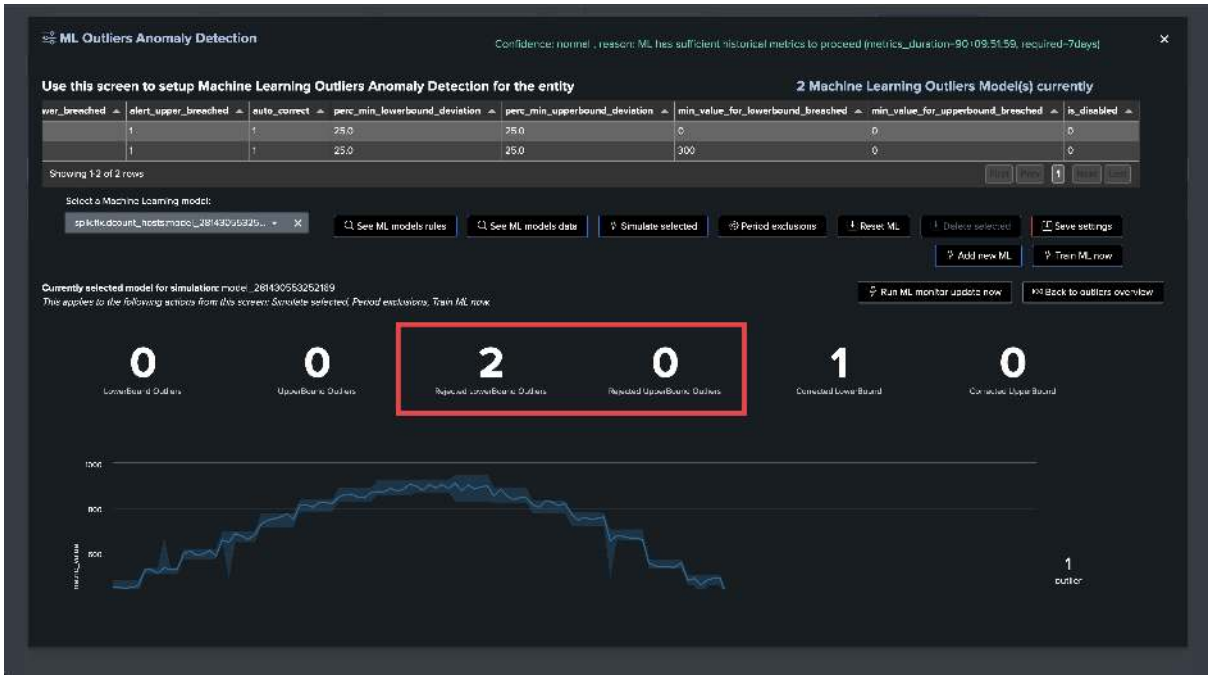
Since TrackMe 2.0.84, you can define minimal thresholds for LowerBound and UpperBound Outliers; values not respecting these thresholds are rejected automatically:

- `min_lower_bound_threshold`: The minimal value for the LowerBound Outliers; values below this threshold are rejected (for LowerBound breaches)
- `max_upper_bound_threshold`: The maximal value for the UpperBound Outliers; values above this threshold are rejected (for UpperBound breaches)

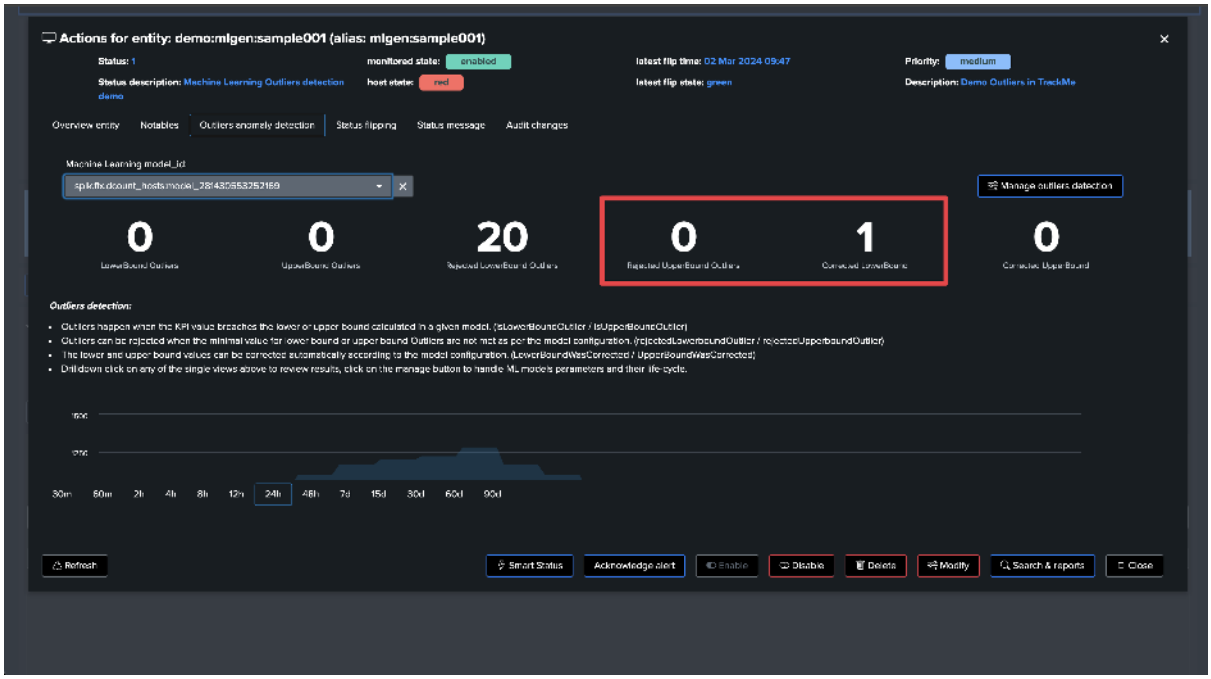
When an Outlier is detected, TrackMe's backend verifies if a min or max threshold has been defined (depending on the type of Outlier); if the value does not respect the threshold, the Outlier is rejected and not taken into account in the entity status.

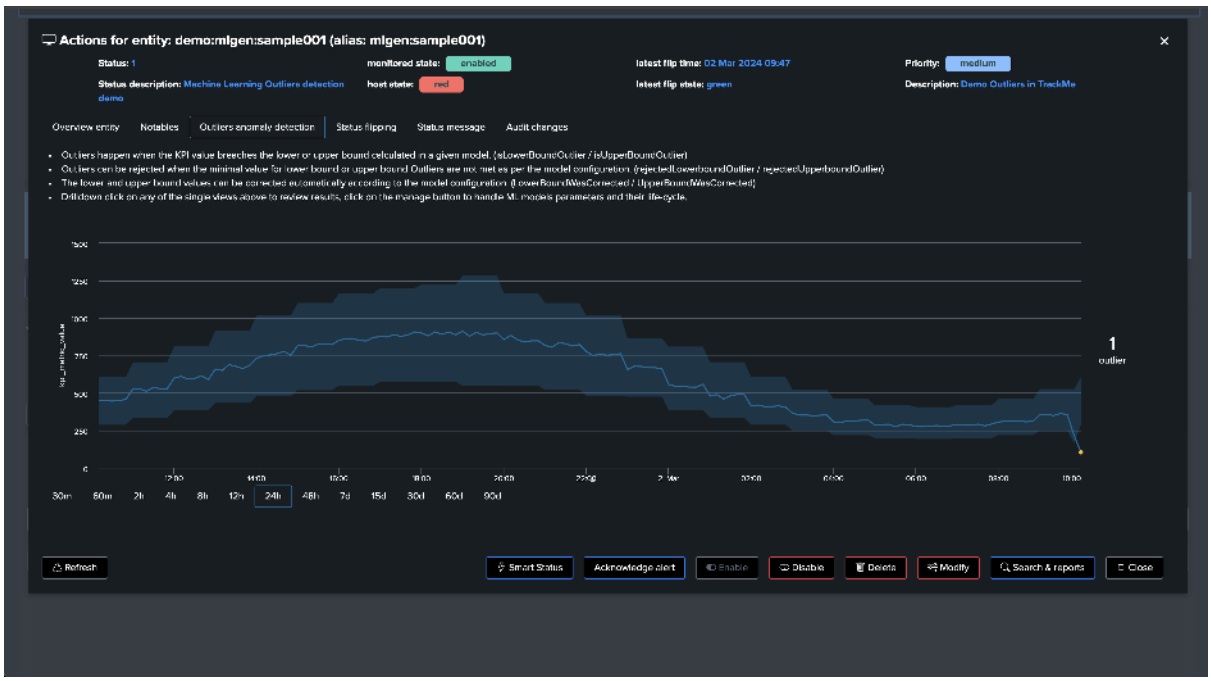
True context Outliers simulation screen:



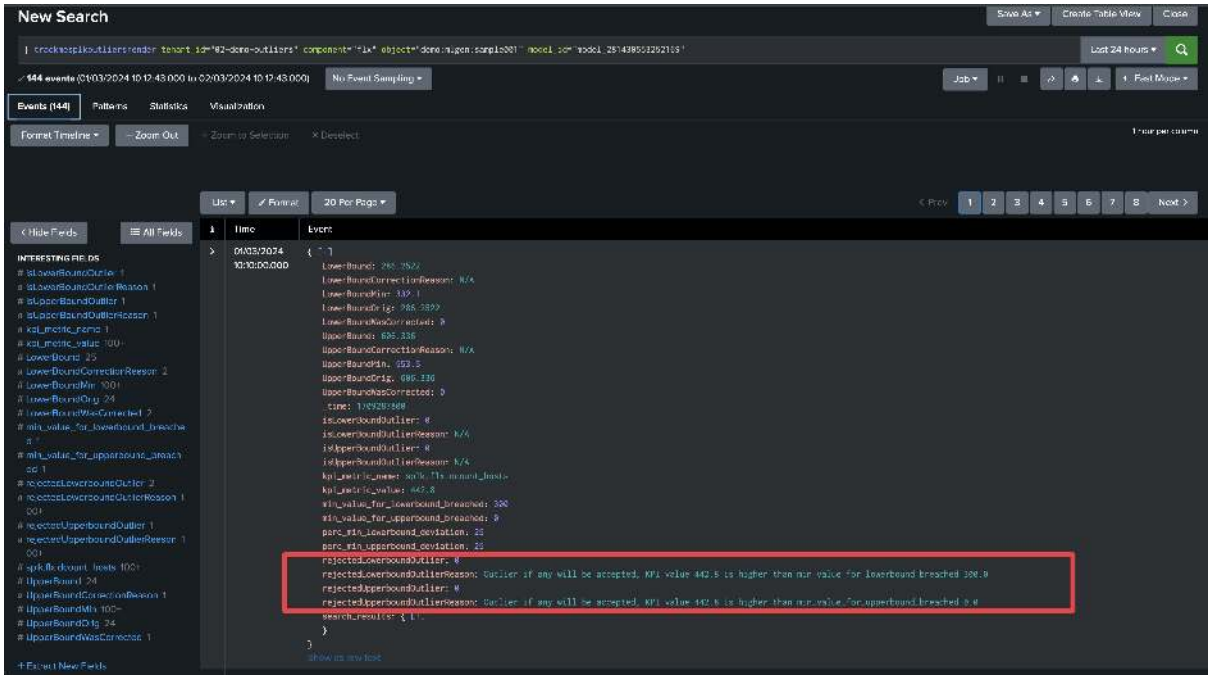


Live Outliers screen:





Rendering TrackMe's commands also show rejected counters and reasons:



7.13.6 Demonstrating Machine Outliers Detection in TrackMe

How ML Outliers Works in TrackMe

In short:

- All components are eligible for Machine Learning Outliers
- Outliers rely on TrackMe-generated metrics only
- This allows running fast and efficient training and rendering searches, with the minimal costs in terms of resources

For the purpose of this demonstration, we create a Flex Object TrackMe tenant which takes into account our ML generator:

- For more information about TrackMe restricted Flex Objects component: *splk-flx - Creating and managing Flex Trackers*

Our Flex tracker:

- Tracker name: "demo"
- Runs every 5 minutes (earliest: -5m, latest: now)

```
index=mlgen ref_sample=*
| stats avg(dcount_hosts) as dcount_hosts, avg(events_count) as events_count by ref_
↪sample

| eval group = "demo"
| eval object = "mlgen" . ":" . ref_sample, alias = ref_sample
| eval object_description = "Demo Outliers in TrackMe"
| eval metrics = "{'dcount_hosts': " . dcount_hosts . ", 'events_count': " . events_
↪count . "}"
| eval outliers_metrics = "{'dcount_hosts': {'alert_lower_breached': 1, 'alert_upper_
↪breached': 1}, 'events_count': {'alert_lower_breached': 1, 'alert_upper_breached':␣
↪1}}"
| eval status=1
| eval status_description="Machine Learning Outliers detection demo"
| table group, object, alias, object_description, metrics, outliers_metrics, status,␣
```

(continues on next page)

(continued from previous page)

```
↪status_description

``` alert if inactive for more than 3600 sec```
| eval max_sec_inactive=3600
```

This Flex Tracker creates entities by monitoring the availability of data in our ML index; it also generates metrics and automates the definition of models which alert on both lower bound and upper bound outliers.

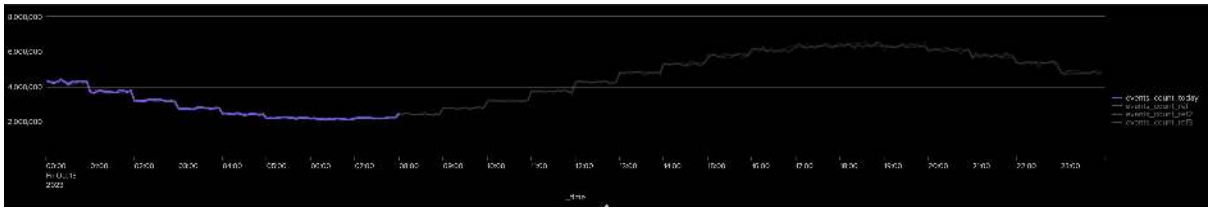
We have our ML generator running and having backfilled the past 90 days of data; it currently does not generate any outliers:

```
index=mlgen ref_sample=sample001 earliest=-90d
| timechart span=1h avg(events_count) as events_count
```

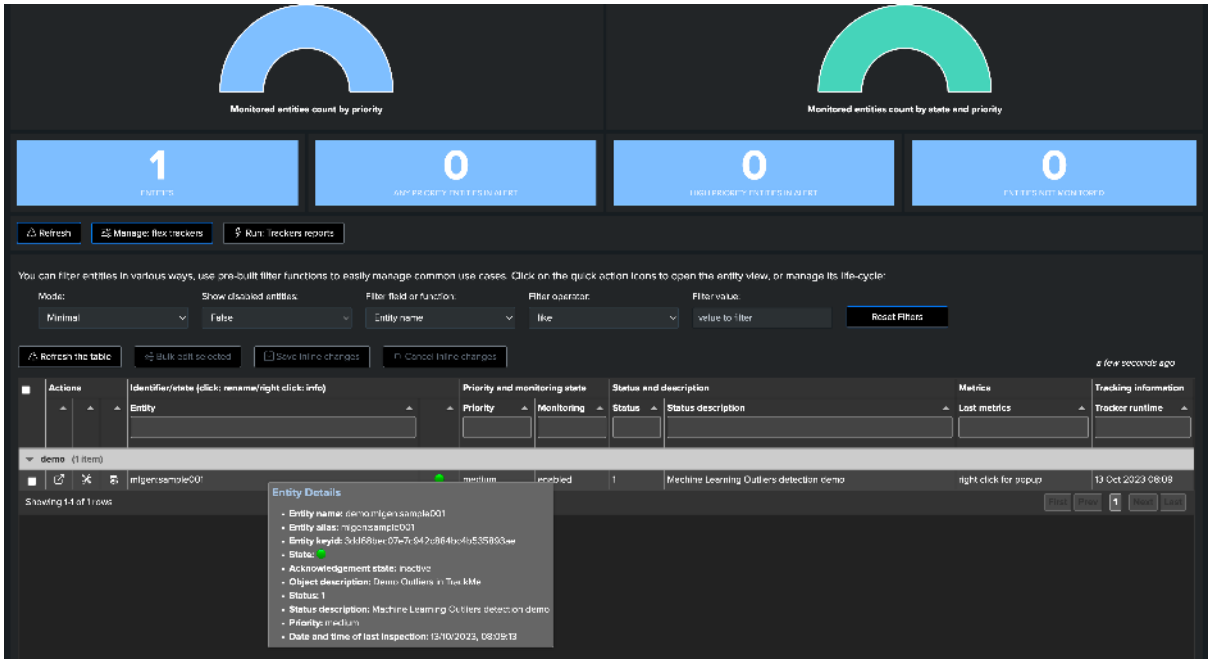
Our ML generator takes into account the weekdays; we can use the following search to compare the relative activity of the current weekdays against the past 4 previous same weekdays:

```
index=mlgen ref_sample=sample001 earliest=@d latest=+1d@d
| timechart span=5m avg(events_count) as events_count_today
| appendcols [
search index=mlgen ref_sample=sample001 earliest=-7d@d latest=-6d@d
| timechart span=5m avg(events_count) as events_count_ref
]
| appendcols [
search index=mlgen ref_sample=sample001 earliest=-14d@d latest=-13d@d
| timechart span=5m avg(events_count) as events_count_ref2
]
| appendcols [
search index=mlgen ref_sample=sample001 earliest=-21d@d latest=-20d@d
| timechart span=5m avg(events_count) as events_count_ref3
]
```

Results:



TrackMe automatically discovered the entity, let's take note of its internal identifier which we will use to manually backfill TrackMe metrics, as if we had been monitoring this entity since the beginning:
```



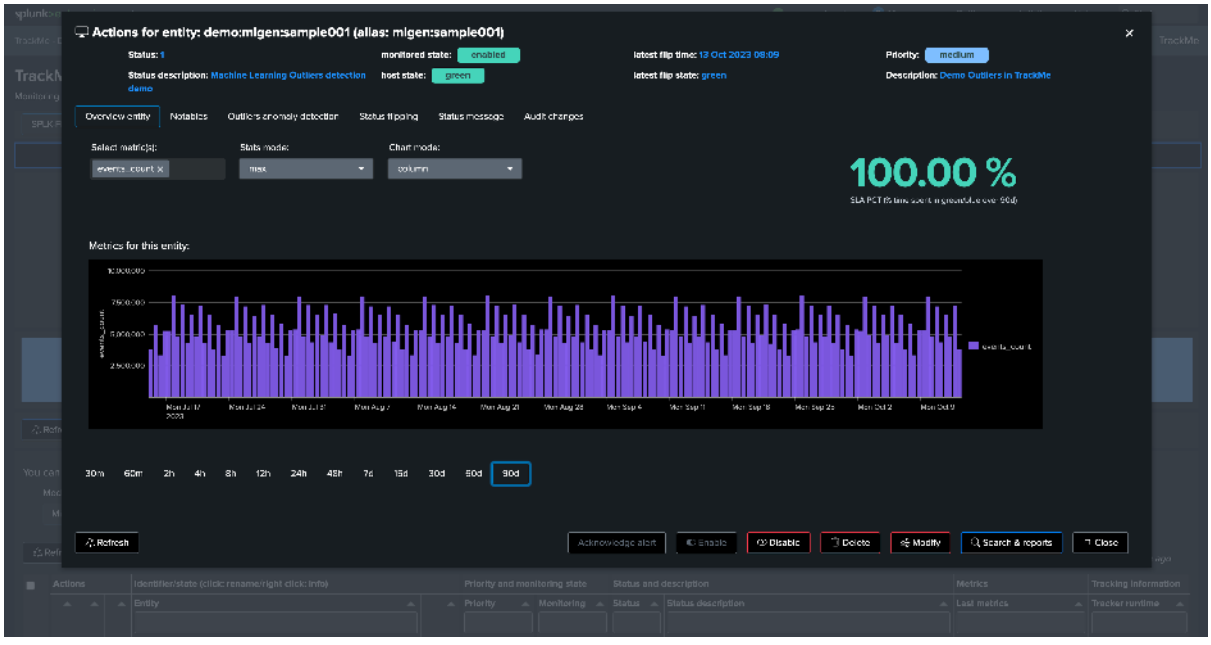
We use mcollect to force backfilling metrics, pay attention to replace with the valid tenant_id value:

```
index=mlgen ref_sample=* earliest=-90d latest=-5m
| bucket _time span=5m
| stats avg(dcount_hosts) as trackme.splk.flx.dcount_hosts, avg(events_count) as␣
↪trackme.splk.flx.events_count by _time, ref_sample

| eval alias=ref_sample
| lookup trackme_flx_tenant_demo-outliers alias OUTPUT tenant_id, _key as object_id,␣
↪object, object_category
| where isnotnull(object_id)

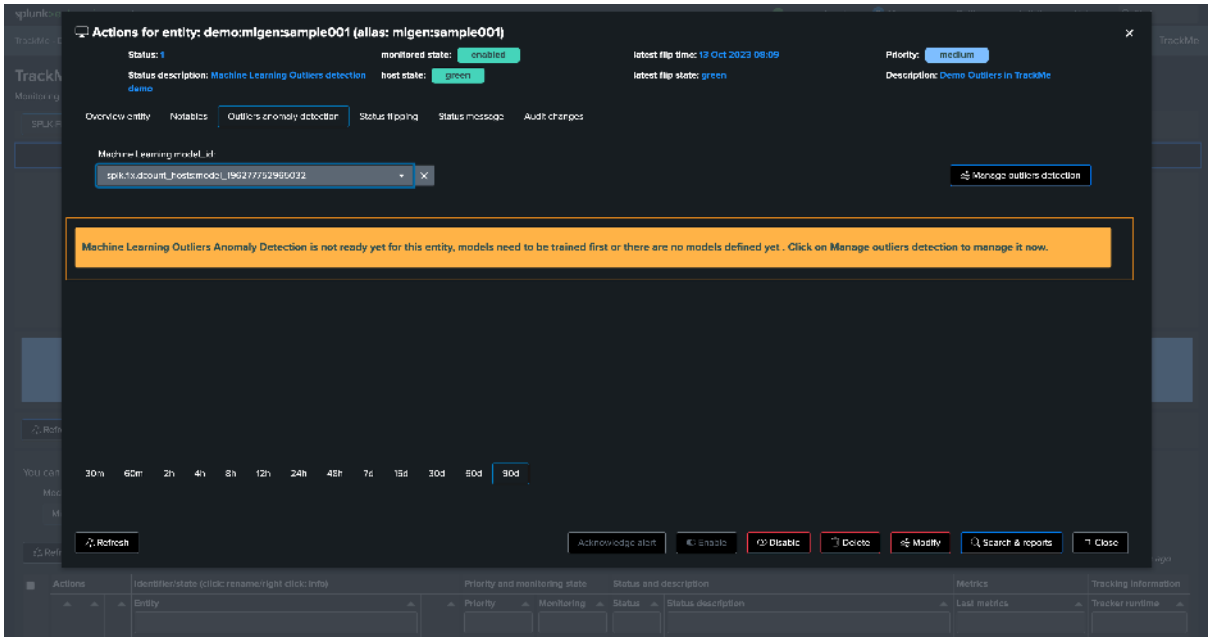
| mcollect index=trackme_metrics split=t object, object_category, object_id, tenant_id
```

Opening the entity shows we have backfilled metrics now:

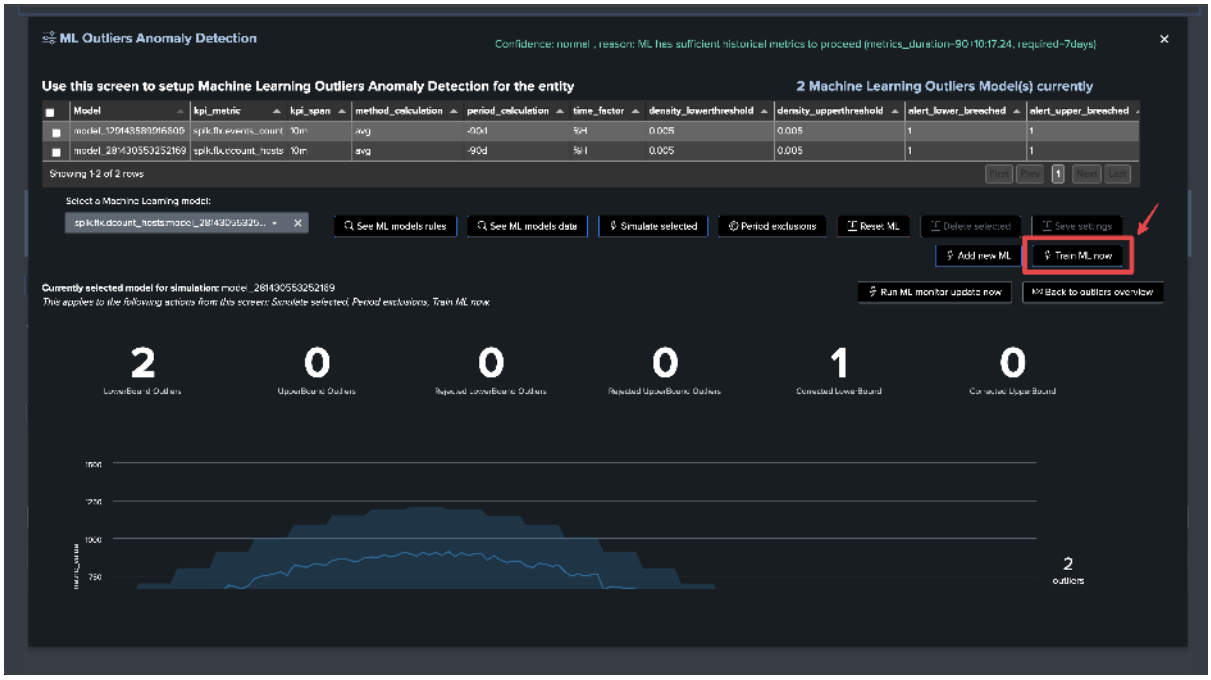


Depending on if TrackMe already ran or not the ML training job for the tenant, ML Outliers may not

be ready yet:

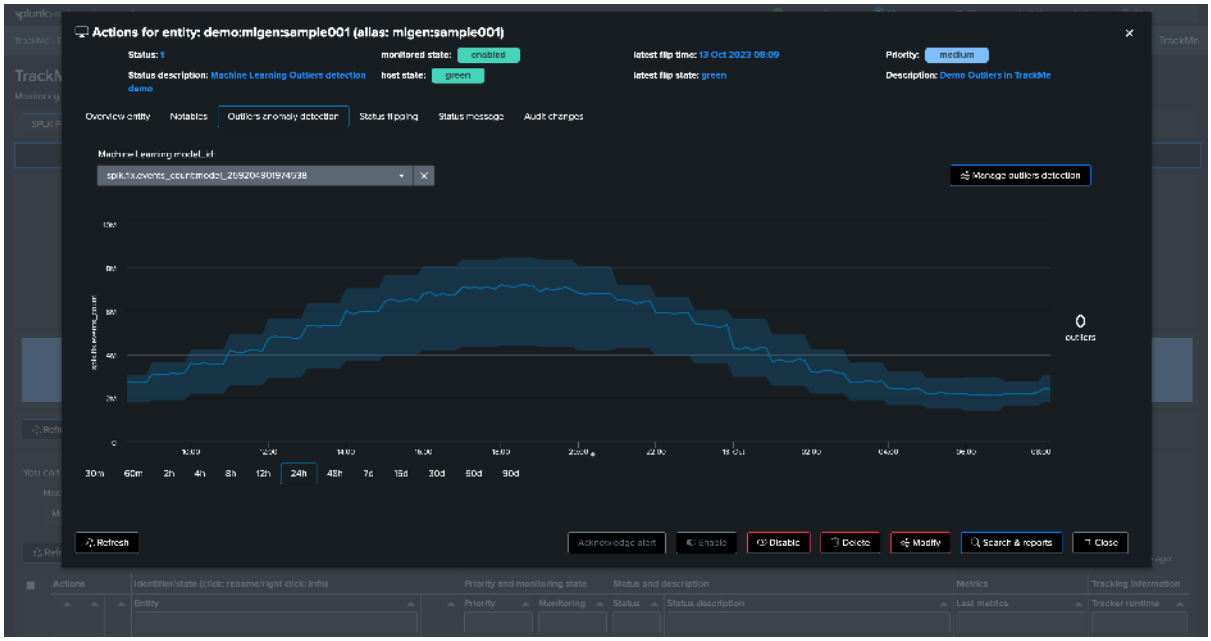


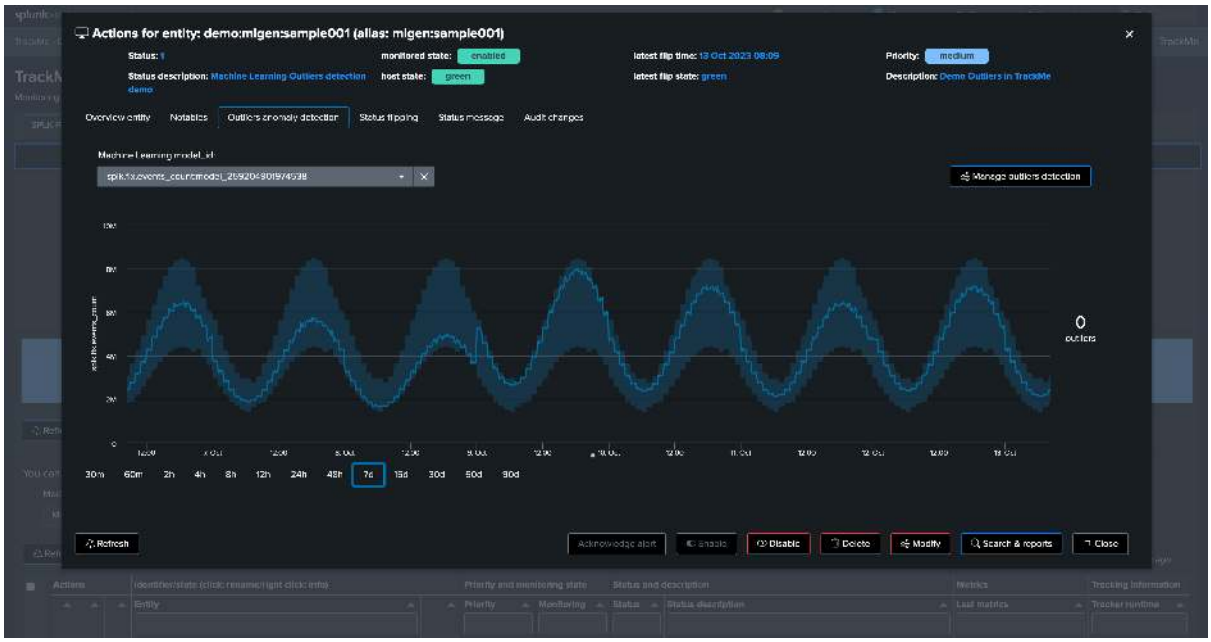
We can either run the mltrain job manually, or train the models via the UI:



If we refresh TrackMe, we can now see ML is ready:

- We have no outliers yet
- TrackMe applied defaults settings for the ML definition (training over the past 30 days, time per hour)

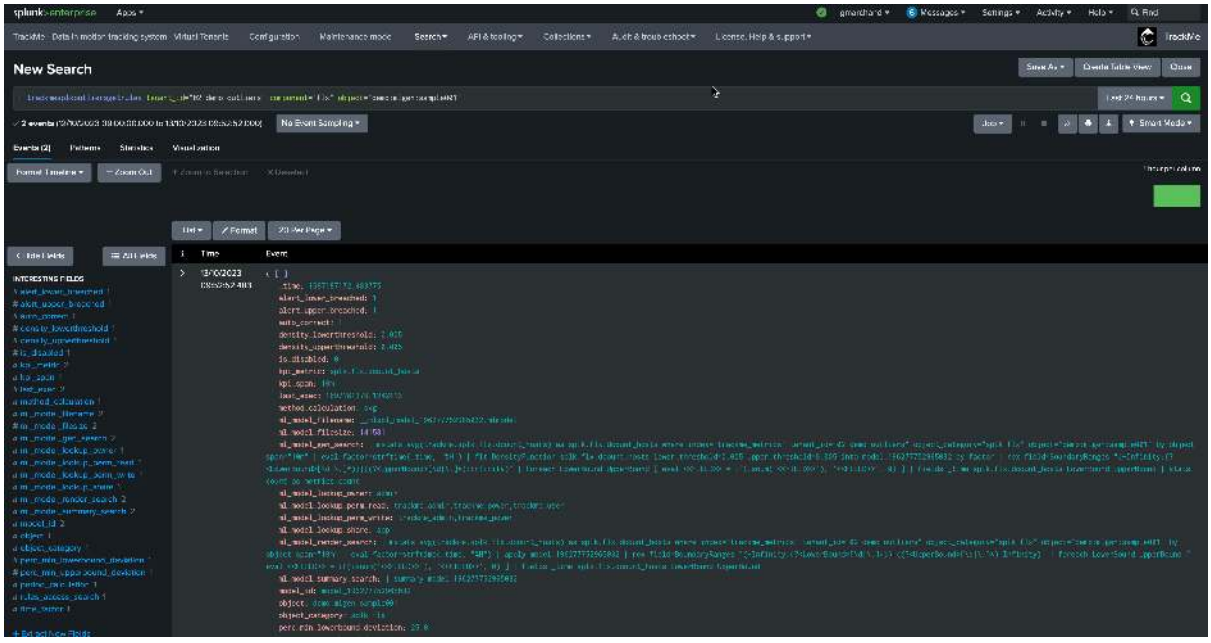




Let's access and review the models definition, for now we will only increase the training period to the past 90 days:

- Click on "Manage Outliers detection"
- Update the models to increase the time range for the calculation
- Manually run a training for each model
- Click on Simulate Selected to review the results (we have selected the event count model), this is looking great for now

You can click on See ML model details to access and review to the models information:



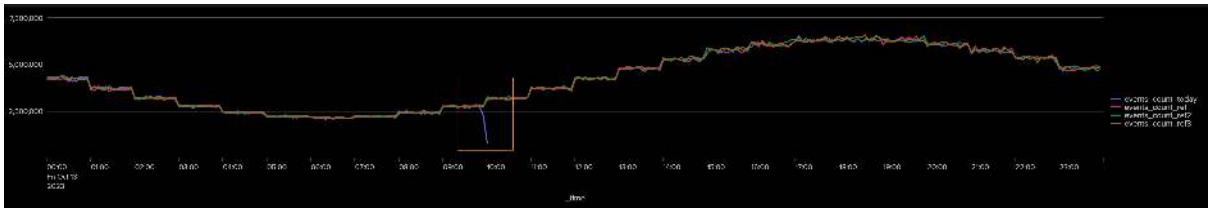
Scenario: Detecting a lower bound outliers

Although we know there is a weekday behavior in the data, for now we will stick with the default settings and we will start generating a lower bound outlier.

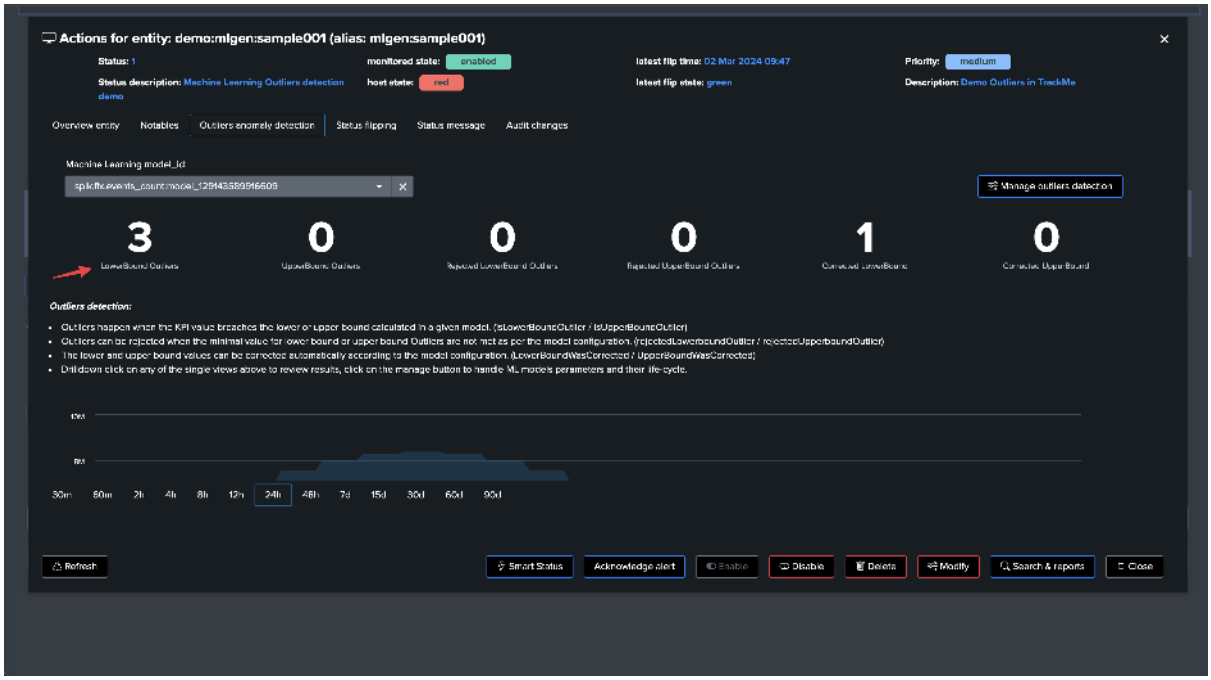
To achieve this, we stop the `run_backfill.sh` and we start `run_gen_lowerbound_outlier.sh`, this basically:

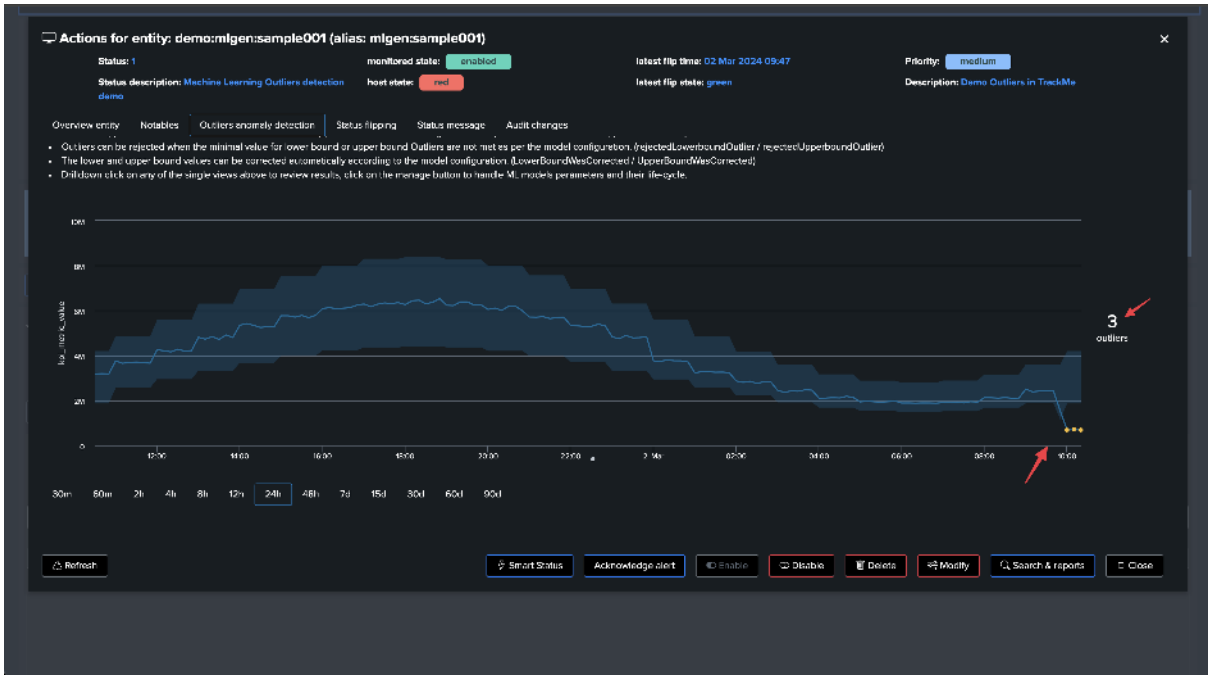
- Influence metrics with a large decrease of the curve, by an approximately 75%, accordingly to the magnitude of the week day / hour range

After a few minutes, we start to see a clear outlier using the previous days' comparison timechart search:

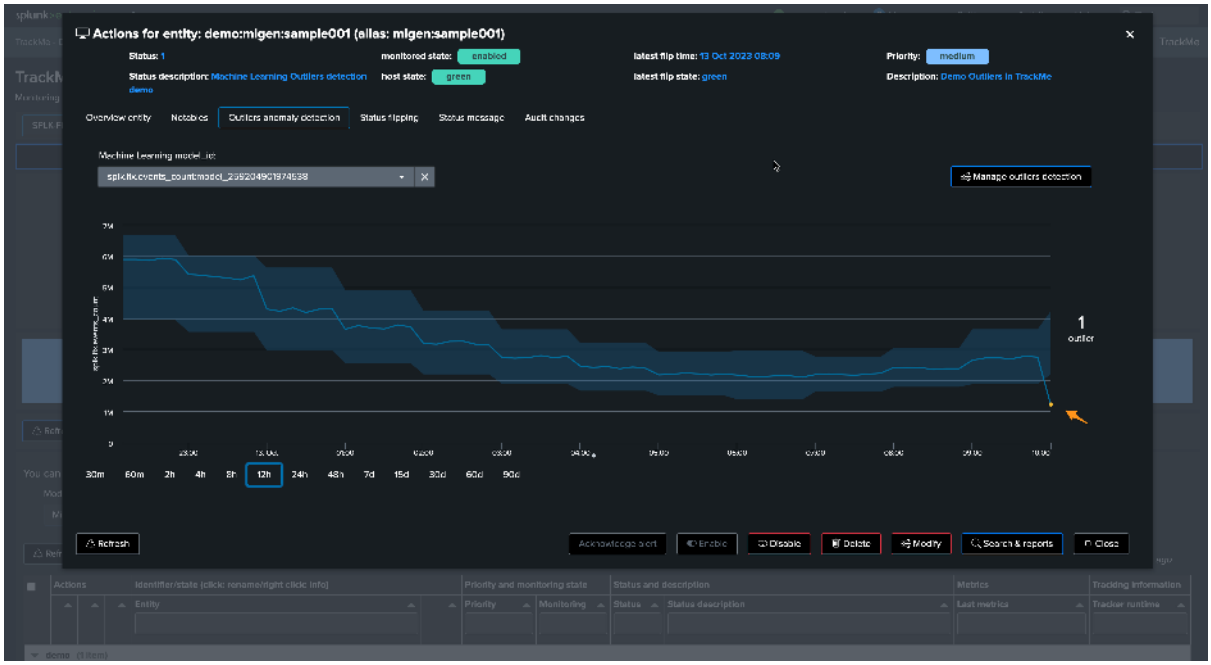


This outlier will also be reflected in TrackMe. It can take 5 to 10 minutes to be detected as an effective outlier:





Let's zoom in the period:

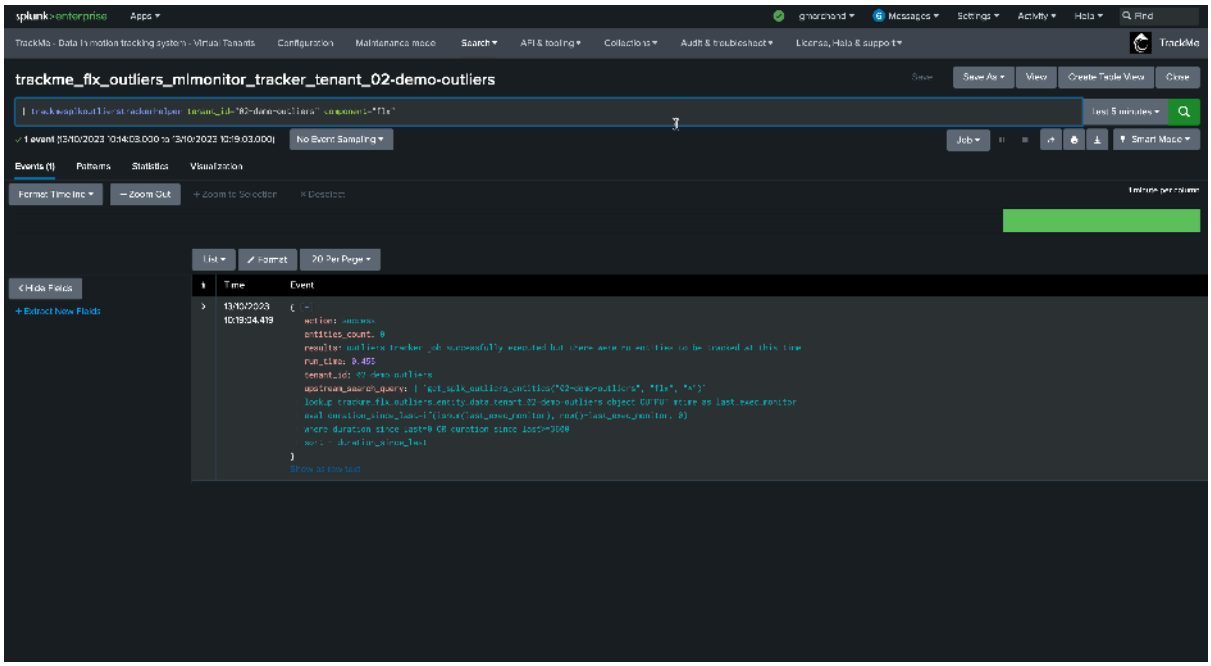


Excellent—the sudden decrease in activity has been detected successfully!

The next phase is to review the Machine Learning rendering phase. This means:

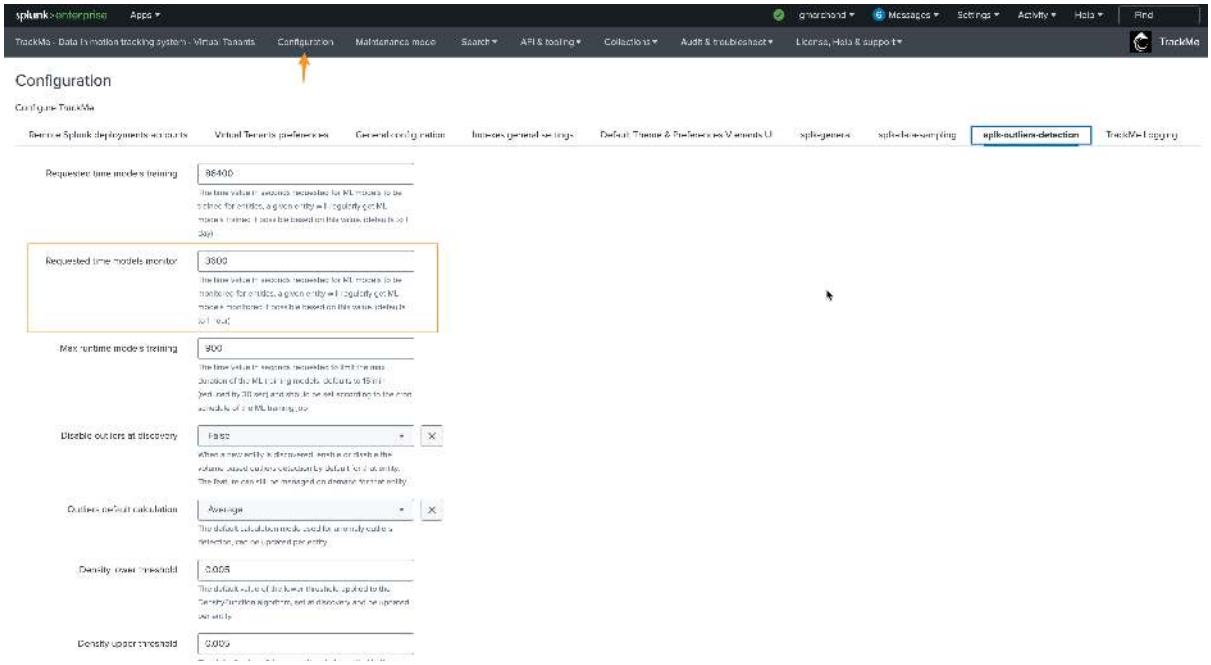
- The `_mlmonitor_` scheduled backend runs on a regular basis and attempts to review as many entities as possible in a given period of time
- Depending on the volume of entities, this process can require some time before the outlier is effectively noticed

Let's run the mlmonitor job:

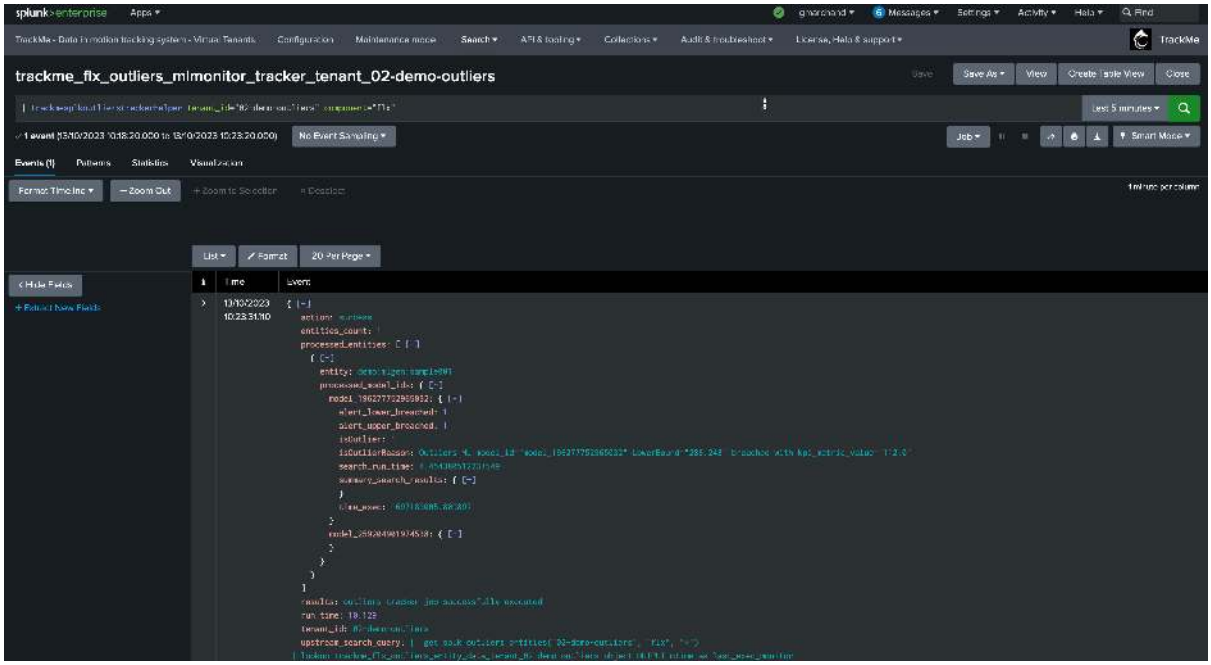


- By default, the ML monitor will attempt to verify any entity that has not been verified for more than an hour

This behavior can be customized in the system-wide configuration:



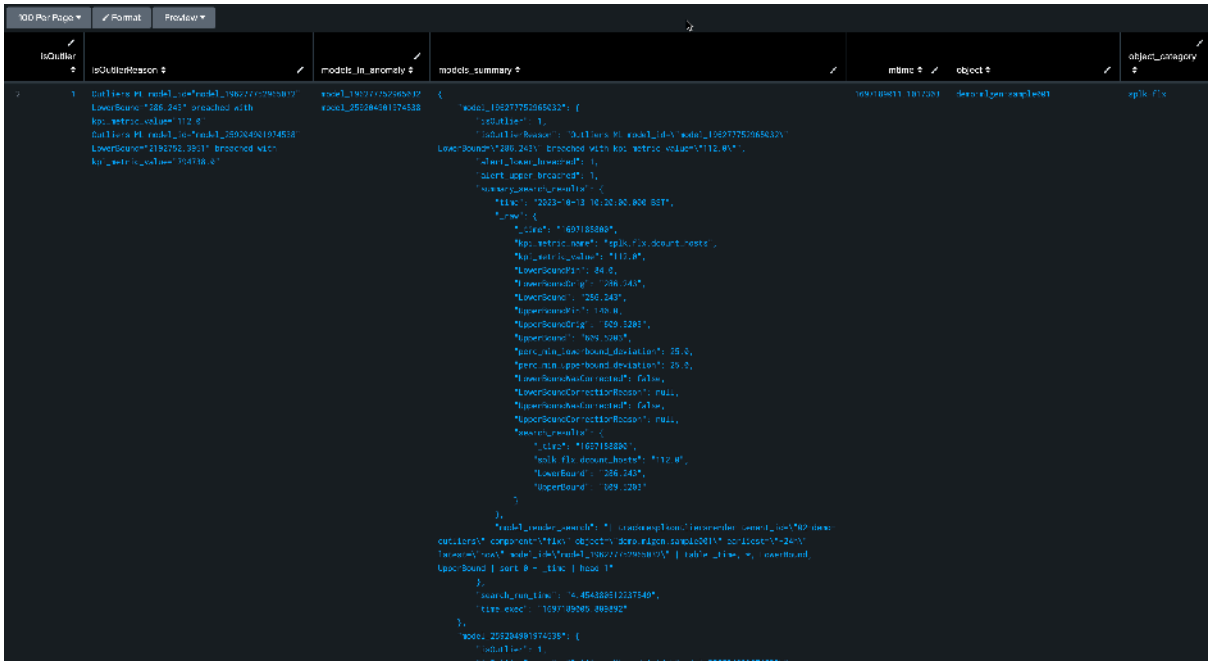
For the purposes of this documentation, we will reduce this to 5 minutes and re-run the monitor backend:



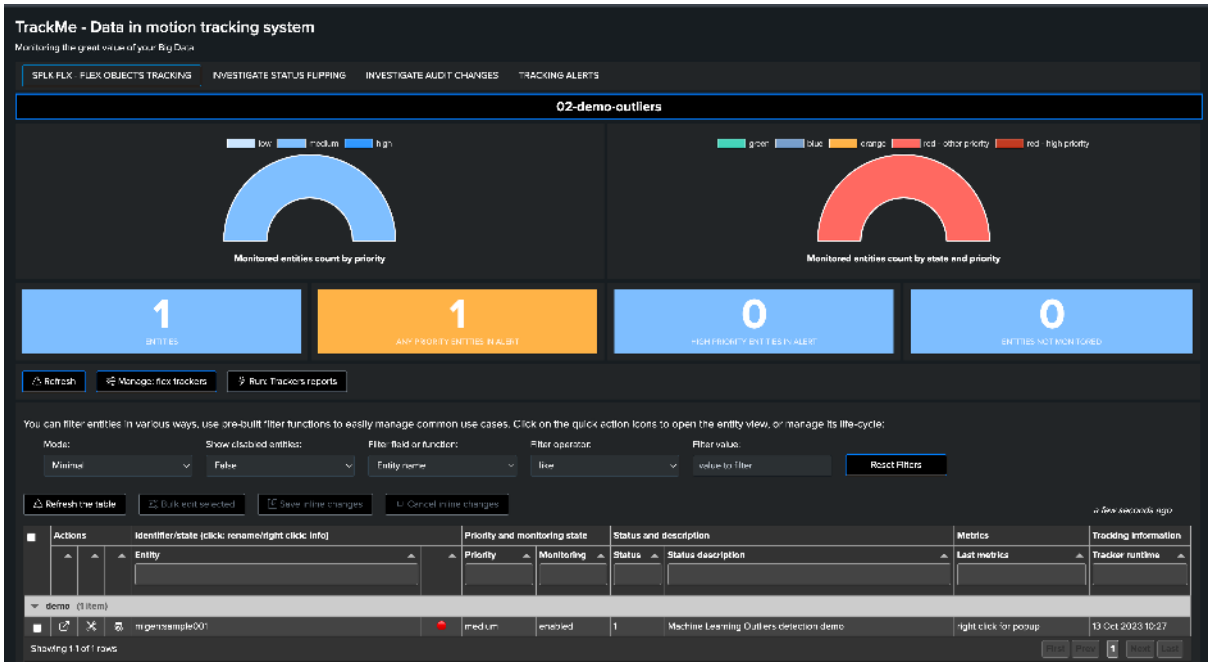
From this output, we already know that the outlier was detected. TrackMe stores the outlier results in a dedicated KVstore collection:

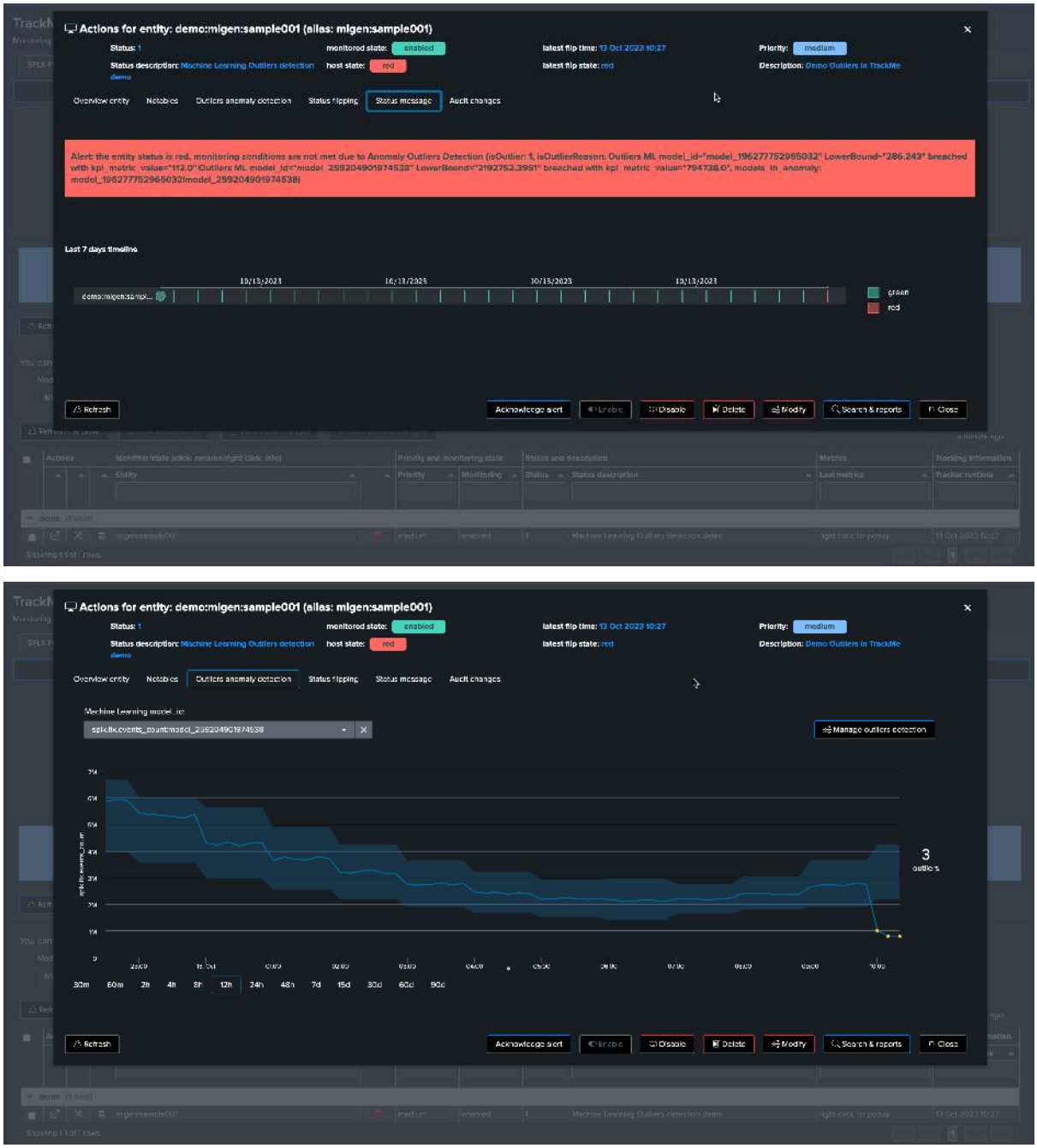
the name of the lookup transforms is trackme_<component>_outliers_entity_data_tenant_<tenant_id>

```
| inputlookup trackme_flx_outliers_entity_data_tenant_02-demo-outliers
```

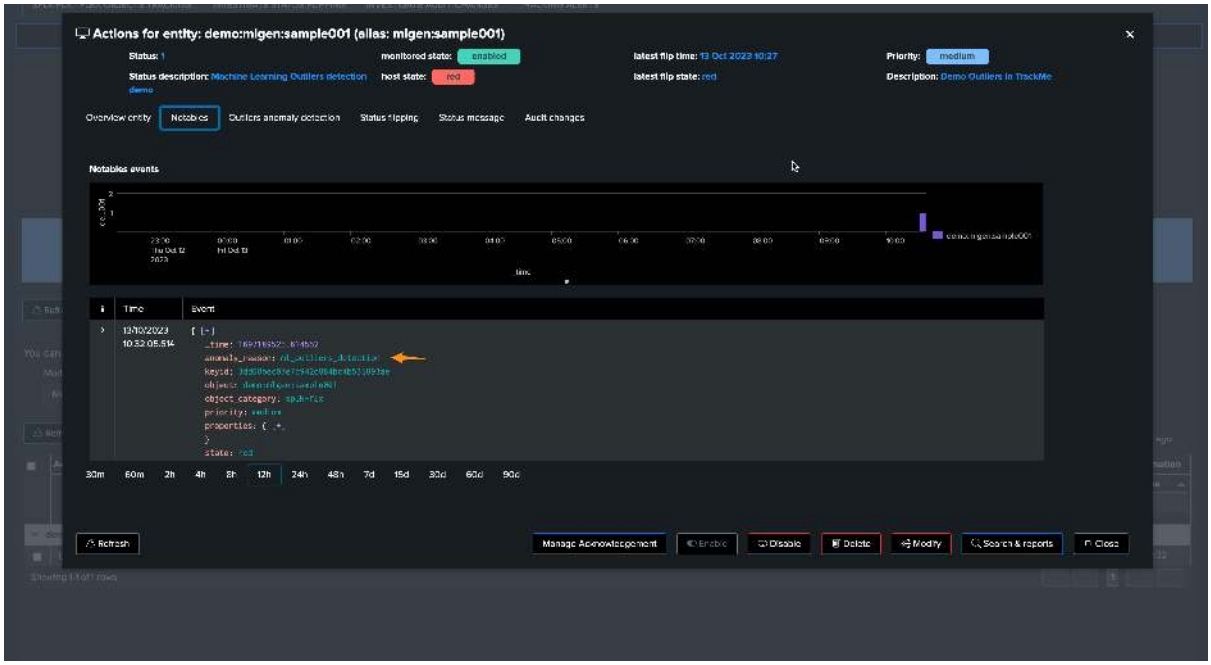


After a few minutes, the TrackMe tracker updates the metadata and the entity appears in red:





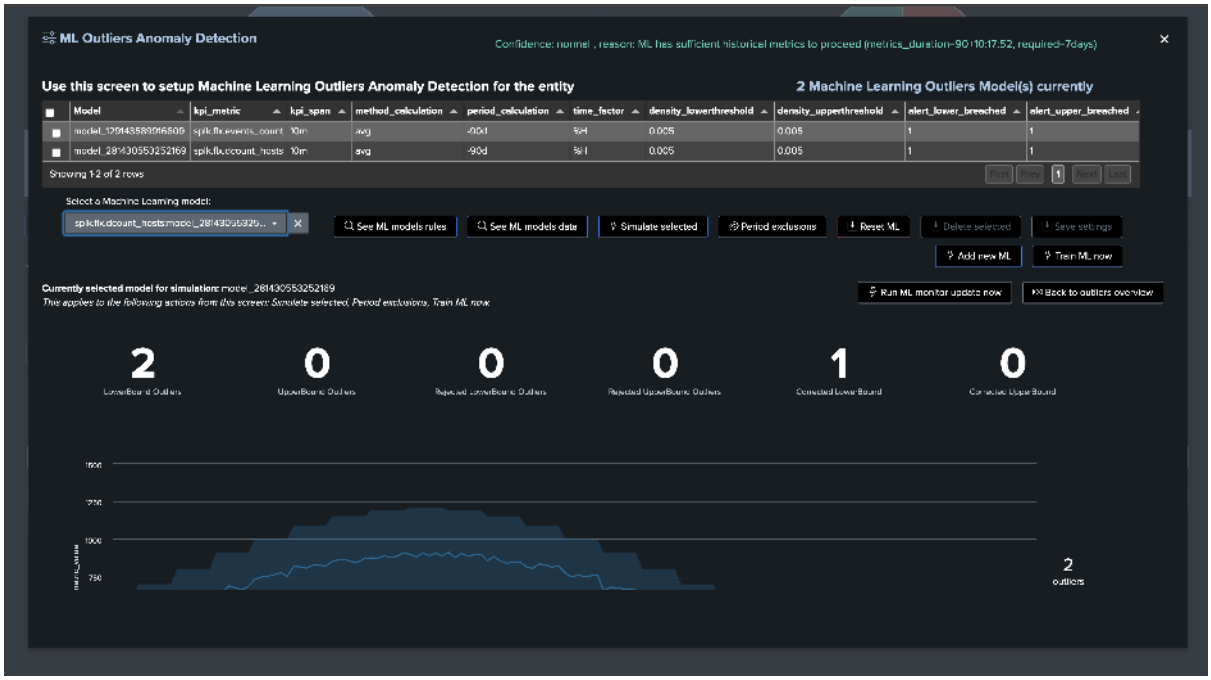
The job is complete, and we have successfully detected an abnormal change in behavior. The entity was impacted, and our alerting configuration would raise an alert accordingly!



Fine tuning the ML models

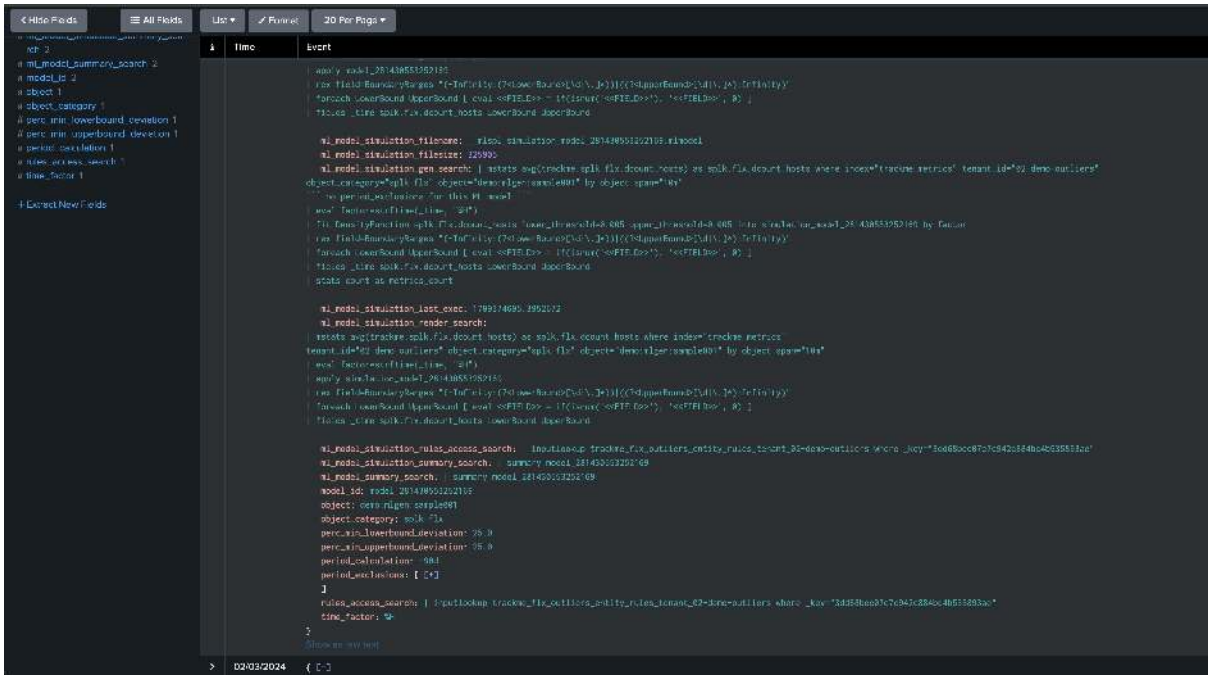
True context simulation since TrackMe 2.0.84

Since this TrackMe release, the principal Simulation screen runs simulations in `true context`, this means that TrackMe will train an ML model dedicated for that simulation, allowing the Outliers detection to 100% reflect the live Outliers detection behaviors.



Once a true context simulation has run, the simulation model details become visible in the entity rules: (click on the button `See ML models rules`)

```
| trackmesplkoutliersgetrules tenant_id="02-demo-outliers" component="flx" object=
↪"demo:mlgen:sample001"
```



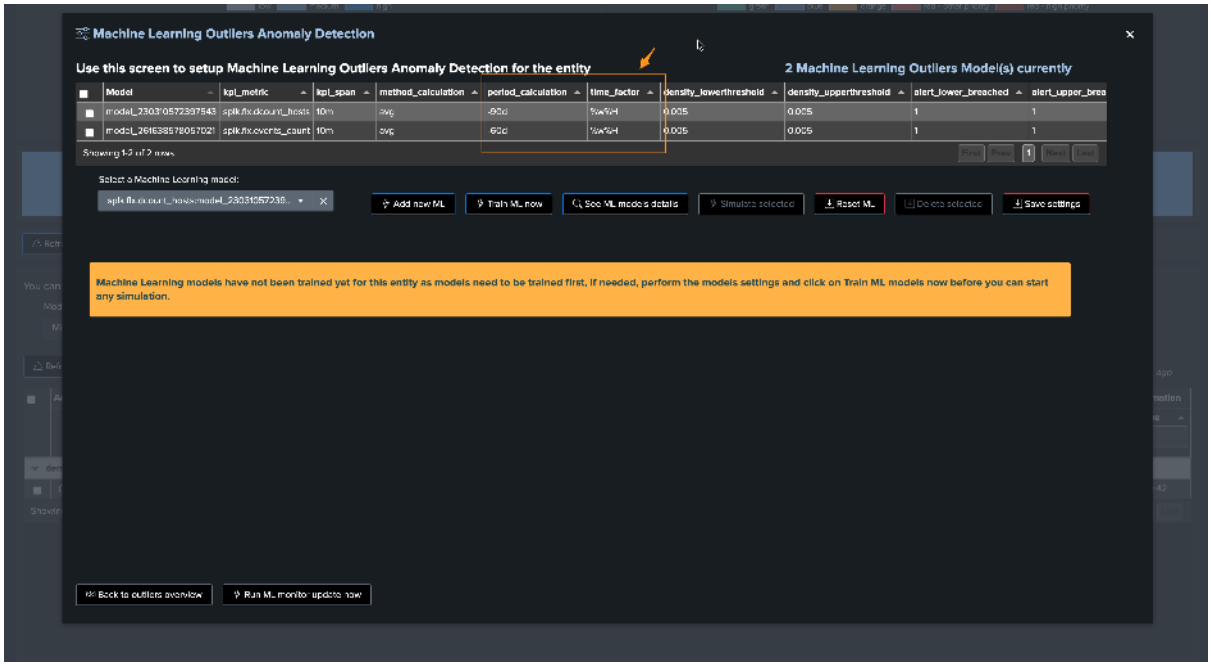
Fine tuning models

As we previously mentioned, our generated data has a concept of weekday behaviors, which we have not yet leveraged in the calculation.

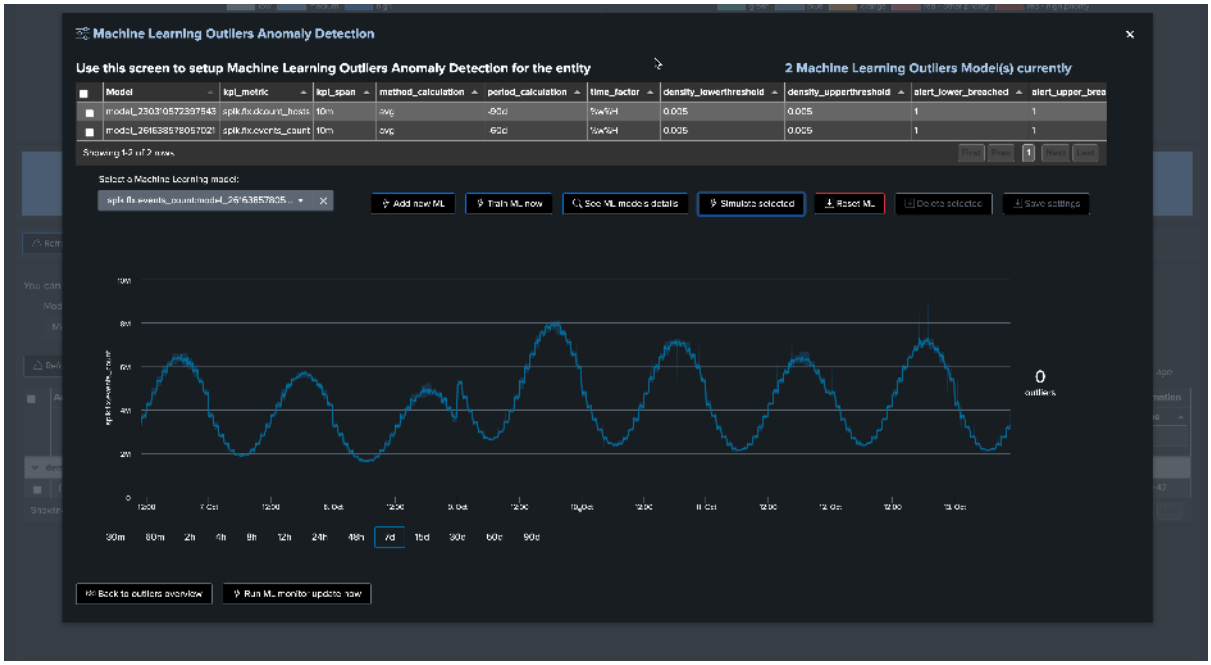
To demonstrate this behavior, we will stop our current data generation, update our sample entity name, and generate a new data set by running the script `run_backfill.sh` again.

- We process the same steps as previously to backfill the metrics using mcollect once the entity was discovered
- Then, we edit the models to increase the period, and this time will ask TrackMe to take into account the weekdays in the outliers calculation

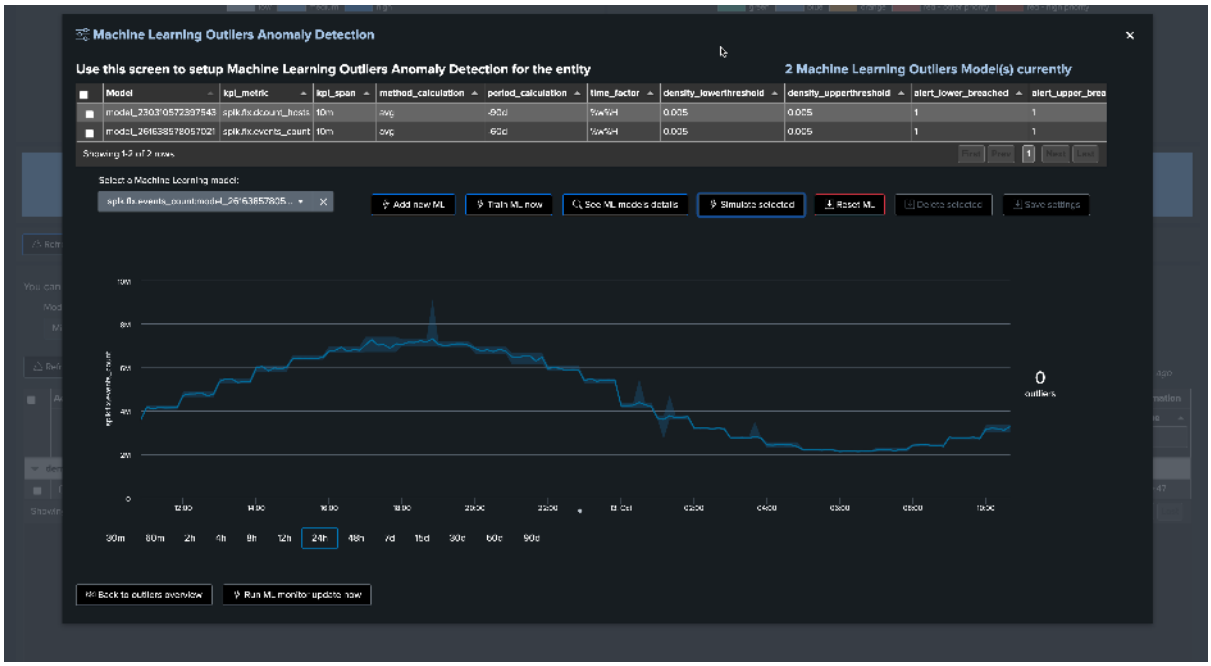
We can observe that the behavior is slightly different, with much closer lower bound and upper bound ranges. This is because our data is very stable and has shown enough consistency over time.



Focus last 24 hours:



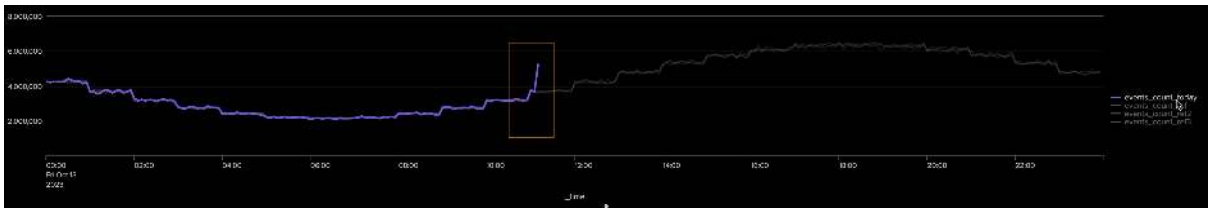
Note that this also means our outlier detection will be much more sensitive, which can lead to false positive alerts.



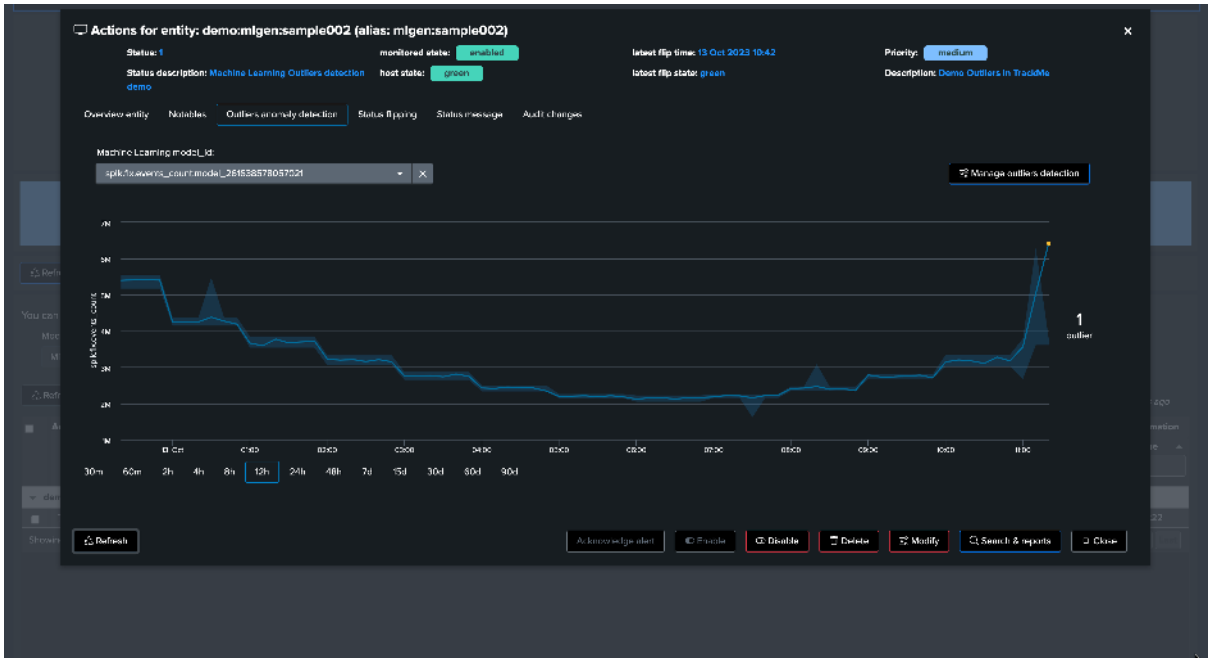
TrackMe implements a concept of "auto-correction" based on minimal variation in lower and upper dimensions, to avoid generating false positives:

This time, we will generate upper bound outliers. We stop the current script and start `run_gen_upper_outlier.sh`, which slightly increases the volume of our metrics.

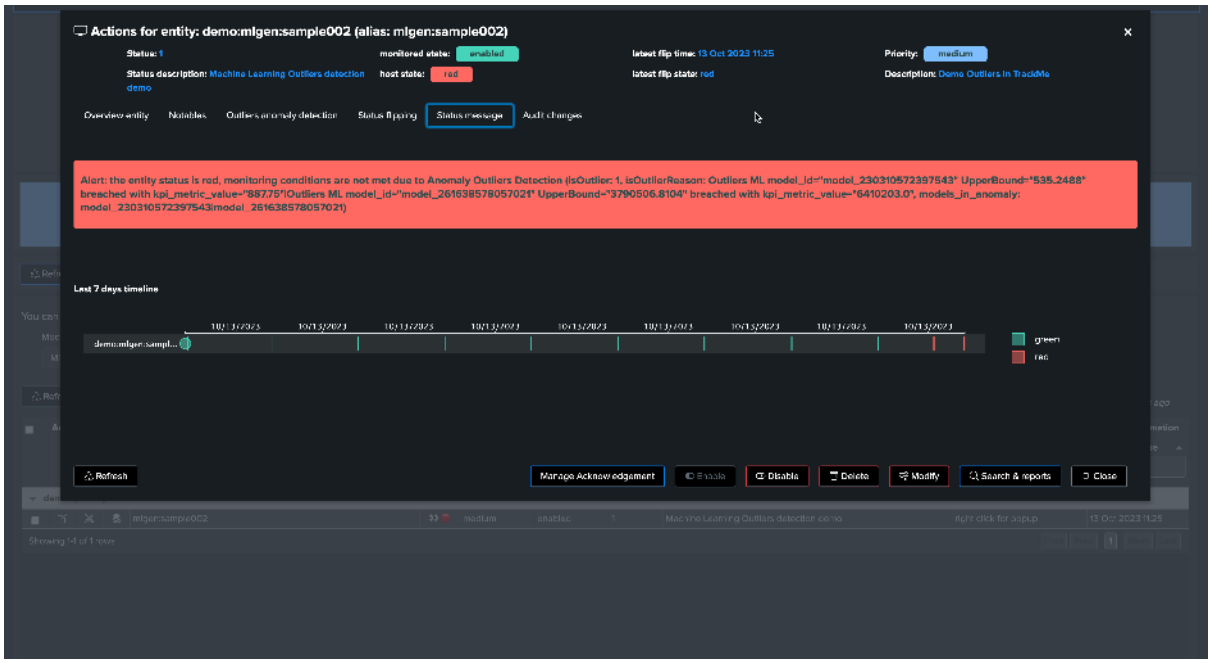
After a few minutes, we can observe the variation:

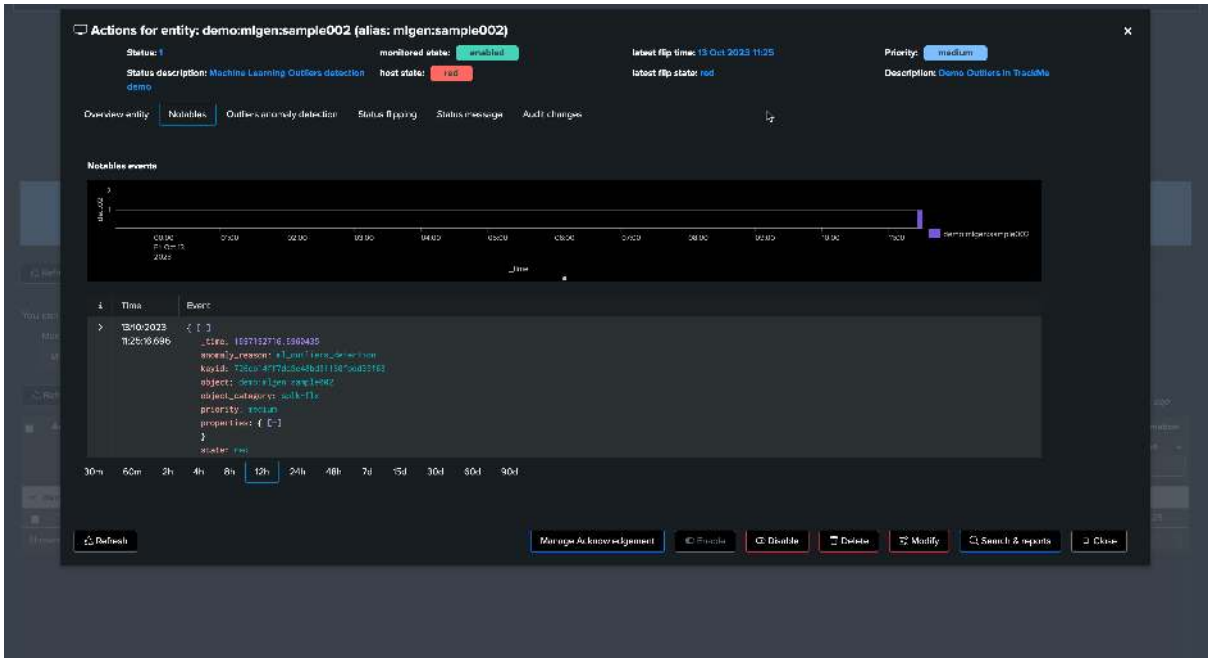


Shortly after, TrackMe notices the upper bound outlier:

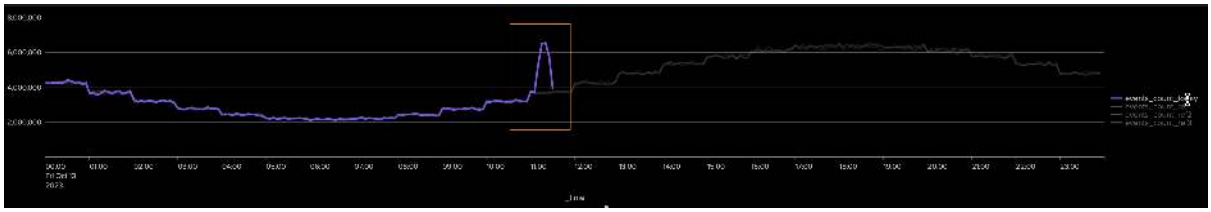


After a run of the ML monitor (which happens automatically), the upper bound condition is detected and the alert is raised accordingly.

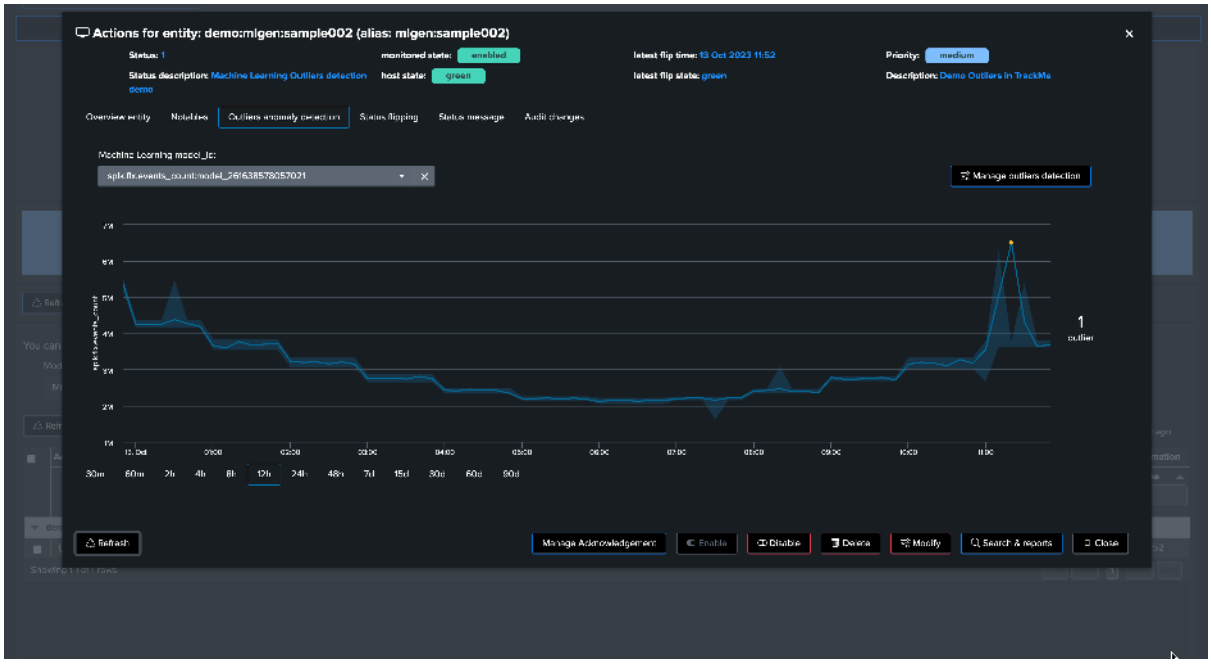


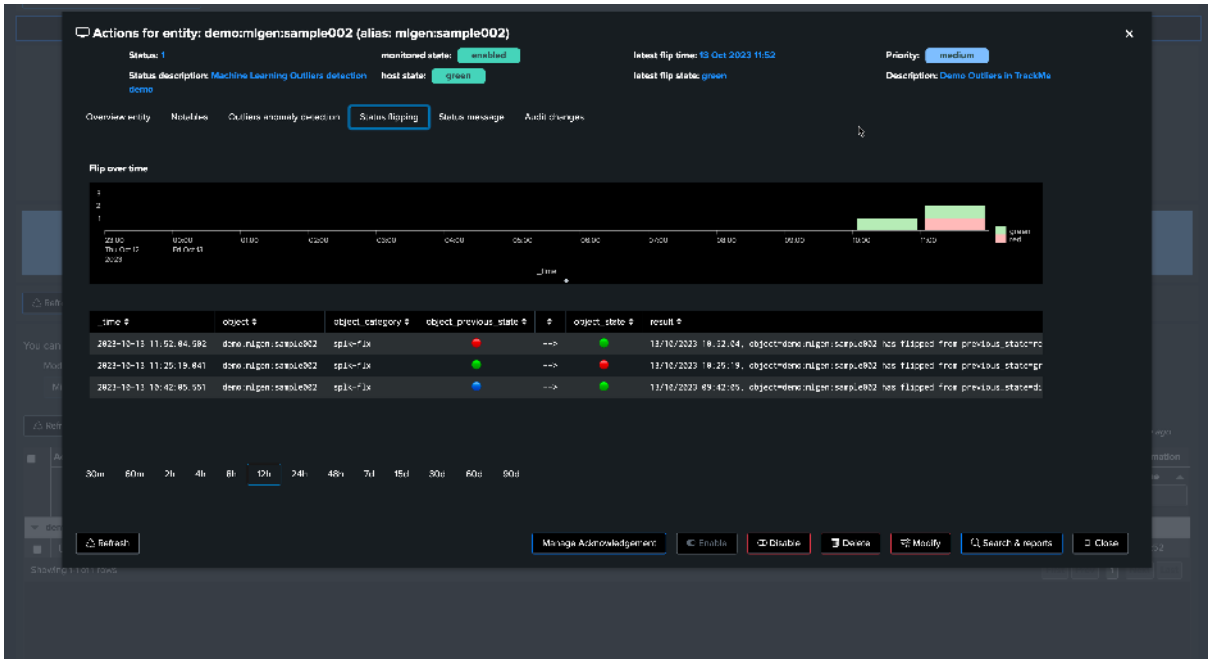


Now, let's assume the "storm" is over. We stop the upper bound outlier generation script and instead call `run_normal.sh`. After some time, outliers are no longer occurring. TrackMe will notice, and the entity will return to a green status.



Soon after, TrackMe sees the entity back to green:





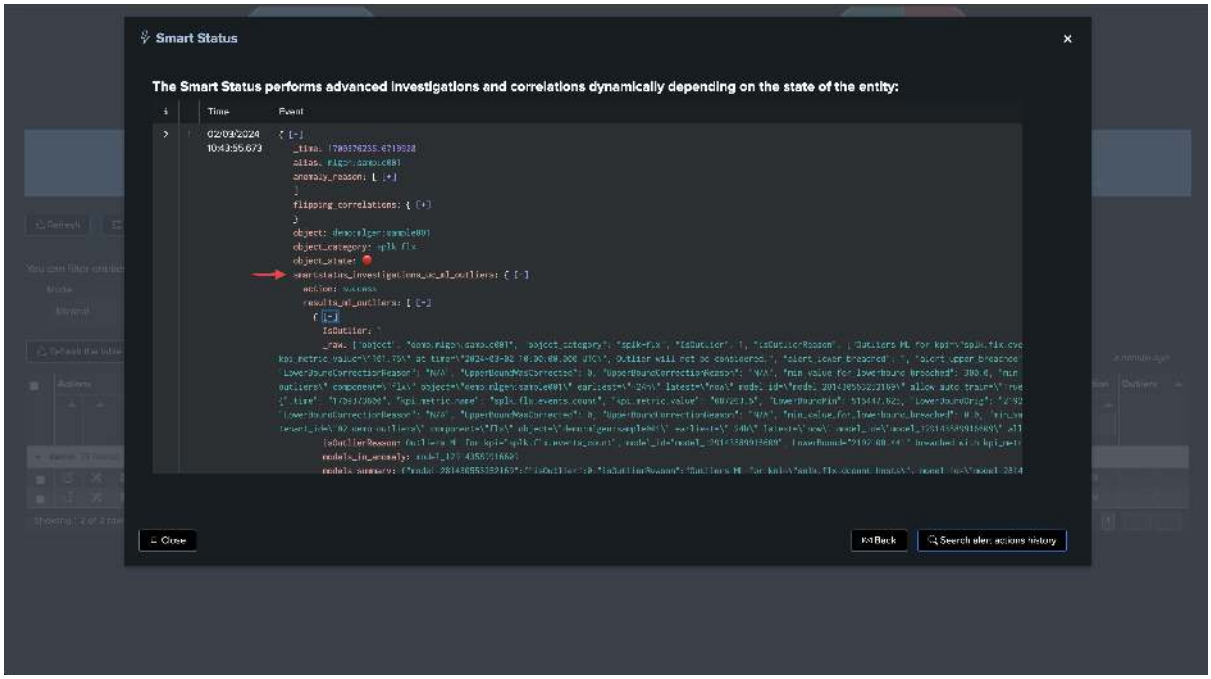
And the job is done!

Note that from TrackMe version 2.0.62, you can exclude the anomaly period from the ML model. This allows the ML model to learn from the data without the incident, making it more accurate in the future.

See: *ML period exception: excluding periods of time*

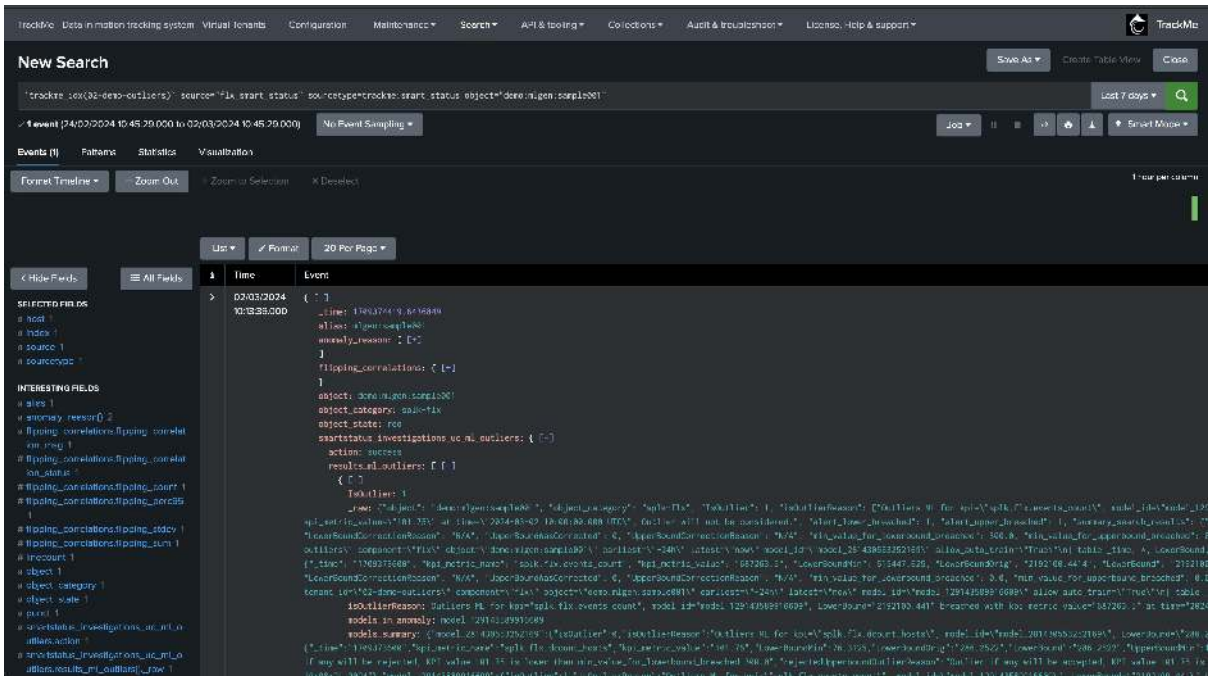
7.13.7 SmartStatus and Outliers

The SmartStatus is a TrackMe feature which automatically runs investigations when a given entity enters an alerting mode (red). When it comes to Outliers, the SmartStatus investigates automatically the Outliers condition:



SmartStatus is also an alert action which indexes its results in TrackMe's summary index, so you can review the actual condition when the alert came through:

```
`trackme_idx(02-demo-outliers)` source="flx_smart_status" sourcetype=trackme:smart_
↪status object="demo:mlgen:sample001"
```



7.13.8 Accessing the ML models

You can access the ML models rules using the following command:

```
| trackmesplkoutliersgetrules tenant_id="<tenant_id>" component="<component>" object="
↪<entity_name>"
```

TrackMe stores the ML models definition in a KVstore:

```
| inputlookup trackme_<component>_outliers_entity_rules_tenant_<tenant_id>
```

7.13.9 Accessing the ML models current results

You can access the ML models data (results) using the following command:

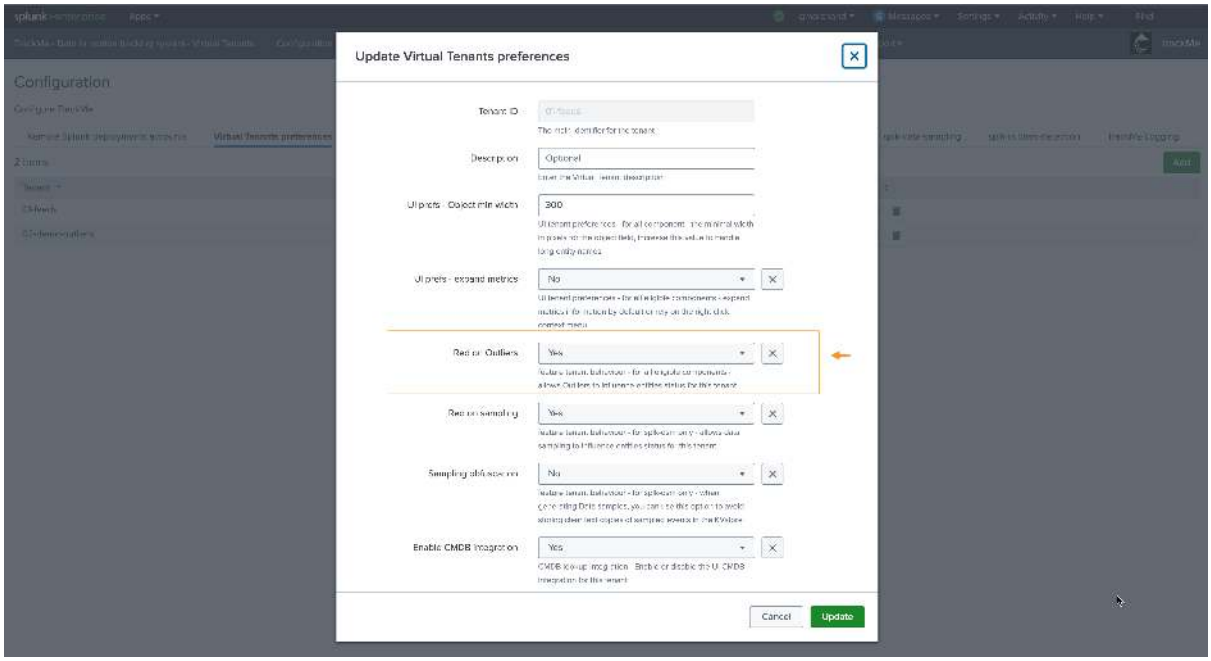
```
| trackmesplkoutliersgetdata tenant_id="<tenant_id>" component="<component>" object="
↪<entity_name>"
```

TrackMe stores the ML models current results in a KVstore:

```
| inputlookup trackme_<component>_outliers_entity_data_tenant_<tenant_id>
```

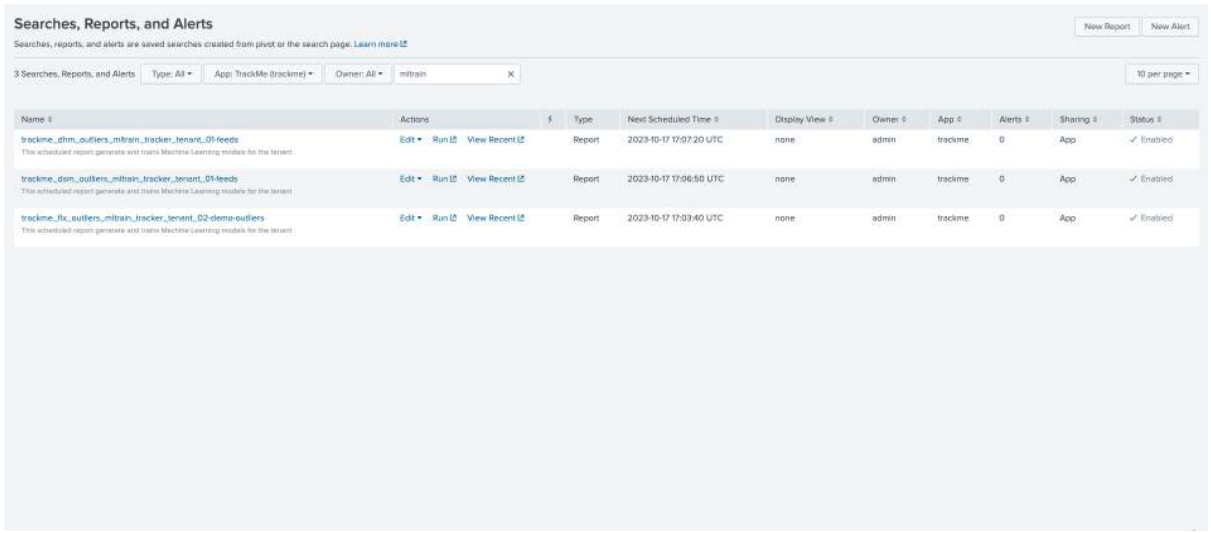
7.13.10 Disabling alerting on Outliers

You can very simply disable alerting on Outliers on a per Virtual Tenant basis, access to the Configuration UI and the Virtual Tenant account:



7.13.11 ML training scheduled jobs

When creating a Virtual Tenant, TrackMe creates a scheduled job called `_ml_train_`, this job is responsible for training the ML models for the entities of the tenant and a given component.



The scheduled ML training job behaves as follow:

- The job is scheduled to run every once per hour
- It runs for a certain amount of time which is driven by an implicit argument `max_runtime` to the command `trackmesplkoutlierstrainhelper` (defaults to 60 minutes minus a margin)

```
max_runtime_sec = Option(
    doc="""
    **Syntax:** **max_runtime_sec=****
    **Description:** The max runtime for the job in seconds, defaults to 60 minutes␣
↪less 120 seconds of margin.""",
    require=False,
    default="3600",
    validate=validators.Match("max_runtime_sec", r"^\d*$"),
)
```

- The job is influenced by system-wide options; see: *ML Outliers system wide options*
- The job runs a first Splunk search to discover entities to be trained
- The jobs log its activity as follows:

```
index=_internal sourcetype=trackme:custom_commands:trackmesplkoutlierstrainhelper
```

For instance:


```
2023-10-17 16:07:02,992 INFO trackmesplkoutlierstrainhelper.py generate 481 {
"tenant_id": "01-feeds",
"action": "success",
"results": "outliers models training job successfully executed",
"run_time": 11.839,
"entities_count": 12,
"processed_entities": [
   {
      "object_category": "splk-dsm",
      "object": "webserver:apache:access:json",
      "search": "| trackmesplkoutlierstrain tenant_id=\"01-feeds\" component=\"dsm\"␣
↪object=\"webserver:apache:access:json\"",
      "runtime": "0.5111777782440186"
   },
   {
      "object_category": "splk-dsm",
      "object": "webserver:nginx:plus:kv",
      "search": "| trackmesplkoutlierstrain tenant_id=\"01-feeds\" component=\"dsm\"␣
↪object=\"webserver:nginx:plus:kv\"",
      "runtime": "0.5510256290435791"
   },
   <redacted>
],
"failures_entities": [],
"search_errors_count": 0,
"upstream_search_query": "| inputlookup trackme_dsm_outliers_entity_rules_tenant_01-
↪feeds where object_category=\"splk-dsm\"\n | `trackme_exclude_badentities`\n |␣
↪lookup local=t trackme_dsm_tenant_01-feeds object OUTPUT monitored_state\n | where␣
↪monitored_state=\"enabled\"\n | eval duration_since_last=if(last_exec!=\"pending\",␣
↪now()-last_exec, 0)\n | where duration_since_last=0 OR duration_since_last>=600\n |␣
↪sort - duration_since_last"
}
```


In some cases and if you wish to reduce the number of changes performed by TrackMe, especially in a co-located SHC context (which we would recommend), you can update the scheduling plan and reduce its frequency.

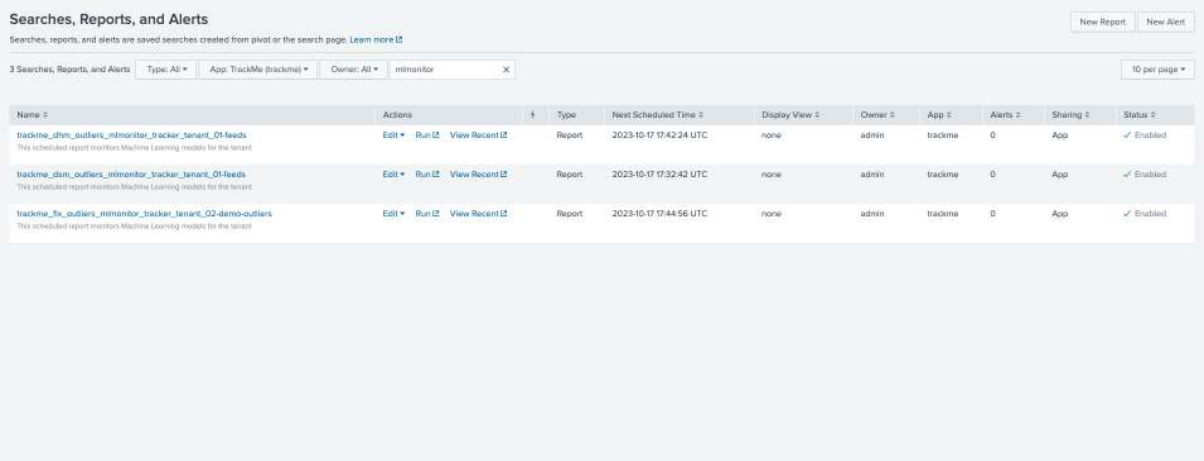
After having loaded the list of entities to be trained, the ML training backend attempts to sequentially train as many entities as possible in the allowed max run time.

- Each search is a highly efficient search relying on TrackMe metrics (mstats search)
- Searches call the `MLTK apply` command to load the previously trained models
- Searches are driven by a TrackMe command called `trackmesplkoutlierstrain`
- logs are available here:

```
index=_internal sourcetype=trackme:custom_commands:trackmesplkoutlierstrain
```

7.13.12 ML monitor scheduled jobs

When creating a Virtual Tenant, TrackMe creates an `_mlmonitor_` scheduled job, this job is responsible for monitoring the ML models for the entities of the tenant and a given component.



The scheduled ML monitor job behaves as follow:

- The job is scheduled to run every 20 minutes
- It runs for a maximum of 15 minutes (to avoid generating skipping searches), influenced by an implicit argument to the TrackMe command:

```
max_runtime = Option(
    doc="""
    **Syntax:** **max_runtime=****
    **Description:** Optional, The max value in seconds for the total runtime of the␣
↪job, defaults to 900 (15 min) which is subtracted by 120 sec of margin. Once the␣
↪job reaches this, it gets terminated""",
    require=False,
    default="900",
    validate=validators.Match("object", r"^\d*$"),
)
```

- The job is influenced by system-wide options; see: *ML Outliers system wide options*
- The job runs a first Splunk search to discover entities to be rendered
- The jobs log its activity as follows:

```
index=_internal sourcetype=trackme:custom_commands:trackmesplkoutlierstrackerhelper
```

- The job processes sequentially entities to be rendered and runs a highly efficient search to render the ML models using mstats, orchestrated by the TrackMe command `trackmesplkoutliersrender`
- The command automatically updates the records in the outliers data KVstore collection

```
| inputlookup trackme_<component>_outliers_entity_data_tenant_<tenant_id>
```

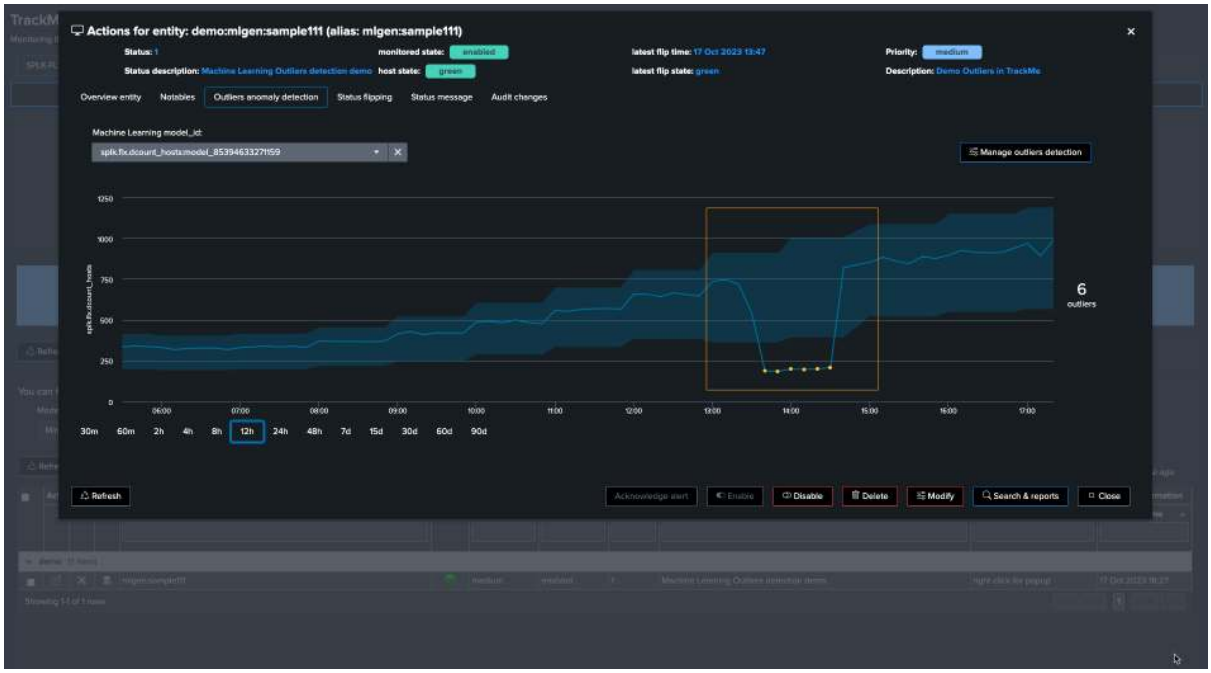
7.13.13 ML period exception: excluding periods of time

From TrackMe version 2.0.62, you can exclude one or more periods of time on an ML model basis:

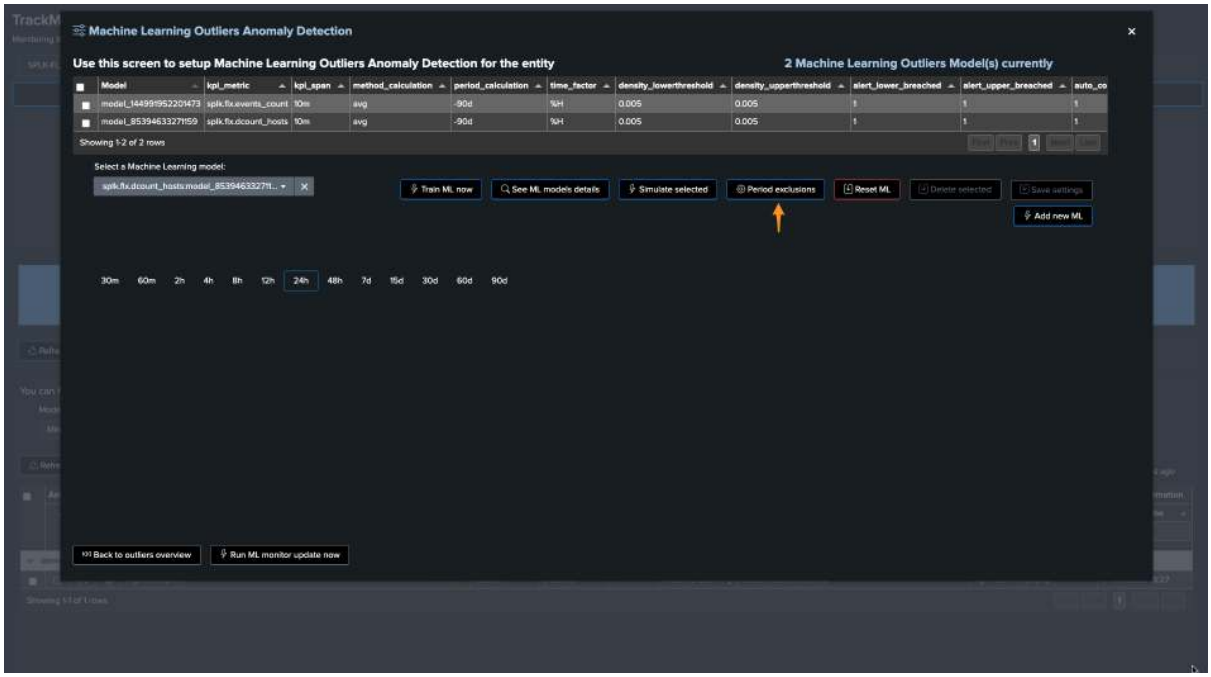
- If an incident occurs, you can exclude the period of time where the incident happened
- Doing so will allow the ML model to learn from the data without the incident, and will therefore be more accurate in the future

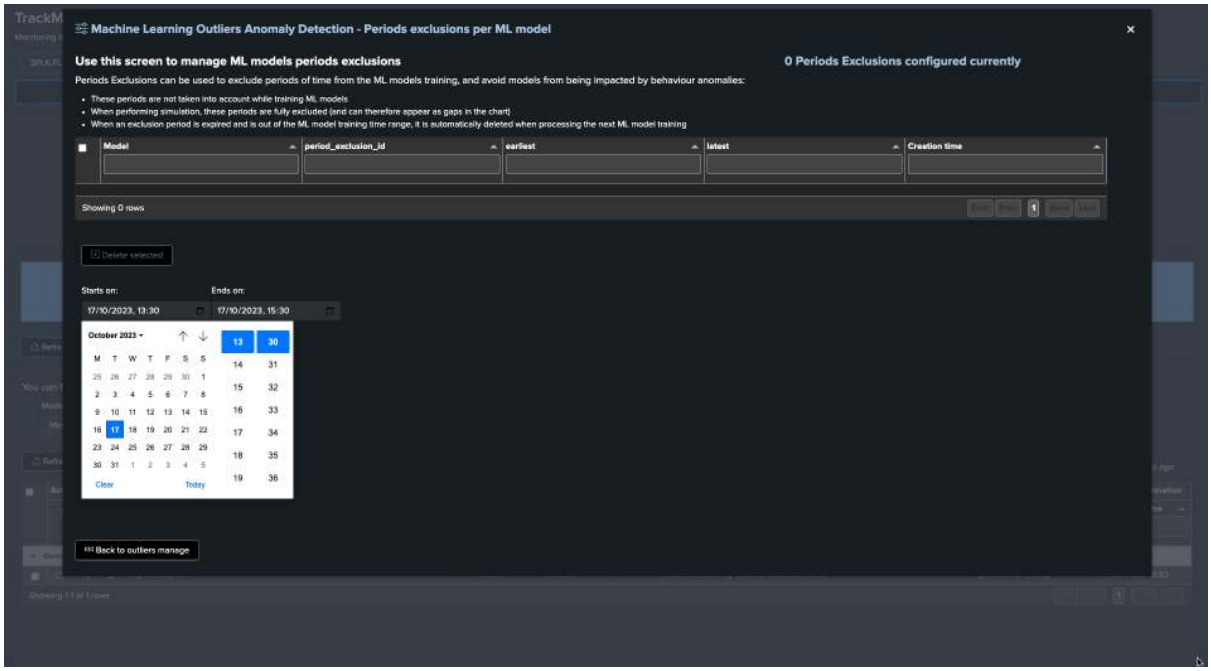
- When the exclusion period is expired because the latest time of the period is now out of the exclusion period, this period is deleted automatically from the ML model during the next ML training phase

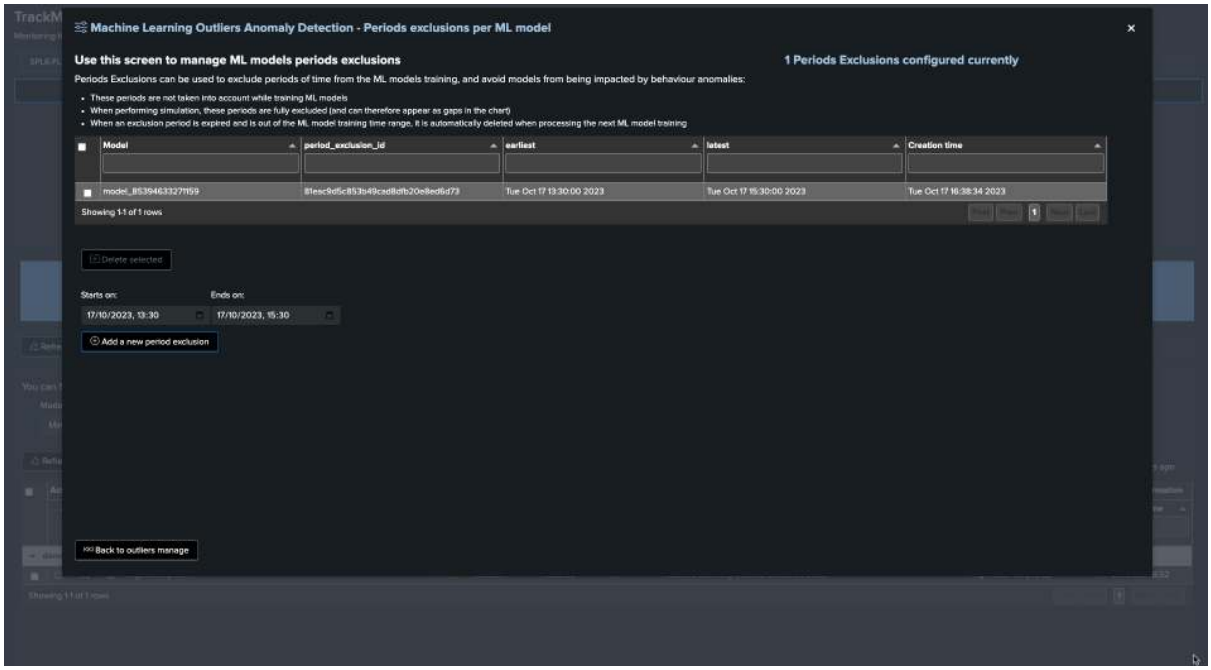
For instance, the following entity was impacted by abnormal behavior, we can exclude the period of time where the incident happened:



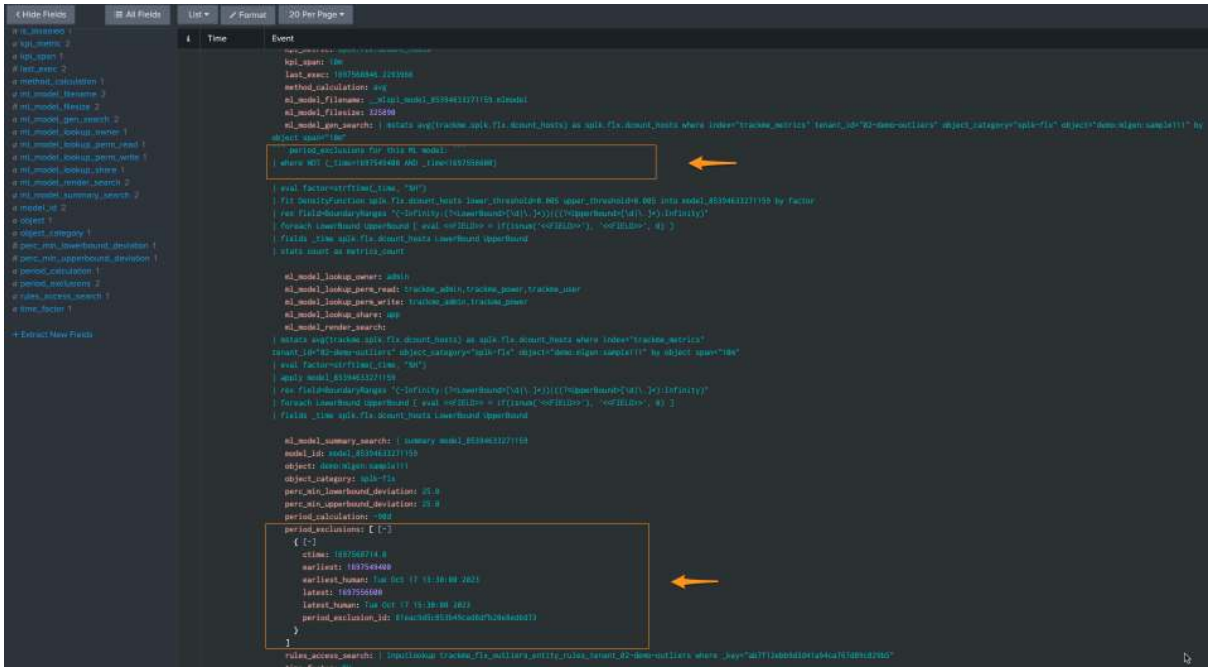
To exclude this period, click on "Manage Outliers detection" and then on "Period Exclusions". Note that exclusions apply per ML model:







When the next ML training happens for this entity, accessing the ML models details will show the exclusion period: (click on See ML model details)



Finally, when the excluded period is out of the time range scope of the ML training, for instance if the ML is trained for the past 30 days and the exclusion period is beyond that, the exclusion period is deleted automatically:

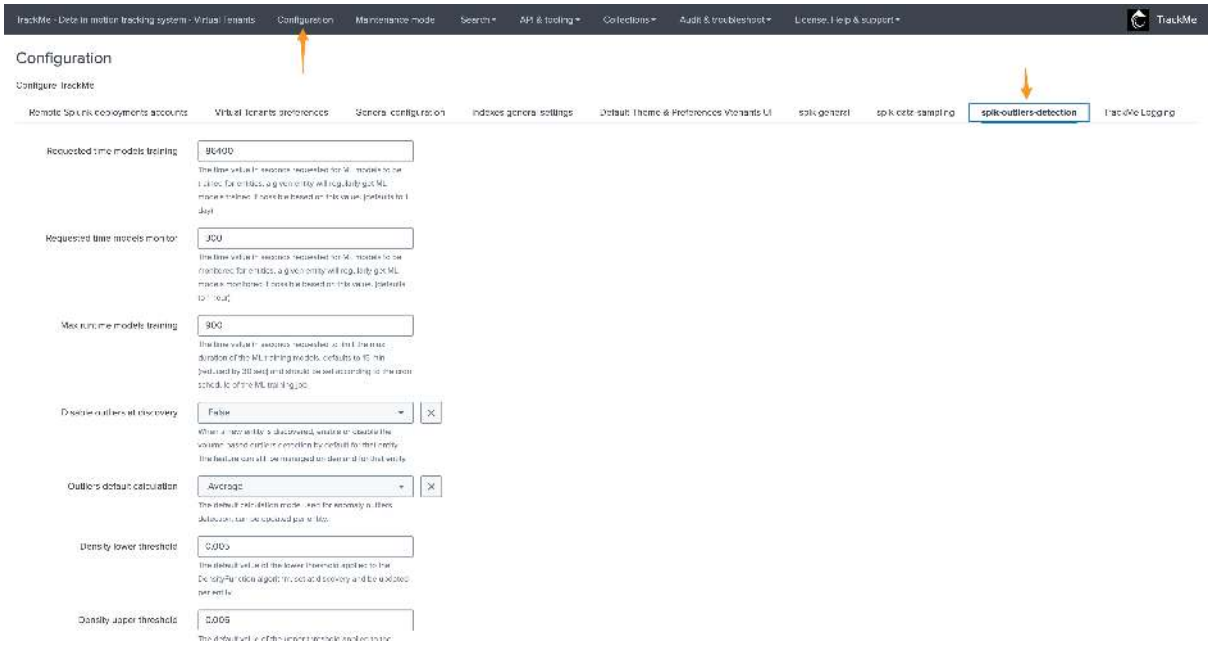
```
index=_internal sourcetype="trackme:custom_commands:trackmesplkoutlierstrain" "period␣
↪exclusion"
```

example:

```
2023-10-17 16:46:15,044 INFO trackmesplkoutlierstrain.py generate 486 tenant_id=02-
↪demo-outliers, object=demo:mlgen:sample111, model_id=model_144991952201473␣
↪rejecting period exclusion as it is now out of the model period calculation: {
   "period_exclusion_id": "180ea7b1a6ec737c823cfc38a6cd9414",
   "earliest": 1682947800,
   "earliest_human": "Mon May  1 13:30:00 2023",
   "latest": 1683041400,
   "latest_human": "Tue May  2 15:30:00 2023",
   "ctime": "1697561162.0"
}
```

7.13.14 ML Outliers system wide options

The following options are applied globally; these influence the Outliers detection behaviours and/or ML models definition: (when entities are discovered or ML is being reset)


Configuration
Configure TrackMe
spk-outliers-detection
Requested time models training
Requested time models monitor
Max runtime models training
Disable outliers at discovery
False
Outlier's default calculation
Average
Density lower threshold
Density upper threshold

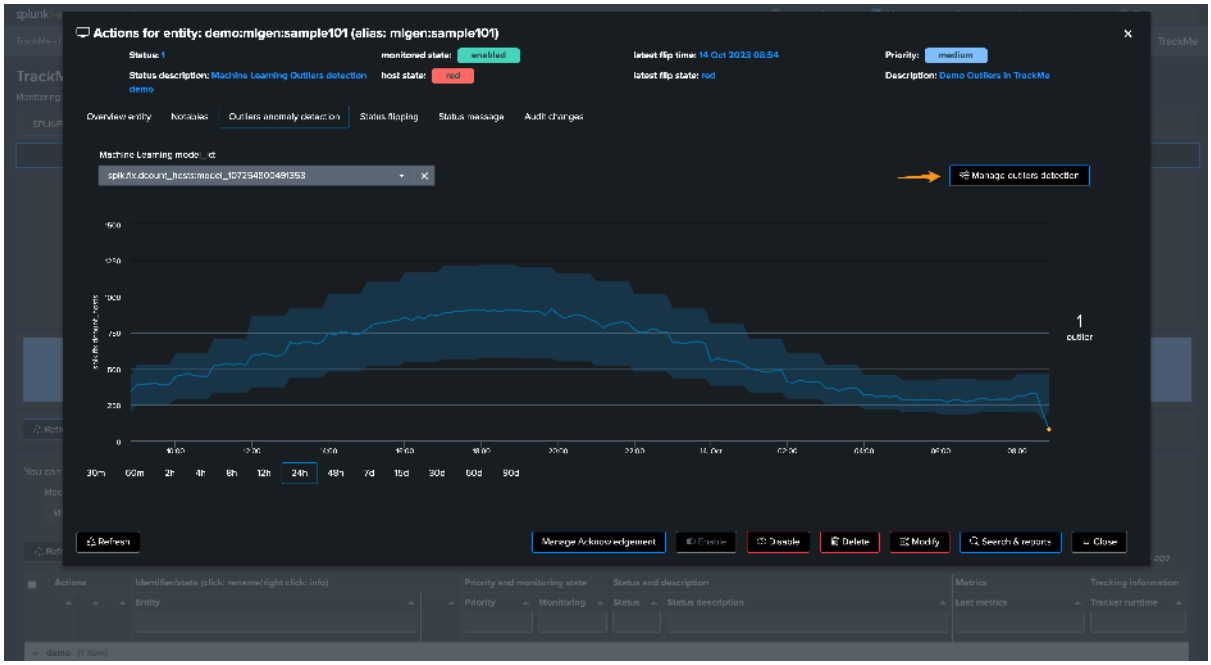
Options:

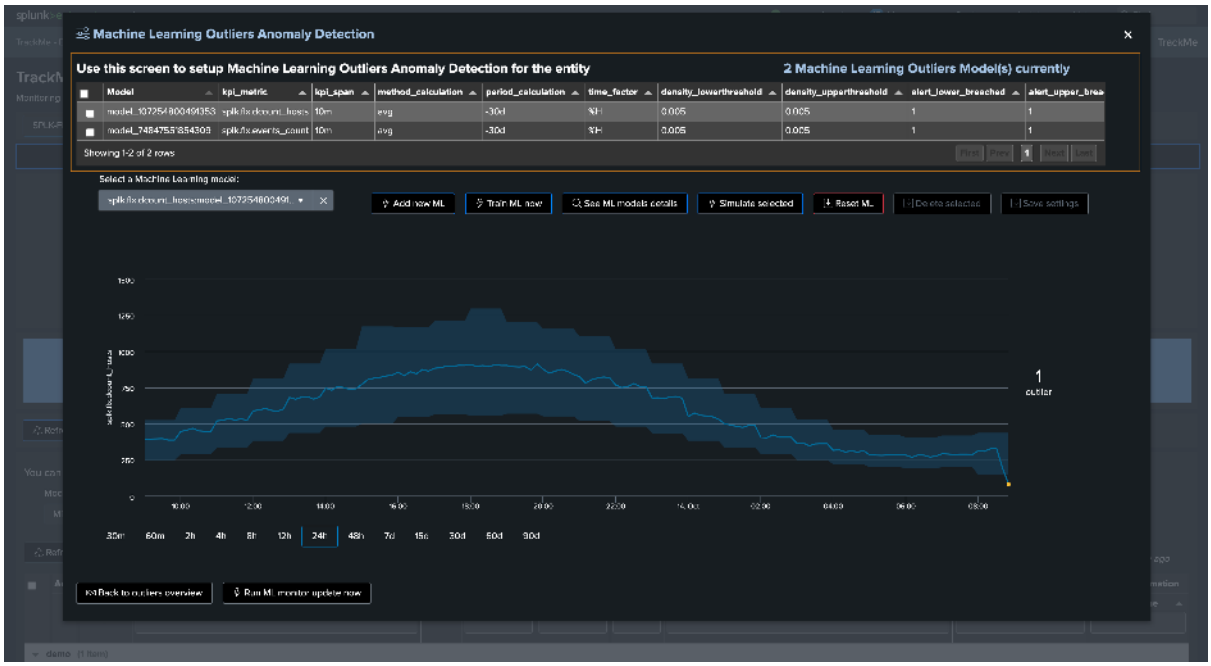
Table 13: ML Outliers system wide options

Option	Purpose
Min days historical metrics for confidence	The minimal number of days of historical metrics required to compute the confidence level of the outliers detection, defaults to 7 days
Requested time models training	The time value in seconds requested for ML models to be trained for entities, a given entity will regularly get ML models trained if possible based on this value. (defaults to 1 day)
Requested time models monitor	The time value in seconds requested for ML models to be monitored for entities, a given entity will regularly get ML models monitored if possible based on this value. (defaults to 1 hour)
Max runtime models training	The time value in seconds requested to limit the max duration of the ML training models, defaults to 15 min (reduced by 30 sec) and should be set according to the cron schedule of the ML training job
Max time since last training	When executing a rendering operation, TrackMe verifies the last time this model was trained, if this time exceeds the value set here, the model will be retrained automatically before rendering. (defaults to 15 days)
Disable outliers at discovery	When a new entity is discovered, enable or disable the volume based outliers detection by default for that entity. The feature can still be managed on demand for that entity.
Outliers default calculation	The default calculation mode used for anomaly outliers detection, can be updated per entity.
Density lower threshold	The default value of the lower threshold applied to the DensityFunction algorithm, set at discovery and be updated per entity.
Density upper threshold	The default value of the upper threshold applied to the DensityFunction algorithm, set at discovery and be updated per entity.
Volume lower breached	Alert when the lower bound threshold is breached for volume based KPIs.
Volume upper breached	Alert when the upper bound threshold is breached for volume based KPIs.
Latency lower breached	Alert when the lower bound threshold is breached for volume based KPIs.
Latency upper breached	Alert when the upper bound threshold is breached for latency based KPIs.
Default period for calculation	The relative period used by default for outliers calculations, applied during entity discovery and can be updated per entity
Default outliers time factor	The default time factor applied for the outliers dynamic thresholds calculation
Default latency kpi metric	The default kpi metric for latency outliers detection
Default volume kpi metric	The default kpi metric for volume outliers detection
Default auto correct	When defining the model, enable or disable auto_correct by default, which uses the concept of auto correction based on min lower and upper deviation.
Perc min lower deviation	If an outlier is not deviant (LowerBound) from at least that percentage of the current KPI value, it will be considered as a false positive.
Perc min upper deviation	If an outlier is not deviant (UpperBound) from at least that percentage of the current KPI value, it will be considered as a false positive.
splk_outliers_mltk_algo	TrackMe uses the MLTK DensityFunction algorithm. You can add custom algorithms as a comma-separated list of values; these will become selectable automatically in the different Outliers configuration screens in TrackMe.
splk_outliers_mltk_algo	If you have multiple algorithms, you can define here which algorithm should be used by default when TrackMe defines the ML models rules, which happens usually at the entities discovery, or when adding/resetting ML models.
splk_outliers_fit_extra_	You can optionally add extra parameters to be added to the MLTK fit command (training phase) at the time of the definition of the ML rules (generally when entities are discovered), for instance: exclude_dist="beta" to exclude Beta distributions for the density function, see MLTK documentation for more information.
splk_outliers_apply_extr	You can optionally add extra parameters to be added to the MLTK

7.13.15 ML Outliers options

Options per ML models can be accessed via the TrackMe UI:





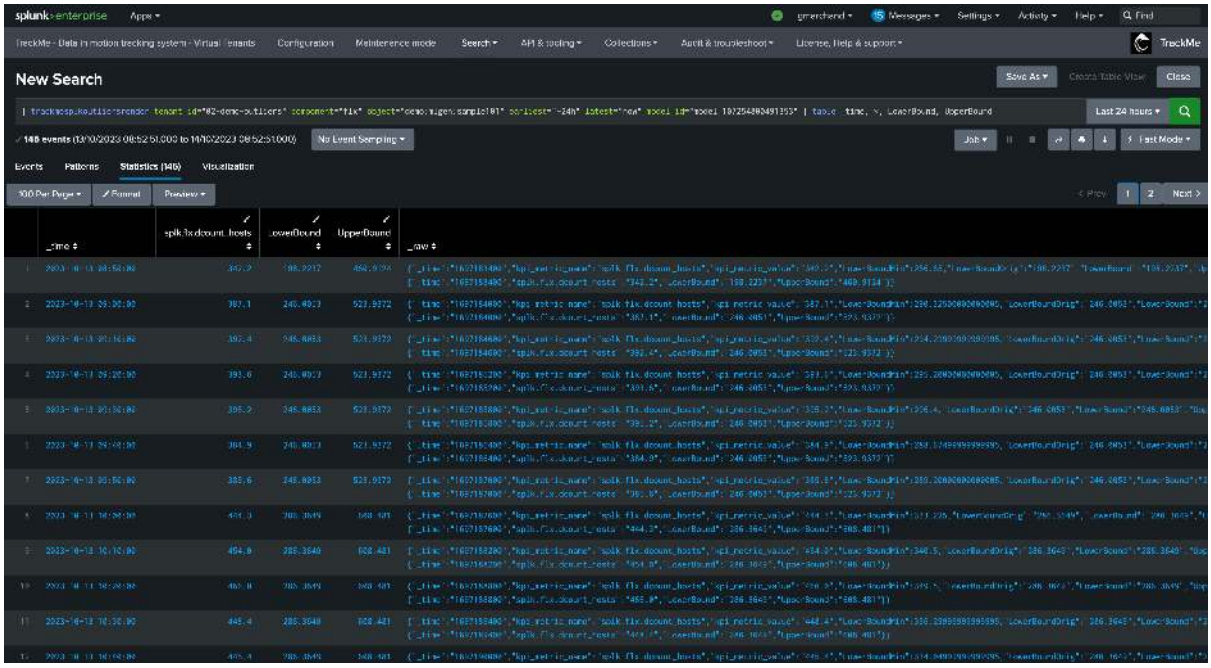
The following options can be defined per ML Outlier model:

Table 14: ML Outliers models options

Option	Purpose
kpi_metric	The Key Performance Indicator associated with the ML model
kpi_span	The span time value used for the calculations
method_calculation	The calculation method to be applied (e.g., average, perc95...)
period_calculation	The period for the calculation for the model training purposes
time_factor	Defines the time-based granularity for the ML model training
density_lowerthreshold	The lower bound threshold for the MLTK density function
density_upperthreshold	The upper bound threshold for the MLTK density function
auto_correct	Enable or disable the auto-correction features; its goal is to limit false positives using the deviation settings
perc_min_lowerbound_d	The min percentage of deviation between the lower bound and the KPI current value
perc_min_upperbound_	The min percentage of deviation between the upper bound and the KPI current value
alert_lower_breached	Alert if the lower threshold is breached
alert_upper_breached	Alert if the upper threshold is breached
min_value_for_lowerbou	The min value for the lower bound to be breached, outliers below this will be rejected
min_value_for_upperbou	The min value for the upper bound to be breached, outliers above this will be rejected
is_disabled	Enable or disable training and monitoring for this model

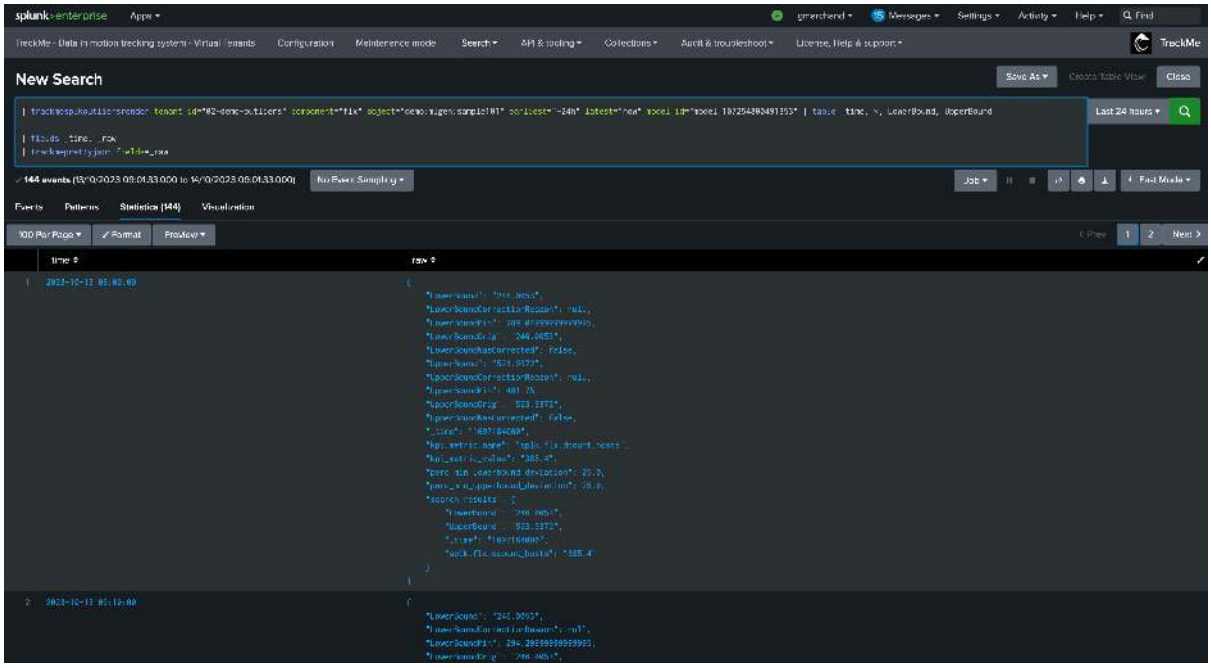
7.13.16 Understanding and Troubleshooting ML rendering results

When TrackMe calls the ML rendering, the following command is called:



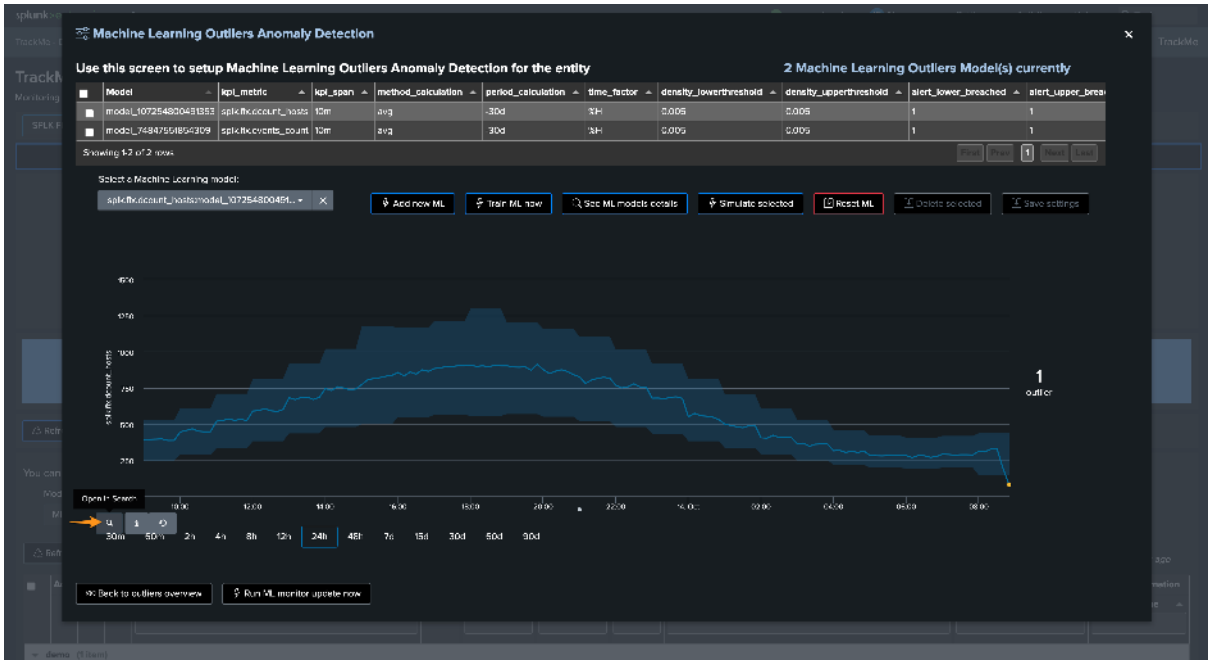
You can go in statistics mode and add the following SPL to review in details the ML decisions:

```
| fields _time, _raw
| trackmeprettyjson fields=_raw
```

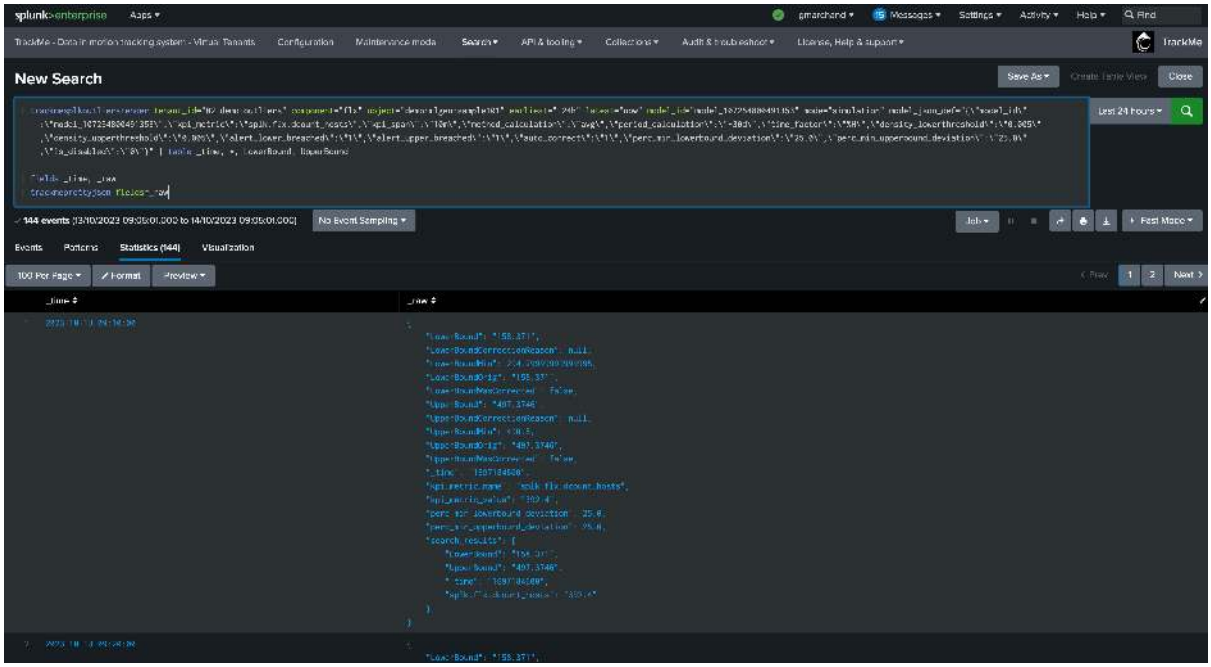


The JSON object contains various information such as the origin MLTK rendering results per time frame, and TrackMe Auto correction decisions.

While in simulation mode, the same command is called with the simulation definition of the ML instead:



You can apply the same SPL lines to review the ML decisions:



7.13.17 Troubleshooting ML training logs

The first level of the training logic is handled via the command `trackmesplkoutlierstrackerhelper`. You can access logs via:

```
index=_internal sourcetype=trackme:custom_commands:trackmesplkoutlierstrackerhelper
```

notes: this command orchestrates the ML training activity for the tenant and the component. It will then proceed by iteration to the training calling the trackmesplkoutlierstrain command

The per-entity ML training logic is handled via the command `trackmesplkoutlierstrain`. You can access logs via:

```
index=_internal sourcetype=trackme:custom_commands:trackmesplkoutlierstrackerhelper
```

7.13.18 Troubleshooting ML rendering (monitoring) logs

Similarly, the ML monitor phase is orchestrated by the `trackmesplkoutlierstrackerhelper` command. You can access logs via:

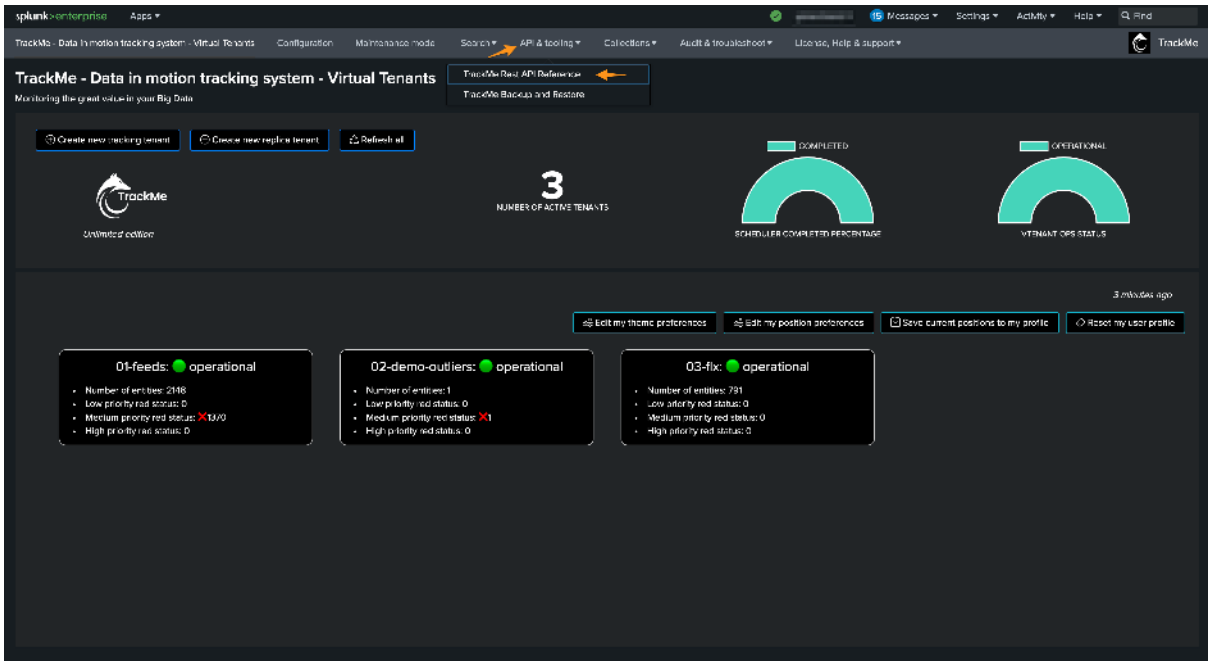
```
index=_internal sourcetype=trackme:custom_commands:trackmesplkoutlierstrackerhelper
```

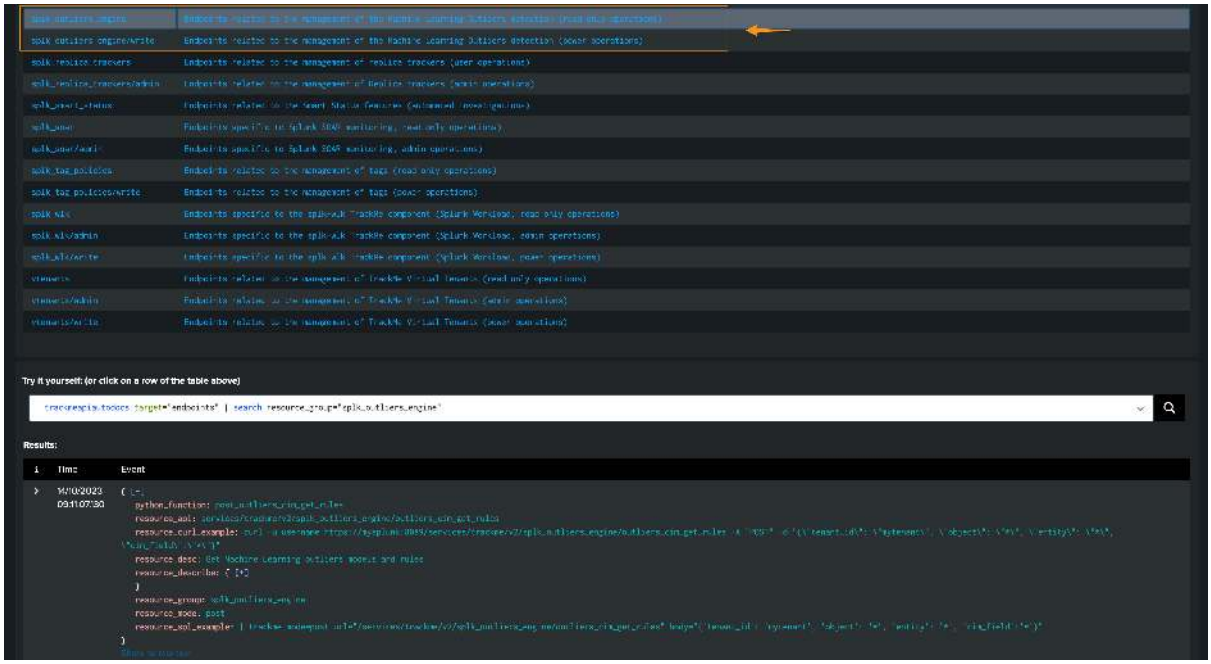
It will then call the command `trackmesplkoutliersrender`. You can access logs via:

```
index=_internal sourcetype=trackme:custom_commands:trackmesplkoutliersrender
```

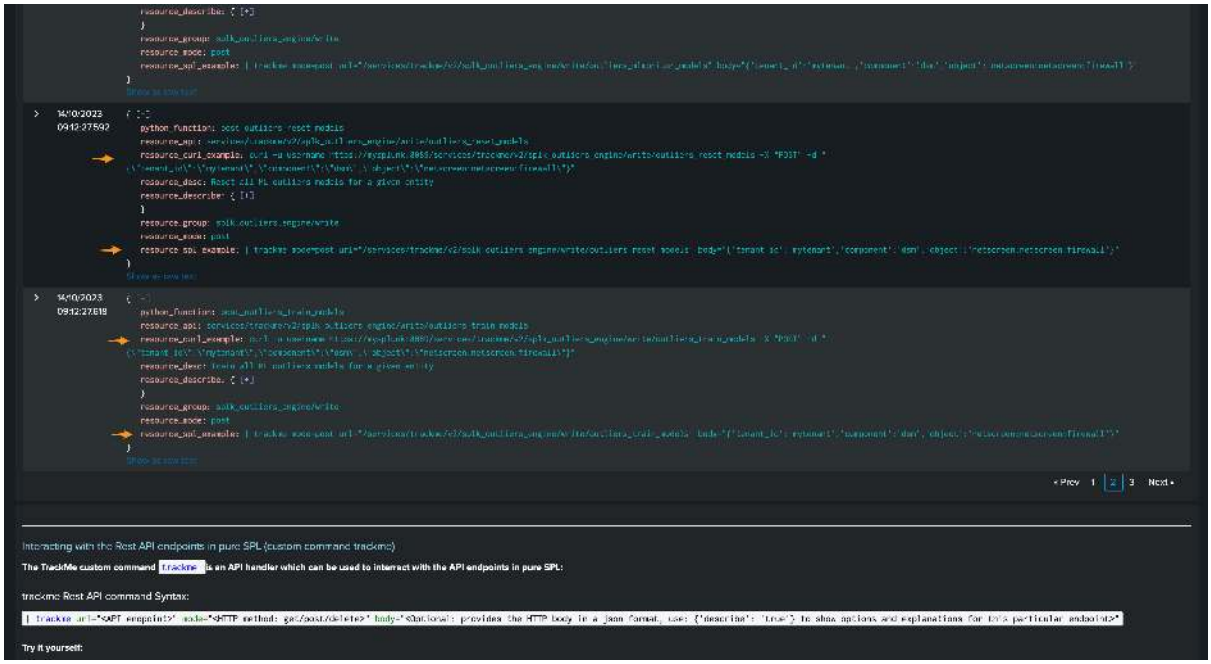
7.13.19 REST API endpoints for ML in TrackMe

TrackMe exposes different REST API endpoints for the ML Outliers purposes; the same endpoints are used by the UI:





TrackMe provides usage examples with every endpoint, both in cURL and in SPL:



You can, for instance, reset ML models or force training via the REST API:

train:

```
| trackme mode=post url="/services/trackme/v2/splk_outliers_engine/write/outliers_
↪train_models" body="{'tenant_id':'02-demo-outliers','component':'flx','object':
↪'demo:mlgen:sample101'}"
```

reset:

```
| trackme mode=post url="/services/trackme/v2/splk_outliers_engine/write/outliers_
↪reset_models" body="{'tenant_id':'02-demo-outliers','component':'flx','object':
↪'demo:mlgen:sample101'}"
```

7.13.20 Expanding ML models results and definition

In TrackMe version 2.0.67, we have included a streaming command `trackmesplkoutliersexpand` which can expand the models results or definition. This can be useful to add more context for custom alerting or reporting.

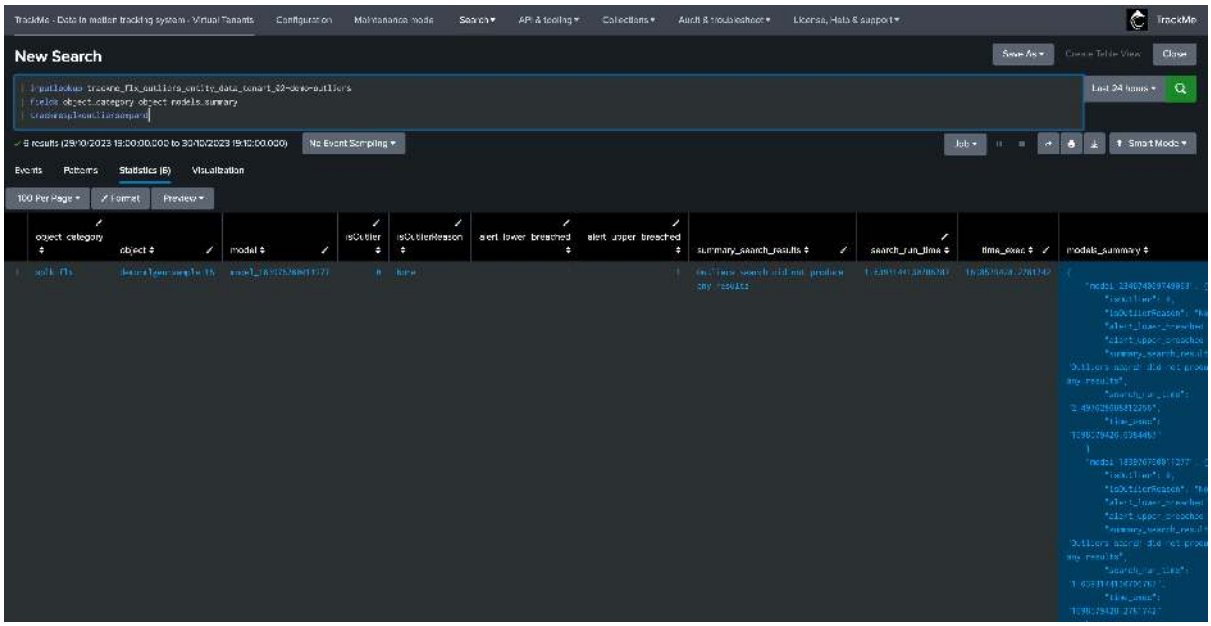
ML Models are stored in complex dictionaries, and even more when there is more than a single model in a given entity. Accessing the models' deep information is challenging; the Splunk `spath` command is designed to handle these use cases properly.

The following example expands results from our Flex tenant 02-demo-outliers:

```
| inputlookup trackme_flx_outliers_entity_data_tenant_02-demo-outliers
```

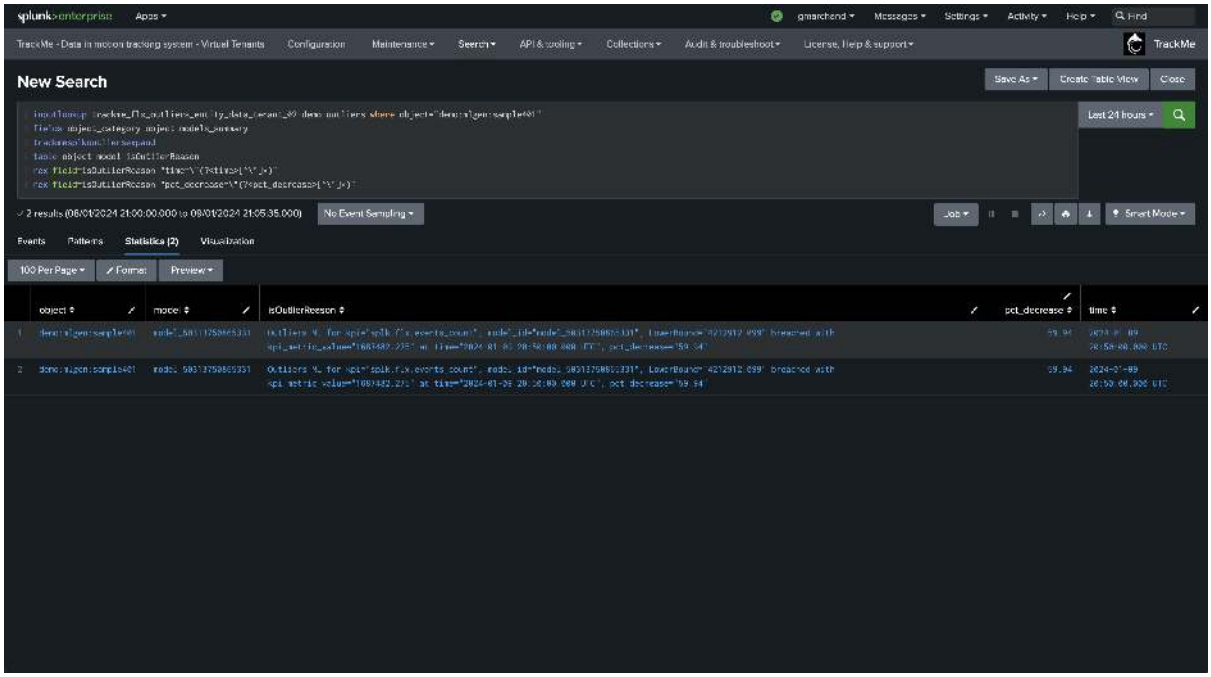
The following command expands the models for each entity; we then get as many rows as we have entities and models, with access to the model details:

```
| inputlookup trackme_flx_outliers_entity_data_tenant_02-demo-outliers
| fields object_category object models_summary
| trackmesplkoutliersexpand
```



For instance, if you wish to extract the time and percentage of decrease when Outliers are detected:

```
| inputlookup trackme_flx_outliers_entity_data_tenant_02-demo-outliers where object=
↪"demo:mlgen:sample401"
| fields object_category object models_summary
| trackmesplkoutliersexpand
| table object model isOutlierReason
| rex field=isOutlierReason "time=\"(?<time>[^\"]*)"
| rex field=isOutlierReason "pct_decrease=\"(?<pct_decrease>[^\"]*)"
```



We can also expand the models definition; note the rename:

```
| inputlookup trackme_flx_outliers_entity_rules_tenant_02-demo-outliers
| fields object_category object entities_outliers
| rename entities_outliers as models_summary
| trackmesplkoutliersexpand
```

7.13.21 Mass deleting ML models

Say we want to mass delete ML models based on a certain criterion like the KPI; we can use the Splunk `map` command in addition with TrackMe REST API endpoints.

The following example deletes models associated with the latency KPI:

```
| inputlookup trackme_dsm_outliers_entity_rules_tenant_02-demo-outliers
| table object_category, object, entities_outliers
| rename entities_outliers as models_summary
| trackmesplkoutliersexpand
| where kpi_metric="splk.feeds.avg_eventcount_5m"
| eval tenant_id="02-demo-outliers"
| table tenant_id, object_category, object, model
| map maxsearches=100 search="| trackme mode=post url=/services/trackme/v2/splk_
↪outliers_engine/write/outliers_delete_models body=\"{'tenant_id': '$tenant_id$',
↪'component': 'dsm', 'object': '$object$', 'models_list': '$model$'}\""
```

7.14 TrackMe Data Sampling - Events and Format Recognition for Quality Inspection in TrackMe

7.14.1 Introduction to TrackMe's Events Format Recognition Engine

This documentation applies from TrackMe 2.1.0:

- TrackMe 2.1.0 welcomed the sampling engine v2.0, which is a major rewrite and improvement of the previous version.

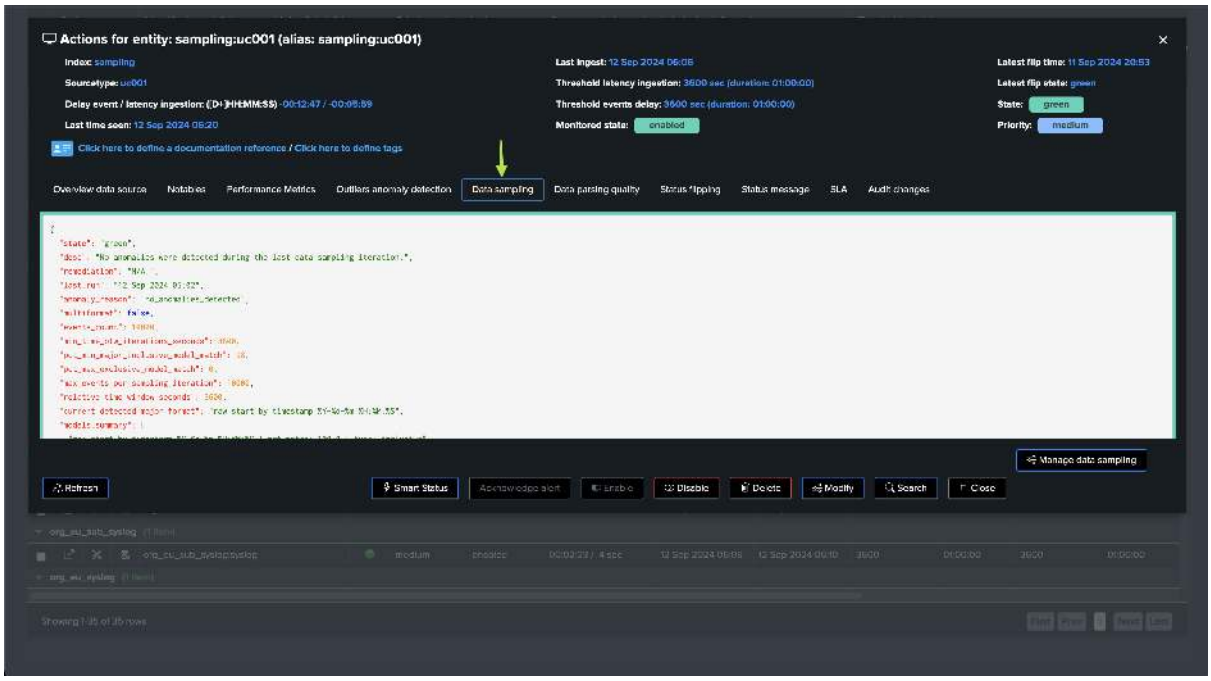
- Therefore, this documentation applies to TrackMe 2.1.0 and later versions.

About TrackMe's data sampling features:

- TrackMe's data sampling engine is a component of the splk-dsm module, which is responsible for the recognition of events and formats in the data being ingested by Splunk.
- The data sampling engine feature provides automated quality assessment of the data being ingested by Splunk, at scale.
- It works by regularly sampling events and applying built-in and custom rules to recognize the format of the events, which is called inclusive matching.
- The engine categorizes events by model match, and tracks the percentage of events matched per model, identifying the **major model** of the data.
- The inclusive major model percentage is then compared against the minimum acceptable threshold, and if the percentage is below the threshold, the entity is marked as in data sampling anomaly.

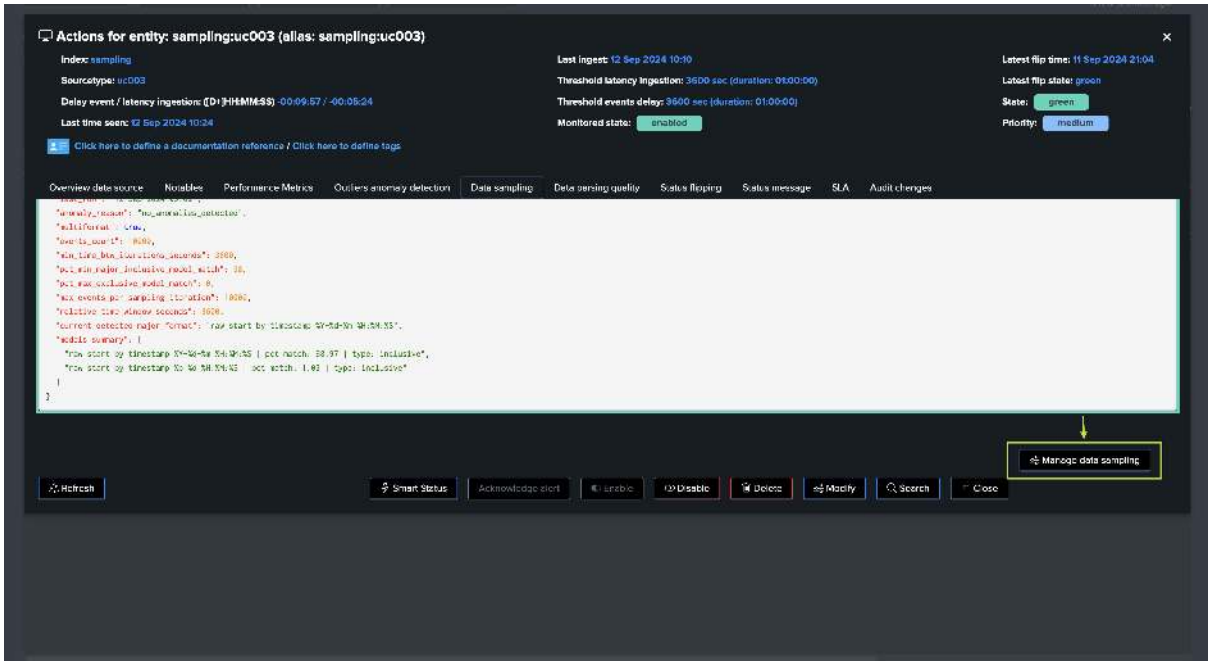
7.14.2 How Does the Data Sampling Stand in TrackMe's Workflow

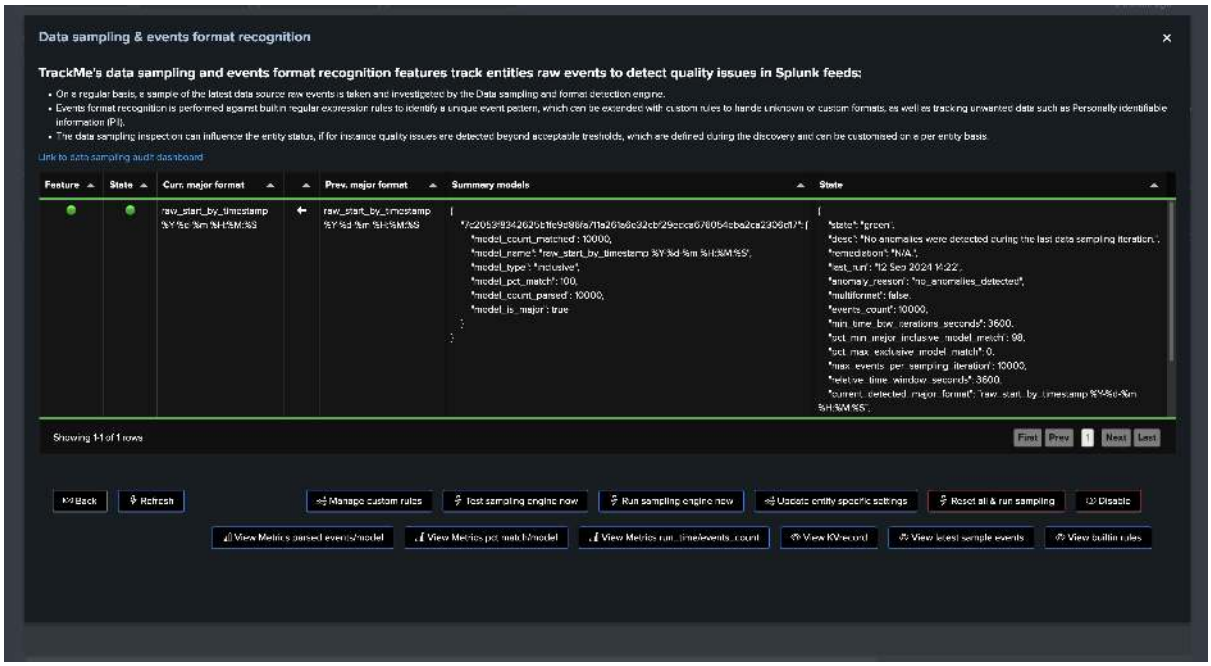
You can find the data sampling status and access to its management in TrackMe's entity view, in the **Data Sampling** tab:



Depending on the status of the data sampling, as well as the global Virtual tenant configuration, the sampling status may or may not influence the entity status.

You access to the full detailed by clicking on "Manage Data sampling" button:





7.14.3 High-Level Overview of the Data Sampling Process

In a very high-level overview, the data sampling process works as follows:

`Step 1: Data Sampling Tracker Execution and Entities to be Sampled Identification`

1. The Virtual Tenant's Data sampling tracker, which is a scheduled job called **trackme_dsm_data_sampling_tracker_tenant_<tenant_id>**, is executed every 20 minutes.

2. The scheduled backend starts by extracting the list of TrackMe entities to be sampled, using the following Splunk search:

Example:

```
| inputlookup trackme_dsm_tenant_mytenant where monitored_state="enabled"
| eval key=_key
```

(continues on next page)

(continued from previous page)

```
| `trackme_exclude_badentities`
| where data_last_time_seen>relative_time(now(), "-24h")
| lookup trackme_dsm_data_sampling_tenant_mytenant object OUTPUT data_sample_feature,␣
↪relative_time_window_seconds, data_sample_last_entity_epoch_processed, min_time_btw_
↪iterations_seconds, data_sample_mtime
``` only consider entities where the last processed epoch (data_sample_last_entity_
↪epoch_processed) is older than data_last_time_seen, or null (entities has not been␣
↪processed yet) ```
| where (isnull(data_sample_last_entity_epoch_processed) OR data_sample_last_entity_
↪epoch_processed<data_last_time_seen)
| eval data_sample_feature=if(isnull(data_sample_feature), "enabled", data_sample_
↪feature) | where (data_sample_feature!="disabled")
``` only consider entities where the min_time_btw_iterations_seconds is older than␣
↪the current time (bigger or equal to the time spent since last run, or null for new␣
↪entities) ```
| eval time_spent_since_last_run=now()-data_sample_mtime
| where (isnull(min_time_btw_iterations_seconds) OR time_spent_since_last_run>=min_
↪time_btw_iterations_seconds)
``` define a priority rank, entities that have been set as disabled_auto should be␣
↪processed last compared to entities in disabled_audo ```
| eval priority_rank=if(data_sample_feature=="enabled", 1, 2)
``` order ```
| sort limit=0 priority_rank, data_sample_mtime
| fields object, key, data_last_time_seen, *
| eval earliest_target=if(isnum(relative_time_window_seconds), data_last_time_seen-
↪relative_time_window_seconds, data_last_time_seen-3600)
```

Notes:

- Some of these settings are defined at the system wide level, the search without variable values is the following, the example above is a simplified version of the search for a tenant named "mytenant"
- To see what is the actual search executed by TrackMe, look at:

```
index=_internal sourcetype=trackme:custom_commands:trackmesamplingexecutor "Executing␣
↪upstream definition search"
```

`Step 2: Run the Data Sampling Process for Each Entity`

1. For each entity to be sampled, the data sampling process is executed.
2. This process means that TrackMe identifies the search to be executed, which depends on the context of the entity as well as various settings such as the relative time window (against the last known event for that entity).
3. The search is executed, each event result from the search goes through the rules, customer rules if any, and the built-in rules.
4. The engine extracts a sample of events per model matched and per entity, which will be stored in the data sampling KVstore collection for investigation purposes.
5. Additionally, the engine performs various calculations, notably the percentage of events matched per model, and the major model of the entity.
6. Finally, the engine generates various metrics (KPIs) in the tenant metrics index, and updates the KVstore record defining the data sampling status, messages, and other metadata.

Notes:

- You can review the logs associated with the data sampling operation of a given TrackMe entity as easy as:

```
index=_internal sourcetype=trackme:custom_commands:trackmesamplingexecutor object="
↪<object value>"
```

Inspecting the anomaly status summary message

In this screen, TrackMe provides a summary of the data sampling status in JSON format, which includes various high level information so you can understand easily the current status, example:

```
{
    "state": "green",
    "desc": "No anomalies were detected during the last data sampling iteration.",
    "remediation": "N/A.",
    "last_run": "12 Sep 2024 05:02",
    "anomaly_reason": "no_anomalies_detected",
    "multiformat": false,
    "events_count": 10000,
    "min_time_btw_iterations_seconds": 3600,
    "pct_min_major_inclusive_model_match": 98,
    "pct_max_exclusive_model_match": 0,
    "max_events_per_sampling_iteration": 10000,
    "relative_time_window_seconds": 3600,
    "current_detected_major_format": "raw_start_by_timestamp %Y-%d-%m %H:%M:%S",
    "models_summary": [
        "raw_start_by_timestamp %Y-%d-%m %H:%M:%S | pct_match: 100.0 | type:␣
↪inclusive"
    ]
}
```

Data sampling statuses

field: state

The data sampling state can be reported as:

Value	Description
green	No anomalies were detected during the last data sampling iteration.
red	Anomalies were detected during the last data sampling iteration, if the Virtual Tenant configuration allows it, it will turn the entity red with anomaly_reason containing **data_sampling**.
orange	Since the discovery of the entity, TrackMe has detected quality issues and anomalies, TrackMe will continue performing sampling in case conditions change, but it will not affect the overall entity status.

The state is also reflected as the border colour of the data sampling tab in the entity view.

field: anomaly_reason

The data sampling anomaly reason can be reported as:

Value	Description
no_anomalies_detected	No anomalies were detected during the last data sampling iteration.
exclusive_rule_match	Anomalies detected, one or more exclusive rules have been matched.
inclusive_rule_match	Anomalies detected, quality issues were detected, the min percentage of the major model matched does not meet requirements which indicates that a too large number of events do not share the same format as the majority of events.
format_change	The major event format (the format previously detected for the majority of events) has changed from <model_name> to <model_name>, this might indicate a non expected quality issue or condition change in the ingestion of this feed in Splunk.
anomalies_at_discovery	Anomalies were detected since the entity discovery, multiple formats were detected and the major model is under the acceptable threshold of percentage of events matched by the major model. The data sampling feature was automatically disabled (disabled_auto) to avoid generating false positives for this entity (the feature will not be allowed to influence the entity status), however TrackMe will continue attempting to process in case conditions for this feed change.
anomalies_since_discovery	Anomalies were detected since the entity discovery, multiple formats were detected and the major model is under the acceptable threshold of percentage of events matched by the major model. The data sampling feature was automatically disabled (disabled_auto) to avoid generating false positives for this entity (the feature will not be allowed to influence the entity status), however TrackMe will continue attempting to process in case conditions for this feed change.

field: models_summary

This field is a list of the models detected during the last data sampling iteration, with the following information:

- The model name (model_name)
- The percentage of events matched by this model (pct_match)
- The type of match (inclusive or exclusive)

Example:

```
"models_summary": [
    "raw_start_by_timestamp %Y-%d-%m %H:%M:%S | pct_match: 100.0 | type: inclusive"
]
```

When multiple models are matched, the list will contain multiple entries, example:

```
"models_summary": [
    "raw_start_by_timestamp %Y-%d-%m %H:%M:%S | pct_match: 98.97 | type: inclusive",
    "raw_start_by_timestamp %b %d %H:%M:%S | pct_match: 1.03 | type: inclusive"
]
```

fields: settings for this entity (min_time_btw_iterations_seconds, etc)

The following fields are settings for this entity, defined when the entity was discovered using system wide corresponding settings, and that can be updated on a per entity basis:

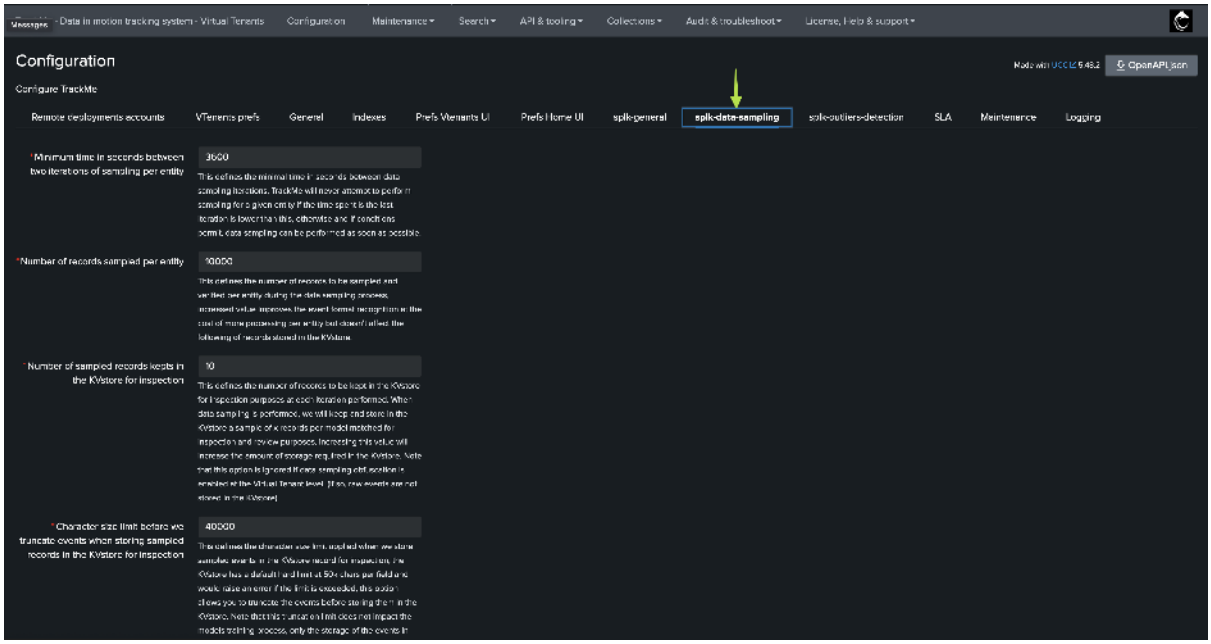
Value	Description	Default Value
min_time_btw_	Minimum time in seconds between two iterations of sampling per entity. TrackMe will never attempt to perform sampling for a given entity if the time since the last iteration is lower than this value.	3600
relative_time_wind	The size in seconds of the time window for the sampling operation, relative to the latest event time known for the entity. This is used to calculate the earliest time for the sampling search.	3600
max_events_per	Defines the number of records to be sampled and verified per entity during the data sampling process. Increasing this value improves event format recognition at the cost of more processing per entity.	10000
pct_min_major_	The minimal percentage of events to match the major inclusive model. If the main model matched has less than this percentage of events matching, the entity state will be impacted.	98
pct_max_exclus	Defines the maximum percentage of events matching an exclusive model that can be accepted per iteration. By default, no events matching an exclusive model are accepted.	95

Hint

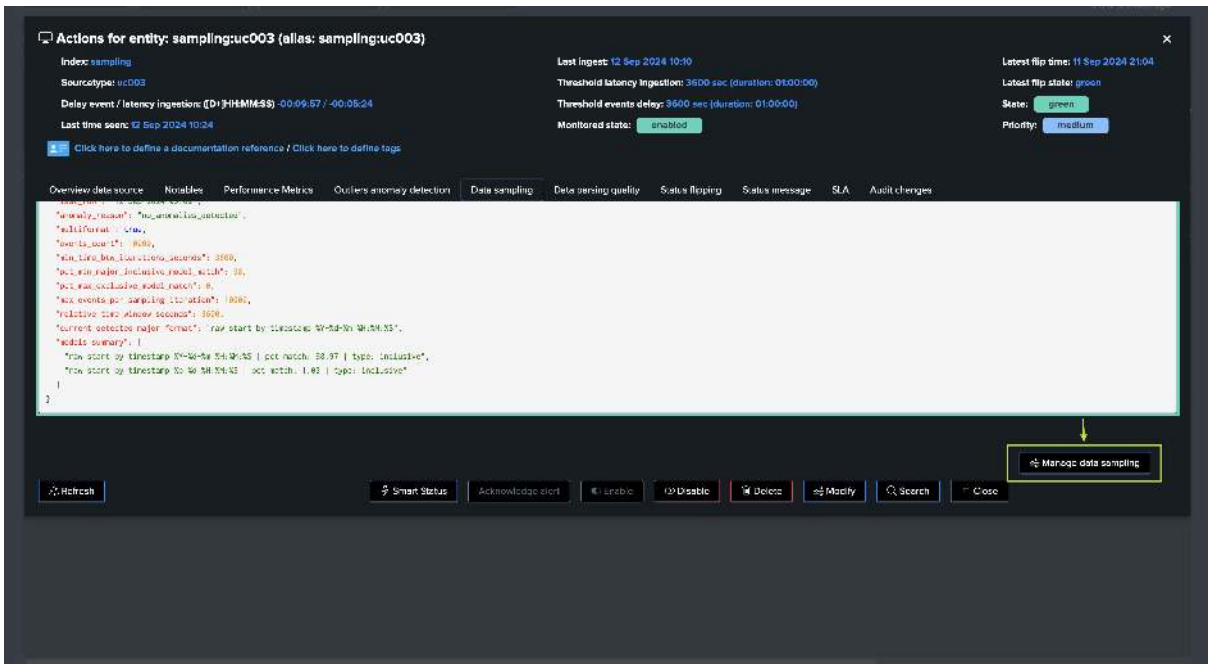
System level settings:

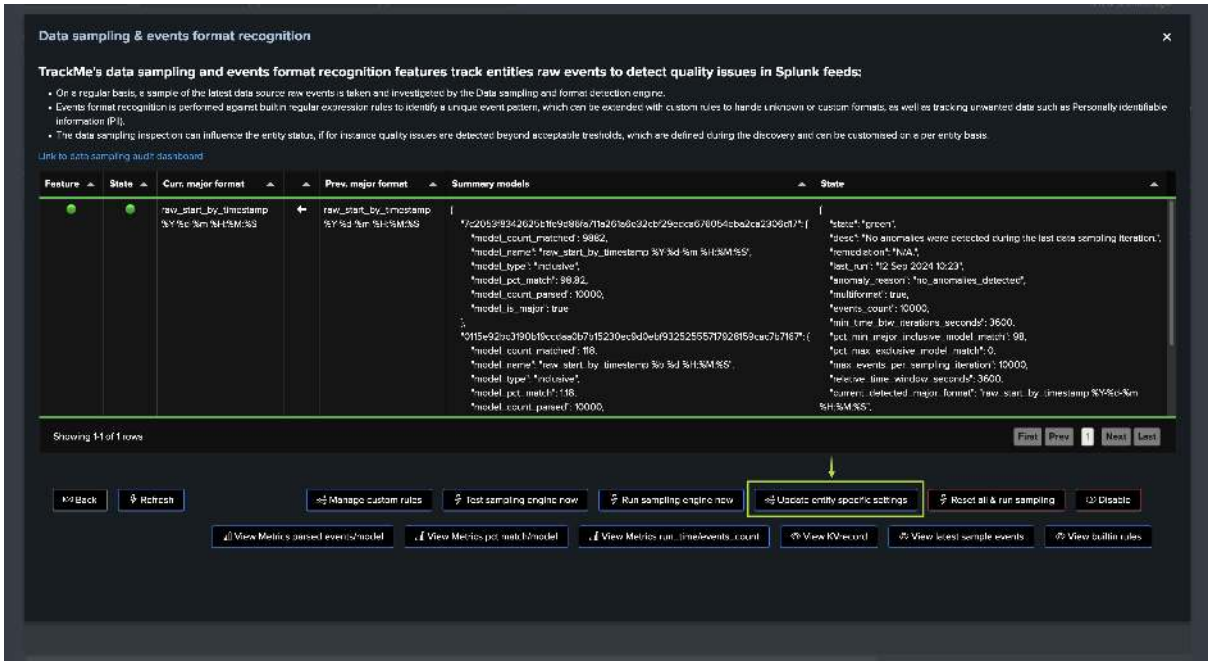
- These settings are applied at the level of TrackMe entities where the data sampling discovers the entity.
- Each of these settings can be updated on a per entity basis, and the entity settings will override the system level settings.

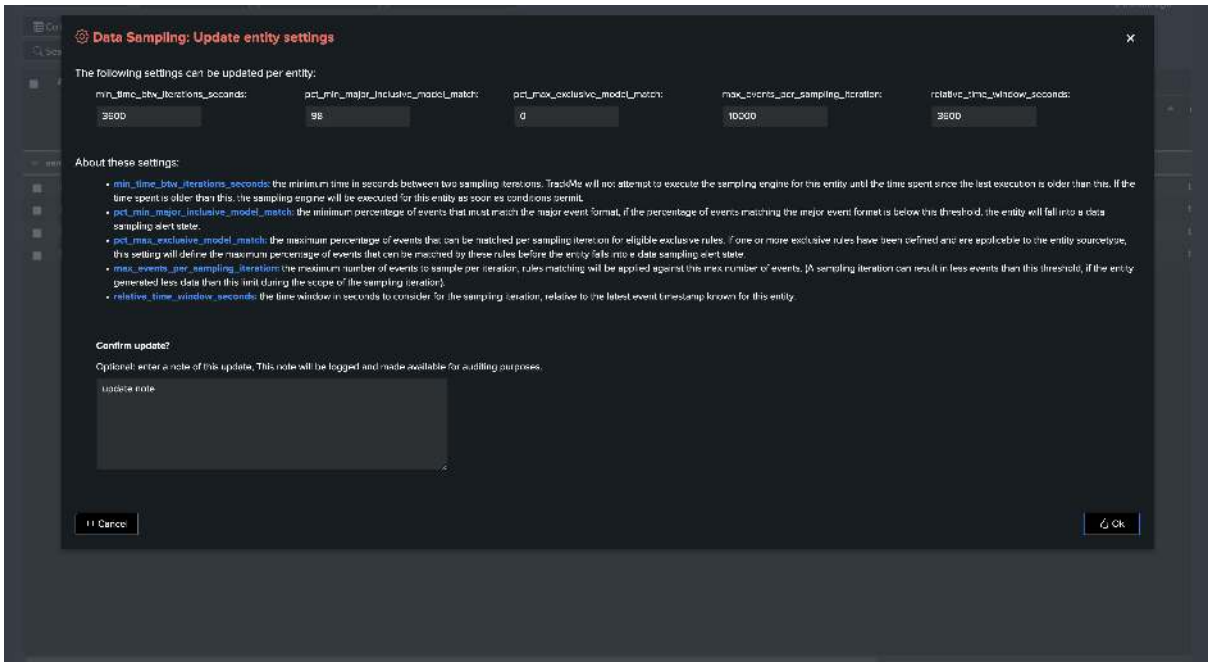
System level settings screen:



Entity level settings management screen:







Inclusive and Exclusive Models Matching, Sourcetype Scope, Major Model and Thresholds

Hint

About these concepts:

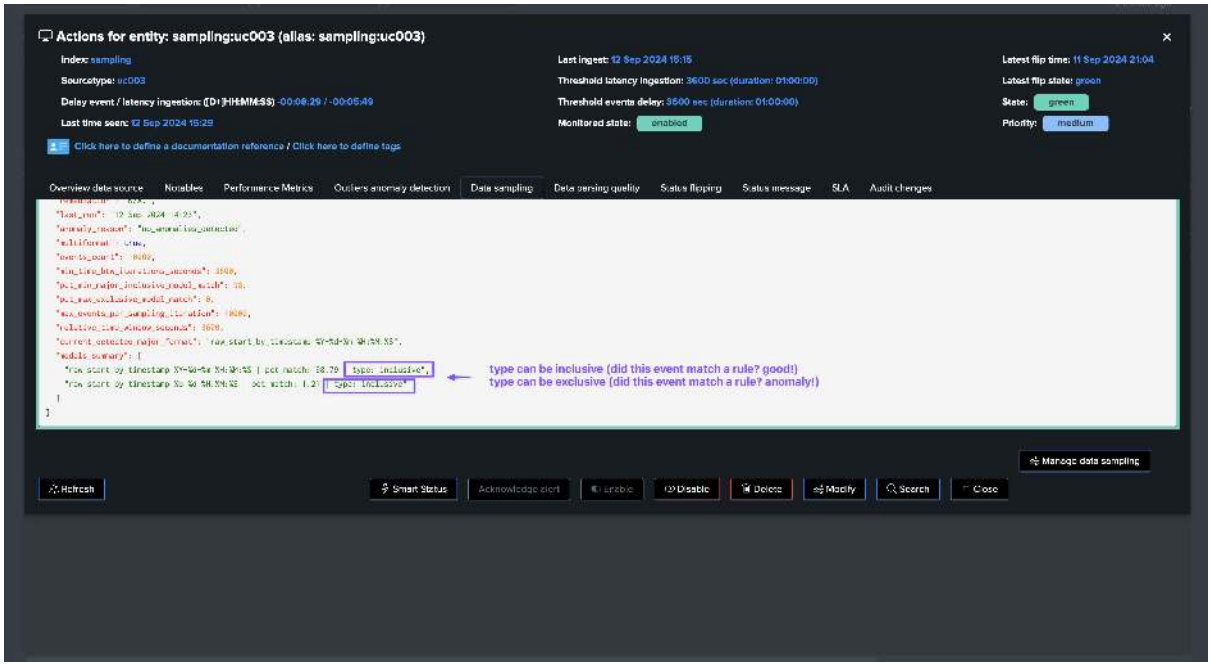
- Built-in rules in TrackMe are always inclusive rules, with sourcetype_scope=*.
- For inclusive rules, when a given event matches a certain rule, the engine stops processing other rules for this event, therefore, an event can only match one rule.
- For exclusive rules, a given event can match multiple rules (as we basically look for "bad things" such as PII data), and the engine will track the percentage of events matching each exclusive rule.
- You can create custom rules which can be inclusive OR exclusive, but not both.

- Each rule is associated with a **sourcetype_scope**, which can be an explicit sourcetype, a list of sourcetypes, and as well any combination of wildcards for each.

Inclusive and exclusive models matching

Data sampling rules in TrackMe can be defined with two types of matching:

- `inclusive`: The rule is applied to the events for format recognition purposes, this means "does the rule regular expression match the event?".
- `exclusive`: The rule is applied to the events, however, if the rule is matched this means that a bad condition is detected "this rule should never match events".



Sourcetype scope

When TrackMe's data sampling engine processes an event, it will only apply the rules that are associated with the sourcetype of the event, relying on the sourcetype_scope:

- All builtin rules are inclusive type rules, with a sourcetype_scope of "*".
- Custom rules can be defined with a specific sourcetype_scope, which can be a single sourcetype, a list of sourcetypes, and wildcards can be used in each value submitted.
- When creating custom rules, the sourcetype_scope predicates which events will be processed by the rule, and which will not.

Major model

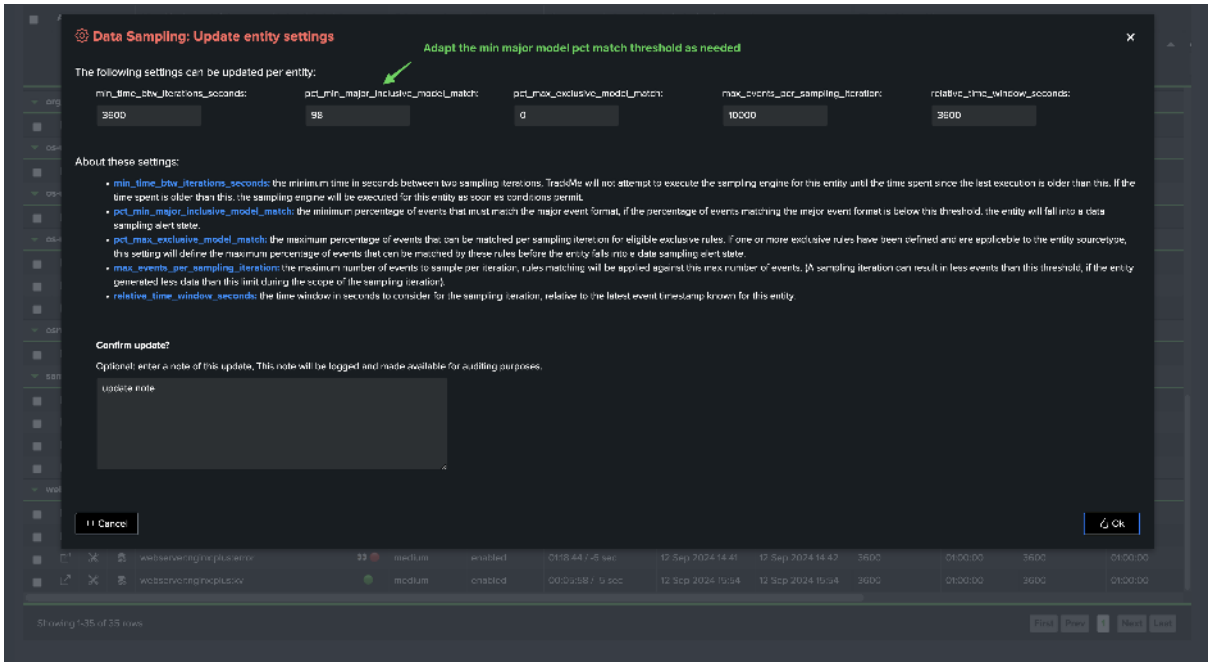
When the data sampling engine performs an iteration for a given entity, it calculates the percentage of events that match each model, and identifies the major model:

- The major model matters for the entity status, as it is the model that the majority of events match.
- This concept applies only to inclusive rules matching.
- During the sampling process, if the percentage of events matching the major model is below the threshold `pct_min_major_inclusive_model_match`, a sampling anomaly status will be raised as `inclusive_rule_match`.

- However, if this happens at the first iteration of the entity, the status will be raised as `anomalies_at_discovery`, and will not be allowed to influence the entity status.
- If from the second iteration, the issue persists, the status will be raised as `anomalies_since_discovery`, TrackMe will continue to attempt to process sampling for this entity in case conditions change, but it will not allow the entity status to be impacted.
- If later on in the life cycle of the Splunk feed, the percentage of events matched for a given model meets the minimal threshold, the status will be updated to `no_anomalies_detected` and it becomes eligible for data sampling.
- Then, if later on a quality issue affects that same entity, the sampling engine would raise an alert for this entity.

In short!

- The concept of major model is used to make the distinction between the models that represent the vast majority of events, so we can exclude minor pollution while focussing on the true events formats.
- We can then detect abnormal variations, such as a format change or a quality issue that is introduced in the feed, while at the same time avoiding false positives.
- Finally, the threshold can be customised per entity, so the engine can adapt to the specificities of all feeds with flexibility.



Max Events count per iteration, relative time period, and minimum interval between iterations

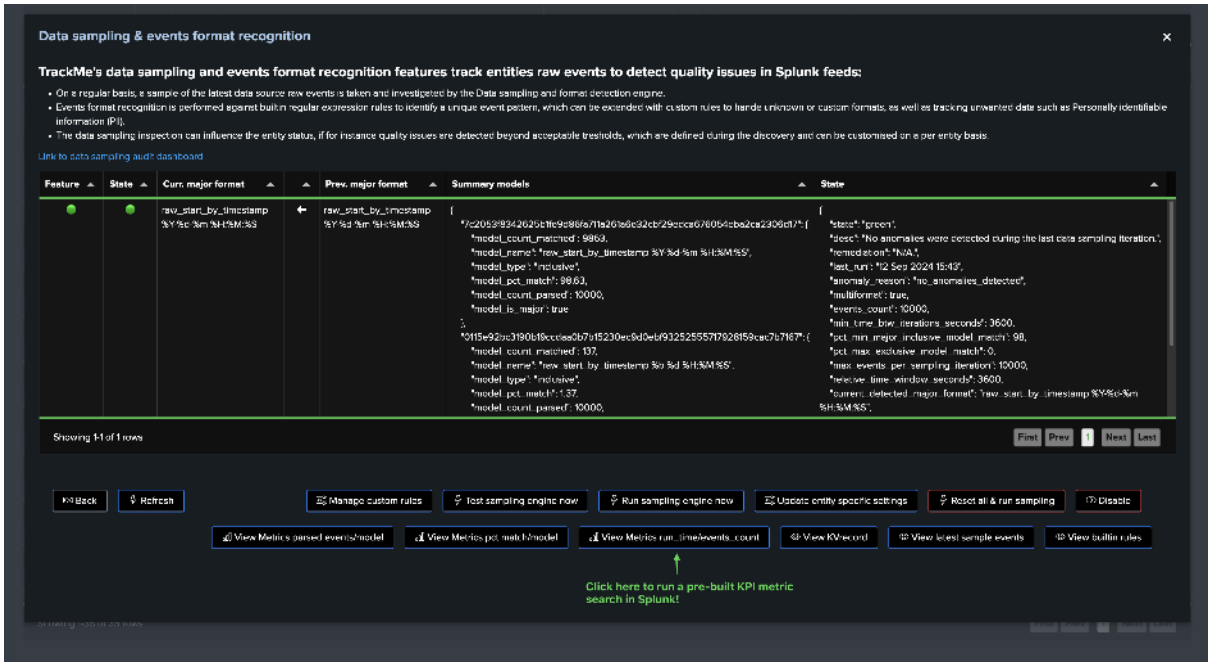
When the engine performs a data sampling iteration, it also takes into account the following concepts and settings:

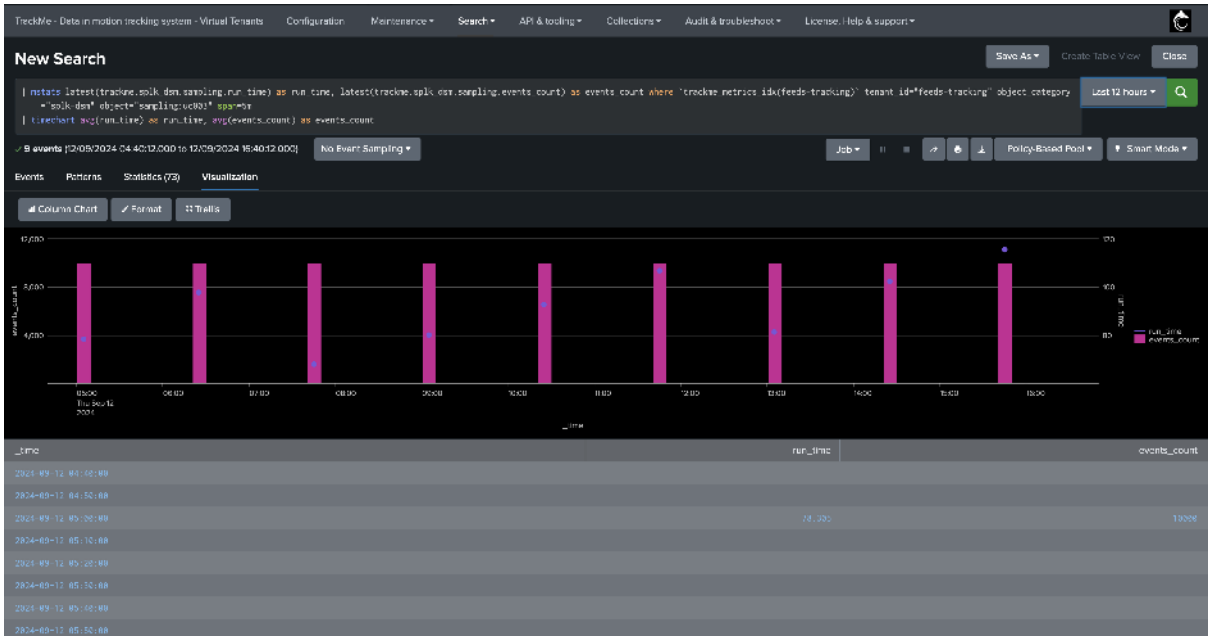
- **max_events_per_sampling_iteration**: This setting defines the maximum number of events that will be sampled per iteration for a given entity. Increasing this value will improve the quality of the data sampling, but will also increase the processing time.
- **relative_time_window_seconds**: This setting defines the time window for the sampling operation, relative to the latest event time known for the entity. This is used to calculate the earliest time for the sampling search. (In short, if the value is 3600 seconds, and the last event time known for the entity is 12:00, the search's earliest time will start at 11:00).

- **min_time_btw_iterations_seconds**: This setting defines the minimum time in seconds between two iterations of sampling per entity. TrackMe will never attempt to perform sampling for a given entity if the time since the last iteration is lower than this value.

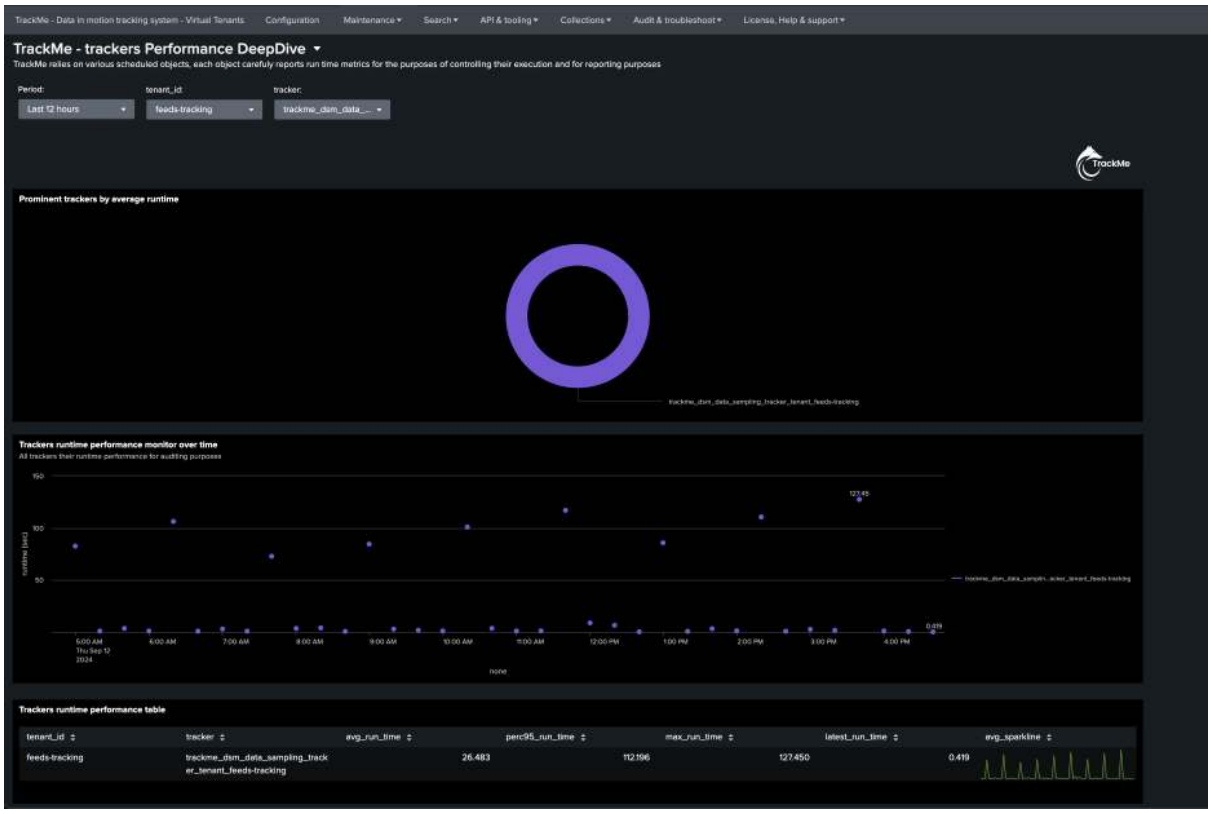
About processing time, truncation and costs:

- **About processing time:** These settings will influence the processing time of each sampling iteration, the more events to be processed and the larger the time window, the more time it will take to process the entity.
- **About truncation:** The sampling engine is not affected by truncation, the regular expression is applied to the full events, truncation is only applied against a subset of the sampled events which we store in the KVstore for investigation purposes.
- **About costs:** The sampling engine generates run_time and processed events count KPIs, so you can easily track and understand the run time costs of the sampling operations.





The total run time costs of the data sampling tracker itself can be tracked using TrackMe's deepdive UI:



About the run time of the data sampling tracker!

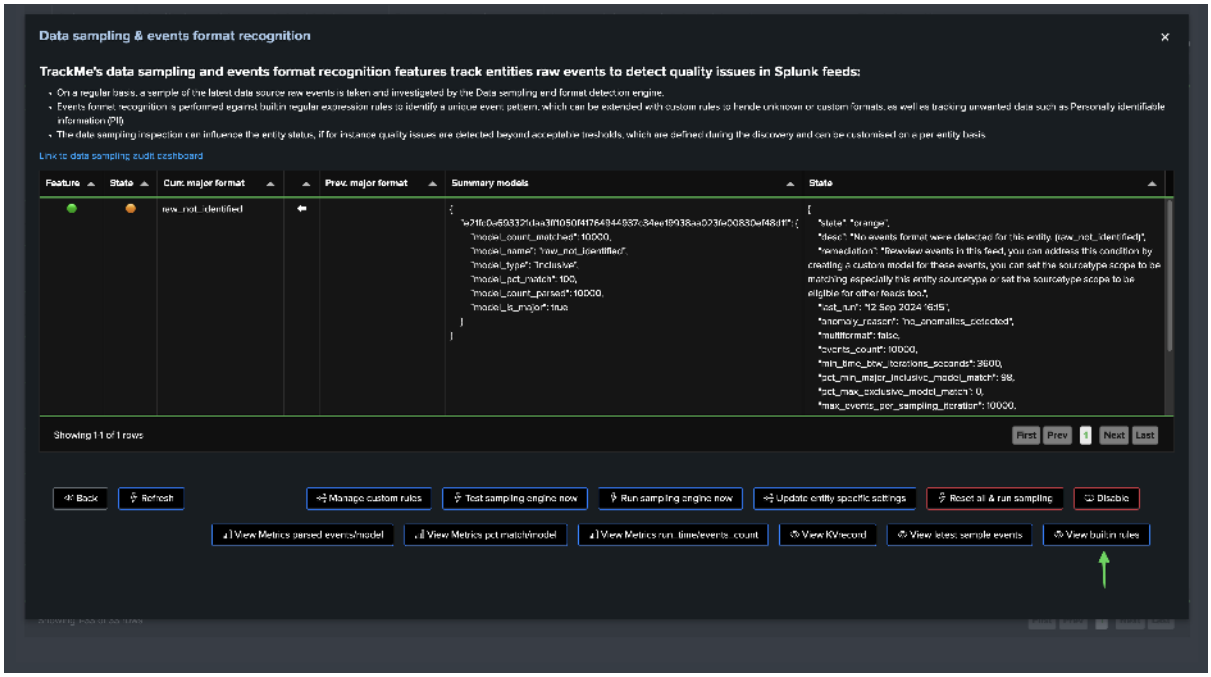
- The data sampling tracker is a scheduled job that runs every 20 minutes, and it will process all entities that are eligible for sampling.
- In fact, it will attempt to process as many entities as possible in its eligible time frame, and will stop automatically before generating skipping searches.
- At the next iteration, it will process the next chunk of entities, which are ordered based on their latest iteration, as well as a priority rank based on if a format was detected or not.

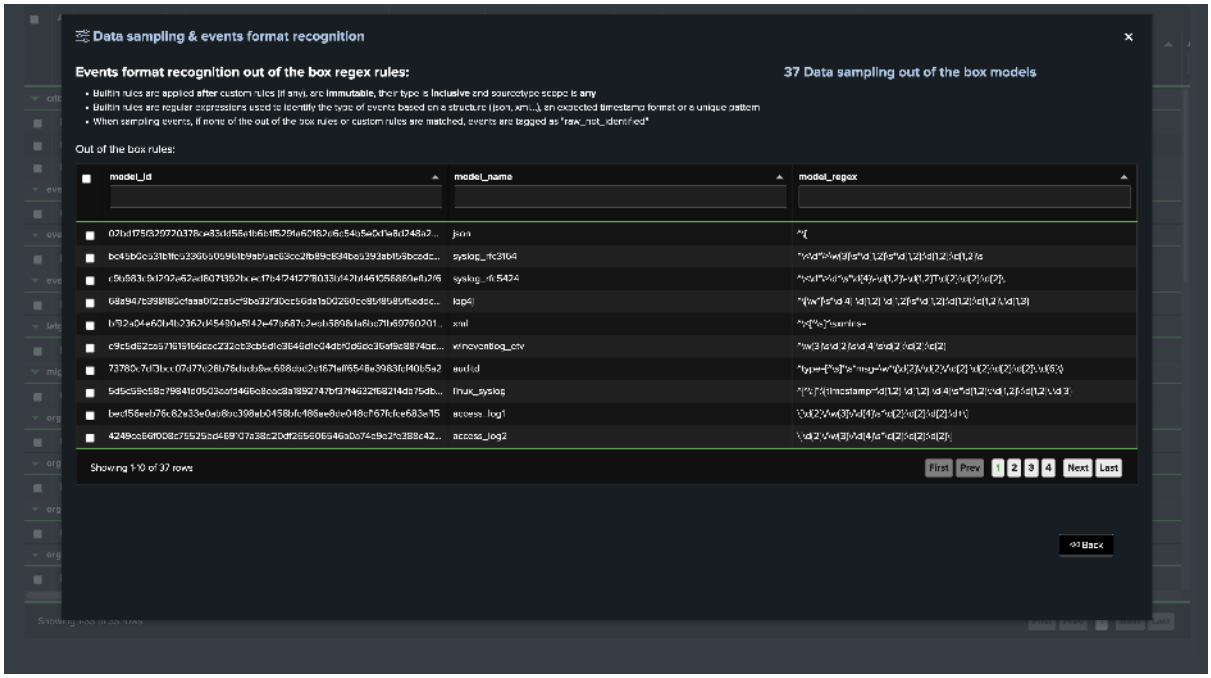
7.14.4 Out-of-the-Box Sampling Rules

TrackMe provides a set of built-in rules that are applied to the events during the data sampling process, you can access these through the sampling management screen:

About out-of-the-box rules:

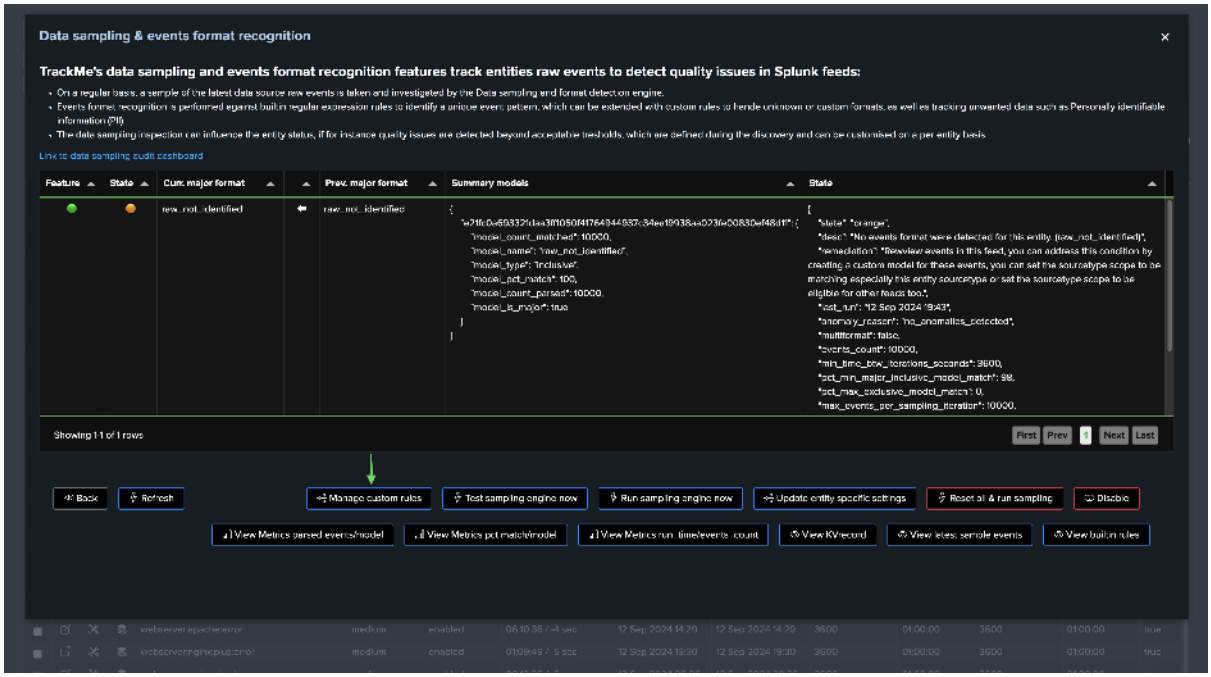
- Out-of-the-box rules are always inclusive rules, with sourcetype_scope=*.
- Out-of-the-box rules are always applied after custom rules, if any.
- For inclusive rules, the engine breaks for a given event at the first rule that matches the event, so an event can only match one rule. (with custom rules applied first)

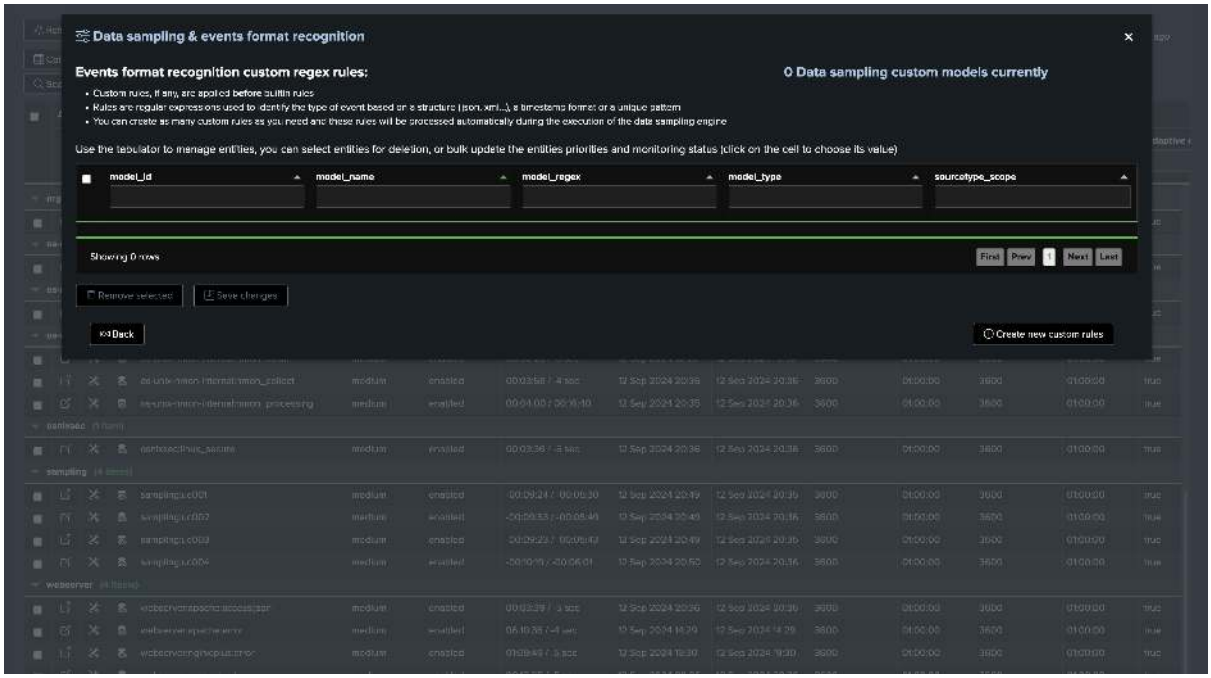




7.14.5 Custom Rules: Creating and Managing Custom Rules

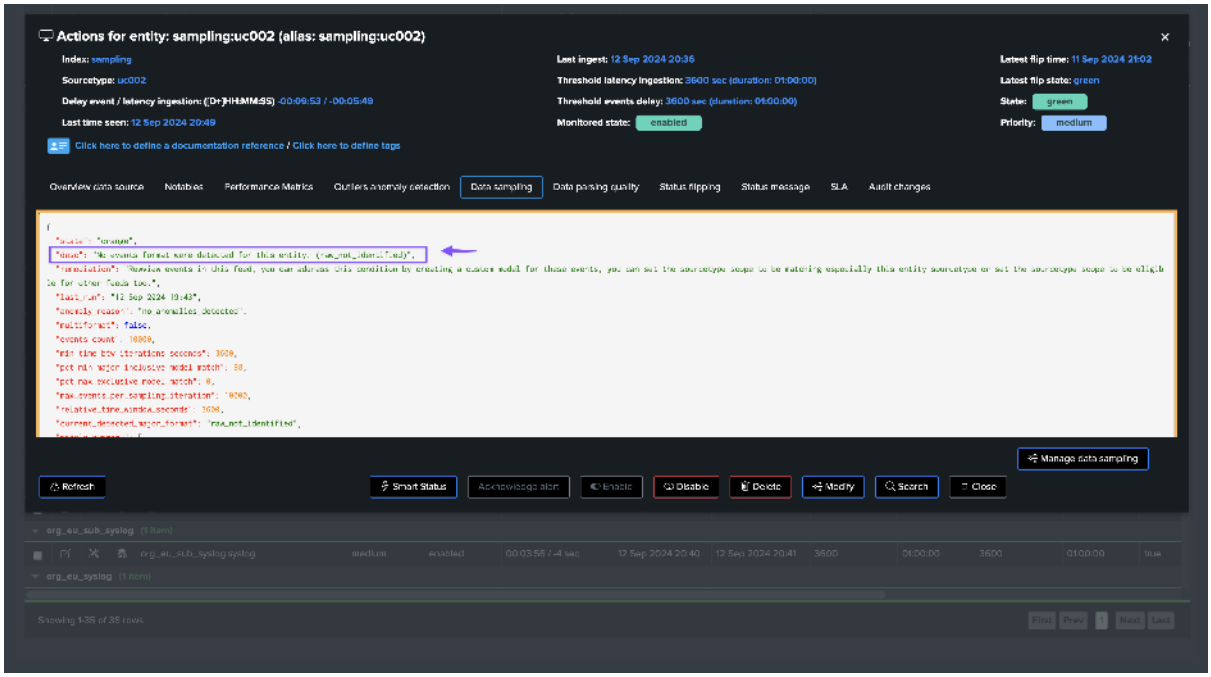
Custom rules can be created to address custom formats, or event formats which are not detected by TrackMe, you can access custom rules management from the entity screen:



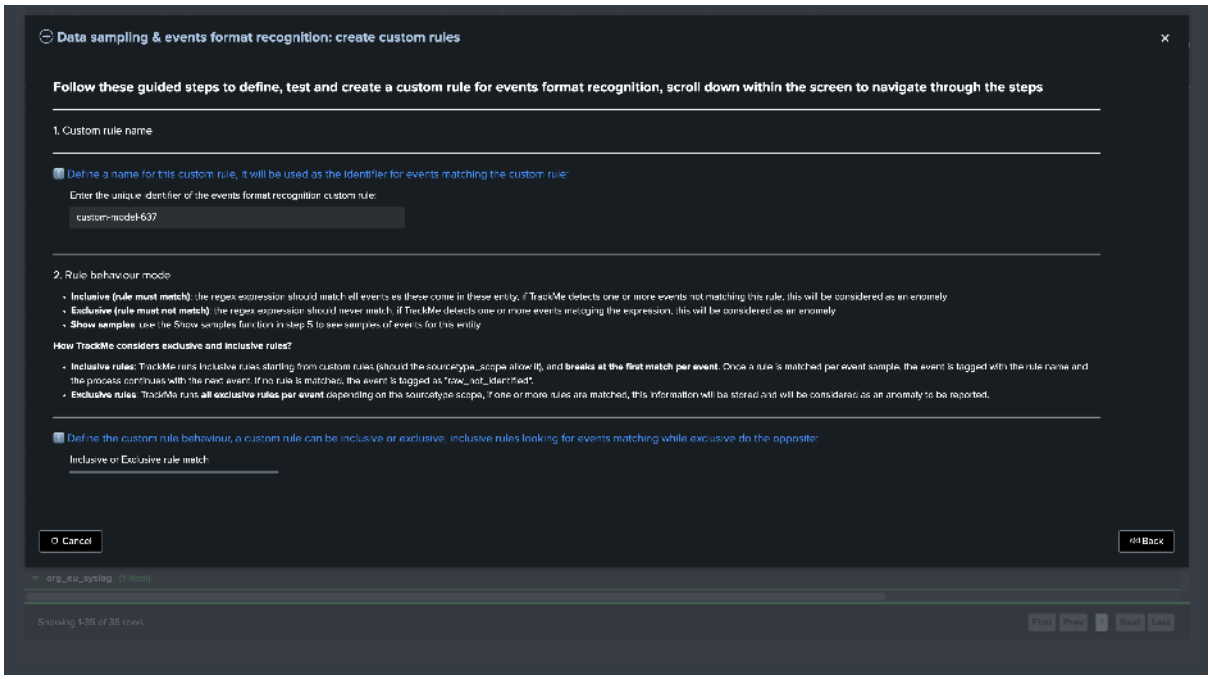


Example of Creating a Custom Rule

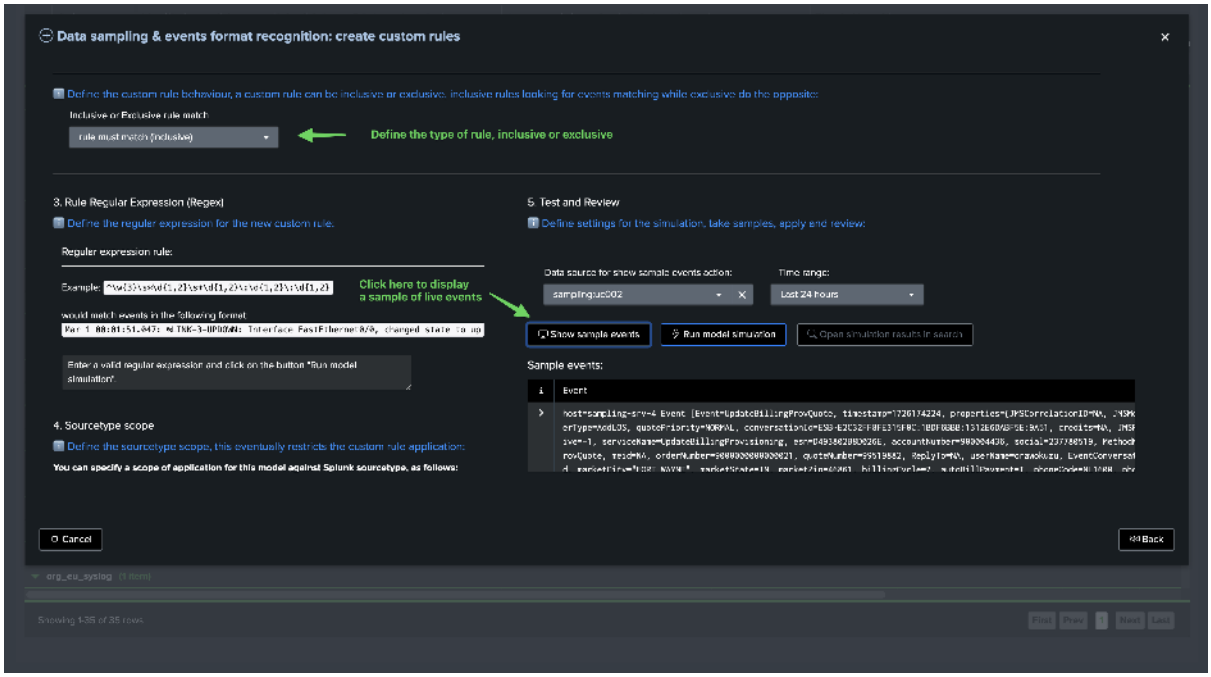
In the following example, we can see that the engine does not recognize the format of the events, and we can create a custom rule to address it, so that the engine can start tracking the format of the events:



The following screen shows the welcome start of the custom rule creation screen:



Scroll down, you can from the UI define the sourcetype scope of the custom rule, as well as displaying a sample of events, defining the regular expression of the rule and testing in real conditions the rule:

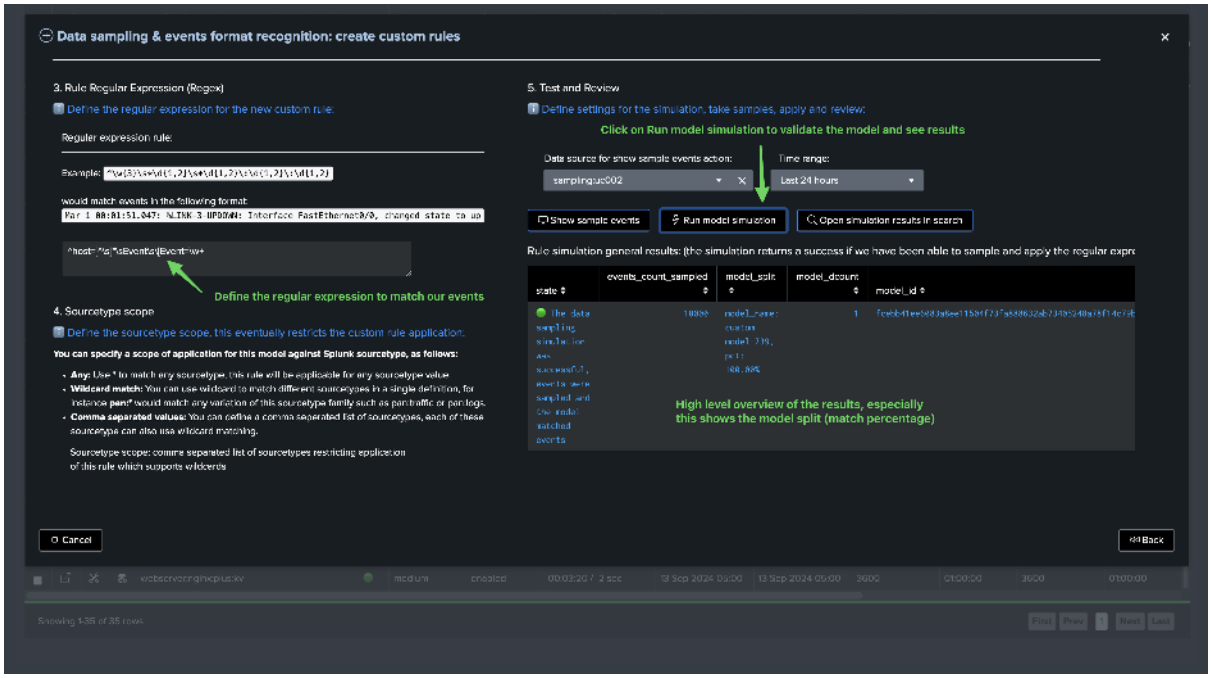


The "Show sample events" button executes underneath the following TrackMe command:

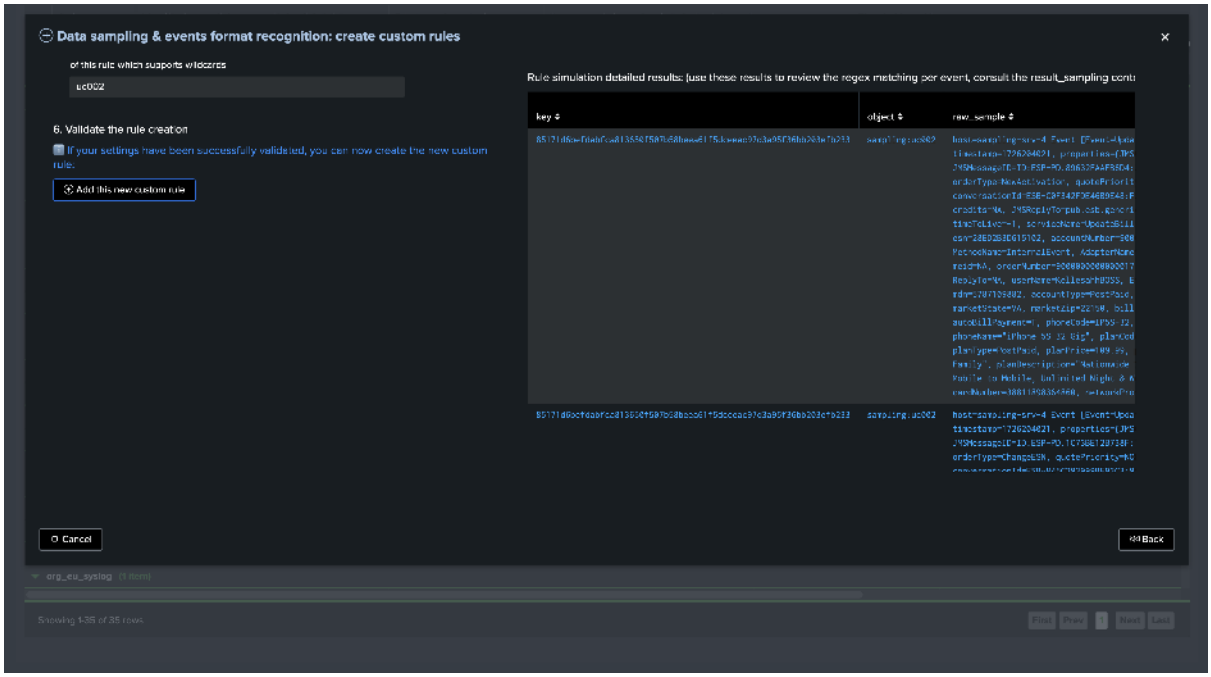
Notes: replace mytenant and myobject with the actual tenant and object values

```
| trackmesamplingexecutor tenant_id="mytenant" object="myobject" mode="get_live_
↪samples" get_samples_max_count="10000" earliest="-24h" latest="now"
```

Define the regular expression for the model and click on "Run model simulation", the top table shows the high-level overview of the model simulation and notably its matching percentage:



Scroll down to see the detailed result of the sampling exercise per event:



The overview table is powered by the following TrackMe command:

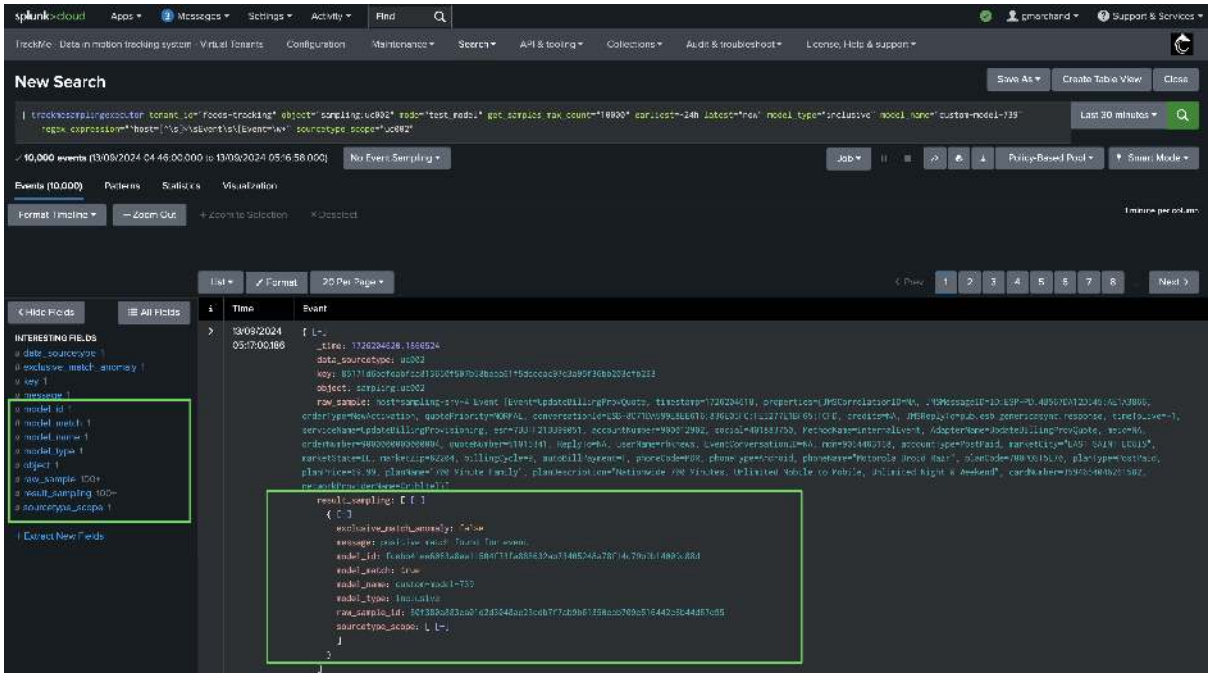
Notes: this command below shows the context for this simulation, tenant_id, object, sourcetype_scope, model_name and model_regex especially are dynamically defined based on your inputs

```
| trackmesamplingexecutor tenant_id="feeds-tracking" object="sampling:uc002" mode=
↪"test_model" get_samples_max_count="10000" earliest="-24h" latest="now" model_type=
↪"inclusive" model_name="custom-model-739" regex_expression="^host=[^\s]*\sEvent\s\
↪[Event=\w+" sourcetype_scope="uc002" | `trackme_data_sampling_format_simulation_
↪results`
```

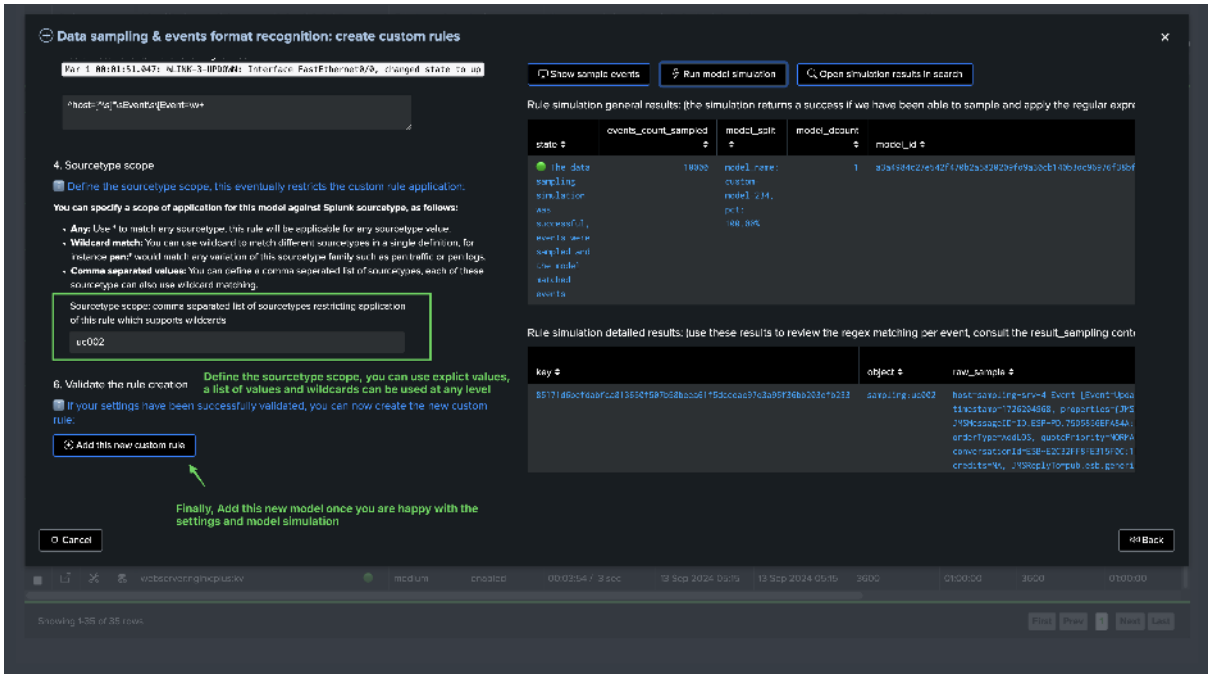
The detailed table is powered by the following TrackMe command:

```
| trackmesamplingexecutor tenant_id="feeds-tracking" object="sampling:uc002" mode=
↪"get_live_samples" get_samples_max_count="10000" earliest="-24h" latest="now"
```

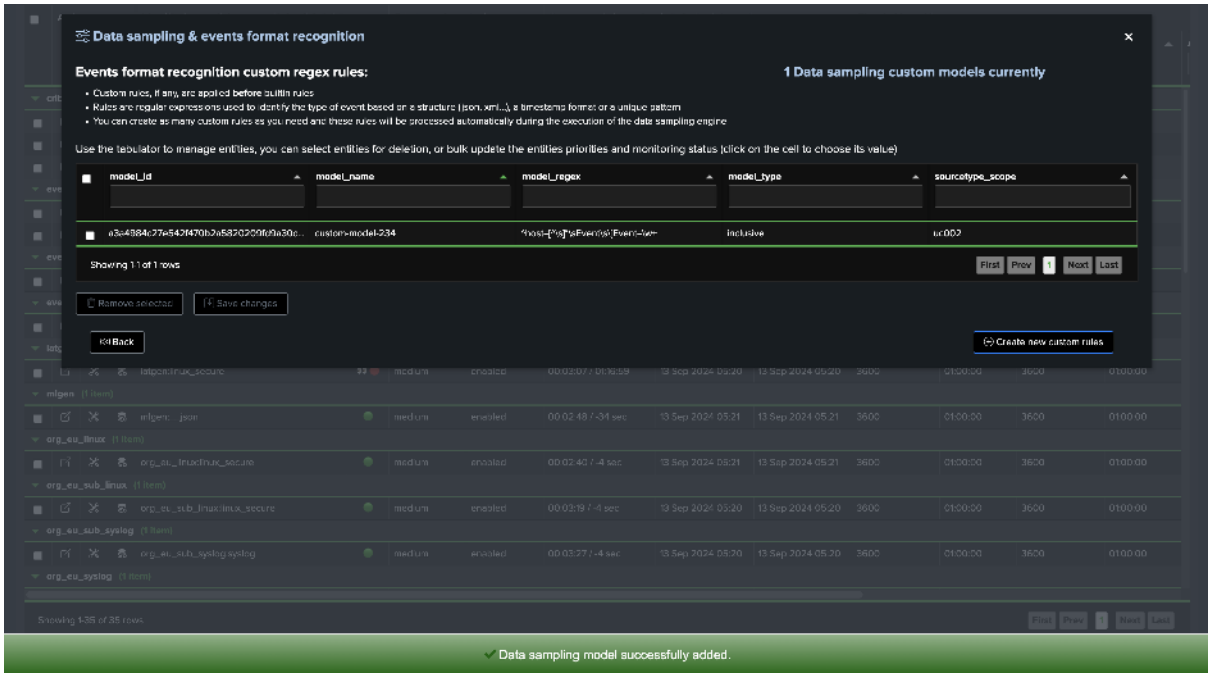
You can click on "Open simulation results in search" to review results in Splunk search UI:

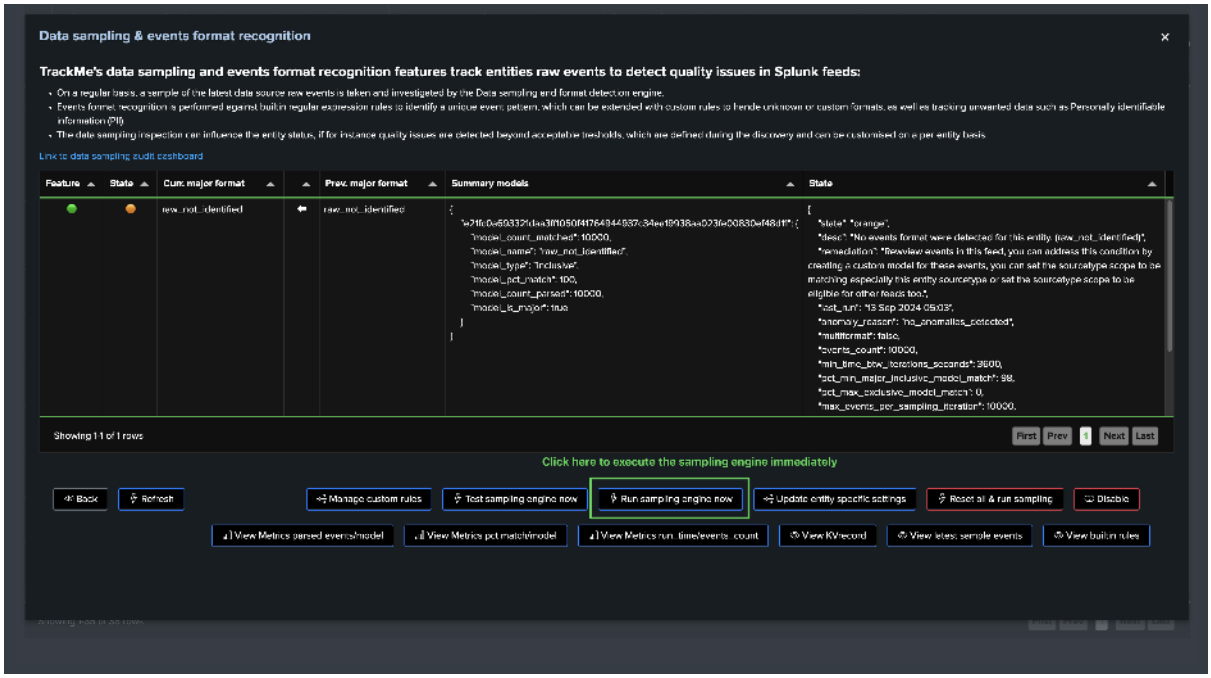


Finally, define the sourcetype scope and add the model:

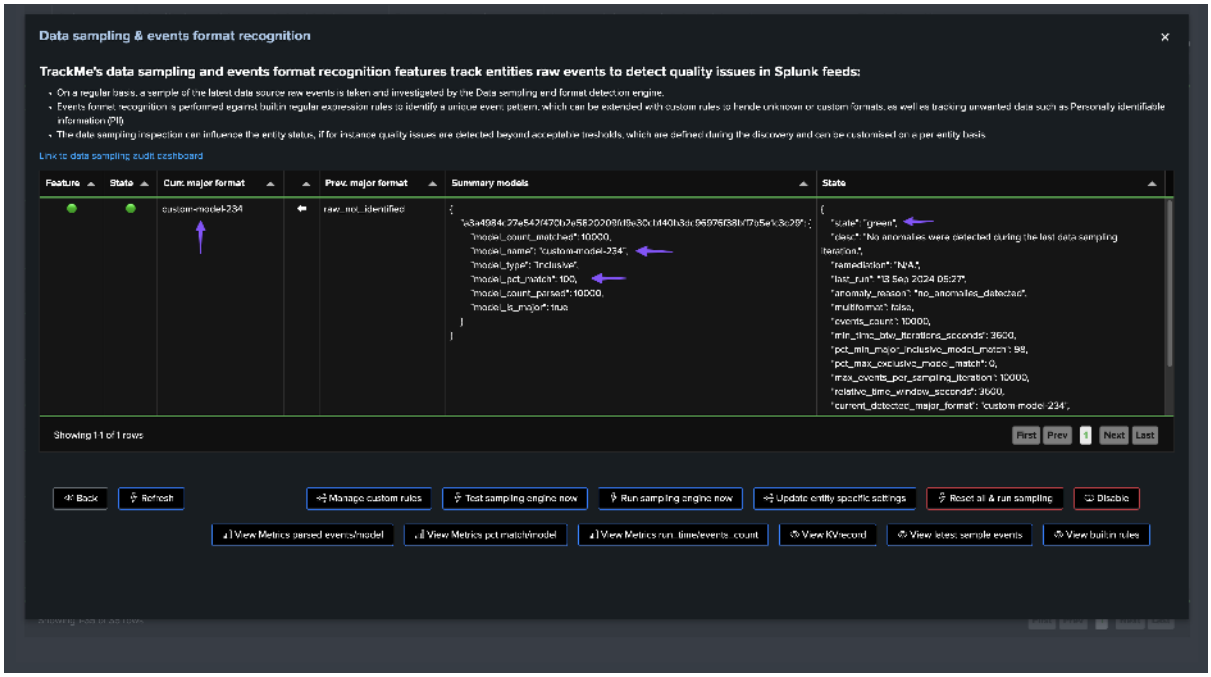


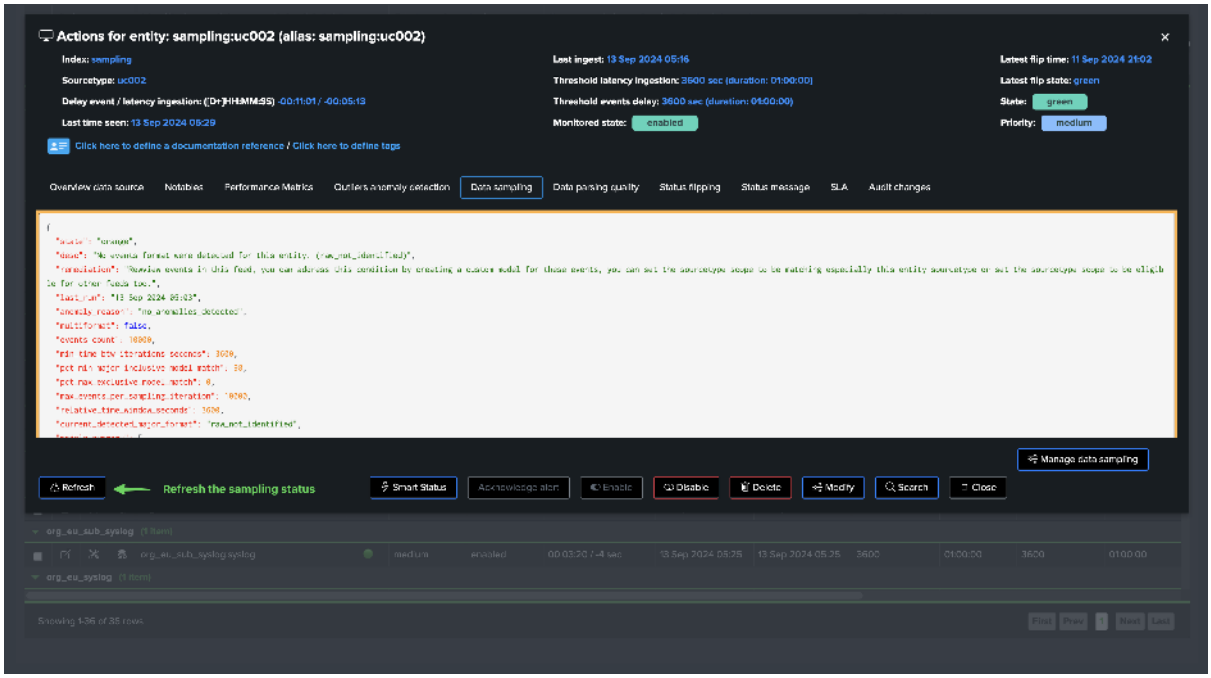
The model has been added, we can request a manual execution so the model is applied immediately:

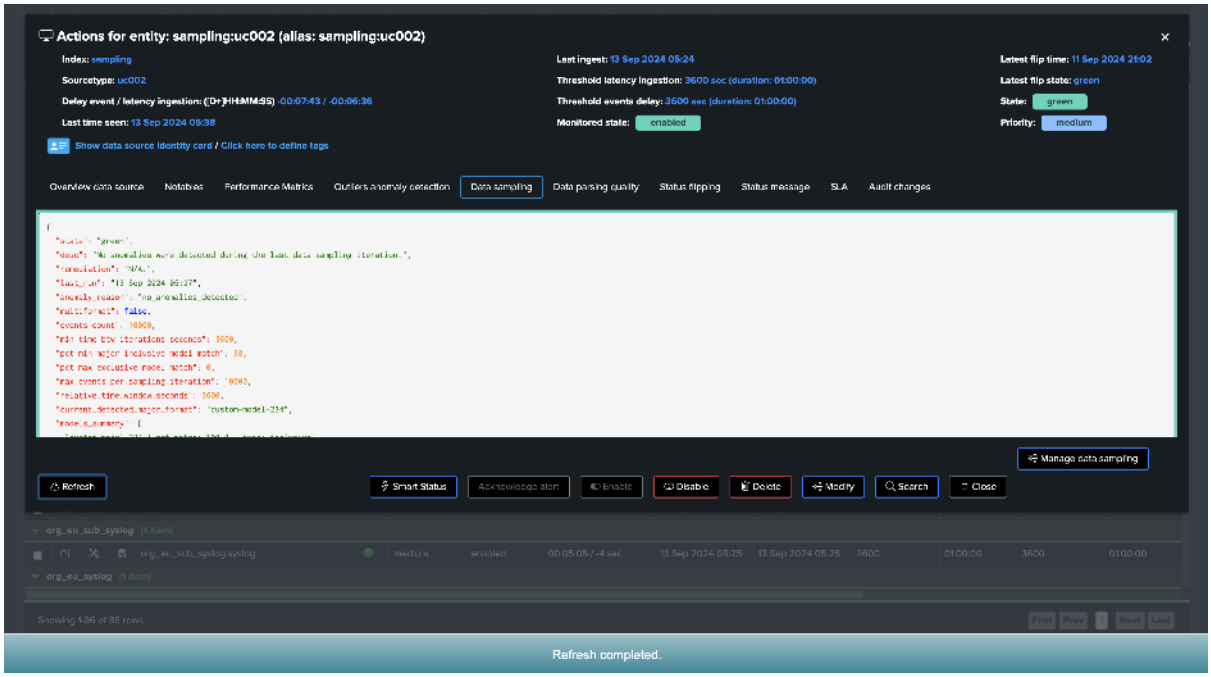




Once executed, the engine took our new model into account, the sampling results updates the status and from this stage, sampling is operational and tracking events for this entity:

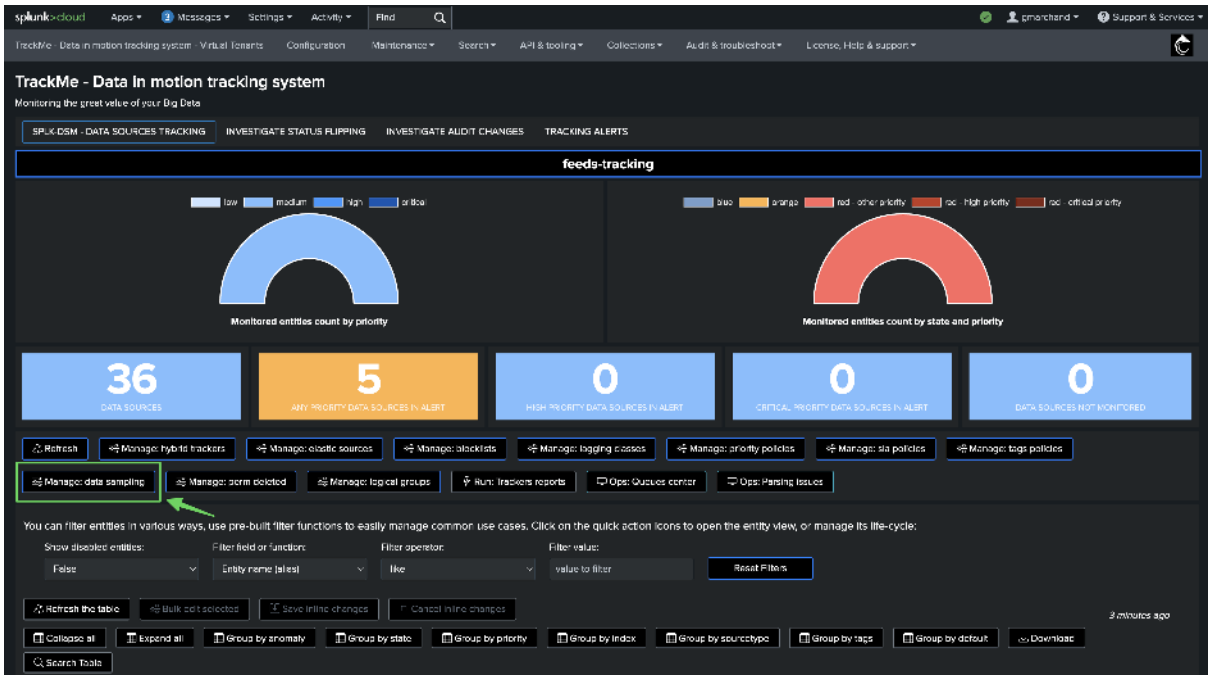

Data sampling & events format recognition
TrackMe's data sampling and events format recognition features track entities raw events to detect quality issues in Splunk feeds:
Feature
State
Curr. major format
Prev. major format
Summary models
State
custom-model-234
raw_not_identified
Showing 1 of 1 rows
Back
Refresh
Manage custom rules
Test sampling engine now
Run sampling engine now
Update entity specific settings
Reset all & run sampling
Disable
View Metrics parsed events/model
View Metrics pct match/model
View Metrics run_time/events_count
View KV record
View latest sample events
View built-in rules

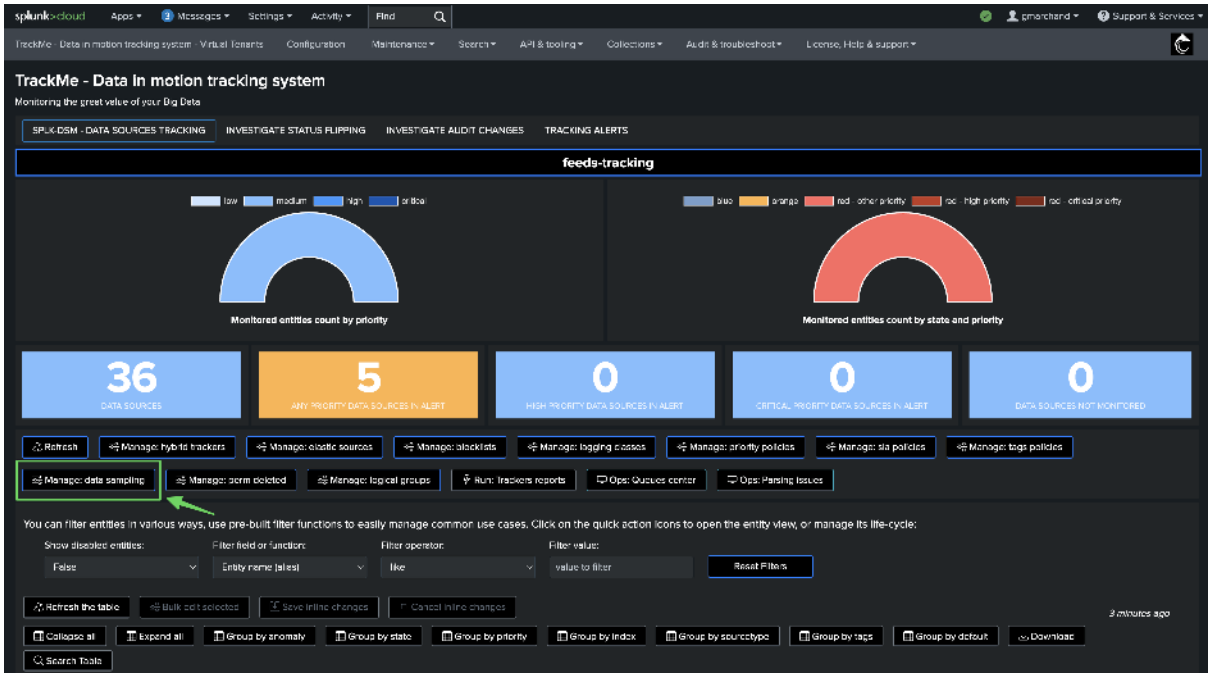

Actions for entity: sampling:uc002 (alias: sampling:uc002)
Index: sampling
Sourcetype: uc002
Last time seen: 13 Sep 2024 05:29
Last ingest: 13 Sep 2024 05:16
Threshold latency ingestion: 3600 sec (duration: 01:00:00)
Threshold events delay: 3600 sec (duration: 01:00:00)
Monitored state: enabled
Latest flip time: 11 Sep 2024 21:02
Latest flip state: green
State: green
Priority: medium
Overview data source
Notables
Performance Metrics
Outliers anomaly detection
Data sampling
Data parsing quality
Status flipping
Status message
SLA
Audit changes
Manage data sampling
Refresh
Refresh the sampling status
Smart Status
Acknowledge alert
Enable
Disable
Delete
Modify
Search
Close



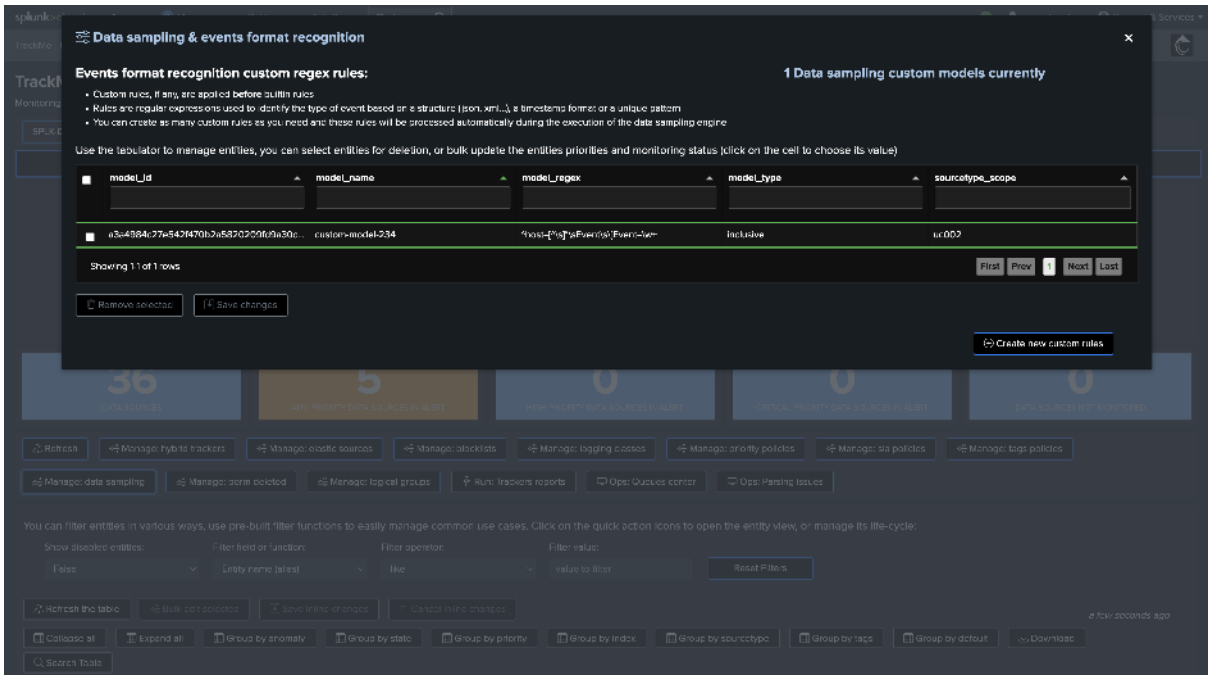
Managing Models

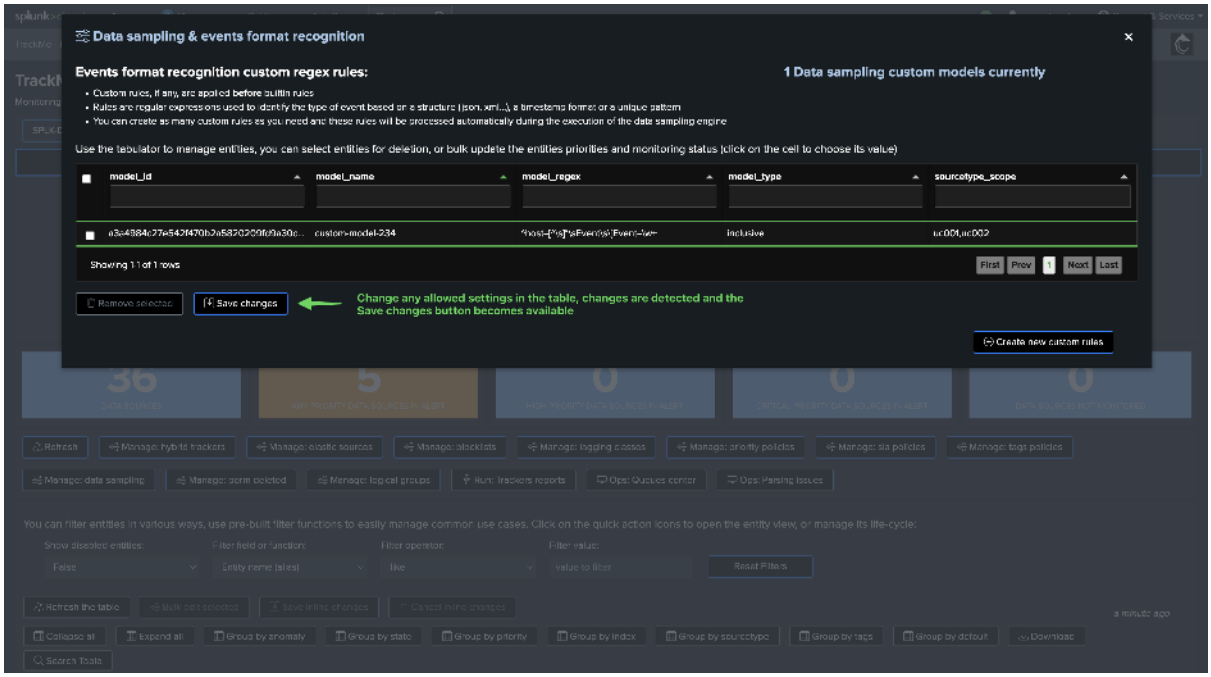
You can manage models at any time, either from the entity screen, or from the data sampling management screen:





From this screen, you can modify the model settings (sourcetype scope, regular expression), or you can delete the model:

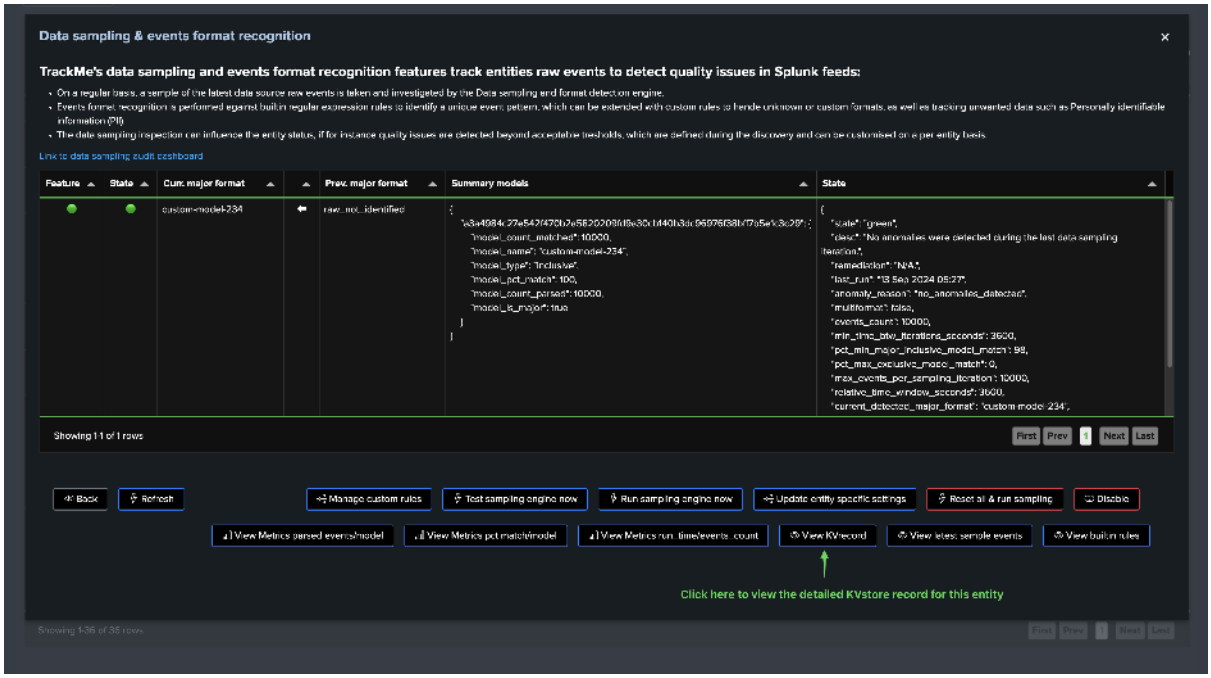


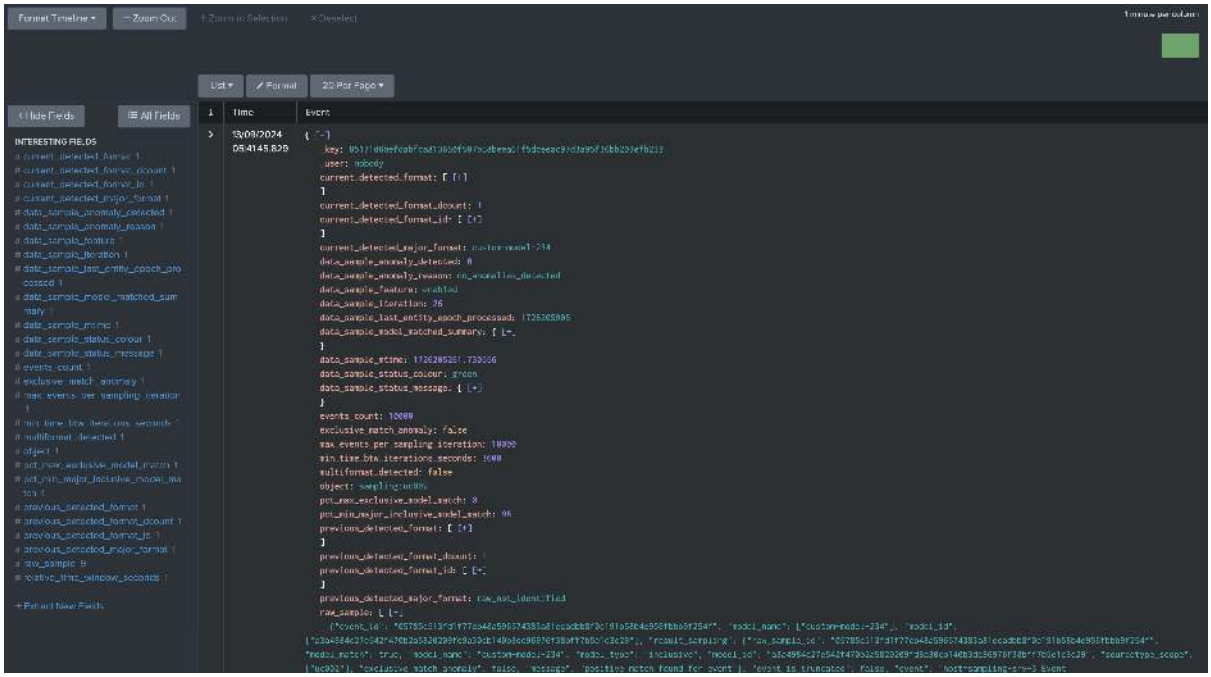


7.14.6 Entity KVstore Record and Data Sampling KVstore Collection

Entity KVrecord and Metadata Fast Access

The sampling engine stores and maintains entity-level Metadata as well as a subset of the sampled events for investigation purposes, you can easily access the KVrecord via the UI:



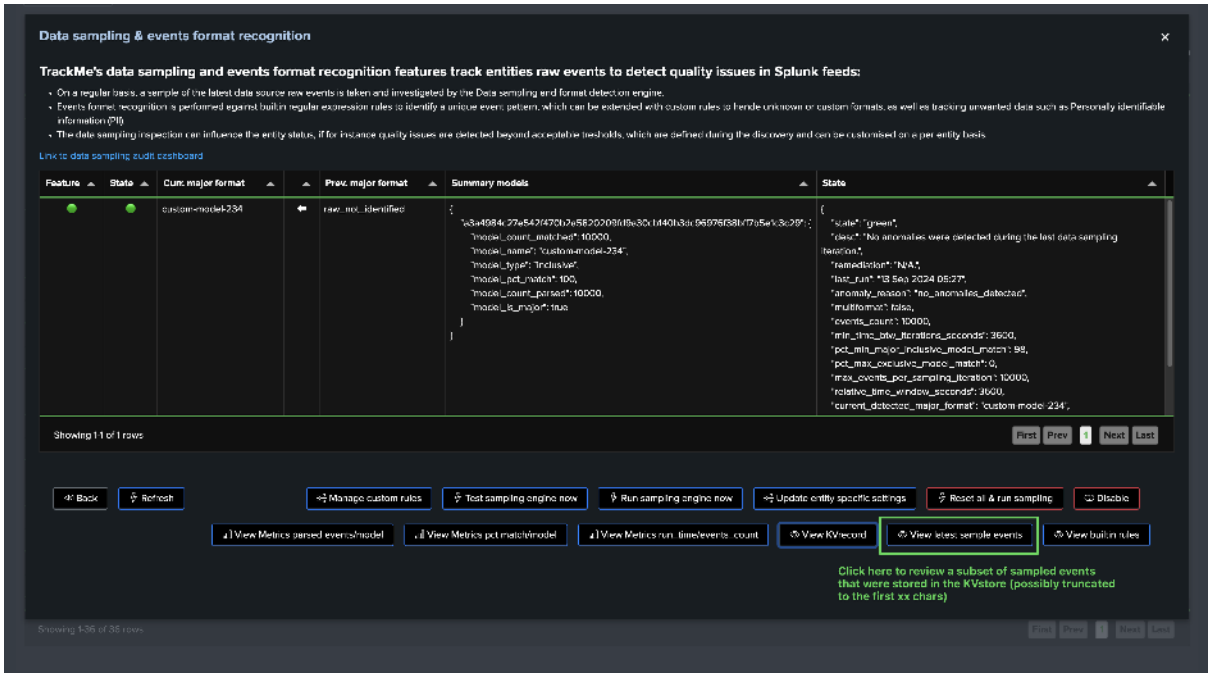


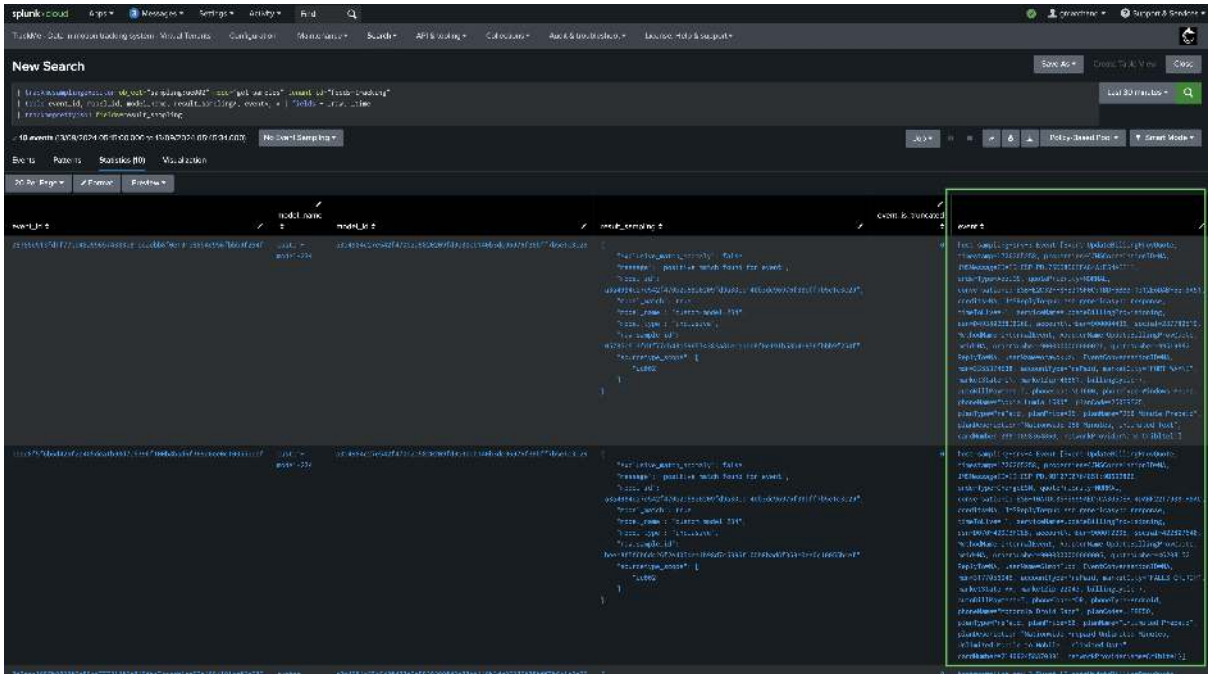
We can see that the engine stores and maintains a subset of sampled events, in fact by default it will take a subset of 10 events per model matched, and it will truncate these events to the first 40k characters, you can also access the stored samples via the button “Show samples”

Hint

About number of events stored in the KVstore collection and truncation:

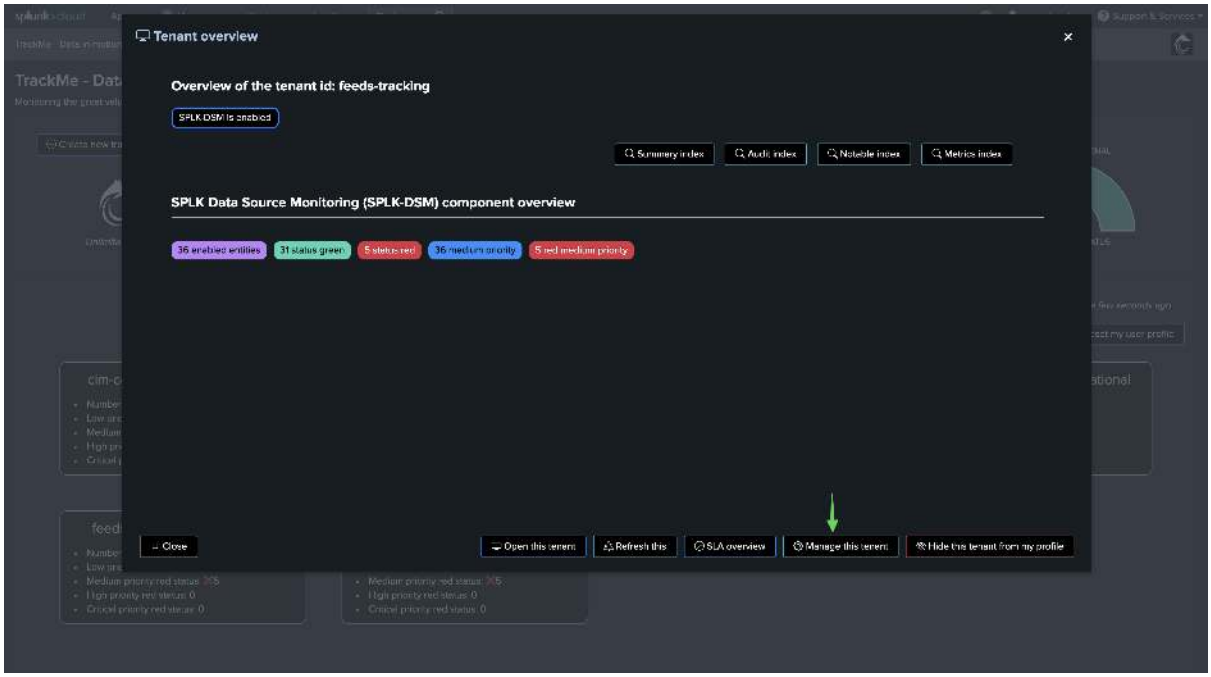
- The number of events stored in the KVstore collection per entity and per model matched is defined by the system-wide setting “splk_data_sampling_no_records_saved_kvrecord”.
- The truncation is only applied when the engine stores events in the KVstore for investigation purposes, not when the regular expression is applied to the event.
- The truncation limit is defined by the system-wide setting “splk_data_sampling_records_kvrecord_truncate_size”.

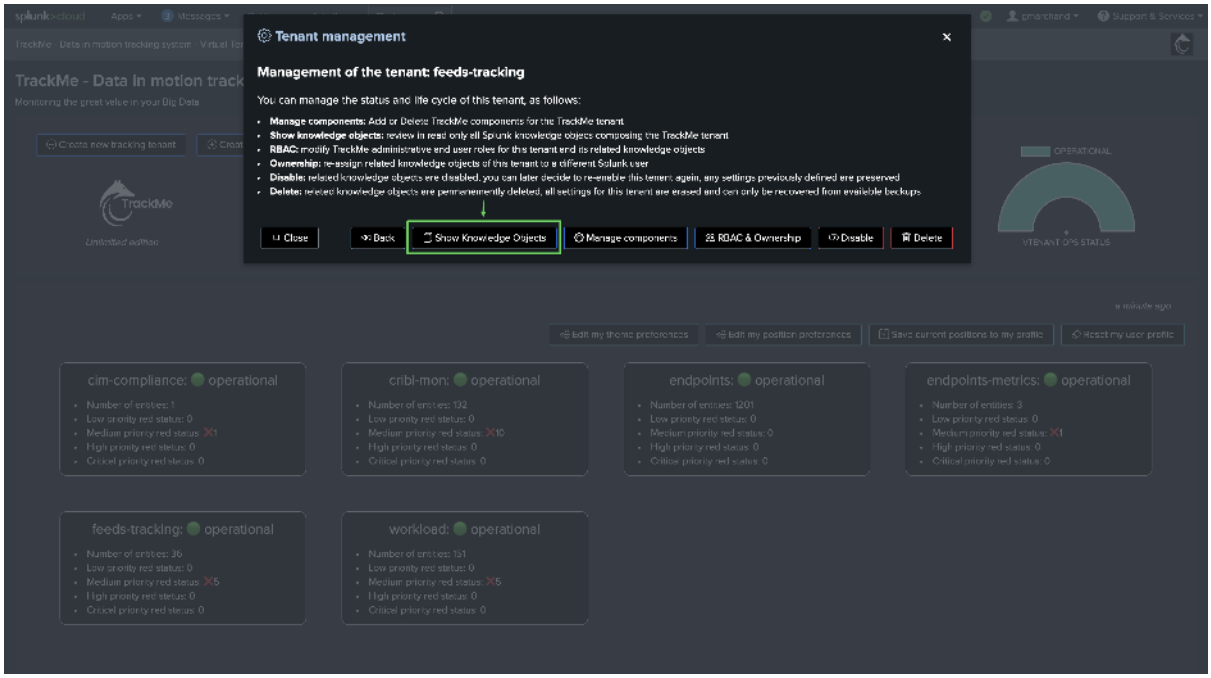


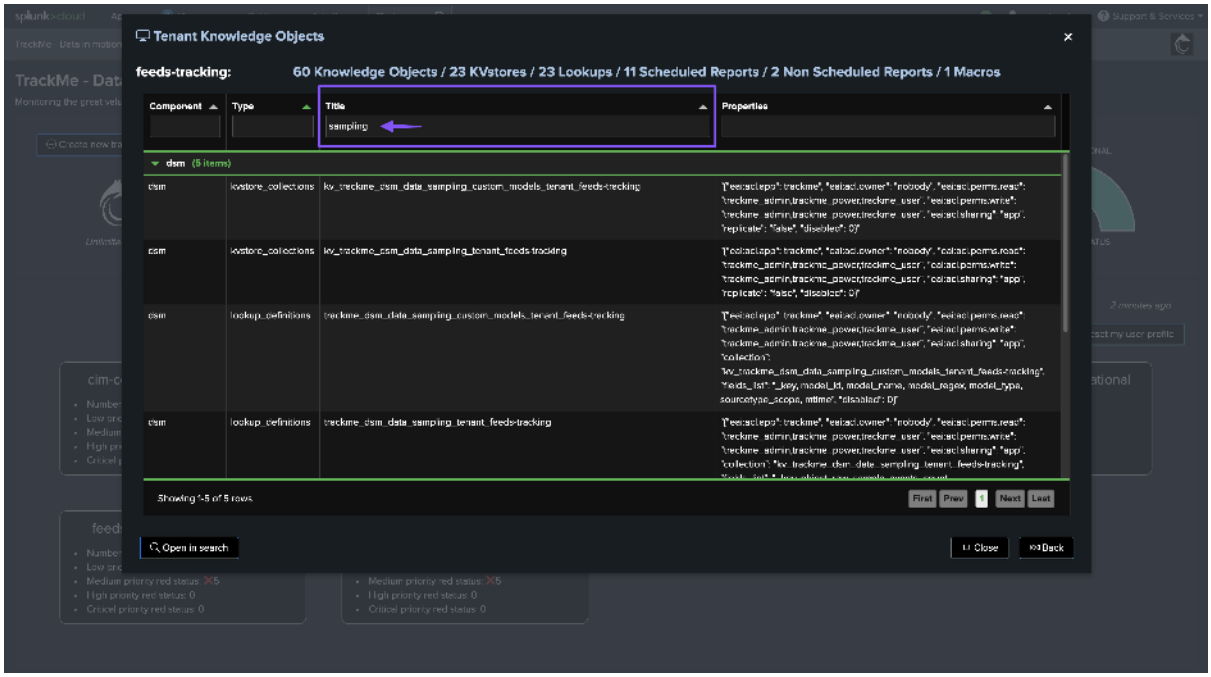


Data Sampling KVstore Collection

The data sampling KVstore collection is created when the Virtual Tenant is created, you can find out the KVstore collection and the transforms name easily from the Virtual Tenant UI:


Tenant overview
Overview of the tenant id: feeds-tracking
SPLK DSM is enabled
Summary index
Audit index
Notable index
Metrics index
SPLK Data Source Monitoring (SPLK-DSM) component overview
Close
Open this tenant
Refresh this
SLA overview
Manage this tenant
Hide this tenant from my profile


Tenant management
Management of the tenant: feeds-tracking
You can manage the status and life cycle of this tenant, as follows:
Manage components: Add or Delete TrackMe components for the TrackMe tenant
Show knowledge objects: review in read only all Splunk knowledge objects composing the TrackMe tenant
RBAC: modify TrackMe administrative and user roles for this tenant and its related knowledge objects
Ownership: re-assign related knowledge objects of this tenant to a different Splunk user
Disable: related knowledge objects are disabled, you can later decide to re-enable this tenant again, any settings previously defined are preserved
Delete: related knowledge objects are permanently deleted, all settings for this tenant are erased and can only be recovered from available backups
Close
Back
Show Knowledge Objects
Manage components
RBAC & Ownership
Disable
Delete
cim-compliance: operational
cribl-mon: operational
endpoints: operational
endpoints-metrics: operational
feeds-tracking: operational
workload: operational



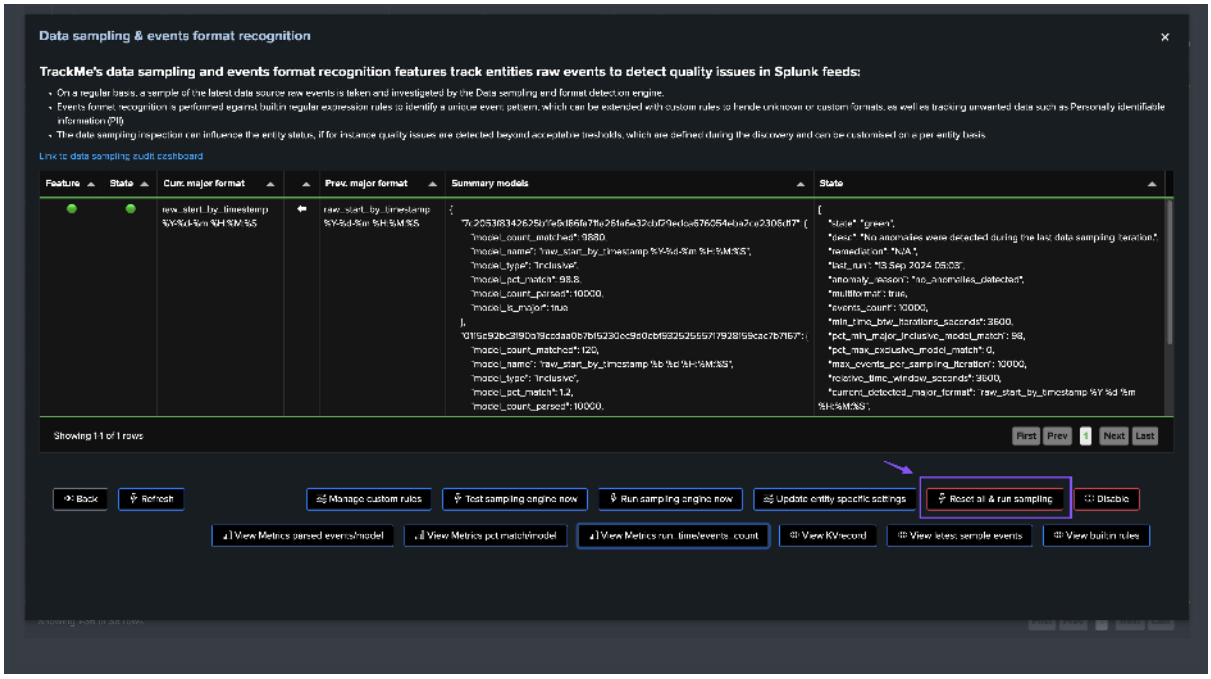
The KVstore collection transforms definition is honoring the following convention:

```
| inputlookup trackme_dsm_data_sampling_tenant_<tenant_id> | eval key=_key
```

7.14.7 Resetting Data Sampling Status and Settings for a Given Entity

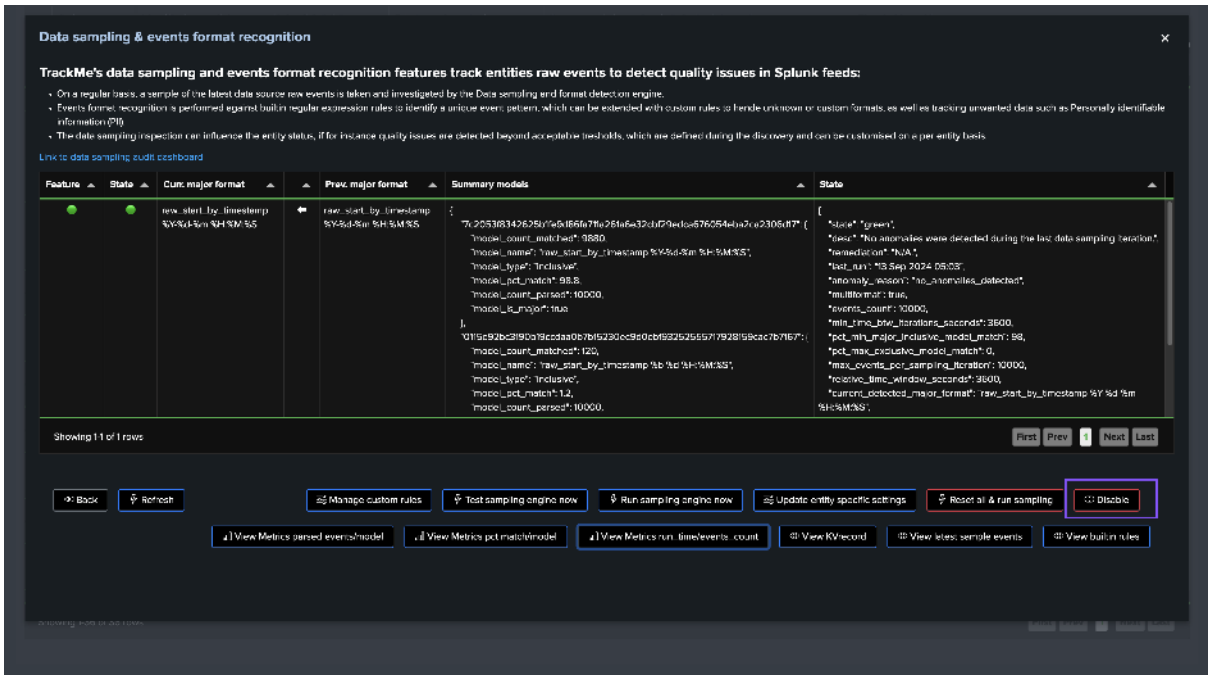
Reset Status & Settings for a Given Entity

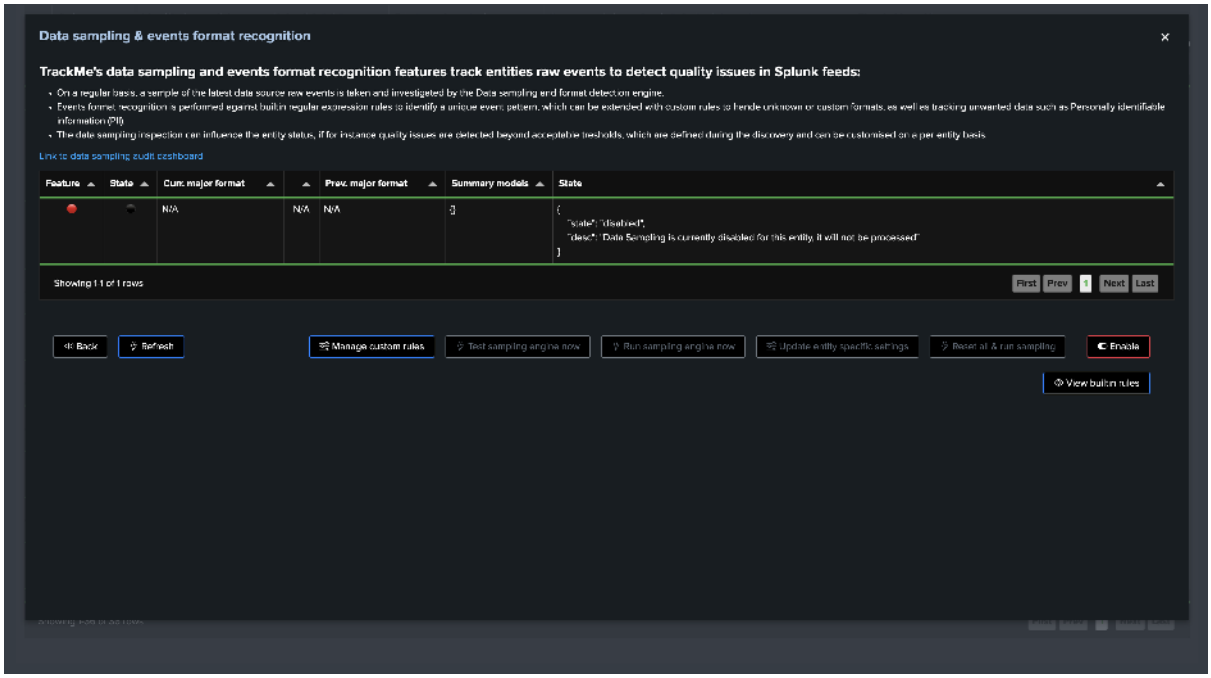
You can reset the data sampling status and settings for a given entity, this will reset the data sampling status, the models, and the settings to the default values, you can access this feature from the entity screen:

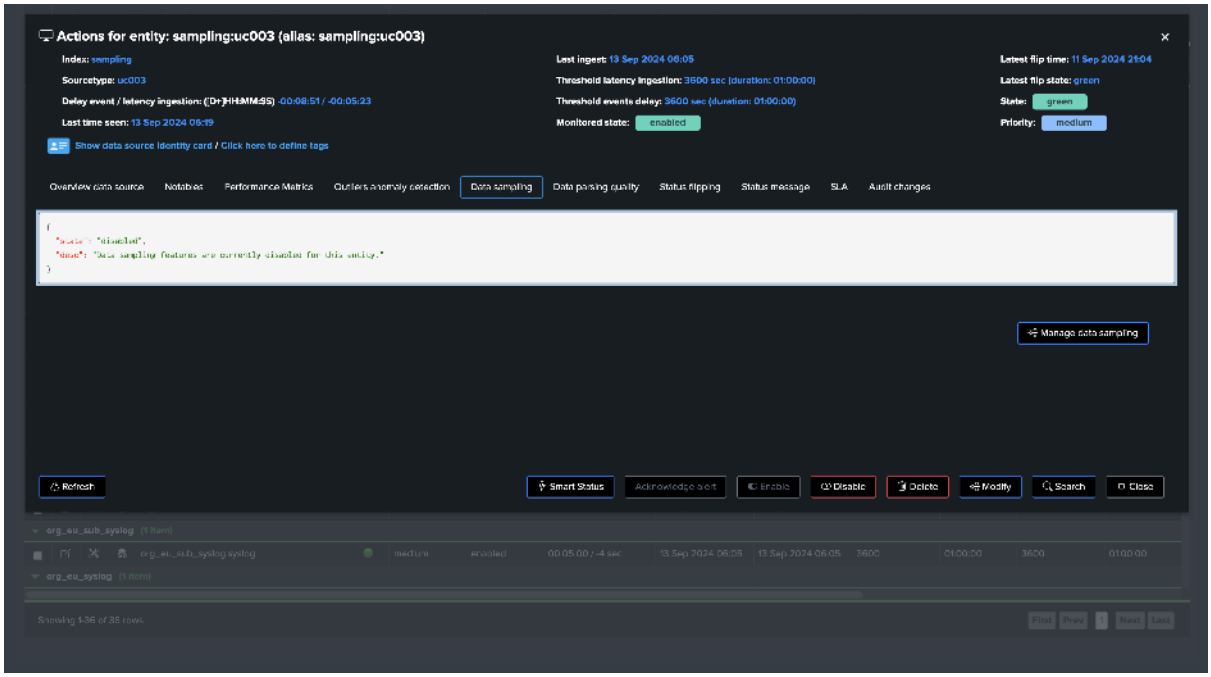


Disable Sampling for a Given Entity

You can disable the data sampling feature for a given entity, the sampling feature will be entirely disabled and the engine will never attempt to process sampling for it:

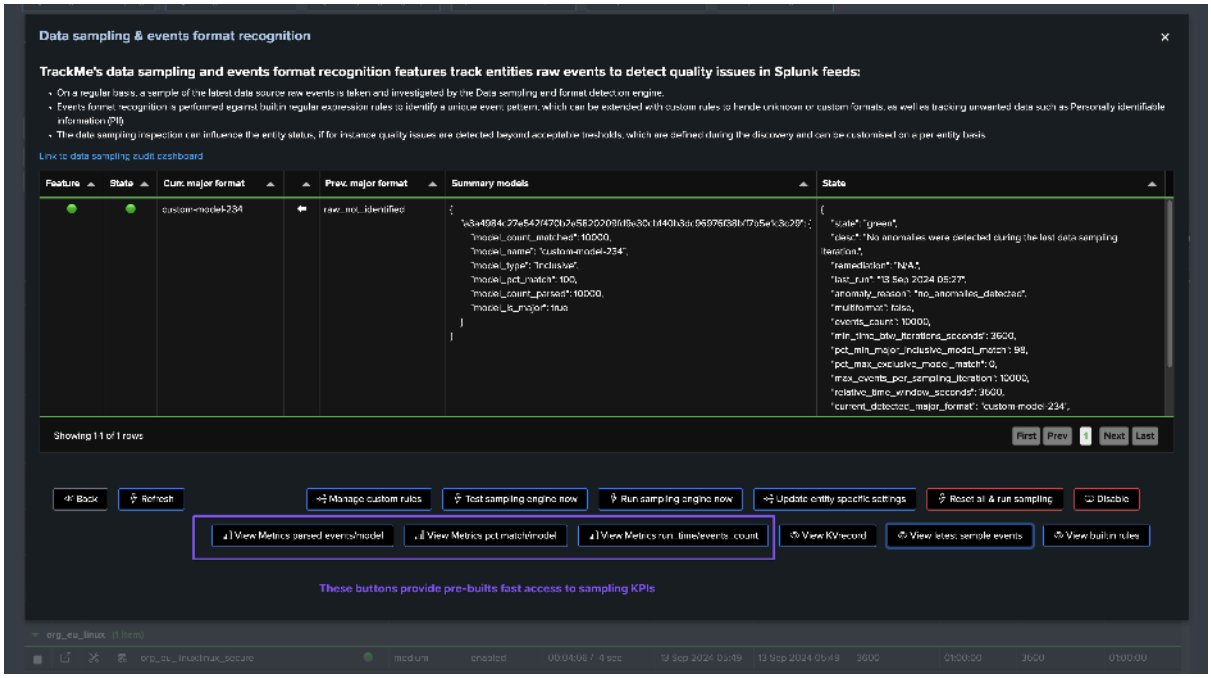




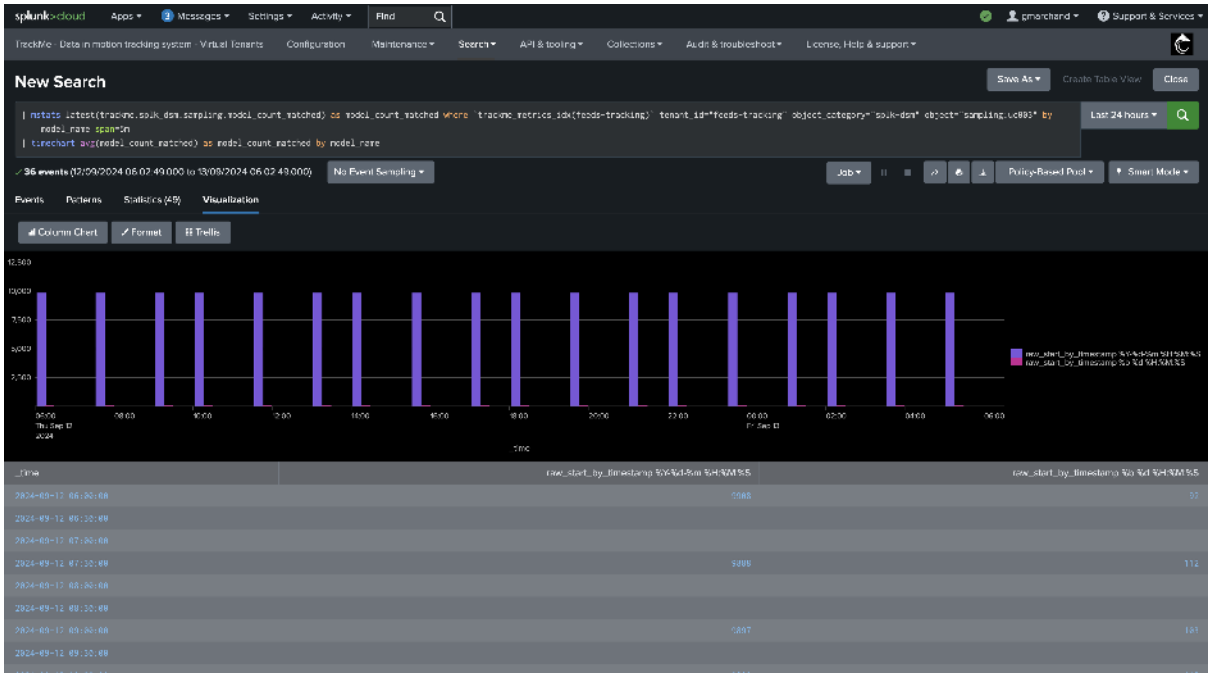


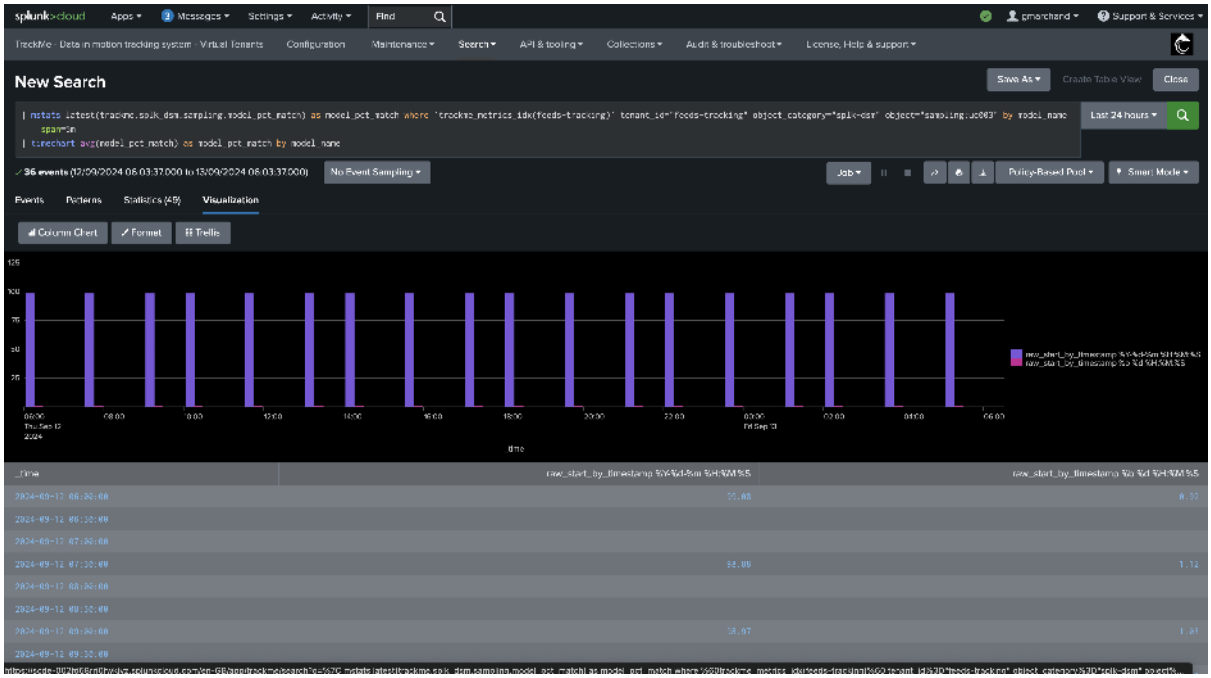
7.14.8 KPIs and Metrics

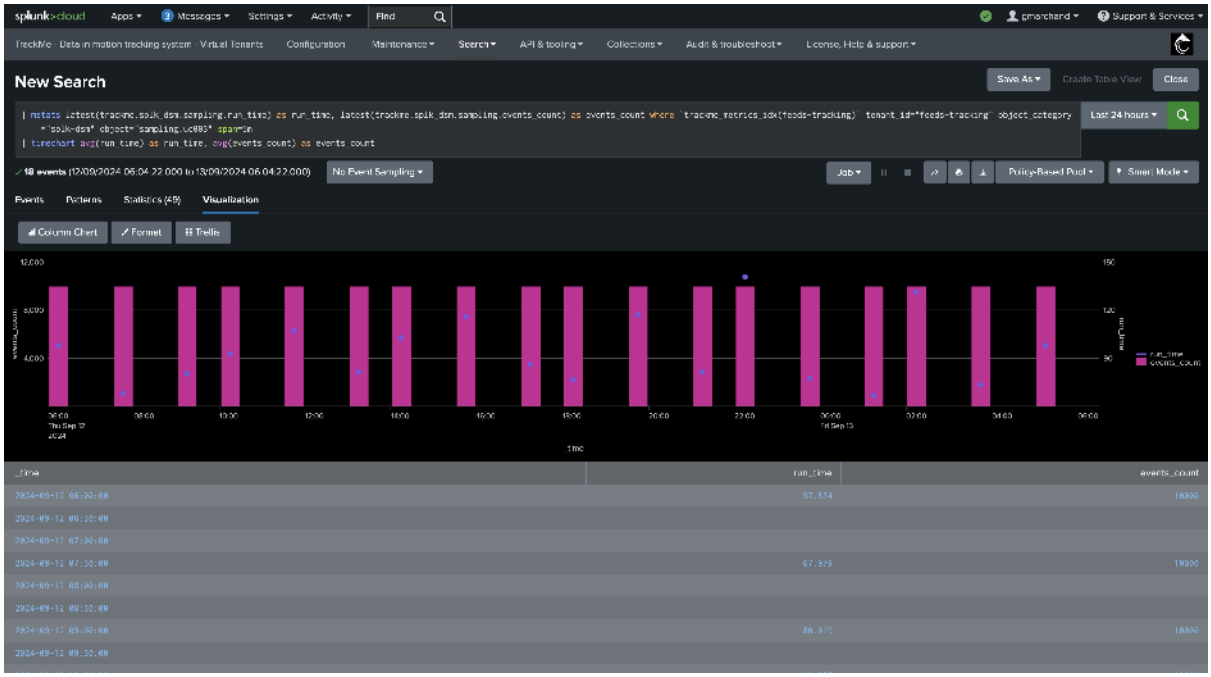
The data sampling engine generates various KPIs and metrics, which are stored in the tenant metrics index, you can access these metrics via the UI:



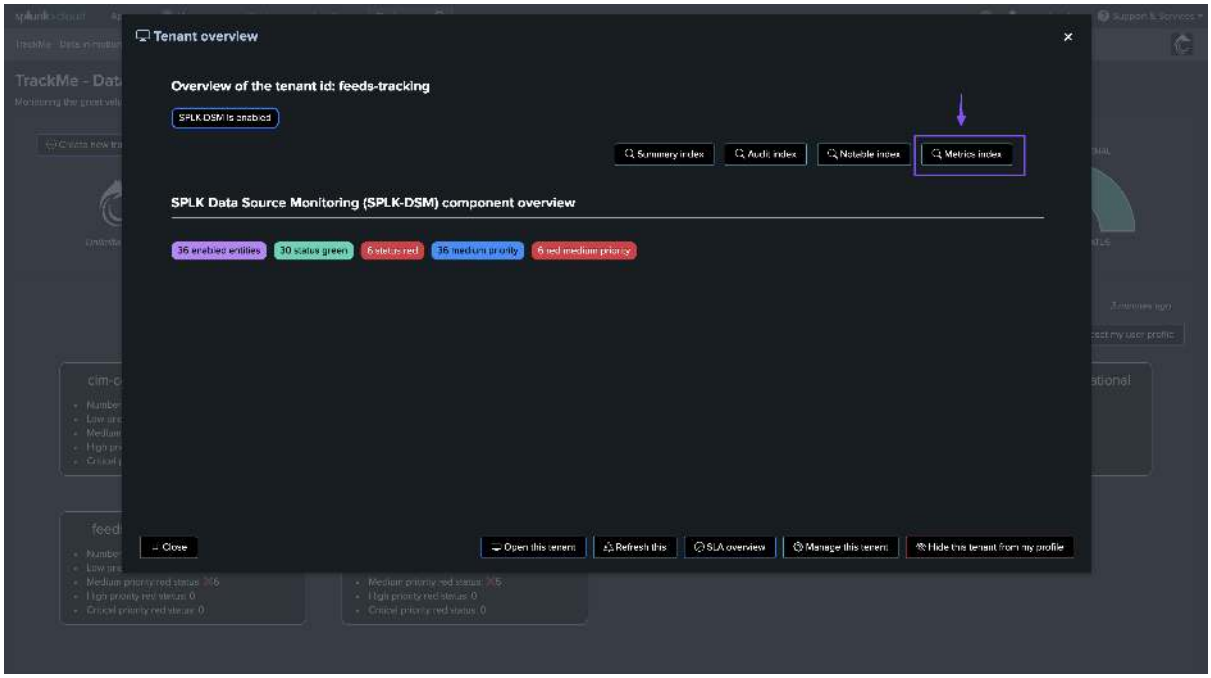
Examples:


splunk>cloud
New Search
Save As
Create Table View
Close
Last 24 hours
Visualization
Column Chart
Format
Trellis
time


splunk>cloud
New Search
Save As
Create Table View
Close
Last 24 hours
Visualization
Column Chart
Format
Trellis
time

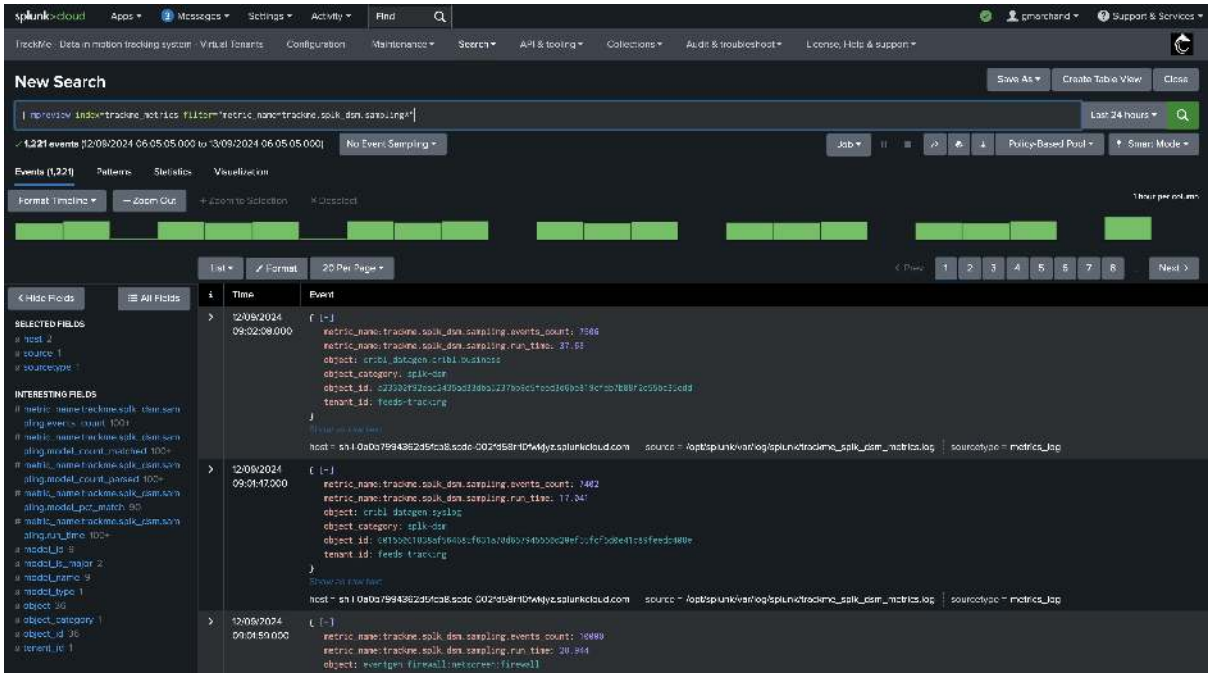


The Virtual Tenants UI provides a quick mpreview access to the tenant-related metrics:



You can also use the following generic mpreview command to access the metrics: (adapt the index name if needed)

```
| mpreview index=trackme_metrics filter="metric_name=trackme.splk_dsm.sampling*"
```



7.14.9 Troubleshooting

Sampling Logging

Data Sampling tracker logs can be inspected through the following search:

```
index=_internal sourcetype=trackme:custom_commands:trackmesamplingexecutor
```

Any errors, unexpected exceptions can be found easily using the log_level:

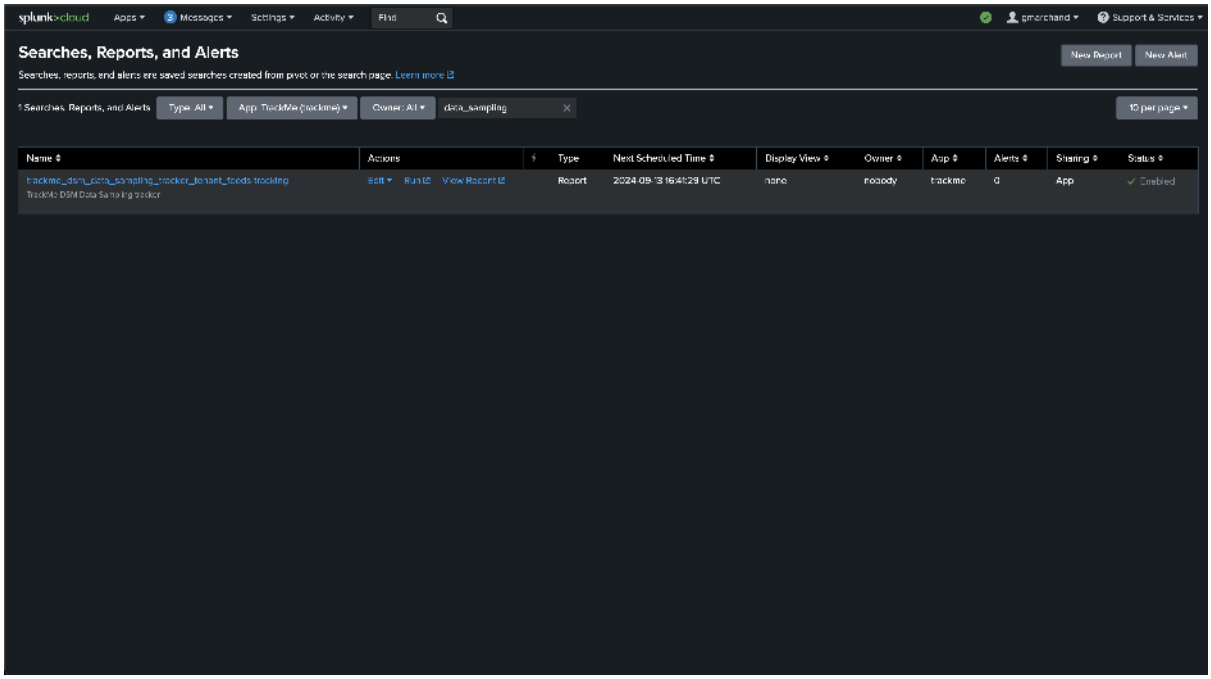
```
index=_internal sourcetype=trackme:custom_commands:trackmesamplingexecutor log_
↪level=error
```

Activity related to a specific Splunk feed, a TrackMe entity, can easily be inspected by filtering on the object value:

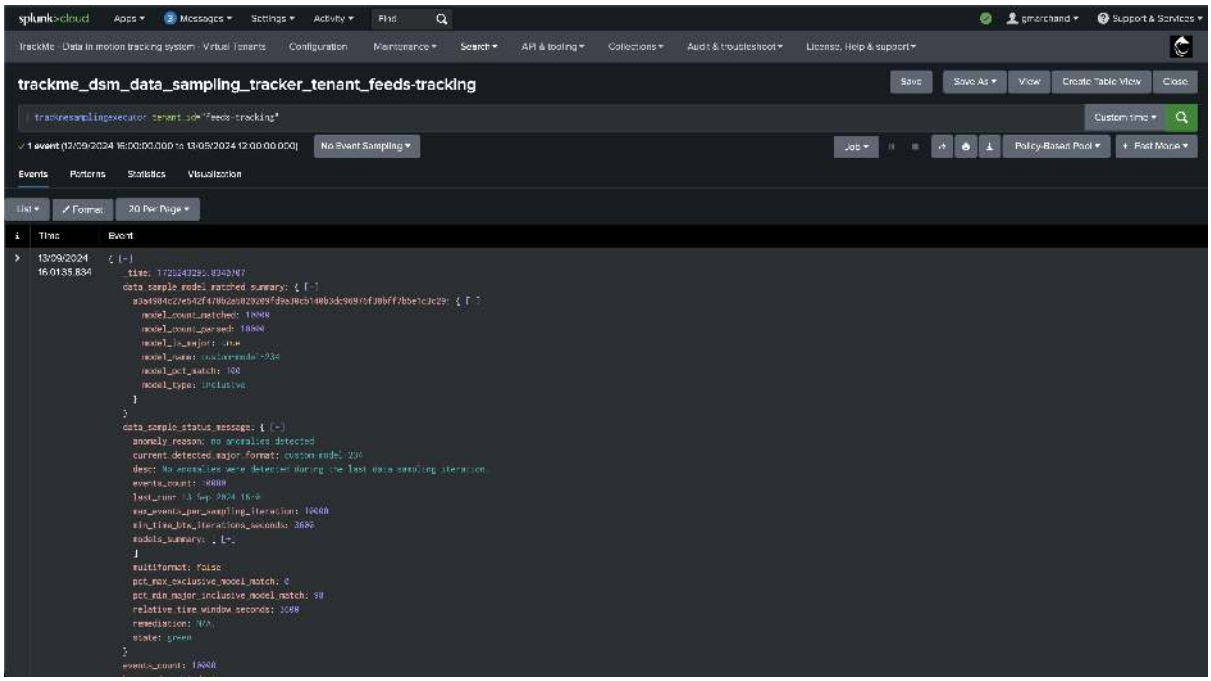
```
index=_internal sourcetype=trackme:custom_commands:trackmesamplingexecutor object="
↪<object value>"
```

Inspecting the Data Sampling Tracker Execution

In Splunk, you can easily review the data sampling scheduled tracker, you can access the previous artifacts, or manually execute the job in case of issues:



Each entity processed results in one JSON event with all the details of the data sampling status:



7.14.10 Annexes

Annexe 1: System-wide Settings

Value	Description	Default Value
splk_data_samp	This defines the minimal time in seconds between data sampling iterations. TrackMe will never attempt to perform sampling for a given entity if the time since the last iteration is lower than this value.	3600
splk_data_samp	Defines the number of records to be sampled and verified per entity during the data sampling process. Increasing this value improves event format recognition but requires more processing per entity.	10000
splk_data_samp	Defines the number of records kept in the KVstore for inspection purposes at each iteration. This sample is stored for inspection and review, and increasing this value requires more KVstore storage.	10
splk_data_samp	Defines the character size limit before truncating events when storing sampled records in the KVstore for inspection. This truncation only affects storage and does not impact the model training process.	40000
splk_data_samp	Defines the minimum percentage of events that must match the major inclusive model. If the main model has less than this percentage of matching events, the entity's state will be impacted.	98
splk_data_samp	Defines the maximum percentage of events matching an exclusive model that can be accepted. By default, no events matching an exclusive model are accepted, but this can be increased per entity.	95
splk_data_samp	Defines the size of the time window for sampling operations in seconds, relative to the latest event time known for the entity. This time window is used to calculate the earliest time for sampling searches.	3600

7.15 Disruption Queue

About the disruption queue concept in TrackMe

- The disruption queue is a feature available for **all components** and all types of entities, this feature was made available in TrackMe 2.1.18.
- This feature allows you to define a period of time in seconds that must be spent before an entity anomaly is considered.
- The **minimal disruption period** is therefore a period of **continuous time** of disruption before we allow an entity to transition to an **alerting state**. (red)
- During this intermediate state, the entity transitions to a **blue state**.
- Once the **disruption period is over**, and if the anomaly persisted, the entity transitions to a **red state**.
- The disruption queue can be leveraged to avoid or reduce the risk of **false positives**, with short life-time anomalies.

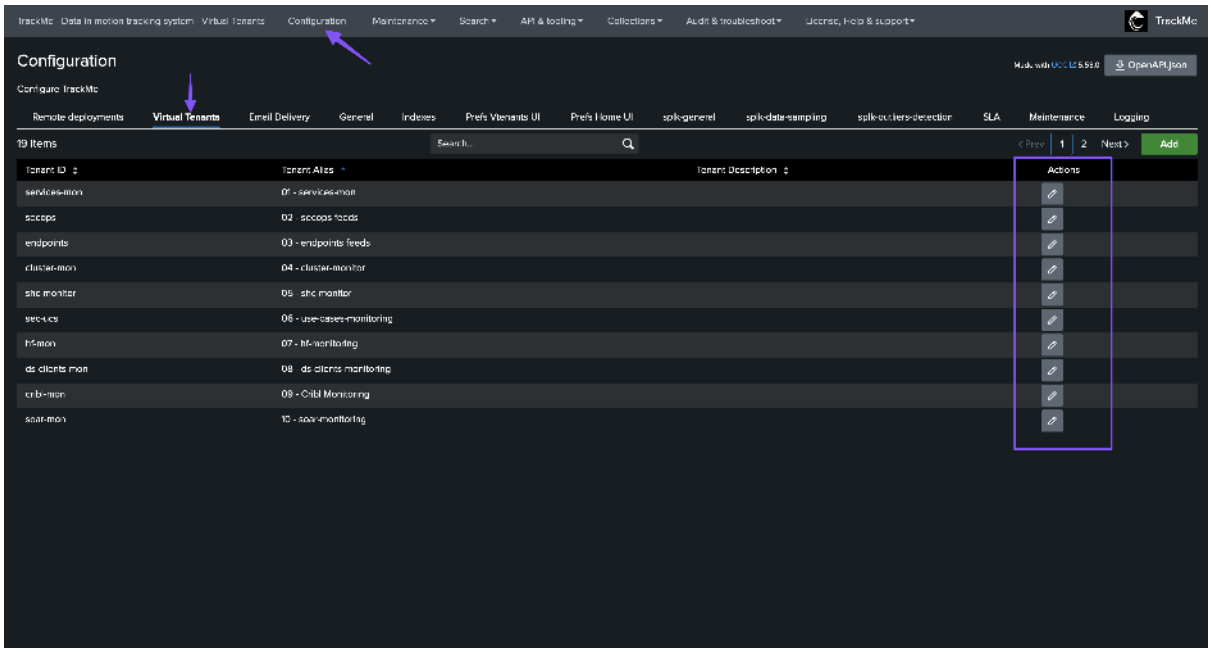
7.15.1 Setting up the disruption queue at the level of the Virtual Tenant (entities discovery)

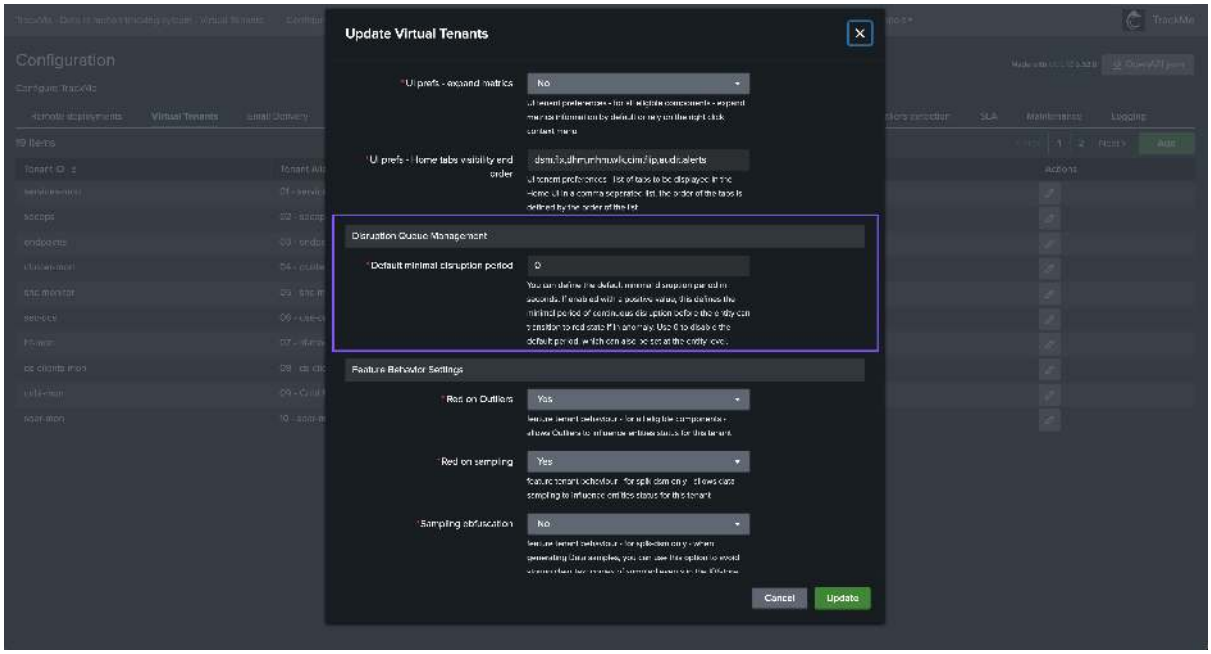
The disruption queue can be configured at the level of the Virtual Tenant, this will apply to all entities discovered in this Virtual Tenant.

Hint

About setting up the disruption queue at the level of the Virtual Tenant:

- If defined at the level of the Virtual Tenant, this disruption queue will be defined for all existing entities and entities to be discovered.
- A given entity can still be updated to have a different disruption queue configuration, this will override the Virtual Tenant configuration.





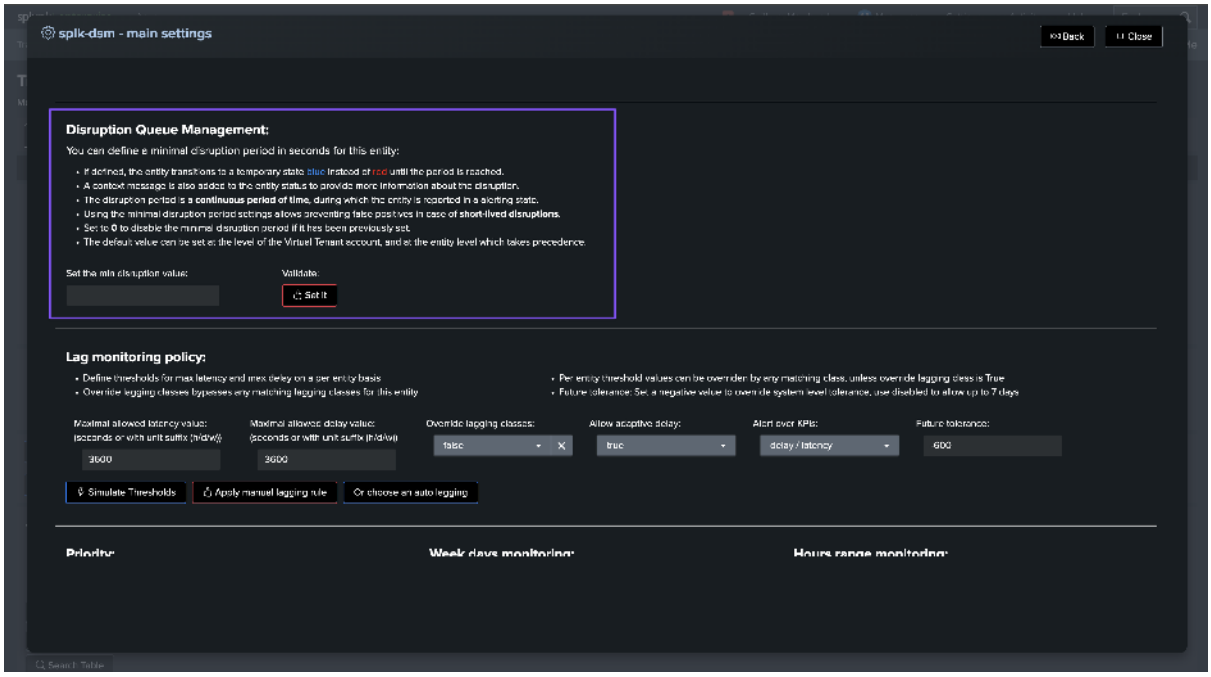
7.15.2 Setting up the disruption queue on a per entity basis

The disruption queue can be configured on a per entity basis through the TrackMe UI and the entity configuration page.

Hint

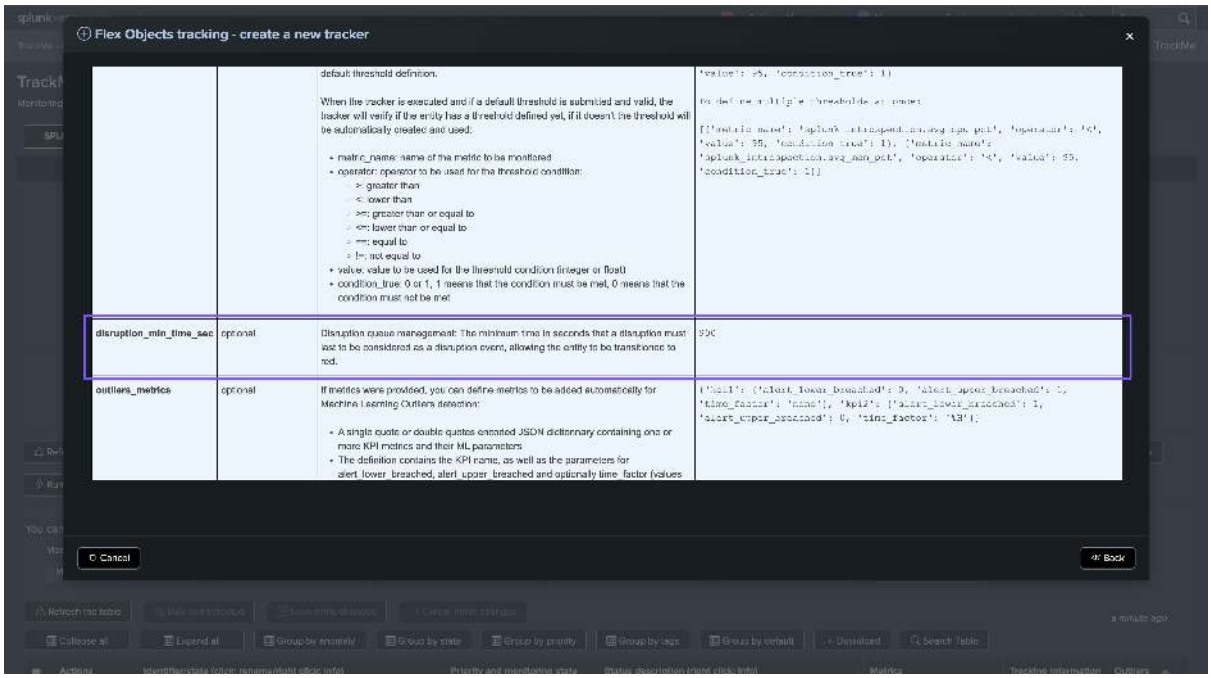
Entity level has precedence over Virtual Tenant level:

- If defined at the level of the entity, this disuption queue will override the Virtual Tenant configuration.
- If no configuration is defined at the level of the entity, the Virtual Tenant configuration will be used. (if configured)



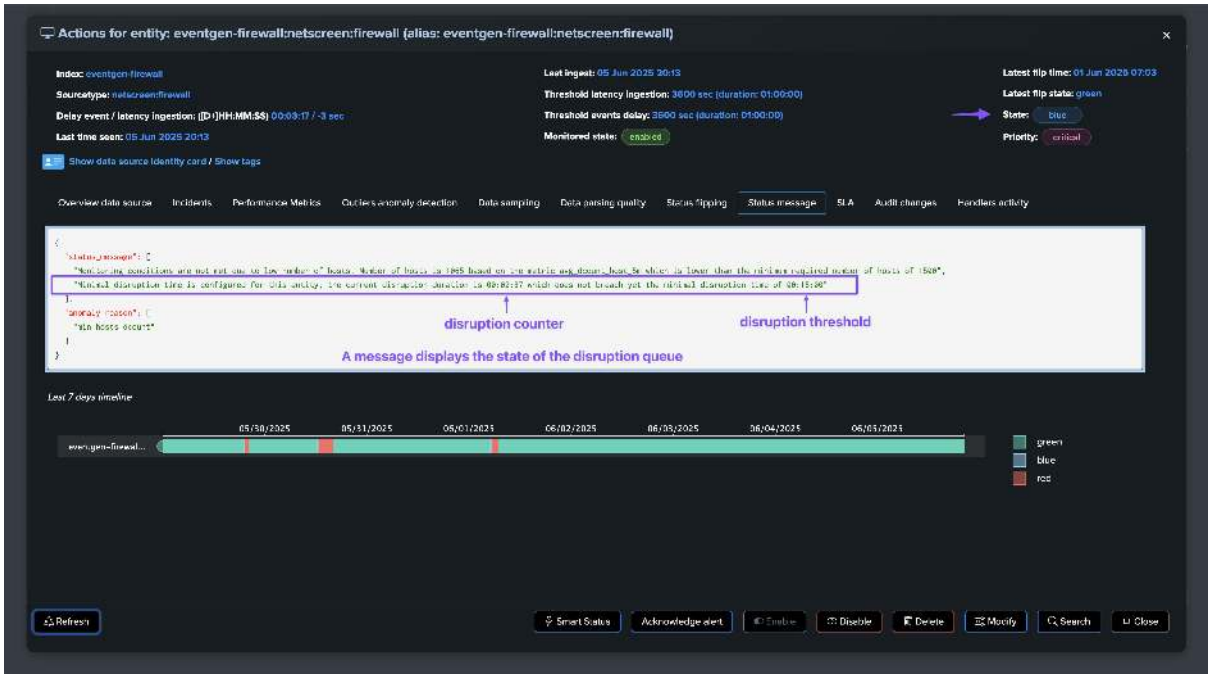
7.15.3 Flex Object specific: setting up the disruption queue at the level of the Flex Object tracker

Especially for the Flex Object component (splk-flx), you can define a default disruption queue for entities associated with this tracker:

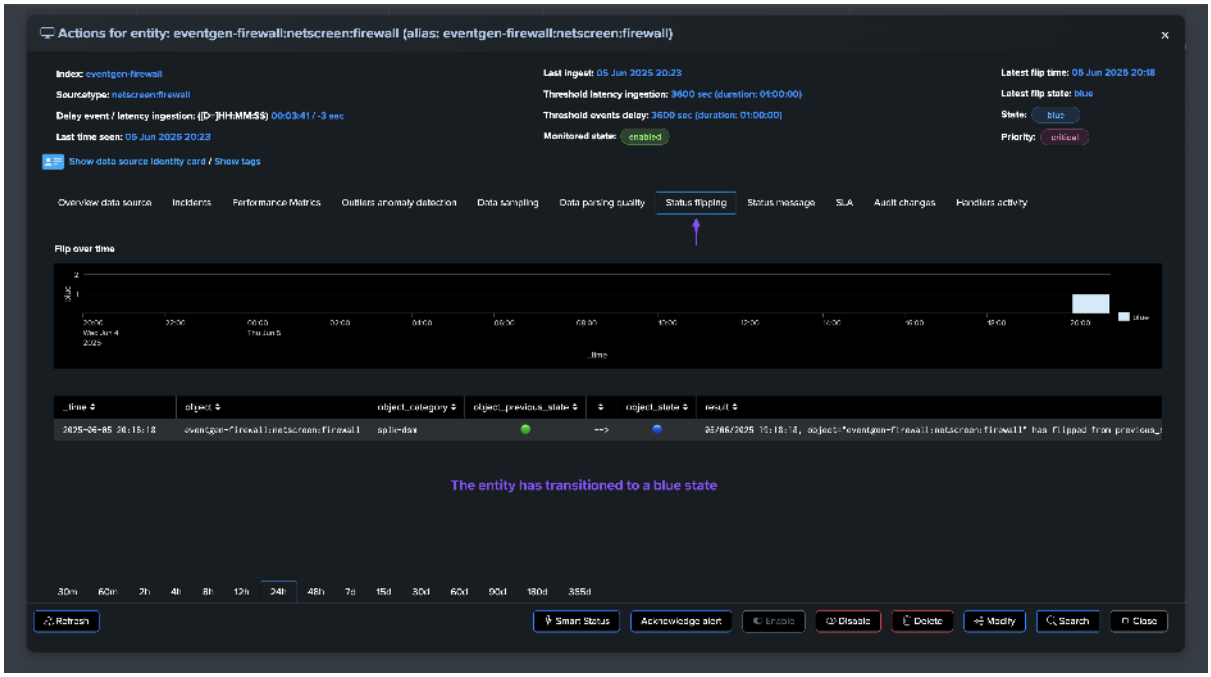


7.15.4 How does the disruption queue work?

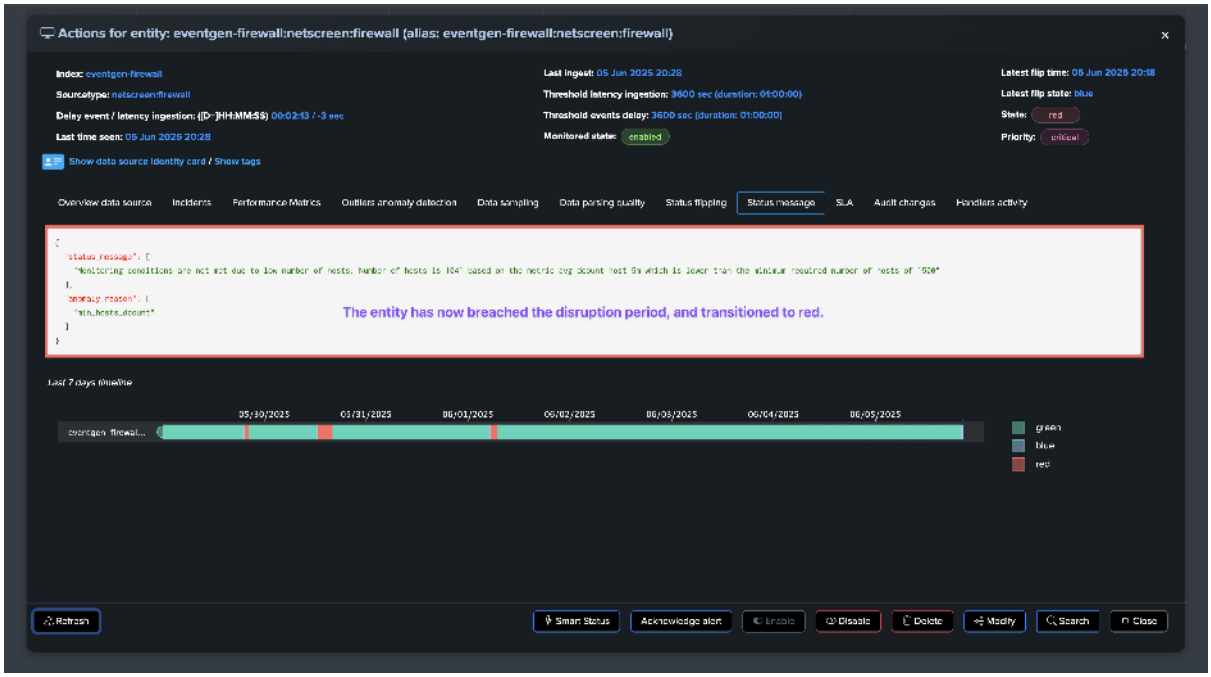
This is quite simple, the disruption queue is counter which starts when an entity is meant to be in alerting state (red):

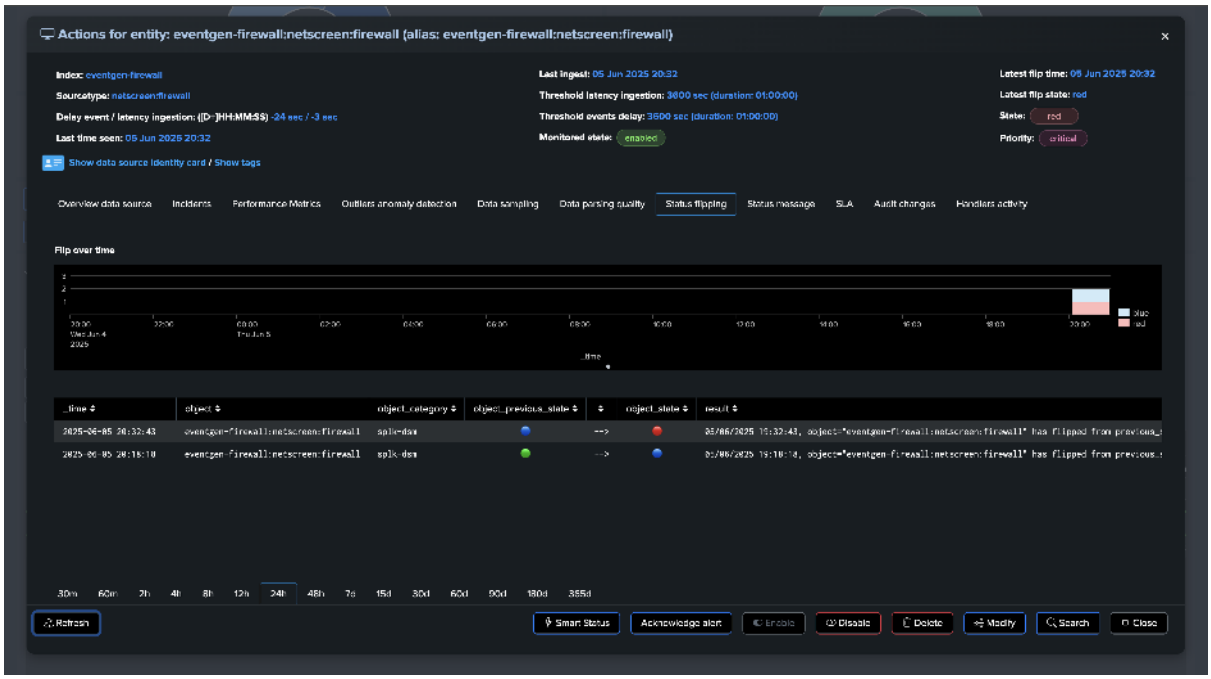


The entity has transitioned to a blue state:



Once this counter reaches the minimal disruption period, the entity will transition to red if the anomaly persists:





7.16 TrackMe Sourcetypes & Metrics

About TrackMe sourcetypes and metrics

- TrackMe generates various events and metrics for the purposes of its activity.
- The destination indexes for both events and metrics are entirely configurable, and potentially on a per Virtual Tenant basis.
- Events generated by TrackMe are JSON events, and parsed using KV_MODE=json.
- Metrics are generated and stored in the Splunk metrics store using metric indexes as per the Virtual Tenant configuration.

7.16.1 TrackMe events & sourcetypes

The following event sourcetypes are generated by TrackMe:

To list all events from a Splunk search:

```
| tstats count where index=trackme_* by sourcetype
```

Sourcetype	Purpose
trackme:audi	TrackMe audit events, for instance when modifications are made against TrackMe entities
trackme:flip	TrackMe flipping state events, flipping events are generated when entities switch from one state to another
trackme:hand	Handler events are generated when TrackMe logics, such as trackers, are executed against a given entity. These events allow identifying which logics are maintaining and performing operations on which entities.
trackme:heal	Health events are generated to monitor the health status of TrackMe components and logics, for instance these events are used to identify when a given Virtual Tenant is degraded due to an issue in a tracker.
trackme:sla_	TrackMe has a concept of SLA tracking, when entities are breaching their SLA and the associated SLA class definition, an SLA breach event will be generated. (one event every 24 hours by default)
trackme:stat	State events are generated when entities are maintained by TrackMe, and contain key information about the entities statuses.
trackme:stat	Stateful events are generated via the concept of state aware alerting, when incidents are opened, updated or closed.
trackme:nota	TrackMe notable events produced by the TrackMe notable alert action
trackme:smar	TrackMe SmartStatus events produced by the TrackMe SmartStatus alert action, when an alert triggers and has the SmartStatus alert action enabled, automated investigations are performed and their results are indexed as SmartStatus events.
trackme:wlk:	These events are specific to the Workload component (splk-wlk), and are generated when an updated version (or first time discovered) of a monitored Splunk scheduled search occurs.

7.16.2 TrackMe main metrics

TrackMe generates various metrics per component, using the following strict convention:

To list all metrics from a Splunk search:

```
| mcatalog values(metric_name) as metrics, values(_dims) as dimensions where␣
↪index=trackme_metrics
```

To view the content of metrics in a practical way such as you would with events in Splunk:

```
| mpreview index=trackme_metrics filter="metric_name=trackme.*"
```

metric_name	Purpose
trackme.splk.feeds.*	Metrics generated by the Feeds tracking components (splk-dsm/splk-dhm/splk-mhm), which contain notably metrics are latency and availability for TrackMe feeds entities.
trackme.splk.flx.*	Metrics generated by the Flex Object splk-flx component. In Flex, a tracker can generate any kind of metrics depending on the use cases, from system metrics to functional or business metrics.
trackme.splk.cim.*	Metrics generated by the CIM compliance component. (splk-cim)
trackme.splk.wlk.*	Metrics generated by the Workload component. (splk-wlk)
trackme.components_r*	Metrics generated when trackers are executed, which trace the health status and run-time of the various TrackMe components and logics such as trackers.
trackme.sla.object_state	These metrics are generated for SLA tracking purposes, and contain the state of the entities in a numerical format.

7.16.3 TrackMe logging events

About TrackMe logging

- TrackMe has essentially two types of components generating log messages, **REST API endpoints** and **custom commands**.
- Both carefully perform message logging, according to the logging level defined in the TrackMe global configurations.

REST API endpoints log messages

To access REST API log messages, you would use the following search:

```
index=_internal sourcetype=trackme:rest_api
```

Custom commands log messages

To access custom commands log messages, you would use the following search:

```
index=_internal sourcetype=trackme:custom_commands:*
```

7.17 TrackMe REST API

TrackMe provides a deep REST API that allows interracting with any of its components, in fact, all interactions from the user interfaces deal with TrackMe's REST API endpoints.

Hint

Since the version 2.0.34, TrackMe implements a strict least privileges approach:

- Endpoints are eventually (depending on the resource group) separated in 3 groups per resource groups, **user level** endpoints, **write level** endpoints and **admin level** endpoints
- According to the RBAC configuration, a given user may not be granted access to write and admin endpoints if the user is a normal user, or the user may not be granted access to the admin endpoints if the user is power user

- These are driven by TrackMe's builtin Splunk capabilities which can be granted to a given Splunk role, by membership or inheritance (see: *Role Based Access Control and ownership*)

The following table summarizes TrackMe's builtin roles, associates capabilities, endpoints accesses and privileges categories:

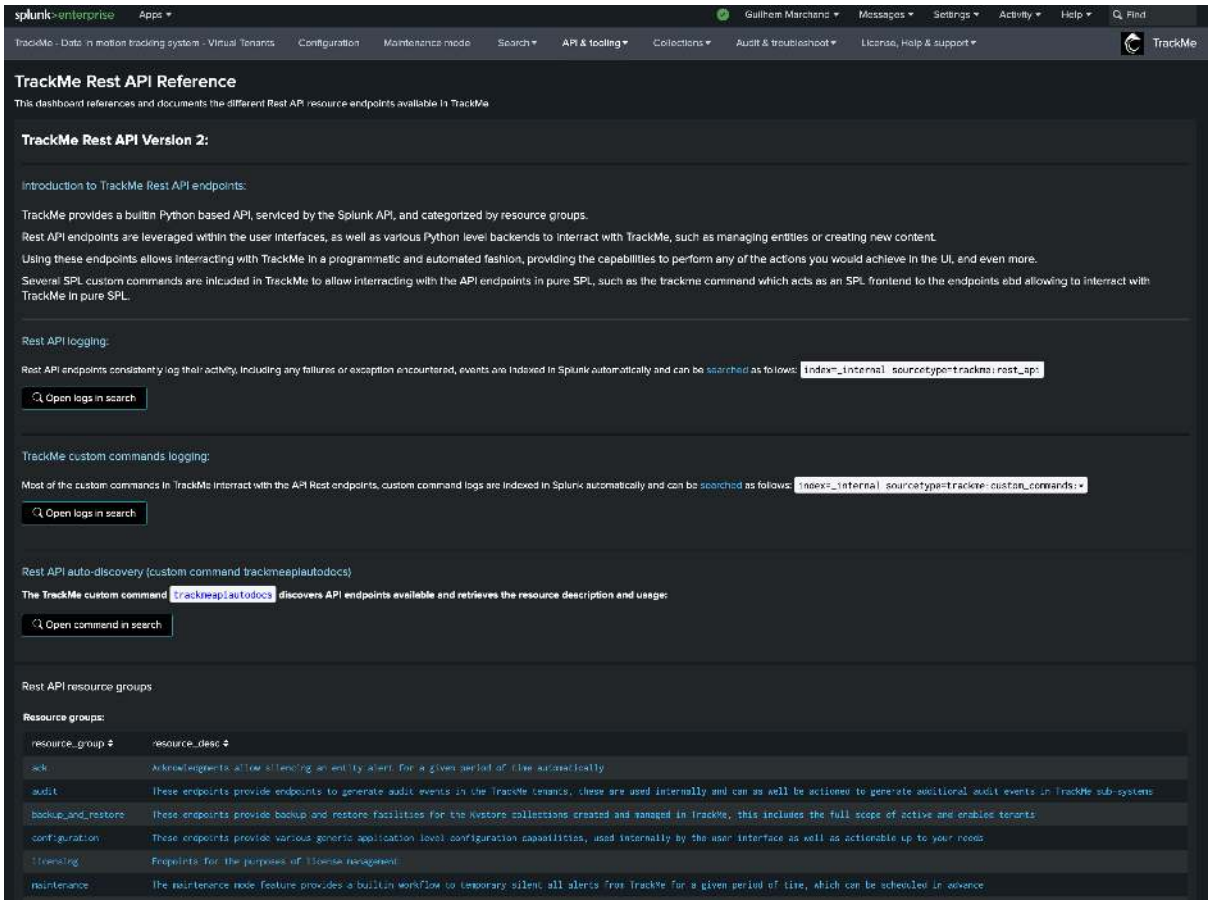
Table 15: TrackMe builtin Roles and Capabilities

Role	Capability	Endpoints root	Description
trackme_user	trackmeuseroperations	all but write and admin	allows read only access in TrackMe
trackme_power	trackmepoweroperations	*/write	allows management of TrackMe entities but not the creation of new content (such as trackers)
trackme_admin	trackmeadminoperations	*/admin	allows content management such as the creation of tenants and trackers

7.17.1 REST API reference

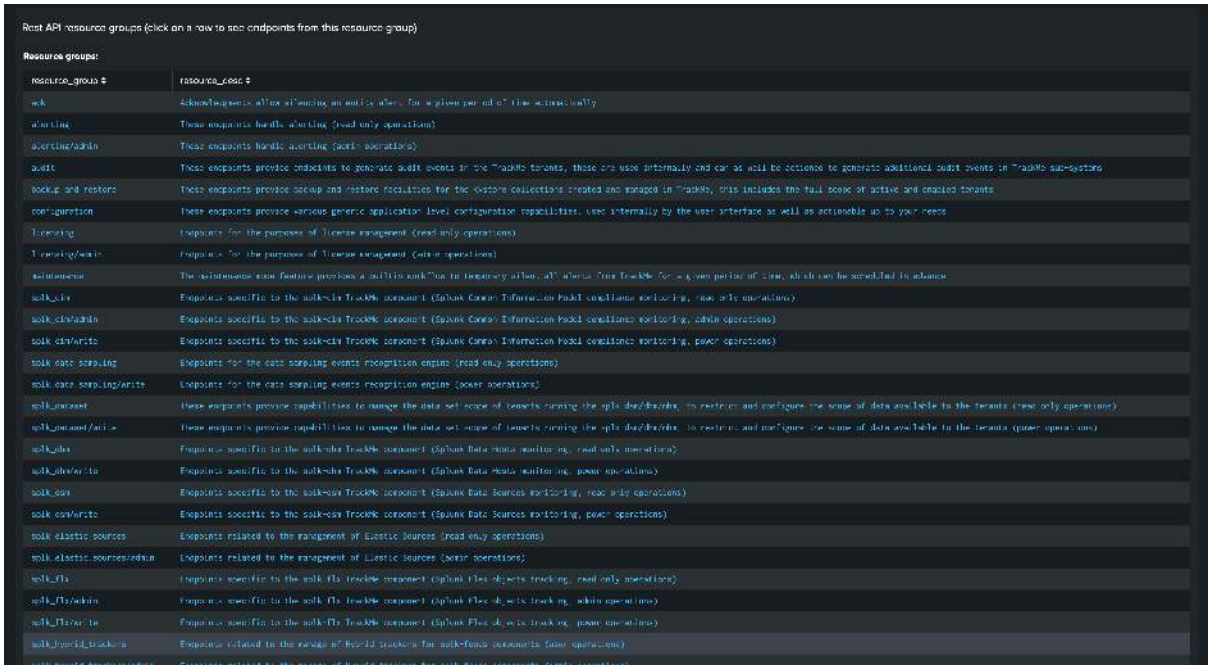
TrackMe comes with a REST API reference user interface which allows you to navigate through the available API endpoints categorized by resource groups:

Go to Navigation bar / API & Tooling / TrackMe REST API Reference:



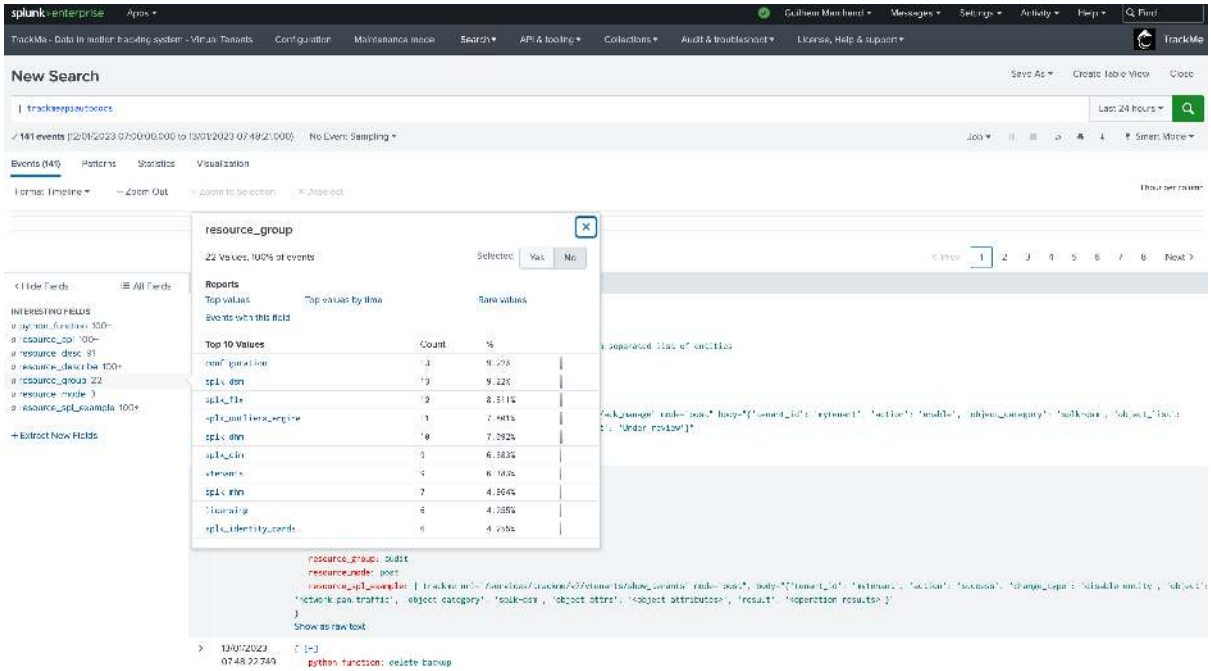
REST API resource groups

At the top level, TrackMe REST API endpoints are categorized by resource groups:



TrackMe has a concept of automatic discovery of the REST API endpoints and their documentation, for this, we leverage a custom command:

```
| trackmeapiautodocs
```



The custom command `trackmeapiautodocs` automatically discovers REST API endpoints and extract their documentation such as the endpoint URI, the expected HTTP mode, options available for that endpoint and the example of use in SPL:

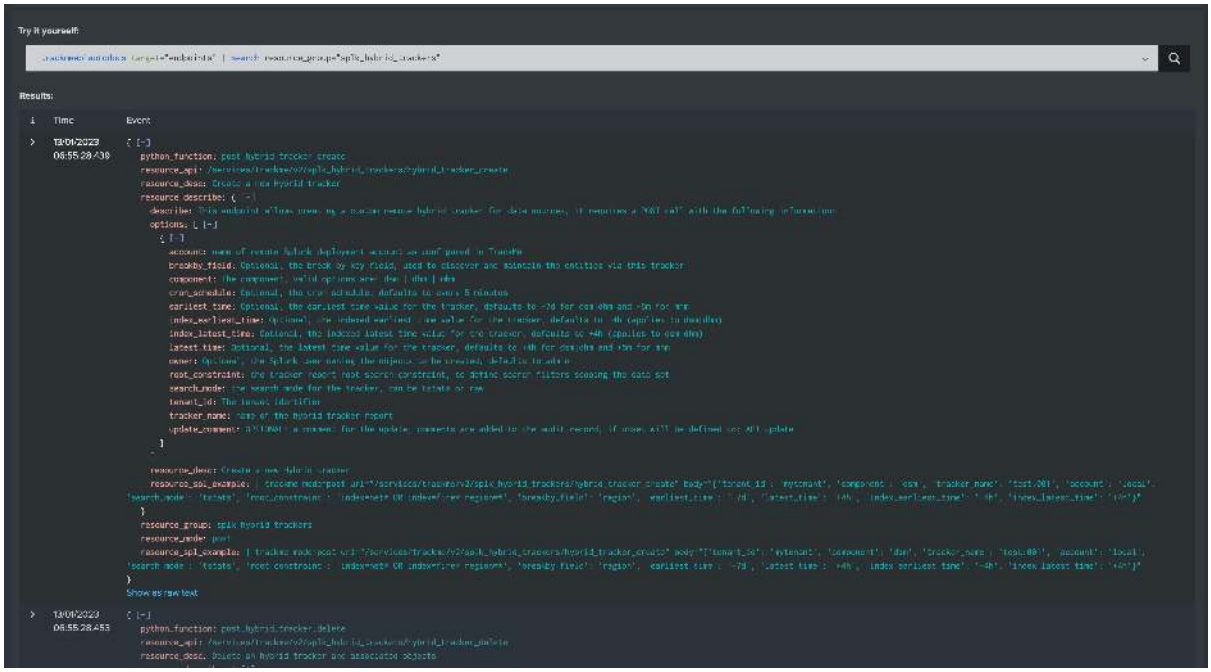
To list all resource groups, you can use the following SPL command:

```
| trackmeapiautodocs target="groups" | table resource_group, resource_desc | sort 0␣
↪resource_group
```

To retrieve the list of API endpoints and their detail, you can use the following SPL command:

Example:

```
| trackmeapiautodocs target="endpoints" | search resource_group="splk_hybrid_trackers"
```



REST API logging access

Any activity of the REST API endpoints is carefully logged, consult *REST API Endpoints Logging*

REST API audit

Any endpoint performing a change in TrackMe, such as deleting an entity, leads to the generation of one or more audit events from that endpoint.

For instance, you can search for all audit events for a given Virtual Tenant:

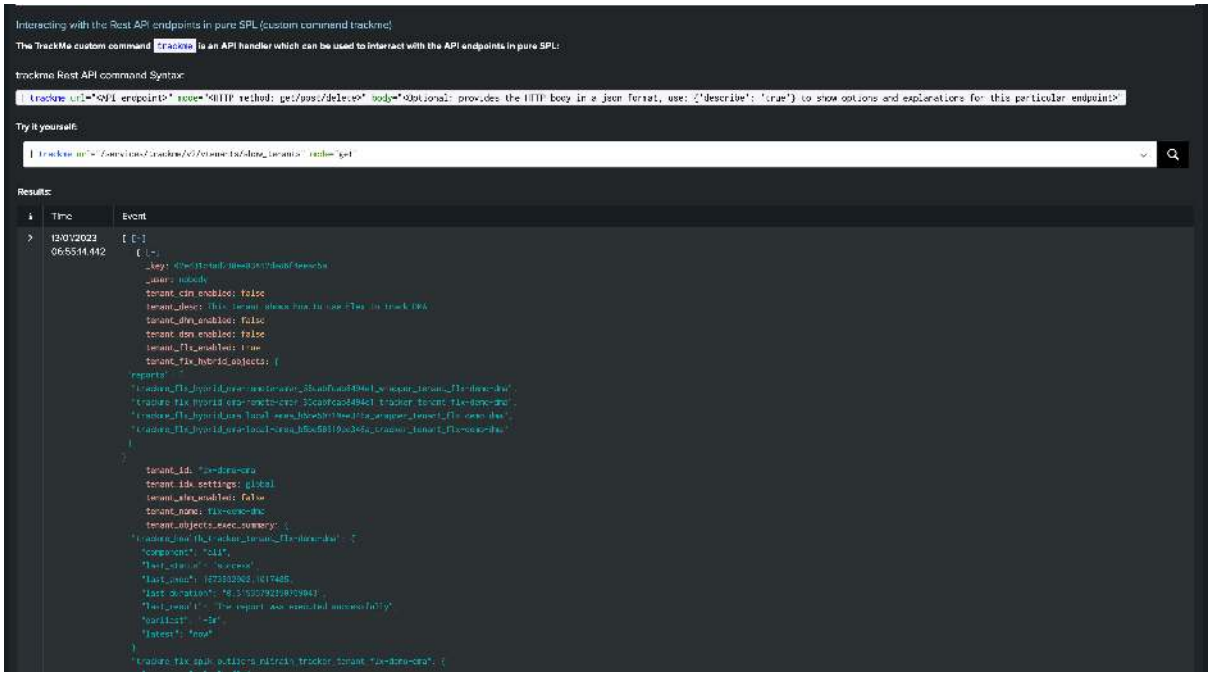
```
index=trackme_audit tenant_id=mytenant
```

REST API usage example in Splunk

REST API endpoints can therefore be used programmatically for any kind of action, within Splunk using the `trackme` REST API wrapper custom command, and externally using any program of your choice, from `curl` to `Postman` and others.

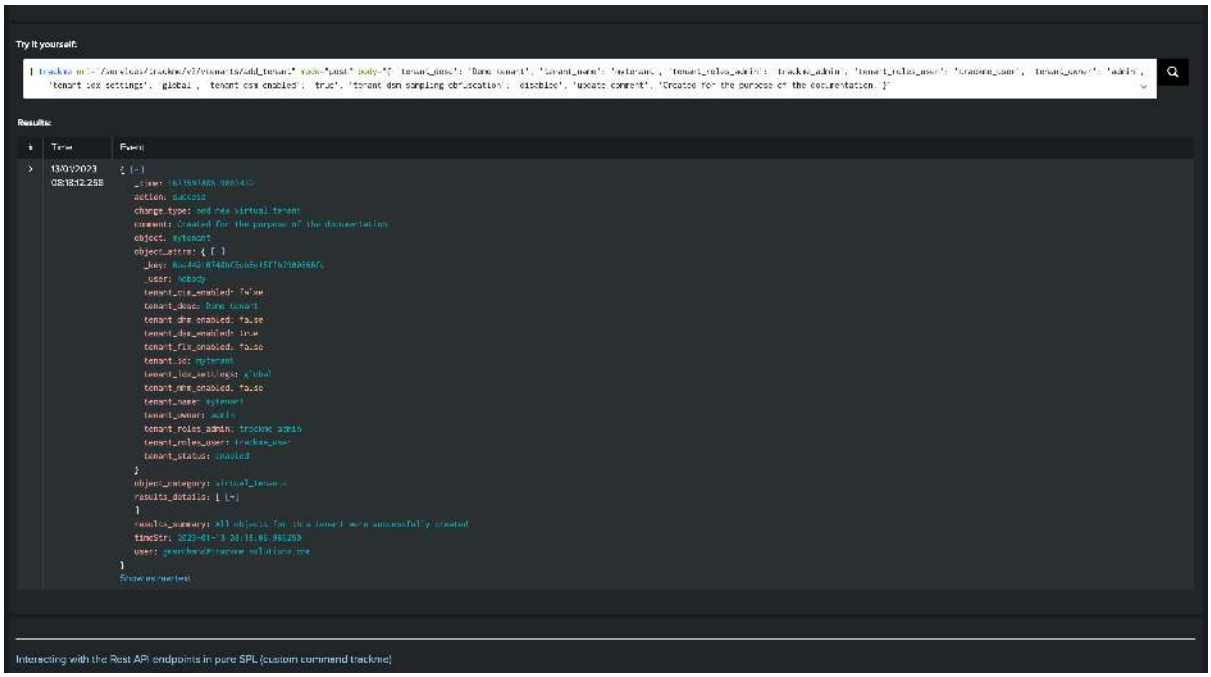
For instance, the usage for retrieving the details of existing Virtual Tenants:

```
| trackme url="/services/trackme/v2/vtenants/show_tenants" mode="get"
```



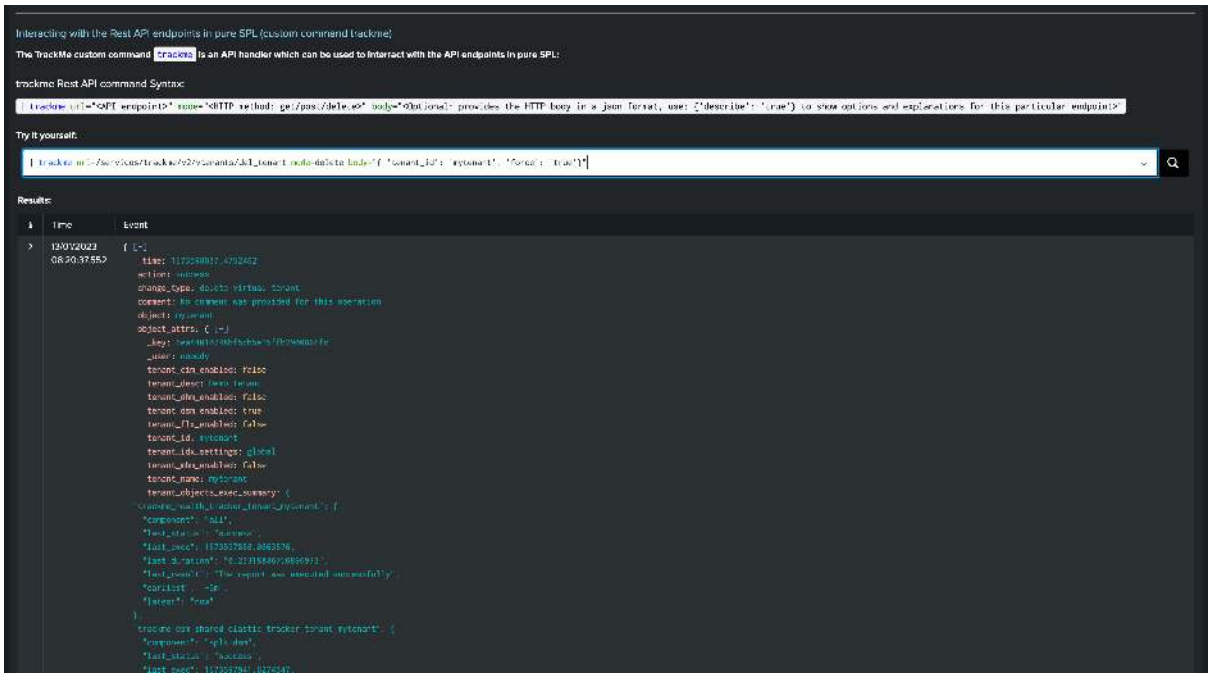
This other example would create a brand new virgin Virtual Tenant for splk-dsm:

```
| trackme url="/services/trackme/v2/vtenants/admin/add_tenant" mode="post" body="{
↪'tenant_desc': 'Demo tenant', 'tenant_name': 'mytenant', 'tenant_roles_admin':
↪'trackme_admin', 'tenant_roles_user': 'trackme_user', 'tenant_owner': 'admin',
↪'tenant_idx_settings': 'global', 'tenant_dsm_enabled': 'true', 'tenant_dsm_sampling_
↪obfuscation': 'disabled', 'update_comment': 'Created for the purpose of the␣
↪documentation.'}"
```



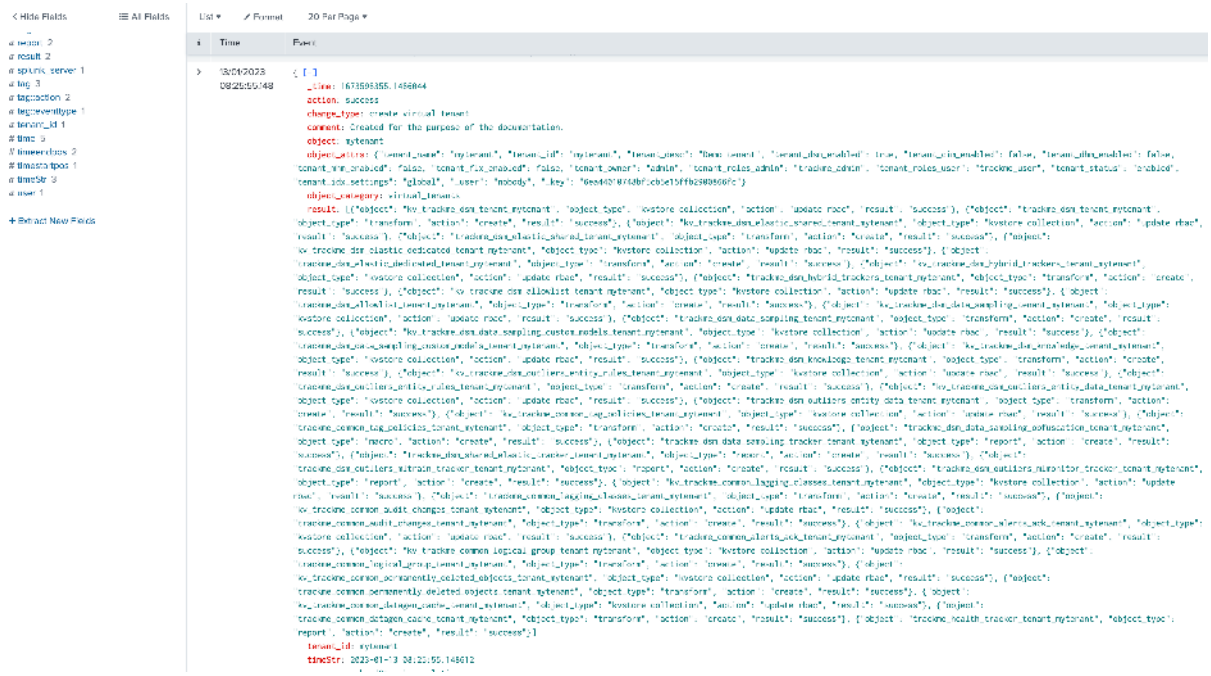
The following example deletes that same Virtual Tenant we've created:

```
| trackme url=/services/trackme/v2/vtenants/admin/del_tenant mode=delete body="{
↪'tenant_id': 'mytenant', 'force': 'true'}"
```



Audit messages are generated and indexed in the Virtual Tenant audit index:

```
index=trackme_audit tenant_id=mytenant
```



REST API usage example outside of Splunk

Authenticating to the Splunk API

Using a bearer token:

The best approach for using the Splunk REST API is to rely on a bearer token to authenticate and access to splunkd:

See: Splunk docs JSON authentication token

Once you have created an authentication token for the user to be used as the service account, using curl specify the bearer token:

```
curl -k -H "Authorization: Bearer <token>"
```

Using basic authentication:

You can also use basic authentication and authenticate to the Splunk API using a username and a password:

```
curl -k -u admin:'ch@ngeM3'
```

Generating an API token:

A last method, but less common since the bearer token were introduced, is to authenticate using basic authentication and retrieve an API token which will be used for the REST calls purposes.

See: Splunk docs API token

Example:

```
curl -k https://localhost:8089/services/auth/login --data-urlencode username=svc_
↪splunk --data-urlencode password=pass

<response>
  <sessionKey>DWGNbGpJgSj30w0GxTAxMj8t0dZKjvjxLYaP^yphdluFN_FGz4gz^
↪NhcgPCLDkjWH3BUQa1Vewt8FTF8KXyyfI09HqjOicIthMuBIB70dVJA8Jg</sessionKey>
  <messages>
    <msg code=""></msg>
  </messages>
</response>

export token="DWGNbGpJgSj30w0GxTAxMj8t0dZKjvjxLYaP^yphdluFN_FGz4gz^
↪NhcgPCLDkjWH3BUQa1Vewt8FTF8KXyyfI09HqjOicIthMuBIB70dVJA8Jg"
```

A token remains valid for the time of a session. (1 hour by default)

The token would be used as following:

```
curl -k -H "Authorization: Splunk $token"
```

Using the REST API from the outside with curl

Consider the following REST API usage example in SPL:

```
| trackme url="/services/trackme/v2/vtenants/trackmeload" mode="post" body={\"mode\":␣
↪\"full\"}
```

Using curl, it could be translated into:

```
curl -u admin https://localhost:8089/services/trackme/v2/vtenants/trackmeload -X POST␣
↪-d '{"mode": "full"}'
```

Hint

JSON format

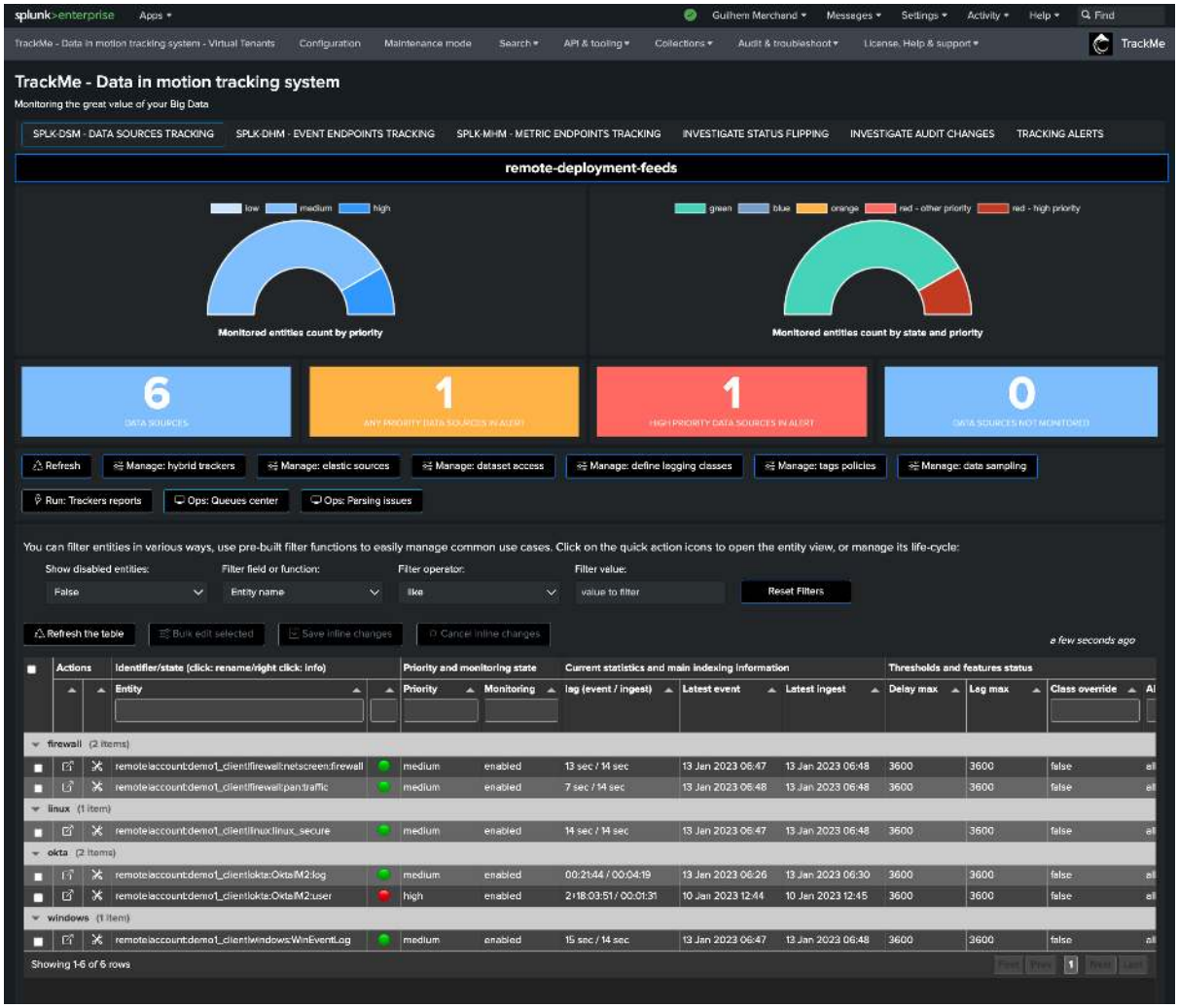
- Note that the trackme SPL command understands JSON submitted with both single or double quotes, however when using a REST API endpoint from the outside of Splunk, you must submit a proper JSON object with double quotes.

7.18 splk-feeds - Creating and Managing Hybrid Trackers

7.18.1 Introduction to Hybrid Trackers

Hybrid Trackers are created and managed through TrackMe. These are scheduled backend jobs that orchestrate entity discovery and management for TrackMe splk-feeds components:

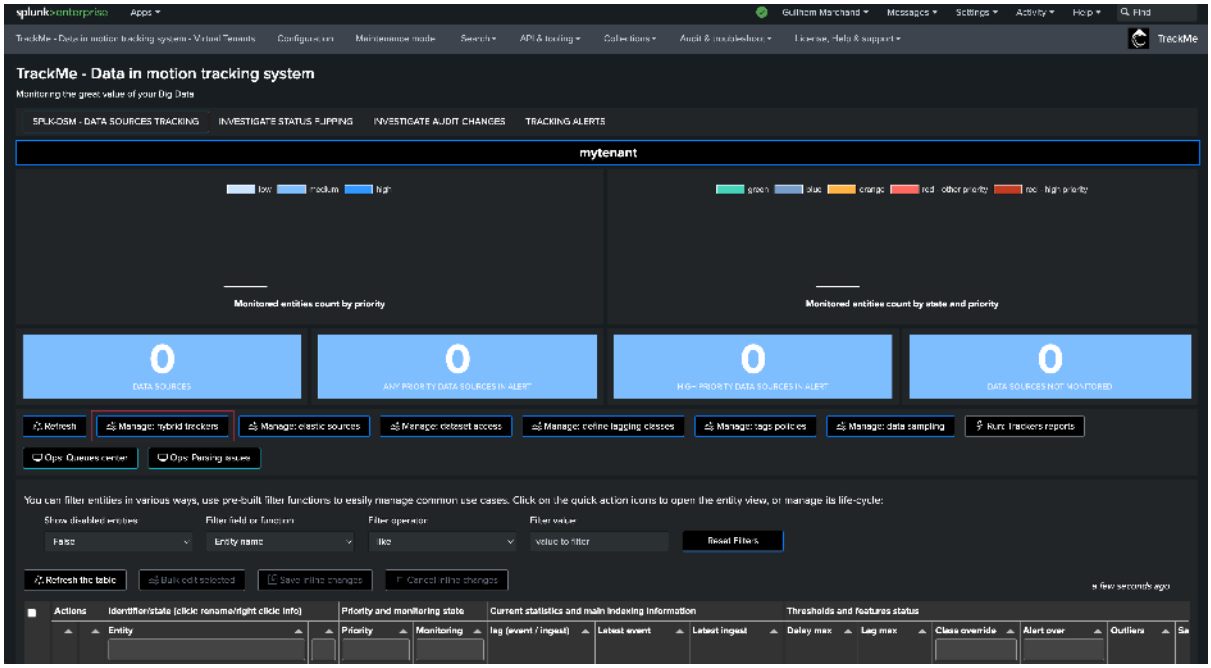
- Hybrid Trackers are scheduled reports that involve various TrackMe backend tools depending on the TrackMe component
- A single Hybrid Tracker can discover and manage a few or many entities according to the needs
- Hybrid Trackers integrate into a main application workflow, which involves concepts such as registering their execution statuses and run time performance
- Hybrid Trackers can be created at any time through guided user interfaces or TrackMe REST endpoints
- In the context of Splunk feeds tracking, Hybrid Trackers can also be created during the initial creation of the Virtual Tenant
- When creating trackers, the related knowledge objects will be owned by the owner defined at the Virtual Tenant level
- TrackMe keeps records of the knowledge objects related to the Hybrid Trackers; therefore, you need to manage their lifecycle through TrackMe

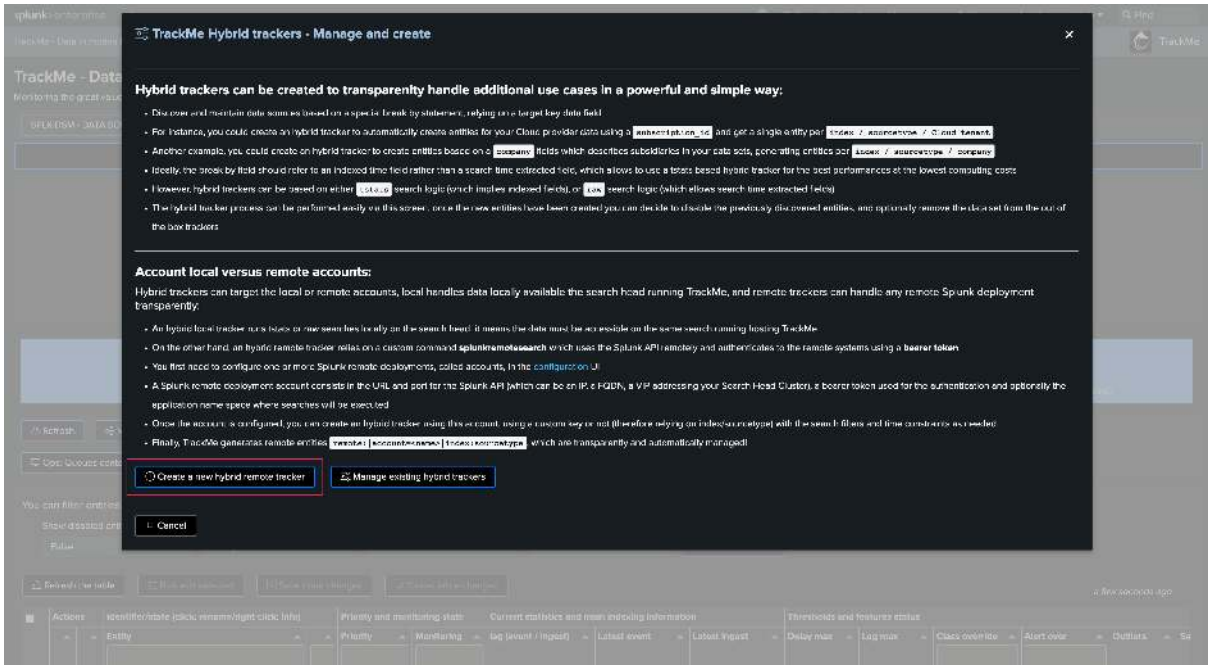


7.18.2 Creating an Hybrid Tracker for splk-feeds

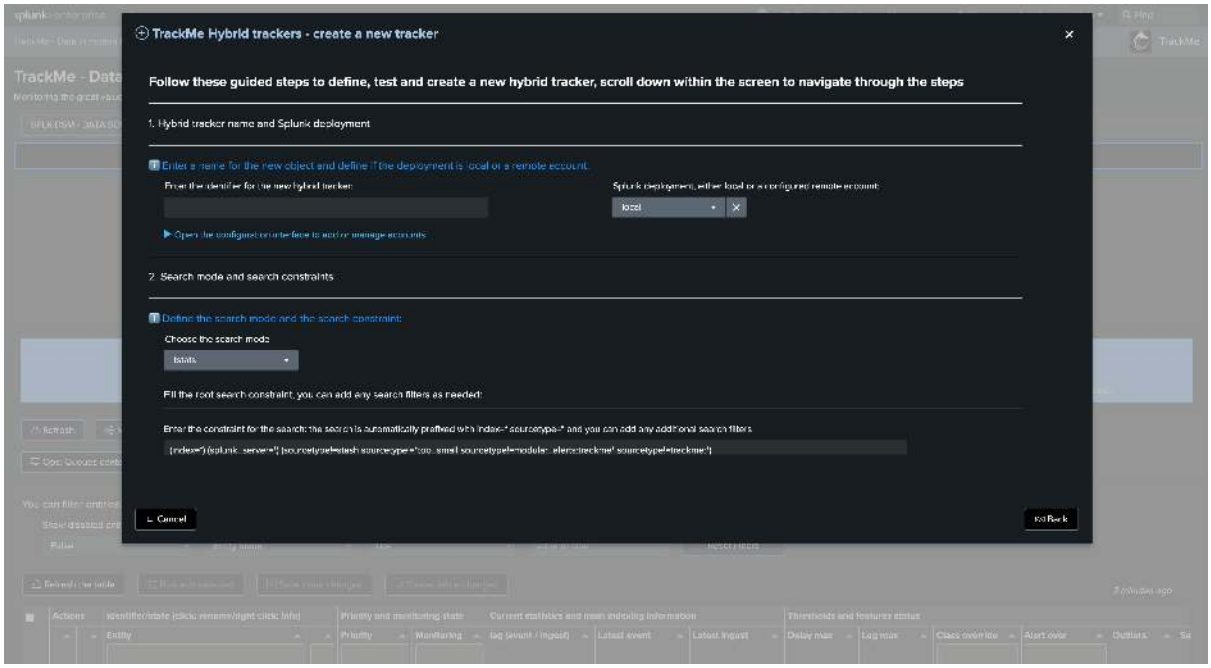
These instructions are related to the splk-dsm component. Options for splk-dhm/splk-mhm may differ, but the underlying logic is similar.

To create a new Hybrid Tracker, access the tenant and click on "Manage: Hybrid Trackers":





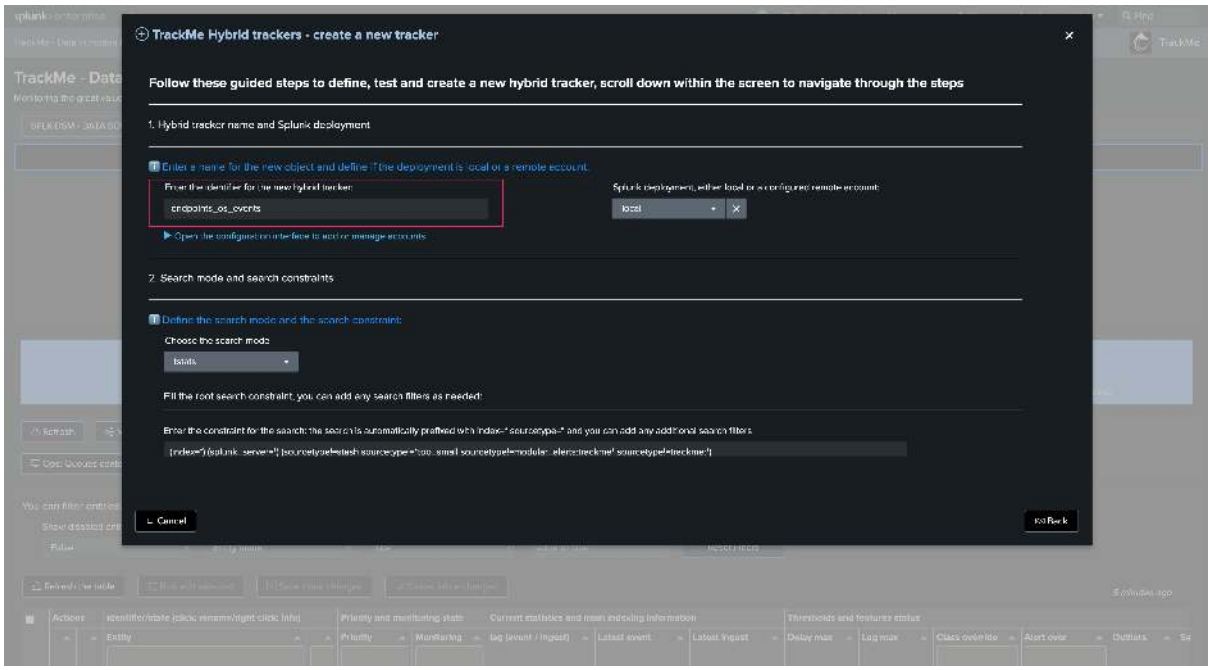
splk-dsm Hybrid Tracker creation wizard:



Once in the creation wizard, follow the guided steps:

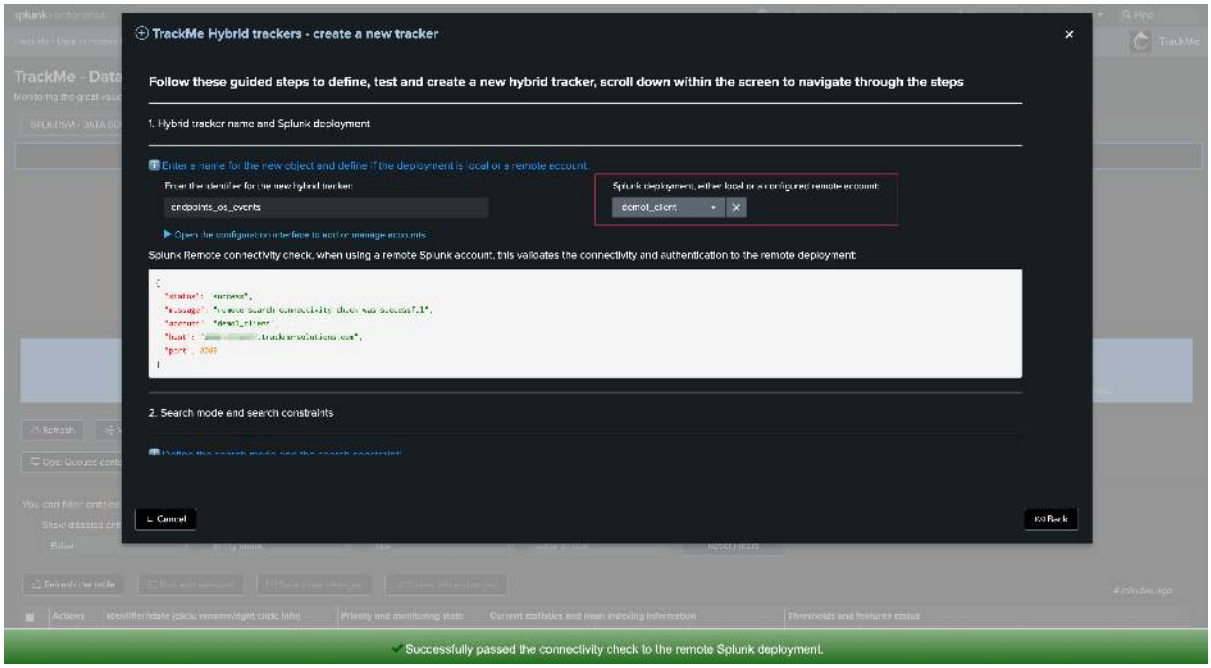
Hybrid Tracker identifier:

- Provide a name for the Hybrid Tracker. This will be included in the name of the Splunk Knowledge Objects related to this tracker
- In the example below, we will name our tracker "endpoints_os_data" as it deals with events originating from Operating Systems



Target Splunk deployment:

- Specify whether the data is searchable locally on the Splunk deployment or if the trackers deal with a remote Splunk deployment
- If a remote Splunk deployment is selected, TrackMe performs a connectivity check to that environment first



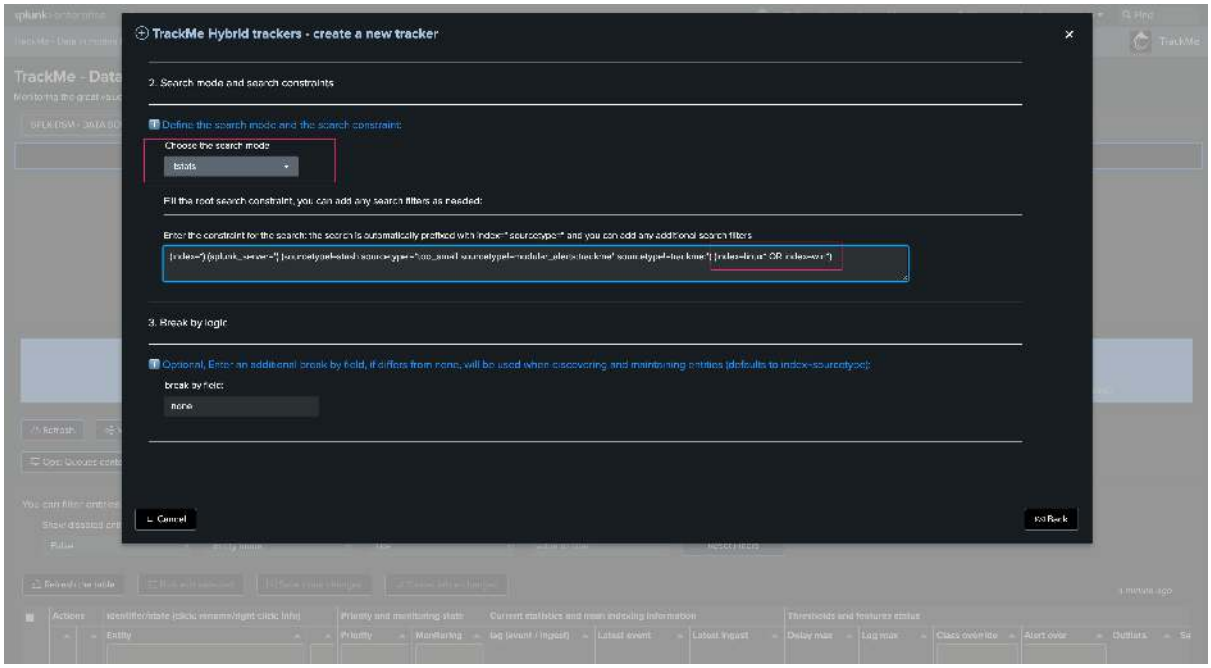
Search mode and search root constraint:

tstats versus raw

- Then, define the search mode. You can choose between tstats and raw
- tstats is generally recommended as it provides much faster and more efficient searches relying on Splunk tsidx files
- However, tstats requires all fields to be indexed fields, while a raw search can deal with search-time extracted fields
- Therefore, raw search provides much more flexibility, but the cost is also much higher
- Depending on your context, raw searches may be fully valid, but if a tstats search can be used equally, use tstats

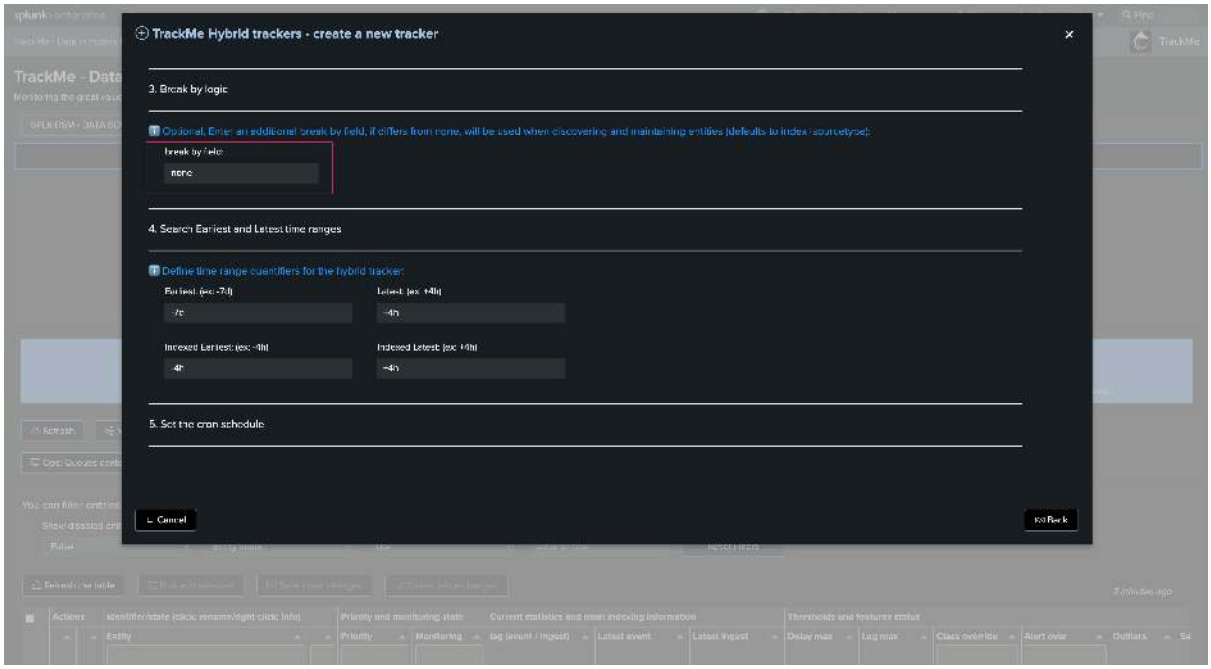
root search constraint:

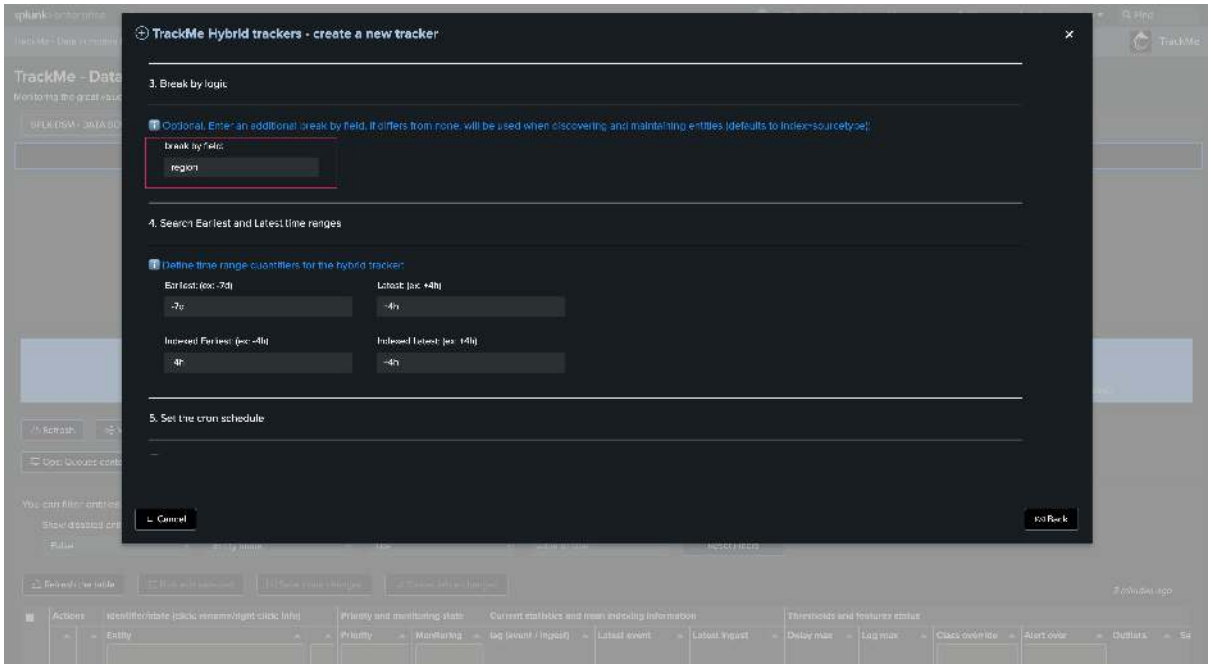
- Define the Splunk root search constraint. The constraint comes pre-filled with normally expected constraints which require valid data, exclude TrackMe related items, etc.
- Add your own search filters according to your needs. In our example, we add an index filter "(index=linux* OR index=win*)"



break by logic:

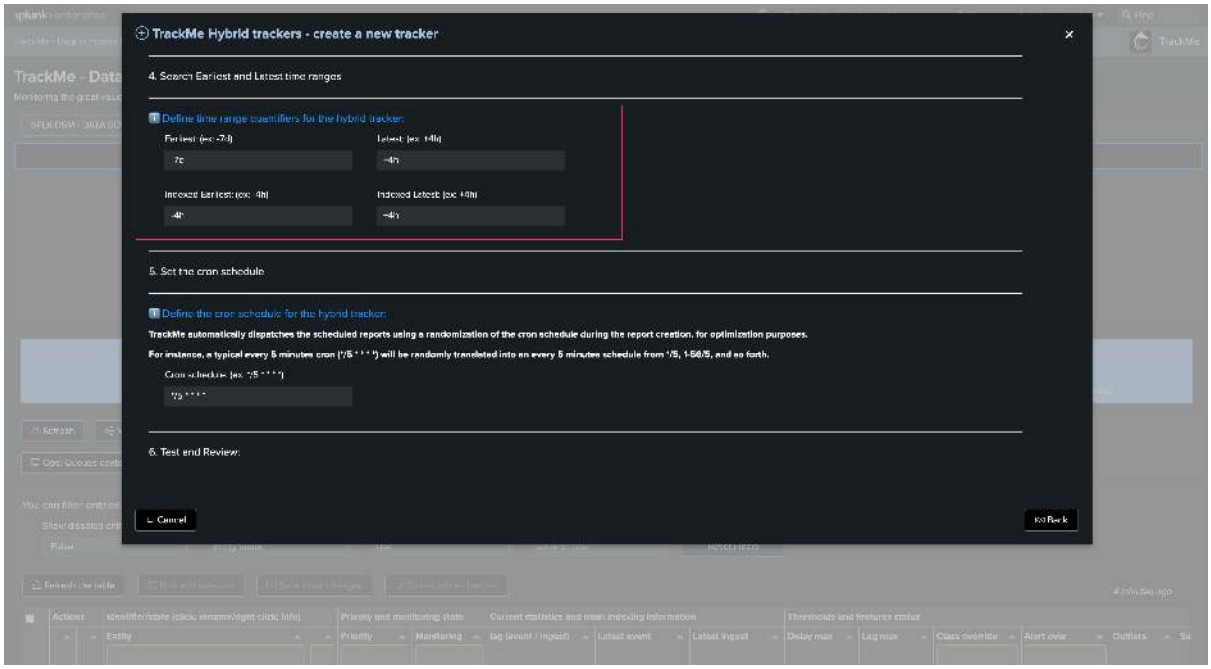
- You can optionally add an additional break by logic field
- This defaults to "none," which means entities are going to match the combo `index + ":" + sourcetype`
- For instance, if we have an indexed field `region`, we can leverage it here to distinguish entities per region. Our entity creation logic becomes `index + ":" + sourcetype + ":" + region`





Time quantifiers:

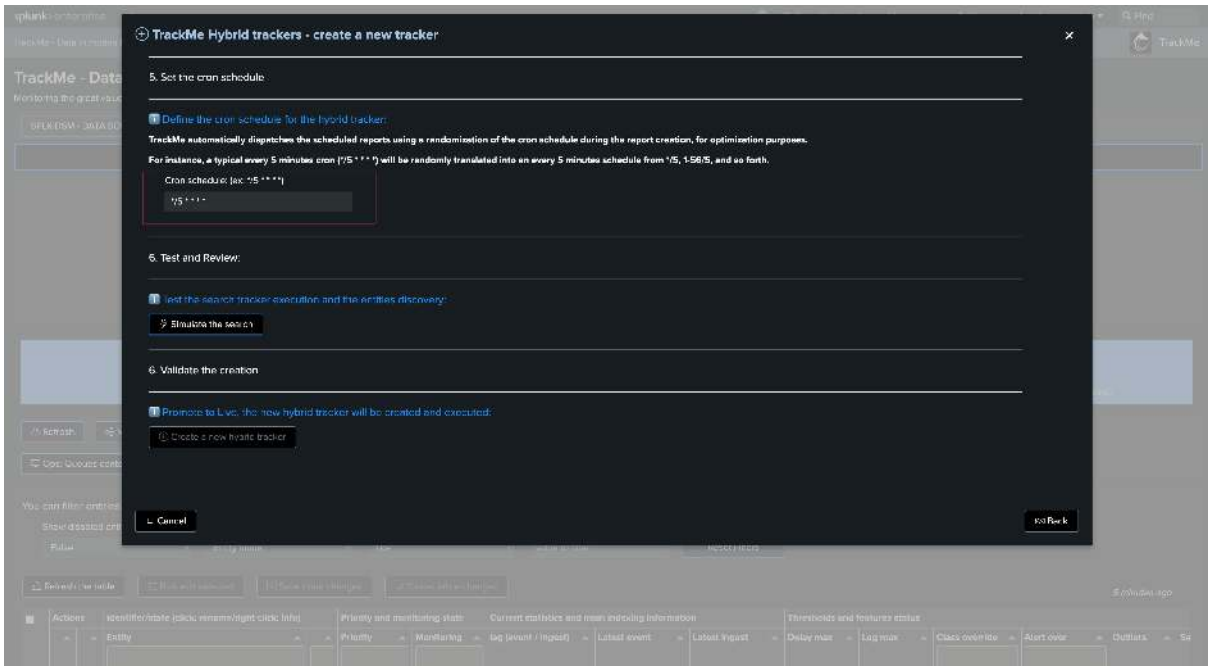
- Review and update if necessary the indexed time earliest and latest, as well as time range earliest and latest
- These time quantifiers drive the period of data that the tracker is going to cover
- Generally, you will want to have a large event time range period to cover data with high latency, while the period for indexed time range can be more restricted for performance optimization purposes
- What will work best and be the most efficient depends a lot on your context and environment. Start with these values, review and adapt if necessary



Cron schedule:

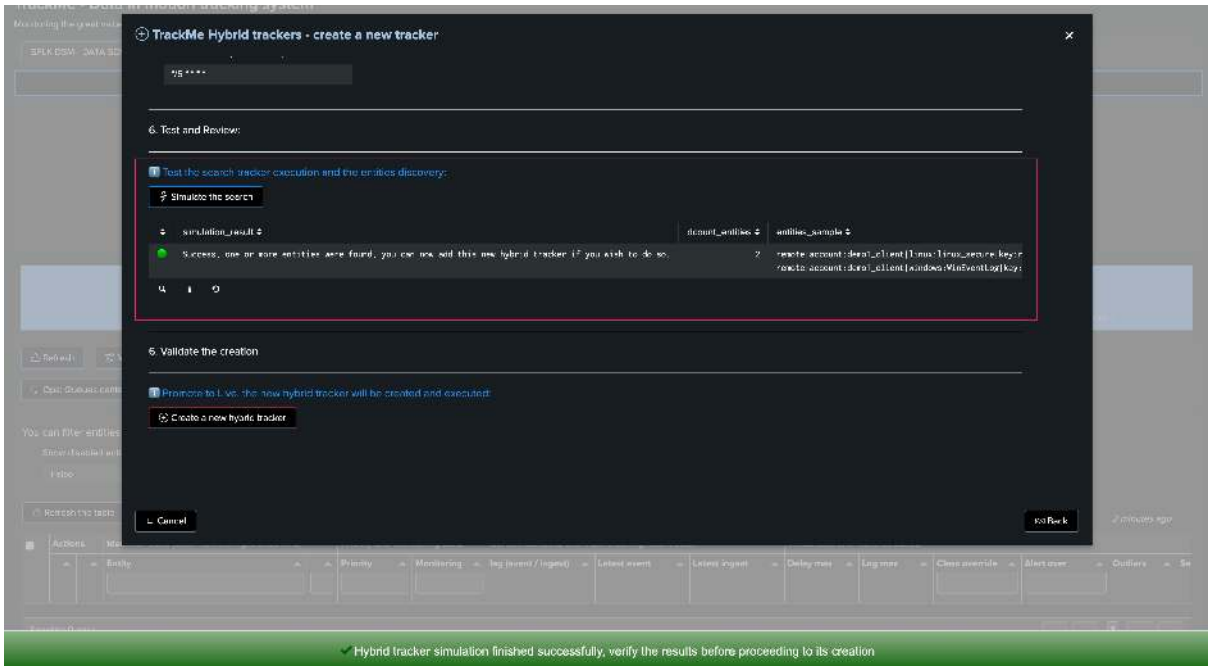
- Define the cron schedule for the Hybrid Tracker

- It defaults to every 5 minutes. Note that TrackMe will automatically dispatch cron schedules for optimization purposes

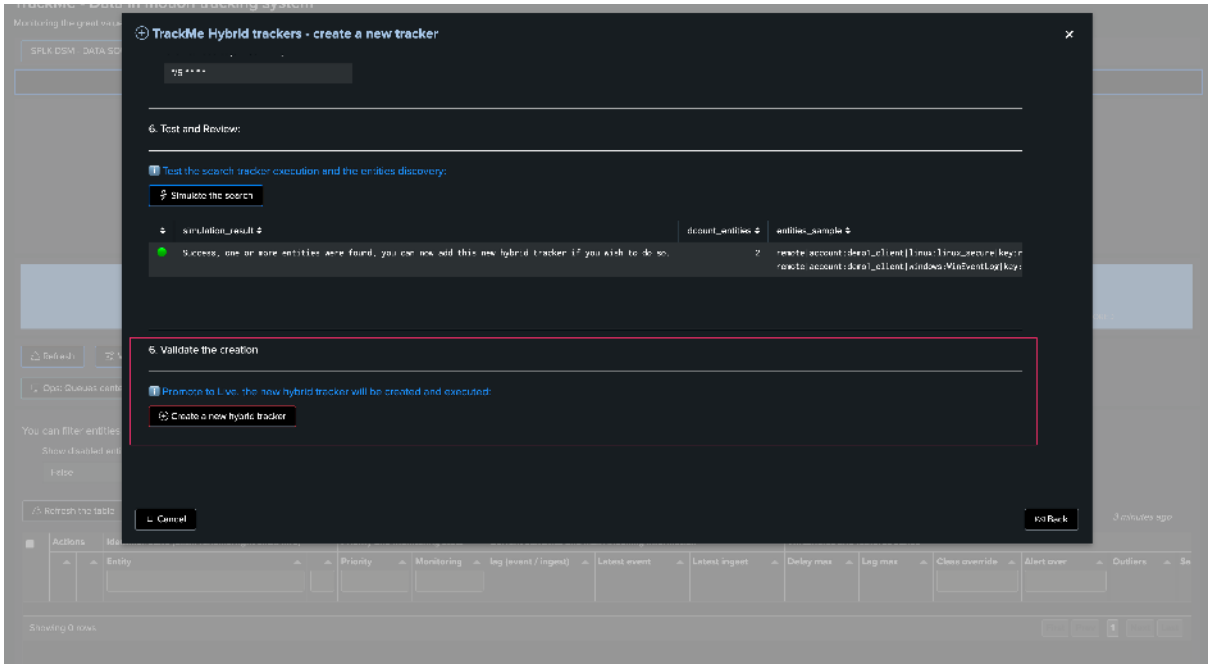


Test and review:

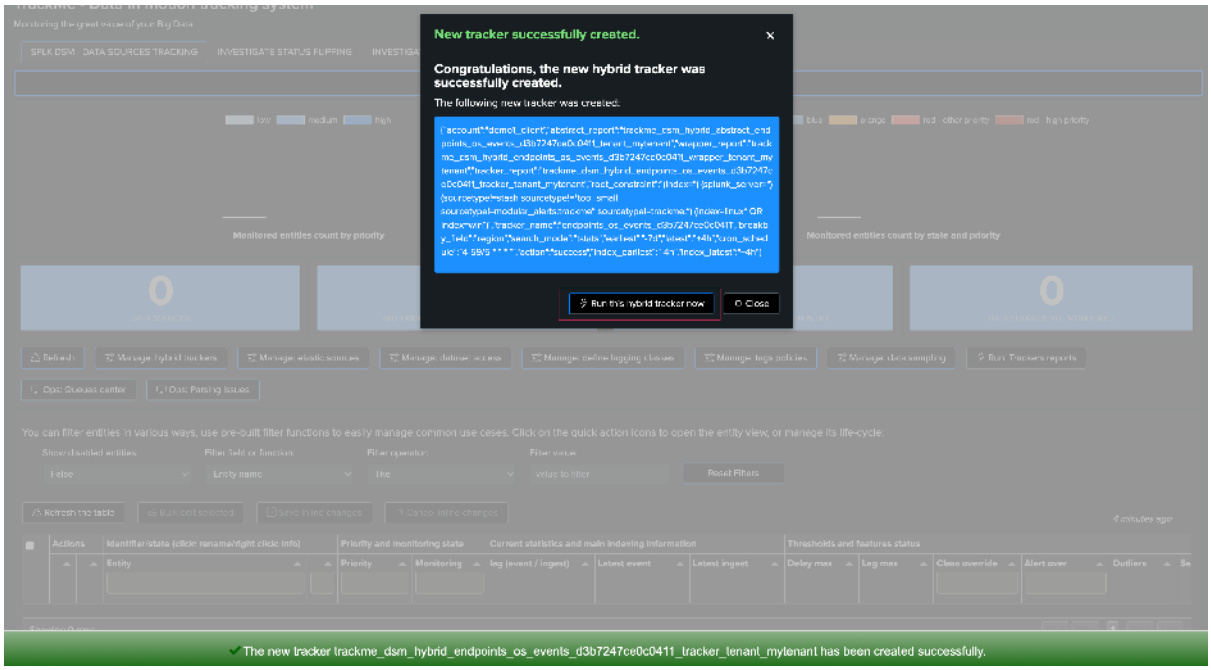
- Click on the button to execute the Hybrid Tracker in preview



Finally, validate the Hybrid Tracker creation:



Once created, you can choose to run the Tracker immediately to discover and create entities in the Virtual Tenant:



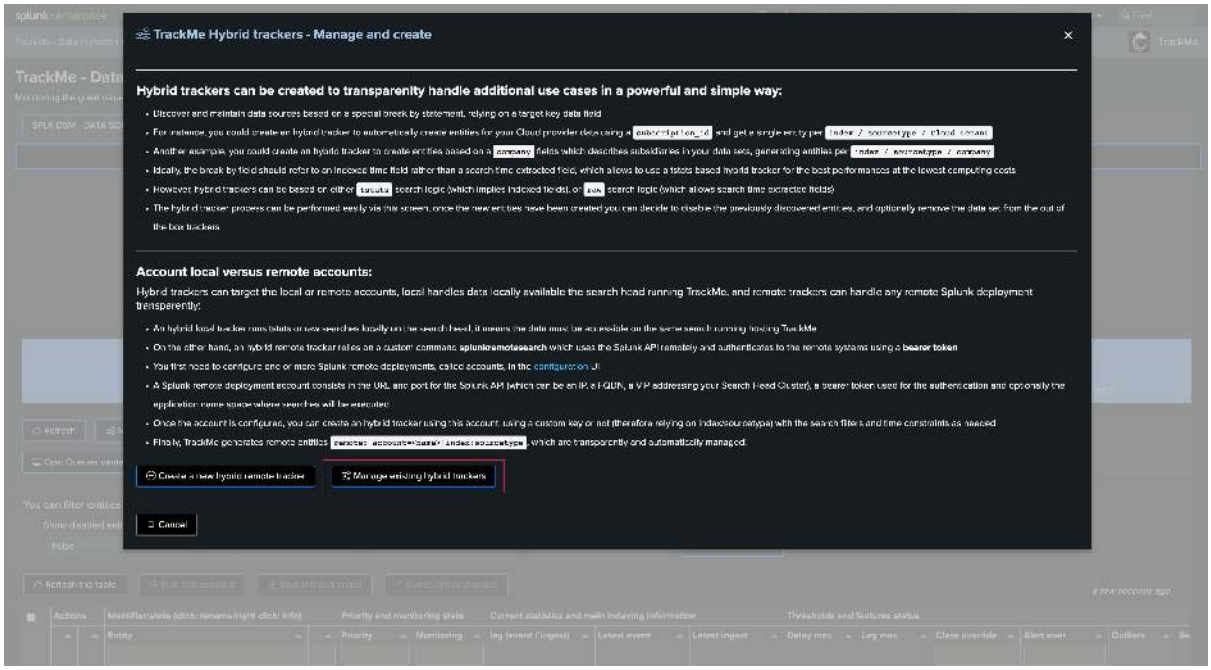
7.18.3 Managing Hybrid Trackers for splk-feeds

Deleting an Hybrid Tracker through the UI

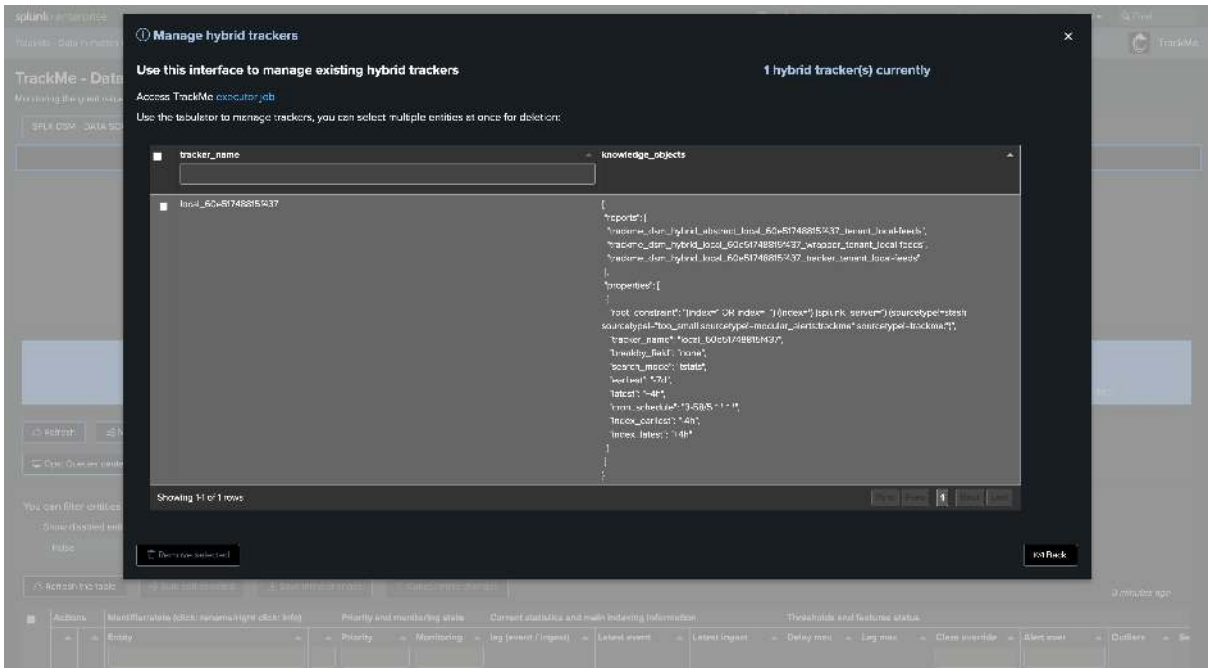
If you want to delete an existing Hybrid Tracker, this operation must be done via TrackMe.

The reason is that the application keeps track of all knowledge objects that were created for a given tenant to honor various features such as managing the lifecycle of the tenant (enabling/disabling, etc.) or the lifecycle of the tracker itself.

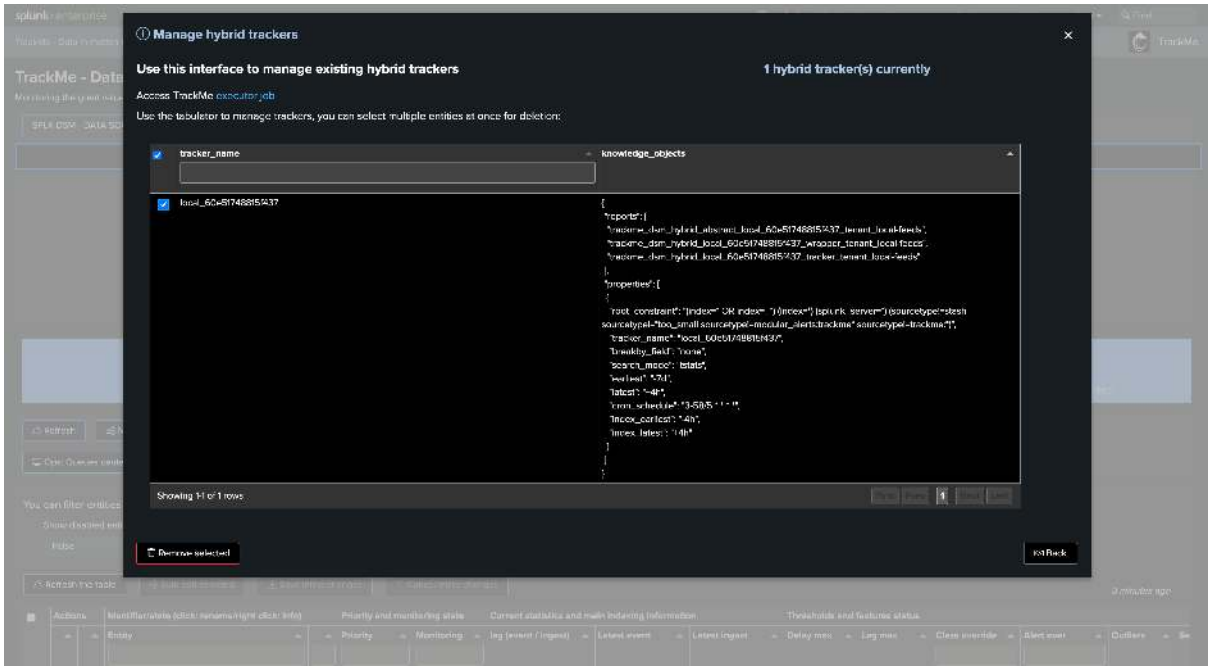
To manage Hybrid Trackers, click on:



The user interface shows available trackers and their related objects:



Select one or more trackers to be deleted:



The related knowledge objects will be deleted, and the Virtual Tenant record will be cleaned up automatically.

For splk-feeds, the entities that were created through these Hybrid Trackers will **not** be deleted. (However, unless another Tracker is created, these will not be maintained anymore)

Deleting an Hybrid Tracker through REST

You can delete a Tracker through the following REST endpoint, example in SPL:

```
| trackme mode=post url="/services/trackme/v2/splk_hybrid_trackers/admin/hybrid_
↪tracker_delete" body="{'tenant_id': 'mytenant', 'component': 'dsm', 'hybrid_
↪trackers_list': 'test:001,test:002'}"
```

7.19 Workload (splk-wlk) - Manage Workload tenants and trackers

7.19.1 Introduction to the Workload component

The TrackMe workload component (splk-wlk) is a licensed users restricted component which tracks the activity from any kind of scheduled search in your Splunk deployment, from reports to alerts and Data Model Acceleration. (DMA)

The Workload component tracks various use cases such as execution errors which can slightly affect the consistency of your Splunk use cases, providing powerful features to detect issues and track the performance of the Splunk scheduling activity.

For the User Guide, consult: *Splunk Workload (splk-wlk)*

Note

Grouping in the Workload component: group and overgroup

- By default, the Workload component groups entities by `app` which represents the Splunk application namespace hosting the scheduled report or alert.
- When creating the tenant and also later on when creating trackers manually, since TrackMe 2.0.70, you can optionally override this behaviour by setting up an `overgroup` value.

- By doing so, the Workload will group entities based on a custom term instead of applications, this can be useful if for instance you want to have multiple Search Head tiers in the same tenant.
- Refer to the *Creating a Workload tenant to host multiple Search Head tiers with overgroup* section for more information.

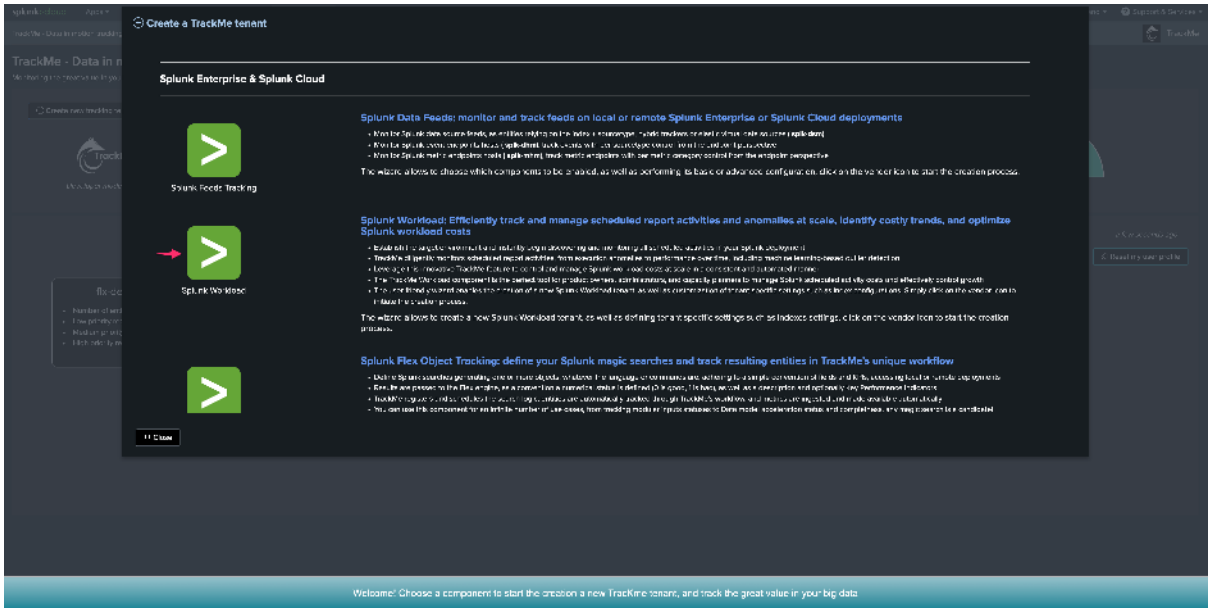
Note

Data collection enhancements in TrackMe v2.0.75

- Since TrackMe 2.0.75, the Workload components has been enhanced to automatically support data collection overlap between execution, relying on the tracking of the last known metrics for a given account/group and type of metrics.
- This allows TrackMe to look at multiple slices of times when tracking the Workload activity (scheduler, introspection, etc) and avoids missing any data point in case of Splunk delaying the execution of the workload trackers. (Splunk skew parameter, time_window and/or over busy scheduler)
- This process is fully automated and transparent, the last seen metadata are stored in a KVstore called `trackme_wlk_last_seen_activity_tenant_<tenant_id>`.

7.19.2 Creating and configuring a Workload Tenant

The creation and configuration of a Workload Tenant is guided through the wizard, click on the "create new tenant" and choose the associated component to start the configuration process:



Workload component in Splunk Cloud

Generally, in Splunk Cloud have two or more Search Head layers:

- The ad-hoc Search Head tier, where TrackMe should be used (standalone or SHC)
- The premium app Search Head tier, running Enterprise Security or ITSI (standalone or SHC)

In a nutshell, the configuration path is the following:

- configure a TrackMe Splunk Remote Deployment which targets the premium app Search Head tier (this is required notably for the versioning tracking of scheduled searches, even through both Search Head tiers access the same indexing tier)

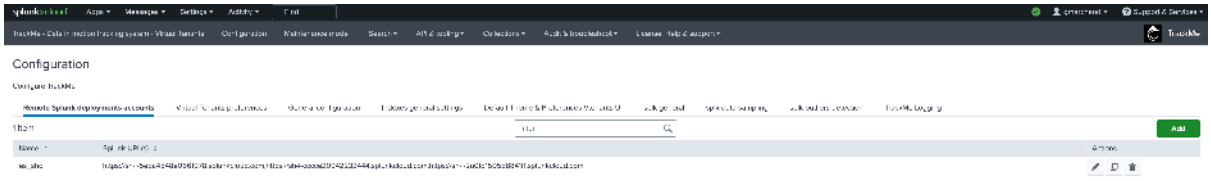
- following these instructions to identify the Search Head tiers names as automatically defined per Splunk Cloud summary data, and recycle this information using a simple subsearch (see in the next sections)
- create and configure the Workload tenant in TrackMe

Step 1: configure the remote Splunk Deployment

For more information about TrackMe's Splunk Remote Deployments capabilities and configuration, consult: *Splunk Remote Deployments (splunkremotesearch)*

Proceed as following:

- Create a bearer token in the premium app Search Head tier, it is good practice to associate the bearer token with a service account dedicated for that purpose



- Identify the Splunk Cloud label per Search Tier, we will use this logic as a subsearch when creating the Workload tenant/trackers:

```
index=summary source="splunk-svc-consumer" earliest=-7d latest=now
| stats values(usage_source) as members by search_head_names
```

Hint

Using the search_head_names label with a subsearch to dynamically restrict the hosts

- In Splunk Cloud, use the following subsearch technique to assign dynamically the list of Search Heads for that given Search Head tier
- If Search Head members are re-created, the subsearch will automatically pick up the new hosts and you have no changes to perform yourself

for instance, if the target Search head tier is called "es" (Enterprise Security), we will use the following subsearch to filter out the hosts dynamically:

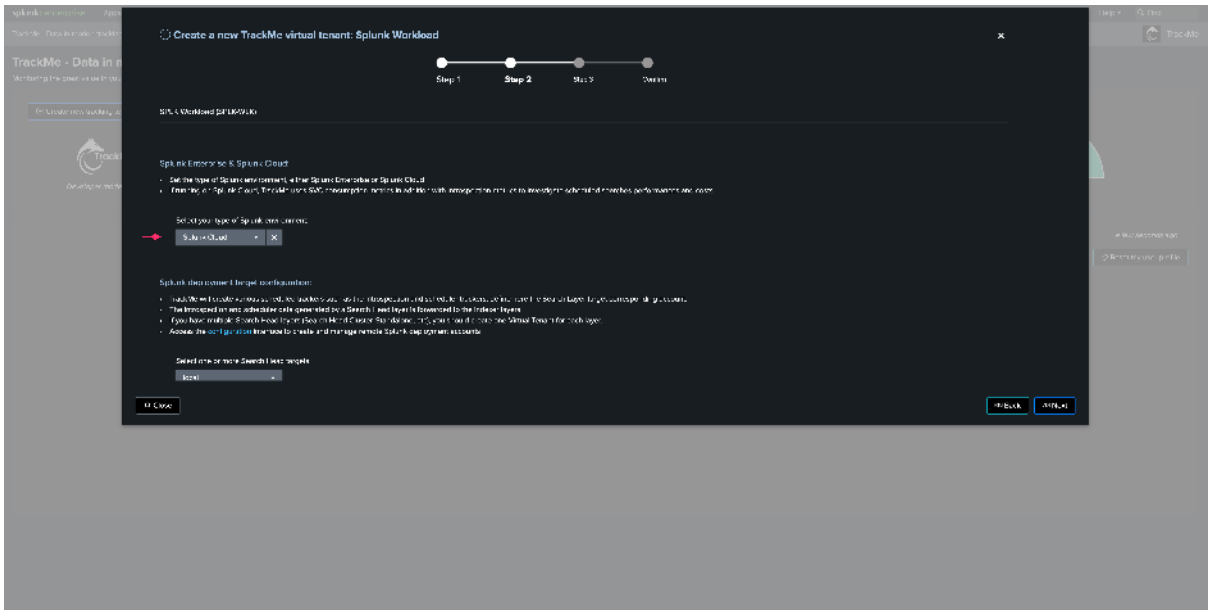
```
[ search index=summary source="splunk-svc-consumer" search_head_names="es" earliest=-
↪24h latest=now | stats count by usage_source | table usage_source | rename usage_
↪source as host | format | eval search = if(search="NOT ()", "host=*", search) ]
```

Once you have validated the connectivity, you can continue and start the configuration of the Workload tenant.

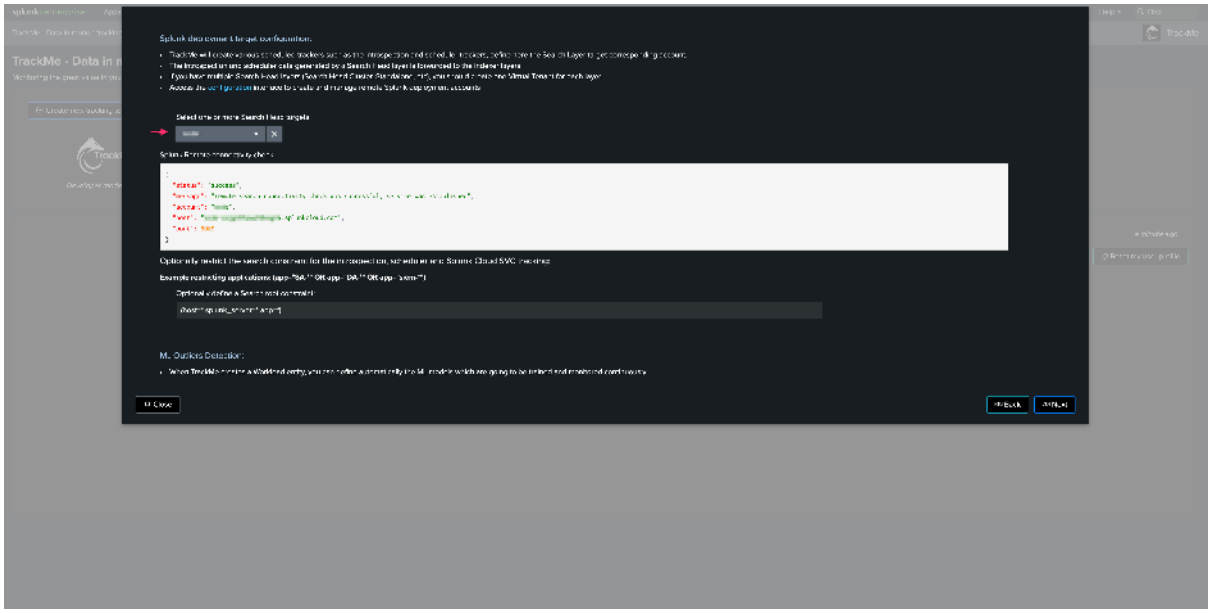
Step 2: create and configure the Workload tenant

Open the assistant wizard, and proceed as follows:

- Select “Splunk Cloud” in the Splunk environment dropdown:



- Select the Remote deployment target:



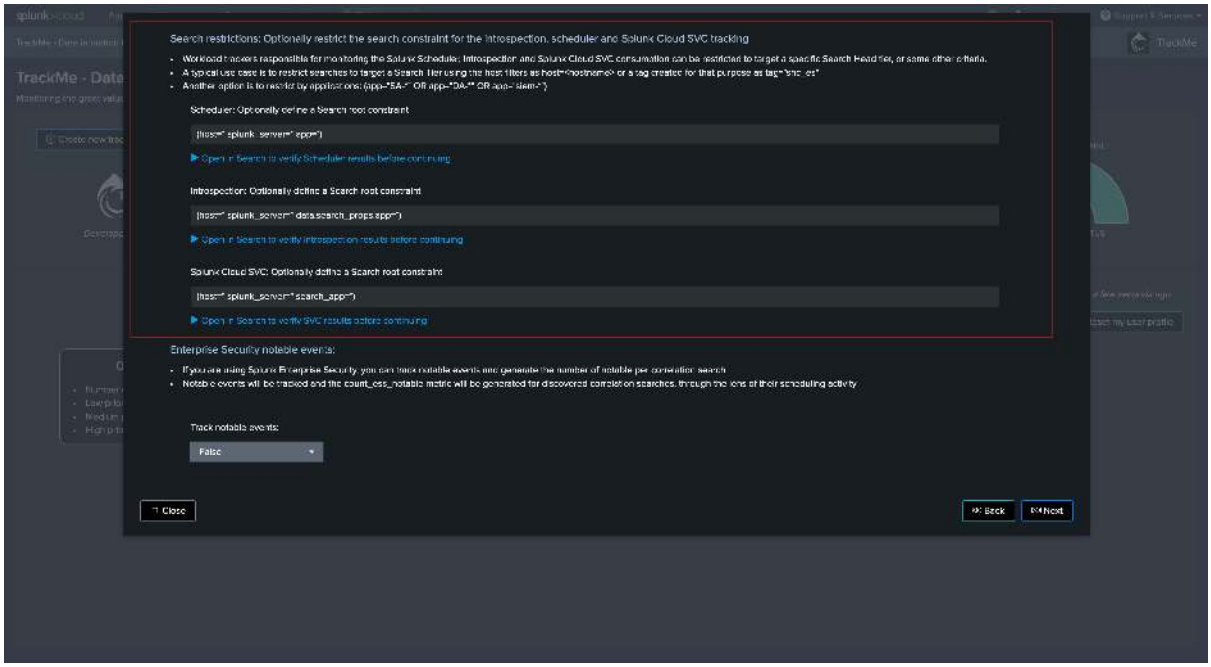
- Set the search restrictions:

Hint

Search restrictions per Workload component

- Since the release 2.0.57, you can define the search restrictions per component, to ease and improve the configuration process
- You can directly validate that the search will return results by clicking on the search preview link

- This opens a pre-populated search interface, it also sets the restrictions and calls splunkremotesearch if the search has to run against a remote Search Head tier



Complete the rest of the setup, it can take up to 5 minutes before entities start to be visible in the Workload tenant.

Workload component in Splunk Enterprise

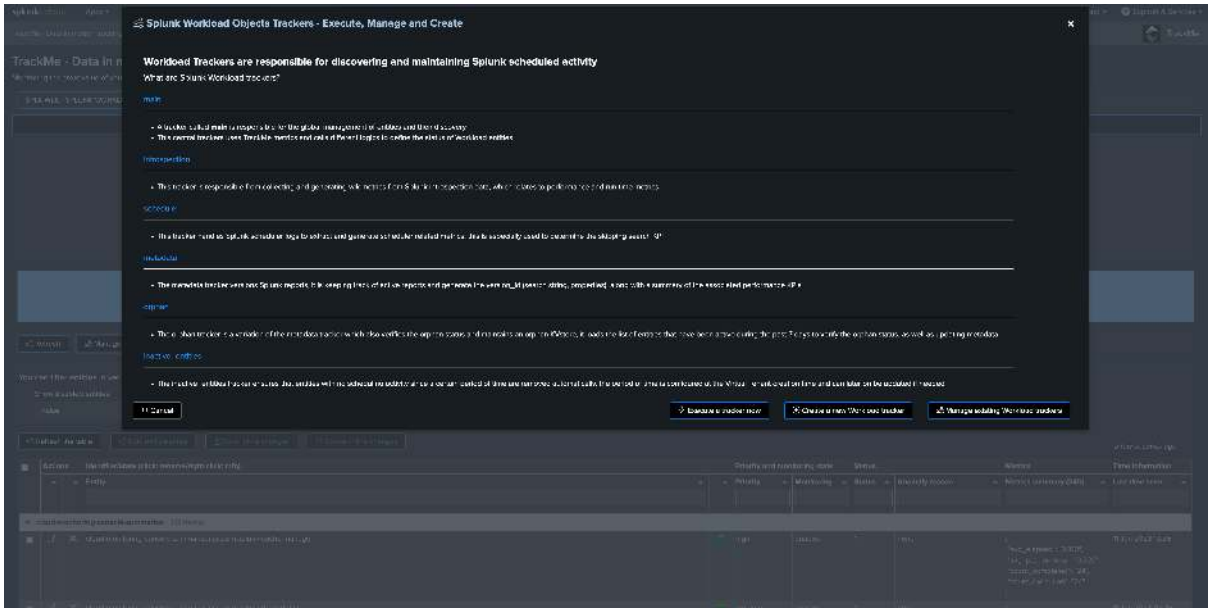
If you are running Splunk Enterprise, then the setup slightly depends on your deployment:

- If you have multiple Search Head tiers, you can use the same tag approach as documented in the Splunk Cloud setup above
- You can also create a custom indexed or search time field on the Search Head tier level, which automatically identifies the Search Head tiers to be targeted
- Use different combinations up to your preference

7.19.3 Managing Workload Trackers

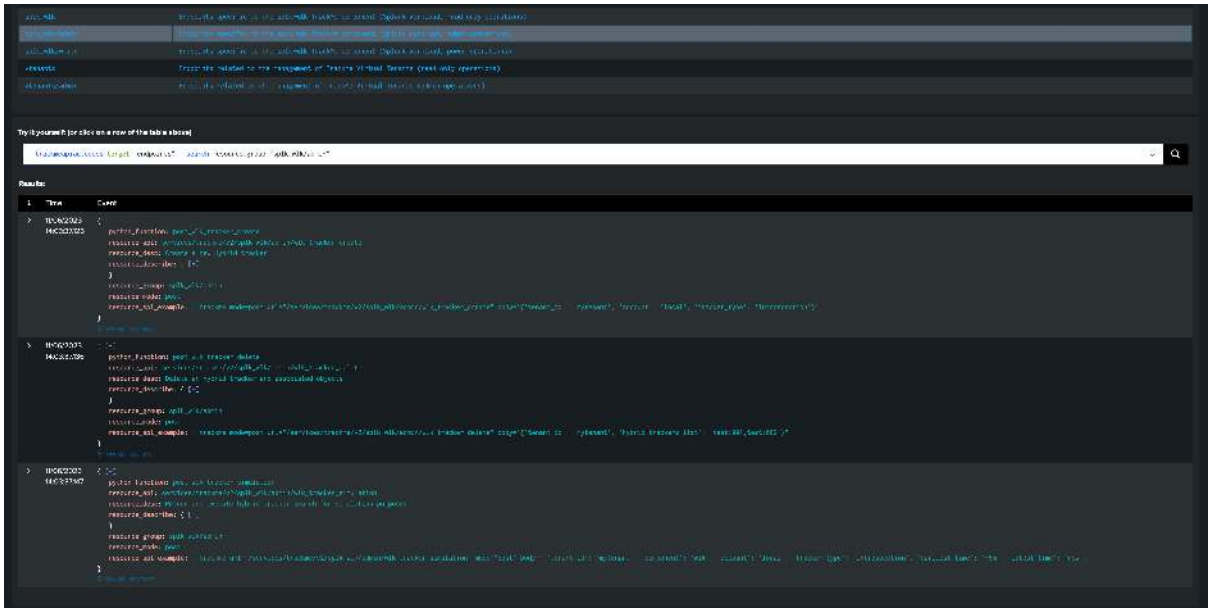
There are a certain number of Workload trackers which are created automatically by the wizard, depending on your Splunk environment.

Through the trackers management user interface, you can eventually add new trackers if needed, either if you want to re-create the trackers or extend the tracking to additional context:



However, the general recommendation is to dedicate a Workload Tenant to a given Search Head tier.

The REST API UI also allows you to manage all types of Workload trackers:



7.19.4 Managing Workload detection

When a Workload Virtual Tenant is created, TrackMe automatically creates a macro which defines the main status definition behaviour for that tenant.

The macro is named as follows:

- `trackme_wlk_set_status_tenant_<tenant_id>`

You can update this macro to modify how TrackMe defines the status of Workload entities, for instance these lines define the grace time period for the detection of delayed entities:

```
``` Calculate the delta in sequence between now and the last execution compared␣
↪against the requested cron schedule sequence, add 1h of grace time, detect if the␣
↪execution has been delayed ```
| eval status=if(cron_exec_sequence_sec>0 AND ( now()-last_seen > (cron_exec_sequence_
↪sec + 3600) ), 2, status)
```
```

This also defines the level of skipping searches (percentage) which leads entities to turn orange or red:

```
``` If there are skipping searches, define two levels of alerting, less than 5% is 3␣
↪(orange), more is 2 (red) ```
| eval status=case(skipped_pct>0 AND skipped_pct<5, 3, skipped_pct>0 AND skipped_pct>
↪=5, 2, 1=1, status)
```

The macro is part of the life cycle of the tenant's knowledge objects, therefore it will be removed automatically if the tenant itself is deleted.

This macro is executed by the "main" tracker, which naming convention is the following:

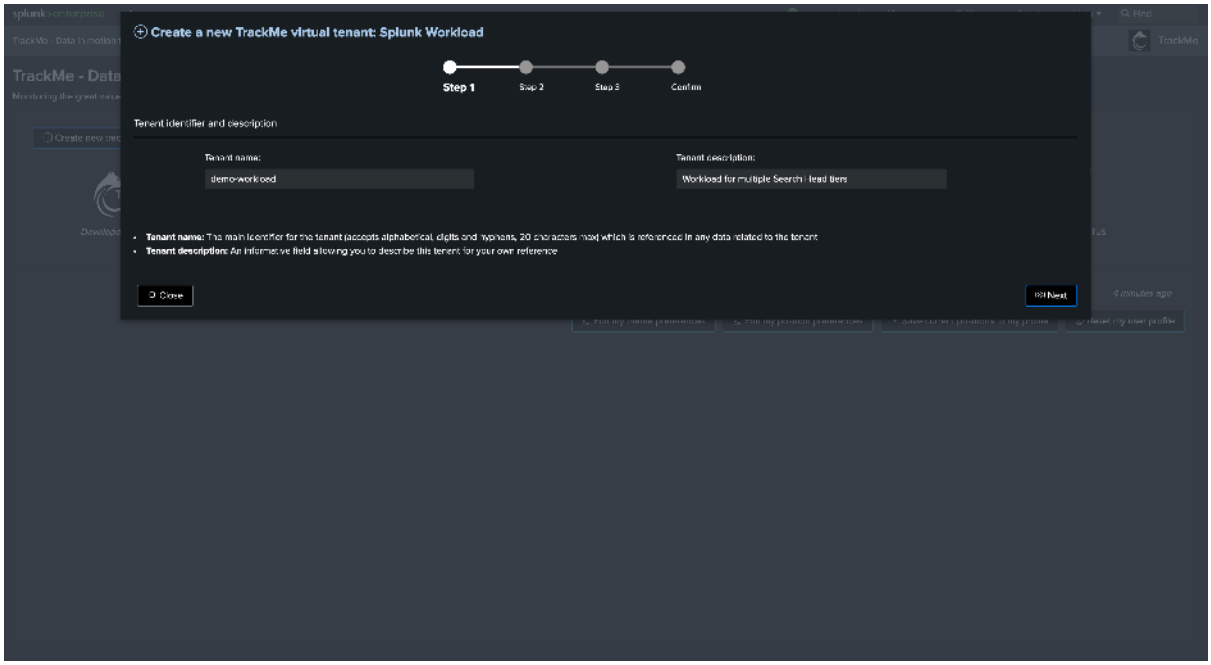
- `trackme_wlk_hybrid_main_ce555e75db1643a_wrapper_tenant_<tenant_id>`

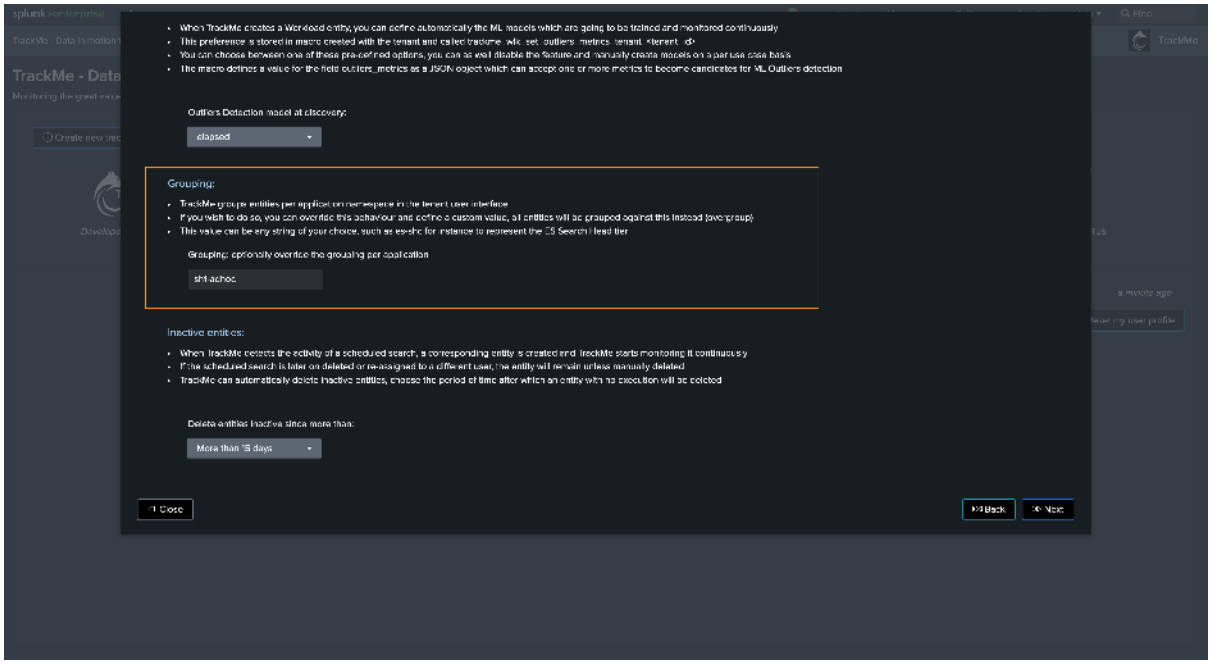
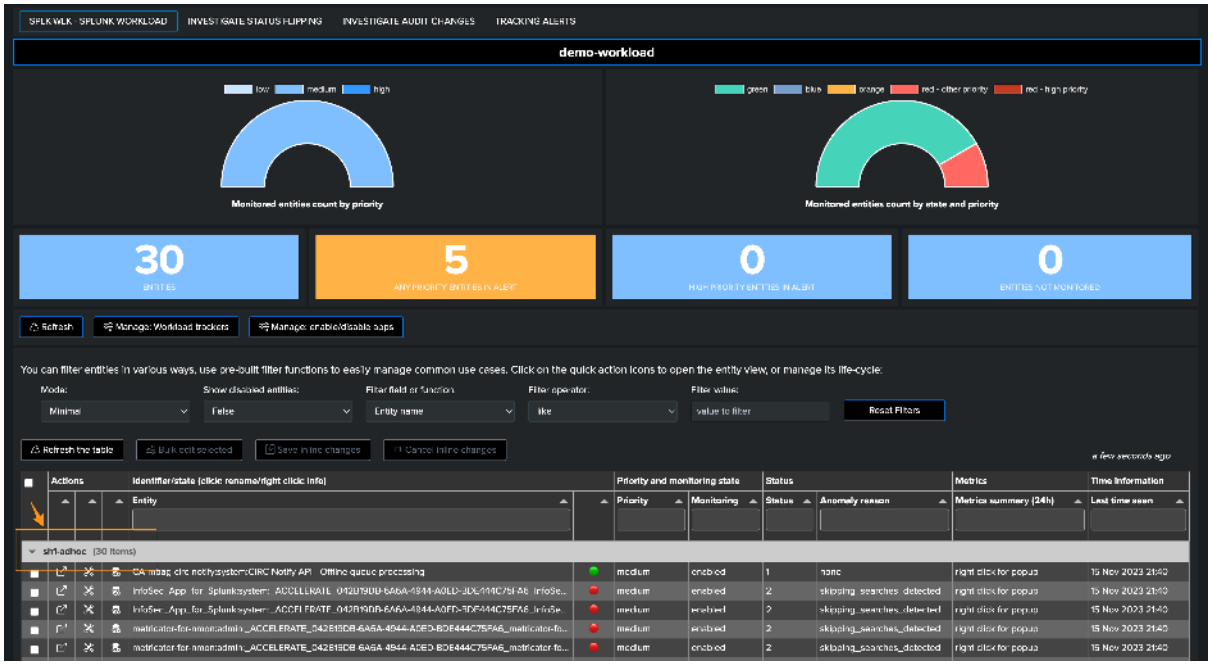
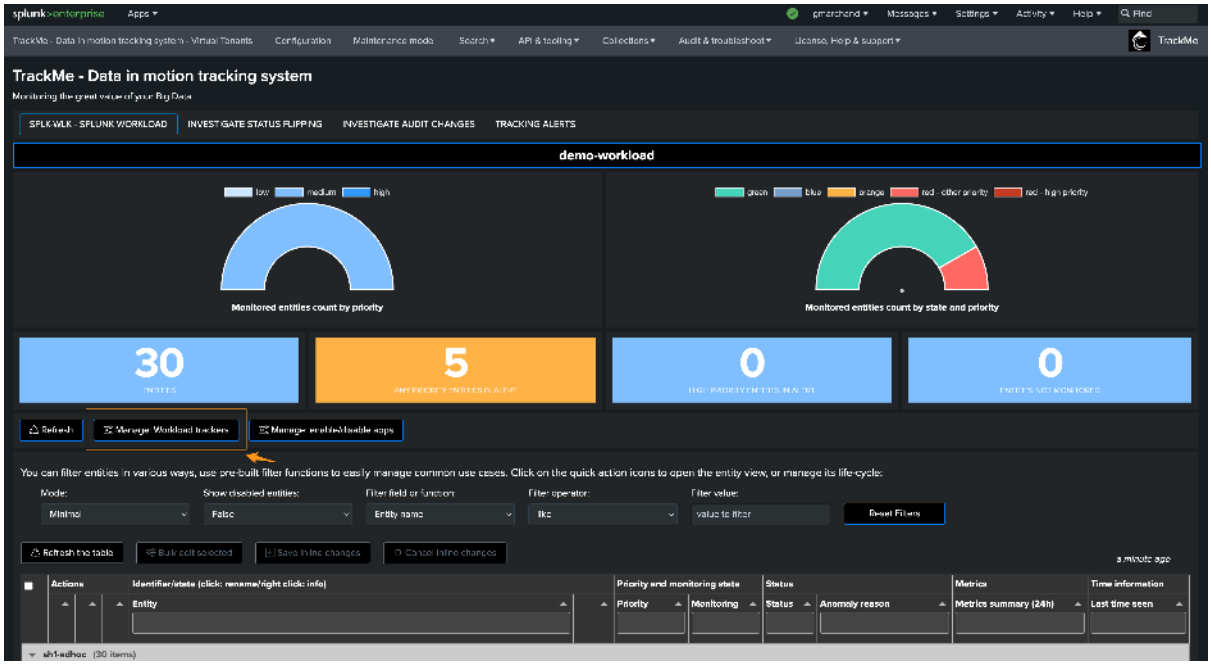
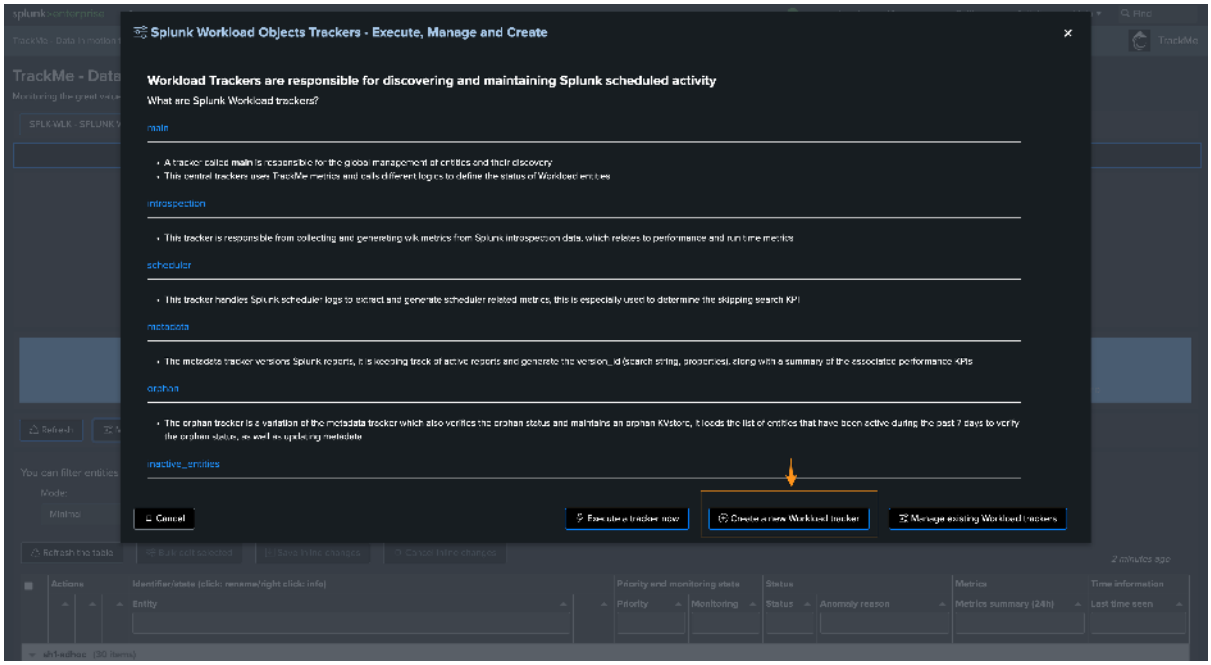
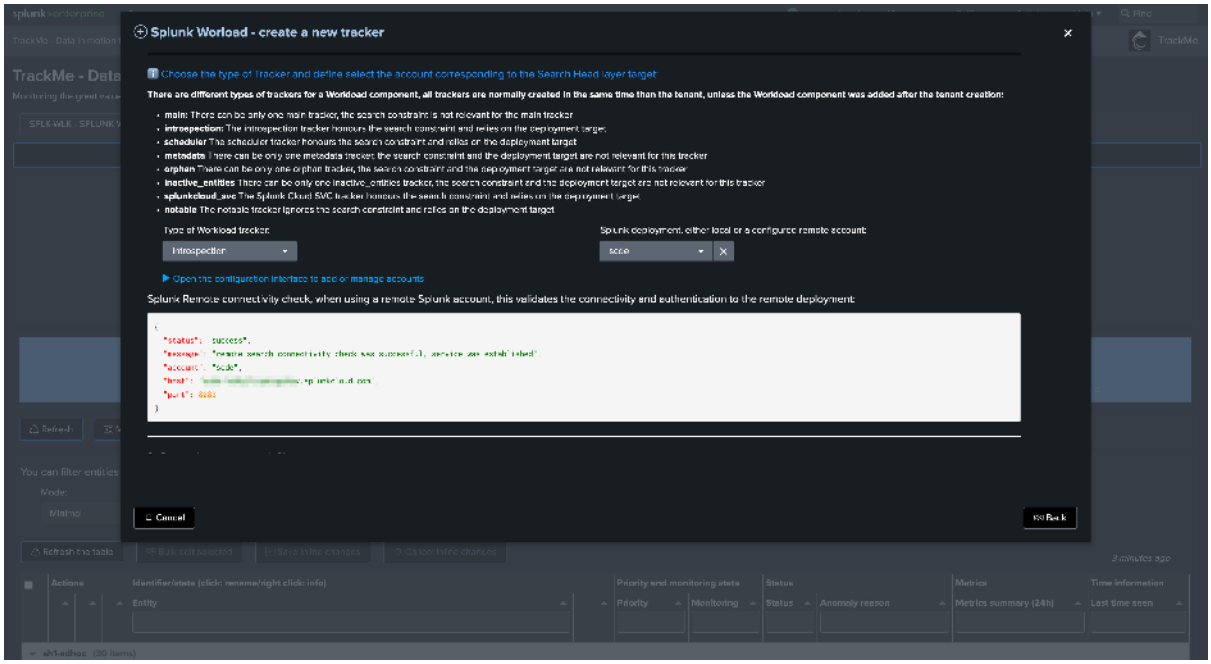
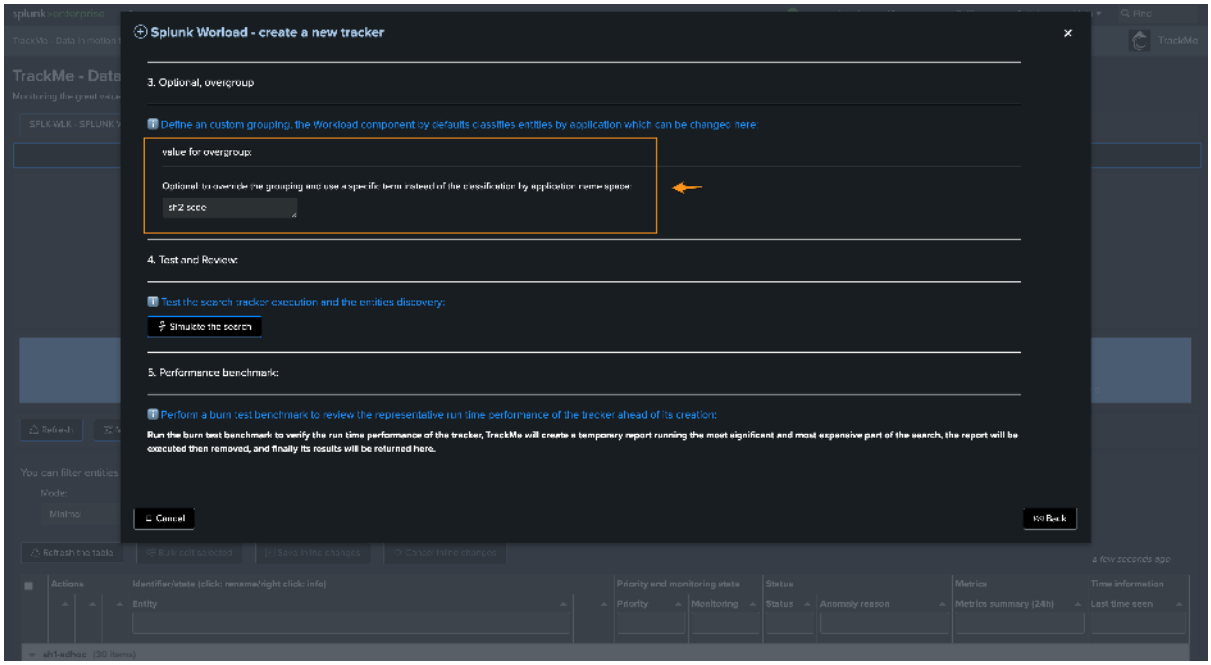
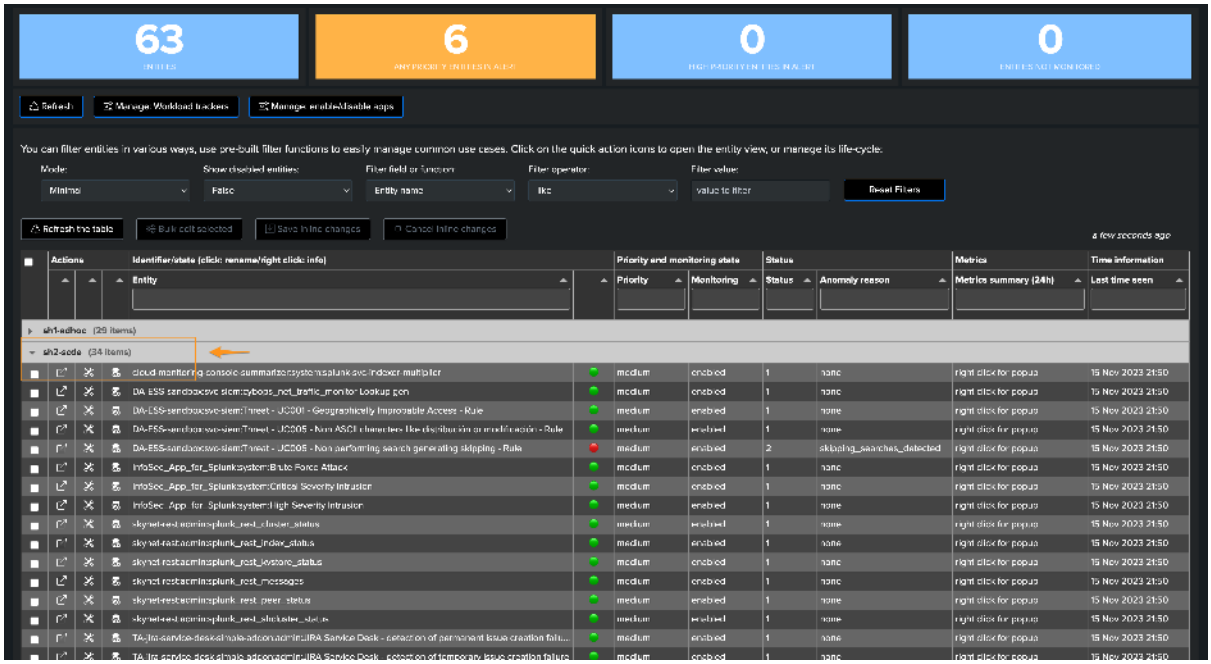
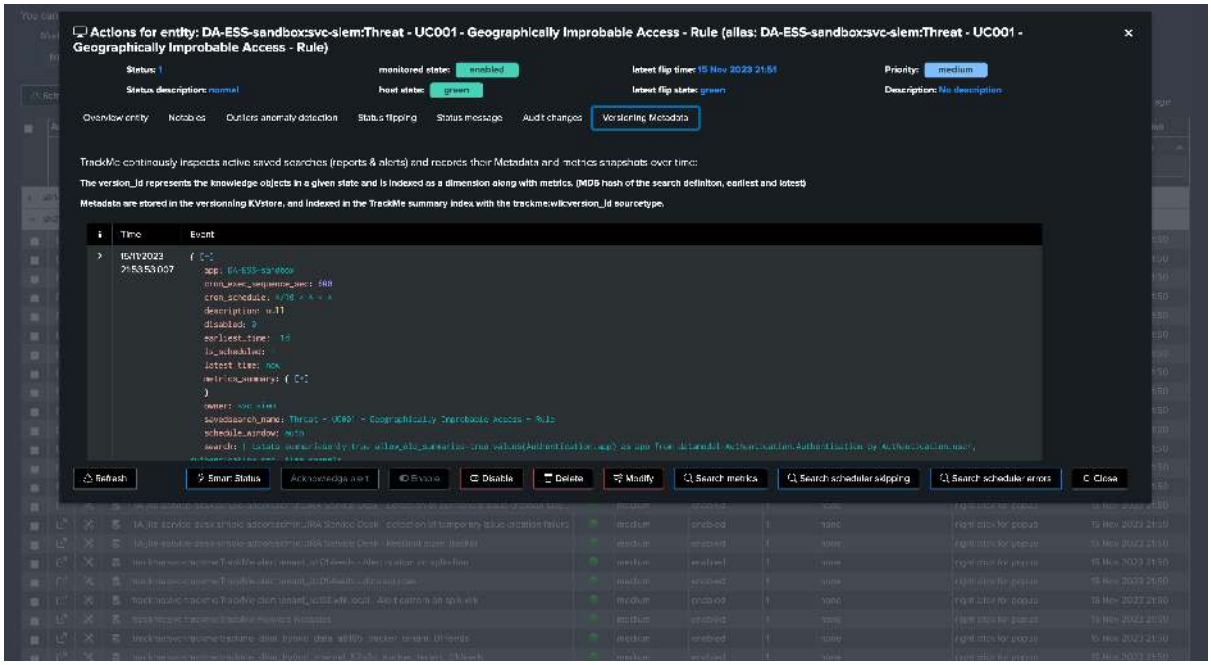
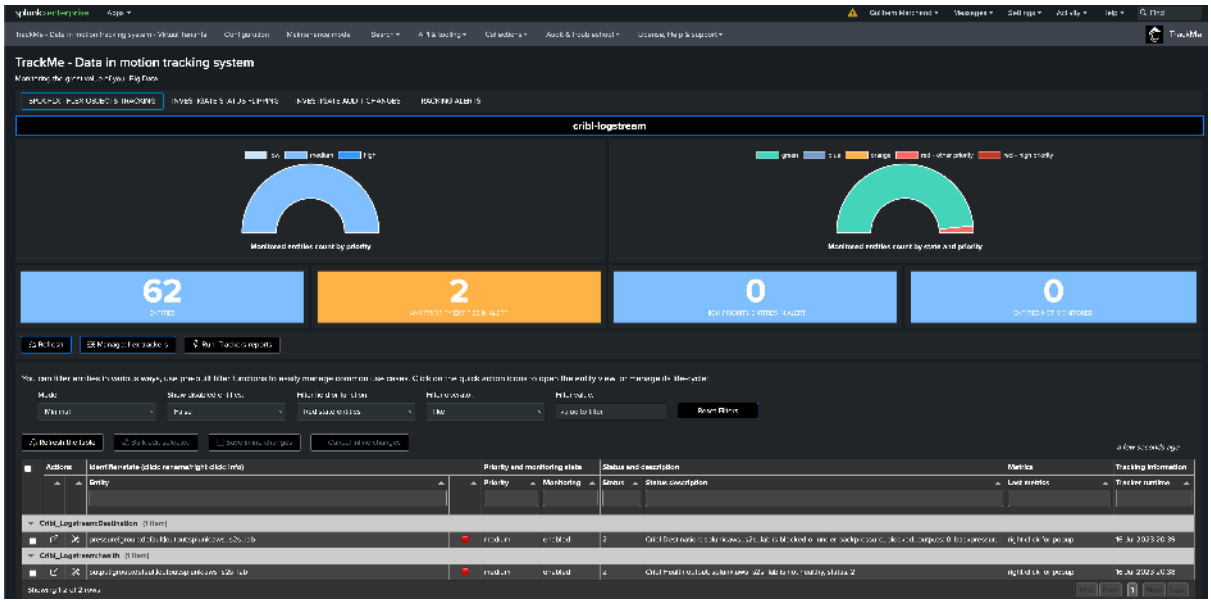
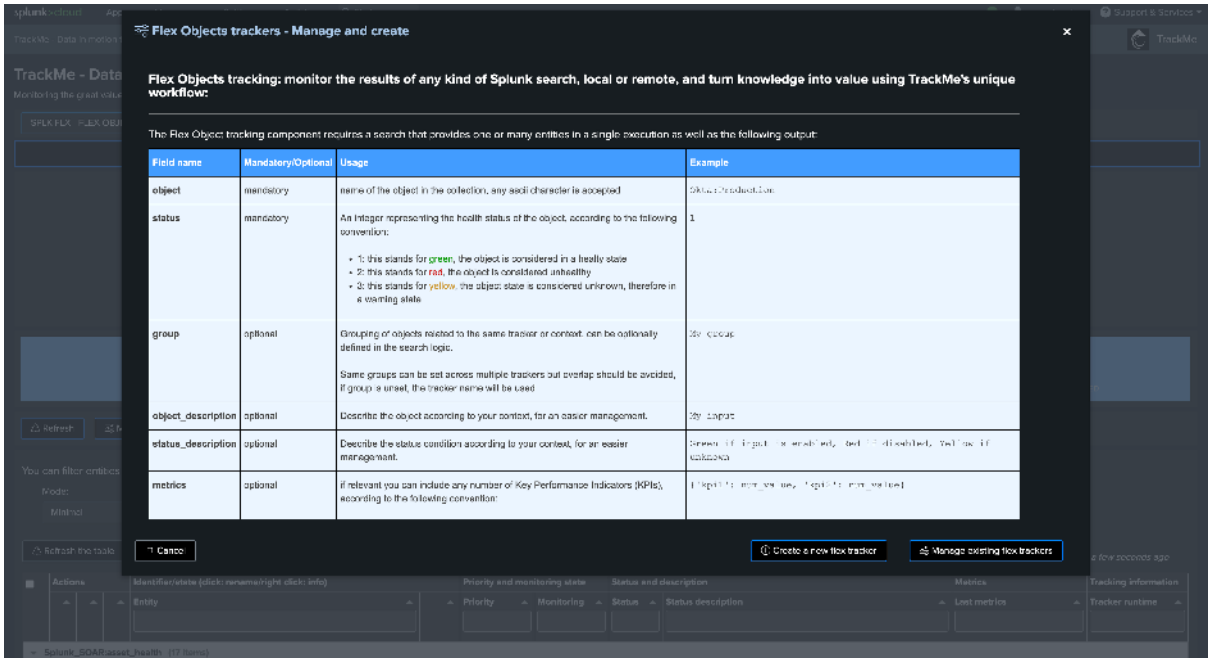
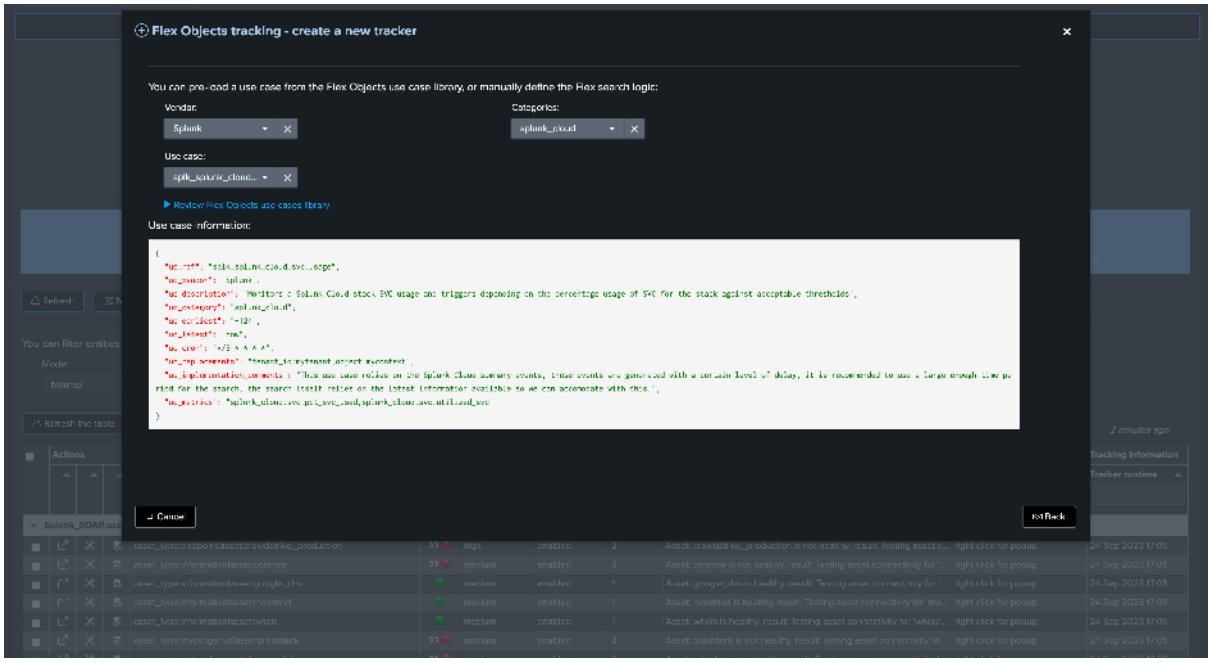
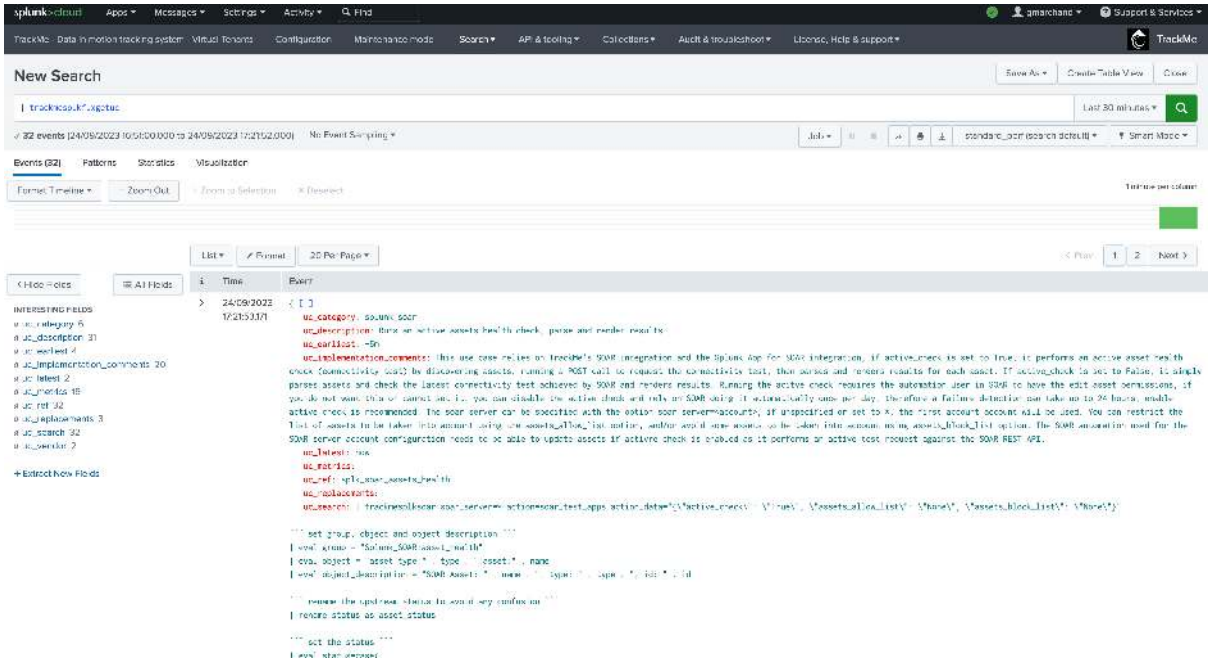
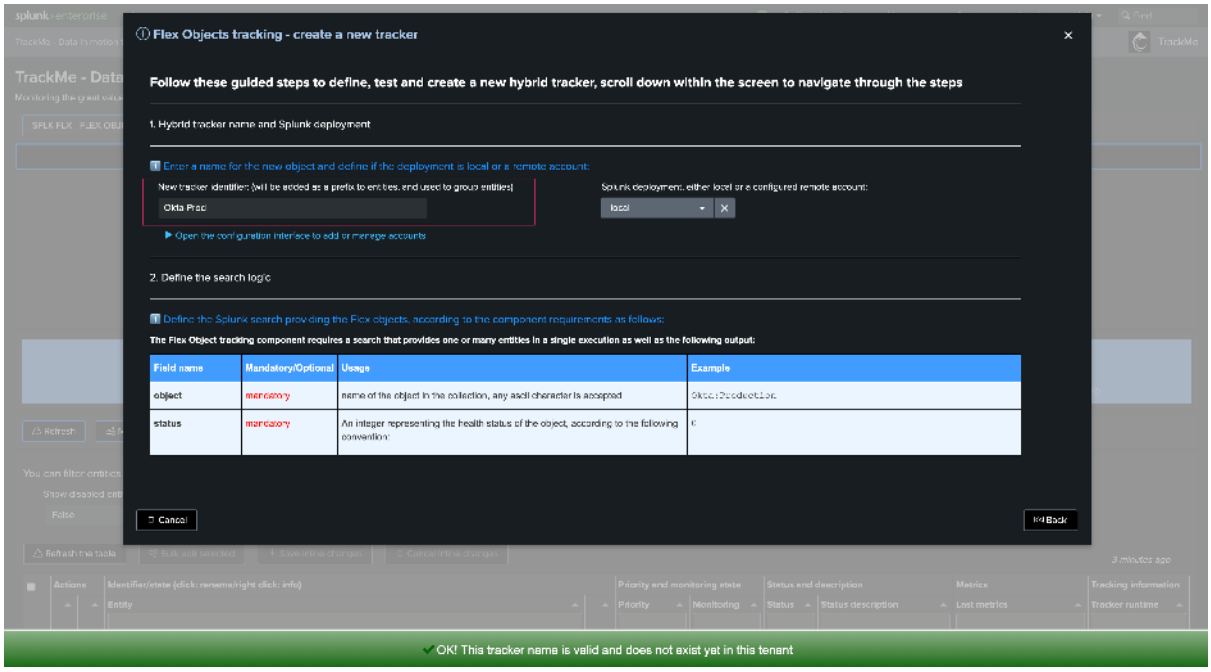
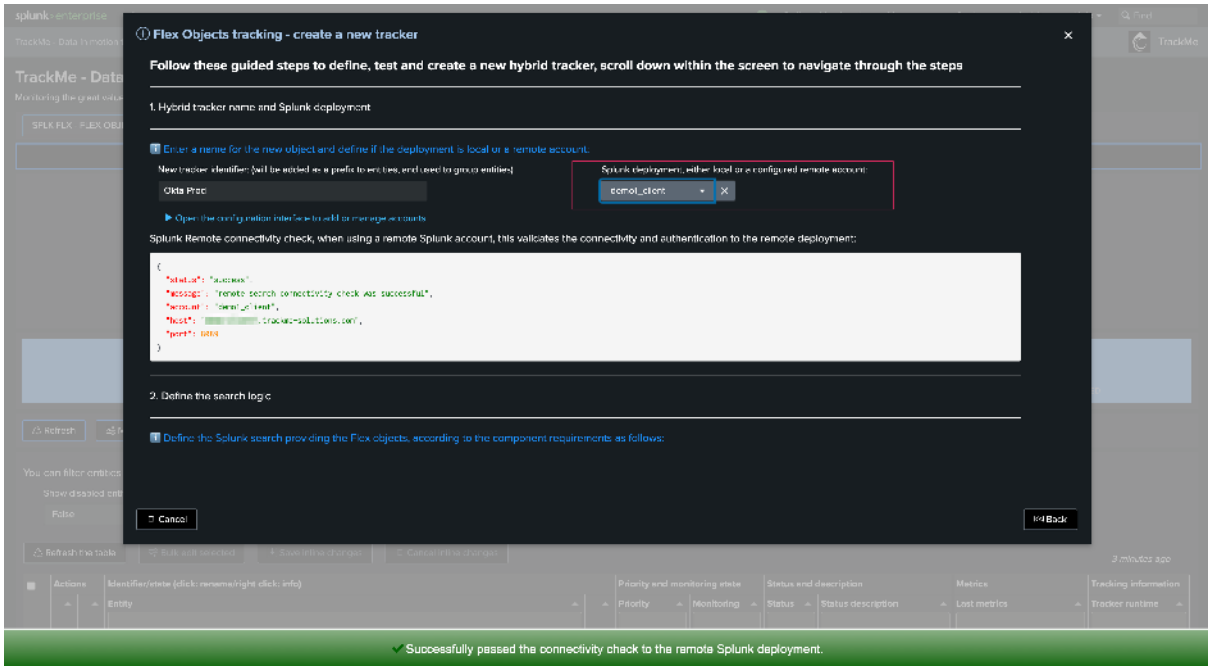
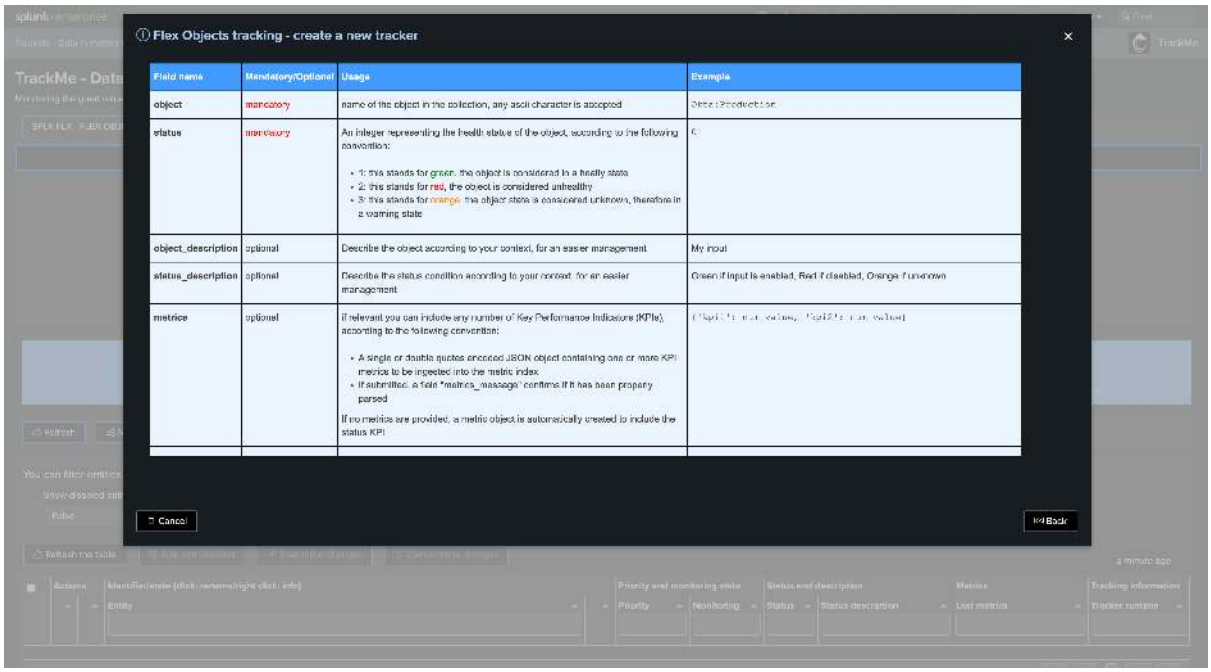
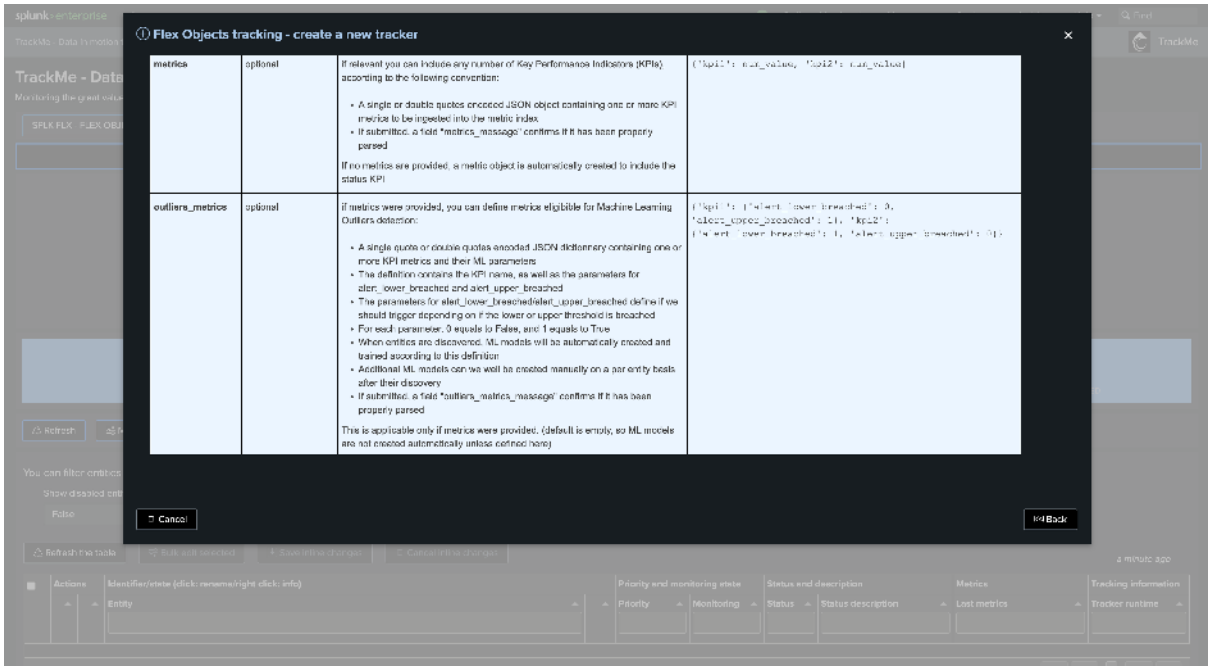
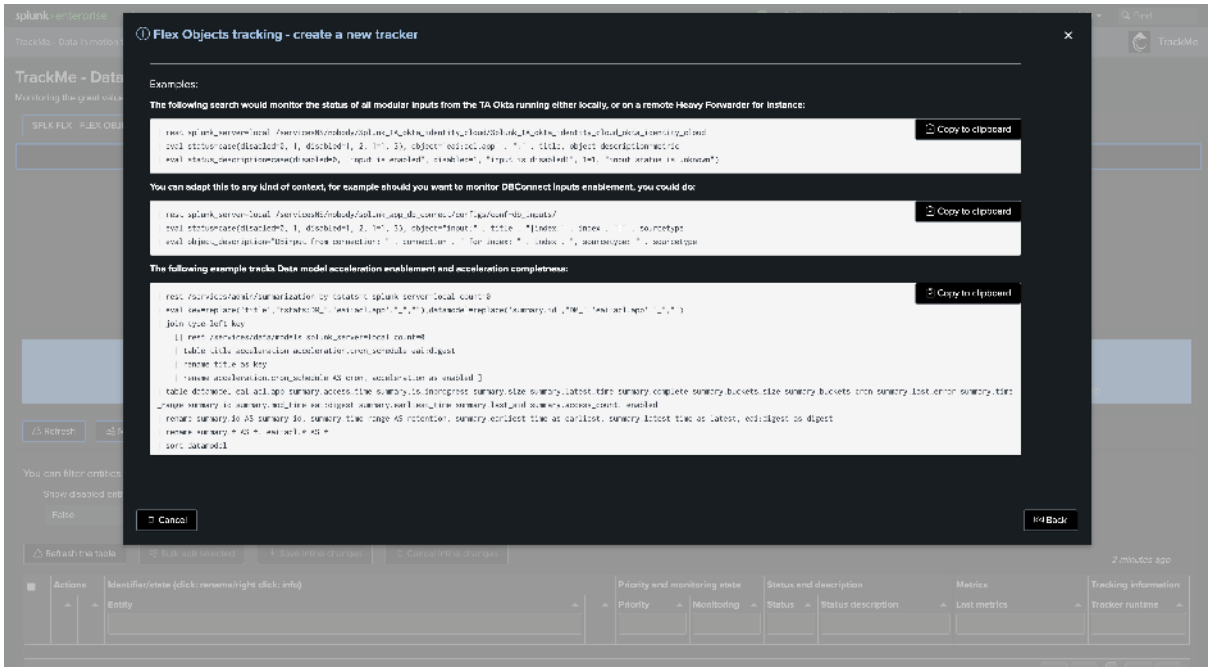
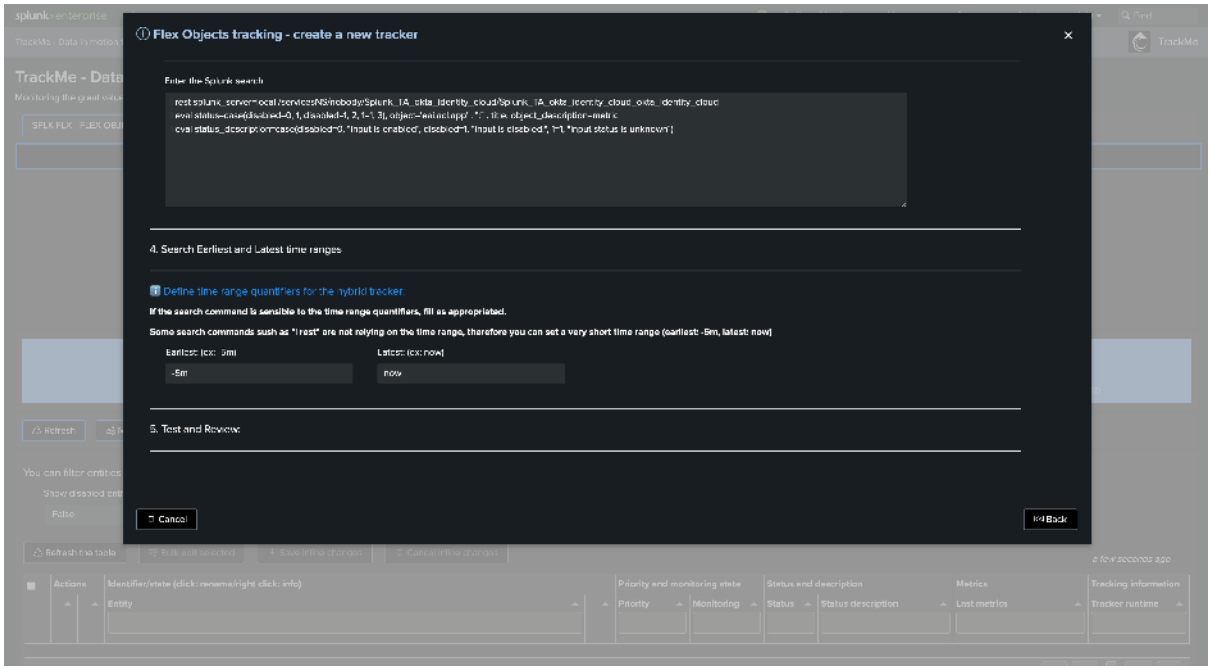
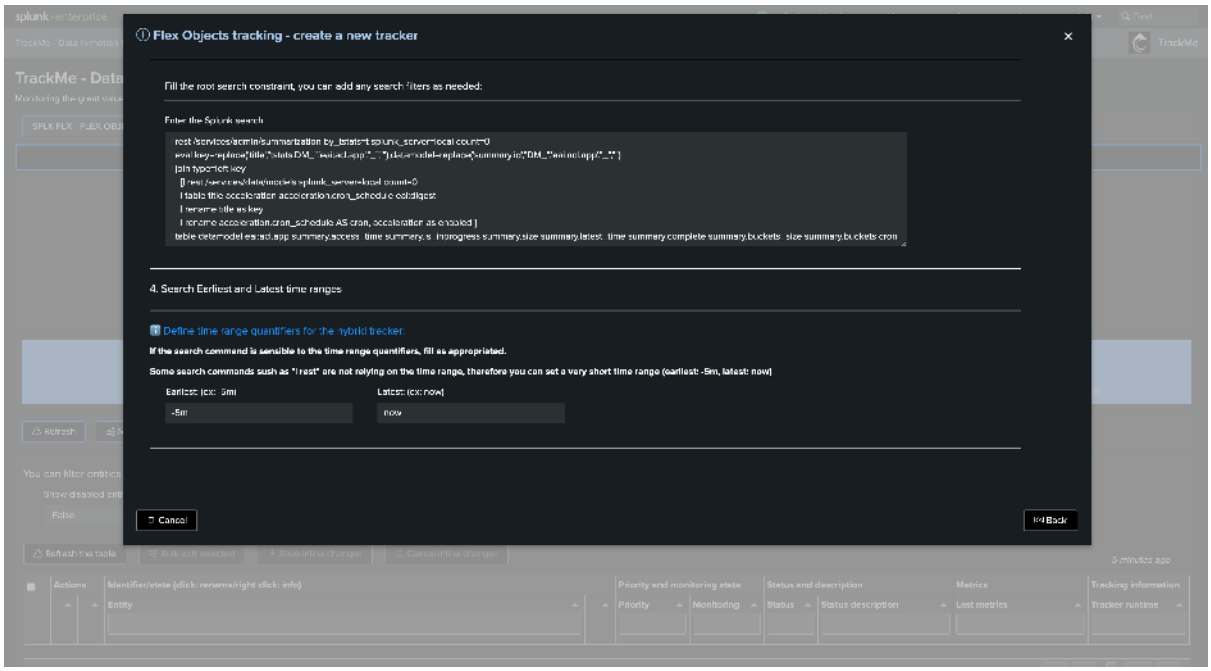
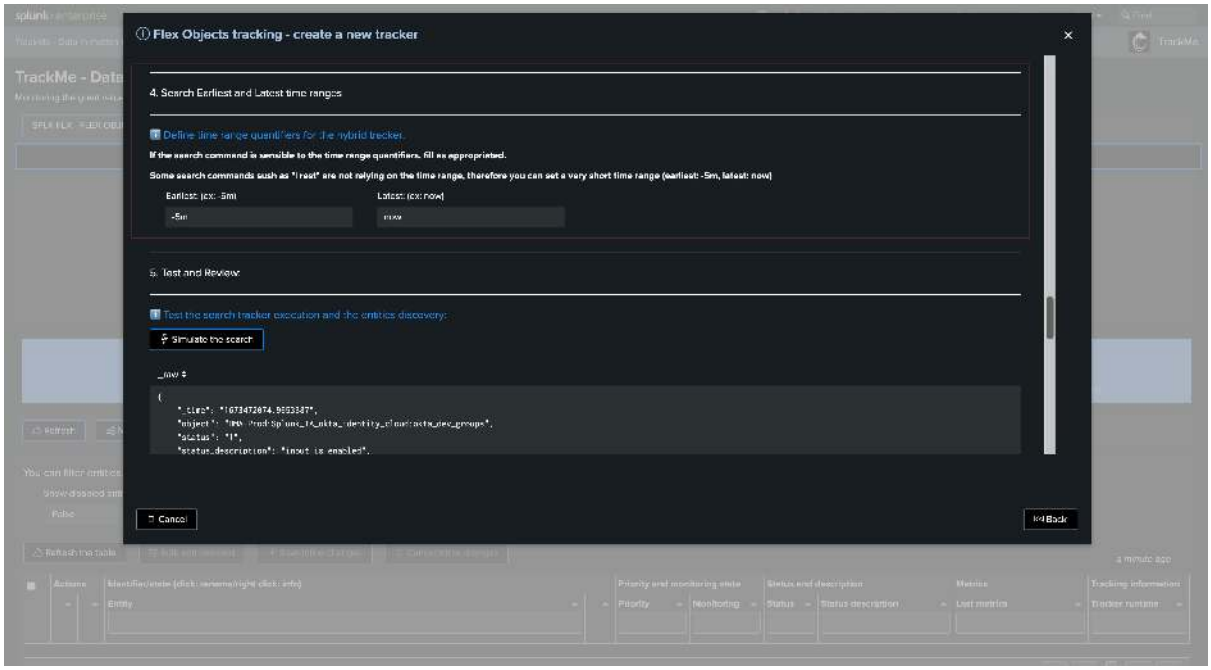
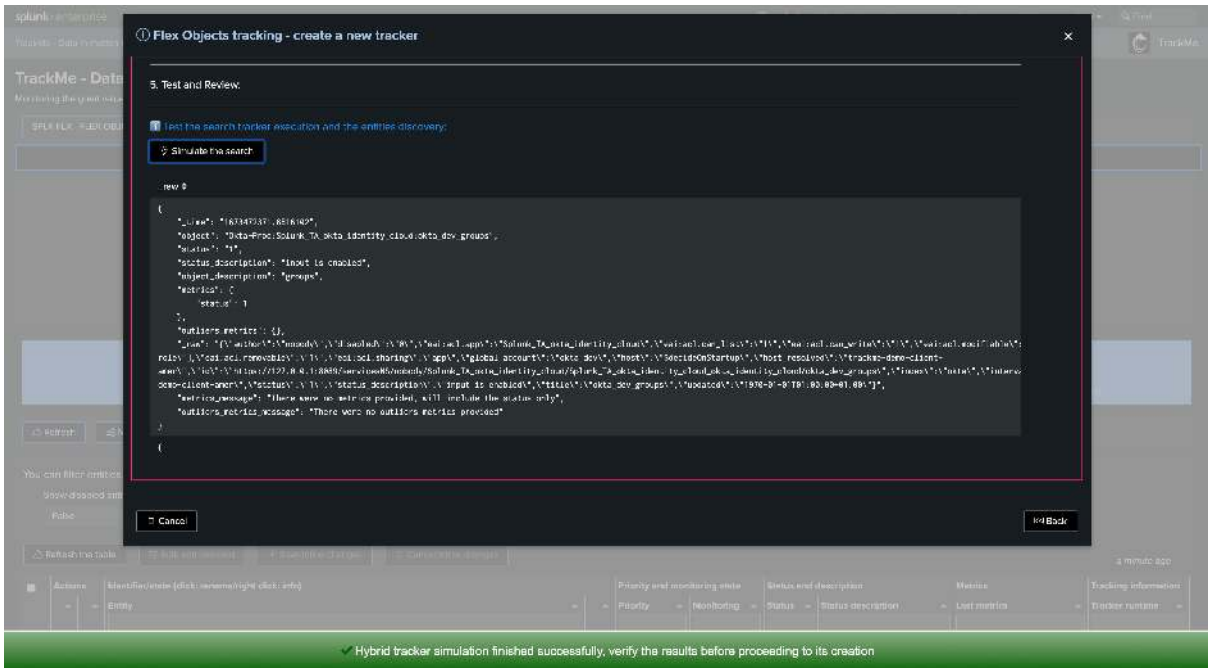
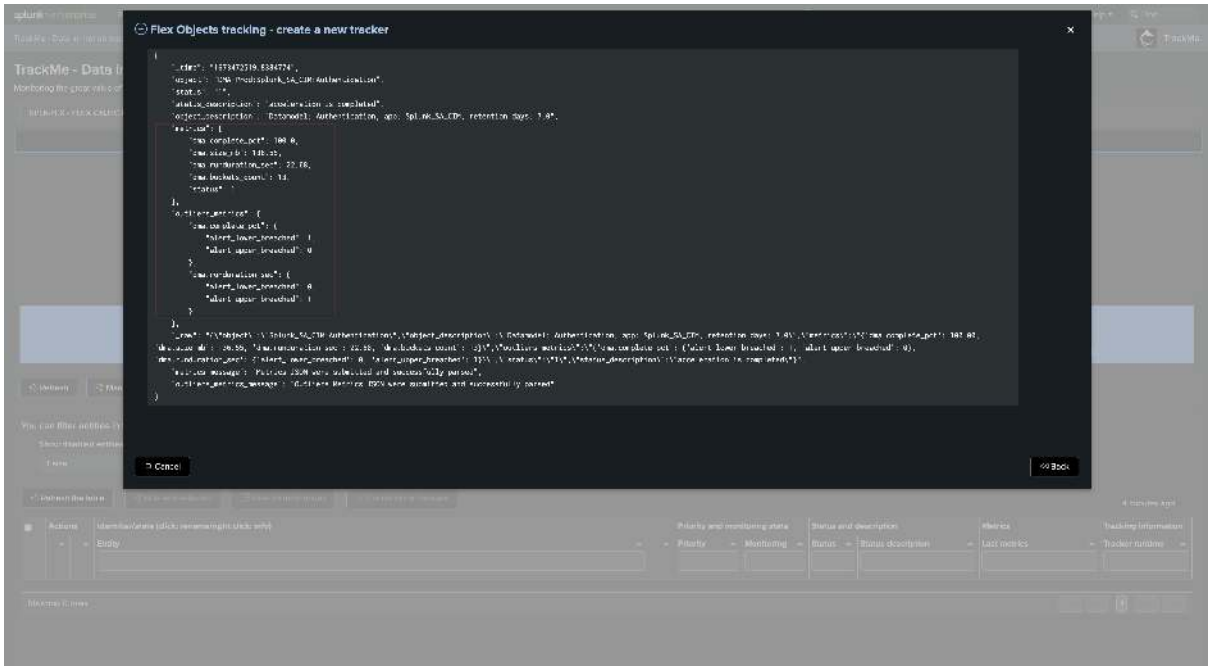
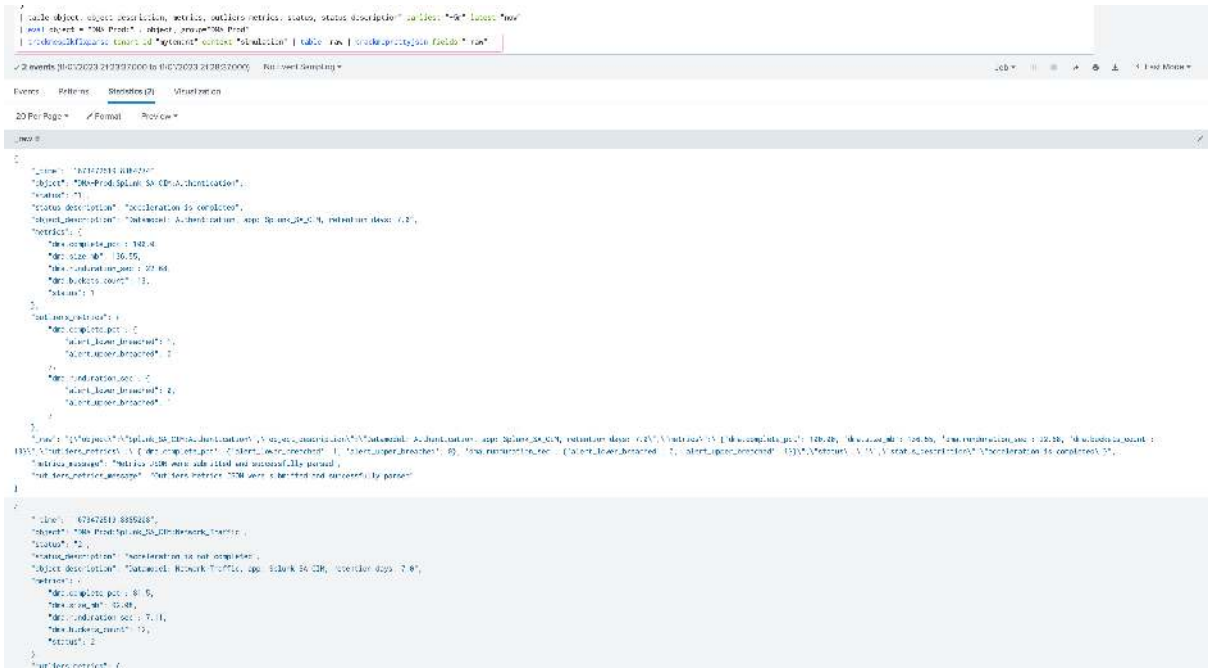
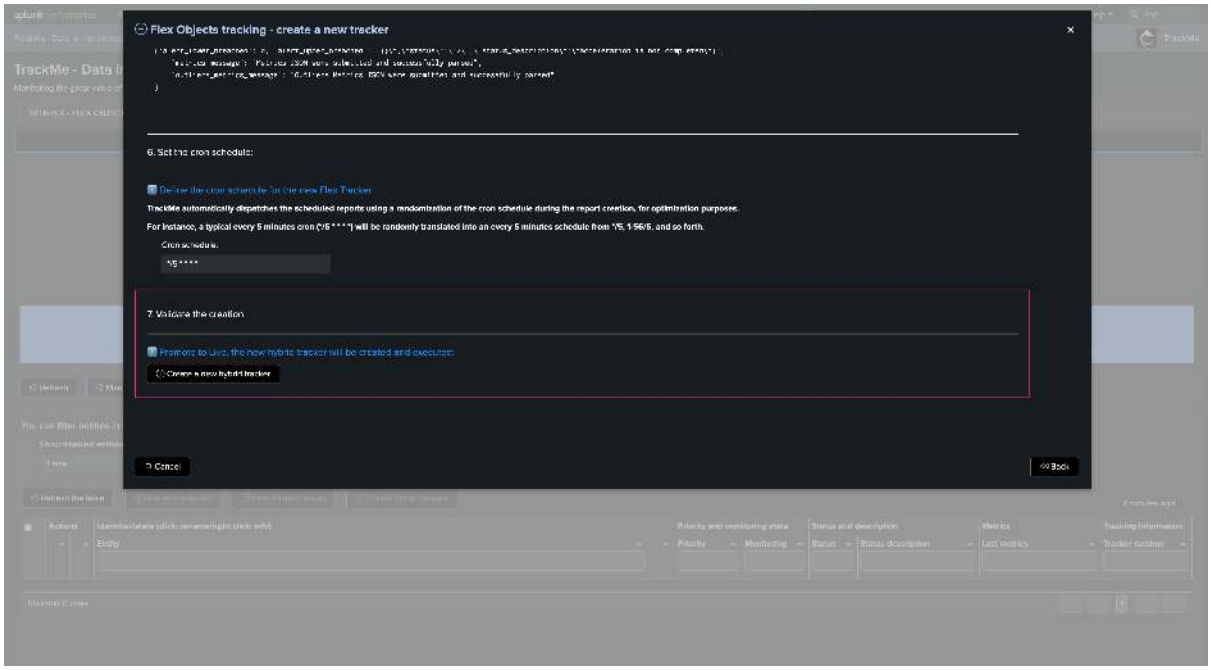
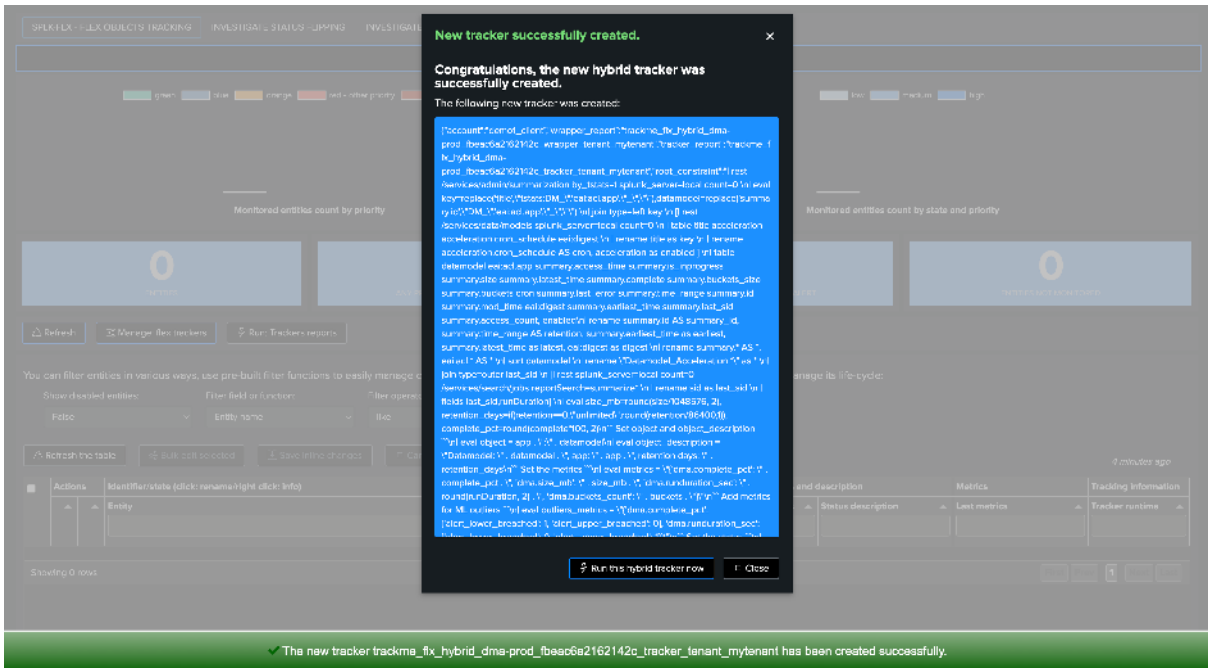
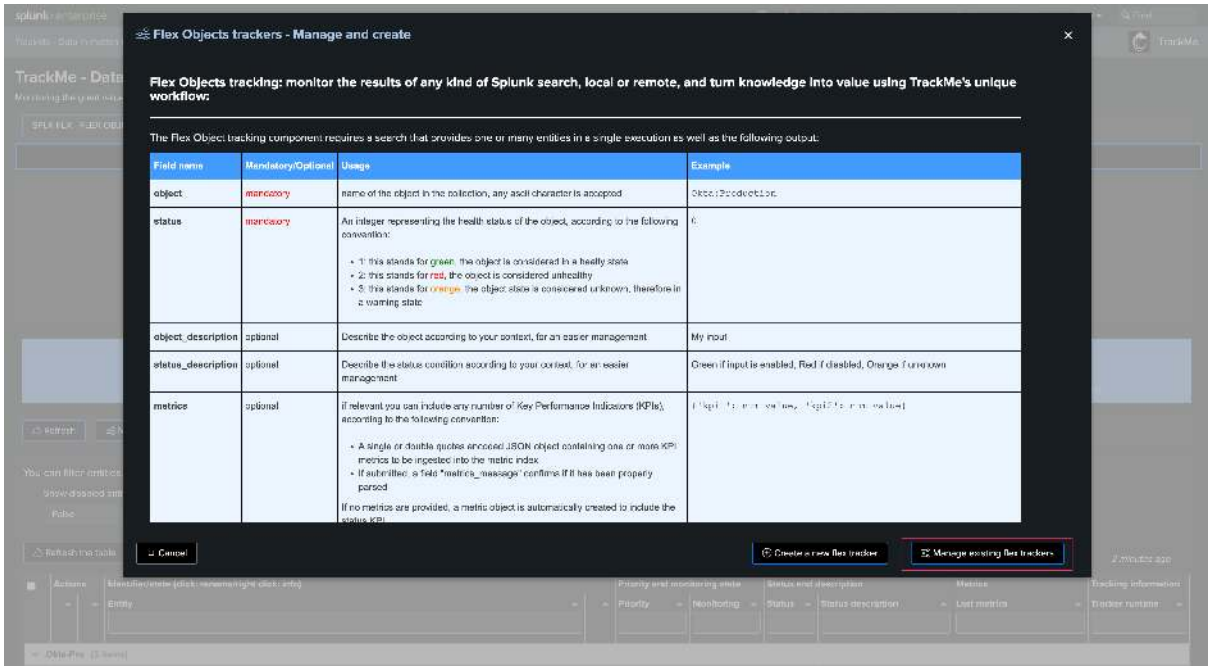
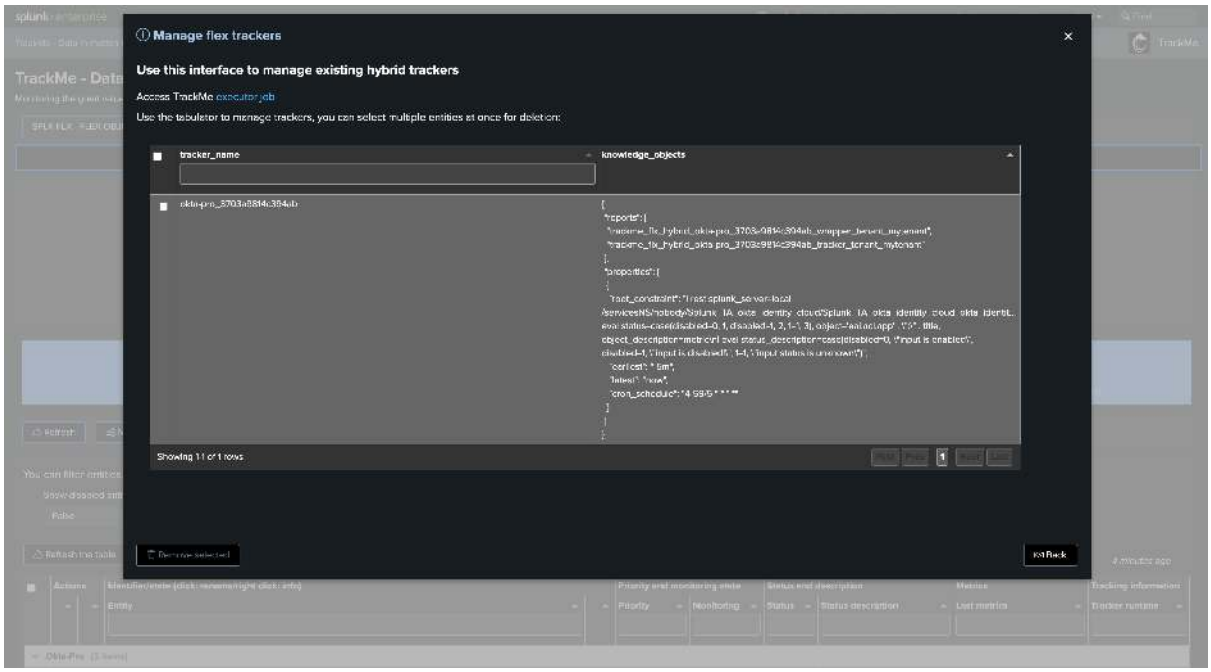
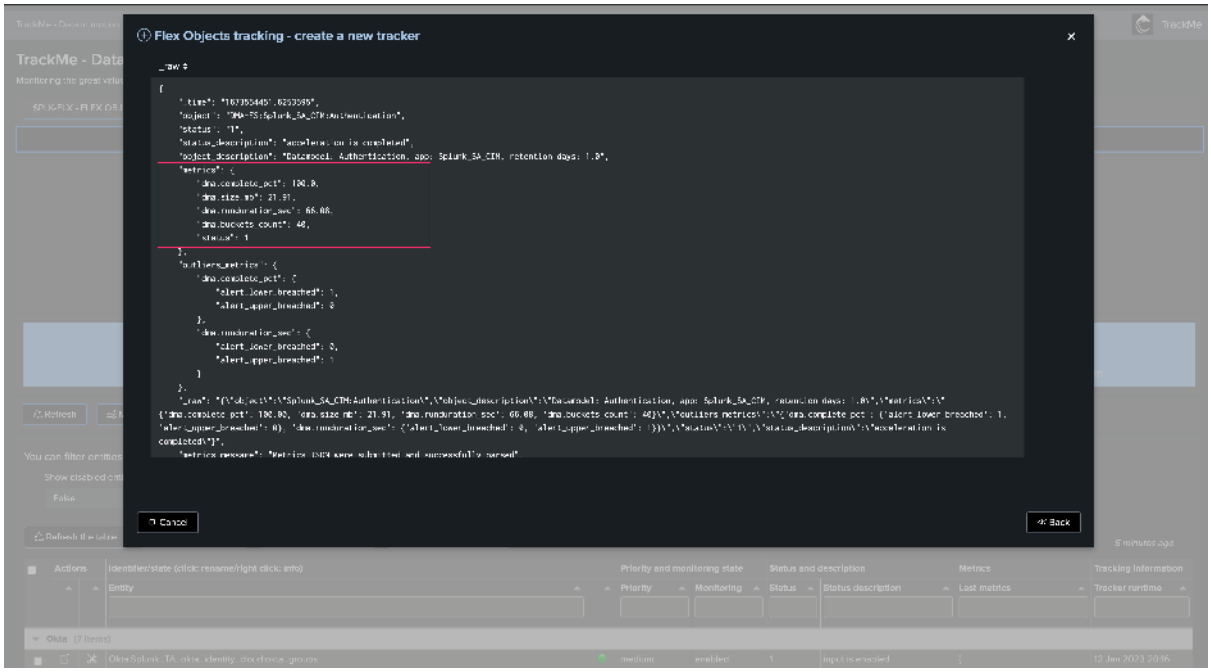
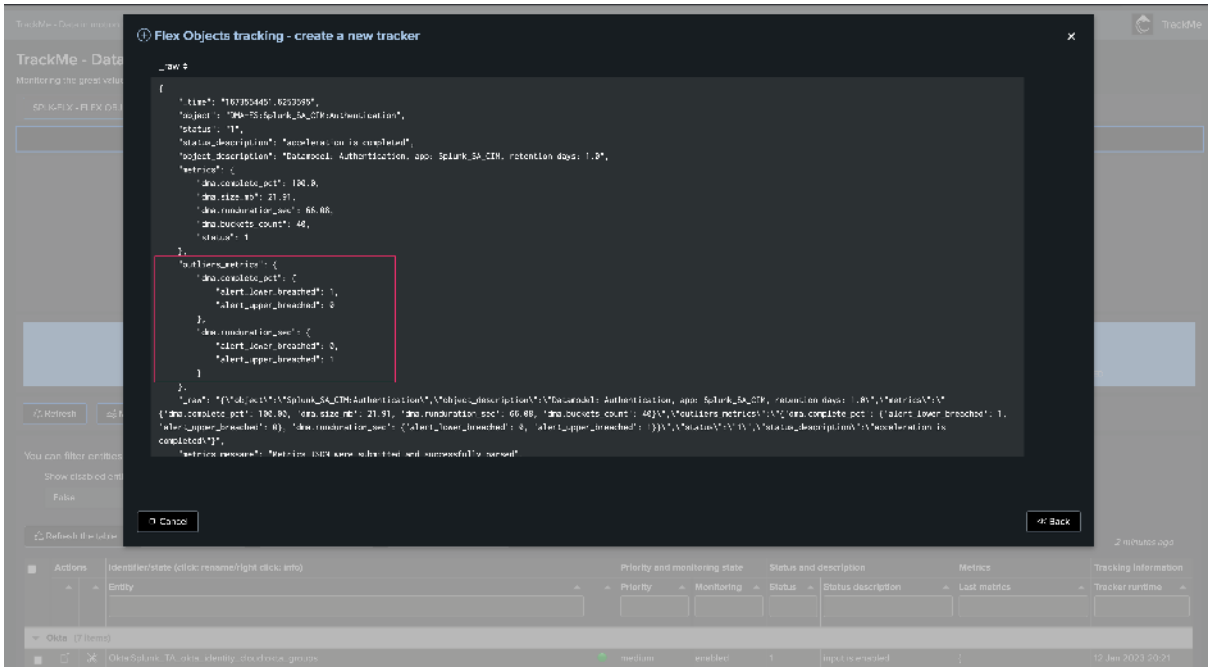
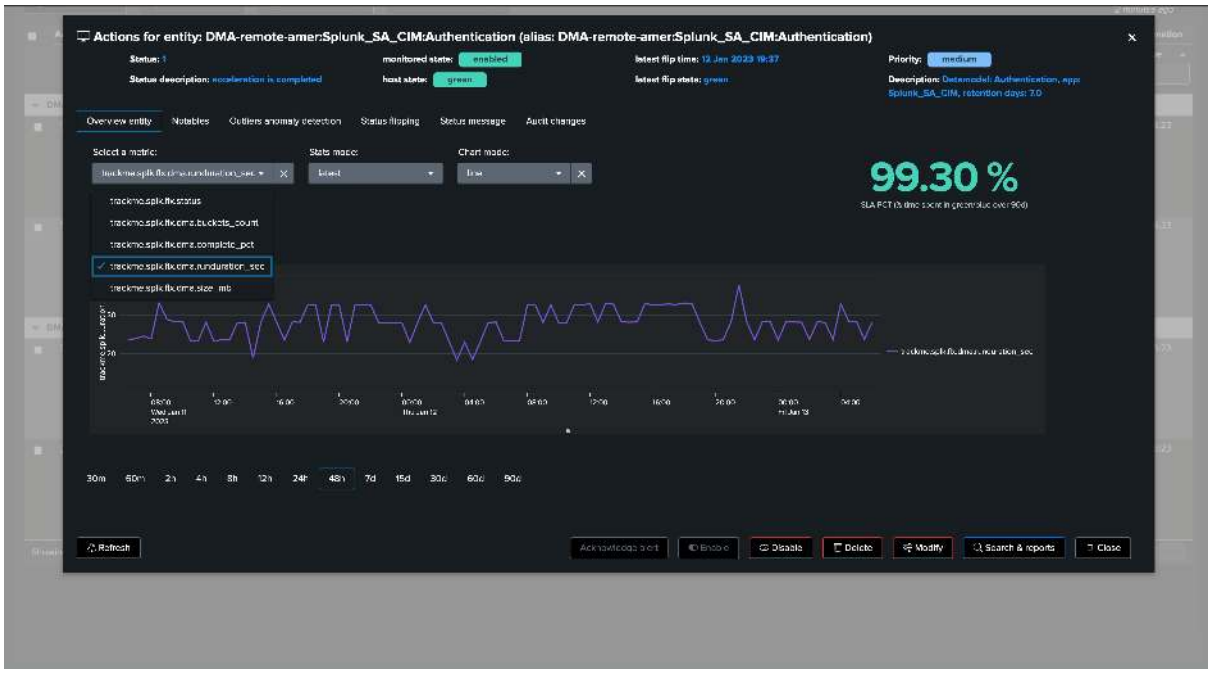
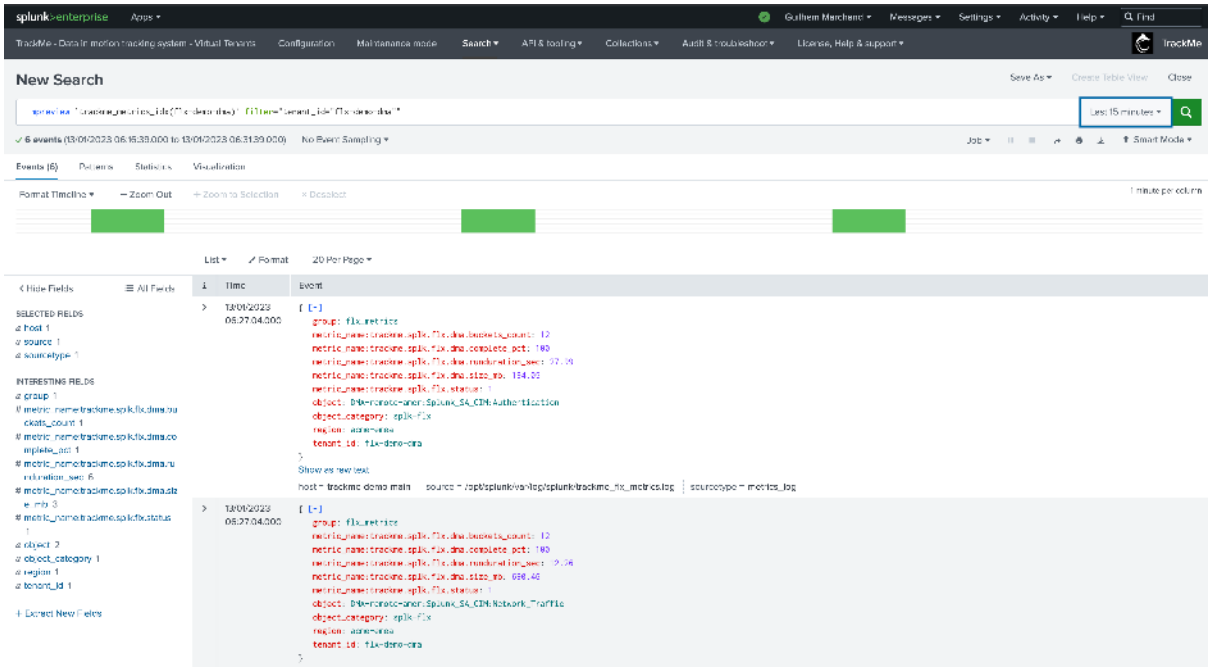
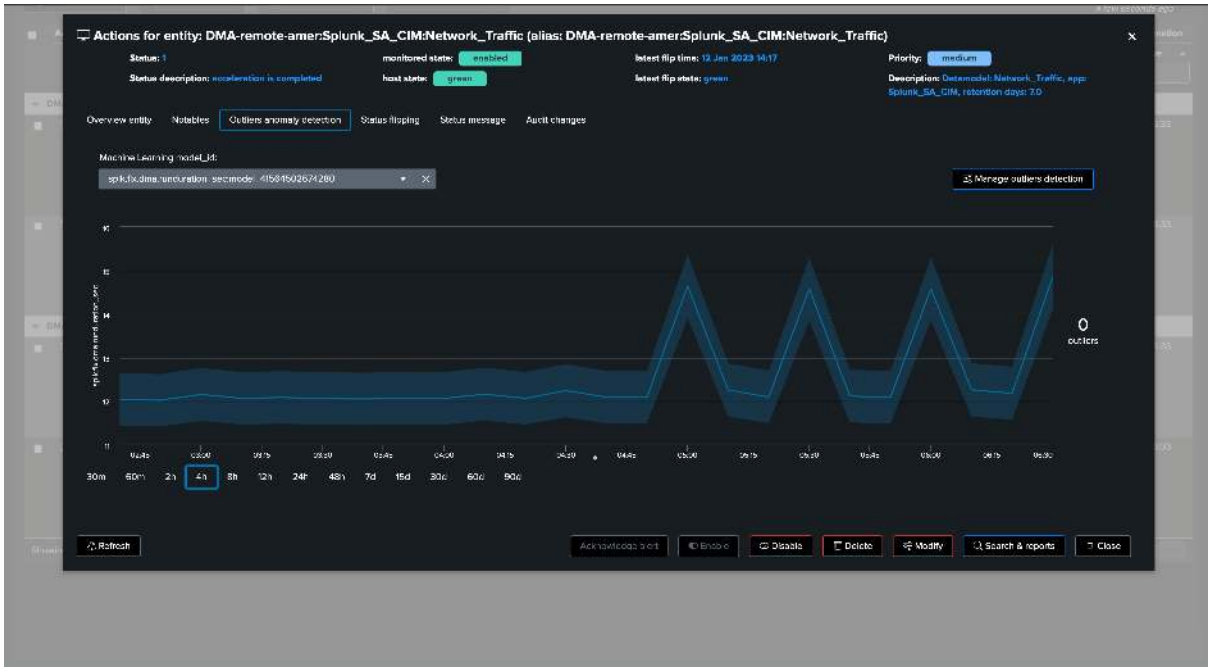
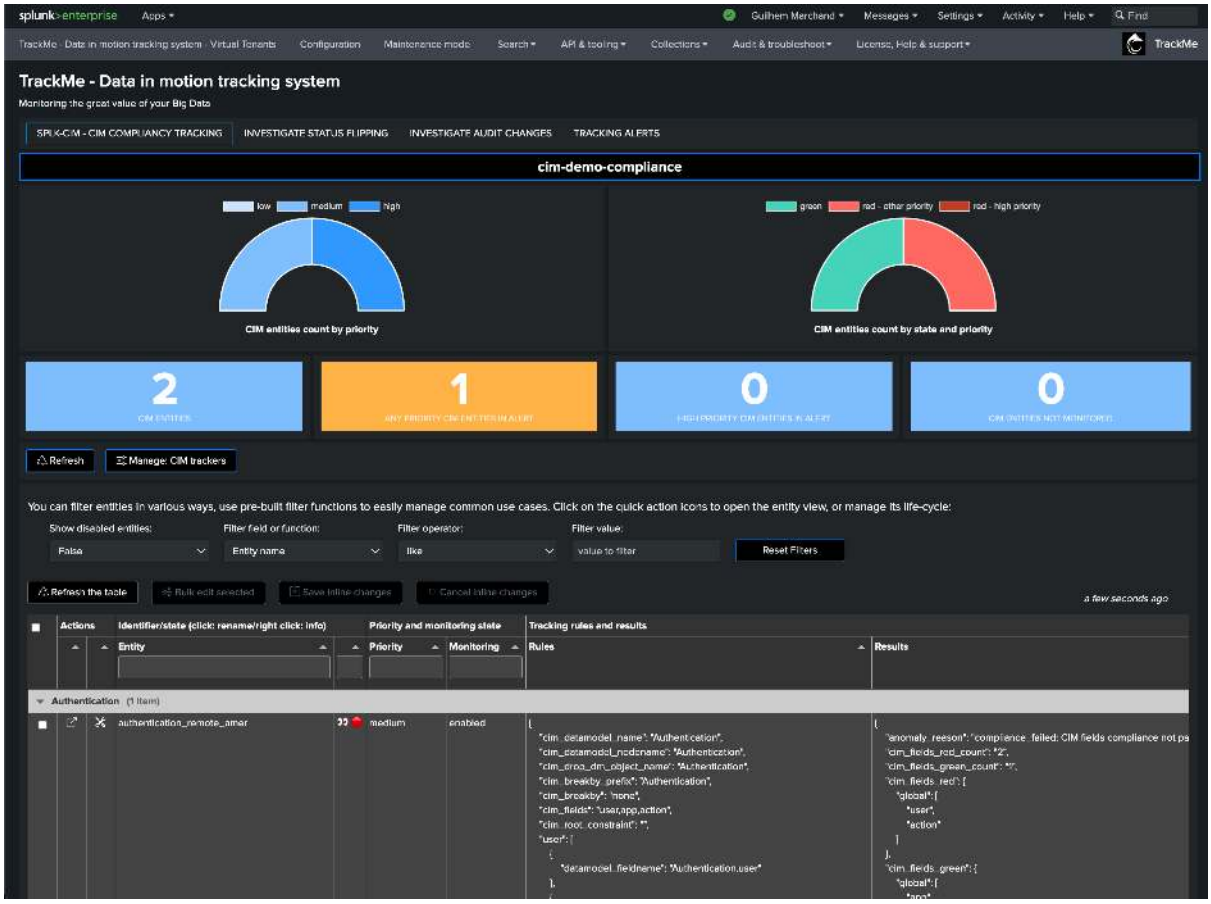
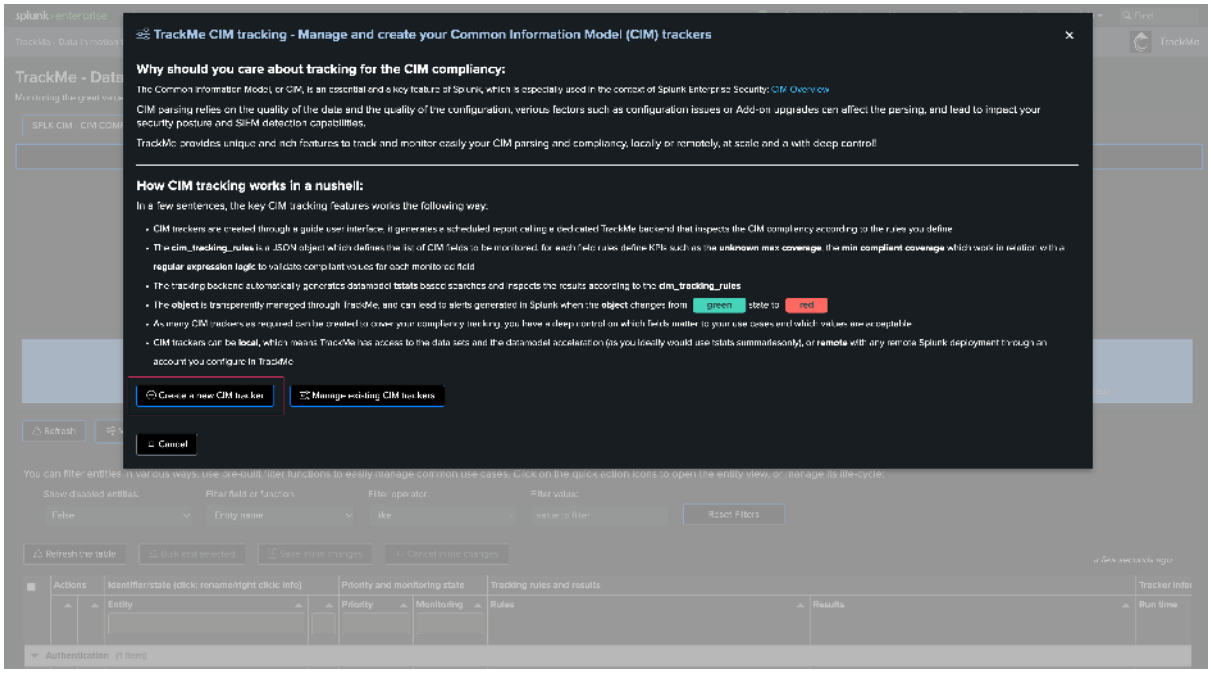
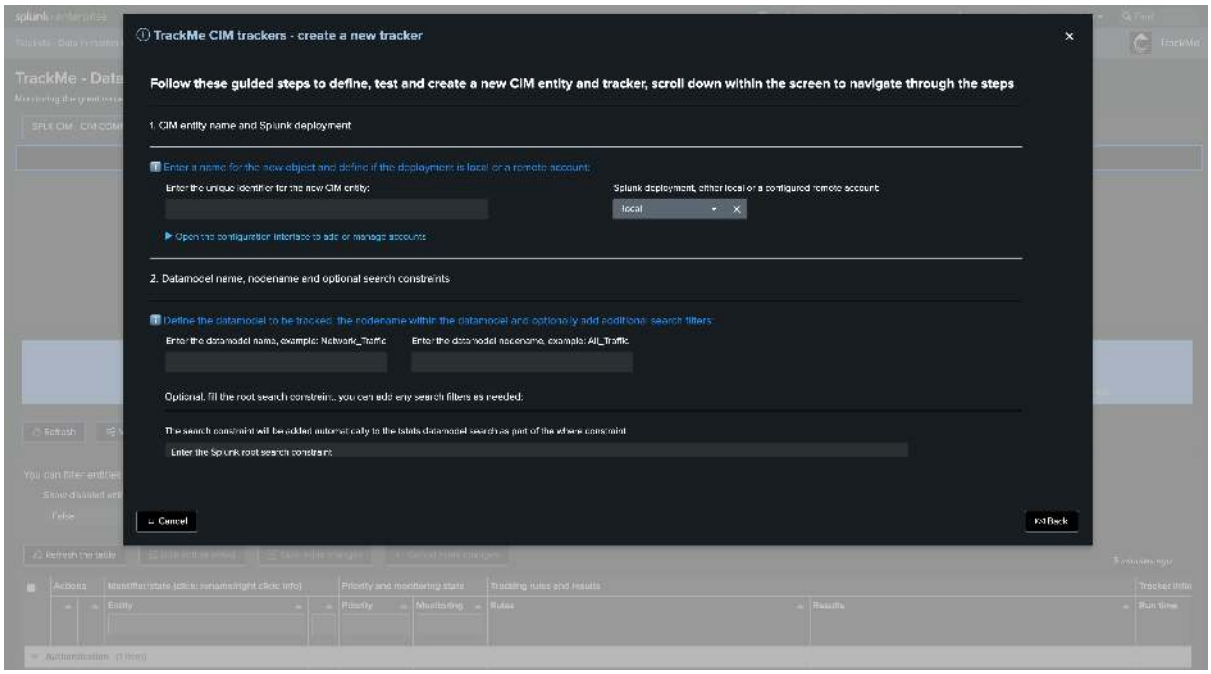
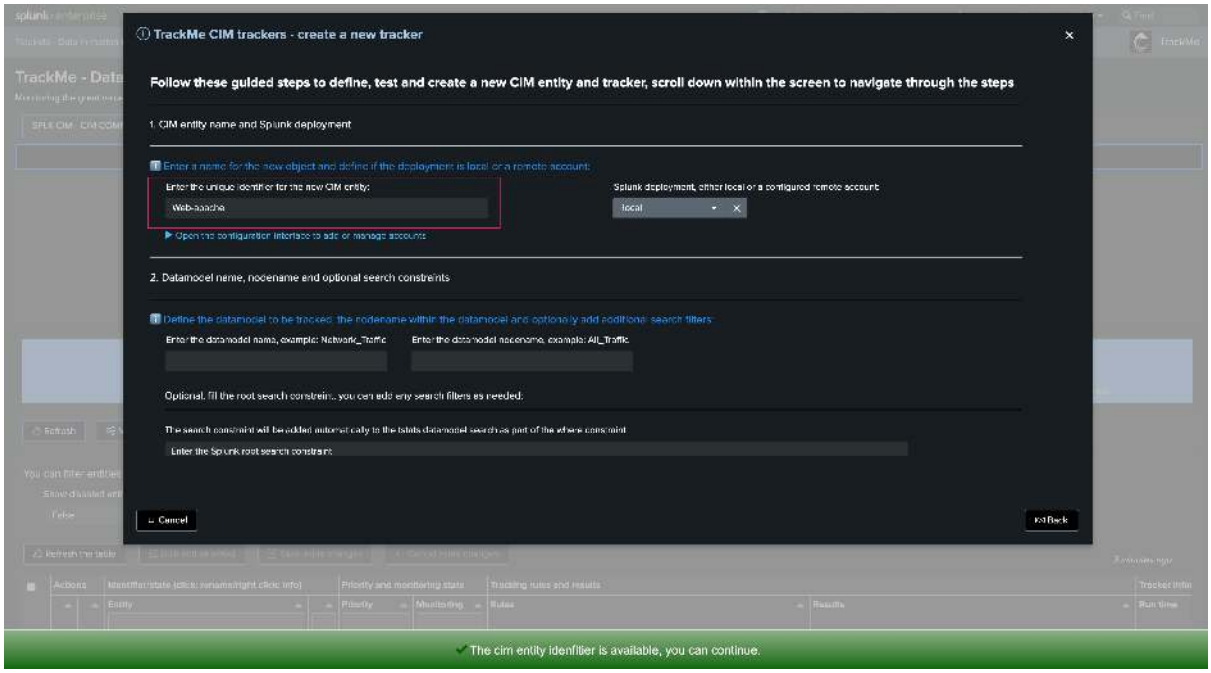
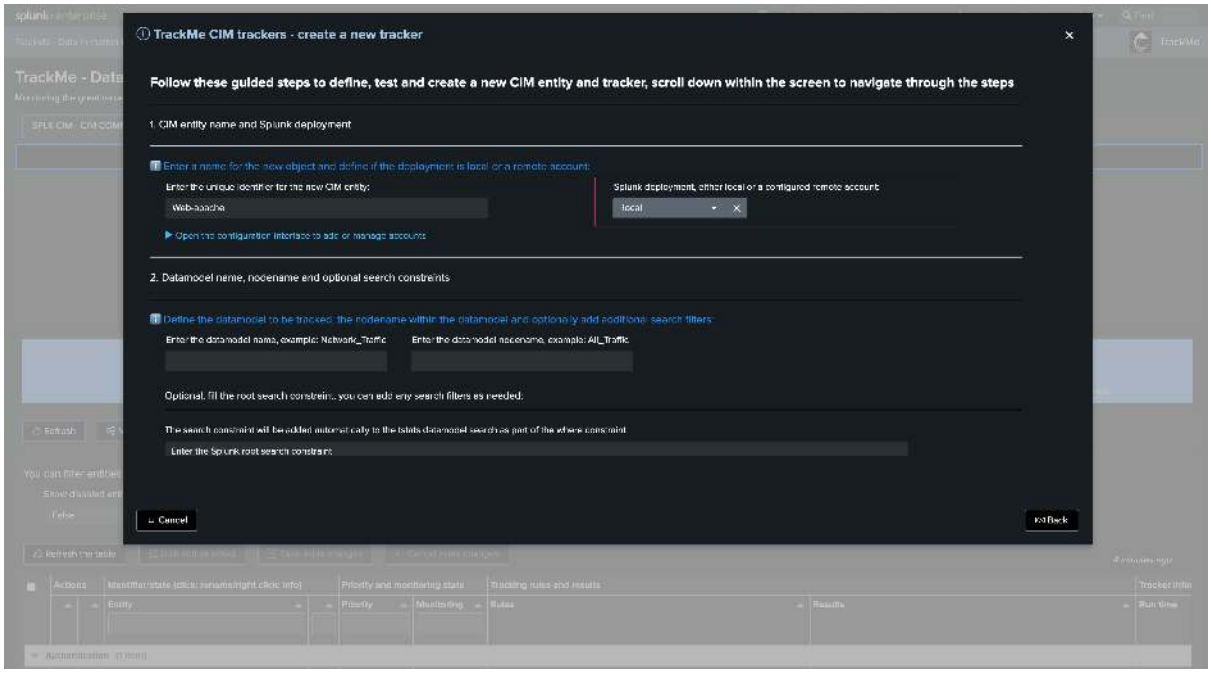
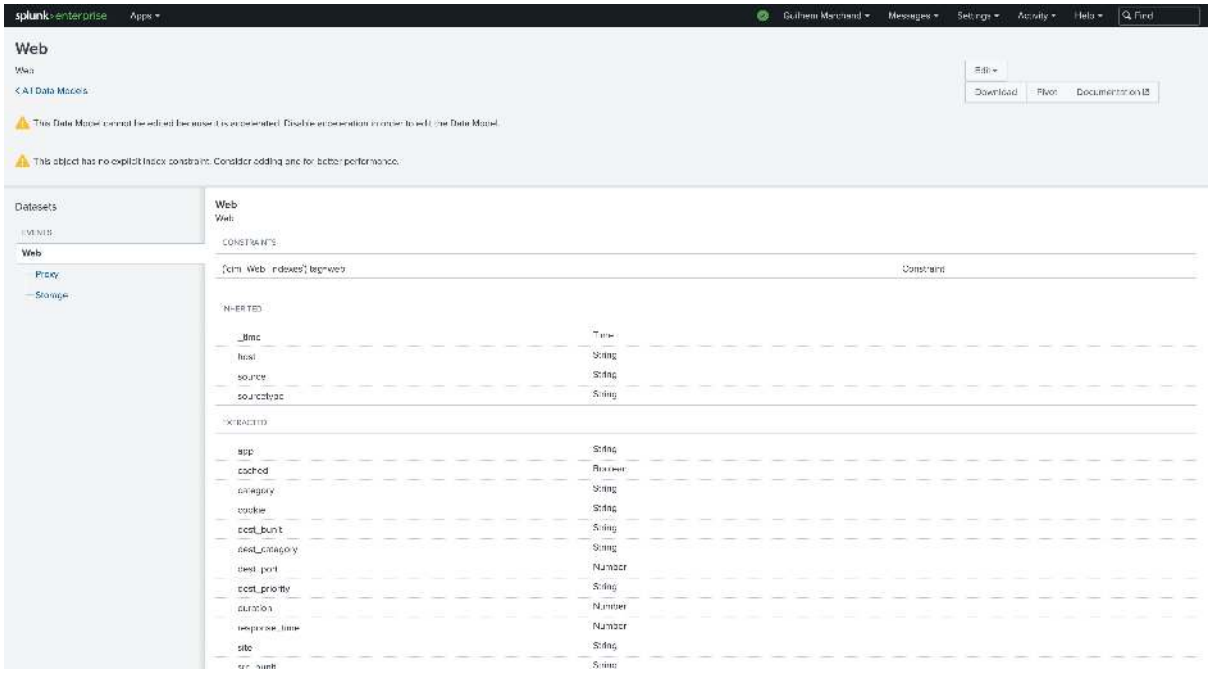
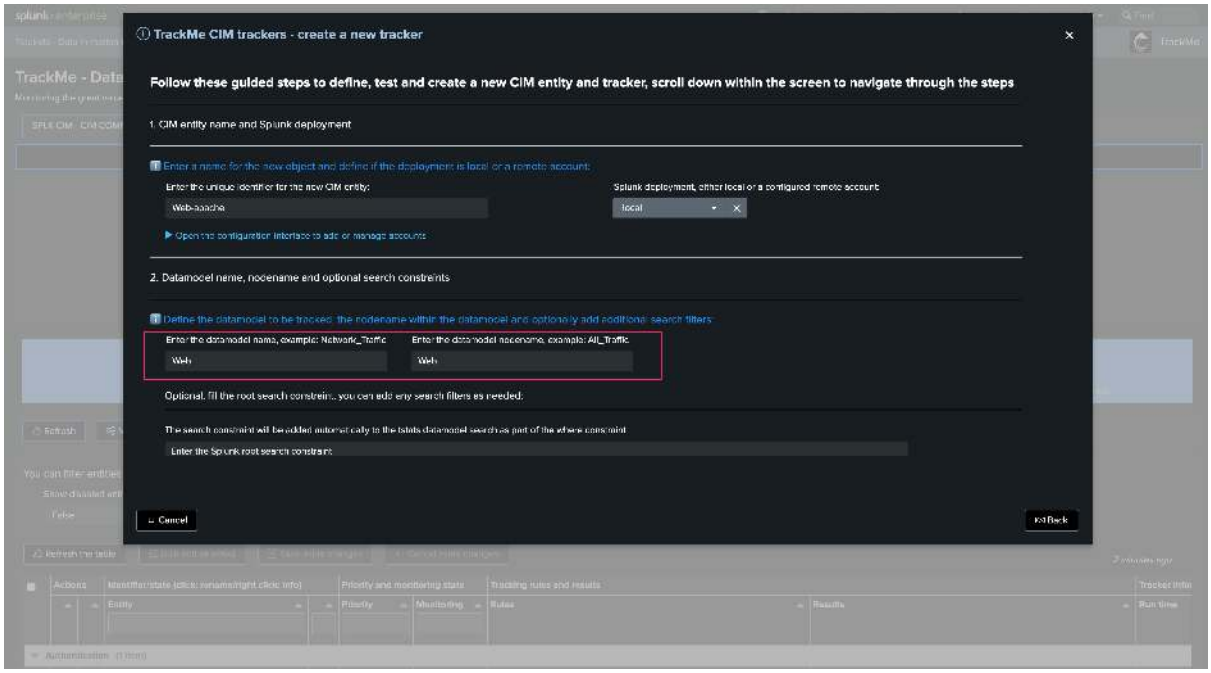
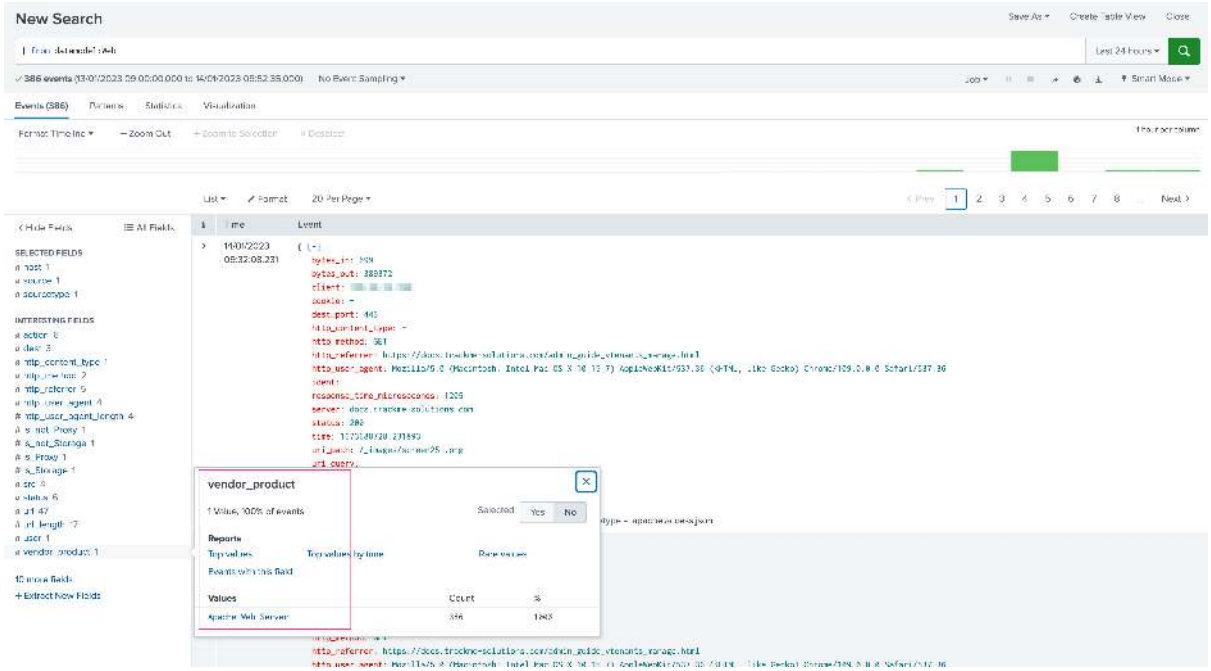
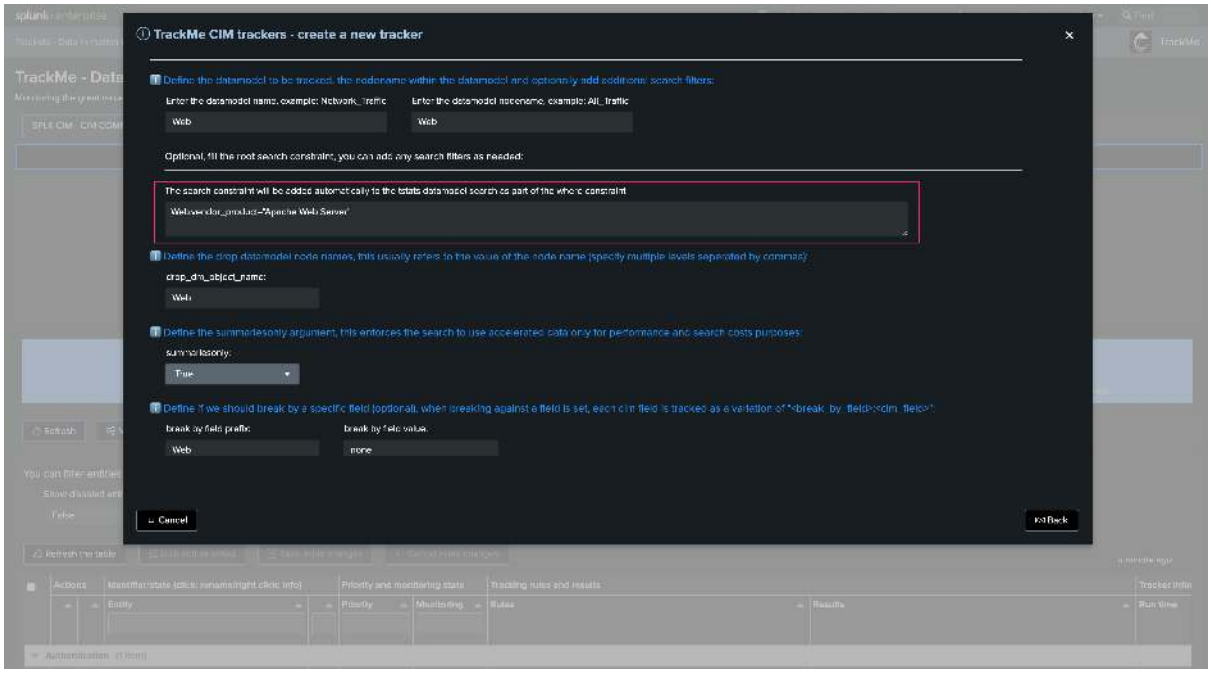
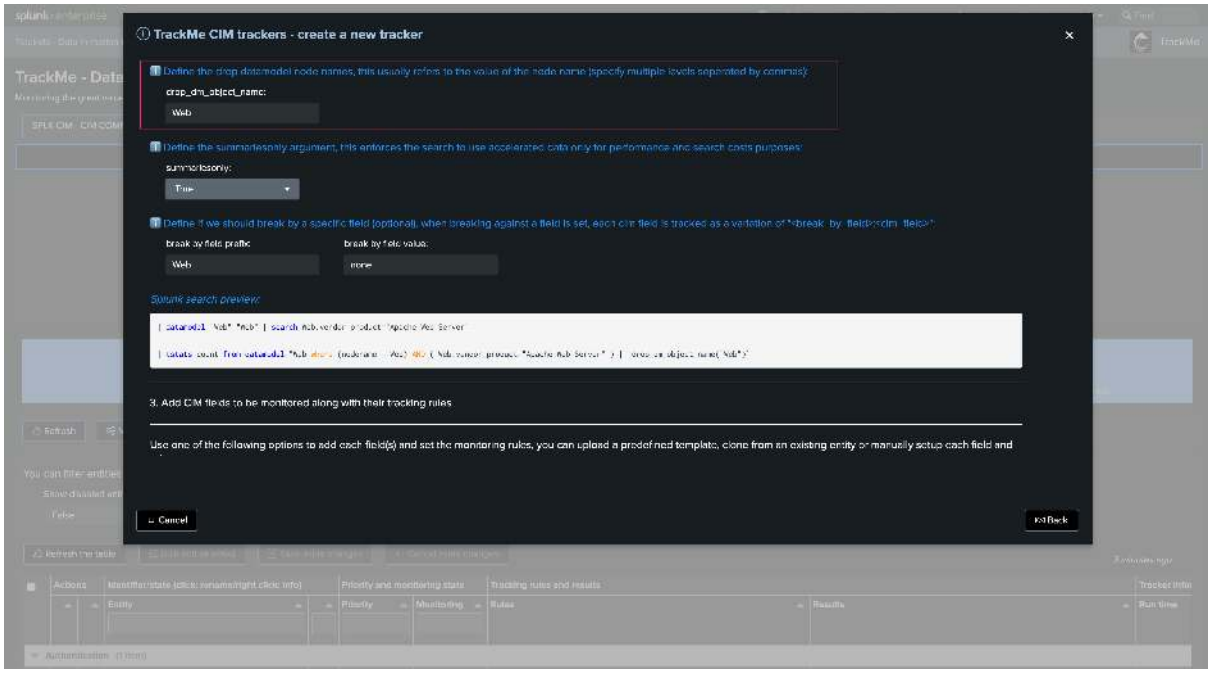
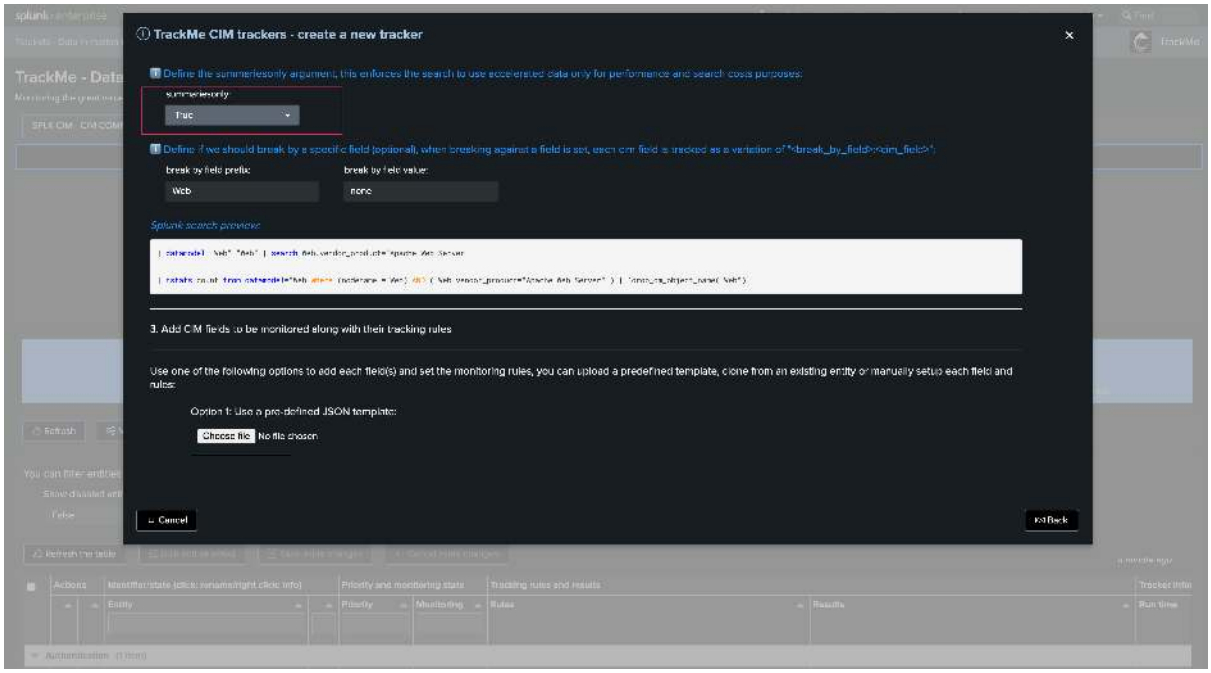
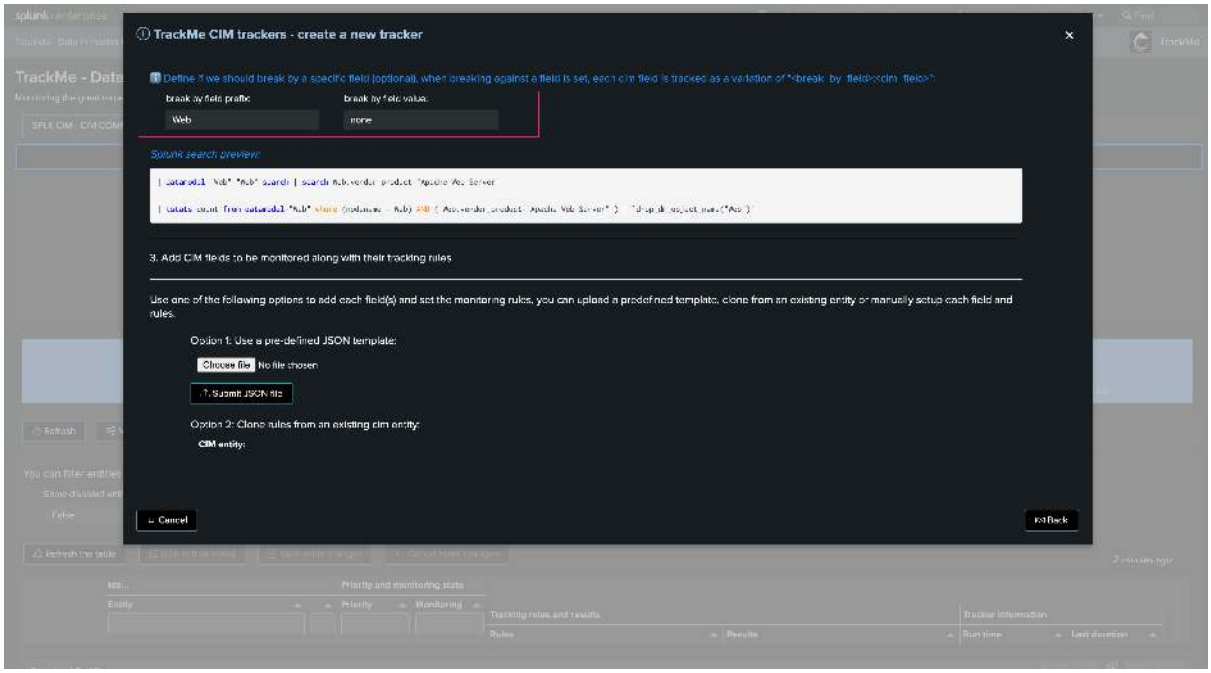
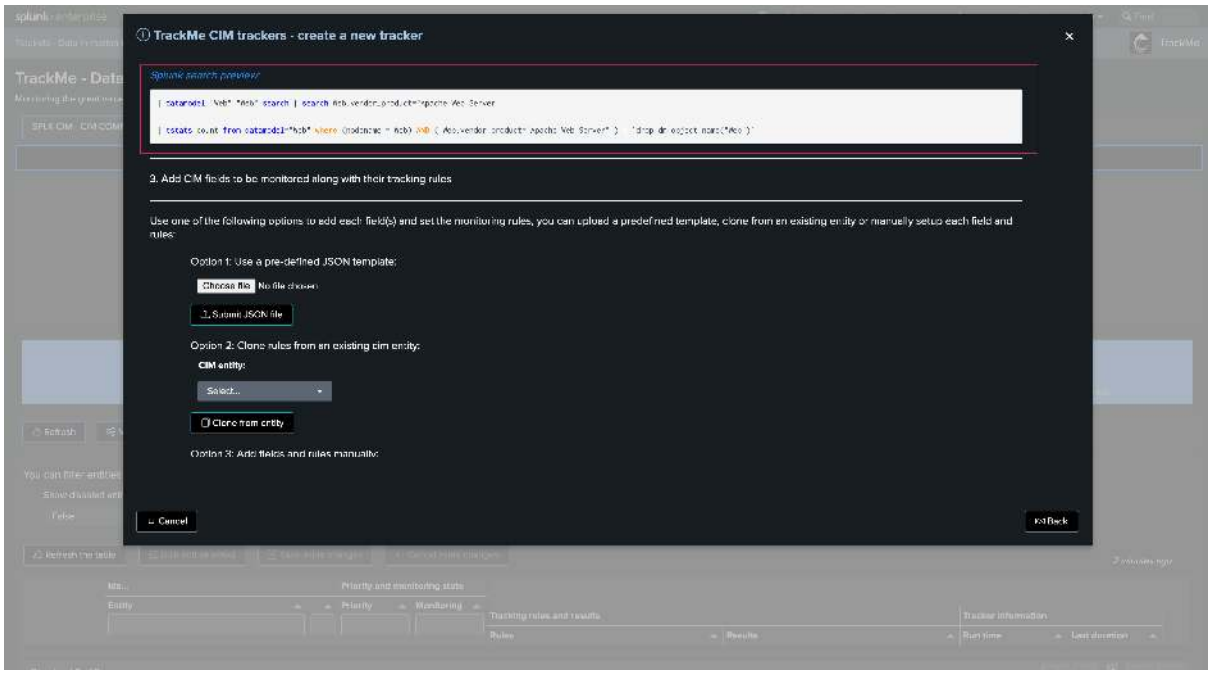
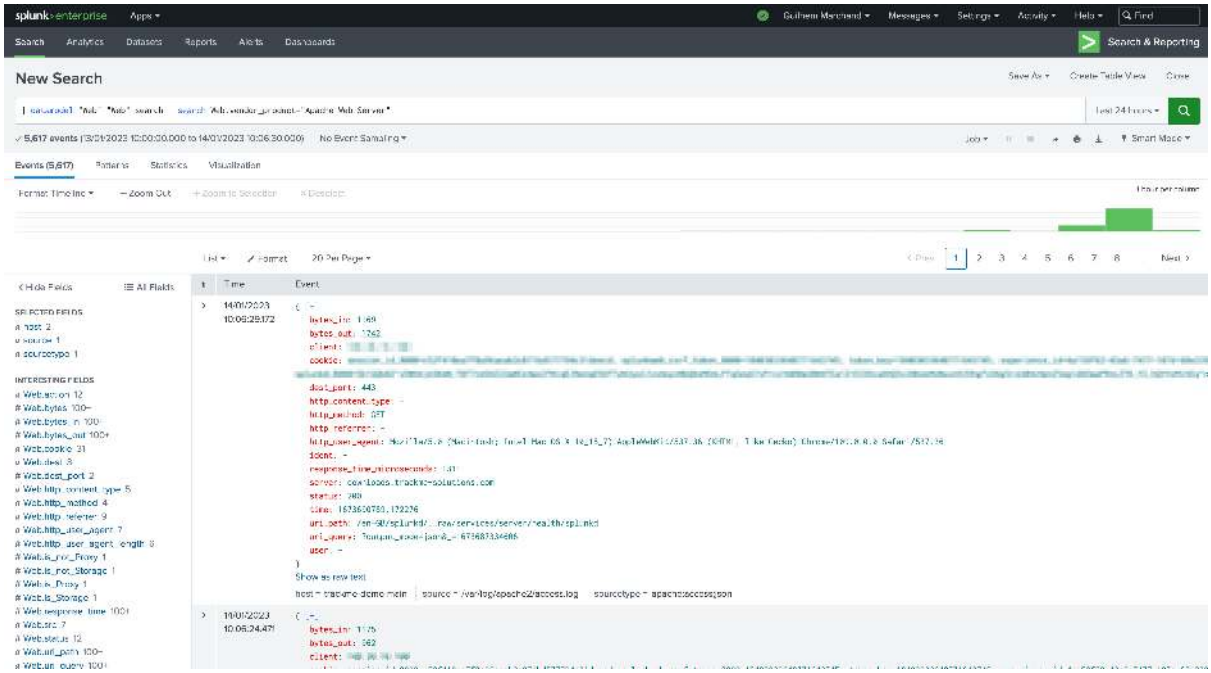
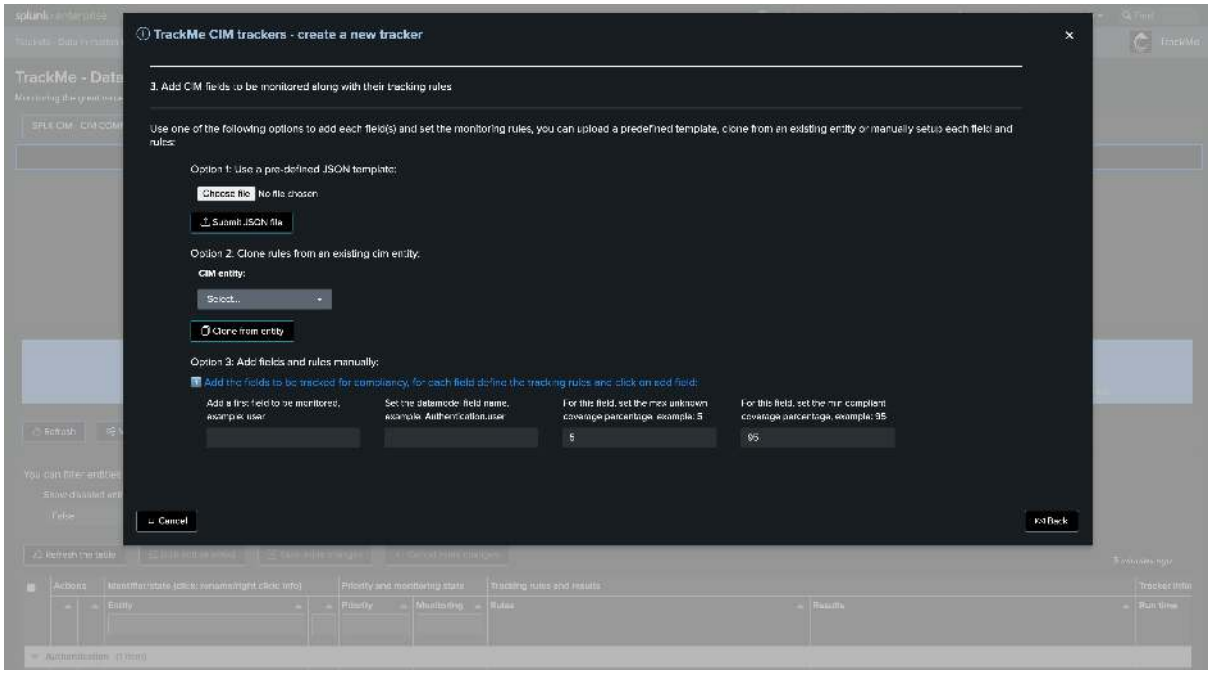
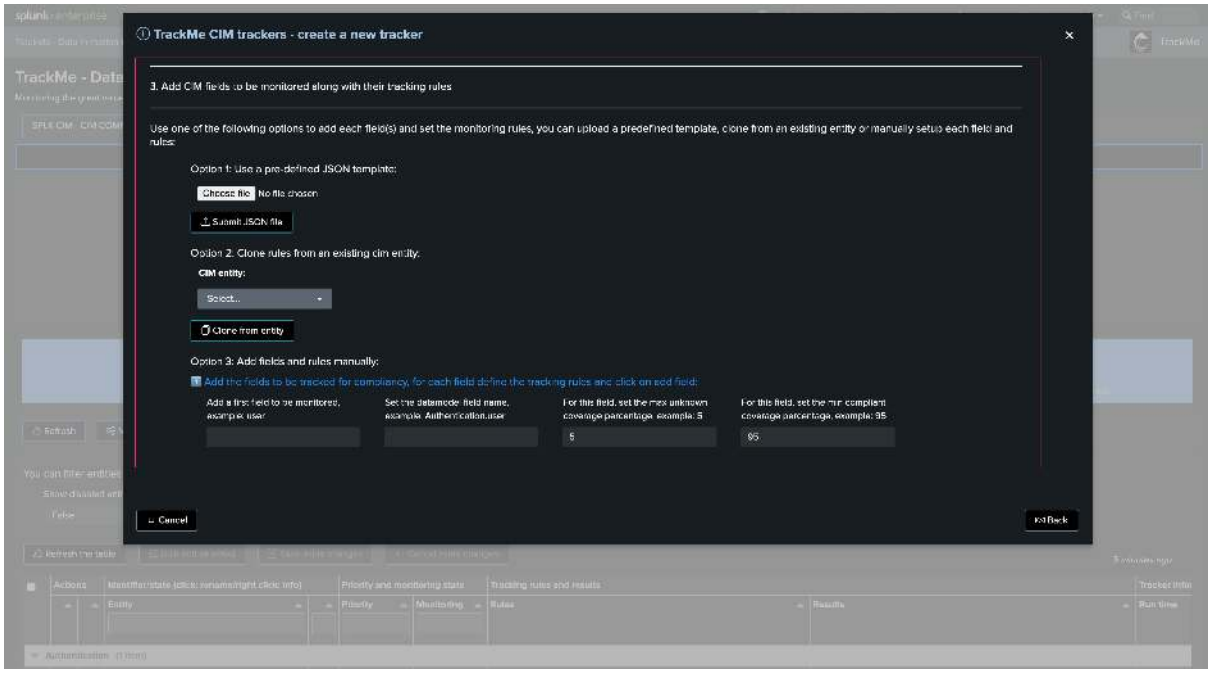
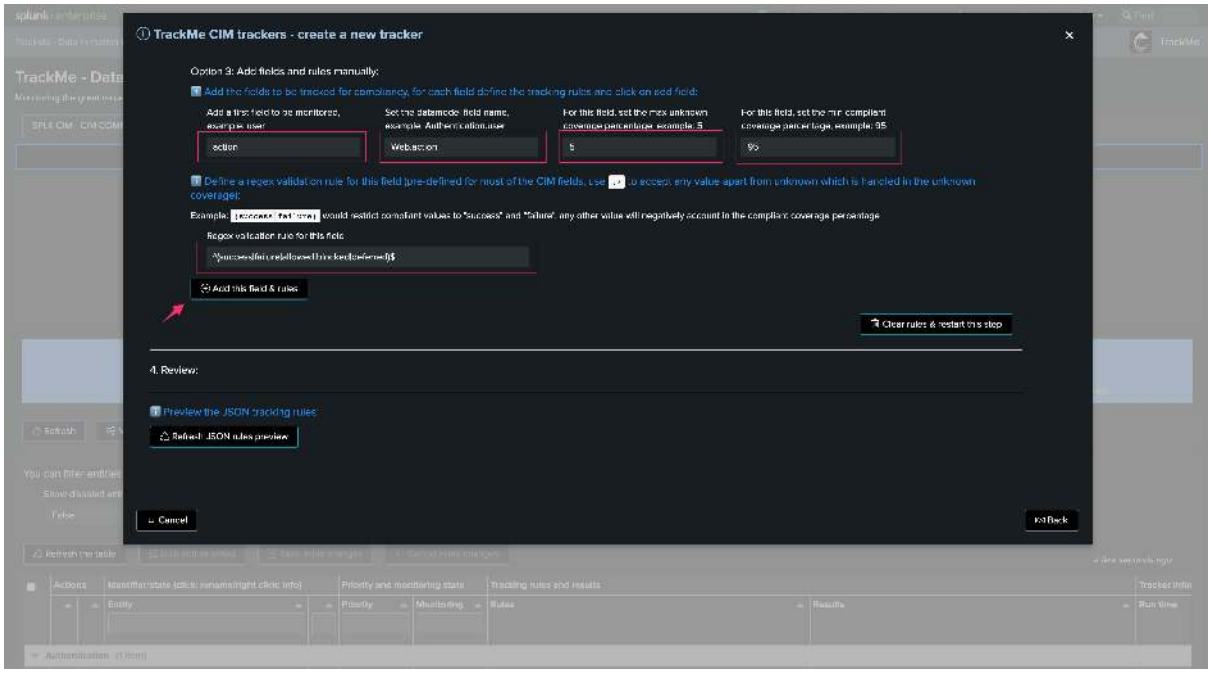
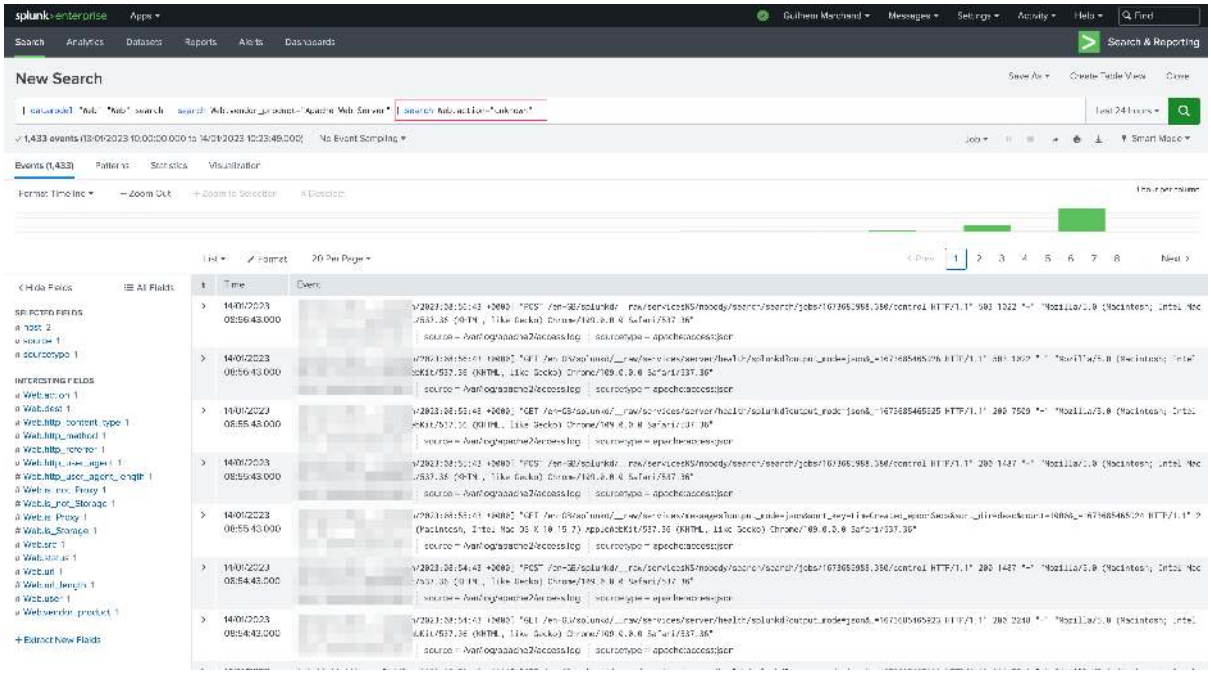
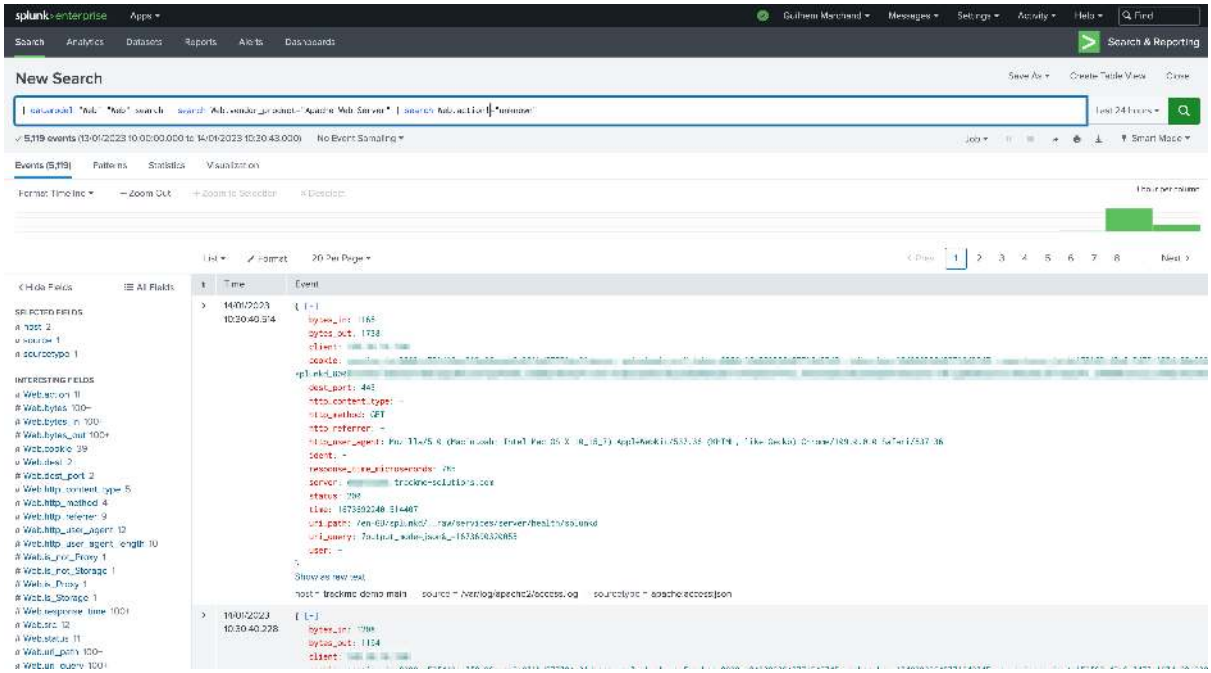
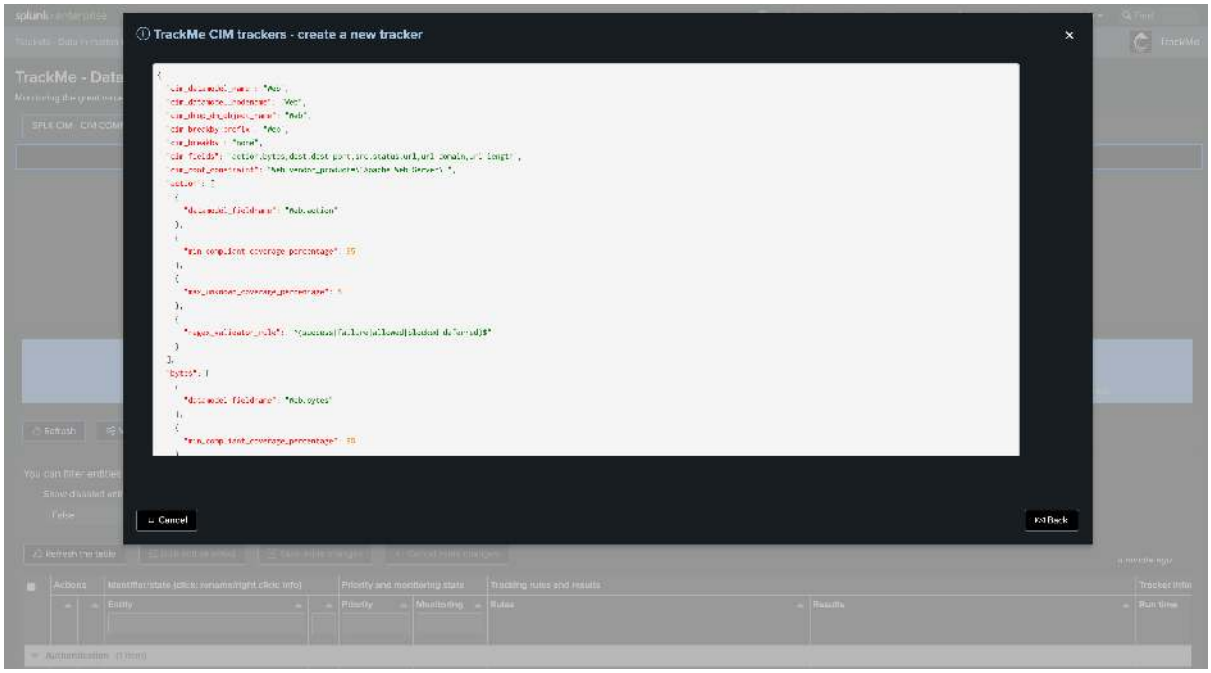
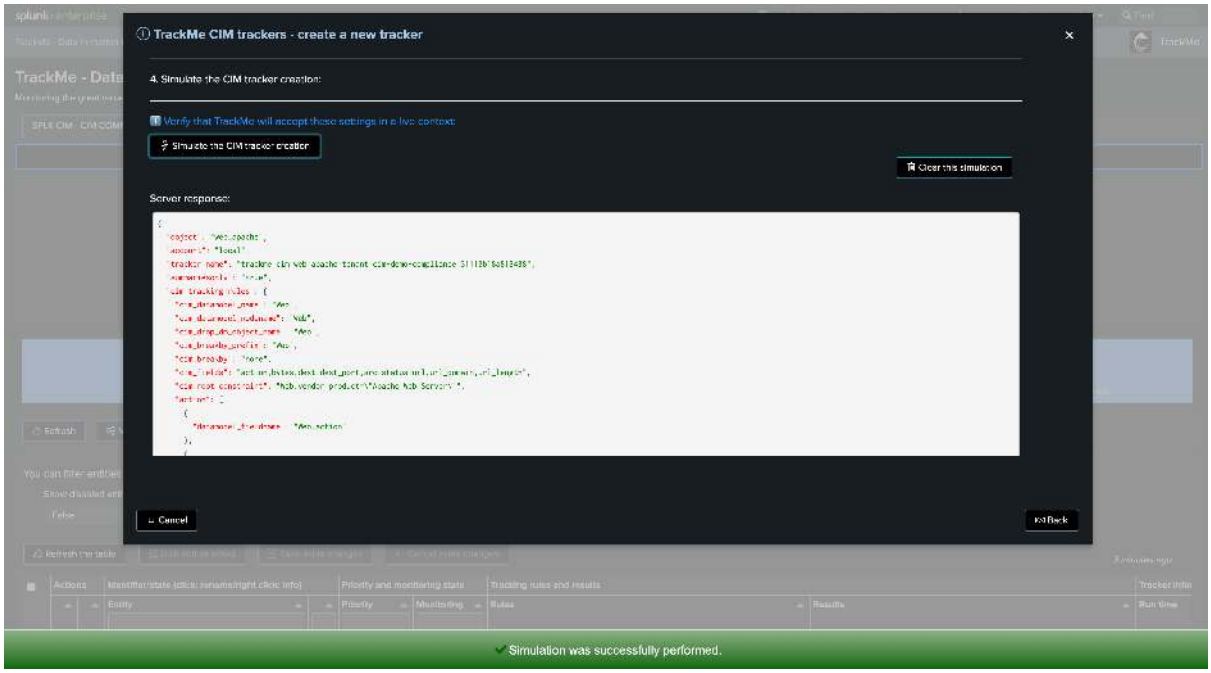
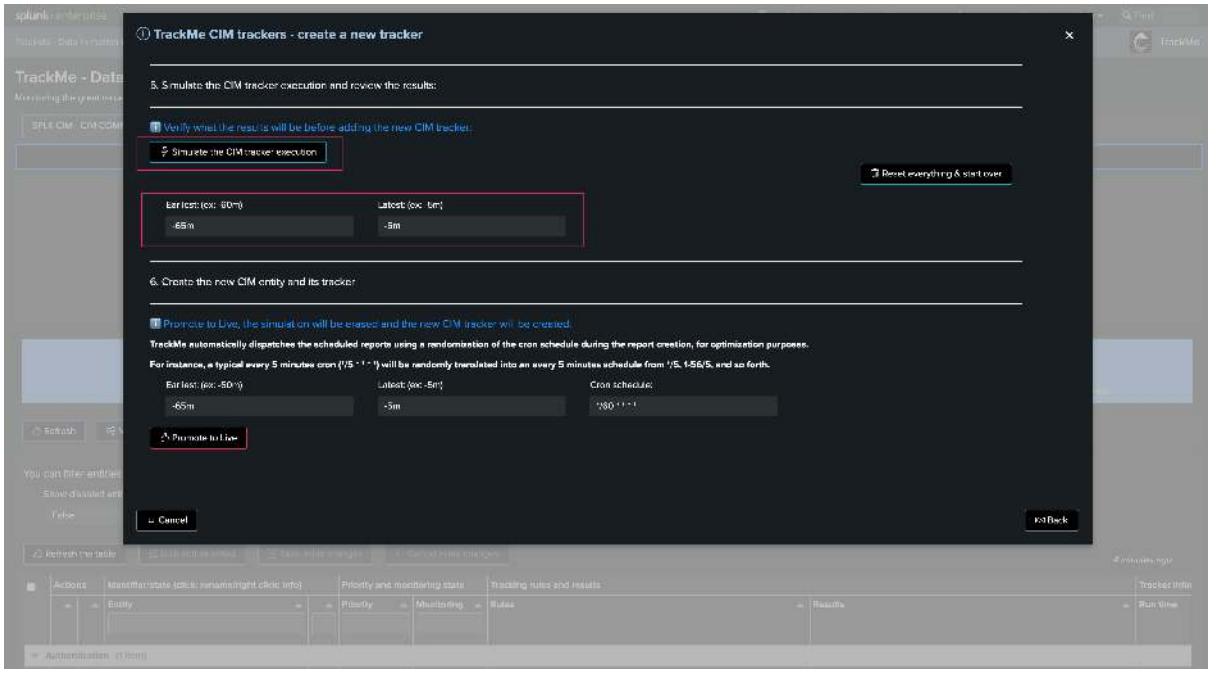
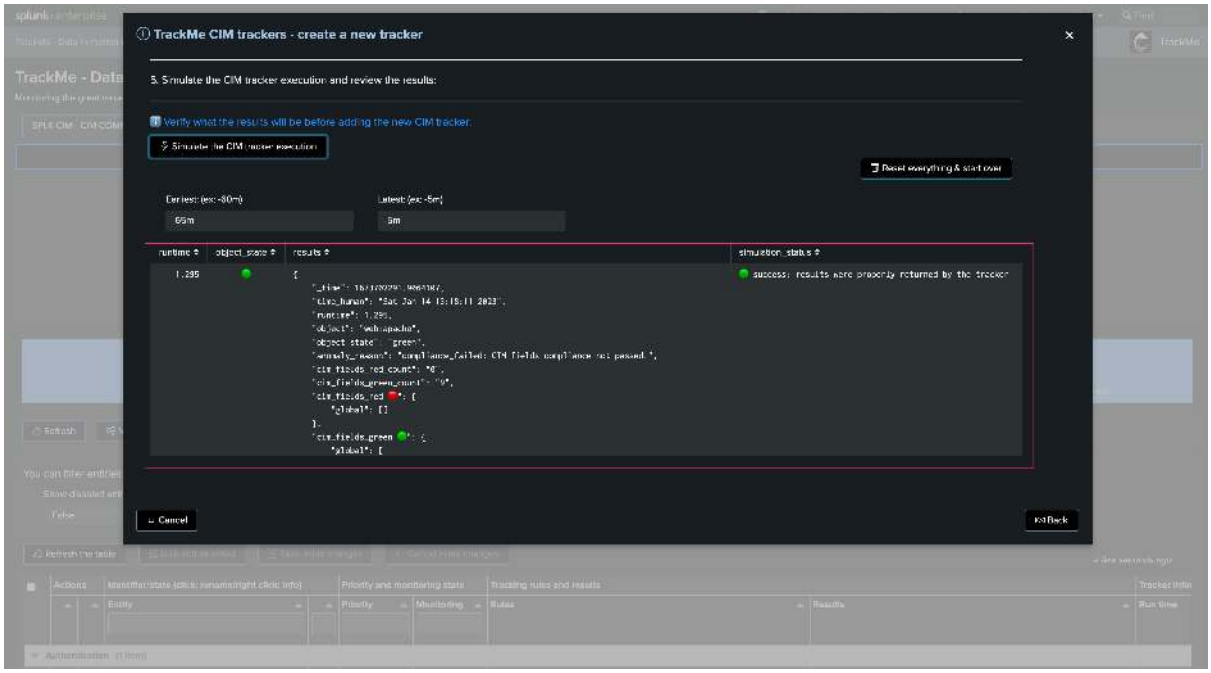
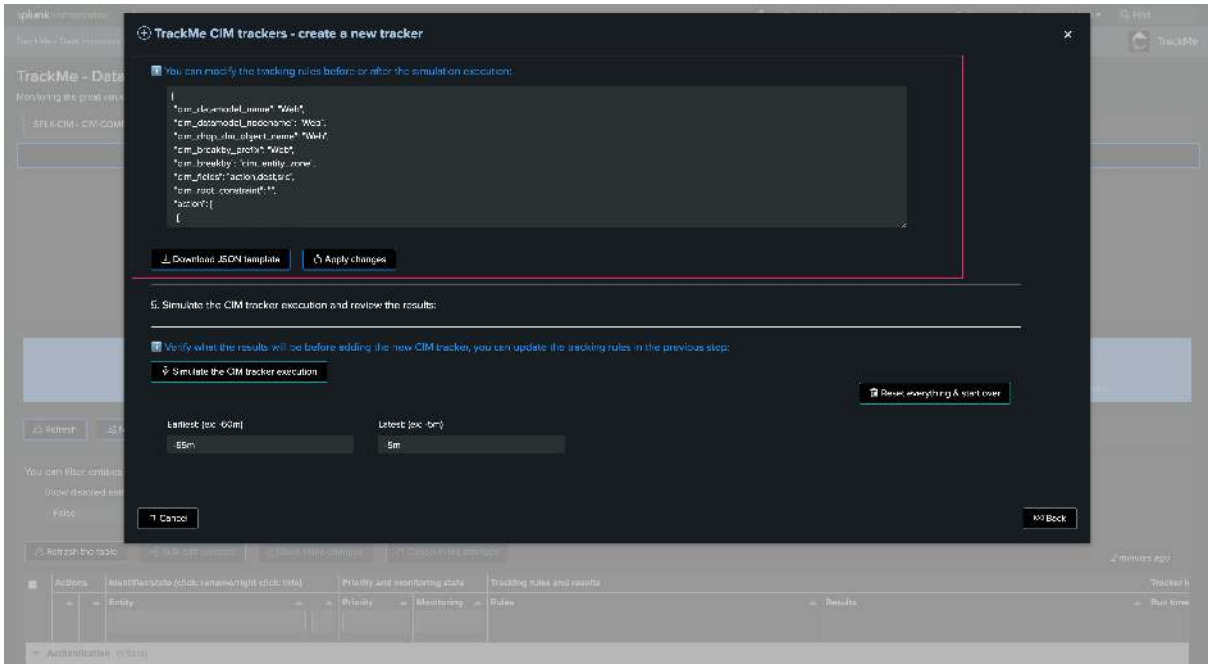
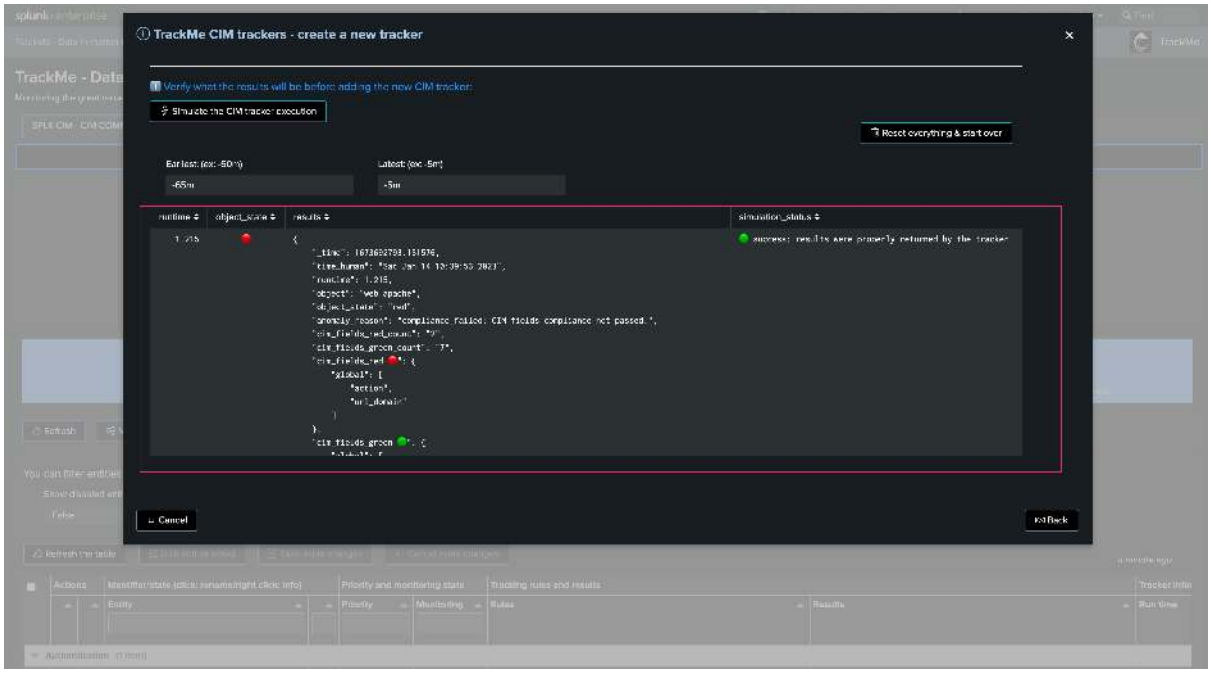
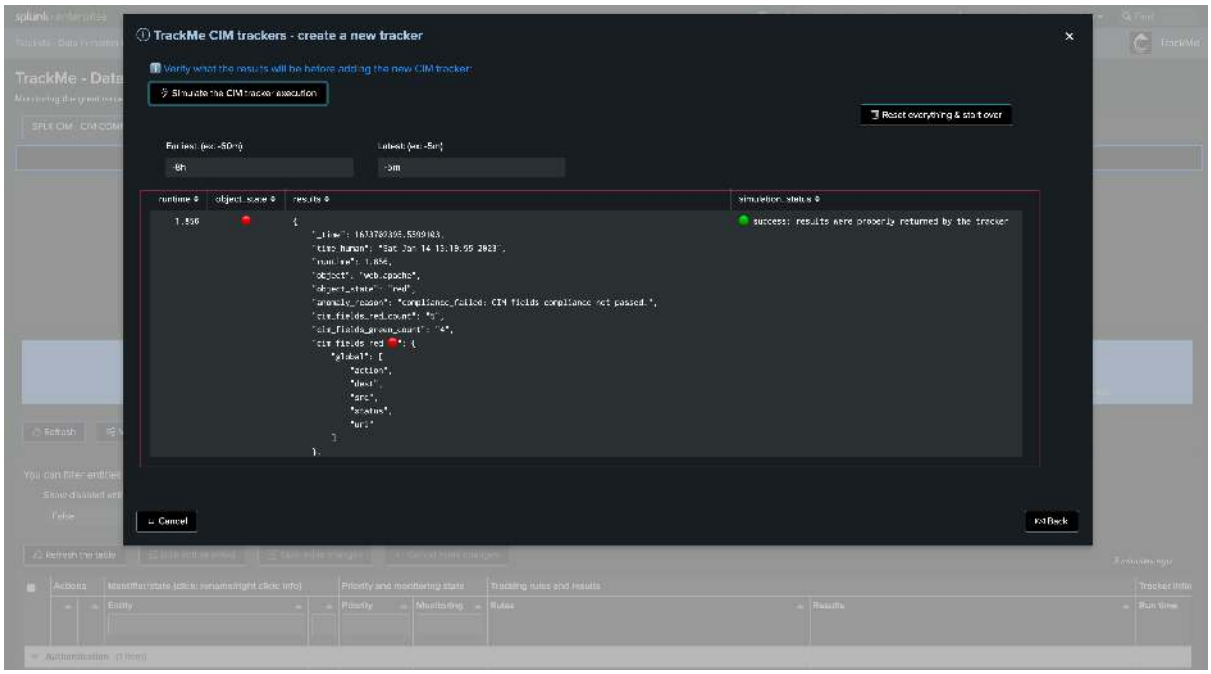
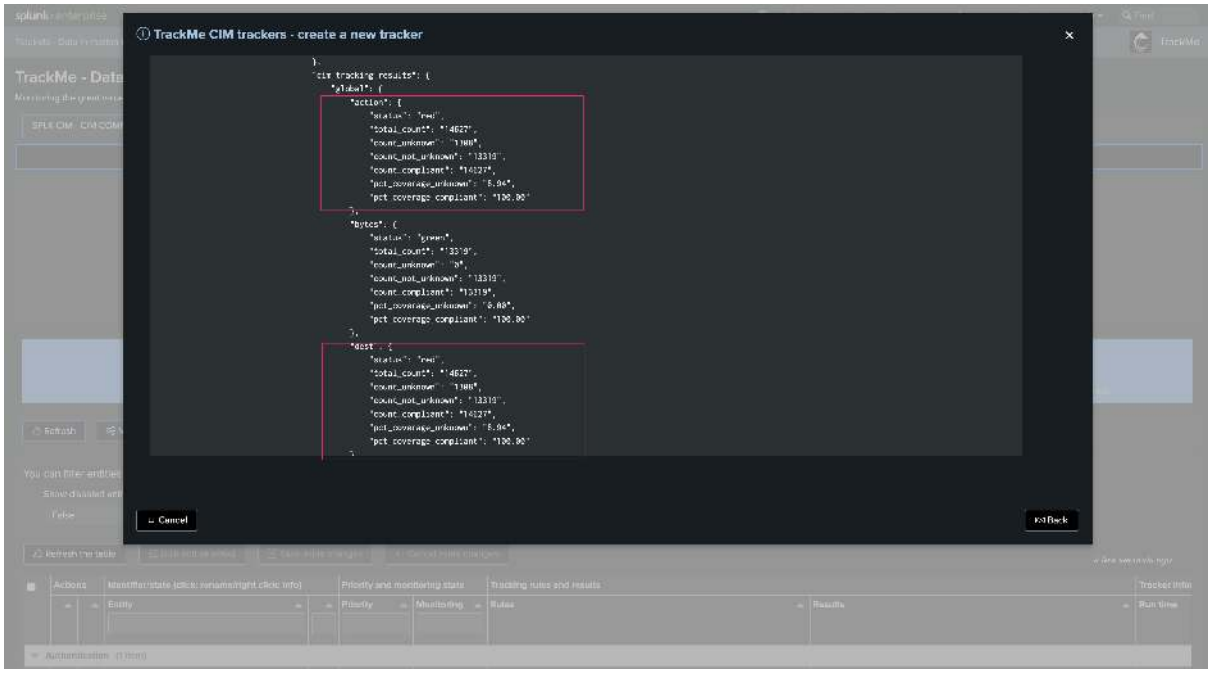
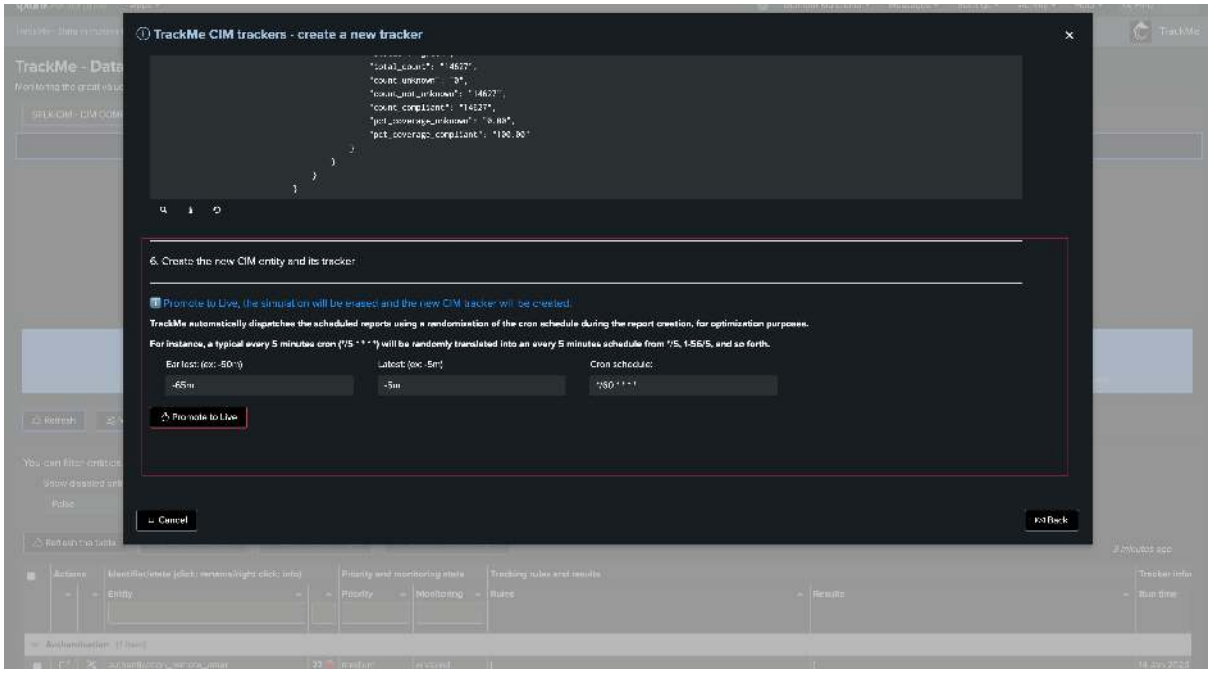
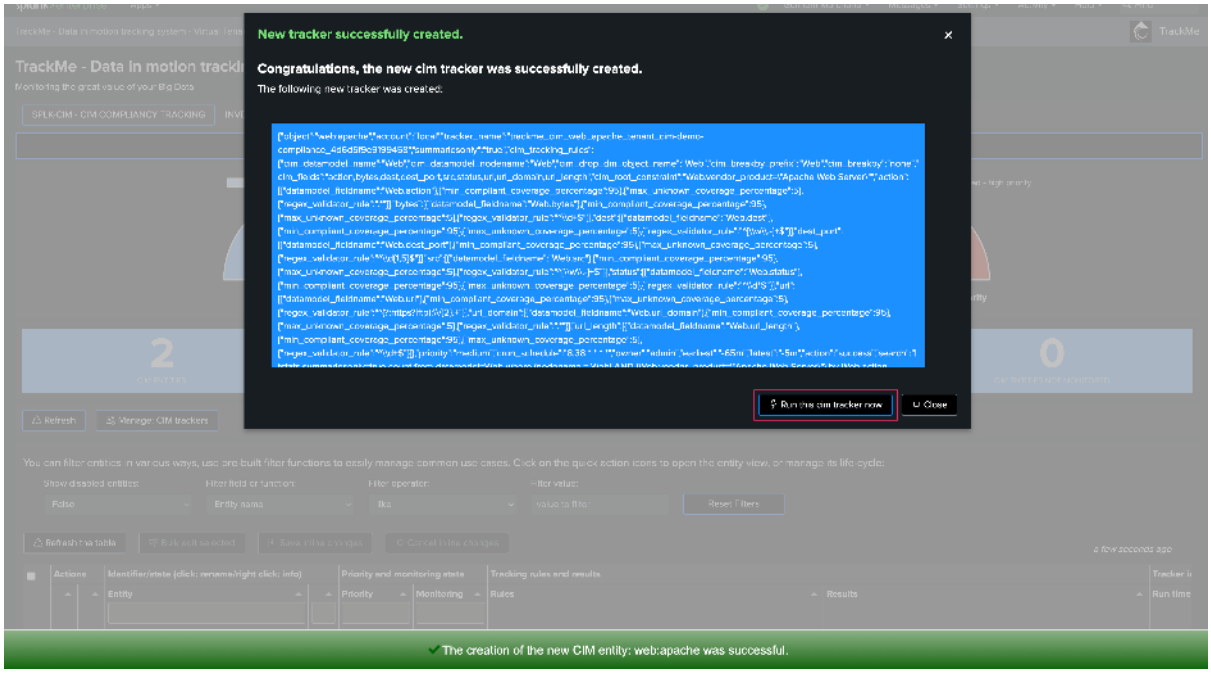
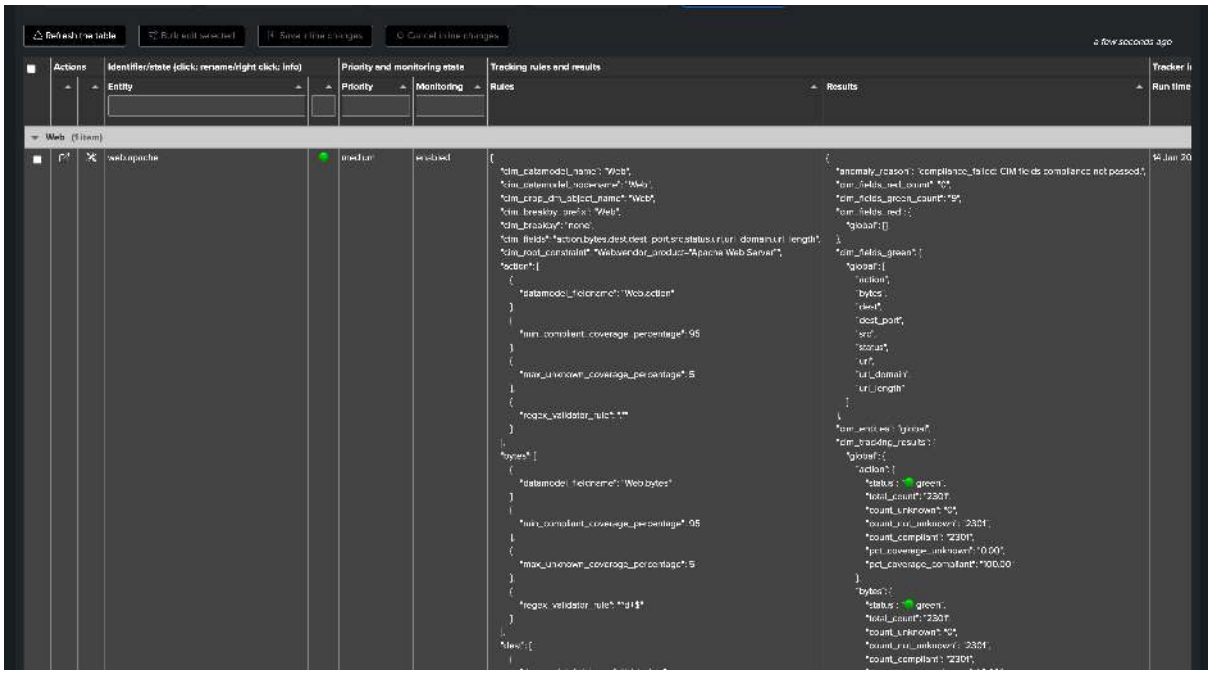
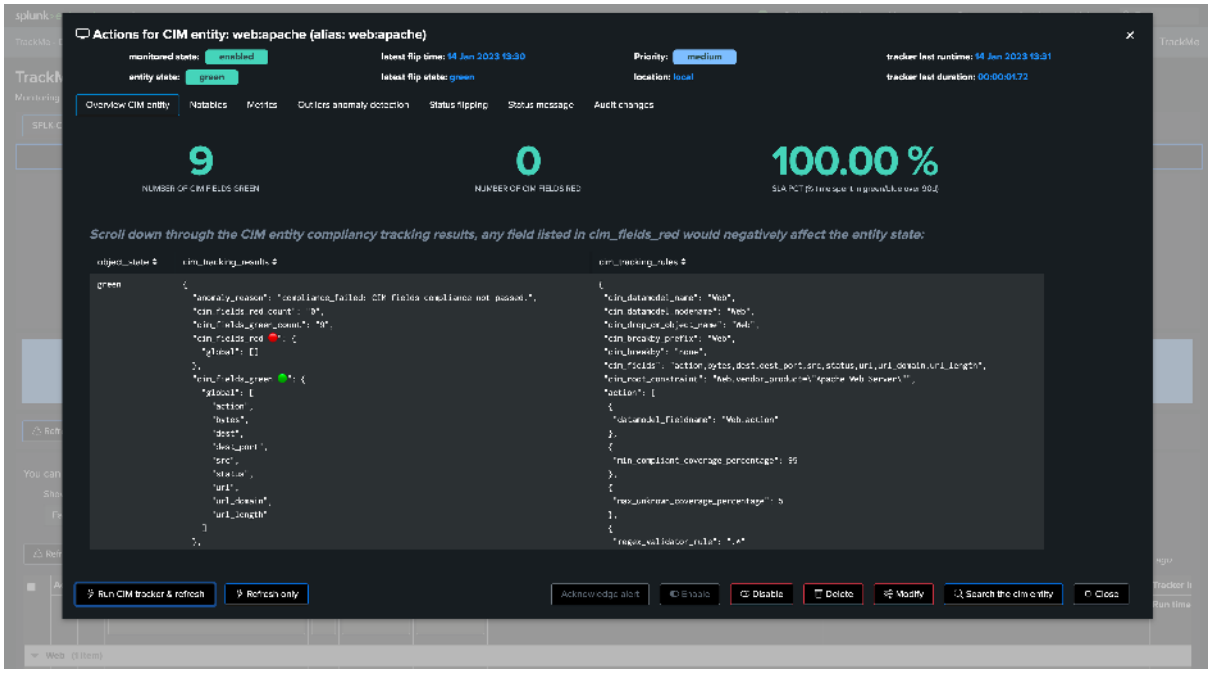
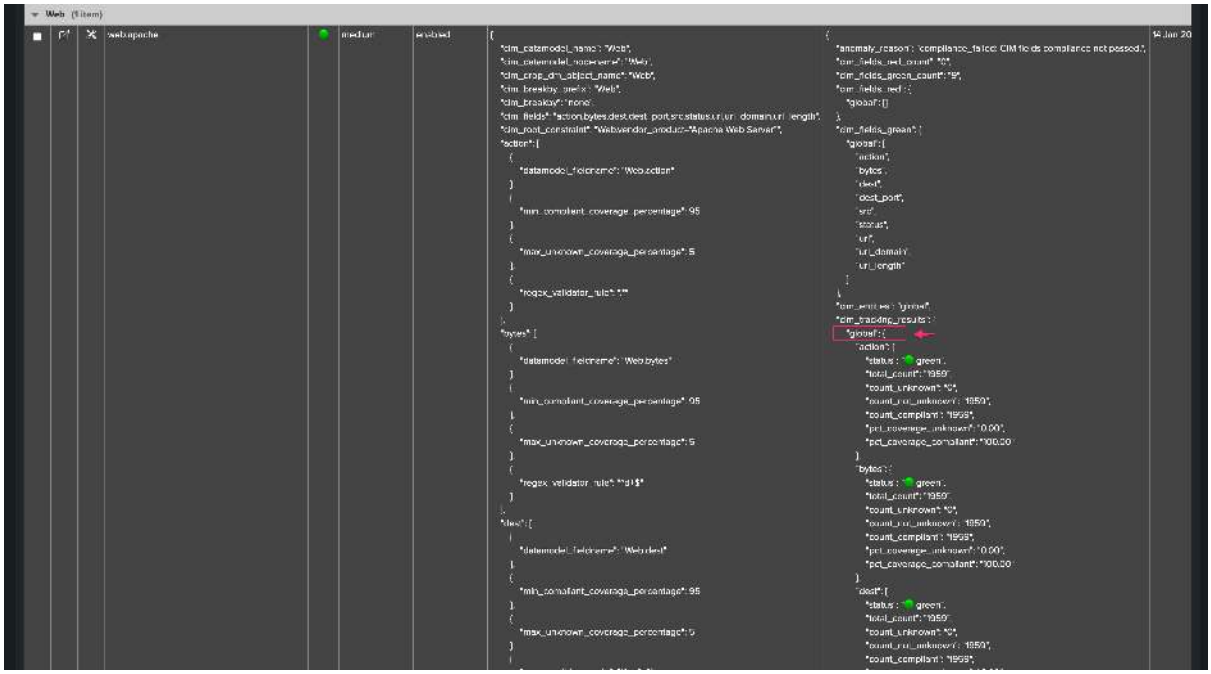
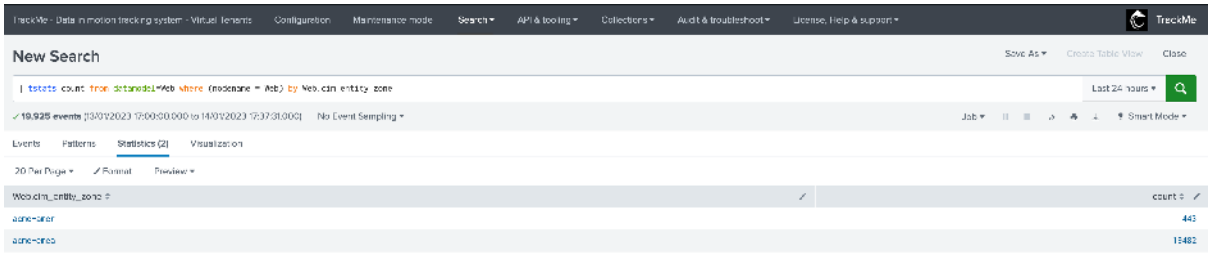
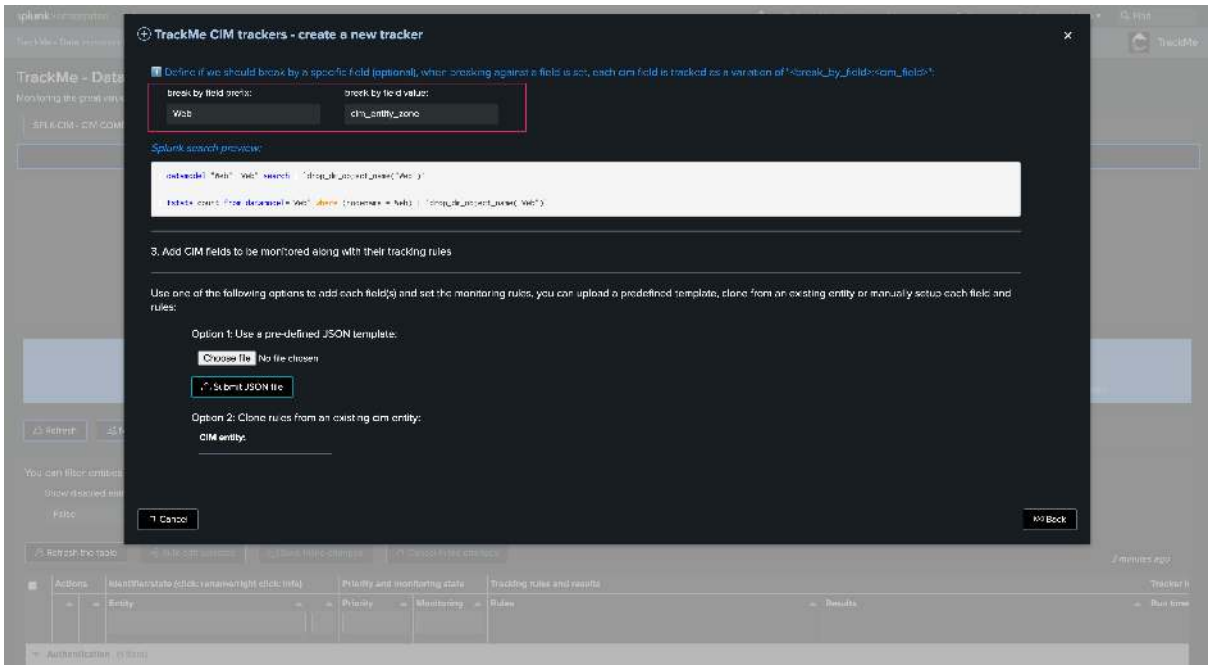
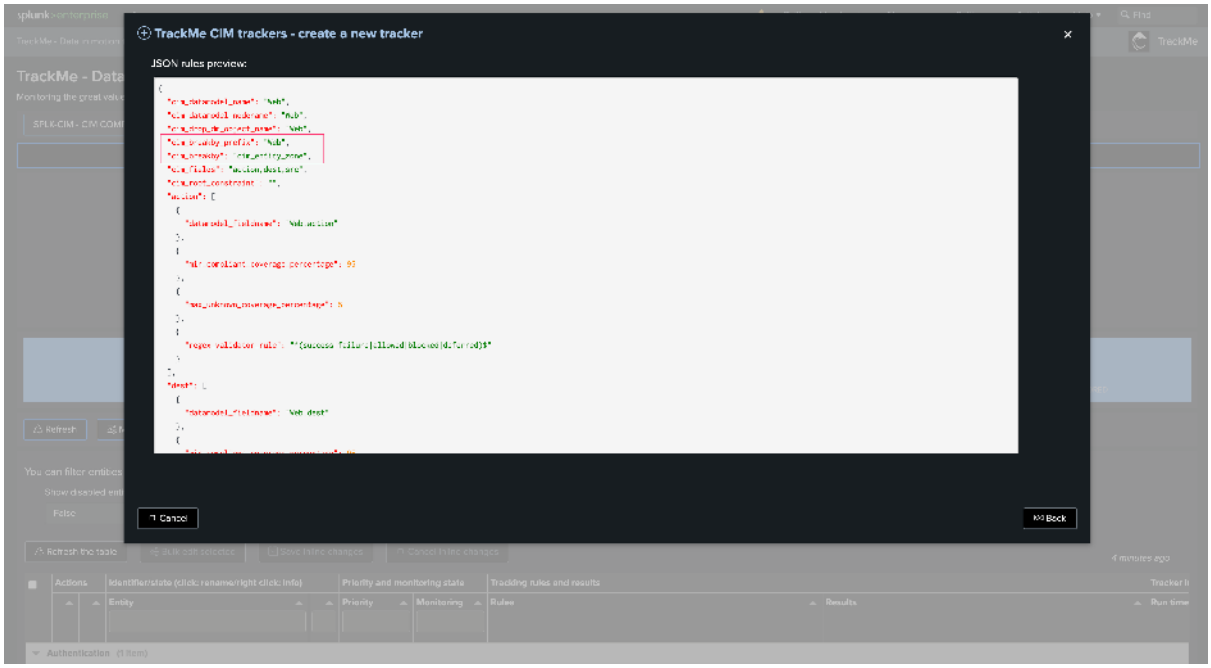
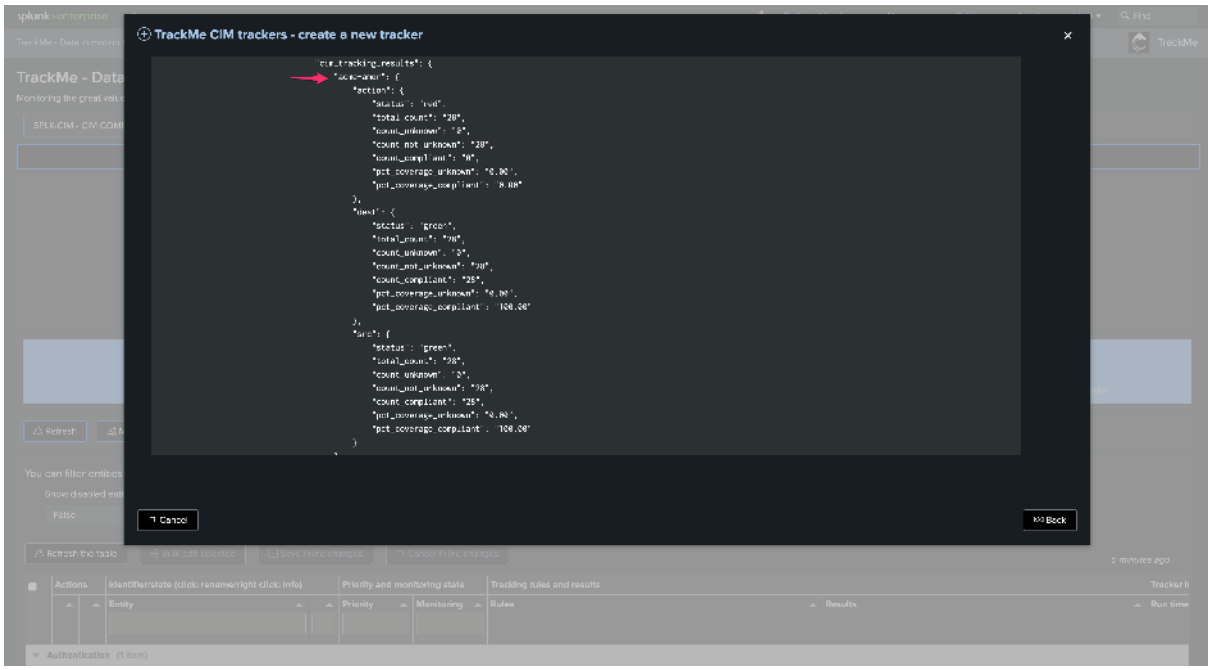
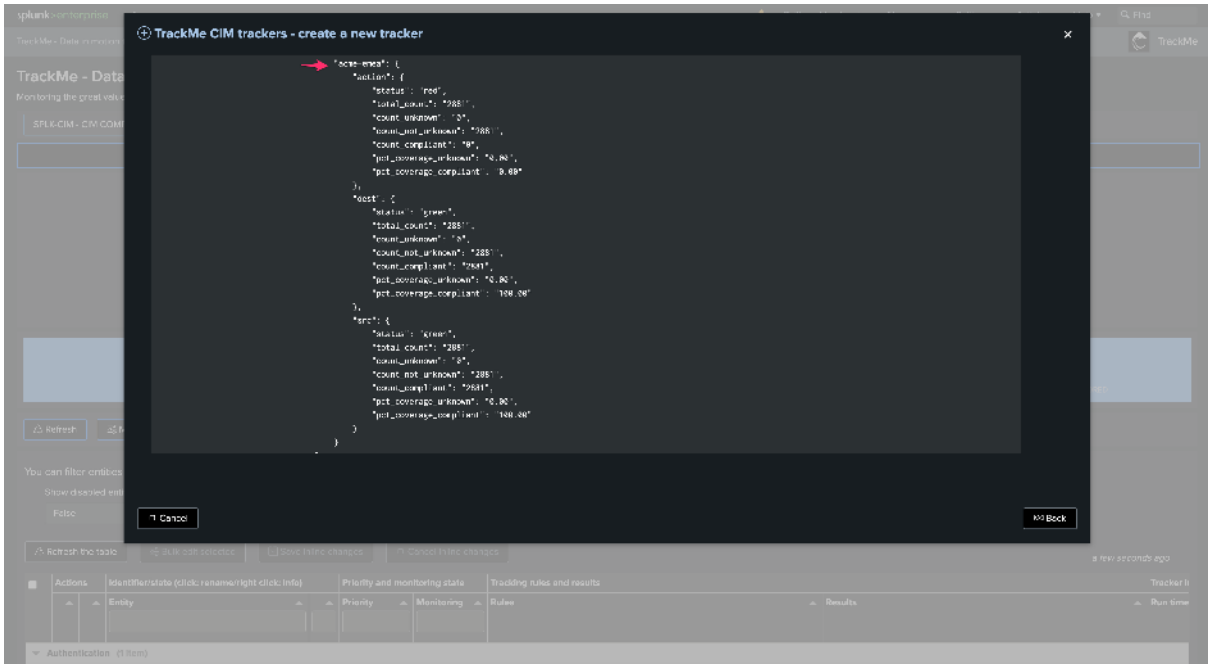
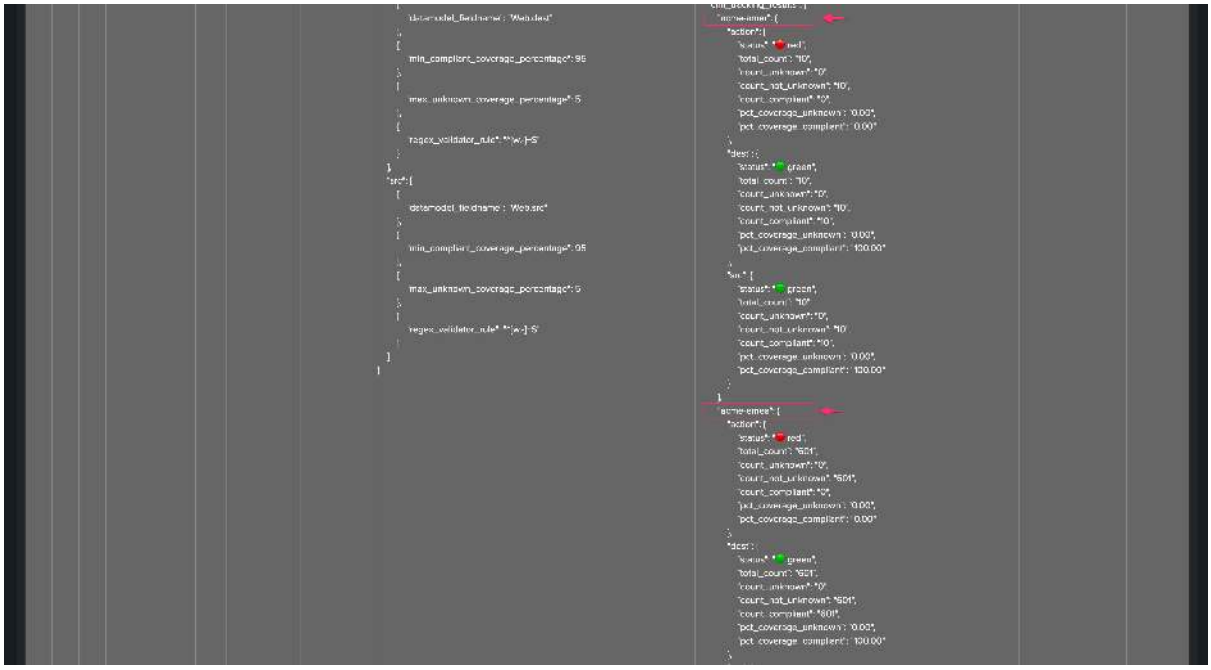
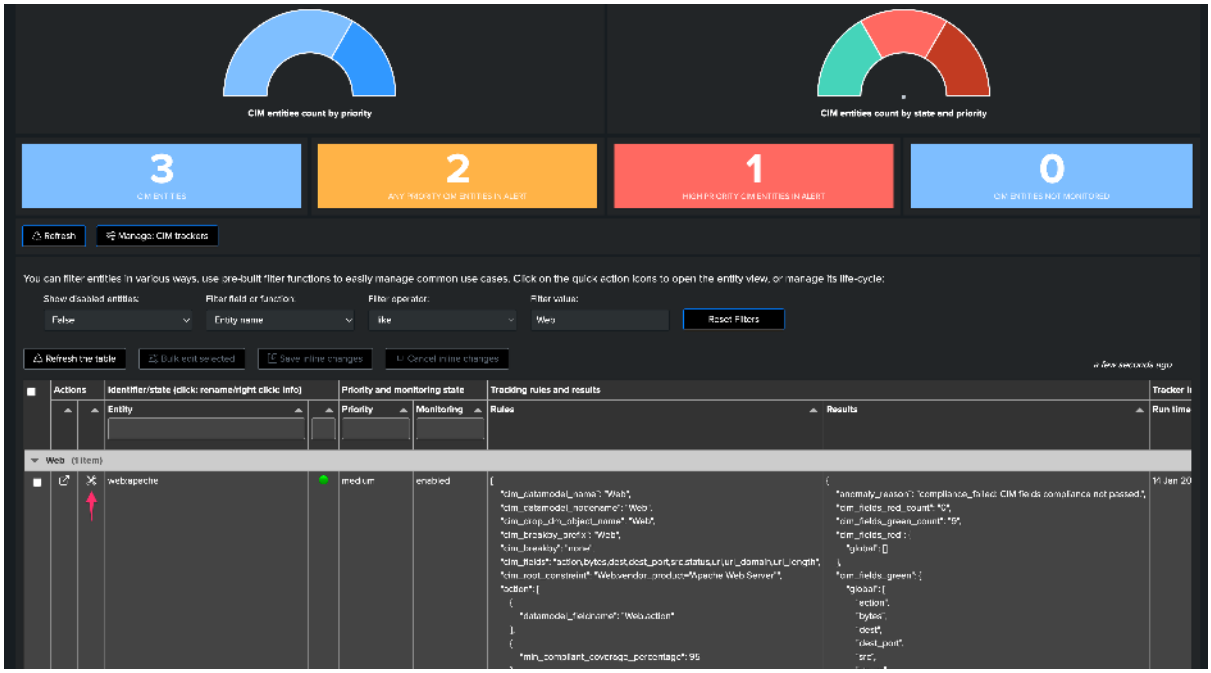
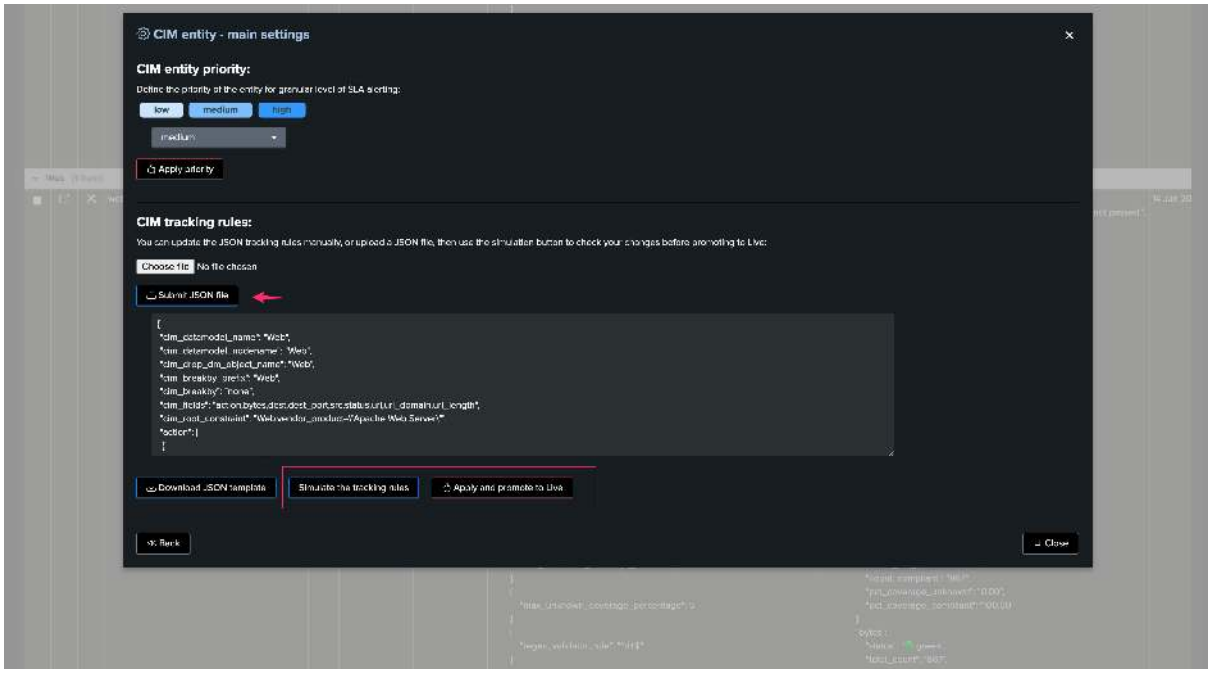
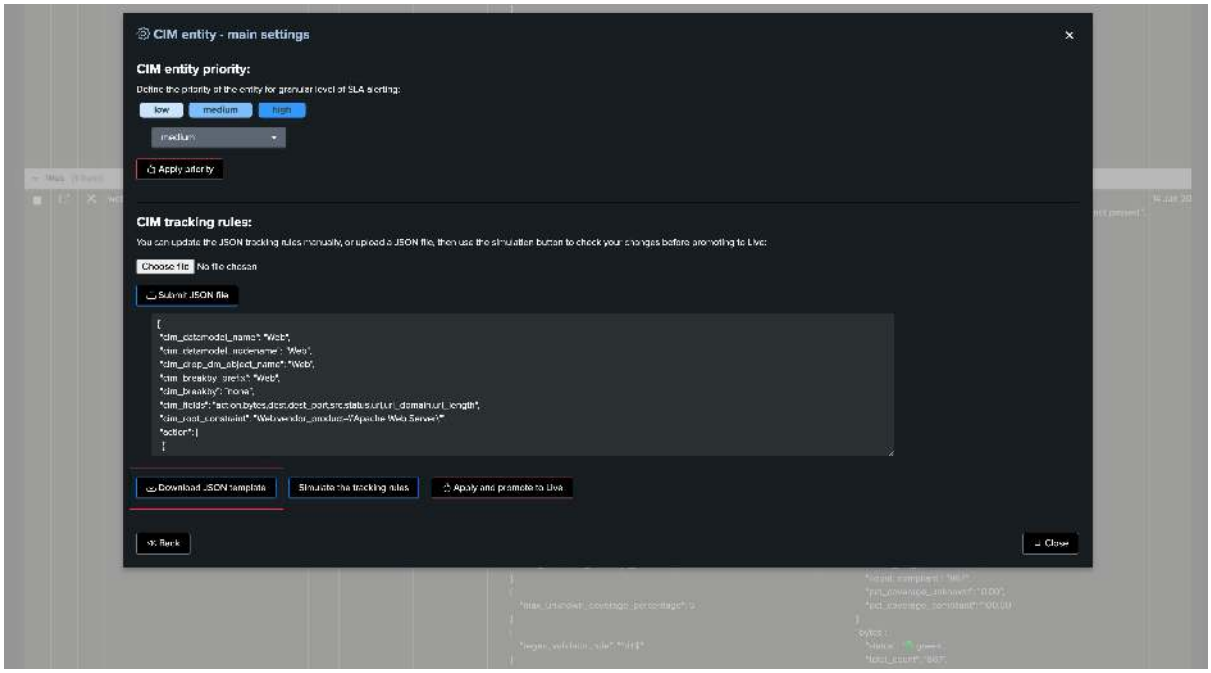
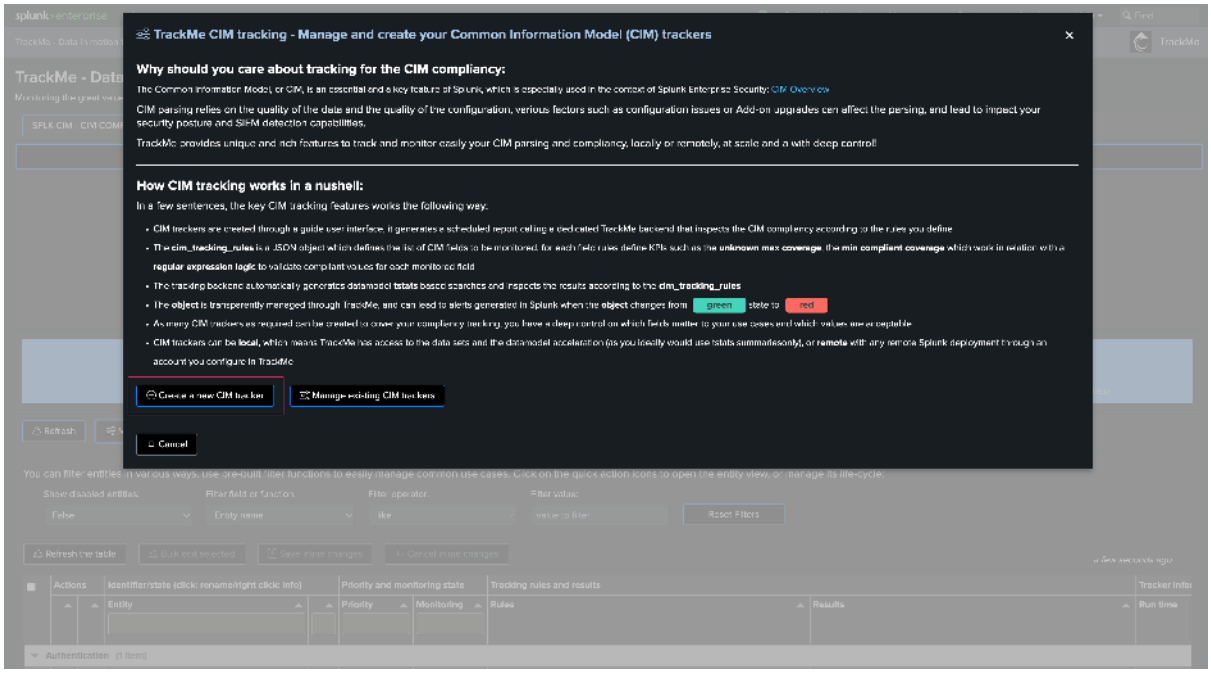
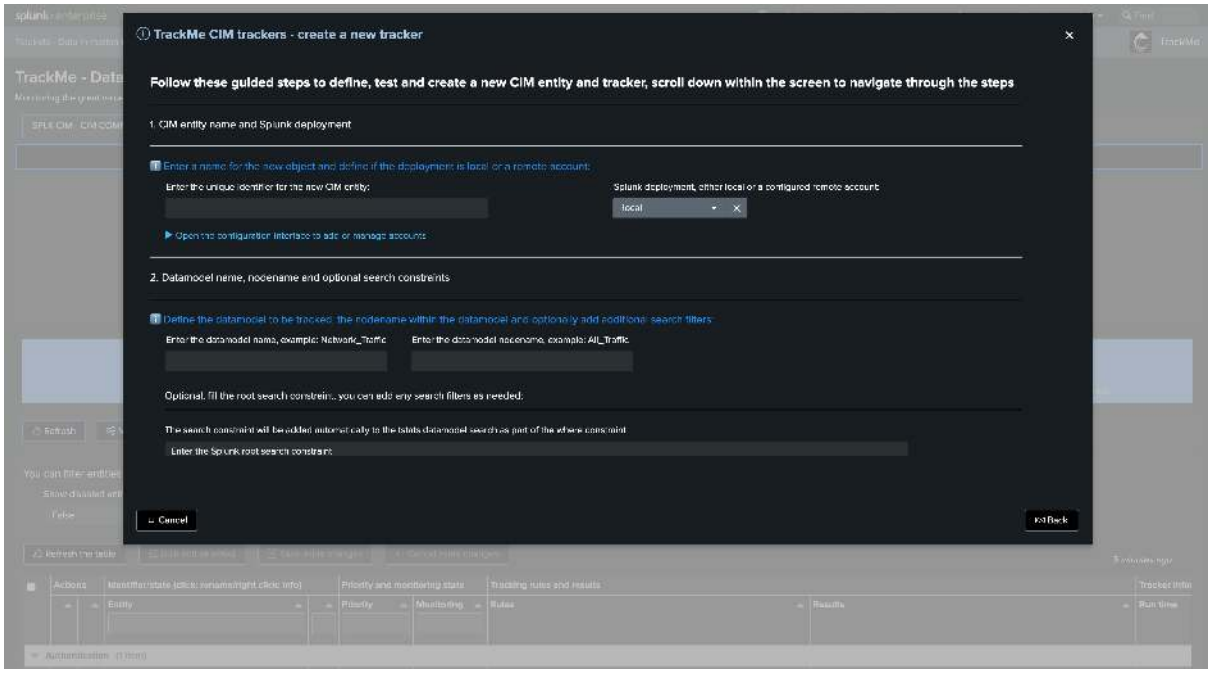
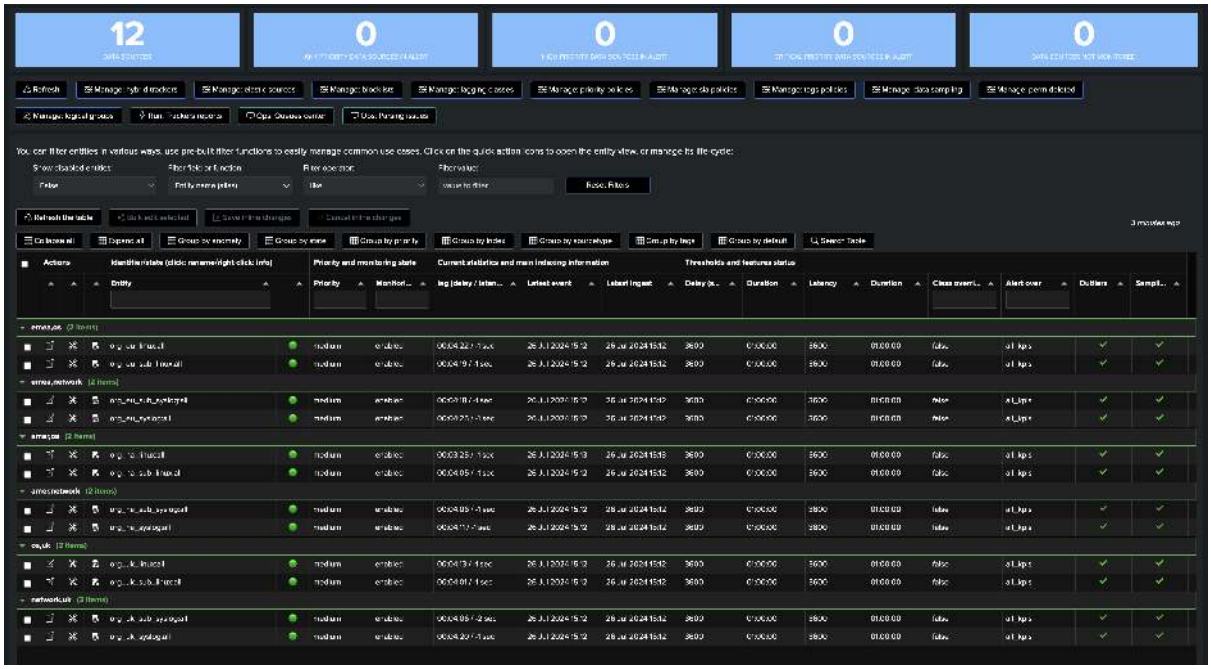
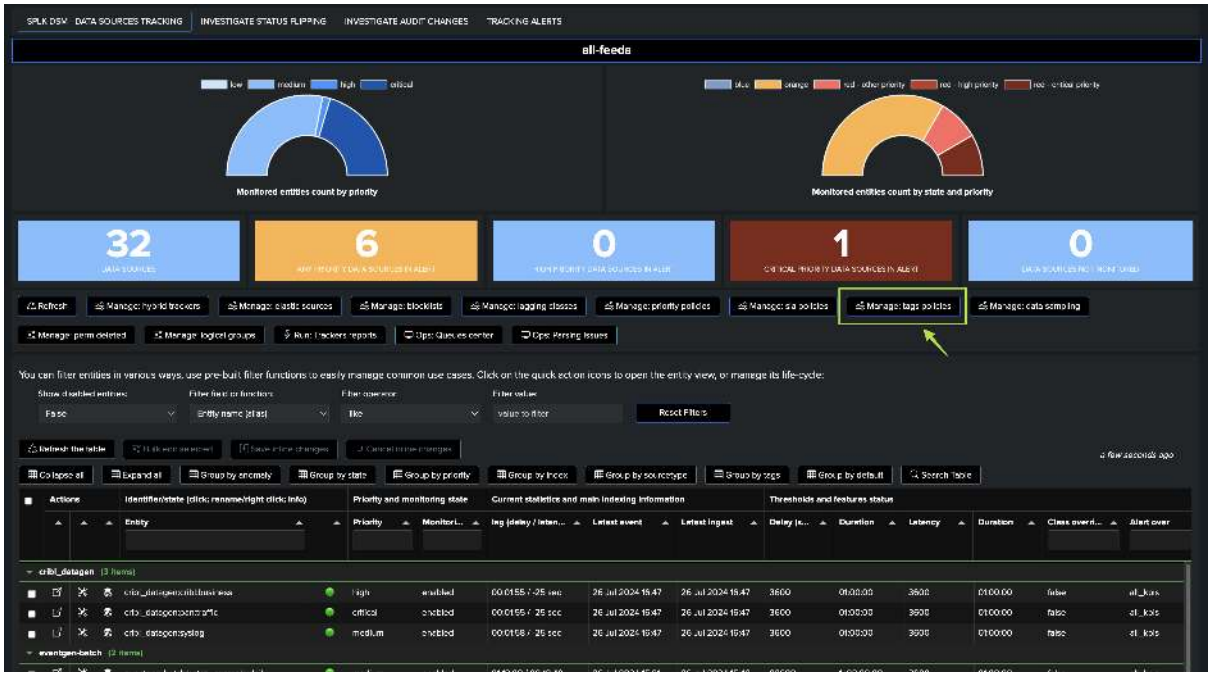
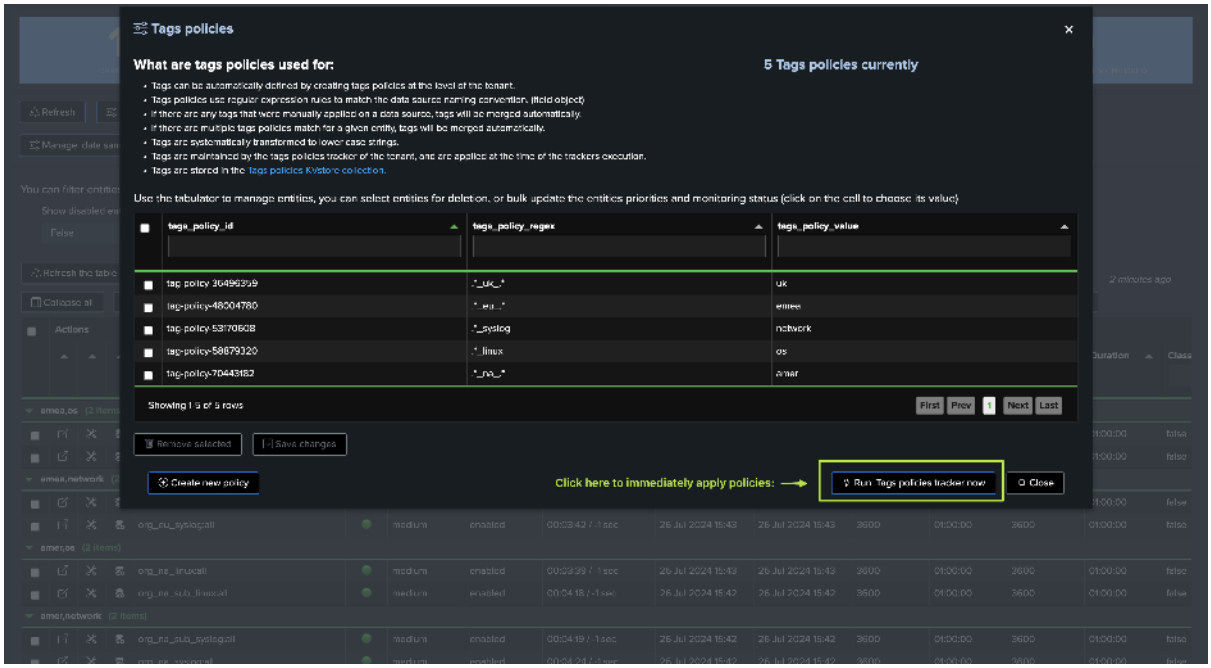
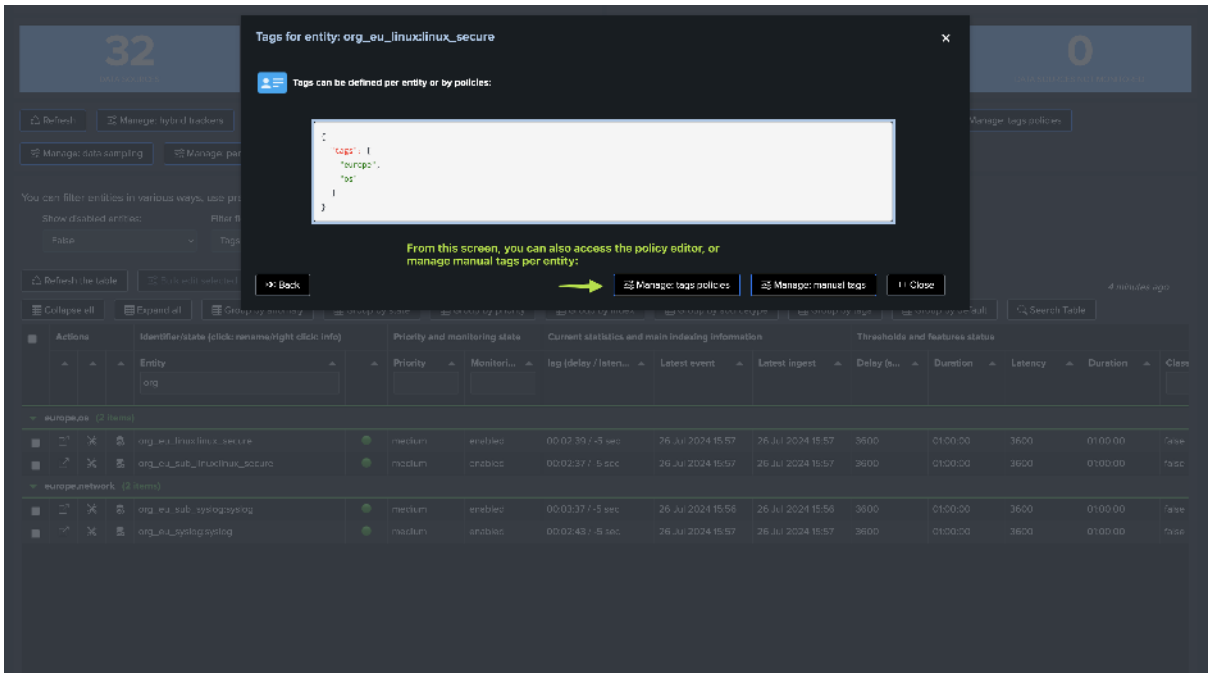
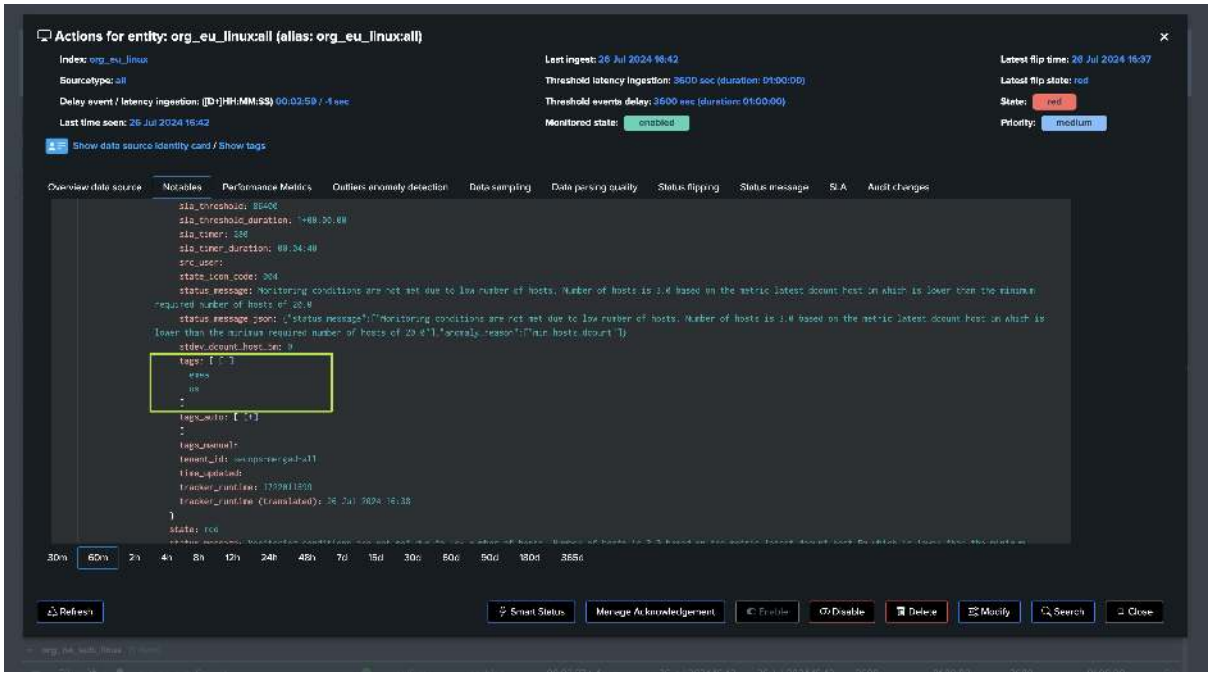
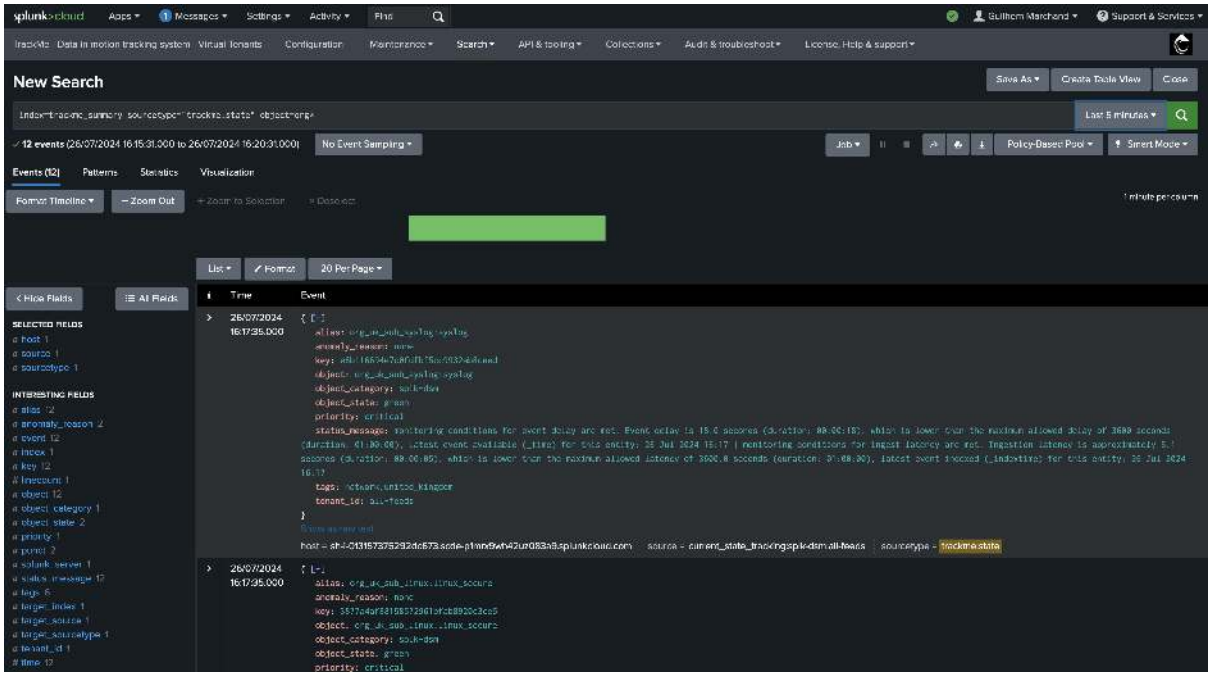
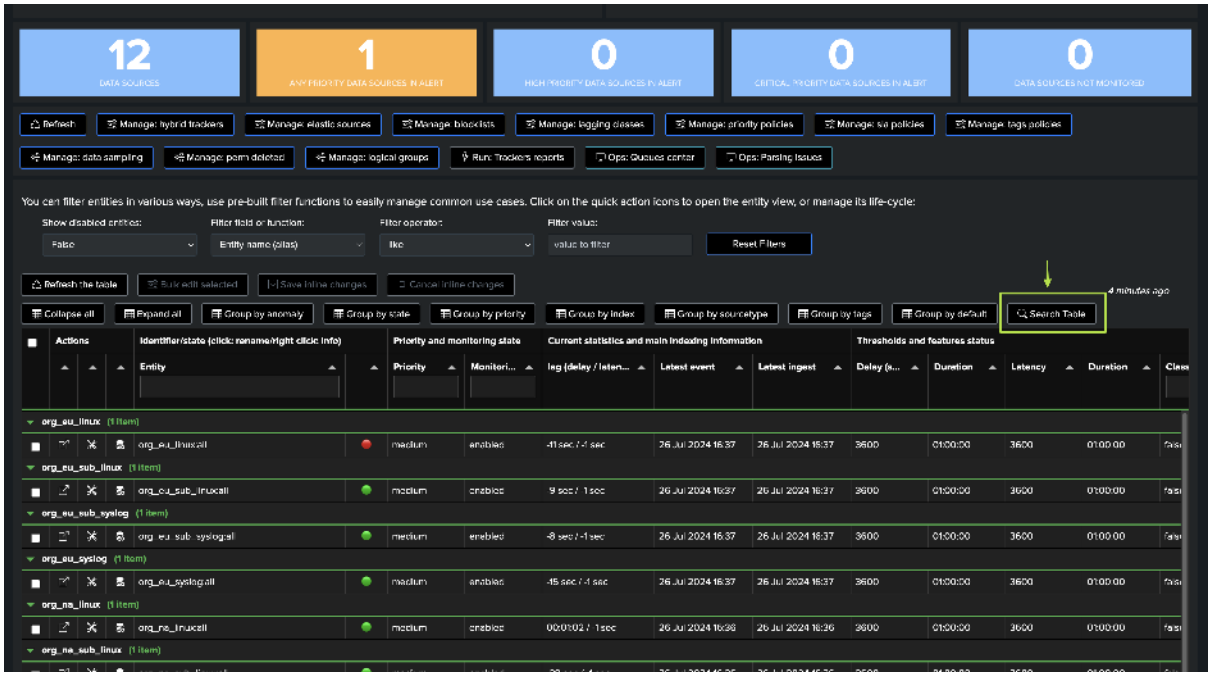
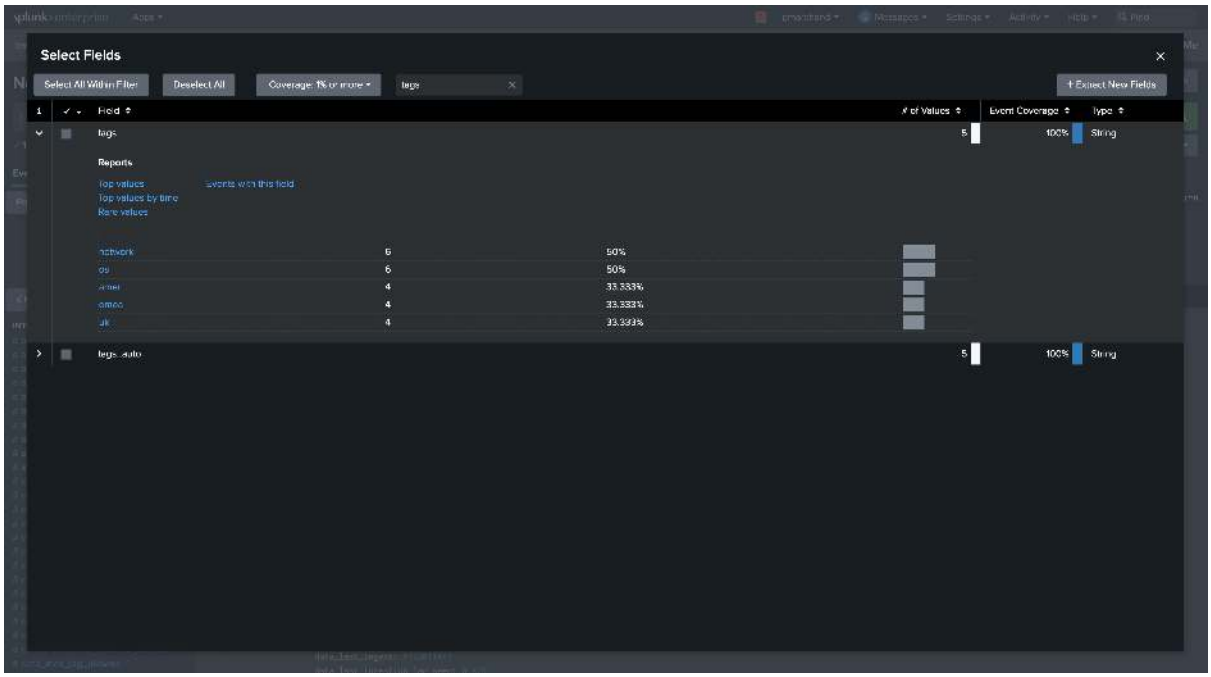
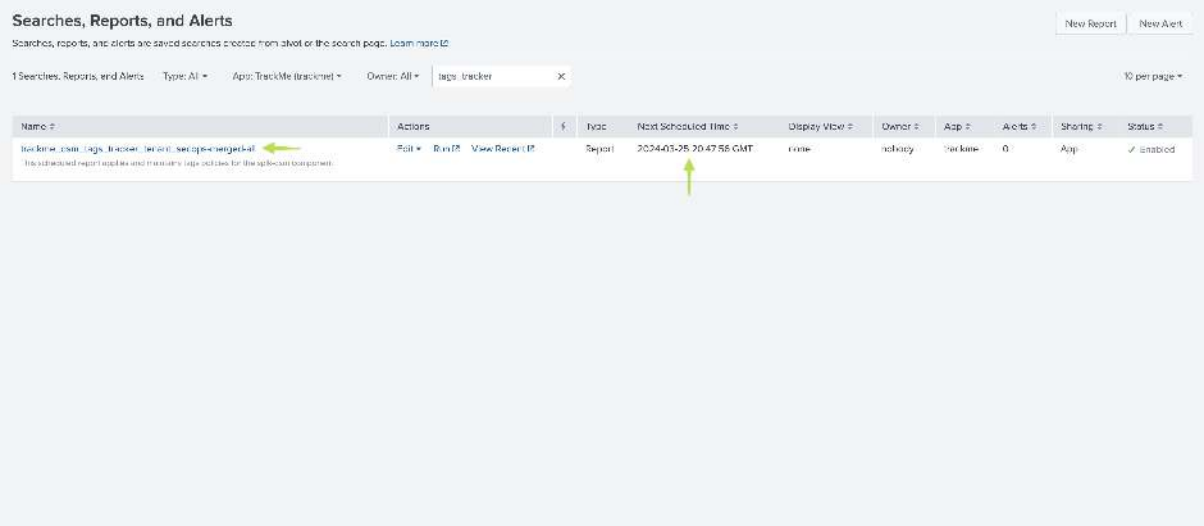
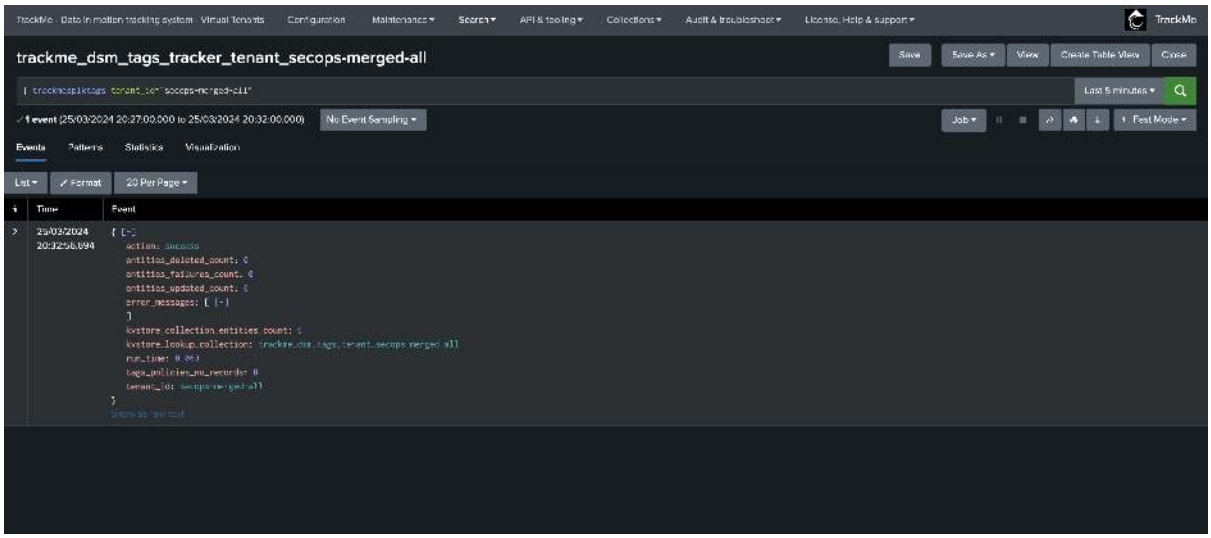
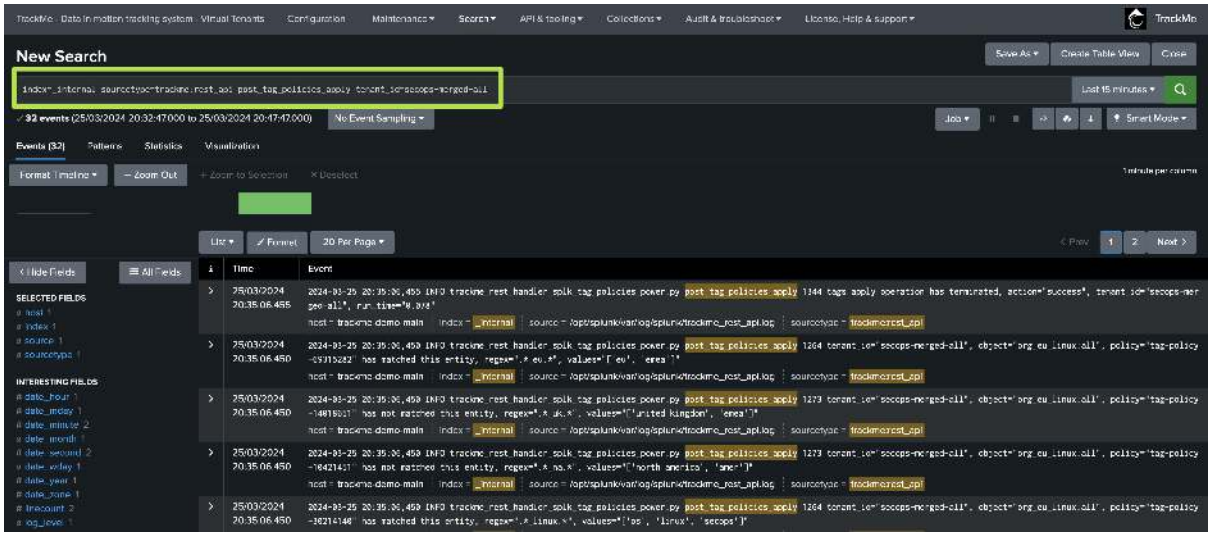
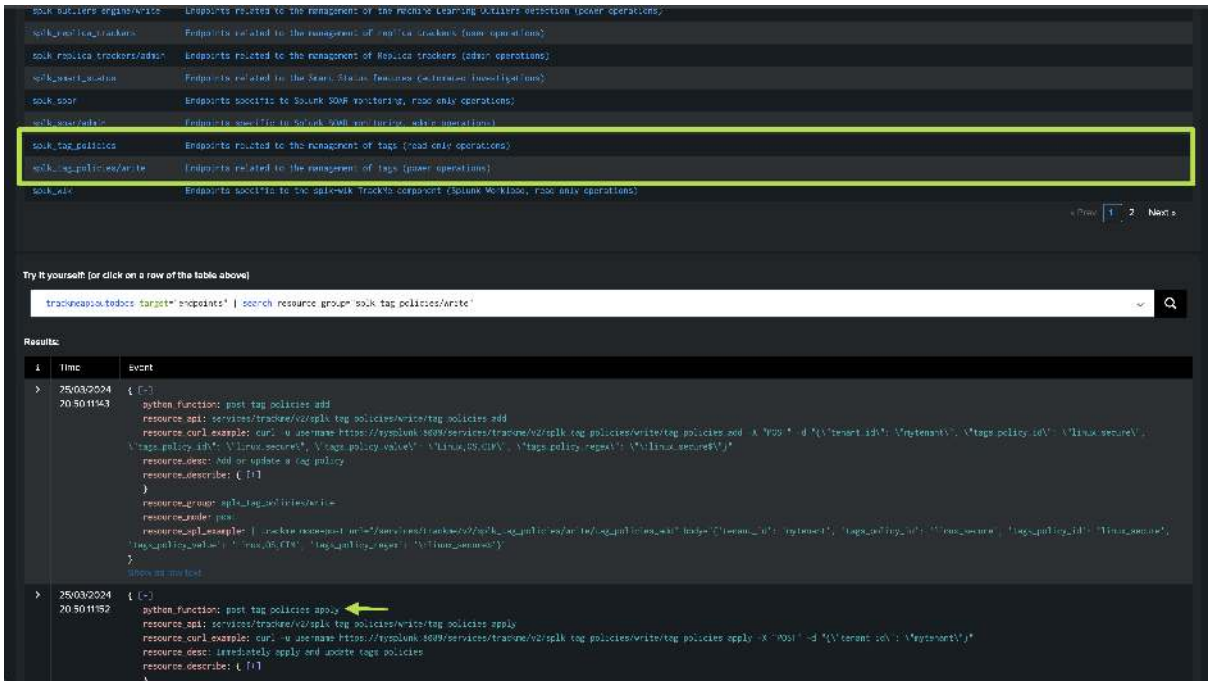
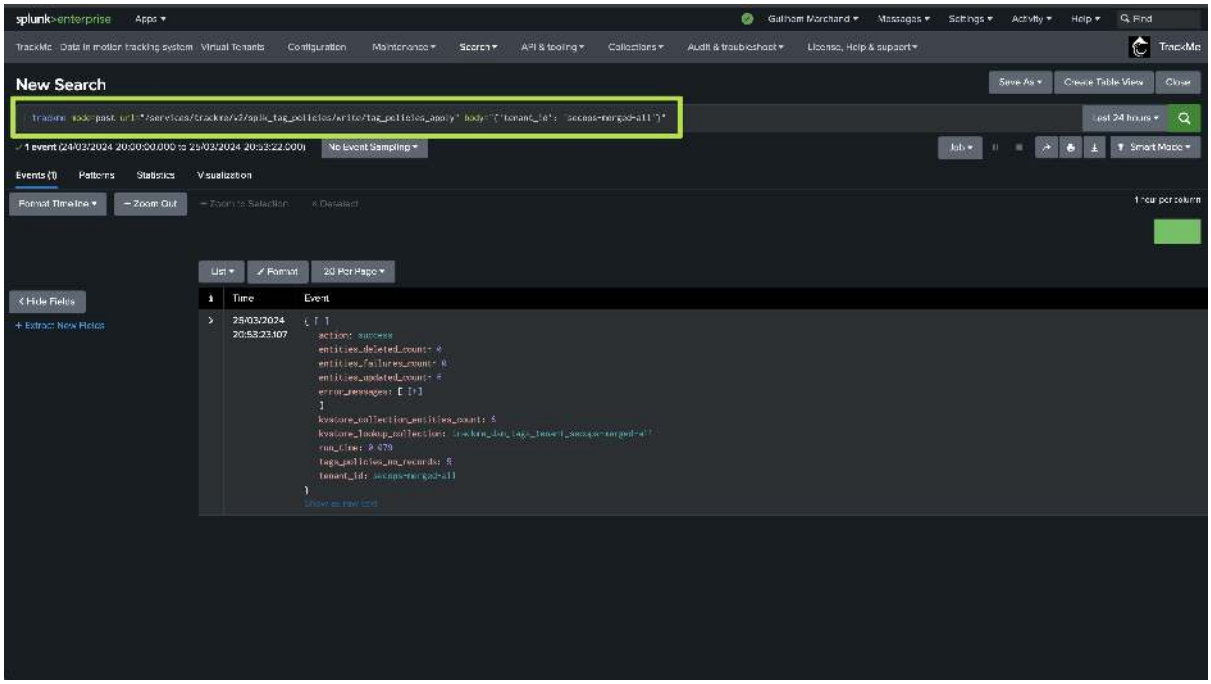
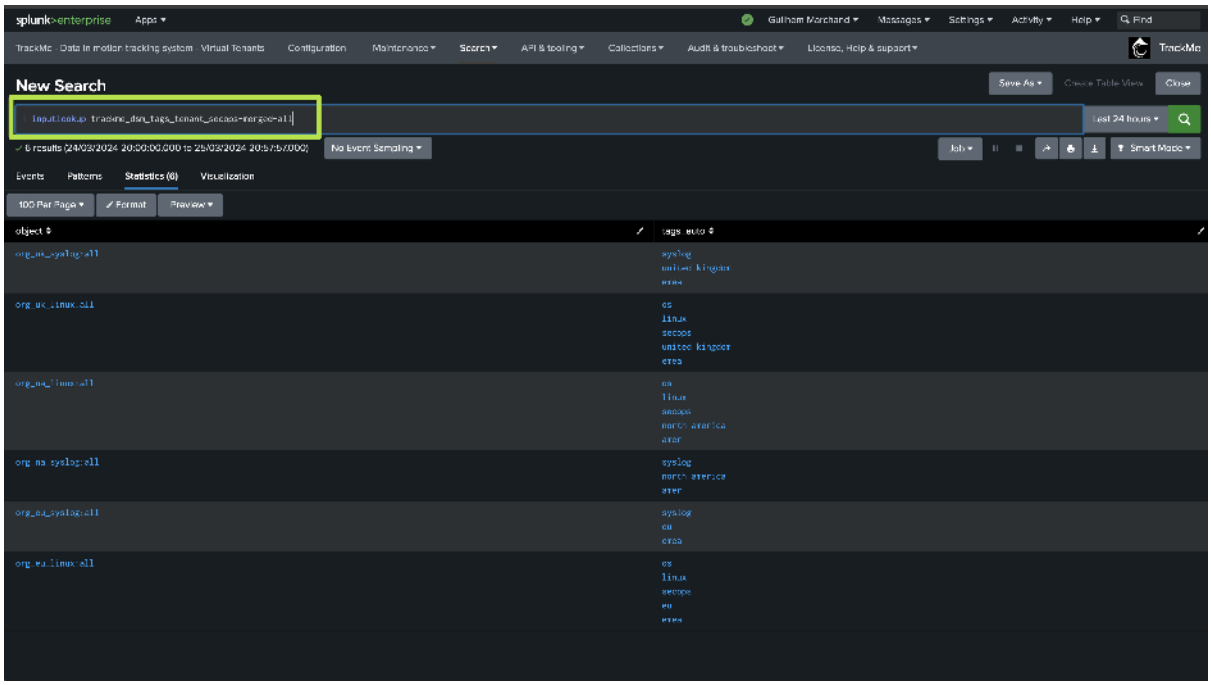
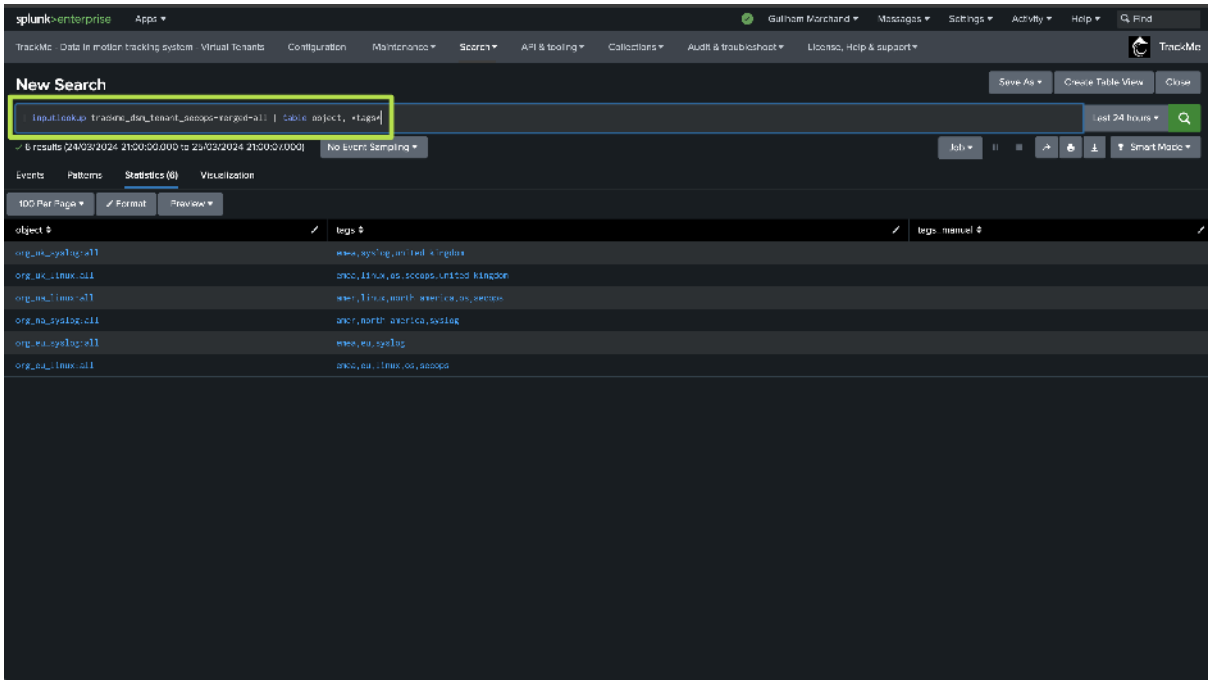
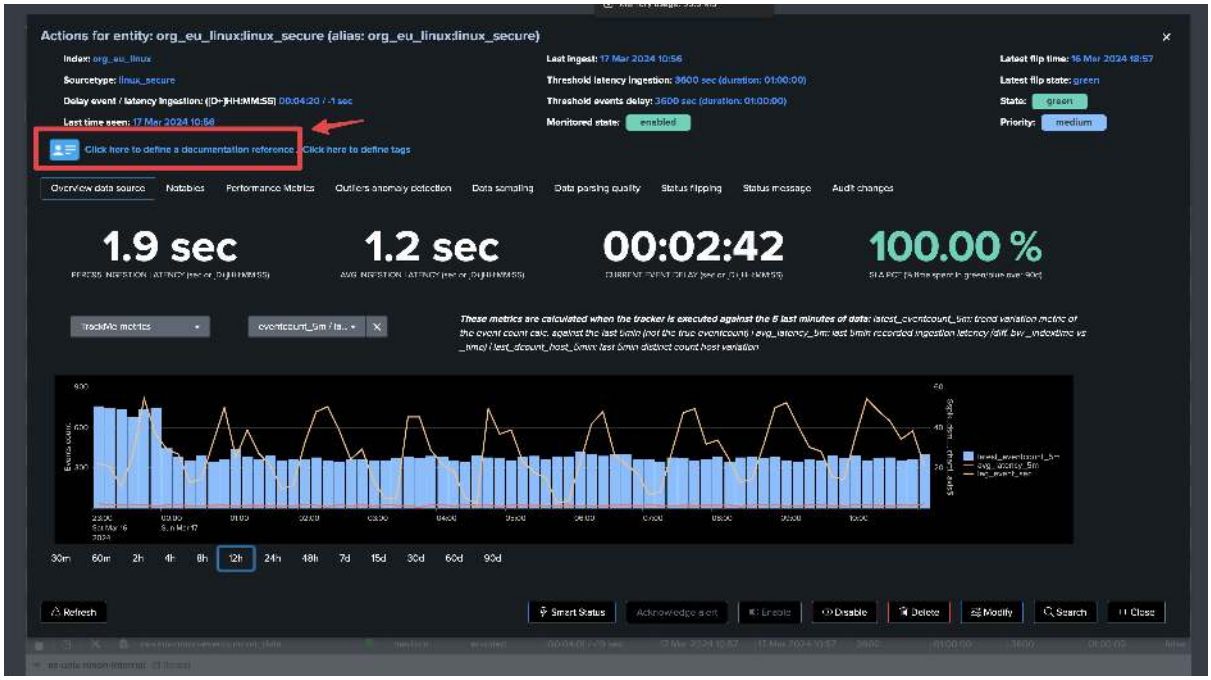
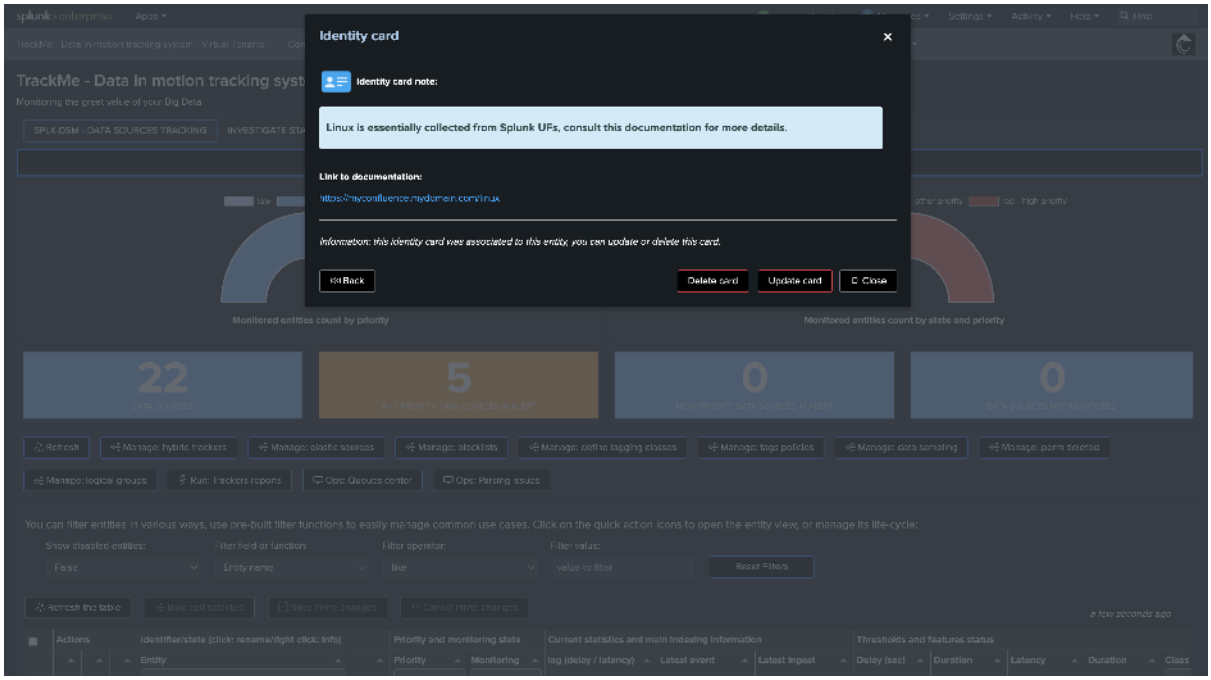
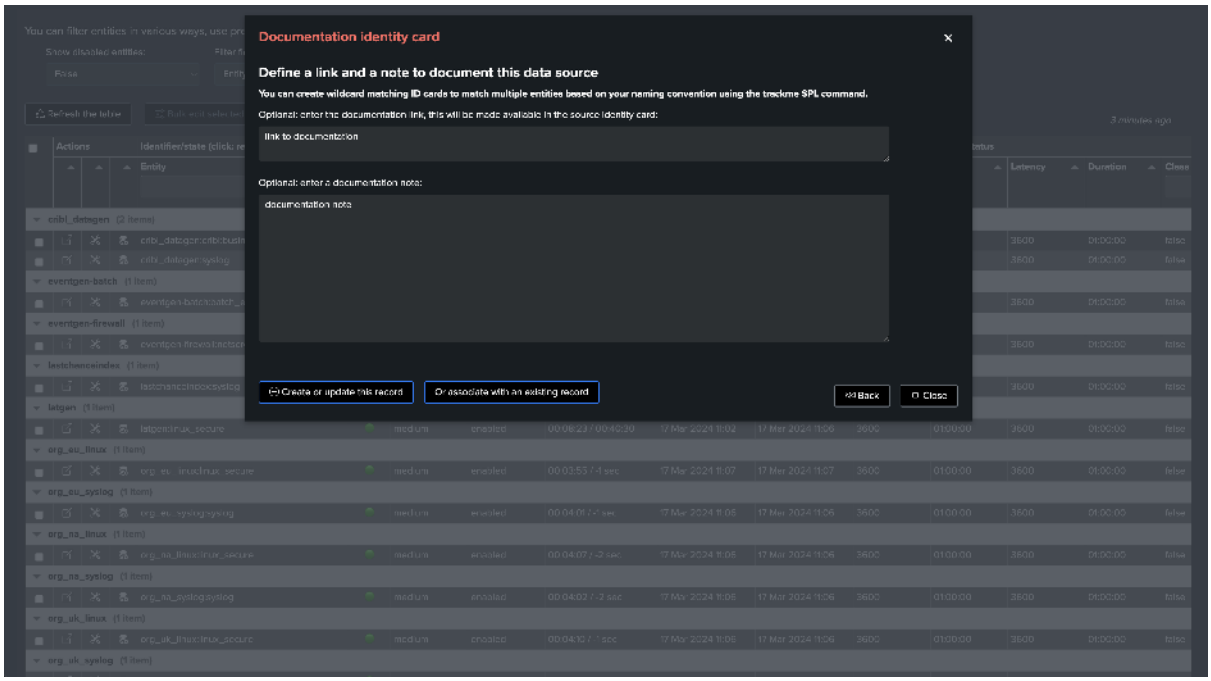
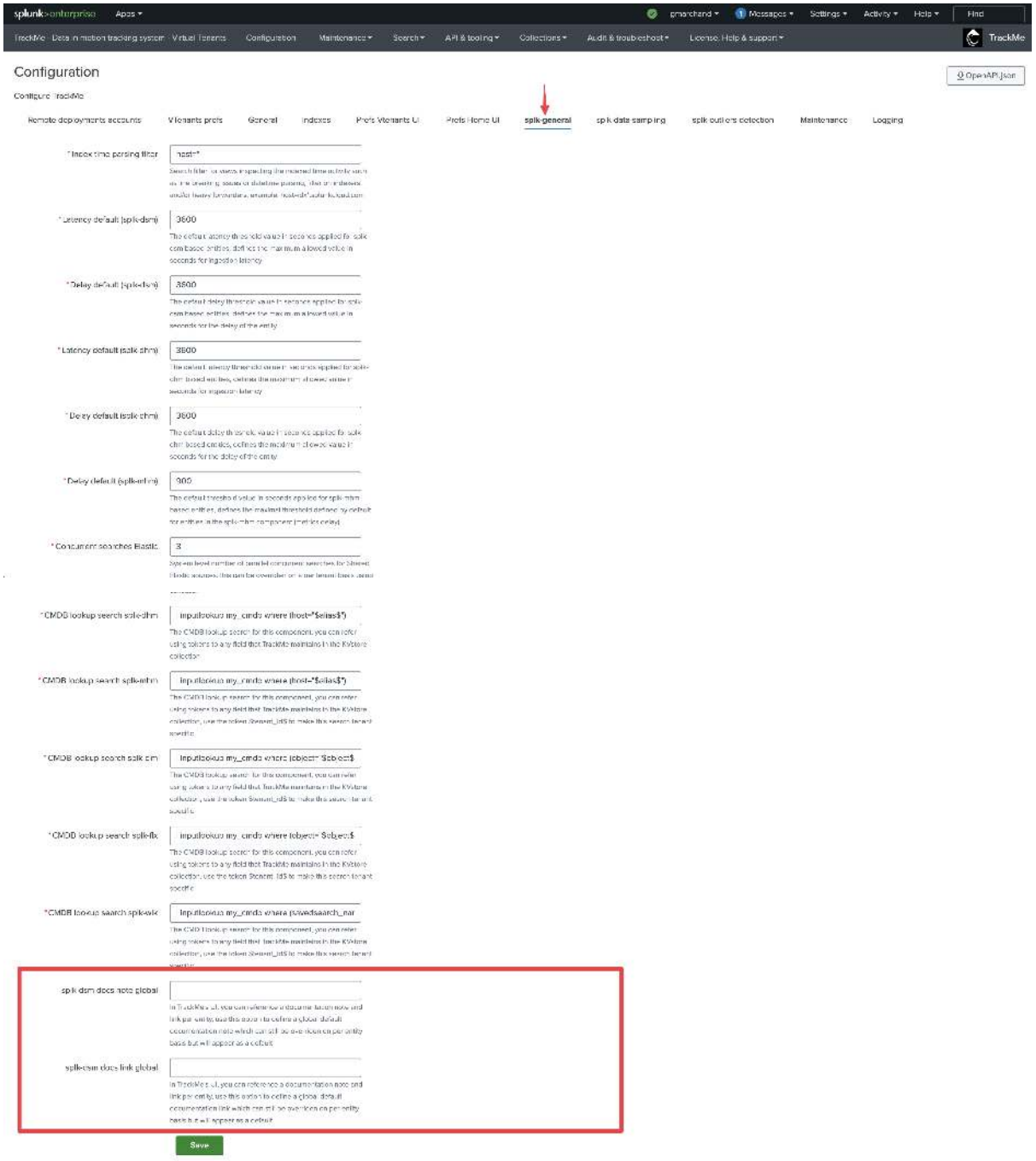
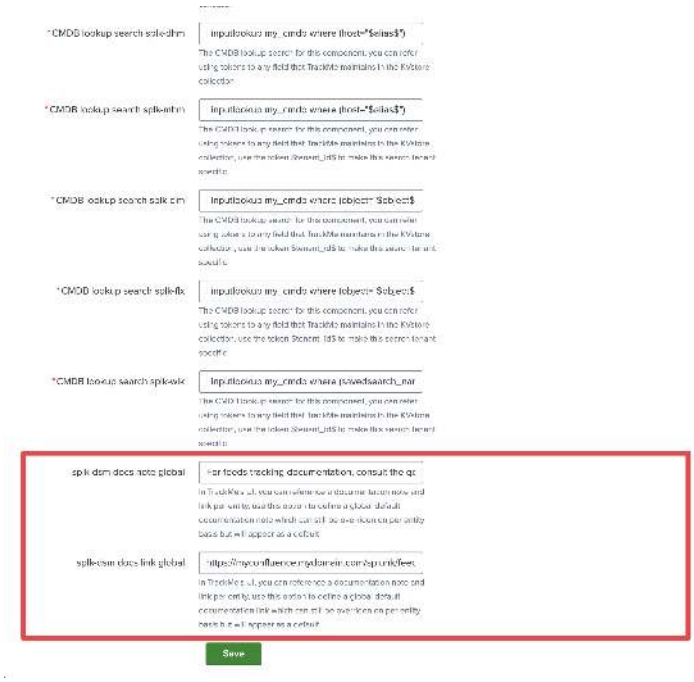
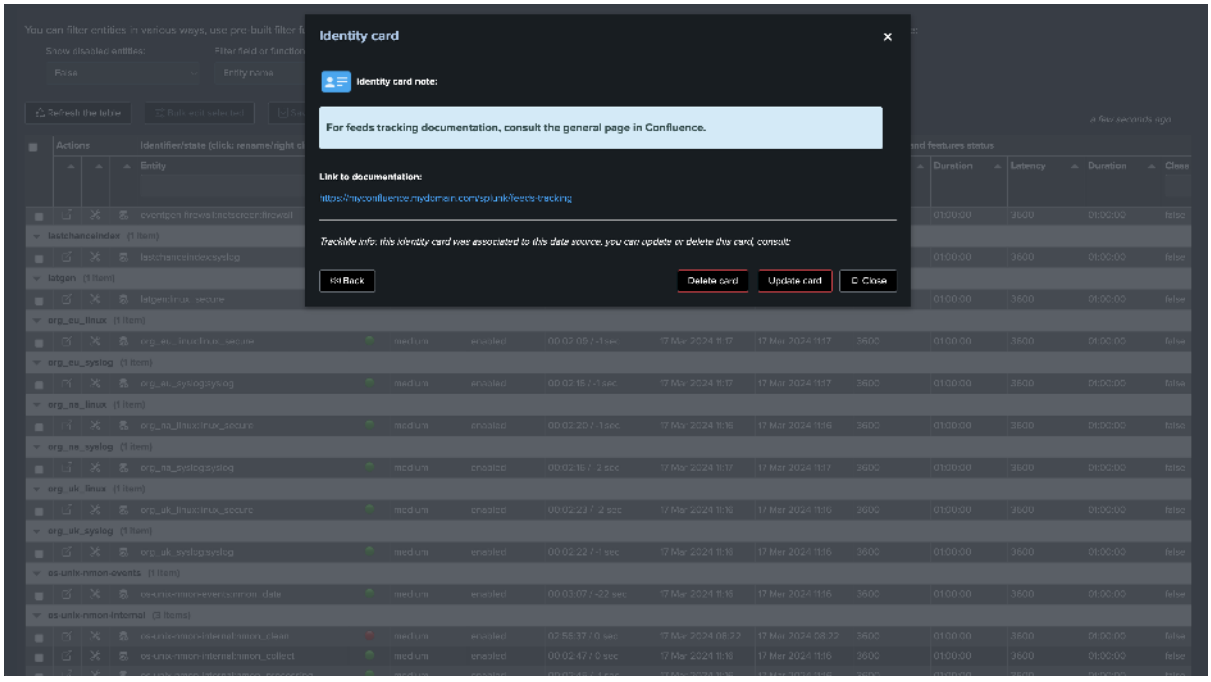
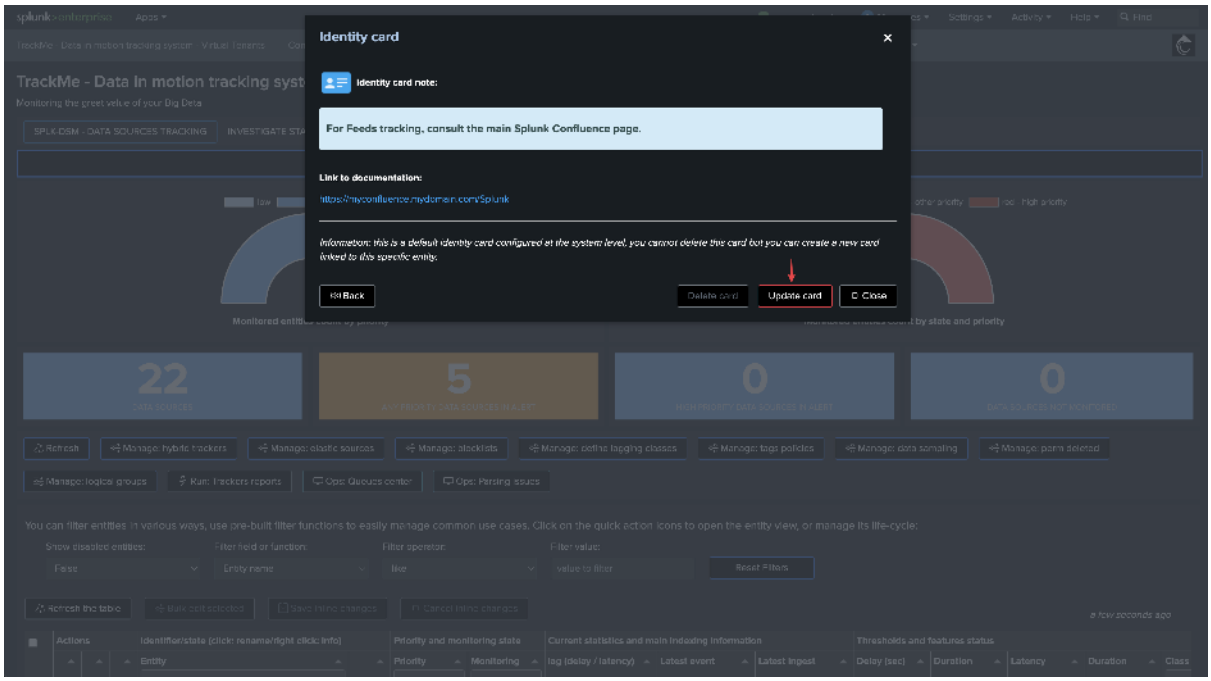
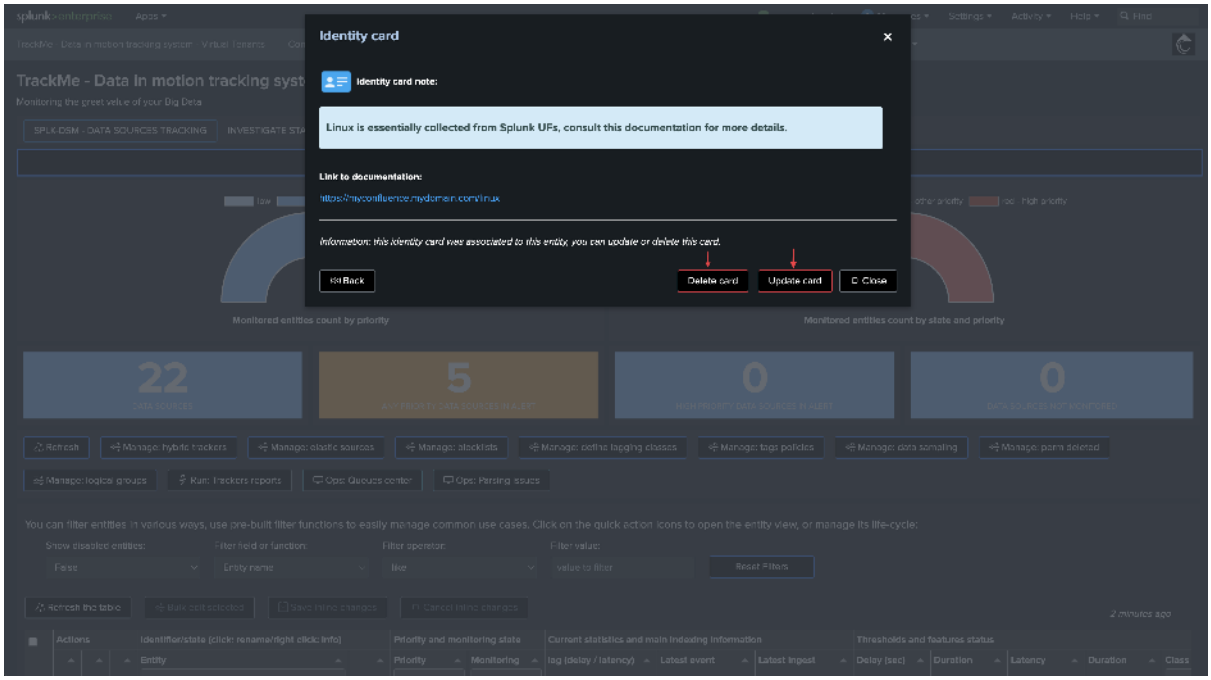
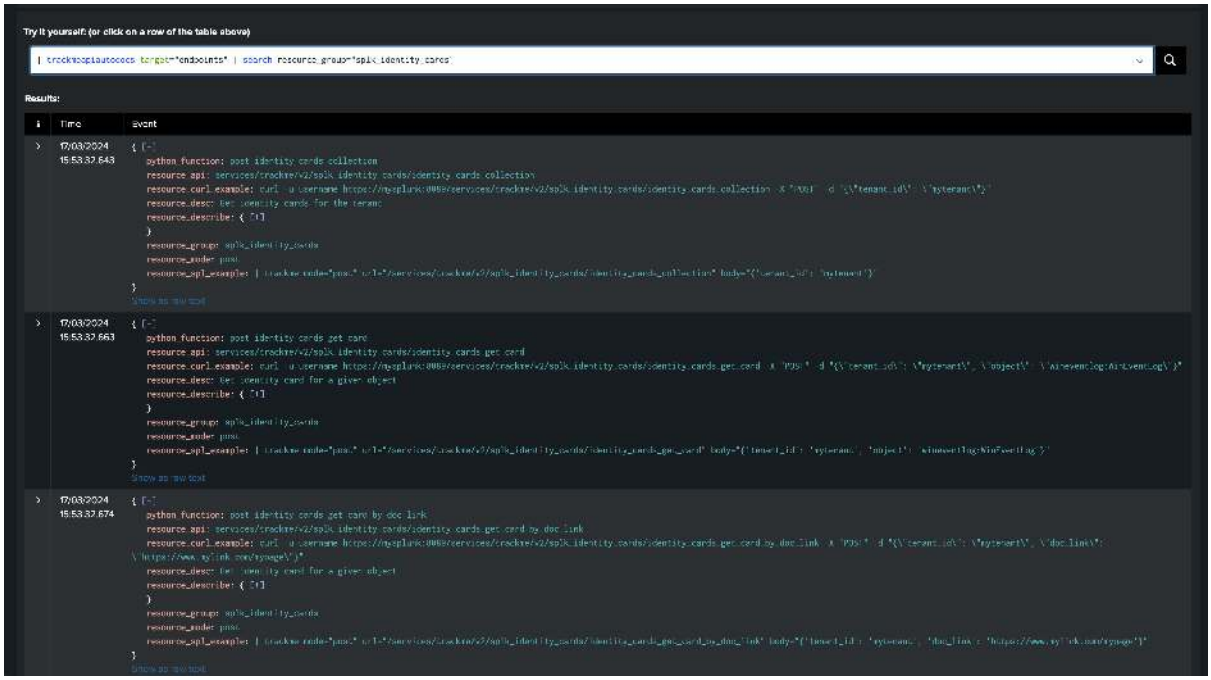
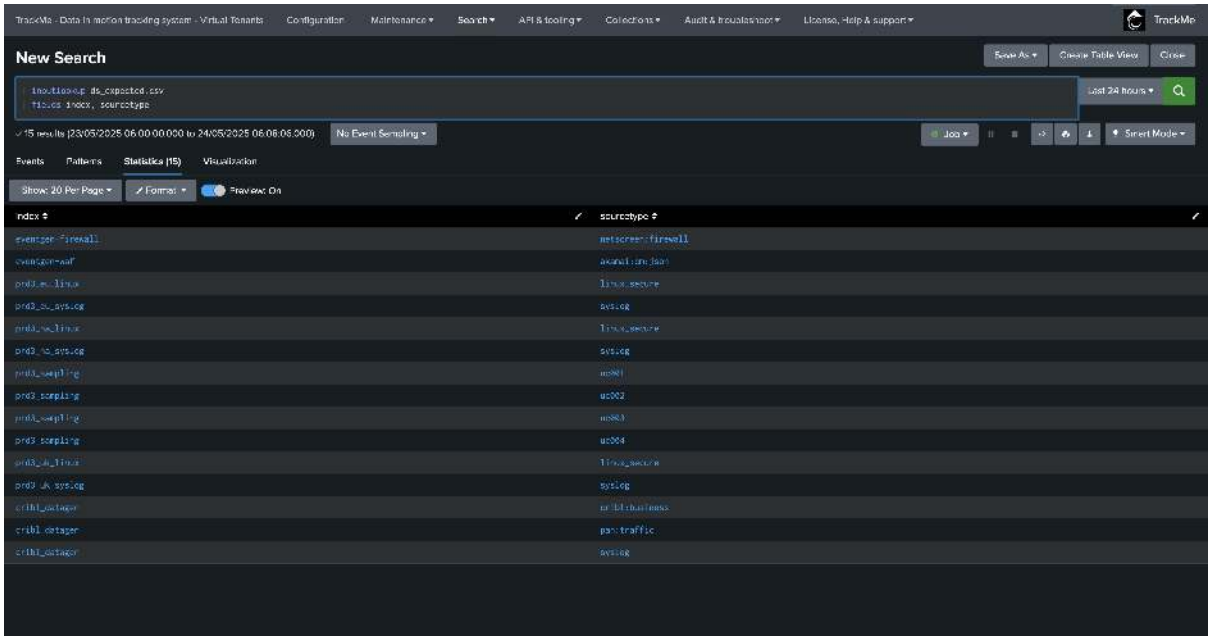
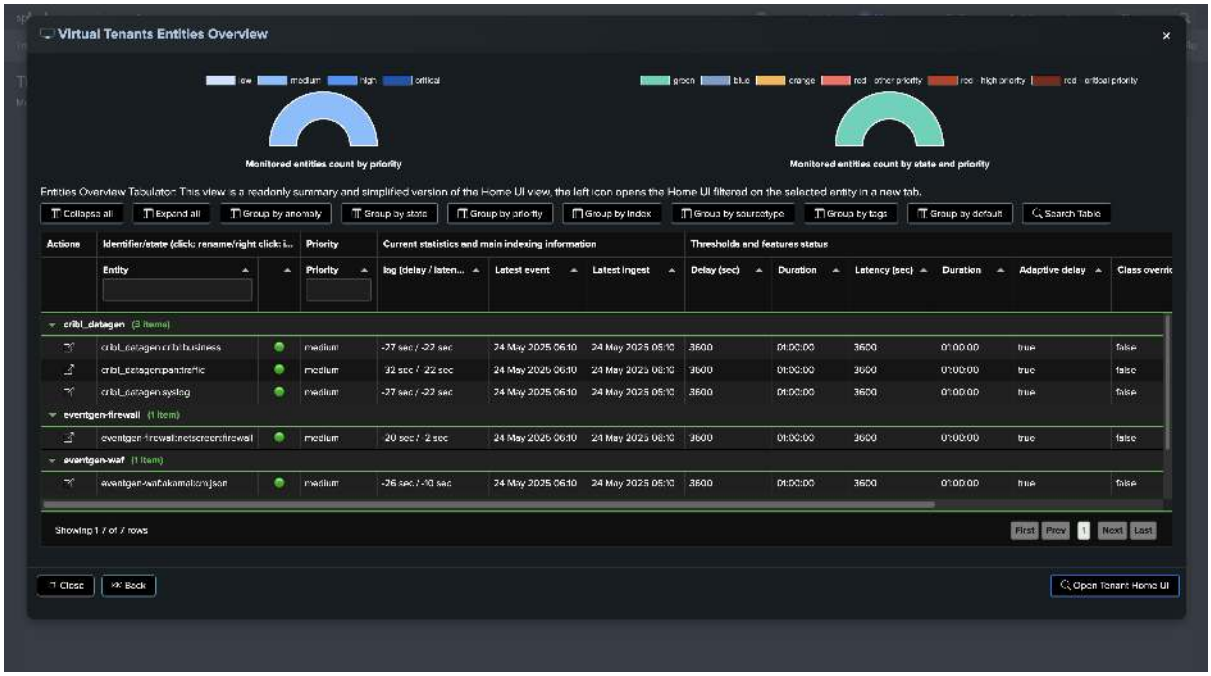
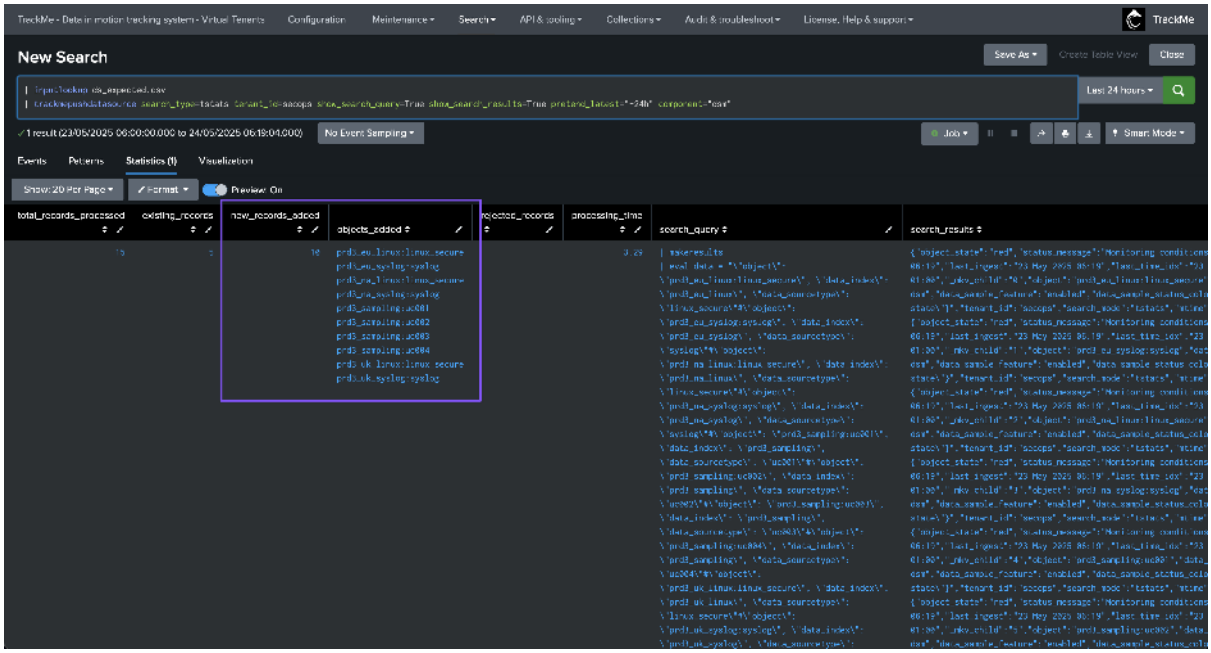
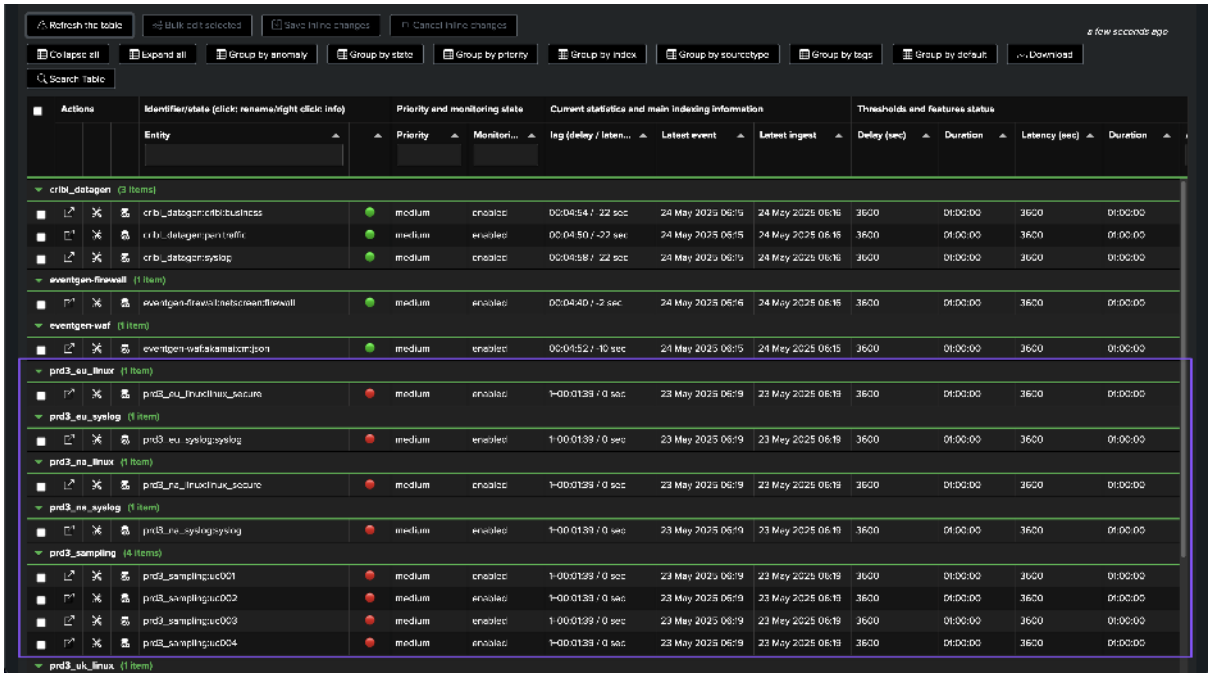
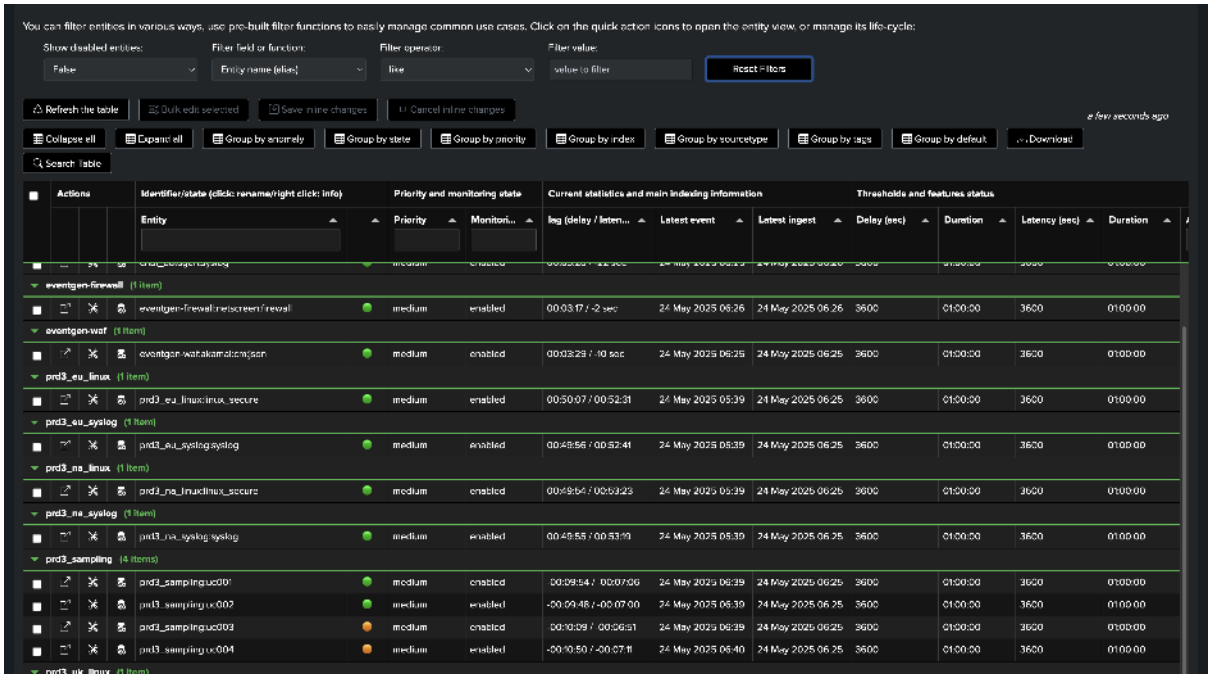
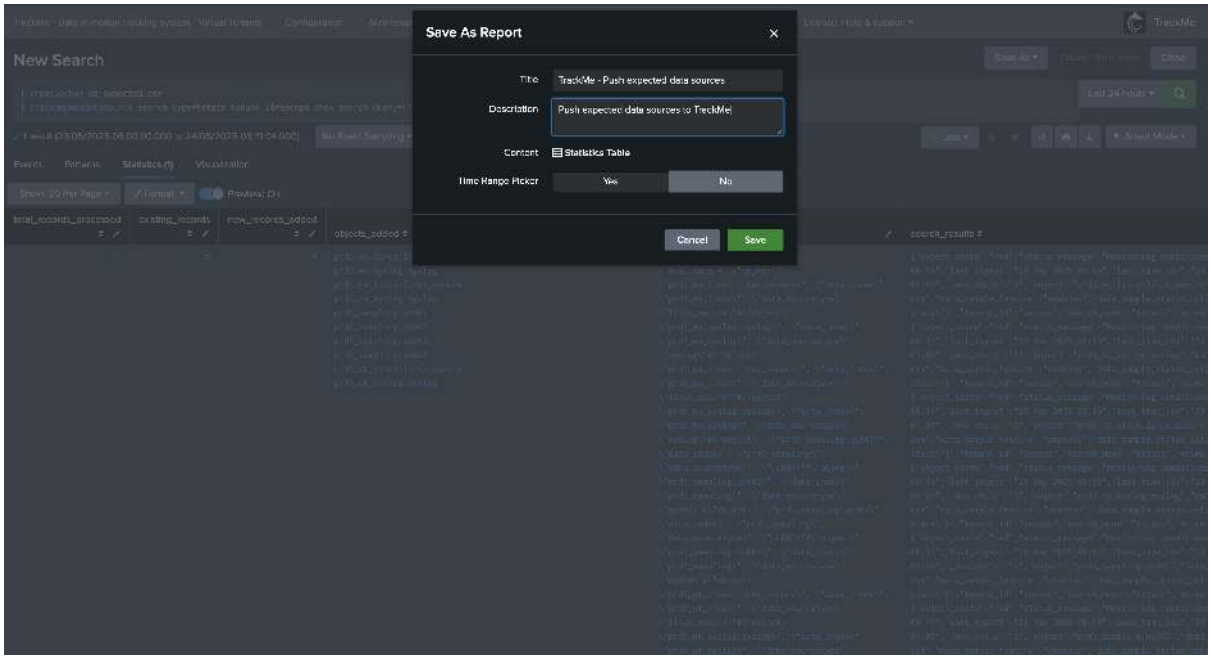
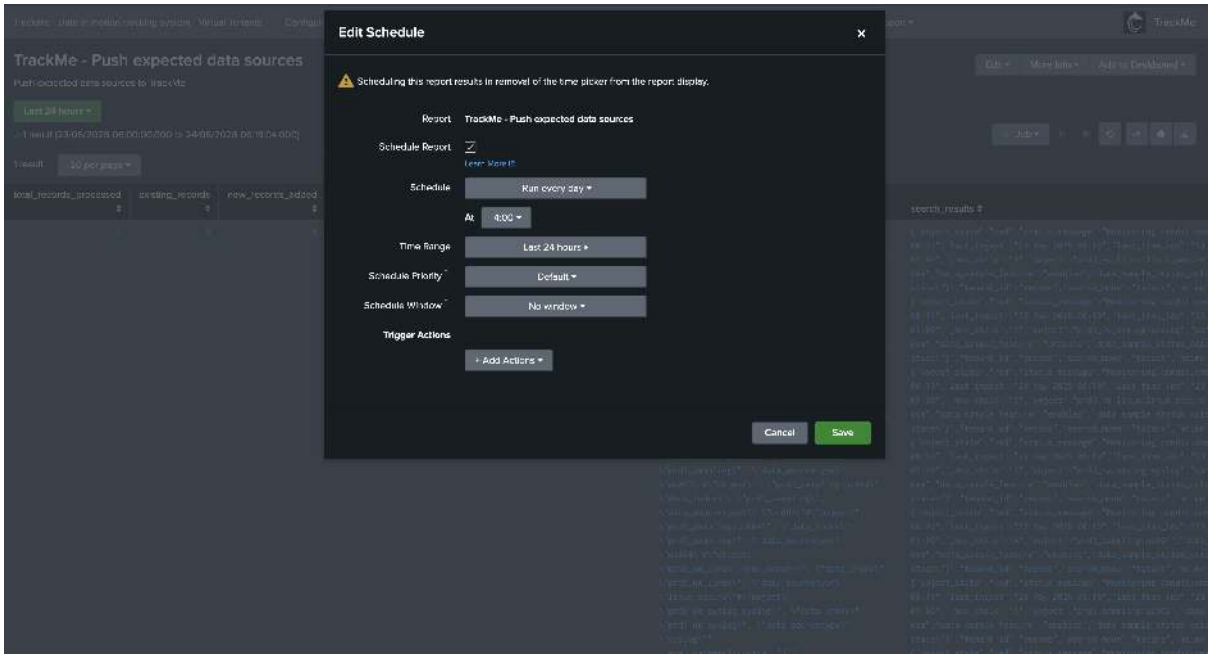
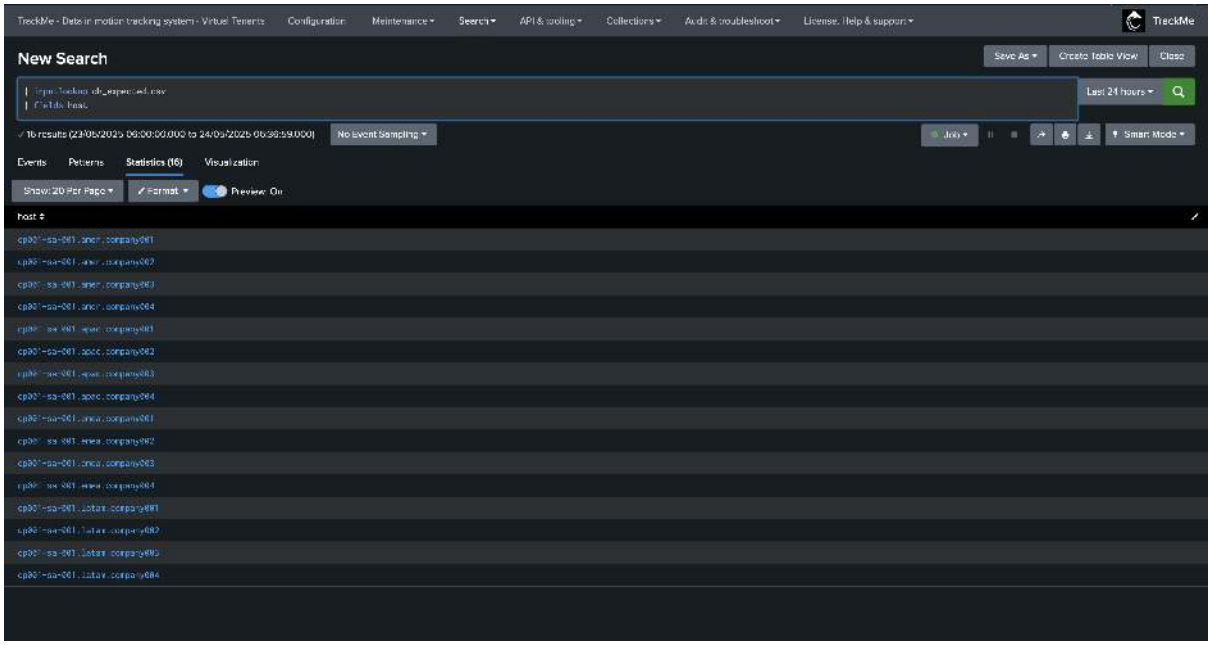
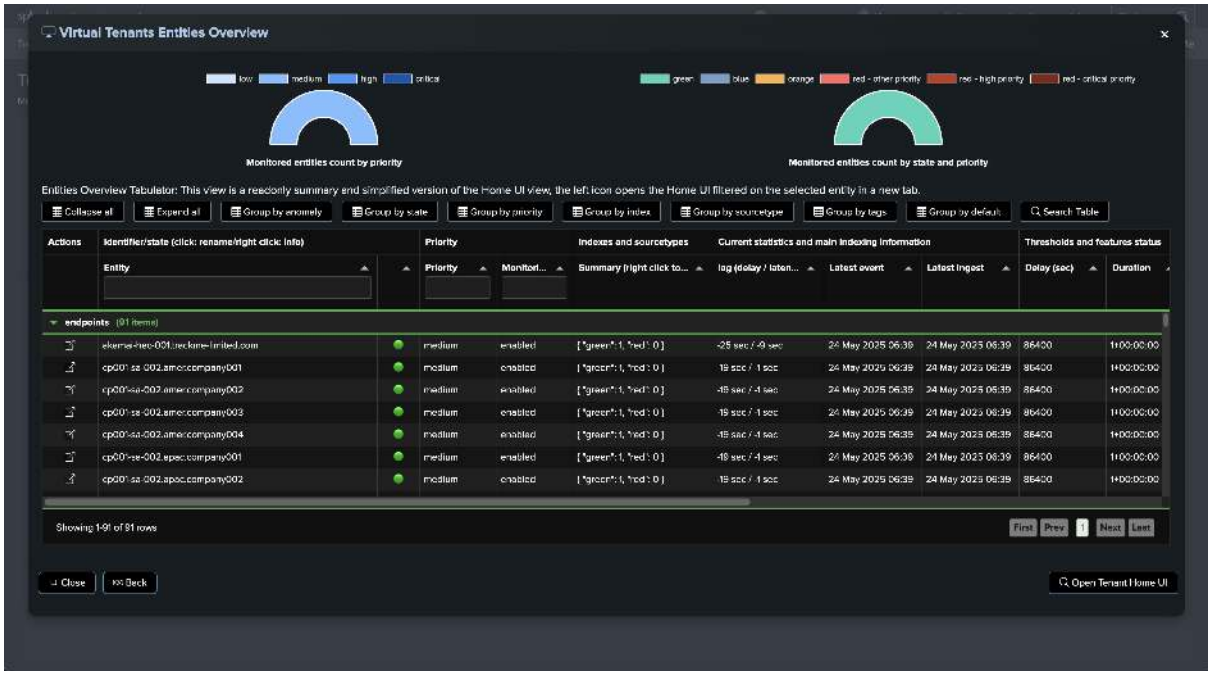
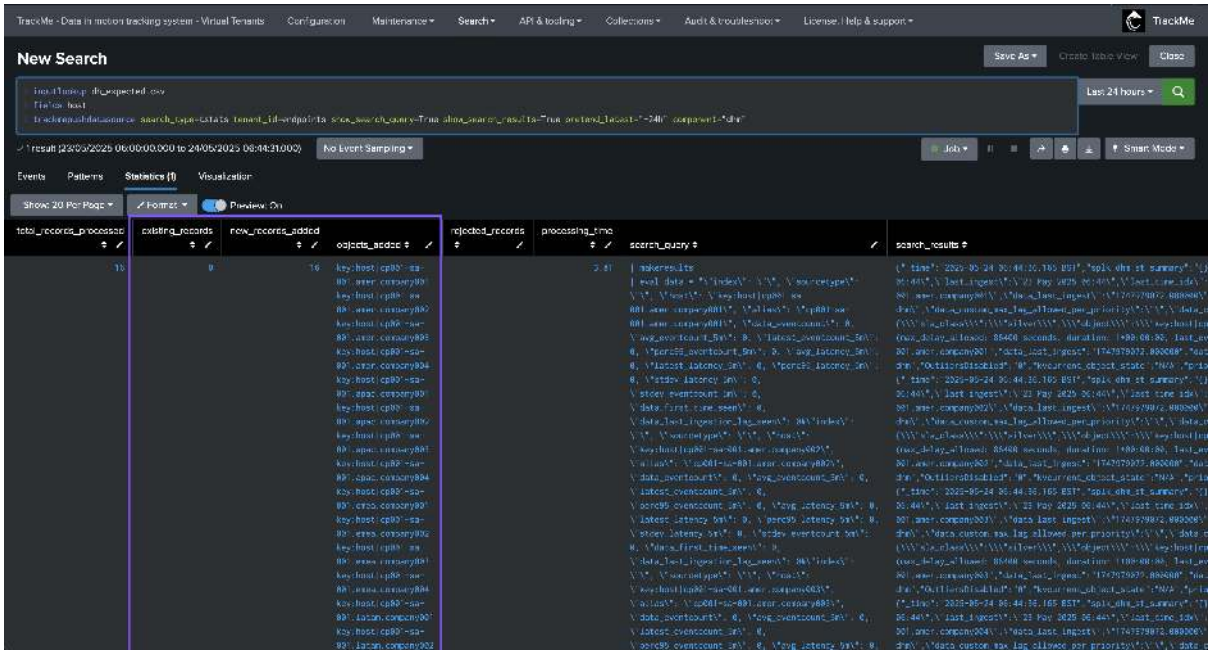
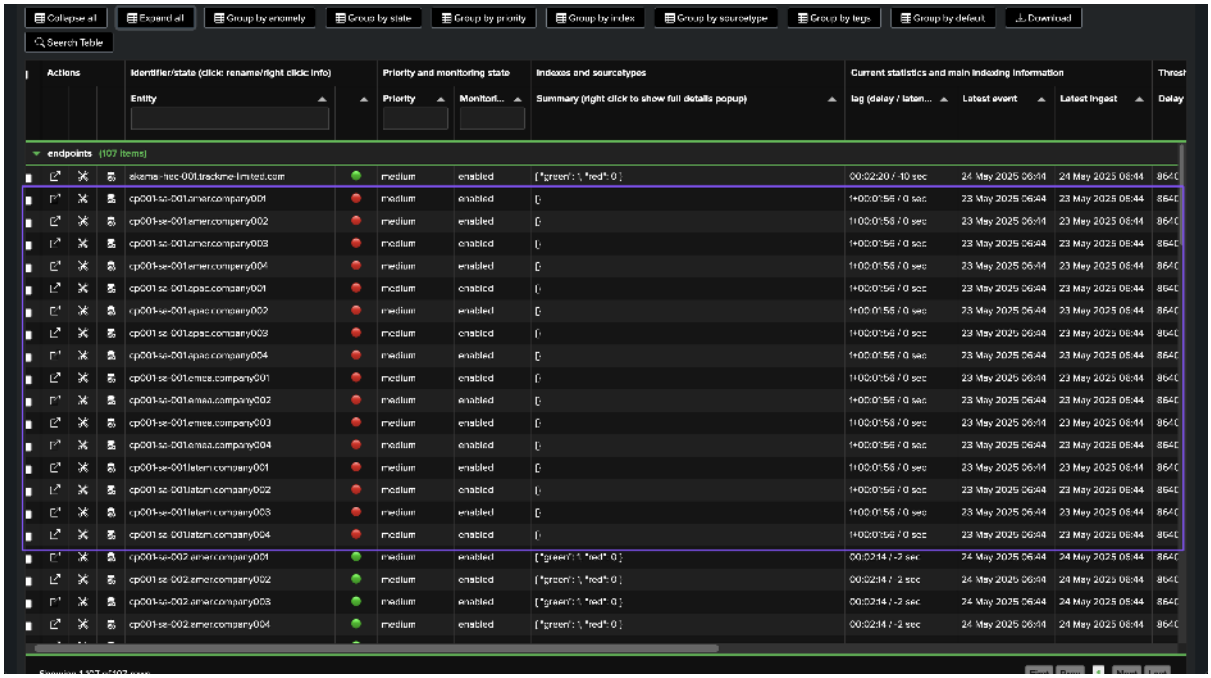
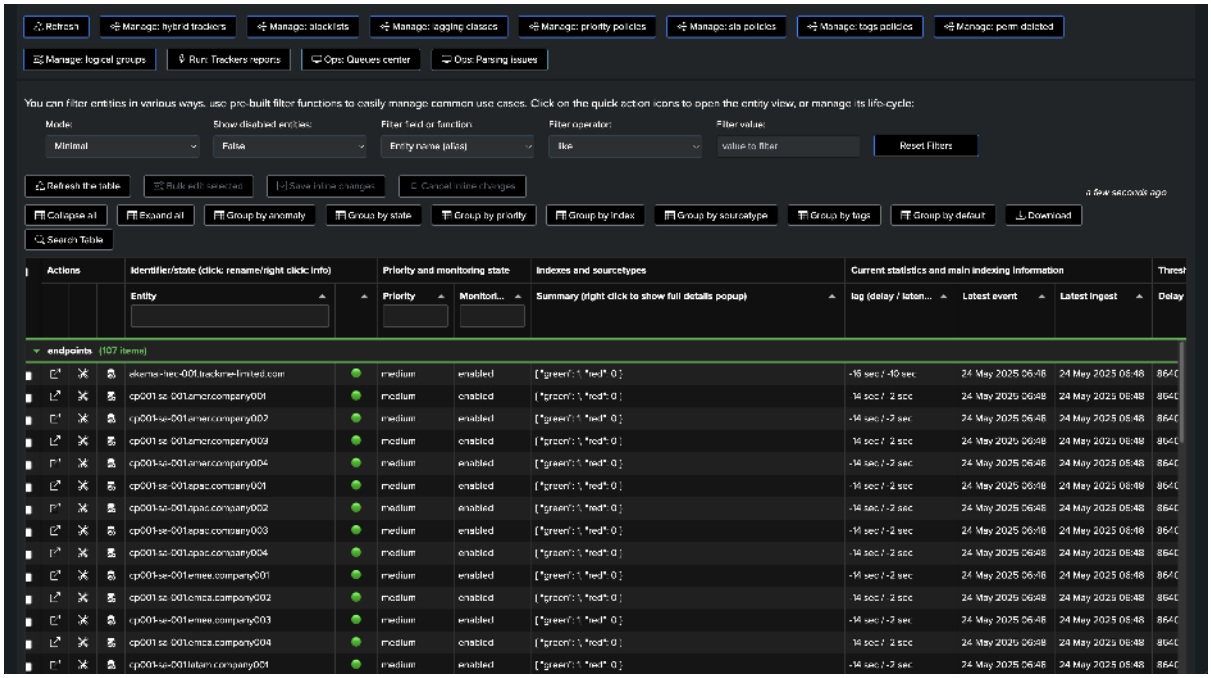
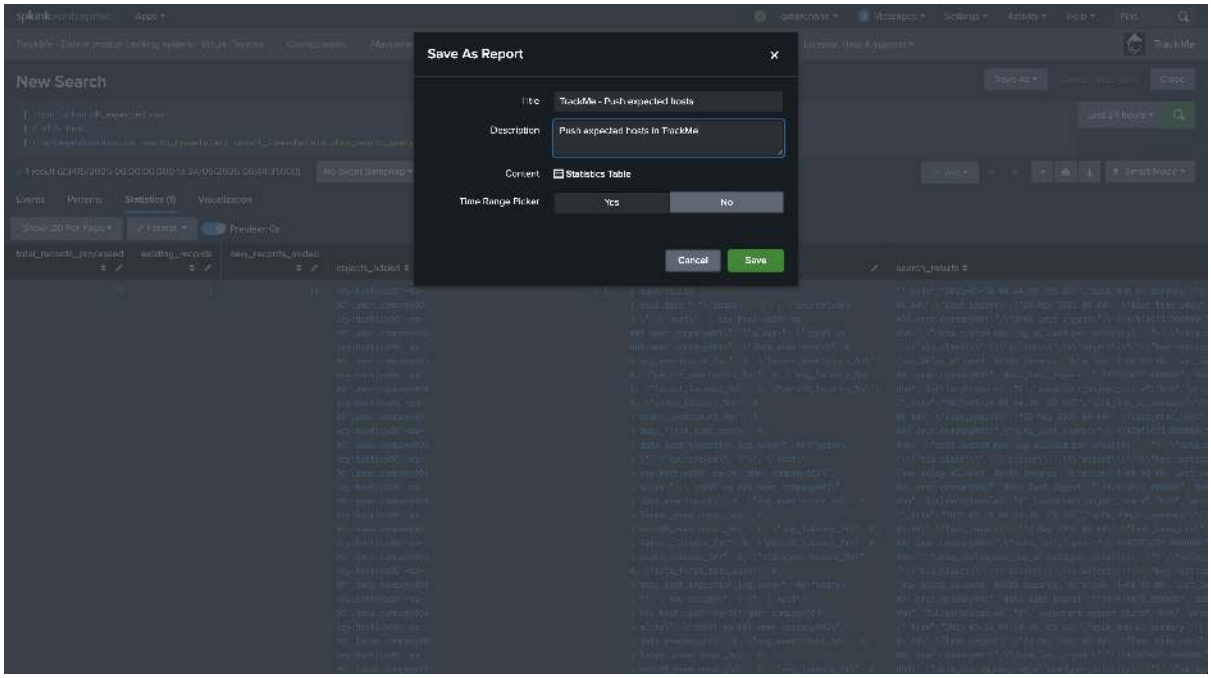
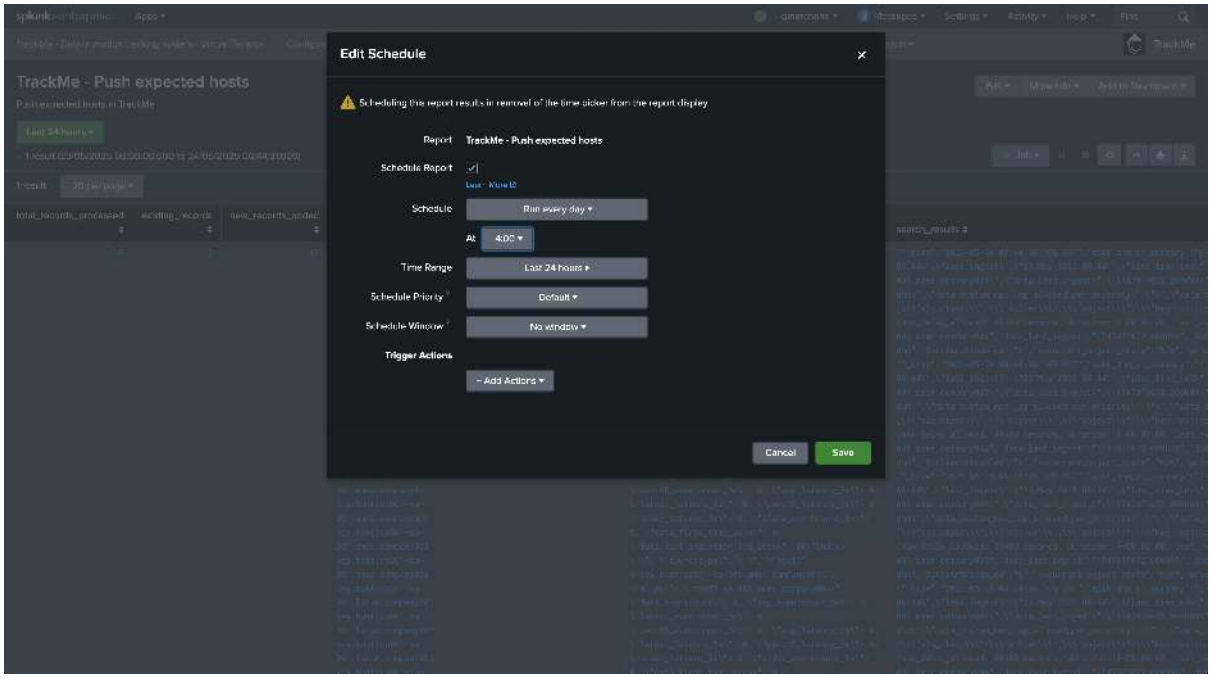
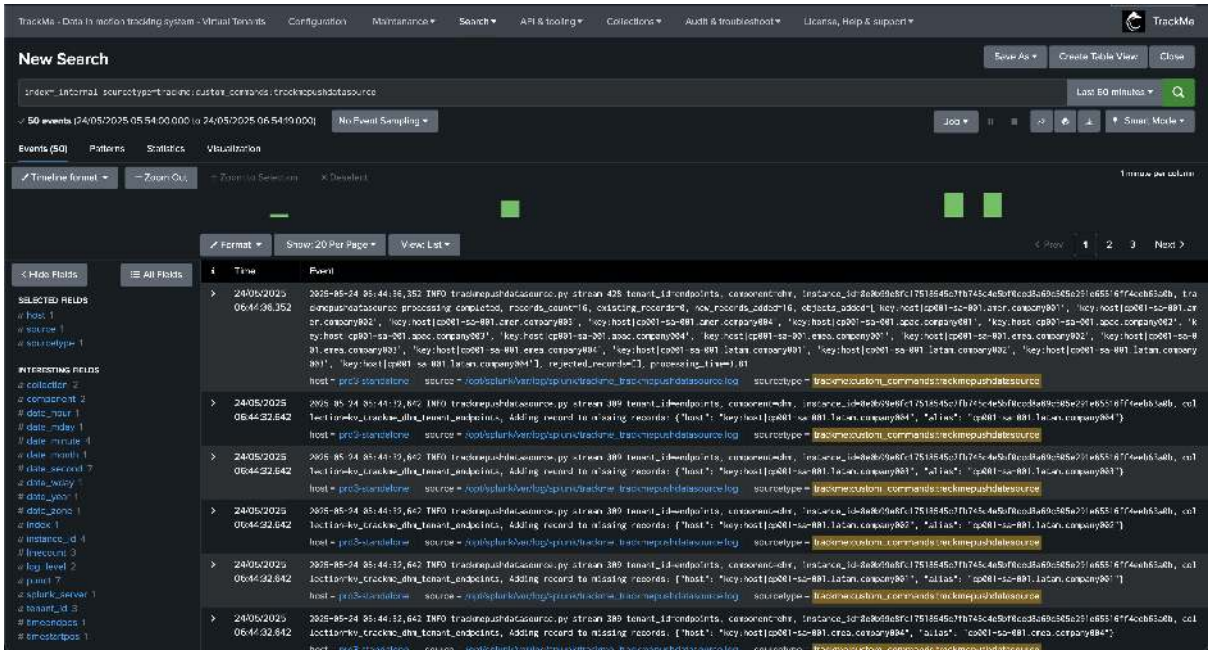
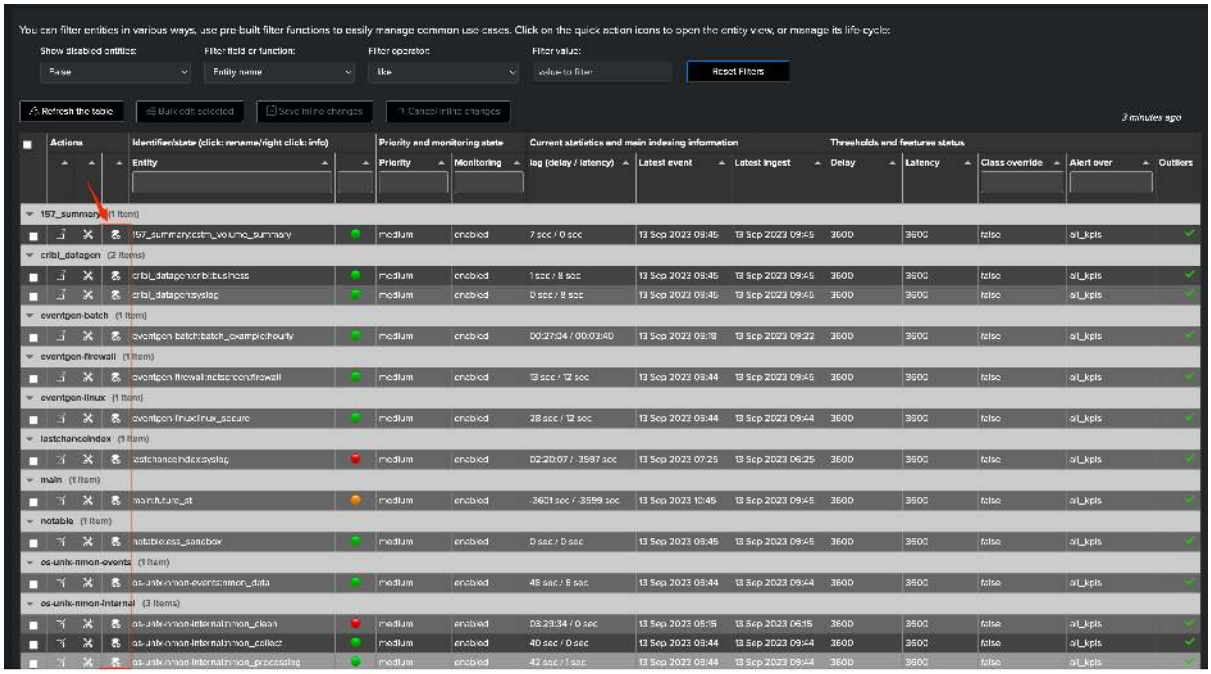
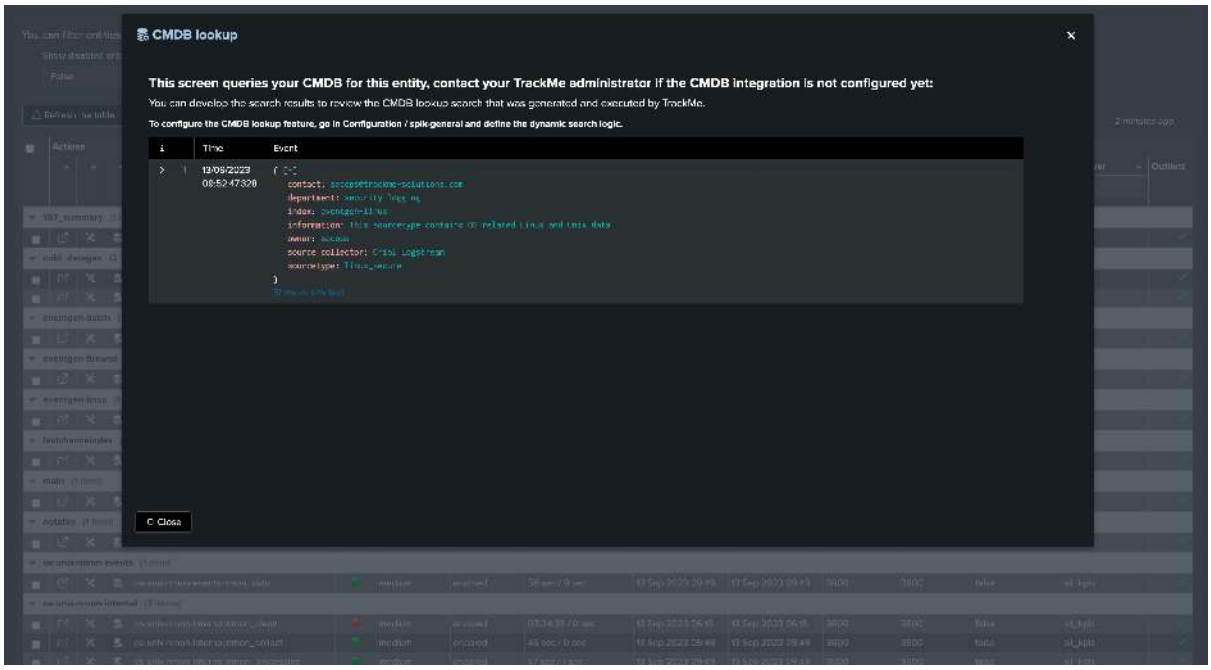
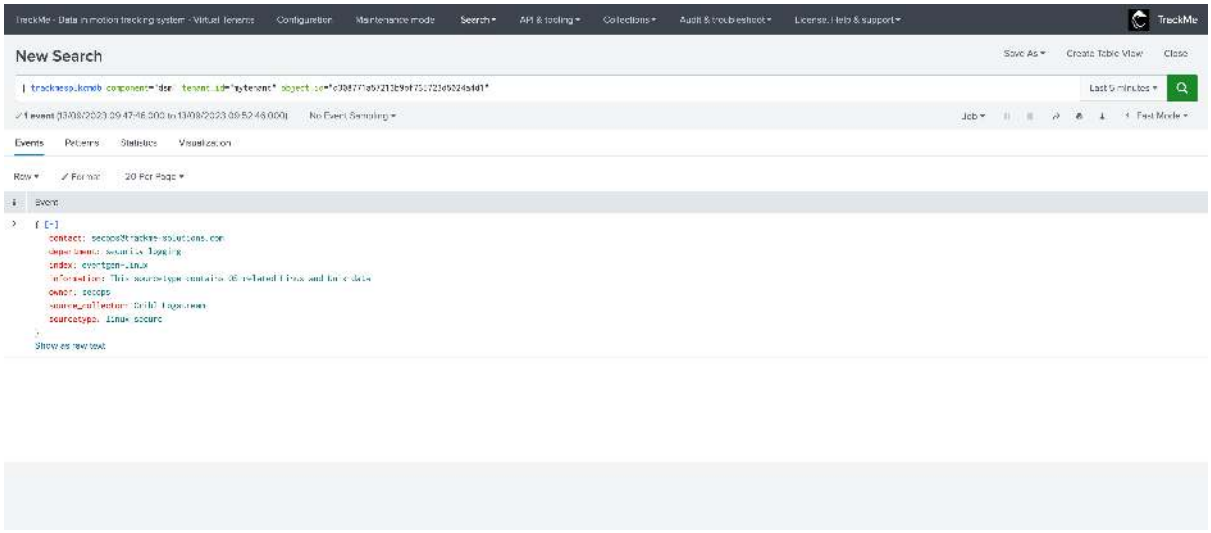
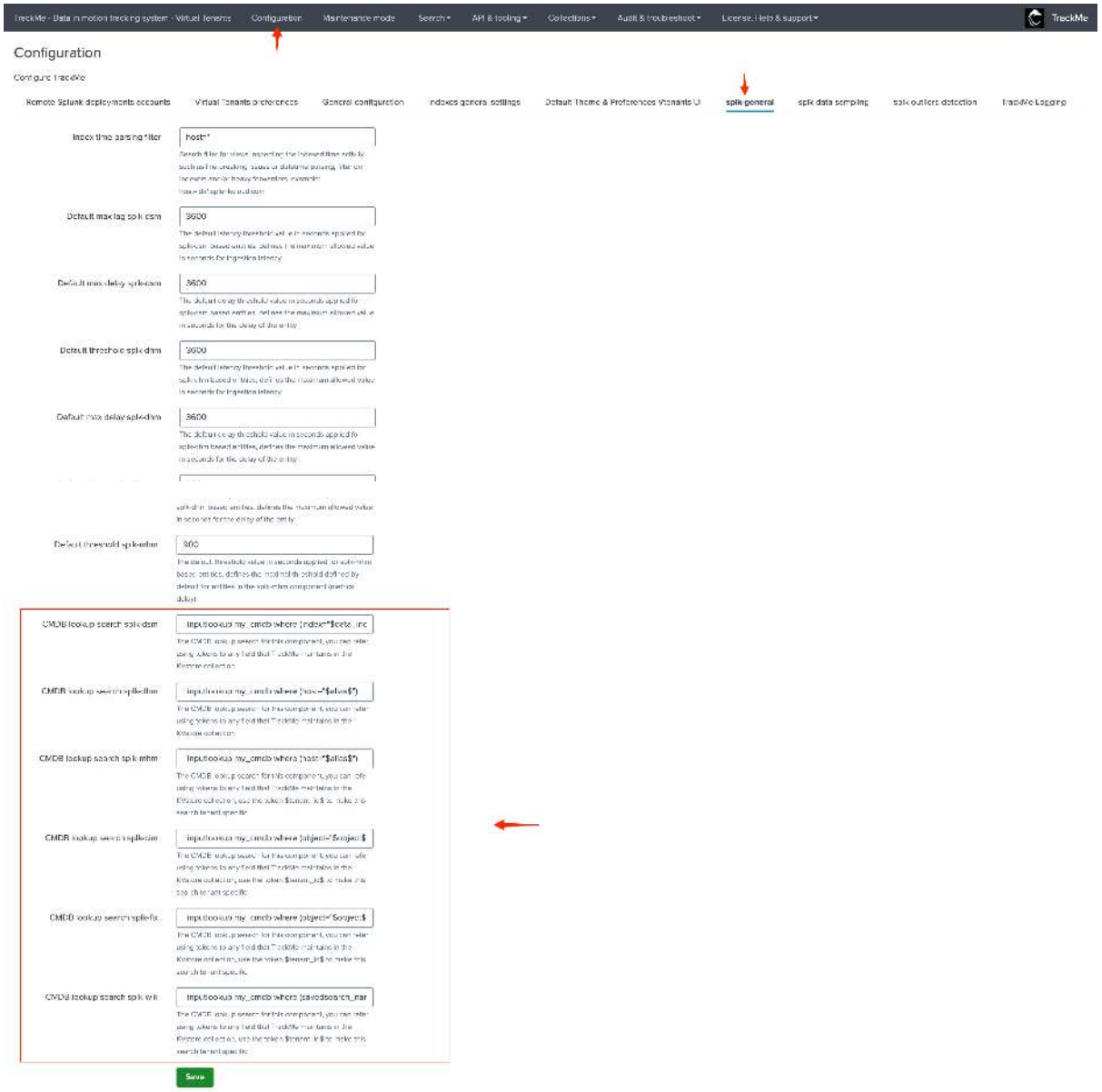
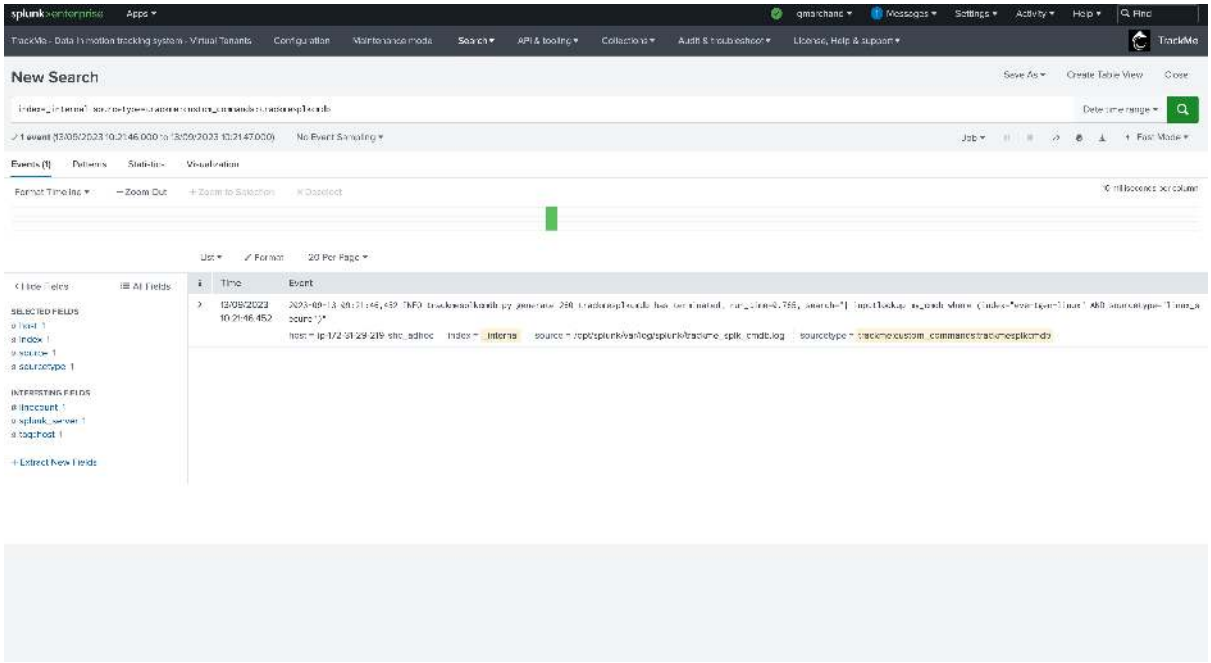
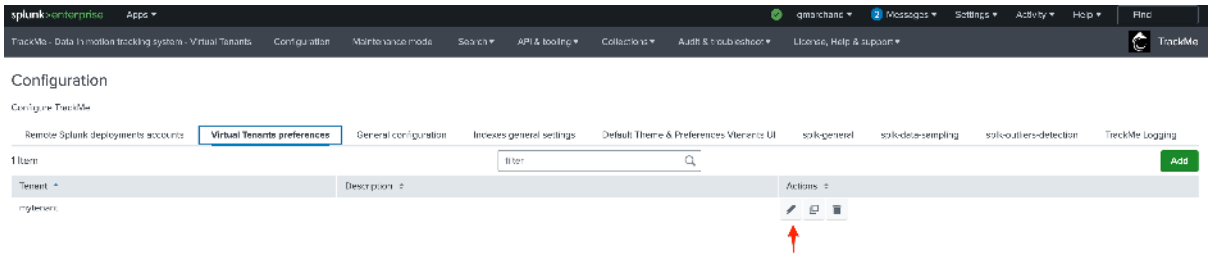
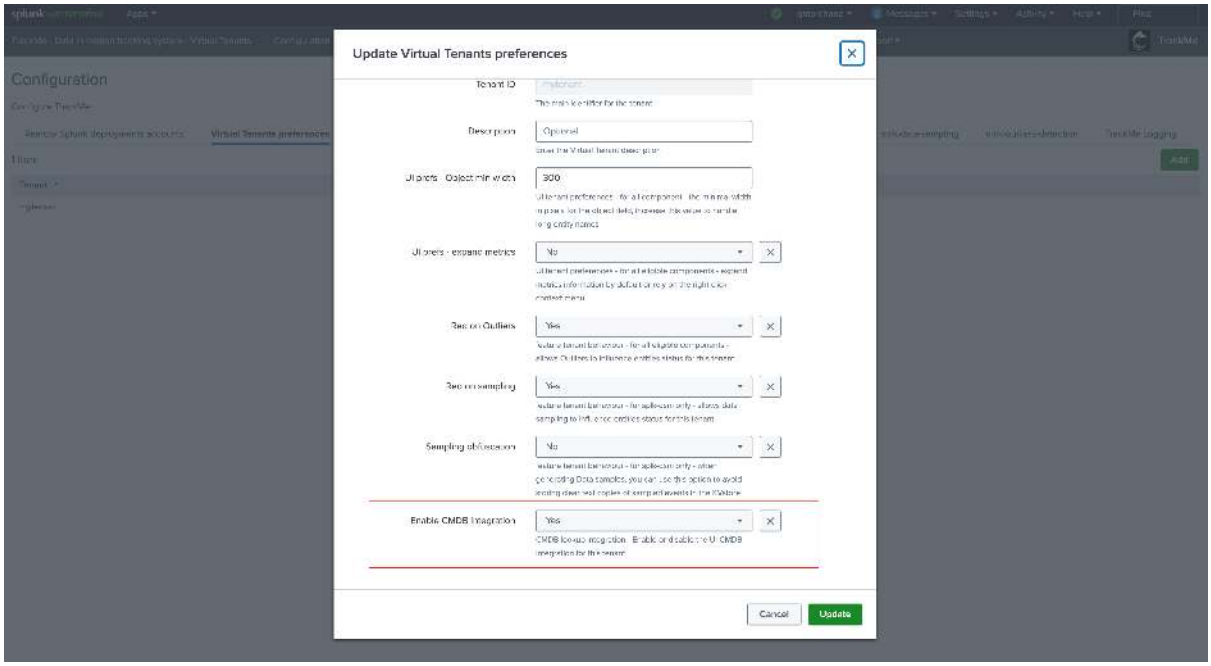
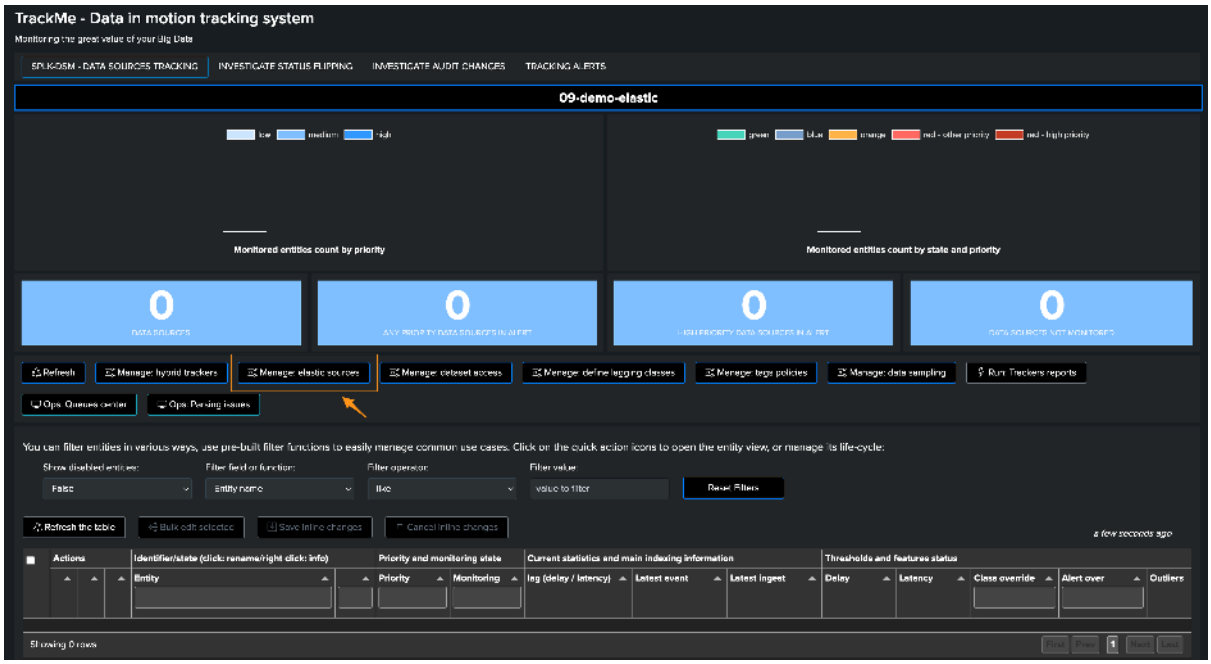
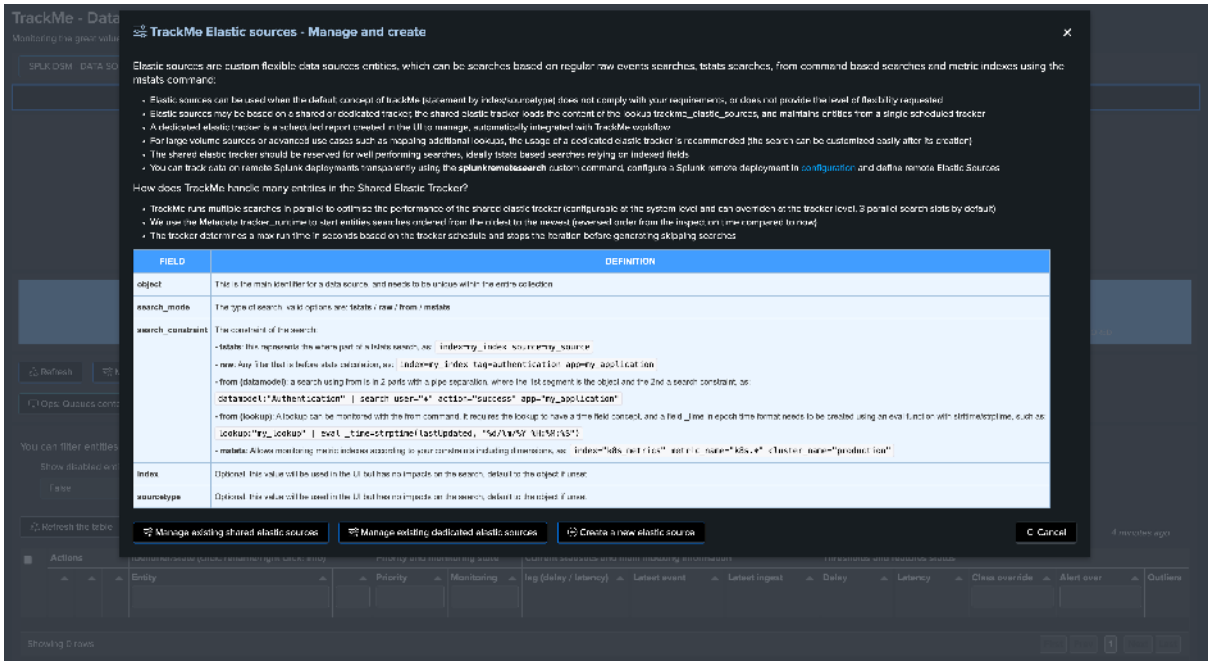
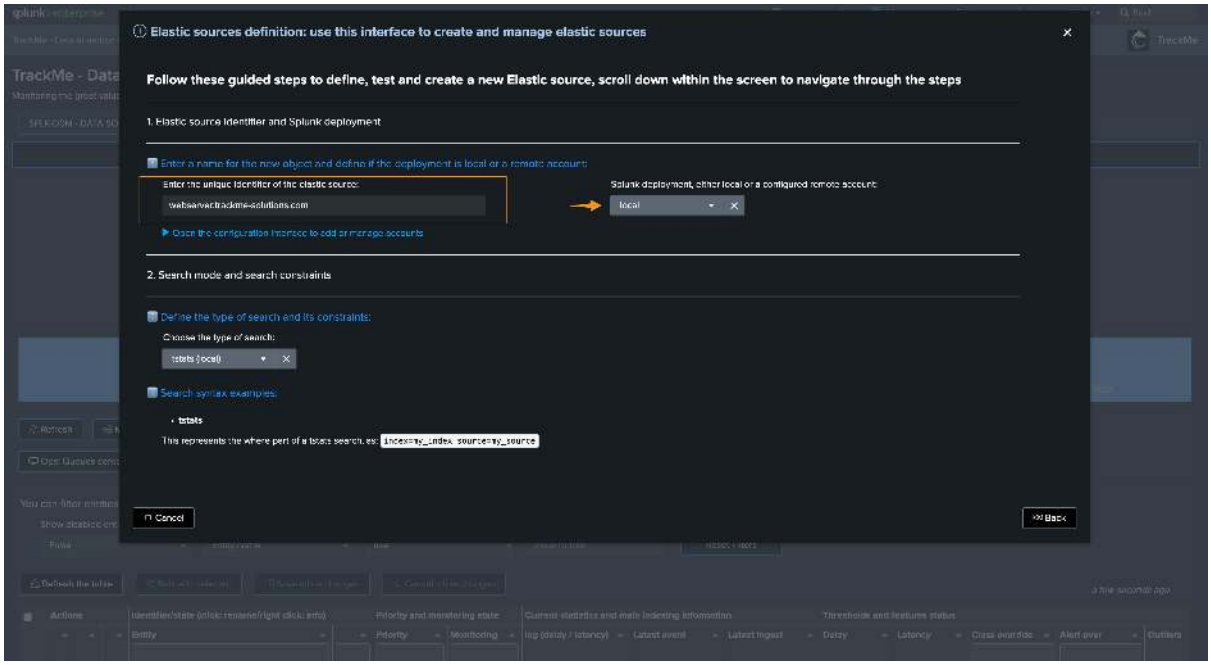
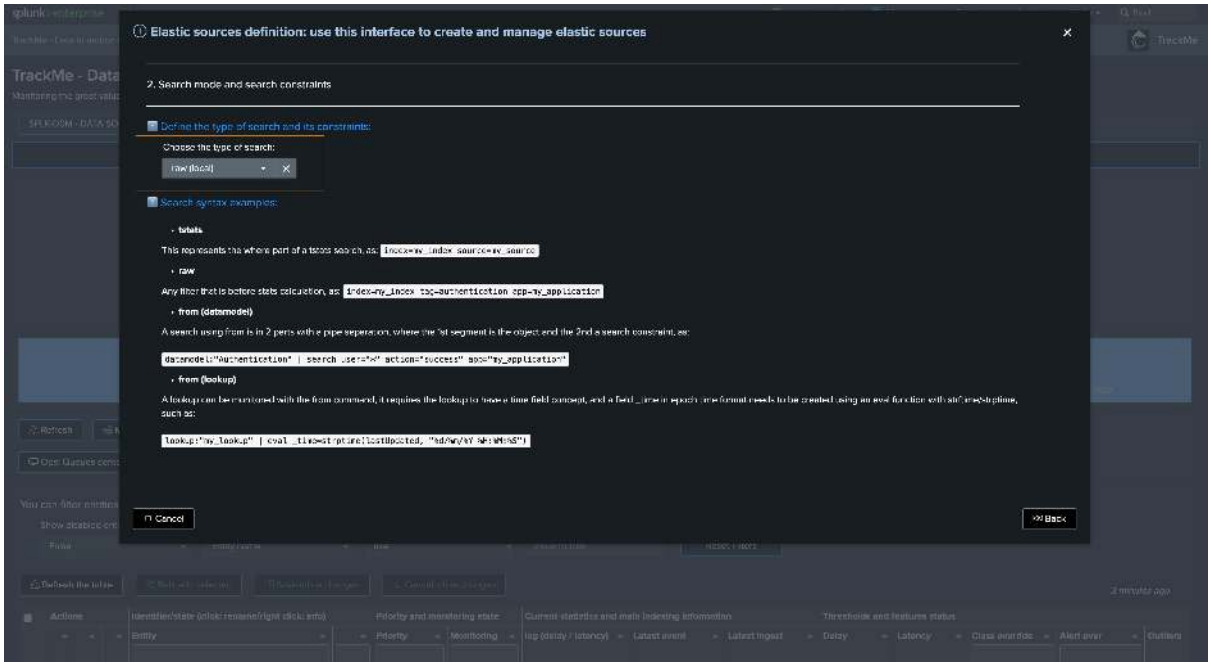
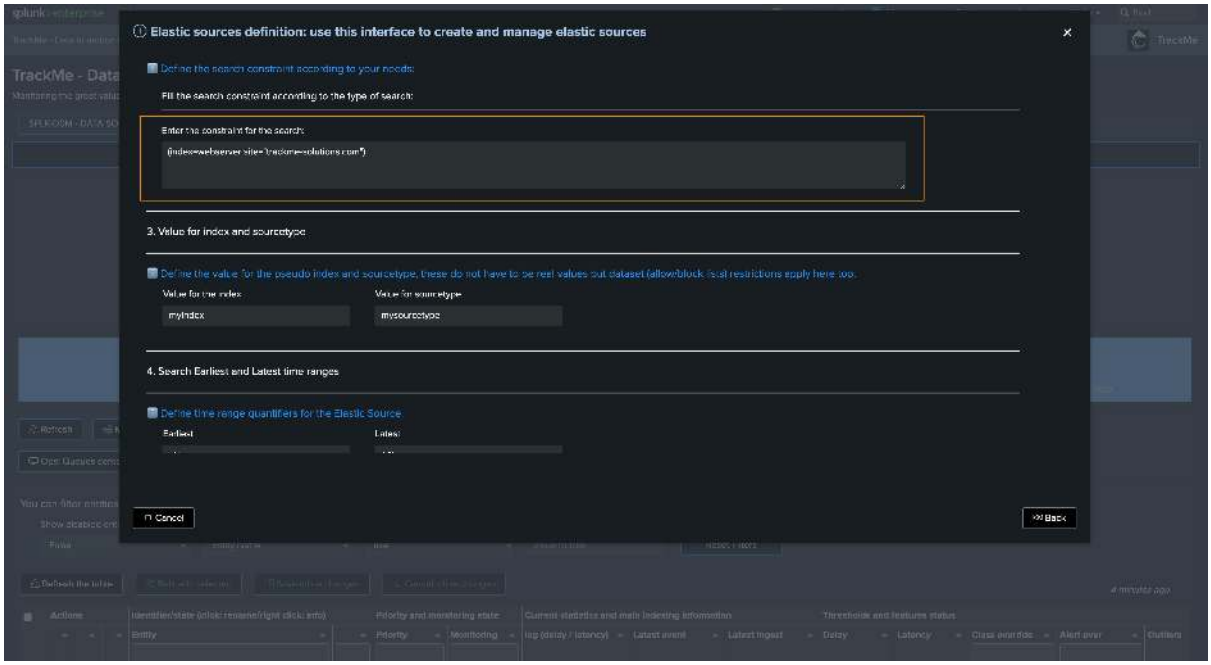
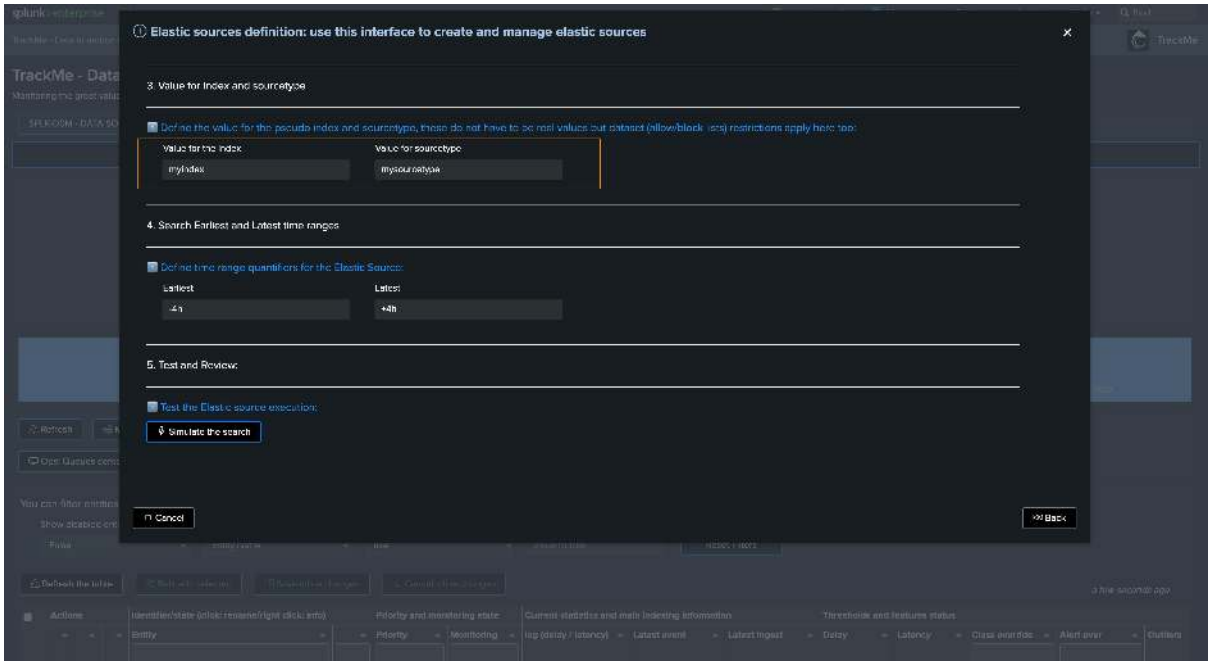
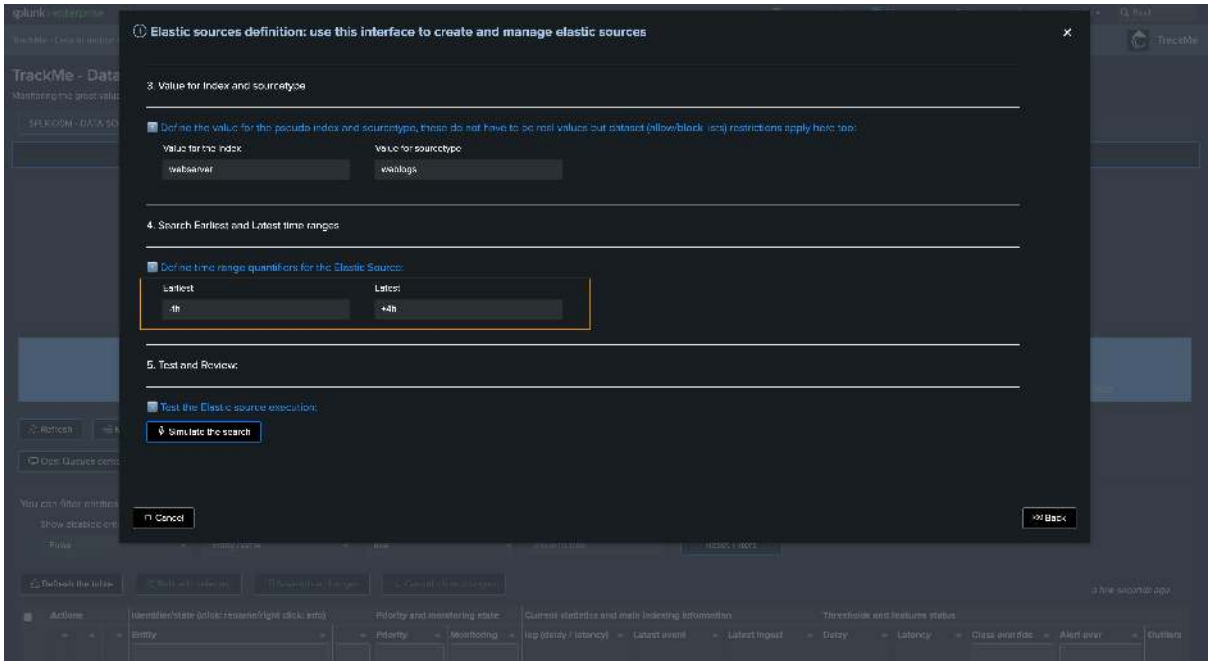
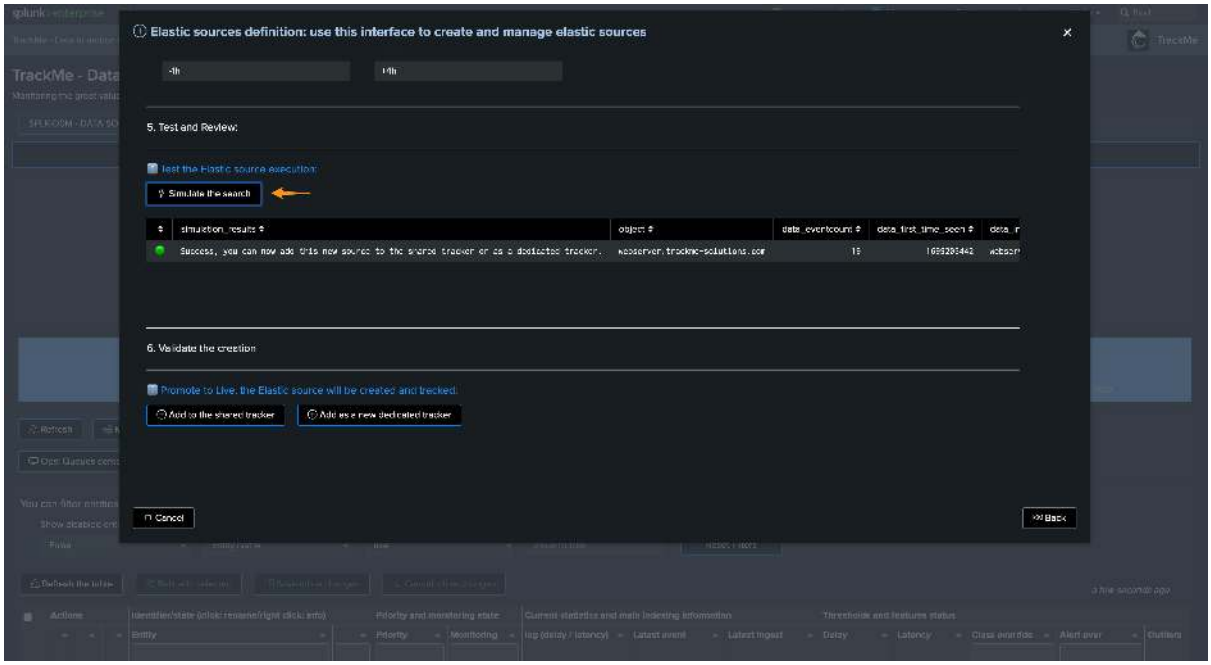
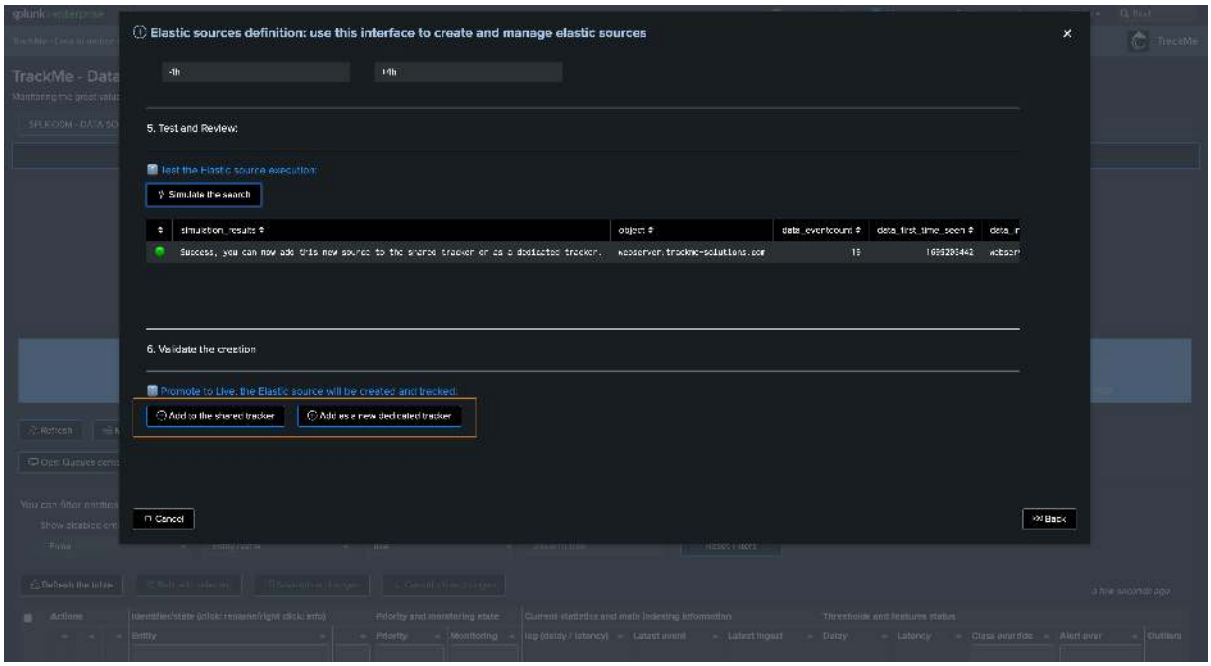
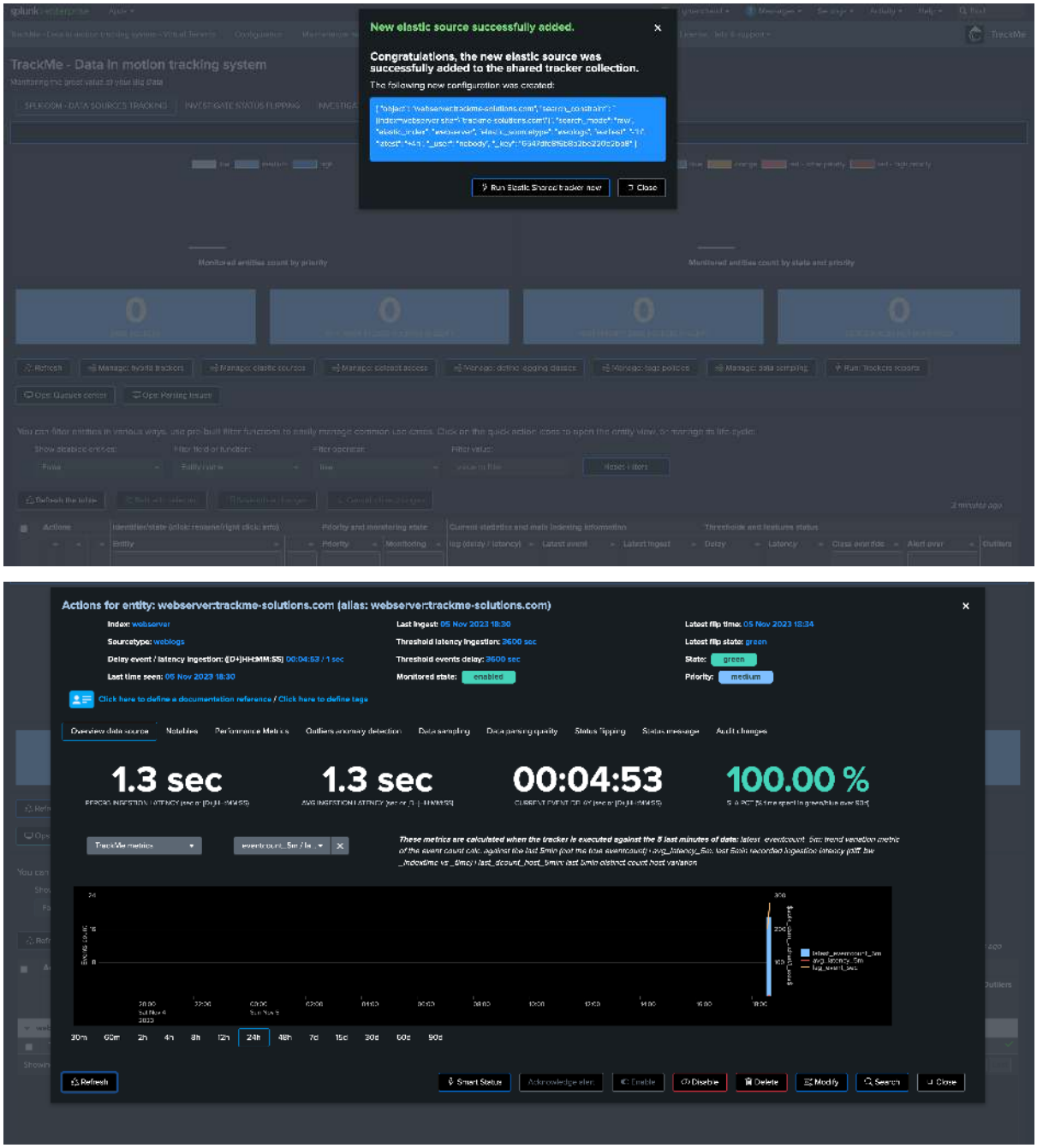
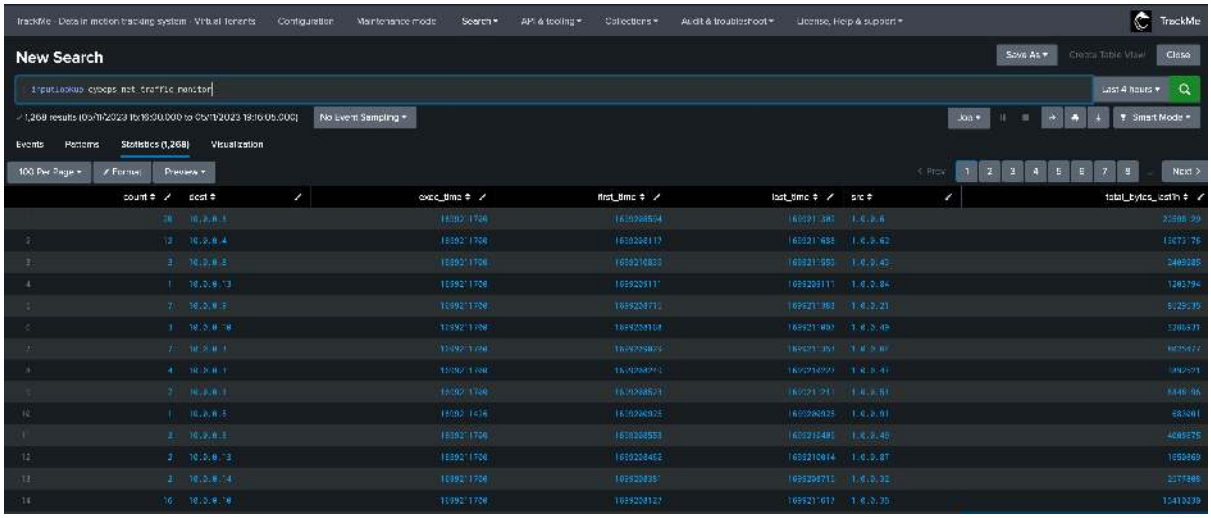
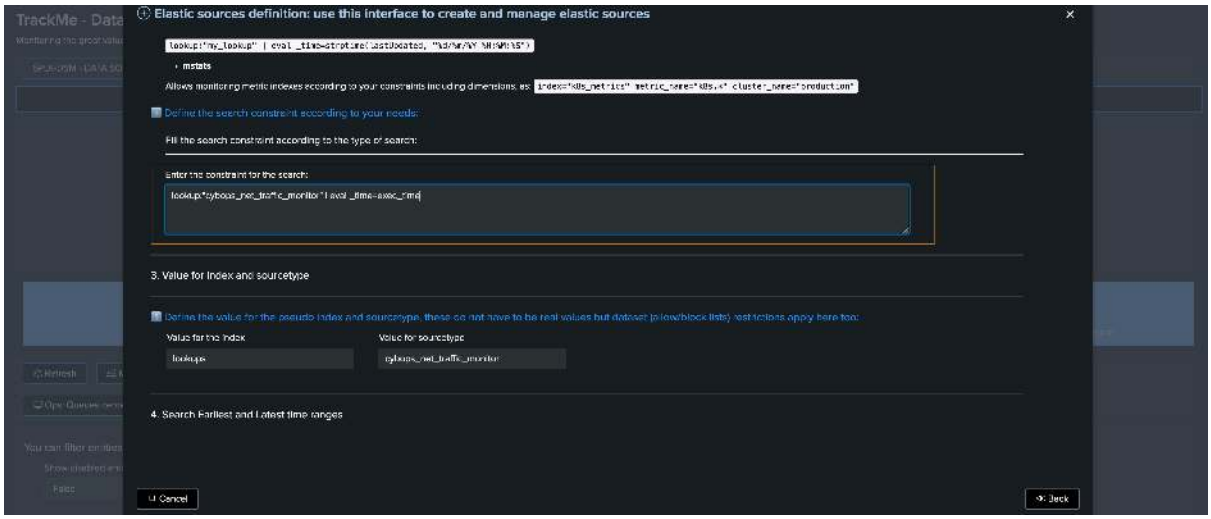
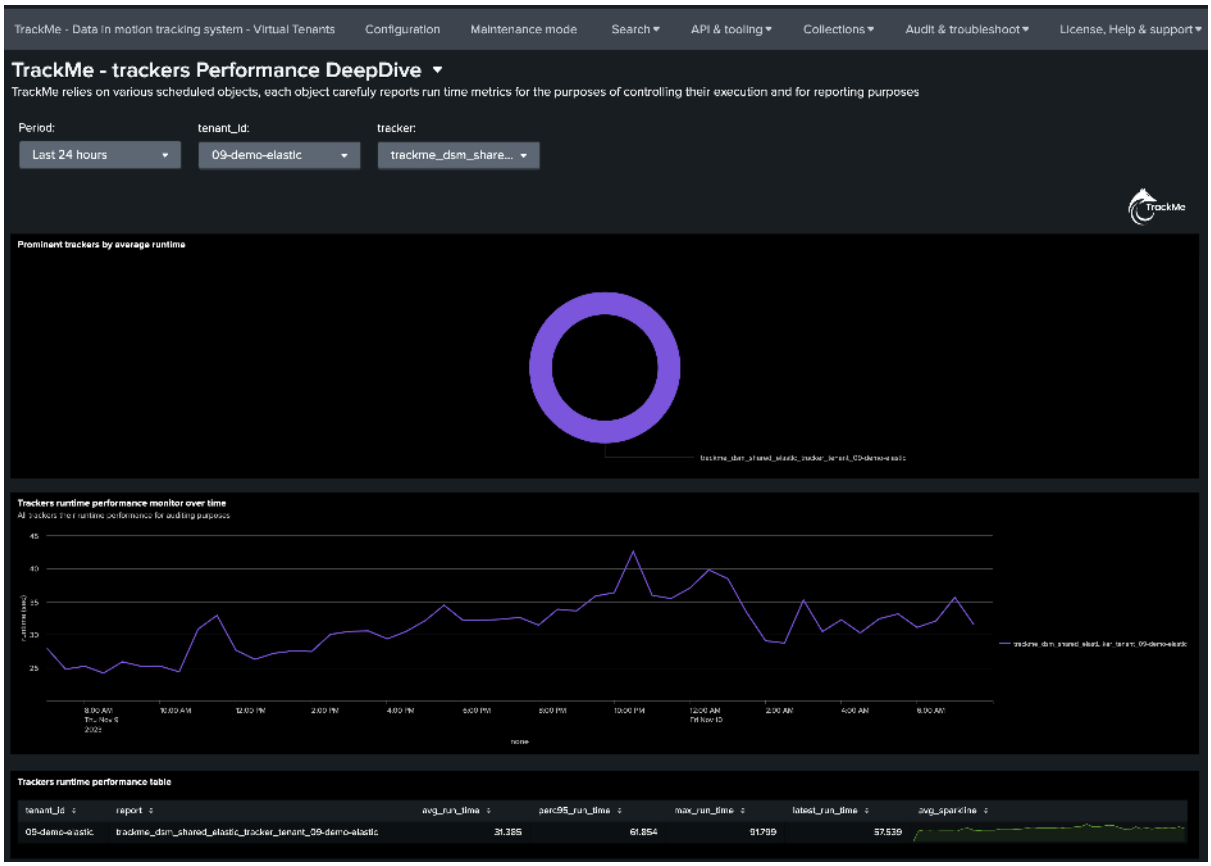
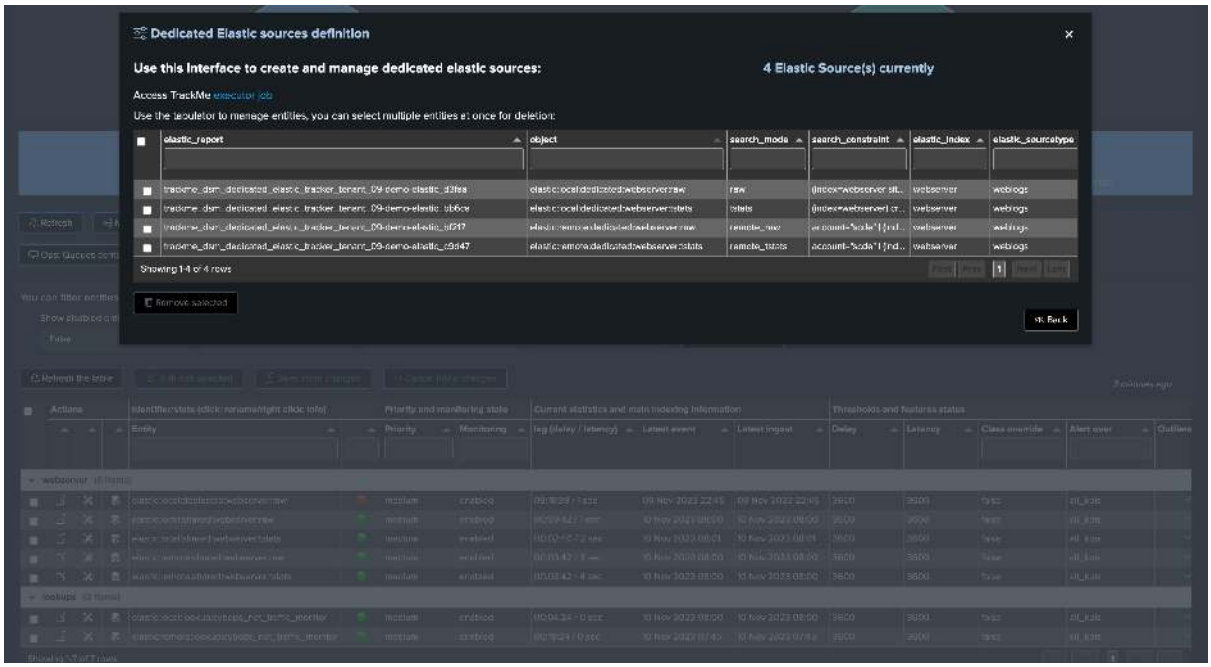
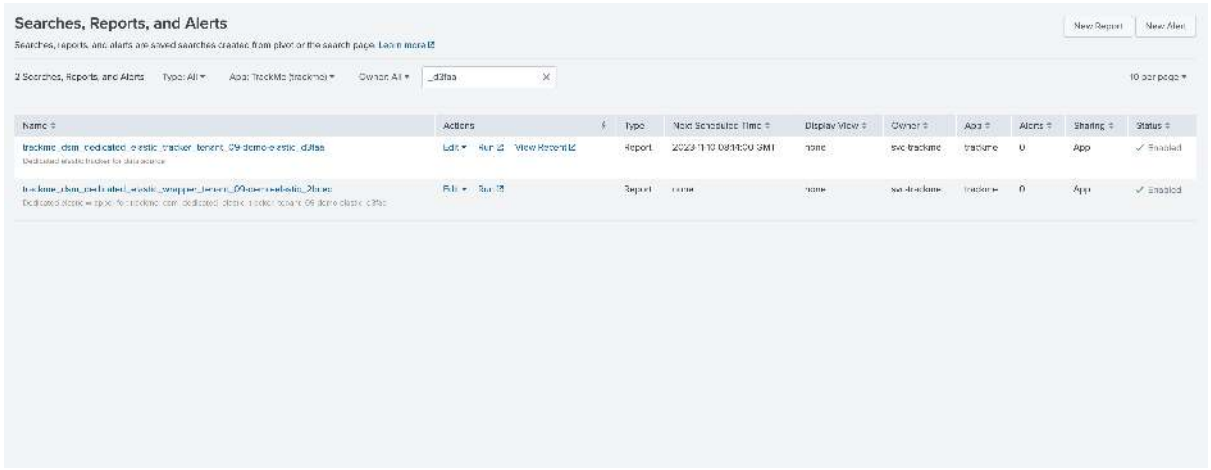
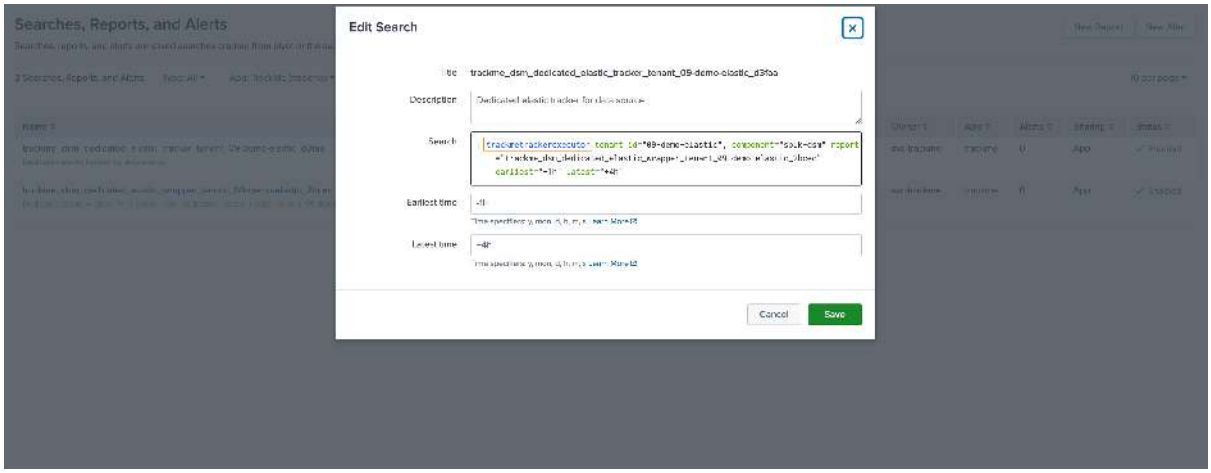
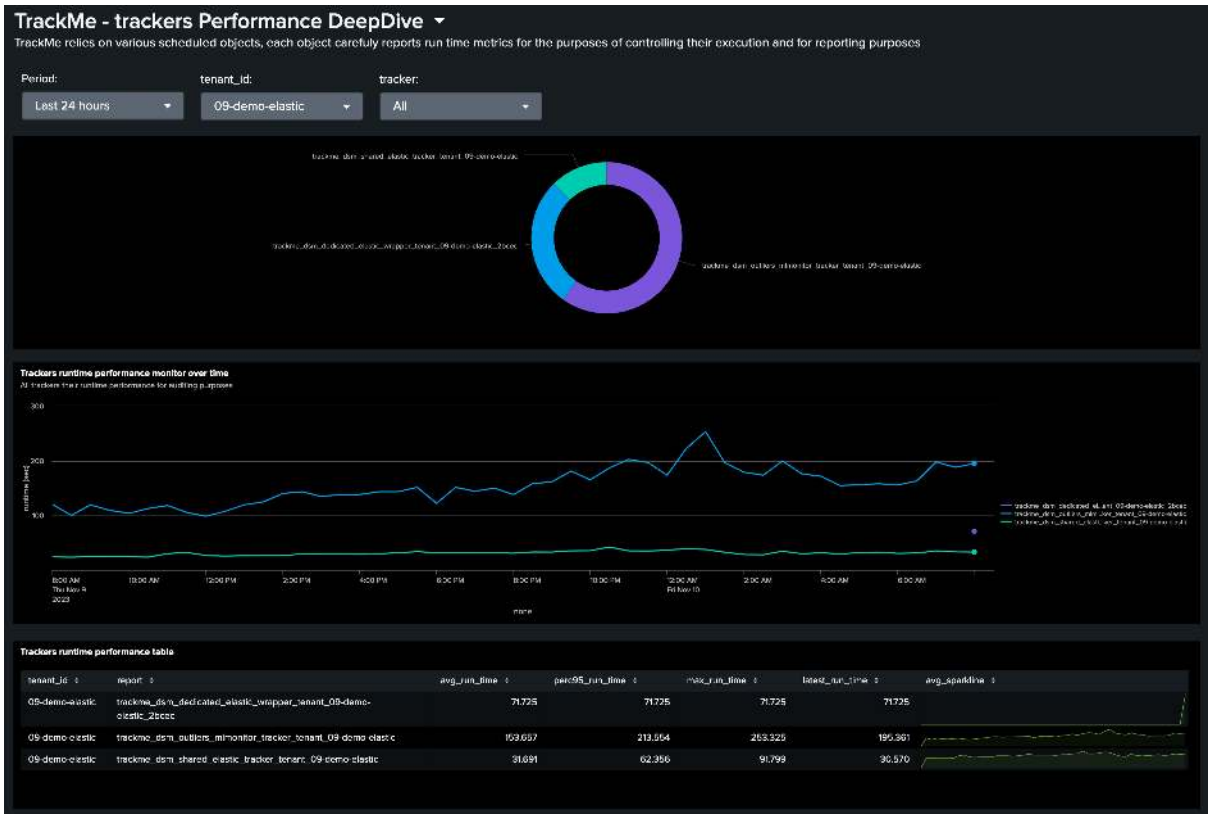
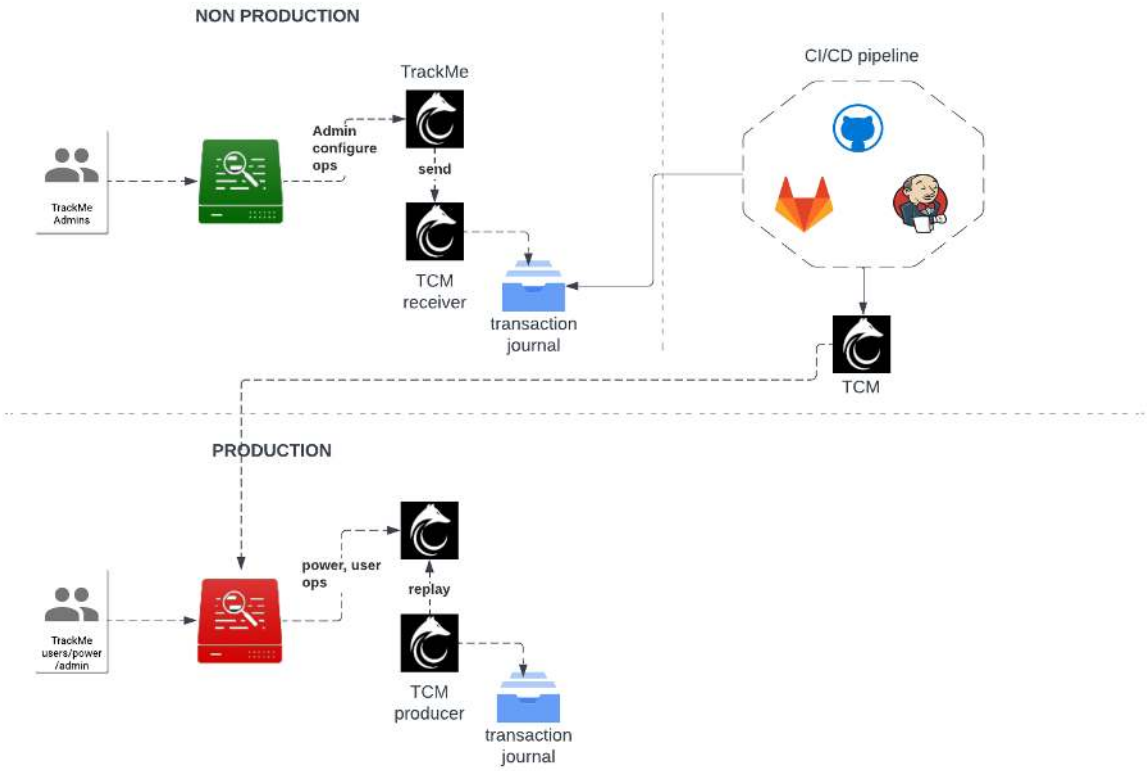
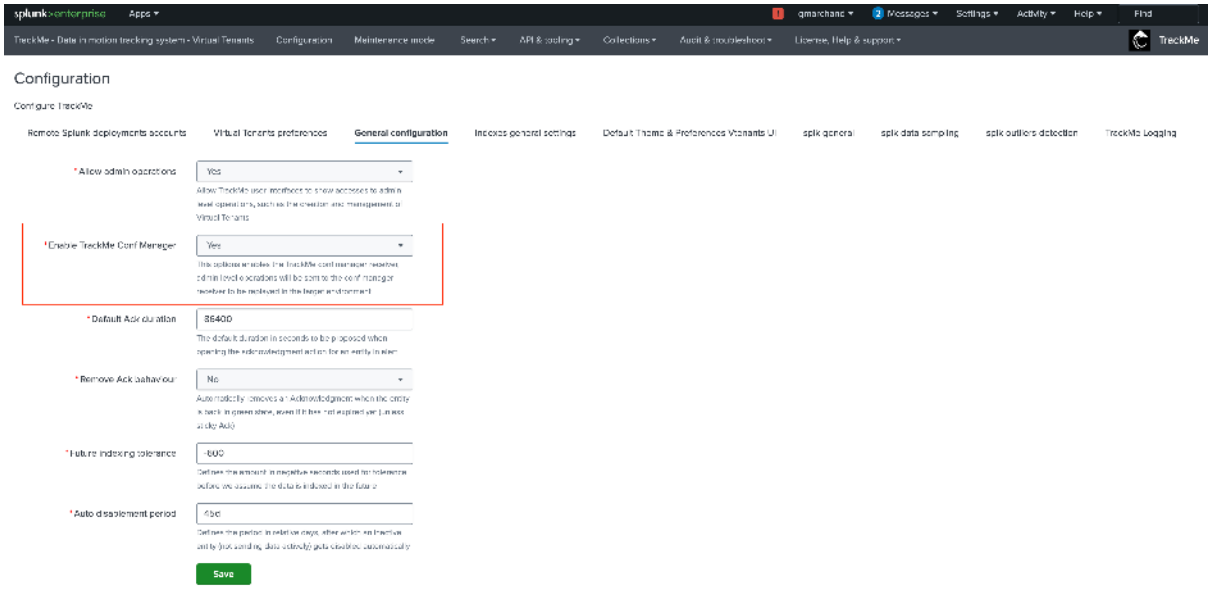
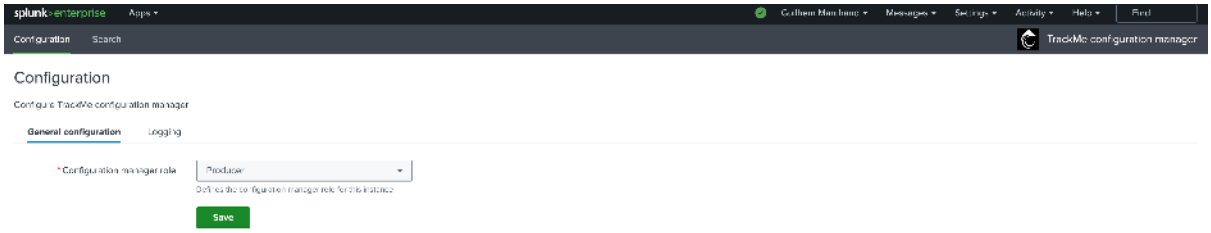
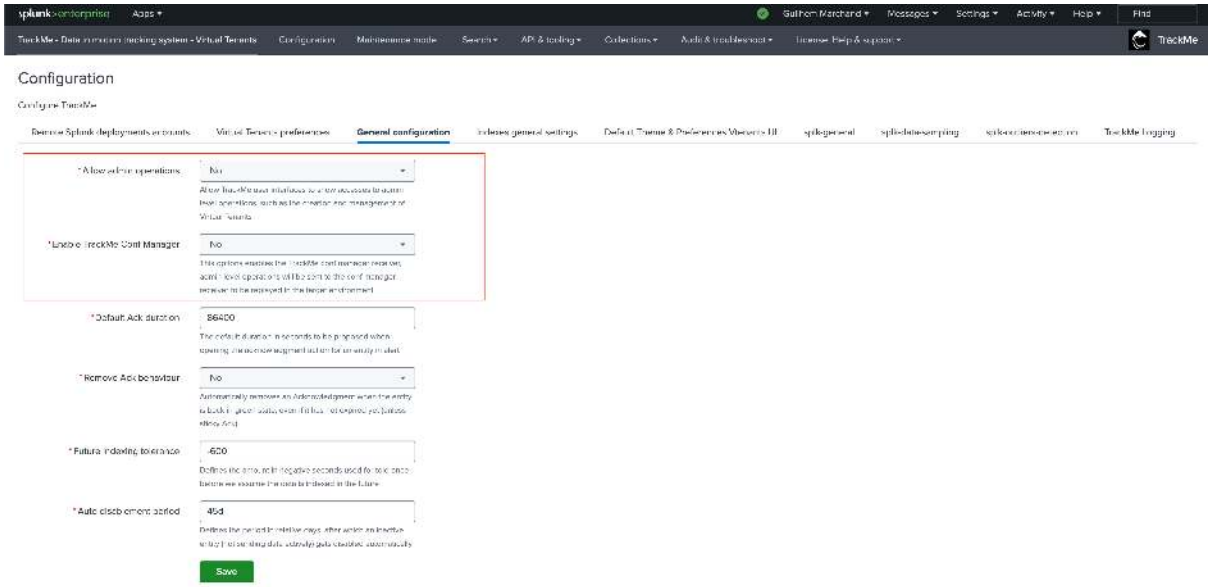
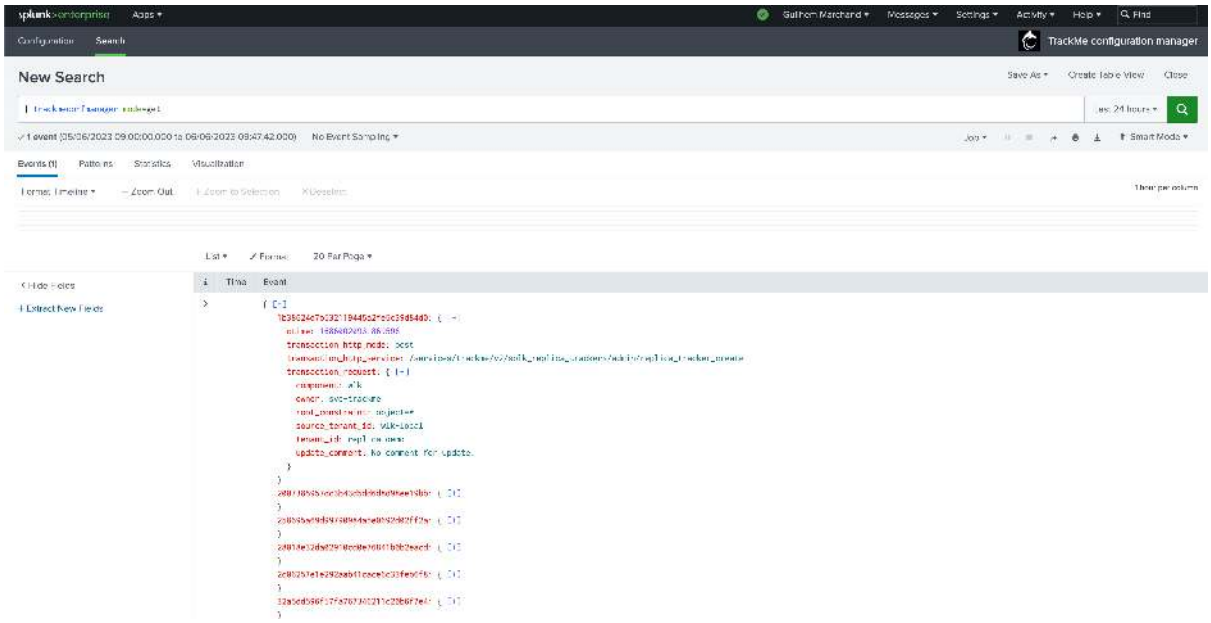
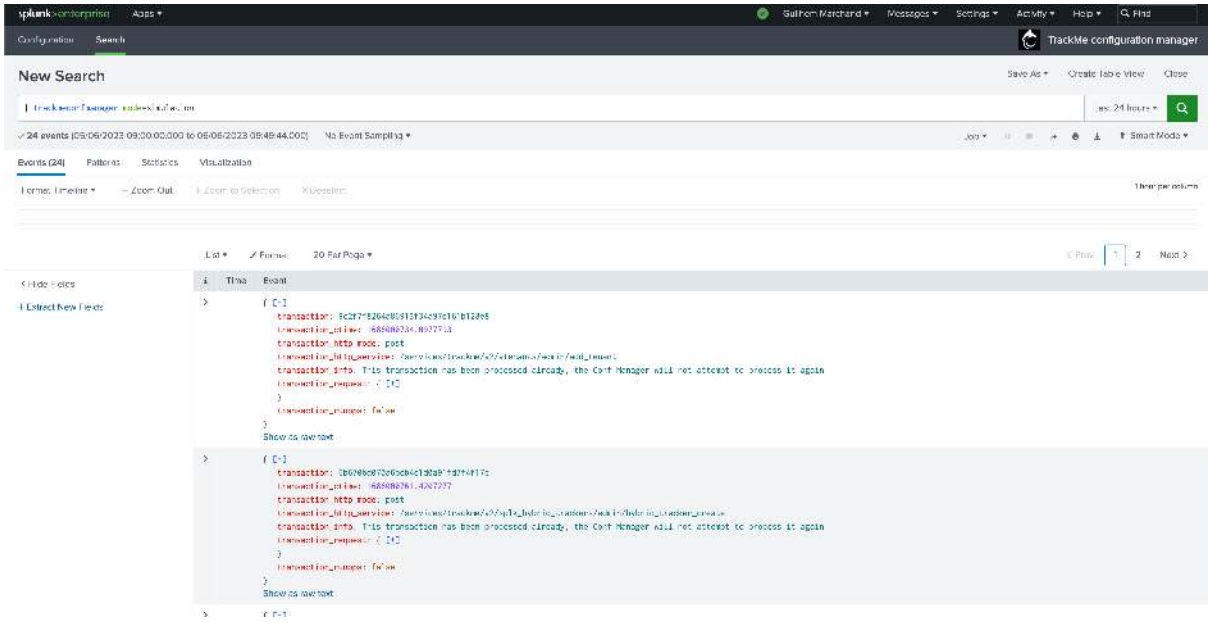
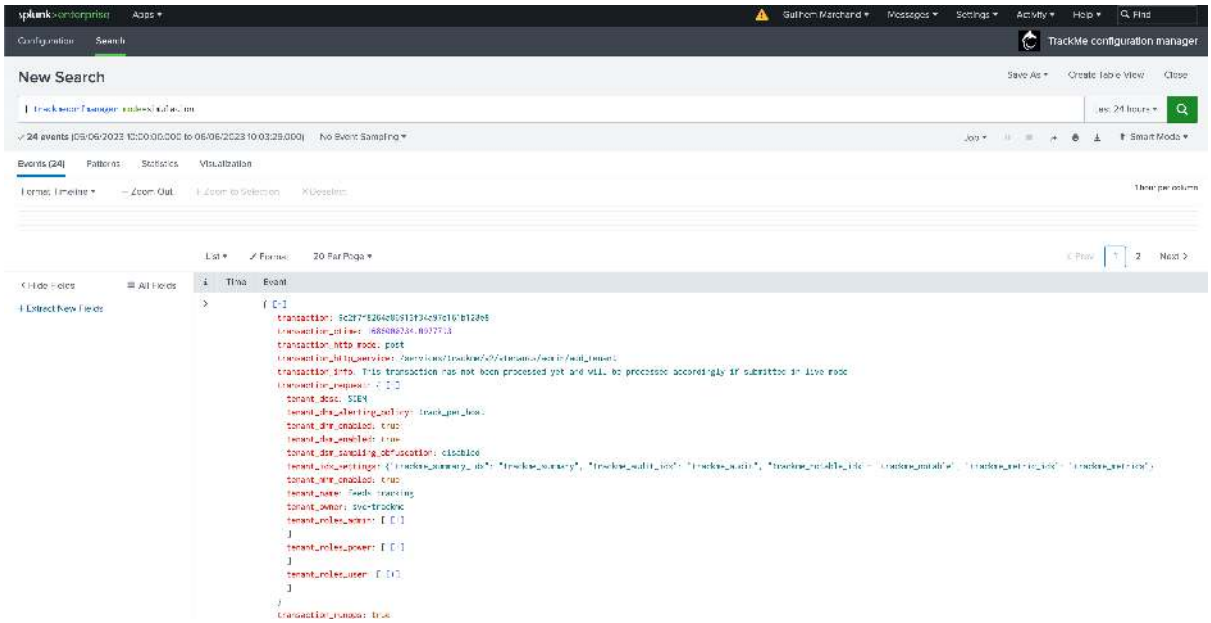
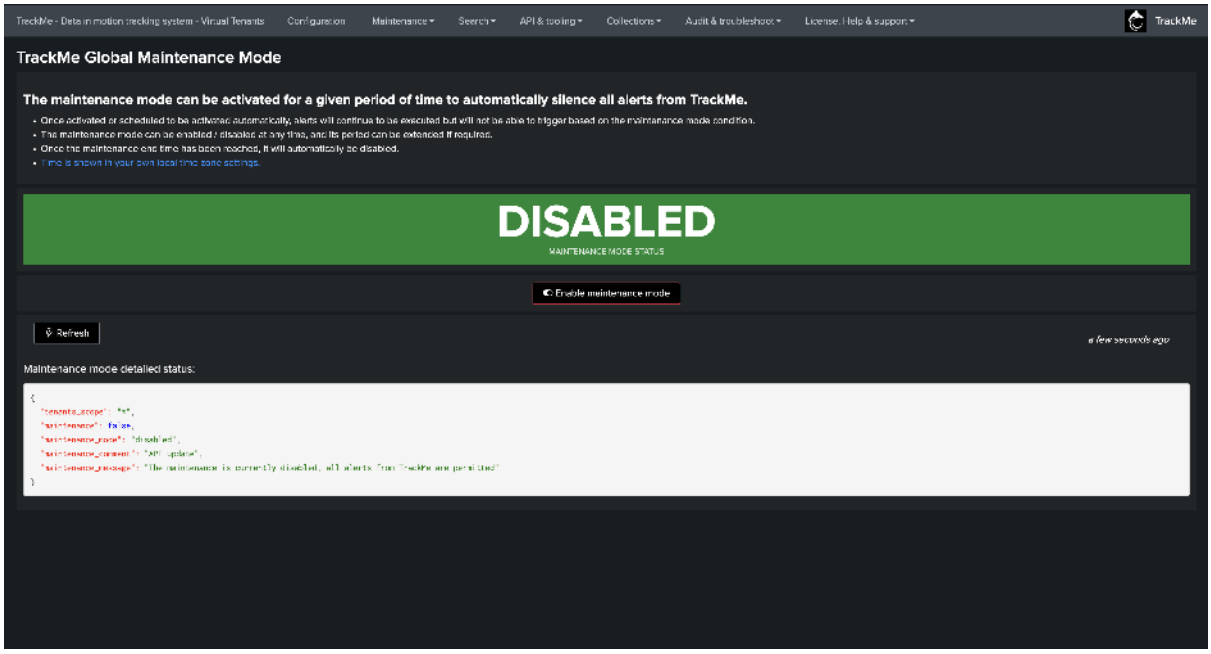
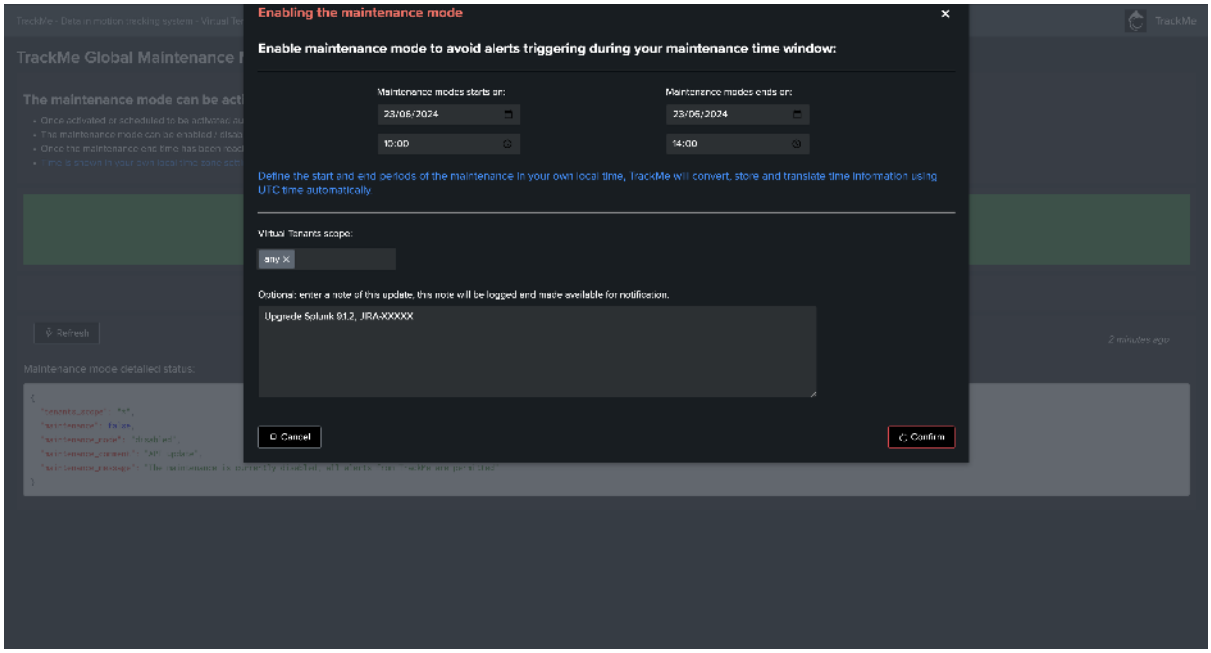
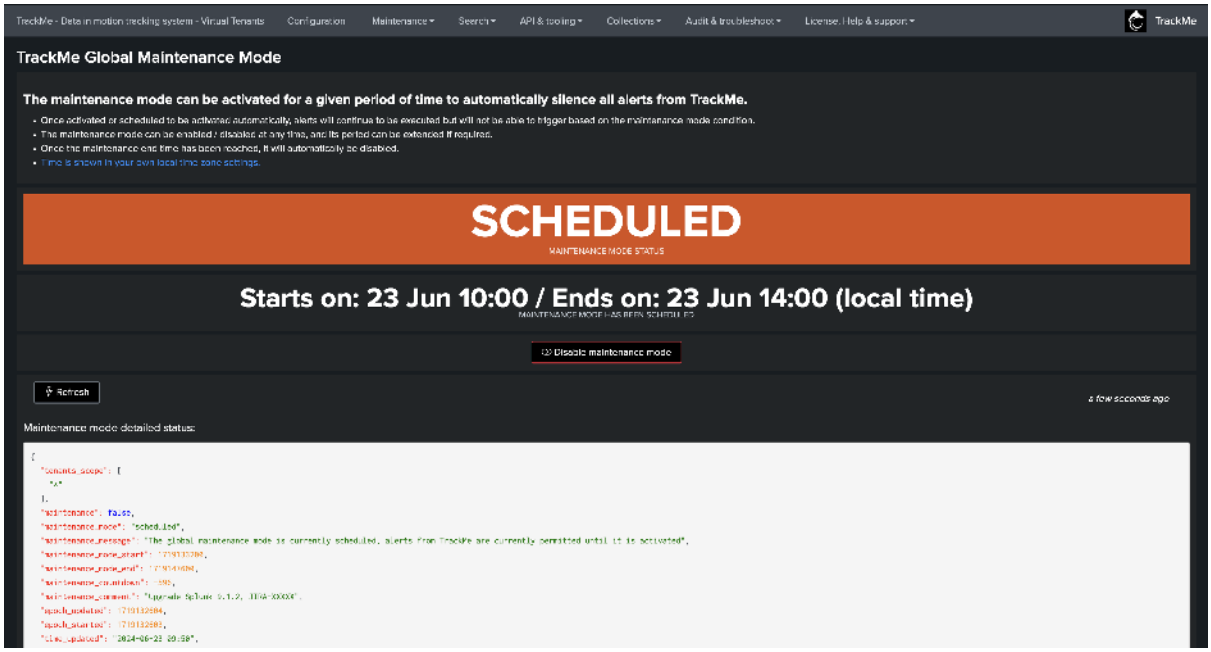
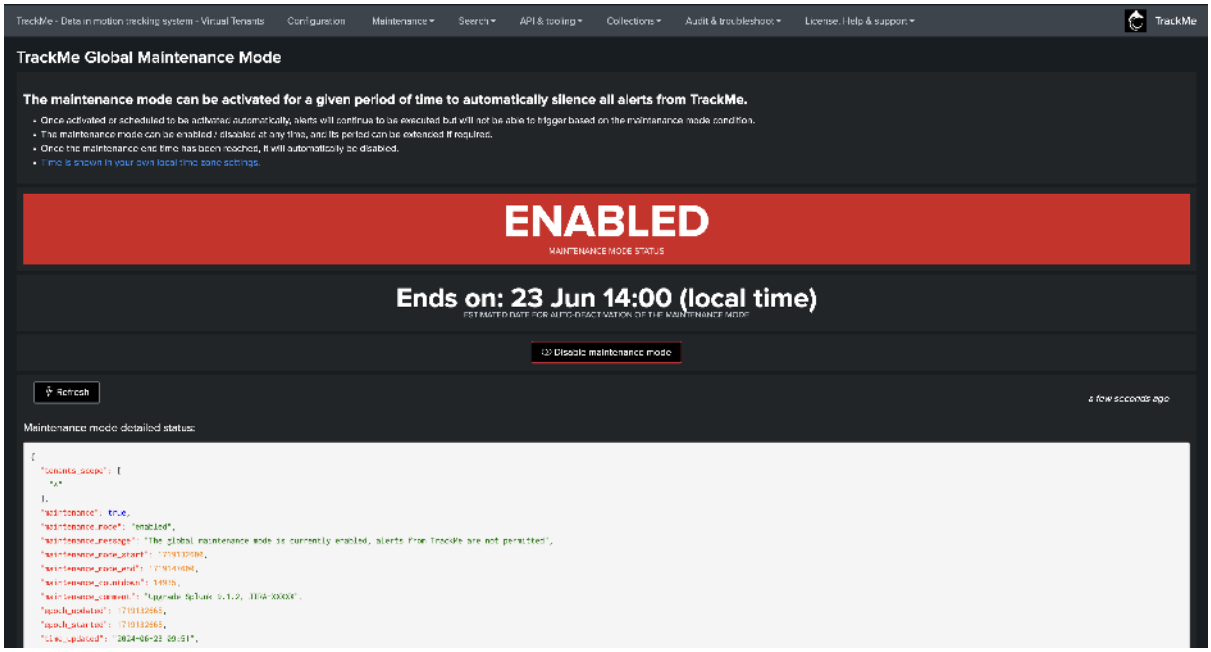
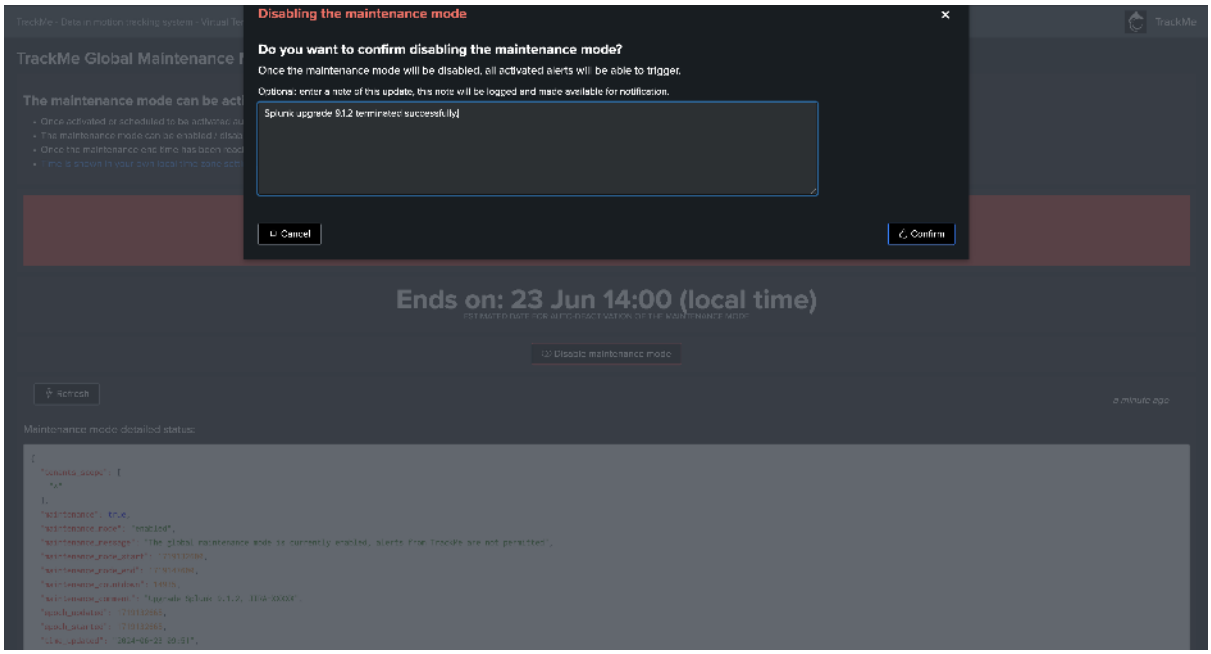
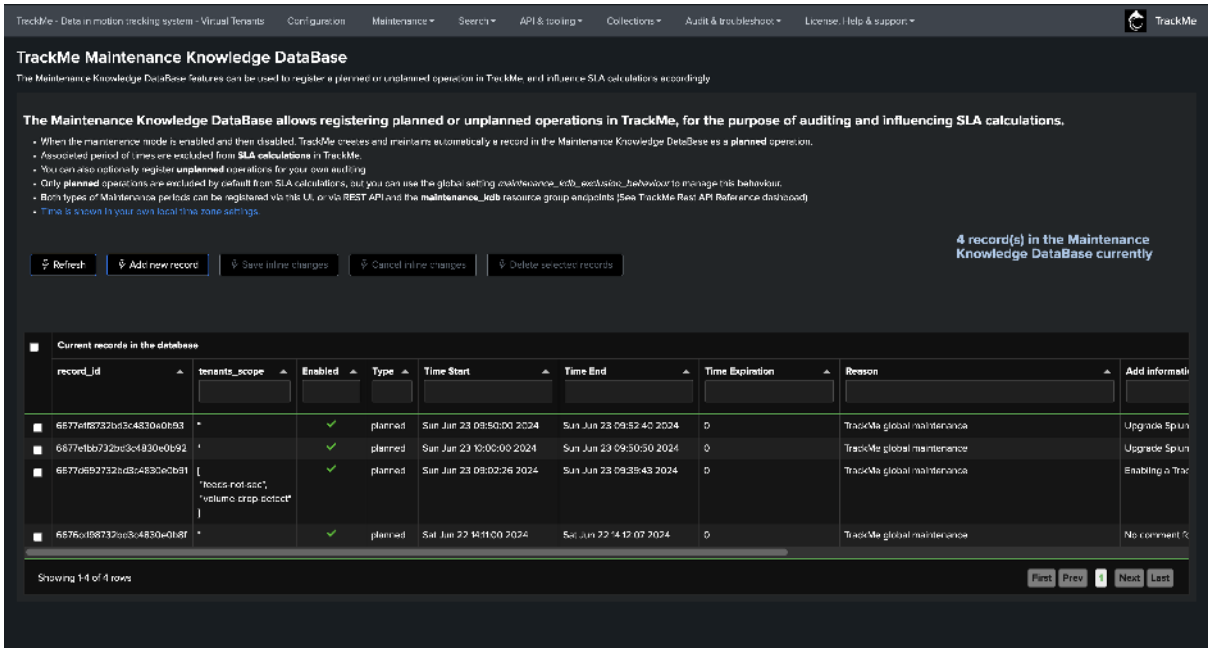
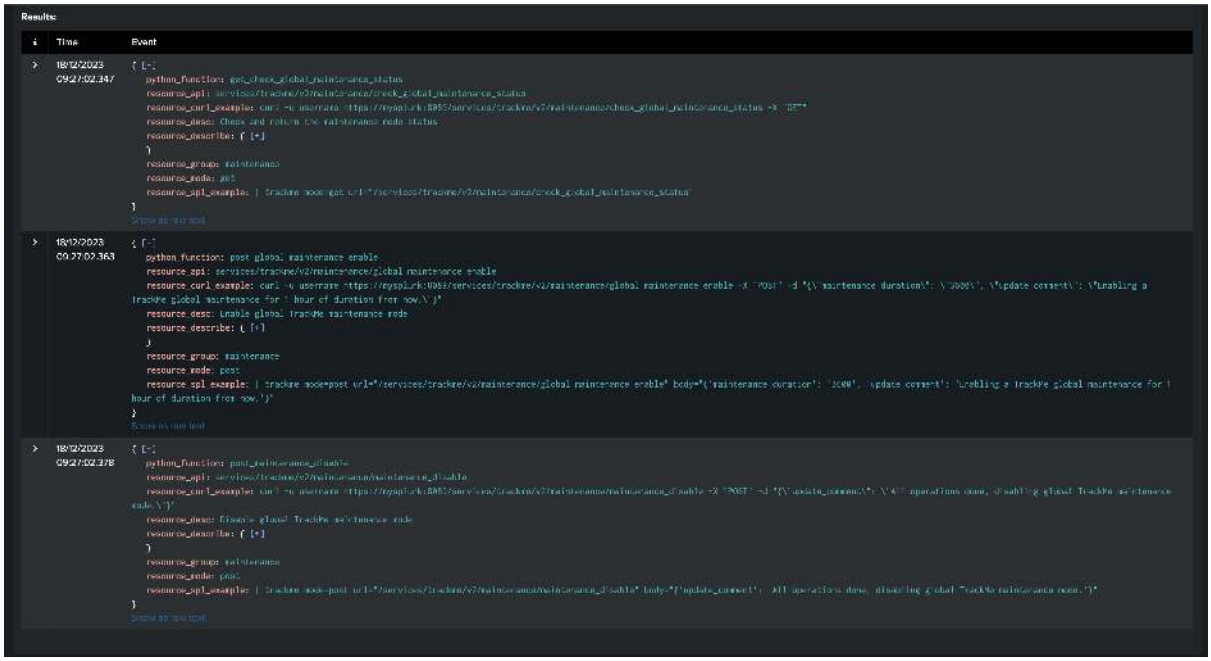
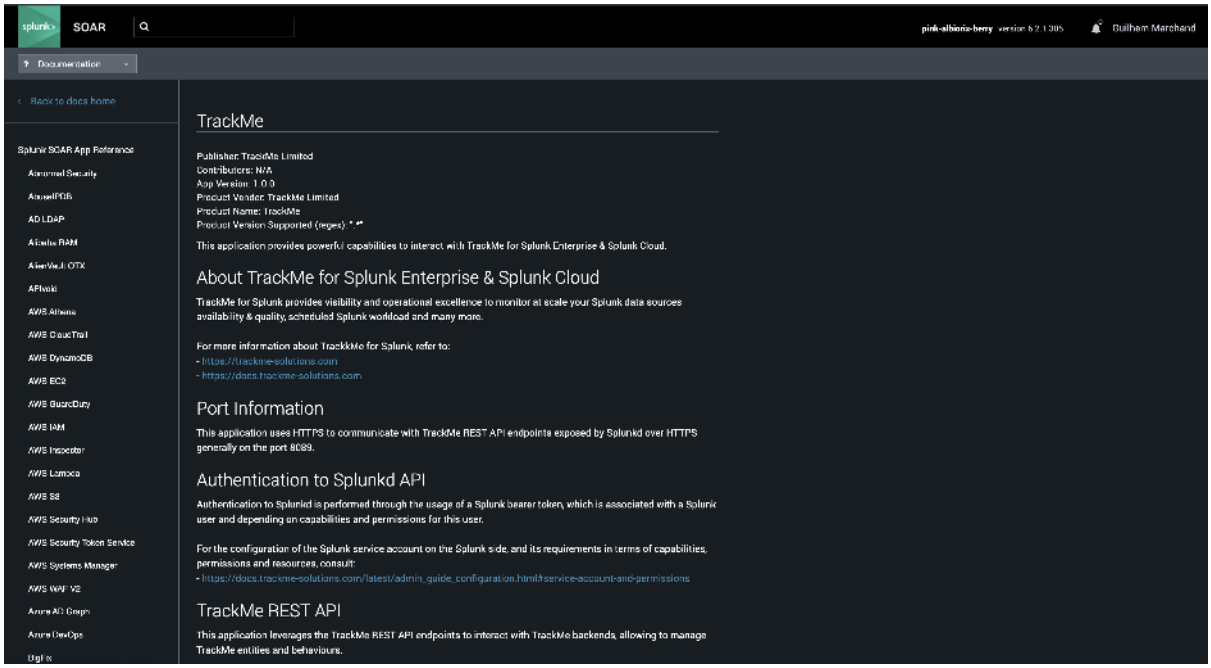
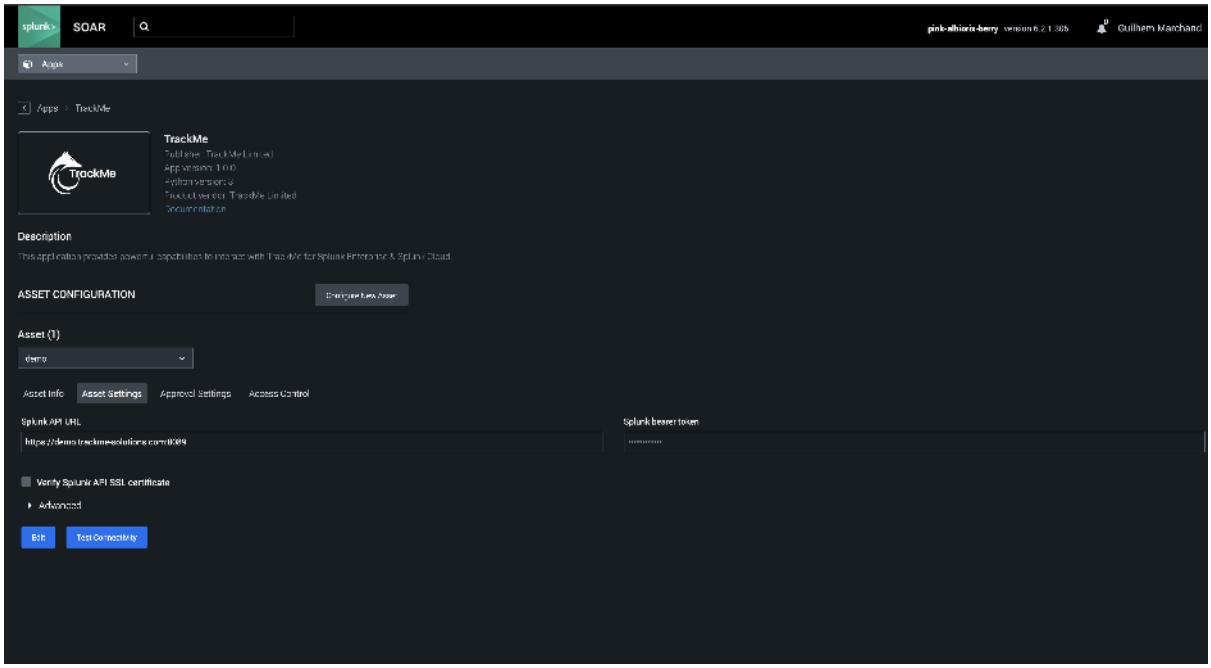
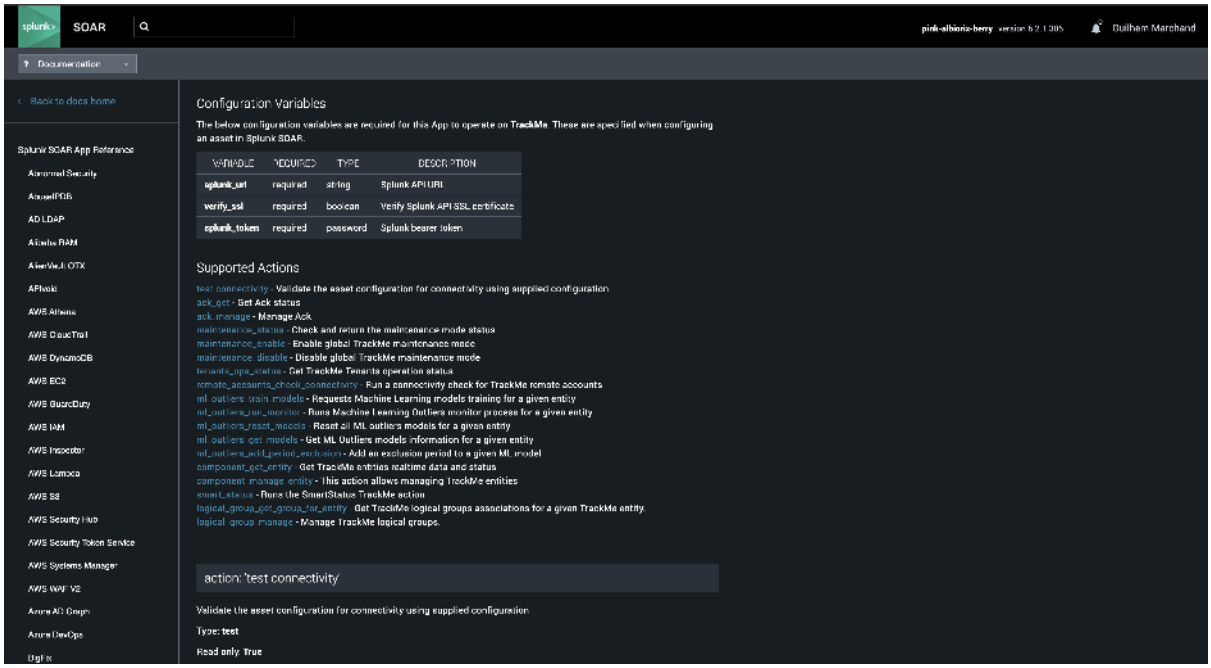
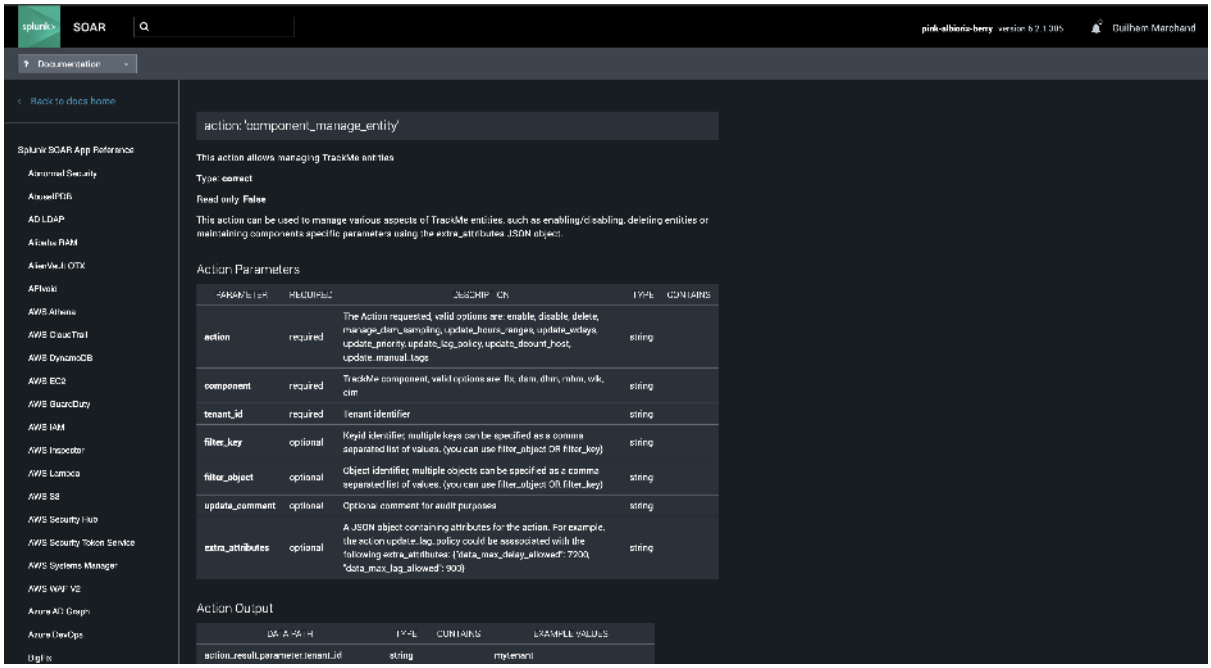
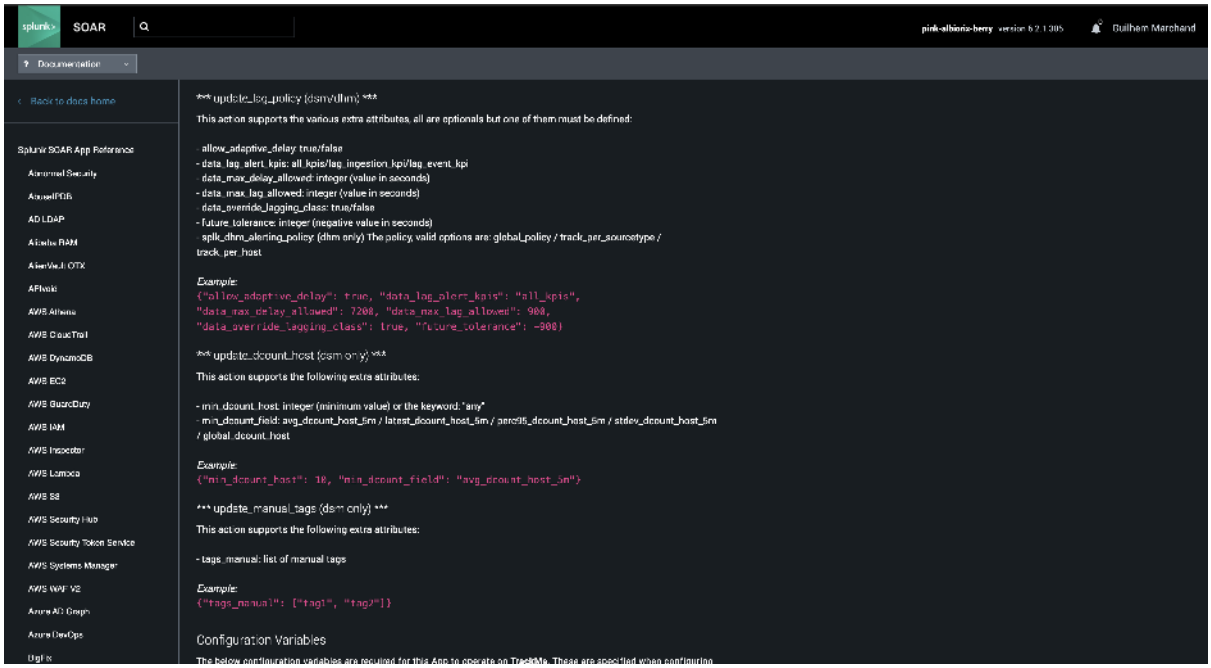
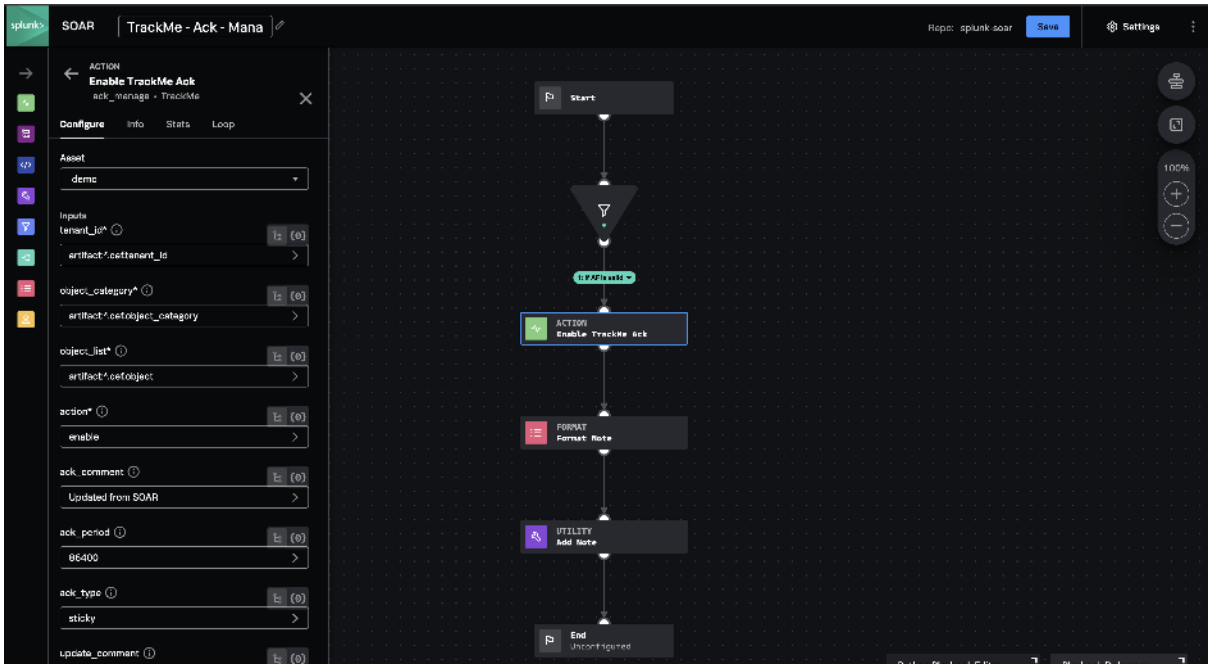
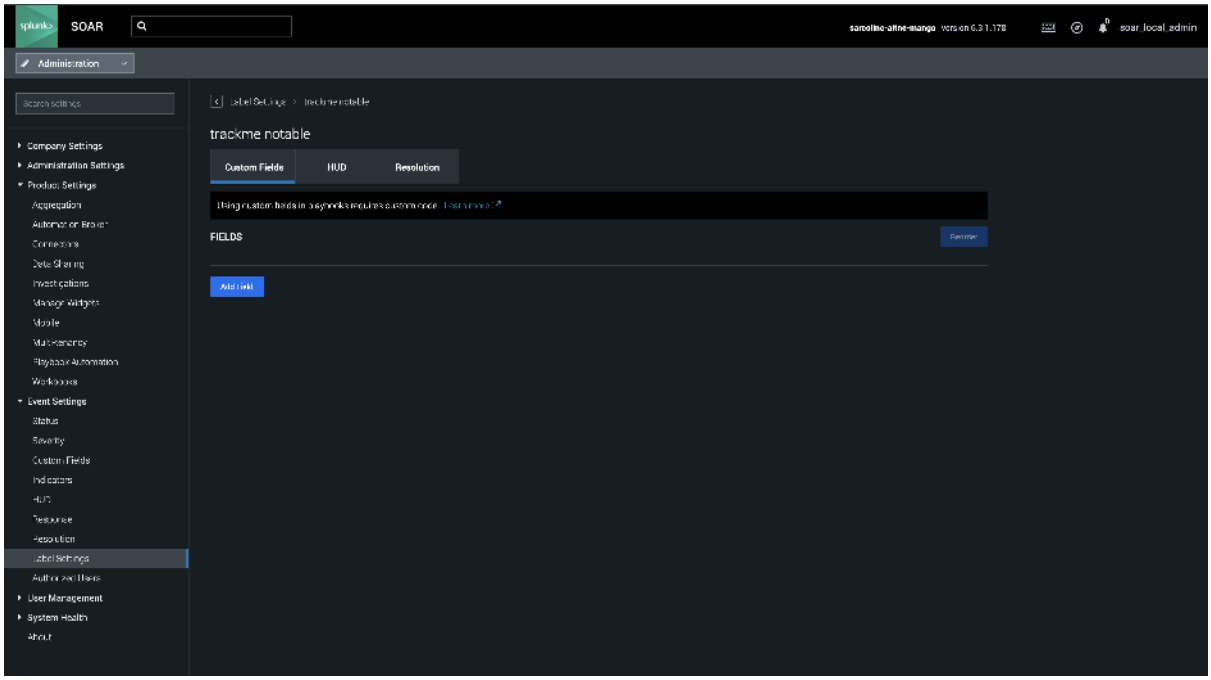
7.19.5 Creating a Workload tenant to host multiple Search Head tiers with overgroup

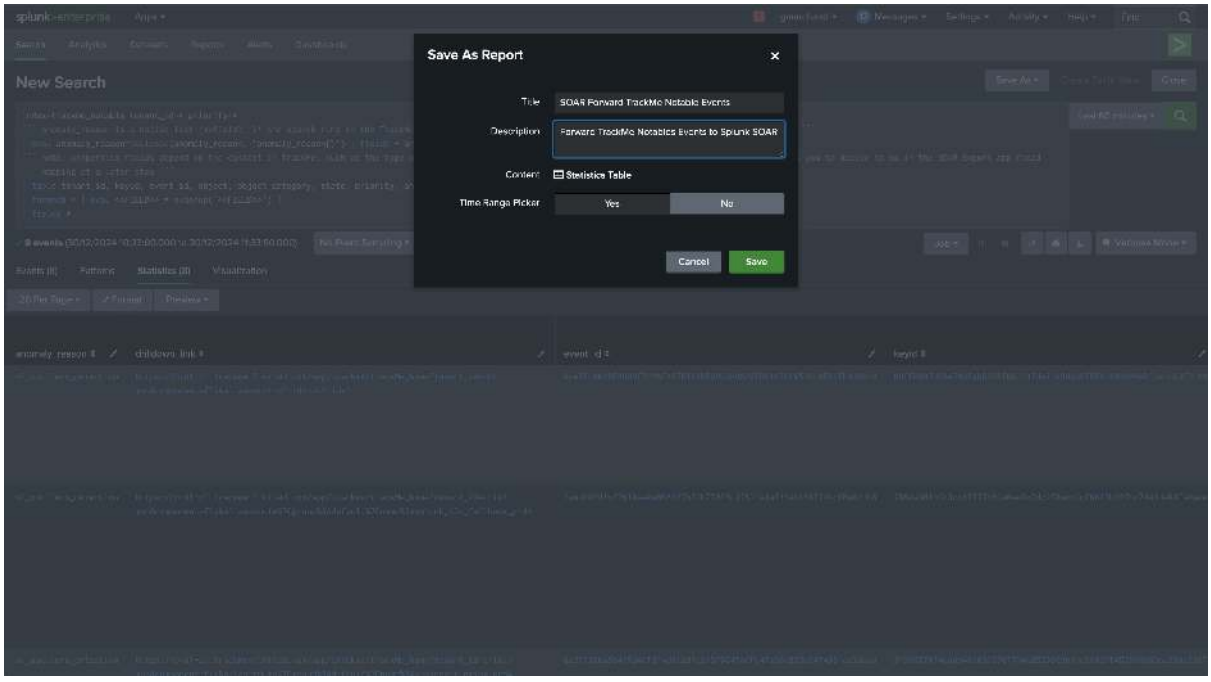
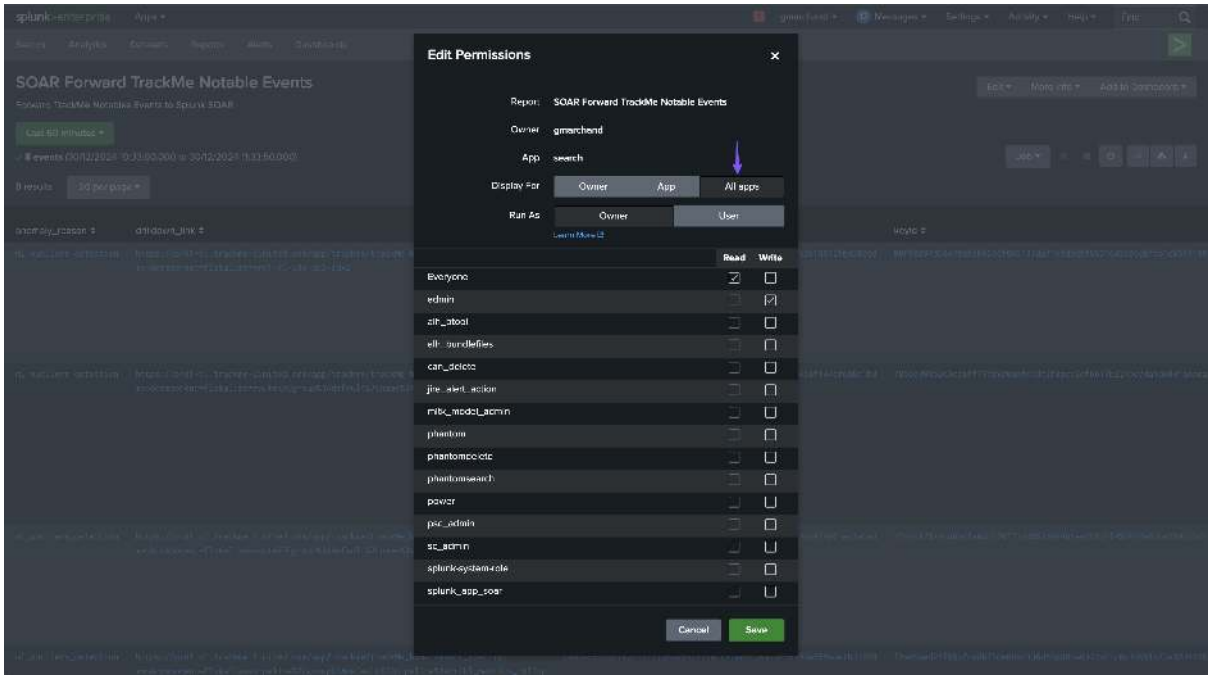
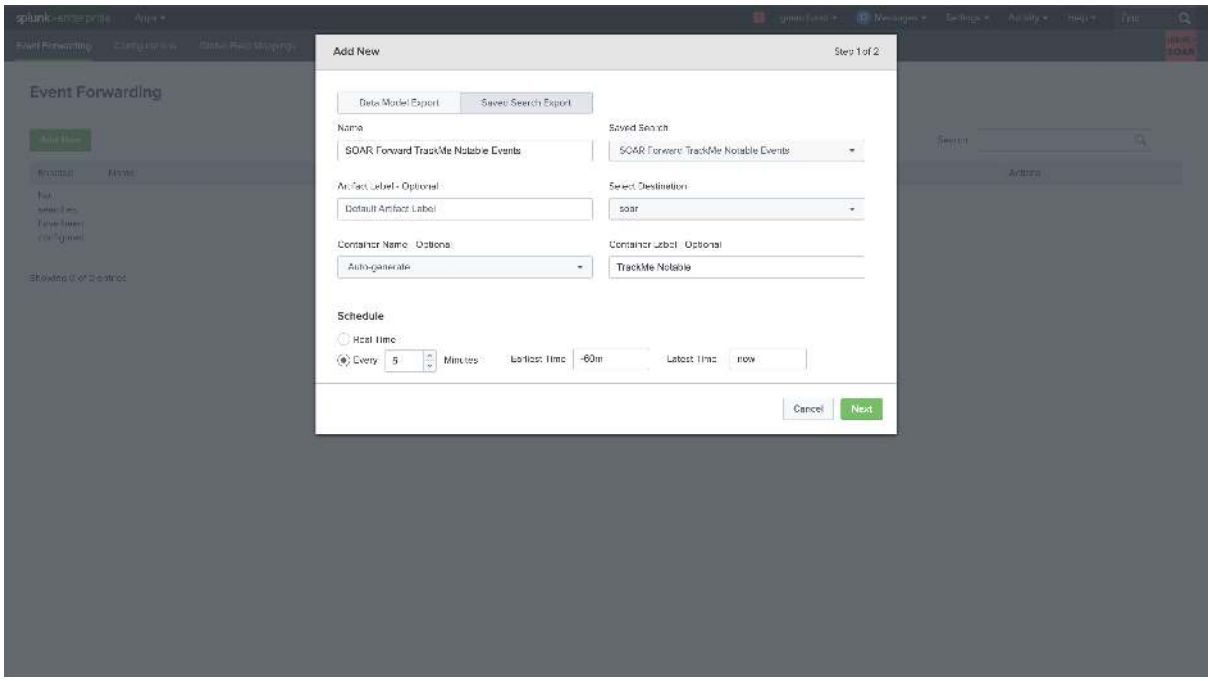
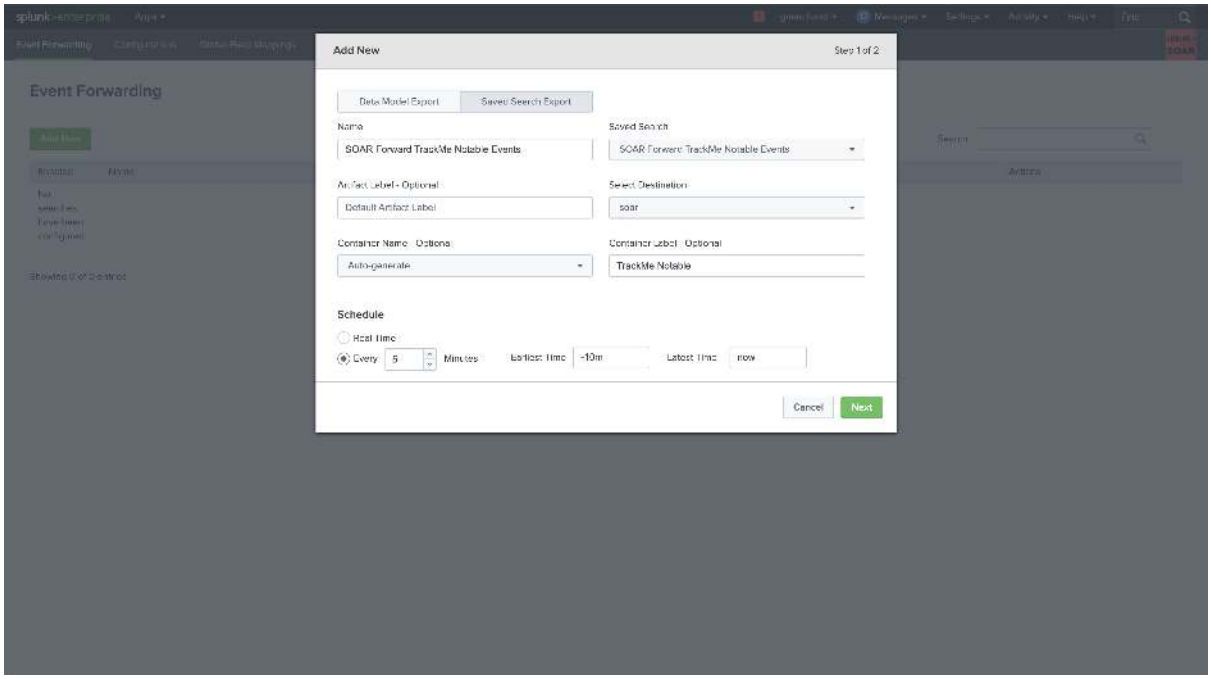
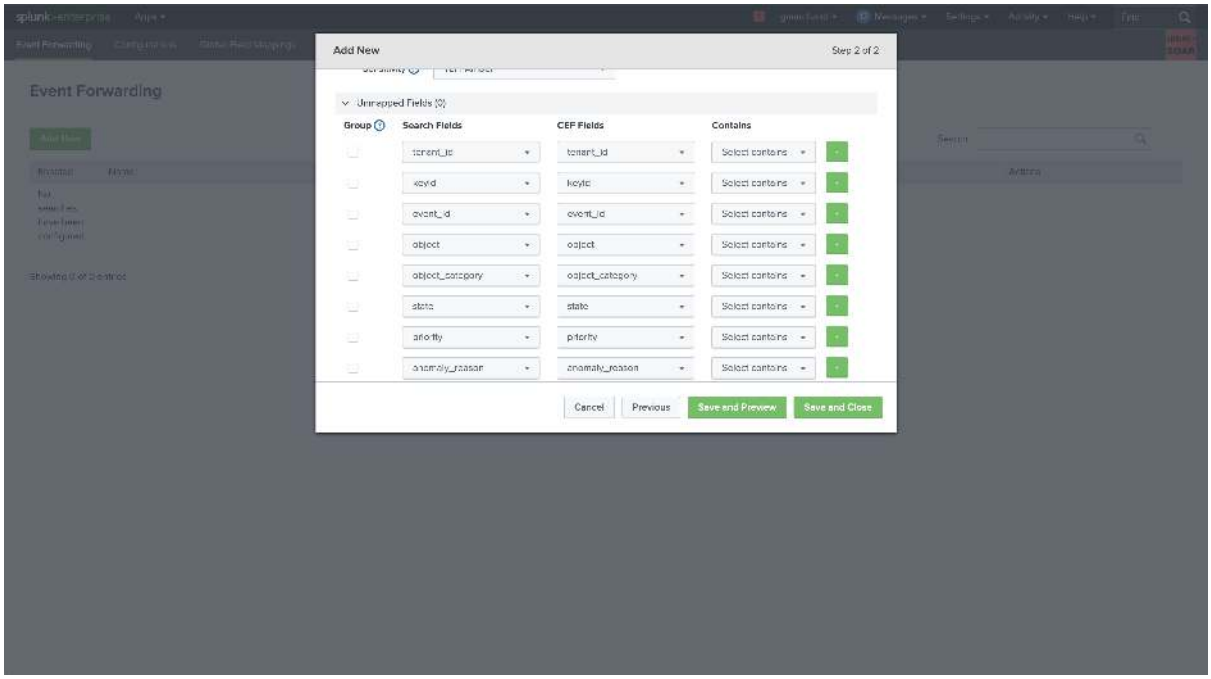
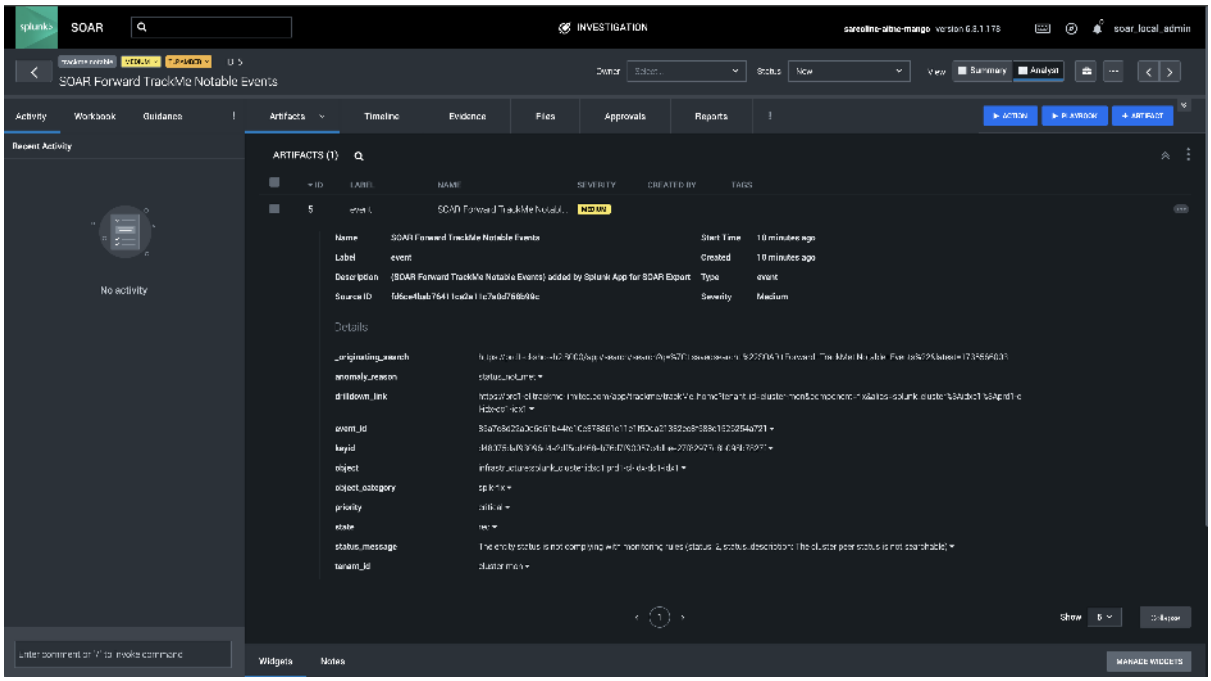
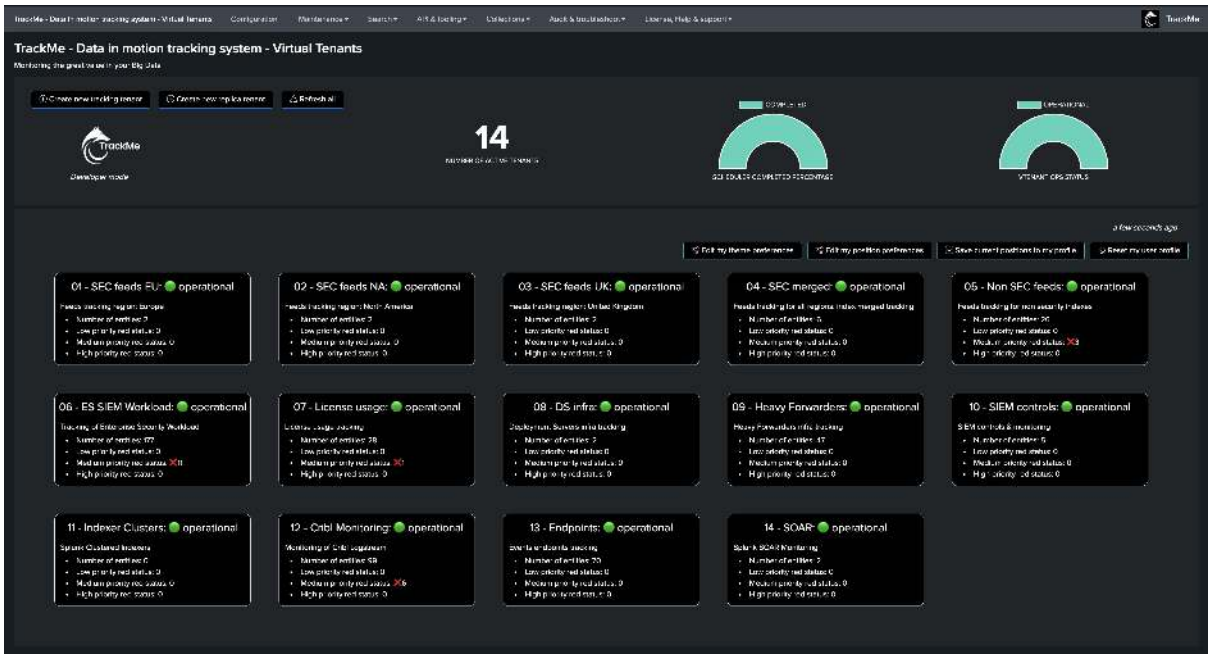
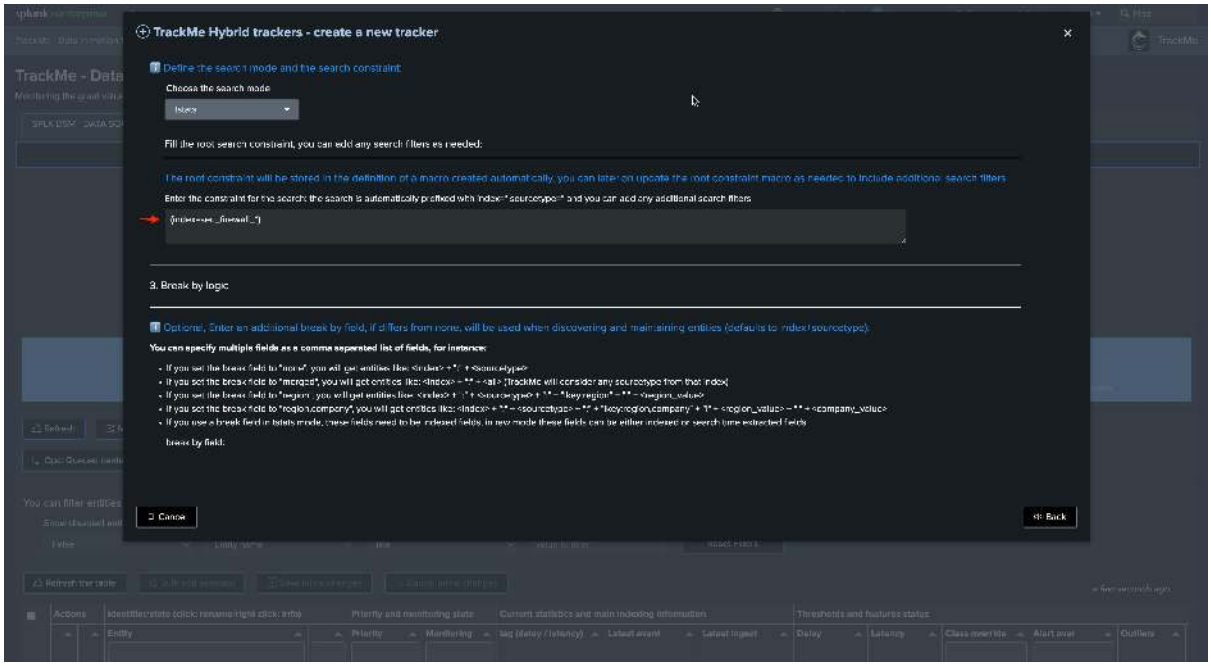
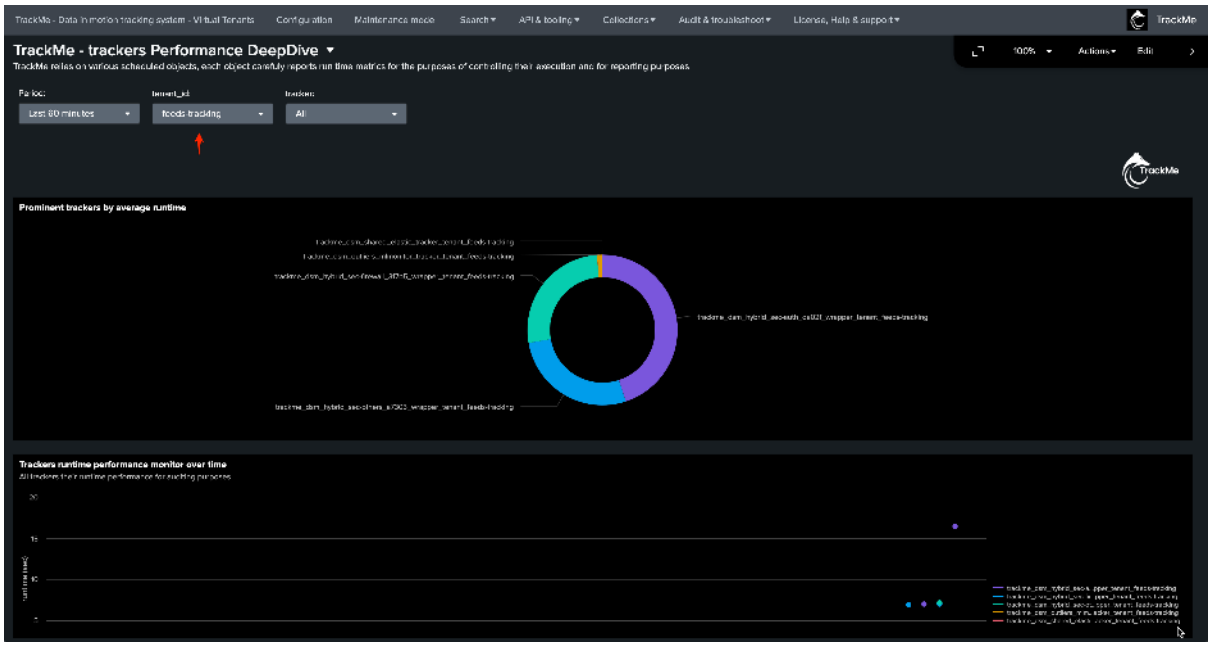
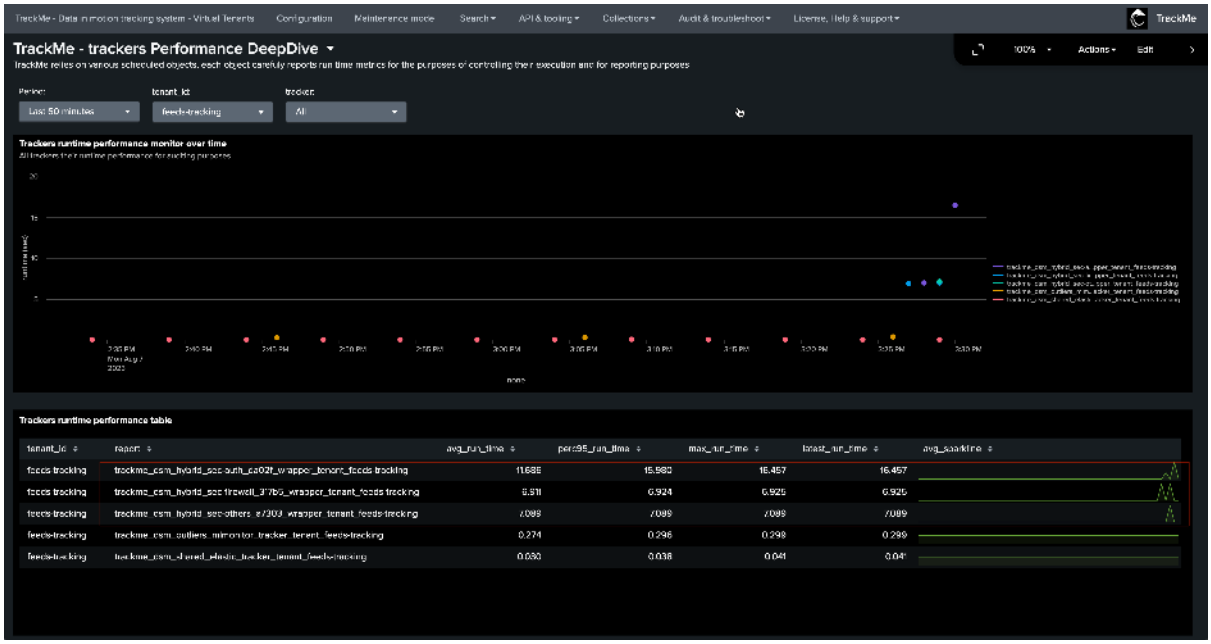
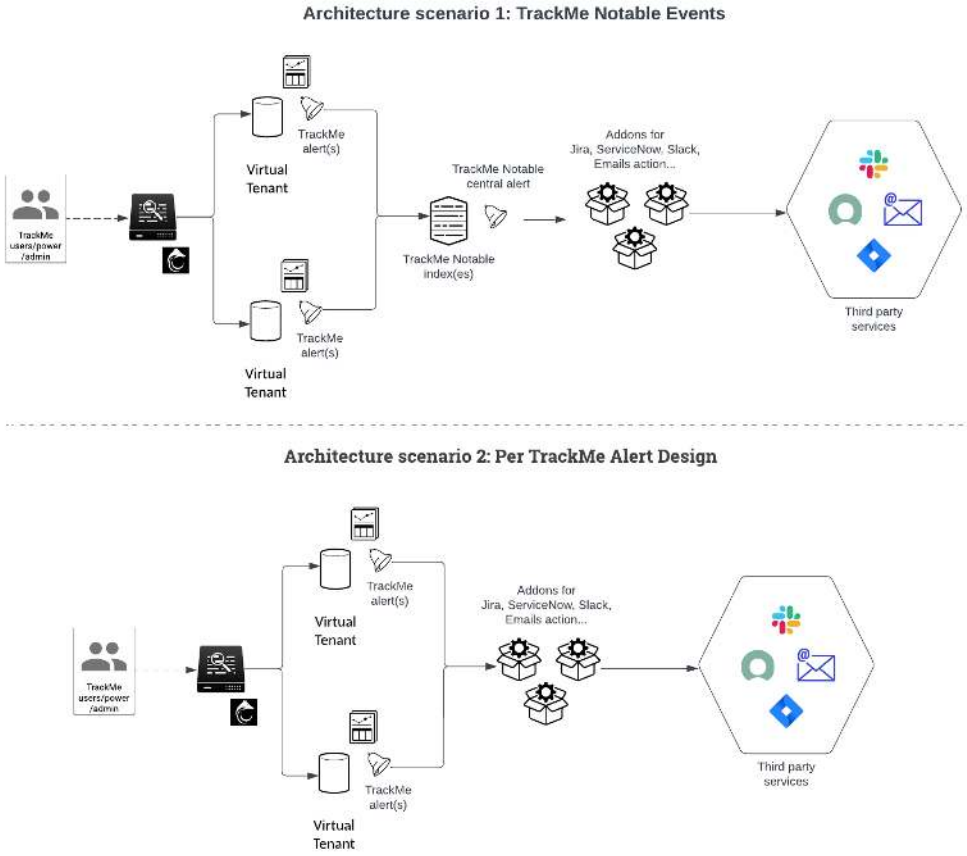
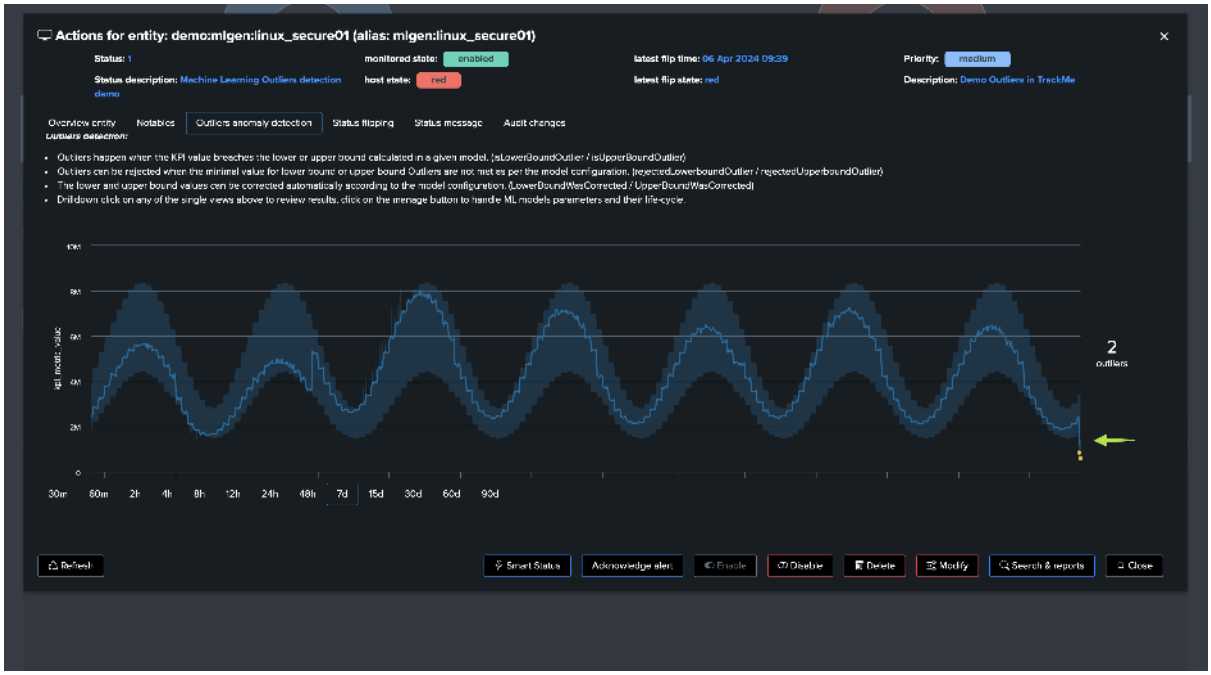
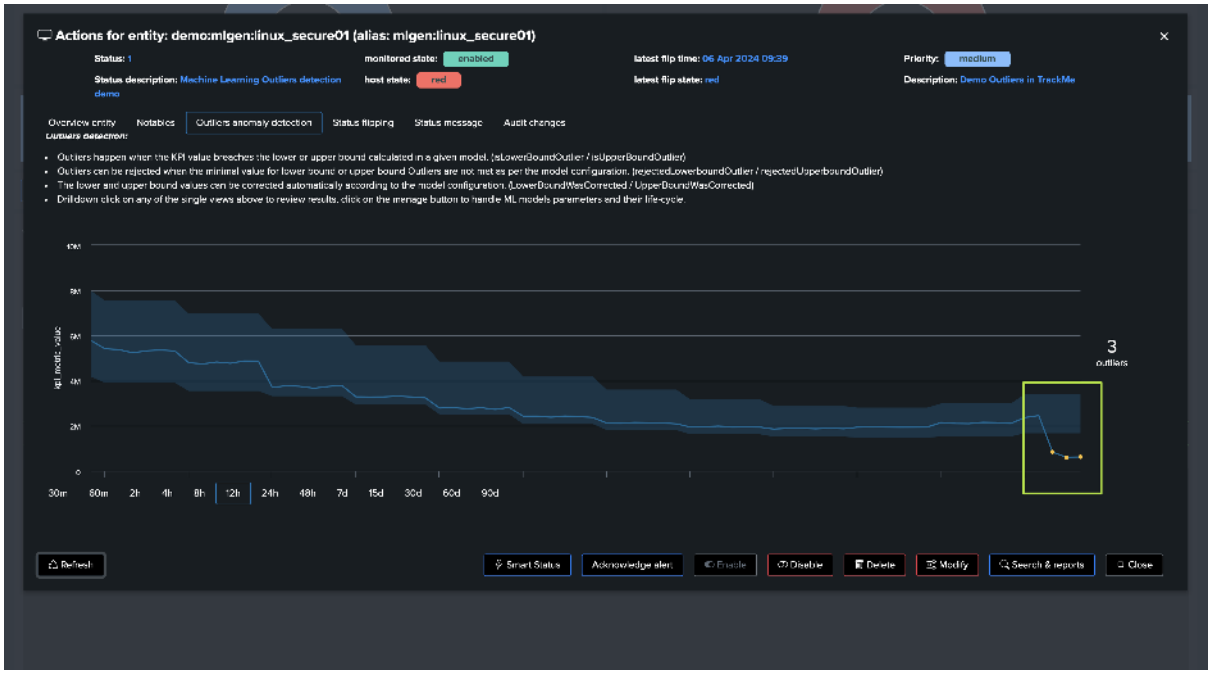
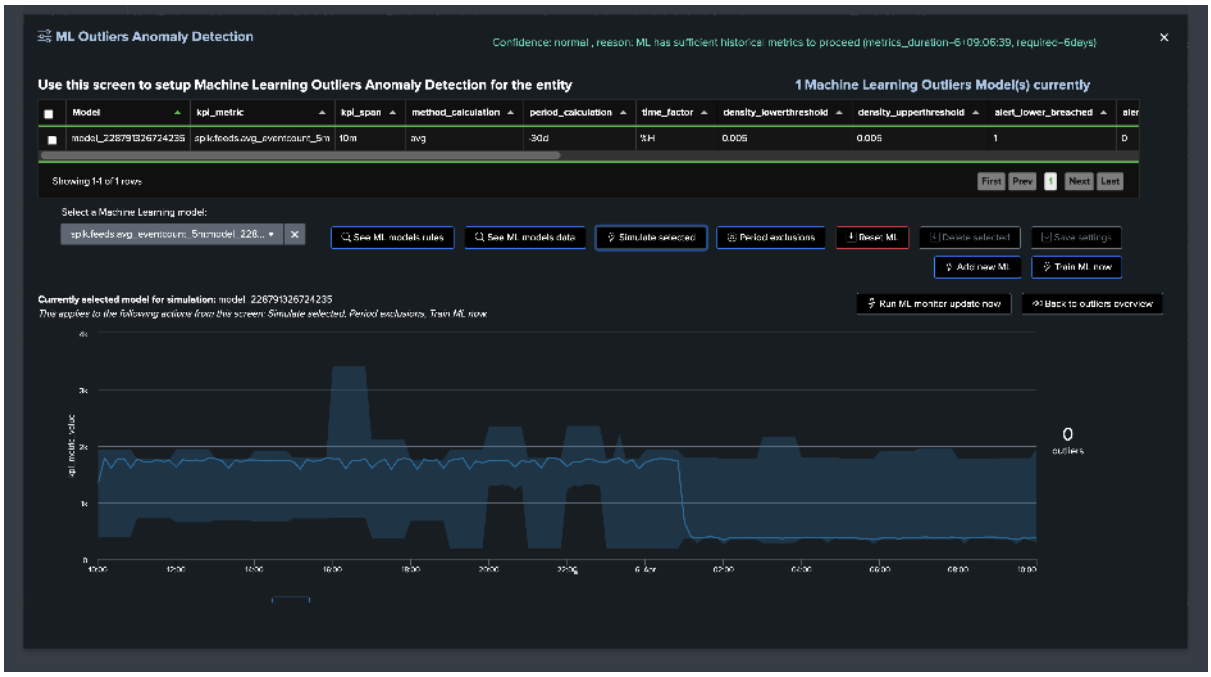
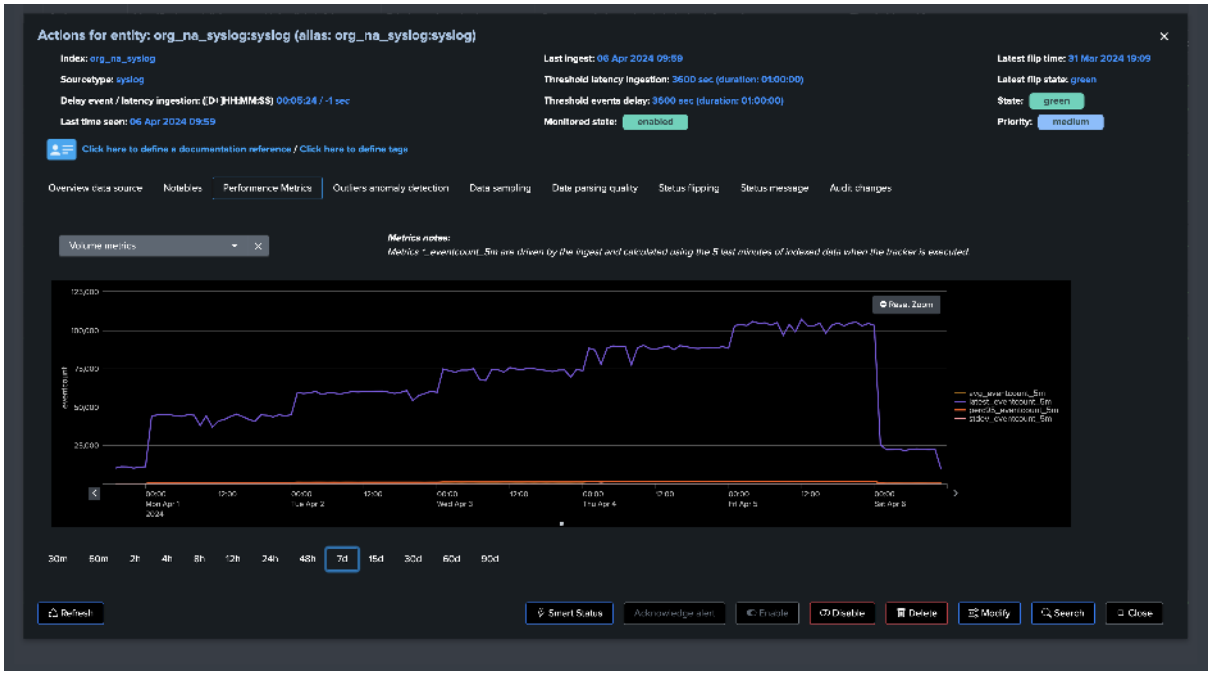
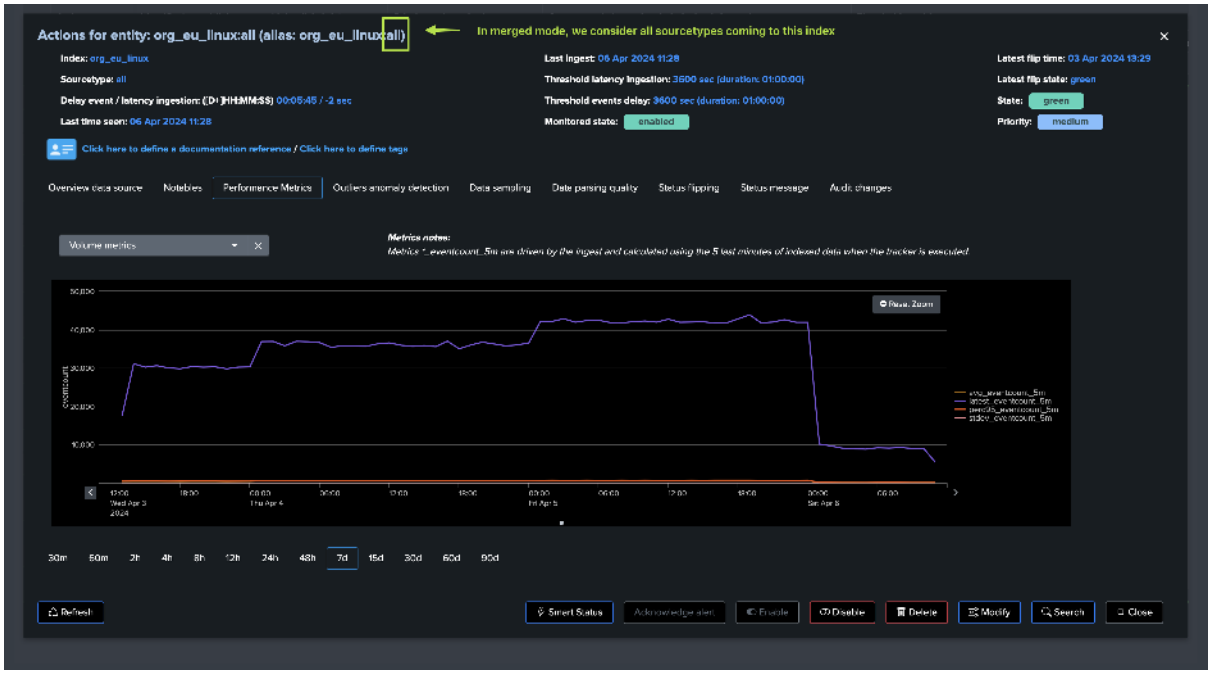
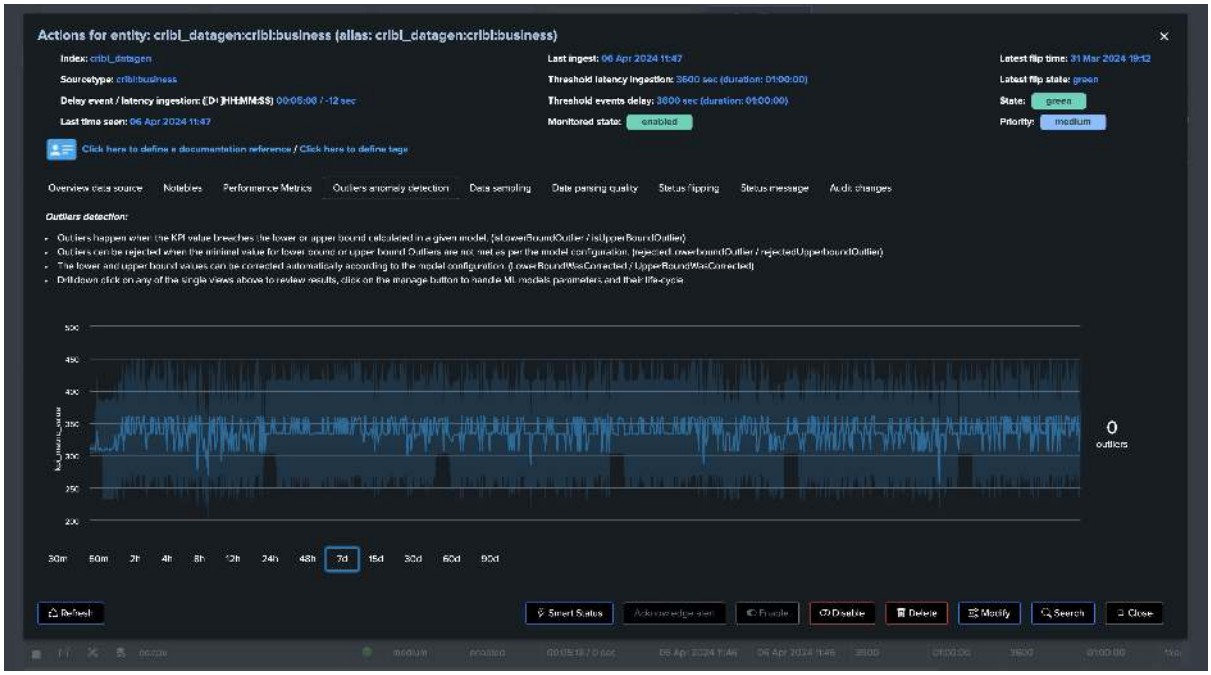
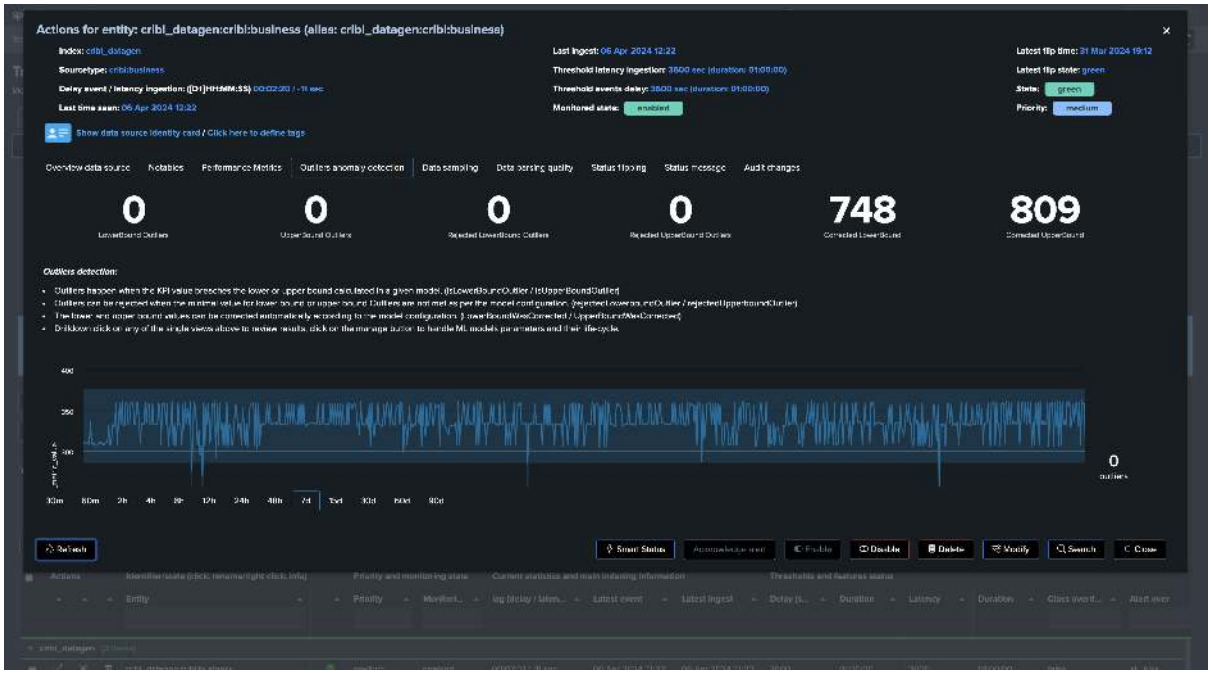
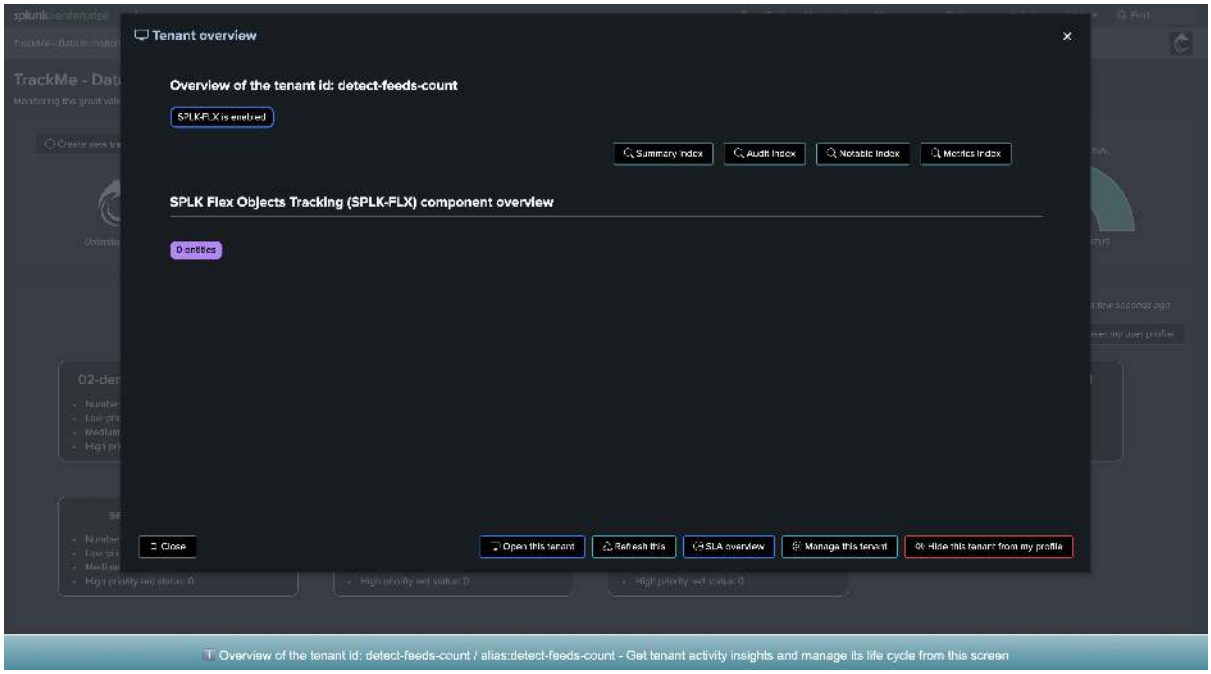
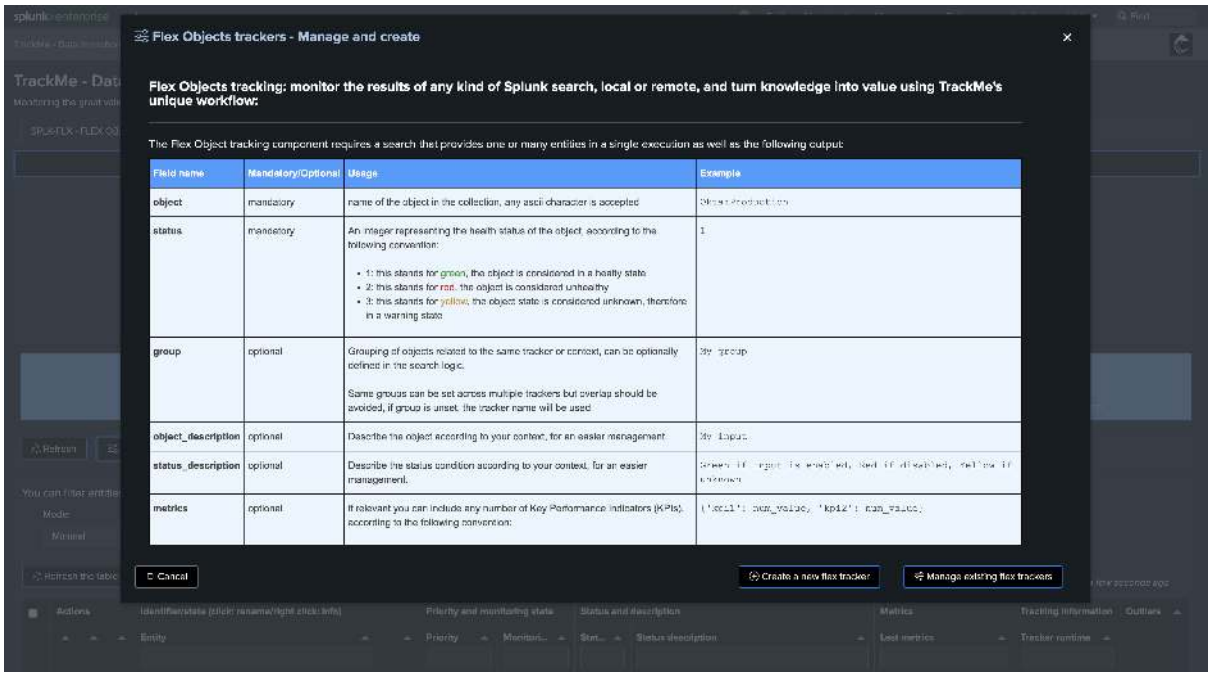
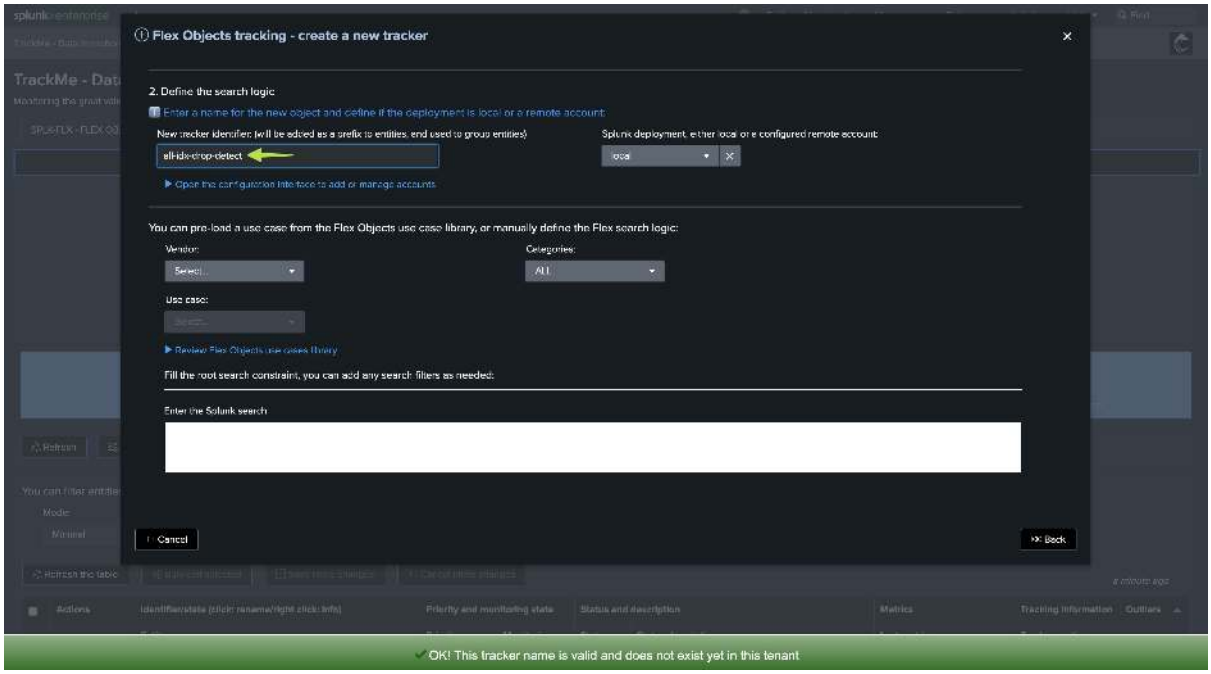
To create a tenant which hosts multiple Search Head tiers, you can use the overgroup feature, follow the steps below:

Step 1: Create the tenant with a first Search Head tier and the corresponding overgroup

We start by creating a brand new Workload tenant, where we define the first Search Head tier constraint, target (account) and overgroup value:



Define everything as needed (account, constraints, etc) then enter a custom value for the overgroup:
```



Once created and after 5/10 minutes, entities will be visible under the overgroup:



Step 2: In the Workload tenant, create the additional trackers for the second Search Head tier

Now that the tenant is created, we can create the additional trackers for the second Search Head tier, the following types of trackers can be created:

- scheduler
- introspection
- Splunk Cloud SVC (optional, for Splunk Cloud only)
- Notable (optional, if you wish to track notable activity)

Note

Common trackers:

- You do not need to create additional trackers for main, metadata and inactive entities
- There can be only one copy of these trackers per tenant, and these trackers can manage entities accordingly as long as these entities refer to a different account than the one used when creating the tenant
- Therefore, we can manage multiple Search Head tiers in a same tenant as long as each Search Head tier corresponds to a different account (which in fact will be the case systematically in the Workload component)



For instance, we will create trackers for a second Search Head tier targeting a remote account and will name the overgroup `sh2-scde`:







After 5/10 minutes, entities are now also created under the second overgroup as expected:



Metadata are managed transparently depending on their parent account, so TrackMe knows which Search Head tier to query for:



7.20 splk-flx - Creating and managing Flex Trackers

7.20.1 Introduction to Flex Trackers

Flex Trackers are created and managed through TrackMe. These are scheduled backend jobs that orchestrate entity discovery and management for the TrackMe splk-flx component:

- The **splk-flx** component stands for Splunk Flex Objects tracking
- This component allows tracking any kind of results from a Splunk search, to manage the resulting entities into TrackMe's unique workflow
- A Flex tracker can, for example, monitor modular input statuses on remote Heavy Forwarders, monitor Data Model Acceleration, or your IoT devices - this can be anything adapted to your

context

- The Flex component expects a certain convention allowing identification of entities and their status. You can automatically define which Key Performance Metrics should be part of it and even define default Machine Learning models for Outliers detection
- A single Tracker can discover and manage a few, or many entities according to the needs
- When creating a tracker, the related knowledge objects will be owned by the owner defined at the Virtual Tenant level
- TrackMe keeps records of the knowledge objects related to the Trackers; therefore, you need to manage its lifecycle through TrackMe

Hint

Flex Objects Library

- The Flex Objects component comes with a use case library of dozens of pre-built searches for Splunk Cloud, Splunk Enterprise, Splunk SOAR and even for third party product such as Cribl Logstream
- The use cases in the library can easily be loaded when creating a new Flex tracker
- See: *Splunk SOAR Cloud & on-premise monitoring and active actions in TrackMe*
- See: *Cribl Logstream monitoring in TrackMe*





7.20.2 Flex Object use cases Library & Tracker creation

You can access the use cases Library through the Hybrid tracker creation wizard:



The user interface also provides a direct link to the TrackMe use cases library provider custom command:

```
| trackmesplkflxgetuc
```



The Flex wizard opens. Start by defining the tracker identifier:

Note: This step is important as this identifier is used to categorize and group the entities in the user interface. Make sure to use any form of convention that makes sense for you:



Choose the environment target, if the target is remote, a connectivity check is immediately performed:



Define the search logic. The wizard shows the fields convention as well as their detailed usage, and several examples:







The following example tracks modular input statuses for the Splunk Okta Add-on on a remote Heavy Forwarder:



Another more advanced example that tracks Data Model Acceleration status on a remote Search Head:



Define the cron schedule. If the use case deals with REST-related searches, the time quantifiers generally do not matter:



Test and review, for instance, with the Okta modular input tracking:



Test and review with the Data Model example. As we specify metrics and outliers metrics, we can observe additional information:



Open in search if you wish to review manually the results in the Splunk Search UI:



Once you are happy with the results, you can proceed to the tracker creation:



After the tracker creation, you can execute it now:



7.20.3 Managing Flex Trackers

Deleting a Flex Tracker through the UI

If you want to delete an existing Flex Tracker, this operation must be done via TrackMe.

The reason is that the application keeps track of all knowledge objects that were created for a given tenant to honor various features such as managing the lifecycle of the tenant (enabling/disabling, etc.) or the lifecycle of the tracker itself.

To manage Flex Trackers, click on:



The user interface shows available trackers and their related objects:



Select one or more trackers to be deleted and proceed. TrackMe will call the related REST endpoint, knowledge objects will be purged, and TrackMe will also clean up the Virtual Tenant records.

TrackMe will **not** automatically purge the entities that were discovered and maintained when the originating tracker is deleted; however, these won't be maintained anymore.

Deleting a Flex Tracker through REST

You can delete a Tracker through the following REST endpoint, example in SPL:

```
| trackme mode=post url="/services/trackme/v2/splk_flx/admin/flx_tracker_delete" body=
↪"{'tenant_id': 'mytenant', 'hybrid_trackers_list': 'Okta:prod'}"
```

7.20.4 Key Performance Indicators in Flex Trackers

When creating a Flex tracker, you can leverage any Key Performance Indicator resulting from the Flex search to generate metrics automatically through TrackMe.

The following example tracks Data Model Acceleration (DMA) completeness and metrics. In short, the logic is the following:

- TrackMe orchestrates and executes a Splunk REST search which returns various information per entity (in this case, a given Common Information Model)
- The information is interpreted by TrackMe as Key Performance Indicators, leading to the generation and ingestion of metrics in the Splunk metric store
- Optionally, these Key Performance Indicators are automatically handled via the Machine Learning Outliers detection engine. ML models will be generated and maintained automatically for these KPIs
- Generating Key Performance Indicators, and therefore metrics, is optional. In some cases, this may not be relevant and is totally expected (such as monitoring statuses of modular inputs)

Example: When defining the DMA tracker, we enable various KPIs resulting from the Flex search:



To achieve this, all we need is to generate metrics from our resulting SPL query as a JSON object:

```
| eval metrics = "{'dma.complete_pct': " . complete_pct . ", 'dma.size_mb': " . size_
↪mb . ", 'dma.runduration_sec': " . round(runDuration, 2) . ", 'dma.buckets_count':
↪" . buckets . "}"
```

Optionally, you can choose which of these KPIs will be candidates for ML Outliers detection, and the basic parameters for the lower/upper threshold breached behavior:



This is also configured via the SPL query, in a resulting JSON formatted object:

```
| eval outliers_metrics = "{'dma.complete_pct': {'alert_lower_breached': 1, 'alert_
↪upper_breached': 0}, 'dma.runduration_sec': {'alert_lower_breached': 0, 'alert_
↪upper_breached': 1}}"
```

Once created, the Flex Tracker automatically generates and ingests these metrics in the metric store and starts to generate and maintain ML models for the purpose of Machine Learning Outliers detection:



Metrics are generated and indexed in the metric index of the Virtual Tenants:



Machine Learning models for Outliers detection will be created and maintained:



7.21 splk-cim - Creating and Managing CIM Trackers

7.21.1 Introduction to CIM Trackers

CIM Trackers are created and managed through TrackMe. These are scheduled backend jobs that orchestrate entity discovery and management for the TrackMe splk-cim component:

- The splk-cim component stands for Common Information Model compliance tracking
- This component allows tracking the compliance of your CIM parsing from the perspective of the CIM data models
- Tracking CIM compliance is a challenging task from various aspects, including complexity, scala-

bility, and computing costs

- TrackMe uniquely tackles these challenges using a powerful set of techniques and backend custom commands that can consider each field individually at scale, applying highly configurable rules in a robust and flexible workflow
- TrackMe provides a suggested regular expression for each CIM field you want to monitor in a given data model, storing the tracking rules in a JSON dictionary
- For each CIM field to be monitored, you can specify the regular expression validating the format, the minimum percentage of compliant results, and the maximum percentage of unknown results
- A single Tracker can manage a few or many CIM fields in a single data model
- You can update CIM tracking rules at any time. When creating CIM trackers, you can upload existing templates that you have previously defined, or clone tracking rules from an existing CIM tracker
- When creating trackers, the related knowledge objects will be owned by the owner defined at the Virtual Tenant level
- TrackMe keeps records of the knowledge objects related to the Trackers, therefore you need to manage their lifecycle through TrackMe



7.21.2 Creating a CIM Tracker

Once you have created a CIM Virtual Tenant, you can create one or more CIM Trackers:



Wizard creation welcoming screen:



First step if so define the name of the CIM entity, in this example we will create a Web data model based entity to track Apache Web Access log compliance:



Then, define if the target is local or remote, in our example the DMA data is locally available but it could be equally a remote Splunk deployment:



Define the Data model name and the Data model node name. Consult the data model structure if needed. In our example:





Add search filter according to the Data model context and your target. In our case, we will leverage the vendor_product Data model field:





Note: You can add any search filter as long as it is available in the context of the Data model. You must use the data model field name

Define the drop Data model context. This field is automatically pre-filled. In complex scenarios, you can add multiple node name drop definitions (as a comma-separated list of values):



Define if we should require only accelerated data. This is highly recommended for optimization and compute costs purposes:



Optionally define a break by field logic. In most cases, use none to keep the default behavior. In some advanced cases, you can use this field to manage many entities from a single Tracker, such as using a custom cim_entity_zone concept you have added to the Data model:



The wizard shows a search preview for your information only:



Before continuing, you may want to verify that these return the expected data:



The next step is to define which fields should be monitored and what the rules for these should be:



There are basically three options here:



- You can upload a previously defined JSON template containing the CIM tracking rules structure. If you have already defined a schema, you can rely on this option to easily repeat the creation process
- You can clone the CIM tracking definition from another existing CIM tracker
- You can add field by field the CIM fields to be monitored

Assuming this is your first CIM tracker, we will add fields manually and explain the different options:

In our example, we are going to monitor a selected list of fields. We selected the most important fields as these are the fields that you would use in your security use cases, especially at the level of the tstats root search or the break by statement:

These are examples for the purposes of the documentation. You should maintain a dictionary of your use cases and identify which CIM fields are mandatory for your security use cases

- action
- bytes
- dest
- dest_port
- src
- status
- url
- url_domain
- url_length

When adding a CIM field, the wizard automatically defines the node name according to your previous settings, as well as a pre-configured regular expression and default acceptable values:



Note: These values can be modified and updated later on once the CIM tracker is created

Consider the following example: During our initial data onboarding setup, we started to ingest Apache access logs with the standard Apache logging format, which wasn't compatible with the Splunk TA requirements for CIM parsing:



If our security use case (our Splunk Enterprise Security) expects a proper format or specific types of values, the use case would simply not be functional or unexpectedly exclude this very important Web server!

This is the purpose of the CIM compliance tracking: ensuring that efforts put into use case development are worthwhile, and that the CIM parsing remains compliant with what was qualified. Once we fixed our onboarding issues:



Once we have added all the fields we want to monitor, a JSON preview shows the CIM Tracking rules structure and details:



We can now simulate the CIM Tracker creation. This step ensures that the JSON structures are well-formed and will be accepted by the TrackMe REST API endpoint:



Next, we can simulate the Tracker execution and review its performance and results. You can at this stage define earliest and latest time quantifiers:





Update tracking rules during the simulation

- You can update the tracking rules during the simulation. Go back to the previous step, edit the rules, and click on apply



Carefully review the Tracker performance and results. In our example and for the purpose of the demonstration, if we increase the time range to capture when we were having CIM parsing issues, the entity would return in a red state. This is totally expected as we identified and fixed these onboarding issues!



We can observe different fields causing trouble:





Once everything is done, we can define earliest and latest time ranges, and the cron schedule. Click on promote to create the CIM Tracker immediately:



After the tracker creation, you can execute it now:



The Tracker is now active, and the entity is available in the Virtual Tenant:





7.21.3 Managing CIM Trackers & Advanced Options

Custom Entity Break By

When creating a CIM tracker, you can optionally set a custom break by entity. By default, the custom entity defaults to none, which is translated as the global entity in the JSON results of the CIM tracking:



Use Cases for a Custom CIM Break By

Consider the following example: Say you have an indexed field which represents a region that is mapped to the cim_entity_zone field.

Some customers will update and customize the CIM data models to leverage the cim_entity_zone field, and will add this as an additional field in the data model.

Your security use cases can therefore filter on and/or break by the cim_entity_zone to distinguish the originating region from a result within a single correlation search in Enterprise Security. In a simplistic example:

```
| tstats count from datamodel=Web where (nodename = Web) by Web.cim_entity_zone
```

From the TrackMe perspective, without a custom break by, you would need to create one tracker per region. However, leveraging the custom break manages this automatically by associating the custom break by field to an entity concept:






TrackMe CIM trackers - create a new tracker
JSON rules preview:
Cancel
Back


TrackMe CIM trackers - create a new tracker
Cancel
Back



Once created, the Tracker shows one top JSON entity per region:



Updating the CIM Tracker JSON Rules

You can update the CIM tracker rules by editing the JSON structure directly. Open the CIM entity edition:





Edition can be made by:

- Uploading a previously defined JSON template
- Manually editing the JSON structure

Downloading the CIM Tracker JSON Rules for Use as Template

The same screen allows you to download the CIM tracking rules of a given CIM entity, for the purposes of using it as the JSON template or manual edition:



CIM Tracking JSON Structure

The following JSON structure is expected:

```
{
    "cim_datamodel_name": "Web",
    "cim_datamodel_nodename": "Web",
    "cim_drop_dm_object_name": "Web",
    "cim_breakby_prefix": "Web",
    "cim_breakby": "none",
    "cim_fields": "action,bytes",
    "cim_root_constraint": "Web.vendor_product=\"Apache Web Server\"",
    "action": [{
            "datamodel_fieldname": "Web.action"
        },
        {
            "min_compliant_coverage_percentage": 95
        },
        {
            "max_unknown_coverage_percentage": 5
        },
        {
            "regex_validator_rule": ".*"
        }
    ],
    "bytes": [{
            "datamodel_fieldname": "Web.bytes"
        },
        {
            "min_compliant_coverage_percentage": 95
        },
        {
            "max_unknown_coverage_percentage": 5
        },
        {
            "regex_validator_rule": "^\\d+$"
```

(continues on next page)

(continued from previous page)

```
        }
    ]
}
```

JSON top structure:

- cim_datamodel_name: the name of the CIM data model
- cim_datamodel_nodename: the node name of the CIM data model
- cim_drop_dm_object_name: a comma-separated list of node names to be dropped
- cim_breakby_prefix: if using a custom break by, the break by field prefix needs to be set
- cim_breakby: the name of the field if using a custom entity break by, defaults to none which corresponds to the global entity, see TBD
- cim_fields: the comma-separated list of CIM fields to be monitored
- cim_root_constraint: the root search constraint when performing the tstats searches

For each CIM field, a JSON dictionary defines:

- datamodel_fieldname: the data model field name including the data model nodename
- min_compliant_coverage_percentage: the minimum compliant coverage percentage. If the percentage returned from the current execution for that field falls below the threshold, the CIM entity turns red
- max_unknown_coverage_percentage: the maximum percentage of unknown values for that field (a common practice in the data model is to set the field value to "unknown" if it is null). If the percentage returned in the current execution falls below this threshold, the CIM entity turns red
- regex_validator_rule: a regular expression used to validate the compliance of CIM values for a given field

Deleting a CIM Tracker Through the UI

If you want to delete an existing CIM Tracker, this operation must be done via TrackMe.

The reason is that the application keeps track of all knowledge objects that were created for a given tenant, to honor various features such as managing the lifecycle of the tenant (enabling / disabling, etc.) or the lifecycle of the tracker itself.

When deleting a CIM Tracker, the TrackMe entity associated with the Tracker **will be automatically deleted**.





Deleting a CIM Tracker Through REST

You can delete a Tracker through the following REST endpoint, example in SPL:

```
| trackme mode=post url="/services/trackme/v2/splk_cim/admin/cim_tracker_delete" body=
↪"{'tenant_id': 'mytenant', 'object_list': 'web:apache'}"
```

7.22 TrackMe Tags enrichment

7.22.1 Introduction to tags enrichments

Hint

About Tags in TrackMe

- Tags are used to provide additional context to the entities
- Tags can be used for filtering purposes in the user interface
- Tags can be used for alerting purposes
- Tags can be used for additional context in the user interface



7.22.2 Implementing tags enrichments in TrackMe (all components)

Extension of Tags features for all components in TrackMe 2.0.98

- Since TrackMe version 2.0.98, tags enrichments are available for all components in TrackMe with the exact same level of features, and user experience.

Tags enrichments in TrackMe can be defined by:

- **Tags policies:** Regular expressions policies which define a number of tags for matching entities, once defined, these policies are applied automatically to the entities.
- **Manual Tags:** In Addition, you can manually add tags to the entities from the user interface.

Accessing, defining and managing tags policies

You can access tags policies from the main TrackMe's tenant screen via the button `Manage: tags policies`:



About Tags policies:

- **Regex based:** Tags policies are regular expressions which define a number of tags for matching entities
- **Tags submission:** These policies are monitored by a scheduled job named `trackme_<component_suffix>_tags_tracker_tenant_<tenant_id>`
- **Multiple matching policies:** Multiple policies can be applied, this means a given entity can receive tags from one or more tags policies
- **Lower case:** Tags are automatically stored in lower case format
- **Removing tags policies:** When tags policies are removed, the tags associated with entities from the removed policies are removed from the entities automatically
- **Tags in the KVstore records:** Tags are stored in the entities KVstore records, and can be used for alerting filtering purposes
- **Tags in the user interface:** Tags are displayed in the user interface for the entities

Tags policies usage example

In the following example, we have a number of entities which have incoming some patterns we can leverage for tags policies purposes:

- geo-locations: the entities naming convention includes the source location of these feeds
- technology information: we can also leverage some of these information to associate these entities with technology related context tags

Example of tags policies definition:

In this example, we are going to rely on the index naming convention to define tags, so we can automatically classify TrackMe entities.

In short, we are going to apply the simple following logic:

geographical location:

regular expression	tags
`.*_na_.*`	`north_america`
`.*_eu_.*`	`europe`
`.*_uk_.*`	`united_kingdom`

technology information:

regular expression	tags
`.*_syslog:.*`	`network`
`.*_linux:.*`	`os`

Once configured in TrackMe's UI:



Tags can be used for filtering purposes in the user interface:

Tags can also be accessed per entity, and modified from here too:



Tags are parts of TrackMe notable events, so you can use these for filtering, enrichment, decision making and so forth:

```
index=trackme_notable tenant_id=<replace with tenant_id>
```



Tags are included in the trackme:state events:

```
index=trackme_summary sourcetype="trackme:state" tenant_id=<replace with tenant_id>
```



Tags can be accessed through the TrackMe get collection command, which renders entities from Decision Maker (same as the UI view):

```
| trackmegetcoll tenant_id=<tenant_id> component=<component_suffix>
```





Troubleshoot tags policies

Tags policies tracker

The first component is the scheduled job which applies the tags policies to the entities, this job is called as follows:

- trackme_<component_suffix>_tags_tracker_tenant_<tenant_id>



This jobs should be scheduled, and properly executed:



You can find its logs as follows:

```
index=_internal sourcetype=trackme:custom_commands:trackmesplktags
```

When you have multiple tenants, you may want to filter on the tenant identifier:

```
index=_internal sourcetype=trackme:custom_commands:trackmesplktags tenant_id=<replace␣
↪with tenant_id>
```

In case of any issues with the tags policies, ensure that the job exists, is scheduled and executes properly and continue with the following steps.

Tags policies REST API endpoint

The scheduled job calls the REST API endpoint to apply the tags policies, you can review the logs for this endpoint as follows:

```
index=_internal sourcetype=trackme:rest_api post_tag_policies_apply
```

With multiple tenants, filter on the tenant identifier:

```
index=_internal sourcetype=trackme:rest_api post_tag_policies_apply tenant_id=
↪<replace with tenant_id>
```



In case of issues, carefully review the logs and verify if there are exceptions or issues with the tags policies:

```
index=_internal sourcetype=trackme:rest_api post_tag_policies_apply log_level=error
```

The REST API endpoint can be applied manually as well, you can easily get the list of endpoints and their usage, review the REST API reference dashboard:



In fact, you can apply the tags policies manually:

```
| trackme mode=post url="/services/trackme/v2/splk_tag_policies/write/tag_policies_
↪apply" body="{'tenant_id': '<replace with the tenant id>'}"
```



Tags policies KVstore collection

When tags policies are applied, TrackMe entities get a record with the list of associated tags, if any:

```
| inputlookup trackme_dsm_tags_tenant_<replace with tenant id>
```



When TrackMe trackers are executed, tags records are automatically crossed enriched with the entities records to form the final entities records tags fields:

```
| inputlookup trackme_dsm_tenant_<replace with tenant id> | table object, *tags*
```



7.23 Feeds (DataSource - splk-dsm) - Documentation Notes & Links

7.23.1 Introduction to Documentation Notes & Links for splk-dsm

Documentation Notes & Links

- The documentation notes for splk-dsm provide easily accessible documentation context directly in the TrackMe UI
- This feature is exclusively available for the Data Source tracking component (splk-dsm)
- You can define **global and default documentation notes and links**, which apply to every entity in TrackMe
- Additionally, you can define **documentation notes and links for individual entities or link them to multiple entities**
- When both global and entity-specific documentation notes are defined, the entity-specific documentation notes take precedence

7.23.2 Where to Find the Documentation Notes & Links in TrackMe UI

When you open the screen of a specific entity, TrackMe shows at the top of the screen a link referring to documentation notes & links, depending on whether documentation notes have been defined yet (global or per entity):

No documentation notes defined yet:



Documentation notes defined:

If documentation notes are defined, the content would appear as:



If documentation notes are not defined yet, the following screen shows up and allows you to create a new documentation reference, or link to an existing documentation reference:



7.23.3 Defining Global Documentation Notes & Links

To define global documentation notes & links, navigate to the "Configure" screen and select the "splk-general" tab:


splunk>enterprise
Apps
TrackMe - Data in motion tracking system - Virtual Tenants
Configuration
Maintenance
Search
API & tooling
Collections
Audit & troubleshoot
License, Help & support
TrackMe
Configuration
Configure TrackMe
OpenAPI.json
Remote deployments accounts
Vtenants prefs
General
Indexes
Prefs Vtenants UI
Prefs Home UI
splk-general
splk data sampling
splk outliers detection
Maintenance
Logging
* Index time parsing filter
host=*
* Latency default (splk-dsm)
3600
* Delay default (splk-fqm)
3600
* Latency default (splk-dhm)
3600
* Delay default (splk-dhm)
3600
* Delay default (splk-wlk)
900
* Concurrent searches Elastic
3
* CMDB lookup search splk-dhm
| inputlookup my_cmdb where (host="$alias$")
* CMDB lookup search splk-wlk
| inputlookup my_cmdb where (host="$alias$")
* CMDB lookup search splk-dsm
| inputlookup my_cmdb where (object="$object$
* CMDB lookup search splk-fqm
| inputlookup my_cmdb where (object="$object$
* CMDB lookup search splk-vtc
| inputlookup my_cmdb where (vtenantsearch_par
splk-dsm docs note global
splk-dsm docs link global
Save

Example:





Note that the documentation notes & links defined globally can be overridden on a per-entity basis, so you can still reference specific documentation notes & links for each entity.

Before TrackMe 2.0.87

- Prior to this release of TrackMe, the global configuration was managed through several Splunk macros (trackme_identity_card_default_url, trackme_identity_card_default_note)
- If you were using this feature before this version, please migrate your configuration to the new global configuration screen

7.23.4 Defining Documentation Notes & Links Per Entity

If a global documentation reference is defined, TrackMe shows an informational message and allows defining an entity-specific documentation reference, or linking to an existing one:



If a specific entity note was defined, however, the screen will allow deleting or updating this reference:



7.23.5 REST API Endpoints

Identity cards are managed through read-only and write REST API endpoints. For detailed usage and examples, consult the REST API Dashboard in TrackMe:

API resource groups:

- `splk_identity_cards`: read-only endpoints (requires trackmeuseroperations capability)
- `splk_identity_cards/write`: write endpoints (requires trackmepoweroperations capability)



7.24 Pushing Expected Sources to TrackMe (Tracking Expected Sources or Hosts in splk-dsm/splk-dhm)

About Pushing Expected Sources

- This guide explains how to push expected sources or hosts to TrackMe Data Sources Monitoring (splk-dsm) or Data Hosts Monitoring (splk-dhm).
- This can be useful if you want to manually insert entities into TrackMe that have not yet been discovered, based on a CMDB or similar knowledge base.
- When pushing expected sources, these entities will be added as if they were discovered by TrackMe under normal circumstances.
- If these entities are not yet sending data to Splunk, or are not covered by the scope of your trackers, they will appear as red in TrackMe.
- As soon as these entities start sending data to Splunk or are covered by the scope of your trackers, they will be updated in TrackMe accordingly.
- This guide requires **TrackMe 2.1.18** or higher.

7.24.1 Pushing Expected Data Sources to TrackMe Data Sources Monitoring (splk-dsm)

In this example, we push expected sources based on a Splunk lookup table containing the list of expected sources:

Suppose we have a Splunk lookup table containing the list of expected sources, based on a list of indexes and sourcetypes:

```
| inputlookup ds_expected.csv
| fields index, sourcetype
```



We have a Virtual Tenant called `secops` that is active and already contains a number of entities, some of which might already be part of the expected sources, but some might not:



To push expected sources to TrackMe from the lookup file, we will call the streaming command `trackmepushdatasource`, which will:

- Parse the records resulting from the `inputlookup` command
- For each pair of index and sourcetype, form the expected entity name in TrackMe (by default, `<index>:<sourcetype>`)
- Verify if this entity already exists in TrackMe
- Add to a search logic that will push entities as they are expected by the discovery process
- Finally, execute the search logic, which pushes entities as needed and expected

We will call the following command:

Replace the tenant name, in our case called secops

```
| inputlookup ds_expected.csv
| trackmepushdatasource search_type=tstats tenant_id=secops show_search_query=True␣
↪show_search_results=True pretend_latest="-24h" component="dsm"
```

Depending on the results, the command returns the list of added entities, rejected records (if any), and other useful information:



In TrackMe, these entities are now visible and in a red state:



As soon as trackers are executed, and if these sources are active and within the scope of the trackers, these entities will appear as green in TrackMe, assuming their status is healthy:

Give it some time; it may take a while before the entities are updated in TrackMe, depending on the conditions.



Finally, save this as a report and schedule it according to your preferences, for instance, once per day:





You're done! Any new source added to the lookup table will be pushed to TrackMe as expected.

7.24.2 Pushing Expected Hosts to TrackMe Data Hosts Monitoring (splk-dhm)

The process for pushing expected hosts is very similar, with the difference that typically you will only push the name of the host as it is expected to appear in Splunk.

In our example, we have a Splunk lookup table containing the list of expected hosts, based on a list of hostnames:

```
| inputlookup dh_expected.csv
| fields host
```



We have a Virtual Tenant called `endpoints` that is active and already contains a number of entities, some of which might already be part of the expected sources, but some might not:



To push expected hosts to TrackMe from the lookup file, we will call the streaming command `trackmepushdatasource`, which will:

- Parse the records resulting from the `inputlookup` command
- For each host, form the expected entity name in TrackMe (by default, `key:host|<host>`)
- Verify if this entity already exists in TrackMe
- Add to a search logic that will push entities as they are expected by the discovery process
- Finally, execute the search logic, which pushes entities as needed and expected

We will call the following command:

Replace the tenant name, in our case called endpoints

```
| inputlookup dh_expected.csv
| trackmepushdatasource search_type=tstats tenant_id=endpoints show_search_query=True␣
↪show_search_results=True pretend_latest="-24h" component="dhm"
```

Depending on the results, the command returns the list of added entities, rejected records (if any), and other useful information:



In TrackMe, these entities are now visible and in a red state. Note that for dhm, the list of indexes and sourcetypes will appear as an empty list for now:



As soon as trackers are executed, and if these sources are active and within the scope of the trackers, these entities will appear as green in TrackMe, assuming their status is healthy:

Give it some time; it may take a while before the entities are updated in TrackMe, depending on the conditions.



Finally, save this as a report and schedule it according to your preferences, for instance, once per day:





You're done! Any new host added to the lookup table will be pushed to TrackMe as expected.

7.24.3 Troubleshooting the Command trackmepushdatasource

If you encounter issues with the command `trackmepushdatasource`, you can use the following search to access the logs:

```
index=_internal sourcetype=trackme:custom_commands:trackmepushdatasource
```



7.24.4 Annexes for Pushing Expected Sources

Annex A: Command trackmepushdatasource Arguments

The following table describes all available arguments for the `trackmepushdatasource` command:

Argument	Required	Default	Description
tenant_id	Yes	None	The tenant identifier
component	Yes	None	The component to use (dsm or dhm)
search_type	Yes	None	The type of search to perform (tstats or raw)
show_search_query	No	False	If true, includes the search query in the summary output
show_search_results	No	False	If true, includes the search results in the summary output
pretend_latest	No	-24h	Relative time value in Splunk format for data_last_time_seen

Annex B: Controlling the Expected Sources Break by Logic for splk-dsm

For expected data sources, if the TrackMe tracker logic includes a specific break by logic, you can submit the value for the object accordingly, which the command will handle automatically.

Example: We use an additional break by logic with an indexed field called `cribl_env`:

```
| inputlookup ds_expected.csv
| fields index, sourcetype, cribl_env
| eval object = index . ":" . sourcetype . ":" . "|key:cribl_env|" . cribl_env
| trackmepushdatasource search_type=tstats tenant_id=secops show_search_query=True␣
↪show_search_results=True pretend_latest="-24h" component="dsm"
```

Annex C: Controlling the Expected Host Metadata for splk-dhm

For expected hosts, you can control the metadata for the host by submitting the host value with the expected metadata key for the object accordingly, which the command will handle automatically.

Example: We use a custom host metadata called `forwarder` instead of the default host metadata:

```
| inputlookup dh_expected.csv
| fields host
| eval host = "key:forwarder|" . host
| trackmepushdatasource search_type=tstats tenant_id=endpoints show_search_query=True␣
↪show_search_results=True pretend_latest="-24h" component="dhm"
```

7.25 CMDB Lookup Integration

7.25.1 Introduction to the CMDB Lookup Integration in TrackMe

The CMDB Lookup integration was introduced in TrackMe Version 2.0.57 and provides the following services:

- In the TrackMe user interface, quick access to the CMDB Lookup feature to provide context and third-party information related to a TrackMe entity
- TrackMe administrators can define a flexible CMDB external integration with dynamic token replacements
- The CMDB Lookup search can use any kind of Splunk Processing Language (SPL) statement, such as searching in a CSV or KVStore based lookup, or calling any Splunk command
- The configuration is made at the system level per TrackMe component, and can be fine-tuned per tenant relying on token variables replacements
- Finally, the integration relies on the built-in TrackMe command `trackmesplkcmdb` which processes and executes the CMDB Lookup logic

7.25.2 CMDB Lookup Feature Overview in the TrackMe User Interface

The TrackMe CMDB Lookup feature is available to TrackMe users by clicking on the CMDB icon in the Tabulator:



The CMDB Lookup integration screen opens and TrackMe automatically executes the integration logic:



In the background, TrackMe executes the CMDB logic via a built-in SPL command and using information from the component, tenant and entity, example:



The CMDB Integration logic is defined at the system-wide configuration level:

For instance:

```
| inputlookup my_cmdb where (index="$data_index$" AND sourcetype="$data_sourcetype$")
```

TrackMe automatically expands and replaces the token values from the entity data.

Any data returned by the CMDB integration will be part of the SPL results in the CMDB Lookup screen.

7.25.3 Configuring the CMDB Lookup Search Logic

TrackMe administrators can configure the CMDB Lookup integration search at the system level, per TrackMe component. Access:

- "Configure" / "spk-general" / Scroll down to the CMDB integration searches



The CMDB Lookup search template is specific per component and contains any number of token variables to be replaced. For instance, for the splk-dsm component, it is by default:

```
| inputlookup my_cmdb where (index="$data_index$" AND sourcetype="$data_sourcetype$")
```

This default template does the following logic:

- A Splunk lookup transform named "my_cmdb" is available in Splunk and in the context of the TrackMe application (you can of course use your own lookup name!)
- Several filters defined as tokens will be replaced automatically by TrackMe. In the default template, it expects a field describing the index as well as a field describing the sourcetype
- The lookup then renders any number of fields and information that is relevant for your context, such as the owner, contact information, etc.

The CMDB Lookup search integration can also use any SPL command and is not necessarily restricted to the use of a Splunk lookup.

In some specific contexts, you could therefore use a custom command of yours, or another SPL command that interacts with a third party such as ServiceNow, Jira, or your ITSM tool.

Token Variables Replacements

When TrackMe calls the `trackmesplkcmdb` command, the following logic is executed:

- The command receives in input a list of arguments such as the component, the tenant identifier, and the object identifier
- The command retrieves the CMDB Lookup integration search template from the TrackMe configuration
- It then processes the token variables replacements accordingly and executes the search transparently
- Results are provided in a JSON format and rendered by the CMDB integration screen
- It also investigates the results and renders a "no results found" specific message if the search did not render any results for this entity

Calling the Command Manually

The custom command `trackmesplkcmdb` accepts the following arguments:

Argument	Description	Default	Mandatory
component	The TrackMe component. Syntax: **component=**. Matches pattern: ^(?:dsm|dhm|mhm|cim|flx|wlk)$	(No default value provided)	Yes
tenant_id	The tenant identifier. Syntax: **tenant_id=**	None	Yes
object	The TrackMe object value. Syntax: **object=**	(No default value provided)	No
object_id	The TrackMe object identifier. Syntax: **object_id=**	(No default value provided)	No

Notes: TrackMe uses the object_id argument which is in fact the KVStore key identifier of the record. When calling the command manually, you can call object_id or object, but not both.

Troubleshooting the CMDB Integration Lookup Feature

TrackMe logs the activity of the command `trackmesplkcmdb` in the following sourcetype:

```
index=_internal sourcetype=trackme:custom_commands:trackmesplkcmdb
```

If the search fails for any reason, an exception would be raised and logged. The output of the exception will also be rendered in the JSON results in the CMDB integration screen.

Finally, you can also observe the runtime performance and the search that TrackMe executed:



Disabling the CMDB Integration for a Given TrackMe Tenant

You can disable the CMDB lookup integration for an existing Virtual Tenant by editing the tenant account preferences:

- "Configure" / "Virtual Tenants preferences"

Then edit the Virtual Tenant account you want to disable the feature for and scroll down to the option:





When disabled, TrackMe will not show the CMDB integration icon anymore for that Virtual Tenant.

7.26 Elastic sources for feeds tracking

7.26.1 Introduction to Elastic sources

Elastic sources feature versus Hybrid Trackers in TrackMe V2

- **In TrackMe V2 and in a large number of use cases, the concept of Elastic Sources is superseded by Hybrid Trackers**
- Indeed, an Elastic Source creates and maintains a single TrackMe entity, where Hybrid Trackers can manage from a few to tens of thousands of entities
- However, there are also use cases where Elastic Sources can be required and can be very valuable to meet your requirements or to address specific use cases such as tracking lookups and more

What does the Elastic sources feature do?

- The Elastic sources feature provides a built-in workflow to create virtual data sources based on any constraints and any Splunk language
- In TrackMe V2, the target can be the **local Search Head tier**, as well as any **remote Splunk deployment or instance** using TrackMe's built-in remote search capabilities
- Elastic Sources allows extending TrackMe feeds tracking capability in a rich and flexible way
- Elastic Sources can be based on `tstats`, `raw`, `from (datamodel and lookup)` and `mstats` searches

Some examples of use cases for Elastic Sources:

- Creating a Virtual entity which is the combination of different indexes, sourcetypes, sources and so forth with additional search filters, using indexed or search time extracted fields
- Creating a Virtual entity based on a Splunk search language not supported by Hybrid Trackers (`from` which allows calling lookups or datamodels)
- Create a Virtual entity based on a `raw` search, as opposed to `tstats` used typically on Hybrid Trackers for performance purposes, and where the use case requires using search time extracted only fields

- Tracking specific Splunk objects such as lookup content over time (CSV based lookup or KVstore), to detect if the lookup is being updated and also apply Machine Learning based outliers detection on the number of records

Shared Elastic versus Dedicated Elastic:

- An Elastic Source can be added to the **Shared tracker**, or created as an independent **Dedicated Tracker**
- Shared Elastic Sources are orchestrated by the Elastic Sources shared tracker
- Dedicated Elastic Sources are orchestrated by their own scheduled report
- The Shared Elastic Sources tracker processes searches in a multithreading parallel way (the max number of concurrent searches is configurable at the TrackMe system level and can be overridden on a per tenant basis)
- When the Shared Elastic Sources tracker reaches its maximal run time (which is automatically calculated depending on its cron schedule and time margin configuration), it stops its execution to avoid generating skipping searches
- On the next iteration, the Shared Elastic Sources tracker will resume its execution starting with entities which have not been inspected previously (but the oldest age of the inspection date)
- On the other hand, Dedicated Elastic trackers are fully independent scheduled reports, their life cycle remains orchestrated by TrackMe but these run independently from each other

7.26.2 Accessing the Elastic source creation UI

First, let's expose how to access the Elastic sources interface, from the data sources tab in the main UI, click on the `Elastic Sources` button:



The following screen appears:



7.26.3 Elastic source example 1: raw search with search time only extracted fields

Hint

tstats versus raw

- These steps are in fact the same whenever you use a raw search, or a tstats based search
- In the first case and in the example below, we use a raw search to benefit from search time extracted fields
- A tstats based search is how highly recommendable when possible, performances are infinitely better than raw searches, and computing costs much lower

In this example, requirements are:

- To monitor the Web data coming from a specific Website (trackme-solutions.com)
- Fields that can be used are only available at search time, creating indexed time fields for this use case would be challenging and possibly overkill for this edge use case (assuming we use tstats Hybrid Trackers globally)
- The volume of data is relatively low, allowing efficient searches to be performed

The following search constraints provides access to the data we need:

```
(index=webserver site="trackme-solutions.com")
```

We will create a new Shared Elastic source in a just a couple of easy steps:

Step 1: define name of the entity

- This defines the name of the entity, stored as the value for `object` and `alias` in TrackMe's collections (you probably want something meaningful)
- Also define the target, `local` means that we looking at data searchable from the local Search head tier, `remote` would target a remote account previously configured



Step 2: choose the language

- You can choose between different options, in our case we will use `raw` because we want to use search time extracted fields



Then we define the search constraint:

- The search constraint should target valid filters, allowing the search to be as efficient as possible and targeting the data we want to monitor



Hint

Defining the search constraint

- The search constraint is meant to be the list of filters providing access to the data you want to monitor
- This can be a simple constraint, or a complex one using OR and AND operators, different levels of parentheses, etc
- However, it is not meant to be the place where you would perform complex regular expression extractions for this instance, this should happen at the props and transforms level instead

In step 3, you should define proper values for index and sourcetypes:

- These are virtual values, they do not impact the search itself but are used to classify the entity in the UI
- You can choose anything you want, but it is recommended to use values that are meaningful to you



Finally in Step 4, define the earliest and latest time ranges:

- These are relevant for both Shared and Dedicated Sources
- The values influence how performing this search will be, especially with raw searches
- In this example, we reduce the earliest to -1h, latest should be in the future to be able to detect events abnormally defined in the future



Finally, after the simulation, we validate the creation as Shared or Dedicated:


Elastic sources definition: use this interface to create and manage elastic sources
5. Test and Review:
Test the Elastic source execution
Simulate the search
simulation_results
object
data_eventcount
data_first_time_seen
Success, you can now add this new source to the shared tracker or as a dedicated tracker.
6. Validate the creation
Promote to Live: the Elastic source will be created and tracked
Add to the shared tracker
Add as a new dedicated tracker
Cancel
Back


Elastic sources definition: use this interface to create and manage elastic sources
5. Test and Review:
Test the Elastic source execution
Simulate the search
simulation_results
object
data_eventcount
data_first_time_seen
Success, you can now add this new source to the shared tracker or as a dedicated tracker.
6. Validate the creation
Promote to Live: the Elastic source will be created and tracked
Add to the shared tracker
Add as a new dedicated tracker
Cancel
Back



7.26.4 Elastic source example 2: tracking lookups update and number of records

In this example, we will demonstrate how we can monitor the content of a Splunk lookup, KVstore or CSV based lookup, and get alerted if the lookup is not updated as expected.

Lookups are very useful in Splunk, we use these all the time as intermediate storage, for enrichment purposes, etc.

However, and this is especially true for security use cases, being capable of detecting if a lookup used in other use cases has stopped being updated or maintained is a critical requirement.

In our example, we have a KVstore based lookup called `cybops_net_traffic_monitor` which is updated on a regular basis and contains some associations and pre-calculations for network devices:



We will use the following Elastic Source definition to monitor our lookup, and we will leverage the exec_time field to define the _time value:

```
| from lookup:cybops_net_traffic_monitor | eval _time=exec_time
```



7.26.5 Administering Shared Elastic Sources

Shared Elastic sources are orchestrated by the Shared Elastic tracker, this tracker is automatically created along with the Virtual Tenant creation. (if the component splk-dsm is enabled)

The tracker is called:

- trackme_dsm_shared_elastic_tracker_tenant_<tenant_id>

The tracker is a scheduler wrapper which calls a Python engine:

```
| trackmeelasticexecutor tenant_id="<tenant_id>" component="splk-dsm"
```

You can find its execution logs using the following search:

```
index=_internal sourcetype=trackme:custom_commands:trackmeelasticexecutor
```

The Shared Elastic tracker loads entities to be processed from a KVstore, the following search shows the content of the KVstore: (update the tenant_id)

```
| inputlookup trackme_dsm_elastic_shared_tenant_<tenant_id>
```

Its execution is influenced by the following configuration item(s), go in Configuration / splk-general:

Table 16: Shared Elastic tracker configuration

Option	Purpose	Default
Concurrent searches Elastic	System level number of parallel concurrent searches for Shared Elastic sources, this can be overridden on a per tenant basis using max_concurrent_searches on the Shared Elastic tracker	3

You can review the performance run time of the Shared Elastic Tracker using the deep dive dashboard:



When the job starts, the following message is shown:

```
2023-11-09 22:21:11,840 INFO trackmeelasticexecutor.py generate 426 tenant_id="09-
↪demo-elastic", component="splk-dsm", report="trackme_dsm_shared_elastic_tracker_
↪tenant_09-demo-elastic", Elastic Sources shared job started, max_concurrent_
↪searches=3, margin_sec=60
```

This message indicates the current number of max concurrent searches, as well as the time margin in seconds to calculate the max run time of the full job.

The max concurrent job is first defined at the global level as explained above, but it can also be overridden on a per tenant basis adding the argument in the search definition:

```
**Syntax:** **max_concurrent_searches=****
**Description:** The max number of searches to be executed in parallel, if set to a␣
↪different value than the system default, this value wins.
```

The margin in seconds defines the time in seconds to be deducted from the total cron translation in seconds, for instance if the job is scheduled to run every 5 minutes, the cron sequence is 300, the job will interrupt at 300-<margin_sec>, this value can be overridden on a per tenant basis by adding the argument:

```
**Syntax:** **margin_sec=****
**Description:** The time in seconds used as a margin when calculating the max␣
↪runtime depending on the cron schedule.
If the search is triggered every 5 minutes, the max runtime will be 5 minutes less␣
↪the margin_sec value.
```

When the job inspects an entity to be processed, it will show the following message: (example)

```
2023-11-09 22:32:28,903 INFO trackmeelasticexecutor.py process_elastic_object 176␣
↪tenant_id="09-demo-elastic", component="splk-dsm", report="trackme_dsm_shared_
↪elastic_tracker_tenant_09-demo-elastic", processing elastic object, object=
↪"elastic:local:lookup:cybops_net_traffic_monitor", last inspection with tracker_
↪runtime="2023-11-09 22:26:57", elastic_report_root_search="| from lookup:cybops_net_
↪traffic_monitor | eval _time=exec_time
| eventstats max(_time) as indextime | eval _indextime=if(isnum(_indextime), _
↪indextime, indextime) | fields - indextime
| eval host=if(isnull(host), "none", host)
| stats max(_indextime) as data_last_ingest, min(_time) as data_first_time_seen, max(_
↪time) as data_last_time_seen, count as data_eventcount, dc(host) as dcount_host
| eval latest_eventcount_5m=data_eventcount
| eval object="elastic:local:lookup:cybops_net_traffic_monitor", data_index="lookups",
↪ data_sourcetype="cybops_net_traffic_monitor" | `trackme_elastic_dedicated_tracker(
↪"09-demo-elastic")`
| eval tenant_id="09-demo-elastic"
| stats count as report_entities_count by tenant_id
| `register_tenant_component_summary(09-demo-elastic, dsm)`"
```

Once the search has been executed, TrackMe shows is summary result and its run time information:

```
2023-11-09 22:32:57,742 INFO trackmeelasticexecutor.py process_elastic_object 245␣
↪tenant_id="09-demo-elastic", component="splk-dsm", object=
↪"elastic:local:lookup:cybops_net_traffic_monitor", report="trackme_dsm_shared_
↪elastic_tracker_tenant_09-demo-elastic", Entity search successfully executed,␣
↪status="success", run_time="28.839"
```

If the system reaches the max number of concurrent searches (Splunk wise), TrackMe will re-attempt automatically for a certain number of times, which is visible with a warning message:

```
2023-11-09 22:46:54,302 WARNING trackmeelasticexecutor.py process_elastic_object 265␣
↪tenant_id="09-demo-elastic", component="splk-dsm", report="trackme_dsm_shared_
↪elastic_tracker_tenant_09-demo-elastic", temporary search failure, current_failures_
↪count=4, max_failures_count=24, exception="HTTP 503 Service Unavailable -- b'{
↪"messages":[{"type":"WARN","text":"Search not executed: The maximum number of␣
↪concurrent historical searches on this instance has been reached., concurrency_
↪category=\\"historical\\", concurrency_context=\\"instance-wide\\", current_
↪concurrency=30, concurrency_limit=30","help":""}]}'"
```

Once all entities to be processed have been processed effectively, the job will show the following final message which includes the total run time: (which runtime is shown in the deep dive dashboard)

```
2023-11-09 22:53:18,138 INFO trackmeelasticexecutor.py generate 535 tenant_id="09-
↪demo-elastic", component="splk-dsm", report="trackme_dsm_shared_elastic_tracker_
↪tenant_09-demo-elastic", Elastic Sources shared job successfully executed, status=
↪"success", run_time="82.069", entities_count="6"
```

7.26.6 Administrating Dedicated Elastic Sources

Dedicated Elastic trackers are independent scheduled reports, their life cycle is orchestrated by TrackMe but they run independently from each other.



Each tracker is composed by two reports, the tracker which is scheduled and the wrapper which is called by the tracker:



The tracker calls the trackmetrackerexecutor command:



Execution logs can be found with the following search:

```
index=_internal sourcetype=trackme:custom_commands:trackmetrackerexecutor
```

Similarly, use the deepdive performance dashboard to review the tracker performance over time:



TrackMe stores dedicated Elastic trackers main Metadata in the following KVstore:

```
| inputlookup trackme_dsm_elastic_dedicated_tenant_<tenant_id>
```

7.27 TrackMe CI/CD management (TCM)

Minimal version required: 2.0.36

- The following documentation and capabilities require version 2.0.36 and later

- It also requires an additional package called TCM, which stands for the TrackMe Configuration Manager
- This package is provided to all TrackMe users at the following URL: https://downloads.trackme-solutions.com/TA-trackme-conf-manager



7.27.1 Introduction to the TrackMe Configuration Manager

In some contexts, it is not allowed to perform any kind of configuration in a Production environment context, and all configuration need to be performed first in a non Production environment, published to a versioning source control and finally released in a strict manner to the target environment(s).

TrackMe is a highly flexible and dynamic solution which implies on the fly creation of knowledge objects and associated content, generally stored in KVstores.

Therefore, typical approaches that work for more basic applications cannot be transposed to complex applications such as TrackMe, to achieve these goals we provide a configuration manager tool called TrackMe Configuration Manager. (TCM)

7.27.2 Downloading TCM

The TCM package is not available on Splunk Base, it can be freely downloaded at the following URL on our download Website:

- https://downloads.trackme-solutions.com/TA-trackme-conf-manager

We do not publish this package in Splunk Base on purpose, notably to allow Splunk Cloud customers to repackage TCM (as described in the next sections) and publish it to Splunk Cloud easily.

7.27.3 Principles

The TCM workflow relies on the following logic:

In non-Production:

- In the non-Production environment, the TCM is deployed and running in **receiver** mode
- TrackMe is configured via a built-in option to enable sending **transactions** to the TCM
- When an admin-level operation is made in non-Production, such as creating a new Virtual Tenant, the relevant transaction is written to the TCM JSON journal file
- The TCM journal is then taken in charge by your CI/CD pipeline, packaged with your TCM release, and published to your versioning control

In Production:

- The TCM is deployed and configured in **producer** mode
- The built-in TCM scheduled backend is enabled
- When the Python backend is executed, **transactions** from the TCM journal are read and verified
- If the transaction has not been processed yet, TCM runs the associated operation by performing the exact same REST API calls to TrackMe that were made in non-Production
- Once the transaction has been performed, a record is written to the TCM resilient KVstore; transactions are uniquely associated with a transaction MD5
- At the next iteration, TCM knows that the transaction is done already and does not need to be processed again

7.27.4 Configuring TrackMe and TCM in non-Production

Note about SHC in non-Production

- Transactions are written to a local file in the TCM application, which would not be replicated amongst the members of the SHC
- Therefore, the TCM can be used in non-Production in an SHC cluster; however, you will need to manage the synchronization of the journal file
- Ideally, you should run TCM in a standalone non-Production environment to avoid having to deal with this question

Non-Production workflow

In non-Production, which means where the configuration of TrackMe is made, the TCM is deployed in its default configuration, which acts in **receiver** mode by default.

TrackMe needs to be instructed to send the transactions to TCM:



Which corresponds to the following option in TrackMe configuration file:

```
[trackme_general]
# Enable the TrackMe Config Manager (TCM) and sets the mode, the TCM is used for CI/
↪CD purposes, the receiver mode sends the transaction to TCM, the replay mode␣
↪replays transactions from TCM
enable_conf_manager_receiver = 0
```

By enabling this option, when TrackMe performs an admin level operation which creates and manages contents such as Virtual Tenants, TrackMe performs a REST call to the TCM API:

```
https://<local splunkd uri>/services/trackme_conf_manager/v1/conf_manager_receiver
```

The TCM API receives the following information:

- `transaction_request`: The request payload for the TrackMe API call, these are all the instructions underneath a REST call made to TrackMe's API endpoints
- `transaction_http_mode`: The requested HTTP mode for the transaction (GET, POST, DELETE)
- `transaction_http_service`: The TrackMe API service that was requested for this transaction, this is the TrackMe API endpoint that was contacted for this operation

TrackMe REST API logs the TCM forwarding actions:

success:

```
trackme_send_to_tcm was successfully executed
```

Example:

```
index=_internal sourcetype=trackme:rest_api trackme_send_to_tcm
```

```
2023-06-05 21:55:38,258 INFO trackme_rest_handler_splk_replica_trackers_admin.py post_
↪replica_tracker_create 224 trackme_send_to_tcm was successfully executed
```

In case of failures:

```
trackme_send_to_tcm has failed with exception=<exception>
```

On the file-system, the TCM writes transactions to a JSON journal file in local:

```
/opt/splunk/etc/apps/TA-trackme-conf-manager/local/trackme_conf_manager_transactions_
↪journal.json
```

Each transaction is a JSON object which represents the REST call to be performed depending on the actions requested in TrackMe, through the UI and the REST API:

Example:


```
"9c2f7f8264a86913f34a97e161b128e8": {
    "transaction_request": {
        "tenant_desc": "SIEM",
        "tenant_name": "feeds-tracking",
        "tenant_roles_admin": [
            "trackme_admin"
        ],
        "tenant_roles_power": [
            "trackme_power"
        ],
        "tenant_roles_user": [
            "trackme_user"
        ],
        "tenant_owner": "svc-trackme",
        "tenant_idx_settings": "{\"trackme_summary_idx\": \"trackme_summary\", \
↪"trackme_audit_idx\": \"trackme_audit\", \"trackme_notable_idx\": \"trackme_notable\
↪", \"trackme_metric_idx\": \"trackme_metrics\"}",
        "tenant_dsm_enabled": true,
        "tenant_dsm_sampling_obfuscation": "disabled",
        "tenant_dhm_enabled": true,
        "tenant_dhm_alerting_policy": "track_per_host",
        "tenant_mhm_enabled": true
    },
    "transaction_http_mode": "post",
    "transaction_http_service": "/services/trackme/v2/vtenants/admin/add_tenant",
    "ctime": 1686000734.0977793
}
```


A transaction is identified by a unique MD5 hash, this same hash will be used by TCM on the producer side (therefore in Production) to identify transactions which have been processed already, and transactions which are pending.

7.27.5 Configuring TrackMe and TCM in Production

Note about SHC in Production

- SHC is fully supported for all TCM replay transactions

TCM packaging and deployment

In Production, therefore the target environment where transactions will be replayed, the TCM must be deployed in Producer mode:

Through the UI:



Which corresponds to the following configuration:

ta_trackme_conf_manager_settings.conf

```
[trackme_general]
conf_manager_role = producer
```

TCM packaging:

Depending on your context:

- If you are a **Splunk Cloud** customer, you need to include the journal file in the default directory, package the application release and deploy it through Splunk ACS (through the API or UI), once Appinspect is passed, TCM is ready
- If you are a **Splunk Enterprise** customer, you can equally publish the journal in default or local, but local has precedence over the default directory

TrackMe configuration

TrackMe does not require any configuration for TCM.

TCM acts independently once it is deployed and configured, and will interact with TrackMe as needed to replay the transactions and apply your settings accordingly.

However, you can disable all admin levels operations through the UI via the following settings (to avoid allowing any admin to generate a conflict between changes) and remove any related settings from TrackMe's user interfaces:



As well, note that forwarding to TCM must be disabled in the Production context. (which is the default)

Viewing the transaction journal

You can access the transaction journal via the following command:

```
| trackmeconfmanager mode=get
```

Example:



Checking transactions statuses

You can check the status of transactions via the following command:

```
| trackmeconfmanager mode=simulation
```

A transaction which has not been completed yet a `transaction_info` message explaining the action which will be taken.

A transaction which has been completed already shows a `transaction_info` message: "This transaction has been processed already, the Conf Manager will not attempt to process it again".

In the following example, all transactions have been proceeded:



In the following example, transactions are pending and would be proceeded:



Enabling the scheduled report trackme_conf_manager_producer

The TCM packages provides a built-in scheduled report which is disabled by default; the report runs the following command:

```
| trackmeconfmanager mode=live
```

This is the process which replays transactions and configures TrackMe according to the transaction journal provided by your CI/CD pipeline.

The process works as follows:

- For each transaction, the Python backend verifies if the transaction can be found in the resilient

KVstore `kv_trackme_conf_manager`:

```
| inputlookup trackme_conf_manager | eval key=_key
```

- If the transaction has not been processed, TCM performs the associated call to the TrackMe REST API
- If the transaction has been processed, then there are no actions to be taken

The TCM transaction replay backend logs its activity into the following logs:

```
index=_internal sourcetype=trackme_conf_manager:trackmeconfmanager
```

7.27.6 Troubleshooting TCM

TCM receiver REST API

When a transaction is sent from the non-Production environment to TCM, activity is logged here:

```
index=_internal sourcetype=trackme_conf_manager:rest_api
```

example:

```
2023-06-05 21:54:12,145 INFO trackme_conf_manager_rest_handler.py post_conf_manager_
↪receiver 216 conf manager received configuration, resp_dict="{'transaction_request
↪': {'tenant_id': 'replica-demo', 'source_tenant_id': 'cim-demo', 'component': 'cim',
↪ 'root_constraint': 'object=*', 'update_comment': 'No comment for update.', 'owner
↪': 'svc-trackme'}, 'transaction_http_mode': 'post', 'transaction_http_service': '/
↪services/trackme/v2/splk_replica_trackers/admin/replica_tracker_create'}"
2023-06-05 21:54:12,158 INFO trackme_conf_manager_rest_handler.py post_conf_manager_
↪receiver 284 {'response': 'transaction received with␣
↪md5=5df36dfae06b237408a0cecd30e87015 and successfully written to the journal',
↪'journal': '/opt/splunk/etc/apps/TA-trackme-conf-manager/local/trackme_conf_manager_
↪transactions_journal.json'}
```

TCM transaction replay (command trackmeconfmanager)

Activity of the trackmeconfmanager command is logged here:

```
index=_internal sourcetype=trackme_conf_manager:trackmeconfmanager
```

example:

```
2023-06-06 09:16:29,916 INFO trackmeconfmanager.py generate 156 verifying transaction␣
↪status, transaction_md5="ca315b5f58e2766e60bf231e86a5f551"
2023-06-06 09:16:29,916 INFO trackmeconfmanager.py generate 162 transaction was not␣
↪found in the KVstore and will be processed in live mode
2023-06-06 09:16:41,030 INFO trackmeconfmanager.py generate 219 transaction was␣
↪registered successfully to the KVstore with key="ca315b5f58e2766e60bf231e86a5f551"
```

7.28 Maintenance mode & knowledge database

7.28.1 Introduction to the maintenance mode features

Hint

The maintenance features are composed of two main components:

- **The Maintenance Mode**, which allows silencing TrackMe alerts for a given period of time
- **The Maintenance Knowledge DataBase**, which allows to store maintenance information associated with the maintenance mode actions, or independently, for audit purposes and also to influence SLA calculations accordingly

Tenants scope selection from Trackme 2.0.97

- Since TrackMe release 2.0.97, you can define the scope of the Virtual Tenants when enabling the maintenance mode.
- This allows defining which Virtual Tenants are affected by the application of the maintenance mode.
- Tenants scope is also automatically taken into account in the maintenance mode knowledge database and for SLA calculation purposes.

7.28.2 TrackMe permissions requirements

To access and manage the Maintenance Mode features, you need to have the following permissions:

- TrackMe admin (capability: trackmeadminoperations)

Regarding the Maintenance Knowledge DataBase, you need to have the following permissions to access the UI in read only:

- TrackMe user (capability: trackmeuseroperations)

To operate changes, adding new Maintenance records or manage existing records, you need to have the following permissions:

- TrackMe admin (capability: trackmeadminoperations)

7.28.3 Maintenance Mode

You can access the Maintenance Mode UI from the maintenance menu on top of TrackMe:



Enabling the Maintenance Mode

You can enable the Maintenance Mode to start immediately or at a given date and time:



If the start date is in the future, TrackMe schedules the start automatically:



When the start period is reached, or if you selected a start for now, TrackMe shows the Maintenance Mode enabled:



You can disable the Maintenance Mode at any time:



Detection and management of the Maintenance Mode status is driven by a Tracker called `trackme_maintenance_mode_tracker`.

7.28.4 Maintenance Knowledge DataBase

Accessing the Maintenance Knowledge DataBase

The interface for the Maintenance Knowledge DataBase is also accessible from the maintenance menu on top of TrackMe:



Maintenance Knowledge DataBase behaviour

Maintenance records are automatically created and updated when enabling a scheduled or immediate maintenance period via the Maintenance Mode user interface and REST API endpoints:

- When the maintenance mode is enabled and then disabled, TrackMe automatically creates and maintains a record in the Maintenance Knowledge DataBase as a planned operation.
- Associated periods of times are excluded from SLA calculations in TrackMe.
- You can also optionally register unplanned operations for your own auditing
- Only planned operations are excluded by default from SLA calculations, but you can use the global setting maintenance_kdb_exclusion_behaviour to manage this behaviour.
- Both types of Maintenance periods can be registered via this UI, or via REST API and the maintenance_kdb resource group endpoints (See TrackMe Rest API Reference dashboard)

7.28.5 REST API endpoints for the Maintenance Mode and Maintenance Knowledge DataBase

Both features rely on different API endpoints, enabling their integration within and out of TrackMe, consult the REST API reference dashboard:


Results:
Time
Event
18/12/2023 09:27:02.347
{ [-]
python_function: get_check_global_maintenance_status
resource_api: services/trackme/v2/maintenance/check_global_maintenance_status
resource_curl_example: curl -u username:'https://myspluk:8089/services/trackme/v2/maintenance/check_global_maintenance_status -X "GET"
resource_desc: Check and return the maintenance mode status
resource_describe: ([+]
)
resource_group: maintenance
resource_mode: get
resource_spl_example: | trackme mode=get url="/services/trackme/v2/maintenance/check_global_maintenance_status"
}
Show as raw text
18/12/2023 09:27:02.363
{ [-]
python_function: post_global_maintenance_enable
resource_api: services/trackme/v2/maintenance/global_maintenance_enable
resource_curl_example: curl -u username:'https://myspluk:8089/services/trackme/v2/maintenance/global_maintenance_enable -X "POST" -d '{\"maintenance_duration\": \"3600\", \"update_comment\": \"Enabling a TrackMe global maintenance for 1 hour of duration from now.\"}'
resource_desc: Enable global TrackMe maintenance mode
resource_describe: ([+]
)
resource_group: maintenance
resource_mode: post
resource_spl_example: | trackme mode=post url="/services/trackme/v2/maintenance/global_maintenance_enable" body="{'maintenance_duration': '3600', 'update_comment': 'Enabling a TrackMe global maintenance for 1 hour of duration from now.'}"
}
Show as raw text
18/12/2023 09:27:02.378
{ [-]
python_function: post_maintenance_disable
resource_api: services/trackme/v2/maintenance/maintenance_disable
resource_curl_example: curl -u username:'https://myspluk:8089/services/trackme/v2/maintenance/maintenance_disable -X "POST" -d '{\"update_comment\": \"API operations done, disabling global TrackMe maintenance mode.\"}'
resource_desc: Disable global TrackMe maintenance mode
resource_describe: ([+]
)
resource_group: maintenance
resource_mode: post
resource_spl_example: | trackme mode=post url="/services/trackme/v2/maintenance/maintenance_disable" body="{'update_comment': 'API operations done, disabling global TrackMe maintenance mode.'}"
}
Show as raw text

Try it yourself! (or click on a row of the table above)
| trackmegetkvtabdocs target="endpoints" | search resource_group="maintenance_kdb"
Results:
Time
Event
18/12/2023 09:31:12.725
{ [-]
python_function: post_maintenance_kdb_get_records
resource_api: services/trackme/v2/maintenance_kdb/maintenance_kdb_get_records
resource_curl_example: curl -u username:https://myspluk:8089/services/trackme/v2/maintenance_kdb/maintenance_kdb_get_records -X "POST" -d '{\"time_format\": \"human\"}'
resource_desc: Retrieve maintenance knowledge records in the database
resource_describe: ([+]
)
resource_group: maintenance_kdb
resource_mode: post
resource_spl_example: | trackme mode=post url="/services/trackme/v2/maintenance_kdb/maintenance_kdb_get_records" body="{'time_format': 'human'}"
}
Show as raw text

Time
Event
18/12/2023 09:32:02.064
{ [-]
python_function: post_maintenance_kdb_add_record
resource_api: services/trackme/v2/maintenance_kdb/admin/maintenance_kdb_add_record
resource_curl_example: curl -u username:https://myspluk:8089/services/trackme/v2/maintenance_kdb/admin/maintenance_kdb_add_record -X "POST" -d '{\"time_start\": \"2023-01-01T00:00\", \"time_end\": \"2023-01-01T01:00\", \"no_days_validity\": 0, \"time_format\": \"%Y-%m-%dT%H:%M\", \"reason\": \"Planned maintenance\", \"type\": \"planned\", \"object\": \"object\", \"add_info\": \"Additional information\", \"update_comment\": \"API update\"}'
resource_desc: Add a new maintenance knowledge record in the database
resource_describe: ([+]
)
resource_group: maintenance_kdb/admin
resource_mode: post
resource_spl_example: | trackme mode=post url="/services/trackme/v2/maintenance_kdb/admin/maintenance_kdb_add_record" body="{'time_start': '2023-01-01T00:00', 'time_end': '2023-01-01T01:00', 'no_days_validity': 0, 'time_format': '%Y-%m-%dT%H:%M', 'reason': 'Planned maintenance', 'type': 'planned', 'object': 'object', 'add_info': 'Additional information', 'update_comment': 'API update'}"
}
Show as raw text
18/12/2023 09:32:02.080
{ [-]
python_function: post_maintenance_kdb_bulk_edit
resource_api: services/trackme/v2/maintenance_kdb/admin/maintenance_kdb_bulk_edit
resource_curl_example: curl -u username:https://myspluk:8089/services/trackme/v2/maintenance_kdb/admin/maintenance_kdb_bulk_edit -X "POST" -d '<newentries_json>'
resource_desc: Perform a bulk edit to one or more entries
resource_describe: ([+]
)
resource_group: maintenance_kdb/admin
resource_mode: post
resource_spl_example: | trackme url="/services/trackme/v2/splk_dsm/write/maintenance_kdb_bulk_edit" mode="post" body="<redacted_json>"
}
Show as raw text
18/12/2023 09:32:02.097
{ [-]
python_function: post_maintenance_kdb_check_expired
resource_api: services/trackme/v2/maintenance_kdb/admin/maintenance_kdb_check_expired
resource_curl_example: curl -u username:https://myspluk:8089/services/trackme/v2/maintenance_kdb/admin/maintenance_kdb_check_expired -X "POST"
resource_desc: Check and verify if a maintenance knowledge record is expired, if so, disable it
resource_describe: ([+]
)
resource_group: maintenance_kdb/admin
resource_mode: post
resource_spl_example: | trackme mode=post url="/services/trackme/v2/maintenance_kdb/admin/maintenance_kdb_check_expired"
}
Show as raw text
18/12/2023 09:32:02.118
{ [-]
python_function: post_maintenance_kdb_manage_record
resource_api: services/trackme/v2/maintenance_kdb/admin/maintenance_kdb_manage_record
resource_curl_example: curl -u username:https://myspluk:8089/services/trackme/v2/maintenance_kdb/admin/maintenance_kdb_manage_record -X "POST" -d '{\"record_id\": \"2345678901\", \"action\": \"disable\", \"update_comment\": \"API update\"}'
resource_desc: Manage an existing maintenance knowledge record in the database
resource_describe: ([+]
)
resource_group: maintenance_kdb/admin
resource_mode: post
resource_spl_example: | trackme mode=post url="/services/trackme/v2/maintenance_kdb/admin/maintenance_kdb_manage_record" body="{'record_id': '2345678901', 'action': 'disable', 'update_comment': 'API update'}"
}

REST API example: enable the maintenance mode for 1 hour

The following example immediately enables the maintenance mode for all tenants for a duration of 1 hour starting now:

```
curl -u username https://mysplunk:8089/services/trackme/v2/maintenance/global_
↪maintenance_enable -X "POST" -d "{\"maintenance_duration\": \"3600\", \"update_
↪comment\": \"Enabling a TrackMe global maintenance for 1 hour of duration from now.\
↪"}"
```

The following example does the same but restricts the scope to a tenant name "secops":

```
curl -u username https://mysplunk:8089/services/trackme/v2/maintenance/global_
↪maintenance_enable -X "POST" -d "{\"maintenance_duration\": \"3600\", \"update_
↪comment\": \"Enabling a TrackMe global maintenance for 1 hour of duration from now.\
↪", \"tenants_scope\": \"secops\"}"
```

REST API example: disable the maintenance mode

The following example disables the maintenance mode:

```
curl -u username https://mysplunk:8089/services/trackme/v2/maintenance/global_
↪maintenance_disable -X "POST" -d "{\"update_comment\": \"Disabling the TrackMe␣
↪global maintenance.\"}"
```

7.29 TrackMe App on SOAR: Automate and interact with TrackMe from Splunk SOAR

About TrackMe App on SOAR

- **TrackMe App on SOAR** is an application for **Splunk SOAR** that allows you to automate tasks and interact with TrackMe transparently from your SOAR environments.
- You can download TrackMe App on SOAR from our release website: https://downloads.trackme-solutions.com/trackme-app-on-soar or our GitHub repository: https://github.com/trackme-limited/trackme-app-on-soar
- With the TrackMe App on SOAR, you can automate various actions in TrackMe, such as retrieving TrackMe entities' realtime statuses, updating their key behavior parameters, or interacting with TrackMe features such as the Machine Learning capabilities.
- Working in collaboration with TrackMe, SOAR users can leverage the TrackMe App on SOAR to extend and enrich the workflow and fulfill any kind of sophisticated requirements.

7.29.1 Overview of the TrackMe App on SOAR

The TrackMe App on SOAR documentation can be consulted directly in SOAR once the application has been installed:



Overview of a TrackMe Asset in SOAR:



You can also consult the README from our GitHub repository:

- https://github.com/trackme-limited/trackme-app-on-soar/blob/main/README.md

7.29.2 SOAR TrackMe actions overview

See the TrackMe App on SOAR built-in documentation for a complete and up-to-date list of actions (or in the README from our GitHub repository):



Each action makes use of the TrackMe API endpoints to interact with TrackMe, and provide various options, you can for instance update the key parameters of TrackMe Entities:





7.29.3 SOAR TrackMe usage example

In the following example, we leverage the TrackMe App on SOAR to update a TrackMe entity's Acknowledgement status:



7.29.4 Forwarding TrackMe Notable Events to SOAR

The ideal and recommended configuration is to forward TrackMe Notable Events to Splunk SOAR using the Splunk App for SOAR Exports. The following documentation details step-by-step the best practices configuration.

Pre-requisites

Splunk App for SOAR Export

The Splunk App for SOAR export should be installed and configured in your Splunk environment so you can forward Splunk events to SOAR. See:

- https://splunkbase.splunk.com/app/3411

Splunk App for SOAR and Splunk/SOAR integration

You should also ideally have already configured Splunk SOAR to forward all events to your Splunk environment. This is notably valuable to allow having overlap in your SOAR forwarding rule and ensure that you never miss any event that should be forwarded to SOAR.

- https://splunkbase.splunk.com/app/6361
- https://docs.splunk.com/Documentation/SOARApp/1.0.71/Install/Configureremotesearch

Create a new Label in Splunk SOAR

You should create a new label for TrackMe Notable Events in SOAR:

- Name: TrackMe Notable



Create the Splunk report for TrackMe Notable Events

Start by creating a fresh new report that will be called in the Splunk App for SOAR Export:

- Name: SOAR Forward TrackMe Notable Events
- Share: the report **must** be shared globally so it can be called from the SOAR Export app
- Earliest / Latest: -60m / now (if we want to use overlapping with SOAR)
- code:

Hint

The following code is an example. You may need to adjust it to your specific environment and use case.

- Ensure to update the subsearch and the label if you choose a different label for TrackMe Notable Events.
- The subsearch technique is used in combination with the SOAR/Splunk integration and the uniqueness of event_id per event.
- With this technique, you ensure to have a safe workflow to avoid missing events in case of temporary issues on the Splunk side and/or delayed execution of the SOAR Export job.

```
index=trackme_notable tenant_id=* priority=*
``` Allows overlap and uses a subsearch benefiting from the SOAR/Splunk integration␣
↪to avoid missing events in case of temporary issues on the Splunk side and/or␣
↪delayed execution of the SOAR Export job ```
| search NOT [ search index="phantom_artifact" container_label="trackme notable" |␣
↪fields cef.event_id | rename cef.event_id as event_id | table event_id | format ]
``` anomaly_reason is a native list (mvfield). If the search runs in the TrackMe␣
↪namespace, TrackMe does this for you; otherwise, you need to handle it explicitly␣
↪```
| eval anomaly_reason=coalesce(anomaly_reason, 'anomaly_reason{}') | fields - anomaly_
↪reason{}
``` Note: properties fields depend on the context in TrackMe, such as the type of the␣
↪component. You do not necessarily need to include these, but doing so allows you to␣
↪decide to do so in the SOAR Export app field mapping at a later step ```
| table tenant_id, keyid, event_id, object, object_category, state, priority, anomaly_
↪reason, status_message, drilldown_link, properties*
| foreach * [ eval <<FIELD>> = mvdedup('<<FIELD>>') ]
| fields *
```


```



Create and configure the Event Forwarding Rule in SOAR Export

In the SOAR Export App, create a new Event Forwarding Rule:

- Name: SOAR Forward TrackMe Notable Events
- Label: TrackMe Notable
- Run every: 5m
- Earliest / Latest: -60m / now (if we want to use overlapping with SOAR)



Choose the severity and sensitivity levels that are relevant to your use case, and configure the field mapping:



Map the fields in the SOAR Export app. For TrackMe Notable Events, ensure to map the following fields: (Note: you can decide later on to include additional properties fields depending on your use cases)

- tenant_id
- keyid
- event_id
- object
- object_category
- state
- priority
- anomaly_reason
- status_message
- drilldown_link



Continue the creation, and if you already have TrackMe Notable events making it to Splunk, new containers will start to be created in SOAR:



CHAPTER

EIGHT

WHITE PAPERS:

8.1 TrackMe's White Papers

About TrackMe's white papers

- TrackMe is a powerful, flexible and large solution that can be used to monitor Splunk environments, from feeds tracking to the search Workload, Performance and infrastructure tracking, functional controls and more!
- With these white papers, tutorials and documentations, we aim at providing and sharing deep knowledge around our most valuable practices and use cases.
- This collection of articles will only get better over time!

8.1.1 Use Cases Demo

- *Use Case Demo: This is All About Feeds*
- *Use Case Demo: 360 Services Monitoring with TrackMe*
- *Use Case Demo: Fields Quality (CIM and non-CIM)*

8.1.2 TrackMe implementation

- *Running a TrackMe Proof of Concept*

For TrackMe beginners

- *QUICK START - Starting with TrackMe: (feed tracking quickstart)*

Feeds Tracking: monitoring Splunk Data availability and quality

- *Use TrackMe to detect abnormal events count drop in Splunk feeds*

Workload Tracking: monitoring and tracking the activity and behaviour of Splunk scheduled searches

- *Monitor Splunk Workload with TrackMe's Workload component*

Infrastructure Monitoring: monitoring and tracking Splunk components and infrastructure

- *Monitor Splunk Indexer Clusters*
- *Monitor Splunk Search Head Clusters*
- *Analyse log messages logging level to detect behaviour anomalies using TrackMe's Flex Object and Machine Learning Anomaly Detection*

Splunk Cloud-specific

- *Tracking Splunk Cloud SVC consumption in TrackMe*

Alerting and Notifications: implementing and designing your alerting and notification strategy

- *Using SLA alerting to build a 2-tier monitoring system*

8.1.3 TrackMe management

- *Backing up and Restoring TrackMe*
- *Performing mass operations in TrackMe*
- *Auto deletion or management of TrackMe entities*

8.2 Running a TrackMe Proof of Concept

Running a TrackMe Proof of Concept (POC) is surely the best way to demonstrate the key values of the product, and how it can help you tackle your monitoring challenges. This document aims to guide you through the main steps to run a successful POC.

- We stand by to help you at any time, so please do not hesitate to reach out to us if you need any assistance: contact@trackme-solutions.com
- TrackMe has different licensing modes, with no licenses TrackMe is running in Community Edition, which implies slight limitations on its features, in the context of the POC which can eventually lead to buying the product, restricting to the community Edition would be very unfortunate and a missed chance for you, and for us.
- Instead, contact us and we will be delighted to provide you a full temporary license, which can cover at least 60 days of usage, which could eventually be renewed for a further 30 days if the buying process is still ongoing.
- TrackMe is a very rich and powerful solution, you may not test every single feature or capability during the POC, but we recommend you to focus on the key features that are most relevant to your monitoring challenges.

8.2.1 Identifying the Key Use Cases

TrackMe can handle many different use cases, below is a non exhaustive list of the most common use cases that TrackMe can help you tackle:

- `Feeds tracking`: Monitor any kind of feeds making to Splunk, from the data source perspective (index / sourcetype / additional Metadata if needed), from the event host perspective, and more.
- `Scheduled search and Splunk Enterprise Security correlation searches`: Track the scheduled activity of your Splunk environment, detect scheduled anomalies and investigate capacity planning, TrackMe as your single pane of glass.
- `TrackMe Flex Objects, monitor your Splunk deployment, configuration, perform SIEM controls, anything!`: Use TrackMe Flex objects, locally or remotely, and monitor any layer of Splunk, from Search Head Cluster and Indexer Clusters health and activity, Knowledge bundle size, licence usages and literally anything you can think of.

8.2.2 Identify where to deploy TrackMe

Review TrackMe requirements and identify where to deploy TrackMe:

- Review the main installation documentation: *Installation of TrackMe*

- For the purpose of a POC, you may want to prioritise a deployment on a standalone server for simplicity, good candidates are usually a standalone Search Head that is already used for monitoring purposes, or the utility node running the Splunk Monitoring Console. (eg. DMC)
- If you are a Splunk Cloud customer, then there are not any questions and TrackMe will run on the ad hoc Search Head tier.
- You can always at a later time decide to migrate TrackMe to a definitive Search Head layer target.

8.2.3 Service account and permissions

- By default, TrackMe runs as the Splunk nobody user (eg. splunk-system-user), it is not strictly required to create a dedicated service account for TrackMe.
- However, it is recommended to create a service account for TrackMe as a good configuration practice, review: *Configuration*
- In the context of a POC, you may totally skip this step and save more time to focus on the use cases instead.
- If this is however a strong requirement and especially in some specific contexts, this can be done since day 1 easily by following the documentation above.

8.2.4 Roles Based Access Control (RBAC)

- TrackMe supports Roles Based Access Control, relying on Splunk native capabilities with our implementation.
- TrackMe comes with 3 builtin Splunk Roles: `trackme_admin`, `trackme_power`, `trackme_user`.
- TrackMe also requires its users to have TrackMe capabilities. (which are provided by the roles above)
- Review: *Role Based Access Control and ownership*

8.2.5 Install TrackMe

- Installing TrackMe is straightforward, Review the main installation documentation: *Installation of TrackMe*
- Ensure you are meeting application dependencies requirements
- Ensure you have defined and published TrackMe indexes in your indexing layer, notably for Splunk Enterprise customers. (in Splunk Cloud, this is automated once the application is deployed)
- Remember that TrackMe does not do anything at all once installed and until you start creating TrackMe Virtual Tenants and trackers, so it is fully safe to have TrackMe installed ahead of time.
- Also, pay attention to TrackMe new releases, we aim at publishing new releases at least once per month, fixing, enhancing and adding new features to enrich the product.
- We also recommend to install and enable the `TrackMe Configuration Manager` app (TCM), this allows registering each administrative action in TrackMe (such as the creation of tenants or trackers) and replay these eventually later on, this is also a good mean for you to understand how TrackMe works underneath!
- Review TrackMe TCM: *TrackMe CI/CD management (TCM)*

8.2.6 Register a TrackMe licence for the POC

- In the context of a POC, we advise you to contact us to get a temporary licence for an extended period of time.
- You can also generate a TrackMe trial licence for 30 directly in TrackMe. (requires your Search Head to have external traffic connectivity with our public licence API services, https/443 outgoing)
- Bellow are the different TrackMe offering and their features and restrictions:

TrackMe Enterprise Edition:

- Up to 6 Virtual Tenants limited
- Up to 8 remote deployments
- Splunk feeds tracking
- Up to 16 Hybrid Trackers
- Unlimited Elastic trackers
- Machine Learning Outliers detection for all components
- Splunk Workload
- Common Information Model compliance tracking (16 trackers)
- Flex Object Tracking (32 trackers)
- Premium support (24 hours SLA, 8.am to 8.pm UK time)

TrackMe Unlimited Edition:

- Unlimited number of Virtual Tenants
- Unlimited remote deployments
- Splunk feeds tracking
- Unlimited number of Hybrid Trackers
- Unlimited Elastic trackers
- Machine Learning Outliers detection for all components
- Splunk Workload
- Common Information Model compliance tracking
- Flex Object Tracking
- Premium support (24 hours SLA, 8h AM to 8h PM UK time)

TrackMe Community Edition:

- 2 Virtual Tenants limited
- 1 remote Splunk deployment limited
- Splunk feeds tracking
- 2 Trackers per component (6 total)
- Unlimited Elastic trackers
- Machine Learning Outliers detection (Splunk feeds)
- Best effort support, with no warranty

8.2.7 Remote deployment and multiple Search Head tiers

- TrackMe provides native capabilities to interact with any remote Splunk deployment or instance, from Search Head tiers to utility nodes such as the Cluster Manager, or Heavy Forwarders.
- When it comes to the TrackMe Workload component (monitoring of Splunk scheduling), we also use this feature to interact with the Splunk API for purposes such as the search versioning.
- Review: *Splunk Remote Deployments (splunkremotesearch)*
- For this, you should identify the targets, and get a bearer token created per target as well as satisfying basic networking requirements.

8.2.8 Design TrackMe tenants

- Ahead of the POC or during the first phases, you will want to consider how tenants and the purpose of each tenant for your TrackMe deployments.
- These decisions are usually influenced by your own technical / functional contexts, as well as the features you want to use in TrackMe.
- A tenant can run multiple TrackMe component (example: splk-dsm for Data source tracking and splk-flx)
- You can dedicate a TrackMe Virtual Tenant for a specific perimeter, or a specific team, or a specific use case, or a mix of all of these depending on your needs.
- Remember that you can experiment, a Virtual Tenant can be created, destroyed, disabled, etc!

A realistic example:



8.2.9 Design TrackMe at scale

- A best practice when creating Tenants and for Feeds tracking is to create Hybrid Trackers after the creation of the Virtual Tenants rather than when creating the Tenants.
- This allows more control and more flexibility, allowing to adopt scaling best practices and design performing and efficient Hybrid Trackers easily.
- Remember that you can leverage various Splunk techniques easily in TrackMe, such as using Splunk indexed fields to filter out or influence TrackMe entities definition and reflect your data pipelines easily.
- Review: *Large Scale Environment and Best Practices Configuration Guide*



- Use out of the box TrackMe tooling to monitor the performing and costs of your Hybrid Trackers easily:





8.2.10 Design your alerting strategy

TrackMe provides a flexible out of the box workflow when it comes to alerting:

- On a per tenant basis, you can create in a very simple step a TrackMe alert
- TrackMe alerts are designed to run automated TrackMe alert actions, such as generating `TrackMe Notable` events, performing `automated Acknowledgements` and run `SmartStatus` investigations.
- A TrackMe architecture best practice for alerting is to rely on TrackMe notable events rather than implementing third party notifications from TrackMe alerts, allowing to perform further correlation, enrichment or filtering on the notable events.
- Review: *Alerting Architecture & Third-Party Integration*



8.3 Use TrackMe to detect abnormal events count drop in Splunk feeds

Detecting abnormal events count drop in Splunk feeds with TrackMe

- This TrackMe whitepaper tutorial demonstrates how you can leverage TrackMe to detect **abnormal events count drop in Splunk feeds**.
- It is a frequent use case requested by TrackMe users. The goal is to be able to easily detect when a feed in Splunk is facing a significant drop in the amount of events received.
- A massive drop in a given feed, whether you look at this from the index perspective or different factors, can be a sign of a problem in the data ingestion pipeline, or even a sign of a problem in the data source itself.
- TrackMe tackles this challenge in different ways, notably by leveraging Machine Learning driven anomaly detection.
- In this tutorial, we will cover the different options available to licensed and unlicensed TrackMe users.
- This tutorial demonstrates use cases around Machine Learning Outliers detection. For deeper insights, refer to the following documentation: *Outliers Anomaly Detection*.
- In this tutorial, we will review the out-of-the-box features of the **splk-dsm** component as well as an additional use case for licensed customers leveraging the **splk-flx** component.

Hint

Before we start, let's summarize our 3 main options and features in TrackMe:

- **splk-dsm:** Out-of-the-box component for feeds tracking, performs Machine Learning based Anomaly Detection against the events count metrics driven by the ingest.
- **splk-flx (licensed customers):** Flex Object tracker component, allows you to orchestrate any kind of SPL logic, translates results into TrackMe entities, turning these into Key Performance Indicators (KPIs) and applying Machine Learning based Anomaly Detection.
- **Use case 1 with Flex:** We will demonstrate how to use the Flex Object tracker to track the events count of Splunk feeds, and apply Machine Learning based Anomaly Detection against these metrics. (These use cases have been added to the Flex Object library in TrackMe 2.0.89)
- **Use case 2 with Flex:** We will demonstrate how to use the Flex Object tracker to track the license usage volume metrics of Splunk feeds, and apply Machine Learning based Anomaly Detection against these metrics.

8.3.1 Objective: Detecting abnormal events count drop in Splunk feeds

Our objective can be summarised in a few charts, as bellow:

Our feed is still making it to Splunk, we get events but the number of events received has significantly decreased:





Let's highlight the main challenges:

- **Feeds behaviors:** All feeds can act differently. Some are continuous with very stable volumes, while others are much more volatile and can slightly change depending on the period of the day or time.
- **Scalability:** We need to be able to scale our detection with no restrictions.
- **Pipeline abstraction:** We also possibly need to abstract the pipeline from the feed itself. Depending on your requirements, the main factors could be simply based on indexes, combinations of indexes and sourcetypes, or even more complex factors.
- **Distinguishing between normal and abnormal behaviors:** Feeds suffering from an abnormal drop in events count are still likely sending events to Splunk. This is a different and more complex use case than tracking feeds interruption.

8.3.2 What does TrackMe do out of the box? (feeds tracking with splk-dsm)

TrackMe provides a main component for data sources tracking (splk-dsm) which performs this detection out of the box:

- When you configure the splk-dsm component, TrackMe performs an association of factors (by default indexes and sourcetypes), creates and maintains entities for each factor, called TrackMe entities.
- Part of the Outliers Anomaly detection framework, TrackMe creates and maintains automatically Machine Learning models based on the events count metrics: `splk.feeds.avg_eventcount_5m`
- Default system wide settings for the Machine Learning models are applied when TrackMe discovers a new entity.
- From this stage, TrackMe will continuously monitor the events count and apply Machine Learning based Anomaly Detection against these entities.
- Machine Learning models can fine-tune on a per entity basis, and different concepts such as ML confidence are actionned by TrackMe to limit the risk of false positives.
- When a given entity is detected as an anomaly, this defines a flag `isAnomaly` which influences the global entity health status, turning possibly an entity from `green` to `red`.

The example below shows a TrackMe data source entity, conditioned by the couple index/sourcetype:

We can observe a slight decrease in the events count. However, this may be normal behavior depending on the period of the week day:





Using merged mode, you can also consider all sourcetypes while using the splk-dsm component to track feeds:



Let's take an additional use case, and let's look at a quite continuous and stable feed:

- In this example, we have a feed which is characterized by a quite stable and continuous activity in terms of events count.
- Therefore, it does not really have a seasonality concept that we should be taking into account.
- We can fine-tune the associated Machine Learning model to use **time_factor: None** which will disable the seasonality concept.

The performance metrics view shows the event count by the ingest over time:

The current Machine Learning model is configured with the default time factor: %H:



By applying time_factor: None, we can disable the seasonality concept, note that TrackMe eventually performs correction if it finds out that the drop is not significant enough to avoid false positives:



More information about this feature in TrackMe

- These metrics are driven by the ingestion. (rolling ingest per 5 minutes)
- Therefore, this is not a strict event counts for a given time period, but instead these metrics demonstrate the ingest behaviour of the feeds from the event counts perspective.
- You can define the default models definition, such as the **time_factor** which conditions how TrackMe handles the seasonality of the events in a given feed.
- You can also define the default **time period** to be considered when TrackMe trains the models.
- Finally, each model can **individually** be fine-tuned as needed, you can specify many settings on per model basis which will apply against a given entity especially.
- In conclusion, detecting abnormal events count drop is an **out of the box feature** of feeds tracking (splk-dsm), so technically TrackMe performs this activity by default and without the need for any extra configuration!

8.3.3 Flex Object (splk-flx): Detect abnormal events count drop using Flex

Hint

Flex Object tracker use case to monitor and detect abnormal events count drop

- The Flex Object tracker component is a very flexible and powerful restricted component of TrackMe. (**splk-flx**)
- These capabilities are available for licensed customers only.
- These use cases were added to the **Flex Object library in TrackMe 2.0.89**.
- It allows to orchestrate any kind of SPL logic, translates results into TrackMe entities, turning these into Key Performance Indicators (KPIs) and applying Machine Learning based Anomaly Detection.
- Because you have full control and within this use case context, we can use a very efficient and costs optimised `tstats` search to track the event count of Splunk feeds according to our needs.

- These metrics will be more “simple” compared to what does the spl-dsm component, these won’t be driven by the ingest but will be strict events counts over a specific time window.
- Equally, we will define the default definition for the Machine Learning models, which can then still be fine-tuned on a per entity basis.
- Finally, because you have full control, you totally define what an entity should, whenever you want to consider indexes as whole, or more complex scenarios.
- One advantage here is that we do not have to break against granular time concepts to calculate complex metrics such as the latency, so we can create **very efficient searches with the lowest cost in compute**.

First, let’s create a new Virtual Tenant

Depending on your needs and preferences, you can add the component to any existing Virtual Tenant, or dedicate a new Virtual Tenant for this use case.

For the purposes of the documentation, we will create a new Virtual Tenant called **detect-feeds-count**.

Once proceeded, we have a new Virtual Tenant ready to be used:



Second, create a new Flex Object tracker

Let's enter the new tenant, and define our Flex Object tracker:



We will create a new Flex Object tracker, called **all-idx-drop-detectt**:



Flex Object tracker source code

Hint

These use cases are available out of the box from TrackMe 2.0.89

- Since TrackMe 2.0.89, we have added these two use cases to the Flex Object library, so you can simply leverage the library to implement the use cases.
- In TrackMe 2.0.98, we added the dictinct count host to the use case, this is also a very valuable KPI to track.
- You can still create the use cases manually (therefore with a prior version of TrackMe), this is what we will demonstrate in this tutorial.

Creating a Flex Object tracker is pretty simple, you can take any of the use cases in the Flex Object library as an example (for instance: splk_license_usage_per_index), we will define our Flex Object tracker as:

Flex object tracker source code:

```
| tstats count as events_count, dc(host) as dcount_host where (index=*) by index
``` fine tune the query above, you can store eventually the root constraint in a␣
↪macro, or simply establish the list here ```

``` define the object, group, alias and description ```
| eval object = index
| eval group="drop-detect"
| eval alias = index
| eval object_description = "Splunk index: " . index

``` set status: in this case this is always going to be green, we will rely on␣
↪outliers detection to detect an abnormal drop in the events count ```
| eval status=1

``` set status_description ```
```

(continues on next page)

(continued from previous page)

```
| eval status_description="Events count drop detection for index: " . index . ",␣
↪latest events_count:" . events_count

``` Set metrics and outliers ```
| eval metrics = "{'splunk.feeds.idx.events_count': " . if(isnum(events_count),␣
↪events_count, 0) . ", 'splunk.feeds.idx.dcount_host': " . if(isnum(dcount_host),␣
↪dcount_host, 0) . "}"

| eval outliers_metrics = "{'splunk.feeds.idx.events_count': {'alert_lower_breached':␣
↪1, 'alert_upper_breached': 0, 'time_factor': 'none'}, 'splunk.feeds.idx.dcount_host
↪': {'alert_lower_breached': 1, 'alert_upper_breached': 0, 'time_factor': 'none'}}"

``` do not alert if the index stops receiving events entirely, setting this to 0␣
↪means this, otherwise you can define a value in seconds, if the index stops␣
↪receiving events, it will turn red for inactivity ```
| eval max_sec_inactive=0
```

We will choose a certain time period, as our search does not have to deal with a time granularity concept (we are not breaking against the time), we have a super efficient and very low cost tracker!

Hint

Fine tuning the Flex Object tracker

- You can fine tune the root constraint (the "where") to match your needs and especially filter on the indexes you want to track.
- You can also modify the break by sequence to match your needs, for instance including the sourcetype or including additional factors such as custom indexed fields.
- Finally, the group is used to logically group entities together, you can also fine tune the group definition to match your needs. (such as using a specific indexed field providing a company or perimeter concept, etc)

Rolling versus Absolute Key Performance Indicator

In next steps, we will define the time period and the frequency of the tracker, depending on these settings and depending on your preferences, we influence the meaning of the KPIs:

- **Rolling Key Performance Indicator:** This is a metric that is frequently executed by the tracker, and the time period is relative to the current time.
- **Absolute Key Performance Indicator:** The time period length matches the execution frequency (cron), so we produce a strict events count for a definitive period of time.
- The meaning of the metrics will be different depending on these settings, although the finality remains the same and both will be equally able to detect abnormal events count dropped.
- We will cover both options in the next steps of this tutorial.

Let's summarize the main ideas:

Rolling metrics pros/cons

Pros:

- Easier to understand. The metric is the real strict event count and would match manual comparison searches.
- Can be easily backfilled, so you don't need to wait for TrackMe to build the historical knowledge before the tracking is operational.

Cons:

- More sensitive to failures. If the tracker schedule is missed and not executed by Splunk (outage), the metrics will be wrong.
- Always looks at 1 hour back compared to the real-time data. We will need at least 1 hour to detect a significant drop in the events count. (However, the use case is about trend analytics, so this last argument is in reality not a real issue)

Absolute metrics pros/cons

Pros:

- Less sensitive to failures, if the tracker schedule is missed and not executed by Splunk, there will be no impact on the detection capabilities.
- Represents a different and valuable way to look at the trending of the events count rather than a basic strict events count.

Cons:

- Cannot be easily backfilled, the trend can only be calculated at point in time and backfilling it is much more challenging.
- Does not represent the basic events count, so one cannot compare with simple Splunk searches, in some cases when doing investigations, analysts can struggle to understand the metrics.

Absolute Key Performance Indicator

The time period length must match the frequency of the tracker, so we have a strict events count for a definitive period of time:

- earliest: -1h@h
- latest: @h
- cron_schedule: `0 * * * *`

Note: you can optionally be more protected about a single failure of Splunk in executing our Tracker once per hour, you can for instance execute it twice per hour, ML calculations will be affected as we do not use a sum but avg/latest and so others, so a duplicated metrics has no impacts.

Rolling Key Performance Indicator

We would for instance execute on a high frequency the Flex Tracker against a bigger time period length:

- earliest: -60m
- latest: now
- cron_schedule: `*/5 * * * *`

Defining the Machine Learning model general rules

In our Flex Object tracker, we have defined the following line:

```
| eval outliers_metrics = "{'splunk.feeds.idx.events_count': {'alert_lower_breached':␣
↪1, 'alert_upper_breached': 0, 'time_factor': 'none'}, 'splunk.feeds.idx.dcount_host
↪': {'alert_lower_breached': 1, 'alert_upper_breached': 0, 'time_factor': 'none'}}"
```

This does the following:

- Automatically define a Machine Learning model definition per entity

- The model will by default alert only for Lower Bound breaches, which means in our context an abnormal drop in the events count.
- Finally, we decided to define by default a time_factor: none, which means we disable the seasonality concept, so we have single LowerBound and UpperBound ranges per entity.
- We could also choose a different pattern between: %H, %H%M, %w%H, %w%H%M

Hint

Machine Learning time_factor

- In this example, we have chosen to define a time_factor: none, which means we disable the seasonality concept.
- You can also do the opposite and enable by default a time_factor of your choice.
- Remember that you can always update the time factor per entity as needed.
- This statement in our Flex Object tracker defines the default definition of the models and is only used once at the time of the entity discovery or if you delete the Machine Learning model for a given entity. (so it is re-created automatically at the next execution)

Backfilling the Key Performance Indicator (Absolute KPI)

ONLY AFTER A FIRST EXECUTION OF THE TRACKER: You can optionally choose to backfill the metrics. This is very useful for the Absolute KPIs:

make sure to replace the tenant_id, the root constraints and the period as per your needs. Here we will be backfilling for the past 30 days which would remain efficient even at large scale

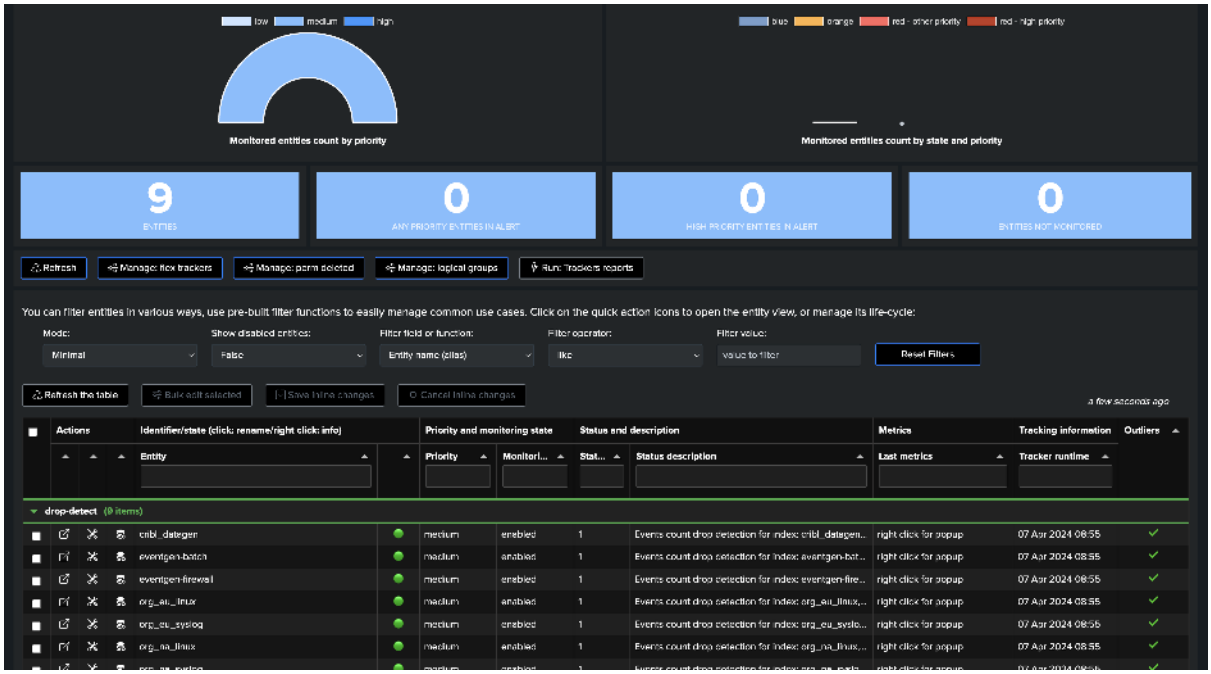
```
| tstats count as trackme.splk.flx.splunk.feeds.idx.events_count, dc(host) as trackme.
↪splk.flx.splunk.feeds.idx.dcount_host where (index=*) earliest=-30d latest=@h by _
↪time, index span=1h

| lookup trackme_flx_tenant_detect-feeds-count alias as index OUTPUT _key as object_
↪id, object
| eval tenant_id = "detect-feeds-count", object_category="splk-flx"
| where isnotnull(object_id)

| mcollect index=trackme_metrics split=t object, object_category, object_id, tenant_id
```

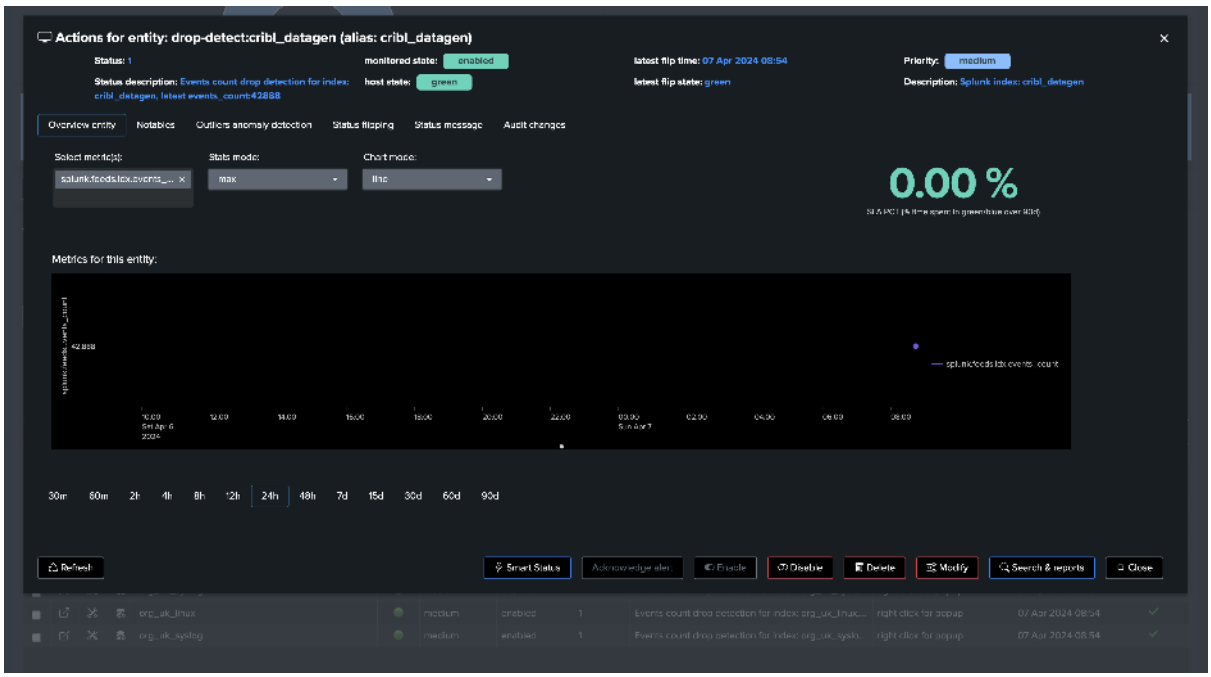
Final, execute and review entities

Finally, once we have first executed the tracker, we will be able to see one TrackMe entity per (active) index:

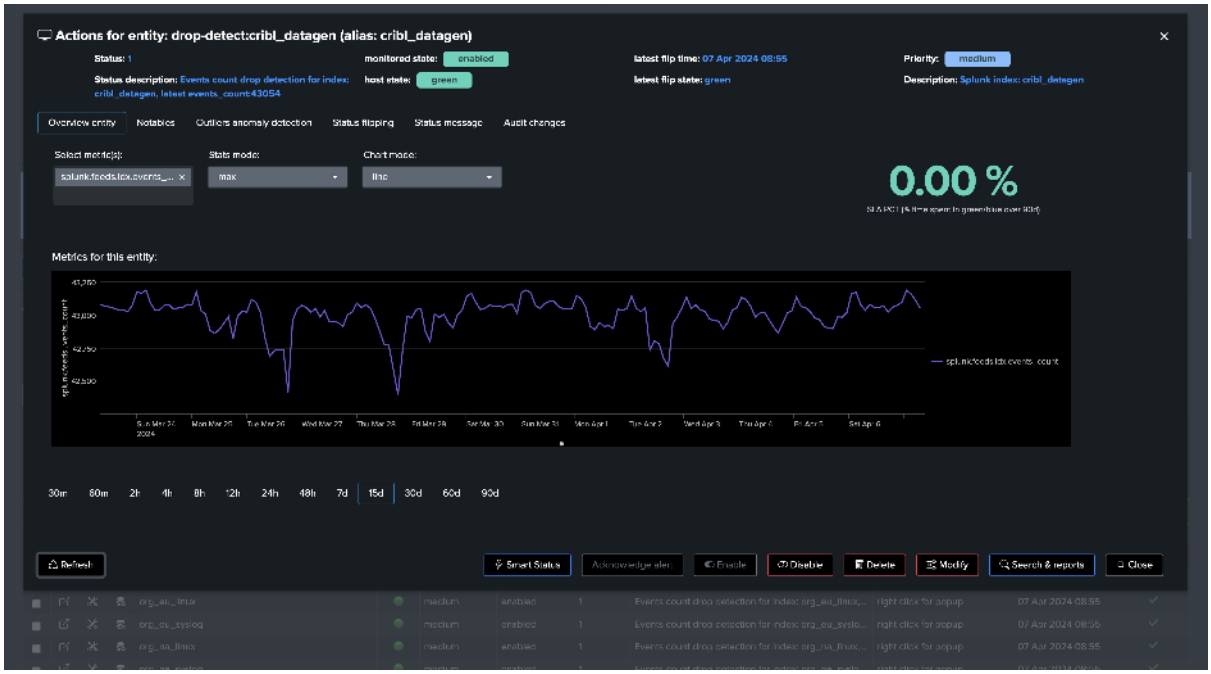


Depending if we wanted to use Rolling or Absolute KPIs, and backfill, we either see the first point of metric or a full backfilled metrics:

Not backefilled:

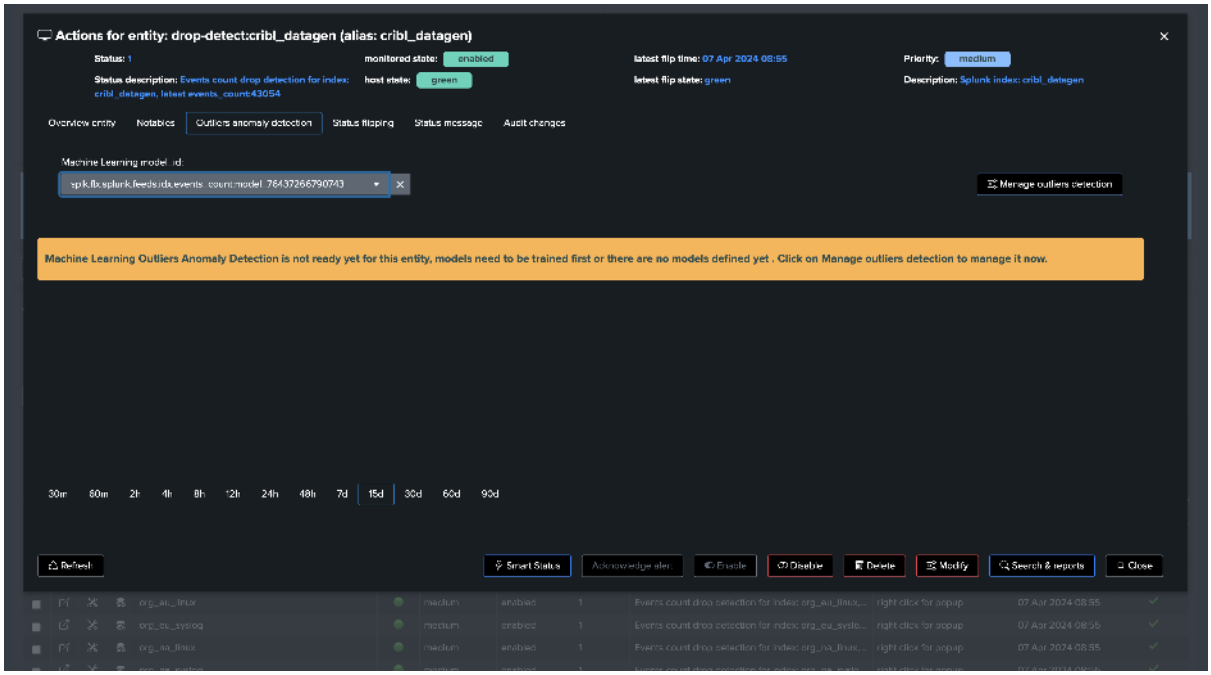


Backfilled:

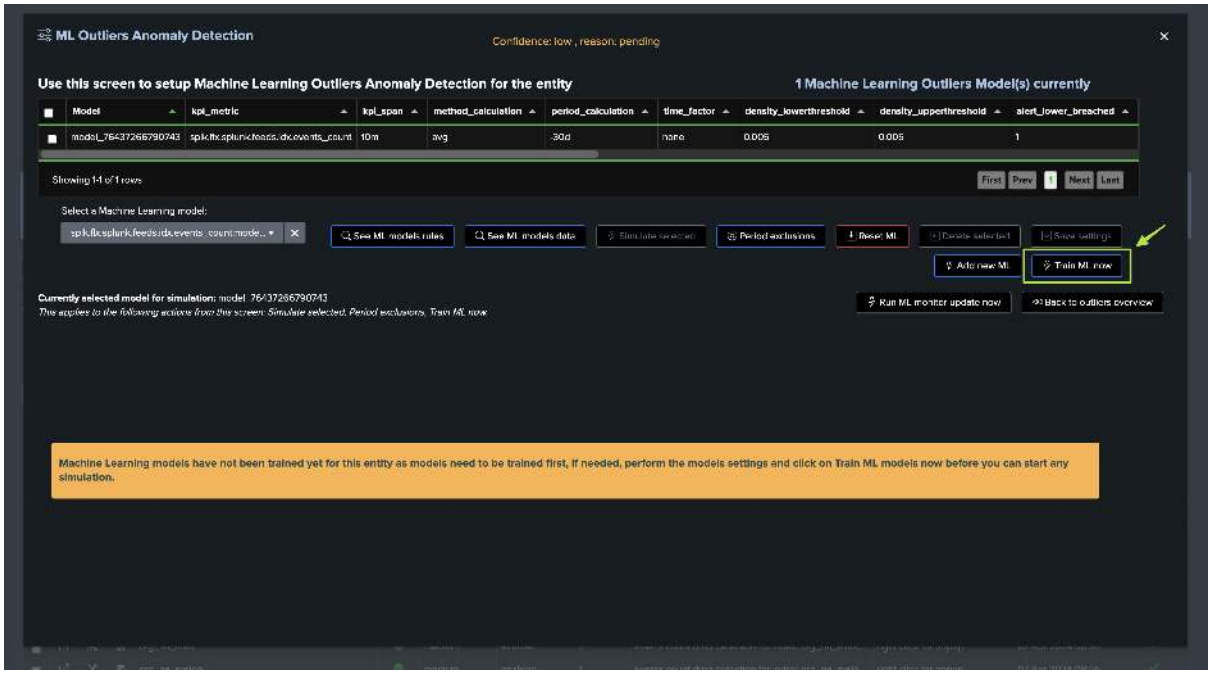


Third, let's have a look at the Machine Learning models

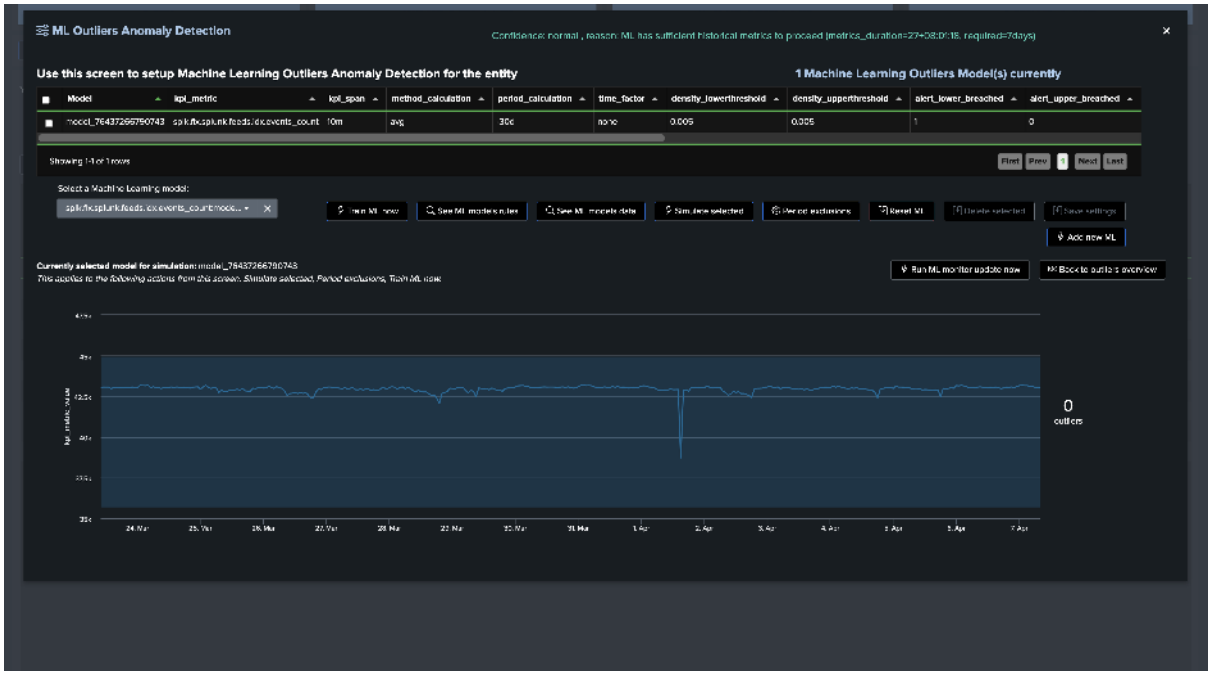
Since we've just created our tenant and entities, the Machine Learning models are not yet trained, we will have to wait for the next execution of the tracker to see the models being trained, or manually trigger it:



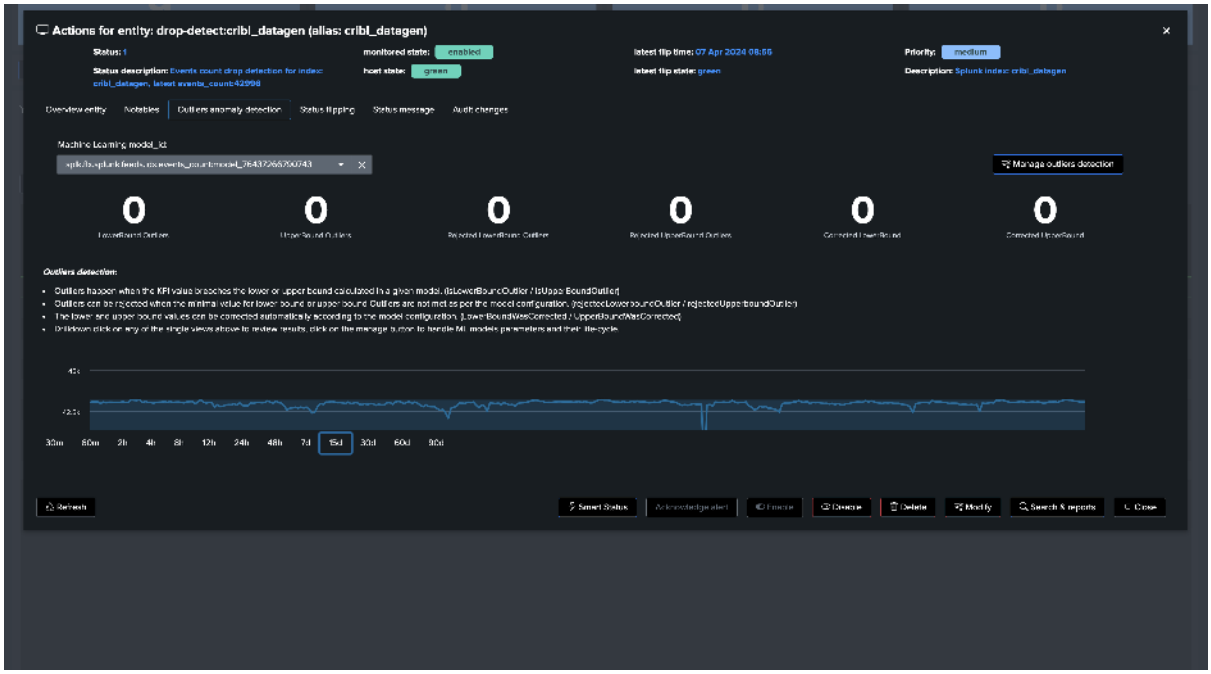
For the purposes of the tutorial, let's run the training now:

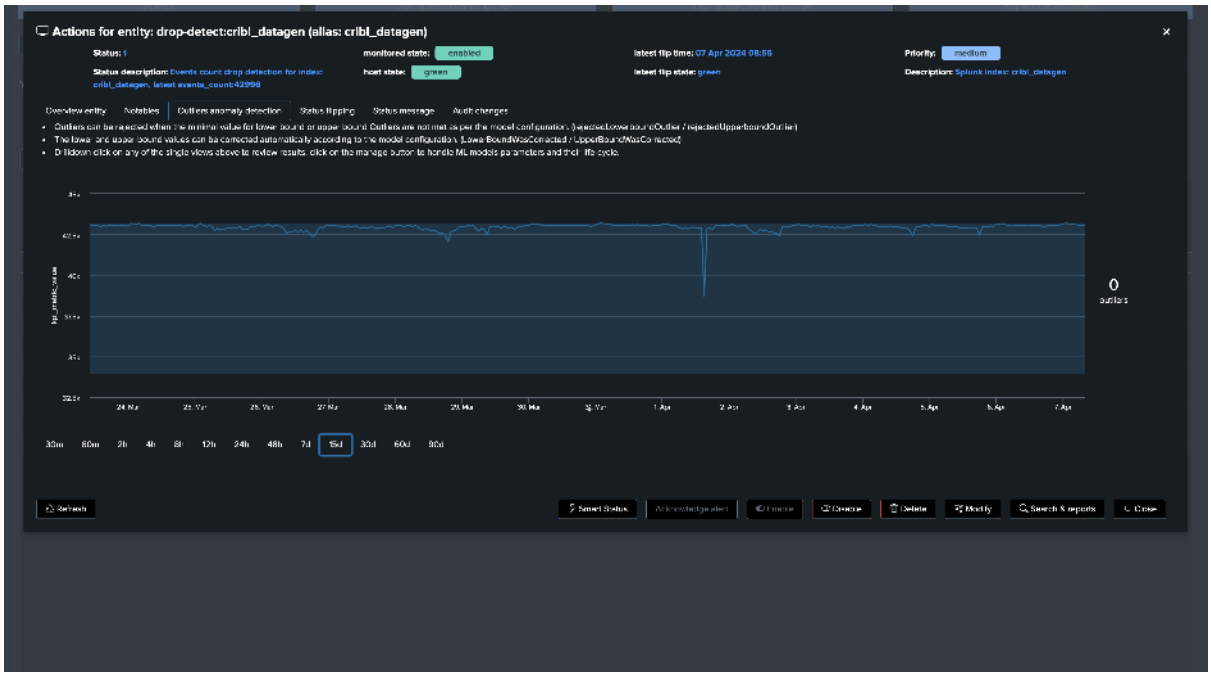


Once trained and simulated, we can see the Machine Learning models being applied:



Going back on the live Machine Learning Outliers detection screen (if the chart does not appear, refresh the page!):





note: Additional options can be set to fine tune Models, notably you can specify minimal values for the LowerBound breaches. (Anything below this value will be rejected as LowerBound Outliers)

Conclusion on drop detection

At this stage, we have:

- A ready and operational framework to continuously track and detect abnormal events count drop in Splunk feeds.
- TrackMe will automatically train and maintain Machine Learning models for each entity, and apply Anomaly Detection against these models.
- Very little fine-tuning is really required, so you can focus on the core of the use case.

- You can also fine tune the most important and critical feeds of your environments easily in TrackMe.

8.3.4 Flex Object use case 2 (splk-flx): Detect abnormal events count drop using Flex and Splunk licence usage

Hint

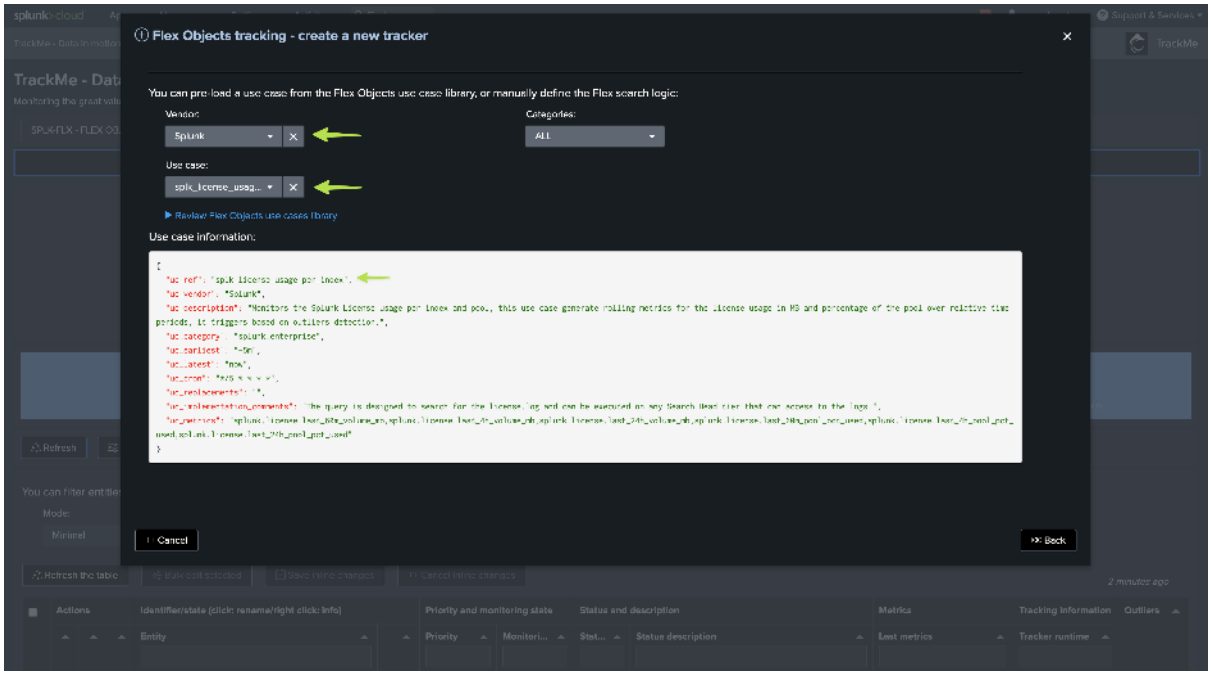
Flex Object tracker use case to monitor and detect abnormal events count drop using Splunk licence statistics

- We also have a **third option** with TrackMe FLex Object tracker and using the Splunk licence usage statistics.
- In fact, TrackMe provides an out of the box Flex Object use case called `splk_license_usage_per_index` which does exactly this!
- This use case leverages the Splunk licence usage statistics which accounts against Splunk indexes, this is available and valid for both **Splunk Enterprise customers and Splunk Cloud customers**.
- Unlike the previous use cases, the calculations are based against **Megabytes volume metrics of incoming data**, rather than events count.
- This is also a very efficient use case, the out of the box use case is based on a rolling metric approach, but you could also modify the scheduling plan as we did in the previous example if you wished to turn it into an absolute metric instead.
- This use case calculates **3 ranges of rolling metrics**: -24h, -4 hours and -60 minutes, and applies Machine Learning based Anomaly Detection against the past 24 hours metrics by default.

We will assume our Flex Object Virtual Tenant is ready to be used, let's call the Out of the Box use case

Go with the creation of a Flex Object tracker and select the out of the box use case:

See after this screenshot for an alternative version of the tracker focusing only on the volume based metrics:



Alternative version of the tracker focusing only on the volume based metrics:

```
index=_internal sourcetype=splunkd source="*/license_usage.log" earliest="-24h"␣
↪latest="now"
| stats sum(b) as b, latest(poolsz) as poolsz by idx, pool
| eval last_24h_volume_mb = round(b/1024/1024, 2), last_24h_pool_pct_used=round(b/
↪poolsz*100, 2)
| fields idx, pool, last_24h_volume_mb, last_24h_pool_pct_used
| eval object = "idx:" . idx

``` get last 4 hours ```
| append [ search index=_internal sourcetype=splunkd source="*/license_usage.log"␣
↪earliest="-4h" latest="now"
| stats sum(b) as b, latest(poolsz) as poolsz by idx, pool
| eval last_4h_volume_mb = round(b/1024/1024, 2), last_4h_pool_pct_used=round(b/
↪poolsz*100, 2)
| fields idx, pool, last_4h_volume_mb, last_4h_pool_pct_used
| eval object = "idx:" . idx
]

``` get last 60 minutes ```
| append [ search index=_internal sourcetype=splunkd source="*/license_usage.log"␣
↪earliest="-60m" latest="now"
| stats sum(b) as b, latest(poolsz) as poolsz by idx, pool
| eval last_60m_volume_mb = round(b/1024/1024, 2), last_60m_pool_pct_used=round(b/
↪poolsz*100, 2)
| fields idx, pool, last_60m_volume_mb, last_60m_pool_pct_used
| eval object = "idx:" . idx
]

| stats first(*) as "*" by object
| foreach last_* [ eval <<FIELD>> = if(isnum('<<FIELD>>'), '<<FIELD>>', 0) ]

| eval group="license|pool:" . pool
| eval alias = idx
| eval object_description = "Splunk license usage for index: " . idx . " / pool: " .␣
↪pool

``` set status: in this case this is always going to be green, we will rely on␣
↪outliers detection to detect an upperBound outlier against the rolling 24 hours␣
↪metrics ```
| eval status=1

``` set status_description ```
| eval status_description=case(
status=1, "last_24h_volume_mb: " . last_24h_volume_mb . ", last_24h_pool_pct_used: " .
↪ last_24h_pool_pct_used,
status=2, "last_24h_volume_mb: " . last_24h_volume_mb . ", last_24h_pool_pct_used: " .
↪ last_24h_pool_pct_used,
status=3, "last_24h_volume_mb: " . last_24h_volume_mb . ", last_24h_pool_pct_used: " .
↪ last_24h_pool_pct_used
)

``` Set metrics and outliers ```
| eval metrics = "{\"splunk.license.last_60m_volume_mb\": " . if(isnum(last_60m_
↪volume_mb), last_60m_volume_mb, 0) . ", \"splunk.license.last_4h_volume_mb\": " .␣
↪if(isnum(last_4h_volume_mb), last_4h_volume_mb, 0) . ", \"splunk.license.last_24h_
```

(continues on next page)
```

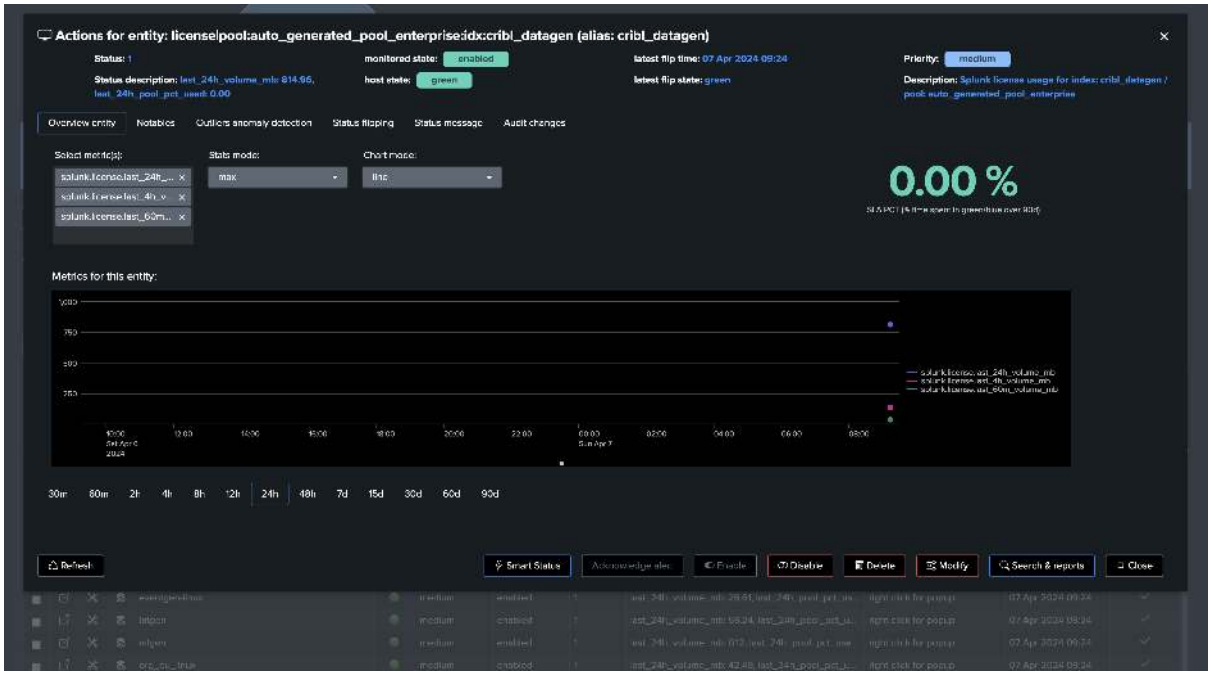
(continued from previous page)

```
↪volume_mb\": " . if(isnum(last_24h_volume_mb), last_24h_volume_mb, 0) ."}"

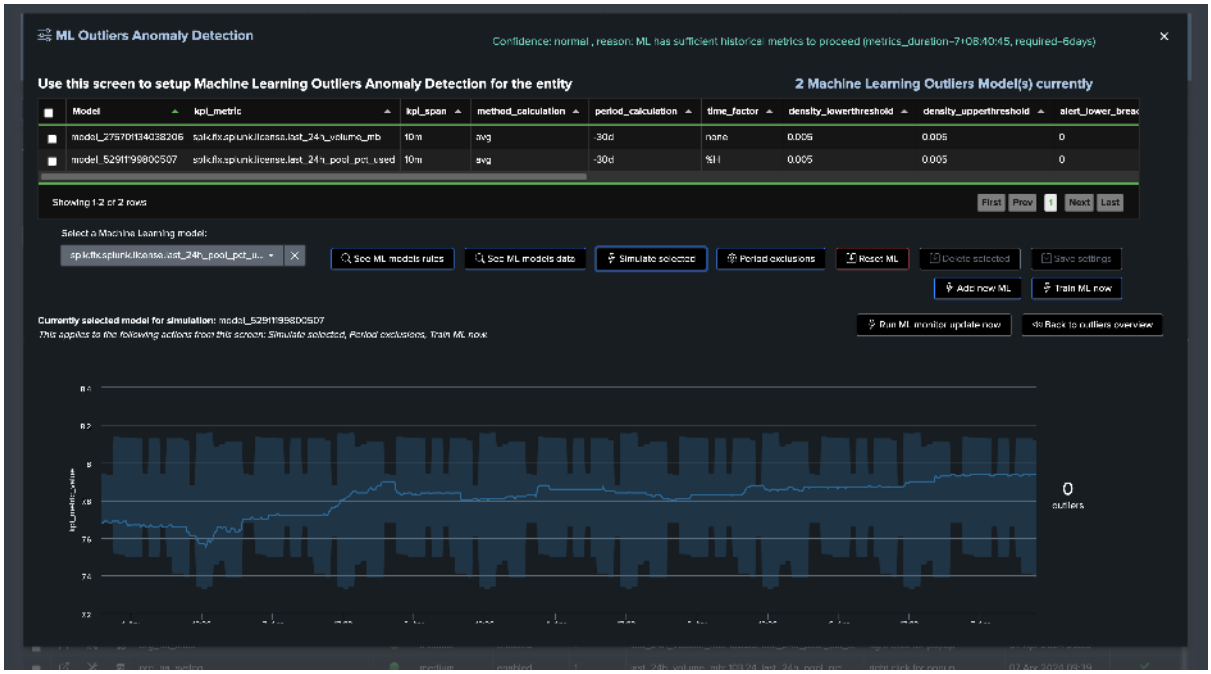
| eval outliers_metrics = "{\"splunk.license.last_24h_volume_mb\": {\"alert_lower_
↪breached\": 0, \"alert_upper_breached\": 1}}"

``` alert if inactive for more than 30 days```
| eval max_sec_inactive=30*24*60*60
```

Once first executed, we have one entity per index, with 3 levels of volume metrics

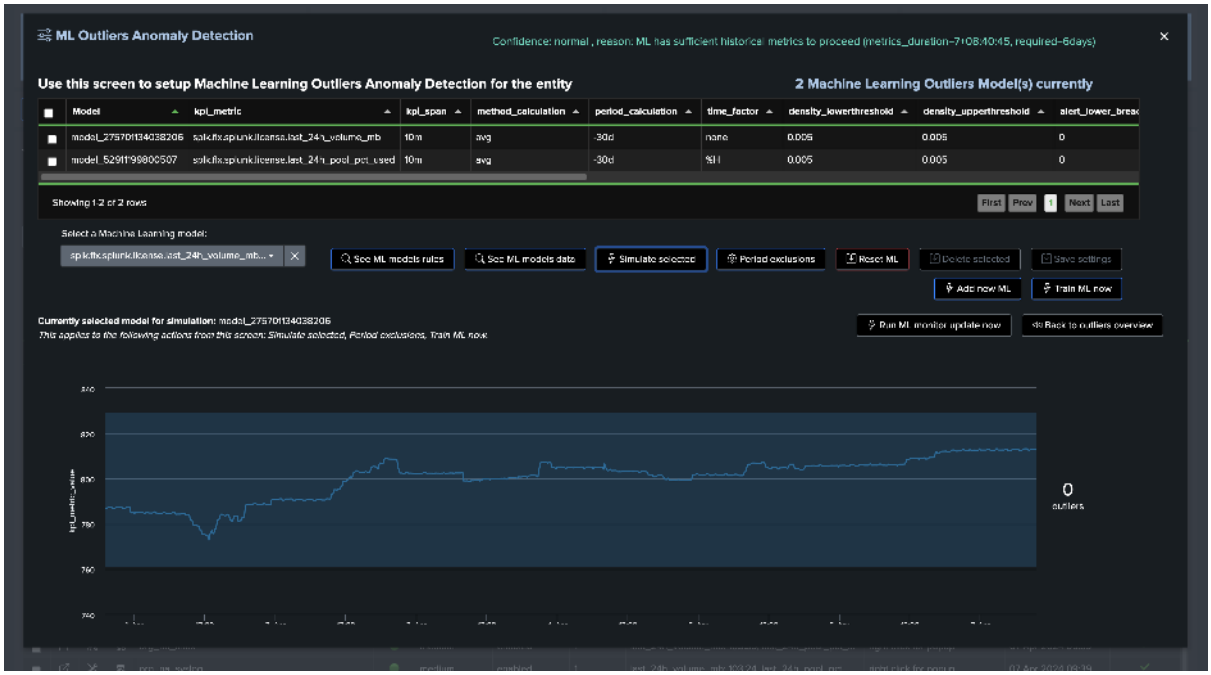


This example shows Machine Learning applied against the licence pool percentage usage with seasonality concept enabled: (for Enterprise customers only)



This example shows Machine Learning applied against the volume metrics with seasonality concept dis-
```

abled:

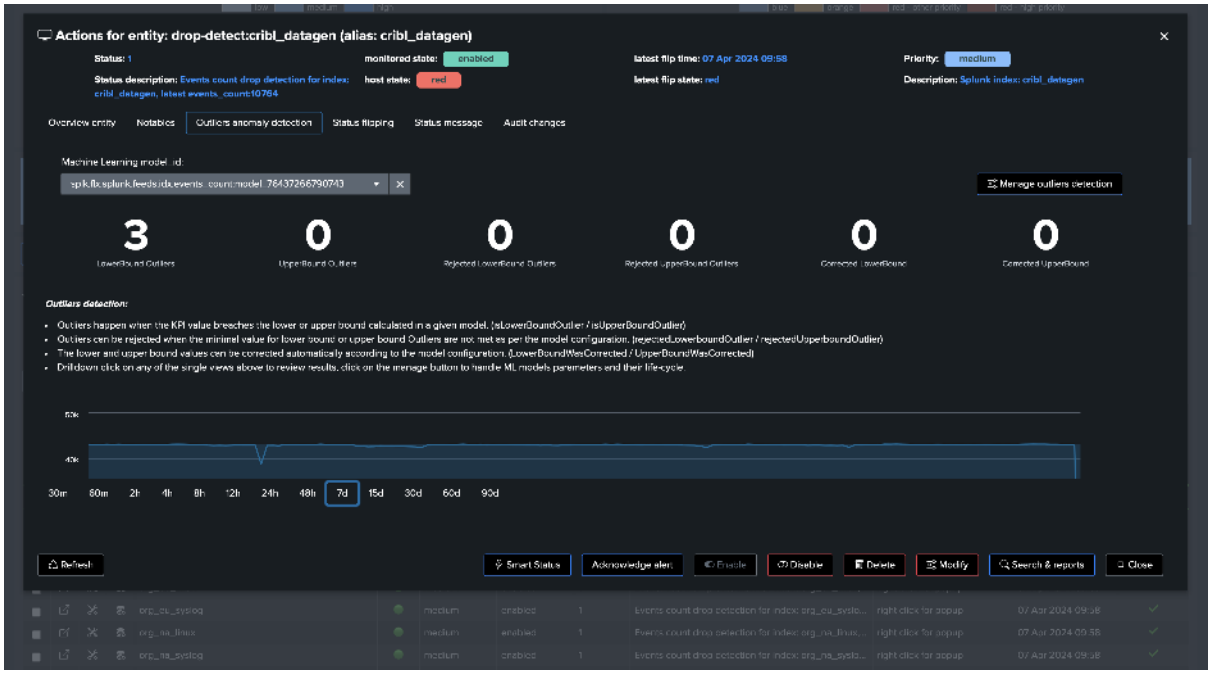


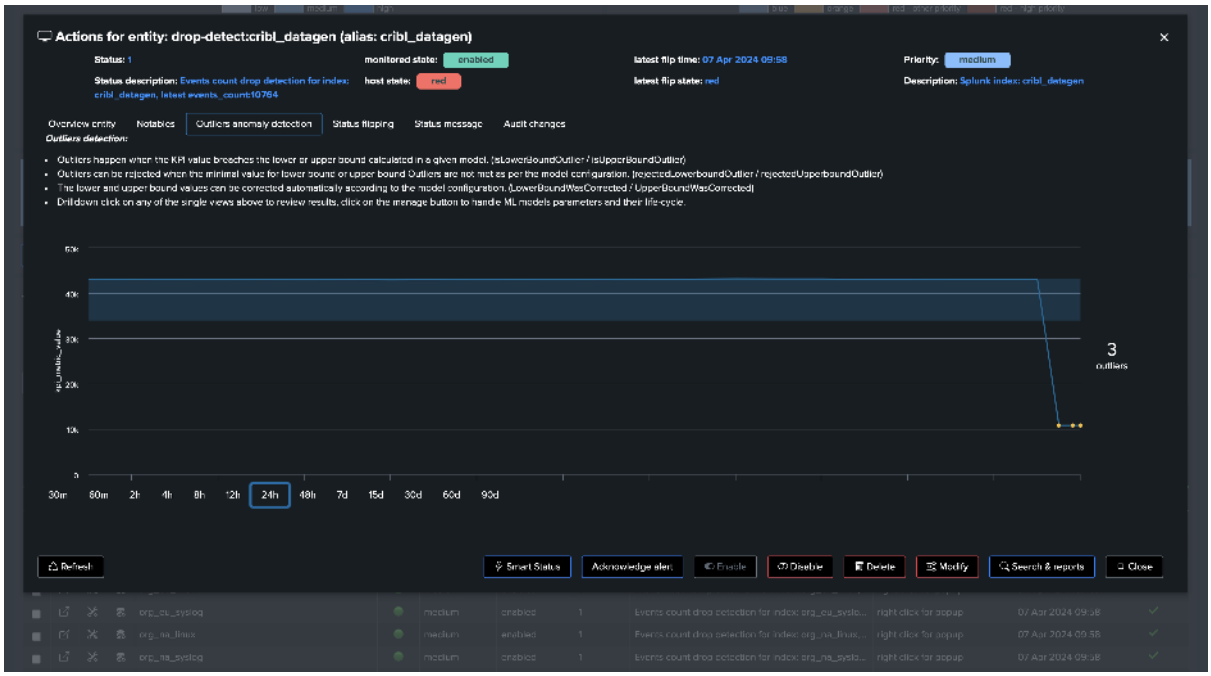
Conclusion: volume based metrics

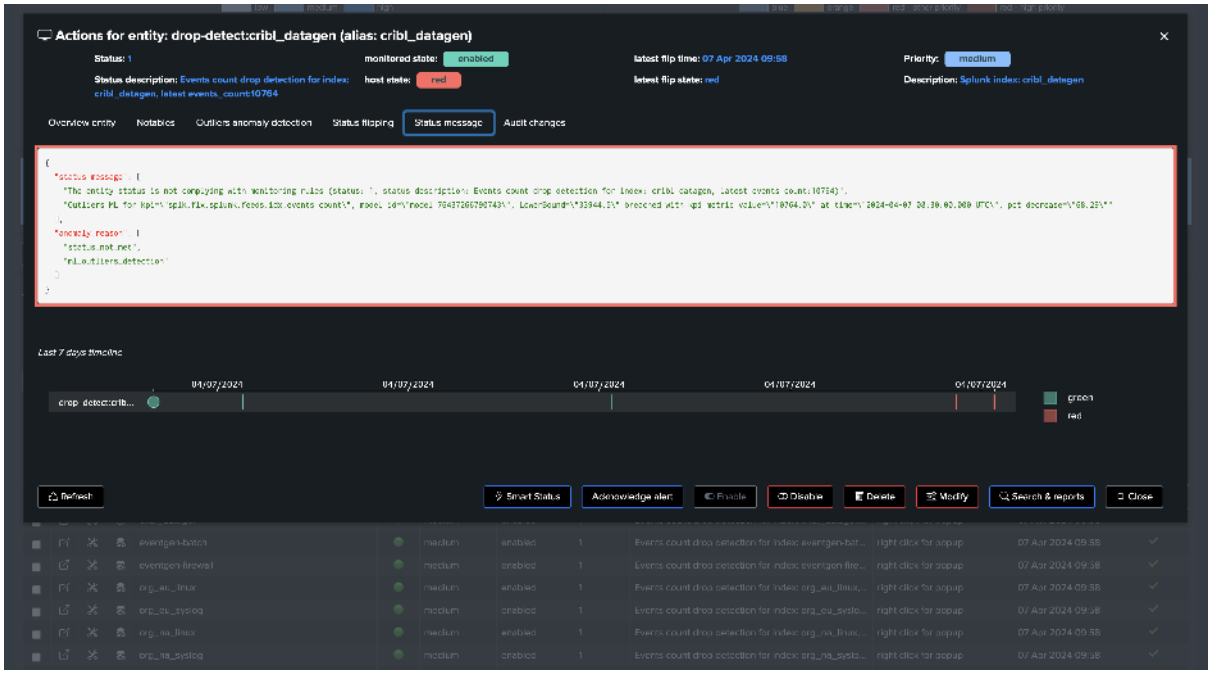
At this stage:

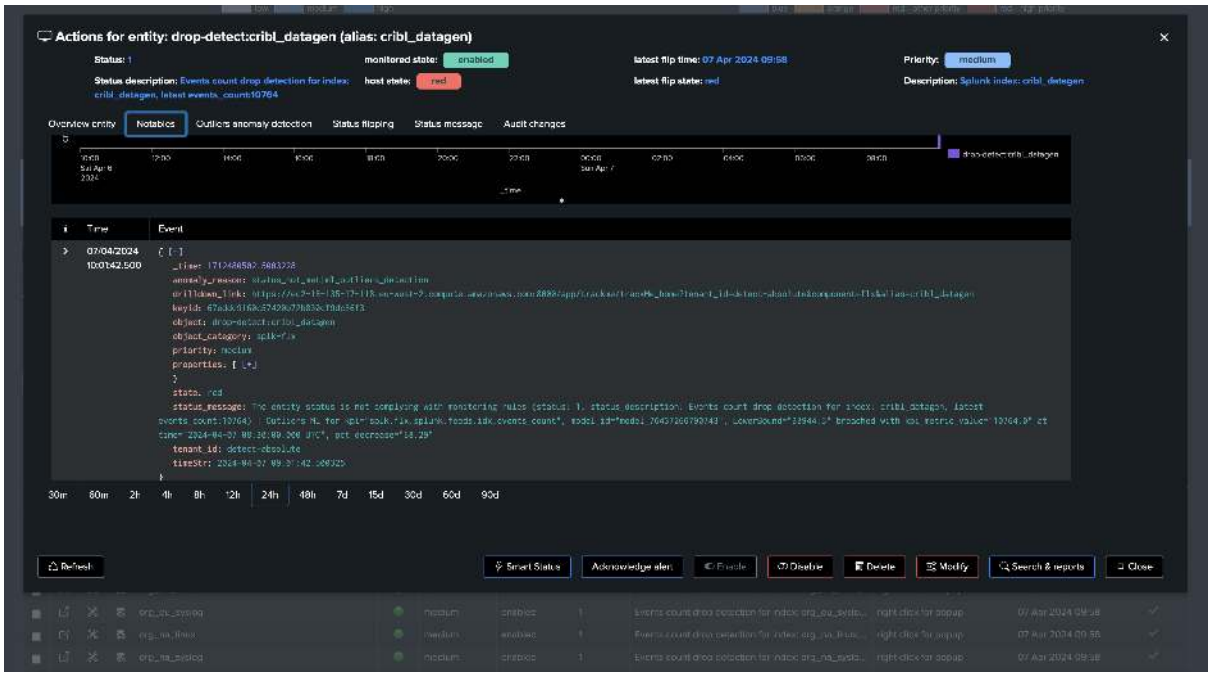
- We have a **ready and operational automated powerful framework** to continuously track and detect abnormal volume drop in Splunk feeds per index.
- TrackMe will automatically train and maintain Machine Learning models for each entity, and apply Anomaly Detection against these models.
- Very little fine-tuning is really required, so you can focus on the core of the use case.
- You can also fine tune the most important and critical feeds of your environments easily in TrackMe.

Screenshots of an entity triggering on Machine Learning detection due to abnormal events count drop:


Actions for entity: drop-detect:cribl_datagen (alias: cribl_datagen)
Status: 1
monitored state: enabled
latest flip time: 07 Apr 2024 09:58
Priority: medium
Status description: Events count drop detection for index: cribl_datagen, latest events_count:10764
host state: red
latest flip state: red
Description: Splunk index: cribl_datagen
Overview entity
Notables
Outliers anomaly detection
Status flipping
Status message
Audit changes
Machine Learning model id:
Manage outliers detection
3
LowerBound Outliers
0
UpperBound Outliers
0
Rejected LowerBound Outliers
0
Rejected UpperBound Outliers
0
Corrected LowerBound
0
Corrected UpperBound
Outliers detection:
Outliers happen when the KPI value breaches the lower or upper bound calculated in a given model. (isLowerBoundOutlier / isUpperBoundOutlier)
Outliers can be rejected when the minimal value for lower bound or upper bound Outliers are not met as per the model configuration. (rejected_lowerboundOutlier / rejectedUpperboundOutlier)
The lower and upper bound values can be corrected automatically according to the model configuration. (LowerBoundWasCorrected / UpperBoundWasCorrected)
Drilldown click on any of the single views above to review results, click on the manage button to handle ML models parameters and their life-cycle.
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
Refresh
Smart Status
Acknowledge alert
Enable
Disable
Delete
Modify
Search & reports
Close


Actions for entity: drop-detect:cribl_datagen (alias: cribl_datagen)
Status: 1
monitored state: enabled
latest flip time: 07 Apr 2024 09:58
Priority: medium
Status description: Events count drop detection for index: cribl_datagen, latest events_count:10764
host state: red
latest flip state: red
Description: Splunk index: cribl_datagen
Overview entity
Notables
Outliers anomaly detection
Status flipping
Status message
Audit changes
Outliers detection:
Outliers happen when the KPI value breaches the lower or upper bound calculated in a given model. (isLowerBoundOutlier / isUpperBoundOutlier)
Outliers can be rejected when the minimal value for lower bound or upper bound Outliers are not met as per the model configuration. (rejected_lowerboundOutlier / rejectedUpperboundOutlier)
The lower and upper bound values can be corrected automatically according to the model configuration. (LowerBoundWasCorrected / UpperBoundWasCorrected)
Drilldown click on any of the single views above to review results, click on the manage button to handle ML models parameters and their life-cycle.
3
outliers
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
Refresh
Smart Status
Acknowledge alert
Enable
Disable
Delete
Modify
Search & reports
Close





8.4 Analyse log messages logging level to detect behaviour anomalies using TrackMe's Flex Object and Machine Learning Anomaly Detection

Detecting increasing volumes, globally and per index with TrackMe

- This TrackMe whitepaper tutorial demonstrates how you can leverage TrackMe to detect **Splunk instances and deployments abnormal conditions by analysing log messages and their logging level through the lens of Machine Learning Outliers detection**.
- This use case tracks Splunk internal logs and their logging level, turning these into KPIs and submitting them to TrackMe's Machine Learning Outliers detection engine.

- The objective is then to detect abnormal trends, such as an increase of the number of errors or a lack of informational messages, which would be symptomatic of potentially serious conditions affecting Splunk instances and deployments.
- The same use philosophy can be applied to any application log messages, and detect abnormal behaviour in any application or system.

Hint

Requires TrackMe licence:

- This use case is designed using TrackMe's restricted component Flex Object.
- This component is only available to licensed users of TrackMe, and is not available with the free community edition of TrackMe.

8.4.1 Objective: Making sense of Splunk log messages logging level and detect Splunk abnormal conditions

The basic logic is to track Splunk internal log messages through the lens of their logging level, similarly to the following Splunk search:

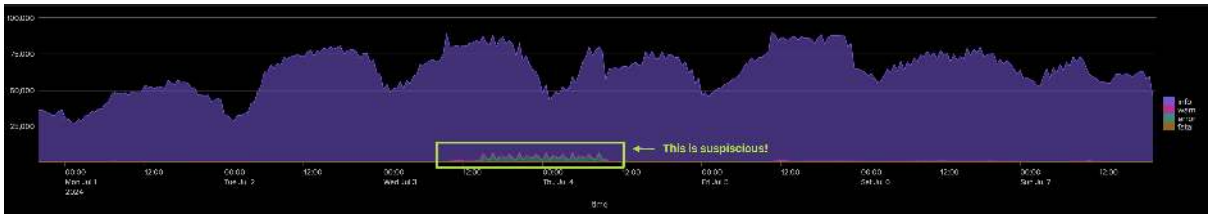
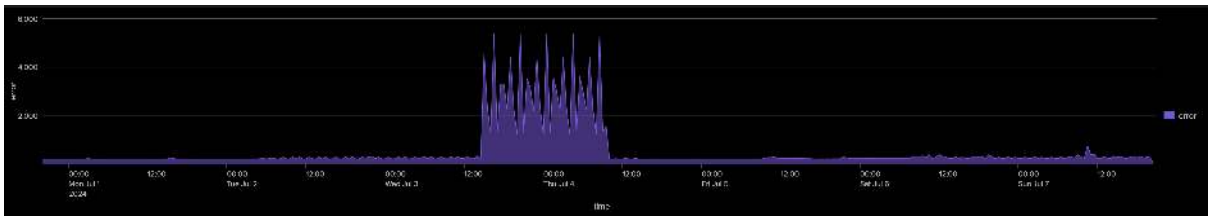
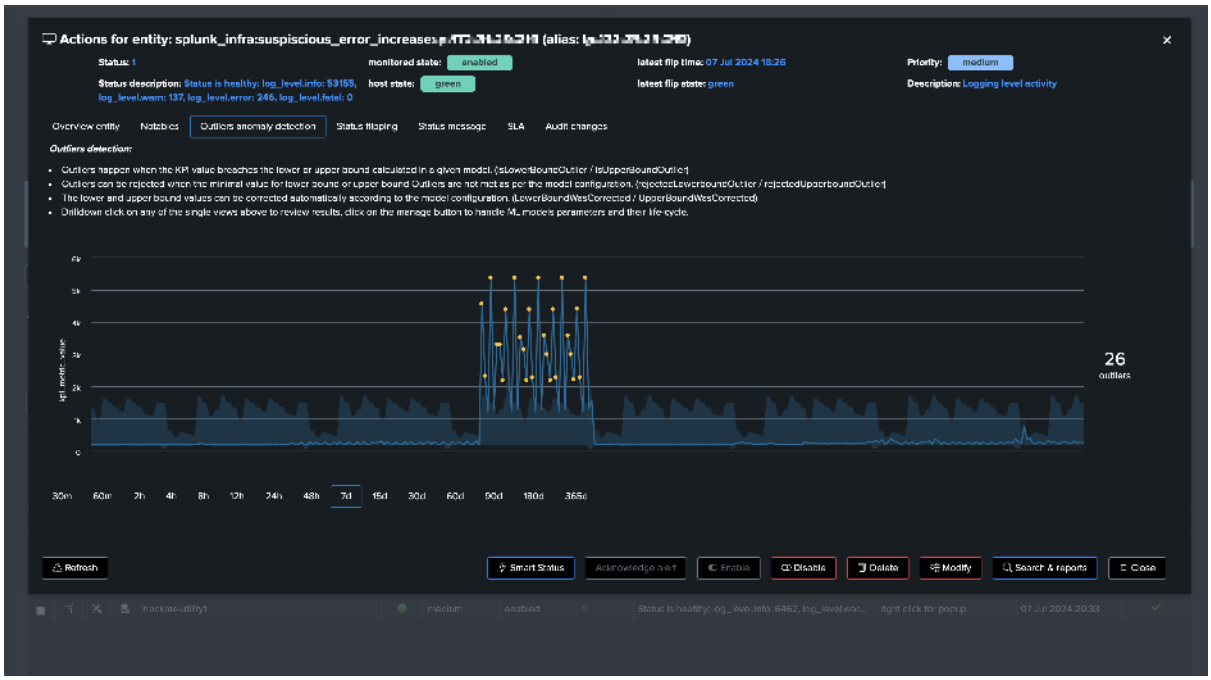
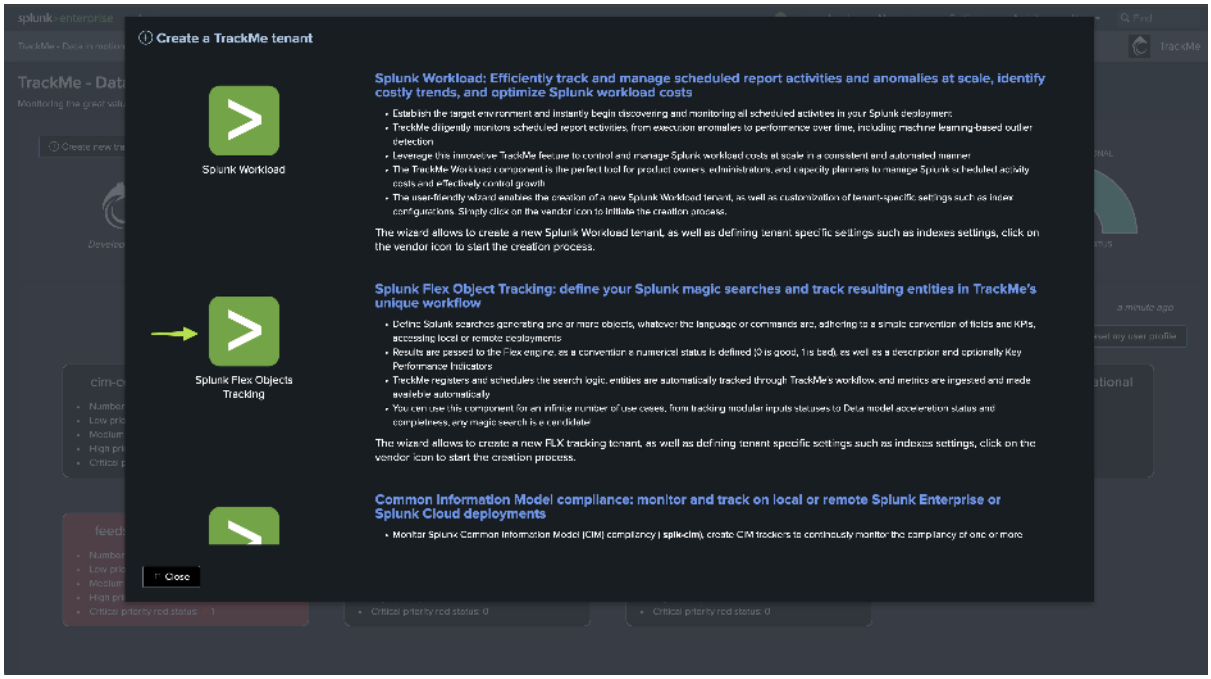
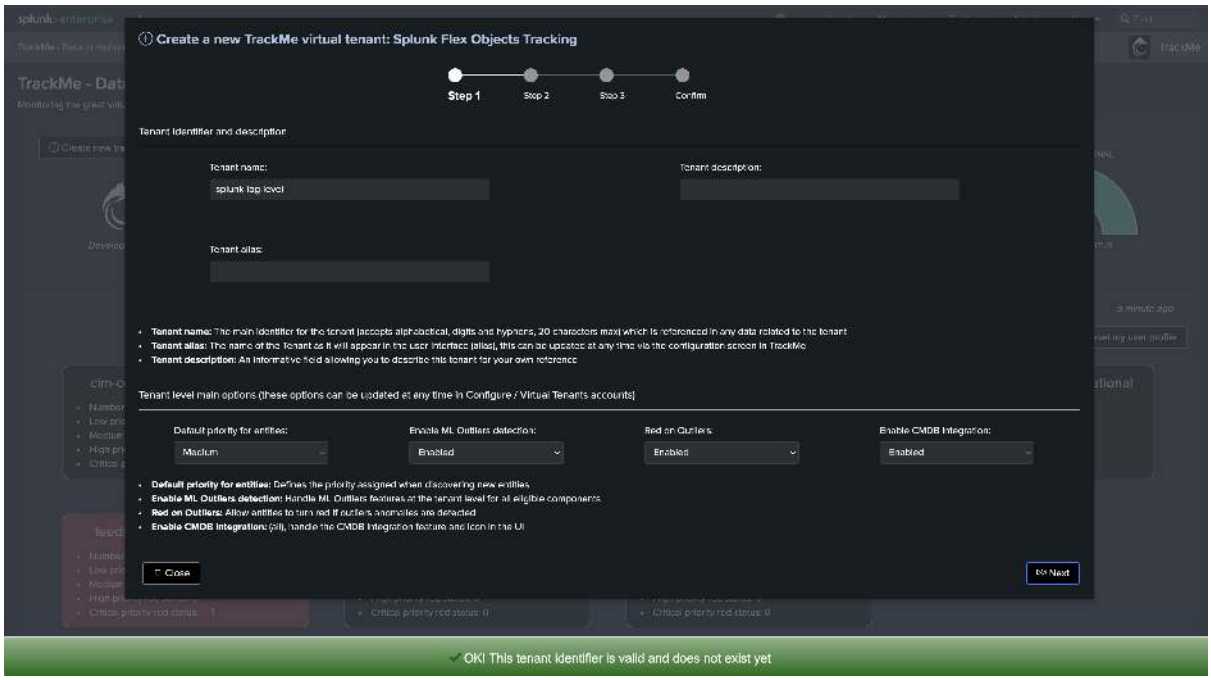
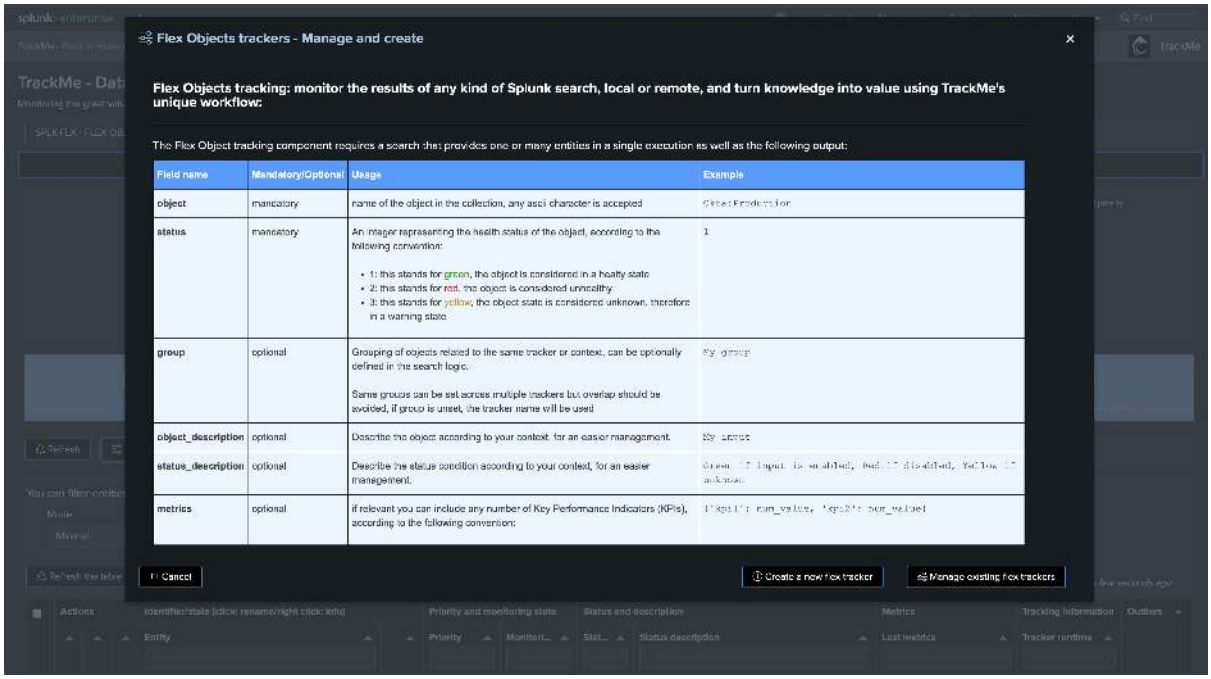
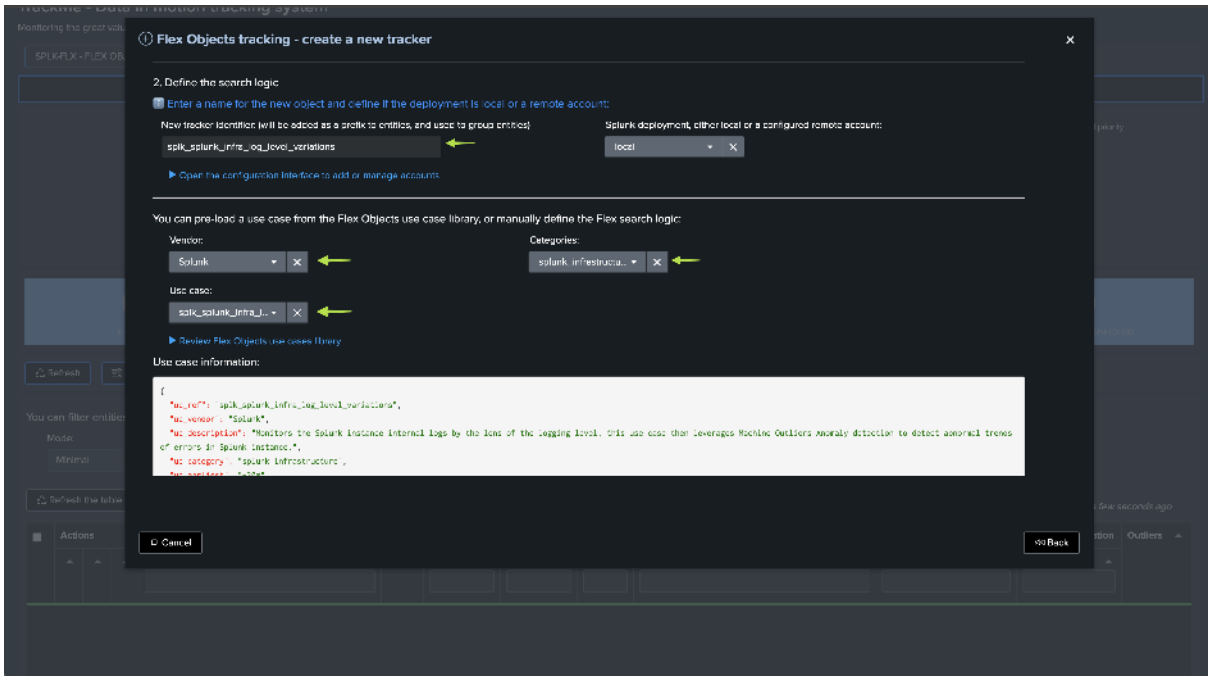
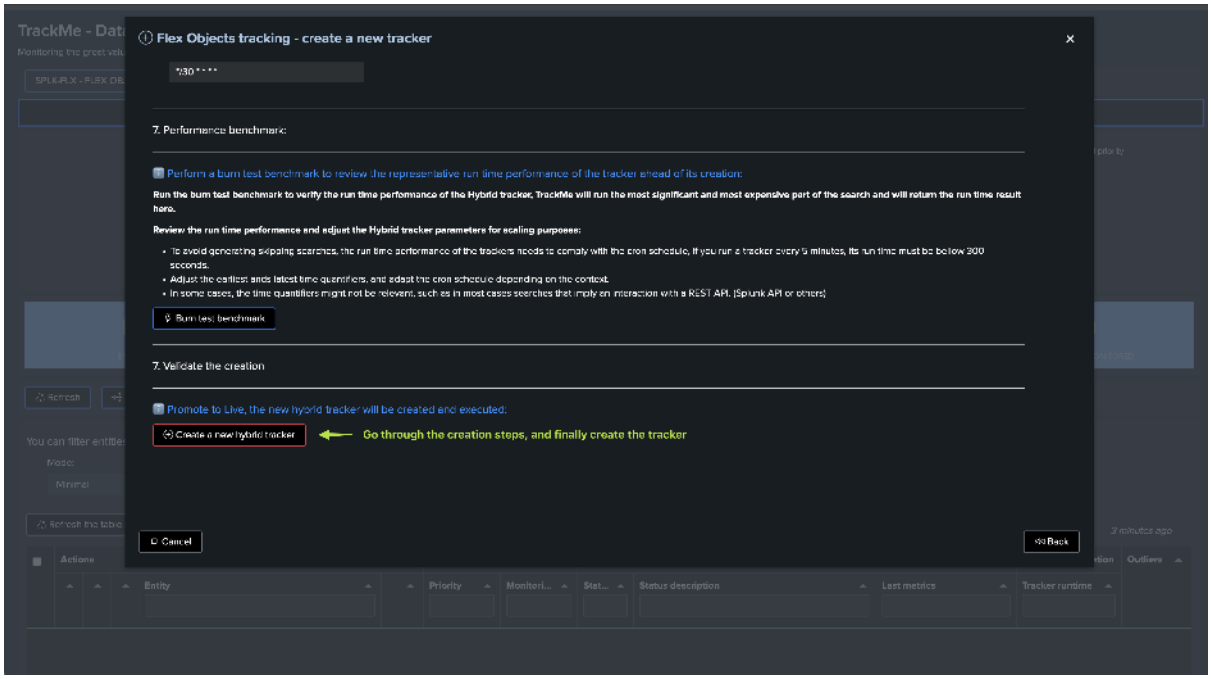
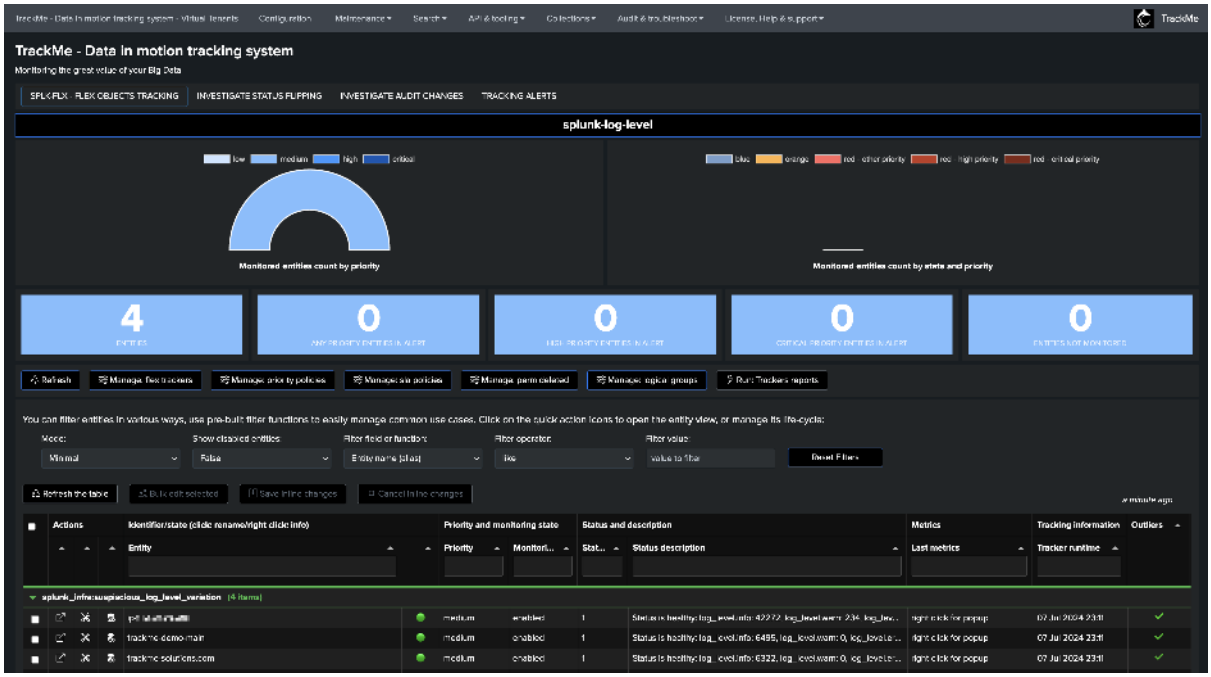
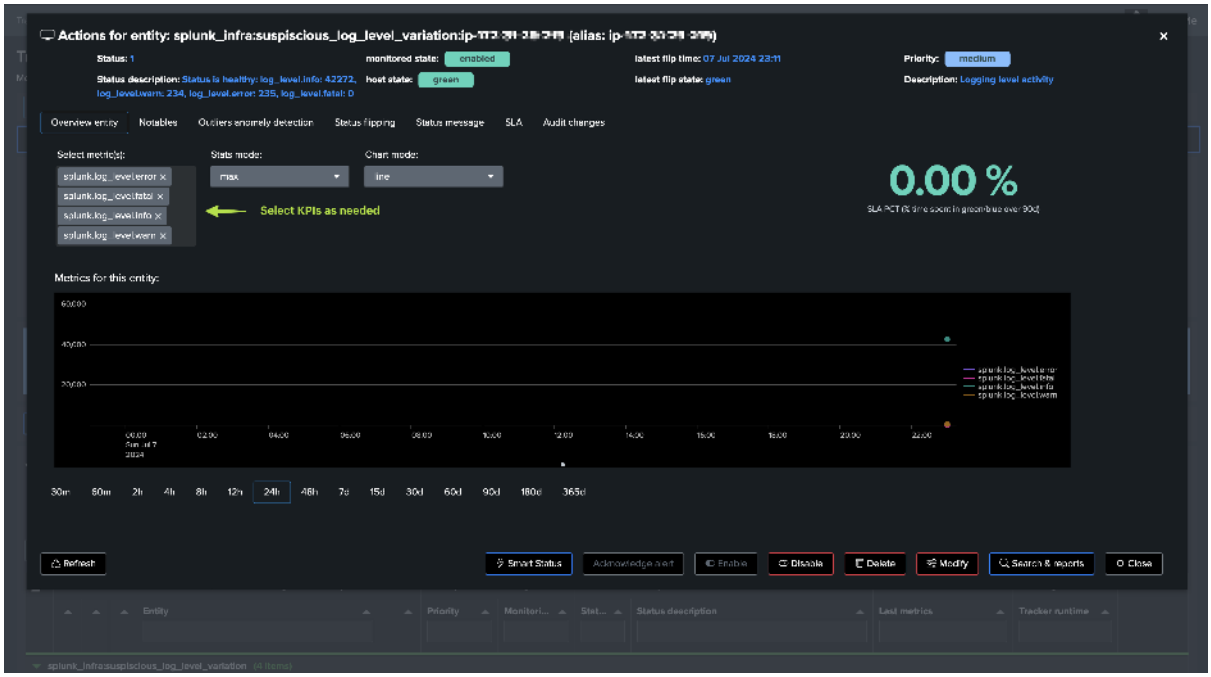
Let's take an example:

```
index=_internal host=* log_level=* (log_level=ERROR OR log_level=INFO OR log_
↪level=WARN OR log_level=FATAL)

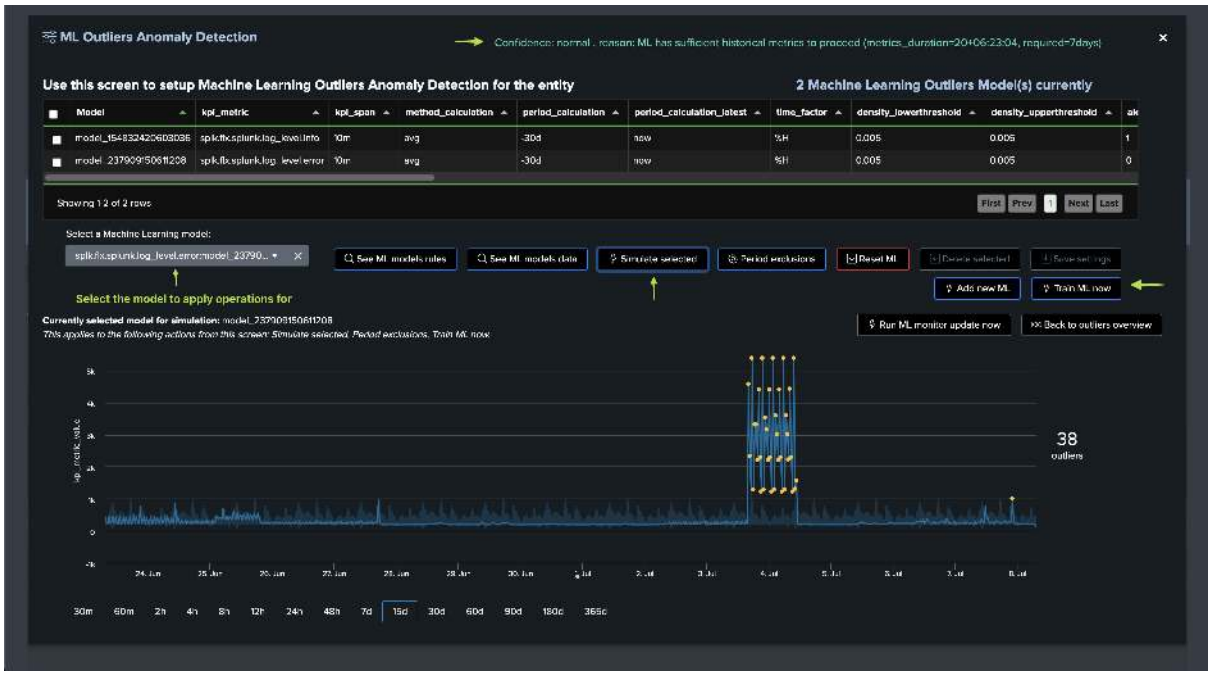
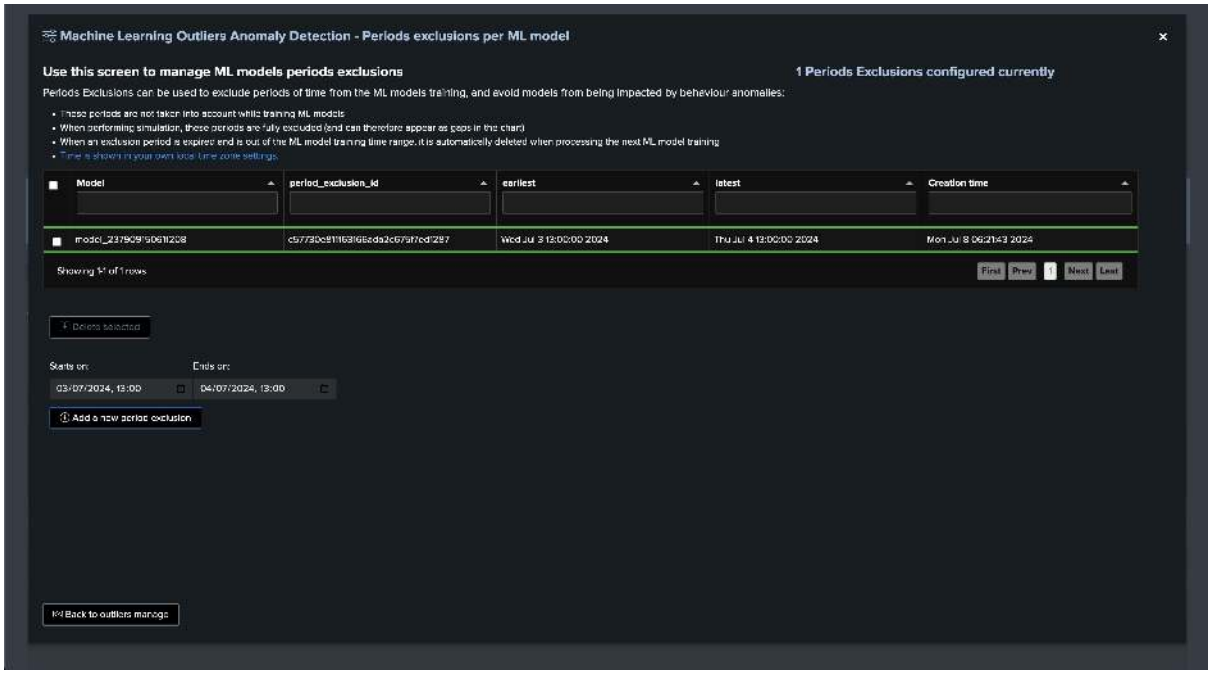
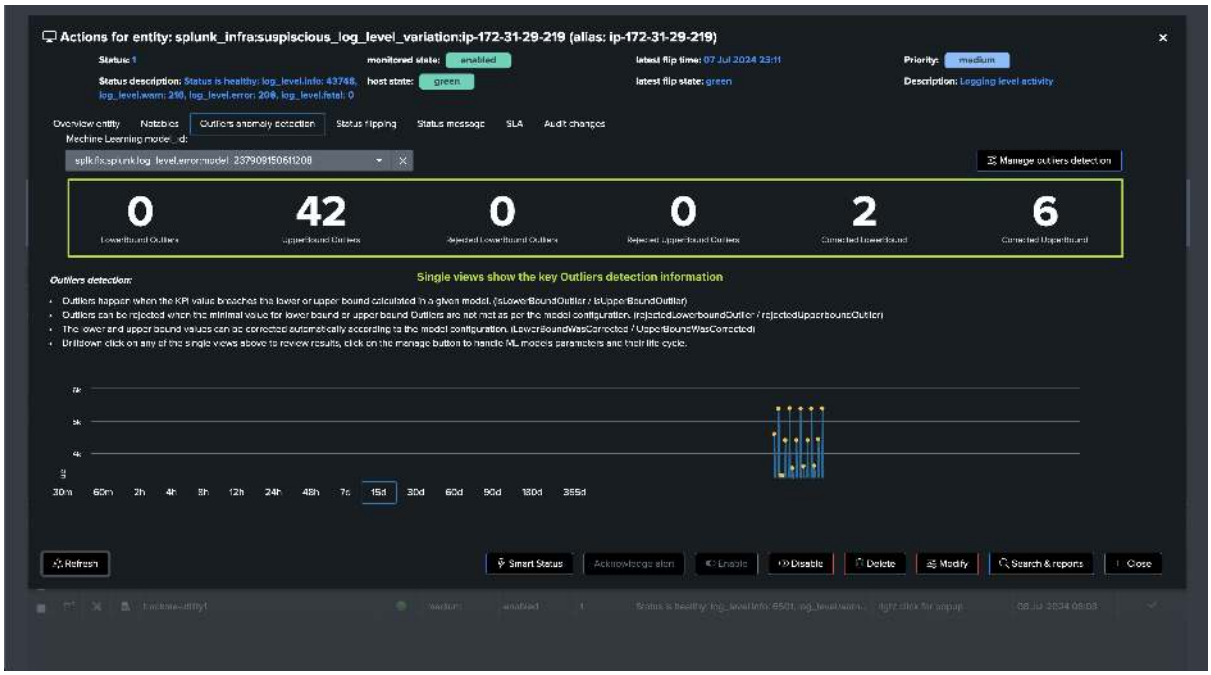
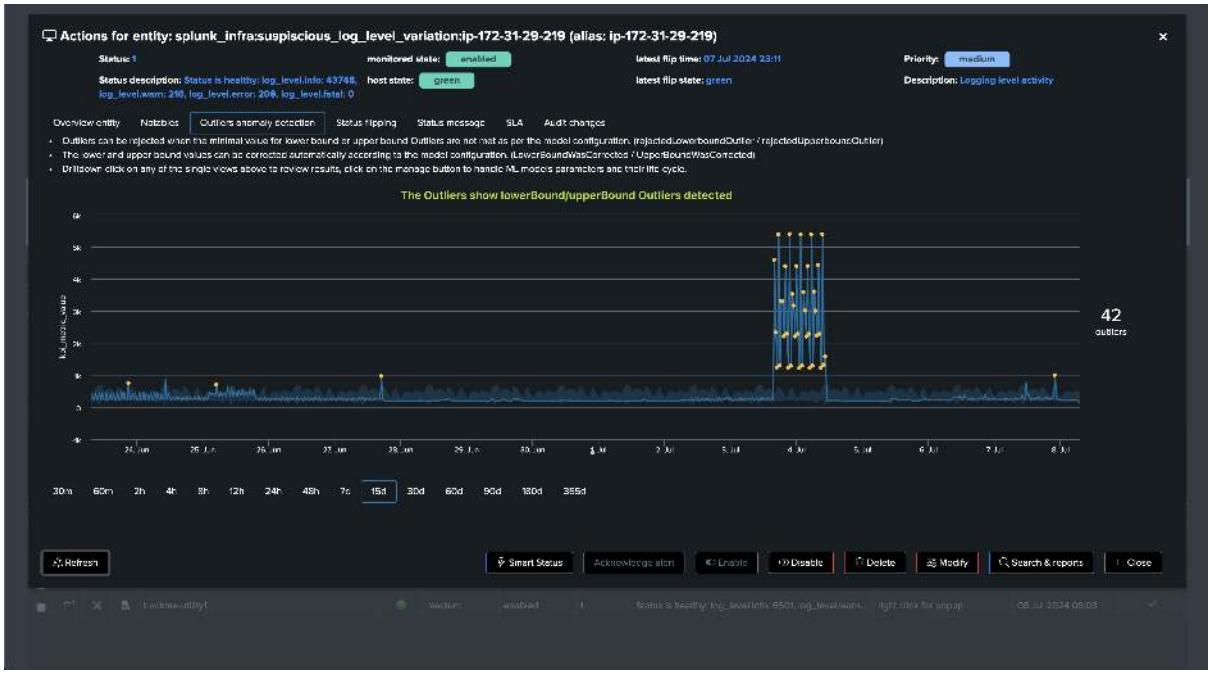
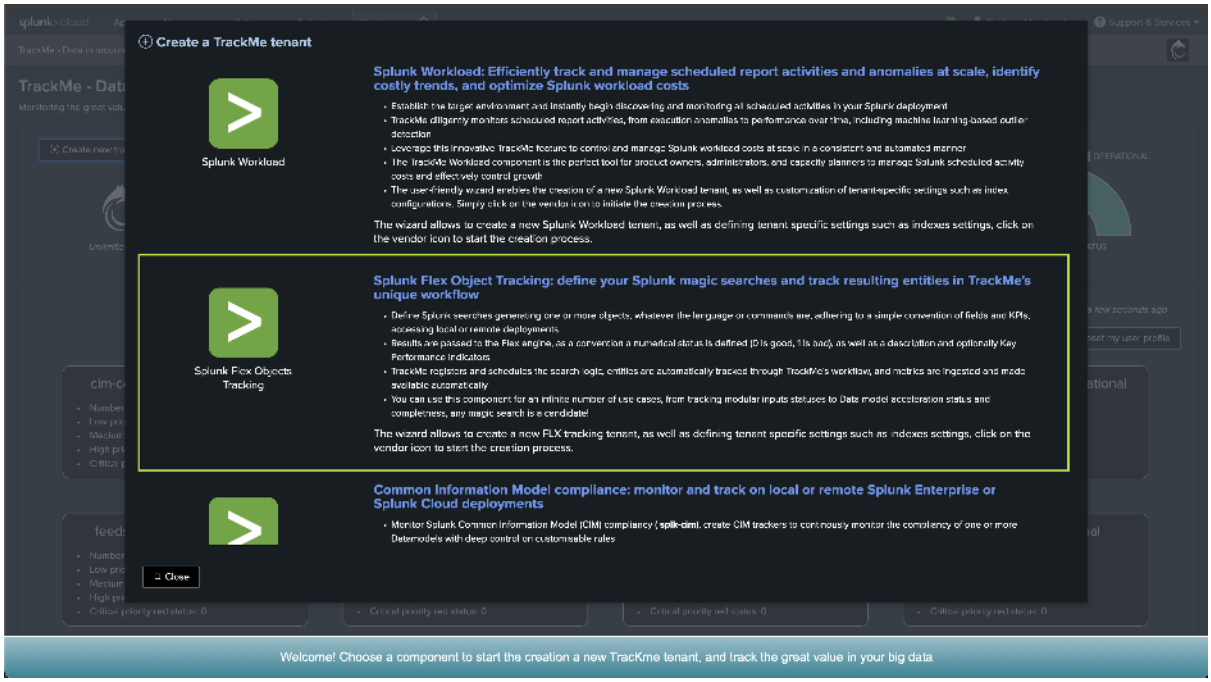
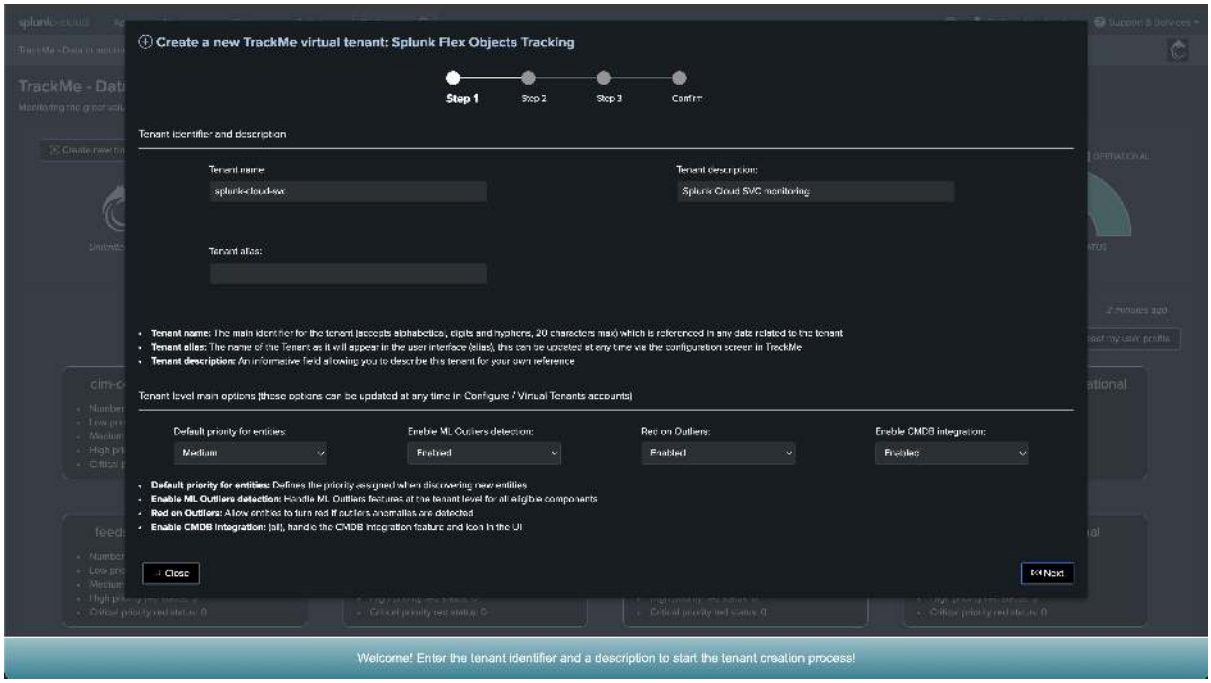
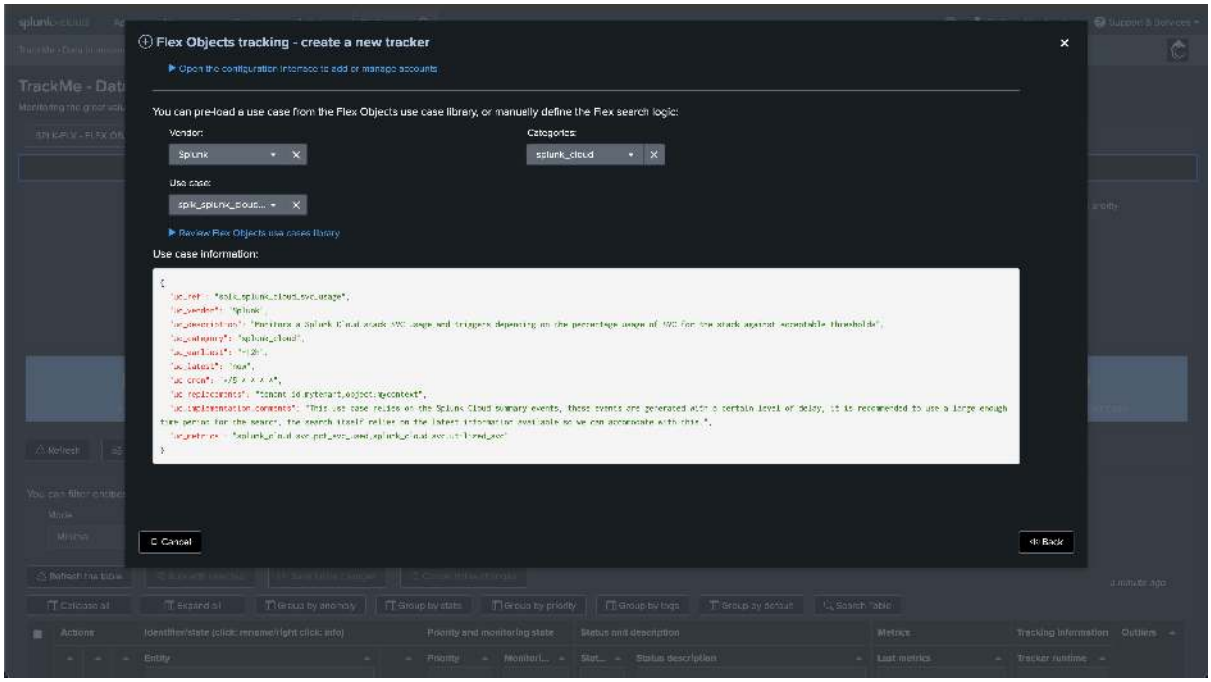
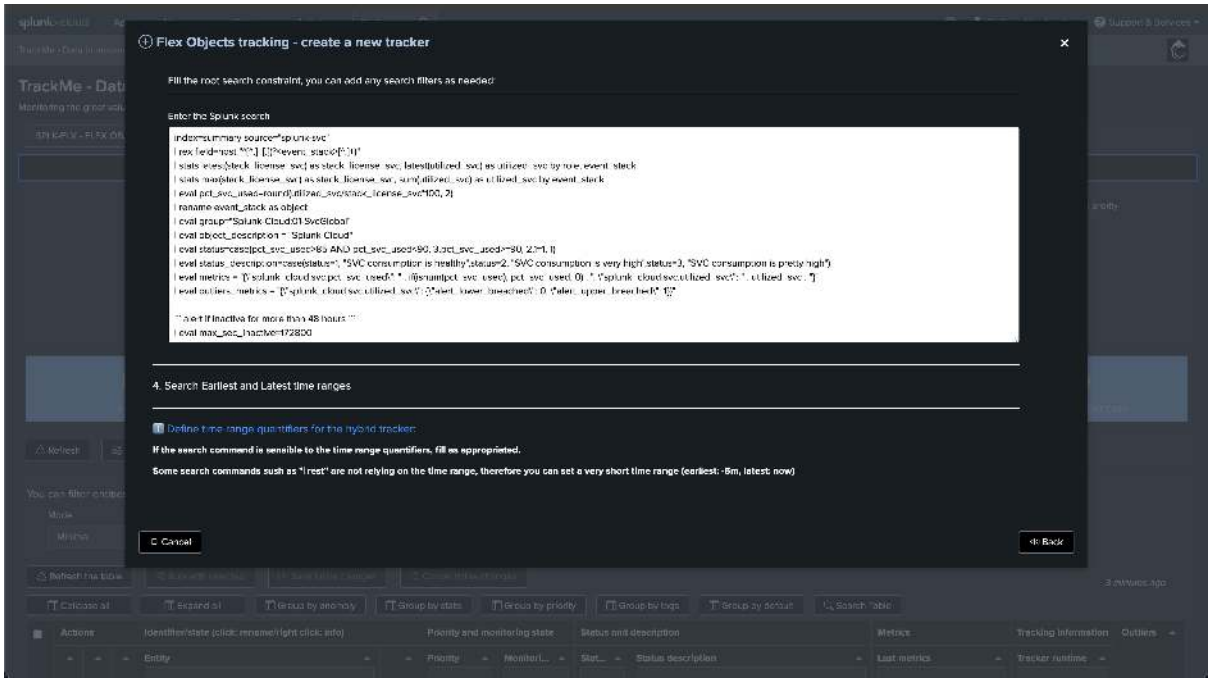
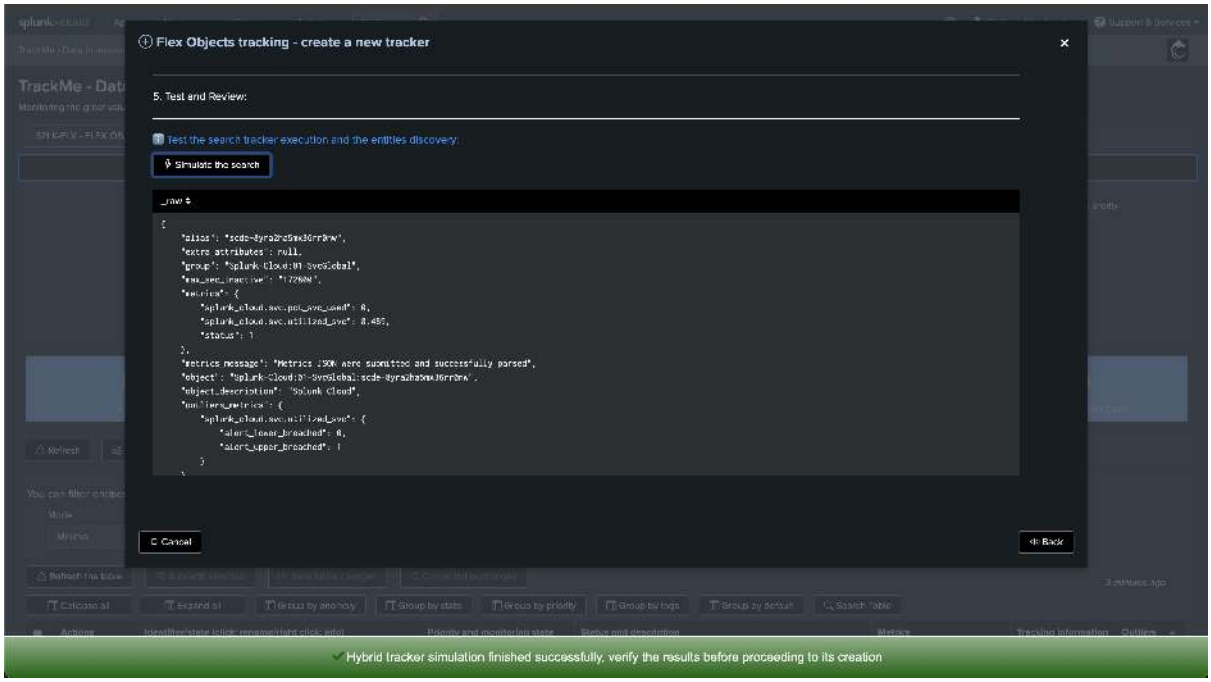
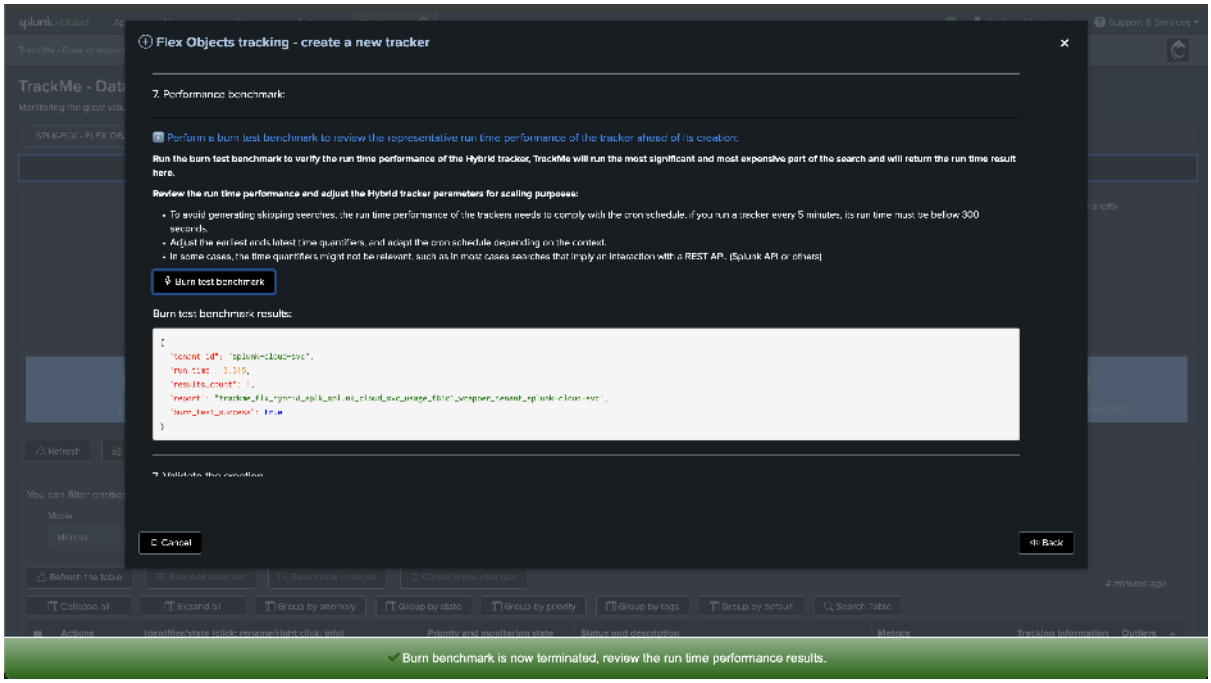
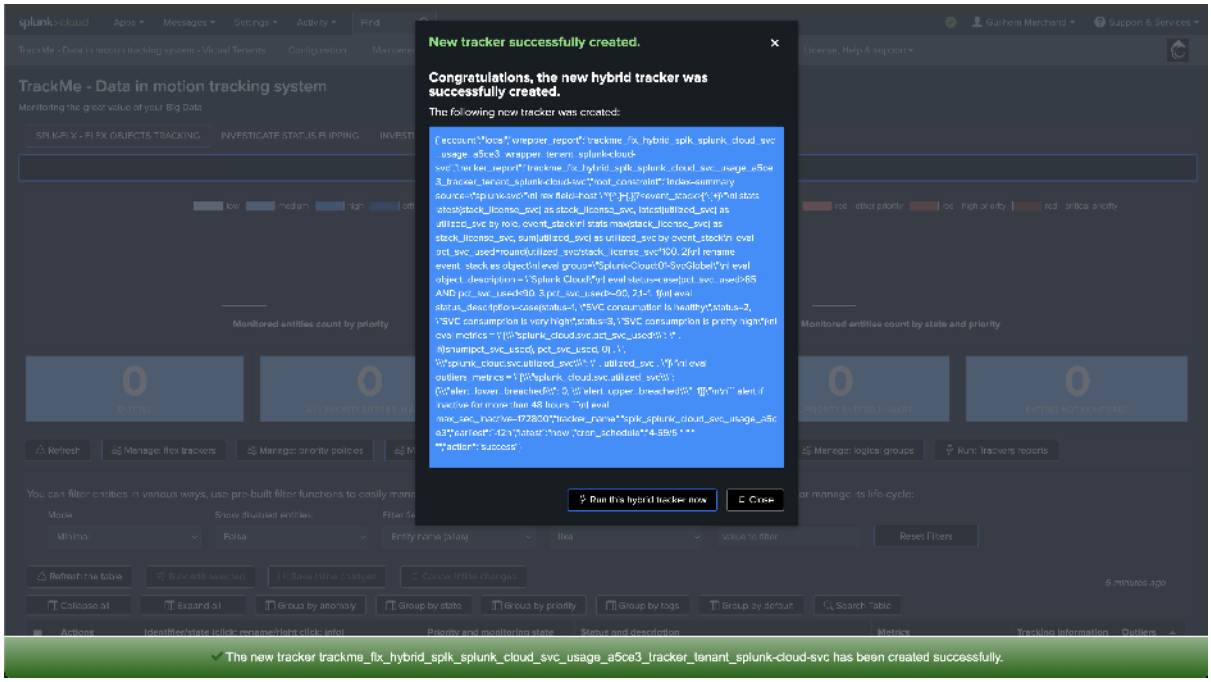
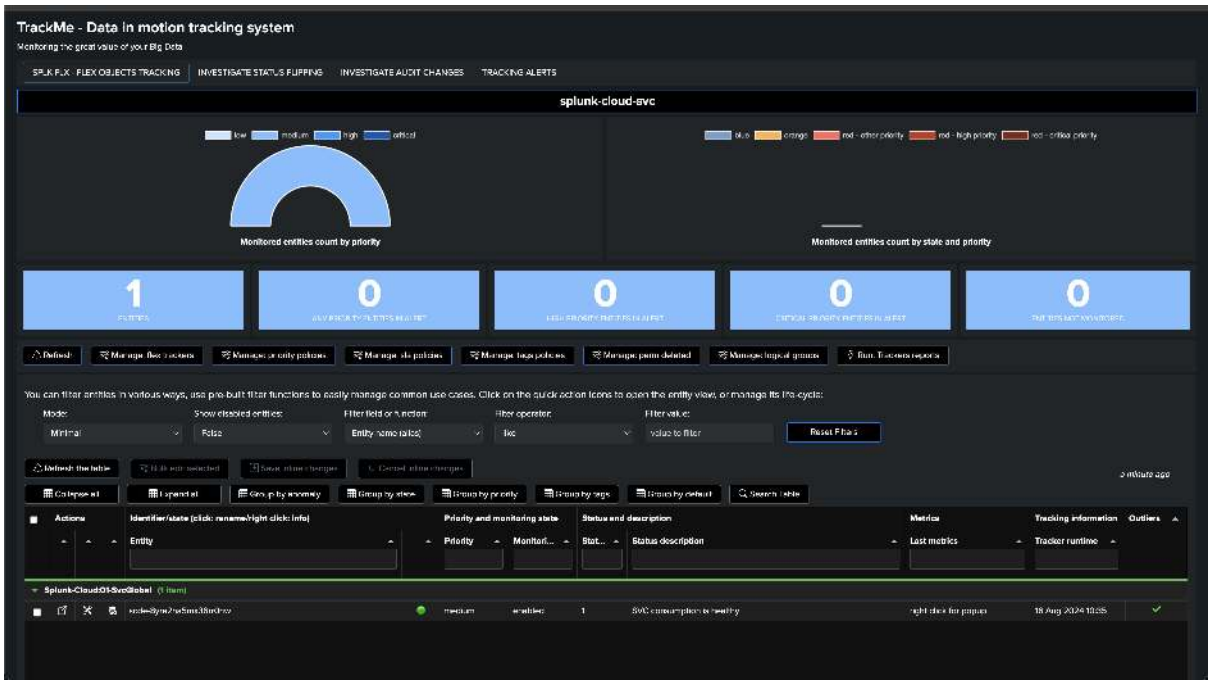
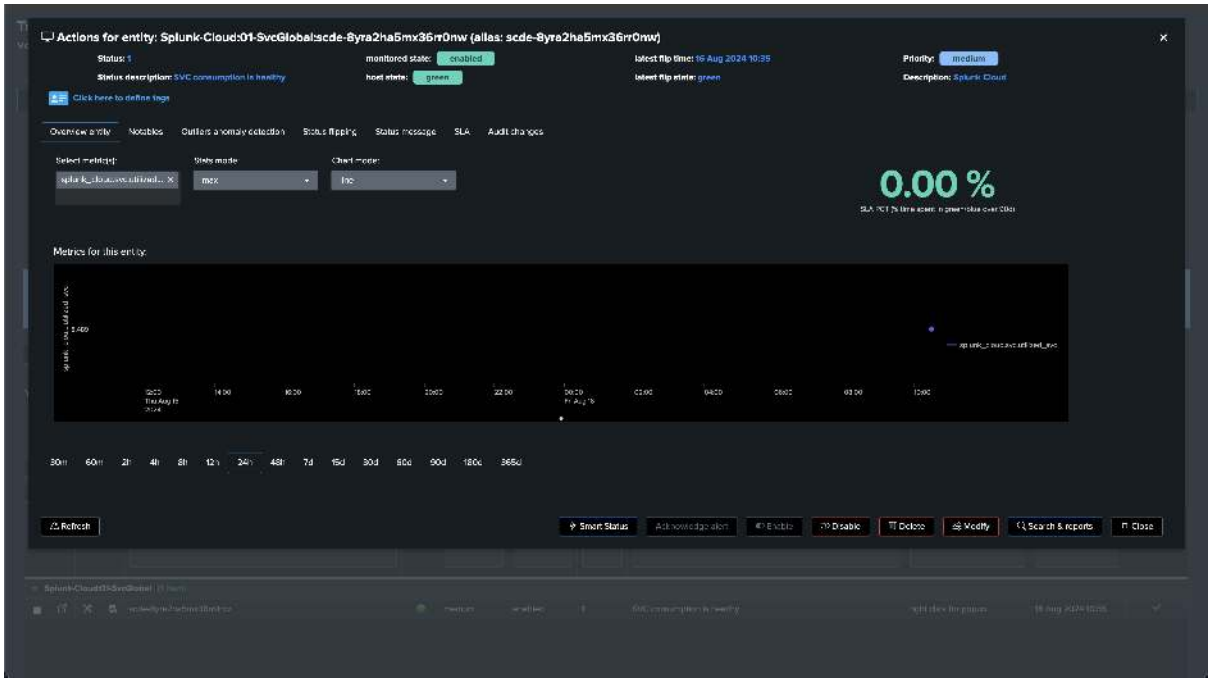
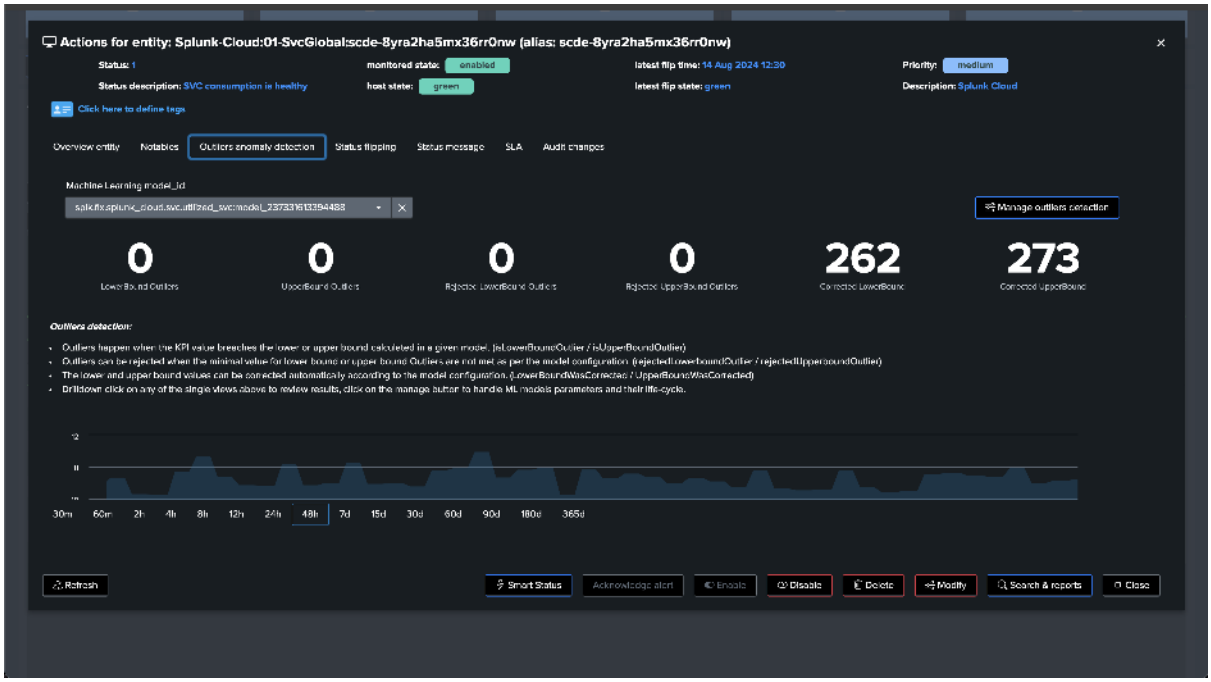
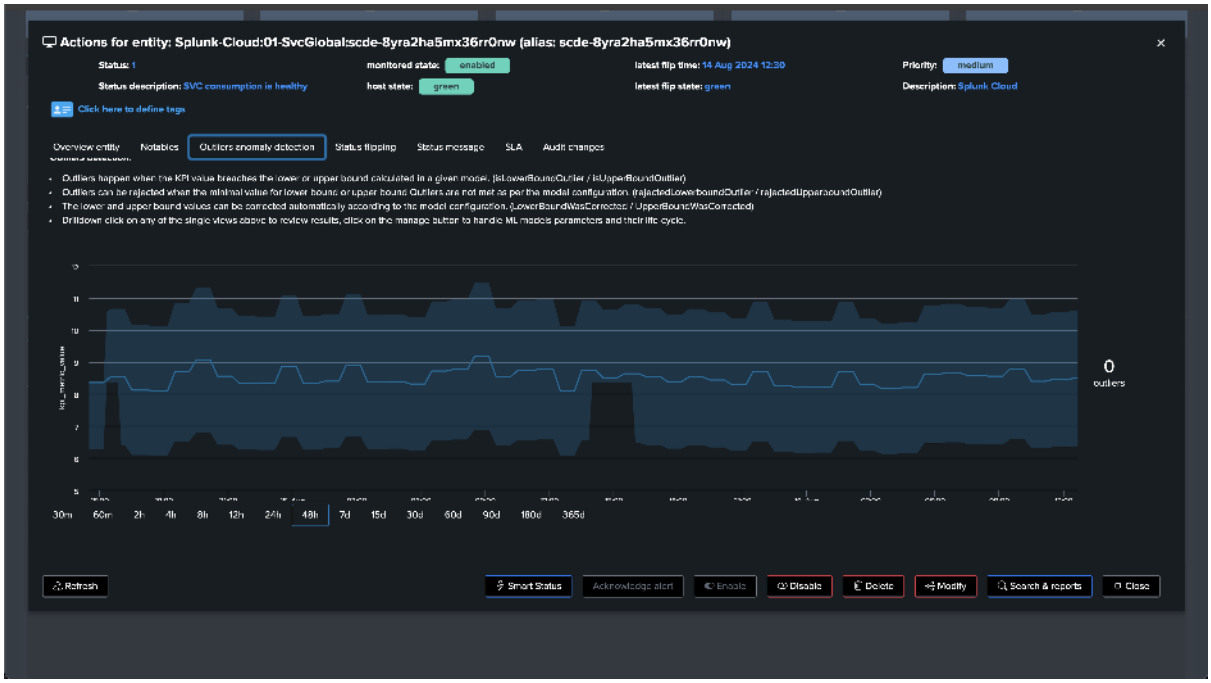
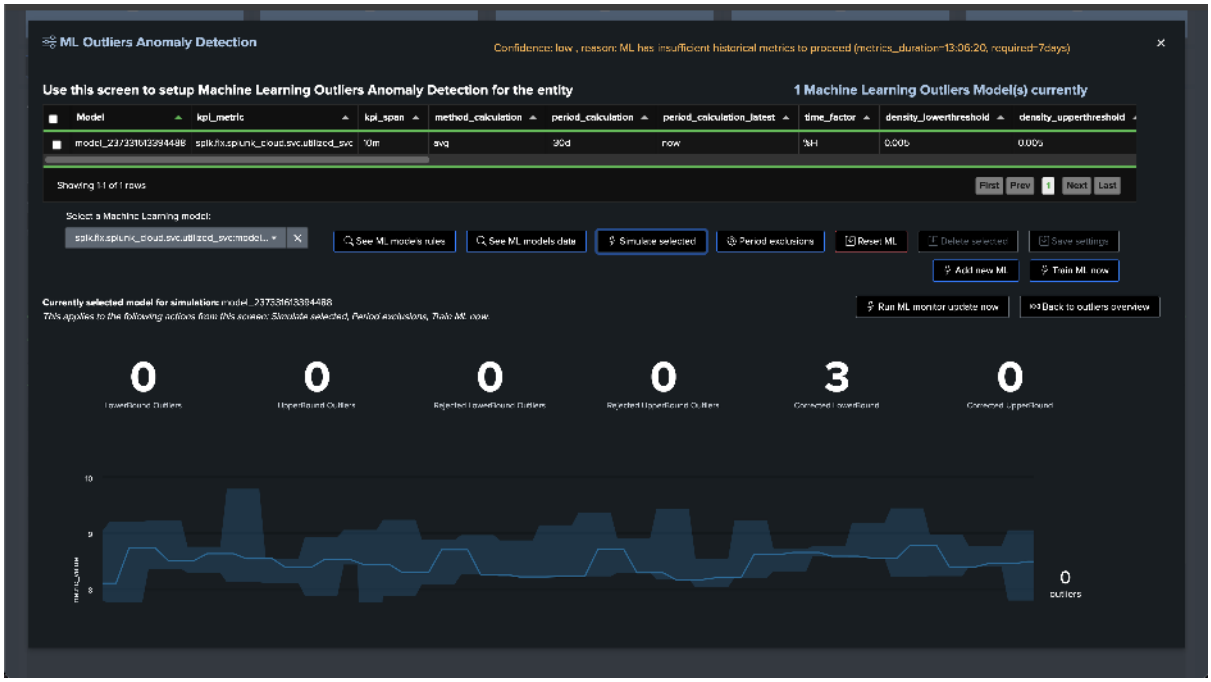
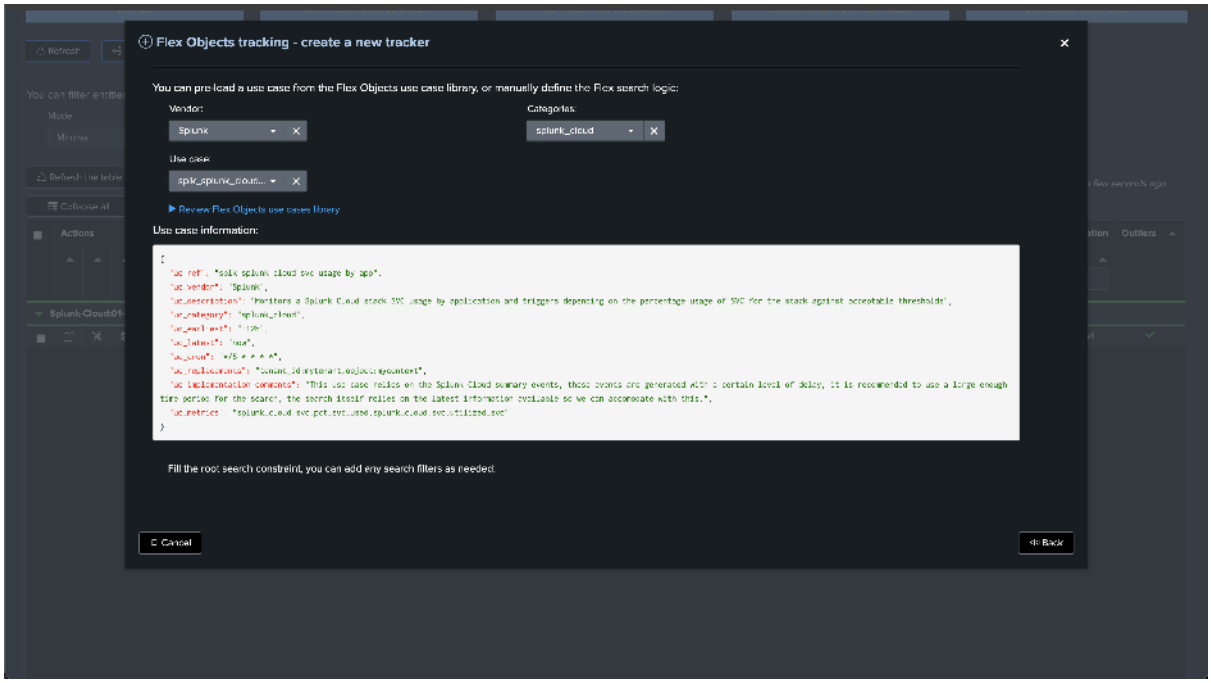
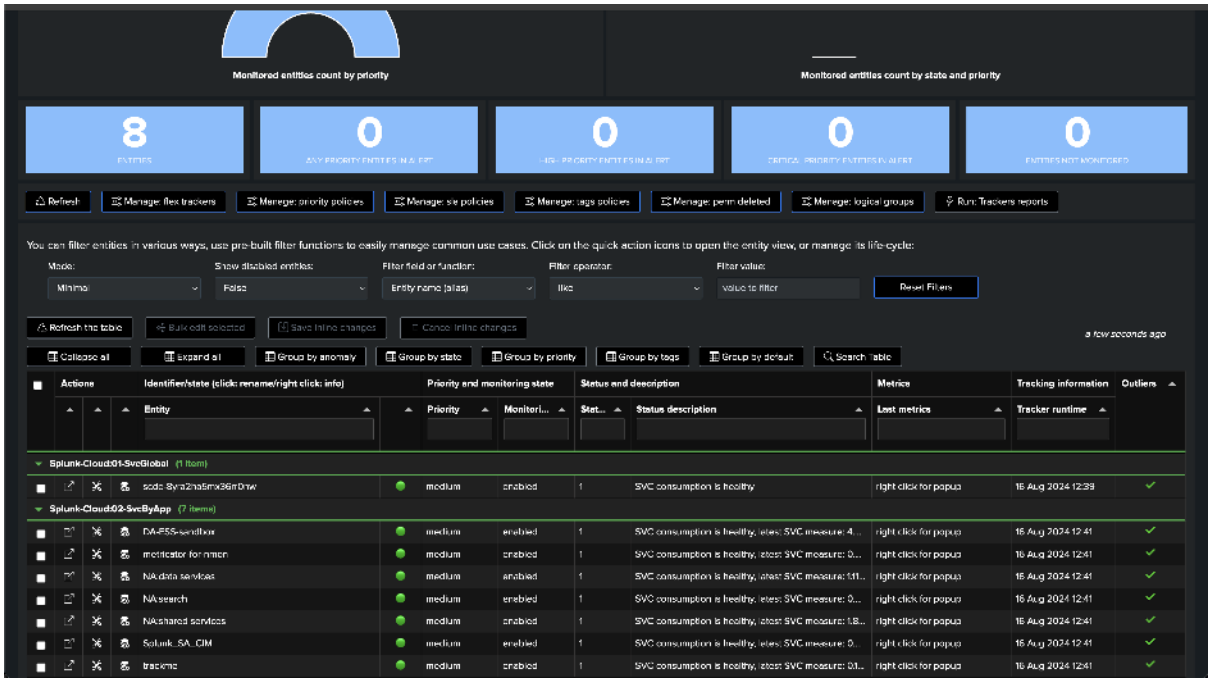
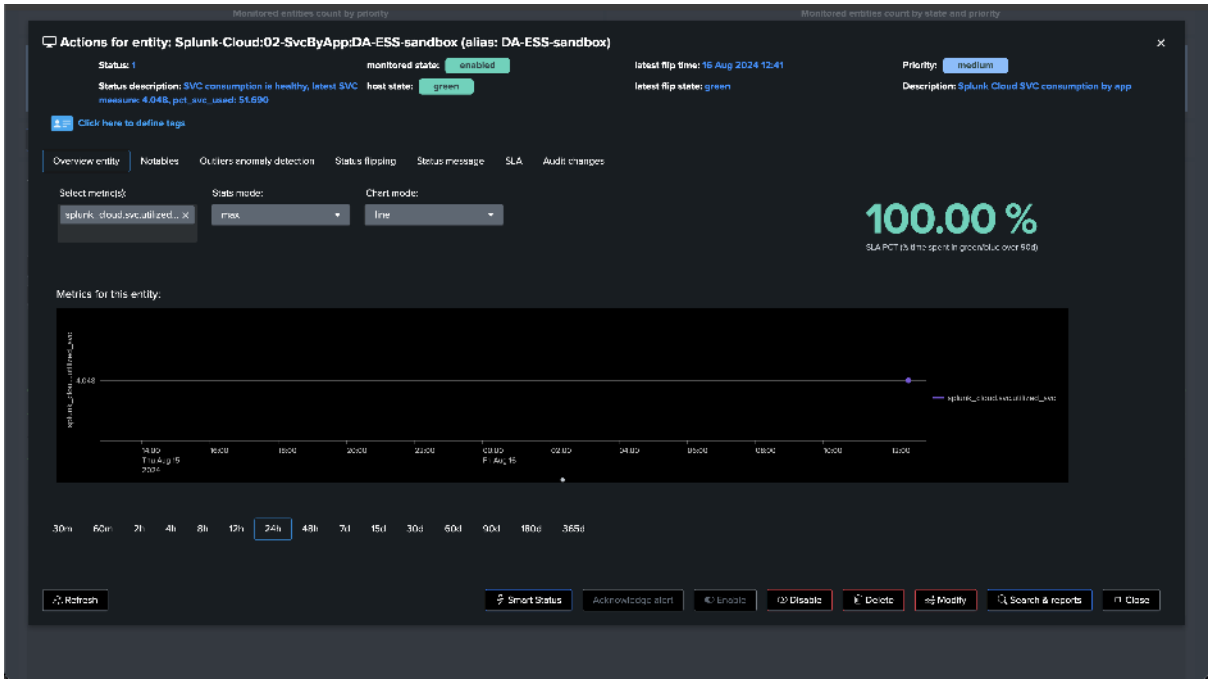
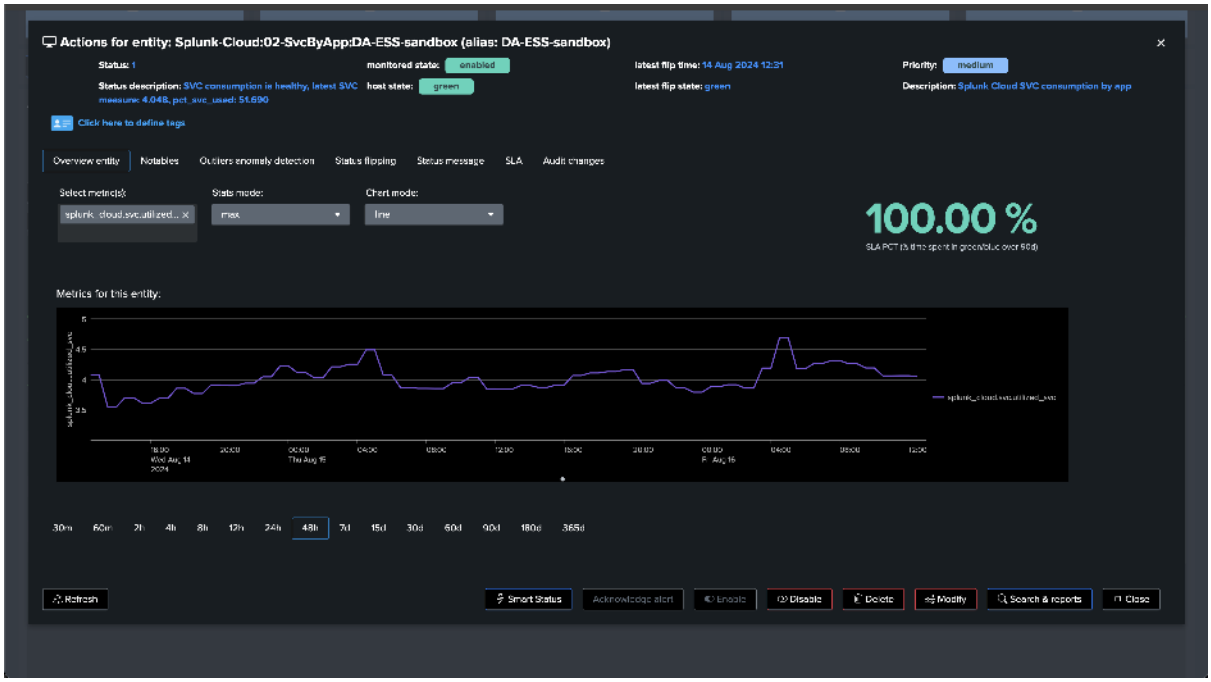
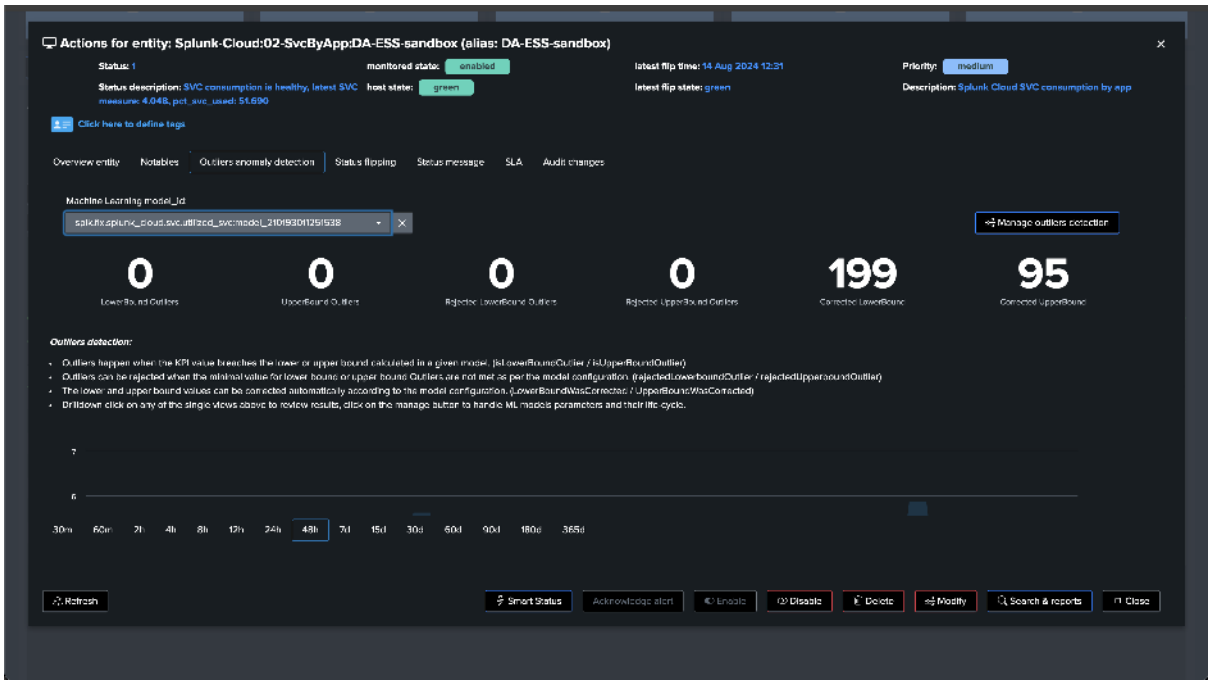
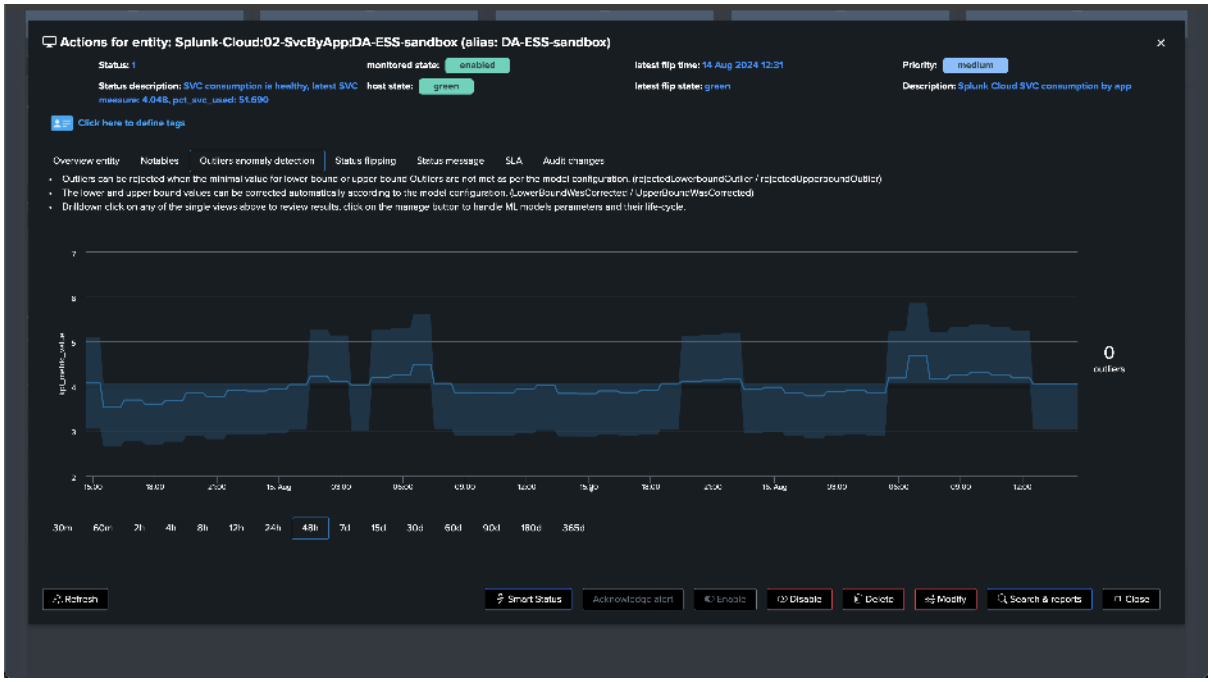
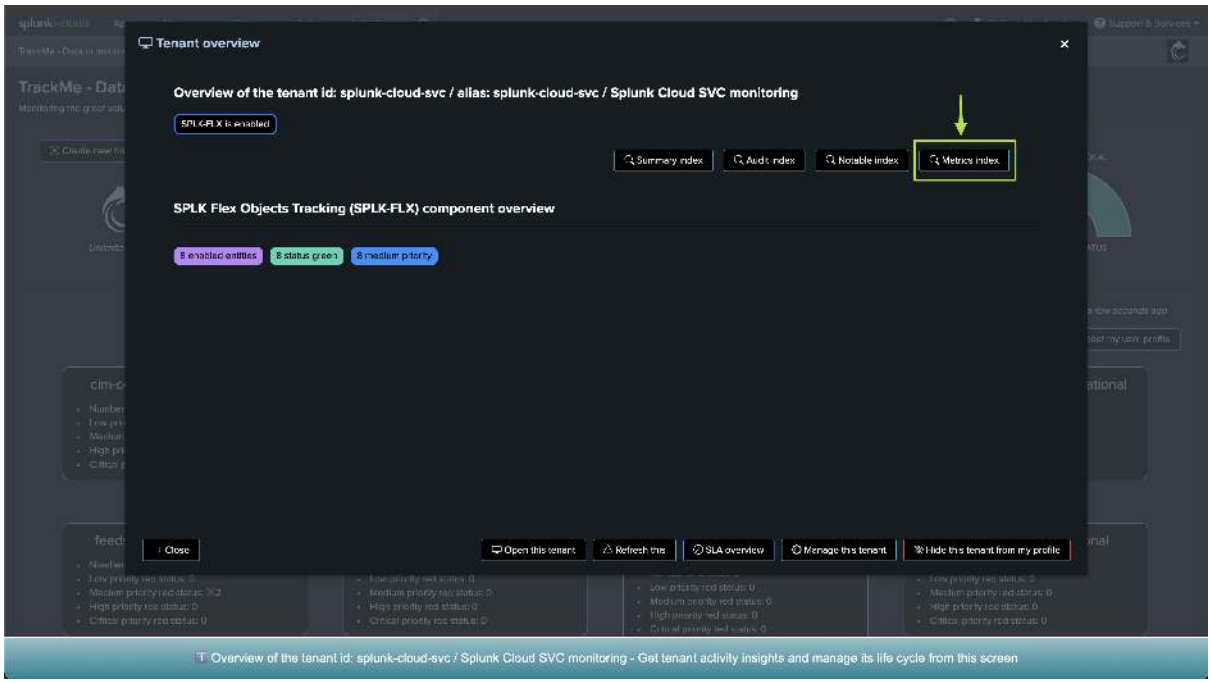
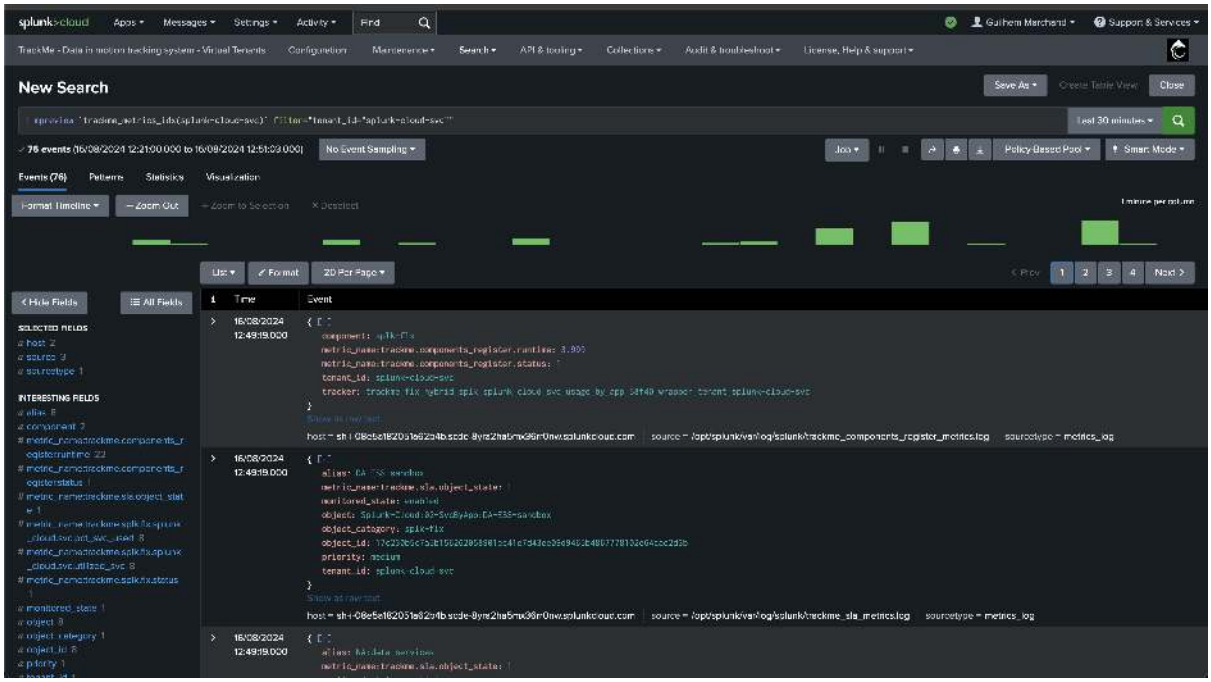
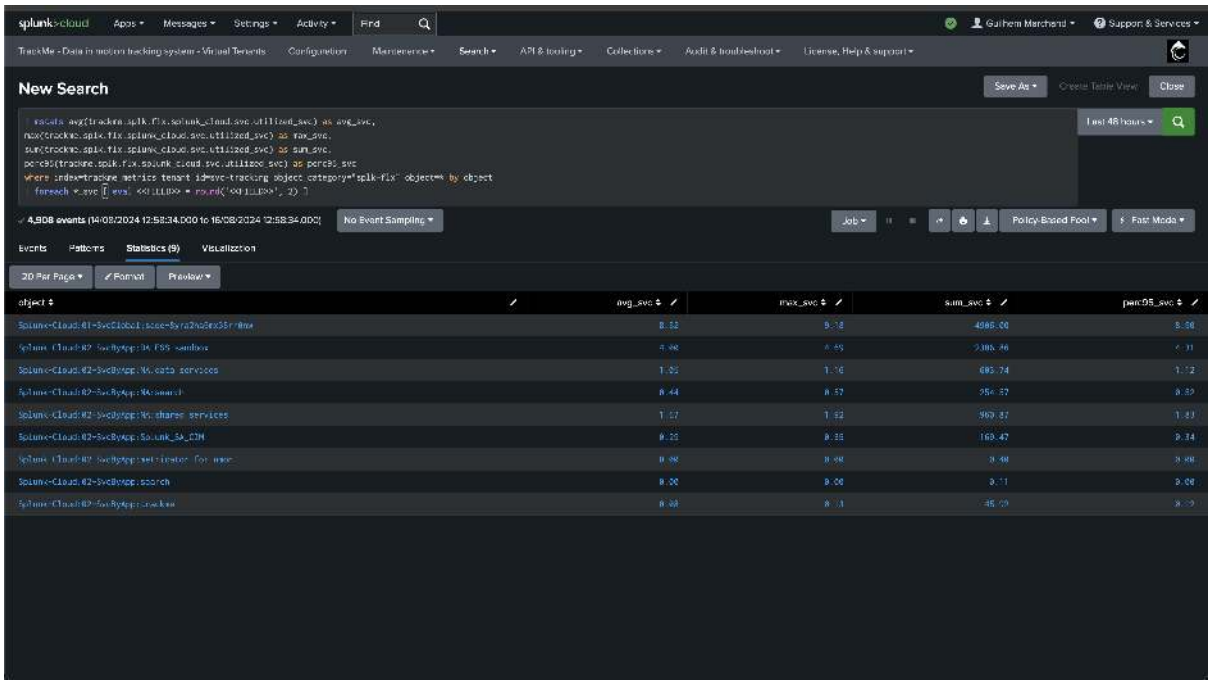
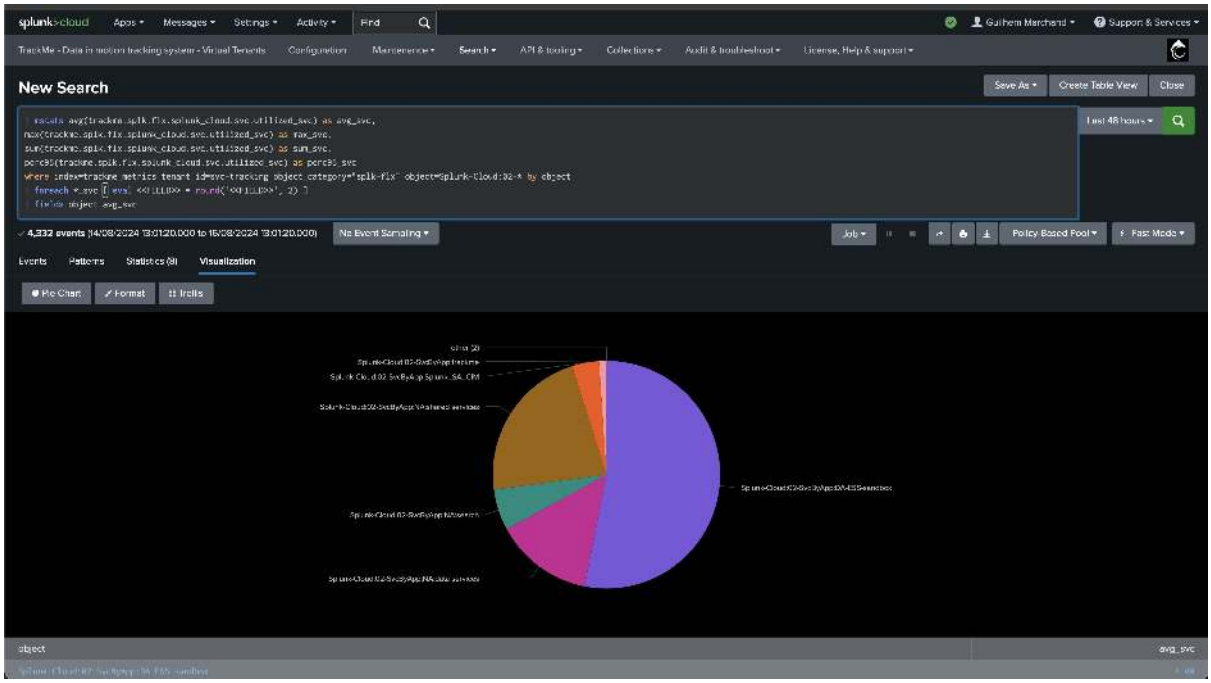
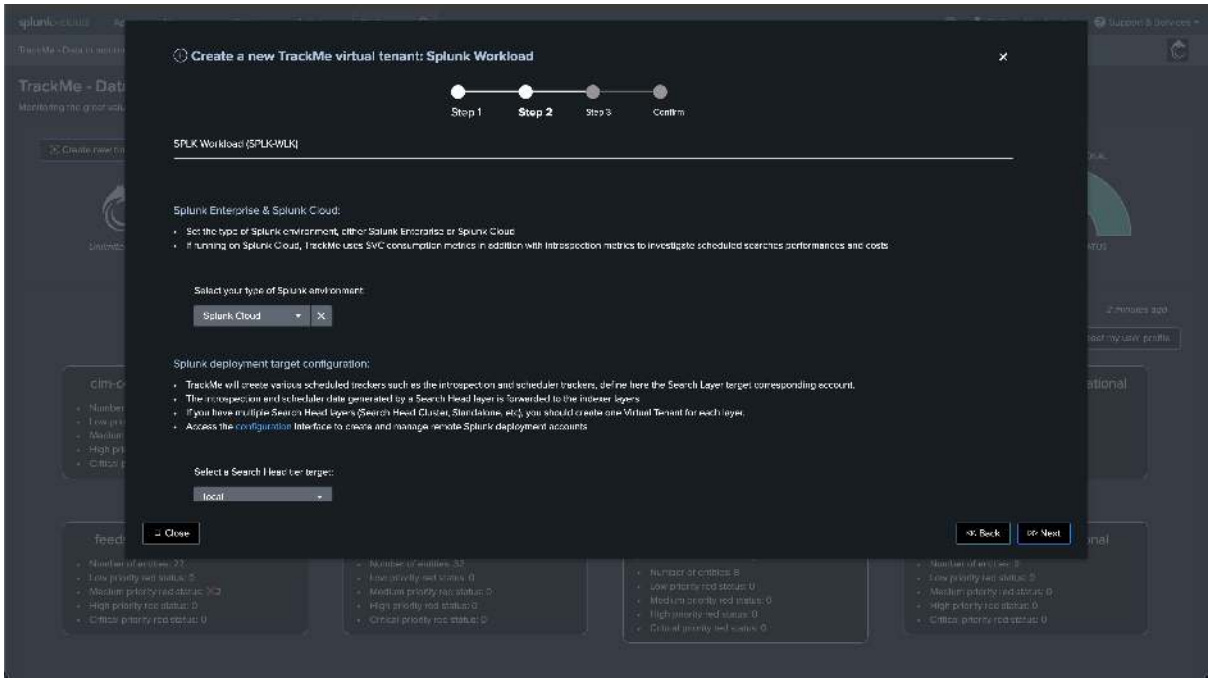
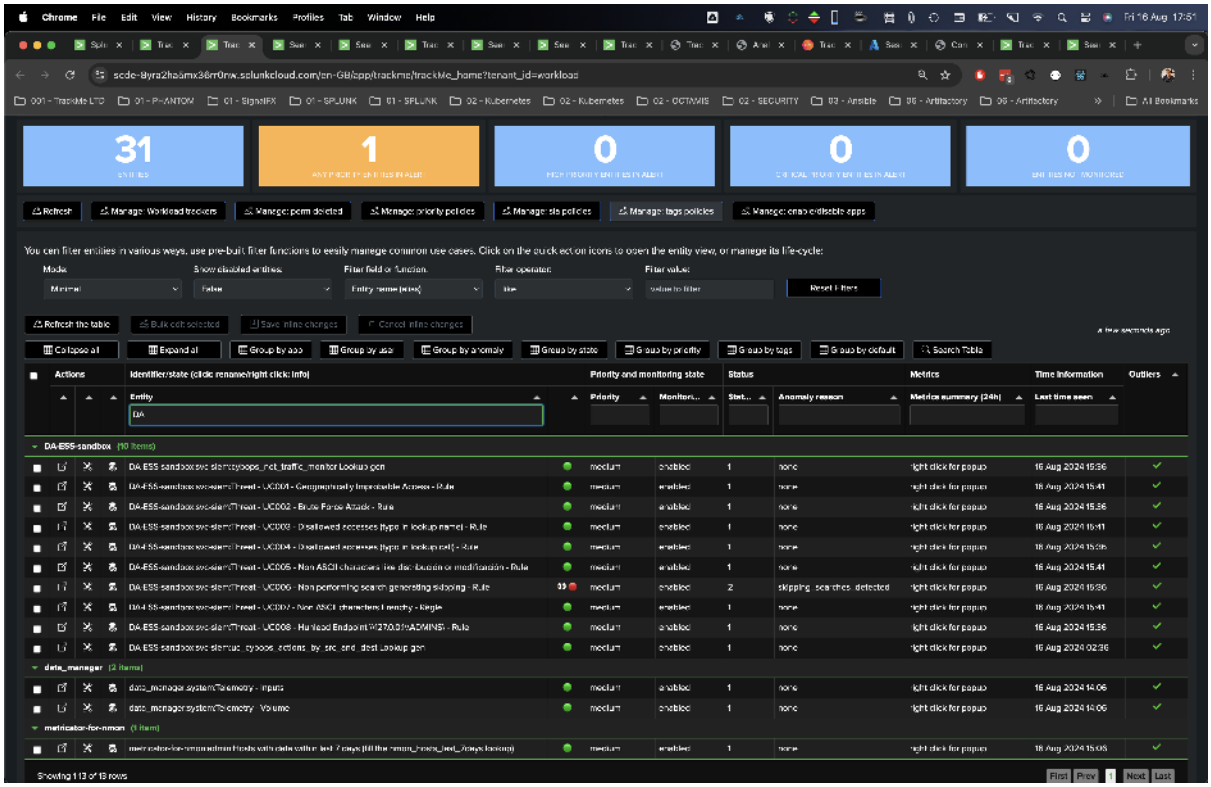
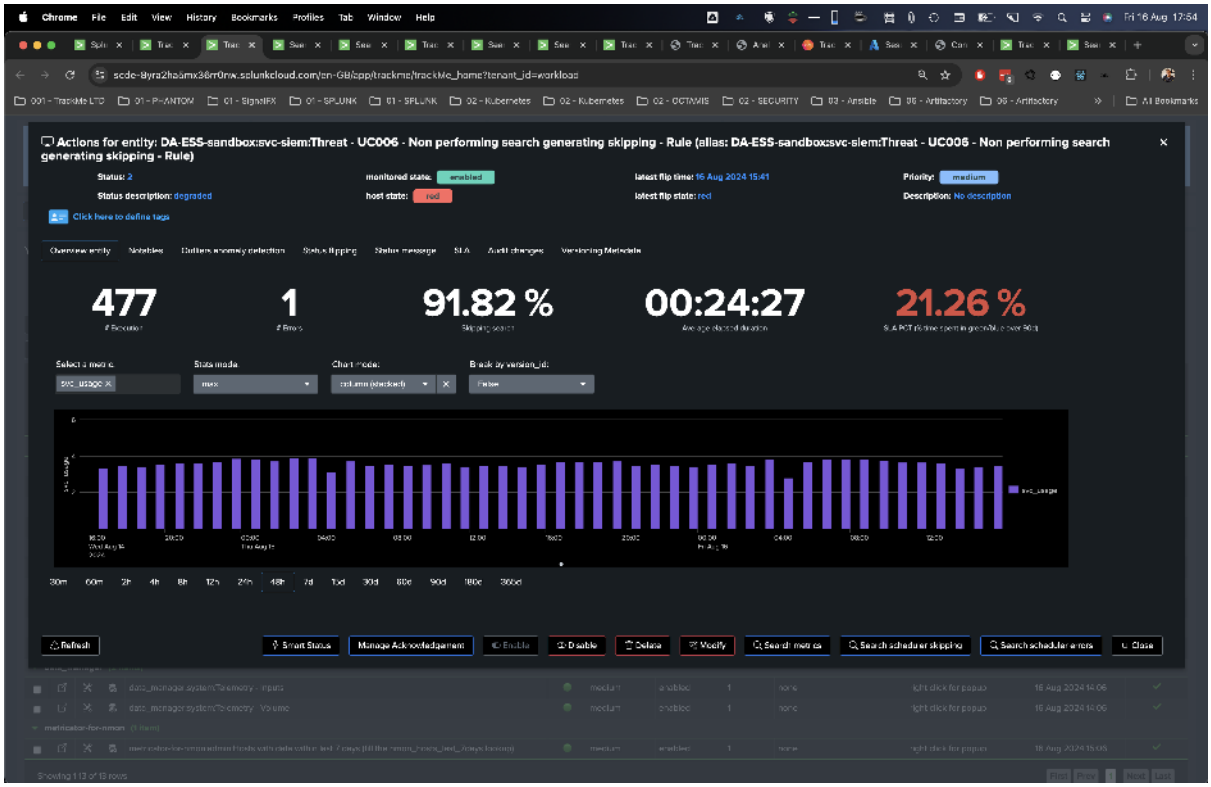
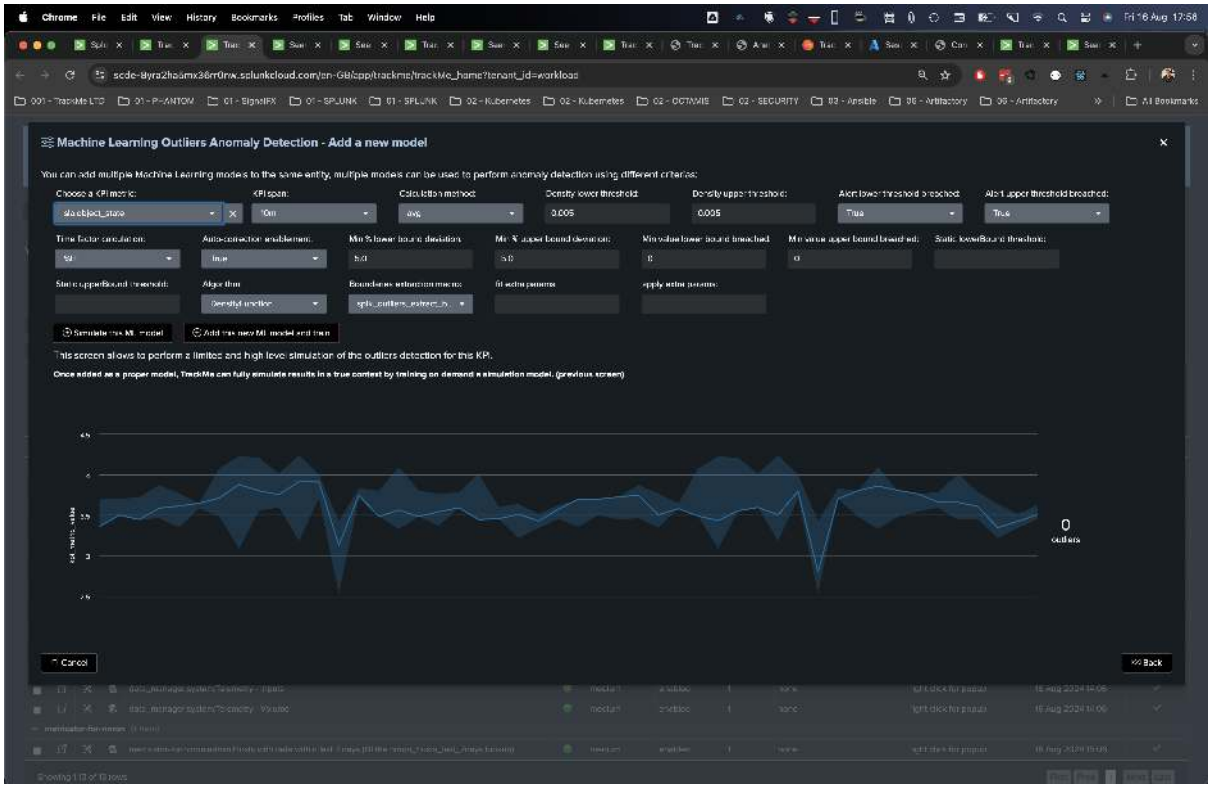
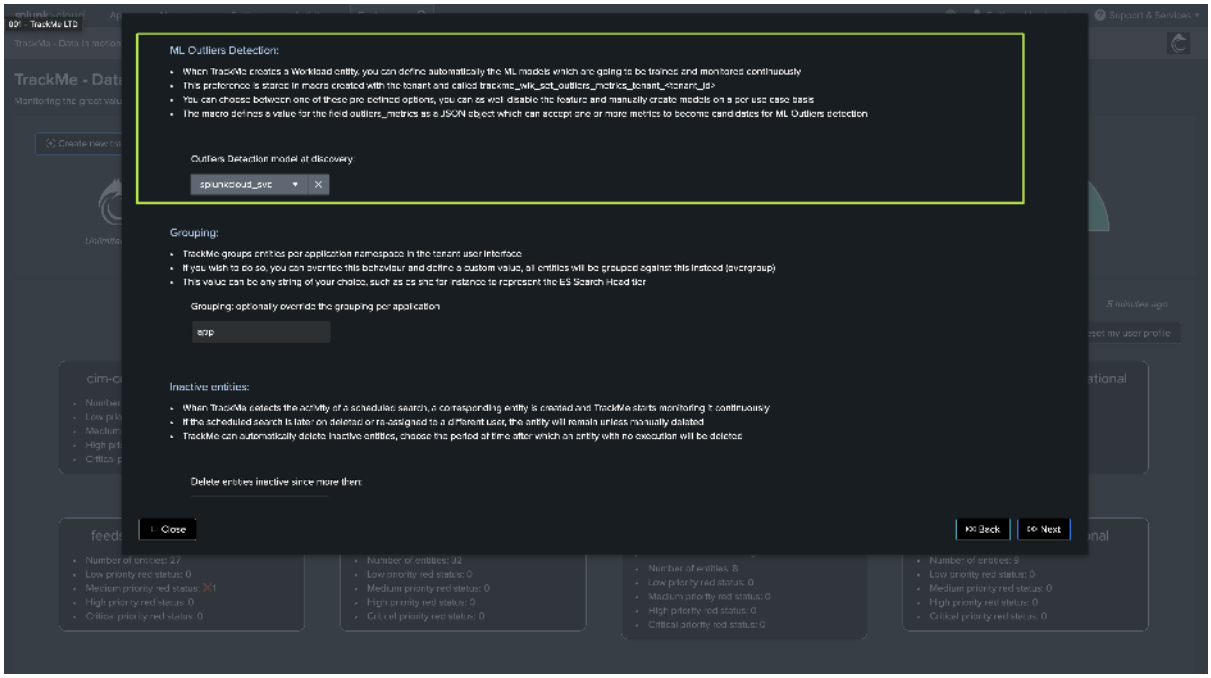
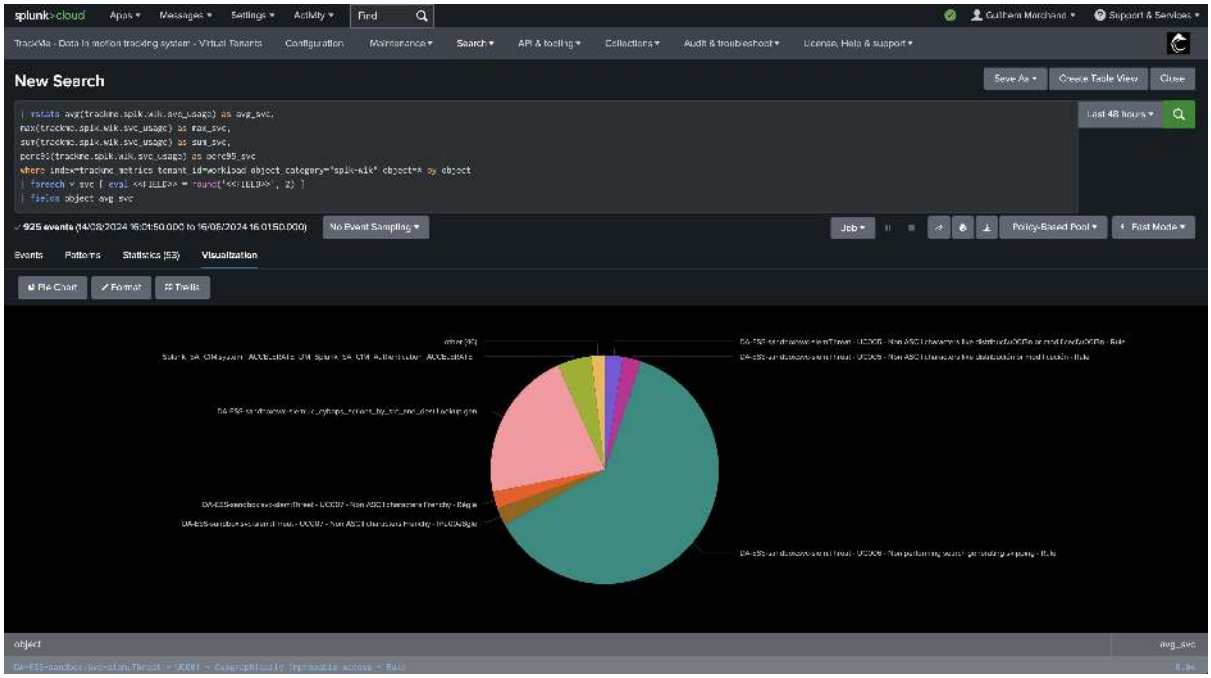
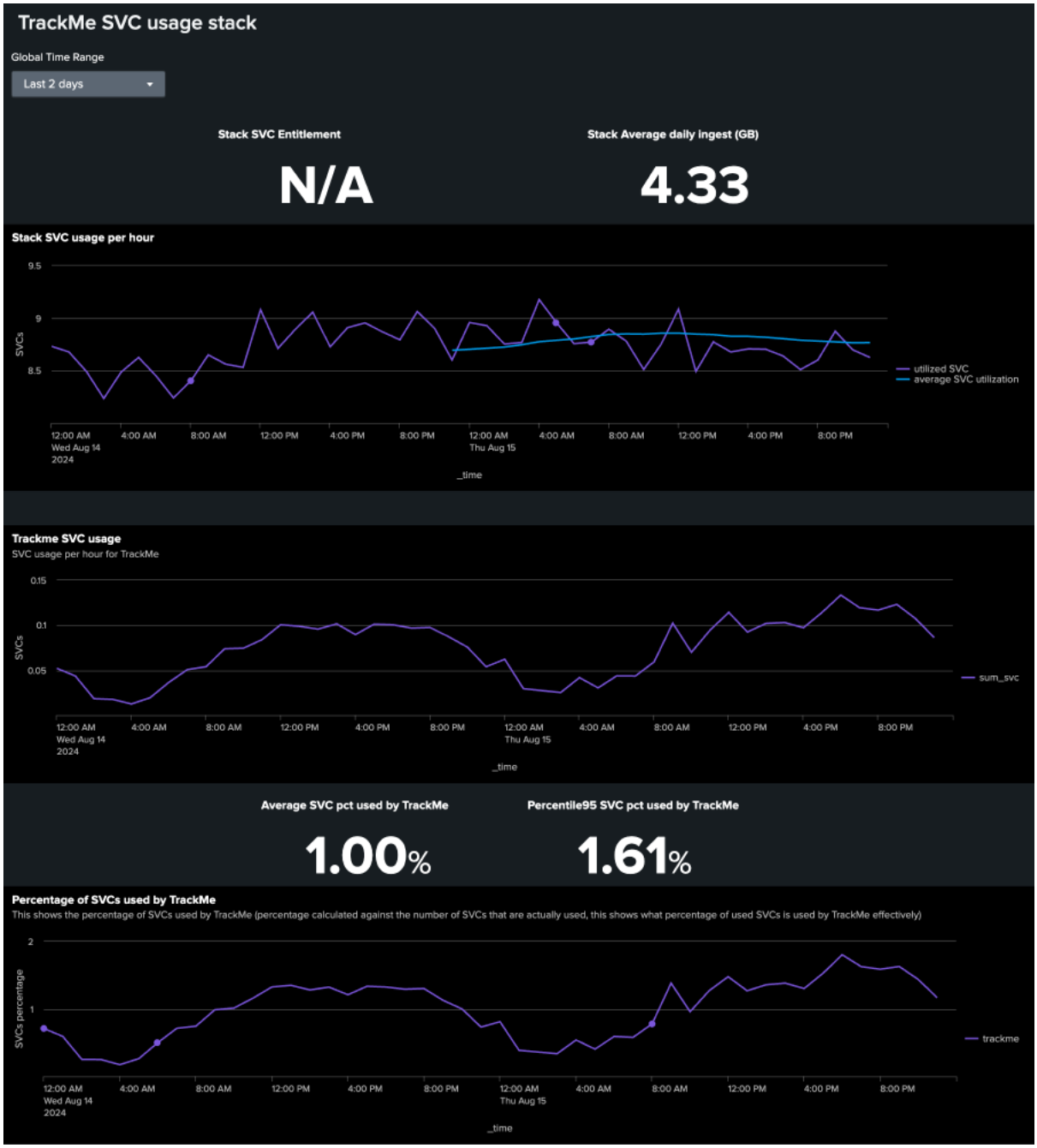
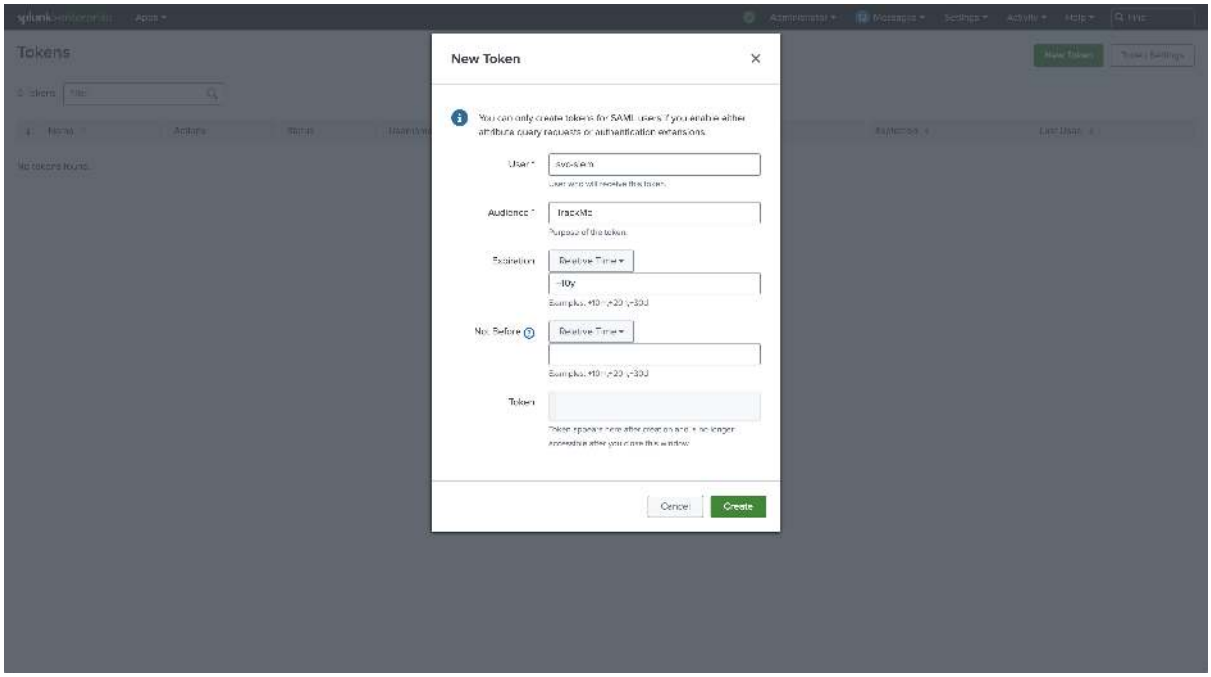
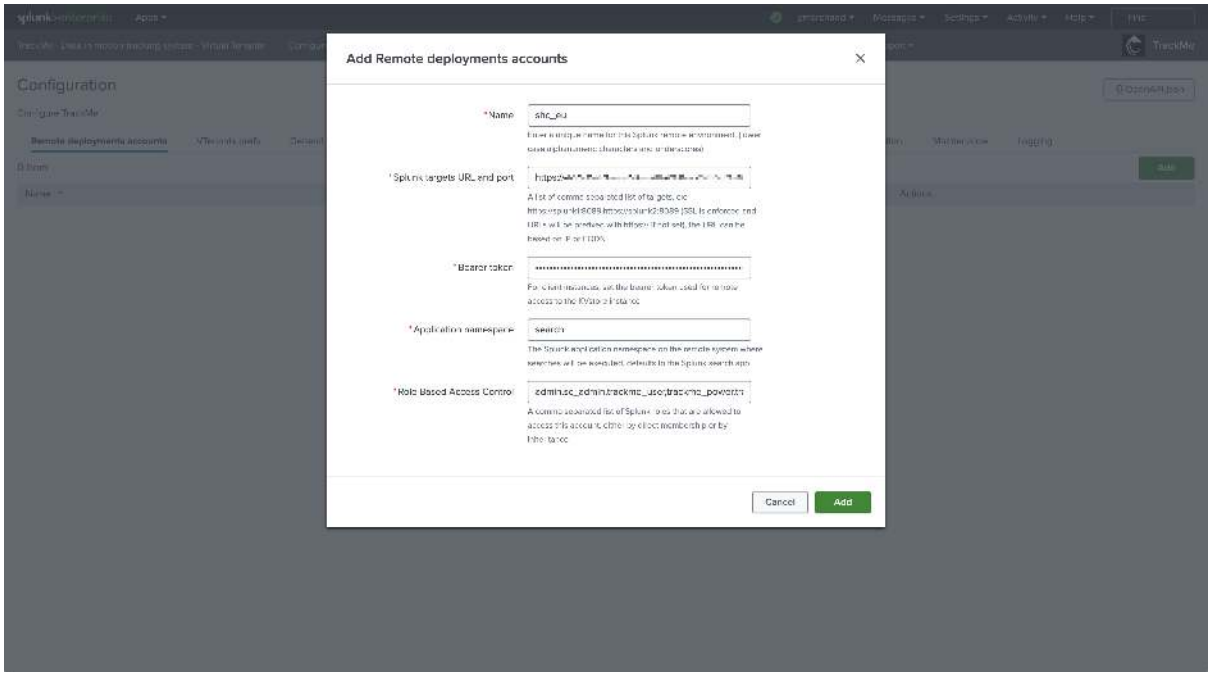
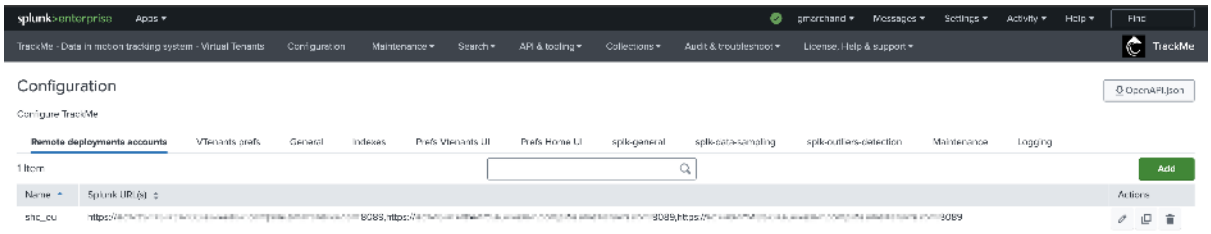
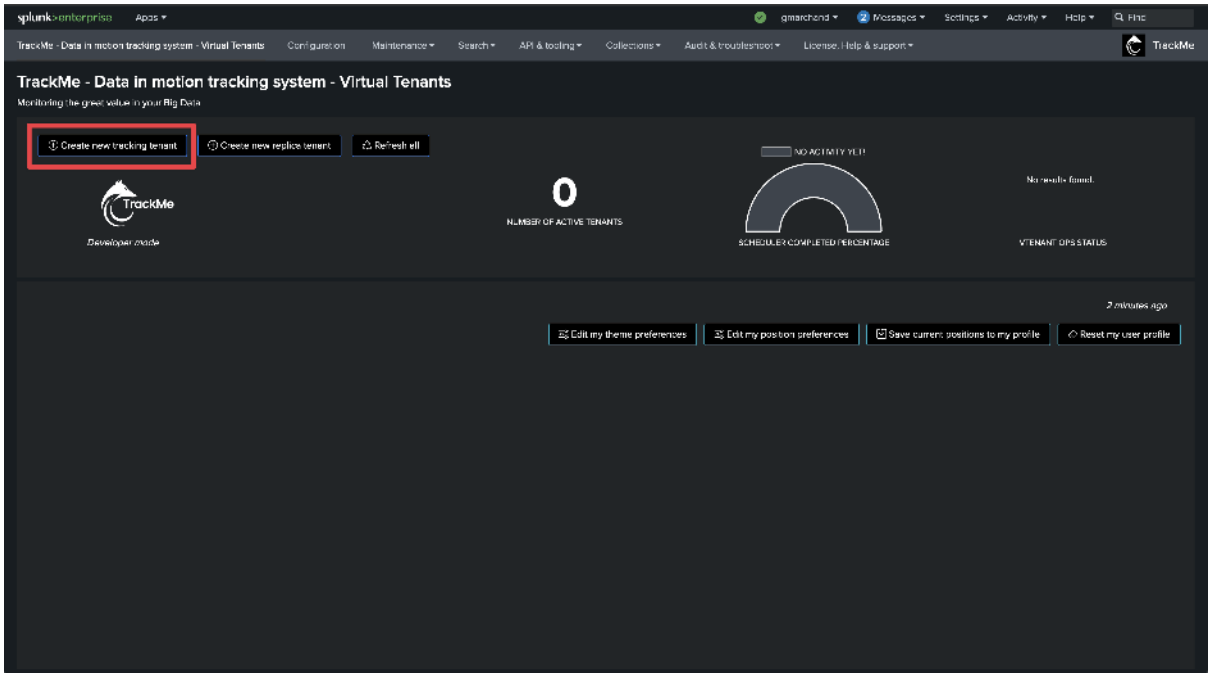
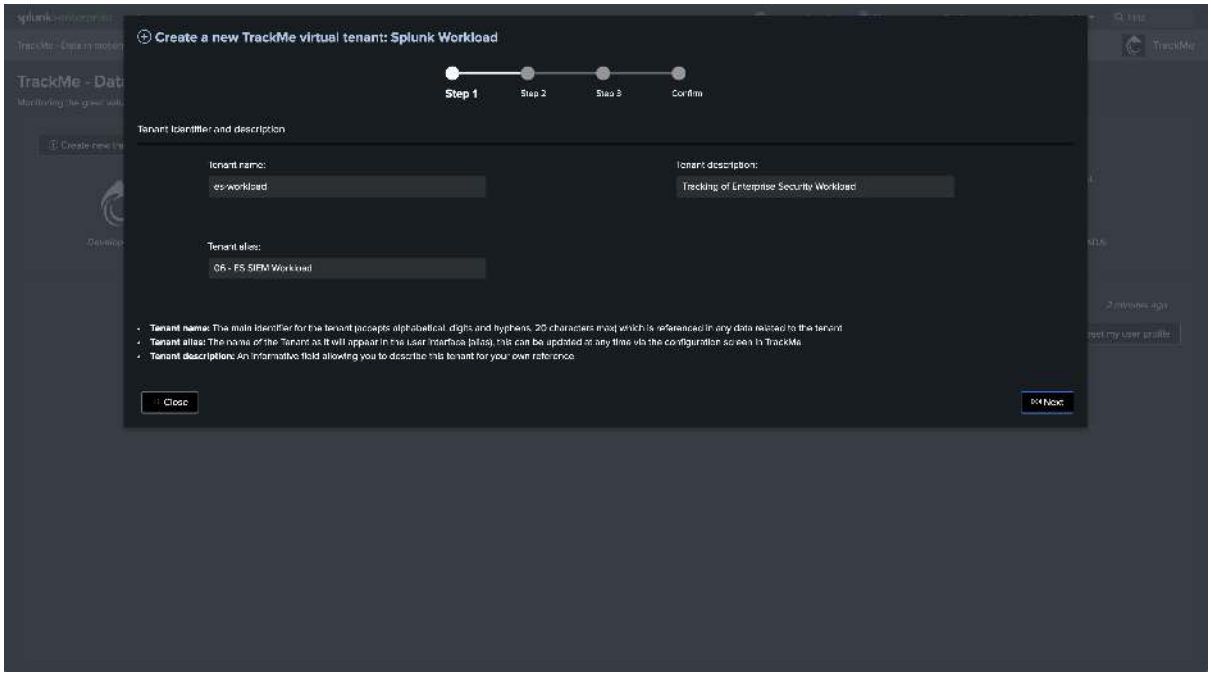
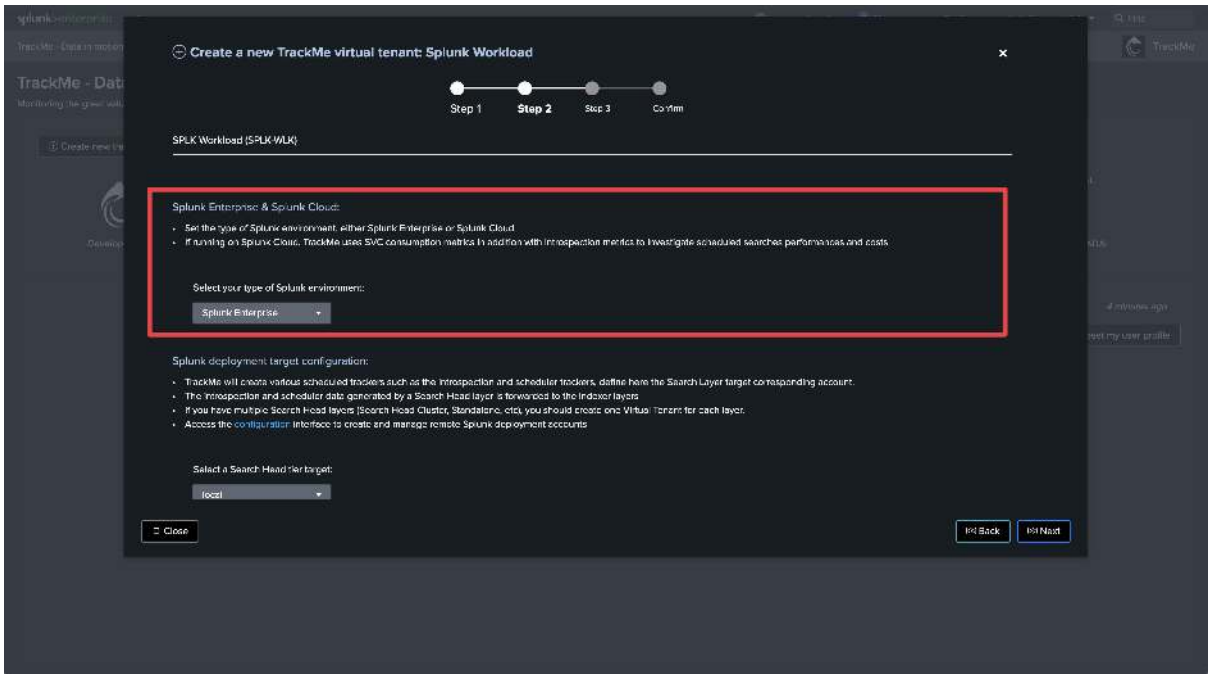
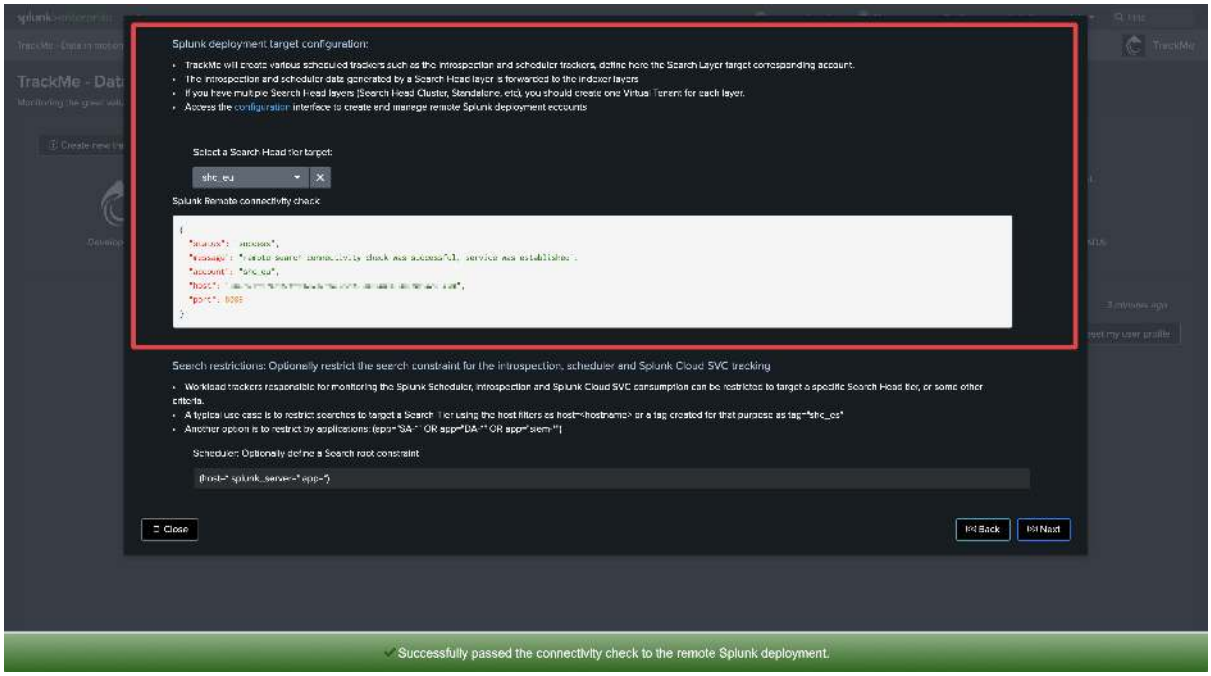
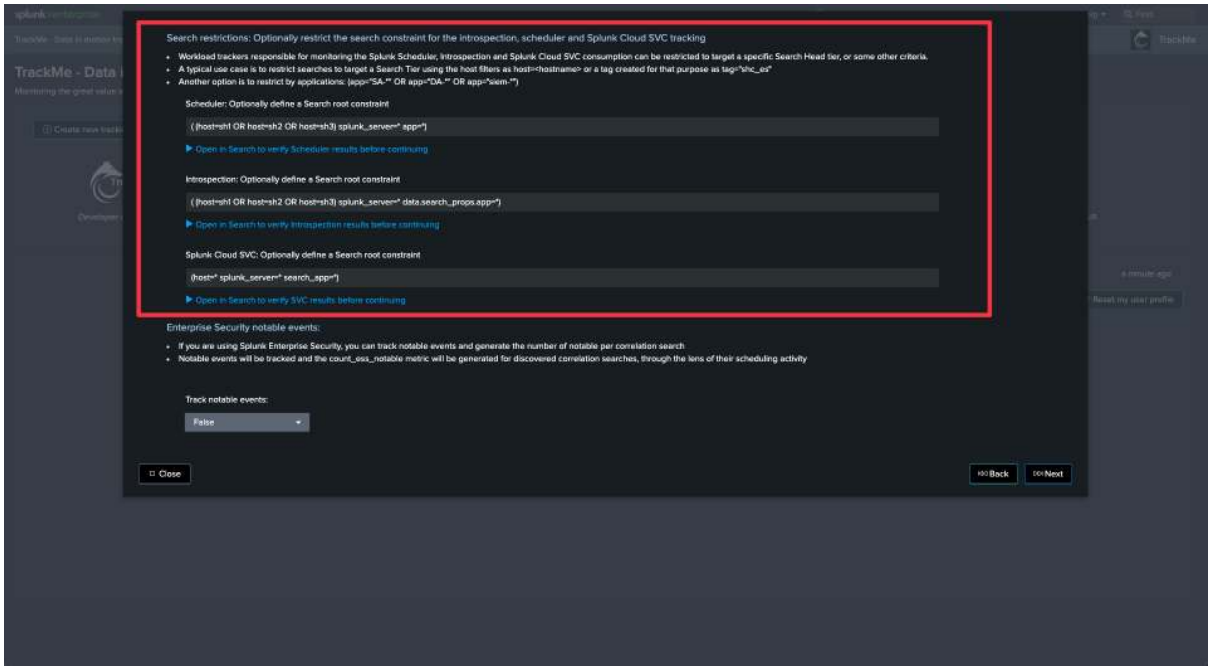
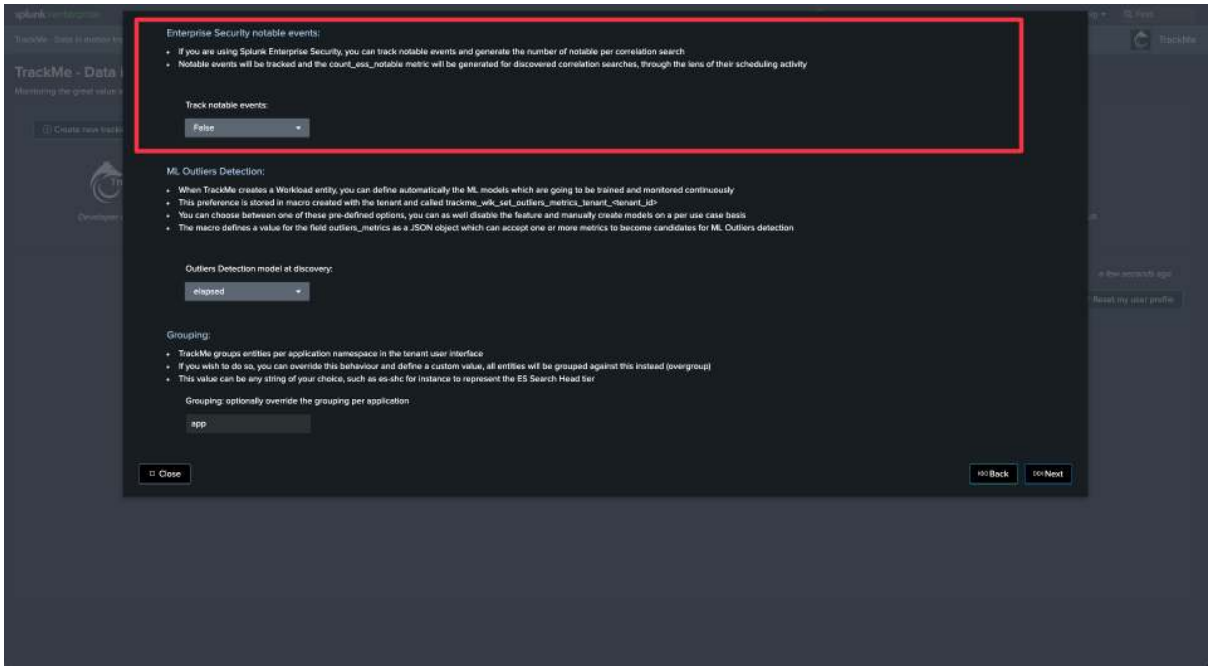
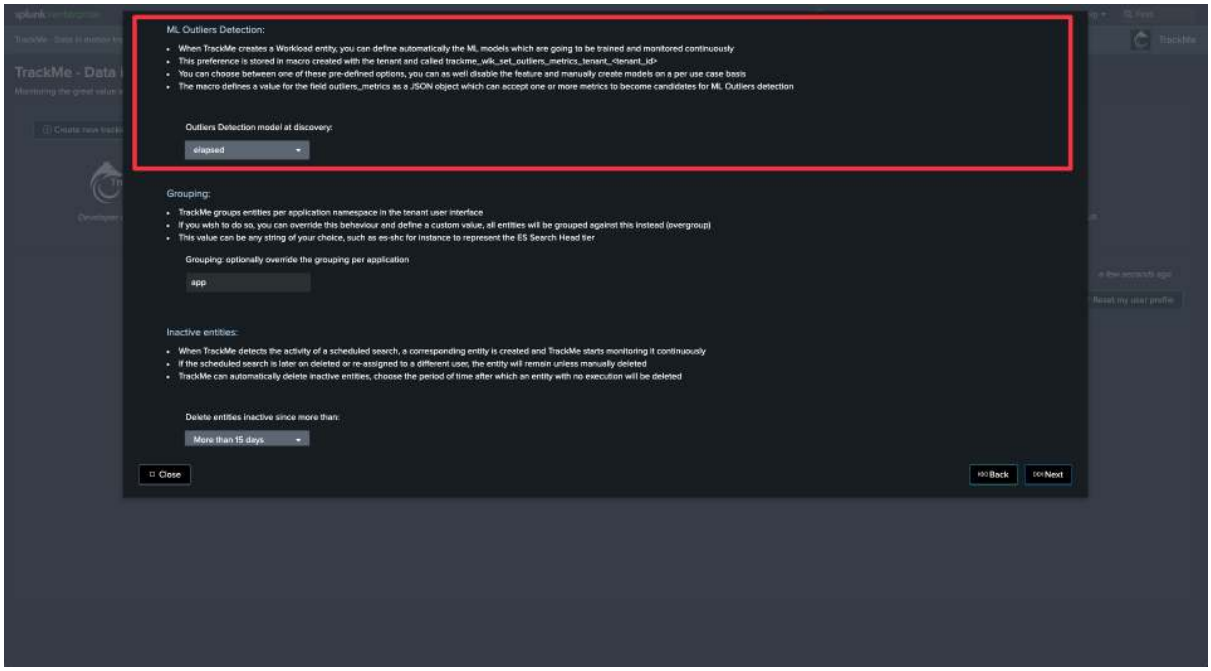
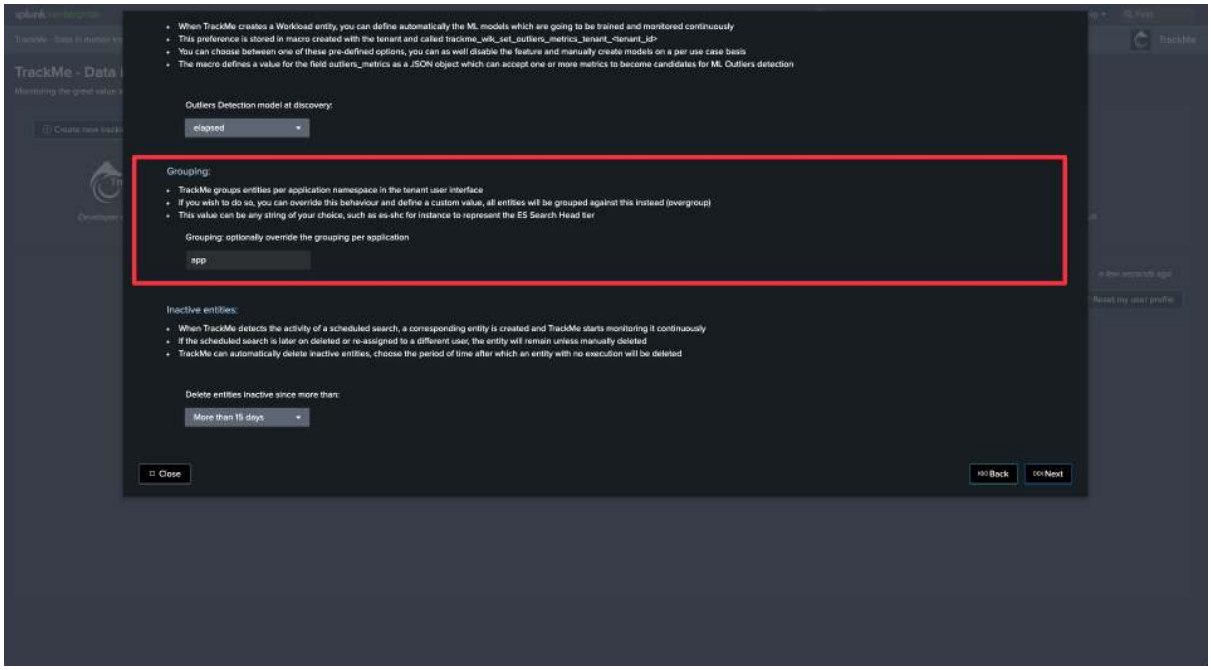
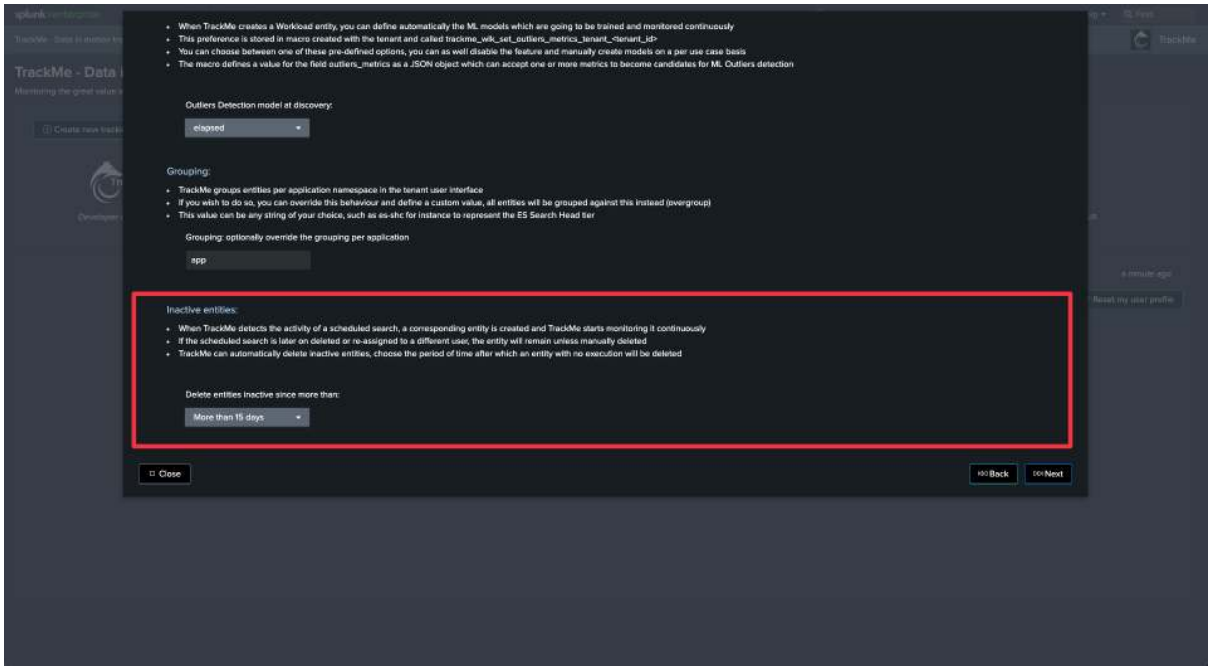
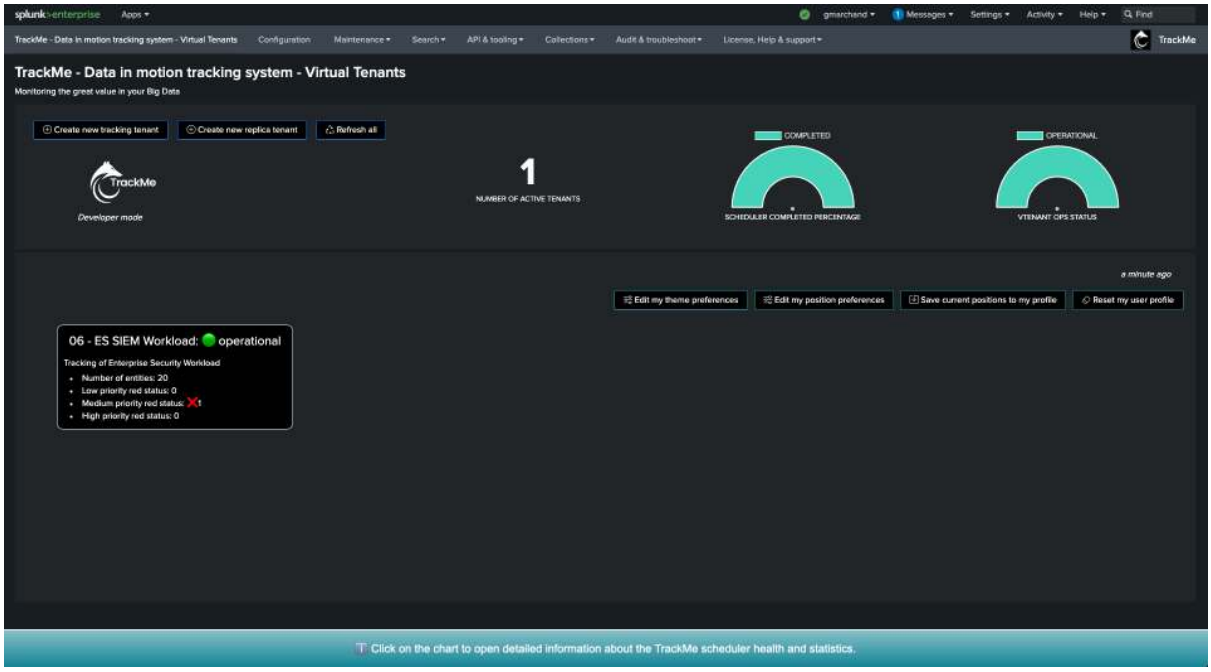
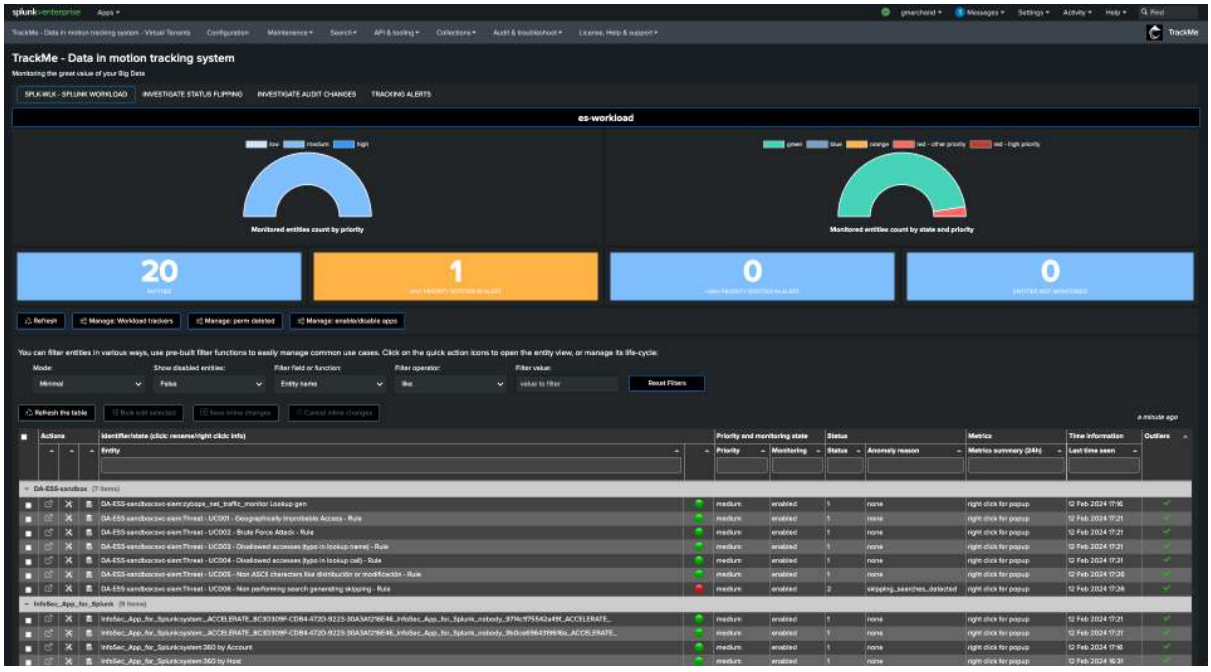
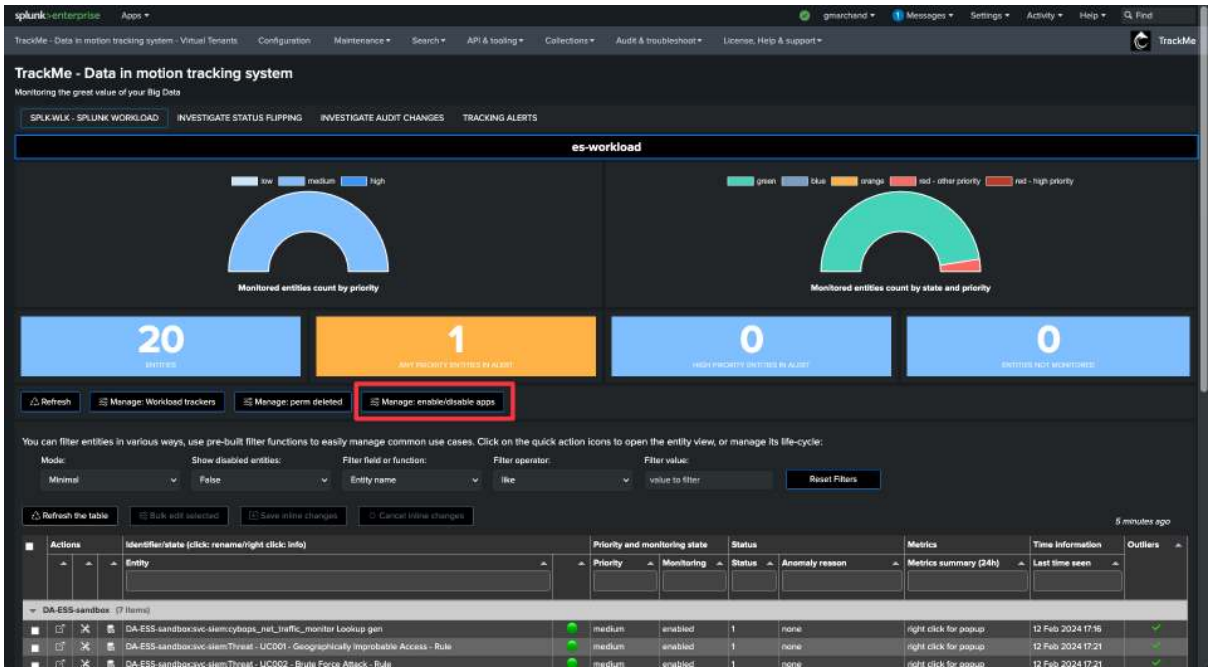
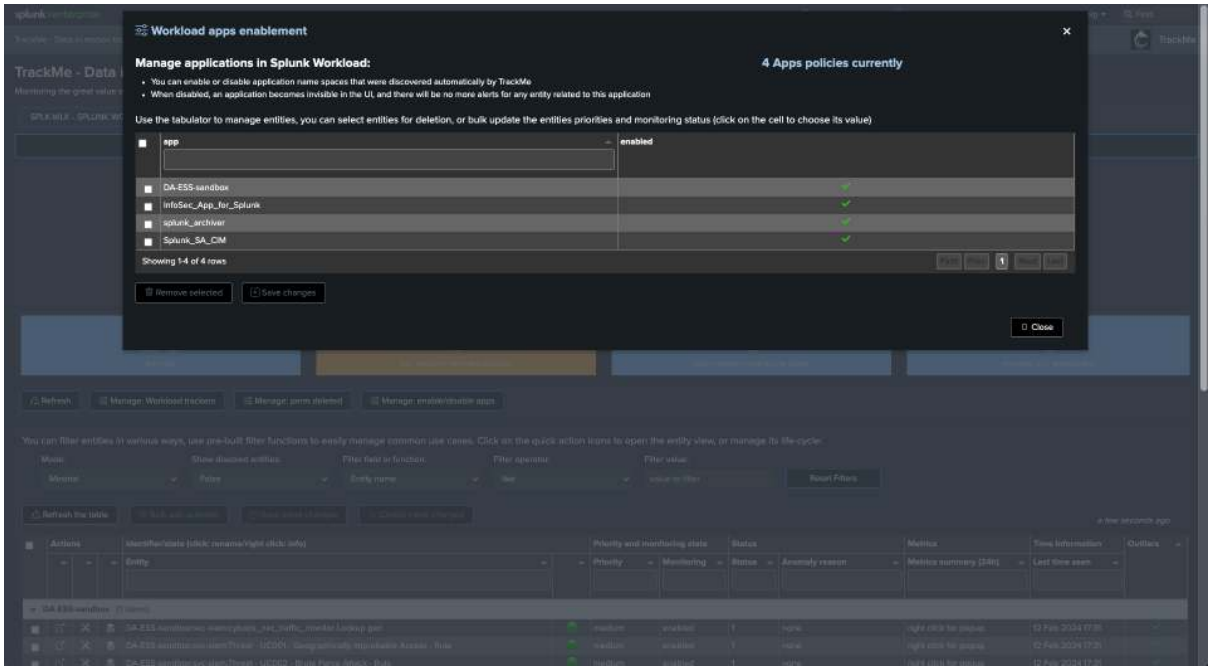
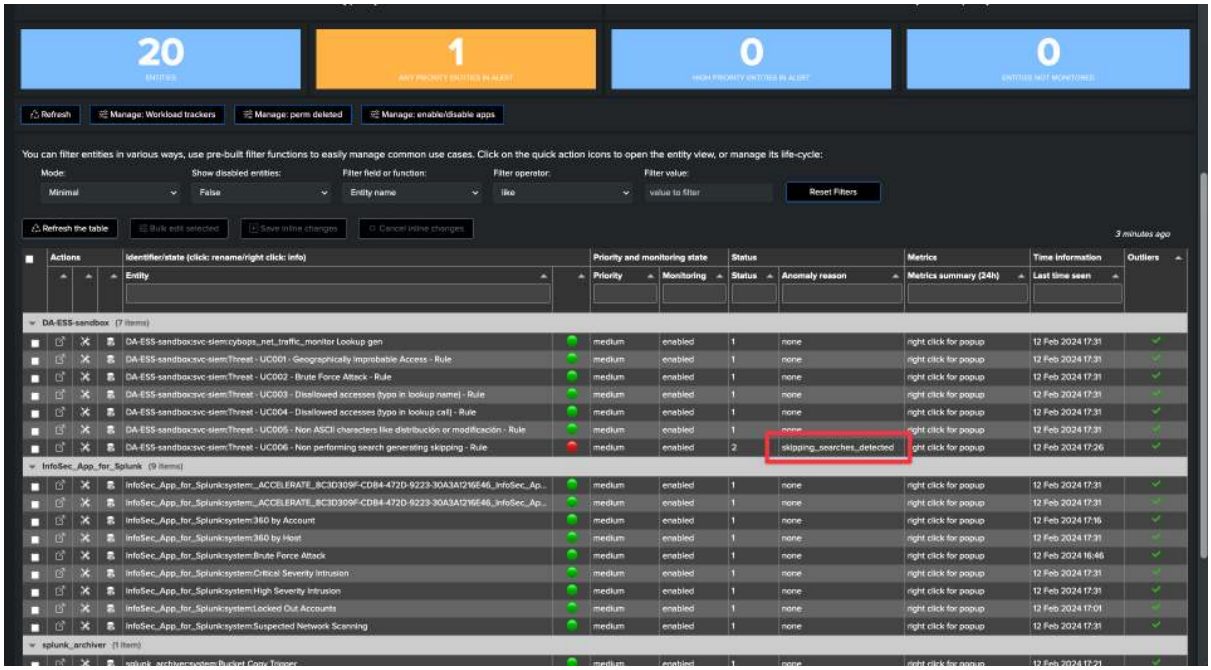
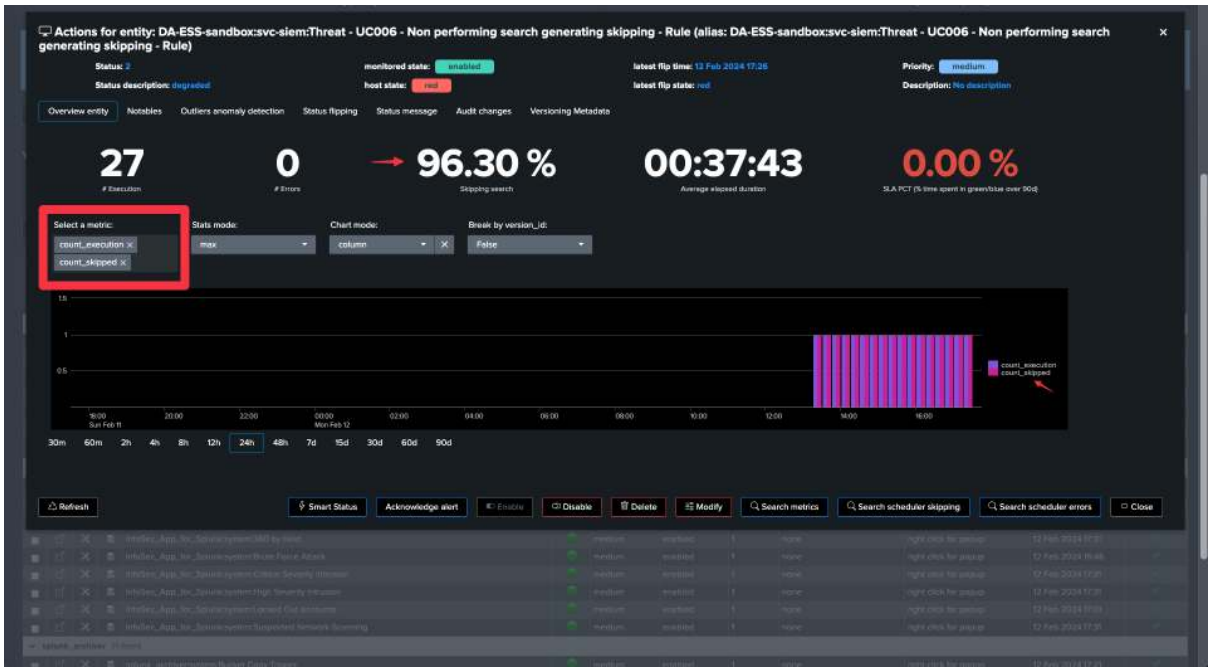
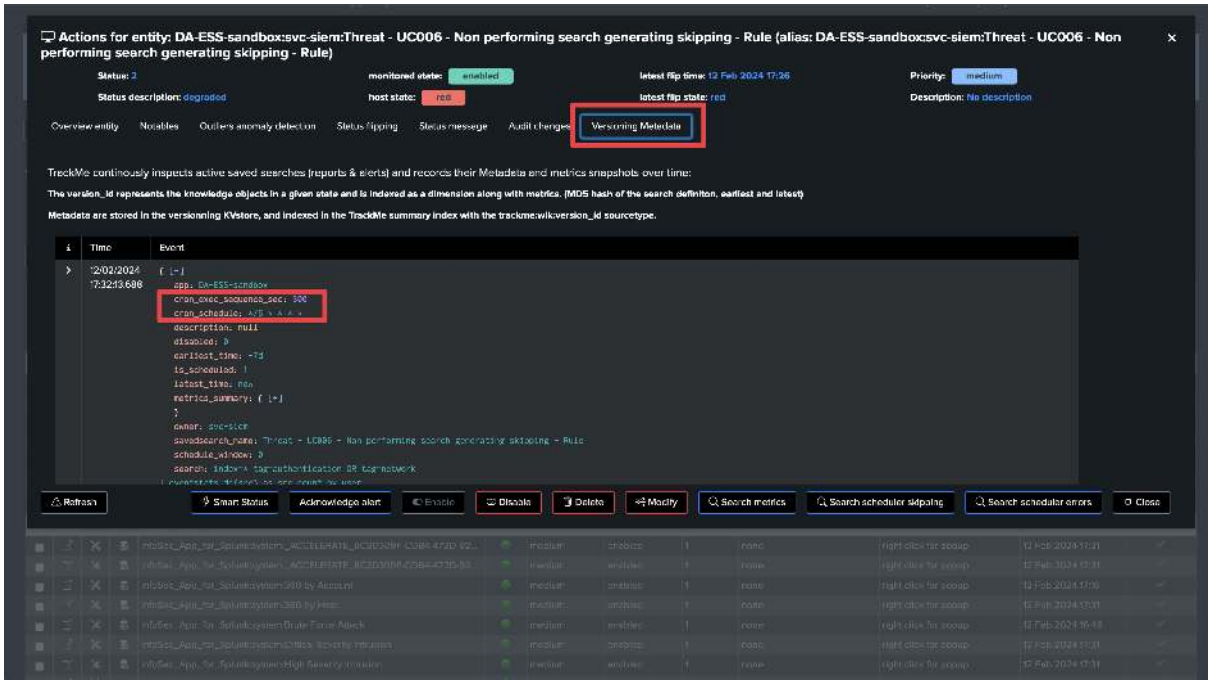
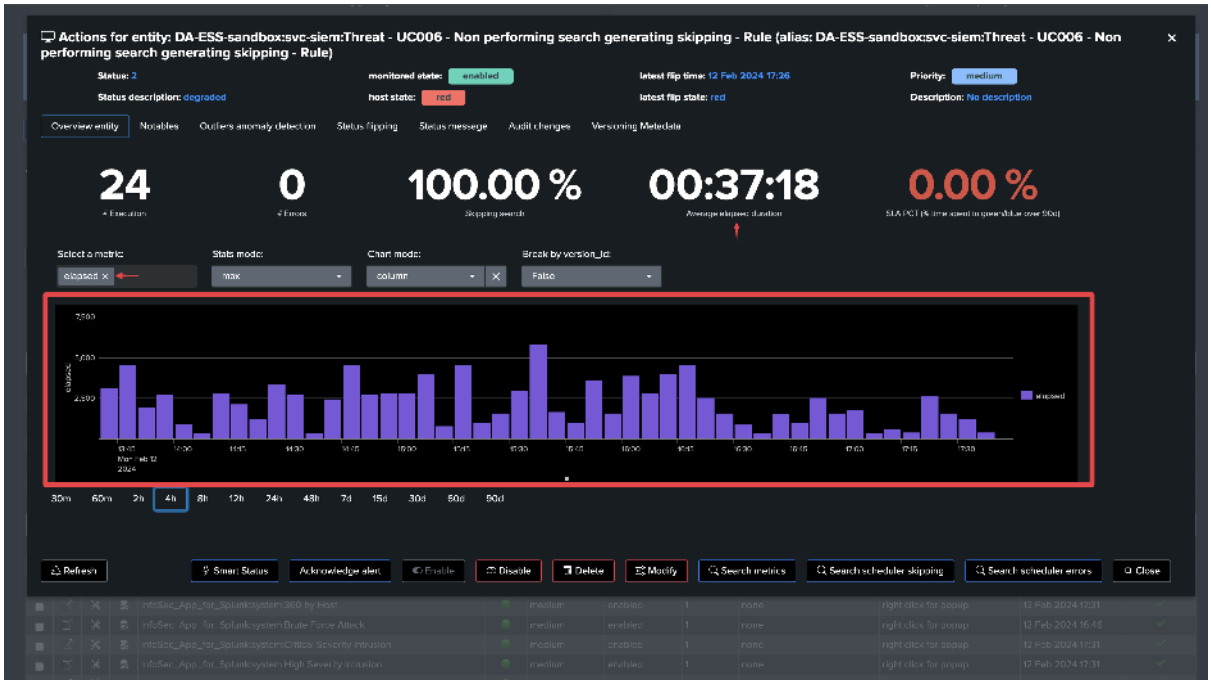
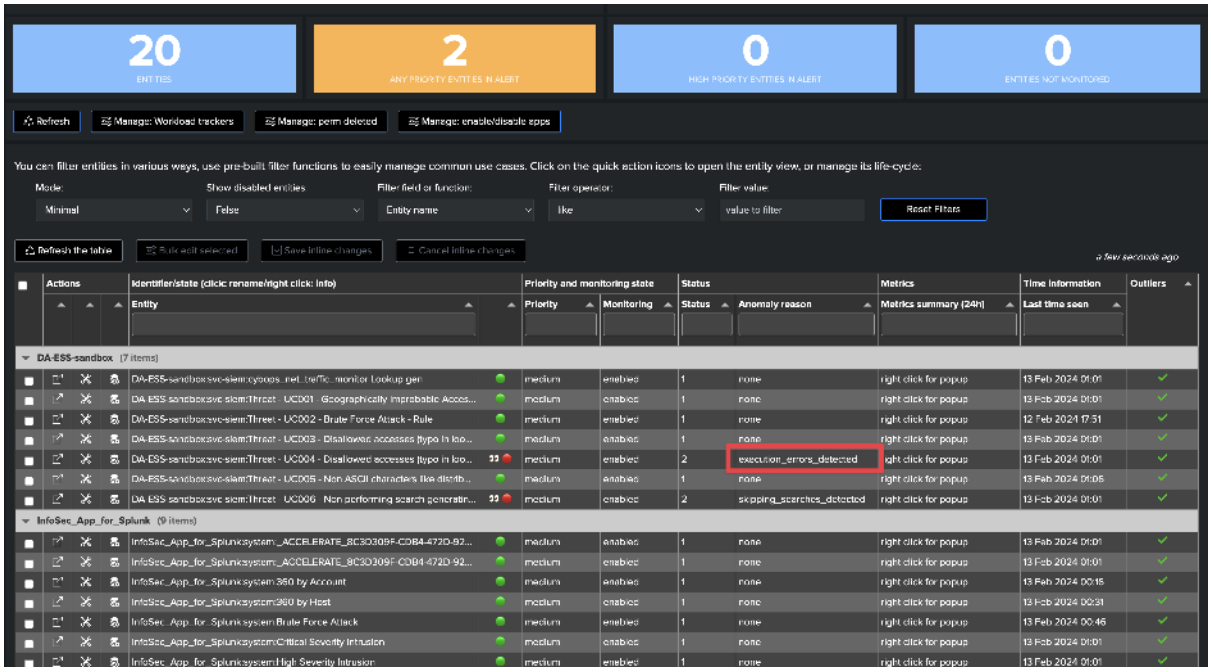
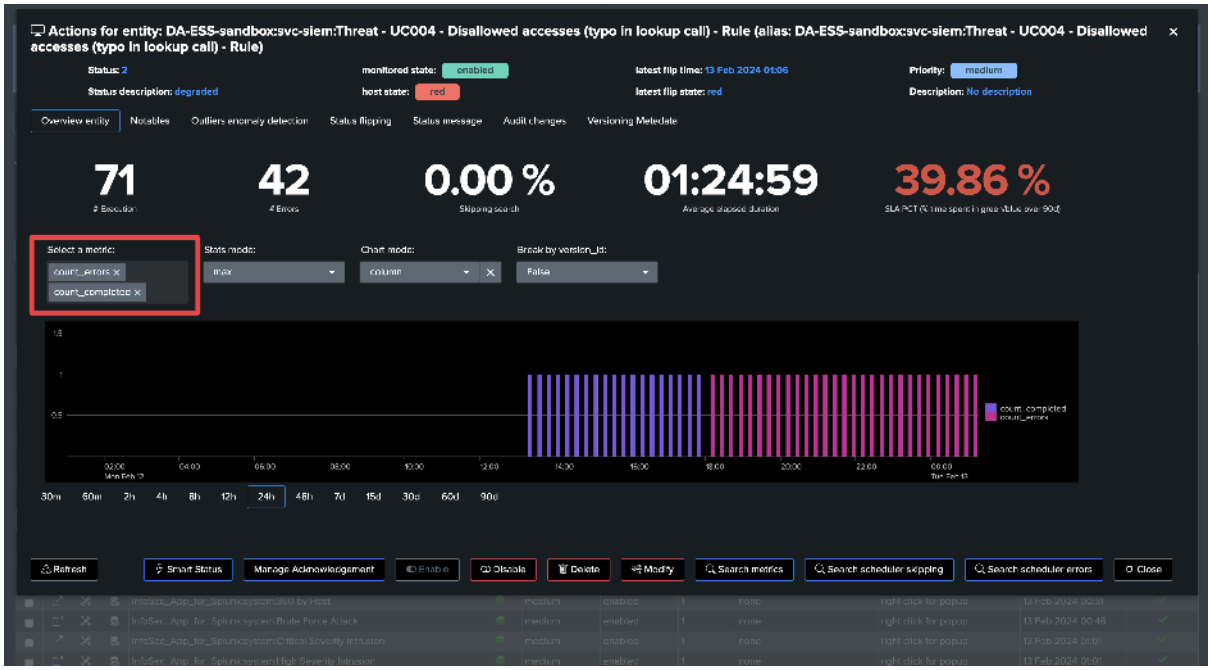
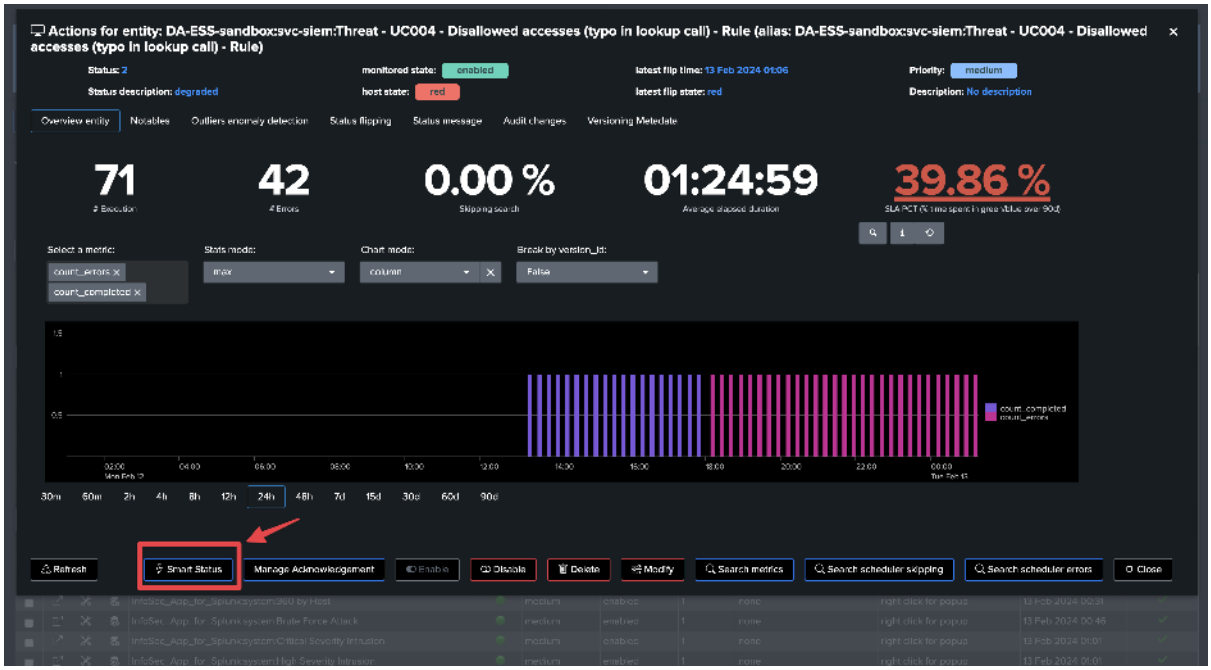
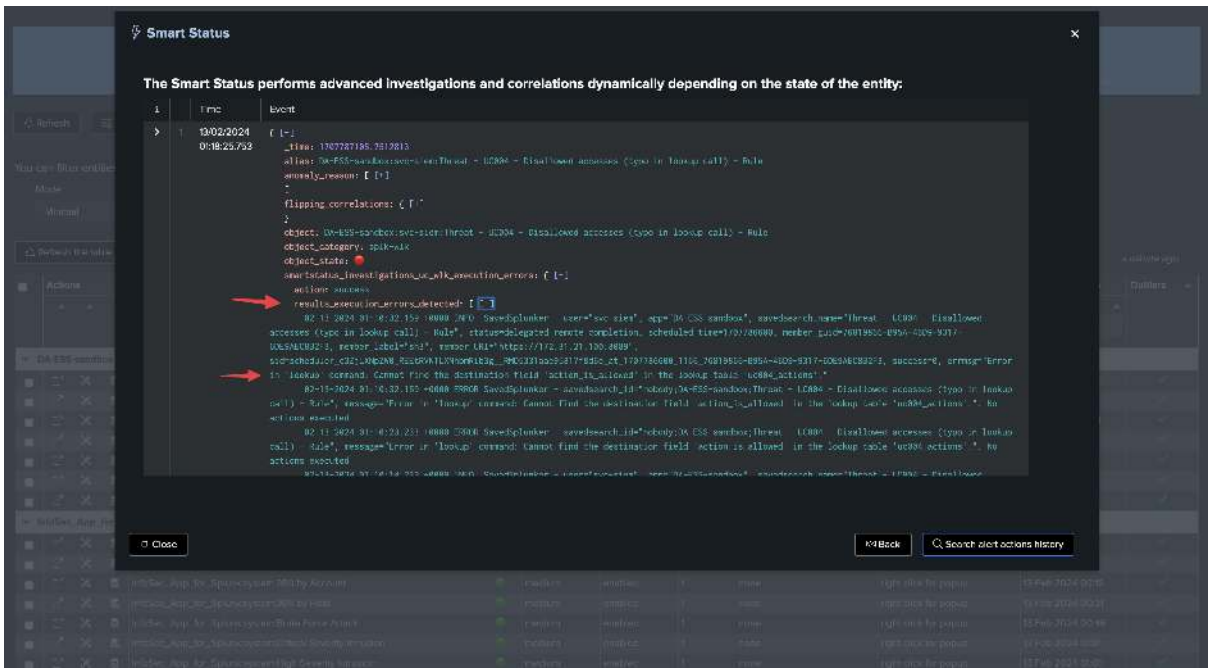
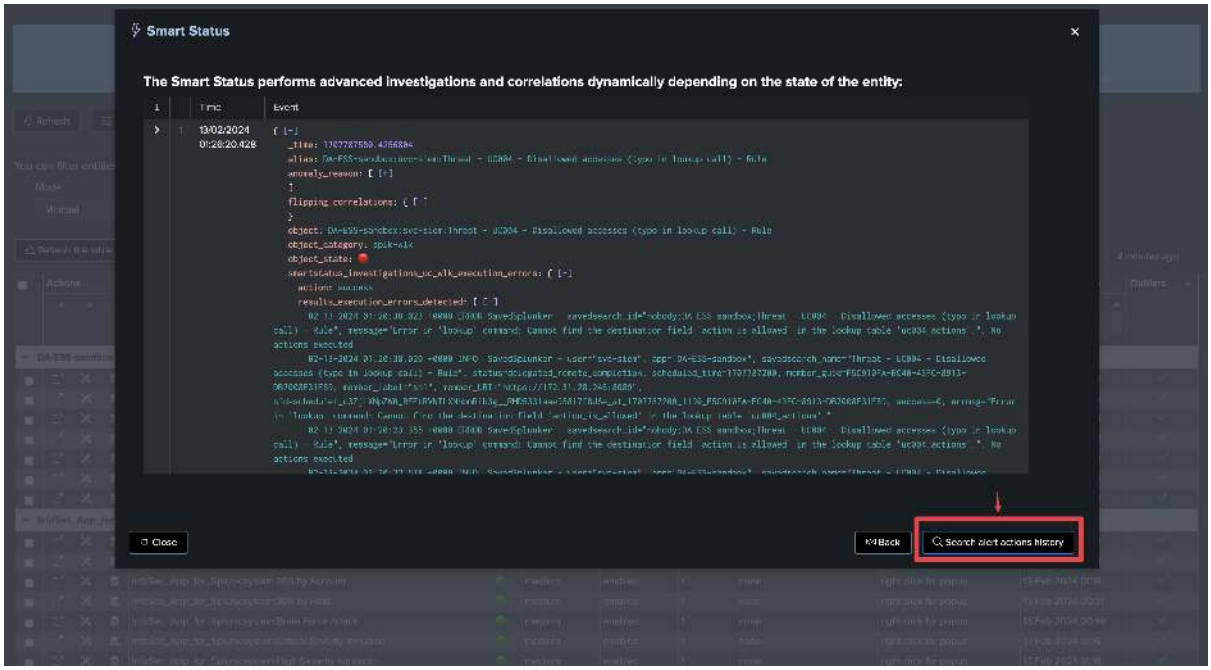
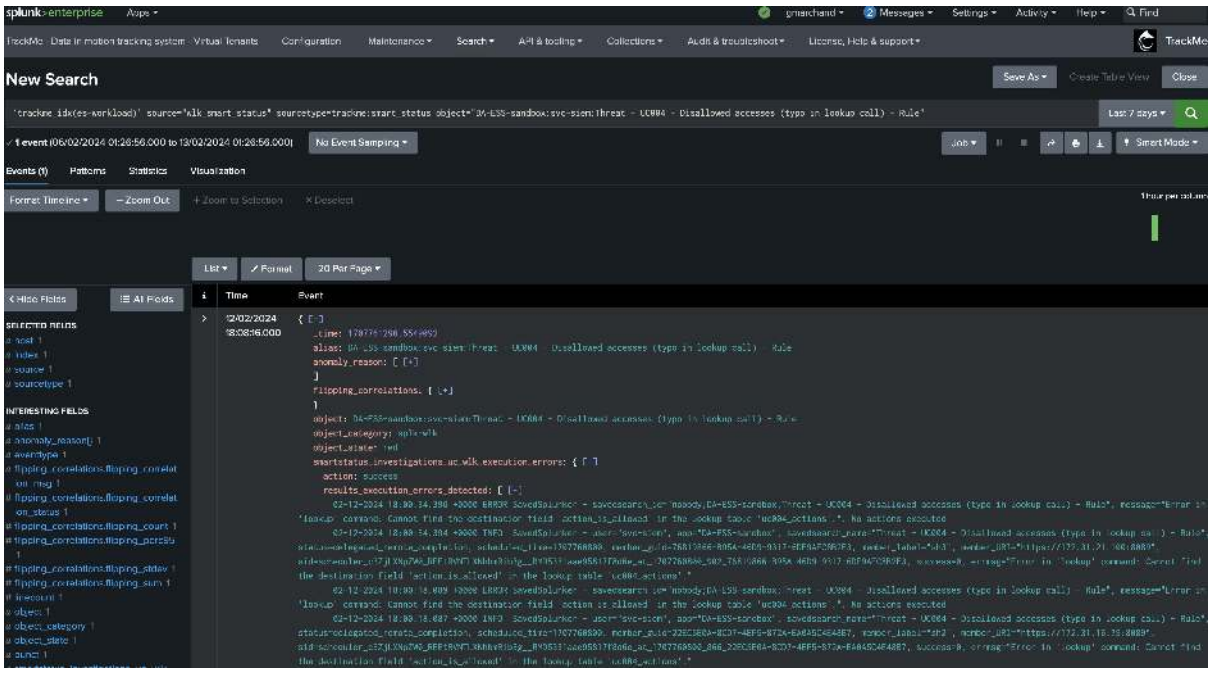
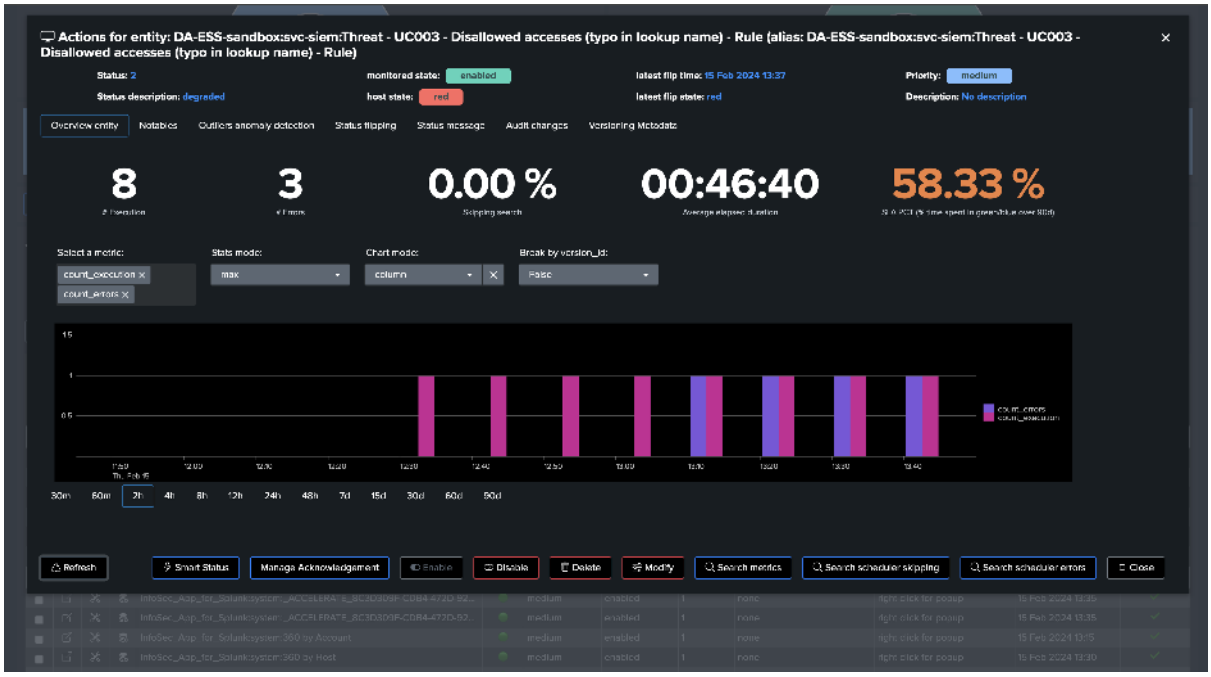
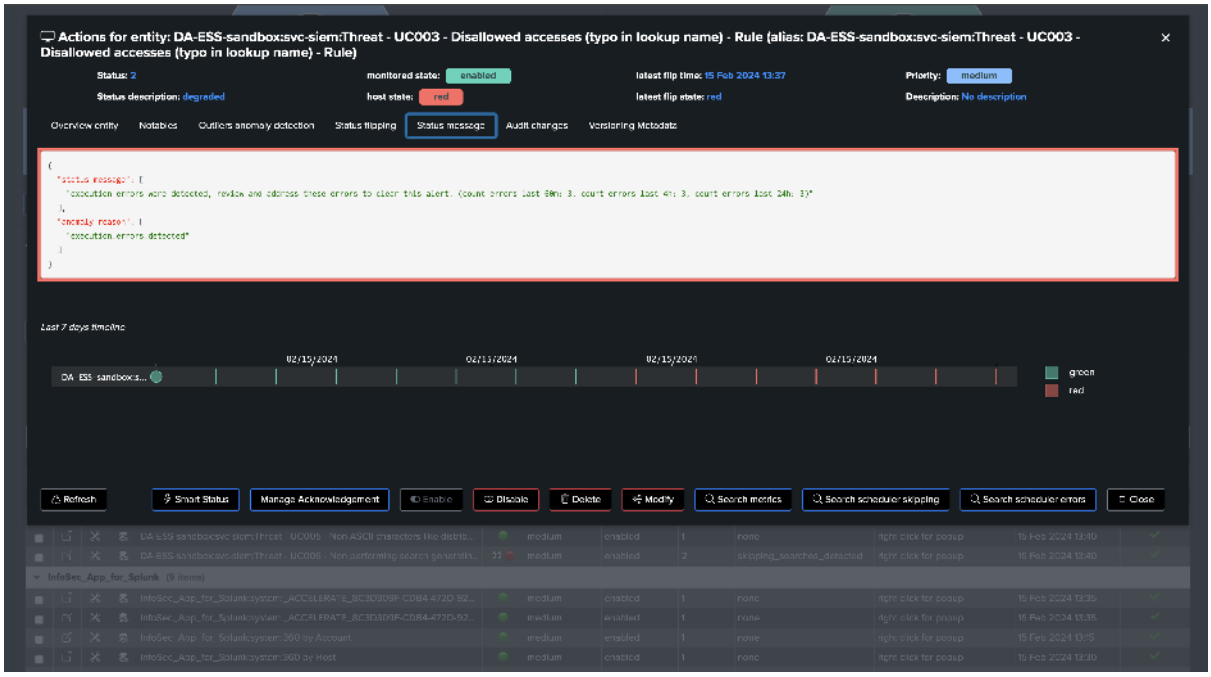
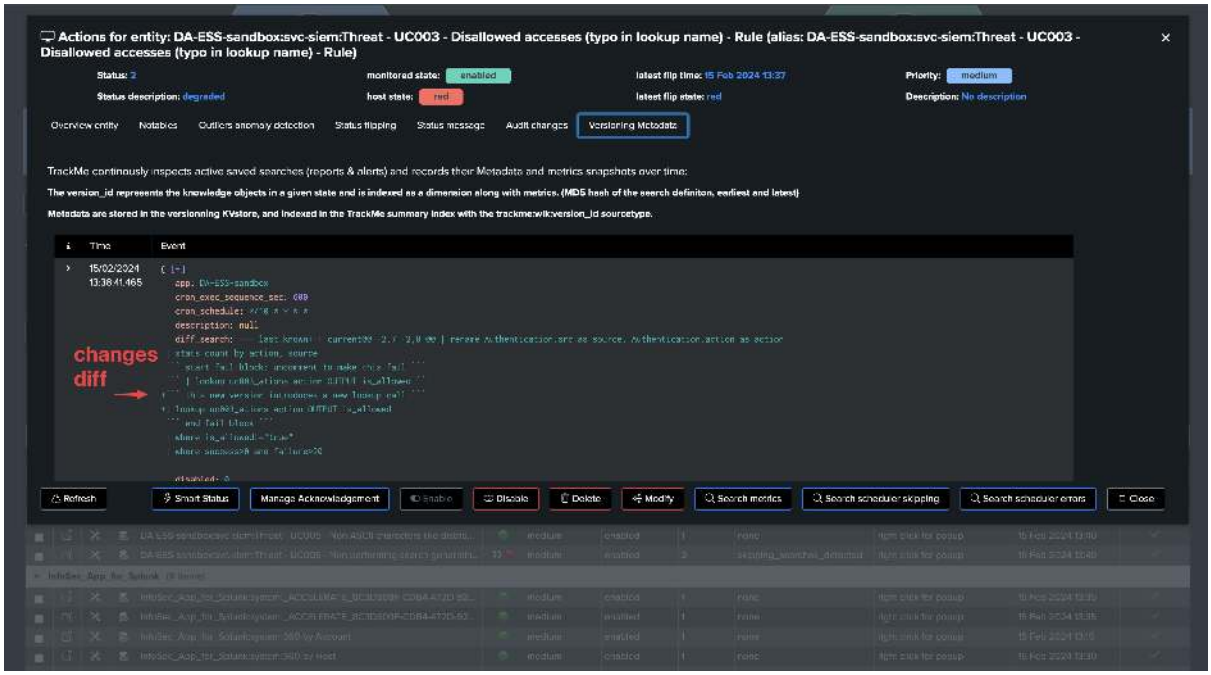
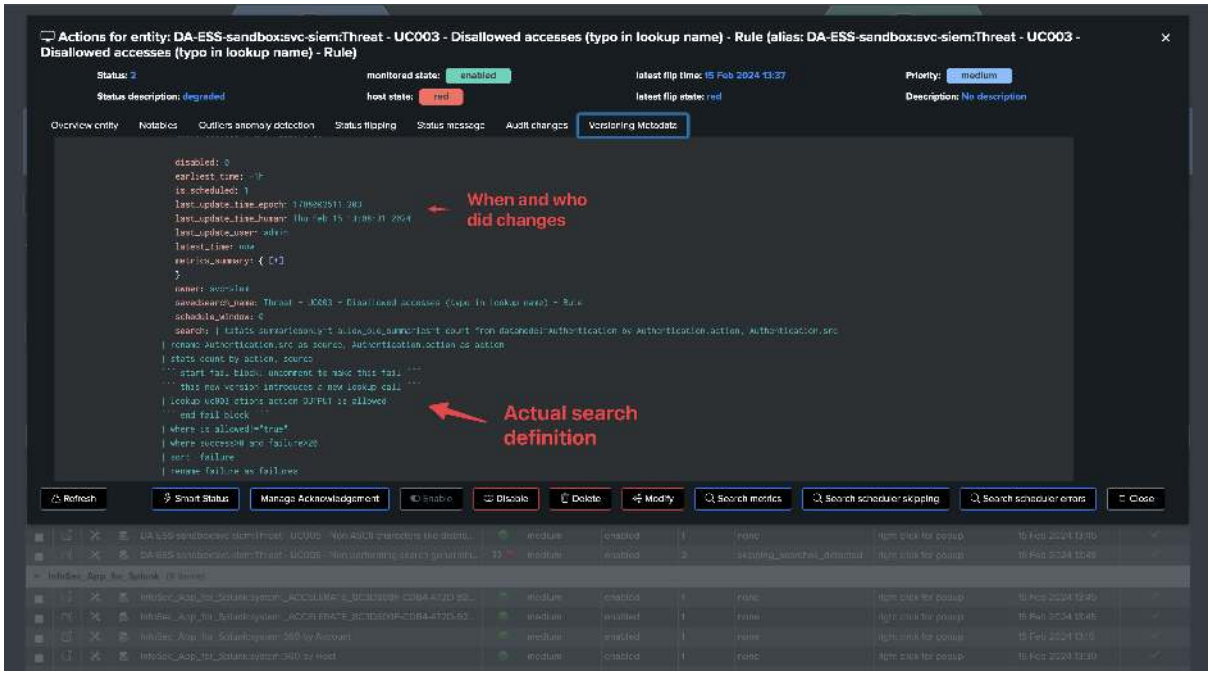
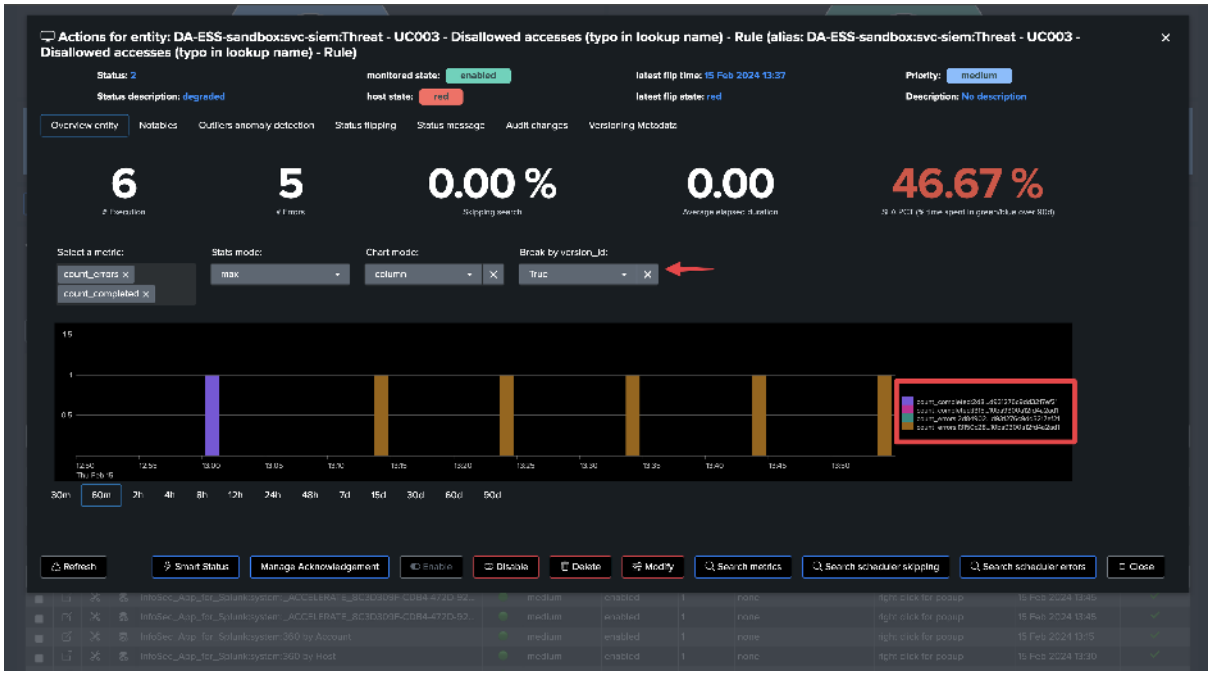
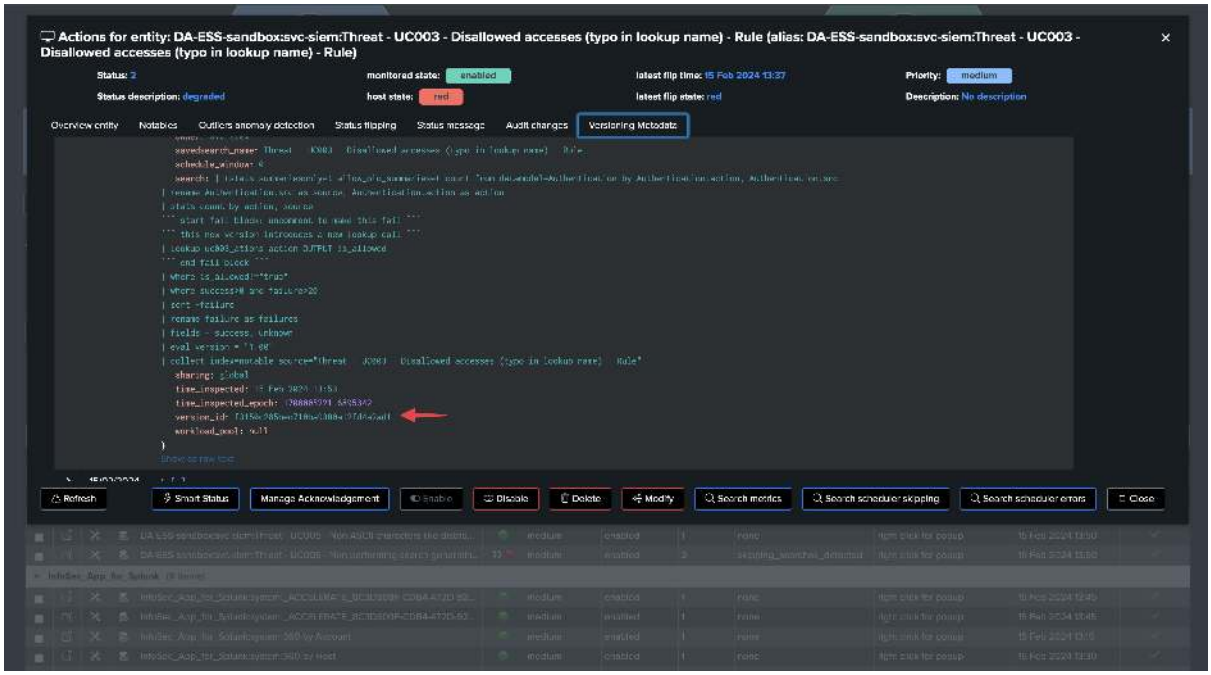
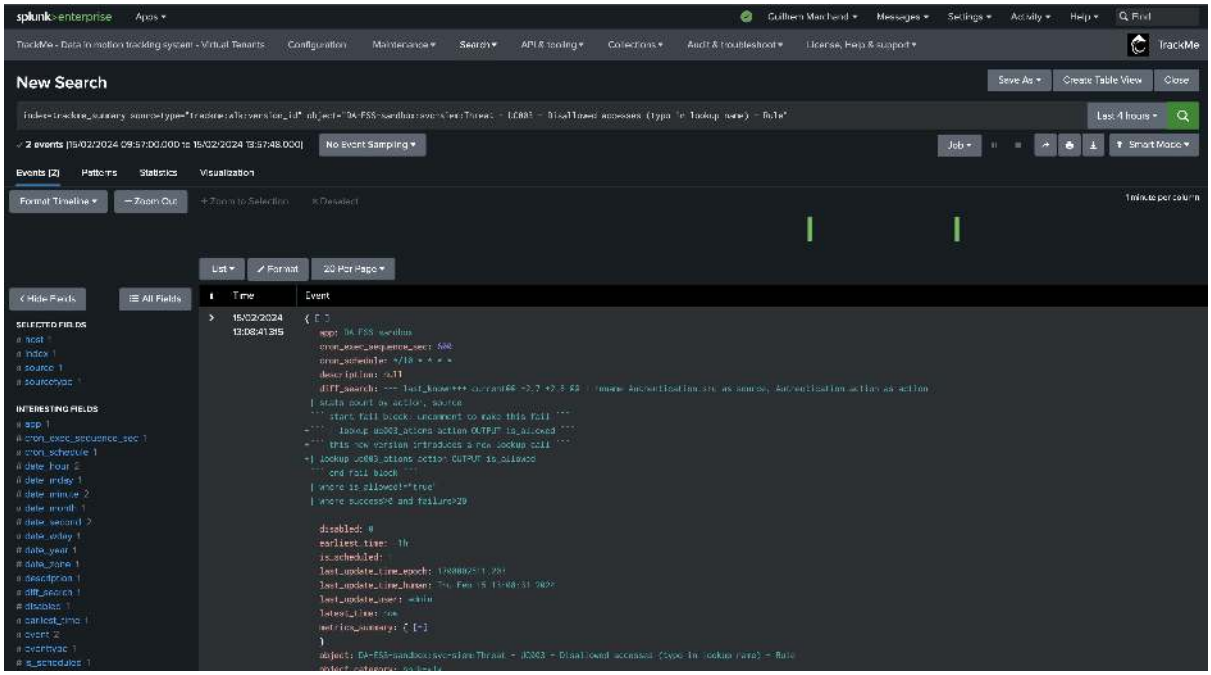
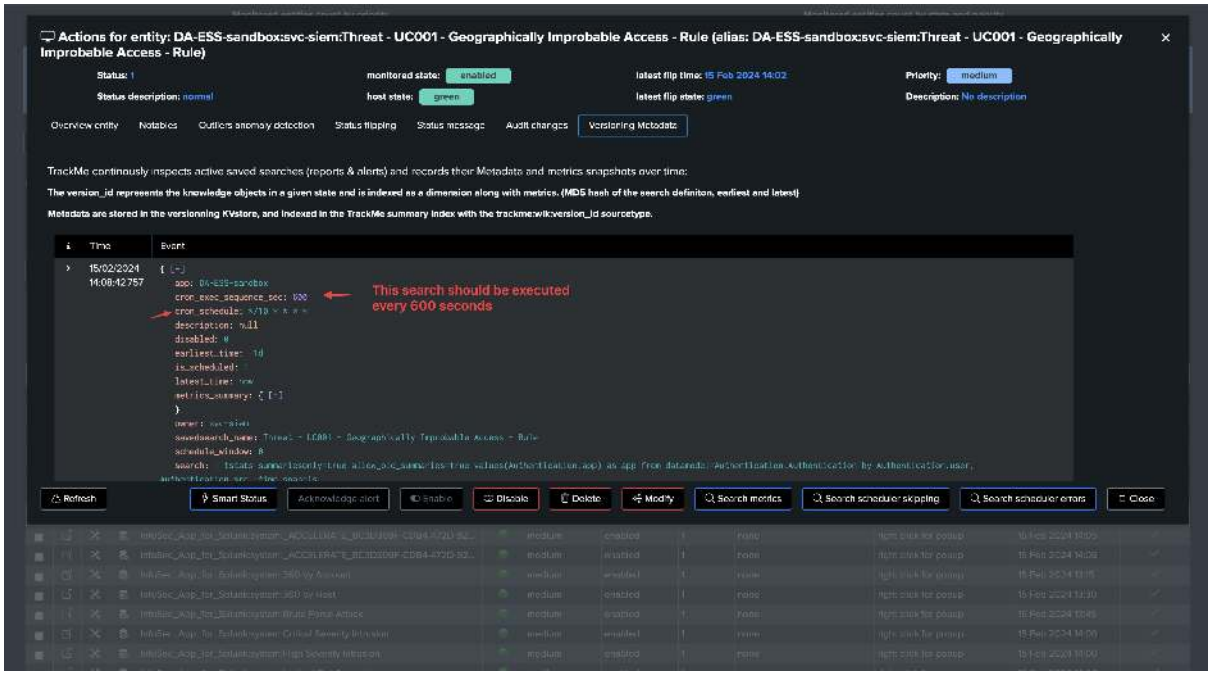
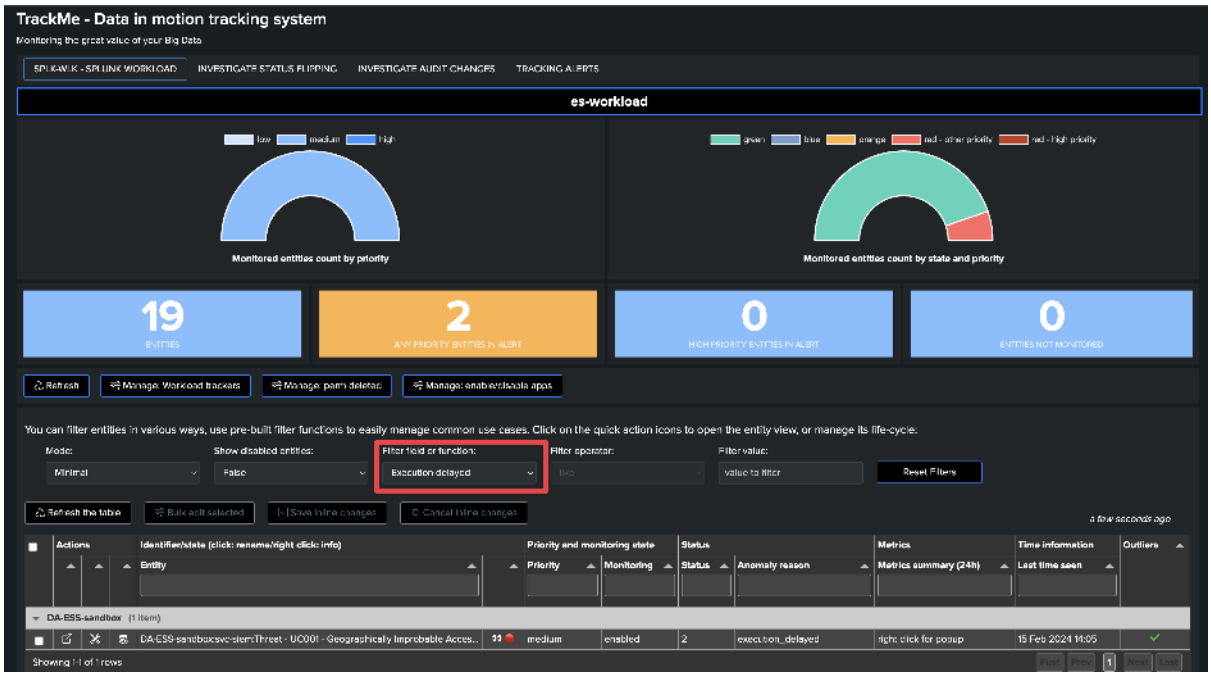
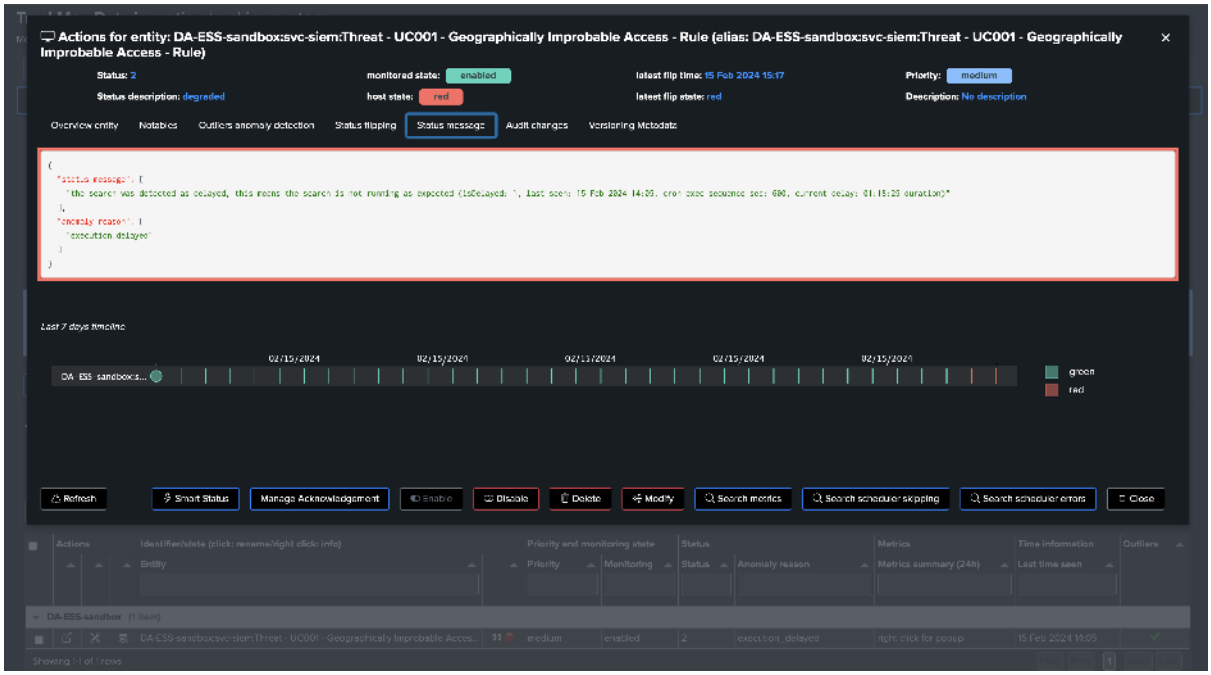
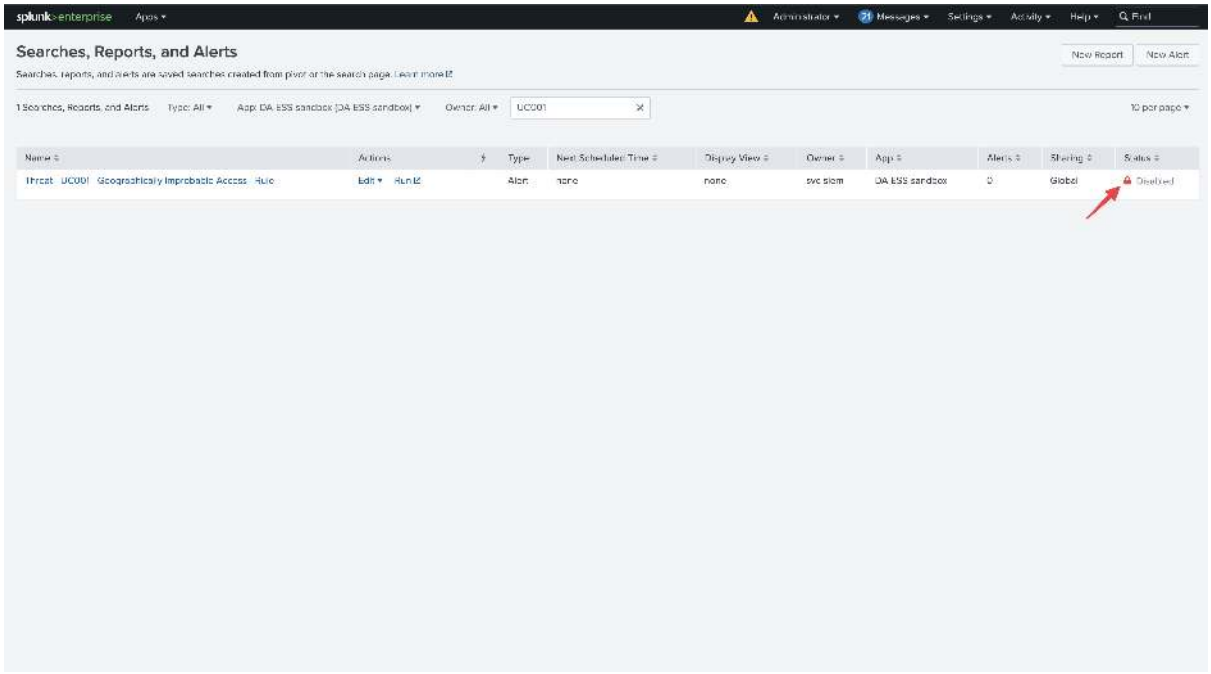
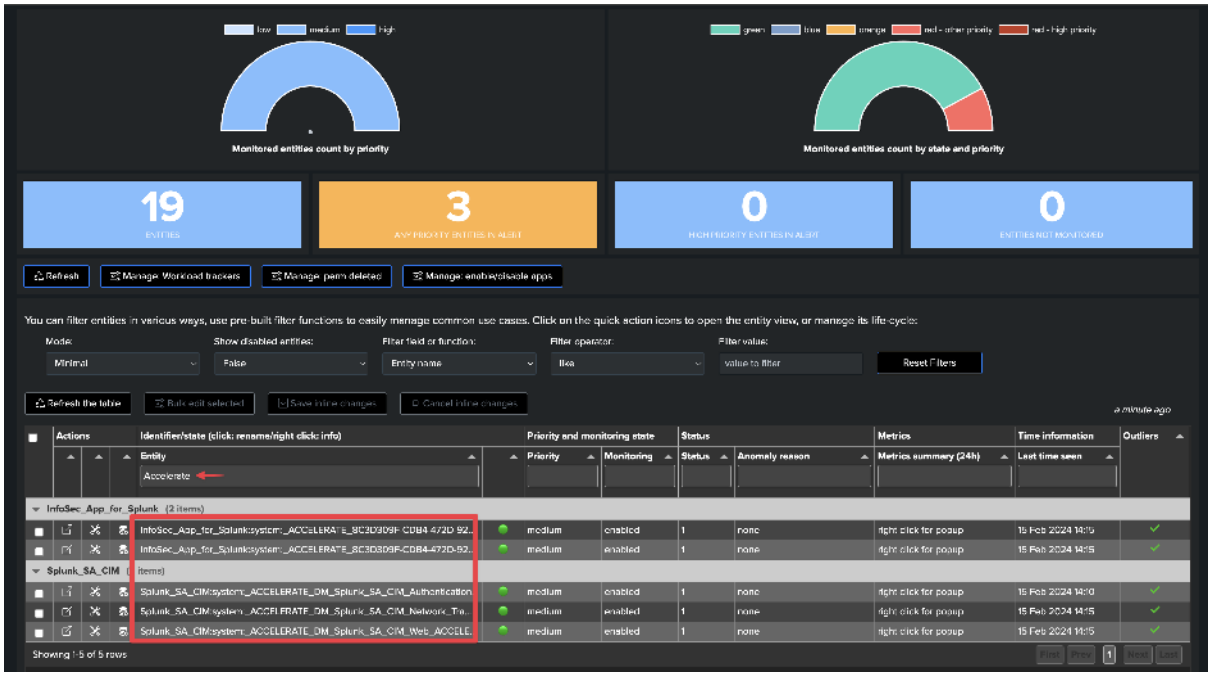
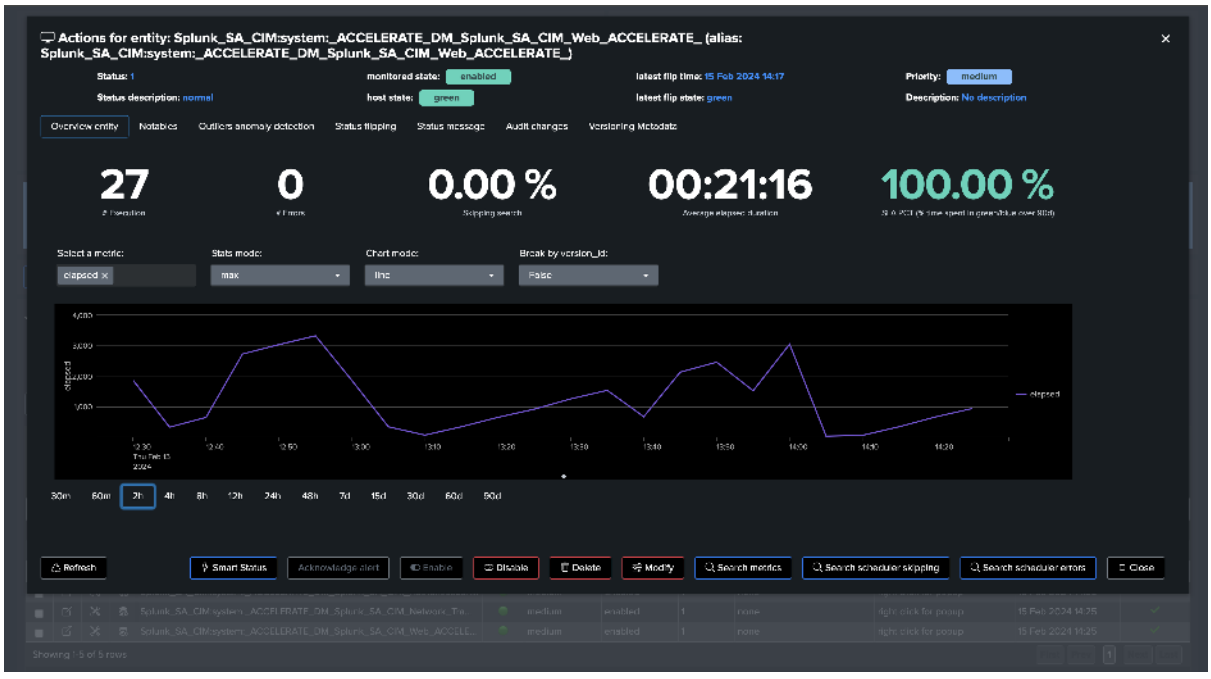
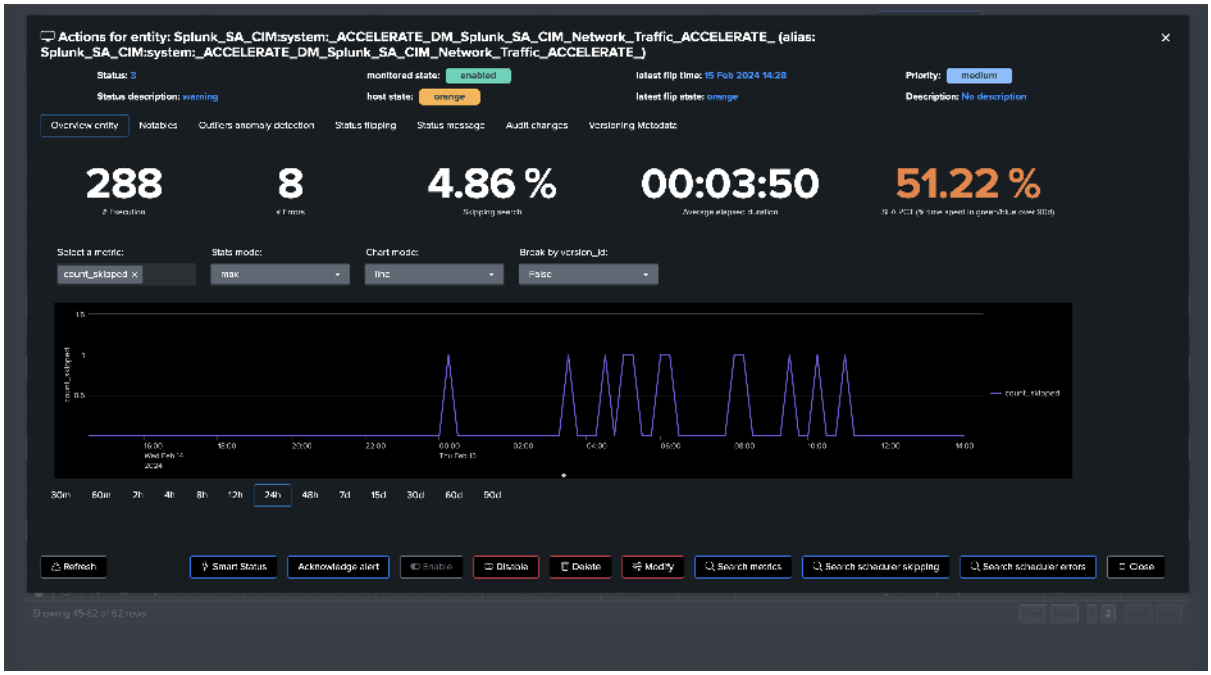
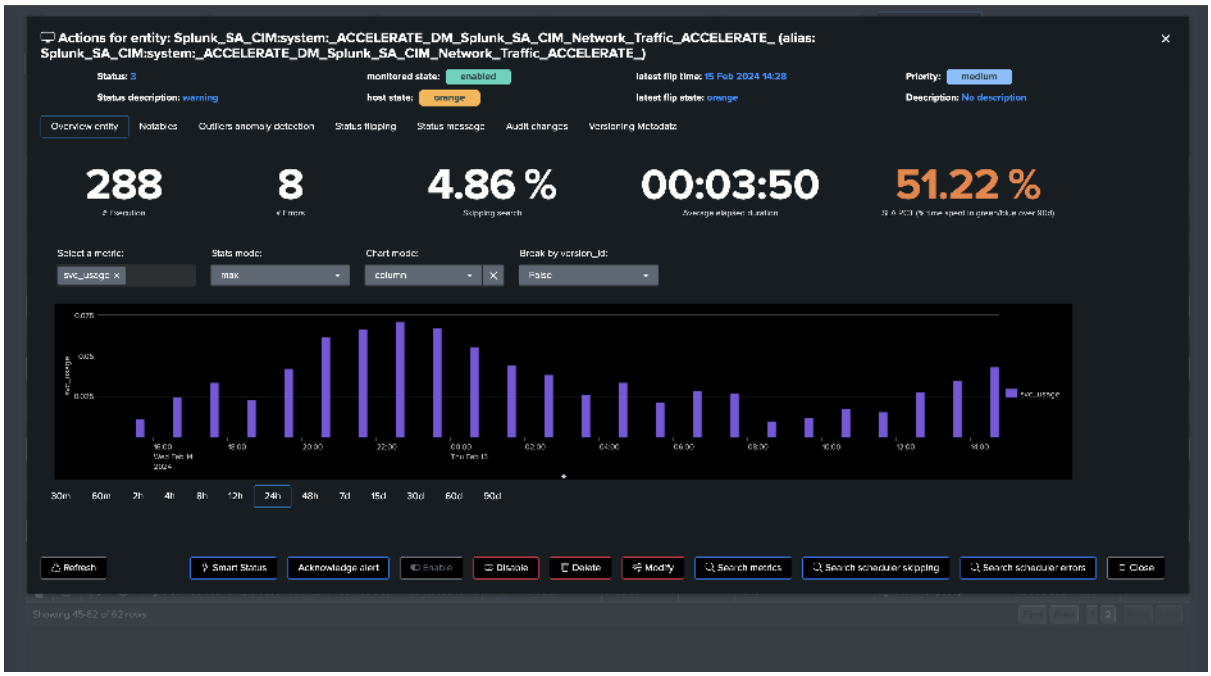
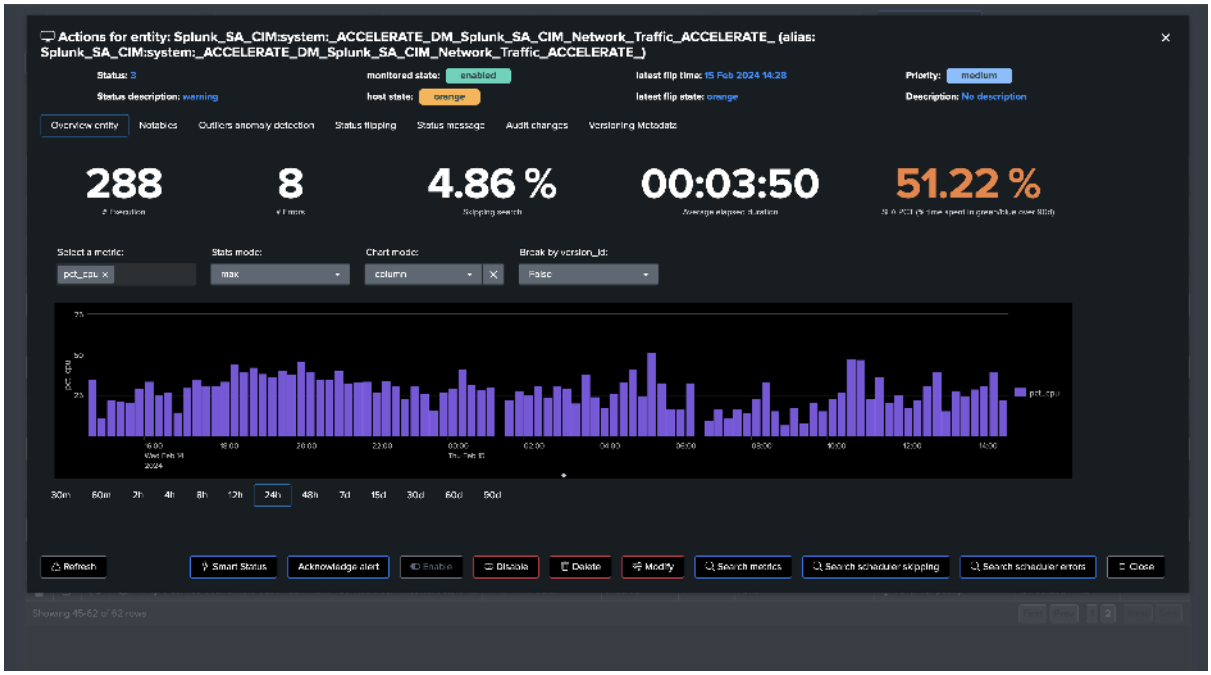
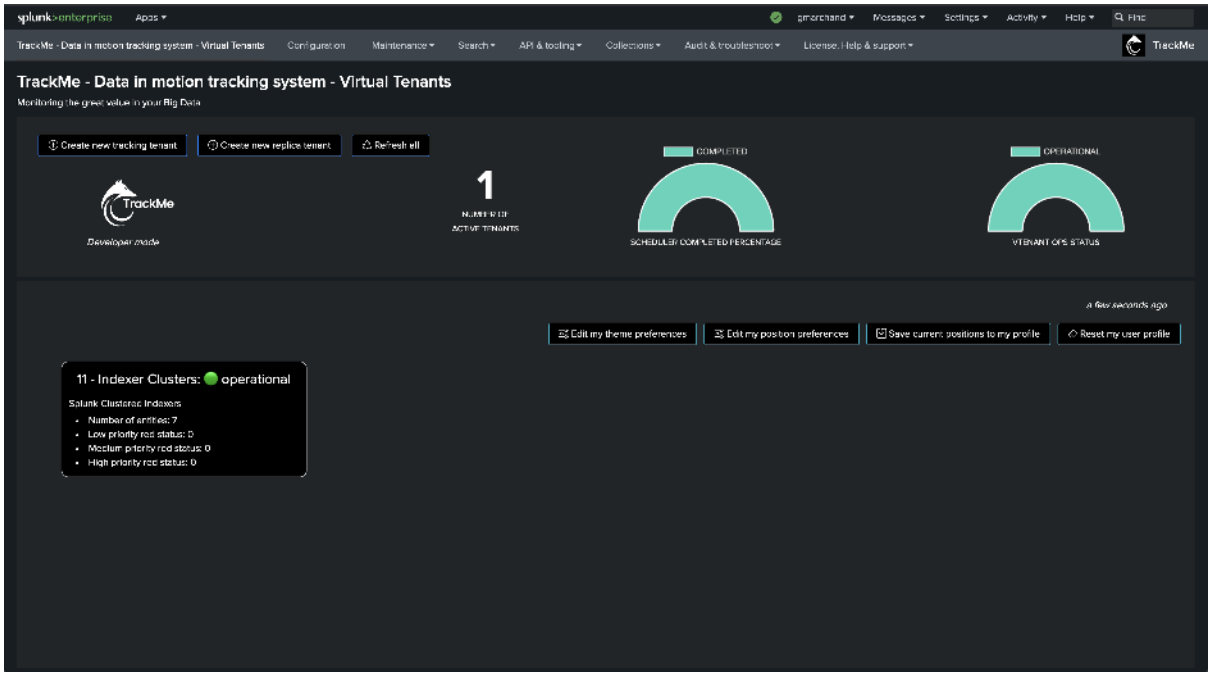
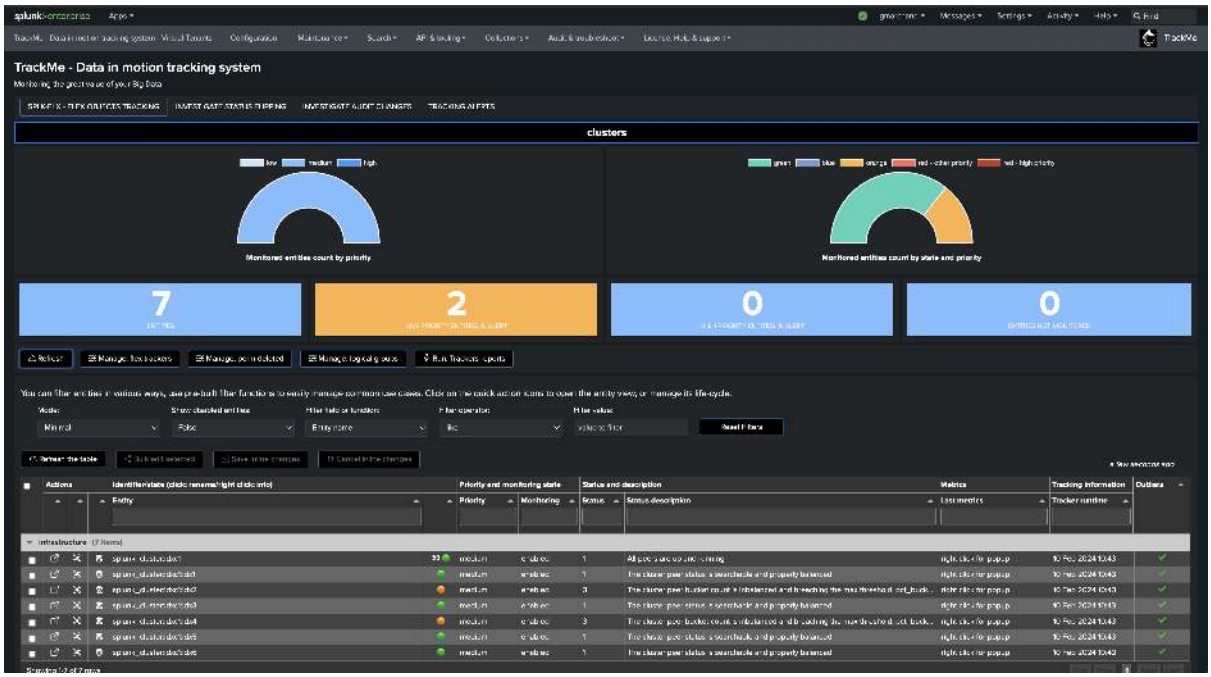
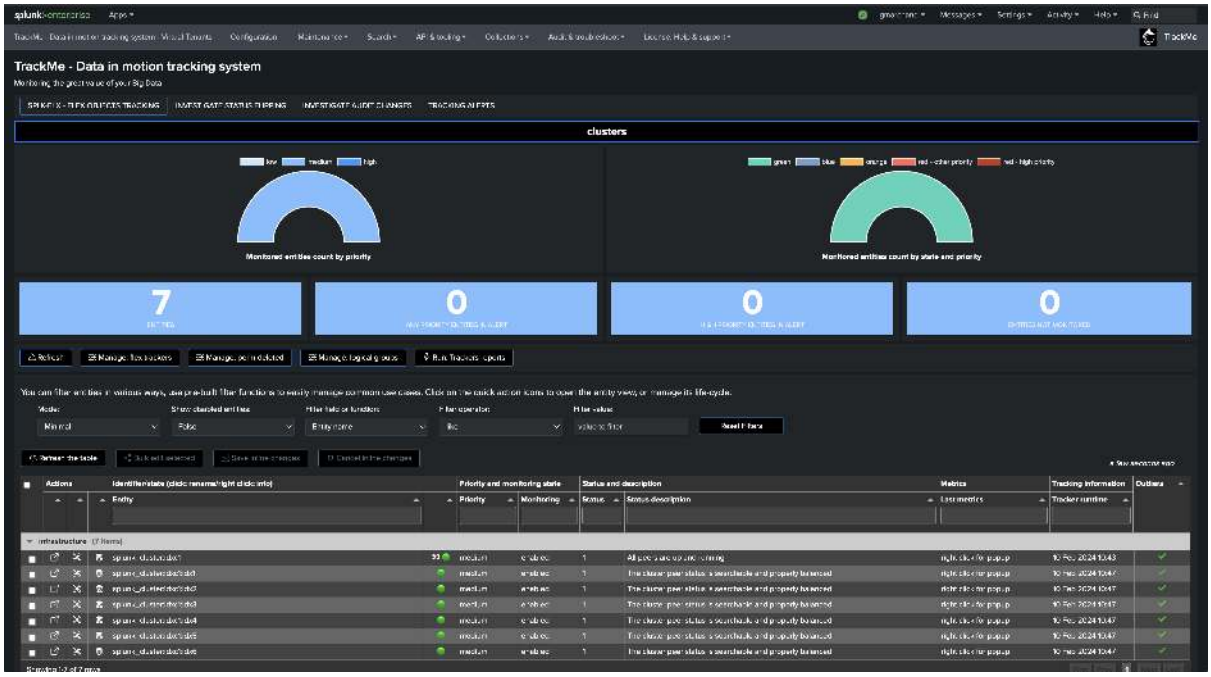
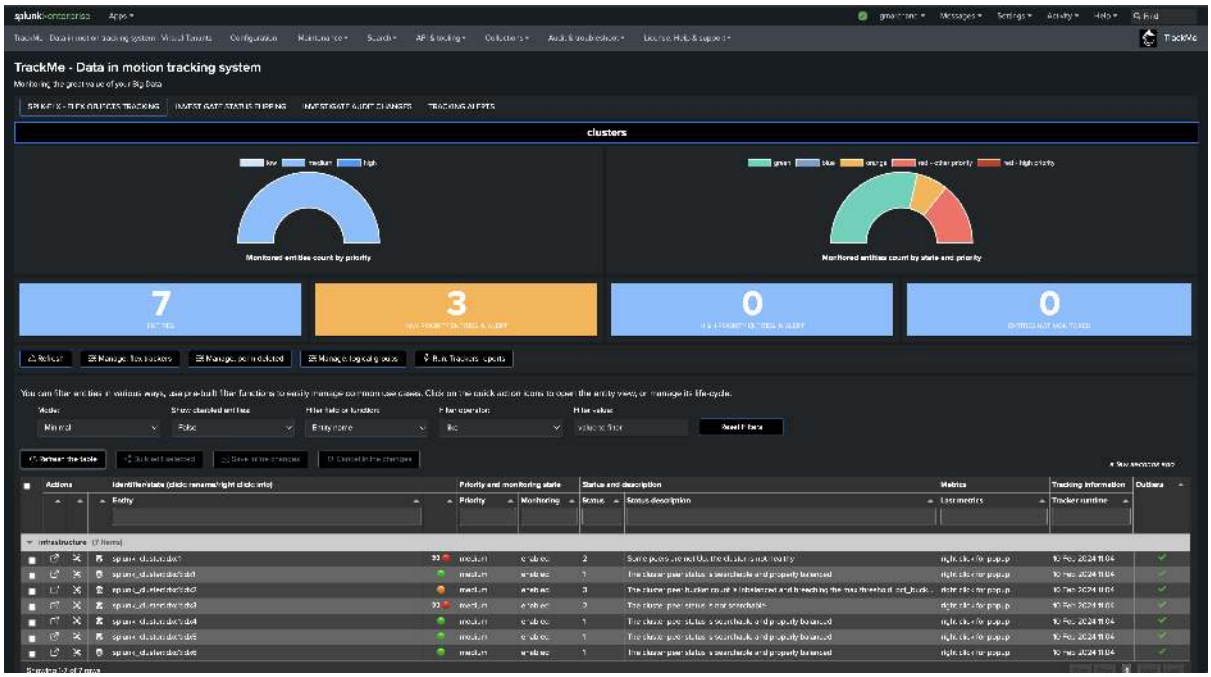
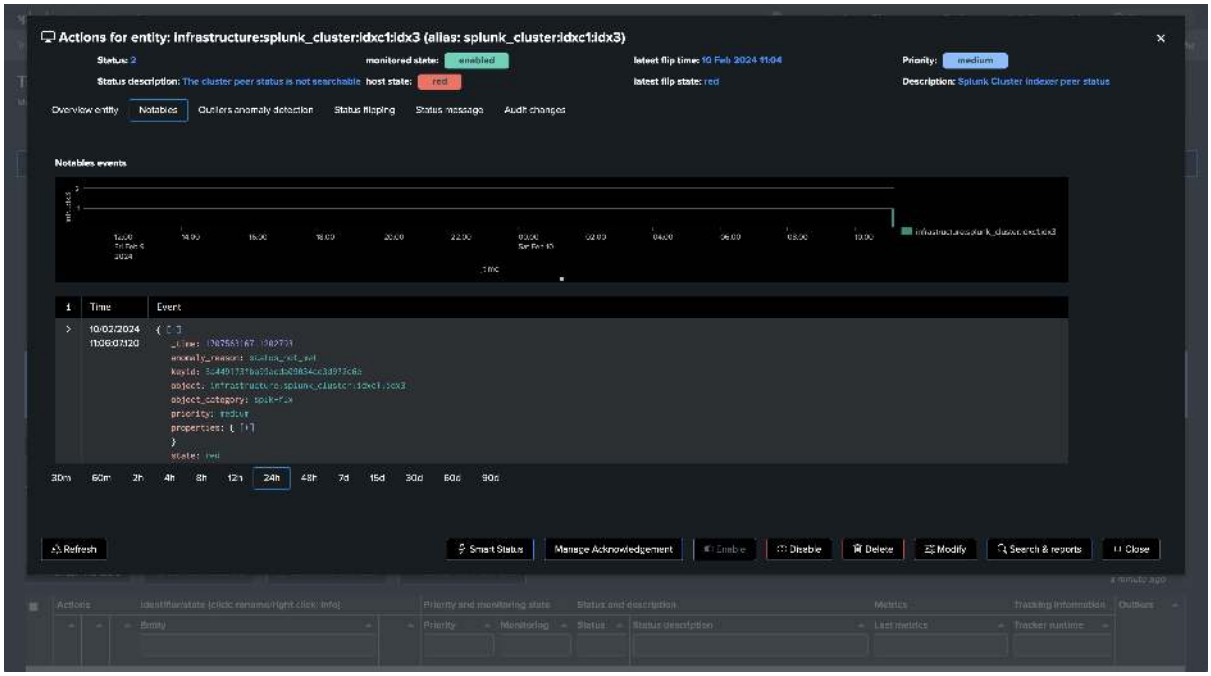
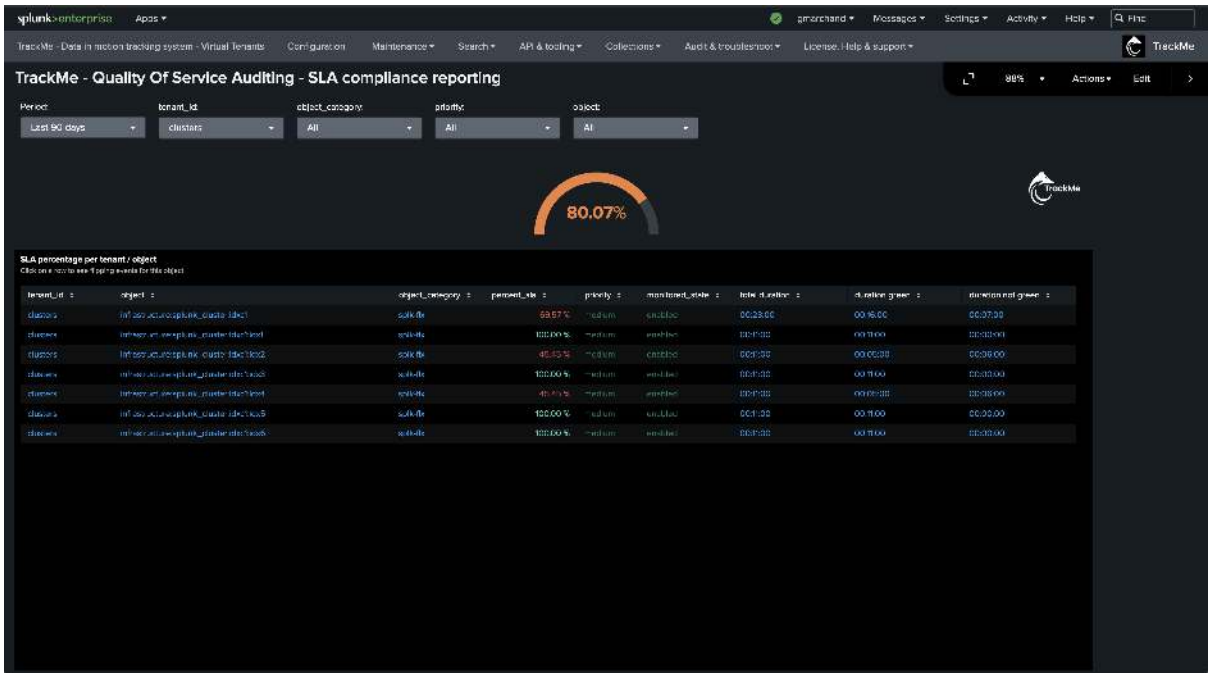
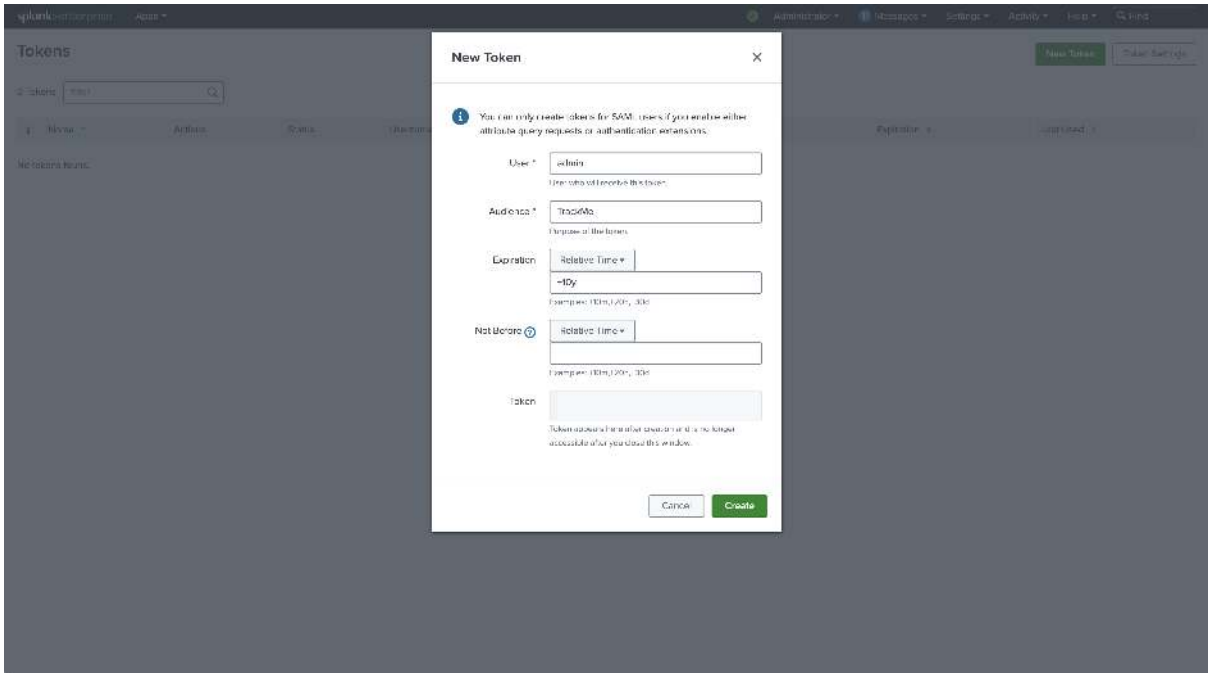
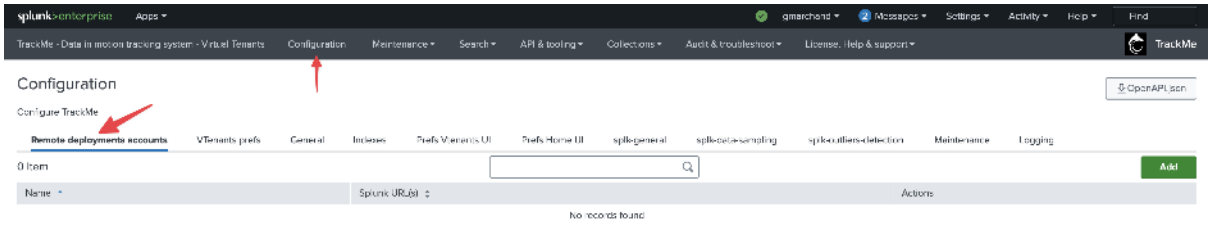
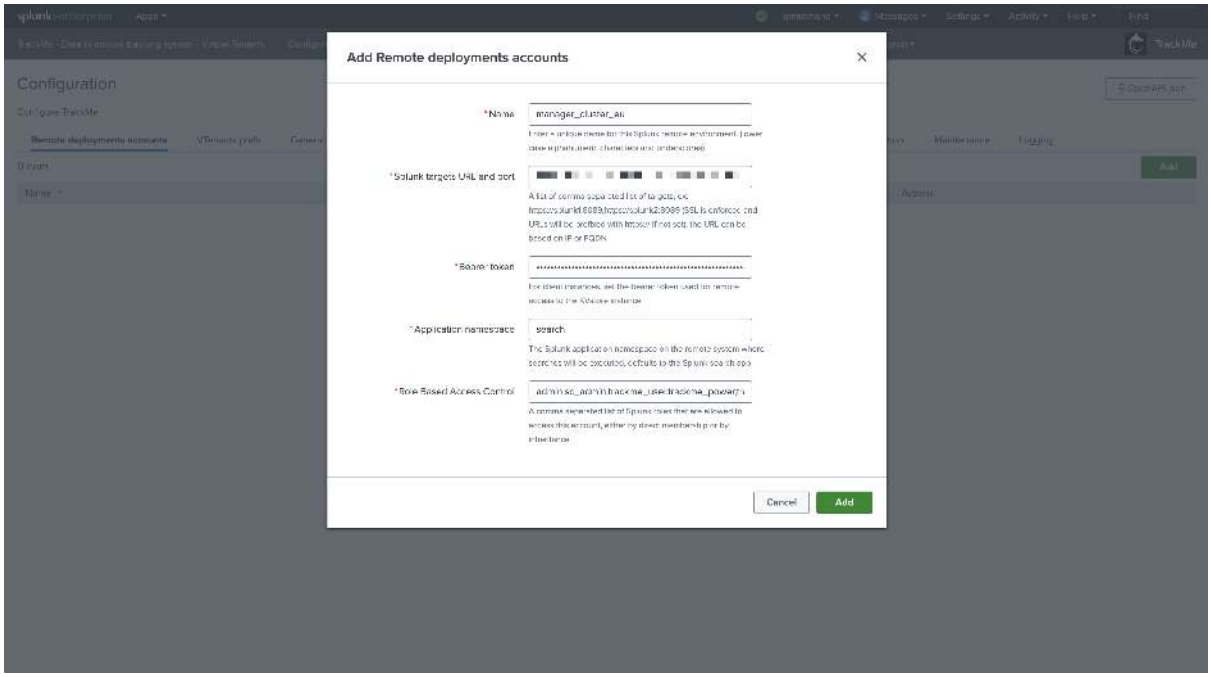
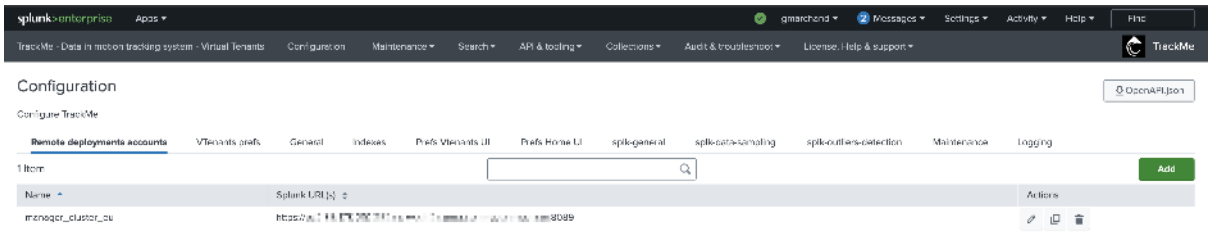
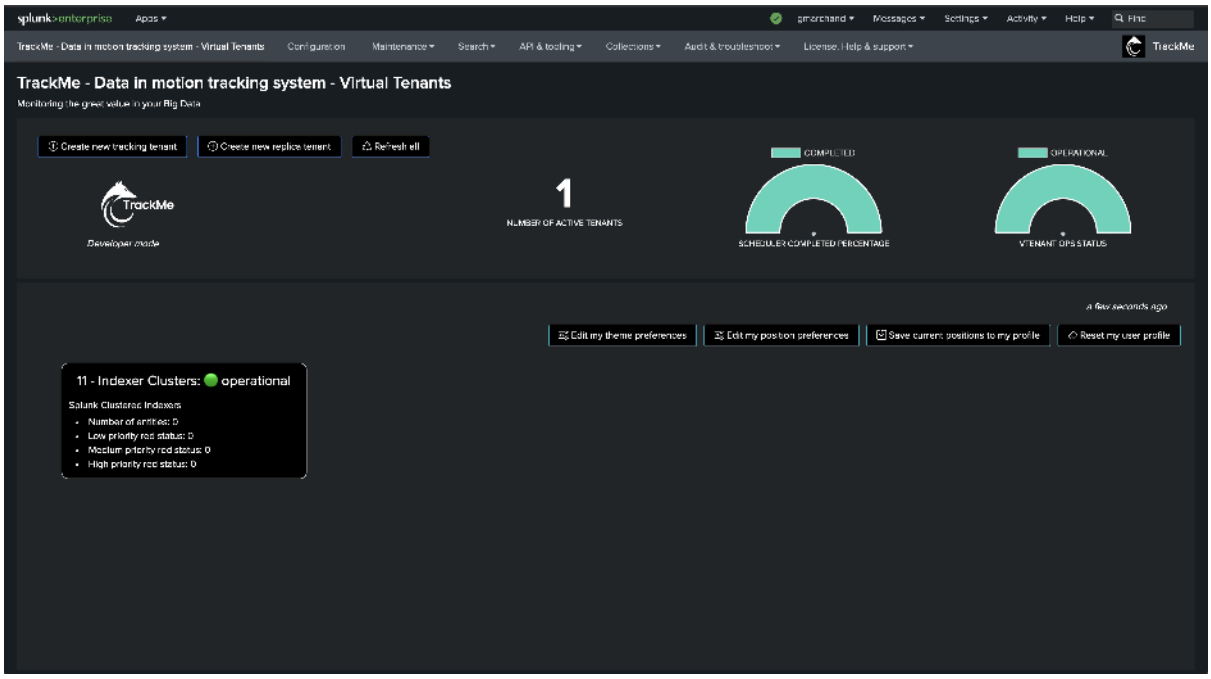
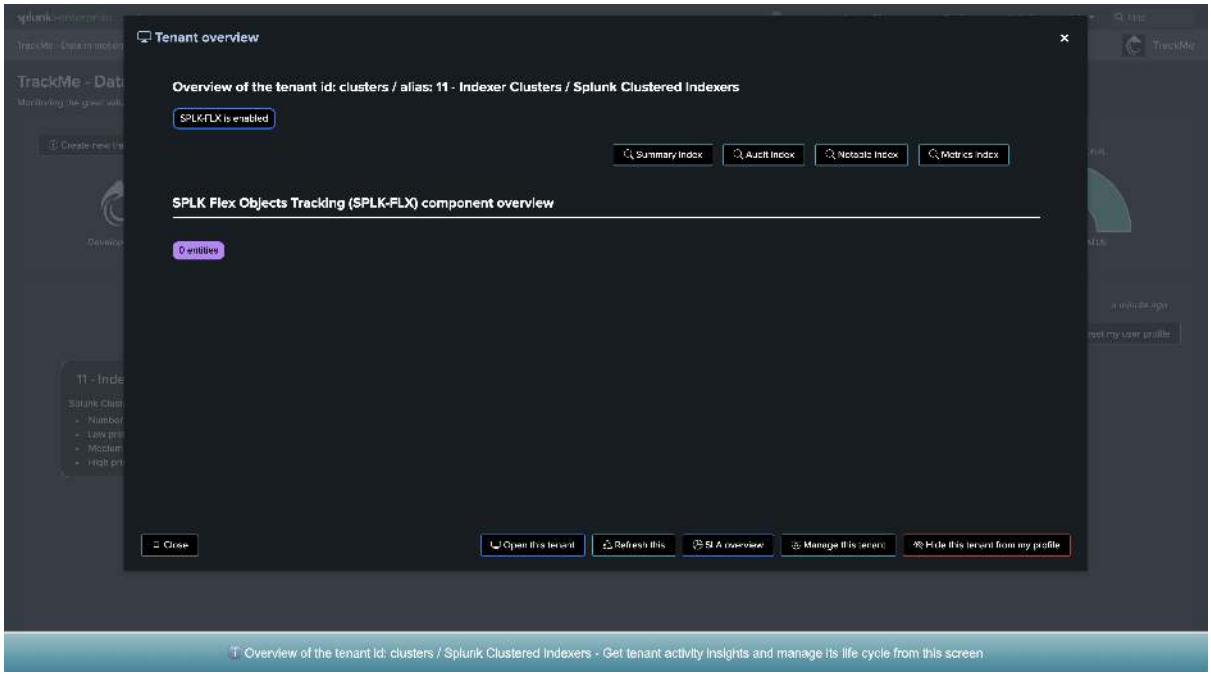
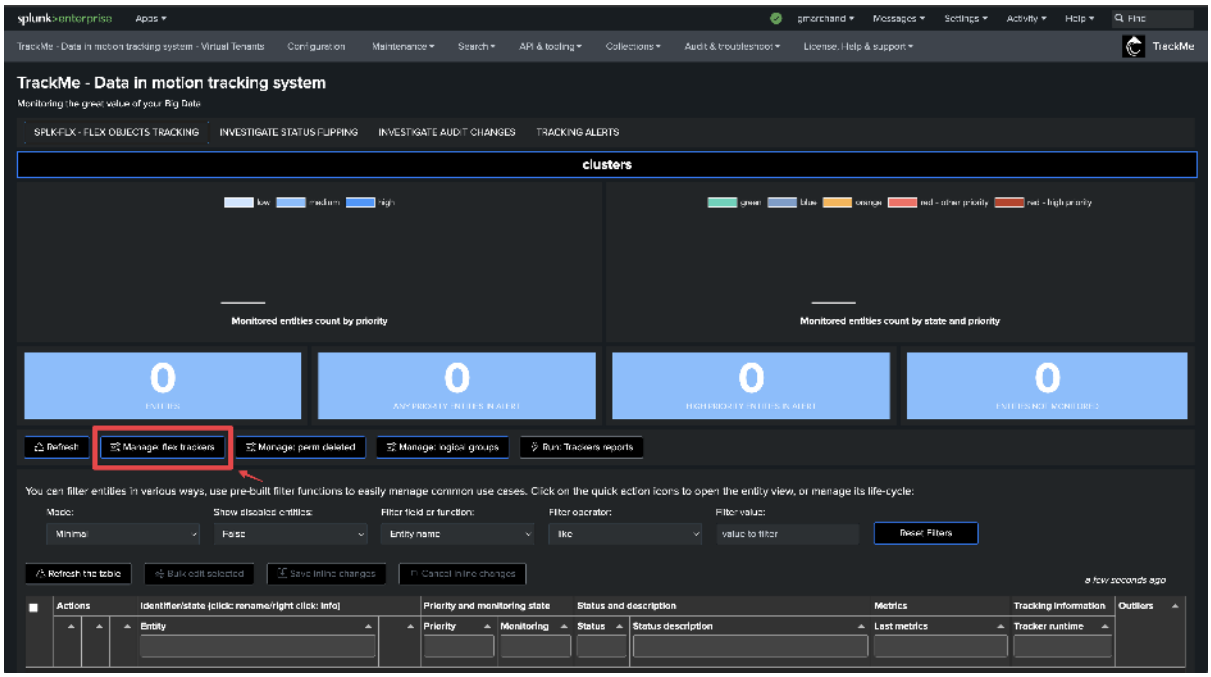
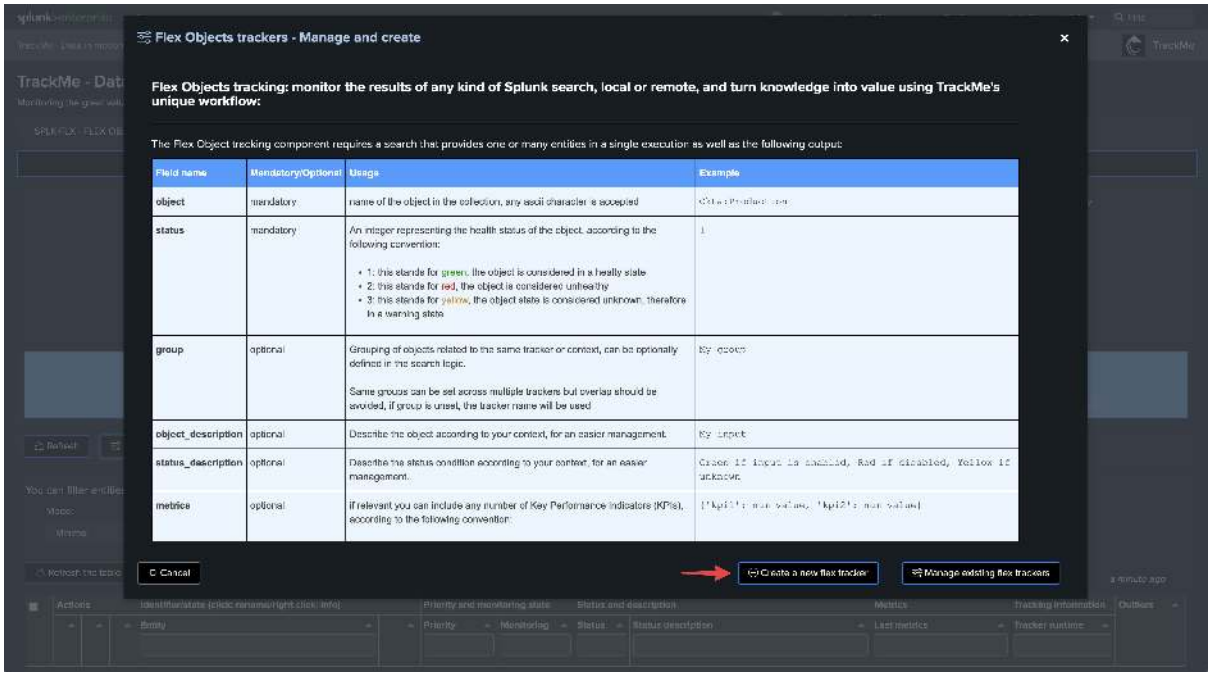
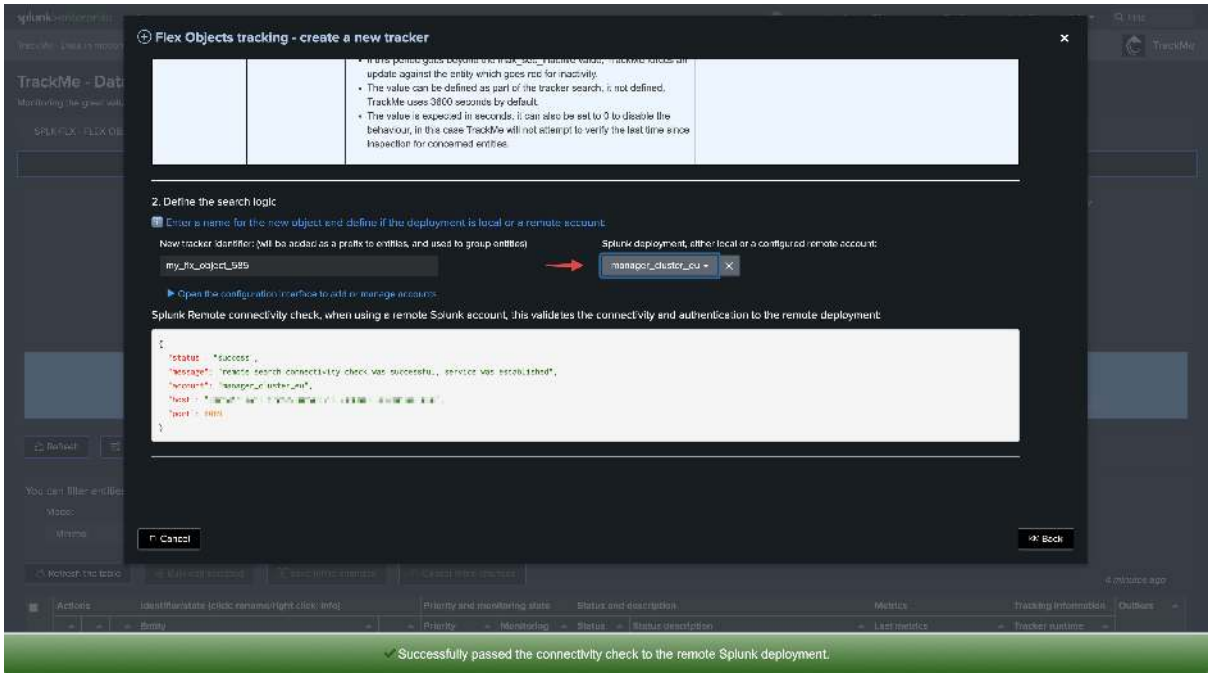
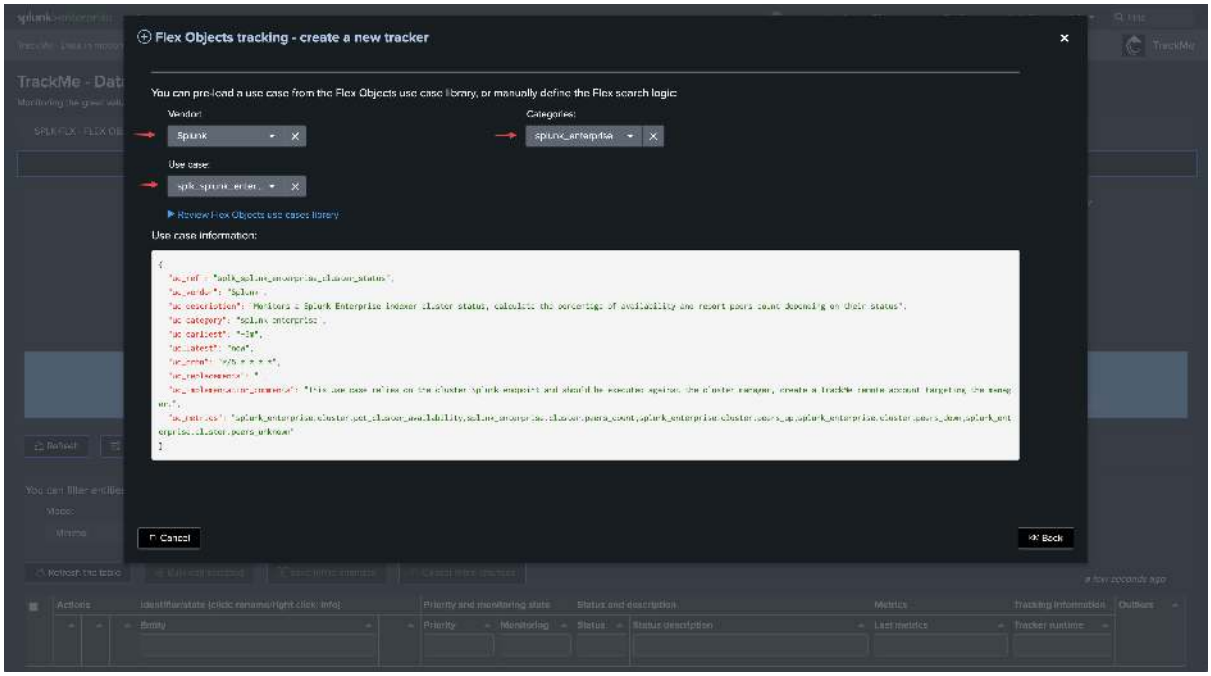
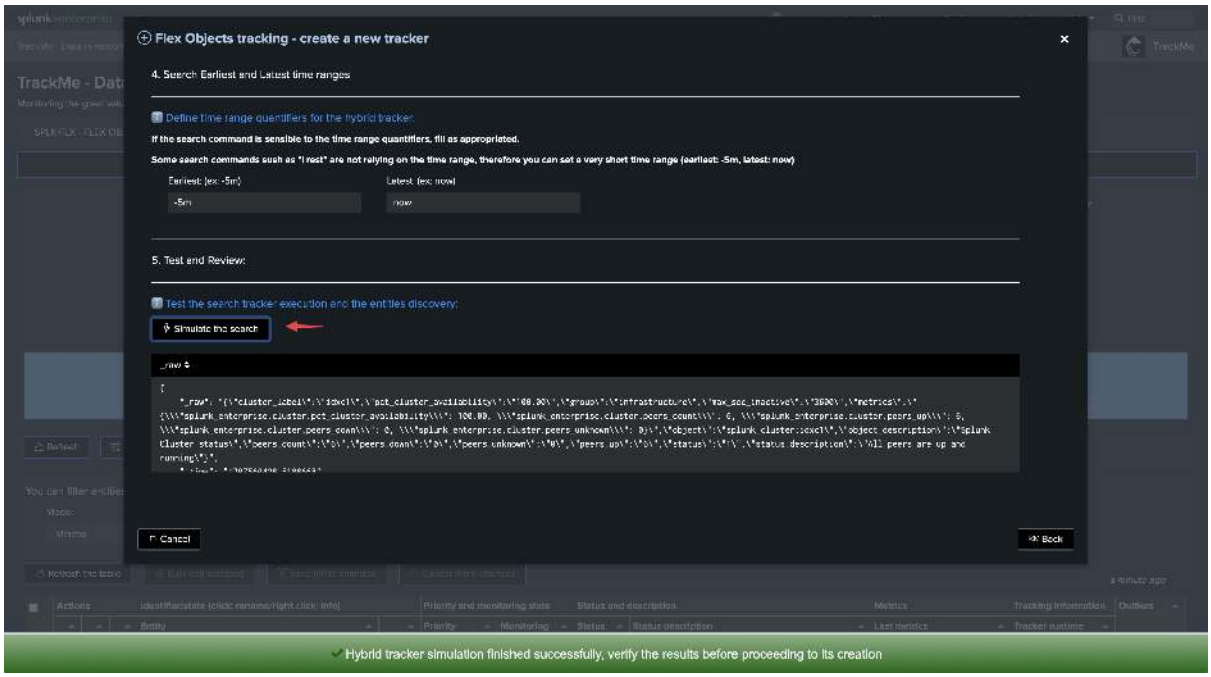
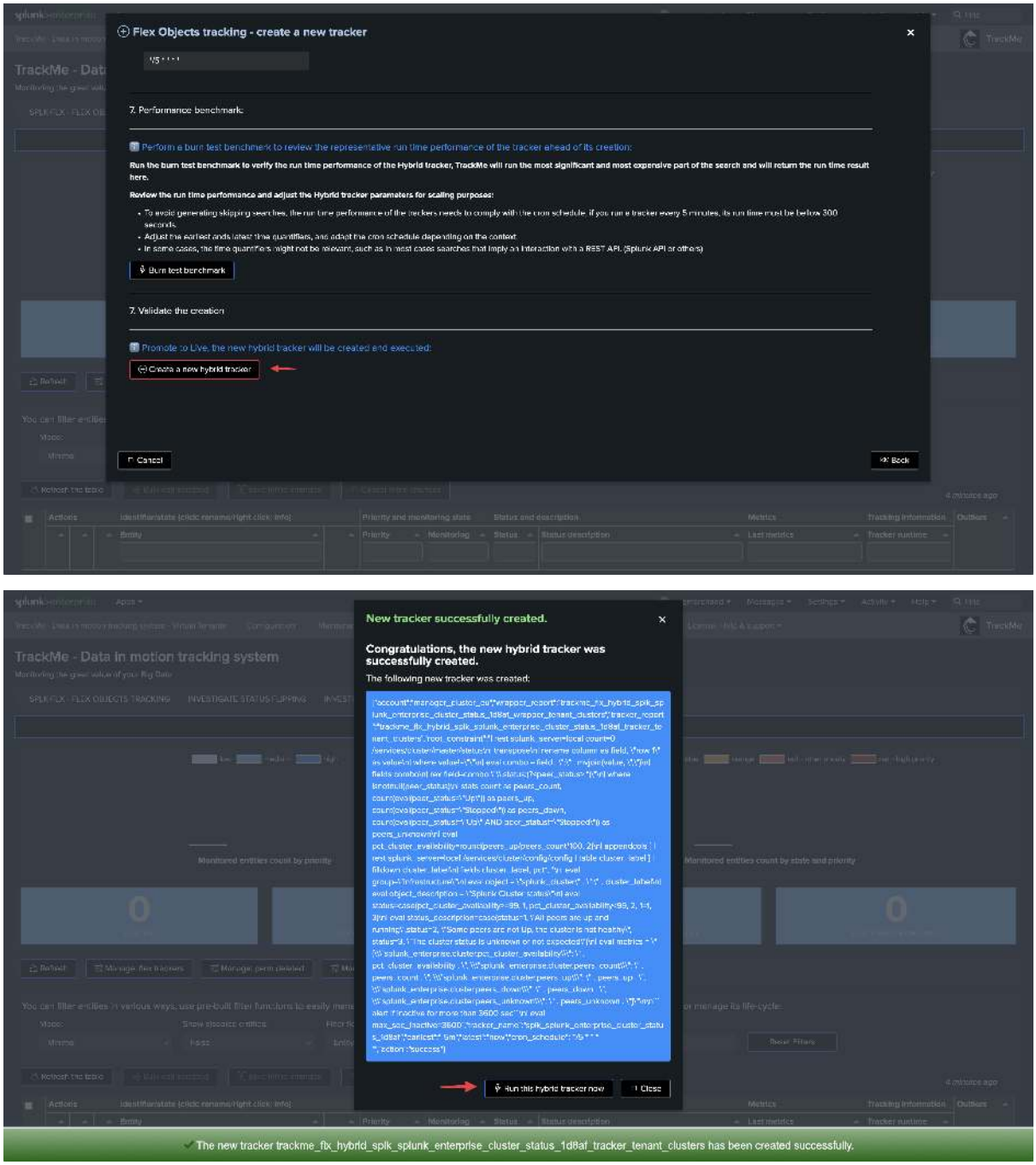
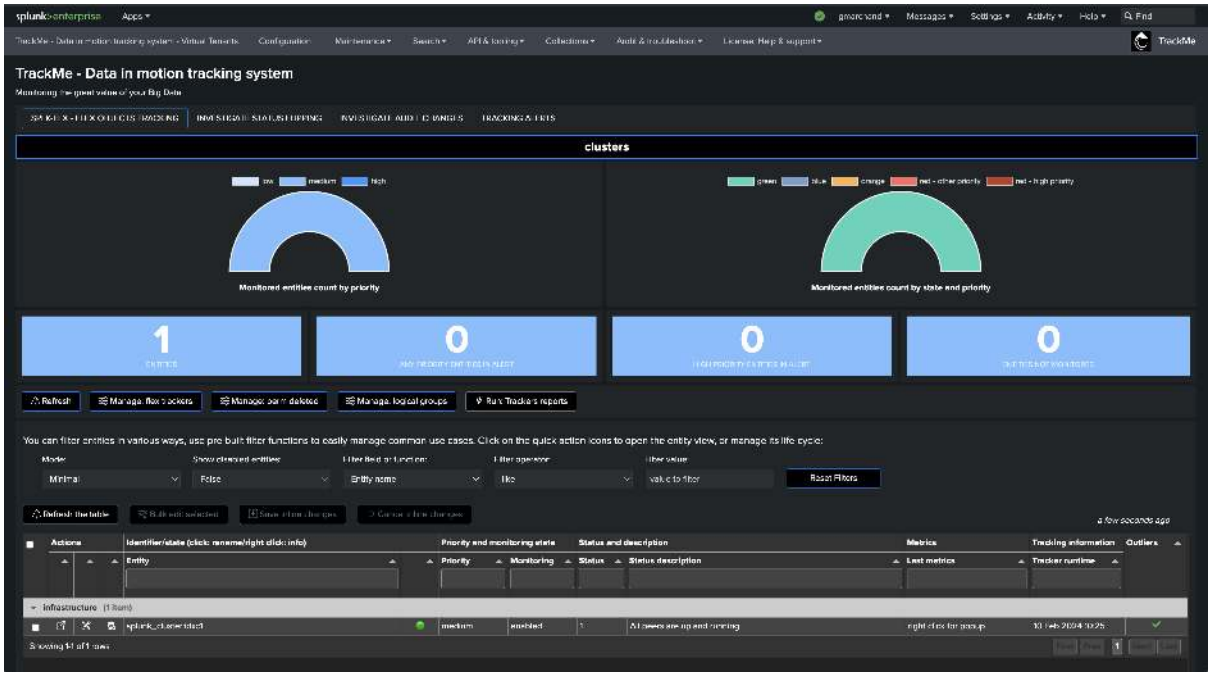
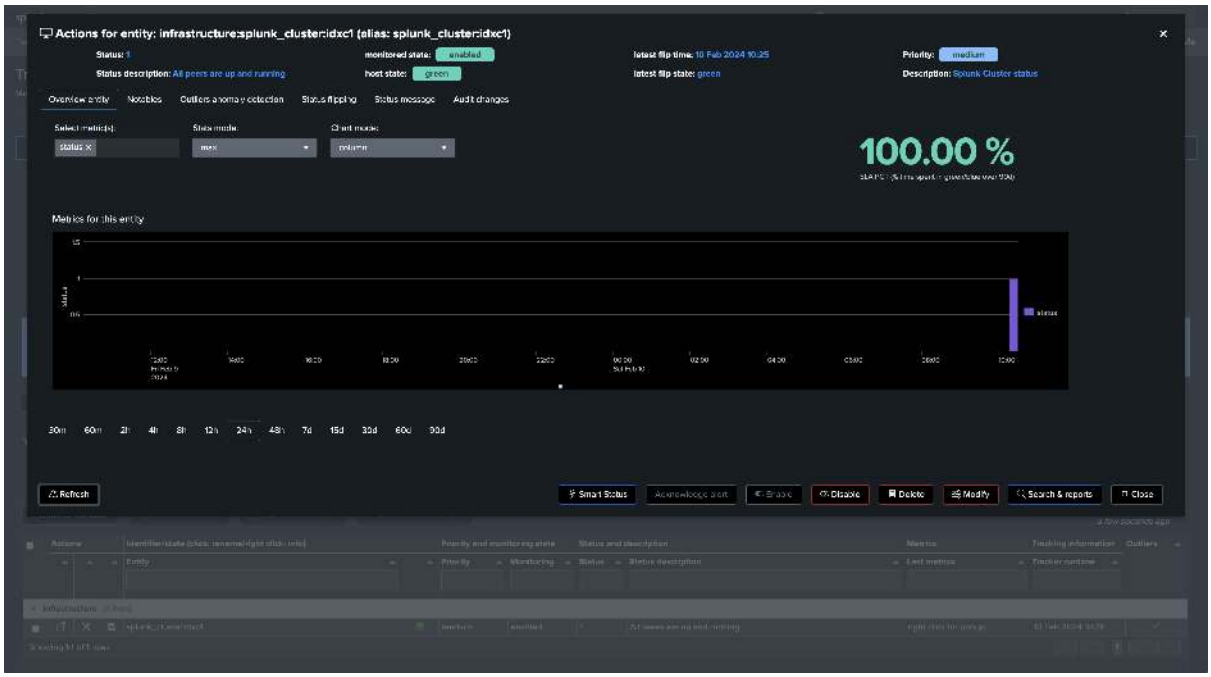
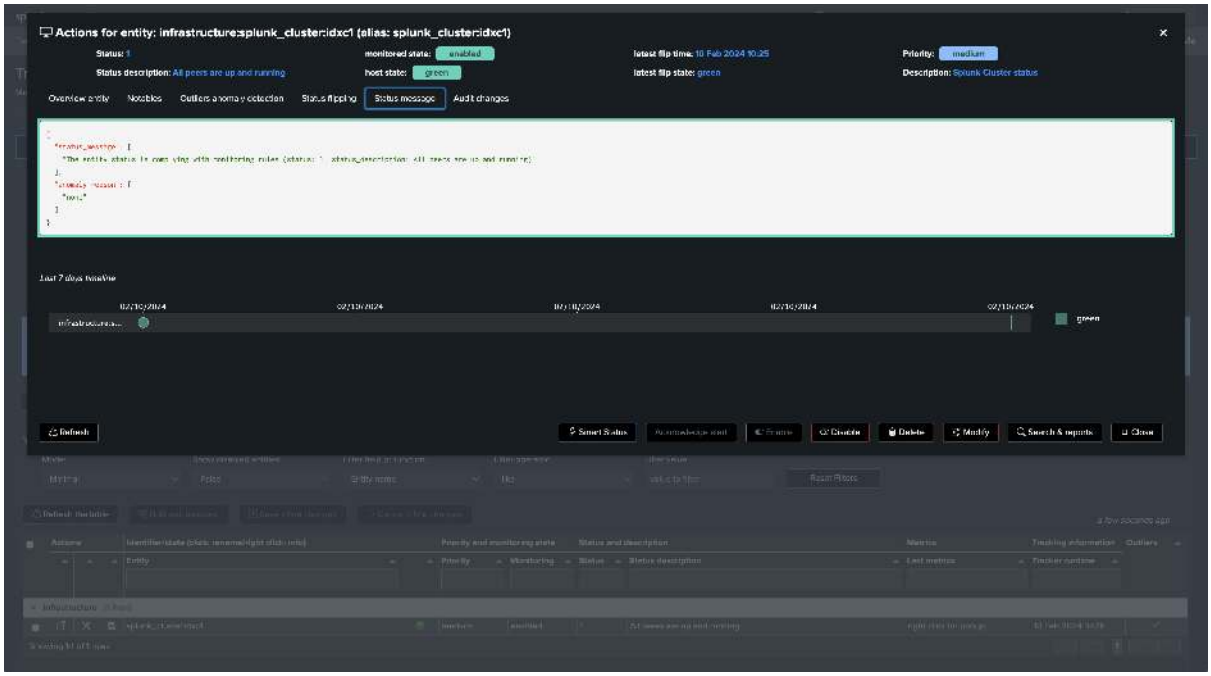
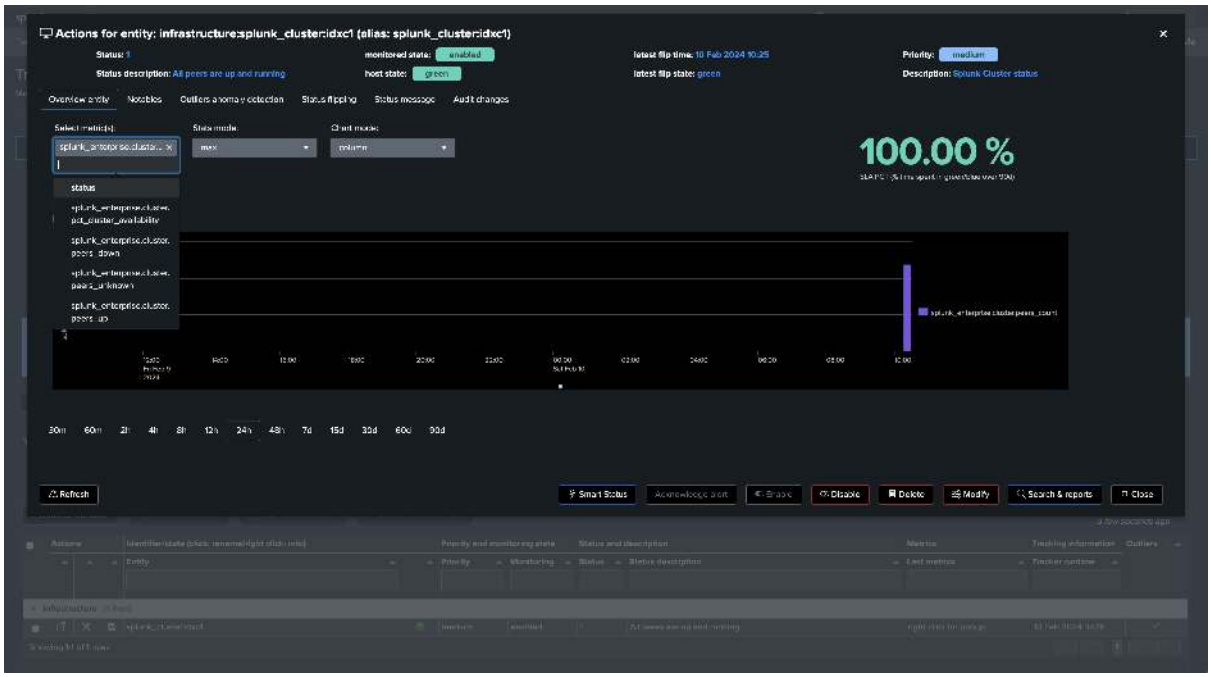
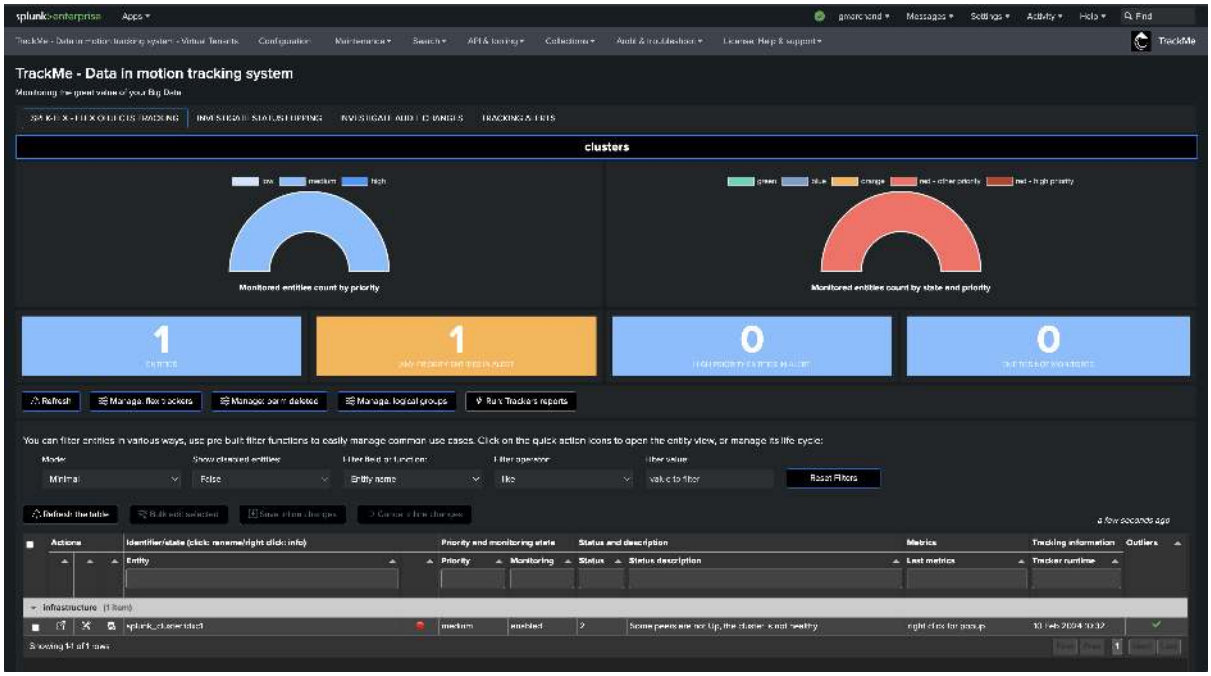
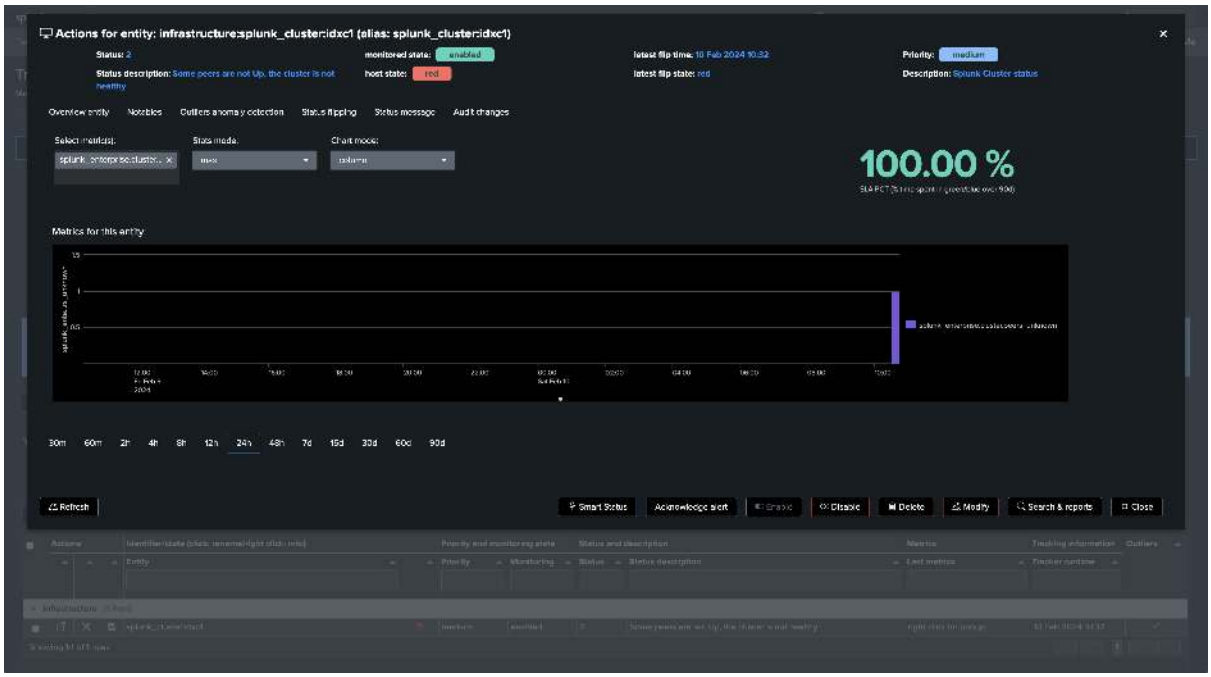
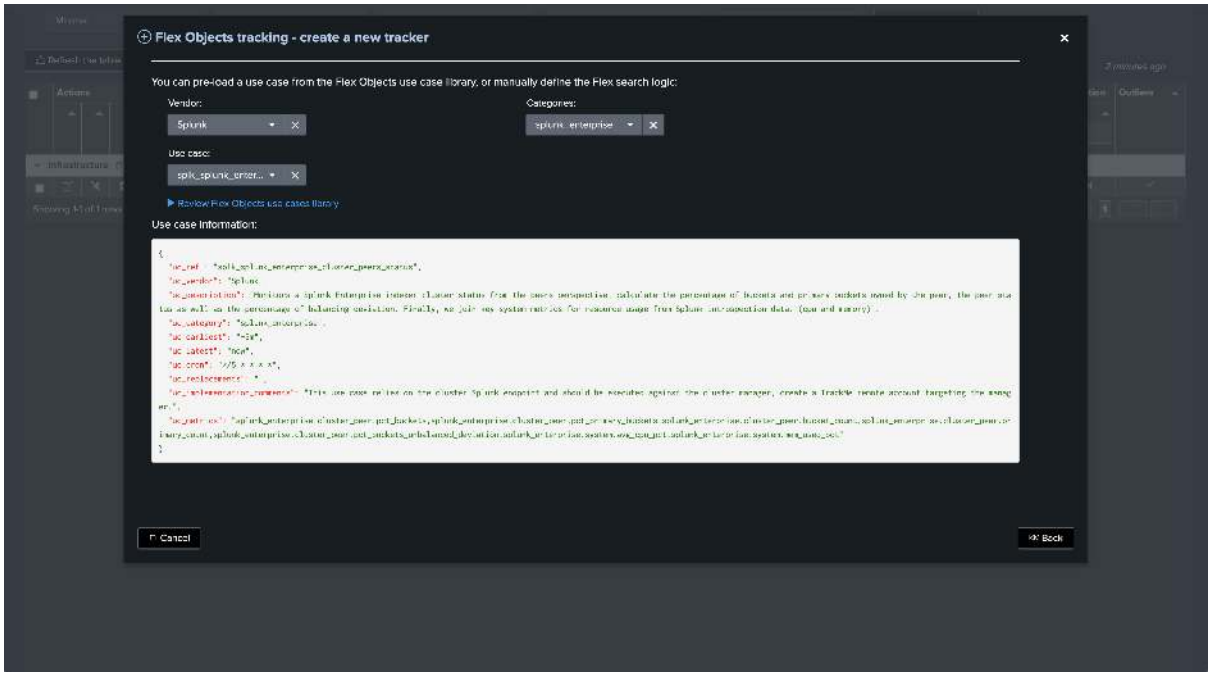
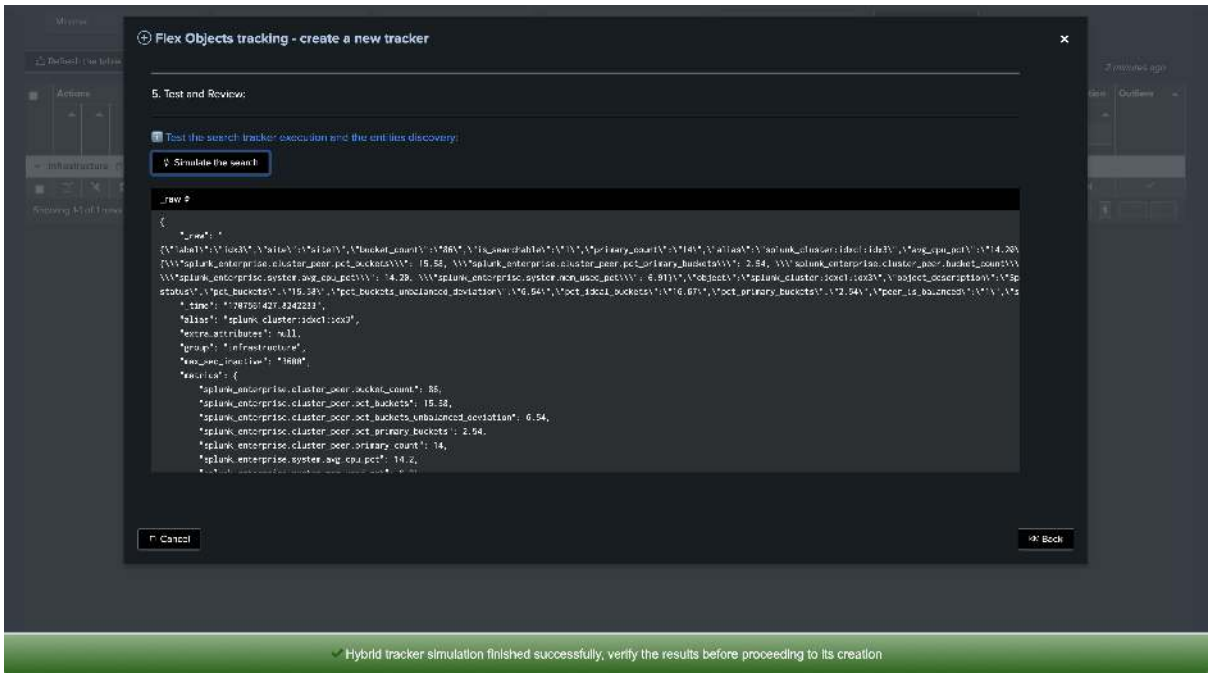
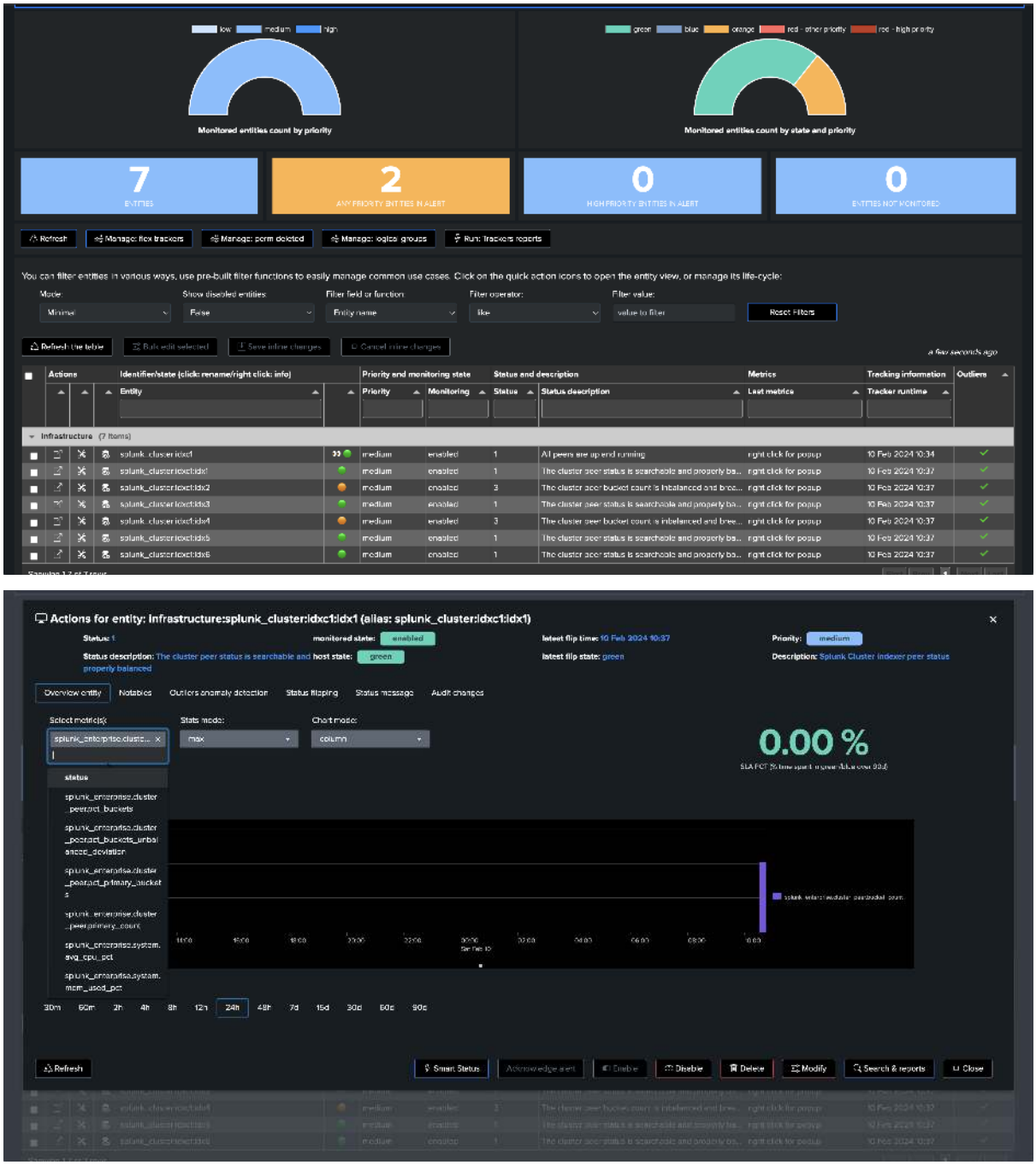
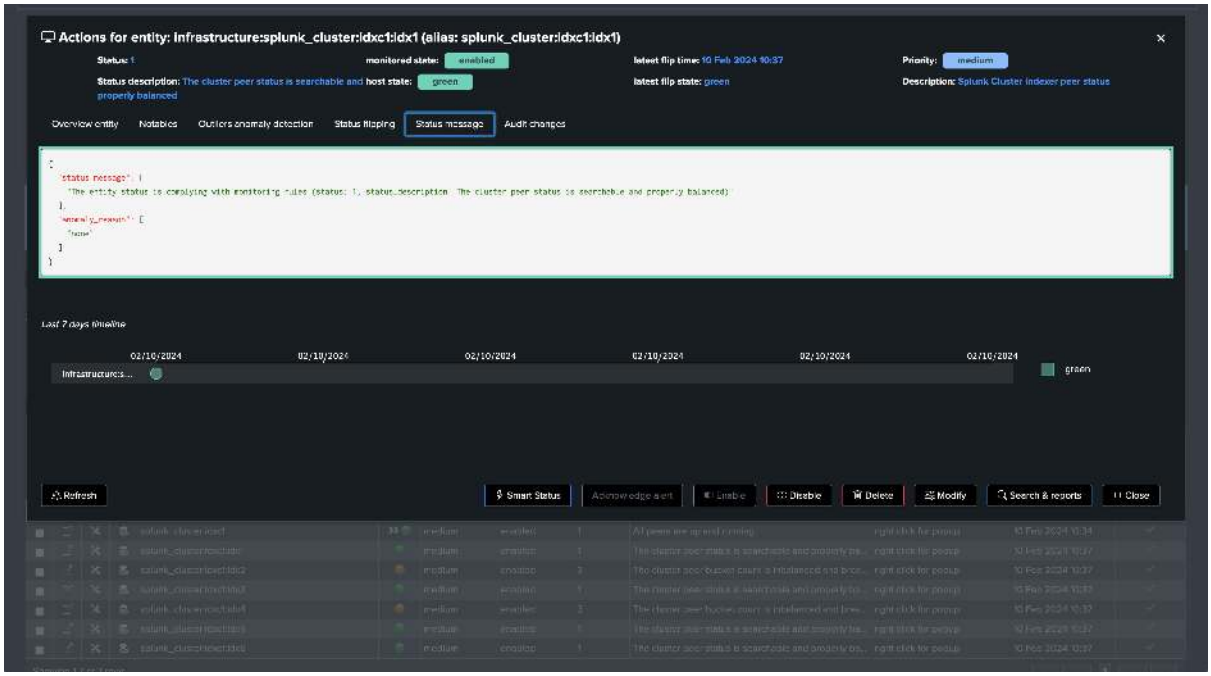
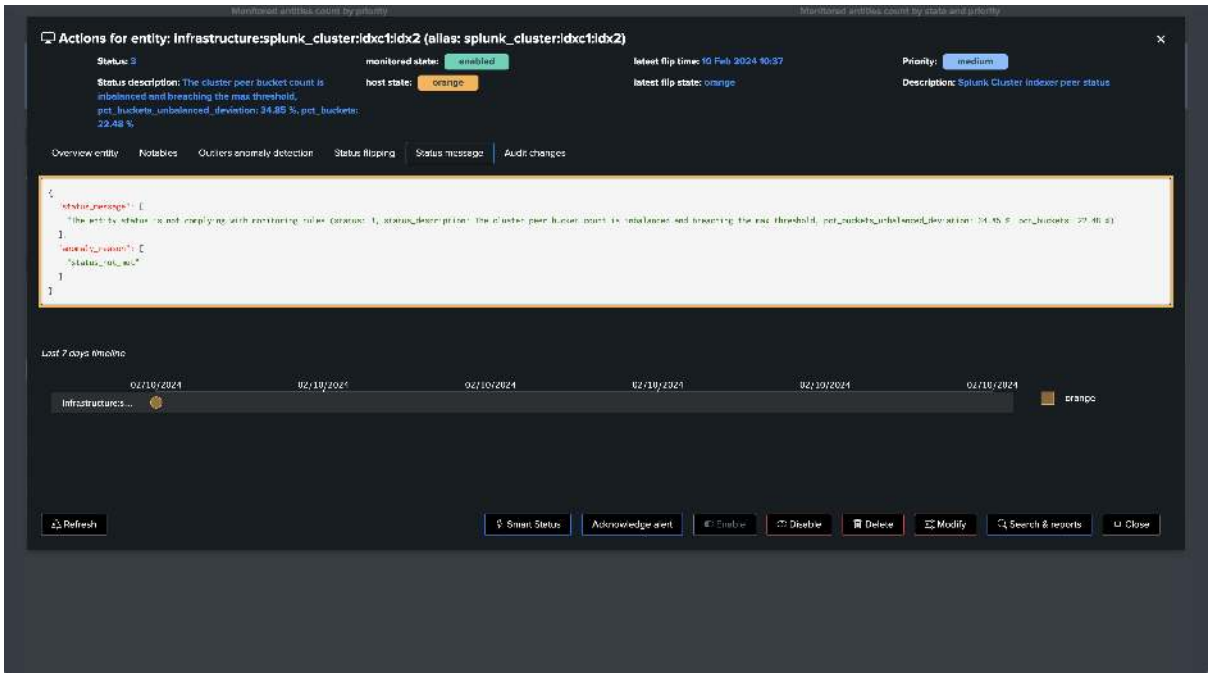
``` The basis break by statement logic happens on a per host basis, you may want to␣
↪update the logic to group by tiers logic, such as grouping by indexer cluster,␣
↪Search head, Search Head Cluster and so forth ```
| bucket _time span=30m
| stats
count(eval(log_level="INFO")) as info,
count(eval(log_level="WARN")) as warn,
count(eval(log_level="ERROR")) as error,
count(eval(log_level="FATAL")) as fatal
by _time, host
| fillnull value=0

| timechart span=30m sum(info) as info, sum(warn) as warn, sum(error) as error,␣
↪sum(fatal) as fatal
```

Several challenges need to be tackled:

- **volume**: the volume of log messages can be very high, running large searches can be very resource consuming, or not even realistic in a continuous monitoring context.
- **variability**: the volume of log messages can vary a lot, depending on time conditions, users activity and so forth.
- **relevance:**: Simply tracking for errors is not meaningful, errors happen all the time, and are not necessarily symptomatic of a problem. The key is to detect abnormal trends.

The following chart shows the challenges quite well:
```



An ad-hoc investigation can possibly detect an abnormal condition, our goal is to leverage TrackMe to do this work efficiently:



TrackMe's Machine Learning Outliers detection:



8.4.2 Implementation in TrackMe

Step 1: Create a Virtual Tenant

First, let's create a new Virtual Tenant for our use case, we may well simply use an existing Virtual Tenant, for the purposes of the documentation we will simply create a new tenant:





Step 2: Create the new Flex Object tracker

The use case was integrated in the Flex Object library in TrackMe 2.0.97, so you can simply call the template splk_splunk_infra_log_level_variations:

Note: You can find the use case search at the end of this document


Flex Objects trackers - Manage and create
Flex Objects tracking: monitor the results of any kind of Splunk search, local or remote, and turn knowledge into value using TrackMe's unique workflow:
The Flex Object tracking component requires a search that provides one or many entities in a single execution as well as the following output:
Field name
Mandatory/Optional
Usage
Example
object
mandatory
name of the object in the collection, any ascii character is accepted
status
mandatory
An integer representing the health status of the object, according to the following convention:
1: this stands for green, the object is considered in a healty state
2: this stands for red, the object is considered unhealthy
3: this stands for yellow, the object state is considered unknown, therefore in a warning state
group
optional
Grouping of objects related to the same tracker or context, can be optionally defined in the search logic.
Same groups can be set across multiple trackers but overlap should be avoided, if group is unset, the tracker name will be used
object_description
optional
Describe the object according to your context, for an easier management.
status_description
optional
Describe the status condition according to your context, for an easier management.
metrics
optional
if relevant you can include any number of Key Performance Indicators (KPIs), according to the following convention:
Cancel
Create a new flex tracker
Manage existing flex trackers


Flex Objects tracking - create a new tracker
2. Define the search logic
Enter a name for the new object and define if the deployment is local or a remote account:
New tracker identifier (will be added as a prefix to entities, and used to group entities)
spk_splunk_infra_log_level_variations
Splunk deployment, either local or a configured remote account:
local
Open the configuration interface to add or manage accounts
You can pre-load a use case from the Flex Objects use case library, or manually define the Flex search logic:
Vendor:
Splunk
Categories:
splunk infrastructu...
Use case:
spk_splunk_infra_l...
Review Flex Objects use cases library
Use case information:
"uc_ref": "spk_splunk_infra_log_level_variations",
"uc_vendor": "Splunk",
"uc_description": "Monitors the Splunk instance internal logs by the kind of the logging level. This use case then leverages Machine Outliers Anomaly detection to detect abnormal trends of errors in Splunk instance.",
"uc_category": "splunk infrastructure",
Cancel
Back



Once the tracker has been executed at least once, we start tracking the log messages logging level on a per Splunk instance basis (which you can customise):





Step 3: (optional) backfill KPIs

You can optionally backfill the KPIs, so ML models can be trained with proper dataset immediately:

Note: ensure to replace the name of the tenant in the Splunk search

```
index=_internal host=* log_level=* (log_level=ERROR OR log_level=INFO OR log_
↪level=WARN OR log_level=FATAL) earliest=-30d@d latest=now

``` The basis break by statement logic happens on a per host basis, you may want to␣
↪update the logic to group by tiers logic, such as grouping by indexer cluster,␣
↪Search head, Search Head Cluster and so forth ```
| bucket _time span=30m
| stats
count(eval(log_level="INFO")) as "trackme.splk.flx.splunk.log_level.info"
count(eval(log_level="WARN")) as "trackme.splk.flx.splunk.log_level.warn"
count(eval(log_level="ERROR")) as "trackme.splk.flx.splunk.log_level.error"
count(eval(log_level="FATAL")) as "trackme.splk.flx.splunk.log_level.fatal"
by _time, host
| fillnull value=0

| eval alias=host
| lookup trackme_flx_tenant_splunk-log-level alias OUTPUT _key as object_id, object,␣
↪object_category, tenant_id

| mcollect index=trackme_metrics split=t object, object_category, object_id, tenant_id
```

Once processed, KPIs should be backfilled depending on if the internal index retention allowed it:
```

Step 4: Review Machine Learning Outliers detection

Once TrackMe has trained Machine Learning models, you can review the results in the Machine Learning Outliers detection dashboard:

Note: You can manually run the mltrain job to force training the models now, or train a particular model through the UI



In our demo case, we know we already have an abnormal period, which can add as an exclusion so the models are not affected by this period:



Detecting abnormal error logging increasing trend:





Detecting insufficient informational logging:

Our detection is now fully ready, and TrackMe will alert us if abnormal conditions are detected on the Splunk instances and deployments.

8.5 Tracking Splunk Cloud SVC consumption in TrackMe

Monitoring the Splunk Cloud SVC consumption using TrackMe

- This white paper aims at providing a detailed guide on how to monitor the Splunk Cloud SVC consumption using TrackMe.

- The Splunk Cloud SVC consumption is a key metric to monitor in order to ensure the Splunk Cloud service is used efficiently and effectively.

8.5.1 Introduction about Splunk Cloud SVC

Whenever you are a Cloud customer paying by the volume, or by the capacity allocated to your environment (stack), SVC consumption is the key indicator in Splunk Cloud which allows you to understand the costs associated with anything that consumes computes in Splunk:

Essentially:

- Indexing related activities
- Users' ad-hoc searches
- Users and applications scheduled searches
- DataModel acceleration (DMA)

In TrackMe, you can leverage this key performance indicator to monitor the consumption of your Splunk Cloud environment, and detect abnormal trends.

To do so, we can leverage two components in TrackMe:

- TrackMe Flex Objects trackers, which can monitor SVC consumptions based on different patterns, and notably leverage Machine Learning to detect abnormal trends.
- TrackMe Workload, which can monitor the consumption of SVC at the levels of scheduled searches.

Splunk generates SVC consumption metrics in the summary index:

Global SVC consumption:

```
index=summary source=splunk-svc
```

Detailed SVC consumption per consumer:

```
index=summary source=splunk-svc-consumer
```

TrackMe leverages these indicators and stores these as metrics into the TrackMe metric indexes, and can leverage basic logics or more advanced logics using Machine Learning Outliers detection.

8.5.2 TrackMe Flex trackers for Splunk Cloud SVC consumption

We will start by creating a dedicated Virtual Tenant for the purposes of tracking Splunk Cloud SVC consumption, we can use the UI and create a new Flex Object enabled tenant:





Or we can use the trackme custom command:

```
| trackme url="/services/trackme/v2/vtenants/admin/add_tenant" mode="post" body="{
↪'tenant_alias':'splunk-cloud-svc','tenant_desc':'Splunk Cloud SVC monitoring',
↪'tenant_name':'splunk-cloud-svc','tenant_roles_admin':['trackme_admin'],'tenant_
↪roles_power':['trackme_power'],'tenant_roles_user':['trackme_user'],'tenant_owner':
↪'nobody','tenant_idx_settings':'{\'trackme_summary_idx\': \'trackme_summary\', \
↪'trackme_audit_idx\': \'trackme_audit\', \'trackme_notable_idx\': \'trackme_notable\
↪', \'trackme_metric_idx\': \'trackme_metrics\'}','tenant_outliers_set_state':'1',
↪'tenant_mloutliers':'1','tenant_cmdb_lookup':'1','tenant_default_priority':'medium',
↪'tenant_flx_enabled':1}"
```

Once the Virtual Tenant is created, we can leverage the TrackMe Flex object library and rely on the two out of the box Flex Object use cases to monitoring SVC:

Hint

TrackMe version 2.0.99

- These two use cases were last updated in TrackMe 2.0.99, make sure you are using this version or later.

TrackMe stack global SVC consumption

Once in the TrackMe UI, create a new Flex Object tracker and select the following use case template:



Scroll down and execute the tracker. You can also customize the tracker if you wish to do so, although this is not mandatory and the use case will work out of the box:




Flex Objects tracking - create a new tracker
5. Test and Review:
Test the search tracker execution and the entities discovery:
Simulate the search
Cancel
Back
Hybrid tracker simulation finished successfully, verify the results before proceeding to its creation


Flex Objects tracking - create a new tracker
7. Performance benchmark:
Perform a burn test benchmark to review the representative run time performance of the tracker ahead of its creation:
Run the burn test benchmark to verify the run time performance of the Hybrid tracker, TrackMe will run the most significant and most expensive part of the search and will return the run time result here.
Review the run time performance and adjust the Hybrid tracker parameters for scaling purposes:
Burn test benchmark
Burn test benchmark results:
Cancel
Back
Burn benchmark is now terminated, review the run time performance results.



Once executed, you will see one TrackMe entity, which corresponds to the Splunk Cloud stack name:



If you open the entity, you will see two SVC kpis:

- `splunk_cloud.svc.pct_svc_used`: the percentage of SVC used in the stack
- `splunk_cloud.svc.utilized_svc`: the amount of SVC used in the stack

Example:



After some time, this will be look like:

Looking at Outliers anomaly detection, the use case automatically created a model for SVC consumption based on the SVC units, it would eventually trigger an outlier alert if an increasing abnormal trend is detected:


Actions for entity: Splunk-Cloud:01-SvcGlobal:scde-8yra2ha5mx36rr0nw (alias: scde-8yra2ha5mx36rr0nw)
Status: 1
monitored state: enabled
latest flip time: 14 Aug 2024 12:30
Priority: medium
Status description: SVC consumption is healthy
host state: green
latest flip state: green
Description: Splunk Cloud
Click here to define tags
Overview entity
Notables
Outliers anomaly detection
Status flipping
Status message
SLA
Audit changes
Machine Learning model_id
Manage outliers detection
0
0
0
0
262
273
Outliers detection:
Refresh
Smart Status
Disable
Delete
Modify
Search & reports
Close


Actions for entity: Splunk-Cloud:01-SvcGlobal:scde-8yra2ha5mx36rr0nw (alias: scde-8yra2ha5mx36rr0nw)
Status: 1
monitored state: enabled
latest flip time: 14 Aug 2024 12:30
Priority: medium
Status description: SVC consumption is healthy
host state: green
latest flip state: green
Description: Splunk Cloud
Click here to define tags
Overview entity
Notables
Outliers anomaly detection
Status flipping
Status message
SLA
Audit changes
0
outliers
Refresh
Smart Status
Disable
Delete
Modify
Search & reports
Close



TrackMe stack app consumer SVC consumption

Similarly, we can now leverage the second out of the box Flex Object use case to monitor the SVC consumption at the app level:



Once created and executed, this use case tracks the SVC consumption by the app consumer as registered by the SVC consumption KPIS:



The same SVC KPI is leveraged per consumer:



After some time, once we have started to collect enough historical knowledge:



Anomaly Outliers detection:





From this stage, if a consumer starts to abnormally consume SVC, TrackMe's Outlier detection will eventually trigger an alert.

Accessing TrackMe's metrics and building your own dashboards and reports

You can easily access to the metrics collected for a given TrackMe Virtual Tenant from the Virtual Tenant home UI:





Query example:

```
| mstats avg(trackme.splk.flx.splunk_cloud.svc.utilized_svc) as avg_svc,
max(trackme.splk.flx.splunk_cloud.svc.utilized_svc) as max_svc,
sum(trackme.splk.flx.splunk_cloud.svc.utilized_svc) as sum_svc,
perc95(trackme.splk.flx.splunk_cloud.svc.utilized_svc) as perc95_svc
where index=trackme_metrics tenant_id=svc-tracking object_category="splk-flx"␣
↪object=* by object
| foreach *_svc [ eval <<FIELD>> = round('<<FIELD>>', 2) ]
```



Building a dashboard based on the above, the following would easily show the bigger SVC consumers:

```
| mstats avg(trackme.splk.flx.splunk_cloud.svc.utilized_svc) as avg_svc,
max(trackme.splk.flx.splunk_cloud.svc.utilized_svc) as max_svc,
```

(continues on next page)

(continued from previous page)

```
sum(trackme.splk.flx.splunk_cloud.svc.utilized_svc) as sum_svc,
perc95(trackme.splk.flx.splunk_cloud.svc.utilized_svc) as perc95_svc
where index=trackme_metrics tenant_id=svc-tracking object_category="splk-flx"␣
↪object=Splunk-Cloud:02-* by object
| foreach *_svc [ eval <<FIELD>> = round('<<FIELD>>', 2) ]
| fields object avg_svc
```



8.5.3 TrackMe Workload component and Splunk Cloud SVC consumption

TrackMe's licensed customers can leverage the Workload component, which can also track SVC consumption on a per Splunk scheduled basis.

When you create the Workload Virtual Tenant, ensure to select "Splunk Cloud" as the type of Splunk environment:



Once configured, the Workload component tracks various aspects of the health of Splunk scheduled, between other KPIs, the Workload component also tracks SVC usage:





We can for instance add an Outlier models against the SVC consumption:



Note that by default, the Workload component would enable ML against the elapsed run time KPI, you can for instance while creating the tenant, or at a later stage, use SVC consumption instead.



Accessing metrics from the Workload component

The following search query can be used to access the SVC consumption metrics collected by the Workload component:

You need to replace the tenant_id name with the one you have created.

```
| mstats avg(trackme.splk.wlk.svc_usage) as avg_svc,
max(trackme.splk.wlk.svc_usage) as max_svc,
sum(trackme.splk.wlk.svc_usage) as sum_svc,
perc95(trackme.splk.wlk.svc_usage) as perc95_svc
where index=trackme_metrics tenant_id=workload object_category="splk-wlk" object=* by␣
↪object
| foreach *_svc [ eval <<FIELD>> = round('<<FIELD>>', 2) ]
```

Based on the previous example, you can easily build very interesting dashboards on top of this:



8.5.4 TrackMe out of the box SVC consumption dashboard

While not directly related to the SVC consumption TrackMe tracks through the components explained above, it is worth mentioning that TrackMe also comes with an out of the box dashboard which is designed to audit the SVC usage that is related to TrackMe itself.

You can find this dashboard in the menu "Audit & Troubleshoot" / "TrackMe SVC usage stack":

Note: the stack SVC entitlement comes as N/A because it is a dev Cloud stack which doesn't come with a normal limit, unlike a proper Cloud stack.



8.6 Monitor Splunk Workload with TrackMe's Workload component

Monitoring Splunk Search Head Clusters with TrackMe Flex Objects

- This tutorial demonstrates the monitoring of **Splunk Workload** with the TrackMe Workload component.
- The Workload component is a component restricted to licensed customers. Please contact Splunk Sales for more information.
- Using these steps will enable TrackMe to continuously monitor Splunk scheduled activity, detect execution anomalies and changes in behavior, as well as performing versioning of searches definition.
- Monitoring the Splunk scheduling workload is a critical part for Splunk. For instance, detecting

when your SIEM correlation searches are affected by any issue is as crucial as detecting when feeds have stopped feeding your use cases.

- With TrackMe's remote search capabilities, you can monitor as many Search Head tiers as you need from a single pane of glass.

8.6.1 Requirements

In most cases, you will want to target a different Search Head tier than the one where the TrackMe instance is running.

For example, if you are running **Splunk Enterprise Security**, this is normally in a different Search Tier. We will therefore start by creating a Splunk Remote Account in TrackMe to access the Search Head tier, programmatically speaking.

8.6.2 Step 1: Create a Splunk Remote Deployment Account for the SHC

The first step is to create a Splunk Remote Deployment Account for the Cluster Manager. For more information about TrackMe Remote Search capabilities and configuration:

Splunk Remote Deployments (splunkremotesearch)

On the Search Head Cluster, create a new Splunk bearer token for the TrackMe Remote Deployment Account:



In TrackMe, click on Configure / Remote Deployment Accounts and add a new account:

- If running a Search Head Cluster, you can specify each of the SHC members in a comma-separated list or use a load balancer URL.
- If running a standalone Search Head, you can specify the Splunk API URL of the Search Head.
- If multiple endpoints are specified, TrackMe automatically dispatches searches randomly amongst available and responding members (it validates connectivity and authentication)





Managing multiple Search Head tiers

- If you have multiple Search Head tiers to be monitored, you can create a Remote Deployment Account for each tier.
- You will then be able to manage and monitor as many Search Head tiers as you need from a single pane of glass in TrackMe.

8.6.3 Step 2: Create a Workload tenant and use the wizard to create trackers automatically

Access the tenant creation wizard and select the Workload component



Choose the tenant_id and other main information



Define the type of Splunk deployment (Enterprise/Cloud)

Splunk Enterprise vs Splunk Cloud

- The main difference here is that if Splunk Cloud is selected, TrackMe will create necessary trackers to monitor the SVC consumption of monitored scheduled searches.
- Select Splunk Cloud if the environment being monitored is a Splunk Cloud environment.



Select the Search Head tier target

Search Head target

- Although some of the search logics TrackMe implements could indeed be run on the TrackMe instance itself if both can search the same indexers, there are much deeper API-related logics which come into account for the Workload components.
- Therefore, it is important to select the right Splunk Remote Deployment Account targeting the desired Search Head tier.



Define Search Restrictions

Search Restrictions

- For each of the main categories of activity, you need to define and/or restrict the Search Heads to be considered for monitoring.
- This is important to avoid monitoring searches that are out of the scope of the Search Head you target, but whose underlying activity can be accessed from the TrackMe instance.
- In the example below, we restrict to a given Search Tier using the filter `(host=sh1 OR host=sh2 OR host=sh3)`



Tracking Enterprise Security Notables

Tracking Notable events

- The Workload component can track notable events generated by correlation search, and turns this into a TrackMe metric.
- You can then use TrackMe to review the number of notable events created by your correlation searches.



ML Outliers Detection

Outliers detection and Workload

- TrackMe can use ML Outliers detection to automatically detect abnormal trends for a given metric.
- In the context of the Workload, TrackMe by default generates and trains models against the run time in seconds of the searches, so it can detect an abnormal trend in the time taken by searches to be executed.
- You can customize and/or disable this behavior.



Grouping

Workload grouping

- In the Workload component, the default behavior is to group TrackMe entities per Splunk application namespace.
- This means you should not monitor multiple Search Head tiers within the same tenant (entities would be mixed all together).
- If you wish to target multiple Search Head tiers in the same tenant, you need to use the grouping to group against a custom pattern.
- You can also simply use this setting to force TrackMe not to group per application namespace.



Inactive entities

Managing entities without any activity

- When a search schedule is disabled, it will become inactive after some time.
- This setting instructs what TrackMe should do with these entities. The default behavior is to remove entities which have not been active for more than 15 days.



Validate the tenant creation and start

After the tenant is created, wait up to 5 minutes before seeing the first entities in TrackMe

- Immediately after the tenant is created, TrackMe starts to orchestrate the Workload detection logic.
- This can take up to 5 minutes before you can start seeing entities created in the Workload component.





8.6.4 Step 3: Start using the Workload component, detect and investigate scheduling issues

Ignoring specific Splunk application namespaces

- You can ignore specific Splunk application namespaces from the Workload component.
- This can be useful to remove Splunk out-of-the-box applications which are not relevant for your monitoring.
- Click on "Manage: enable/disable apps" to access the configuration screen





Use case: detecting skipping searches

What are skipping searches and how these affect Splunk workload

- Skipping searches happen when a given search takes too long to be executed and is then skipped by the scheduler.
- The current cron schedule defines how often the search is executed and indirectly defines how long in seconds the search can run before a new search should normally be executed.
- If Splunk encounters that a search is already running at this stage, its execution will be skipped.

Depending on the percentage of skipping searches, TrackMe sets the entities status to orange or red:





TrackMe's versioning, cron schedule and cron sequence

- TrackMe performs versioning of all monitored searches and detects when a search is updated.
- It also extracts the cron schedule and uses the `croniter` Python library to calculate the maximum run time sequence in seconds of the search.
- You can use this information to easily understand, without quitting TrackMe, the reasons for skipping searches.

In this example, we can see that the cron schedule is set to run every 5 minutes, and that the maximum run time sequence is 300 seconds:



However, The run time execution is well over acceptable values, and the search is skipped:



This search is clearly encountering quality issues, and needs to be reviewed and redesigned to perform properly or have a more adapted cron schedule.

Use case: detecting searches suffering from execution anomalies

Execution errors typically indicate non-working searches and may slightly affect your Splunk use cases

- There are plenty of reasons why a Splunk search could start failing at some point in its lifecycle.
- This may happen due to temporary infrastructure issues, incorrect syntax, breaking change introduction, third-party application installation or updates... plenty!
- This gets also more complex and challenging with the continuous growth of the environment and use cases. Very often, use cases fail silently and are not detected until a major issue is encountered.
- With TrackMe's Workload component, anomaly executions are detected at scale, made easily available and automatically investigated for use cases and platform owners to review as soon as possible.

Searches suffering from execution anomalies are tagged with the anomaly_reason=execution_errors_detected



Open the entity and review when execution errors were detected



In a single click, you can use the SmartStatus to retrieve a meaningful sample of the Splunk scheduler execution errors:

Hint

SmartStatus provides access to the _internal execution errors to use cases owners with no access to the _internal index

- With the SmartStatus and TrackMe's RBAC management, use case owners can transparently access to the execution errors from the _internal, even if they do not have access to the _internal index.
- This is possible via controlled elevation of privileges, users that are granted access to the TrackMe API endpoints through the RBAC configuration, will be able to execute the SmartStatus action which itself is executed with system level privileges.
- Finally, TrackMe also executes the SmartStatus automatically via the builtin alert action, and indexes the results of it into the associated TrackMe summary index.


Actions for entity: DA-ESS-sandbox:svc-siem:Threat - UC004 - Disallowed accesses (typo in lookup call) - Rule (alias: DA-ESS-sandbox:svc-siem:Threat - UC004 - Disallowed accesses (typo in lookup call) - Rule)
Status: 2
monitored state: enabled
latest flip time: 13 Feb 2024 01:06
Priority: medium
Status description: degraded
host state: red
latest flip state: red
Description: No description
Overview entity
Notables
Outliers anomaly detection
Status flipping
Status message
Audit changes
Versioning Metadata
71
42
0.00 %
01:24:59
39.86 %
Select a metric:
Stats mode:
Chart mode:
Break by version_id:
count_errors
count_completed
max
column
False
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
Refresh
Smart Status
Manage Acknowledgement
Enable
Disable
Delete
Modify
Search metrics
Search scheduler skipping
Search scheduler errors
Close


Smart Status
The Smart Status performs advanced investigations and correlations dynamically depending on the state of the entity:
Time
Event
13/02/2024 01:18:25.793
Close
Back
Search alert actions history





Use case: Use TrackMe's version to establish relations between search updates and changes in behavior

Using TrackMe's Workload search versioning, you can easily detect changes and make the link between these changes and how these impact the search behavior

- Once TrackMe discovers a search entity, it starts to continuously track the code version of the search.
- TrackMe creates a dimension called `version_id` which is the unique MD5 hash of the search definition and time quantifiers.
- This dimension is indexed with the Workload metrics and represents the unique instance of a search at some point in time.

- When a change to the search is detected, a new value of the `version_id` is generated, the search version KVstore record is updated and TrackMe generates a version event which is indexed in the TrackMe summary index of the tenant.
- TrackMe also issues a diff of the search definition, time quantifiers and attempts to identify when and who updated the search, which information is added to the versioning record and event.
- Finally, you can review changes easily in TrackMe UI, whether this search is local to the TrackMe host or hosted on any remote deployment, without quitting TrackMe for a second.

Say you have a search which suddenly started to generate anomalies execution, TrackMe captures this fact and impacted the entity:





Looking at the version tab, you can see the changes made to the search, and when these were made:





From the metrics perspective, we break by the version_id and identify when the change was made and how this impacted the search behavior:





You can also look at the corresponding versioning events:

```
index=trackme_summary sourcetype="trackme:wlk:version_id" object=*
```



Use case: Detect delayed searches

By monitoring scheduled searches, TrackMe also detects if a search has stopped being executed

- Using its versioning capabilities, TrackMe interprets the cron schedule and defines the `cron_exec_sequence_sec` (using the Python library croniter).
- This value in seconds represents the time expected between two executions of the search.
- If a search that was discovered suddenly stops being executed, TrackMe will detect it and tag the entity as delayed.
- TrackMe applies a 60-minute delay grace time before impacting the entity.
- This is a vital verification, especially for SIEM environments. Being capable of detecting use cases that are not executed anymore is very important and should lead to the investigation of the issue.

In the versioning tab, you can see the cron_schedule as well as the cron_exec_sequence_sec:



TrackMe detects that the search has stopped being executed, and tags the entity as delayed:





The search was indeed disabled at the Splunk level, which might have been intentional, or not, and should be reviewed:



Use case: Monitoring Data Models Acceleration and Reports Acceleration behaviors and performance

Splunk Accelerated Searches in the Workload component

- TrackMe also discovers and monitors searches that relate to Splunk Data Models (DMA) and Reports Acceleration.
- Especially for DMA, it is important to monitor the level of Skipping searches per Data Model, as well as the elapsed time performance of these searches.
- TrackMe will, for instance, automatically alert if these are suffering from high skipping searches

ratios.

- Finally, TrackMe also uses Machine Learning to detect Outliers regarding the performance of the searches and may alert if an abnormal increasing trend is detected.

DMA and report acceleration searches are automatically discovered and monitored by TrackMe:



All metrics are available, such as the elapsed: (run time in seconds)



This environment is struggling to maintain DMA for instance, indicating underneath issues that can affect your security use cases:



Splunk Cloud customers can easily review related SVC usage:



Splunk Enterprise & Cloud can also review CPU usage and memory consumption metrics:



TrackMe's unique ML Outliers detection clearly catched the bad situation:

There are plenty more use cases that can be covered by the Workload component, and the above are just a few examples.

Some more documentation:

- *Workload (splk-wlk) - Manage Workload tenants and trackers*
- *Splunk Workload (splk-wlk)*

8.7 Monitor Splunk Indexer Clusters

Monitoring Splunk Indexer Clusters with TrackMe Flex Objects

- This tutorial demonstrates the monitoring of **Splunk Indexer Clusters** with TrackMe Flex Objects.
- TrackMe Flex Objects is a component restricted to licensed customers. Please contact Splunk Sales for more information.
- Using these steps will enable TrackMe to continuously monitor the health of your Splunk Indexer Clusters and alert you when issues are detected.
- With TrackMe's remote search capabilities, you can monitor as many Indexer Clusters as you need from a single pane of glass.

The following screen shows the final results in TrackMe, starting from Step 1 to easily implement the monitoring of your Splunk Indexers Clusters.


splunk>enterprise
TrackMe - Data in motion tracking system - Virtual Tenants
Create new tracking tenant
Create new replica tenant
Refresh all
TrackMe
Developer mode
1
COMPLETED
SCHEDULER COMPLETED PERCENTAGE
OPERATIONAL
TENANT OPS STATUS
a few seconds ago
Edit my theme preferences
Edit my position preferences
Save current positions to my profile
Reset my user profile
11 - Indexer Clusters: operational
Splunk Clusters Indexers
Number of entities: 7
Low priority red status: 0
Medium priority red status: 0
High priority red status: 0


TrackMe - Data in motion tracking system
clusters
Monitored entities count by priority
Monitored entities count by state and priority
7
2
0
0


TrackMe - Data in motion tracking system
clusters
7
0
0
0


TrackMe - Data in motion tracking system
clusters
7
3
0
0





8.7.1 Step 1: Create a Splunk Remote Deployment Account for the Cluster Manager

The first step is to create a Splunk Remote Deployment Account for the Cluster Manager. For more information about TrackMe Remote Search capabilities and configuration:

Splunk Remote Deployments (splunkremotesearch)

On the Cluster Manager, create a new Splunk bearer token for the TrackMe Remote Deployment Account:



In TrackMe, click on Configure / Remote Deployment Accounts and add a new account:







Hint

Managing multiple Indexer Clusters

- If you have multiple Indexer Clusters, you can create a Remote Deployment Account for each Cluster Manager.
- You will then be able to manage and monitor as many Indexer Clusters as you need from a single pane of glass in TrackMe.

8.7.2 Step 2: Create a Flex Object tenant for monitoring Indexer Clusters

Now, let's create a new tenant for the purposes of monitoring our Indexer Clusters. We can achieve this via the UI wizard or with a simple command line:

```
| trackme url=/services/trackme/v2/vtenants/admin/add_tenant mode=post body="{'tenant_
↪name': 'clusters', 'tenant_alias': '11 - Indexer Clusters', 'tenant_desc': 'Splunk␣
↪Clustered Indexers', 'tenant_roles_admin': 'trackme_admin', 'tenant_roles_power':
↪'trackme_power', 'tenant_roles_user': 'trackme_user', 'tenant_owner': 'admin',
↪'tenant_idx_settings': 'global', 'tenant_flx_enabled': 'true'}"
```

8.7.3 Step 3: Create Flex Object trackers using the Flex Objects library

Hint

Find the use cases definition (SPL) in annex of this documentation

- See: *Annex: Use cases definition (SPL)*

Enter the newly created tenant and start the Flex Object wizard using the built-in Flex Object library:




Tenant overview
Overview of the tenant id: clusters / alias: 11 - Indexer Clusters / Splunk Clustered Indexers
SPLK-FLX is enabled
Summary Index
Audit Index
Notable Index
Metrics Index
SPLK Flex Objects Tracking (SPLK-FLX) component overview
0 entities
Close
Open this tenant
Refresh this
SLA overview
Manage this tenant
Hide this tenant from my profile
Overview of the tenant id: clusters / Splunk Clustered Indexers - Get tenant activity insights and manage its life cycle from this screen


TrackMe - Data in motion tracking system
Monitoring the great value of your Big Data
SPLK-FLX - FLEX OBJECTS TRACKING
INVESTIGATE STATUS FLIPPING
INVESTIGATE AUDIT CHANGES
TRACKING ALERTS
clusters
Monitored entities count by priority
Monitored entities count by state and priority
Refresh
Manage flex trackers
Manage: perm deleted
Manage: ogiss groups
Run Trackers reports
You can filter entities in various ways, use pre-built filter functions to easily manage common use cases. Click on the quick action icons to open the entity view, or manage its life-cycle:
Mode:
Show disabled entities:
Filter field or function:
Filter operator:
Filter value:
Reset Filters
Refresh the table
Bulk edit selected
Save inline changes
Cancel inline changes
a few seconds ago
Actions
Identifier/state (click: rename/right click: info)
Entity
Priority and monitoring state
Priority
Monitoring
Status and description
Status
Status description
Metrics
Last metrics
Tracking information
Tracker runtime
Outliers



Select the use cases and remote accounts:

We will start with the use case "splk_splunk_enterprise_cluster_status" which tracks the global indexer cluster status using the manager API endpoints.




Flex Objects tracking - create a new tracker
You can pre-load a use case from the Flex Objects use case library, or manually define the Flex search logic:
Vendor:
Splunk
Categories:
splunk_enterprise
Use case:
Review Flex Objects use cases library
Use case information:
Cancel
Back


Flex Objects tracking - create a new tracker
4. Search Earliest and Latest time ranges
Define time range quantifiers for the hybrid tracker:
If the search command is sensible to the time range quantifiers, fill as appropriated.
Some search commands such as "| rest" are not relying on the time range, therefore you can set a very short time range (earliest: -5m, latest: now)
Earliest (ex: -5m)
-5m
Latest (ex: now)
now
5. Test and Review:
Test the search tracker execution and the entities discovery:
Simulate the search
Cancel
Back
Hybrid tracker simulation finished successfully, verify the results before proceeding to its creation



Once executed, TrackMe immediately starts monitoring the global indexer cluster status, generates KPIs for the cluster, and alerts you when issues are detected.


splunk>enterprise
Apps
TrackMe
TrackMe - Data in motion tracking system
clusters
Monitored entities count by priority
Monitored entities count by state and priority
1
0
0
0
Refresh
Manage flex objects
Manage saved searches
Manage logical groups
Run TrackMe reports
You can filter entities in various ways, use pre built filter functions to easily manage common use cases. Click on the quick action icons to open the entity view, or manage its life cycle:
Mode
Minimal
Show disabled entities
False
Filter field
Entity name
Filter operator
like
Filter value
value to filter
Reset Filters
Refresh the table
a few seconds ago
Actions
Identifier/state (click: rename/right click: info)
Priority and monitoring state
Status and description
Metrics
Tracking information
Outliers
Entity
Priority
Monitoring
Status
Status description
Last metrics
Tracker runtime
Infrastructure (1 item)
splunk_clusteridxc1
medium
enabled
All peers are up and running
right click for pop up
10 Feb 2024 10:25
Showing 1 of 1 rows


Actions for entity: infrastructure:splunk_clusteridxc1 (alias: splunk_clusteridxc1)
Status: 1
Status description: All peers are up and running
monitored state: enabled
host state: green
latest flip time: 10 Feb 2024 10:25
latest flip state: green
Priority: medium
Description: Splunk Cluster status
Overview entity
Notables
Outliers anomaly detection
Status flipping
Status message
Audit changes
Select metric(s)
Stats mode
Chart mode
status
max
column
100.00 %
Metrics for this entity
20m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
Refresh
Smart Status
Acknowledge alert
Enable
Disable
Delete
Modify
Search & reports
Close





If a cluster peer goes down for any reason, the status of the TrackMe entity turns red and an alert is generated.





Let's now create a second Flex Object tracker using the library with the use case "splk_splunk_enterprise_cluster_peers" which tracks the peers' statuses individually.





Once executed, one TrackMe entity is created per cluster peer, TrackMe starts monitoring the peer status and begins collecting main KPIs of interest for the peers.



Notably, TrackMe monitors the distribution of the buckets within the members and alerts you when the distribution is not properly balanced between the members.





You now have a fully monitored Indexer Cluster, enabling TrackMe to continuously monitor the health of your Splunk Indexer Clusters, track SLA availability, and easily alert you when issues are detected on your Splunk indexers cluster(s) infrastructure.

8.7.4 Annex: Use cases definition (SPL)

Use case splk_splunk_enterprise_cluster_status

```
| rest splunk_server=local count=0 /services/cluster/master/status
| transpose
| rename column as field, "row 1" as value
| where value!=""
| eval combo = field . ":" . mvjoin(value, ",")
| fields combo
```

(continues on next page)

(continued from previous page)

```
| rex field=combo "\.status:(?<peer_status>.*)"
| where isnotnull(peer_status)
| stats count as peers_count, count(eval(peer_status="Up")) as peers_up,␣
↪count(eval(peer_status="Stopped")) as peers_down, count(eval(peer_status!="Up" AND␣
↪peer_status!="Stopped")) as peers_unknown
| eval pct_cluster_availability=round(peers_up/peers_count*100, 2)
| appendcols [ | rest splunk_server=local /services/cluster/config/config | table␣
↪cluster_label ] | filldown cluster_label
| fields cluster_label, pct*, *
| eval group="infrastructure"
| eval object = "splunk_cluster" . ":" . cluster_label
| eval object_description = "Splunk Cluster status"
| eval status=case(pct_cluster_availability>=99, 1, pct_cluster_availability<99, 2,␣
↪1=1, 3)
| eval status_description=case(status=1, "All peers are up and running",status=2,
↪"Some peers are not Up, the cluster is not healthy", status=3, "The cluster status␣
↪is unknown or not expected")
| eval metrics = "{\"splunk_enterprise.cluster.pct_cluster_availability\": " . pct_
↪cluster_availability . ", \"splunk_enterprise.cluster.peers_count\": " . peers_
↪count . ", \"splunk_enterprise.cluster.peers_up\": " . peers_up . ", \"splunk_
↪enterprise.cluster.peers_down\": " . peers_down . ", \"splunk_enterprise.cluster.
↪peers_unknown\": " . peers_unknown . "}"

``` alert if inactive for more than 3600 sec```
| eval max_sec_inactive=3600
```

Use case 'splk_splunk_enterprise_cluster_peers'

```
| rest splunk_server=local count=0 /services/cluster/master/peers
| fields label site bucket_count is_searchable primary_count

``` aggreg ```
| eventstats sum(bucket_count) as total_bucket_count, dc(label) as dcount_peers

``` calculate percentage of buckets ```
| eval pct_buckets = round(bucket_count/total_bucket_count*100, 2), pct_primary_
↪buckets = round(primary_count/total_bucket_count*100, 2), pct_ideal_
↪buckets=round(100/dcount_peers, 2)
| eval pct_buckets_unbalanced_deviation = round(abs(pct_buckets - pct_ideal_buckets)/
↪pct_ideal_buckets*100, 2)

``` set the cluster_label automatically ```
| appendcols [ | rest splunk_server=local /services/cluster/config/config | table␣
↪cluster_label ] | filldown cluster_label

``` set group ```
| eval group="infrastructure"

``` set object, alias and object_description ```
| eval object = "splunk_cluster" . ":" . cluster_label . ":" . label, alias=object
| eval object_description = "Splunk Cluster indexer peer status"

``` set status, if the peer is not searchable, then it is red ```
| eval status=case(is_searchable=1, 1, is_searchable!=1, 2, 1=1, 3)

``` if the peer buckets deviation percentage is higher than 25%, then the peer is␣
```

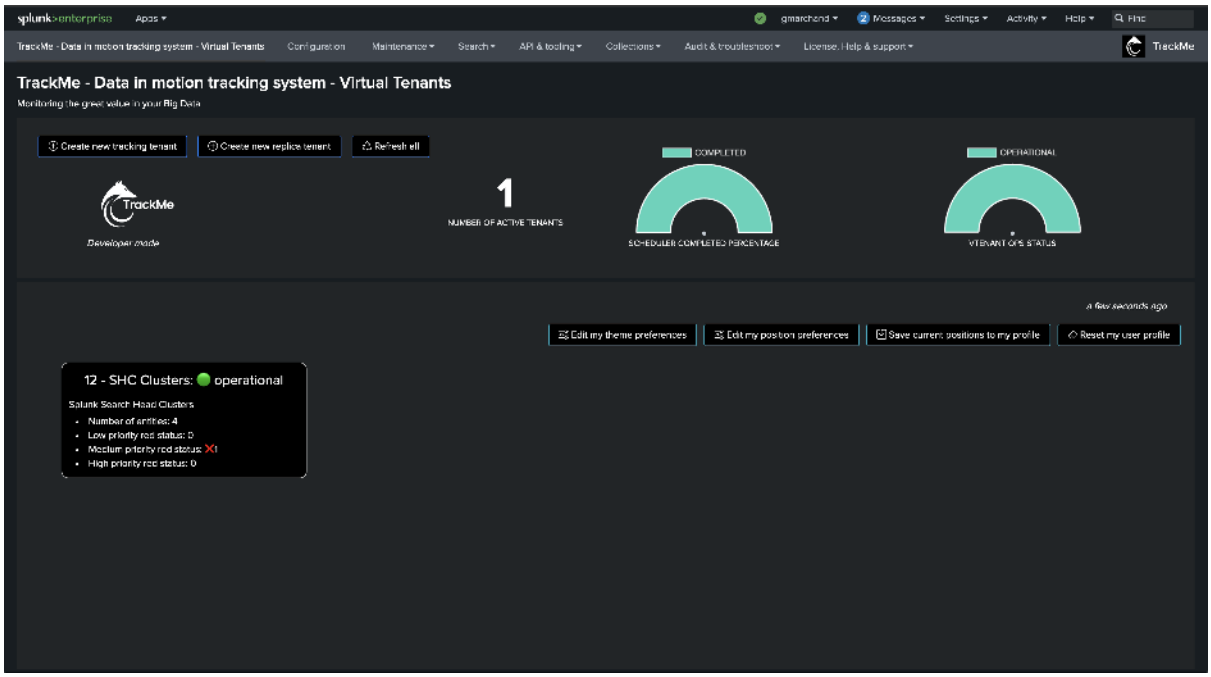
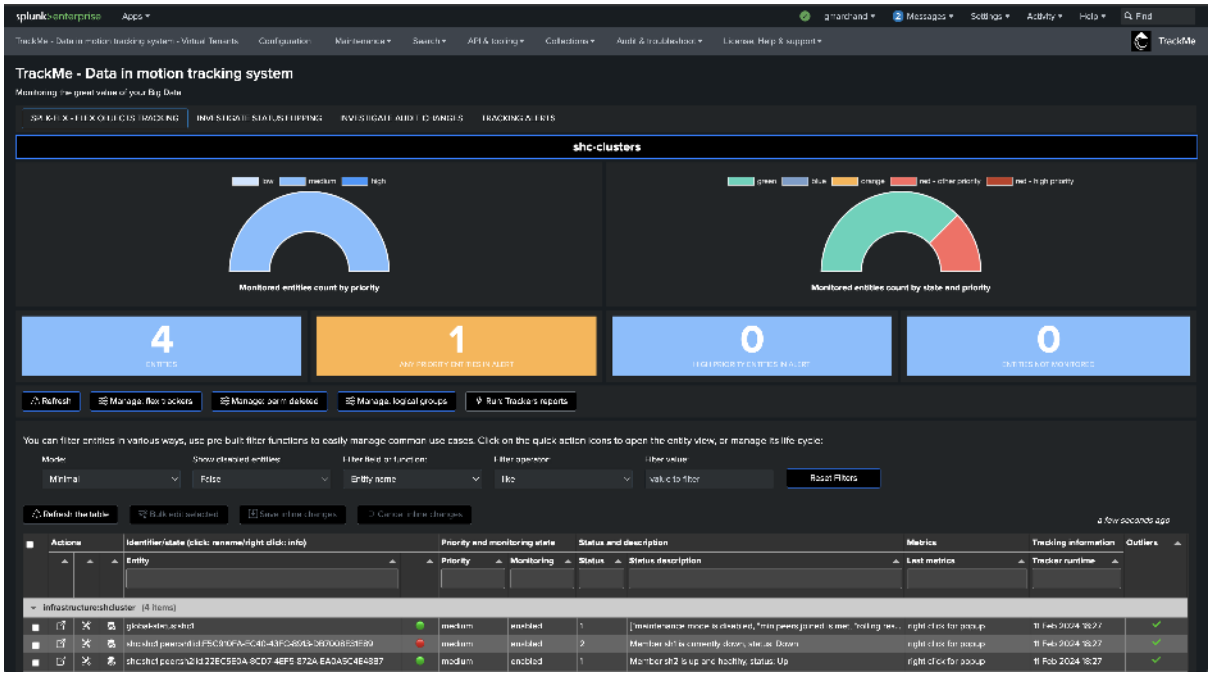
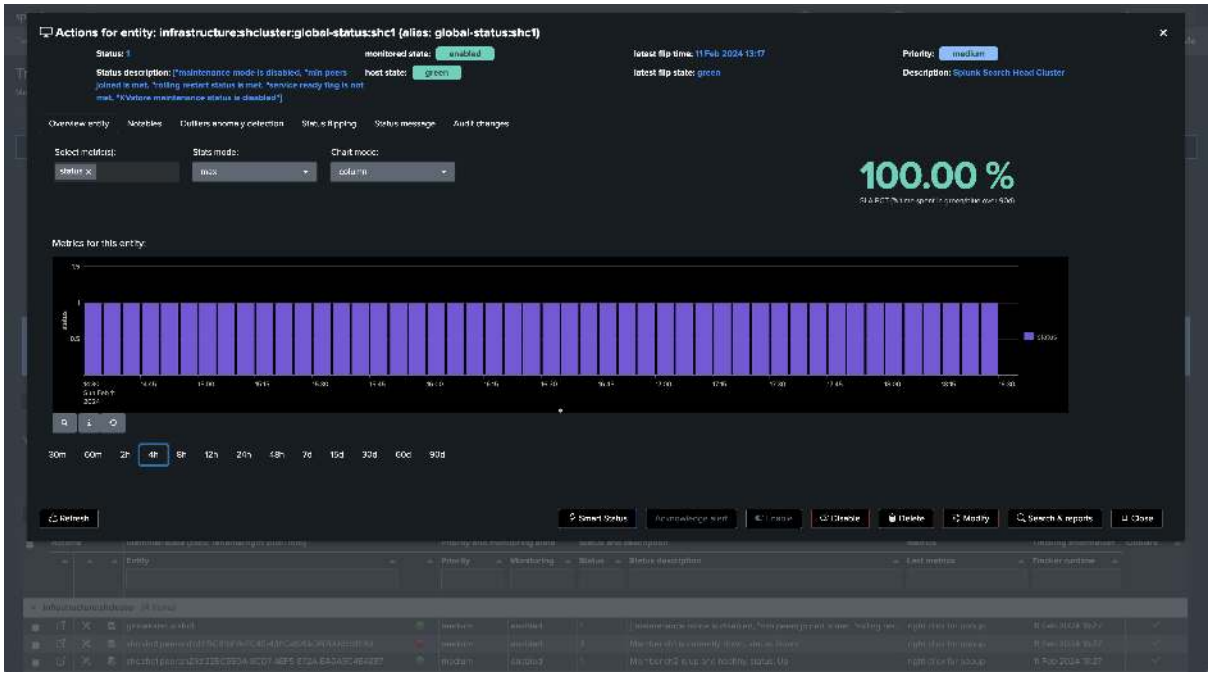
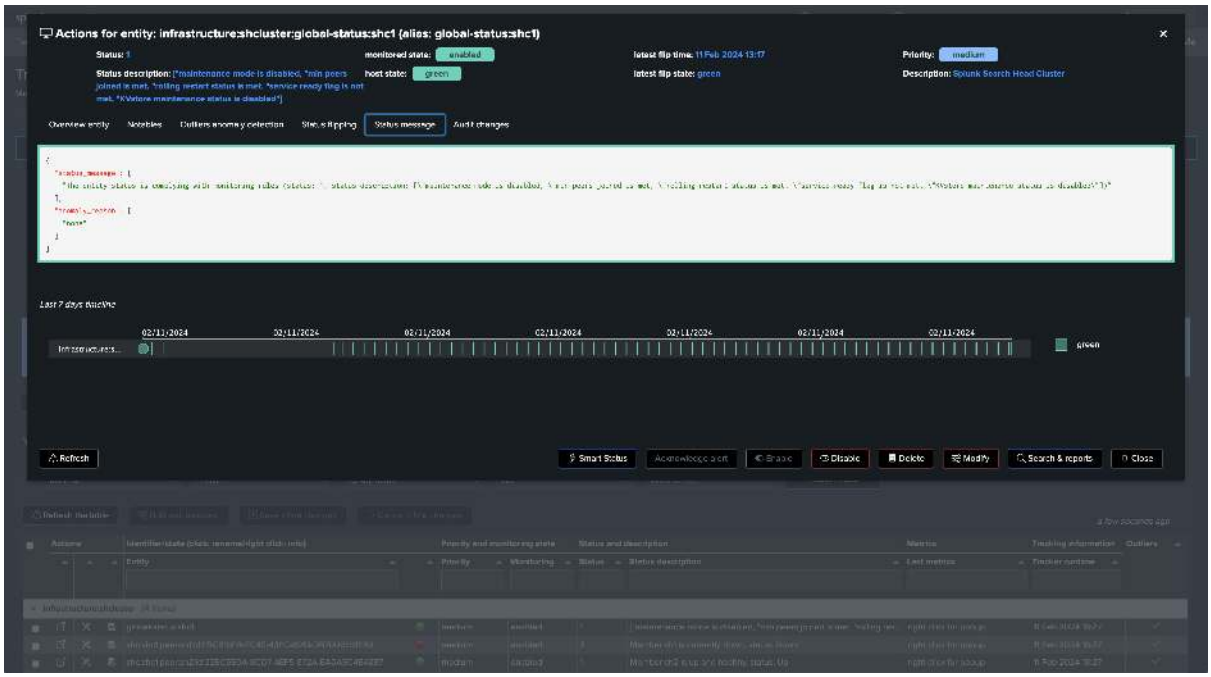
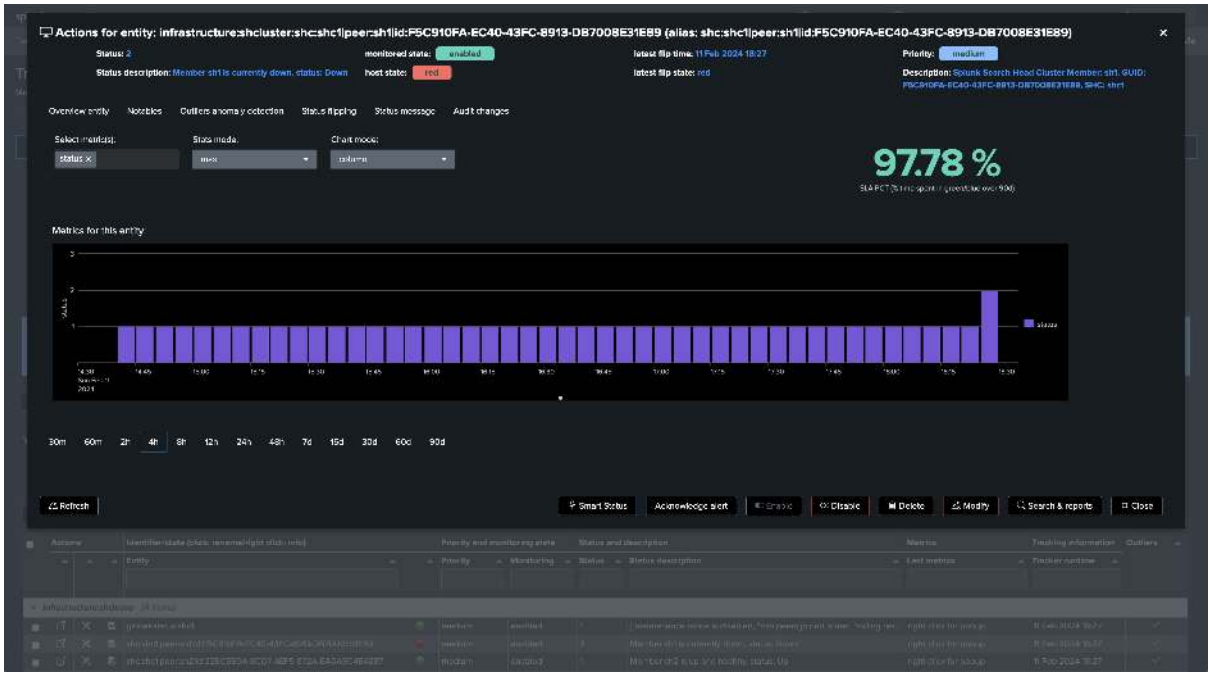
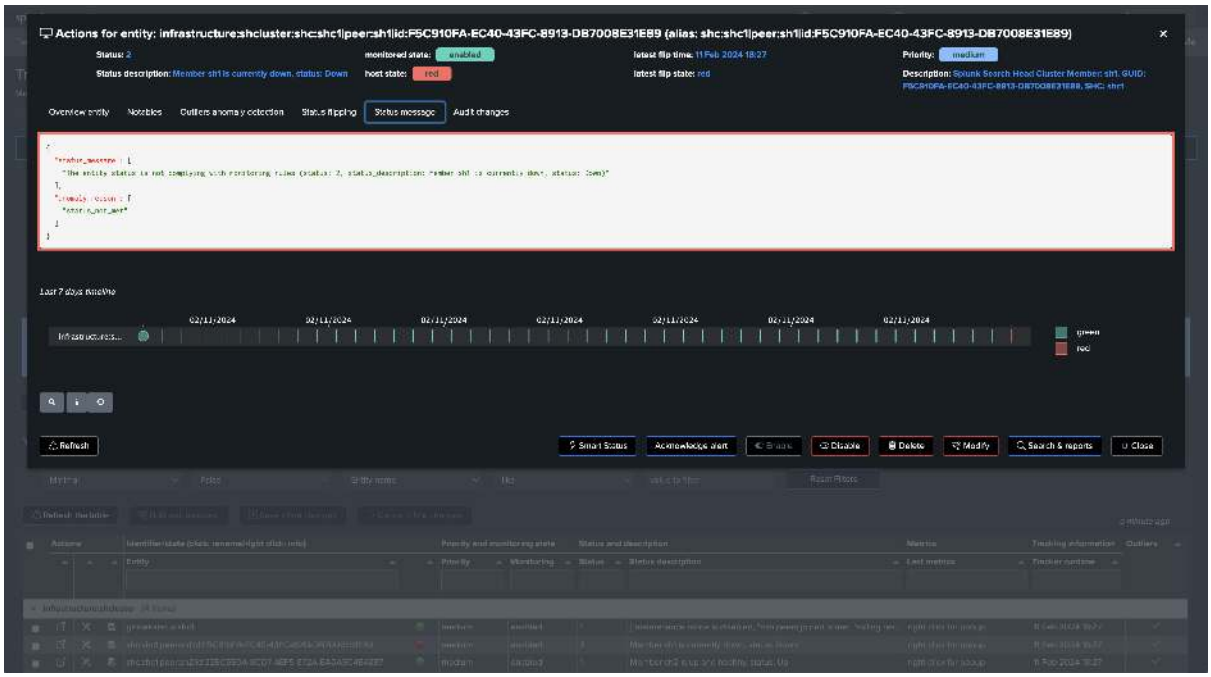
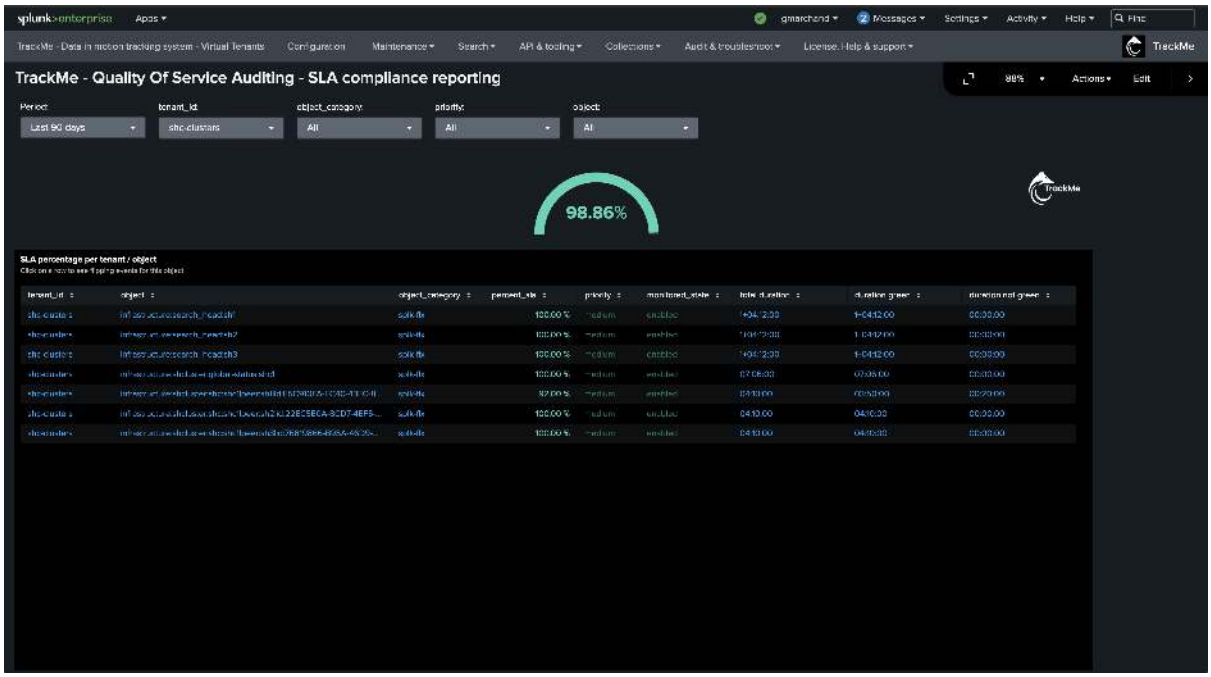
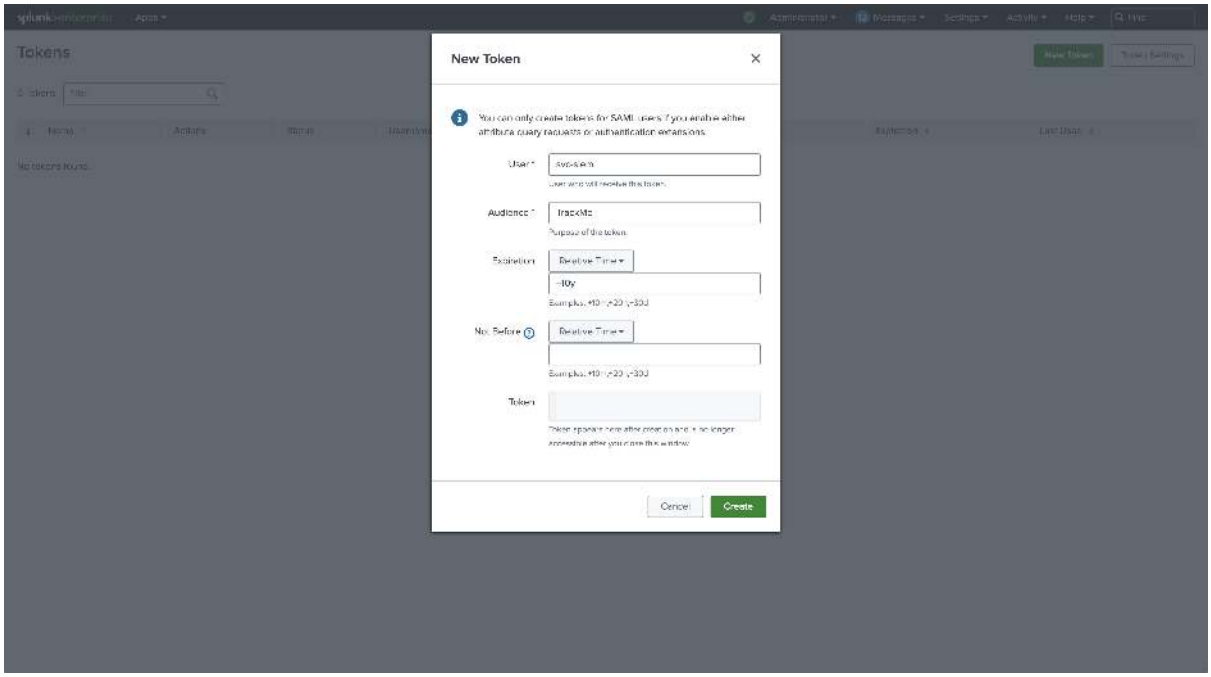
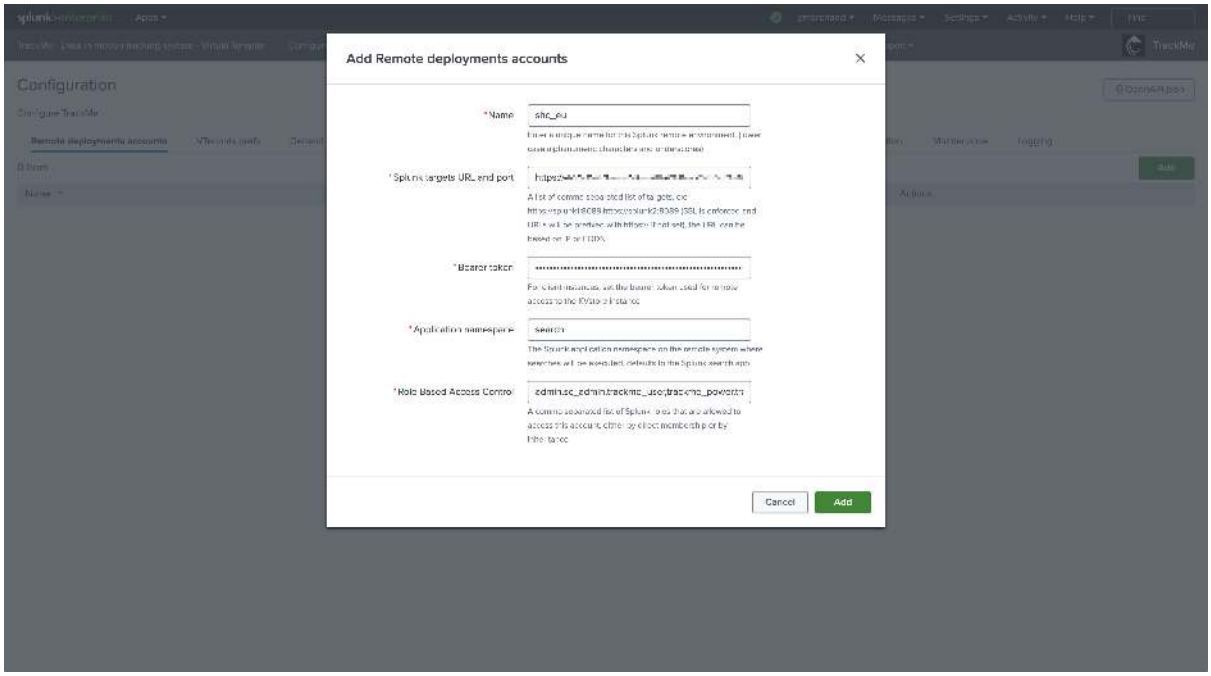
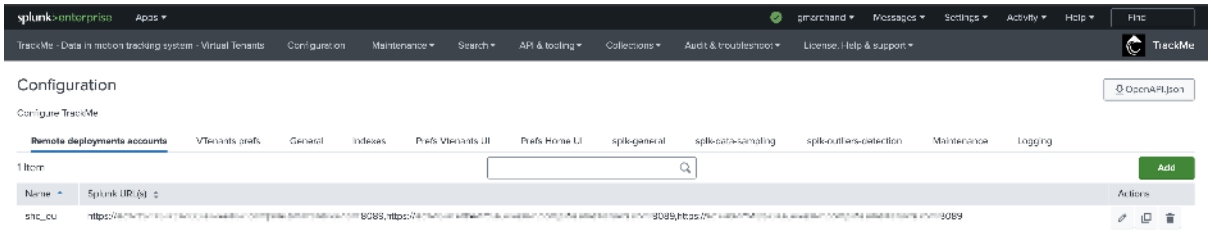
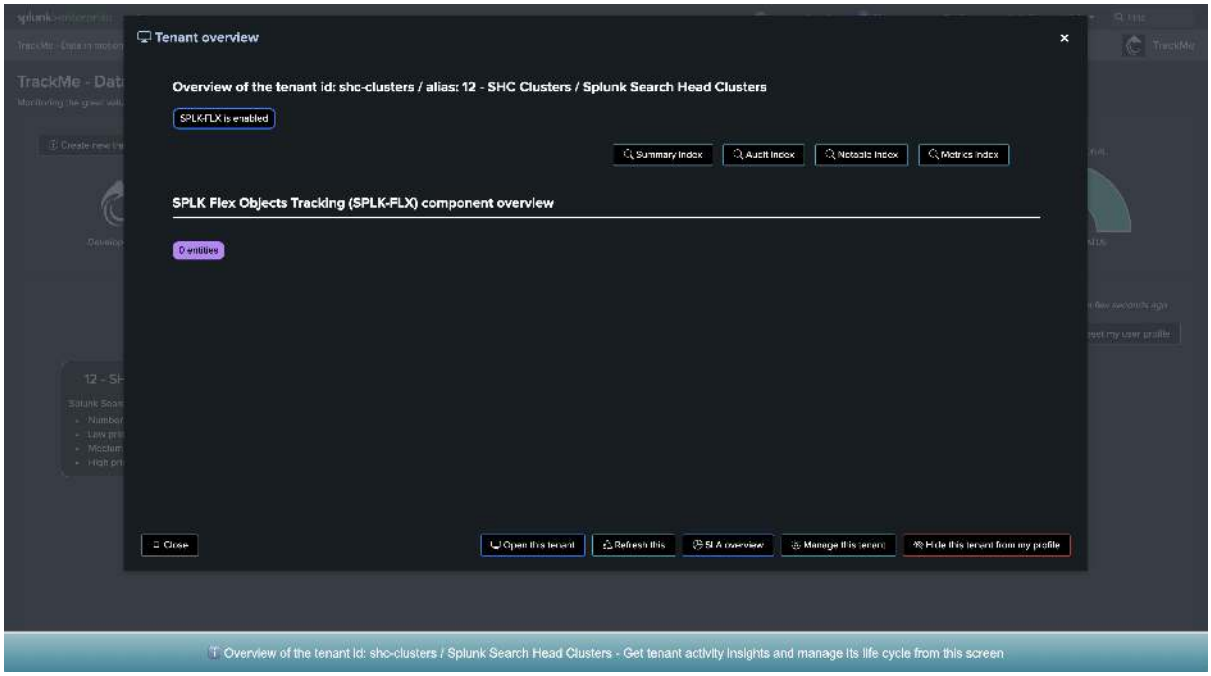
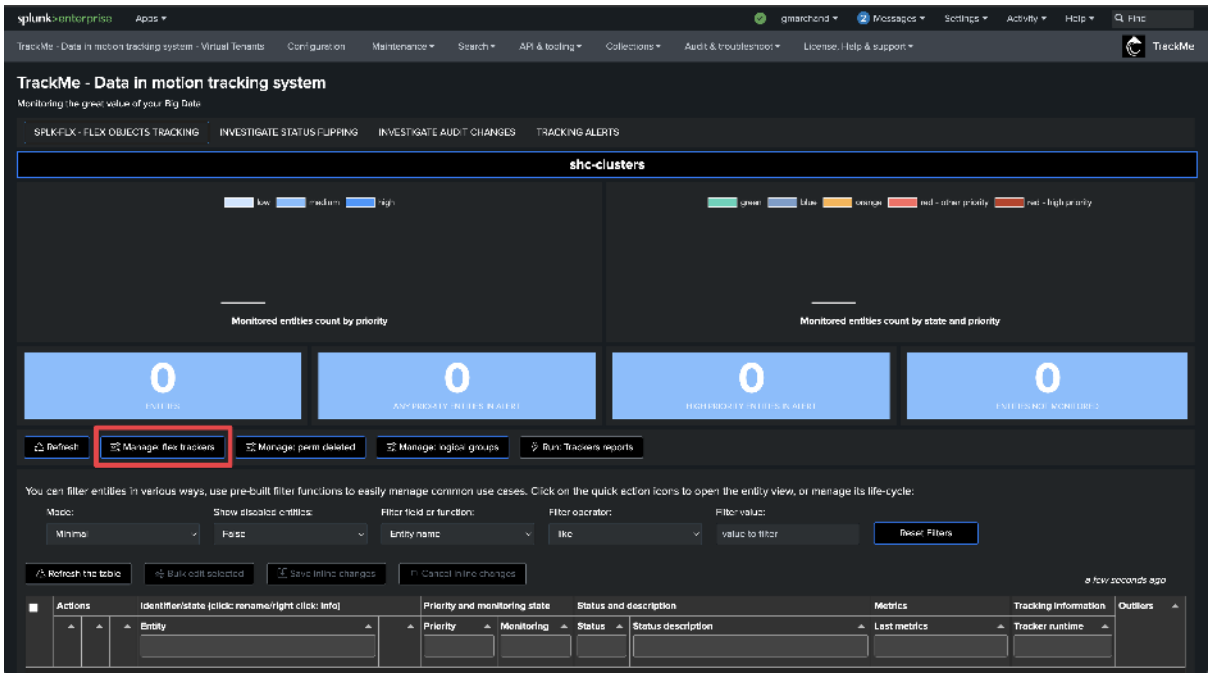
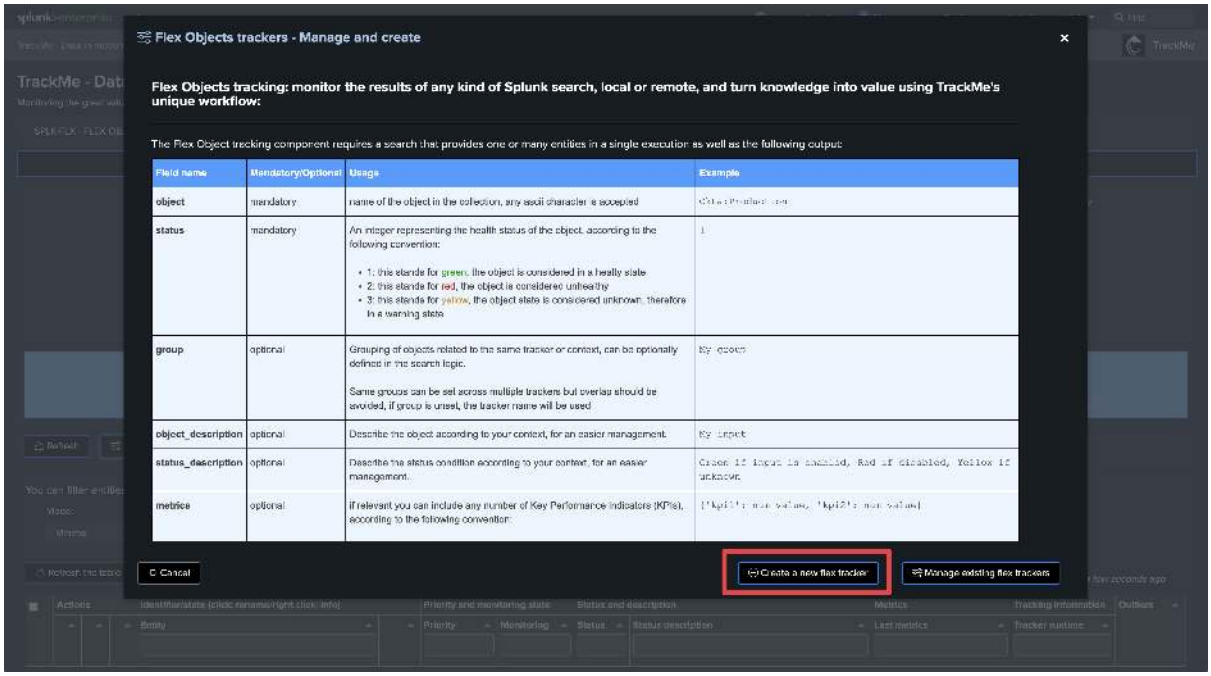
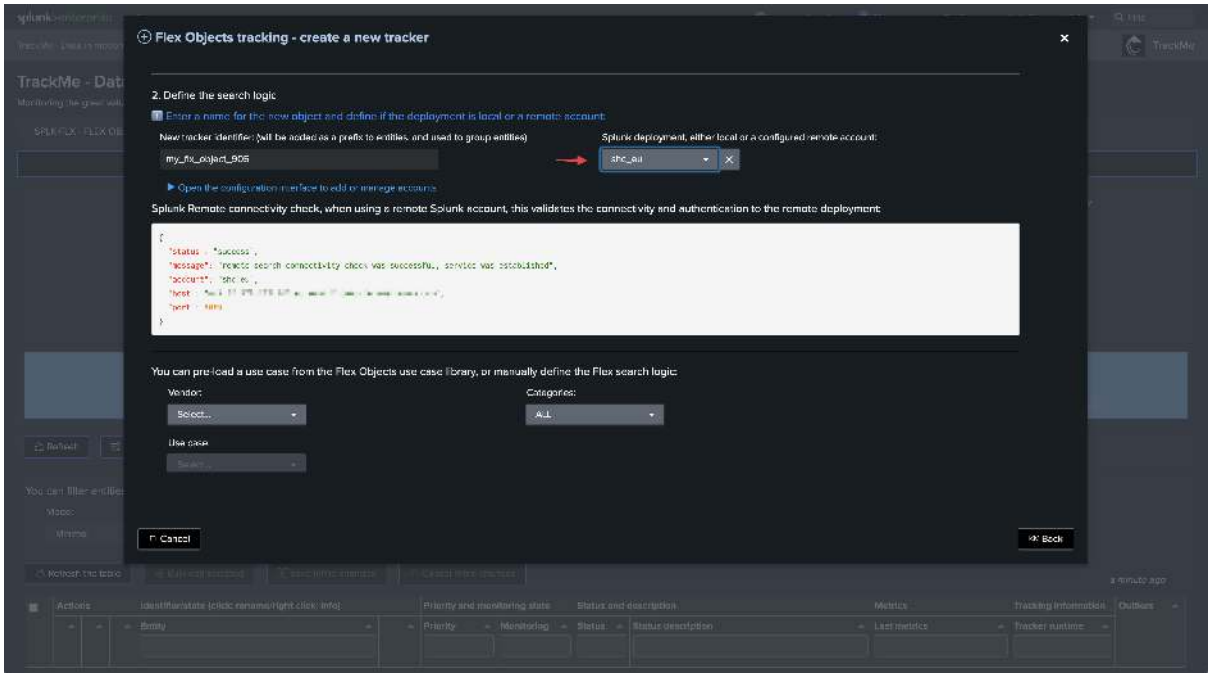
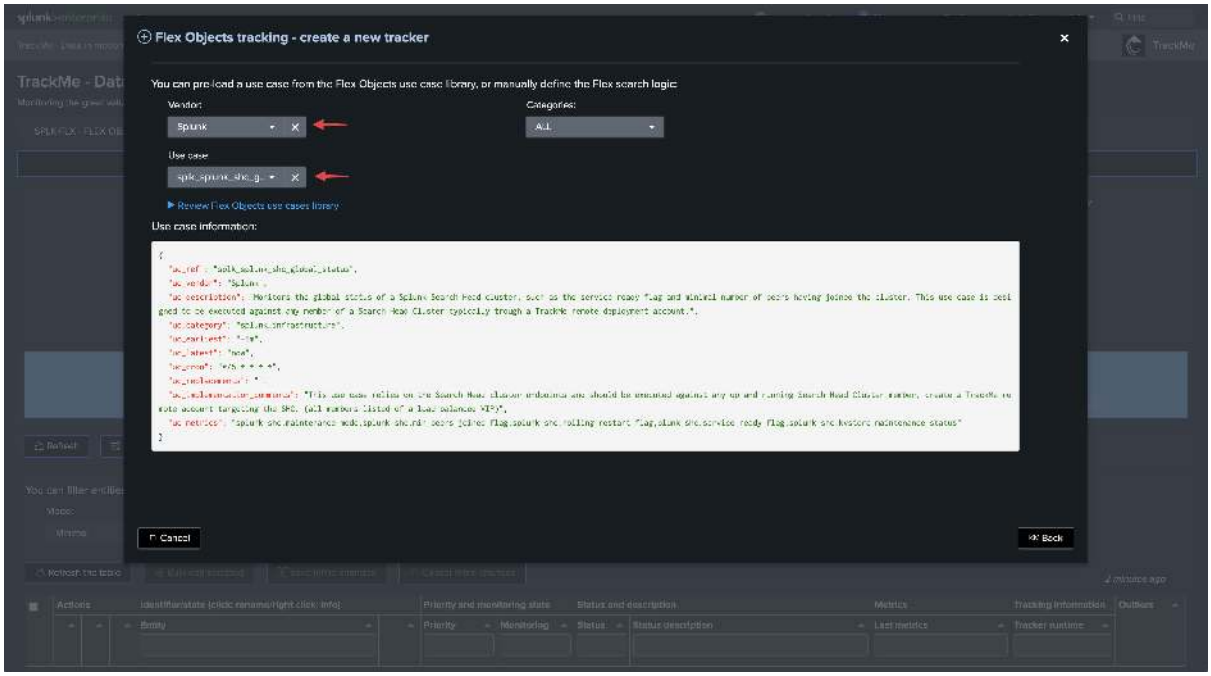
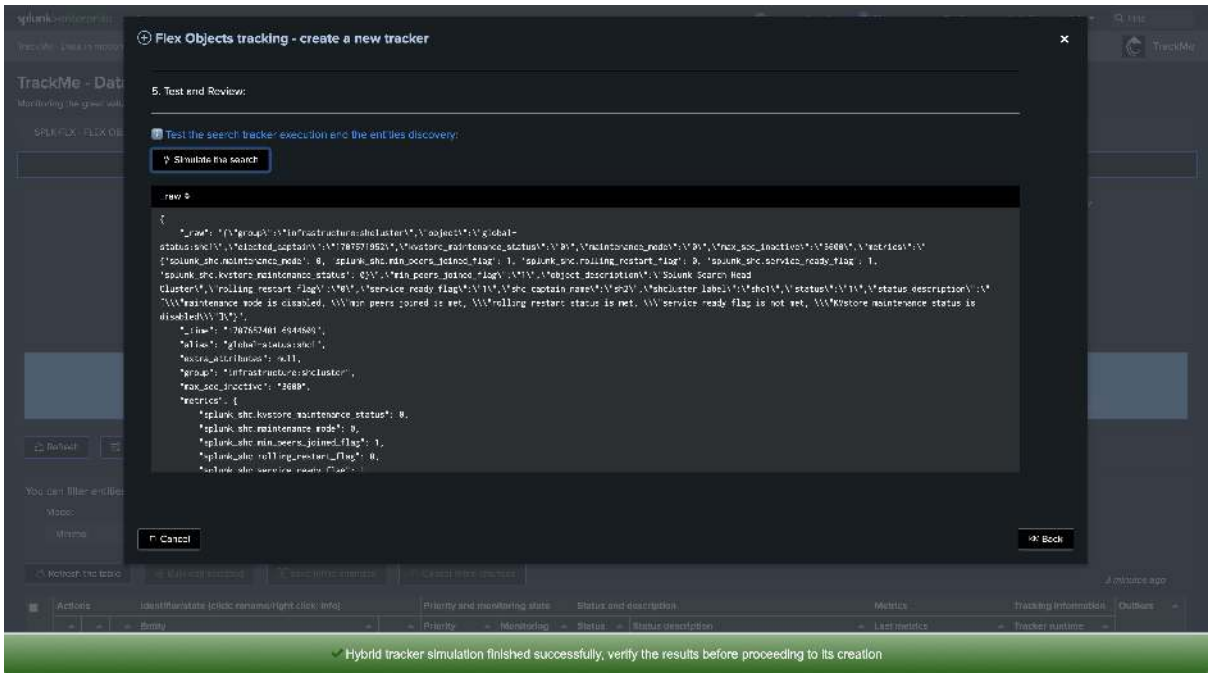
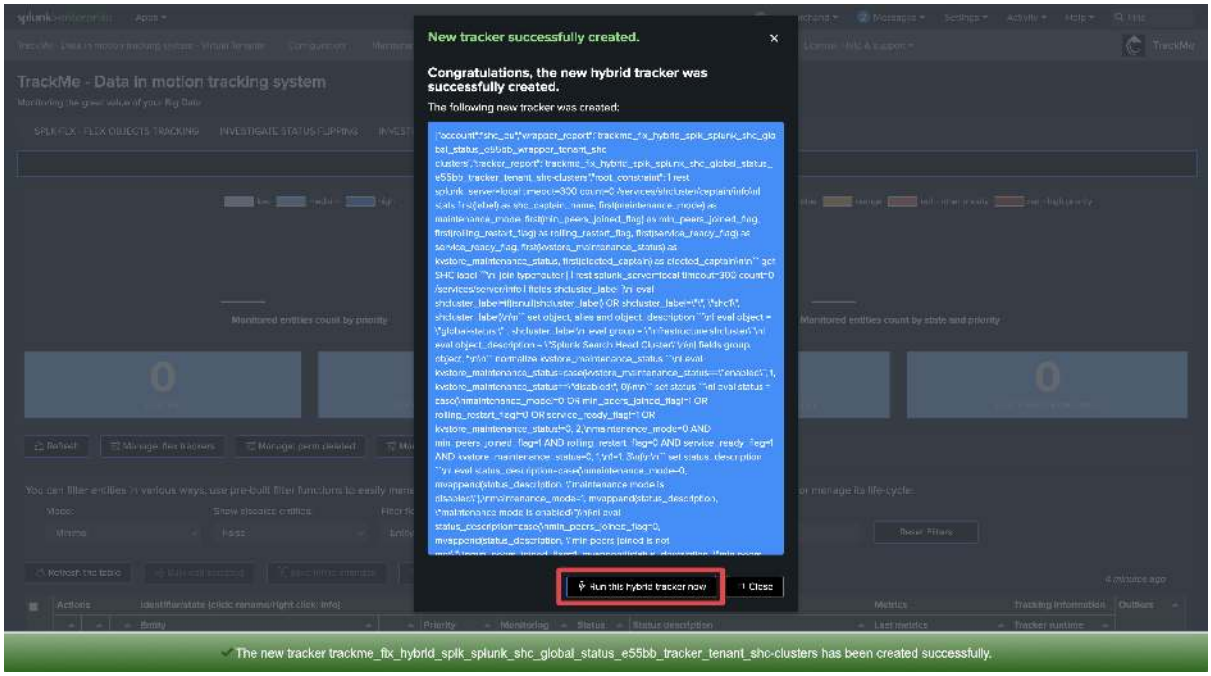
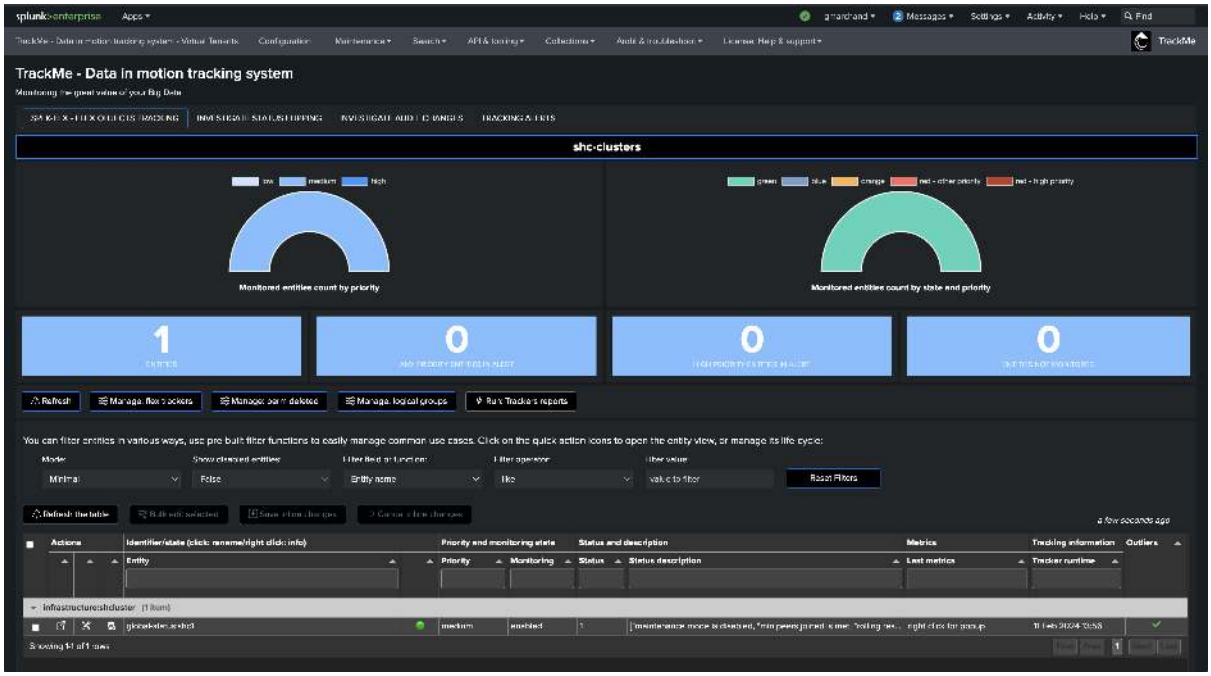
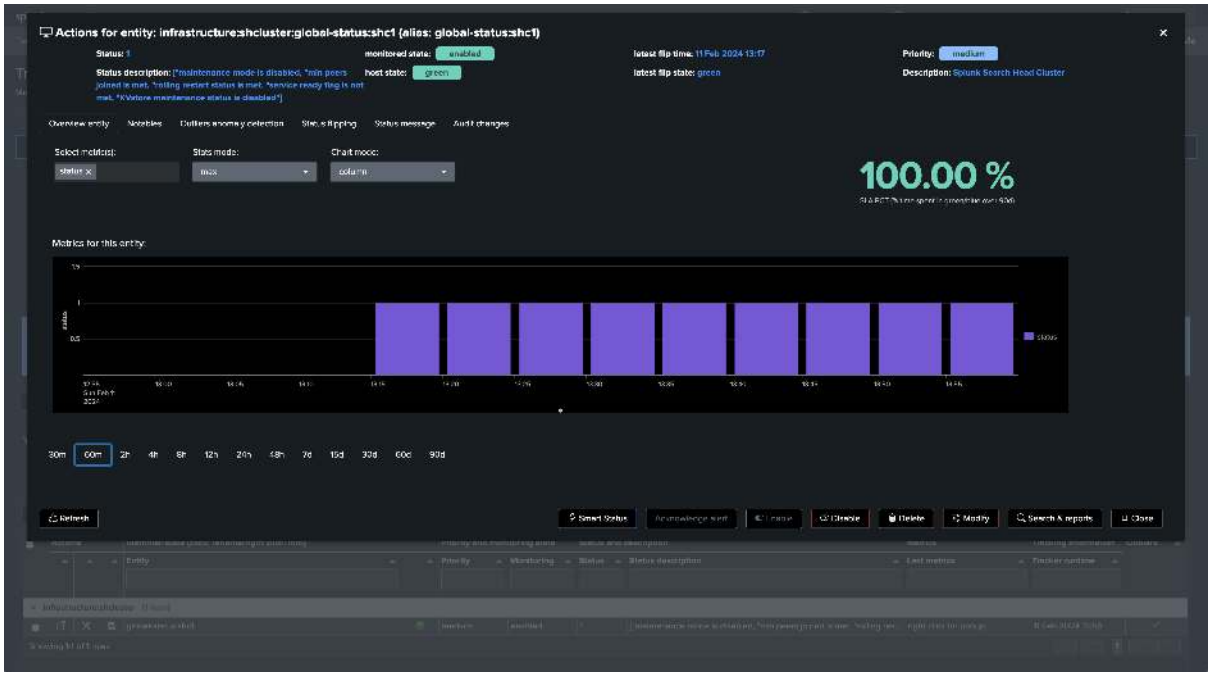
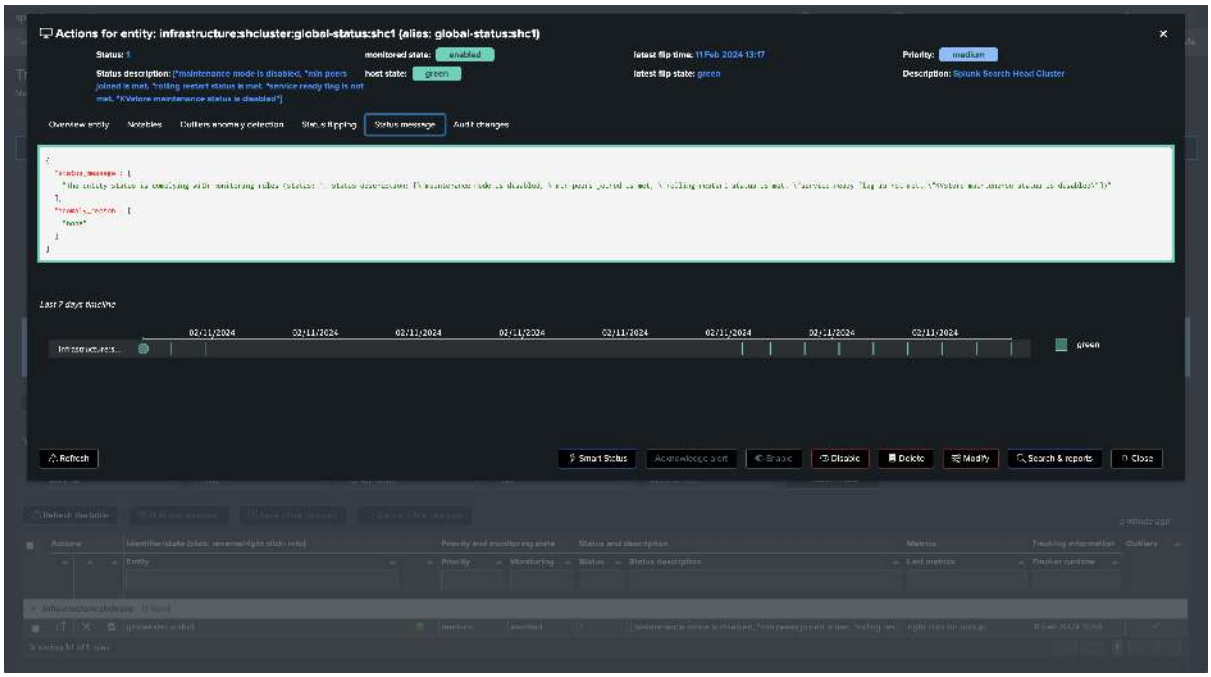
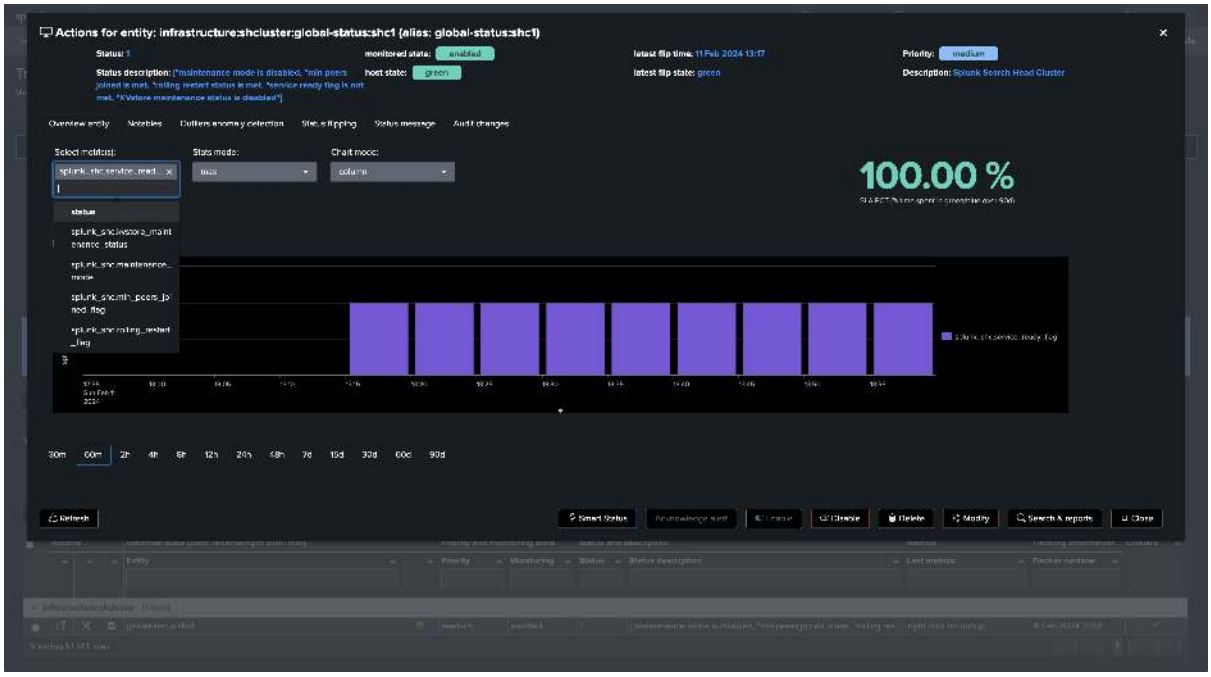
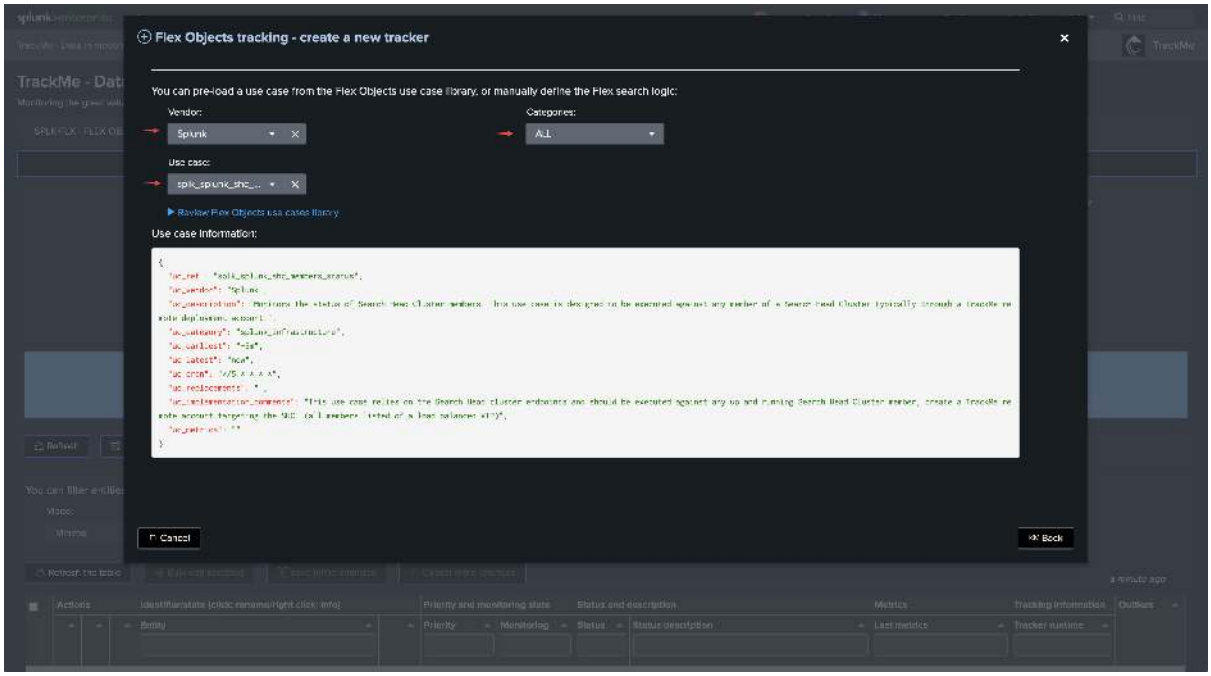
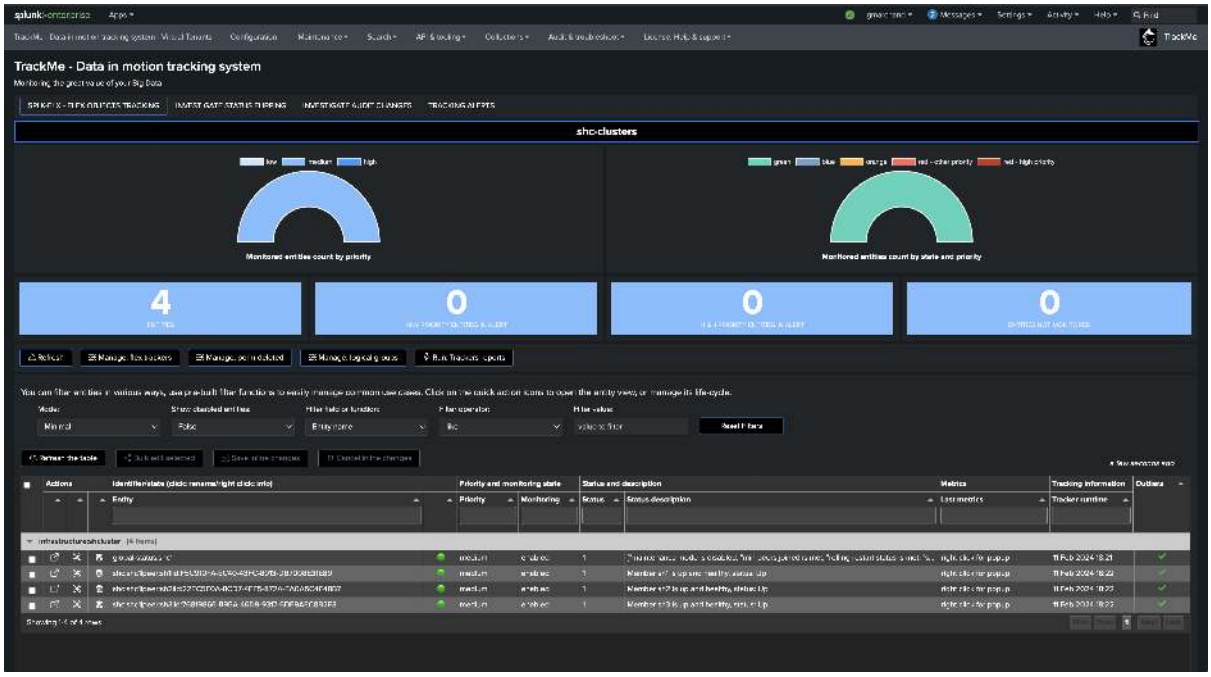
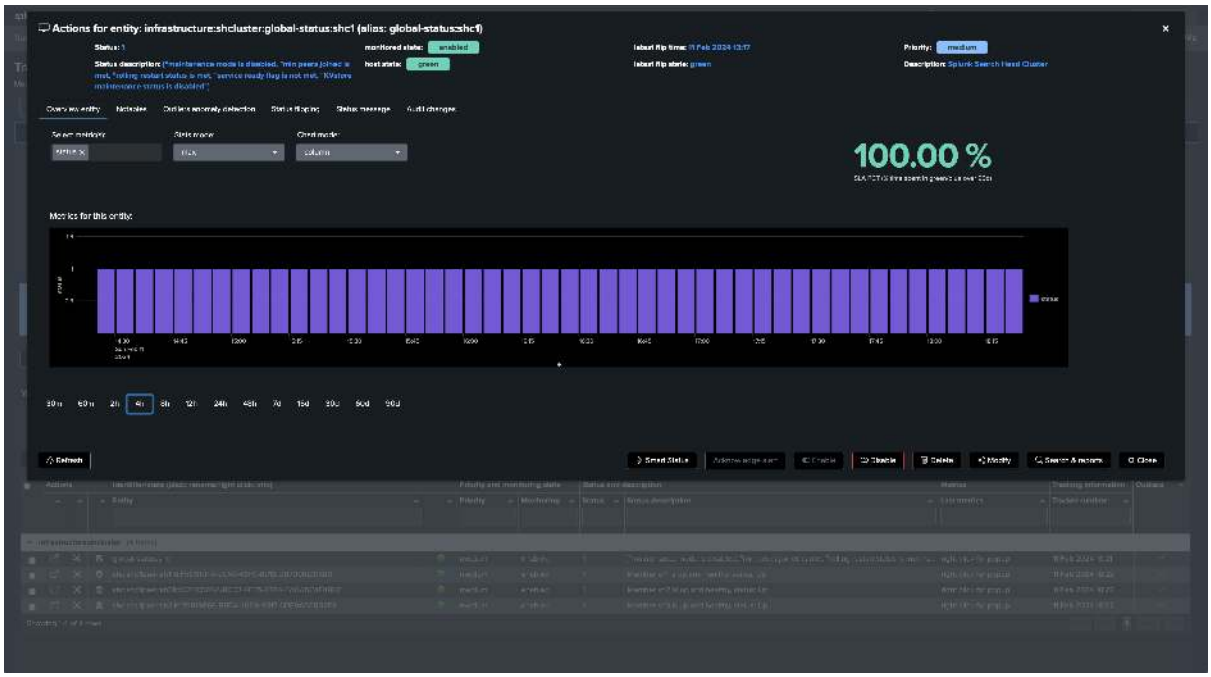
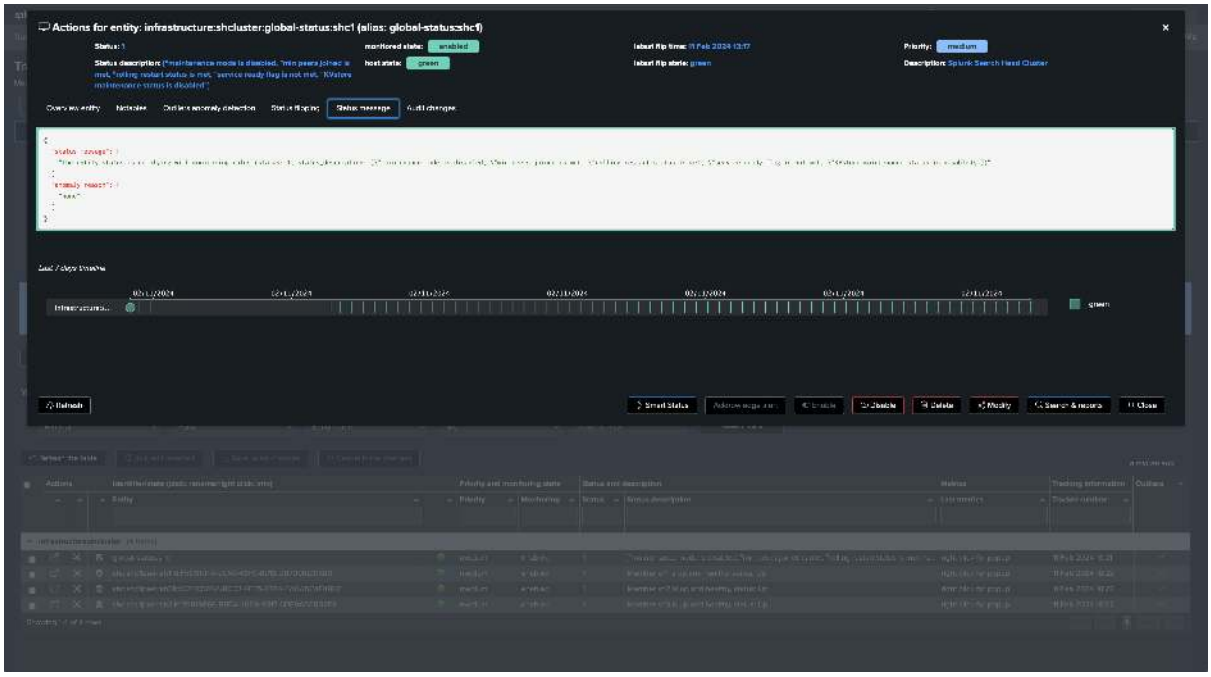
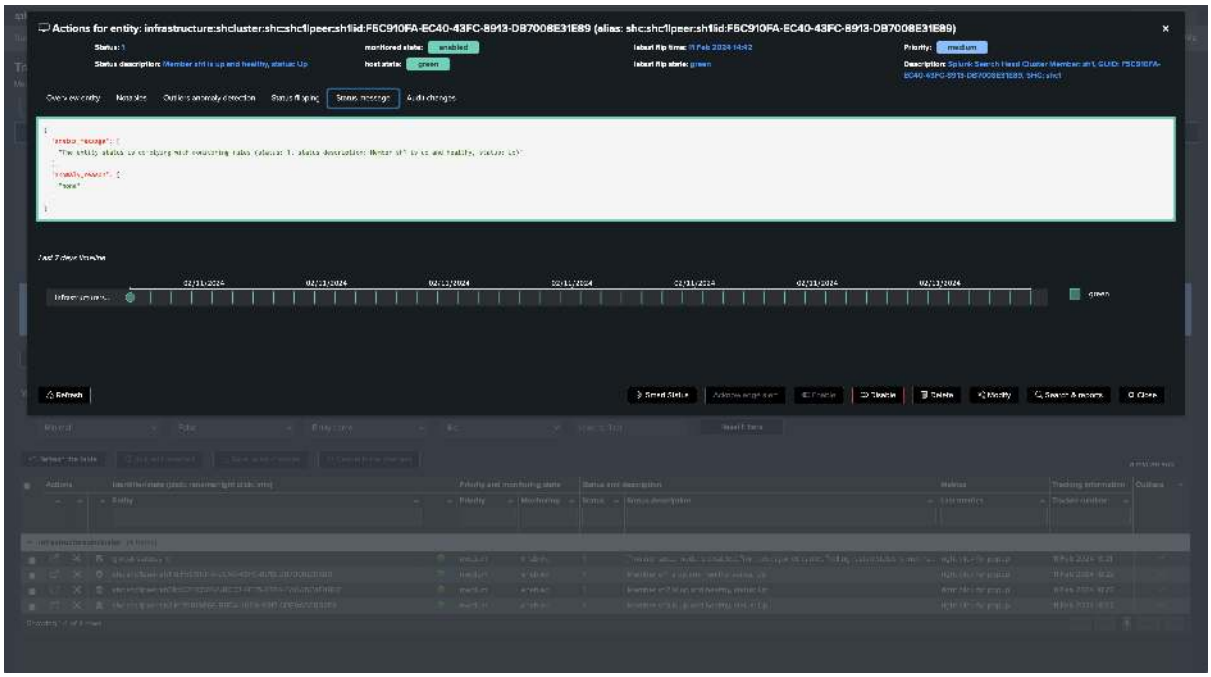
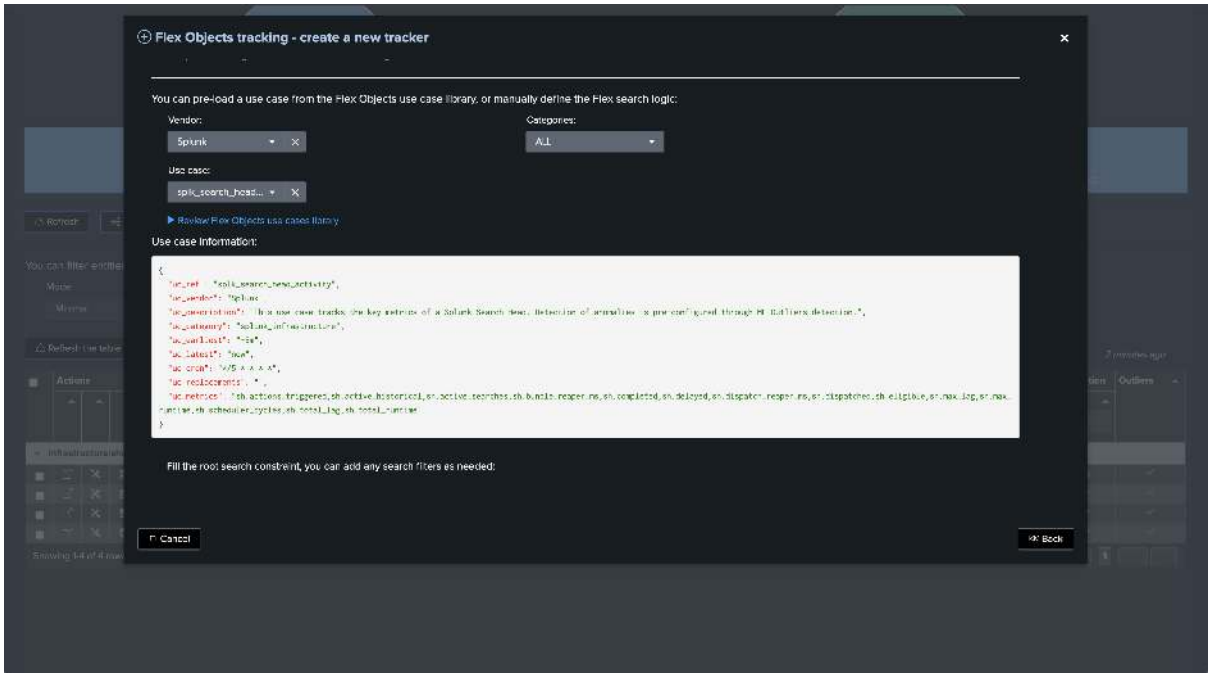
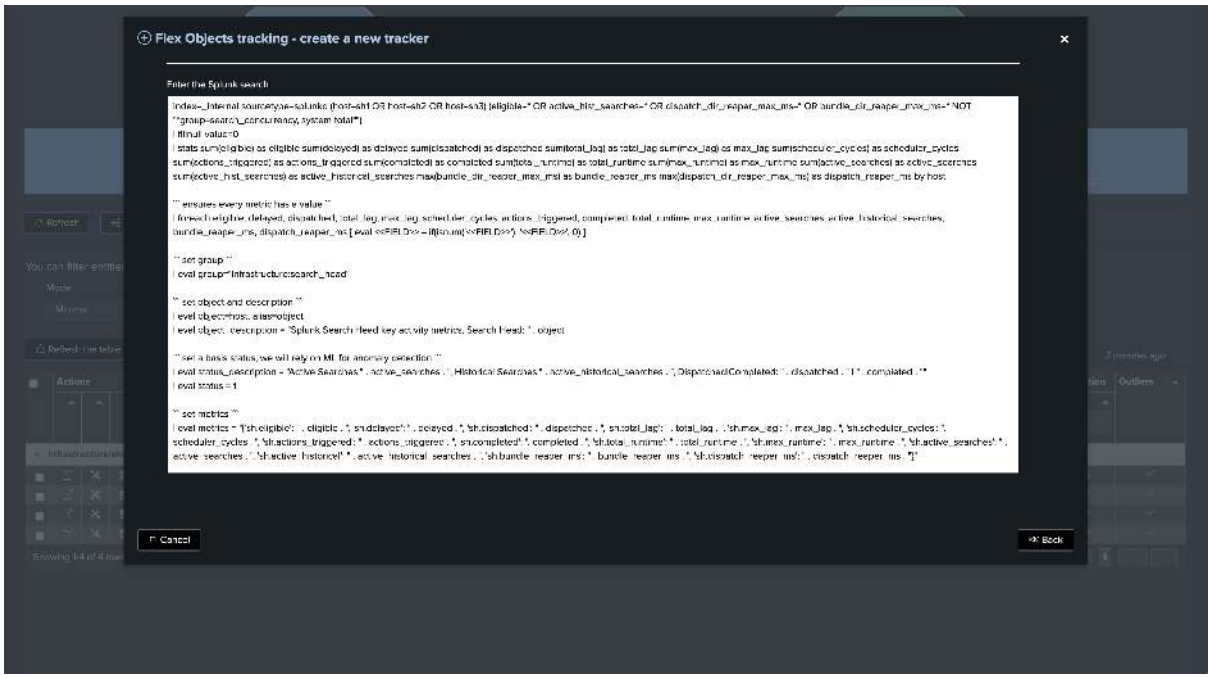
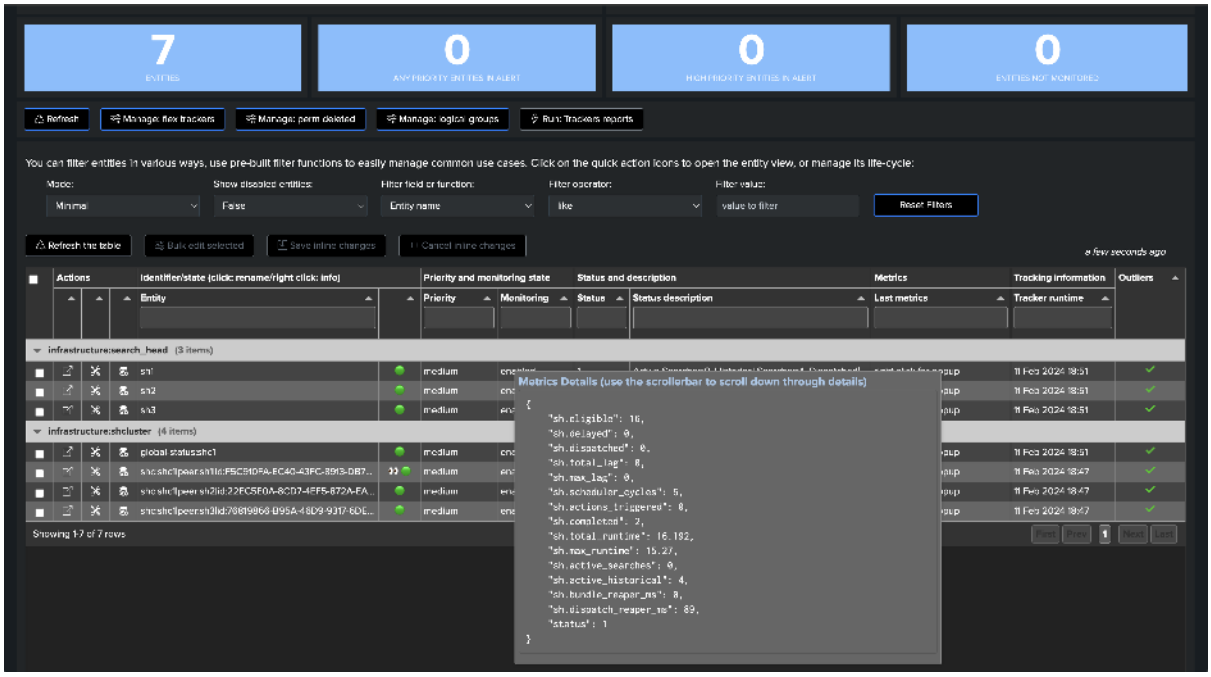
(continues on next page)

(continued from previous page)

```
↪highly inbalanced, we set the status to yellow, but you can also set this to red if␣
↪you prefer ```
| eval peer_is_balanced=if(pct_buckets_unbalanced_deviation>25, 0, 1)
``` yellow if breached, set to red if preferred ```
| eval status=if(peer_is_balanced!=1, 3, status)

``` join external CPU metrics ```
| join type=outer label [ search index=_introspection sourcetype=splunk_resource_
↪usage component=Hostwide
| eval cpu_total_pct=('data.cpu_system_pct'+'data.cpu_user_pct'), mem_used_pct=('data.
↪mem_used'/'data.mem'*100)
| stats avg(cpu_total_pct) as avg_cpu_pct, avg(mem_used_pct) as mem_used_pct by host
| eval avg_cpu_pct=round(avg_cpu_pct, 2), mem_used_pct=round(mem_used_pct, 2) |␣
↪rename host as label ]
| eval avg_cpu_pct=if(isnum(avg_cpu_pct), avg_cpu_pct, 0), mem_used_pct=if(isnum(mem_
↪used_pct), mem_used_pct, 0)

``` set status_description ```
| eval status_description=case(
status=1 AND is_searchable=1 AND pct_buckets_unbalanced_deviation<=25, "The cluster␣
↪peer status is searchable and properly balanced",
status=2 AND is_searchable!=1, "The cluster peer status is not searchable",
(status=2 OR status=3) AND peer_is_balanced!=1, "The cluster peer bucket count is␣
↪inbalanced and breaching the max threshold, pct_buckets_unbalanced_deviation: " .␣
↪pct_buckets_unbalanced_deviation . " %, pct_buckets: " . pct_buckets . " %",
status=3, "The cluster peer status is unknown or not expected"
)

``` set metrics ```
| eval metrics = "{\"splunk_enterprise.cluster_peer.pct_buckets\": " . pct_buckets .
↪", \"splunk_enterprise.cluster_peer.pct_primary_buckets\": " . pct_primary_buckets .
↪ ", \"splunk_enterprise.cluster_peer.bucket_count\": " . bucket_count . ", \"splunk_
↪enterprise.cluster_peer.primary_count\": " . primary_count . ", \"splunk_enterprise.
↪cluster_peer.pct_buckets_unbalanced_deviation\":" . pct_buckets_unbalanced_
↪deviation . ", \"splunk_enterprise.system.avg_cpu_pct\": " . avg_cpu_pct  . ", \
↪"splunk_enterprise.system.mem_used_pct\": " . mem_used_pct . "}"

``` alert if inactive for more than 3600 sec```
| eval max_sec_inactive=3600
```

8.8 Monitor Splunk Search Head Clusters

Monitoring Splunk Search Head Clusters with TrackMe Flex Objects

- This tutorial demonstrates the monitoring of **Splunk Search Head Clusters** with TrackMe Flex Objects.
- TrackMe Flex Objects is a component restricted to licensed customers. Please contact Splunk Sales for more information.
- Using these steps will enable TrackMe to continuously monitor the health of your Splunk Search Head Clusters and alert you when issues are detected.
- With TrackMe's remote search capabilities, you can monitor as many Search Head Clusters as you need from a single pane of glass.
```

The following screen shows the final results in TrackMe, starting from Step 1 to easily implement the monitoring of your Splunk Search Head Clusters.






Actions for entity: infrastructure:shcluster:global-status:shc1 (alias: global-status:shc1)
Status: 1
monitored state: enabled
latest flip time: 11 Feb 2024 13:17
Priority: medium
Status description: ["maintenance mode is disabled, *min peers joined is met, *rolling restart status is met, *service ready flag is set, *KVstore maintenance status is disabled"]
host state: green
latest flip state: green
Description: Splunk Search Head Cluster
Overview entity
Notables
Outliers anomaly detection
Status flipping
Status message
Audit changes
Select metric(s):
Stats mode:
max
Chart mode:
column
100.00 %
Metrics for this entity:
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
Refresh
Smart Status
Acknowledge alert
Enable
Disable
Delete
Modify
Search & reports
Close


Actions for entity: infrastructure:shcluster:global-status:shc1 (alias: global-status:shc1)
Status: 1
monitored state: enabled
latest flip time: 11 Feb 2024 13:17
Priority: medium
Status description: ["maintenance mode is disabled, *min peers joined is met, *rolling restart status is met, *service ready flag is set, *KVstore maintenance status is disabled"]
host state: green
latest flip state: green
Description: Splunk Search Head Cluster
Overview entity
Notables
Outliers anomaly detection
Status flipping
Status message
Audit changes
Last 7 days timeline
02/11/2024
green
Refresh
Smart Status
Acknowledge alert
Enable
Disable
Delete
Modify
Search & reports
Close


Actions for entity: infrastructure:shcluster:shc:shc1|peer:sh1|id:F5C910FA-EC40-43FC-8913-DB7008E31E89 (alias: shc:shc1|peer:sh1|id:F5C910FA-EC40-43FC-8913-DB7008E31E89)
Status: 2
monitored state: enabled
latest flip time: 11 Feb 2024 18:27
Priority: medium
Status description: Member sh1 is currently down, status: Down
host state: red
latest flip state: red
Description: Splunk Search Head Cluster Member: sh1, GUID: F5C910FA-EC40-43FC-8913-DB7008E31E89, SHC: shc1
Overview entity
Notables
Outliers anomaly detection
Status flipping
Status message
Audit changes
Select metric(s):
Stats mode:
Chart mode:
Status
max
column
97.78 %
Metrics for this entity
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
Refresh
Smart Status
Acknowledge alert
Disable
Delete
Modify
Search & reports
Close


Actions for entity: infrastructure:shcluster:shc:shc1|peer:sh1|id:F5C910FA-EC40-43FC-8913-DB7008E31E89 (alias: shc:shc1|peer:sh1|id:F5C910FA-EC40-43FC-8913-DB7008E31E89)
Status: 2
monitored state: enabled
latest flip time: 11 Feb 2024 18:27
Priority: medium
Status description: Member sh1 is currently down, status: Down
host state: red
latest flip state: red
Description: Splunk Search Head Cluster Member: sh1, GUID: F5C910FA-EC40-43FC-8913-DB7008E31E89, SHC: shc1
Overview entity
Notables
Outliers anomaly detection
Status flipping
Status message
Audit changes
Last 7 days timeline
02/11/2024
green
red
Refresh
Smart Status
Acknowledge alert
Disable
Delete
Modify
Search & reports
Close



8.8.1 Step 1: Create a Splunk Remote Deployment Account

Hint

Interacting with Splunk API through TrackMe's remote search capabilities

- This use case relies on a Splunk / *rest* command which is a Splunk REST API call through native SPL.
- Therefore, this command needs to be executed on a remote Splunk target, in the context of monitoring Search Head Clusters, the command can be executed on any up and running SHC member.
- Using TrackMe's remote search capabilities, you can transparently execute the command on a remote target, and process results locally on the TrackMe guest.

The first step is to create a Splunk Remote Deployment Account for the Cluster Manager. For more information about TrackMe Remote Search capabilities and configurations:

Splunk Remote Deployments (splunkremotesearch)

On the Search Head Cluster, create a new Splunk bearer token for the TrackMe Remote Deployment Account:



In TrackMe, click on Configure / Remote Deployment Accounts and add a new account:

- You can specify each of the SHC members in a comma separated list or use a load balancer URL.
- If multiple endpoints are specified, TrackMe automatically dispatches searches randomly amongst available and responding members (it validates connectivity and authentication)





Hint

Managing multiple Search Head Clusters

- If you have multiple Search Head Clusters, you can create a Remote Deployment Account for each SHC.
- You will then be able to manage and monitor as many Search Head Clusters as you need from a single pane of glass in TrackMe.

8.8.2 Step 2: Create a Flex Object tenant for Search Head Clusters monitoring

Now, let's create a new tenant for the purposes of monitoring our Search Head Clusters. We can achieve this via the UI wizard or with a simple command line:

```
| trackme url=/services/trackme/v2/vtenants/admin/add_tenant mode=post body="{'tenant_
↪name': 'shc-clusters', 'tenant_alias': '12 - SHC Clusters', 'tenant_desc': 'Splunk␣
↪Search Head Clusters', 'tenant_roles_admin': 'trackme_admin', 'tenant_roles_power':
↪'trackme_power', 'tenant_roles_user': 'trackme_user', 'tenant_owner': 'admin',
↪'tenant_idx_settings': 'global', 'tenant_flx_enabled': 'true'}"
```

8.8.3 Step 3: Create Flex Object trackers using the Flex Objects library

Hint

Find the use cases definition (SPL) in annex of this documentation

- See: *Annex: Use cases definition (SPL)*

Enter the newly created tenant and start the Flex Object wizard using the built-in Flex Objects library:


Tenant overview
Overview of the tenant id: shc-clusters / alias: 12 - SHC Clusters / Splunk Search Head Clusters
SPLK-FLX is enabled
Summary Index
Audit Index
Notable Index
Metrics Index
SPLK Flex Objects Tracking (SPLK-FLX) component overview
0 entities
Close
Refresh this
SLA overview
Hide this tenant from my profile
Overview of the tenant id: shc-clusters / Splunk Search Head Clusters - Get tenant activity insights and manage its life cycle from this screen


TrackMe - Data in motion tracking system
SPLK-FLX - FLEX OBJECTS TRACKING
INVESTIGATE STATUS FLIPPING
INVESTIGATE AUDIT CHANGES
TRACKING ALERTS
shc-clusters
Monitored entities count by priority
Monitored entities count by state and priority
Refresh
Manage: flex trackers
Manage: perm deleted
Manage: ogbse groups
Run Trackers reports
Reset Filters
Actions
Entity
Priority
Monitoring
Status
Status description
Metrics
Last metrics
Tracking information
Tracker runtime
Outliers



Select the use cases and remote accounts:

We will start with the use case "splk_splunk_shc_global_status" which tracks the SHC globally using the SHC API endpoints.




Flex Objects tracking - create a new tracker
You can pre-load a use case from the Flex Objects use case library, or manually define the Flex search logic
Vendor:
Splunk
Categories:
ALL
Use case
Review Flex Objects use cases library
Use case information:
Cancel
Back


Flex Objects tracking - create a new tracker
5. Test and Review:
Test the search tracker execution and the entities discovery:
Simulate the search
Cancel
Back
Hybrid tracker simulation finished successfully, verify the results before proceeding to its creation



Once executed, TrackMe immediately starts monitoring the SHC global status, generates and indexes metrics, and alerts you when issues are detected.




Actions for entity: infrastructure:shcluster:global-status:shc1 (alias: global-status:shc1)
Status: 1
monitored state: enabled
latest flip time: 11 Feb 2024 13:17
Priority: medium
Status description: ["maintenance mode is disabled", "min peers joined is met", "rolling restart status is met", "service ready flag is set", "KVstore maintenance status is disabled"]
host state: green
latest flip state: green
Description: Splunk Search Head Cluster
Overview entity
Notables
Outliers anomaly detection
Status flipping
Status message
Audit changes
Select metric(s):
Stats mode:
max
Chart mode:
column
100.00 %
Metrics for this entity:
30m
60m
2h
4h
8h
12h
24h
48h
7d
15d
30d
60d
90d
Refresh
Smart Status
Acknowledge alert
Disable
Delete
Modify
Search & reports
Close


Actions for entity: infrastructure:shcluster:global-status:shc1 (alias: global-status:shc1)
Status: 1
monitored state: enabled
latest flip time: 11 Feb 2024 13:17
Priority: medium
Status description: ["maintenance mode is disabled", "min peers joined is met", "rolling restart status is met", "service ready flag is set", "KVstore maintenance status is disabled"]
host state: green
latest flip state: green
Description: Splunk Search Head Cluster
Overview entity
Notables
Outliers anomaly detection
Status flipping
Status message
Audit changes
Last 7 days timeline
02/11/2024
green
Refresh
Smart Status
Acknowledge alert
Disable
Delete
Modify
Search & reports
Close



Repeat the same operation with the use case "splk_splunk_shc_members_status" which tracks the SHC members individually, so we instantly detect which members are having issues:



After the first execution of this second tracker, we have a global view of the Search Head Cluster, as well as the view on a per member basis, TrackMe will alert automatically when issues are detected.


TrackMe - Data in motion tracking system
shc-clusters
Monitored entities count by priority
Monitored entities count by state and priority
4
0
0
0


Actions for entity: infrastructure:shcluster:global-status:shc1 (alias: global-status:shc1)
100.00 %
Metrics for this entity





Finally, we can also opt out for the Search Head activity Flex use case, which focuses on collecting and monitoring the activity of the Search Heads:

Hint

Search Head activity Flex use case

- This use case relies on Splunk indexed data, unlike the rest related use cases which are basically interactions with the Splunk REST API.
- Therefore, If TrackMe has access to the same indexers from a search perspective, it is not required to target the remote account in this case.
- On the other hand, or if you prefer to do so, if the TrackMe guest is on a different indexing layer, using the remote account is mandatory.

- Finally, make sure to restrict the list of host to the Search Heads only, or this use case will also monitor any Search Head type of activity from other Search Heads







Note that unlike the previous use cases, we alert based on the detection of abnormal behaviours of the Search Heads using Machine Learning, notably TrackMe monitors the typical volume of search executions on the Search Head.

8.8.4 Annex: Use cases definition (SPL)

Use case splk_splunk_shc_global_status

```
| rest splunk_server=local timeout=300 count=0 /services/shcluster/captain/info
| stats first(label) as shc_captain_name, first(maintenance_mode) as maintenance_mode,
↪ first(min_peers_joined_flag) as min_peers_joined_flag, first(rolling_restart_flag)␣
↪as rolling_restart_flag, first(service_ready_flag) as service_ready_flag,␣
↪first(kvstore_maintenance_status) as kvstore_maintenance_status, first(elected_
↪captain) as elected_captain
```

(continues on next page)

(continued from previous page)

```
``` get SHC label ```
| join type=outer [ | rest splunk_server=local timeout=300 count=0 /services/server/
↪info | fields shcluster_label ]
| eval shcluster_label=if(isnull(shcluster_label) OR shcluster_label="", "shc1",␣
↪shcluster_label)

``` set object, alias and object_description ```
| eval object = "global-status:" . shcluster_label
| eval group = "infrastructure:shcluster"
| eval object_description = "Splunk Search Head Cluster"

| fields group, object, *

``` normalize kvstore_maintenance_status ```
| eval kvstore_maintenance_status=case(kvstore_maintenance_status=="enabled", 1,␣
↪kvstore_maintenance_status=="disabled", 0)

``` set status ```
| eval status = case(
maintenance_mode!=0 OR min_peers_joined_flag!=1 OR rolling_restart_flag!=0 OR service_
↪ready_flag!=1 OR kvstore_maintenance_status!=0, 2,
maintenance_mode=0 AND min_peers_joined_flag=1 AND rolling_restart_flag=0 AND service_
↪ready_flag=1 AND kvstore_maintenance_status=0, 1,
1=1, 3
)

``` set status_description ```
| eval status_description=case(
maintenance_mode=0, mvappend(status_description, "maintenance mode is disabled"),
maintenance_mode=1, mvappend(status_description, "maintenance mode is enabled")
)
| eval status_description=case(
min_peers_joined_flag=0, mvappend(status_description, "min peers joined is not met"),
min_peers_joined_flag=1, mvappend(status_description, "min peers joined is met")
)
| eval status_description=case(
rolling_restart_flag=0, mvappend(status_description, "rolling restart status is met"),
rolling_restart_flag=1, mvappend(status_description, "rolling restart status is in␣
↪progress or status is not met")
)
| eval status_description=case(
service_ready_flag=1, mvappend(status_description, "service ready flag is met"),
service_ready_flag=0, mvappend(status_description, "service ready flag is not met")
)
| eval status_description=case(
kvstore_maintenance_status=0, mvappend(status_description, "KVstore maintenance␣
↪status is disabled"),
kvstore_maintenance_status=1, mvappend(status_description, "KVstore maintenance␣
↪status is enabled")
)
| eval status_description="[\"" . mvjoin(status_description, ", \"") . "\"]"

``` set metrics ```
| eval metrics = "{" . "'splunk_shc.maintenance_mode': " . maintenance_mode . ",
↪'splunk_shc.min_peers_joined_flag': " . min_peers_joined_flag . ", 'splunk_shc.
```

(continues on next page)

(continued from previous page)

```
↪rolling_restart_flag': " . rolling_restart_flag . ", 'splunk_shc.service_ready_flag
↪': " . service_ready_flag . ", 'splunk_shc.kvstore_maintenance_status': " . ␣
↪kvstore_maintenance_status . "}"

``` alert if inactive for more than 3600 sec```
| eval max_sec_inactive=3600
```

Use case "splk_splunk_shc_members_status

```
| rest splunk_server=local timeout=300 count=0 /services/shcluster/status
| transpose
| rename column as category, "row 1" as value

``` extract information ```
| eval cluster_member_name=case(match(category, "^peers\.[^\.]*\.label$"), value)
| rex field=category "^peers\.(?<cluster_member_guid>[^\.]*)\.label$"
| eval cluster_member_status=case(match(category, "^peers\.[^\.]*\.status$"), value)

``` aggregate ```
| stats list(cluster_member_guid) as cluster_member_guid, list(cluster_member_name)␣
↪as cluster_member_name, list(cluster_member_status) as cluster_member_status

``` build entity from mv fields ```
| eval cluster_member_info=mvzip(mvzip(cluster_member_guid, cluster_member_name),␣
↪cluster_member_status)

``` order ```
| fields cluster_member_info, cluster_captain, service_ready_flag, rolling_restart_
↪flag

``` expand entities and extract ```
| mvexpand cluster_member_info
| eval cluster_member_info=split(cluster_member_info, ",")
| eval cluster_member_guid=mvindex(cluster_member_info, 0), cluster_member_
↪name=mvindex(cluster_member_info, 1), cluster_member_status=mvindex(cluster_member_
↪info, 2)

``` order ```
| fields cluster_member_guid, cluster_member_name, cluster_member_status

``` get SHC label ```
| join type=outer [ | rest splunk_server=local timeout=300 count=0 /services/server/
↪info | fields shcluster_label ]

``` set object, alias and object_description ```
| eval object = "shc:" . shcluster_label . "|peer:" . cluster_member_name . "|id:" .␣
↪cluster_member_guid
| eval group = "infrastructure:shcluster"
| eval object_description = "Splunk Search Head Cluster Member: " . cluster_member_
↪name . ", GUID: " . cluster_member_guid . ", SHC: " . shcluster_label

``` set status ```
| eval status = case(
cluster_member_status="Up", 1,
cluster_member_status="Down", 2,
1=1, 3
```

(continues on next page)

(continued from previous page)

```
)

``` set status description ```
| eval status_description = case(
status=1, "Member " . cluster_member_name . " is up and healthy, status: " . cluster_
↪member_status,
status=2, "Member " . cluster_member_name . " is currently down, status: " . cluster_
↪member_status,
status=3, "Member " . cluster_member_name . " status is unknown or unexpected,␣
↪status: " . cluster_member_status
)

``` alert if inactive for more than 3600 sec```
| eval max_sec_inactive=3600
```

Use case splk_splunk_shc_activity

```
index=_internal sourcetype=splunkd host=* (eligible=* OR active_hist_searches=* OR␣
↪dispatch_dir_reaper_max_ms=* OR bundle_dir_reaper_max_ms=* NOT "*group=search_
↪concurrency, system total*")
| fillnull value=0
| stats sum(eligible) as eligible sum(delayed) as delayed sum(dispatched) as␣
↪dispatched sum(total_lag) as total_lag sum(max_lag) as max_lag sum(scheduler_
↪cycles) as scheduler_cycles sum(actions_triggered) as actions_triggered␣
↪sum(completed) as completed sum(total_runtime) as total_runtime sum(max_runtime) as␣
↪max_runtime sum(active_searches) as active_searches sum(active_hist_searches) as␣
↪active_historical_searches max(bundle_dir_reaper_max_ms) as bundle_reaper_ms␣
↪max(dispatch_dir_reaper_max_ms) as dispatch_reaper_ms by host

``` ensures every metric has a value ```
| foreach eligible, delayed, dispatched, total_lag, max_lag, scheduler_cycles,␣
↪actions_triggered, completed, total_runtime, max_runtime, active_searches, active_
↪historical_searches, bundle_reaper_ms, dispatch_reaper_ms [ eval <<FIELD>> =␣
↪if(isnum('<<FIELD>>'), '<<FIELD>>', 0) ]

``` set group ```
| eval group="infrastructure:search_head"

``` set object and description ```
| eval object=host, alias=object
| eval object_description = "Splunk Search Head key activity metrics, Search Head: " .
↪ object

``` set a basis status, we will rely on ML for anomaly detection ```
| eval status_description = "Active Searches:" . active_searches . ", Historical␣
↪Searches:" . active_historical_searches . ", Dispatched|Completed: " . dispatched .
↪" | " . completed . ""
| eval status = 1

``` set metrics ```
| eval metrics = "{'sh.eligible': " . eligible . ", 'sh.delayed': " . delayed . ",
↪'sh.dispatched': " . dispatched . ", 'sh.total_lag': " . total_lag . ", 'sh.max_lag
↪': " . max_lag . ", 'sh.scheduler_cycles': ". scheduler_cycles . ", 'sh.actions_
↪triggered': " . actions_triggered . ", 'sh.completed': ". completed . ", 'sh.total_
↪runtime': " . total_runtime . ", 'sh.max_runtime': " . max_runtime . ", 'sh.active_
↪searches': " . active_searches . ", 'sh.active_historical': " . active_historical_
```

(continues on next page)
```

(continued from previous page)

```
↪searches . ", 'sh.bundle_reaper_ms': " . bundle_reaper_ms . ", 'sh.dispatch_reaper_
↪ms': " . dispatch_reaper_ms . "}"

``` set outliers metrics ```
| eval outliers_metrics = "{'sh.completed': {'alert_lower_breached': 1, 'alert_upper_
↪breached': 0, 'time_factor': '%w%H'}}"

| eval max_sec_inactive=3600
```

8.9 Backing up and Restoring TrackMe

Hint

TrackMe Backup and Restore new features since 2.1.5: backup, restore, clone and migrate TrackMe Knowledge Objects & KVstore collections

- Since TrackMe 2.1.5, a new set of features have been introduced to allow for more flexibility and capabilities in managing TrackMe Knowledge Objects and KVstore collections.
- TrackMe now allows for the **backup, restore, clone and migrate** of TrackMe **Knowledge Objects** and **KVstore collections records**, entirely within TrackMe.
- These features leverage new TrackMe endpoints and utilities, and a refreshed dashboard allowing access to command examples and reviewing available backups.
- These new capabilities leverage the Splunk API regarding backing up and restoring Knowledge Objects, and works whether you are a **Splunk Enterprise or Cloud customer**, in a **standalone or Search Head Cluster** context.
- This new sets of incredibly **powerful and flexible** features allow TrackMe to be entirely autonomous regarding its backup and restore capabilities, and to be able to manage its own Knowledge Objects and KVstore collections records in a very granular way.
- This documentation has been updated to reflect these new capabilities, and to provide guidance on how to use them.

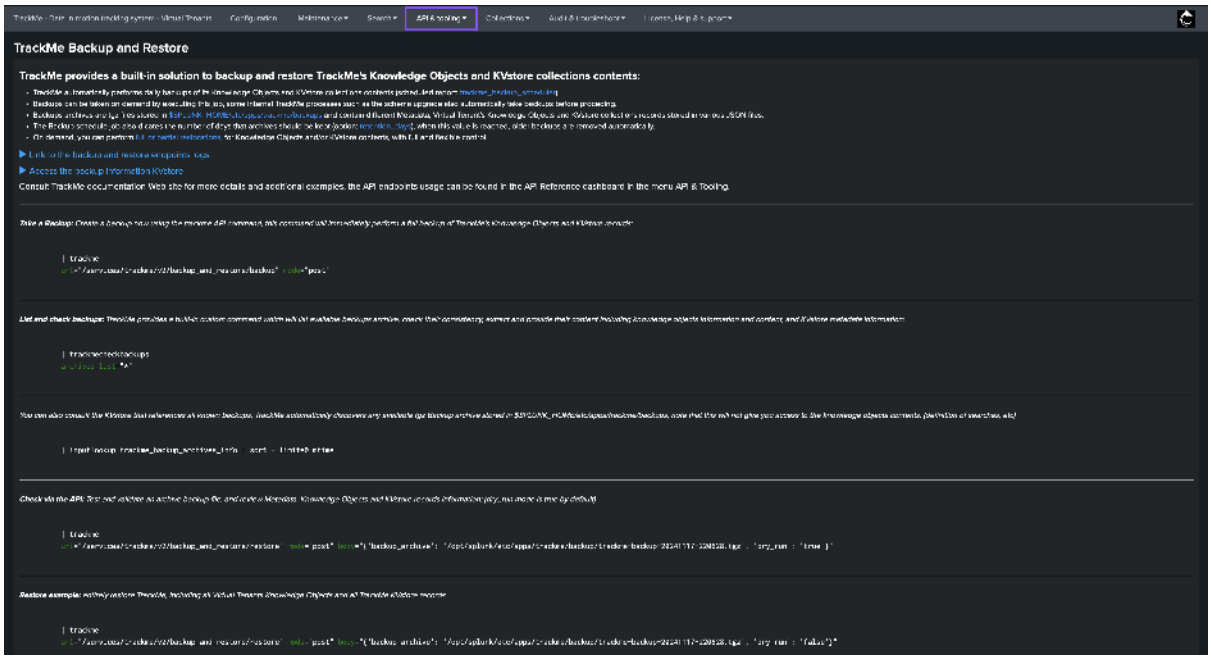
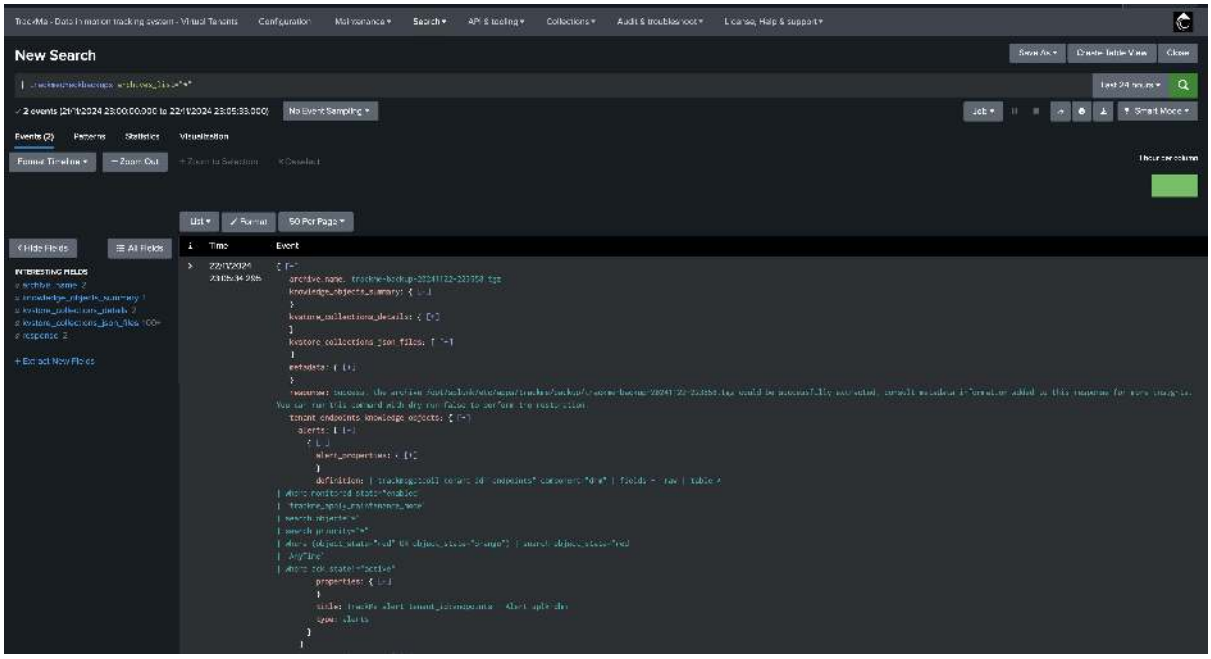
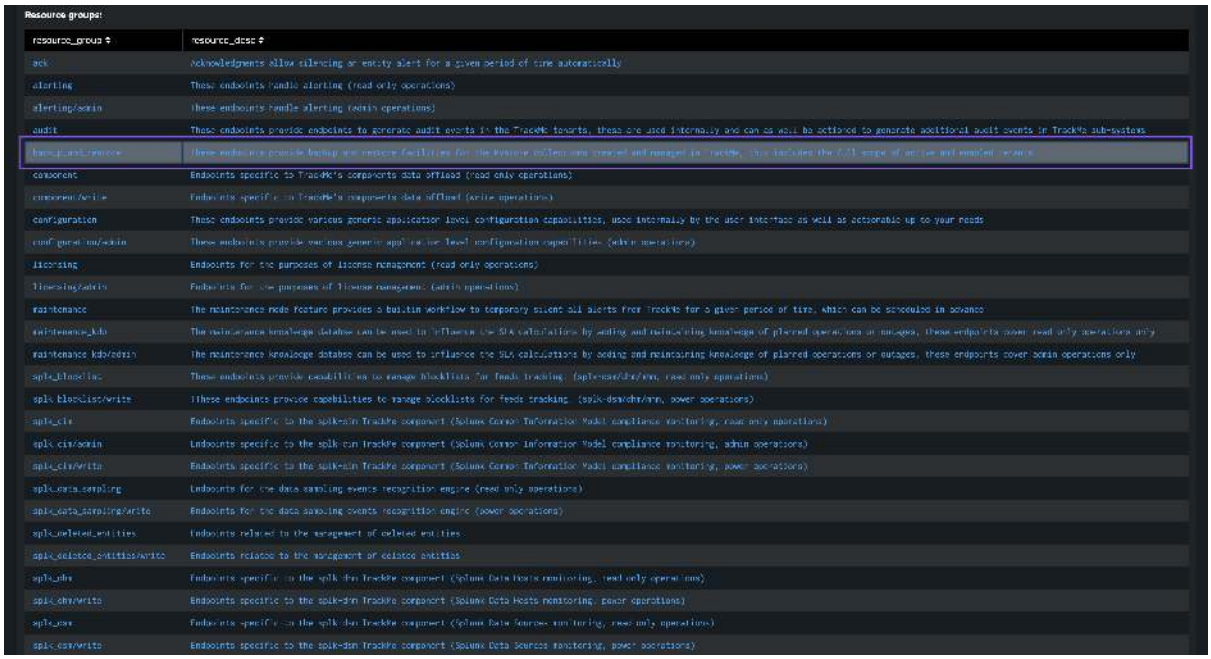
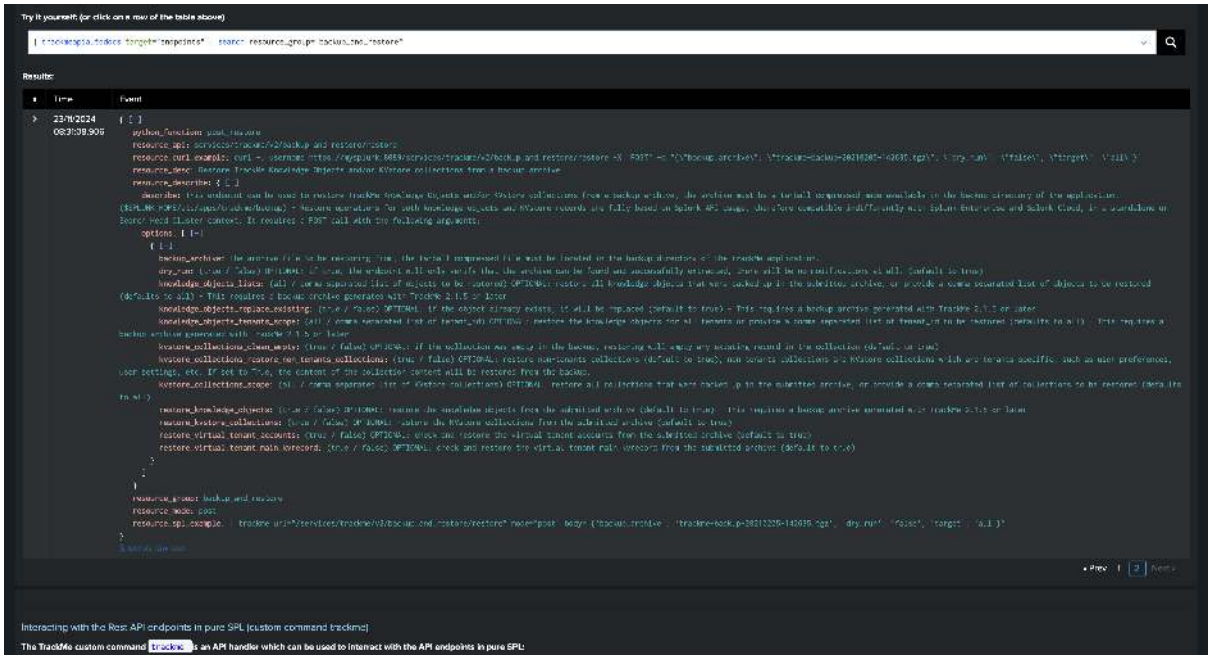
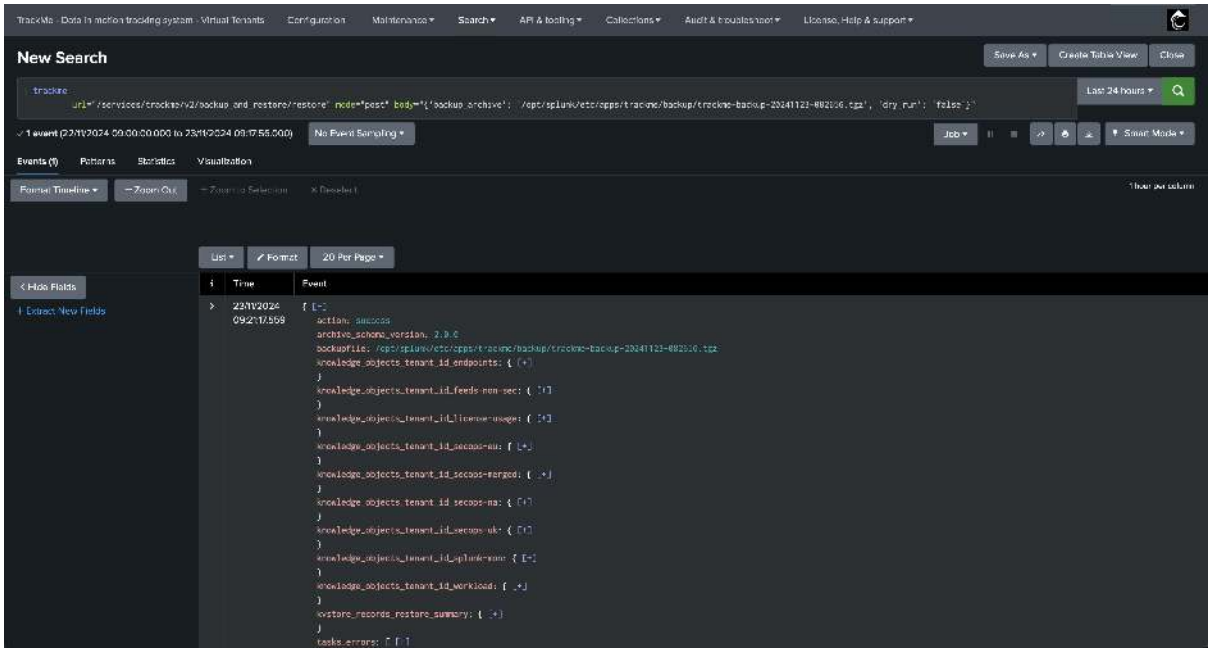
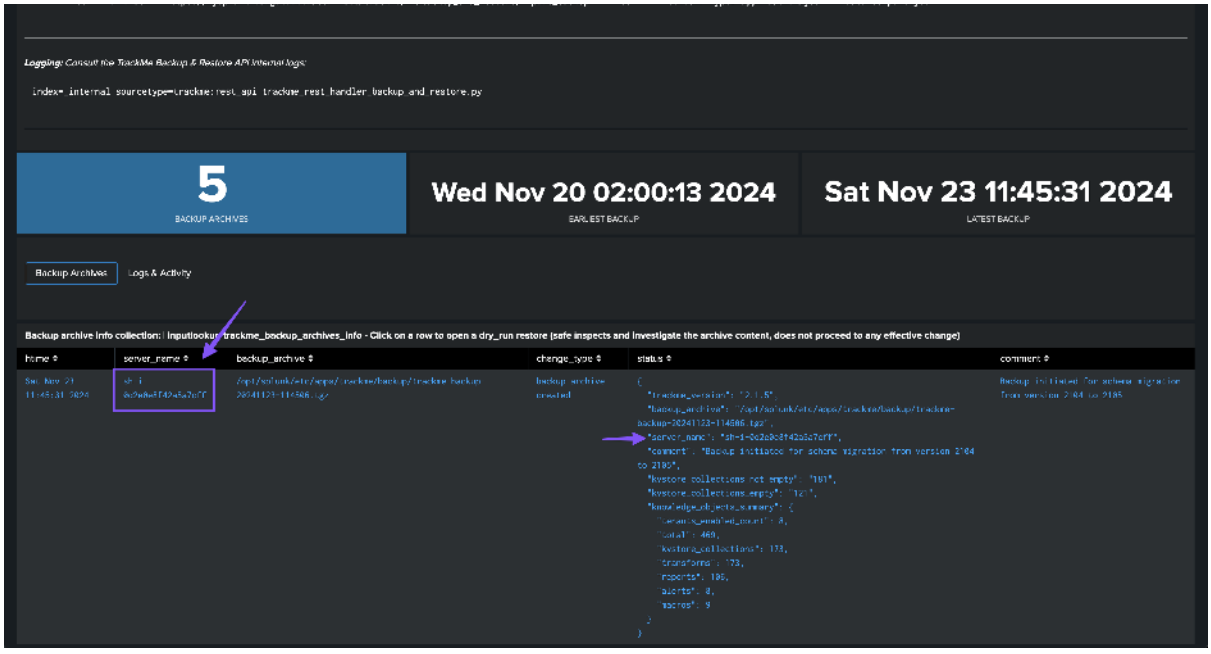
Note

Limitations:

- **Disabled** Virtual Tenants are **not** backed up, and therefore **not** restorables, when performing a restore operation, TrackMe will automatically force purge a Virtual Tenant if it was disabled when the backup was performed.
- **Alert actions restoration:** TrackMe can restore any alert that was created with TrackMe, however only TrackMe built-in alerts actions and Splunk Email alert actions are restored. (external alert actions will need to be re-configured manually once restored)

8.9.1 Introduction to backup and restore

TrackMe is a complex Splunk application, backing up and restoring means that the following need to be taken into account:

- In TrackMe, Splunk Knowledge Objects are created on the fly, this includes KVstore definitions, transforms, macros, reports, and alerts.
```

- TrackMe heavily relies on KVstores, the application stores various states, as well as its own data and configuration in KVstores.
- When backing up and restoring TrackMe, both aspects have to be managed together, you cannot restore TrackMe configuration files without restoring the KVstore data and vice versa.
- TrackMe provides builtin advanced capabilities to backup and restore its Knowledge Objects and resilient data stores. (KVstores)

8.9.2 Backing up TrackMe

Automated backups in TrackMe

TrackMe automatically performs backups on the following circumstances:

- Every days, the scheduled report `trackme_backup_scheduler` executes during the night and performs a full backup of TrackMe.
- When TrackMe is upgraded to a new release, TrackMe performs a backup when the first Virtual Tenant is upgraded through a process called `schema upgrade`.

TrackMe backups location:

- TrackMe backups are stored in the backup directory at the root of the TrackMe application directory.
- The backup directory is located at: `$SPLUNK_HOME/etc/apps/trackme/backup/`
- The format of the files are compressed tarballs, with the following naming convention: `trackme-backup-YYYYMMDD-HHMMSS.tgz`

Reviewing TrackMe backups

To review the available backups, you can use the builtin dashboard "TrackMe Backup and Restore" available in the menu API &amp Tooling:

The first part of the dashboard shows various use cases and command examples:



Scroll down to see available backups in this TrackMe deployments, or access logs:

TrackMe logs the backup context in the comment metadata, if the backup is executed during the context of the release upgrade, the comment mentions this and the schema version upgrade information.

Checking, Inspecting and Reviewing TrackMe backups

The most practical way to easily check, inspect and review TrackMe's backups, including Knowledge Objects and their definition, is to use a built-in command called trackmecheckbackups:

```
| trackmecheckbackups archives_list="*"
```

What this backend does:

- Lists and iterates over all available TrackMe backups.
- Verifies that the backup archives can successfully be extracted.
- Access and iterate through the archives metadata.
- Render the archives contents and metadata in pure SPL and on a per archive basis.

Notes:

- The command does not give access to the KVstore records themselves, but gives detailed metadata for each kvstore collection. (number of records, etc)
- Knowledge Objects can be fully accessed, and you can for instance review the definition of a given search, alert or macro at some points in time via the archive.



Accessing the Backup metadata KVstore

TrackMe stores the metadata of the backups in a dedicated KVstore collection, you can access this collection using the following SPL command:

```
| inputlookup trackme_backup_archives_info | sort - limit=0 mtime
```

The KVstore is used in the dashboard to provide high-level information about the backups, such as the number of archives, the size of the archives, the creation date, etc... This KVstore is also used by the schema upgrade process to determine the latest backup available.

Finally, note that TrackMe does not restore the KVstore records for this specific KVstore, the reason is that TrackMe automatically discovers available backups, and will re-create any missing records for a backup which would not be currently known to the system.

Anatomy of a TrackMe backup

A TrackMe backup archive contains the following elements:

- The backup archive itself, a compressed tarball containing all the files and directories of the backup.
- A full metadata file, containing the metadata of the backup, such as the creation date, the size of the archive, etc.
- A light metadata file, containing a summary of the backup, such as the number of Knowledge Objects, the number of KVstore collections, etc.

Light metadata content example:

```
{
"trackme_version": "2.1.5",
"backup_archive": "/opt/splunk/etc/apps/trackme/backup/trackme-backup-20241123-020021.
↪tgz",
"server_name": "xxxxxxxxxxx",
"comment": "Backup initiated by splunk-system-user, date: Sat Nov 23 02:00:21 2024",
"kvstore_collections_not_empty": "199",
"kvstore_collections_empty": "148",
"knowledge_objects_summary": {
        "tenants_enabled_count": 9,
```

(continues on next page)

(continued from previous page)

```
        "total": 522,
        "kvstore_collections": 191,
        "transforms": 191,
        "reports": 120,
        "alerts": 9,
        "macros": 11
    }
}
```

Example:

```
cd $SPLUNK_HOME/etc/apps/trackme/backup/
```

```
-rw------- 1 splunk splunk 713476 Nov 22 22:36 trackme-backup-20241122-223558.tgz
-rw------- 1 splunk splunk  45149 Nov 22 22:36 trackme-backup-20241122-223558.tgz.
↪full.meta
-rw------- 1 splunk splunk    518 Nov 22 22:36 trackme-backup-20241122-223558.tgz.
↪light.meta
```

The content of the archive itself:

- One JSON file per KVstore collection, containing the records of the collection. (Note: the file will be empty if the collection is empty)
- One JSON file per Virtual Tenant containing the Knowledge Objects of the Virtual Tenant.
- One JSON file per Virtual Tenant containing the Virtual Tenant account information.

```
kv_<kvstore_transform_reference>.json
... <repeated for each KVstore>
tenant_<tenant_id>_knowledge_objects.json
tenant_<tenant_id>_vtenant_account.json
```

8.9.3 Restoring TrackMe

Restoring TrackMe can be achieved with flexibility, and different options are available depending on your needs:

- `full restoration`: You can entirely restore TrackMe Knowledge Objects and KVstore records from a backup.
- `full restoration targeting all or a list of Virtual Tenants, without deleting and replacing existing Knowledge Objects`: You can restore TrackMe Knowledge Objects and KVstore records from a backup, without altering existing Knowledge Objects.
- `full restoration targeting all or a list of Virtual Tenants, with or without replacing Knowledge Objects, and without restoring KVstore records`: You can restore TrackMe Knowledge Objects from a backup, without restoring KVstore records.
- `partial restoration of Knowledge Objects`: You can restore a specific list of Knowledge Objects from a backup.
- `partial restoration of KVstore collections records`: You can restore a specific list of KVstore collections and their records from a backup.
- `mixed restoration`: You can mix and match the above options to achieve a specific restoration scenario.

The next steps will guide you through the restoration process based on these different scenarios.

Restore endpoint API and its options

The TrackMe API restore endpoint handle these operations, you can review backup and restore endpoints and their usage in the API Reference dashboard:





You can also run:

```
| trackmeapiautodocs target="endpoints" | search resource_group="backup_and_restore"
```

Endpoint options:

- `backup_archive`: The archive file to be restored, the tarball compressed file must be located in the backup directory of the trackMe application.
- `dry_run`: (true / false) OPTIONAL: if true, the endpoint will only verify that the archive can be found and successfully extracted, there will be no modifications at all. (default to true)
- `knowledge_objects_lists`: (all / comma separated list of objects to be restored) OPTIONAL: restore all knowledge objects that were backed up in the submitted archive, or provide a comma

separated list of objects to be restored (defaults to all) - This requires a backup archive generated with TrackMe 2.1.5 or later

- `knowledge_objects_replace_existing`: (true / false) OPTIONAL: if the object already exists, it will be replaced (default to true) - This requires a backup archive generated with TrackMe 2.1.5 or later
- `knowledge_objects_tenants_scope`: (all / comma separated list of tenant_id) OPTIONAL: restore the knowledge objects for all tenants or provide a comma separated list of tenant_id to be restored (defaults to all) - This requires a backup archive generated with TrackMe 2.1.5 or later
- `kvstore_collections_clean_empty`: (true / false) OPTIONAL: if the collection was empty in the backup, restoring will empty any existing record in the collection (default to true)
- `kvstore_collections_restore_non_tenants_collections`: (true / false) OPTIONAL: restore non-tenants collections (default to true), non tenants collections are KVstore collections which are tenant specific, such as user preferences, user settings, etc. If set to True, the content of the collection content will be restored from the backup.
- `kvstore_collections_scope`: (all / comma separated list of KVstore collections) OPTIONAL: restore all collections that were backed up in the submitted archive, or provide a comma separated list of collections to be restored (defaults to all)
- `restore_knowledge_objects`: (true / false) OPTIONAL: restore the knowledge objects from the submitted archive (default to true) - This requires a backup archive generated with TrackMe 2.1.5 or later
- `restore_kvstore_collections`: (true / false) OPTIONAL: restore the KVstore collections from the submitted archive (default to true)
- `restore_virtual_tenant_accounts`: (true / false) OPTIONAL: check and restore the virtual tenant accounts from the submitted archive (default to true)
- `restore_virtual_tenant_main_kvrecord`: (true / false) OPTIONAL: check and restore the virtual tenant main record from the submitted archive (default to true)

Restore operations examples

Full restoration of TrackMe

Example:

```
| trackme
url="/services/trackme/v2/backup_and_restore/restore" mode="post" body="{'backup_
↪archive': '/opt/splunk/etc/apps/trackme/backup/trackme-backup-20241117-220628.tgz',
↪'dry_run': 'false'}"
```

This example above will:

- For each Virtual Tenant, restore the Virtual Tenant account
- For each Virtual Tenant, restore the central KVstore record
- For each Virtual Tenant, restore Knowledge Objects, for each Knowledge Object, if the Knowledge Object exists already, it will be deleted and re-created from the backup
- For each Virtual Tenant KVstore collection, restore the KVstore collection records (if the collection was empty in the backup, the collection will be emptied)
- Restore central and non-tenants related KVstore collections records

Restore example: restore a specific Virtual Tenant, including all Virtual Tenants Knowledge Objects and all TrackMe KVstore records for this Virtual Tenant only

```
| trackme
     url="/services/trackme/v2/backup_and_restore/restore" mode="post" body="{
↪'backup_archive': '/opt/splunk/etc/apps/trackme/backup/trackme-backup-20241117-
↪220628.tgz', 'dry_run': 'false', 'knowledge_objects_tenants_scope': 'my_tenant',
↪'knowledge_objects_replace_existing': 'true', 'kvstore_collections_clean_empty':
↪'true', 'kvstore_collections_restore_non_tenants_collections': 'false'}"
```

This example above will:

- For the Virtual Tenant "my_tenant", restore the Virtual Tenant account
- For the Virtual Tenant "my_tenant", restore the central KVstore record
- For the Virtual Tenant "my_tenant", restore Knowledge Objects, for each Knowledge Object, if the Knowledge Object exists already, it will be deleted and re-created from the backup

Restore example: restore one or more specific Knowledge Objects, and nothing else (if the report calls itself another report, the parent report will be also restored automatically)

```
| trackme
url="/services/trackme/v2/backup_and_restore/restore" mode="post" body="{'backup_
↪archive': '/opt/splunk/etc/apps/trackme/backup/trackme-backup-20241117-220628.tgz',
↪'dry_run': 'false', 'knowledge_objects_lists': 'my-tracker-001,my-tracker-002',
↪'restore_virtual_tenant_accounts': 'false', 'restore_virtual_tenant_main_kvrecord':
↪'false', 'knowledge_objects_replace_existing': 'true', 'restore_kvstore_collections
↪': 'false'}"
```

This example above will:

- Restore the Knowledge Objects "my-tracker-001" and "my-tracker-002" from the backup
- If the Knowledge Object exists already, it will be deleted and re-created from the backup
- No Virtual Tenant account will be restored
- No Virtual Tenant KVstore record will be restored
- No KVstore collection records will be restored

Restore example: restore one or more specific KVstore collections contents, and nothing else (kvstore_collections_clean_empty allows controlling if the collection should be empty, if it was actually empty during the backup)

```
| trackme
url="/services/trackme/v2/backup_and_restore/restore" mode="post" body="{'backup_
↪archive': '/opt/splunk/etc/apps/trackme/backup/trackme-backup-20241117-220628.tgz',
↪'dry_run': 'false', 'kvstore_collections_scope': 'kv_trackme_dsm_tenant_mytenant,kv_
↪trackme_dsm_priority_policies_tenant_test-restore', 'kvstore_collections_clean_empty
↪': 'true', 'restore_knowledge_objects': 'false', 'restore_virtual_tenant_accounts':
↪'false', 'restore_virtual_tenant_main_kvrecord': 'false'}"
```

This example above will:

- Restore the KVstore collections "kv_trackme_dsm_tenant_mytenant" and "kv_trackme_dsm_priority_policies_tenant_test-restore" from the backup
- If the collection was empty in the backup, the collection will be emptied
- No Knowledge Objects will be restored

- No Virtual Tenant account will be restored
- No Virtual Tenant KVstore record will be restored

8.9.4 Exporting and Importing TrackMe backups, for migration or cloning purposes

TrackMe provides importing and exporting capabilities that can be used to migrate or clone TrackMe configurations and data between different TrackMe deployments:

Exporting a TrackMe backup (source system)

You can use the export REST API to export a TrackMe backup archive as a base64 string, which can then be used to call the import endpoint to a different Splunk instance running TrackMe (use cases: migration, cloning...) - This example shows the usage with curl

```
curl -k -u admin https://mysplunk-source:8089/services/trackme/v2/backup_and_restore/
↪export_backup -X "POST" -d "{\"archive_name\": \"trackme-backup-20241120-064752.tgz\
↪"}" --output export.json
```

*Review the resulting file export.json.**

Importing a TrackMe backup (target system)

Then, you can use the import REST API to import a TrackMe backup archive from the base64 string, TrackMe automatically verifies the archive, if the archive is valid, it will be discovered and integrated into the available archives immediately - This example shows the usage with curl

```
curl -k -u admin https://mysplunk-target:8089/services/trackme/v2/backup_and_restore/
↪import_backup -X "POST" -H "Content-Type: application/json" --data @export.json
```

Expected response:

```
{"success":"Backup imported successfully"}
```

Note:

- TrackMe will automatically identify the backup file on the target system, and will include it in the available backups.
- After an import, if you open the TrackMe Backup and Restore dashboard, you will see the imported backup in the list of available backups.

Migrating or cloning the TrackMe deployment

Once you have imported the backup on the target system, you can then proceed with the restoration process as described in the previous section, to restore the Knowledge Objects and KVstore records.

```
| trackme
url="/services/trackme/v2/backup_and_restore/restore" mode="post" body="{'backup_
↪archive': '/opt/splunk/etc/apps/trackme/backup/trackme-backup-20241117-220628.tgz',
↪'dry_run': 'false'}"
```



8.9.5 Splunk Search Head Cluster specific considerations

In a Splunk Search Head Cluster context, especially in Splunk Cloud where you cannot access the filesystem, you may need to consider the following:

- The backup file on the filesystem is Search Head member specific, backup files are **not** replicated to other members.
- This means that when calling a restore operation, you need to make sure that either the backup file is available on the member commands are executed, or that you can target the specific member of the Search Head cluster where the backup file is hosted.

Identifying the backup file member location:

- TrackMe stores the information about the server name where the tgz backup file was generated.
- You can see this information in the Backup and Restore dashboard under the `server_name` field.



Therefore:

- You can use the export endpoint to export the backup file and by targetiing the SHC member where it is located.
- Then use the import endpoint to import the backup file on the target SHC member.
- Finally, you can execute the restore operation on the target SHC member.
- These steps can be used whenever you need to restore in the same environment, or if the objective is to migrate or clone the TrackMe deployment from a SHC to another. (same or different Splunk deployment)

Hint

*Accessing Splunk Cloud specific Search heads API**

- In Splunk Cloud, you can access the Search Head members API using the following URL: `https://<search_head_member_name>.<cloud stack name>.splunkcloud.com:8089`
- To get the fully qualified name of the server you can connected to, in Splunk Web, click on top right "Support and Services", "About"
- You can easily found out the list of Search Head members in the Cloud monitoring console, or by looking splunkd internal logs.
- Finally, remember that you must allow public IP network where your connection starts from in the Splunk Cloud Splunk REST API allow list in Splunk Cloud.

Examples:

- Step 1 Export from the source SHC member

```
curl -k -u admin https://<shc member source>:8089/services/trackme/v2/backup_and_
↪restore/export_backup -X "POST" -d "{\"archive_name\": \"trackme-backup-20241120-
↪064752.tgz\"}" --output export.json
```

- Step 2: Import to the SHC member target

```
curl -k -u admin https://<shc member target>:8089/services/trackme/v2/backup_and_
↪restore/import_backup -X "POST" -H "Content-Type: application/json" --data @export.
↪json
```

- Step 3: Restore on the SHC member target

```
curl -k -u admin https://<shc member target>:8089/services/trackme/v2/backup_and_
↪restore/restore -X "POST" -H "Content-Type: application/json" -d "{\"backup_archive\
↪": \"/opt/splunk/etc/apps/trackme/backup/trackme-backup-20241120-064752.tgz\", \
↪"dry_run\": \"false\"}"
```

8.9.6 Accessing logs and troubleshooting

TrackMe logs all backup and restore operations:

```
index=_internal sourcetype=trackme:rest_api trackme_rest_handler_backup_and_restore.py
```

8.10 Auto deletion or management of TrackMe entities

8.10.1 Introduction to automated management of TrackMe entities

About this paper:

- In some use cases, you may want to automatically manage entities in TrackMe based on different criteria.
- For instance, if you track hosts in TrackMe, you could automatically delete machines that have been tagged as decommissioned in your CMDB.
- This way, you would automatically keep your TrackMe entities in sync with your CMDB and purge entities as needed.

8.10.2 Automatically deleting entities in TrackMe with a custom search

In this example, we are managing hosts in TrackMe through the splk-dhm component. We can access the real-time view very easily using the following command:

Note: Click on the "Search table" button in TrackMe's UI to access this search.

Hint

Replace tenant and component name:

- Replace "mytenant" with the tenant ID you want to delete entities from.

```
| trackmegetcoll tenant_id=hosts-dsm-tracking component=dsm
```

Then, let's add a table to focus on specific fields:

```
| trackmegetcoll tenant_id=host-dhm-tracking component=dhm
| table keyid, object, alias, object_state
```

Now, let's assume that we want to filter on entities that have stopped forwarding data to Splunk, therefore are currently in a red state.

```
| trackmegetcoll tenant_id=host-dhm-tracking component=dhm
| table keyid, object, alias, object_state

``` select entities in alert ```
| where object_state="red"
```

Then, we will cross with our CMDB which is available in Splunk, we will use the alias field which contains the true host as seen in Splunk, you obviously can add any additional logic and/or customise the logic to match your context and requirements:

```
| trackmegetcoll tenant_id=host-dhm-tracking component=dhm
| table keyid, object, alias, object_state

``` select entities in alert ```
| where object_state="red"

``` cross with your CMDB ```
| lookup my_hosts_cmdb ci_id as alias OUTPUT ci_status
| where isnotnull(ci_status) AND ci_status!=""
```
```

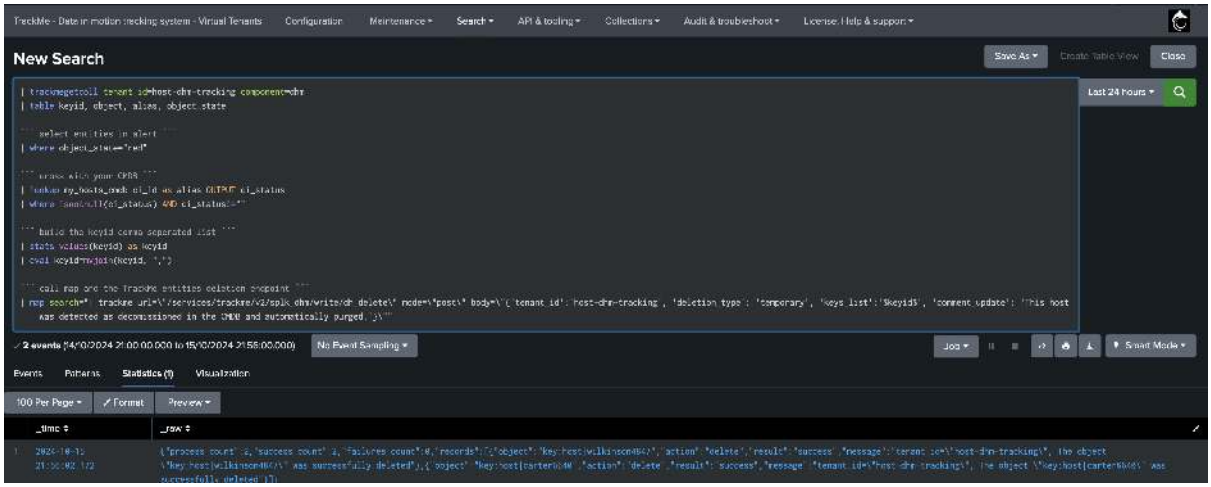
The next step is to turn this into a comma-separated list of keyids, so that we can use it in a delete command using TrackMe's API endpoints and built-in pure SPL. We will also use the map command from Splunk:

```
| trackmegetcoll tenant_id=host-dhm-tracking component=dhm
| table keyid, object, alias, object_state

``` select entities in alert ```
| where object_state="red"

``` cross with your CMDB ```
| lookup my_hosts_cmdb ci_id as alias OUTPUT ci_status
| where isnotnull(ci_status) AND ci_status!=""

``` build the keyid comma separated list ```
| stats values(keyid) as keyid
| eval keyid=mvjoin(keyid, ",")
```

Last, and final step, call the endpoint:

```
| trackmegetcoll tenant_id=host-dhm-tracking component=dhm
| table keyid, object, alias, object_state

``` select entities in alert ```
| where object_state="red"

``` cross with your CMDB ```
| lookup my_hosts_cmdb ci_id as alias OUTPUT ci_status
| where isnotnull(ci_status) AND ci_status!=""

``` build the keyid comma separated list ```
| stats values(keyid) as keyid
| eval keyid=mvjoin(keyid, ",")

``` call map and the TrackMe entities deletion endpoint ```
| map search="| trackme url=\"/services/trackme/v2/splk_dhm/write/dh_delete\" mode=\
↪"post\" body=\"{'tenant_id':'host-dhm-tracking', 'deletion_type': 'temporary',
↪'keys_list':'$keyid$', 'comment_update': 'This host was detected as decommissioned␣
↪in the CMDB and automatically purged.'}\""
```

We will schedule this search to run automatically. It doesn't need to run frequently, and most likely once per day or every few hours would be more than enough.

Any entities that are detected as decommissioned in the CMDB will be automatically purged from TrackMe. We also have TrackMe audit events should we need to review the deletion history.
```



8.10.3 Alternative using the Flex object component (splk-flex)

Licensed TrackMe users can also leverage instead the Flex object component to track hosts, the same logic can be used very easily too:

```
| trackmegetcoll tenant_id=hosts-flx-tracking component=flx
| table keyid, object, alias, object_state, object_description

``` select entities in alert ```
| where object_state="red"

``` cross with your CMDB ```
| lookup my_hosts_cmdb ci_id as alias OUTPUT ci_status
| where isnotnull(ci_status) AND ci_status!=""

``` build the keyid comma separated list ```
| stats values(keyid) as keyid
| eval keyid=mvjoin(keyid, ",")

``` call map and the TrackMe entities deletion endpoint ```
| map search="| trackme url=\"/services/trackme/v2/splk_flx/write/flx_delete\" mode=\
↪"post\" body=\"{'tenant_id':'hosts-flx-tracking', 'deletion_type': 'temporary',
↪'keys_list':'$keyid$', 'comment_update': 'This host was detected as decommissioned␣
↪in the CMDB and automatically purged.'}\""
```

8.10.4 Alternative logic: disabling rather than deleting entities

Alternatively, we may want to disable entities rather than deleting them. The same logic would apply, and we would simply call a different endpoint.

With our splk-dhm example:

```
| trackmegetcoll tenant_id=host-dhm-tracking component=dhm
| table keyid, object, alias, object_state

``` select entities in alert ```
| where object_state="red"

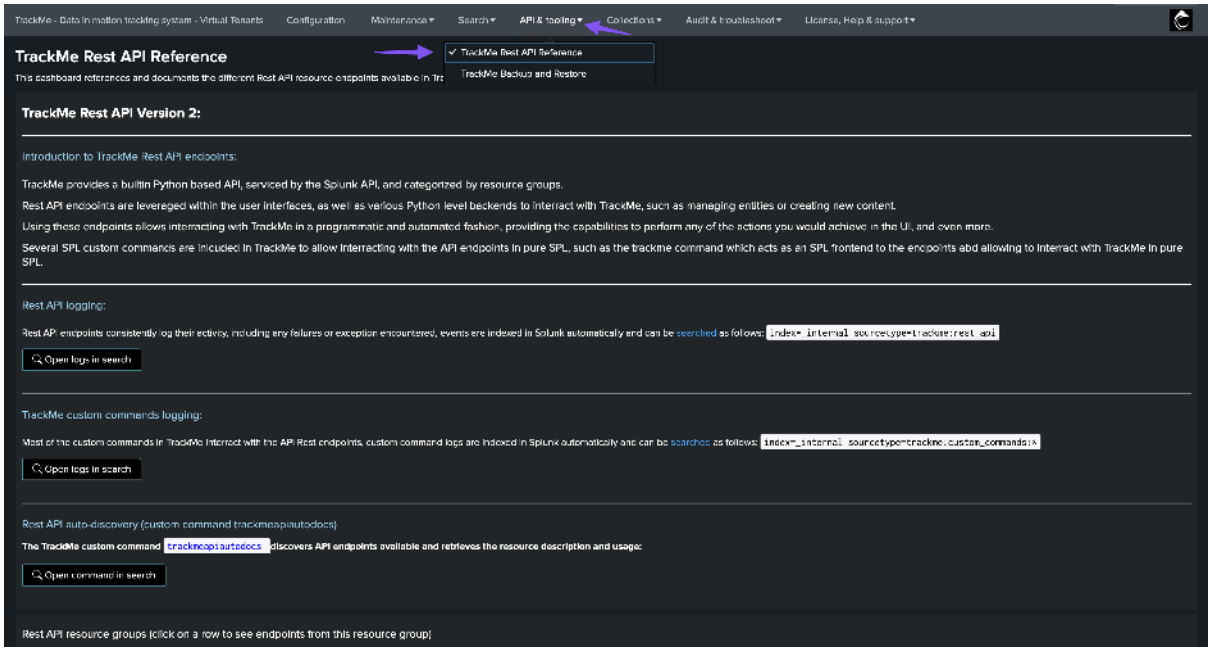
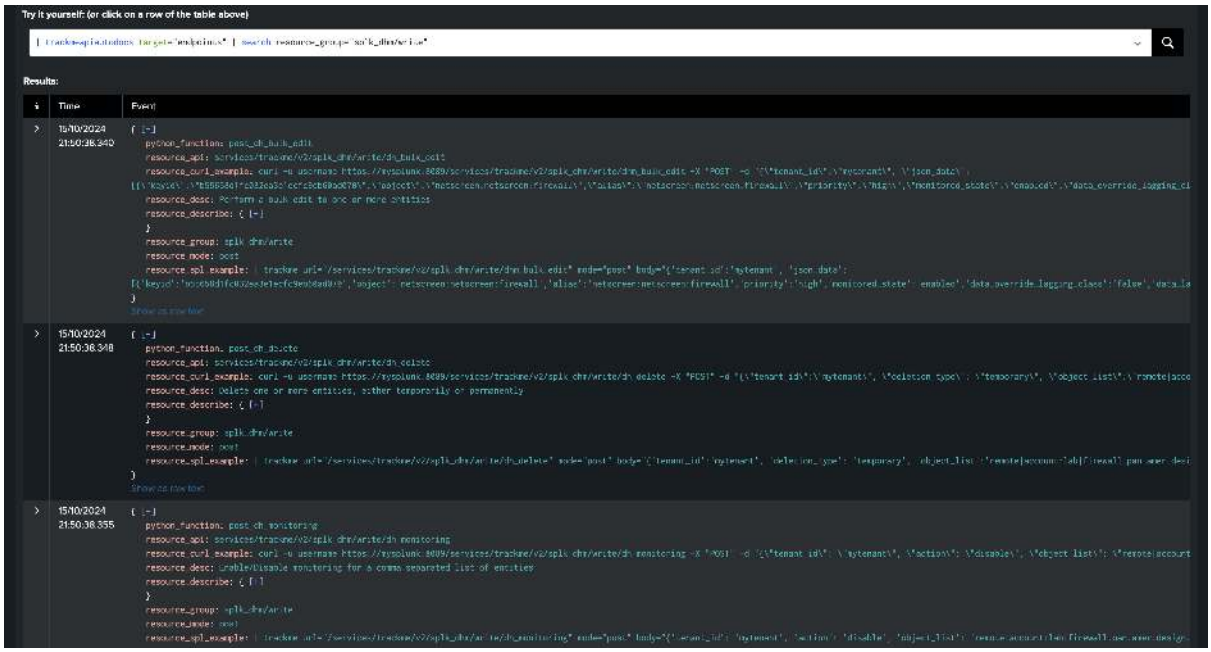
``` cross with your CMDB ```
| lookup my_hosts_cmdb ci_id as alias OUTPUT ci_status
| where isnotnull(ci_status) AND ci_status!=""
```

(continues on next page)

(continued from previous page)

```
``` build the keyid comma separated list ```
| stats values(keyid) as keyid
| eval keyid=mvjoin(keyid, ",")

``` call map and the TrackMe entities deletion endpoint ```
| map search="| trackme url=\"/services/trackme/v2/splk_dhm/write/dh_monitoring\"␣
↪mode=\"post\" body=\"{'tenant_id':'host-dhm-tracking', 'action': 'disable', 'keys_
↪list':'$keyid$', 'comment_update': 'This host was detected as decommissioned in the␣
↪CMDB and automatically disabled.'}\""
```

With our splk-flx example:

```
| trackmegetcoll tenant_id=host-flx-tracking component=dhm
| table keyid, object, alias, object_state

``` select entities in alert ```
| where object_state="red"

``` cross with your CMDB ```
| lookup my_hosts_cmdb ci_id as alias OUTPUT ci_status
| where isnotnull(ci_status) AND ci_status!=""

``` build the keyid comma separated list ```
| stats values(keyid) as keyid
| eval keyid=mvjoin(keyid, ",")

``` call map and the TrackMe entities deletion endpoint ```
| map search="| trackme url=\"/services/trackme/v2/splk_dhm/write/flx_monitoring\"␣
↪mode=\"post\" body=\"{'tenant_id':'host-flx-tracking', 'action': 'disable', 'keys_
↪list':'$keyid$', 'comment_update': 'This host was detected as decommissioned in the␣
↪CMDB and automatically disabled.'}\""
```

Hint

The same type of logic applies to anything in TrackMe:

- In fact, you can use the same type of logic in many variations, whatever component you are dealing with.
- You can, for example, decide to have entities with a monitored state defined as disabled when entities are discovered, then wait for at least 7 days of historical knowledge to be collected before enabling the monitoring, or any other concept that makes sense for your context.
- Consult the REST API reference dashboard in TrackMe, menu **API & Tooling** to find which endpoints are available for each component, as well as their usage and usage examples.





8.11 Performing mass operations in TrackMe

8.11.1 Introduction to mass operations in TrackMe

About mass operations in TrackMe

- There are different contexts in which mass operations can be performed in TrackMe.
- Depending on the context, you can perform mass operations directly in the TrackMe UI (bulk edit), in some contexts you can also directly use TrackMe REST API endpoints and eventually take advantage of Splunk SPL techniques.
- The purpose of this whitepaper is to document various typical mass operations that can be performed in TrackMe, and to provide guidance on how to perform them.

8.11.2 Mass deleting entities

The following example demonstrates how to mass delete entities in TrackMe:

Hint

Replace tenant and component name:

- Replace "mytenant" with the tenant ID you want to delete entities from.
- "dsm" is the component name, simply replace it with the component you want to delete entities from. (dsm/dhm/mhm/wlk/flx)

Context:

- In this example, we assume that we have a large list of entities that should be removed from splk-dsm, for instance due to an issue in Splunk which has led to the unwanted creation of a large number of sourcetypes.

Step 1: Get entities

```
| inputlookup trackme_dsm_tenant_mytenant | eval keyid=_key
``` in this example, we want to delete all entities related to the main index, which␣
↪unexpecty was fed with a large number of unwanted entities ```
| search data_index="main"
```

Step 2: Collect keys and turn into a CSV list

```
| inputlookup trackme_dsm_tenant_mytenant | eval keyid=_key
``` in this example, we want to delete all entities related to the main index, which␣
↪unexpecty was fed with a large number of unwanted entities ```
| search data_index="main"
| table keyid
| stats values(keyid) as keys
| eval keys=mvjoin(keys, ",")
```

Step final: Delete entities

```
| inputlookup trackme_dsm_tenant_mytenant | eval keyid=_key
``` in this example, we want to delete all entities related to the main index, which␣
↪unexpecty was fed with a large number of unwanted entities ```
| search data_index="main"
| table keyid
| stats values(keyid) as keys
| eval keys=mvjoin(keys, ",")
``` finally call the endpoint, which supports multiple entities deletion using the␣
↪Splunk map command```
| map search="| trackme url=/services/trackme/v2/splk_dsm/write/ds_delete mode=post␣
↪body=\"{'tenant_id':'mytenant', 'deletion_type': 'temporary', 'keys_list':'$keys$'}\
↪""
```

Run the search above, if dealing with a very large number of entities, this search can take some time to complete, send the search to background and monitor the progress.

8.11.3 Mass enabling/disabling entities

Here is a customer use case:

- TrackMe is configured to automatically disable the monitoring state when new entities are discovered.

- The purpose is to avoid creating noise until we have enough historical knowledge of the feeds' behaviors, so that TrackMe's features such as the adaptive delay will eventually adapt the delay values if needed.
- Then, a custom scheduled search inspects entities, and when we have reached a state of having enough historical knowledge, disabled entities are enabled automatically.

Configuring TrackMe to disable the monitoring status at the time of the feed discovery:

The default monitored state is handled via a Splunk macro in TrackMe, which definition in macros.conf is as follows:

```
[trackme_default_monitored_state]
definition = eval monitored_state=if(isnull(monitored_state), "enabled", monitored_
↪state)
iseval = 0
```

To disable the monitoring state at the time of the feed discovery, you can override the macro definition to:

```
eval monitored_state=if(isnull(monitored_state), "disabled", monitored_state)
```

Creating a scheduled report which leverages TrackMe's metrics to define a confidence status, then call map and TrackMe's REST API to enable entities:

The following search would be scheduled at your convenience, and handle entities enabling based on a confidence status:

Hint

Replace tenant:

- Replace "mytenant" with the tenant ID value.

```
| mstats latest(trackme.splk.feeds.lag_event_sec) as lag_event_sec where␣
↪index=trackme_metrics tenant_id="mytenant" object_category="splk-dsm" by object␣
↪span=1d

| stats min(_time) as first_time by object
| eval metrics_duration=now()-first_time
| eval confidence=if(metrics_duration<(7*86400), "low", "normal")
| where confidence="normal"

| lookup trackme_dsm_tenant_mytenant object OUTPUT _key as keyid, monitored_state,␣
↪priority
| where (isnotnull(keyid) AND monitored_state="disabled")

|  stats values(keyid) as keys
|  eval keys=mvjoin(keys, ",")

|  map search="| trackme url=\"/services/trackme/v2/splk_dsm/write/ds_monitoring\"␣
↪mode=\"post\" body=\"{'tenant_id': 'mytenant', 'action': 'enable', 'keys_list': '
↪$keys$'}\""
```

8.11.4 Maintaining entities priority in a Splunk lookup file

A frequent question from TrackMe users is how they can maintain the priority of entities in a lookup file, and how to update the priority of entities in bulk.

In this example, we will demonstrate how to update the priority of entities in bulk for splk-dhm (Data Hosts tracking) component.

Step 1: Prepare the lookup file

Create a lookup file which contains at the minimum the following fields:

- host: the machine hostname as it is seen in Splunk (the host metadata)
- priority: the priority value you want to set for the entity.

While it is not fully mandatory, calling the lookup through a lookup definition is recommended so you can run a case-insensitive lookup call, you should be able to run:

```
| inputlookup hosts_lookup_cmdb
```

About case-sensitive match and lookup share

- Ensure to create a definition for the lookup so you can configure Splunk to allow case-insensitive lookup matches.
- Also, make sure both the lookup file and lookup permissions are set to be shared so TrackMe can access these.

Step 2: Create a report that will be scheduled and will update and maintain the priority of entities based on the lookup content

This is really simple, TrackMe stores the priority in the main KVstore collection, updating it based on a third party lookup can therefore be handled through a simple SPL logic:

Hint

Replace tenant:

- Replace "mytenant" with the tenant ID value.

Hint

About the alias field and components:

- In the example below, we use the TrackMe internal field called `alias`.
- In the context of splk-dhm, the alias will be the exact host value as seen in Splunk.
- In fact, TrackMe stores the unique object value in a field called object and is prefixed with a TrackMe metadata *key:* `, which is why we use the alias field instead to perform the lookup call.

```
| inputlookup trackme_dhm_tenant_mytenant | eval keyid=_key
| lookup hosts_lookup_cmdb host as alias OUTPUT priority as lookup_priority
| table keyid, lookup_priority, priority, *
| where isnotnull(lookup_priority)
| where priority!=lookup_priority
``` At this stage only entities to be managed will be updated ```
| eval priority=lookup_priority, mtime=now()
``` Update ```
| outputlookup append=t key_field=keyid trackme_dhm_tenant_mytenant
```

Save this search as a report and schedule it at your convenience, it doesn't need to be executed very often so once every 4 hours for instance is likely enough.

Finally, note that this report will not return any results unless a priority update is needed, not returning any results simply indicates that all entities are up to date according to the lookup file.

Maintaining priority through a third party lookup would override any change made through the UI or REST API:

- Note that this logic above would override a priority change a TrackMe power user would make through the UI or REST API.
- Therefore, this needs to be known and documented to avoid any risks of confusion.

About other components:

- The logic or variations of this logic can be applied to any TrackMe component.
- Simply replace the component suffix, in this example `_dhm` with the associated component suffix. (dsm, mhm, flx, wlk, cim)

8.11.5 Mass updating entities priority

The following example demonstrates how to mass update entities priority in TrackMe:

Hint

Replace tenant:

- Replace "mytenant" with the tenant ID value.
- Replace the component, in this example we use splk-dsm, replace it with the component you want to update the priority for. (notably the KVstore collection prefix, dsm/dhm/mhm/flx/wlk/cim)

Step 1: Get entities

This example updates entities associated with indexes starting with "org", replace for your context

```
| inputlookup trackme_dsm_tenant_mytenant | eval keyid=_key
``` in this example, we want to update all entities related to indexes starting by␣
↪org* ```
| search data_index="org*"
```

Step 2: Collect keys and turn into a CSV list

```
| inputlookup trackme_dsm_tenant_mytenant | eval keyid=_key
``` in this example, we want to update all entities related to indexes starting by␣
↪org* ```
| search data_index="org*"
| table keyid
| stats values(keyid) as keys
| eval keys=mvjoin(keys, ",")
```

Step final: Update entities priority

```
| inputlookup trackme_dsm_tenant_mytenant | eval keyid=_key
``` in this example, we want to update all entities related to indexes starting by␣
↪org* ```
| search data_index="org*"
```

(continues on next page)
```

(continued from previous page)

```
| table keyid
| stats values(keyid) as keys
| eval keys=mvjoin(keys, ",")
``` finally call the endpoint, which supports multiple entities using the Splunk map␣
↪command```
| map search="| trackme url=/services/trackme/v2/splk_dsm/write/ds_update_priority␣
↪mode=post body=\"{'tenant_id': 'mytenant', 'priority': 'high', 'keys_list': '$keys$
↪'}\""
```

Hint

API endpoints per component: (look into menu API & Tooling / TrackMe REST API Reference for all API endpoints documentation)

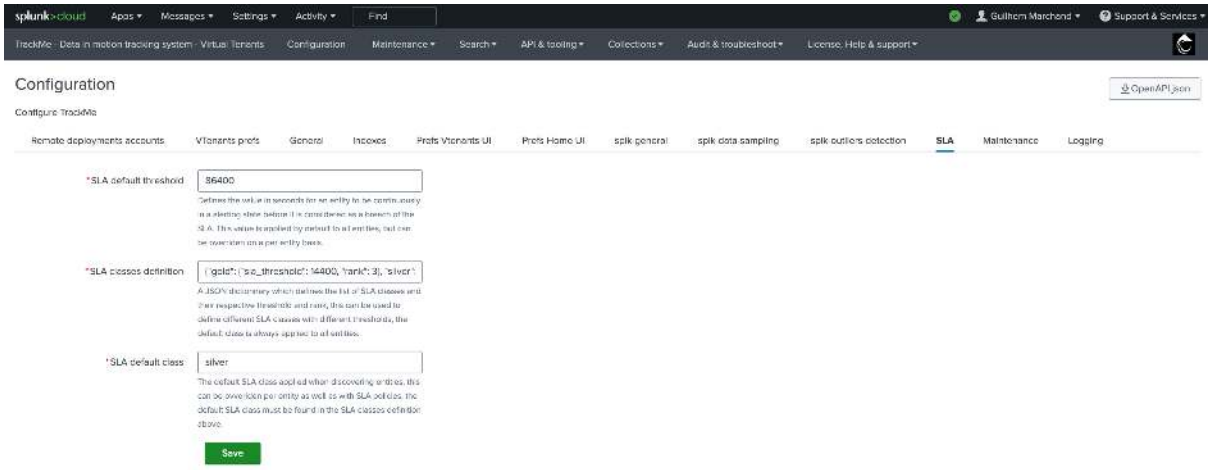
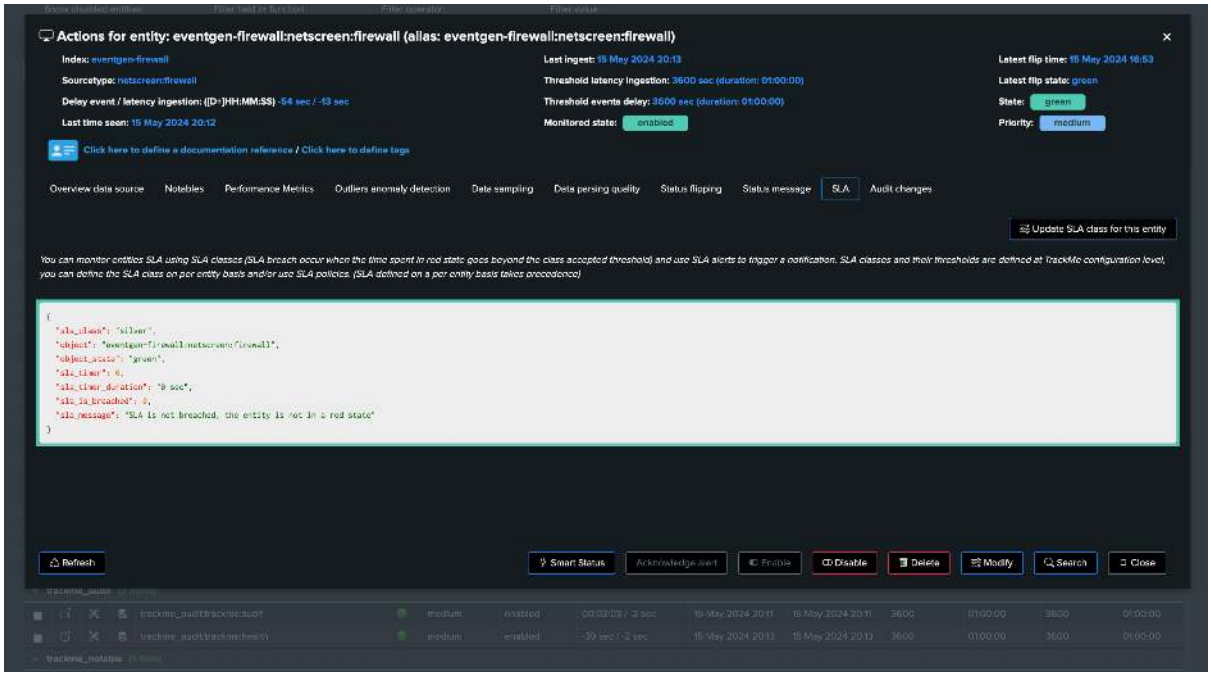
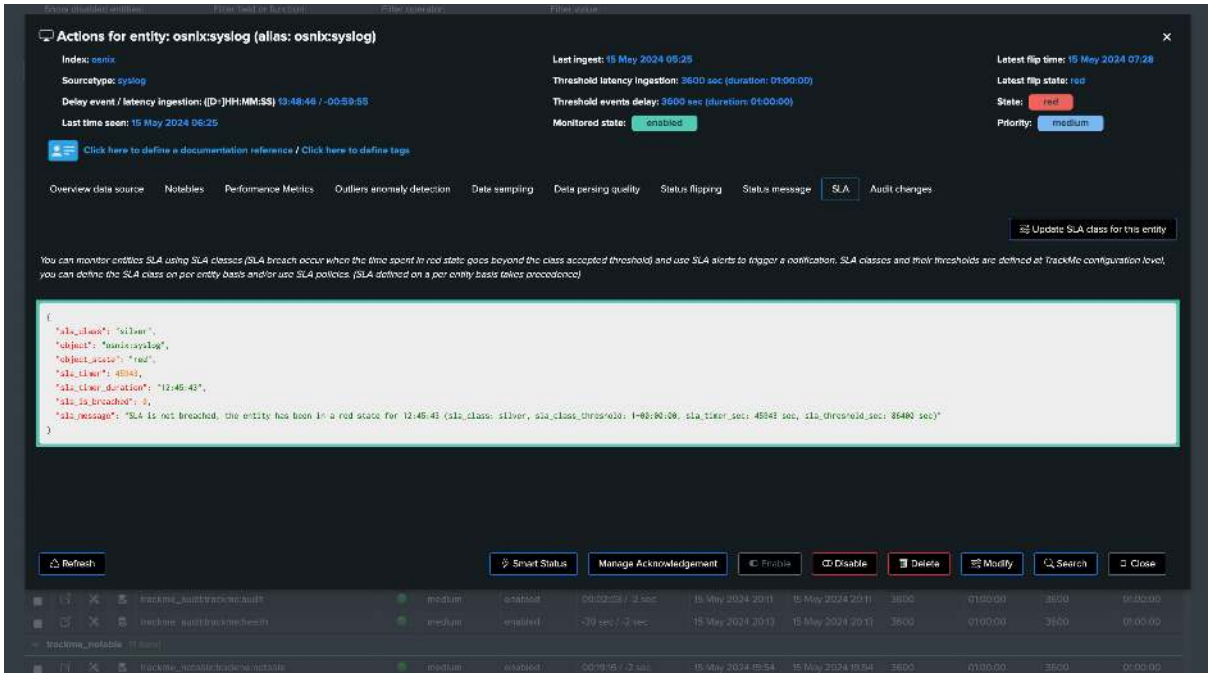
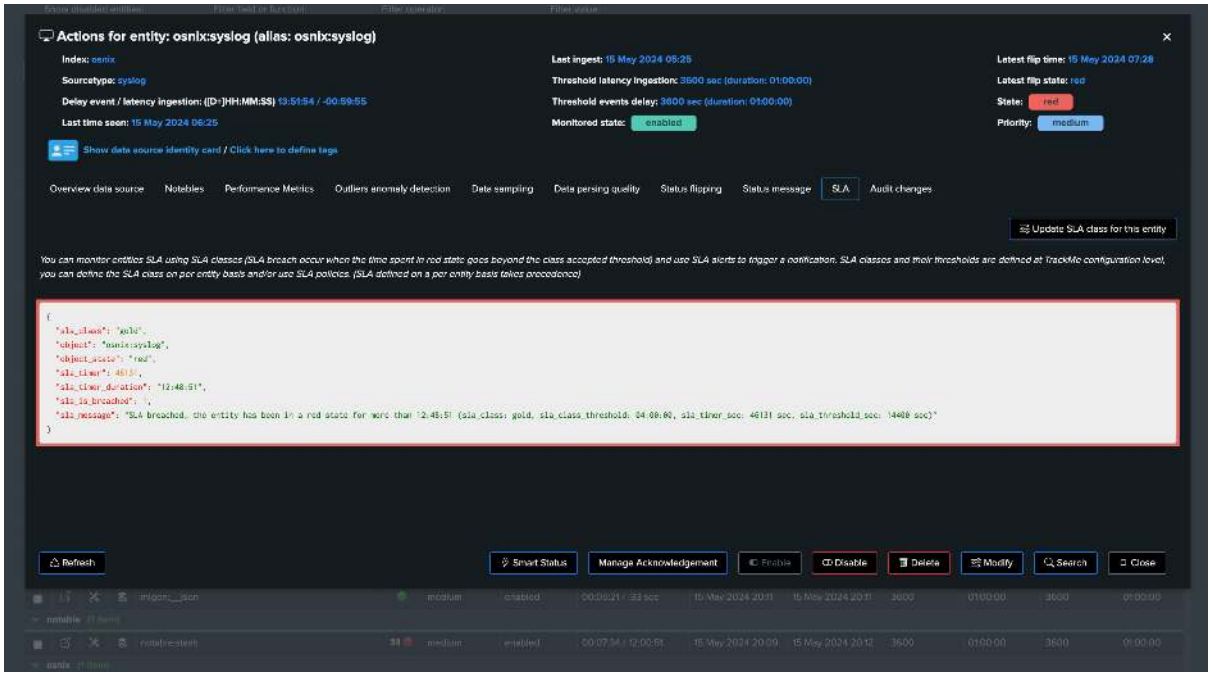
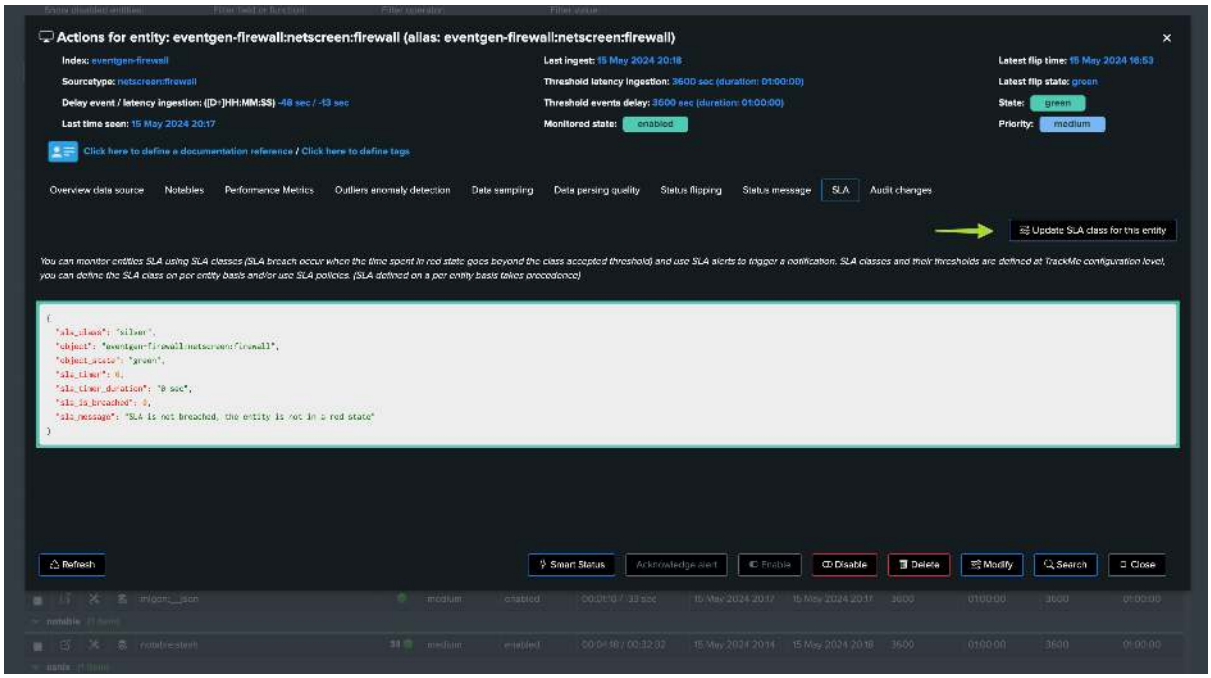
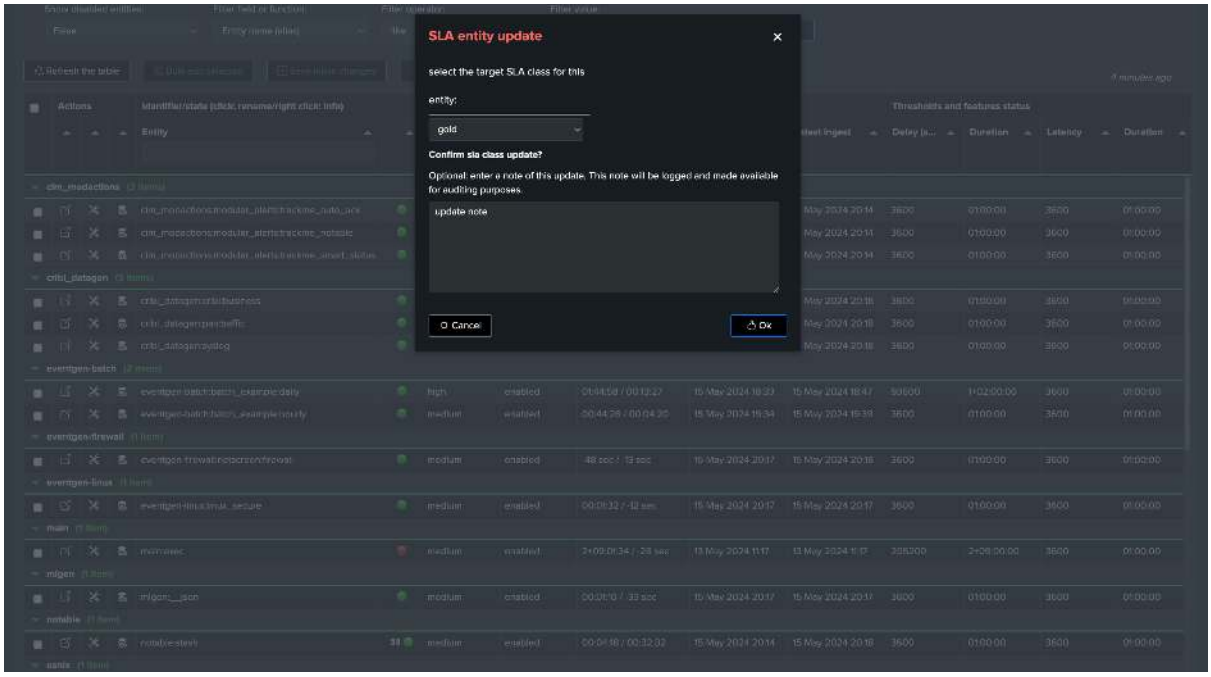
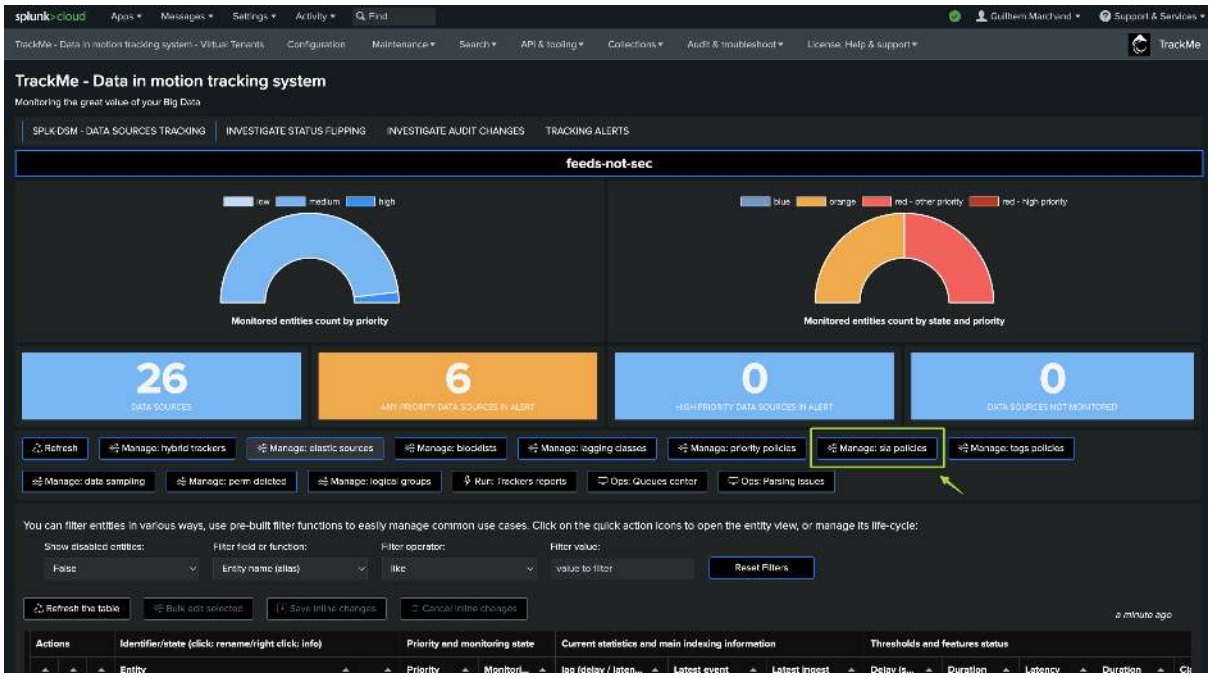
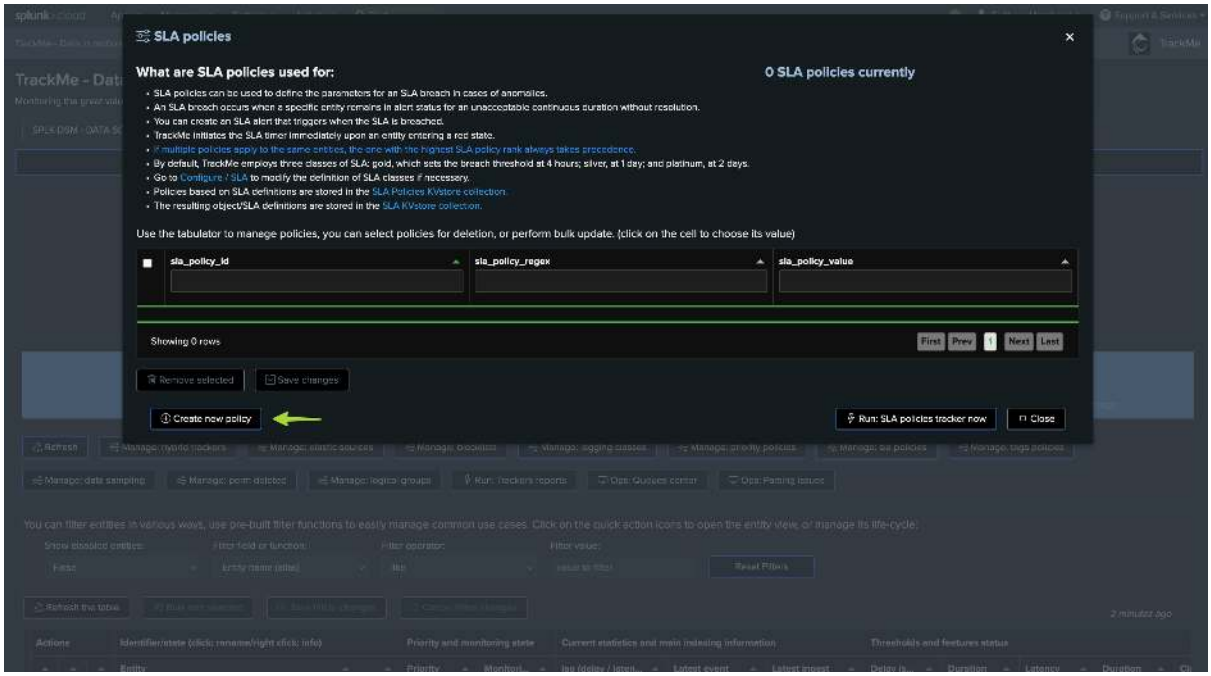
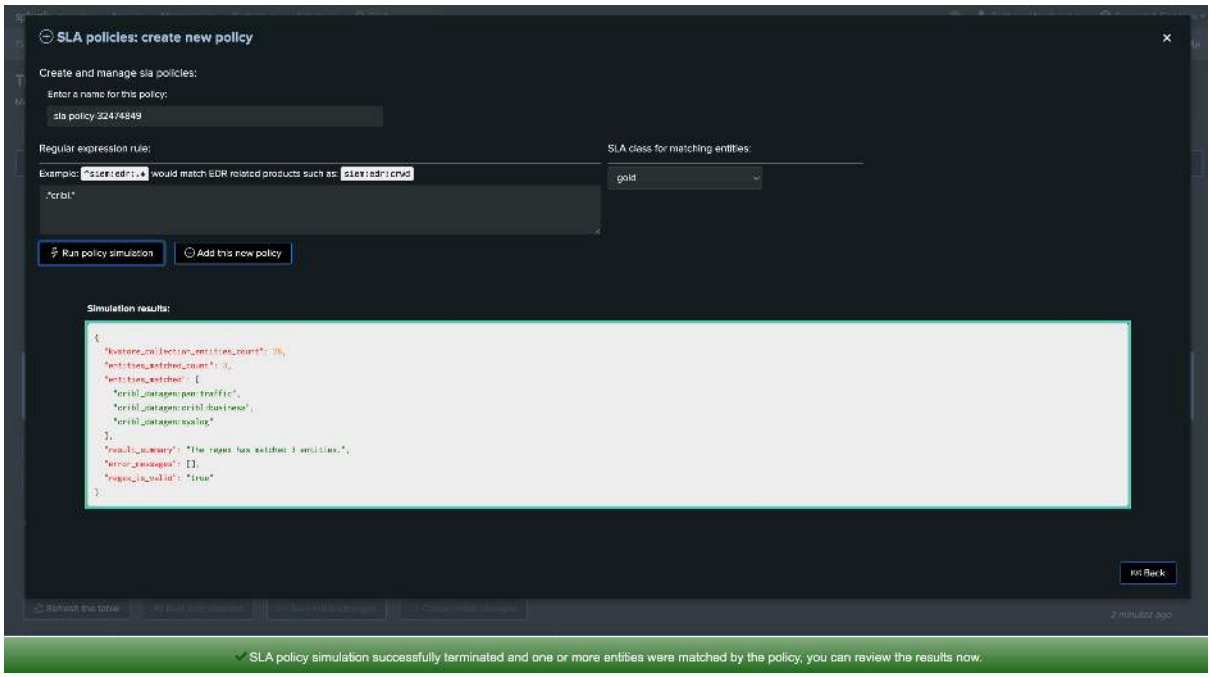
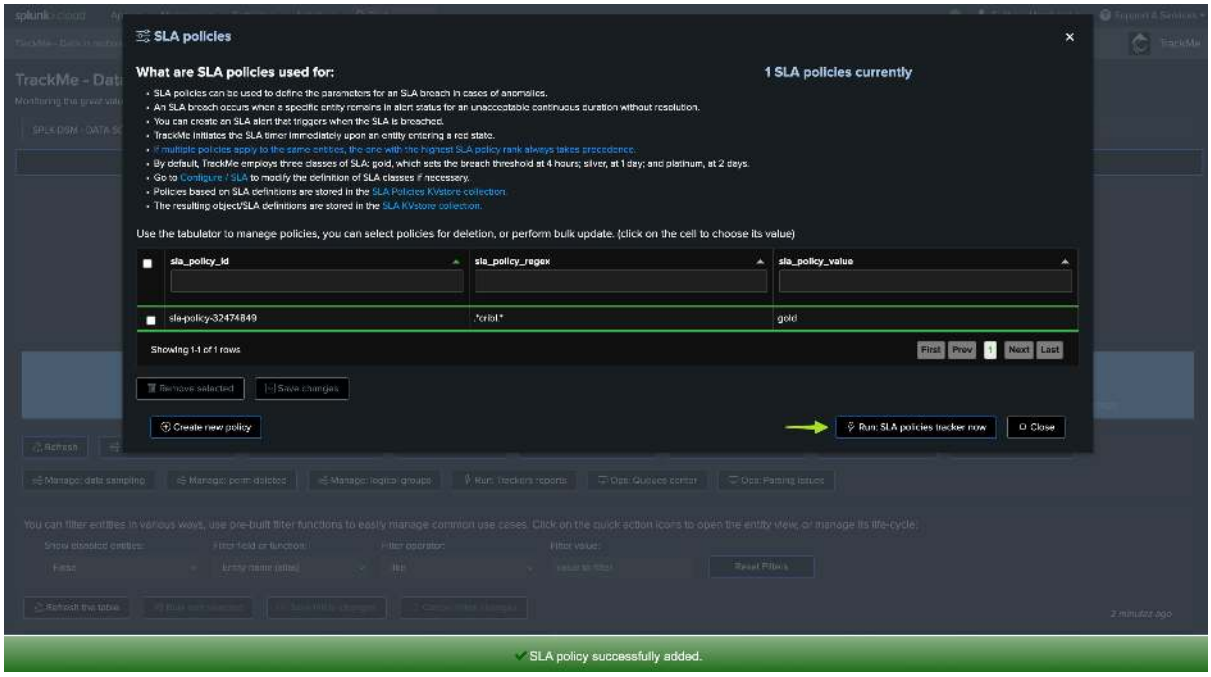
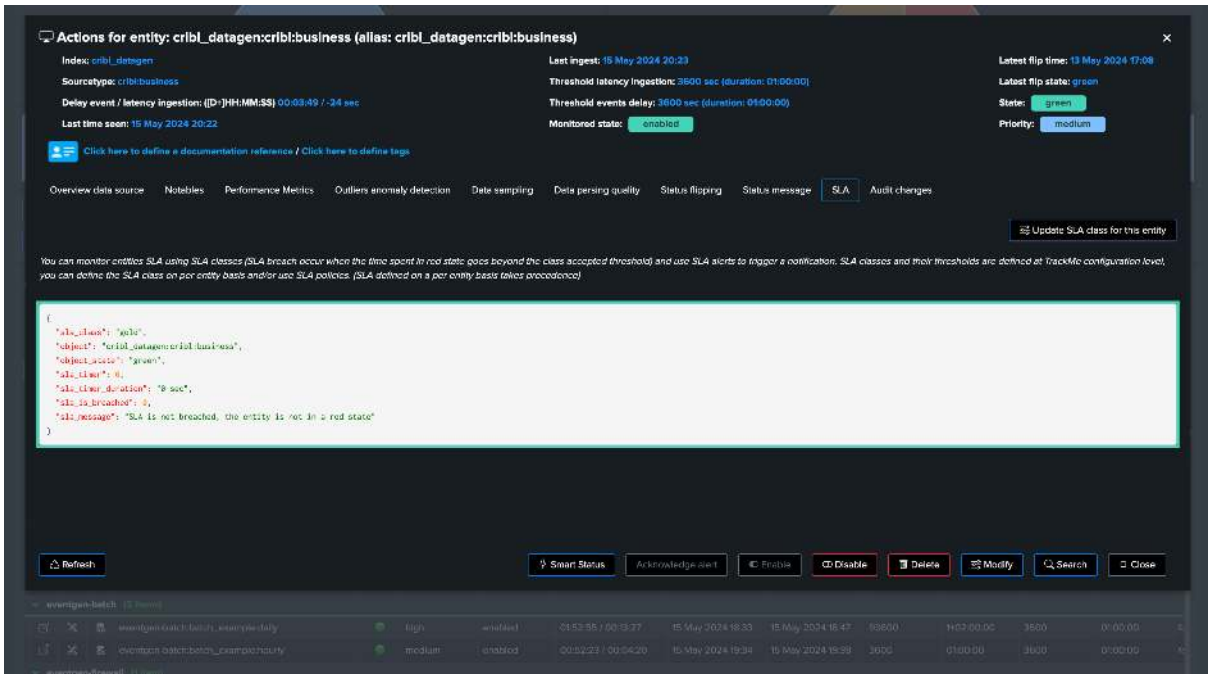
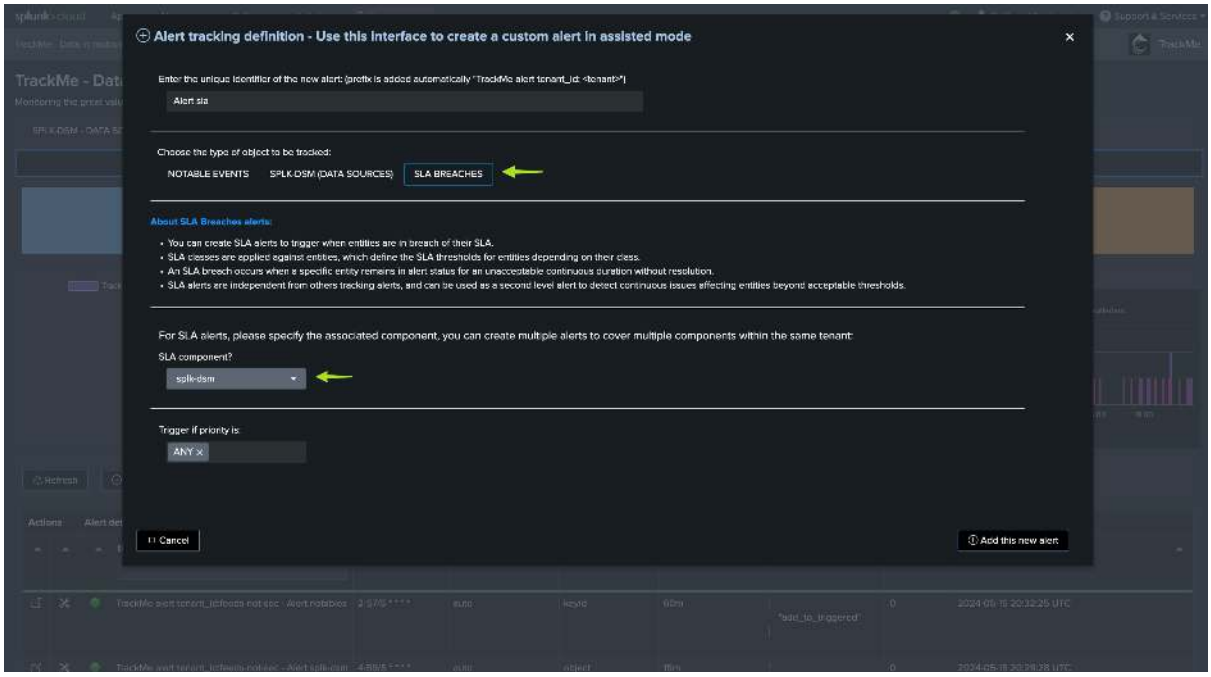
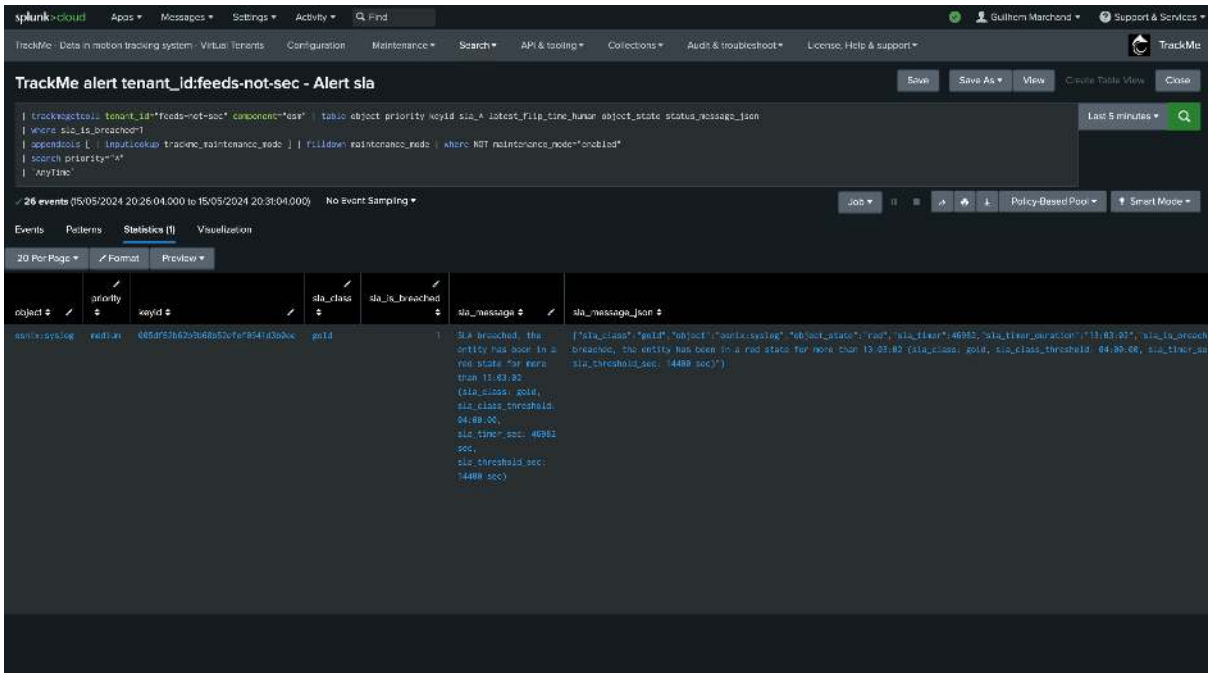
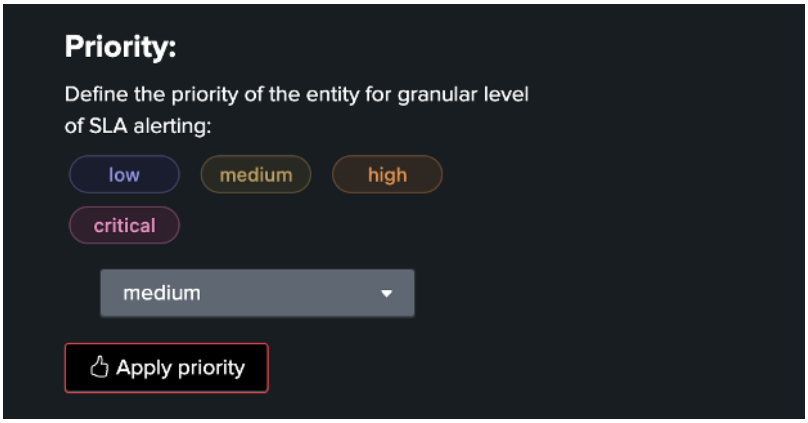
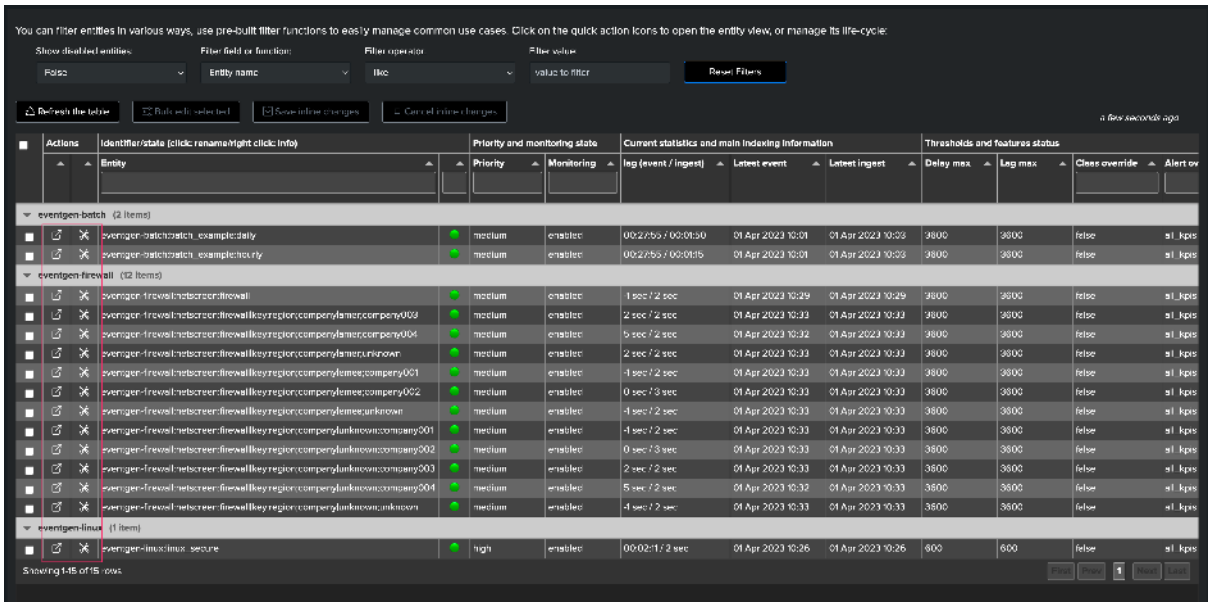
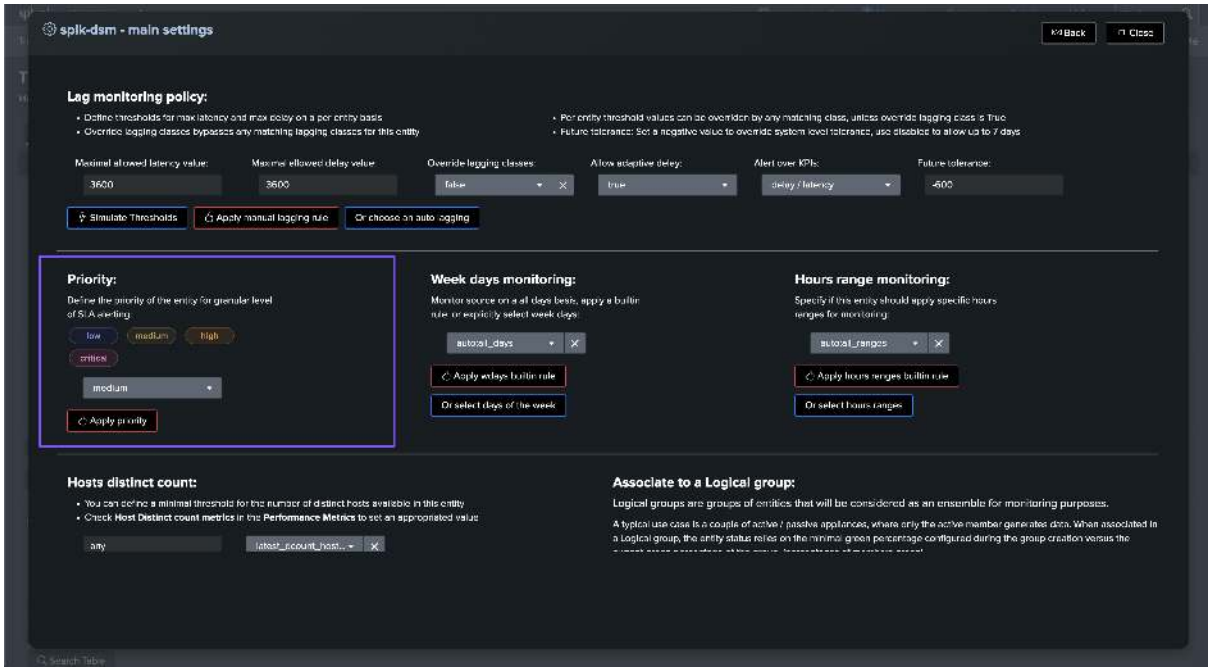
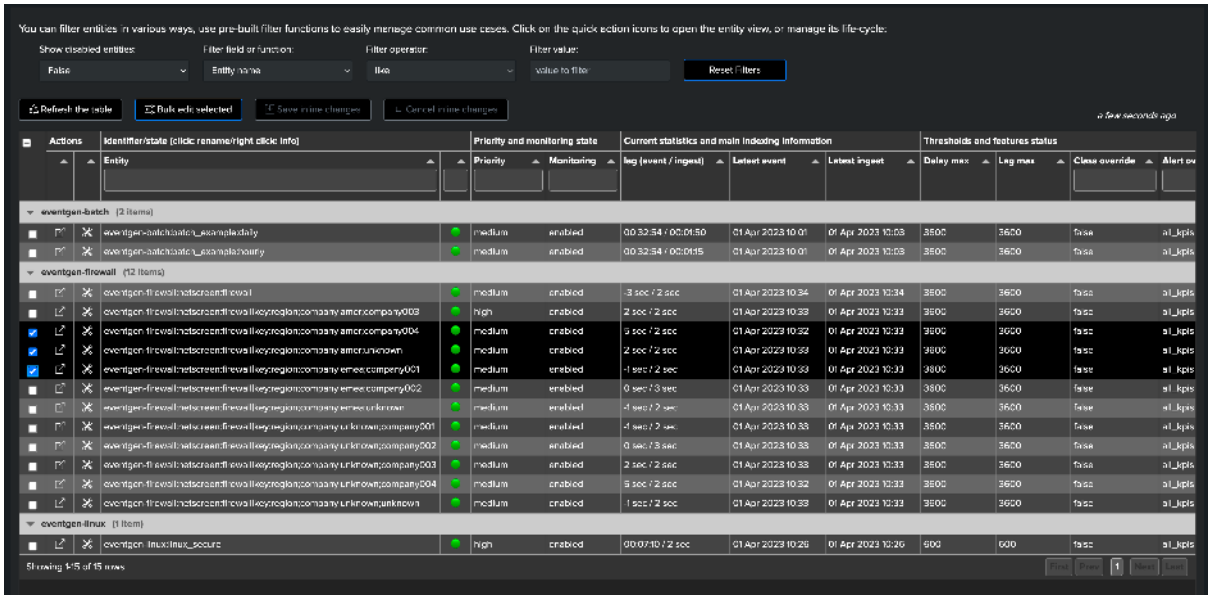
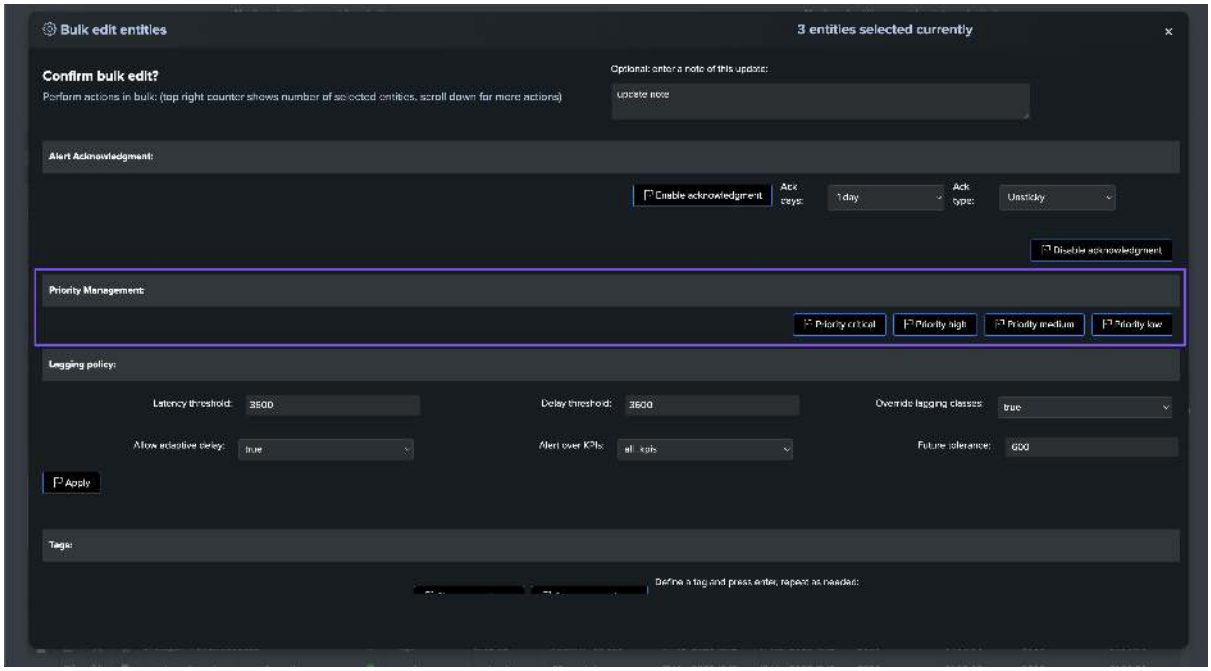
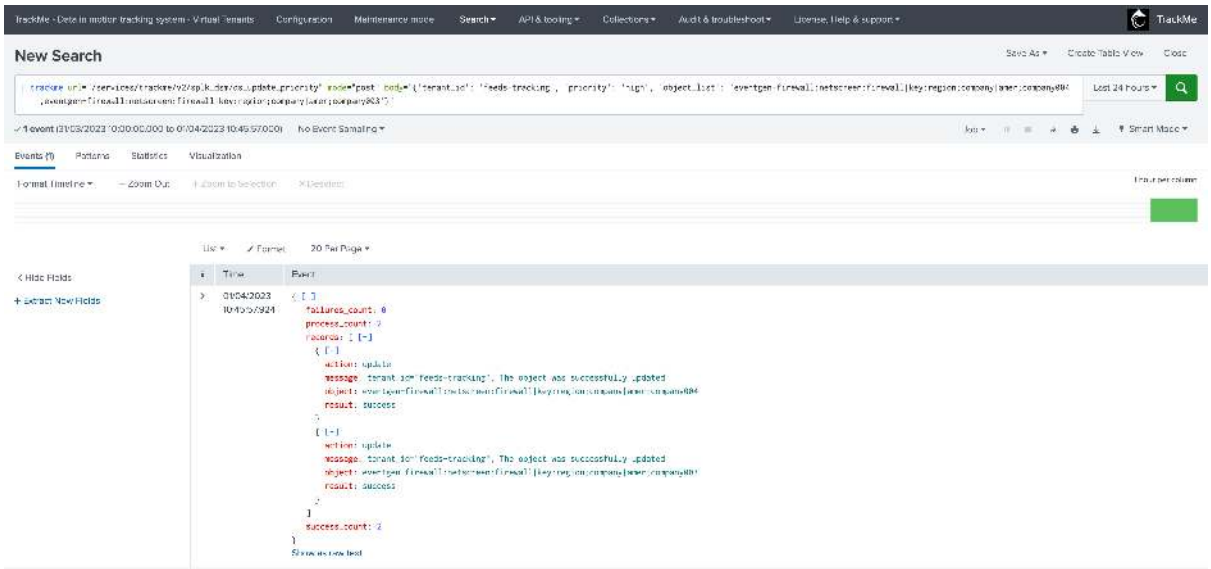
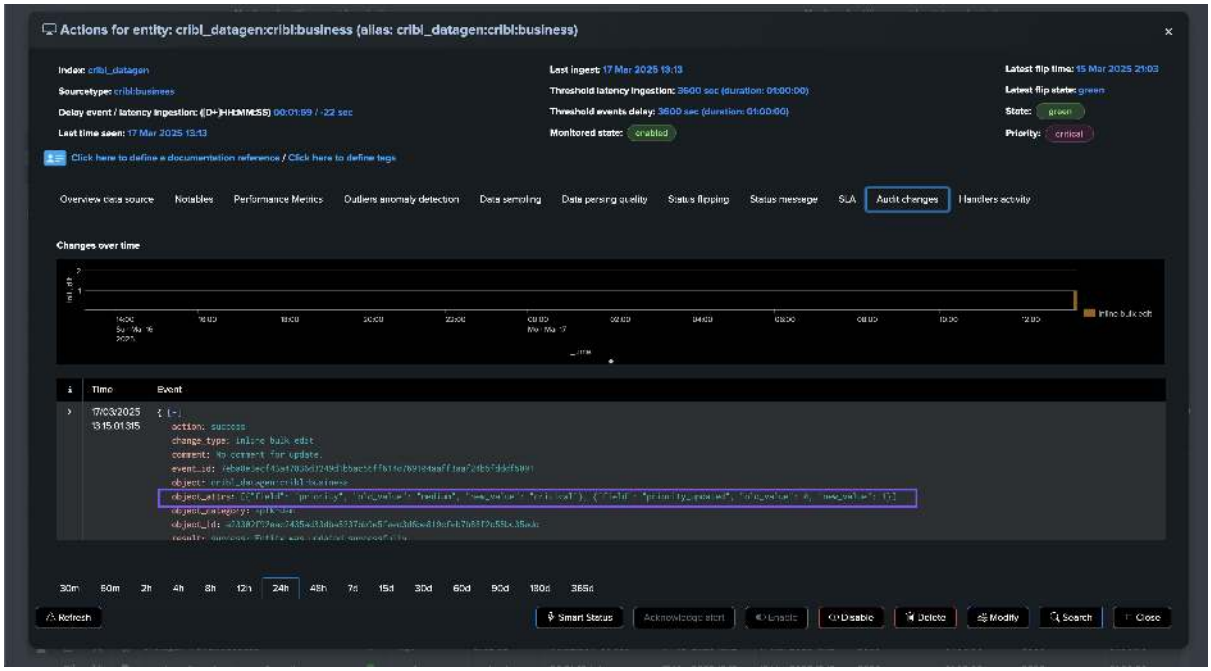
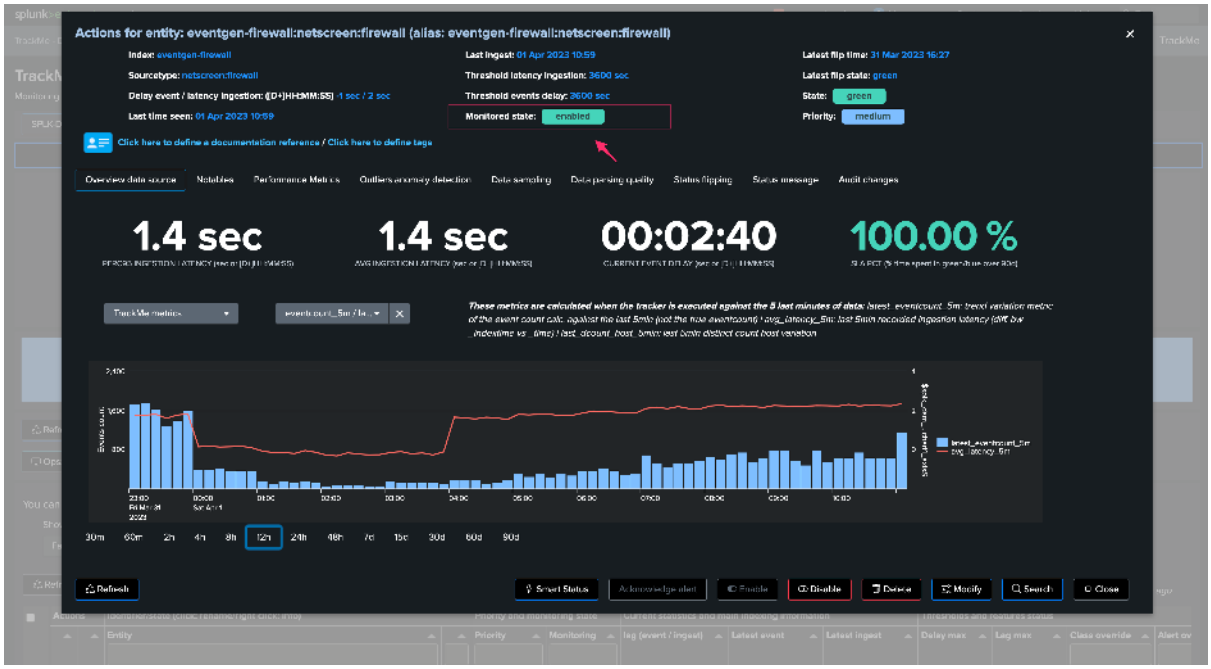
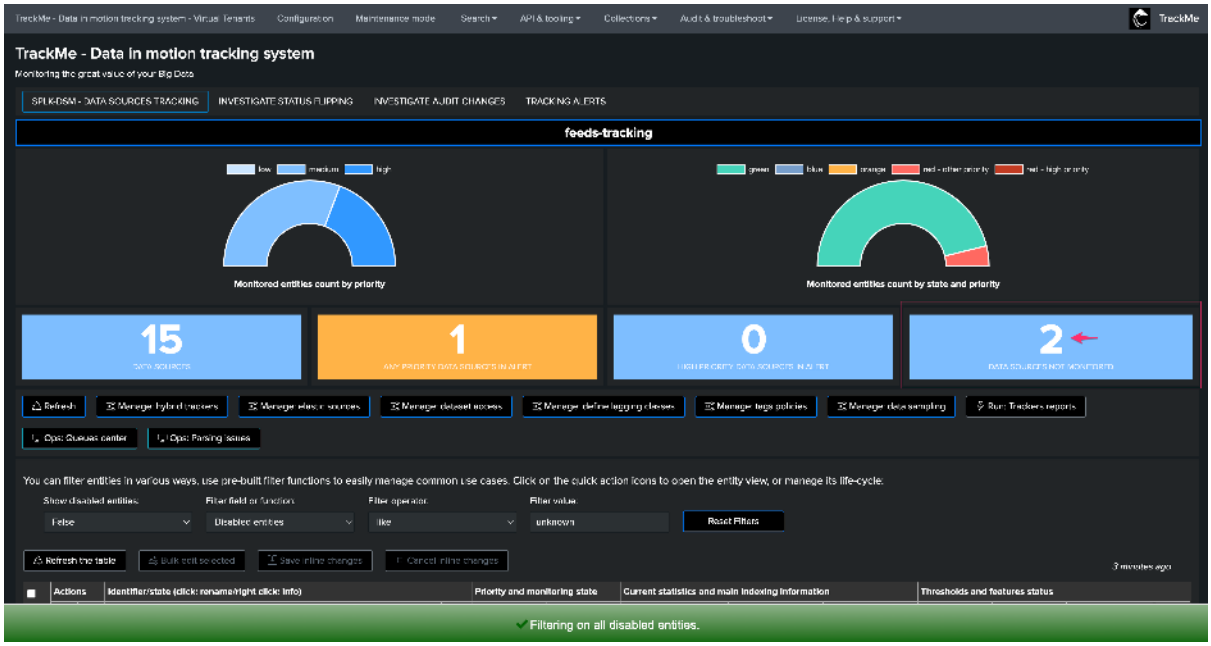
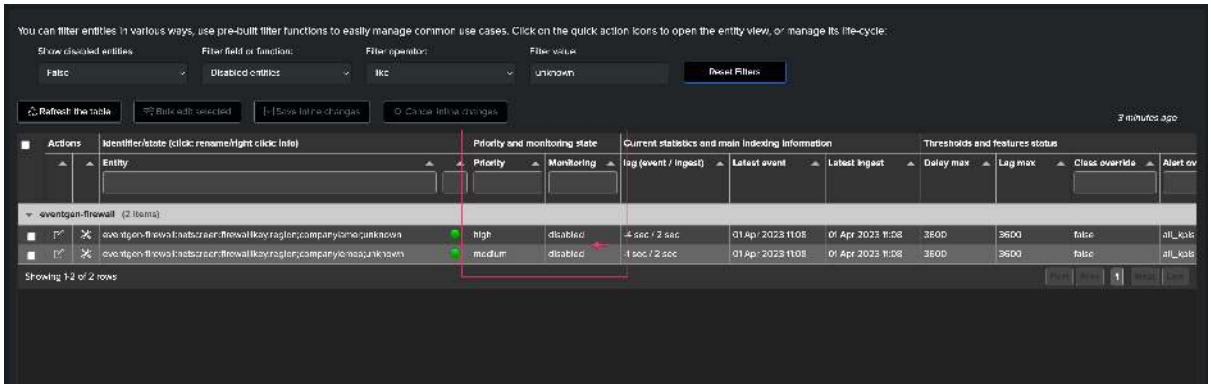
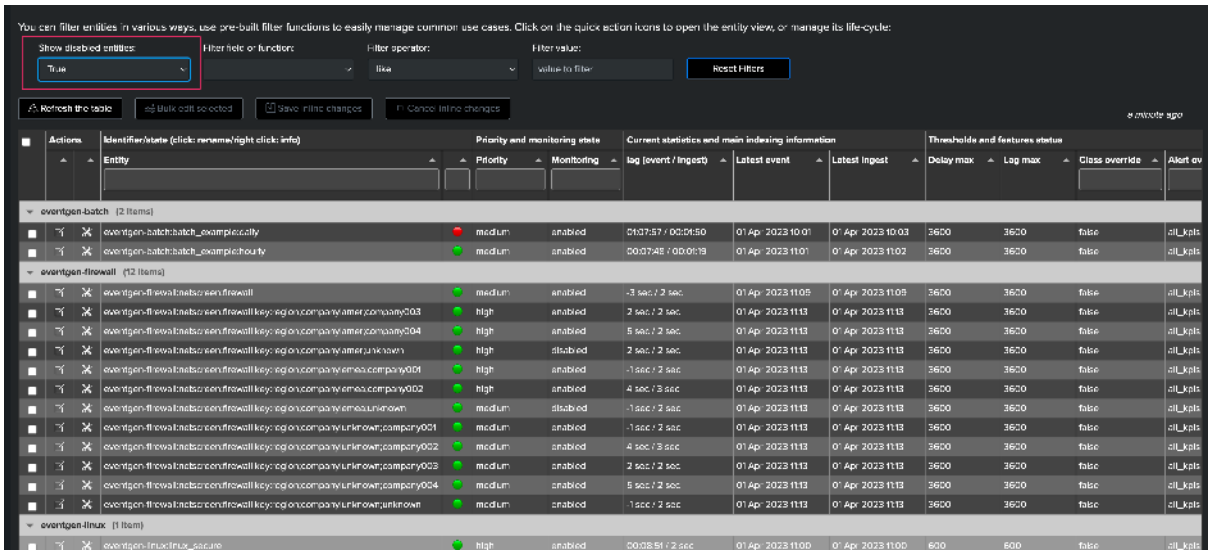
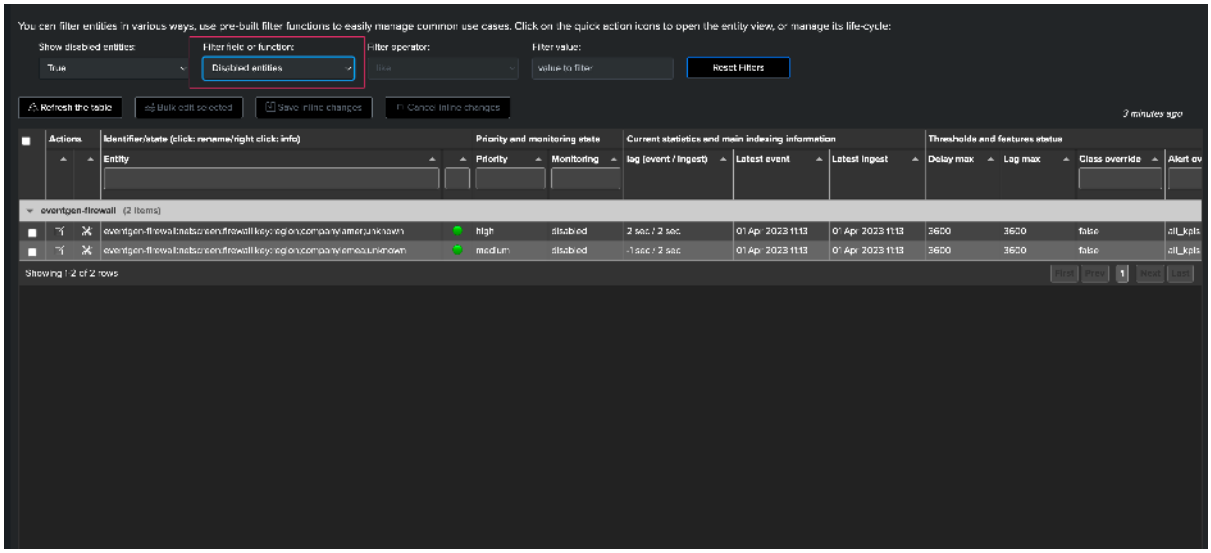
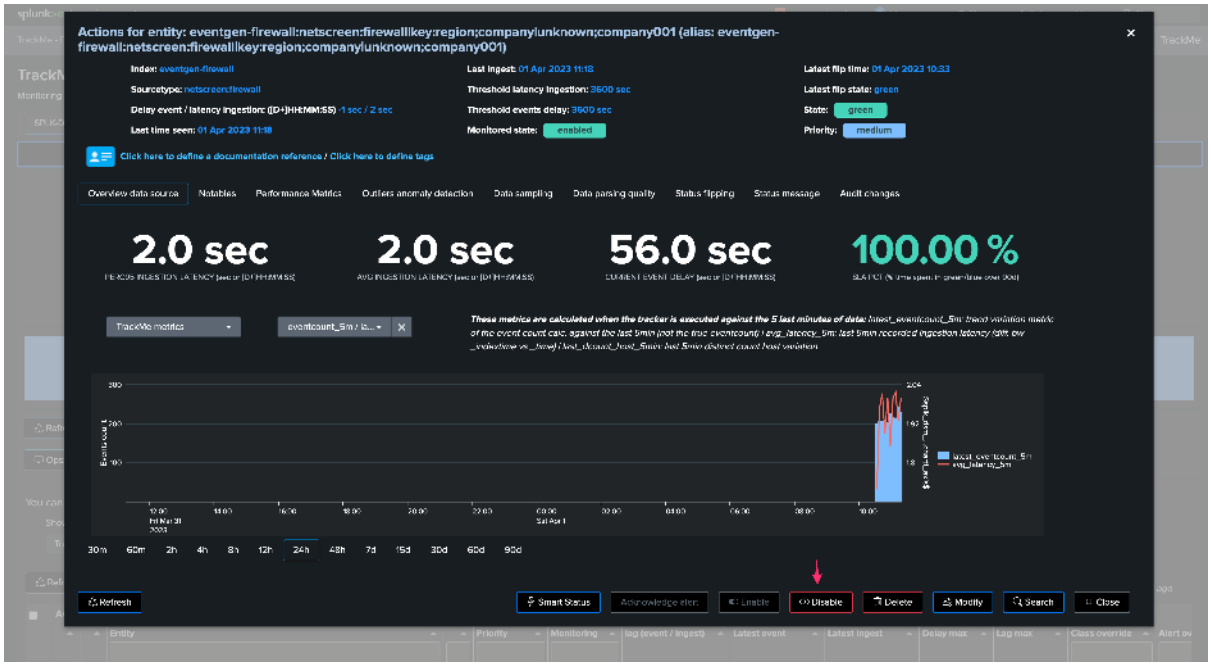
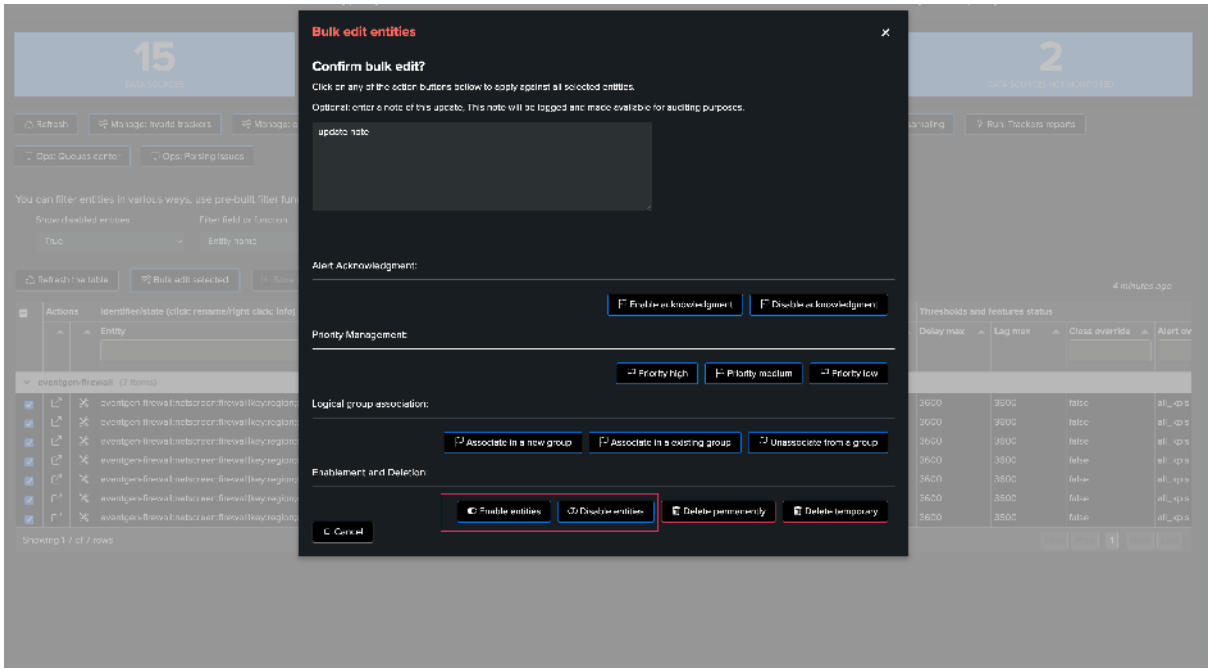
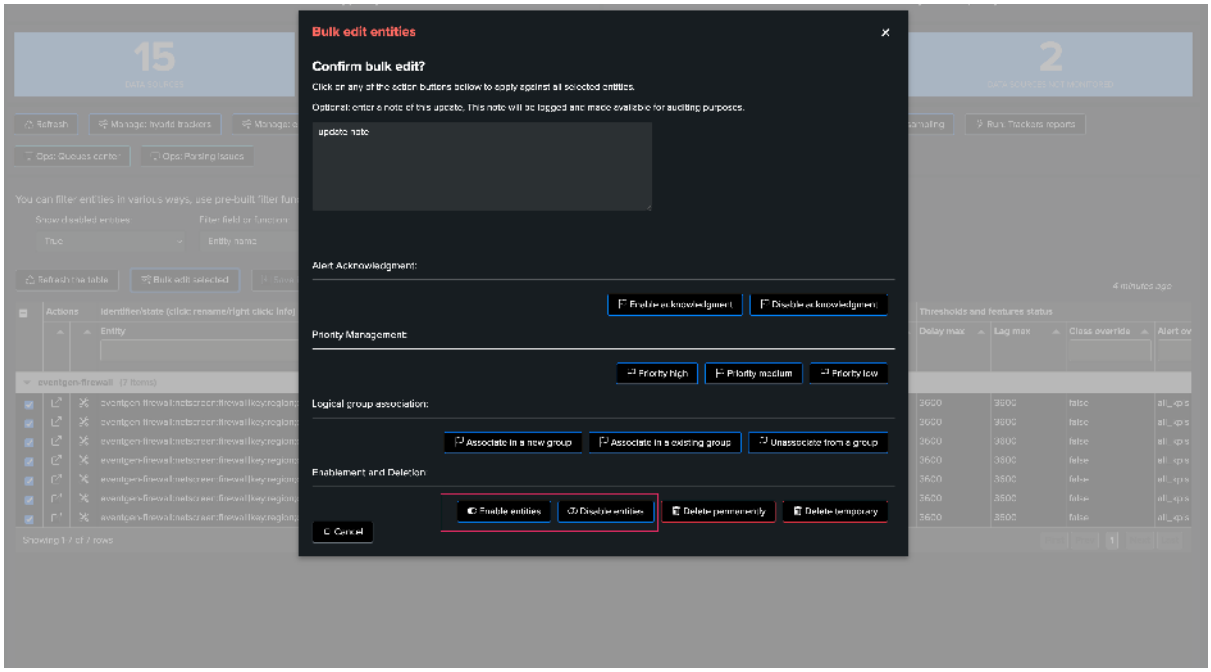
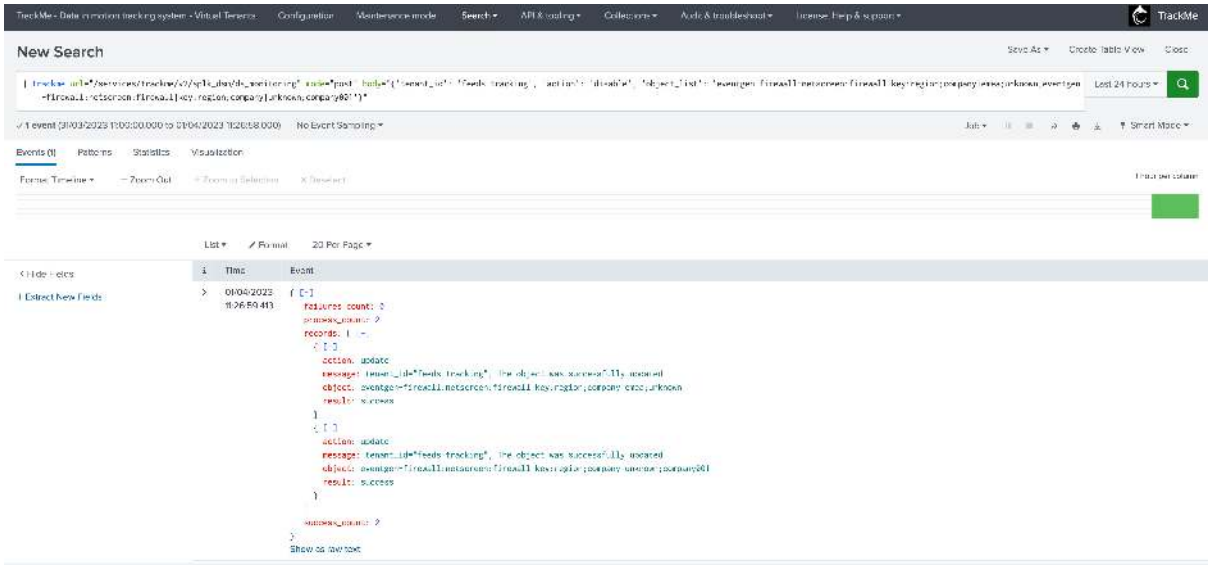
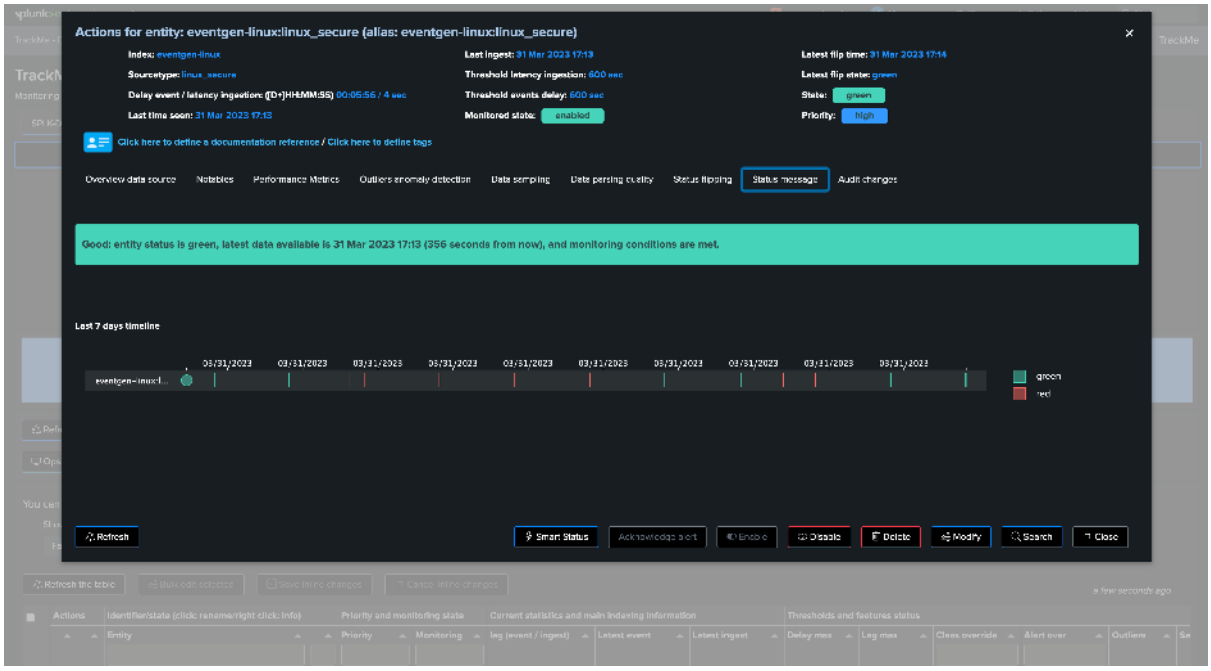
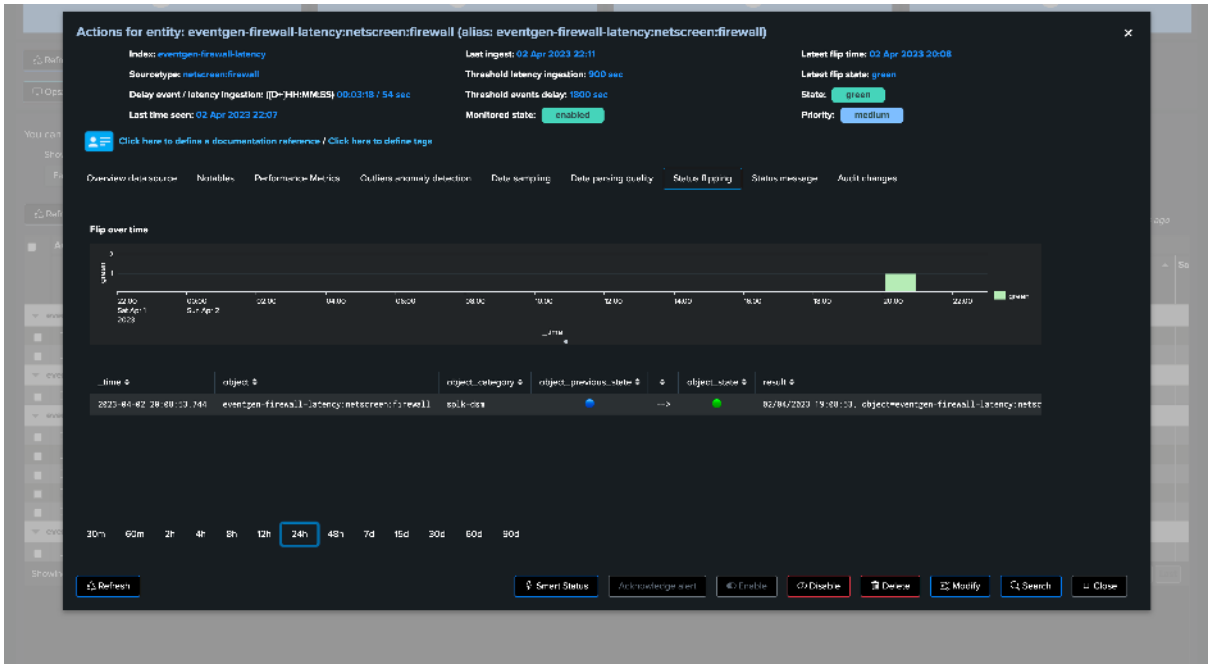
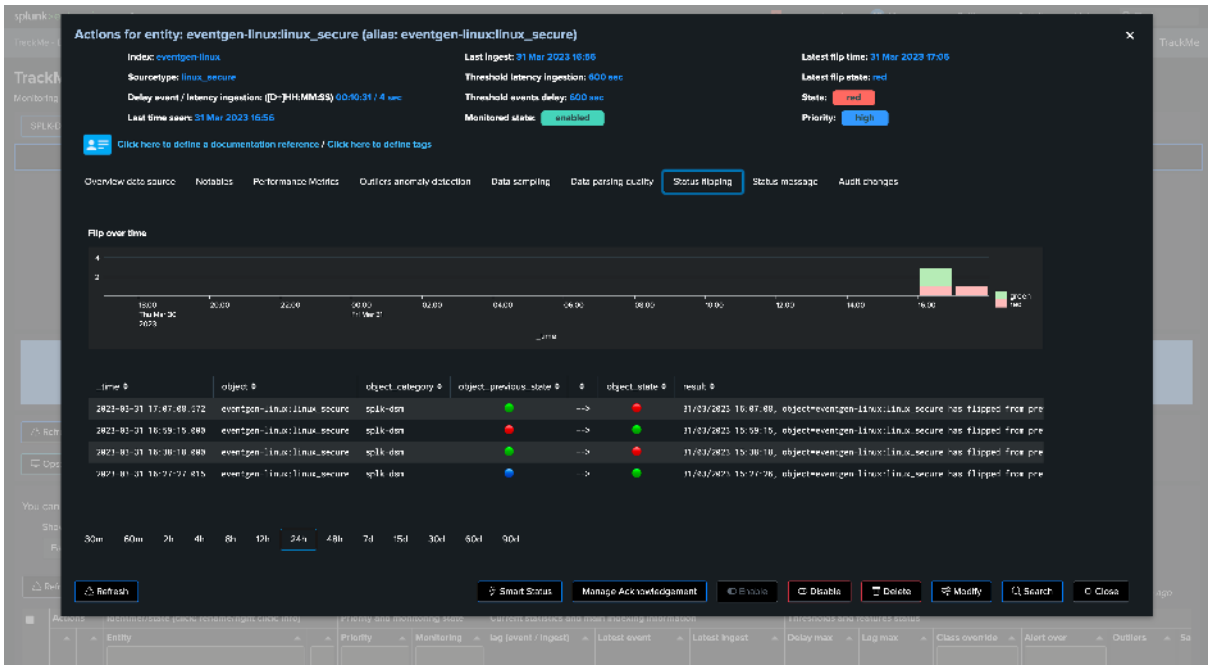
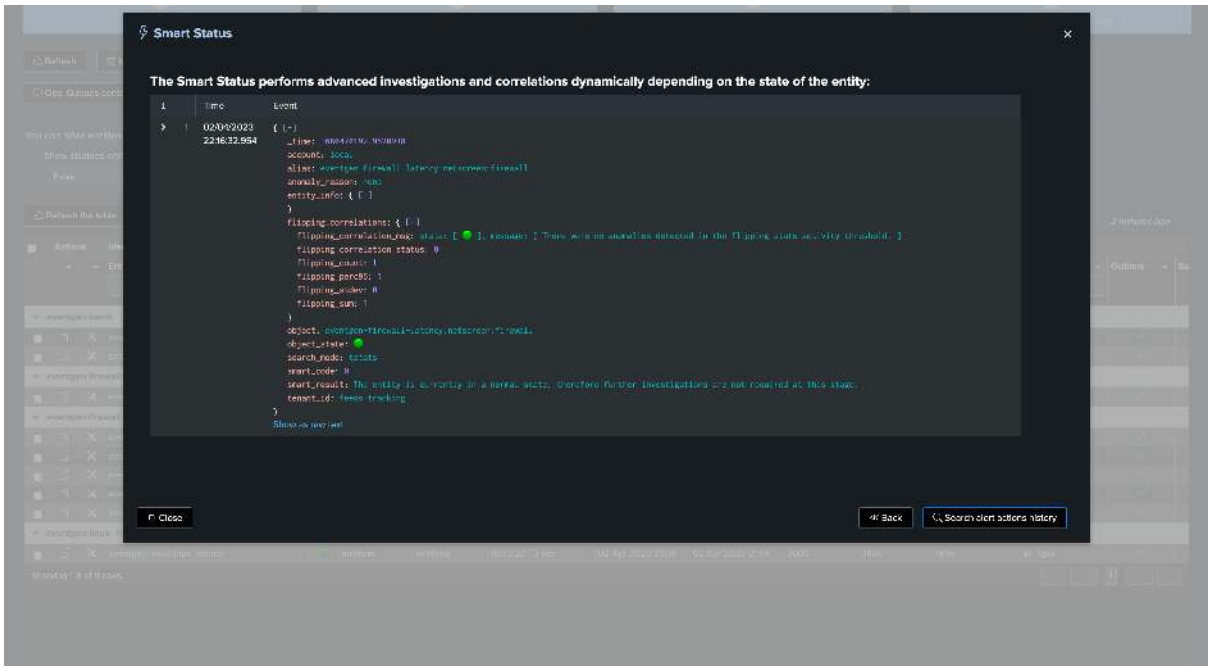
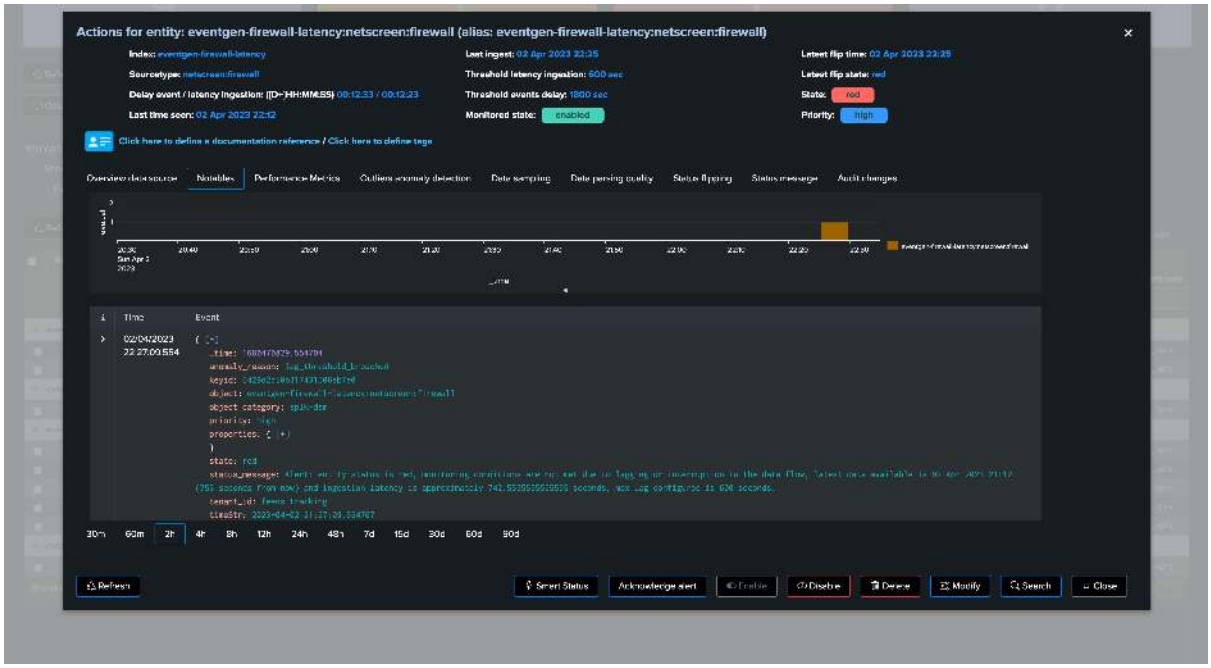
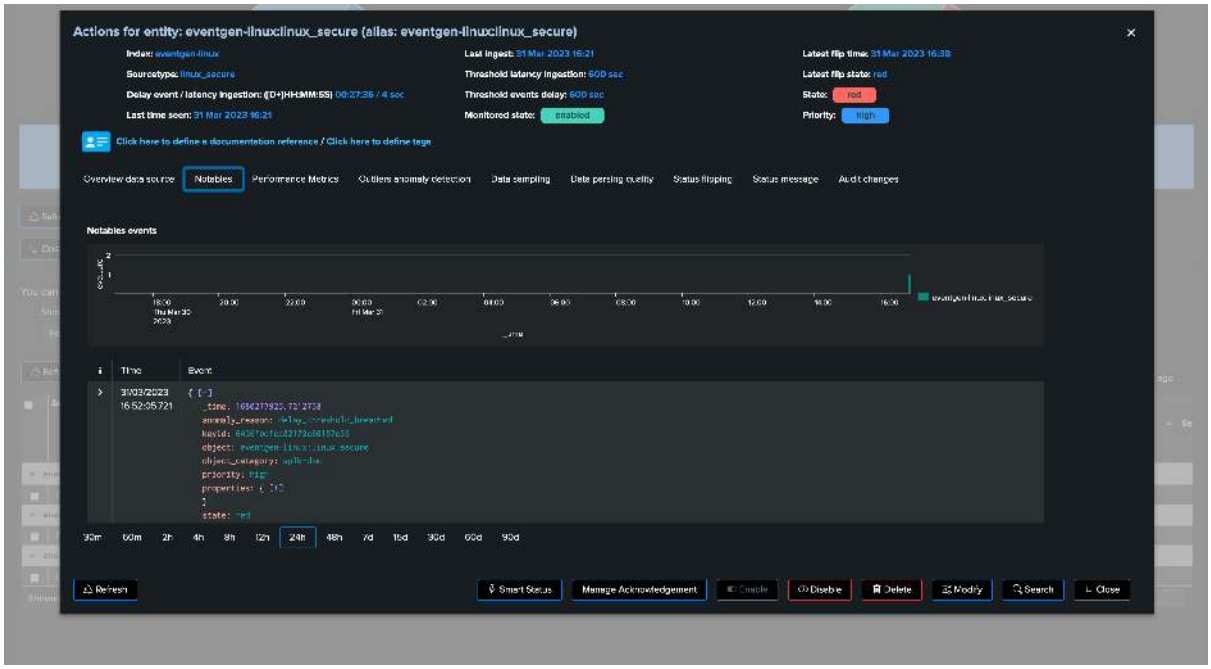
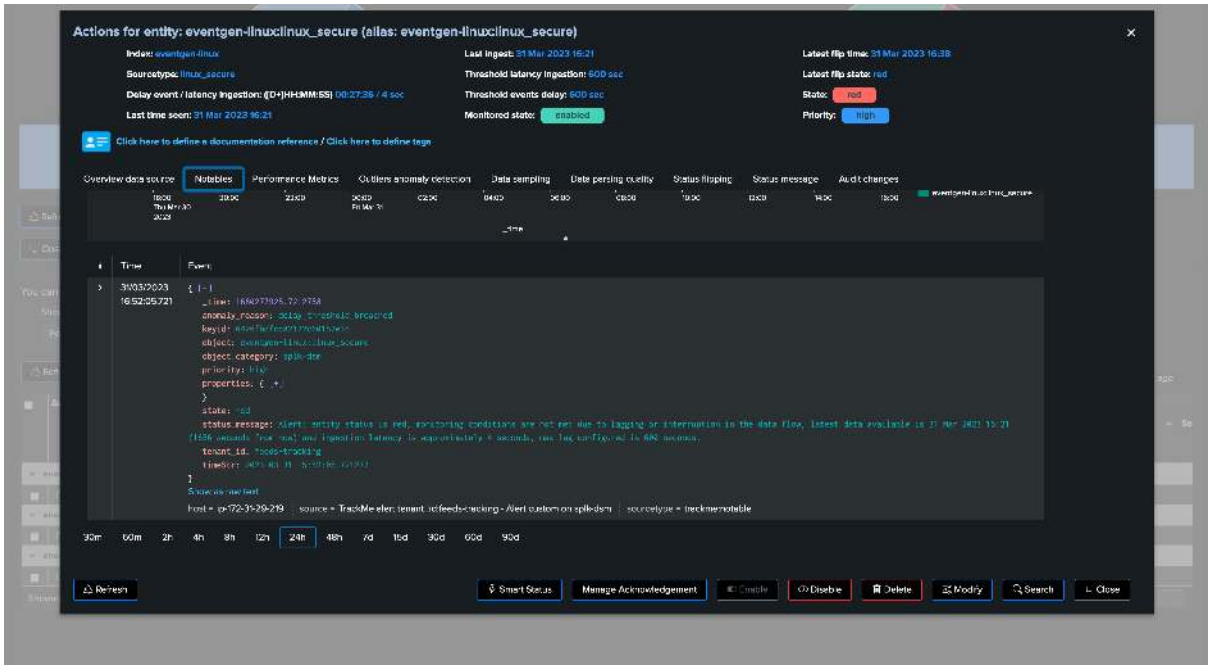
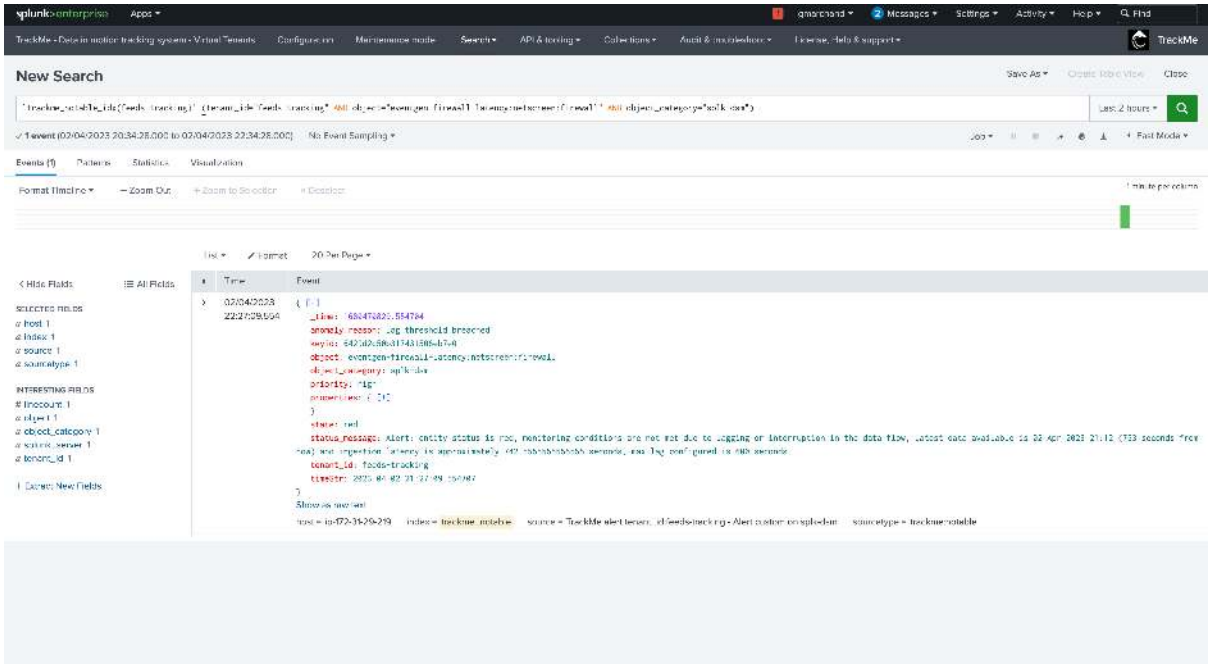
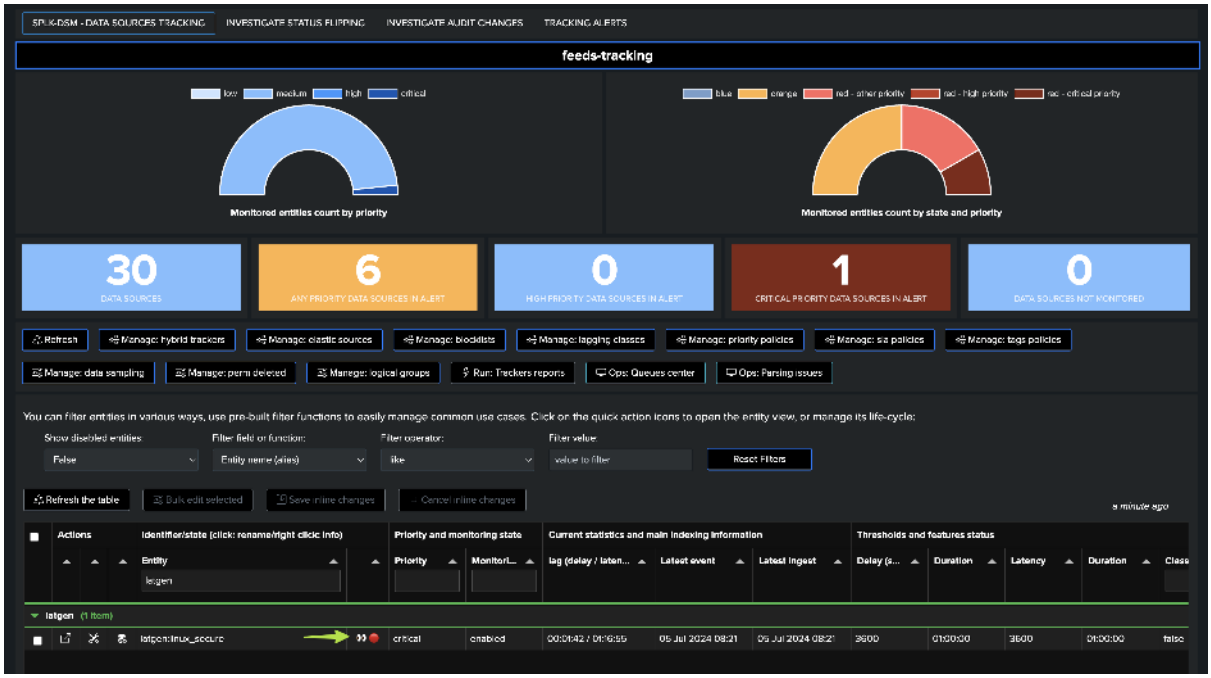
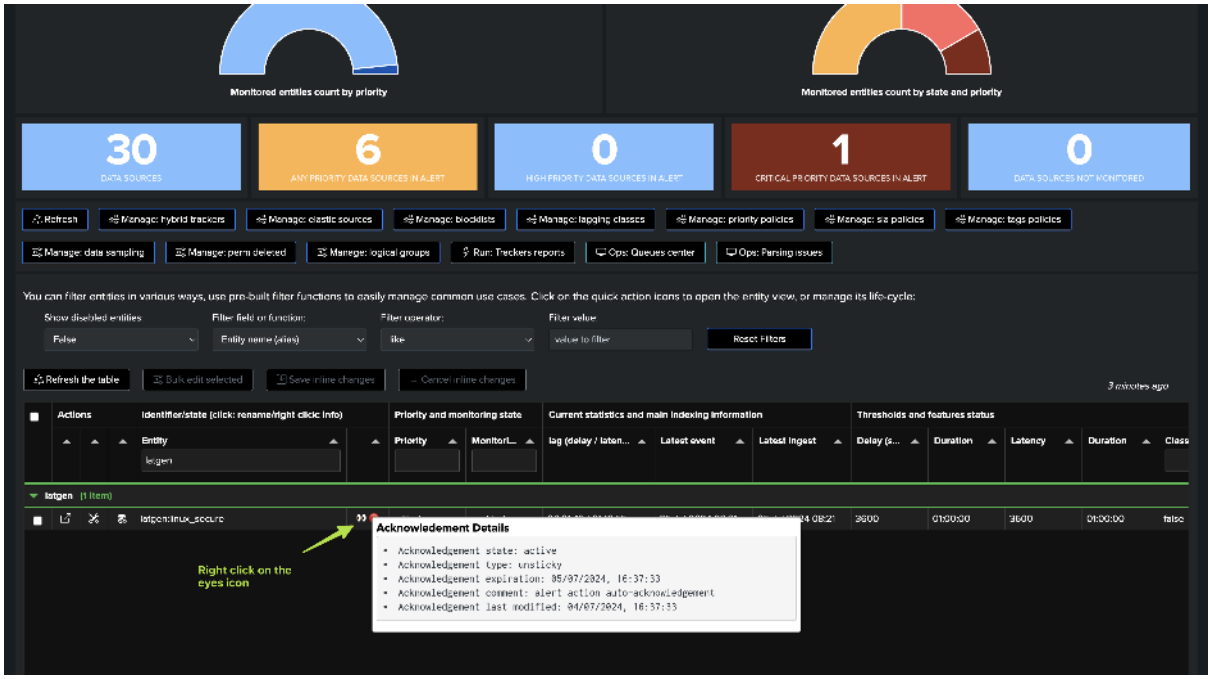
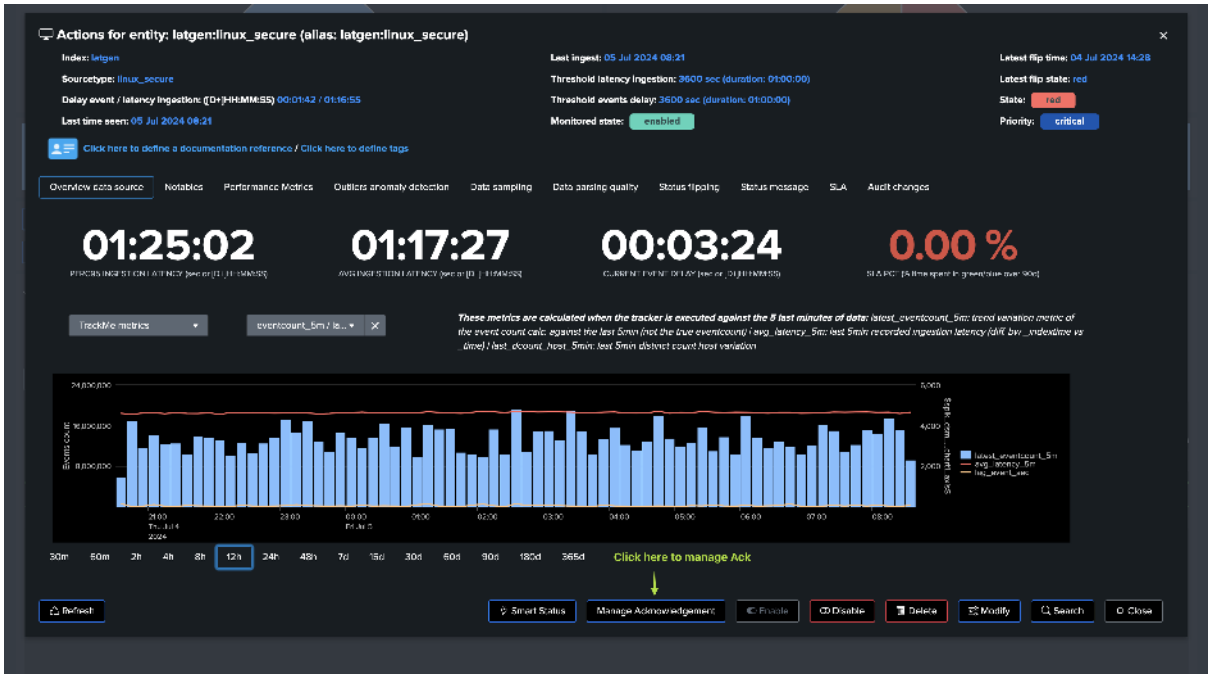
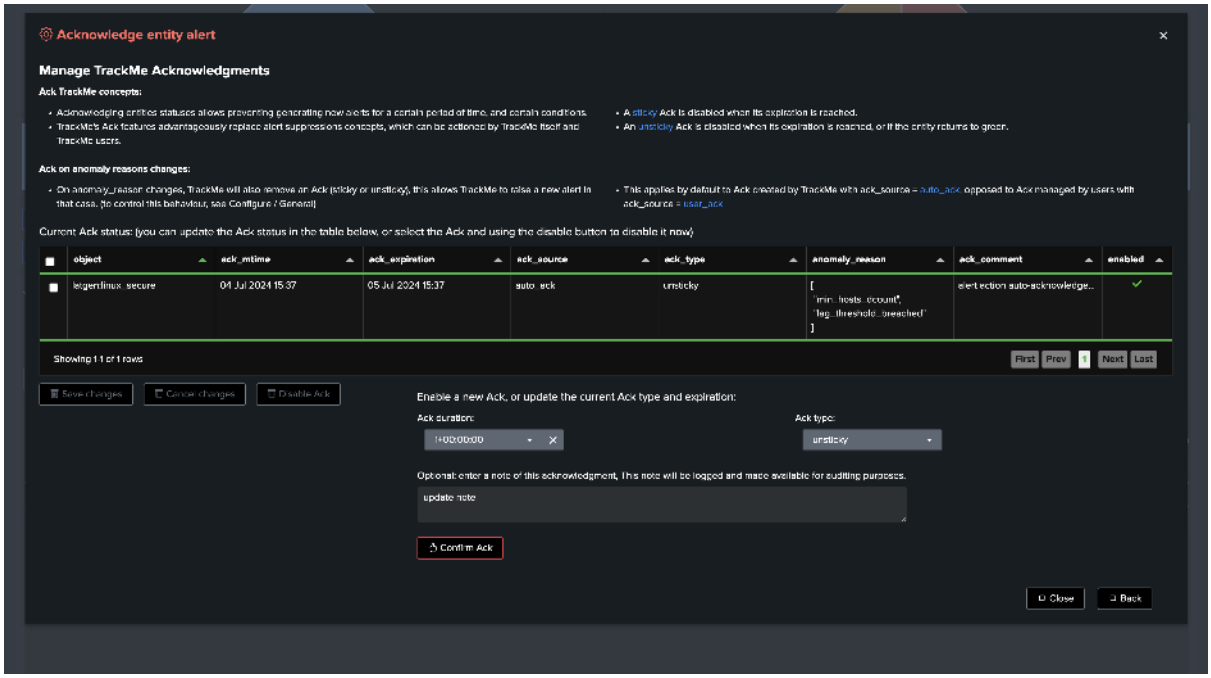
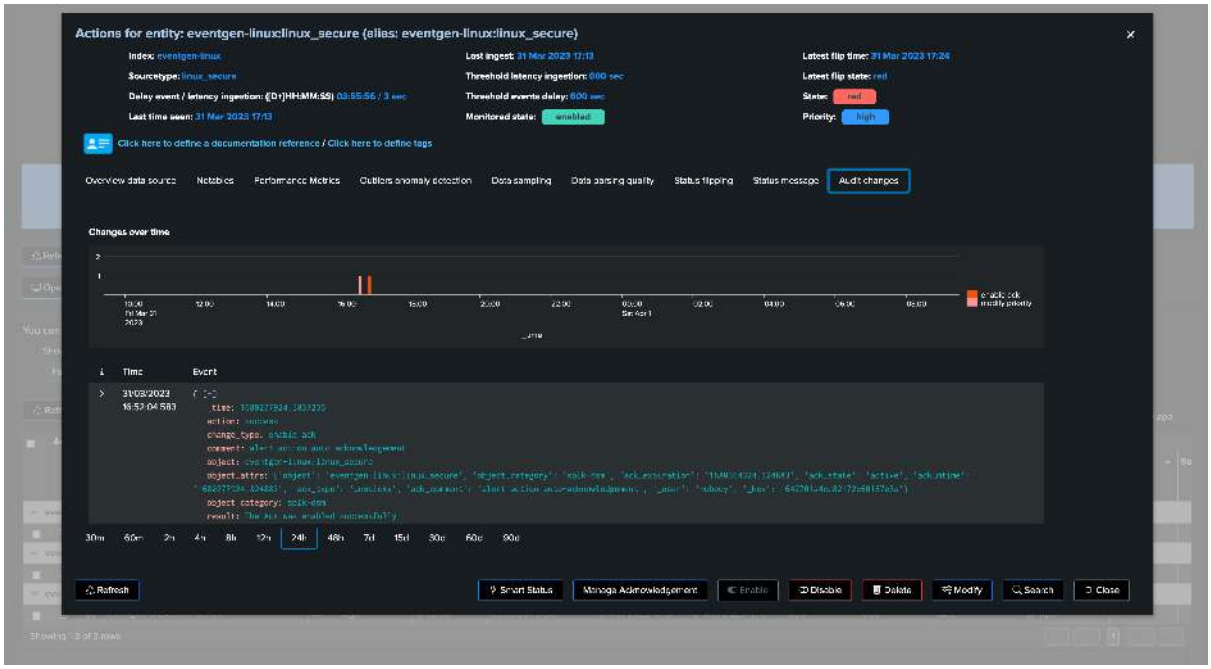
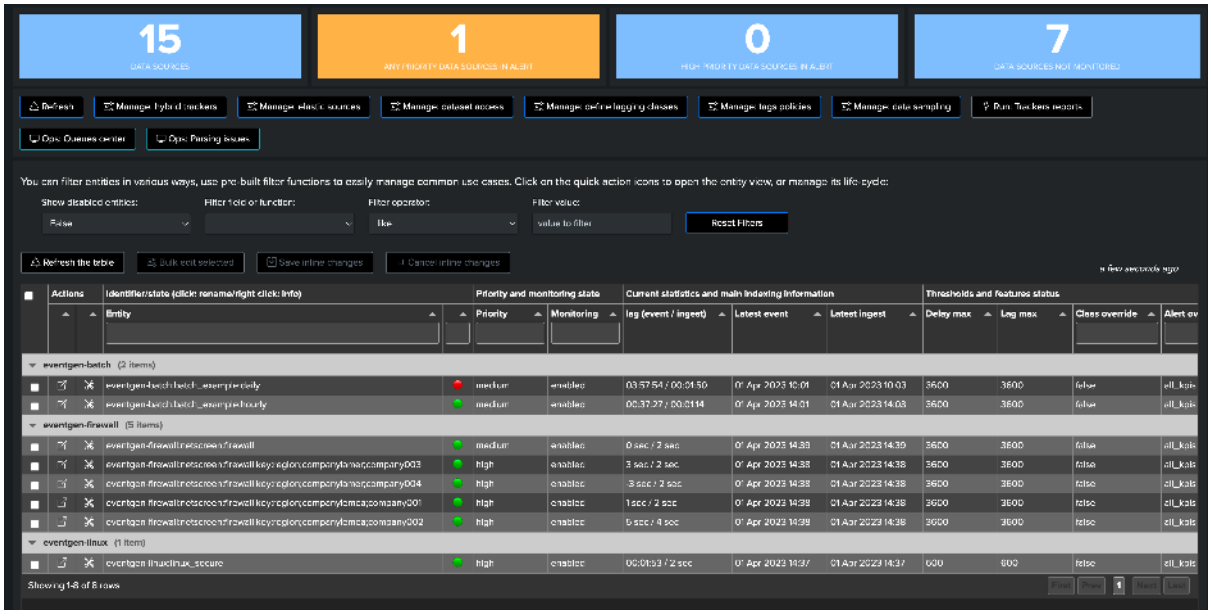
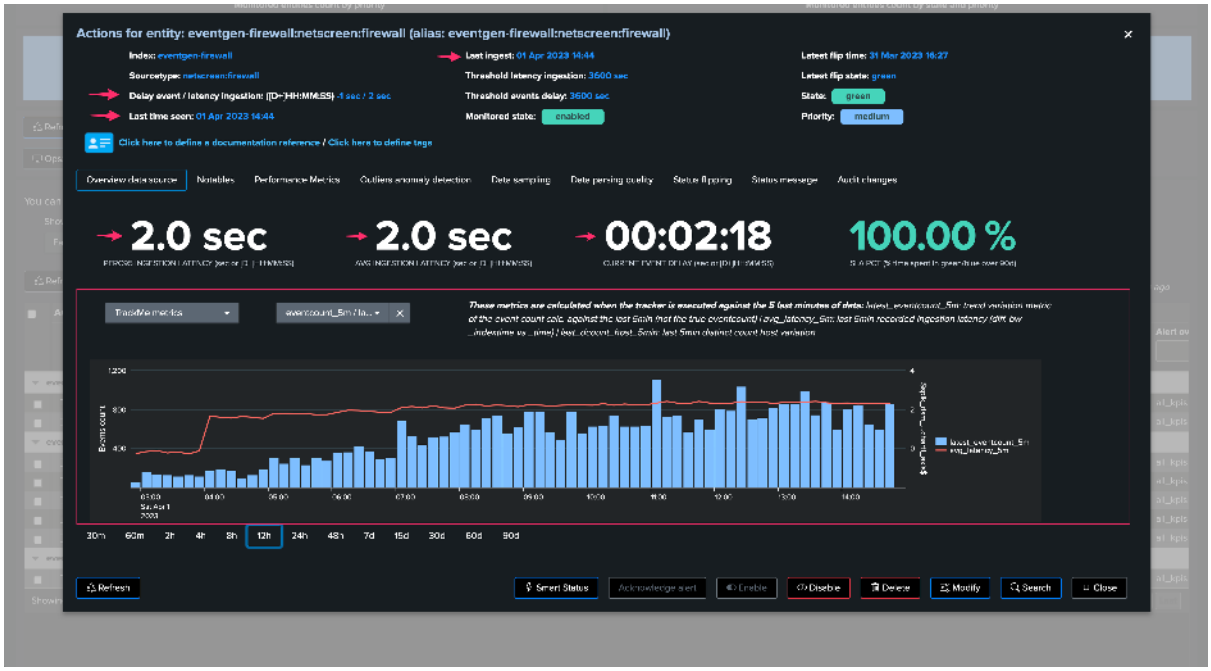
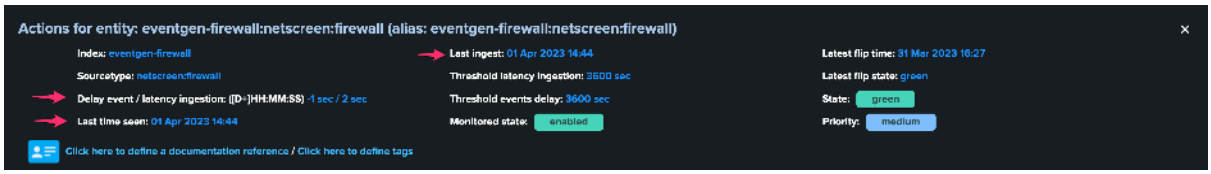
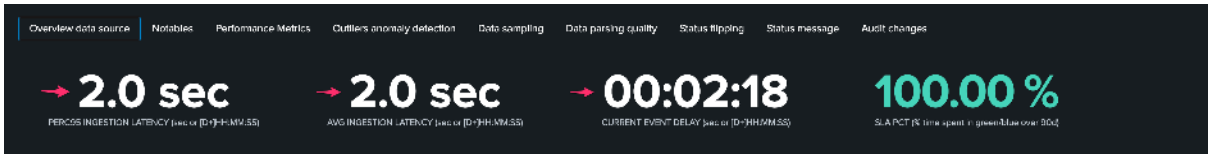
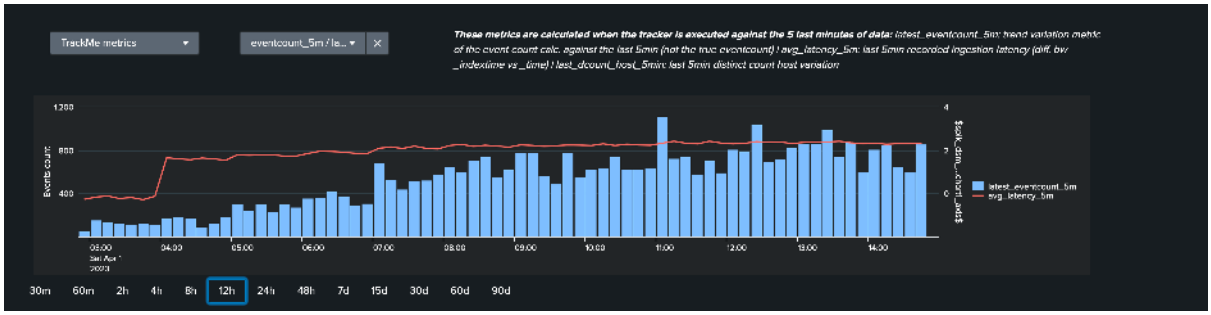
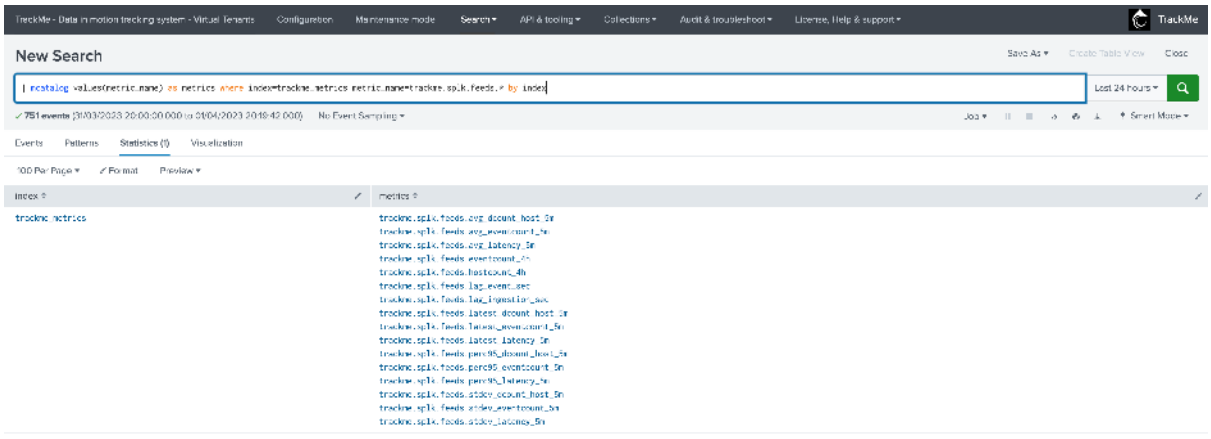
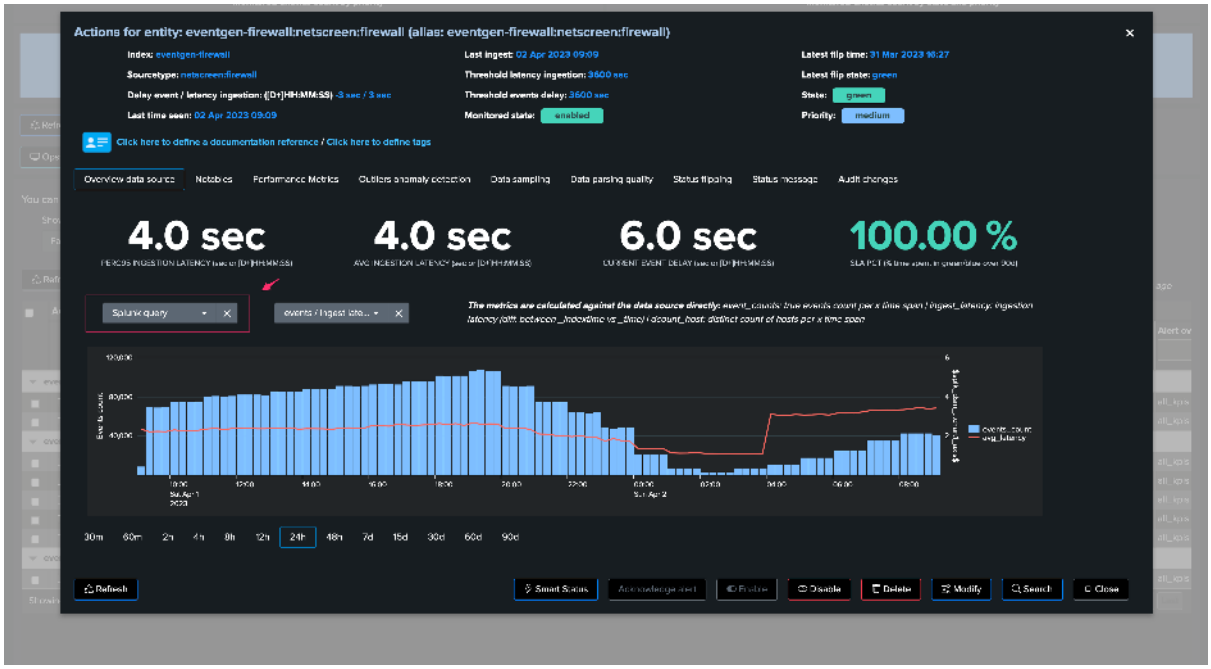
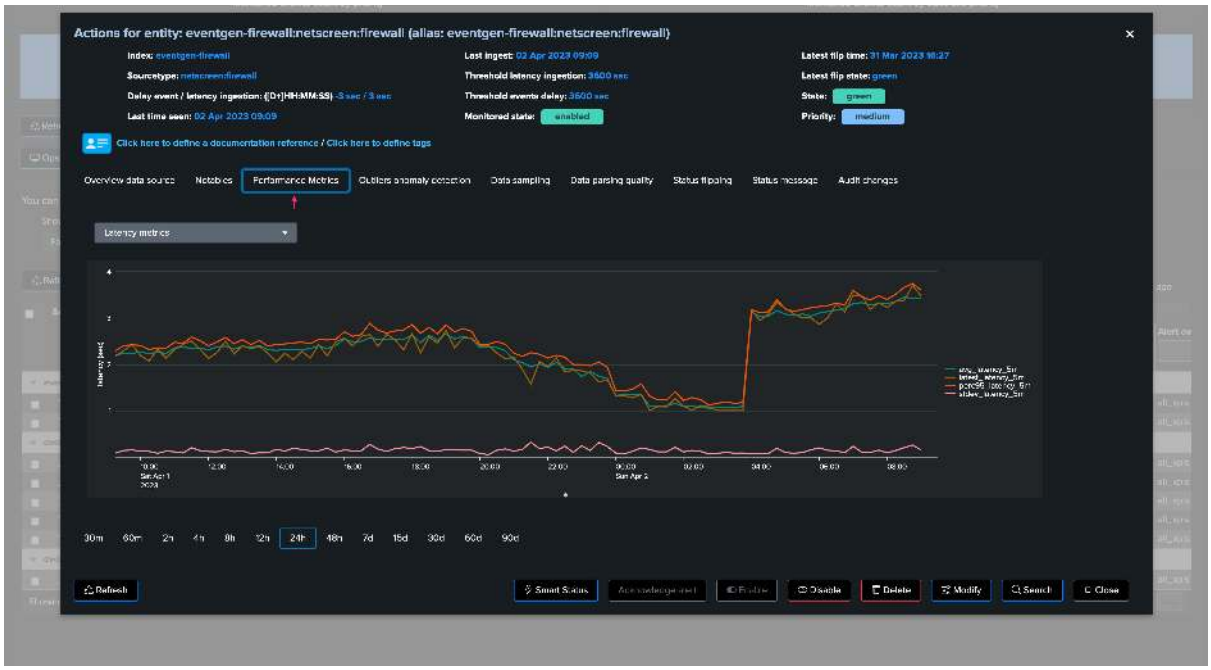
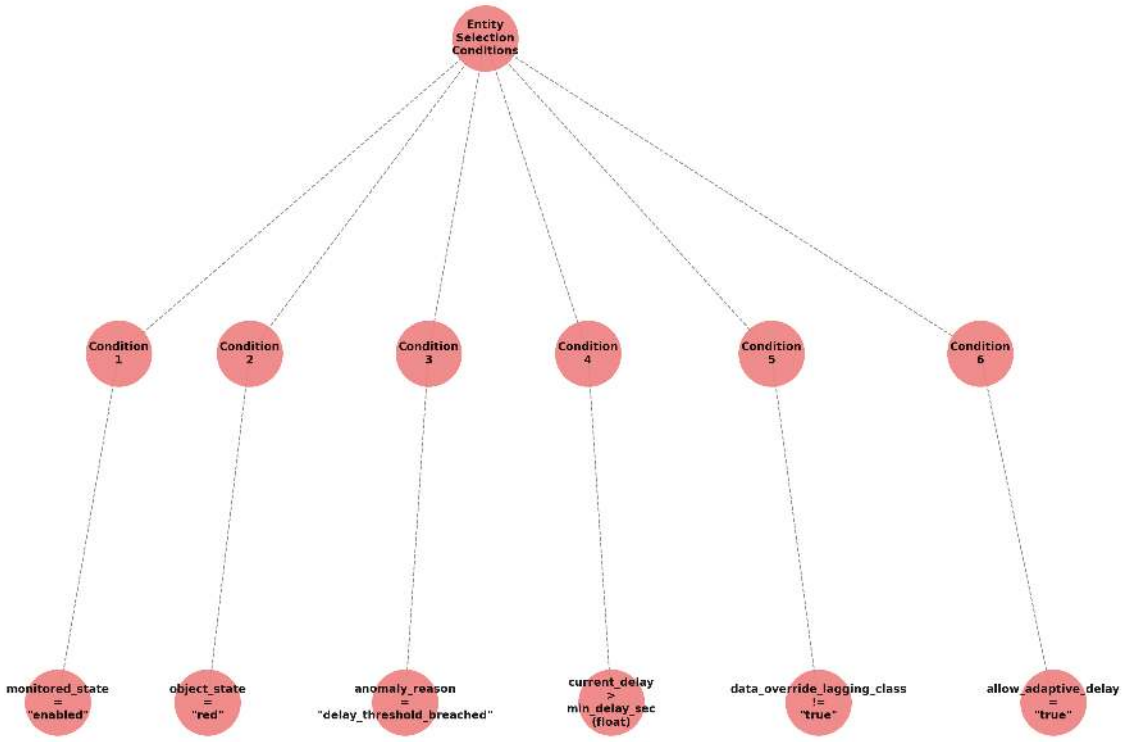
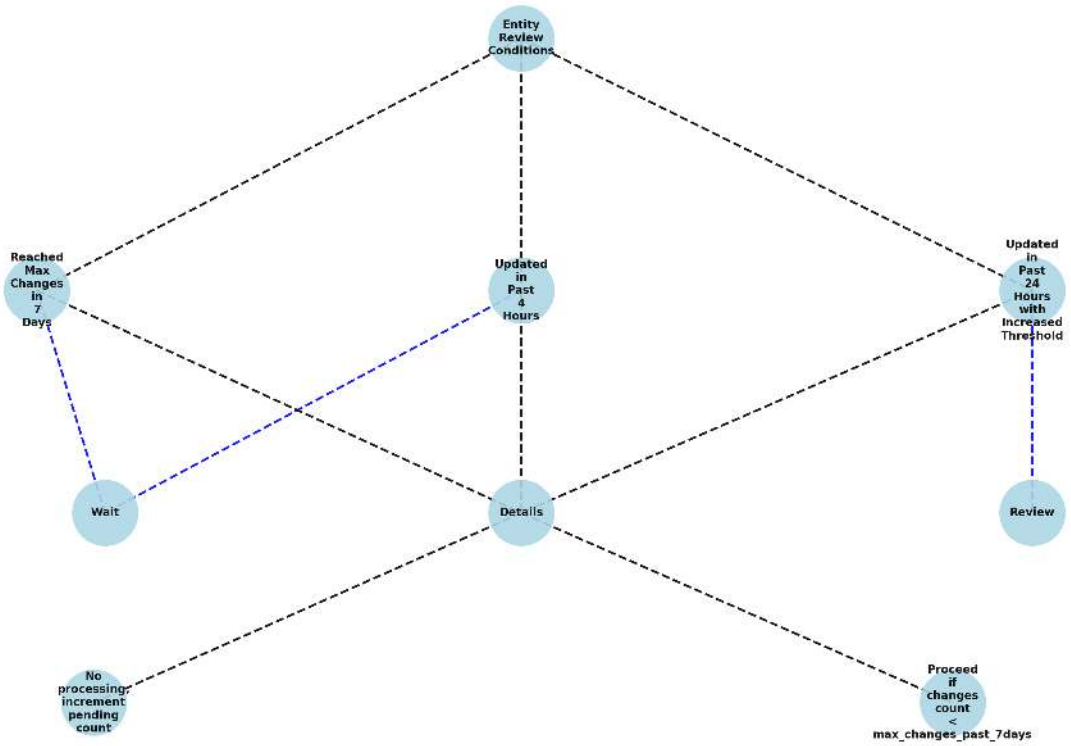
- splk-dsm: /services/trackme/v2/splk_dsm/write/ds_update_priority
- splk-dhm: /services/trackme/v2/splk_dhm/write/dh_update_priority
- splk-mhm: /services/trackme/v2/splk_mhm/write/mh_update_priority
- splk-wlk: /services/trackme/v2/splk_wlk/write/wlk_update_priority
- splk-flx: /services/trackme/v2/splk_flx/write/flx_update_priority
- splk-cim: /services/trackme/v2/splk_cim/write/cim_update_priority

8.12 Using SLA alerting to build a 2-tier monitoring system

About SLA alerting

- **SLA alerting** concepts were introduced in TrackMe 2.0.92.
- The concepts of SLA alerting can be leveraged to design a 2-tier monitoring system, where a first alert is emitted when TrackMe entities switch to a red state, and a second independent alert is emitted when the SLA is breached.
- This can be translated as "alert: this TrackMe entity is red!", then later on "alert: This entity has now spent too long in alert and is breaching its SLA!".
- Entities are associated with a given SLA class, each SLA class has a threshold in seconds and a numerical rank value.
- Available SLA classes and their configuration are defined at the level of the TrackMe system configuration. (A JSON object which defines the list of classes and their parameters)
- You can then define the SLA class per entity (which if defined will take precedence over policy-based classes), or via SLA policies, which are regular expressions orchestrated by TrackMe and handle the SLA class of each matching entities.
- Finally, you can create an SLA alert, which leverages the SLA information to emit an alert when the SLA is breached.

8.12.1 SLA Classes and Thresholds

SLA classes are defined at the level of the TrackMe system configuration, SLA classes define a threshold in seconds and a numerical rank value:
```



The default SLA classes are:

```
{
    "gold": {
        "sla_threshold": 14400,
        "rank": 3
    },
    "silver": {
        "sla_threshold": 86400,
        "rank": 2
    },
    "platinum": {
        "sla_threshold": 172800,
        "rank": 1
    }
}
```

Notes:

- You can add / remove / change classes as needed.
- You can define the default class to be used when TrackMe entities are discovered. (parameter: default_sla_class)
- The **Threshold** value is in seconds. For an SLA to be breached, a given TrackMe entity must be in alert for a continuous amount of time exceeding the threshold.
- The **rank** value is a numerical value, it is used to handle any conflict when applying SLA policies, the highest rank value will always win.

8.12.2 SLA Tab in TrackMe UI

The SLA feature is first translated into a tab called SLA which describes the current SLA status for the selected TrackMe entity:

The entity is in green state, therefore the SLA cannot be breached:



The entity is red state, however it has not yet breached the SLA threshold by spending a continuous amount of time in alert higher than the threshold:



The entity is red state, and it has breached the SLA threshold by spending a continuous amount of time in alert higher than the threshold:



8.12.3 Defining the SLA class per entity

You can define the SLA class manually on per entity basis: (via the UI or via the associated endpoint)

Hint

If the SLA has been defined manually, it will take precedence against policies based classes. (see next section)





8.12.4 Defining SLA policies to assign SLA classes automatically

You can define policies, which are regular expressions applied and orchestrated by TrackMe automatically:



For instance, say we want to match entities containing "cribl":





Let's execute the SLA tracker now:



From this stage, all matching entities will get the highest ranked policy and its associated threshold:



8.12.5 SLA Alerts

From TrackMe alert tabs, you can now create an SLA alert:

Hint

The TrackMe component must be defined and match your target (there would be one alert per component in the tenant)



Notes:

- This alert is independent from the TrackMe main alerts and notable alert
- If it triggers, it means that the SLA has been breached for one or more entities
- If the SLA is breached, the concept is to say that basically we had a first alert, the issue is not fixed after an acceptable amount of time so we generate a second alert once the SLA threshold has been breached. (2 tiers alerting system)

Example:



CHAPTER

NINE

USER GUIDE:

9.1 Entities Priority

9.1.1 Introduction to TrackMe Entities Priority

In TrackMe, all components and entities have a concept of "priority". The priority can be:

- low
- medium
- high
- critical



When TrackMe discovers and creates an entity, a default priority value is assigned to the entity.

By default, entity priorities are assigned as "medium", which can be changed by TrackMe administrators in the Virtual Tenant preferences.

9.1.2 Why Use Priority

Priority is a simple concept that allows you to categorize entities depending on their importance in your context.

This serves different purposes:

- This facilitates understanding if there are important entities affected by issues currently
- The TrackMe main user interface provides different features to highlight when high priority entities are affected
- You can filter on high priority entities, or use custom filters and drilldown actions to access entities with a certain priority value
- This allows qualifying entities over time to improve the coverage of your environment
- As well, you can create different alerts filtering on certain priorities, to handle different actions depending on the priority

- For example, you may want to generate an incident in your ITSM tool for high priority entities, while other priority types lead to email alerts

9.1.3 Updating the priority

You can update the priority for one or more entities, as needed. The priority is a KVstore field which is persistent.

To update an entity priority, you can:

- Open the entity main screen, click on the "Modify" button and assign the priority accordingly
- You can update priorities in bulk edit mode, select one or more entities, click on the bulk action button and define the priority as needed.
- The priority can as well be updated using the associated REST API endpoint for the component

Updating the priority of a single entity

Open the entity Modification screen (click on the entity icon then Modification, or the configure icon right to the open icon), and set the priority as needed:





Updating the priority in bulk

Select one of more entities to be updated, and click on the bulk edit button:





Updating the priority with the REST API

Open the REST API reference to find the endpoint for that component, for instance with splk-dsm:

```
| trackme url="/services/trackme/v2/splk_dsm/write/ds_update_priority" mode="post"␣
↪body="{'tenant_id': 'mytenant', 'priority': 'high', 'object_list': 'eventgen-
↪firewall:netscreen:firewall|key:region;company|amer;company004,eventgen-
↪firewall:netscreen:firewall|key:region;company|amer;company003'}"
```



9.1.4 Priority change audit

All changes of the priority through TrackMe are audited.

When a user updates the priority, an audit event associating the Virtual Tenant, the component and the entity is created.

You can review audit changes for a given entity in the "Audit" tab of the entity main screen.



9.1.5 Recommendations for Using TrackMe Entities Priority

When managing high-scale environments, one of the challenges is often dealing with the large number of entities that can be discovered.

To effectively handle this situation, consider the following recommendations:

- Start progressively: Limit the scope of the Hybrid Trackers responsible for entity discovery as much as possible. This approach can help you maintain control over the number of entities being discovered and monitored.
- Leverage priority: Use TrackMe's entities priority feature to distinguish valuable entities from those that are less important. For instance, tagging entities that you have reviewed and qualified as "high" or "critical" can help you focus on the most critical components in your environment.
- Adopt a tiered alerting approach: By associating high priority entities with TrackMe alerts, you ensure that only qualified entities trigger alerts. This method helps to:
  - Improve your monitoring posture by focusing on the most important entities
  - Reduce false positive alerts
  - Minimize alert fatigue by reducing the number of alerts generated from lower priority entities

By following these recommendations, you can better manage the entities in your high-scale environment and ensure that your monitoring efforts are focused on the most critical components.

9.2 Entity Monitoring State

9.2.1 Introduction to the monitored state

In TrackMe, all entities have a field defining the "monitoring status". This field is a persistent field stored in the KVstore named "monitored_state" which expects two possible values:

- enabled: the entity is actively monitored
- disabled: the entity remains but is not considered actively nor leading to alerts



9.2.2 Purposes of Monitoring Status

The Monitoring Status field serves the following purposes:

- The TrackMe main user interface filters by default on entities that are actively monitored
- Entities can be disabled and hidden from the main UI without being deleted
- TrackMe alerts only consider entities which are actively monitored

9.2.3 Use Cases for Monitoring Status

There are different conditions where this feature can be useful:

- The Hybrid Tracker data scope means some entities that are out of interest are discovered, and these entities should not be considered anymore
- A given entity representing a data provider has issues that cannot be addressed at the moment, and you want to fully exclude these entities without permanently deleting them

9.2.4 TrackMe User Interface

The main user interface filters out by default on monitored entities. However, currently disabled entities are accounted for in a single informative view, and you can also ask TrackMe to reveal entities that are currently disabled.

Drilldown: click on the single view, this automatically filters out on disabled entities:





You can ask TrackMe to include disabled entities in the Tabulator:



You can use the the filter function to only show disabled entities in the Tabulator:



9.2.5 Updating the Monitored State

You can enable or disable the monitoring state on a per-entity basis, via bulk editing and via the REST API endpoints.

Enabling / Disabling an entity

Open the entity main screen and click on Enable / Disable:



Enabling / Disabling in bulk

Select one or more entities, click on the bulk action button:





Enabling / Disabling in REST

Open the REST API user interface to find the endpoint associated with the component, you can enable / disable multiple entities at once:

```
| trackme url="/services/trackme/v2/splk_dsm/write/ds_monitoring" mode="post" body="{
↪'tenant_id': 'feeds-tracking', 'action': 'disable', 'object_list': 'eventgen-
↪firewall:netscreen:firewall|key:region;company|emea;unknown,eventgen-
↪firewall:netscreen:firewall|key:region;company|unknown;company001'}""
```



9.3 Status Message

9.3.1 Introduction to the Status Message

The TrackMe status message is a human-readable status that describes the reasons why TrackMe defines the entity status. It is the fastest and easiest way for an analyst to understand the reasons behind the status of a TrackMe entity. TrackMe entities can have the following statuses:

- Green: The entity is considered to be in a healthy state.
- Red: The entity is considered to be in a non-healthy state.
- Orange: This is an intermediate status, indicating a condition such as the detection of data in the future.
- Blue: The entity is a member of a logical group that fulfills the monitoring conditions but is in a non-healthy state.

The status is influenced by different factors, such as Key Performance Indicators and additional features such as the Machine Learning Outliers detection.

9.3.2 Factors Influencing the Status

These conditions differ depending on the type of TrackMe component.

For instance, with the splk-dsm TrackMe component (which stands for Data Source Monitoring), the essential factors can be:

- If the entity is suffering from latency at the ingestion (the maximal acceptable threshold for the latency for this entity was breached)
- If the entity is suffering from delay (the data flow is interrupted according to the entity threshold)
- Quality anomalies were found by the Data Sampling TrackMe feature (events format recognition)
- Outliers were detected by the Machine Learning engine

And more.

The “status_message” field is a readable translation of the machine status for that entity, adding context to easily and quickly understand the reasons behind the status.

9.3.3 Availability of the Status Message

The status message is made available in different parts of TrackMe, such as:

- A graphical object in the user interface for the analyst to review
- As part of a notable event created by a TrackMe alert
- As part of the alert results themselves
- Stored in the main KVstore of the TrackMe Virtual Tenant and component, in the status_message field

9.3.4 Reviewing the Status Message

The status message of an entity is easily accessible from the tab called "Status message" in the entity main screen. The message content is associated with the color code of the entity state, such as red, green, etc.

9.3.5 Examples of Status Messages

Here are some examples of status messages:

- Green:

```
Good: entity status is green, latest data available is 31 Mar 2023 16:58 (479 seconds␣
↪from now), and monitoring conditions are met.
```



Red:

```
Alert: entity status is red, monitoring conditions are not met due to lagging or␣
↪interruption in the data flow, latest data available is 31 Mar 2023 16:56 (631␣
↪seconds from now) and ingestion latency is approximately 4 seconds, max lag␣
↪configured is 600 seconds.
```

The TrackMe status message is an essential feature for quickly understanding the reasons behind an entity's status, making it easier to determine what needs to be addressed to monitor and track data availability and quality effectively.

9.4 Status Flipping Feature

9.4.1 Introduction to flipping events

The Status Flipping feature is a powerful and informative part of the TrackMe application for Splunk, designed to monitor data availability and quality for Splunk Enterprise and Splunk Cloud. This feature is available for all TrackMe components, including:

- splk-dsm (Data Source Monitoring)
- splk-dhm (Data Host Monitoring)
- splk-mhm (Metric Host Monitoring)
- splk-cim (Common Information Model compliance tracking)
- splk-flx (Flex Object tracking)



This guide will provide an overview of the Status Flipping feature and how it helps to detect and track changes in the state of entities.

9.4.2 Status Flipping: Understanding Entity States

In TrackMe, entities have a state that represents their current health status. These states include:

- Green: The entity is considered to be in a healthy state.
- Red: The entity is considered to be in a non-healthy state.
- Orange: This is an intermediate status, indicating a condition such as the detection of data in the future.
- Blue: The entity is a member of a logical group that fulfills the monitoring conditions but is in a non-healthy state.

9.4.3 How Status Flipping Works

The Status Flipping feature's purpose is to track changes in the state of entities. Here's how it works:

- Discovery: When an entity is discovered by TrackMe trackers, a flipping event is generated in the trackme_summary index. The entity switches from the "discovered" status to its state (green, red, or orange) based on the detection.
- Status Change: When an entity's status changes because TrackMe has detected an issue, a new event is created in the trackme_summary index. This event shows the previous status, the new status, the reason for the change (captured in a field called "anomaly_reason"), and time-related information.
- Entity View: TrackMe provides a tab in the entity view that displays the flipping status events over time in a chart, as well as a table with the flipping events ordered by the most recent events.

9.4.4 Flipping Event Examples

Here are some examples of what flipping events might look like:

Example: Entity being discovered:

```
31/03/2023 14:52:16, object=osnixsec:linux_secure has flipped from previous_
↪state=discovered to state=green with anomaly_reason=none
```

Example: Entity switching from green to red:

```
31/03/2023 15:02:16, object=eventgen-linux:linux_secure has flipped from previous_
↪state=green to state=red with anomaly_reason=delay_threshold_breached
```

Example: Entity switching from red to green:

```
31/03/2023 15:59:15, object=eventgen-linux:linux_secure has flipped from previous_
↪state=red to state=green with anomaly_reason=none
```

9.4.5 Reviewing TrackMe Flipping Events

You can access flipping events that fired for a given entity in the "Status Flipping" tab of the main entity screen. This view provides the following information:

- Overtime Chart: An overtime chart of the flipping events, which helps to visualize the frequency and distribution of these events over time.
- Table View: A table that displays the entity-related notable flipping events ordered by the latest flipping events.

This view provides quick access to the details of each event, making it easier to investigate and understand the frequency of the flipping events for this entity and the reasons behind these changes.



9.4.6 Searching Flipping Events in Splunk

You can search for flipping events in Splunk using the following search:

```
`trackme_idx(mytenant)` sourcetype=trackme:state tenant_id="mytenant" object="myobject
↪"
```

Replace "mytenant" with the name of your TrackMe Virtual Tenant and "myobject" with the name (identifier) of the TrackMe entity you want to search for.

9.4.7 Using Flipping Status Events as Key Performance Indicators

TrackMe leverages the flipping status events as a form of Key Performance Indicator (KPI) to measure the entity's behavior concerning how often it changes from one state to another. Frequent status changes can indicate potential issues, such as bad behavior or the need for fine-tuning the entity's configuration.

By keeping an eye on the flipping status events, users can gain valuable insights into the performance and stability of their entities, allowing for proactive measures to ensure optimal performance and data quality.

The SmartStatus feature for instance performs a correlation regarding flipping events:



9.5 Notable Events

9.5.1 Introduction to Notable Events

TrackMe provides a feature called "TrackMe Notable Events". When a TrackMe alert is created, a TrackMe alert action is automatically enabled for that alert (called trackme_notable). When the alert is executed and fires alerts for given entities, TrackMe generates a notable event which is stored in the Splunk notable index that is associated with the TrackMe Virtual Tenant.

The notable events contain summary information regarding the entity that was concerned by the alert, as well as all properties of that entity. This acts as an instant snapshot of the entity in the state it was in when the alert fired. Therefore, using notable events facilitates the understanding and investigation of why TrackMe alerted about a given entity and what were the reasons for the alert to trigger.



9.5.2 How TrackMe Notable Events Work

TrackMe notable events are designed to provide valuable information when a TrackMe alert fires for a specific entity. Here's how the process works:

- Alert Creation: When a TrackMe alert is created, a corresponding TrackMe alert action (trackme_notable) is automatically enabled.
- Alert Execution: When the TrackMe alert is executed, it evaluates the monitored entities based on the alert's conditions.
- Alert Firing: If the alert's conditions are met for a given entity, the alert fires for that specific entity.
- Notable Event Generation: Once the alert fires, TrackMe generates a notable event for the concerned entity, which is then stored in the associated Splunk notable index. Notable events are JSON-formatted events generated by the built-in TrackMe alert action trackme_alert.

Please note that notable events are only created when a TrackMe alert fires for that entity.

9.5.3 Searching Notable Events in Splunk

Notable events can be searched with the following Splunk search:

```
trackme_notable_idx(mytenant) tenant_id="mytenant" object="myobject"
```

Replace "mytenant" with the TrackMe Virtual Tenant name, and "myobject" with the entity name.

9.5.4 Reviewing TrackMe Notable Events

You can access Notable events that fired for a given entity in the "Notables" tab of the main entity screen. This view provides the following information:

- Overtime Chart: An overtime chart of the notable events, which helps to visualize the frequency and distribution of these events over time.
- Table View: A table that displays the entity-related notable events ordered by the latest notable events.

This view provides quick access to the details of each event, making it easier to investigate and understand the reasons behind the alert.


Actions for entity: eventgen-linux:linux_secure (alias: eventgen-linux:linux_secure)
Notables
Performance Metrics
Outliers anomaly detection
Data sampling
Data parsing quality
Status flipping
Status message
Audit changes
Notables events
Refresh
Smart Status
Manage Acknowledgement
Disable
Delete
Modify
Search
Close


Actions for entity: eventgen-linux:linux_secure (alias: eventgen-linux:linux_secure)
Notables
Performance Metrics
Outliers anomaly detection
Data sampling
Data parsing quality
Status flipping
Status message
Audit changes
Refresh
Smart Status
Manage Acknowledgement
Disable
Delete
Modify
Search
Close



9.5.5 Notable Events: Essential and Valuable Feature

The TrackMe notable event is an essential and valuable feature for understanding and investigating the reasons behind an alert.

It provides a snapshot of the entity's state when the alert was triggered, making it easier to determine what caused the issue and how to address it.

By leveraging notable events, you can enhance your monitoring and incident response capabilities, ultimately leading to a more efficient and reliable system.

9.6 Acknowledgments

9.6.1 Introduction to Acknowledgments

Hint

Acknowledgments in TrackMe

- Acknowledgments are a powerful feature in TrackMe that allows control of alert suppression for TrackMe entities, with advanced and sophisticated capabilities.
- Acknowledgments can be emitted automatically by TrackMe alerts using built-in TrackMe Auto-Ack alert actions, and/or manually by TrackMe users.
- A **sticky** Ack is disabled when its expiration is reached. An **unsticky** Ack is disabled when its expiration is reached, or if the entity returns to green.
- On **anomaly_reason changes**, TrackMe will also remove an Ack (sticky or unsticky); this allows TrackMe to raise a new alert in that case. (To control this behavior, see Configure / General)

Update to Ack in TrackMe 2.0.97 with anomaly reason change detection

- In TrackMe 2.0.97, we have made improvements to the TrackMe Ack concepts to allow for more flexibility and capabilities in managing acknowledgments.

- This new feature allows TrackMe Ack to be influenced by changes in anomalies affecting entities.
- Conditioned by system-level configurable options (See Configure / General / Expire Ack on anomaly reason change behavior, Expire Ack on anomaly reason change min time since, Expire Ack on anomaly reason only for auto ack), this new feature completes and enhances the Ack capabilities in TrackMe.
- If an entity that turned red due to an Outliers detection, for instance, and later on is also affected by an additional condition such as a lag breach, the Ack will be automatically expired so that a new alert can be raised transparently by TrackMe.

9.6.2 How TrackMe Acknowledgment Works

TrackMe Acknowledgment can be enabled either automatically or manually for entities:

Automatic Acknowledgments

Acknowledgments can be enabled automatically for entities as part of TrackMe alerts. When an alert is configured and fires for a given entity, the TrackMe alert action called "TrackMe auto acknowledge" enables the Acknowledgment for a certain period of time (configurable per alert). The TrackMe main user interface reflects the fact that the entity was acknowledged with a specific "eyes" icon on the left of the status icon. Automatic Acknowledgments can still be managed manually by an analyst within the user interface. They can be disabled, and their expiration period can be extended.

Manual Acknowledgments

Acknowledgments can be enabled manually via the user interface.

- The expiration period can be defined (e.g., 7 days) and can be extended later on.
- At any time, an acknowledgment can also be removed, allowing the entity to be fired again by an alert.

Sticky Acknowledgments versus Unsticky Acknowledgment

- You can define the type of Acknowledgment. There are two types of Acknowledgment: sticky and unsticky.
- A sticky Acknowledgment means that even if the entity moves from a non-healthy state to a healthy state (red to green), the Acknowledgment remains active for the period of time it was created. In that case, if the entity goes again to a non-healthy status, it will not generate a new alert.
- An unsticky Acknowledgment, on the other hand, means that if the entity goes back to green, the Acknowledgment is disabled automatically by the system-wide Acknowledgment tracker. The entity can therefore lead to a new alert.

Additional Information on Acknowledgment Management

When an active alert is acknowledged, it remains visible in the user interface and monitored. However, no more alerts will be generated based on the built-in alerting for that source:

- An acknowledgment is valid for a period of time and will be disabled automatically once its expiration time has been reached.
- If an active acknowledgment is acknowledged again, its expiration is extended for a new cycle.
- An unsticky acknowledgment is automatically disabled when the source returns to a green state.
- A sticky acknowledgment remains active even if the entity goes back to green and will be purged only when it expires.
- An unsticky acknowledgment remains active as long as the entity does not return to green; if it does, the acknowledgment will be purged.

When an entity was acknowledged, an "eyes" icon is added to the left of the status icon:



You can right click on the entity name to access the context menu:



When an entity is not a healthy status, the button Manage acknowledgments is enabled automatically:



You can use this screen to manage acknowledgments:



Acknowledgments are audited, you can review their activities in the audit tab of the entity:



9.6.3 Conclusion

The TrackMe Acknowledgment feature provides a convenient way for analysts to manage and control the visibility of non-healthy entities in the user interface while preventing new alerts from firing for the same entity.

This feature enables efficient handling of issues related to data providers, ensuring a smooth monitoring experience for TrackMe users.

9.7 Splunk Feeds KPIs (splk-feeds)

9.7.1 Introduction to latency and delay

This documentation focuses on the Key Performance Indicators (KPIs) available with the TrackMe component called “splk-feeds”, which stands for Splunk Feeds Tracking.

The category of components includes:

- splk-dsm (Data Source Monitoring)
- splk-dhm (Data Host Monitoring)
- splk-mhm (Metric Host Monitoring)

In the context of these components, we focus on the metrics that can be categorized as part of “lagging metrics”.

In TrackMe, this essentially means:

- latency: how fast events are received and indexed, in regards to when these events were produced by the data provider
- delay: how late in the past is this data provider, in other terms, when did we receive the latest event for this provider

9.7.2 Latency in the Splunk Context

In the Splunk context, latency is the time taken between when the event was produced and when the event was received and written.

To calculate the latency, we operate a simple formula:

```
latency = (_indextime - _time)
```

9.7.3 Understanding Latency's Importance

Latency is an important concept because it can impact the reliability of Splunk use cases and even imply considerable consequences from an investment, business, and people's real-life perspective.

TrackMe was originally designed to detect and alert when Splunk is suffering from latency in large environments with a multitude of data sources and typologies of events.

9.7.4 Delay in the Splunk Context

Delay is another KPI related to the quality assessment activity.

It is the interruption of a data flow which can happen for various reasons, such as process issues, configuration issues, or network issues.

TrackMe calculates delay using the formula:

```
delay = now() - last_time
```

9.7.5 Difference between Latency and Delay

Latency and delay are two distinct concepts in the context of Splunk and TrackMe.

Here's a comparison between the two:

Latency

- Refers to the time difference between when an event was produced by the data provider and when it was received and indexed by Splunk
- Calculated as: latency = (_indextime - _time)
- Often caused by network issues, misconfigured timezones, or incorrect timestamp parsing
- Affects the reliability of search results, scheduled reports, and alerts due to late indexing of events
- Can lead to missed events or incoherent results, especially in time-constrained searches

Delay

- Refers to the time difference between the current time (when the Hybrid Tracker runs) and the latest event received for a data provider
- Calculated as: delay = now() - last_time
- Often caused by data flow interruptions, such as process failures, configuration issues, or network problems
- Indicates potential issues with data flow and can impact the quality of service
- TrackMe allows setting different thresholds for delay depending on the data source's expected behavior

While latency and delay are related, they are not the same concept.

TrackMe manages both KPIs individually, allowing users to set different thresholds for latency and delay to adapt to various use cases and data sources.

9.7.6 Significance of Delay

Delay is a major KPI regarding the quality of service.

Detecting interruptions in the data flow is a key to quality.

TrackMe allows managing both thresholds efficiently in various use cases, regardless of whether data sources continuously generate events or generate events sporadically or in a batch fashion.

9.7.7 How TrackMe Handles Latency and Delay

TrackMe is designed to manage latency and delay efficiently, providing a consistent and powerful workflow and framework for monitoring data sources in large environments.

Here's how TrackMe handles latency and delay:

1. TrackMe runs scheduled Hybrid Trackers to calculate latency and delay values for each entity.
2. The Hybrid Trackers are executed based on a cron schedule, which means the accuracy of delay values will vary accordingly. For example, if the tracker is executed every 5 minutes, there can be up to a 5-minute delay before new events are considered.
3. TrackMe stores the latest latency and delay values for each entity and provides a user interface to view and manage these values.
4. TrackMe allows users to set different threshold values for latency and delay. When these thresholds are exceeded, alerts can be generated to notify users of potential issues.
5. TrackMe supports a variety of use cases, including continuous event generation and sporadic or batch event generation. This flexibility allows TrackMe to adapt to the specific needs of different data sources.

9.7.8 Reviewing latency and delay in TrackMe

TrackMe Tabulator

When accessing TrackMe main user interface, several information related to the latency and delay are displayed in the Tabulator:



For each entity:

- The column "**lag (event / ingest)**" provides a summary of delay and latency (in this order)
- The column "**Latest event**" informs about the latest event that was detected by TrackMe (therefore, the latest event from the _time perspective)

- The column "**Latest ingest**" informs about the latest event that was indexed (therefore, the max value of _indextime)
- The column "**Delay max**" is the current threshold value for the delay
- The column "**Lag max**" is the current threshold value for the latency

TrackMe entity view

When opening the main screen of an entity, more information is provided for that entity especially:



Let's describe the top information first:



- **Delay event / latency ingestion: ([D+]HH:MM:SS):** shows the current and latency, these are shown as duration (the Splunk function tostring duration) if the the value exceeds 60 seconds
- **Last time seen:** shows the last event seen, from the _time perspective
- **Last ingest:** shows the last indexed event, from the _indextime perspective

Then we have some high level single view statistics:



These statistics are calculated against either the TrackMe metrics, or direct Splunk queries against the data sources (see the next section):

- **PERC95 INGESTION LATENCY (sec or [D+]HH:MM:SS)** Shows the percentile 95 calculation against the latency
- **AVG INGESTION LATENCY (sec or [D+]HH:MM:SS)** Shows the average calculation against the latency

- **CURRENT EVENT DELAY (sec or [D+]HH:MM:SS)** Shows the current delay for the entity

Finally, we have the over time charts:



Source data for the calculation in TrackMe overview entity screen:

The left dropdown selector allows to operate between two modes:

- TrackMe metrics
- Splunk queries

TrackMe metrics

When using **TrackMe metrics**, the overview relies on the metrics that Hybrid Trackers generate and index in the TrackMe metric index defined for this tenant.

You can access the metric index from the Virtual Tenant user interface (indexes shortcuts), otherwise use the following search:

```
| mpreview `trackme_metrics_idx(mytenant)` filter="tenant_id="mytenant""
```

where "mytenant" is to be replaced with the name of the Virtual Tenant.

Metrics are prefixed with "trackme_trackme.splk.feeds.", you can as well use the mcatalog command:

```
| mcatalog values(metric_name) as metrics where index=trackme_metrics metric_
↪name=trackme.splk.feeds.*
```



Splunk Queries

When using **Splunk Queries**, the overview performs direct queries against Splunk data, consider that:

- These queries process Splunk data directly (most of the time these will be tstats based queries)
- Processing require more time than using TrackMe metrics (which are very performing mstats based searches), the user experience is much slower

- Long time range searches on very large environment will likely not be doable in reasonable amount of time, unlike TrackMe metrics
- This requires access to the indexes, unlike TrackMe Metrics



For the most optimized experience and fastest results, use TrackMe metrics.

Performance metrics tab

The "Performance metrics" tab provides a deeper access to all TrackMe metrics:



9.7.9 Conclusion on latency and delay

TrackMe stands out for its consistent and scalable approach to addressing a critical challenge faced by Splunk customers.

Monitoring latency and data flow interruptions is crucial for maintaining the quality and reliability of the insights gained from Splunk.

Latency and data flow interruptions can have severe consequences, leading to inconsistent results and jeopardizing the business and security detections that rely on the Splunk platform. Being proactive in detecting and addressing latency and delay issues is essential for successfully managing and operating large-scale Splunk environments.

By utilizing TrackMe, organizations can effectively monitor and manage latency and delay, ensuring that the data driving their decisions and actions remains accurate and up-to-date, ultimately improving their overall Splunk experience.

9.8 Splunk Feeds Thresholds (Delay and Latency, Machine Learning adaptive thresholding)

9.8.1 1. Introduction to adaptive thresholding

In TrackMe, Splunk feeds latency and delay Key Performance Indicators (KPIs) are continuously monitored.

When a given entity breaches the predefined threshold values, this impacts the state of the entity, and depending on the configuration, the entity status will turn red, potentially leading to an alert being emitted by TrackMe.

By default and when discovering entities, TrackMe applies a 1-hour maximal threshold (3600 seconds) for both delay and latency.

The purpose of this documentation is to describe the options that TrackMe provides to configure the threshold values accordingly.

9.8.2 2. Adaptive delay thresholds with Machine Learning (since TrackMe v2.0.72)

Hint

Since TrackMe v2.0.72, we implement a **Machine Learning** driven approach to automatically adapt the delay threshold values based on the historical knowledge TrackMe has accumulated.

- This feature is handled by a dedicated tracker called **trackmesplkadaptivedelay** inspecting entities reporting delay threshold breaches
- It identifies the number of days of accumulated metrics for this entity (to define the confidence level, provided as an argument to the tracker, by default 7 days minimum required)
- If conditions allow it, TrackMe updates automatically the delay threshold value
- The more knowledge TrackMe accumulates over time, the more accurate the threshold values will be
- Various important enhancements were made in further TrackMe releases to improve the accuracy and behavior of the adaptive threshold features in TrackMe

Hint

Since TrackMe 2.0.96, control the review period using the argument review_period_no_days

- This argument was introduced in this release, it allows you to control the period of time for the review of TrackMe entities that were updated.
- Valid options are: 7, 15 or 30 days, the default value is 30 days.

- Update the argument in each tracker configuration if you wish to change the default value.
- argument: `review_period_no_days=30`

Hint

Since TrackMe 2.1.10, control the max SLA percentage using the argument max_sla_percentage

- This argument was introduced in this release, it allows you to control the max SLA percentage for the adaptive thresholding.
- This defines the threshold for the SLA percentage under which the adaptive thresholding will **NOT** attempt to update the threshold value for a given entity.
- This feature **reduces the risk of updating a stable entity which is affected by a true positive anomaly**.
- If the currently observed SLA percentage is not lower than the max SLA percentage (defaults to 90%), the entity will not be managed.
- Update the argument in each tracker configuration if you wish to change the default value.
- argument: `max_sla_percentage=100`

2.1. Behavior

The adaptive threshold tracker monitors the status of feed entities currently in alert due to delay threshold breach (anomaly_reason=delay_threshold_breached).

This tracker invokes the command trackmesplkadaptivedelay for entities matching specific conditions, which then investigates historical metrics collected by TrackMe using Machine Learning. TrackMe uses the density function to calculate the UpperBound value per entity, and automatically updates entities when appropriate.

Entities filtering, when the trackers loads, it will take into account entities as:

- monitored_state is "enabled", status is "red"
- anomaly_reason is "delay_threshold_breached"
- data_override_lagging_class is "false" and allow_adaptive_delay is "true"
- current_delay is more than min_delay_sec (argument submitted to the adaptive tracker, 3600 seconds by default)

Entities selection behavior can be summarized in the following mindmap:



In Addition:

- TrackMe verifies its internal processing logs to define the list of entities previously managed in the past 7 days.
- Entities that were updated since the past 4 hours will not yet be reviewed, to allow some time before re-inspecting potential thresholds after an update.
- Entities that were updated since more than 4 hours, within the past 24 hours and where the threshold value was increased are added for further review.
- Beyond these conditions, entities that were updated during the past 7, 15 or 30 days depending on the settings are regularly reviewed and updated if needed and depending on the conditions and the stability of the feeds.

Entities review behavior can be summarized in the following mindmap:



2.2 Dynamic Threshold Logic Attribution

As a basis, TrackMe automatically runs the following mstats search (over 30 days of metrics by default), example:

```
| mstats latest(trackme.splk.feeds.lag_event_sec) as lag_event_sec where `trackme_
↪metrics_idx(mytenant)` tenant_id="mytenant" object_category="splk-dsm" object=
↪"myobject" by object span=5m

``` ML calculations for this object ```
| fit DensityFunction lag_event_sec lower_threshold=0.005 upper_threshold=0.005 by␣
↪object
| rex field=BoundaryRanges "(-Infinity:(?<LowerBound>[\d|\.]*))|((?<UpperBound>[\d|\.
↪]*):Infinity)"
| foreach LowerBound UpperBound [ eval <<FIELD>> = if(isnum('<<FIELD>>'), '<<FIELD>>',
↪ 0) ]
| fields _time lag_event_sec LowerBound UpperBound

``` retain the UpperBound and perform additional calculations ```
| stats first(UpperBound) as UpperBound, perc95(lag_event_sec) as perc95_lag_event_
↪sec, min(lag_event_sec) as min_lag_event_sec, max(lag_event_sec) as max_lag_event_
↪sec, stdev(lag_event_sec) as stdev_lag_event_sec | eval UpperBound=round(UpperBound,
↪ 0)
| foreach *_lag_event_sec [ eval <<FIELD>> = round('<<FIELD>>', 0) ]

``` round by the hour, and go at the next hour range ```
| eval adaptive_delay = (round(UpperBound/3600, 0) * 3600) + 3600, adaptive_delay_
↪duration = tostring(adaptive_delay, "duration")
```

When reviewing entities for further re-processing after an initial update, TrackMe uses a more sophisticated variation of this logic which works against different time frames (-24h, -7, -30d) and aggregates the results, this allows TrackMe to beter take into account a feed
```

that returns to stability after an interruption and in a shorter time frame:

```
| mstats latest(trackme.splk.feeds.lag_event_sec) as lag_event_sec where `trackme_
↪metrics_idx(01-feeds)` tenant_id="mytenant" object_category="splk-dsm" object=
↪"myobject" earliest="-30d" latest="now" by object span=5m

``` ML calculations for this object ```
| fit DensityFunction lag_event_sec lower_threshold=0.005 upper_threshold=0.005 by␣
↪object
| rex field=BoundaryRanges "(-Infinity:(?<LowerBound>[\d|\.]*))|((?<UpperBound>[\d|\.
↪]*):Infinity)"
| foreach LowerBound UpperBound [ eval <<FIELD>> = if(isnum('<<FIELD>>'), '<<FIELD>>',
↪ 0) ]
| fields _time object lag_event_sec LowerBound UpperBound

``` retain the UpperBound and perform additional calculations ```
| stats first(UpperBound) as UpperBound, perc95(lag_event_sec) as perc95_lag_event_
↪sec, min(lag_event_sec) as min_lag_event_sec, max(lag_event_sec) as max_lag_event_
↪sec, stdev(lag_event_sec) as stdev_lag_event_sec by object | eval␣
↪UpperBound=round(UpperBound, 0)
| foreach *_lag_event_sec [ eval <<FIELD>> = round('<<FIELD>>', 0) ]

``` round by the hour, and go at the next hour range ```
| eval adaptive_delay = (round(UpperBound/3600, 0) * 3600) + 3600, adaptive_delay_
↪duration = tostring(adaptive_delay, "duration")

``` rename ```
| rename LowerBound as LowerBound_30d, UpperBound as UpperBound_30d, perc95_lag_event_
↪sec as perc95_lag_event_sec_30d, min_lag_event_sec as min_lag_event_sec_30d, max_
↪lag_event_sec as max_lag_event_sec_30d, stdev_lag_event_sec as stdev_lag_event_sec_
↪30d, adaptive_delay as adaptive_delay_30d, adaptive_delay_duration as adaptive_
↪delay_duration_30d

| join type=outer object [

| mstats latest(trackme.splk.feeds.lag_event_sec) as lag_event_sec where `trackme_
↪metrics_idx(01-feeds)` tenant_id="mytenant" object_category="splk-dsm" object=
↪"myobject" earliest="-7d" latest="now" by object span=5m

``` ML calculations for this object ```
| fit DensityFunction lag_event_sec lower_threshold=0.005 upper_threshold=0.005 by␣
↪object
| rex field=BoundaryRanges "(-Infinity:(?<LowerBound>[\d|\.]*))|((?<UpperBound>[\d|\.
↪]*):Infinity)"
| foreach LowerBound UpperBound [ eval <<FIELD>> = if(isnum('<<FIELD>>'), '<<FIELD>>',
↪ 0) ]
| fields _time object lag_event_sec LowerBound UpperBound

``` retain the UpperBound and perform additional calculations ```
| stats first(UpperBound) as UpperBound, perc95(lag_event_sec) as perc95_lag_event_
↪sec, min(lag_event_sec) as min_lag_event_sec, max(lag_event_sec) as max_lag_event_
↪sec, stdev(lag_event_sec) as stdev_lag_event_sec by object | eval␣
↪UpperBound=round(UpperBound, 0)
| foreach *_lag_event_sec [ eval <<FIELD>> = round('<<FIELD>>', 0) ]

``` round by the hour, and go at the next hour range ```
| eval adaptive_delay = (round(UpperBound/3600, 0) * 3600) + 3600, adaptive_delay_
```

(continues on next page)
```

(continued from previous page)

```
↪duration = tostring(adaptive_delay, "duration")

``` rename ```
| rename LowerBound as LowerBound_7d, UpperBound as UpperBound_7d, perc95_lag_event_
↪sec as perc95_lag_event_sec_7d, min_lag_event_sec as min_lag_event_sec_7d, max_lag_
↪event_sec as max_lag_event_sec_7d, stdev_lag_event_sec as stdev_lag_event_sec_7d,␣
↪adaptive_delay as adaptive_delay_7d, adaptive_delay_duration as adaptive_delay_
↪duration_7d

]

| join type=outer object [

| mstats latest(trackme.splk.feeds.lag_event_sec) as lag_event_sec where `trackme_
↪metrics_idx(01-feeds)` tenant_id="mytenant" object_category="splk-dsm" object=
↪"myobject" earliest="-24h" latest="now" by object span=5m

``` ML calculations for this object ```
| fit DensityFunction lag_event_sec lower_threshold=0.005 upper_threshold=0.005 by␣
↪object
| rex field=BoundaryRanges "(-Infinity:(?<LowerBound>[\d|\.]*))|((?<UpperBound>[\d|\.
↪]*):Infinity)"
| foreach LowerBound UpperBound [ eval <<FIELD>> = if(isnum('<<FIELD>>'), '<<FIELD>>',
↪ 0) ]
| fields _time object lag_event_sec LowerBound UpperBound

``` retain the UpperBound and perform additional calculations ```
| stats first(UpperBound) as UpperBound, perc95(lag_event_sec) as perc95_lag_event_
↪sec, min(lag_event_sec) as min_lag_event_sec, max(lag_event_sec) as max_lag_event_
↪sec, stdev(lag_event_sec) as stdev_lag_event_sec by object | eval␣
↪UpperBound=round(UpperBound, 0)
| foreach *_lag_event_sec [ eval <<FIELD>> = round('<<FIELD>>', 0) ]

``` round by the hour, and go at the next hour range ```
| eval adaptive_delay = (round(UpperBound/3600, 0) * 3600) + 3600, adaptive_delay_
↪duration = tostring(adaptive_delay, "duration")

``` rename ```
| rename LowerBound as LowerBound_24h, UpperBound as UpperBound_24h, perc95_lag_event_
↪sec as perc95_lag_event_sec_24h, min_lag_event_sec as min_lag_event_sec_24h, max_
↪lag_event_sec as max_lag_event_sec_24h, stdev_lag_event_sec as stdev_lag_event_sec_
↪24h, adaptive_delay as adaptive_delay_24h, adaptive_delay_duration as adaptive_
↪delay_duration_24h

]

``` aggregate the UpperBound, if for any reason one the UpperBound is not returned as␣
↪expected, we will use the 7d value ```
| eval UpperBound=case(
isnum(UpperBound_30d) AND isnum(UpperBound_7d) AND isnum(UpperBound_24h),␣
↪avg(UpperBound_30d+UpperBound_7d+UpperBound_24h/3, 2),
1=1, UpperBound_7d
)
| eval adaptive_delay = (round(UpperBound/3600, 0) * 3600) + 3600, adaptive_delay_
↪duration = tostring(adaptive_delay, "duration")
```

TrackMe carefuly logs the searches performed as well as their result.

2.3 Tracker Level Key Arguments

min_delay_sec

This defines the minimum delay value in seconds for entities to be considered (2 hours by default).

```
min_delay_sec=3600
```

max_auto_delay_sec

This defines the maximal delay value that the adaptive backend can set, if the automated delay calculation goes beyond it, this value will be used instead, expressed in seconds.

Behavior change in TrackMe 2.0.84

- Before this version, the behavior was to refuse updating entities if the calculation was leading to a superior value than the max_auto_delay_sec.
- From this version, TrackMe will instead use this value and will update the entity, which then enters the cycle of review automatically.

```
max_auto_delay_sec=604800
```

max_changes_past_7days

This defines the maximal number of changes that can be performed in a 7 days time frame, once reached we will not update this entity again until the counter is reset.

```
max_changes_past_7days=10
```

min_historical_metrics_days

The minimal number of accumulated days of metrics before we start updating the delay threshold, expressed in days.

```
min_historical_metrics_days=7
```

review_period_no_days

This argument was introduced in TrackMe 2.0.96, it allows you to control the period of time for the review of TrackMe entities that were updated.

```
review_period_no_days=30
```

2.4 Updating Delay Thresholds Automatically

After performing these investigations, the command updates the delay threshold value for selected entities, and generates an audit record with corresponding results (context: automated adaptive delay update). Audit messages can be found with the following search:

```
`trackme_audit_idx` tenant_id=* "automated adaptive delay update"
| table _time, tenant_id, object_category, object, action, comment
| sort - 0 _time | trackmeprettyjson fields=comment
```

2.5 Adaptive tracker output & Activity log traces

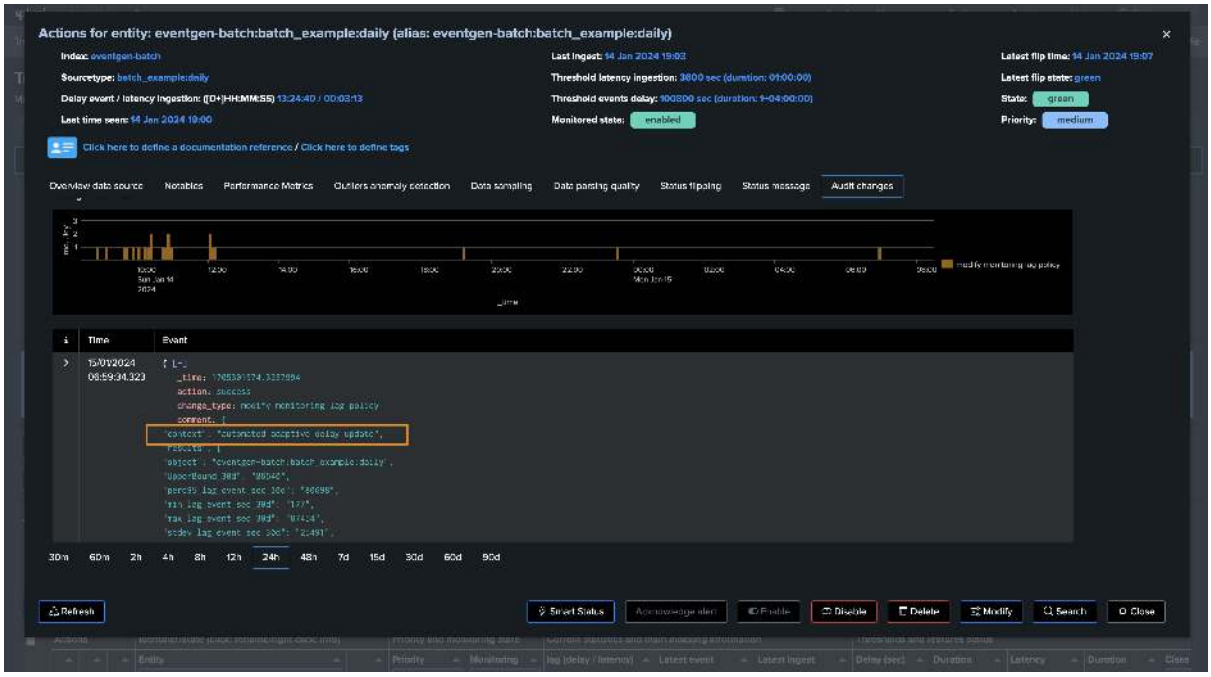
The activity log traces can be found with the following search:

```
index=_internal sourcetype=trackme:custom_commands:trackmesplkadaptivedelay
```

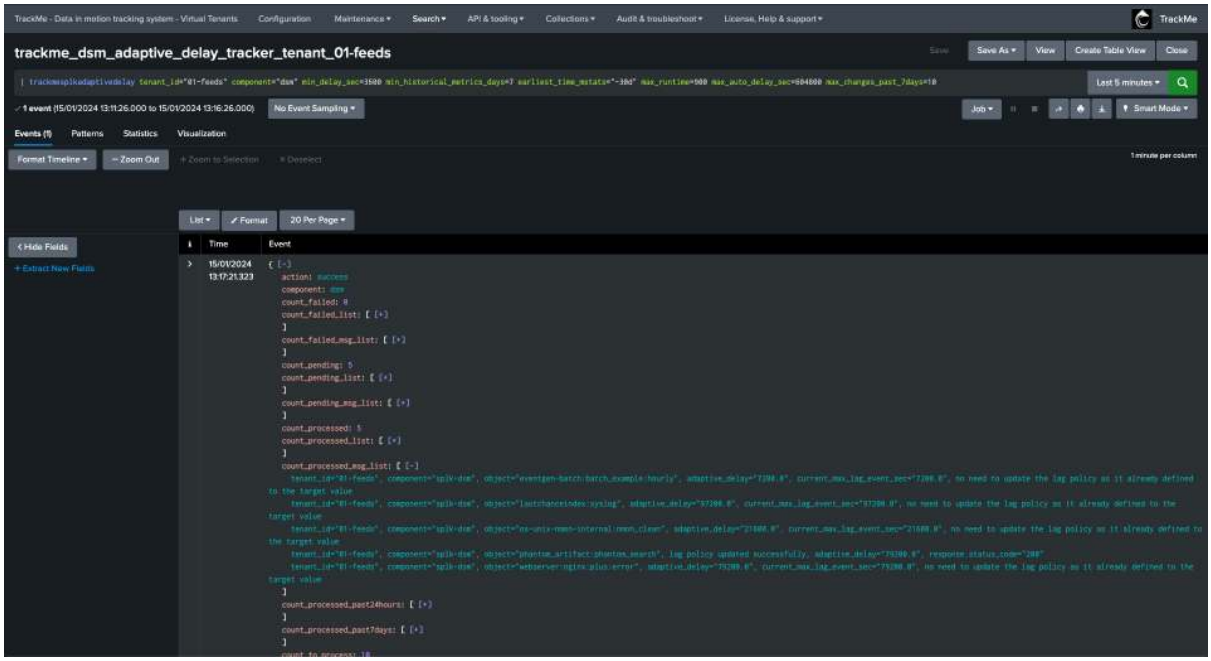
To easily review decisions made for a given entity, add the object as part of the search filter:

```
index=_internal sourcetype=trackme:custom_commands:trackmesplkadaptivedelay object=
↪"myobject"
```

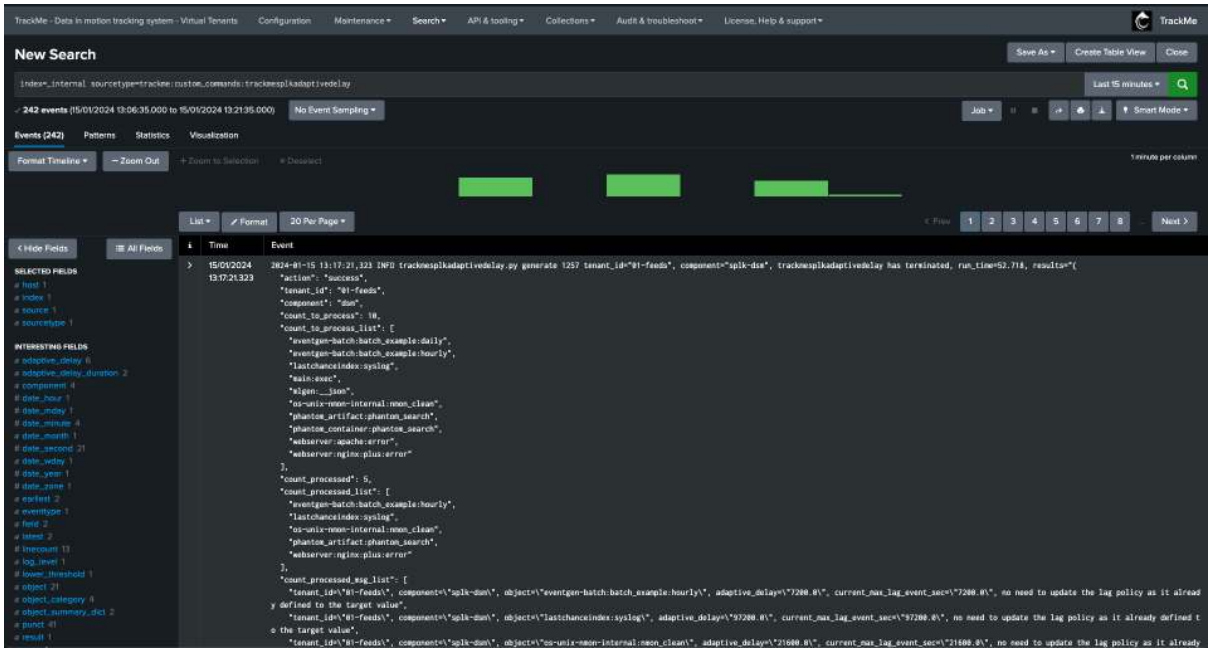
TrackMe also generates an audit event every time the Adaptive Threshold performs an update, you can find this easily via TrackMe's user interface:



When the Tracker runs, it register in its output every entity that was considered as well as actions performed for it:

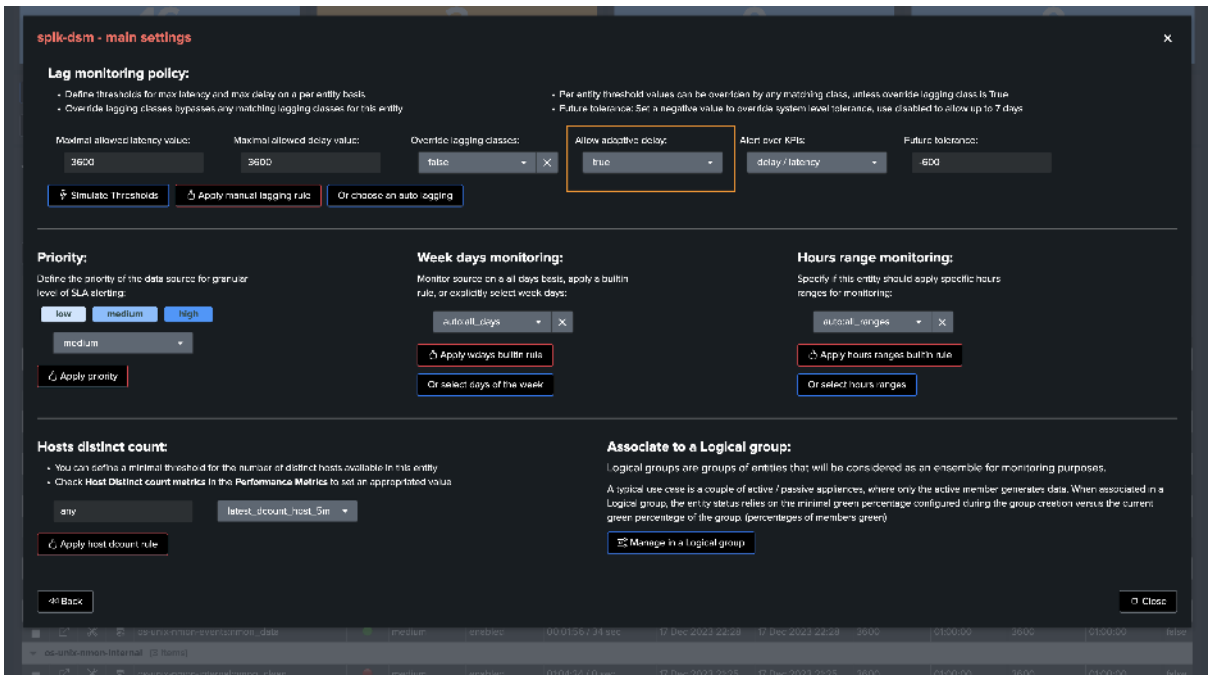


The job output is also registered in the activity logs:



2.6 Preventing an Entity from Being Automatically Managed

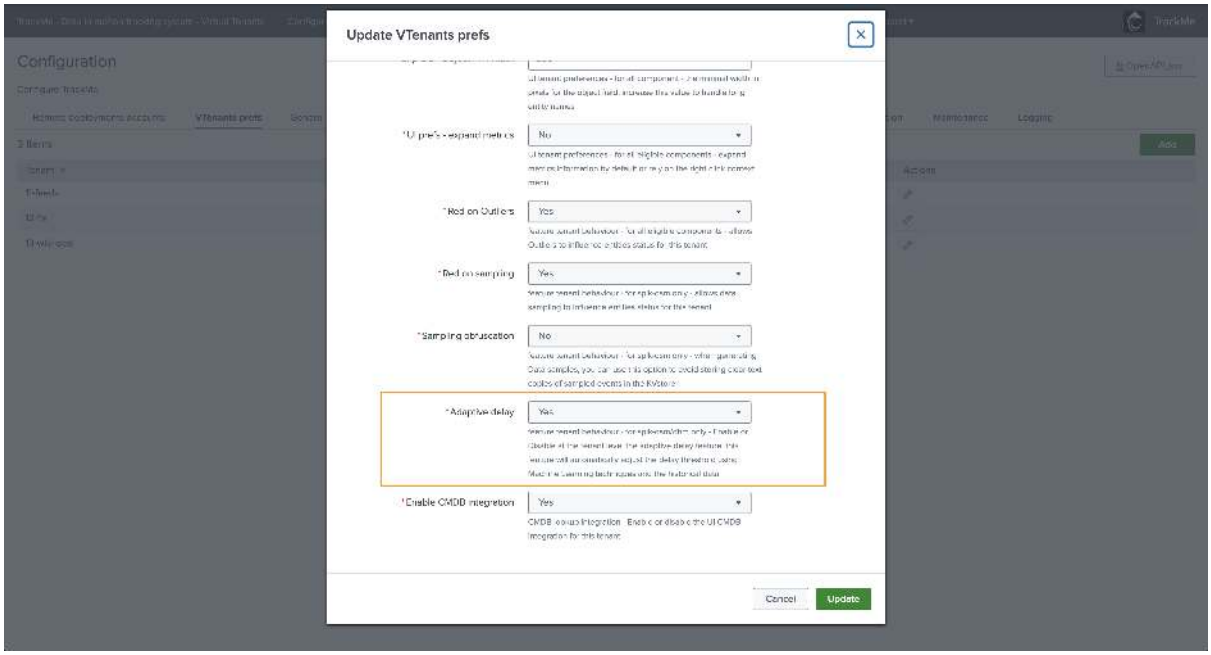
Via the UI, you can set the value of allow_adaptive_delay to False, which prevents TrackMe from automatically updating the delay threshold for a given entity.



2.7 Disabling Adaptive Delay Thresholding at the Tenant Level

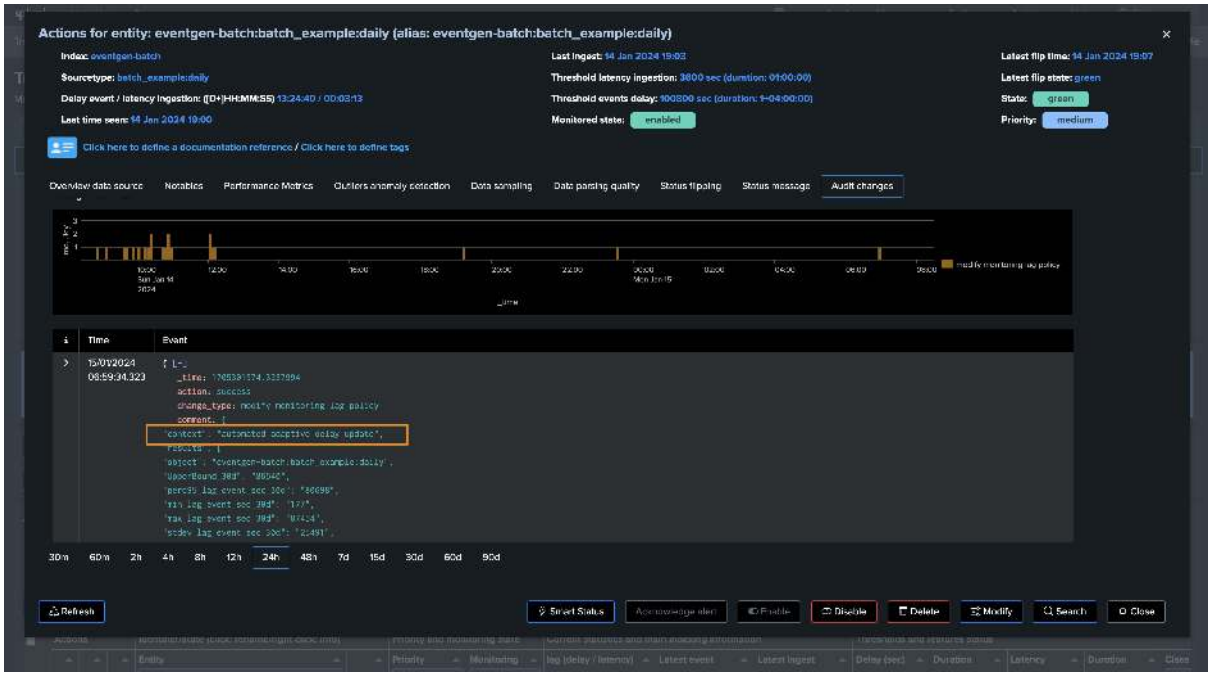
Since TrackMe V2.0.75, you can disable the adaptive delay thresholding feature at the Virtual Tenant account level (setting: adaptive_delay).

Go in Configure / VTenants prefs:



2.8 Audit dashboard

Review the audit dashboard called "TrackMe - Adaptive delay threshold audit" available in the menu **Audit & Troubleshoot**.

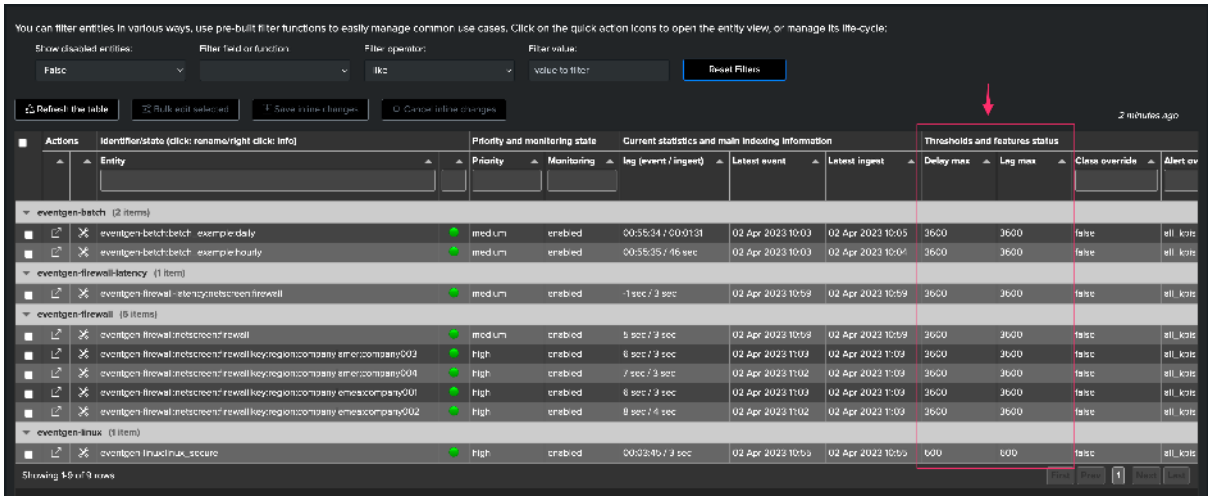


9.8.3 3. Reviewing Current Thresholds

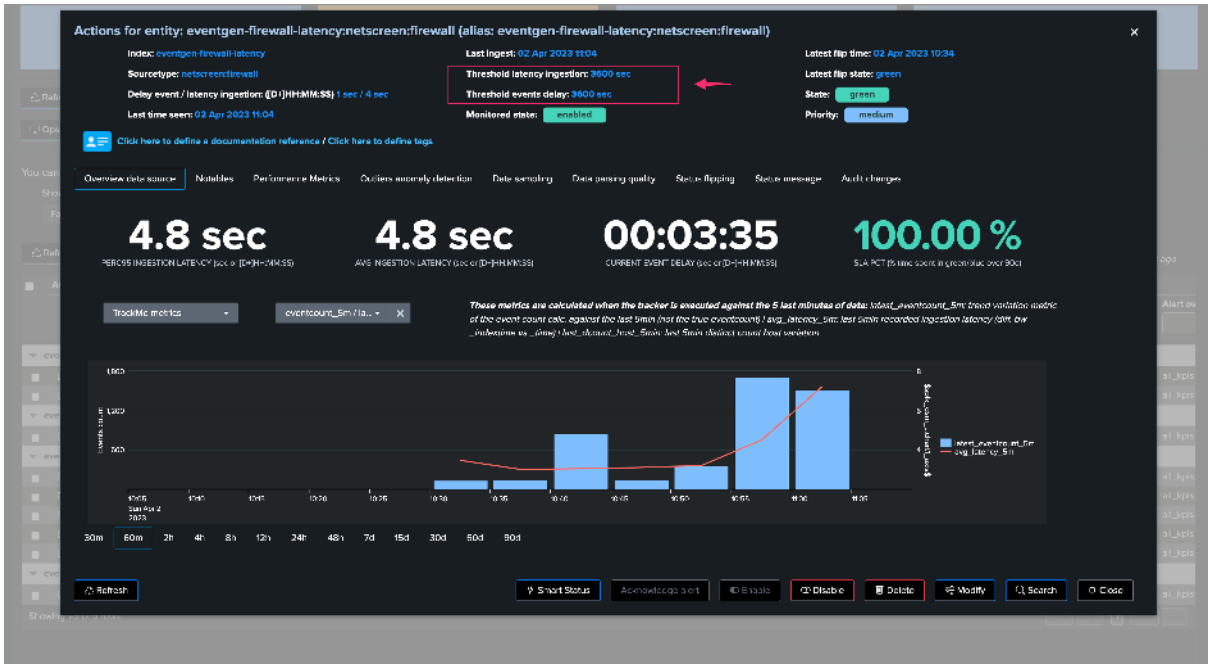
Currently set thresholds are shown in different parts of the TrackMe main user interface:

- In the main user interface of the Virtual Tenant, the Tabulator shows a two-column element showing both threshold values
- When opening the entity main screen

Viewing thresholds from the Tabulator:



Viewing thesholds from the entity view:



9.8.4 4. Defining Custom Threshold Values

There are mainly two approaches, which can be combined:

- Defining global rules that define the threshold values based on custom criteria, these are called "Lagging classes" in TrackMe
- Defining custom threshold values for a given entity, optionally overriding Lagging classes, if any

9.8.5 5. Lagging Classes for Thresholds Management

A best practice approach is to configure lagging classes.

Lagging classes can be defined using the following criteria: - Based on the index - Based on the sourcetype - Based on the priority level defined for the entity

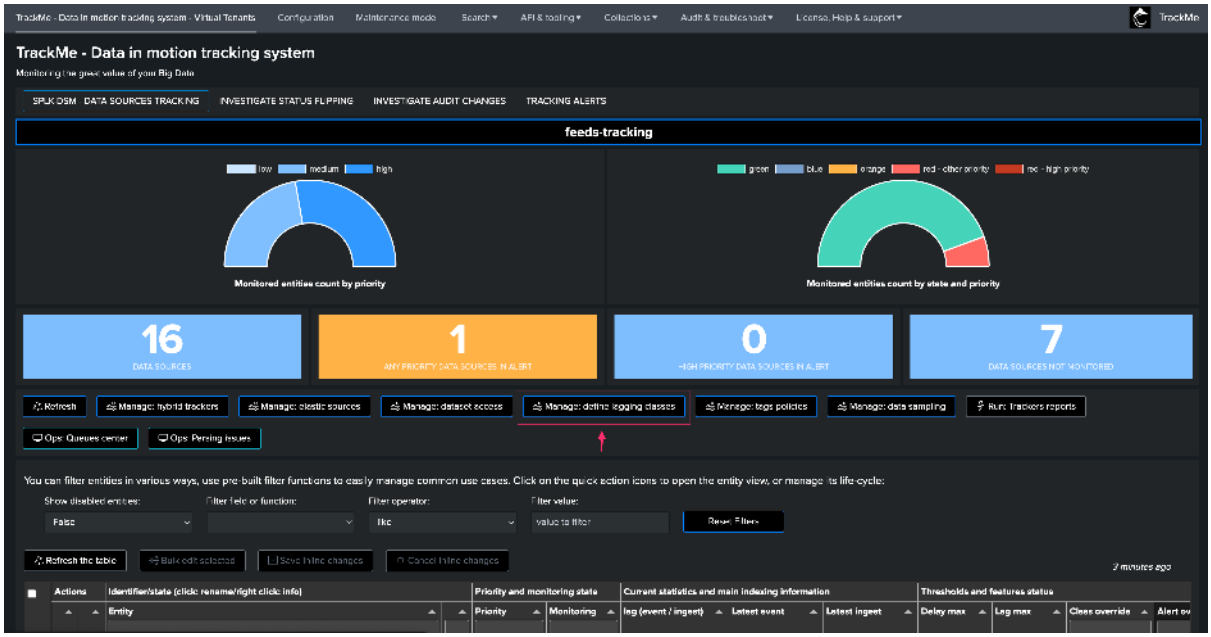
When a lagging class is defined and matches an entity, TrackMe defines the values of the thresholds accordingly.

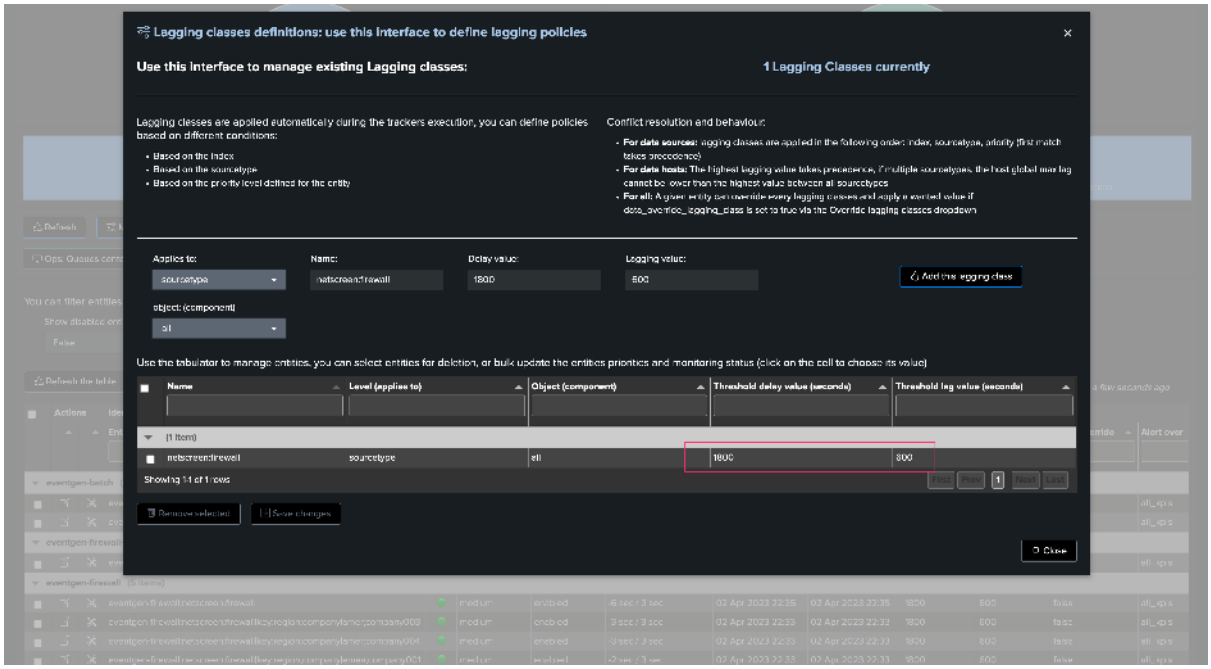
These values can be overridden on a per-entity basis, allowing managing generic use cases for a data provider while still being able to manage specific use cases per entity.

Example: defining a custom lagging class for a sourcetype:

In this example, we define a custom lagging class for the sourcetype "netscreen-firewall", with the following values:

- latency: 10 minutes (600 seconds)
- delay: 2 hours (7200 seconds)





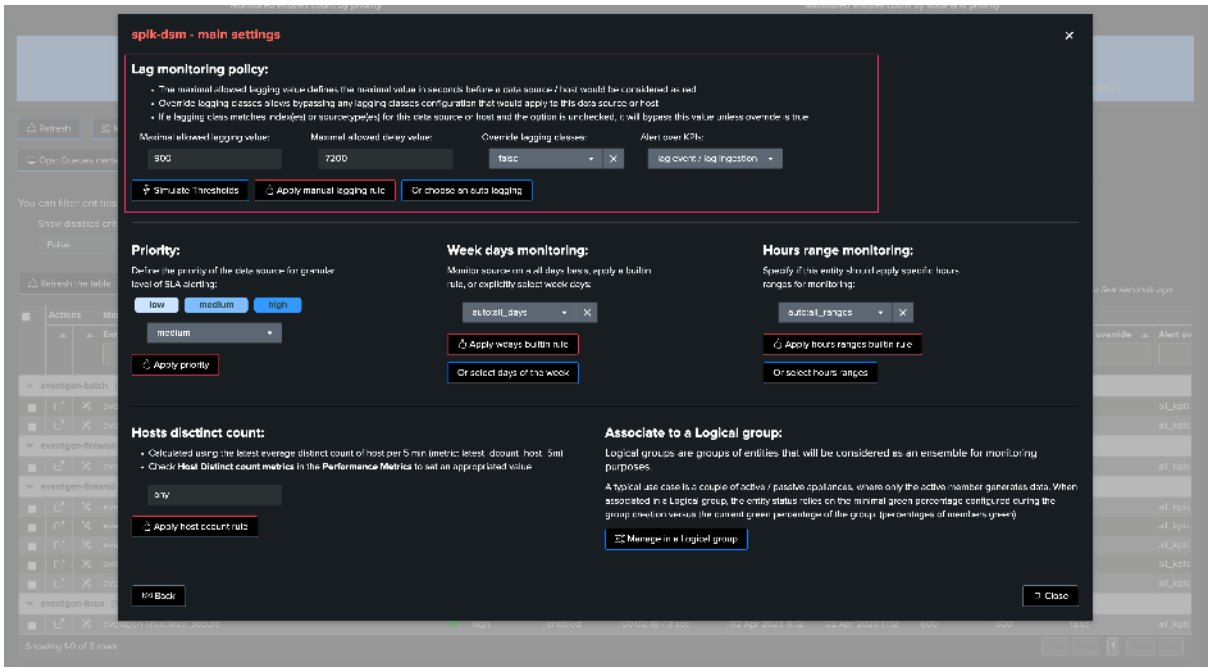
Once Hybrid Trackers have been executed at least once, and the entities are active, thresholds have been updated automatically:

9.8.6 6. Per Entity Thresholds

To manually define thresholds for a given entity, proceed as follows:

- Open the entity main screen, and access the modification screen
- On top of this screen, define the thresholds as needed
- Set the override value to True if you are using lagging classes, to avoid these values from being overridden by a system-wide rule
- Click on apply

Example: defining custom thresholds



In this screen, you can:

- set the maximal acceptable value for latency
- set the maximal acceptable value for delay
- define if we should override lagging classes, if any matching (default to false, set true if needed)
- define if we should alert on both KPIs (default is both), or only one of the two

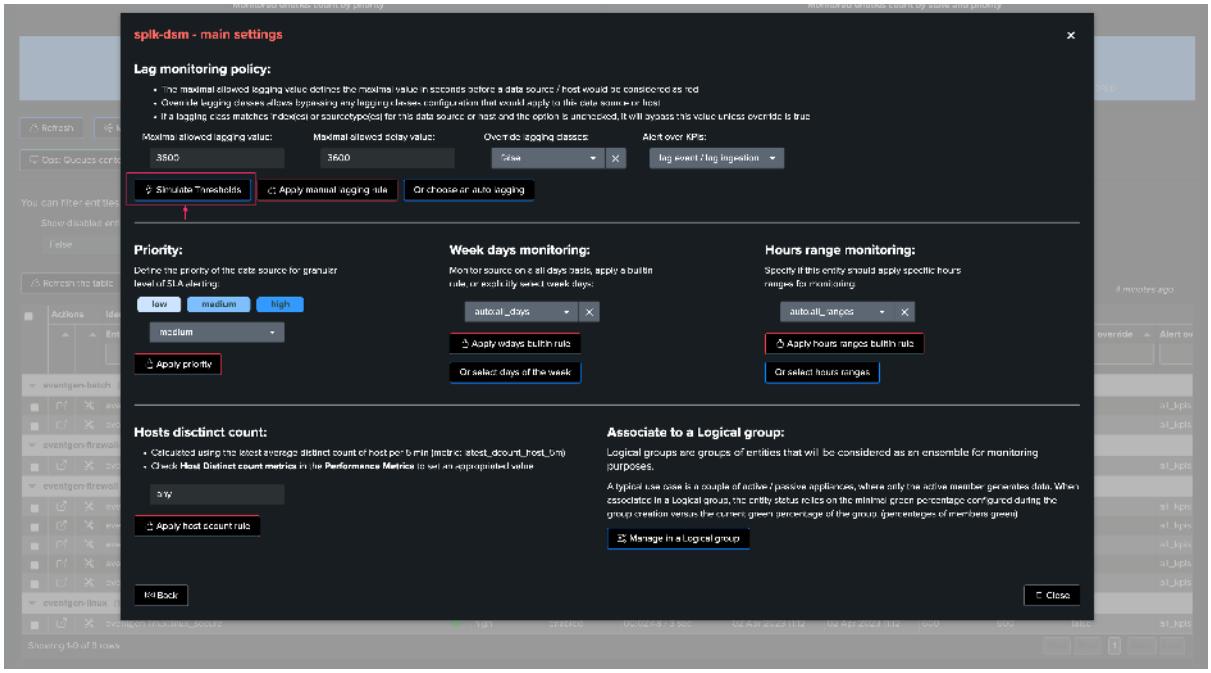
9.8.7 7. Simulating Threshold Values

TrackMe provides a feature that allows simulating how thresholds would be breached based on your inputs.

Open the thresholds settings screen, and click on the simulate thresholds button:

- Provide input values for both latency and delay thresholds
- Click on simulate, TrackMe will apply these thresholds against TrackMe summary events and show an event for each breach that would result from your settings

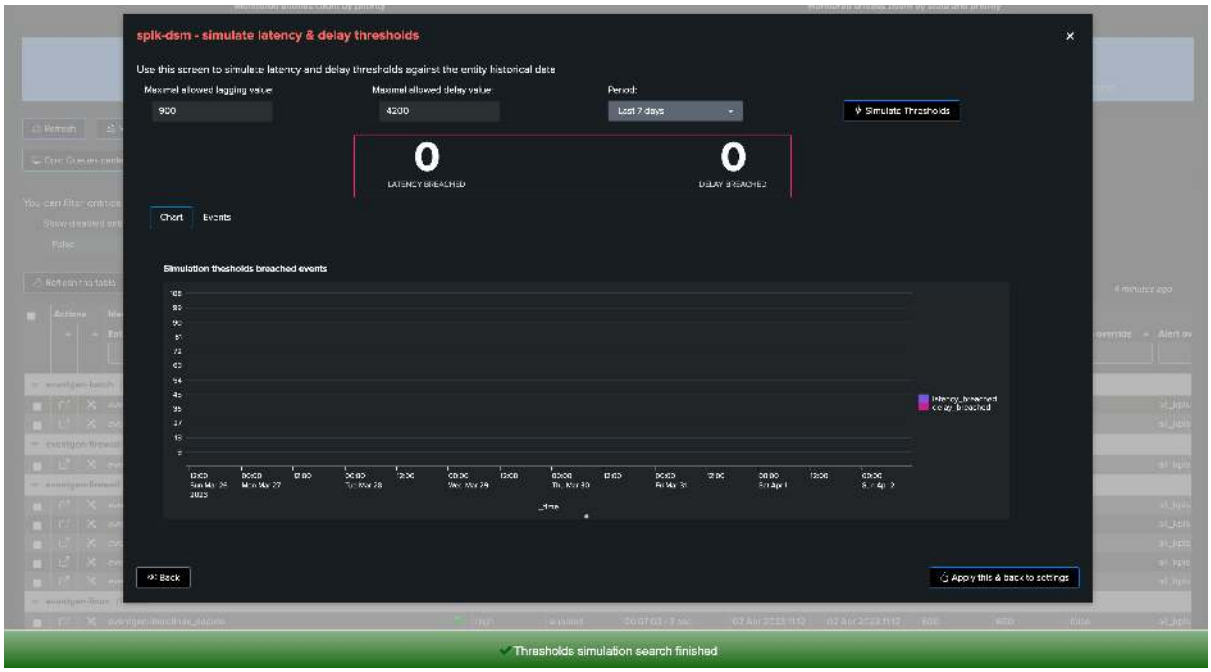
Example:



For instance, this provider sends data to Splunk once per hour, however it does not suffer from latency when it does it:

The threshold simulation screen's purpose is to use TrackMe summary events to simulate your settings against the historical knowledge TrackMe has accumulated, before you set these effectively.

In this example, as the data is generated once per hour, we could set up a small value for latency, and for instance 1 hour + 10 minutes of additional time for the delay:



Finally, we can click on the apply button to prefill our thresholds settings, and apply as needed.

9.8.8 8. Anatomy of an Entity suffering from index time Latency

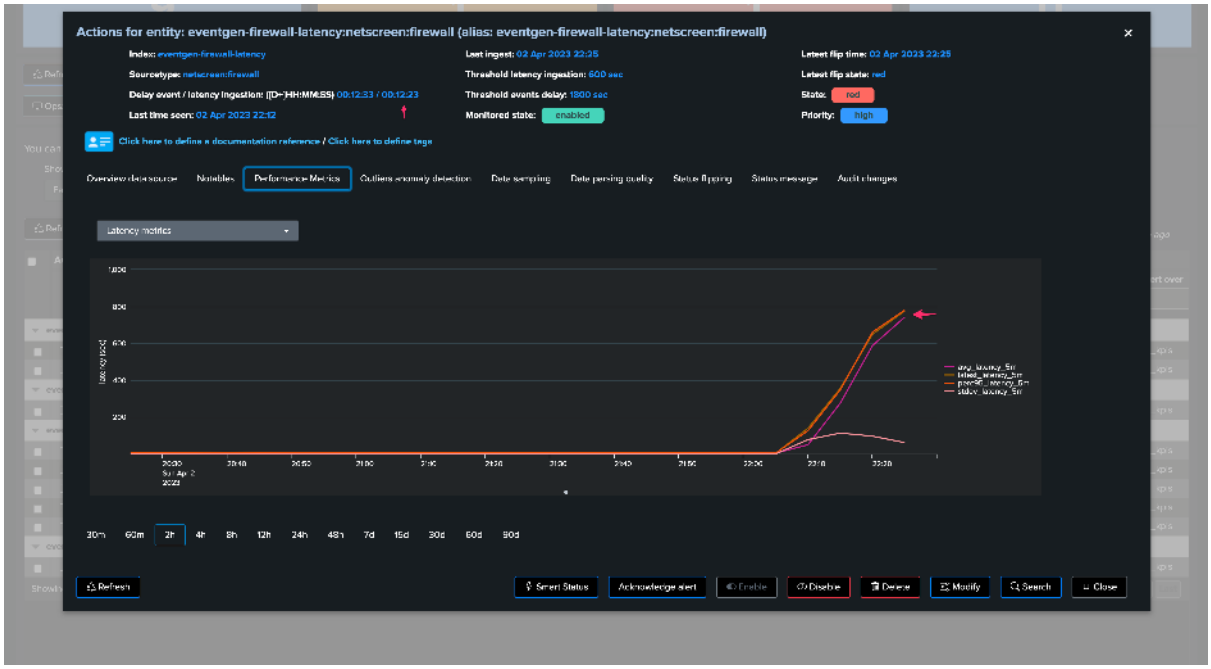
In the following example, we are reviewing an entity which is suffering from latency at the ingestion time:

Some additional details:

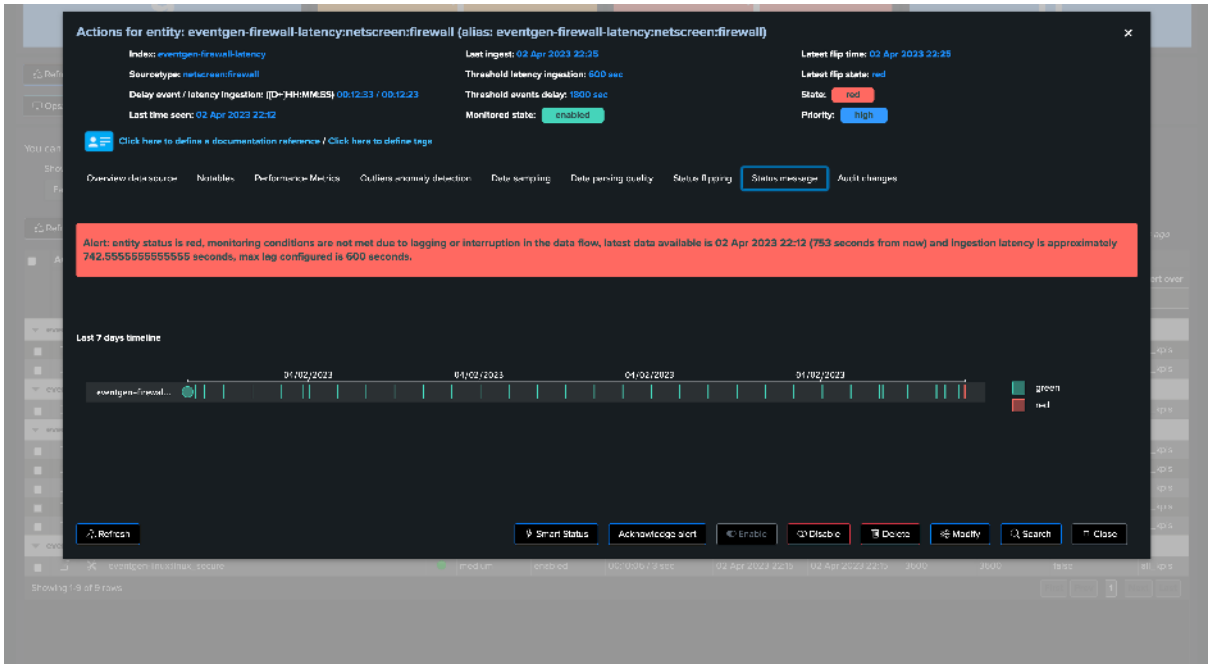
- Latency means that we receive and index events with potentially some high amounts of time between when these events were produced (based on the events timestamps) and when these were received
- This may imply as well that we have **delay**, but not necessarily, you can receive for instance a mix of pseudo real time events and events with latency
- In addition, the measures between latency and delay will differ, which is in most of the cases to be expected

The following entity breached a pre-defined set of thresholds for delay and latency, TrackMe shows:

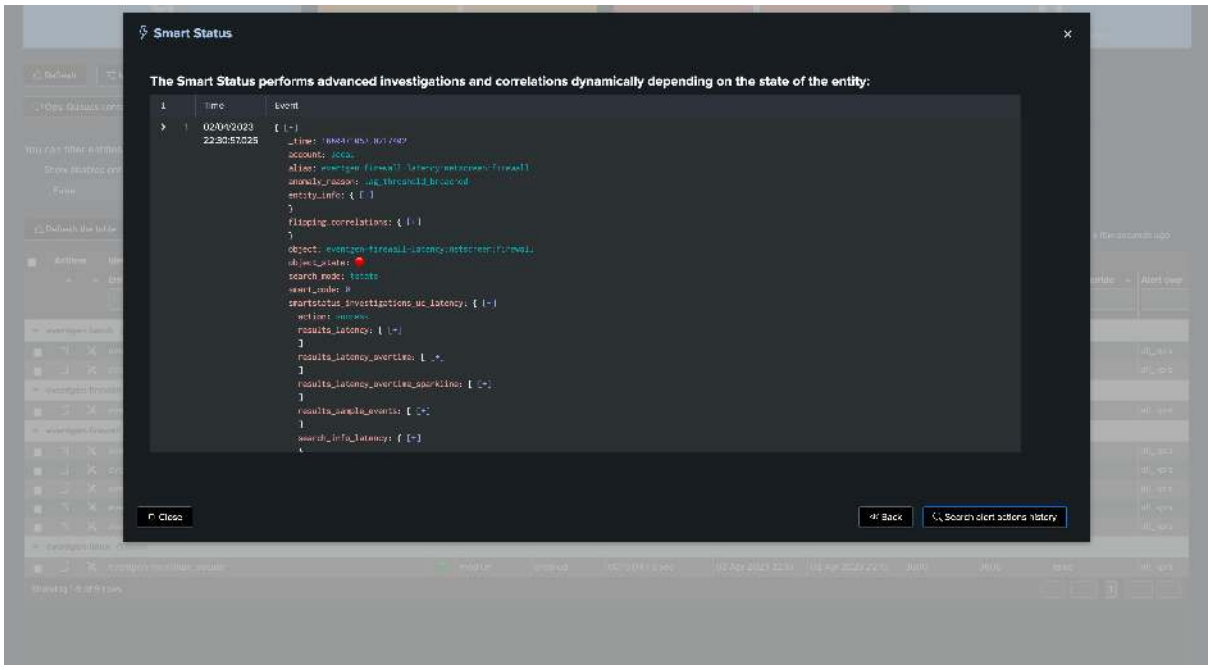
We can observe latency increasing for this entity in both the overview chart (based on TrackMe metrics and Splunk queries), as well as the Performance Metrics chart:

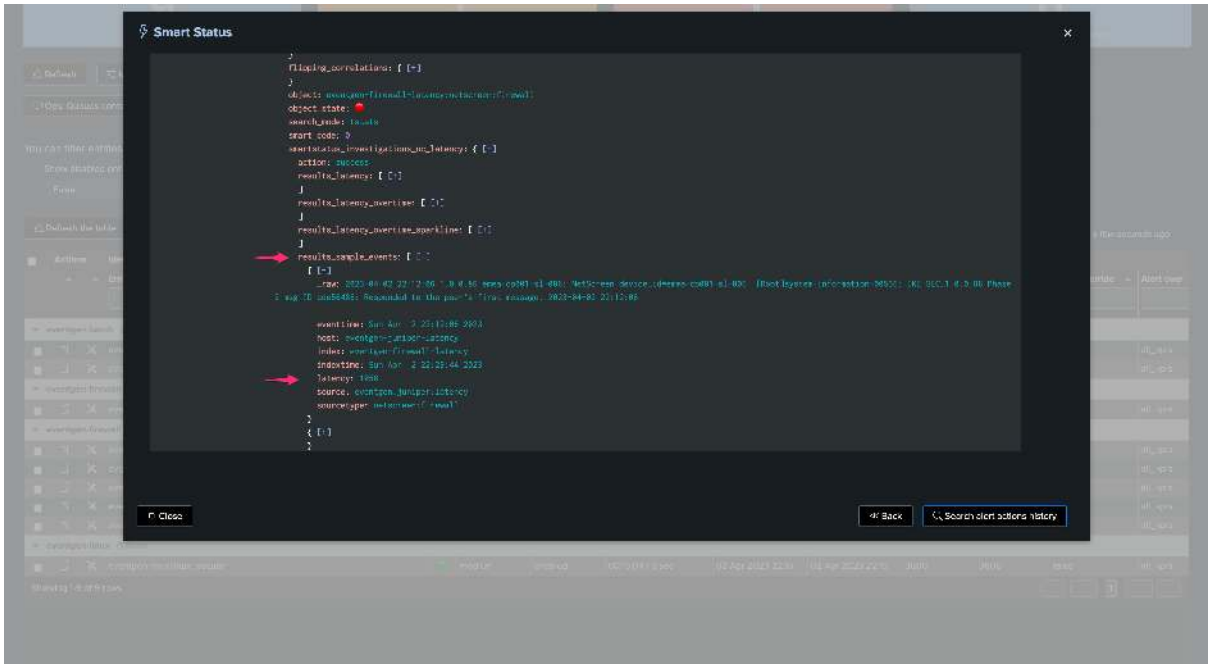


The Status Message shows a clear explanation about the issue that affects the entity:



The Smart Status function provides automated investigations too:



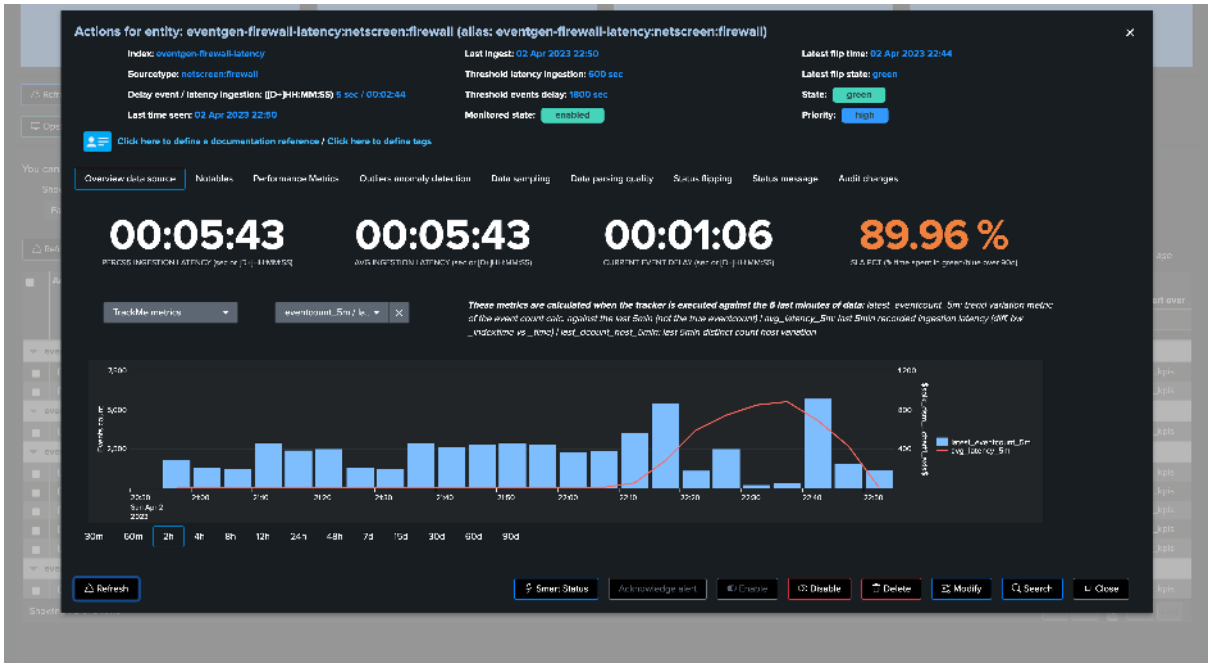


Shall we want to manually review events with latency, for instance searching for events indexed with more than 15 minutes of latency during the past 4 hours of indexed data with events within the past 4 hours:

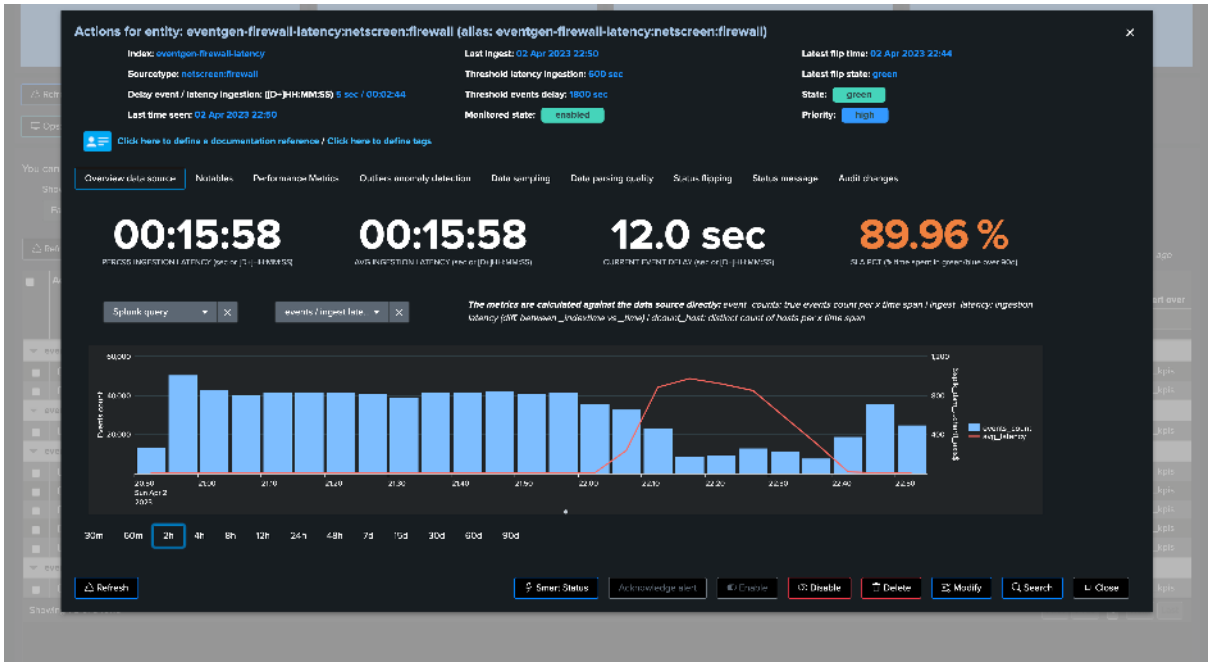
```
index=myindex sourcetype=mysourcetype _index_earliest="-4h" _index_latest=
↪"+4h" earliest="-4h" latest="+4h"
| eval indextime=_indextime, latency=(_indextime-_time)
| where latency>900
| eval indextime=strftime(indextime, "%c")
| table _time indextime latency _raw
```

Once the issue is addressed and the root cause was fixed, and the latency is back to acceptable thresholds, TrackMe will return the entity status back to green after some time:

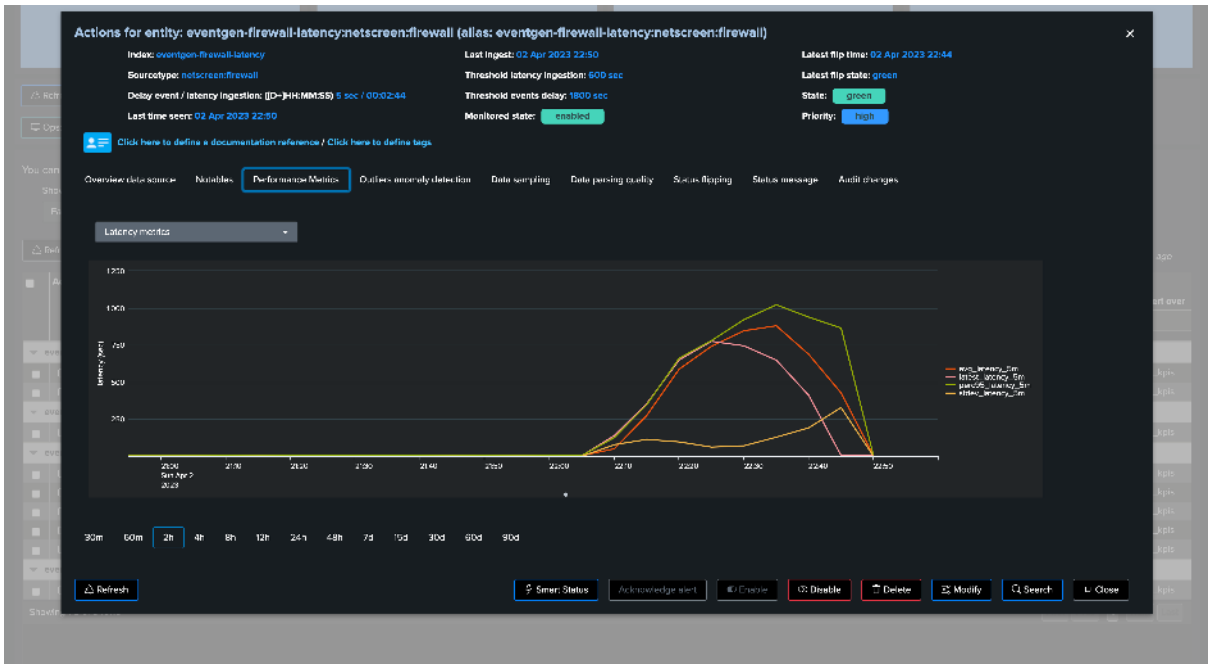
TrackMe metrics view:



Splunk query view:



Performance metrics tab:

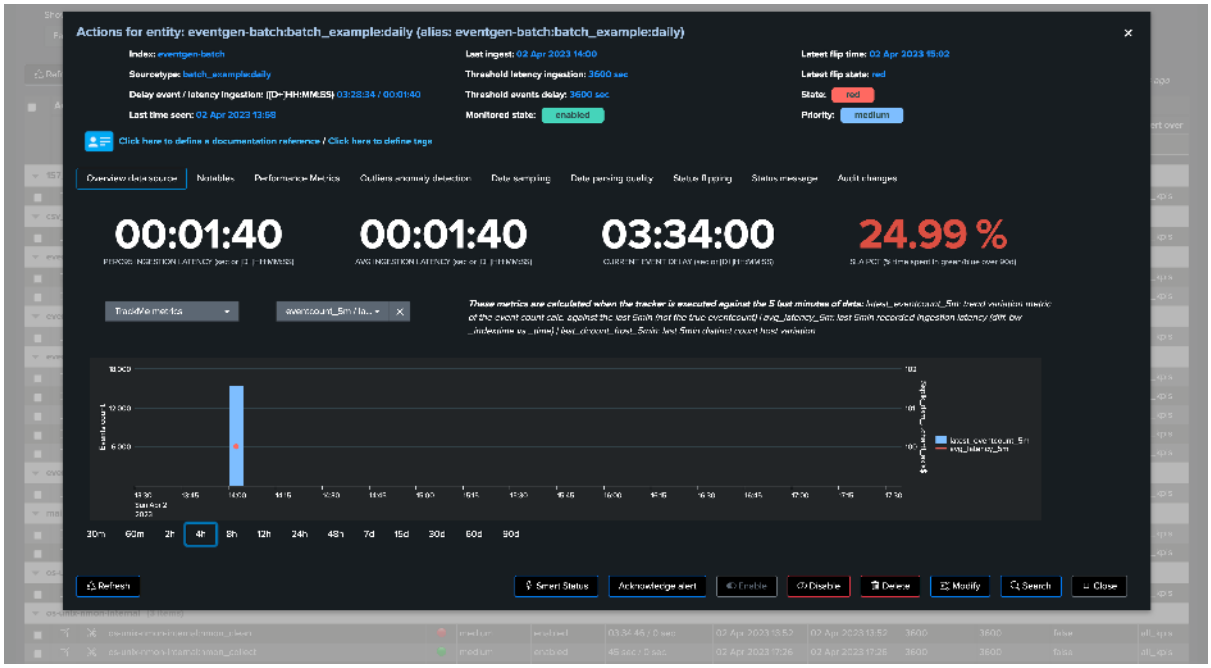


9.8.9 9. Anatomy of an Entity with Delay with no Latency

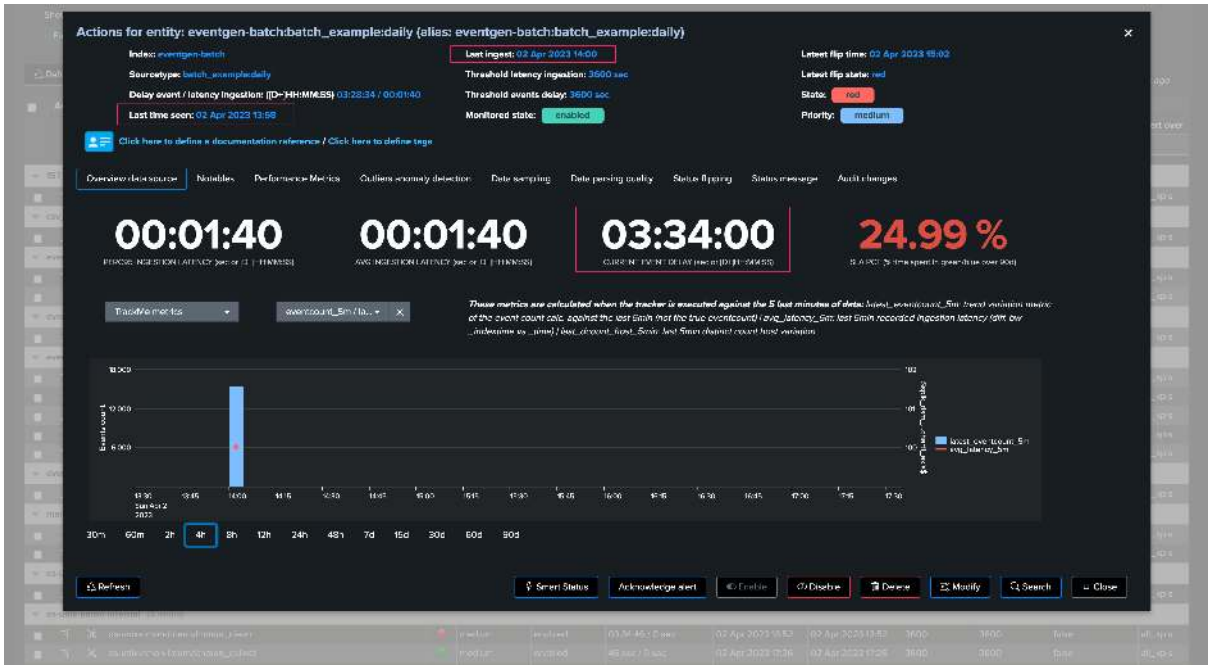
Batch data once per day example

In the following example, we are reviewing an entity which is suffering from delay:

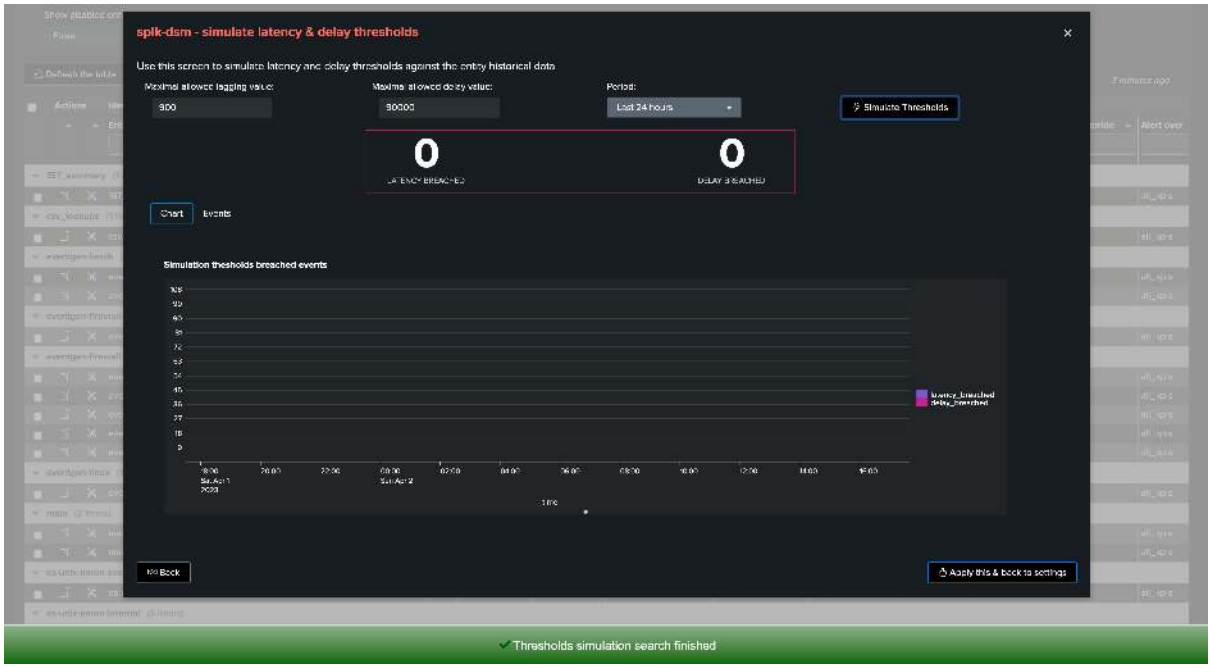
In this use case, the entity does not have any latency, however, it generates data in a batched manner, for instance this provider generates a bunch of Splunk events once per day:



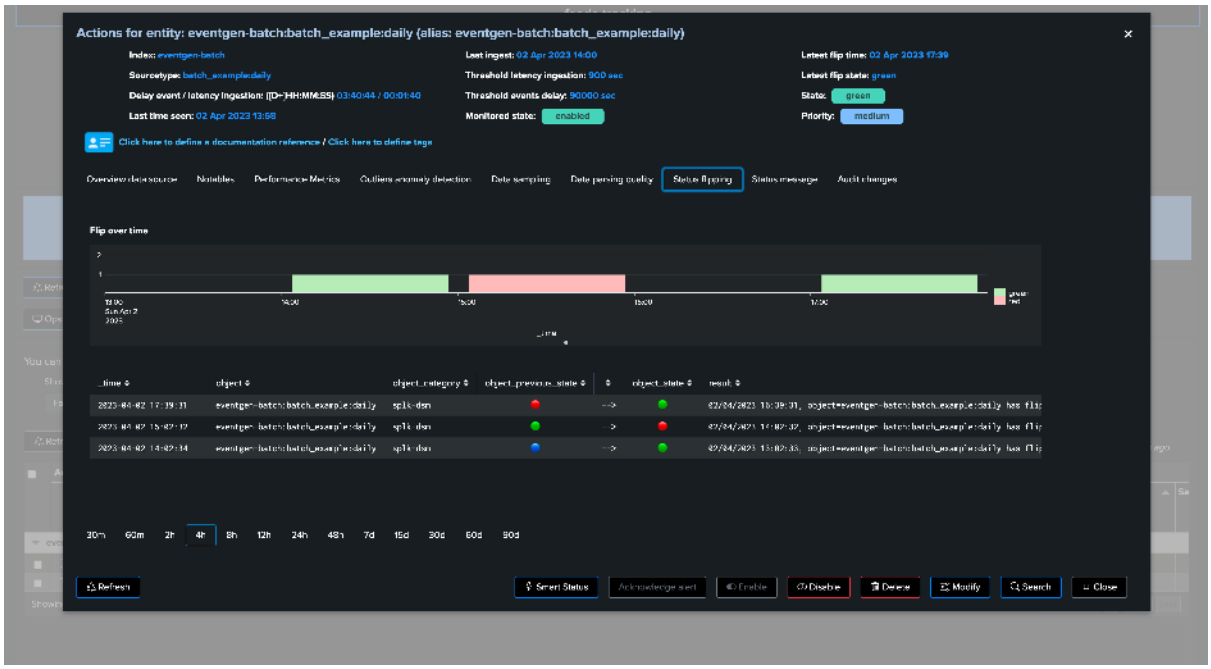
TrackMe shows the current delay, as well as the latest event (from the _time perspective) and the latest ingest event (therefore from the _indextime perspective):



In this context, we could set a max delay of a little bit more than 24 hours (say 24 hours + 10 minutes), the latency may remain low as we do not expect to ingest events with latency when these are produced:



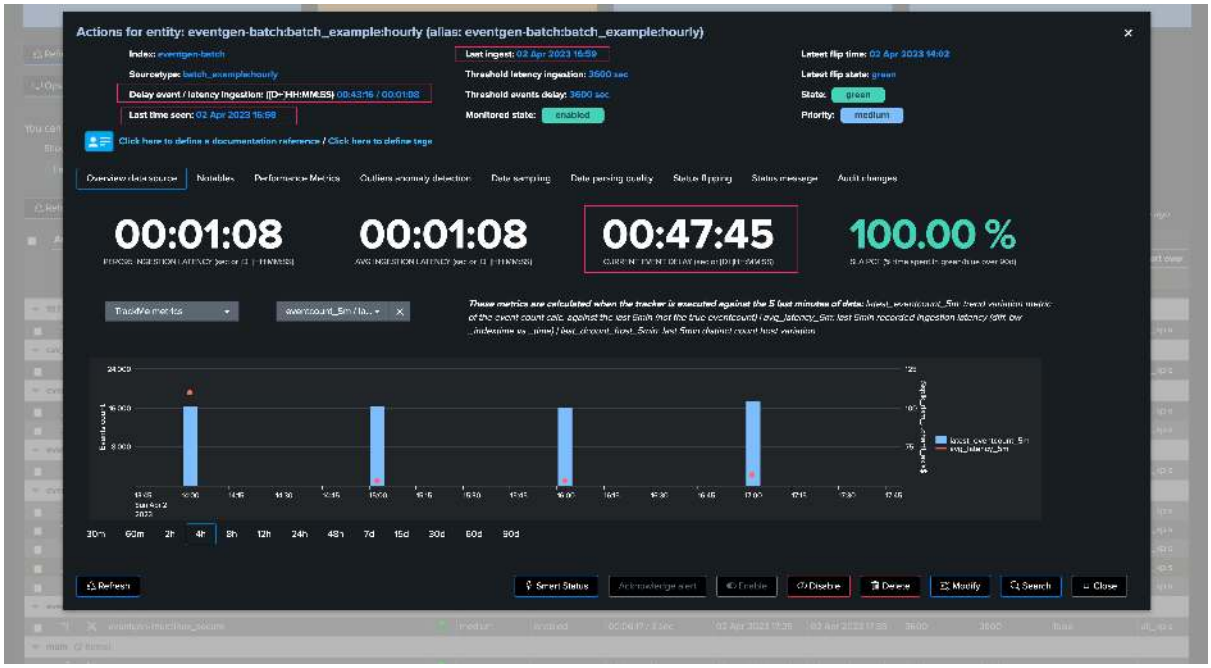
Once we apply our new thresholds, the entity returns to green and will remain healthy unless our monitoring conditions would not be met:



Batch data once per hour example

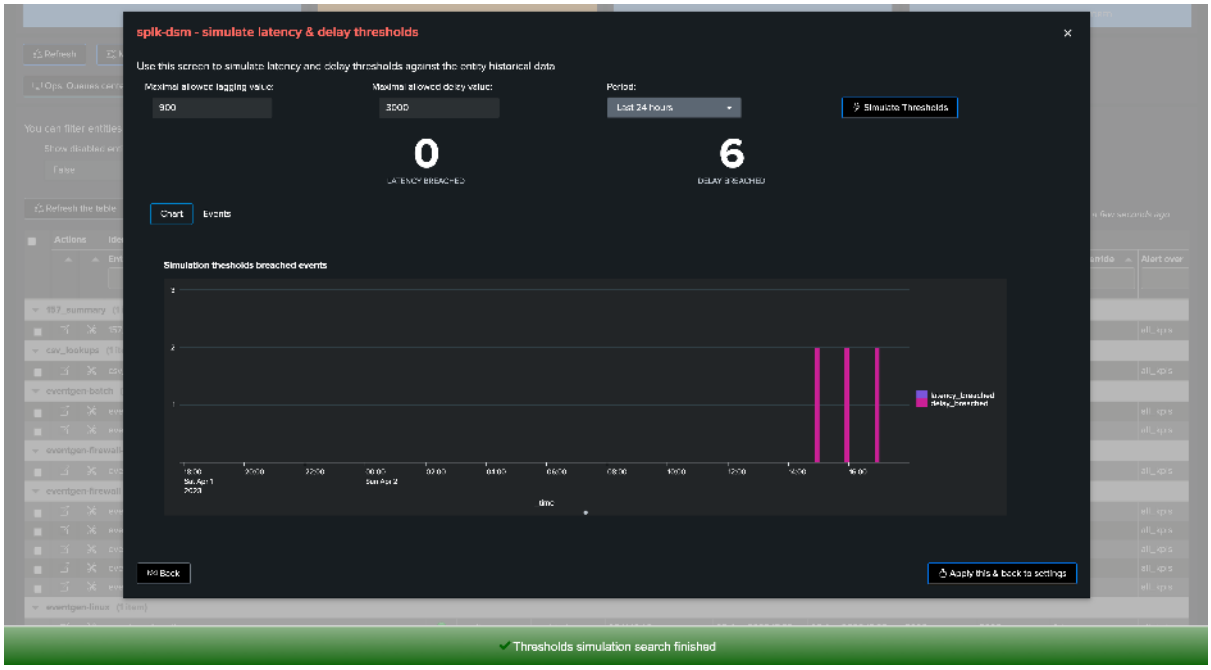
In this second example, we have another data source which generates data once per hour, similarly we may not expect latency, but it is likely that the data source alert could trigger with a default delay threshold of 3600 seconds if the provider is a little bit late:

Note that we clearly observe the delay, when data was generated and indexed, against as fast we received and indexed these events:

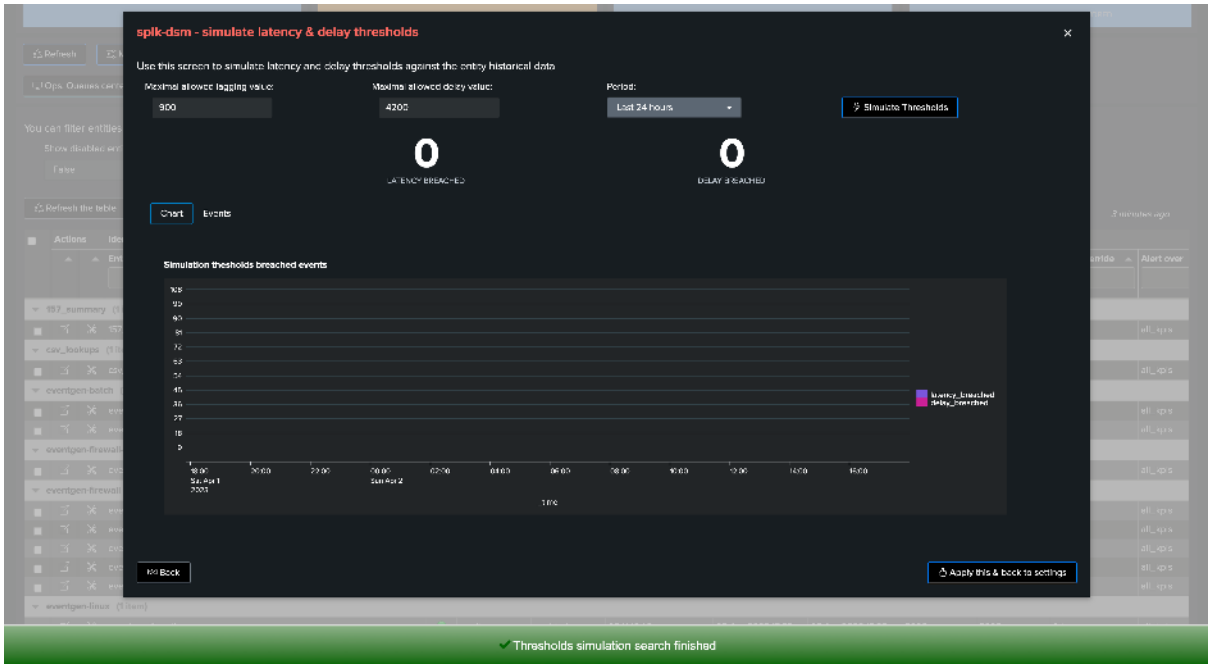


We may or may not have had alerts for this entity yet, depending on the context we may tolerate or not some levels of delay, for instance we could set to 1 hour + 10 minutes if the data is quite critical:

If we go too low (say for 3000 seconds for example), then we can observe that would get alerts for this data source due to delay:



In our example, we will set it to 4200 seconds, 1 hour and 10 minutes:



Again, accepting a certain delay in the delivery of events does not mean that these events should be indexed with latency, both KPIs need to be taken into account independently.

9.8.10 Conclusion

Thresholds definition is an important part of TrackMe configuration and entities lifecycle.

TrackMe provides different meaningful features to observe, review, and define threshold values that make sense for your context.

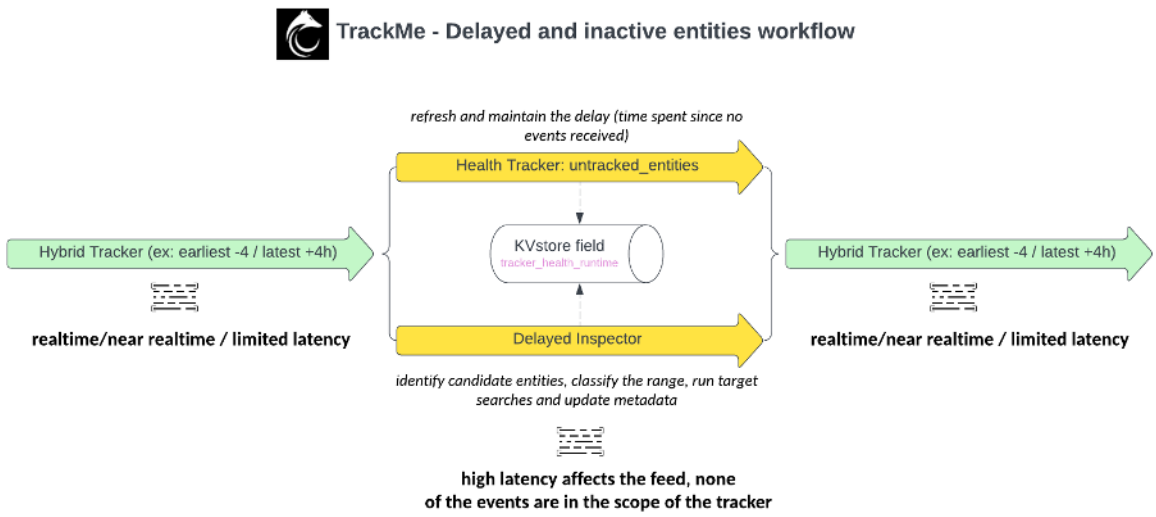
Because every context is different, TrackMe provides flexible features to allow managing latency and delay thresholds as needed.

9.9 Splunk Feeds Delayed & Inactive Entities (splk-feeds)

Hint

New delayed entities inspector introduced in TrackMe 2.1.10

- **TrackMe 2.1.10** introduced a new automated process called **delayed entities inspector**
- This new backend process applies to `splk-dsm` and `splk-dhm` components, and is automatically created with the Virtual Tenant and component.
- This new process ensures to maintain a near up to date view of entities that fall out of the scope of hybrid trackers, due to high delay and/or latency.
- It is configurable at the level of the Virtual Tenant, detects when a given entity is no longer covered by the range of an hybrid tracker, and will trigger a target search to refresh the entity metadata.
- The `delayed entities inspector` uses a multi-ranges approach to classify entities based on their current delay, which influences the frequency at which the tracker will attempt to refresh the knowledge of the entity.
- This process allows to slightly reduce the risk of false positives, and avoids confusion with out of date delay knowledge, while maintaining scalability and performance.



9.9.1 1. Introduction for delayed entities

TrackMe discovers and maintains Splunk feeds and their resulting entities using one or more primary scheduled logics, called **Hybrid Trackers**.

Hint

The following documentation describes TrackMe processes regarding the management of entities which are delayed or become inactive if these entities have not sent data anymore for a long period of time.

Trackers have a time range which defines the earliest and latest time of the events that can be processed by the tracker, for instance:

- `earliest: -4h` / `indexed earliest: -4h`
- `latest: +4h` / `indexed latest: +4h`

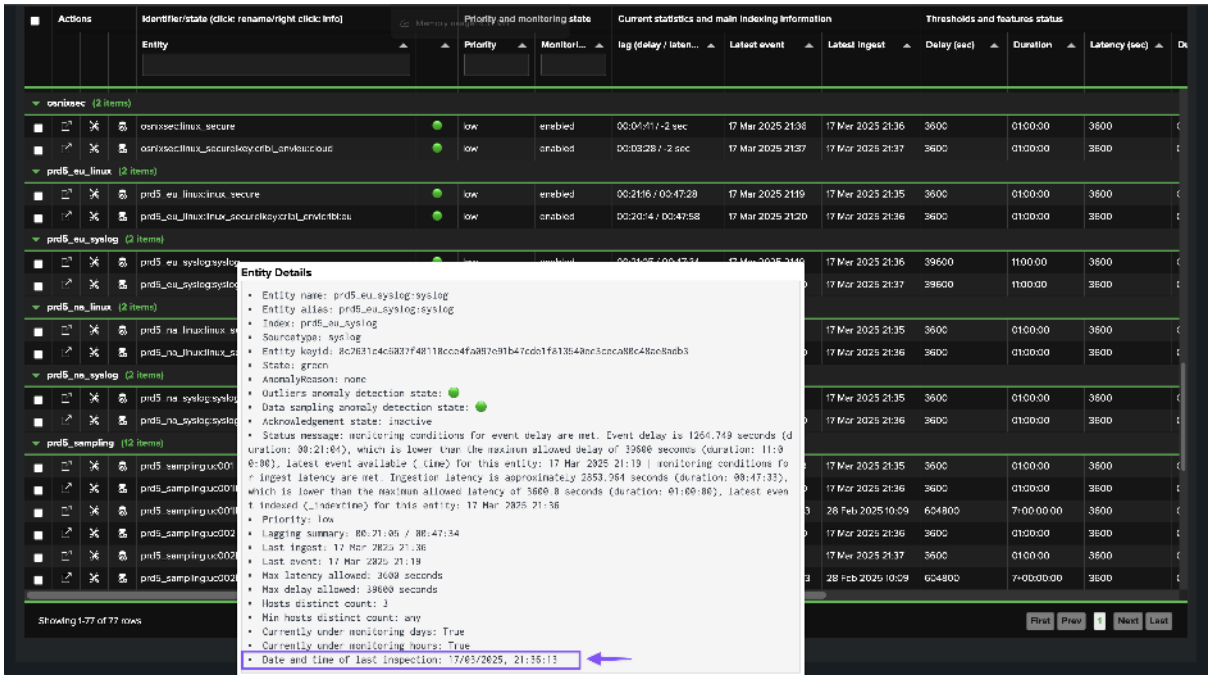
If the entity stops sending data to Splunk, or sends events with a latency that go beyond these limits, at some point the events will become out of the range of trackers, to maintain their state and TrackMe knowledge, the following happens:

- Every 5 minutes, the health tracker is executed and runs various maintenance and verification routines for the whole Virtual Tenant.
- For splk-dsm and splk-dhm components, the task `untracked_entities` is executed and detects entities that are no longer covered by any tracker, maintaining the delay metadata knowledge in TrackMe.
- In parallel, the delayed entities inspector regularly reviews and performs target searches to refresh the metadata knowledge of entities that are no longer covered by any tracker and covered by the `untracked_entities` task from the Health Tracker.

9.9.2 1. Date and time of last feed inspection

The date and time of the last inspection is stored as an epoch time format in the field called "tracker_runtime".

The TrackMe UI makes this information available in a human readable format in the contextual menu, to access to this menu, right click on the entity name within the Tabulator:



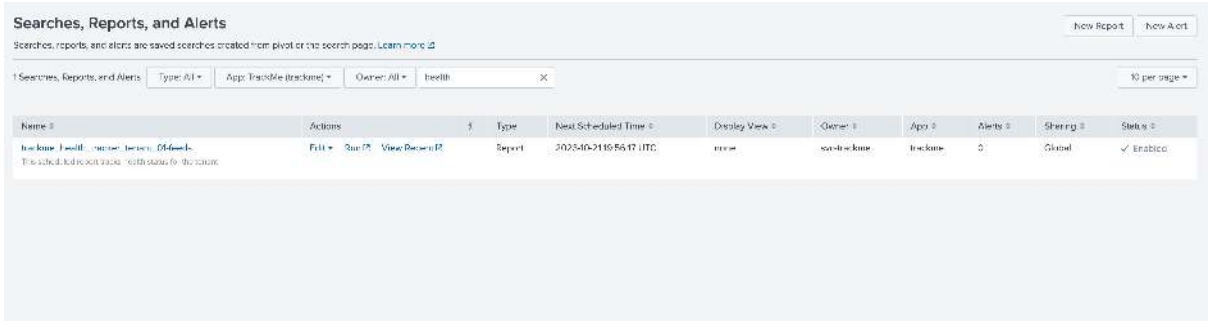
9.9.3 3. Health Tracker (untracked_entities)

Hint

Health Tracker and delayed entities

- The health tracker includes a task called `untracked_entities` which is responsible for detecting entities that are no longer covered by any tracker. (out of range)
- Entities can become out of range of trackers if these are suffering from high delay and/or latency, which makes them unreachable by the main trackers.
- The health tracker `untracked_entities` task ensures to maintain and refresh the calculated delay value, according to the current knowledge of the last event received for the entity.

The TrackMe health tracker is created automatically along with the Virtual Tenant, it performs various verifications and is responsible for various things like maintaining the schema version. (upgrade procedures for TrackMe)



It also ensures that inactive entities for the splk-dsm/splk-dhm components are updated regularly, logs for inactive entities updates can be found here:

```
index=_internal sourcetype=trackme:custom_commands:trackmetrackerhealth task=
↪"untracked_entities"
```

You can include the tenant_id and component if you want to focus on a specific Virtual Tenant and

component:

```
index=_internal sourcetype=trackme:custom_commands:trackmetrackerhealth tenant_id=
↪"mytenant" component="splk-dsm" task="untracked_entities"
```

A typical log activity if there are inactive entities will be similar to:

```
2023-09-24 08:41:22,647 INFO trackmetrackerhealth.py generate 556 tenant_id="01-feeds
↪", component="splk-dsm", task="untracked_entities", results="{'tenant_id': '01-feeds
↪', 'report_entities_count': '2', 'objects': ['webserver:apache:error',
↪'webserver:nginx:plus:error']}"
```

Using this workflow, TrackMe ensures that entities Metadata remain up to date even if they are not covered by any tracker for any reason, such as inactivity (feed interruption) or any other reason.

9.9.4 4. Delayed entities inspector

Hint

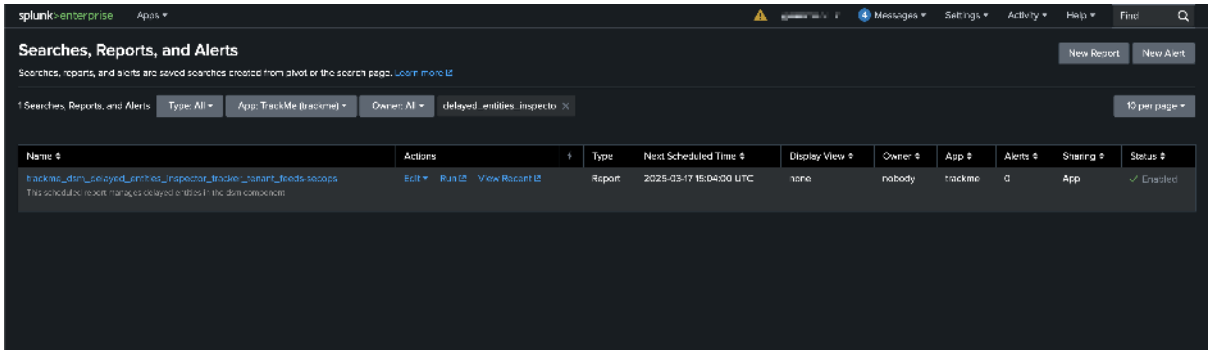
Delayed entities inspector

- The delayed entities inspector is a new process introduced in TrackMe 2.1.10.
- This process performs target searches to refresh the metadata knowledge of entities that are no longer covered by any tracker.
- This process is automatically created with the Virtual Tenant and component.

How does it work?

The delayed entities inspector is a scheduled job called:

- `trackme_<component>_delayed_entities_inspector_tracker_tenant_<tenant_id>`



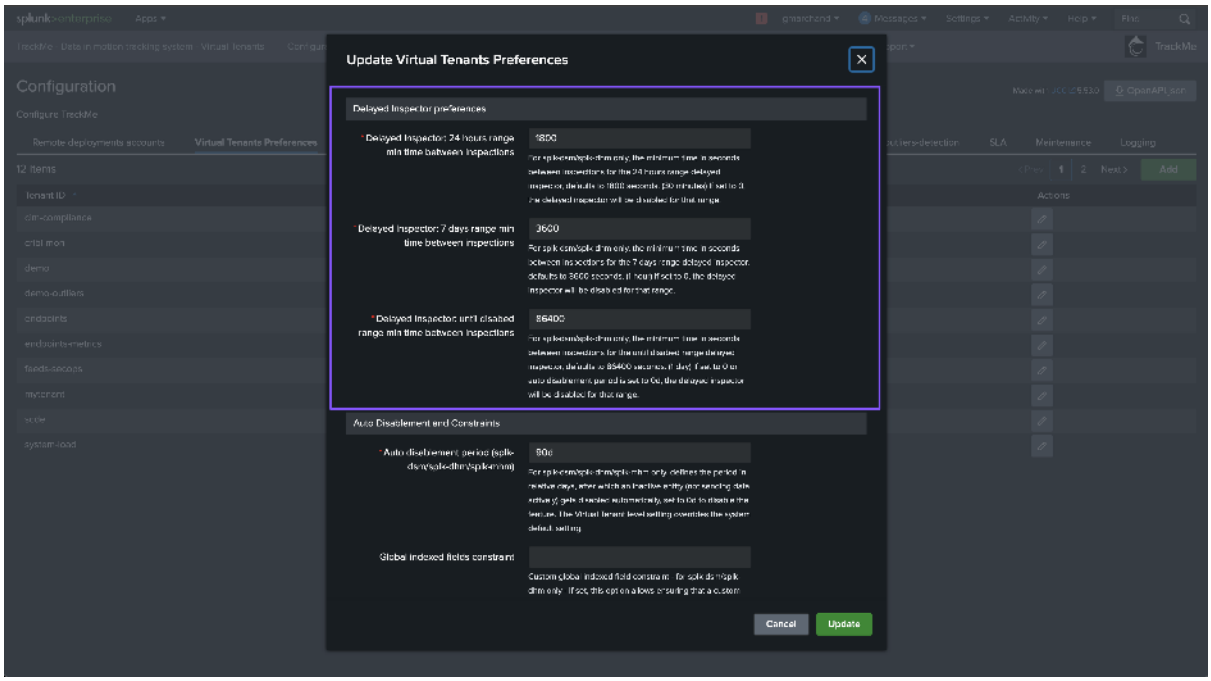
The job calls a custom command `trackmesplkfeedsdelayedinspector` which performs the following actions:

- Retrieves the list of entities that have been managed by the health tracker `untracked_entities` task, relying on the field `tracker_health_runtime`
- Classify entities based on these configurations and the following ranges:
- `24 hours range`: If the delay is less than 24 hours, the delayed entities inspector will attempt to refresh the knowledge based on the 24 hours range. (every 30 minutes by default)
- `7 days ranges`: If the delay is less than 7 days, the delayed entities inspector will attempt to refresh the knowledge based on the 7 days range. (every 1 hour by default)
- `Until disabled`: If the delay is greater than 7 days and until the auto disablement period is reached, the delayed entities inspector will attempt to refresh the knowledge of the entity based on the range setting. (once every 24 hours by default)

- Iterates over entities and performs a target search to refresh the metadata knowledge of the entity.

9.9.5 Virtual Tenant delayed inspector configuration

The delayed inspector configuration is available in the Virtual Tenant configuration page, in the `Delayed entities inspector` section.



9.9.6 Execution of the delayed entities inspector searches

The delayed entities inspector iterates over entities and performs a target search to refresh the metadata knowledge of the entity.

Its activity can be tracked through the main logs:

```
index=_internal sourcetype=trackme:custom_commands:trackmesplkfeedsdelayedinspector
```

Typically, the logs will show two steps per entity, which notably details the search executed, its runtime and results:

- `Executing entity delayed tracking resulting search=`
- `delayed_entity_search_results=`

You can track executed searches and related events for a given entity by using the following search:

```
index=_internal sourcetype=trackme:custom_commands:trackmesplkfeedsdelayedinspector␣
↪tenant_id="mytenant" component="splk-dsm" object="myentity"
```

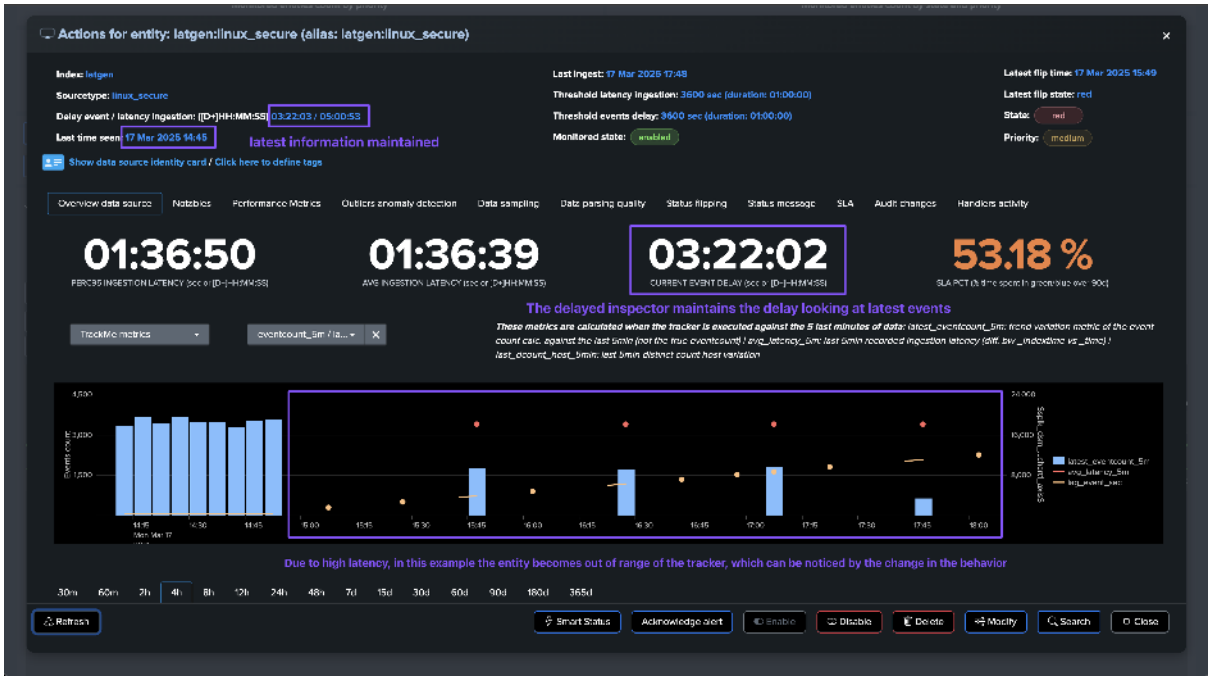
Hint

Handlers events introduced in TrackMe 2.1.11

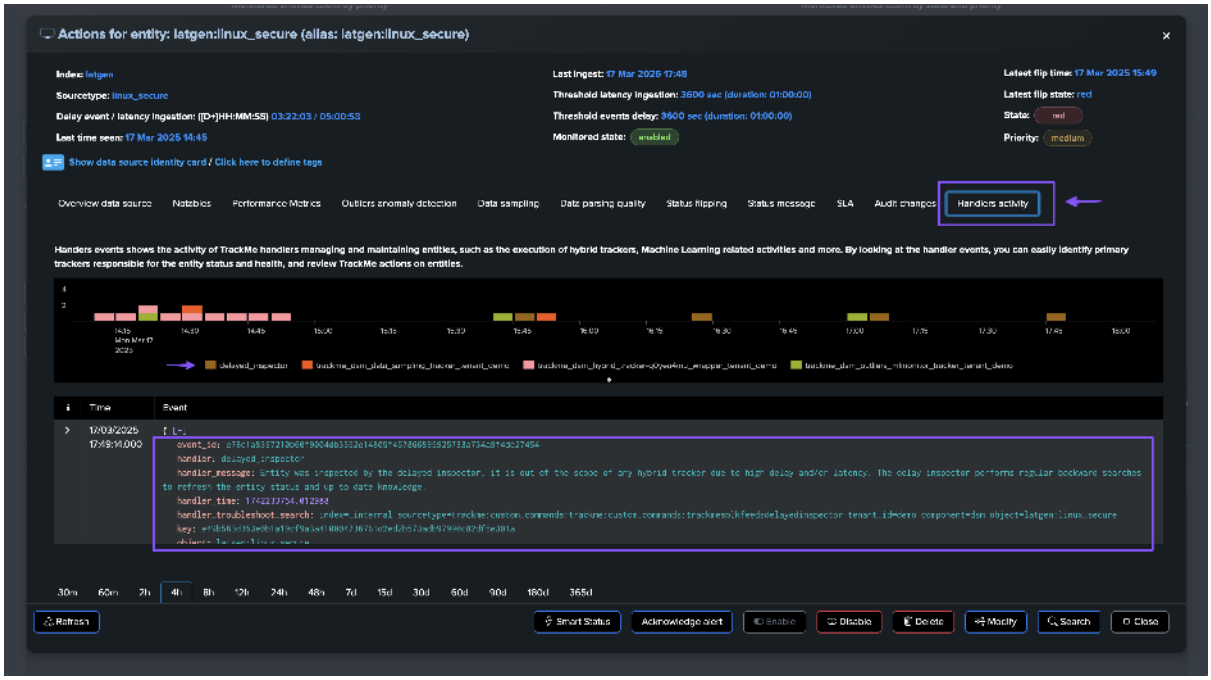
- Introduced in TrackMe 2.1.11, the Handler events allows easily tracking the execution of the various TrackMe logics, from hybrid trackers to any meaningful process and notably the delayed entities inspector.

- You can easily observe and review the activity of TrackMe such as the delayed entities inspector by looking at the `Handler events` tab or the `trackme:handlers` sourcetype in TrackMe's summary indexes.

An entity affected by high latency and/or delay and becoming out of range, therefore maintained by the delayed entities inspector will show like the following:



From TrackMe 2.1.11, the Handler events will also show the activity associated with the delayed entities inspector:



As soon as the entity issues are fixed, the feed has finished catching up or the underneath issue is resolved, the entity is again managed by the hybrid tracker:

9.10 Logical groups (entities ensemble association)

About Logical Groups

- The Logical Groups feature in TrackMe allows you to associate multiple entities into an ensemble, for the purposes of influencing the status of each entity depending on the status of the group itself
- For instance, you can associate two entities into a group with a 50% minimal green threshold; as long as one of the two remains compliant, the group will be considered healthy and the entities will be in a blue state
- The Logical Groups feature is available for the components of the Splunk feeds tracking family, as well as Flex Objects (splk-flx)

9.10.1 Introduction to logical groups

In TrackMe and for the Feeds tracking components, you can associate multiple entities into a logical group.

Logical groups are stored in a dedicated KVstore collection per tenant:

- `trackme_common_logical_group_tenant_<tenant_id>`

A logical group record is composed of:

- `object_group_name`: The name of the logical group
- `object_group_members`: The members of the group (multi-value field)
- `object_group_min_green_percent`: The minimal percentage of entities in a healthy state for the group to be considered healthy
- `object_group_mtime`: The epoch time recorded for the last modification of the group

A logical group influences the status of entities that are members of a group as follows:

- If an entity is not healthy (red, orange) but the logical group itself is healthy, the entity status will be blue instead (because the minimal green percentage of healthy entities in the group is respected)

- If an entity is not healthy and the group itself does not meet the minimal percentage of healthy entities in the group, the entity status will be in a non-healthy state (red, orange)

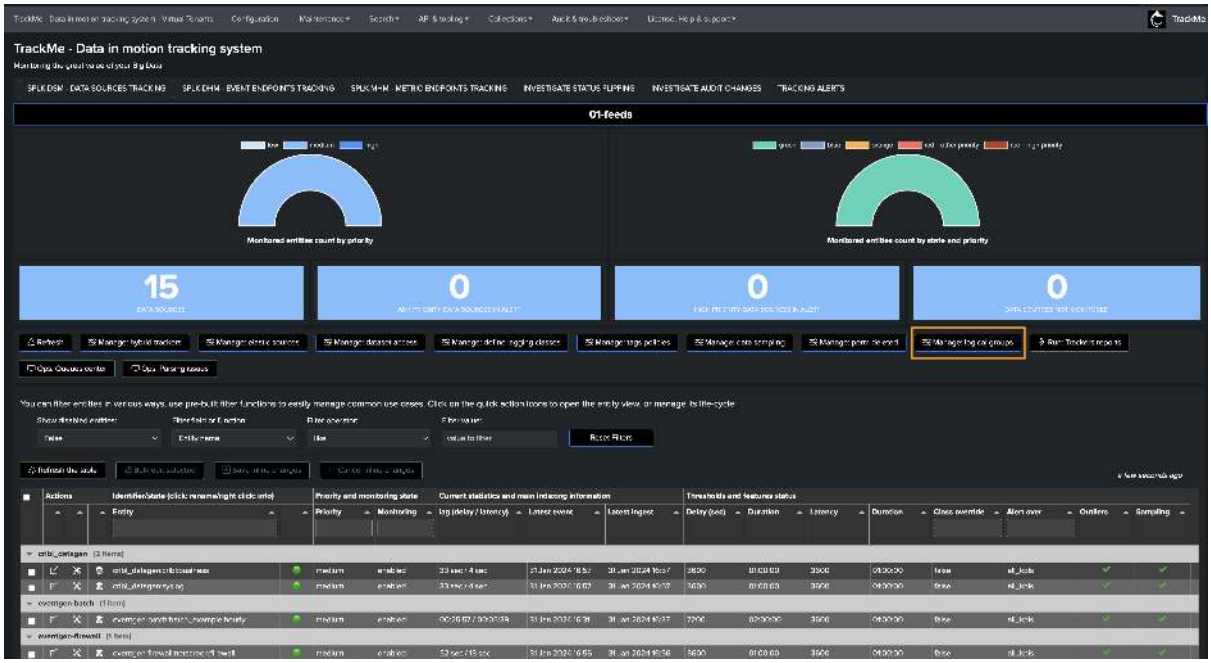
9.10.2 Use cases for logical groups

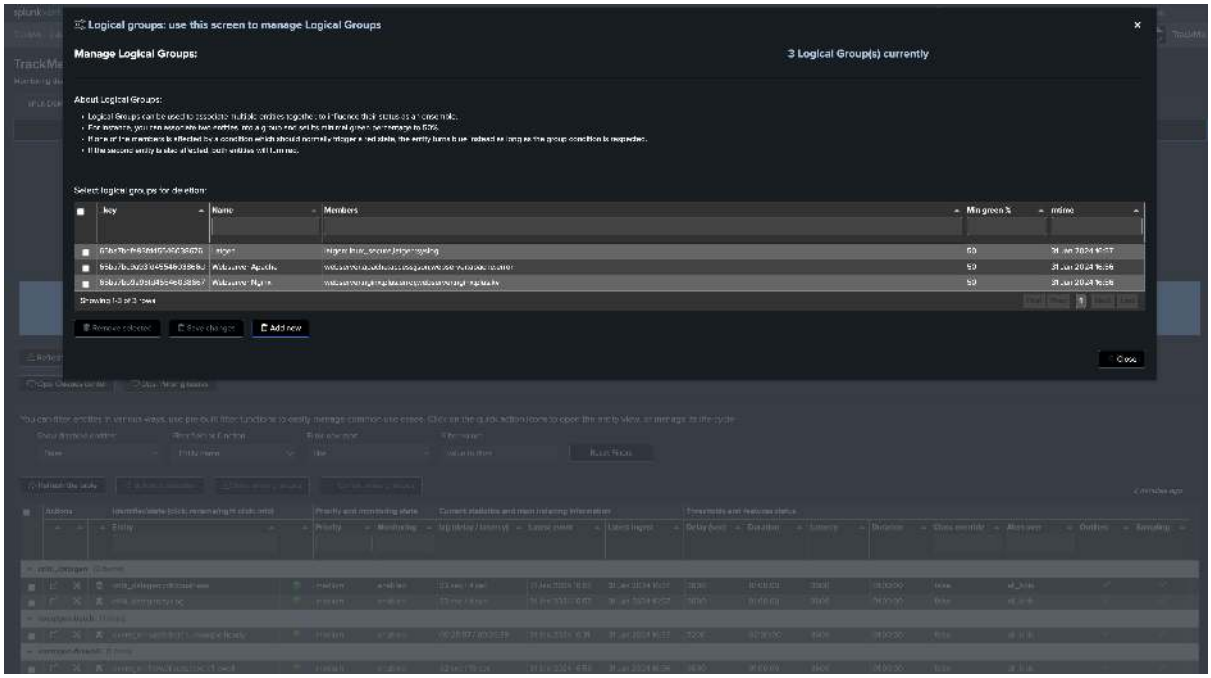
There are various use cases where logical groups can be beneficial:

- Multiple entities represent the same underlying data pipeline, however these entities need to be considered independently for different reasons
- Group multiple entities according to your needs, and avoid alerting if only a single entity is not healthy

9.10.3 Creating, updating and deleting logical groups via the central Logical Group screen

Since TrackMe 2.0.83, in addition to the per-entity screen and the bulk edit (see the next sections), you can also create, update and delete logical groups via the central Logical Group screen available from the TrackMe main menu:





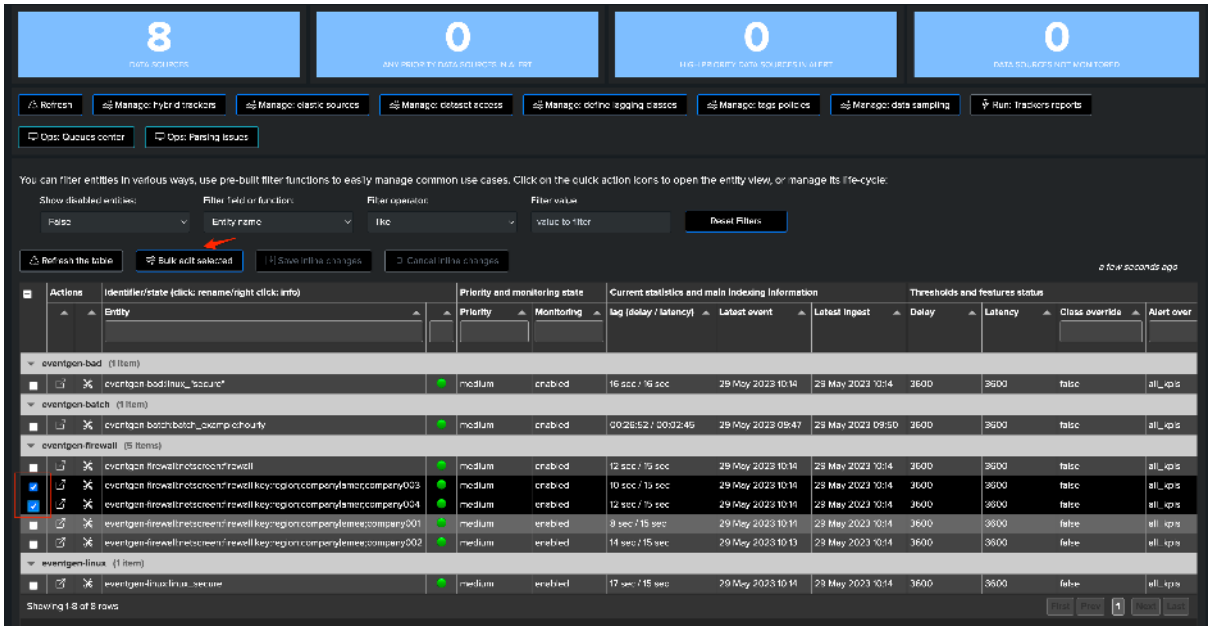
9.10.4 Creating logical groups and associating entities

Logical groups can be defined through TrackMe user interfaces, in SPL via the REST API endpoint and the `trackme` command, as well as via dedicated custom command utility `trackmeautogroup`.

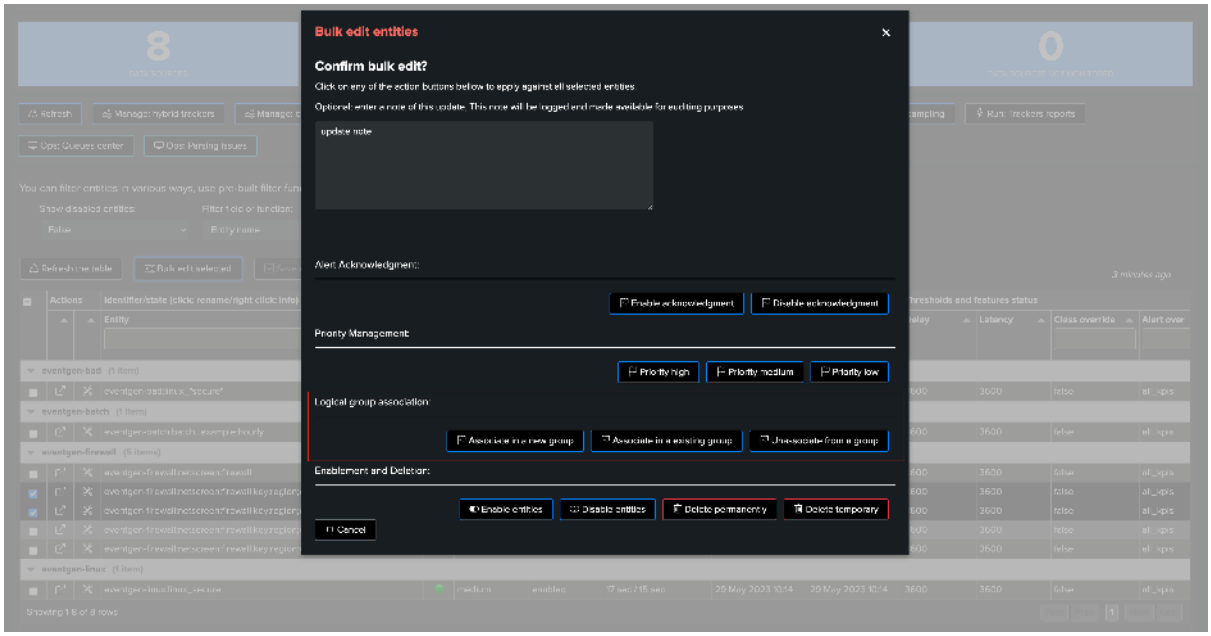
Logical groups via the UI and bulk edit

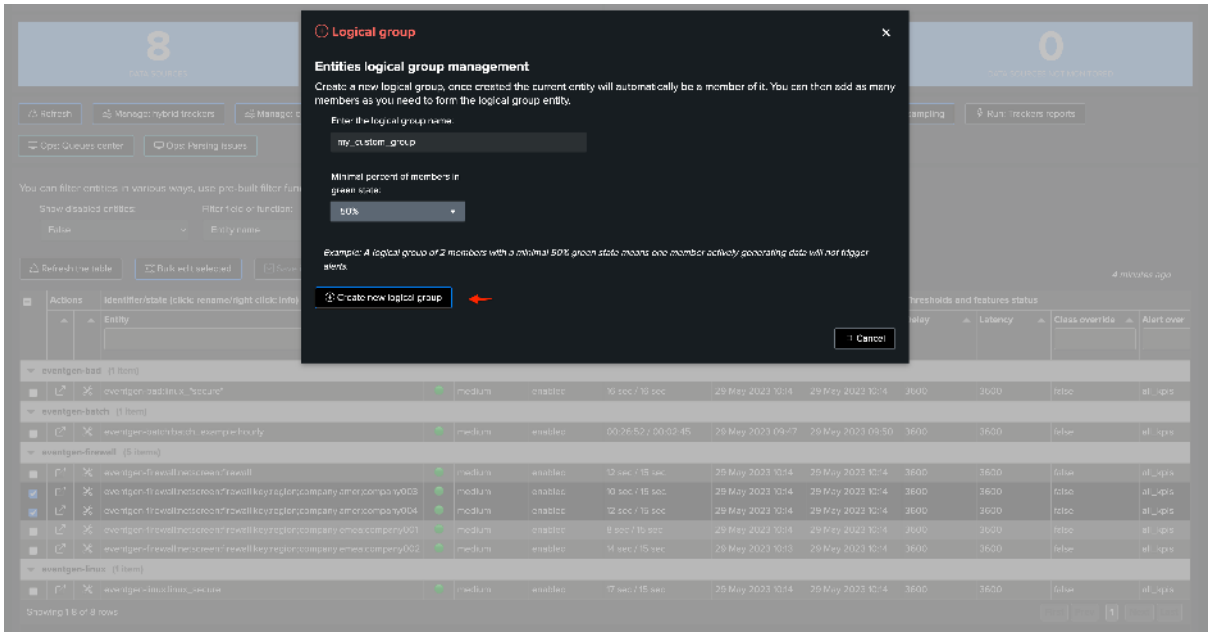
You can associate entities into a new or existing logical group from the bulk edition feature:

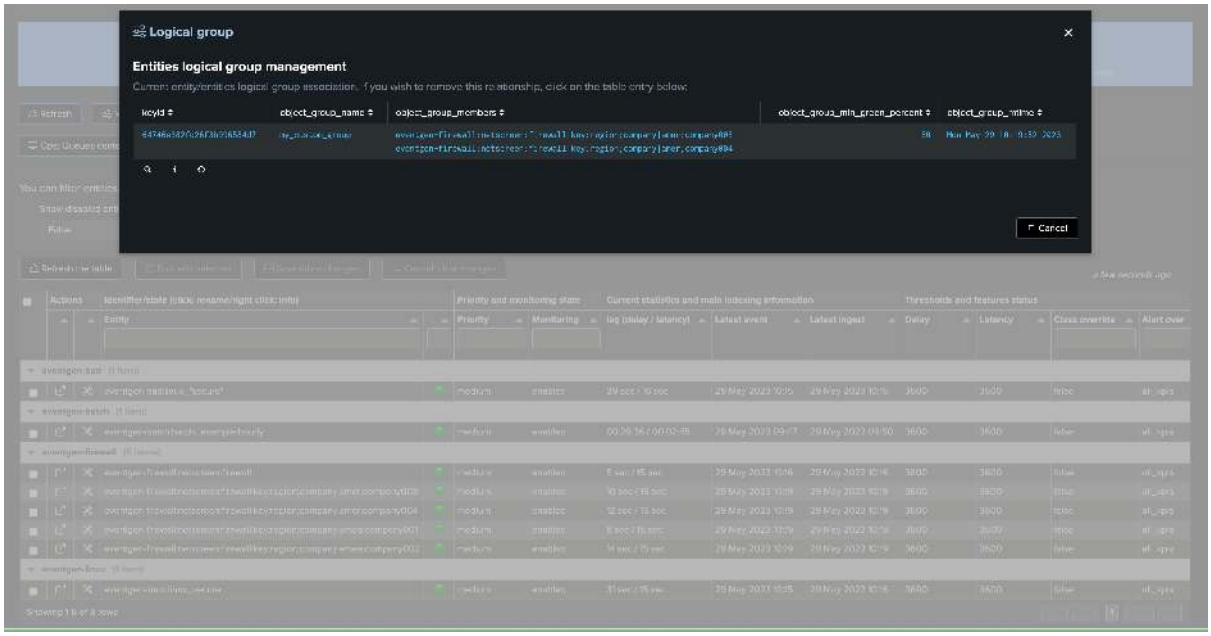
select one or more entities to be associated:



you can create a new group or choose an existing group:


Bulk edit entities
Confirm bulk edit?
Click on any of the action buttons below to apply against all selected entities
Optional: enter a note of this update. This note will be logged and made available for auditing purposes
update note
Alert Acknowledgment:
Enable acknowledgment
Disable acknowledgment
Priority Management:
Priority high
Priority medium
Priority low
Logical group association:
Associate in a new group
Associate in a existing group
Un-associate from a group
Enablement and Deletion:
Enable entities
Disable entities
Delete permanently
Delete temporary
Cancel


Logical group
Entities logical group management
Create a new logical group, once created the current entity will automatically be a member of it. You can then add as many members as you need to form the logical group entity.
Enter the logical group name:
my_custom_group
Minimal percent of members in green state:
50%
Example: A logical group of 2 members with a minimal 50% green state means one member actively generating data will not trigger alerts.
Create new logical group
Cancel

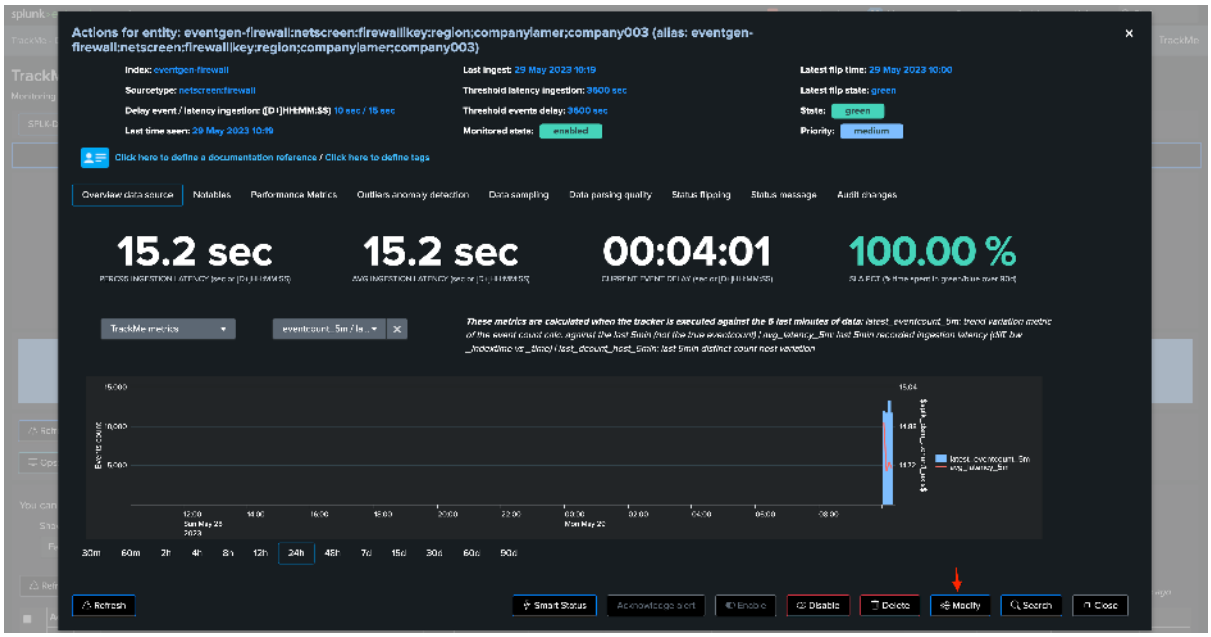


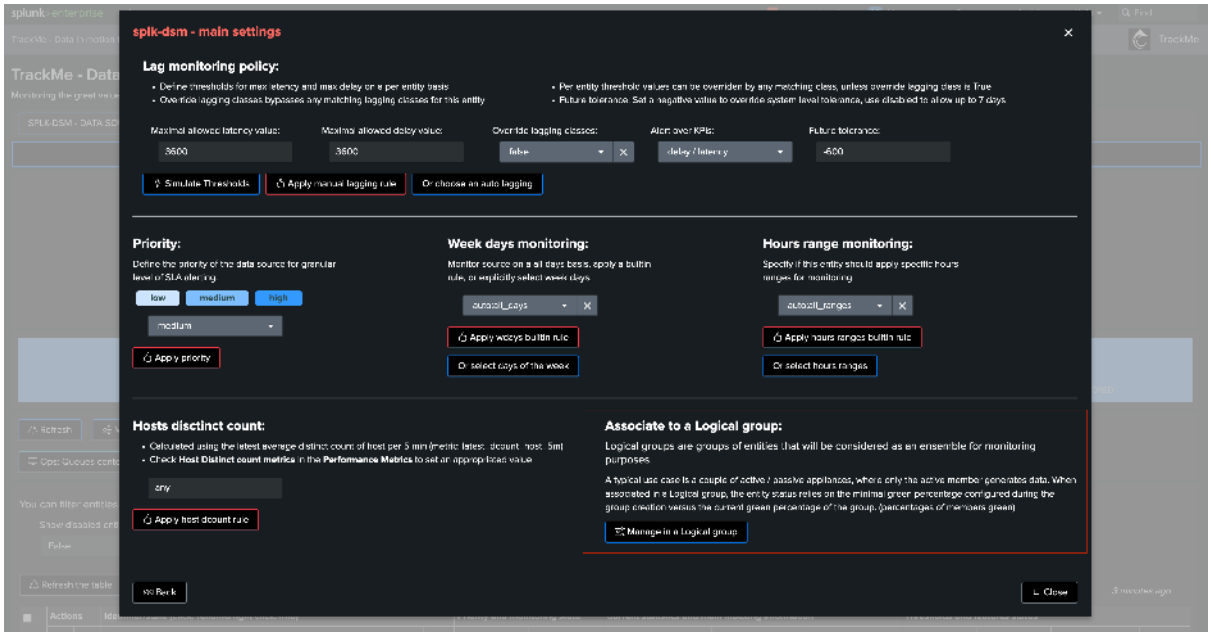
From the bulk edit screen you can:

- Associate selected entities into a new logical group
- Associate selected entities into an existing logical group
- Remove the association of these entities with their current logical group

Logical groups per entity via the modification screen

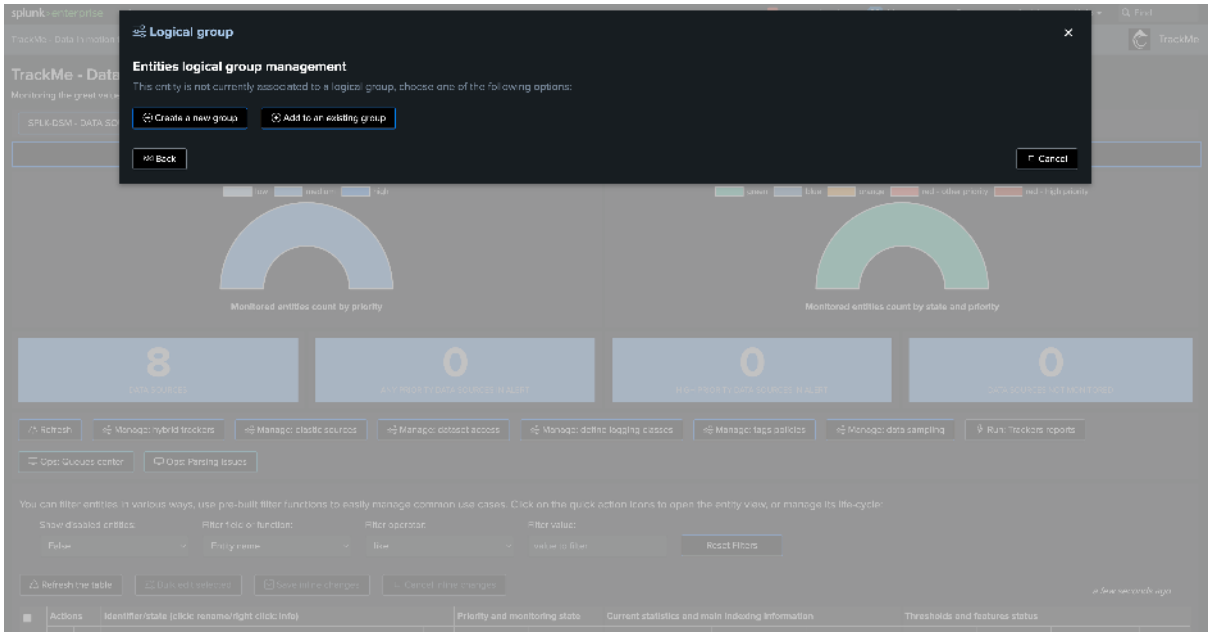
On a per-entity basis, click on the Modify button and access the logical group setting screen:





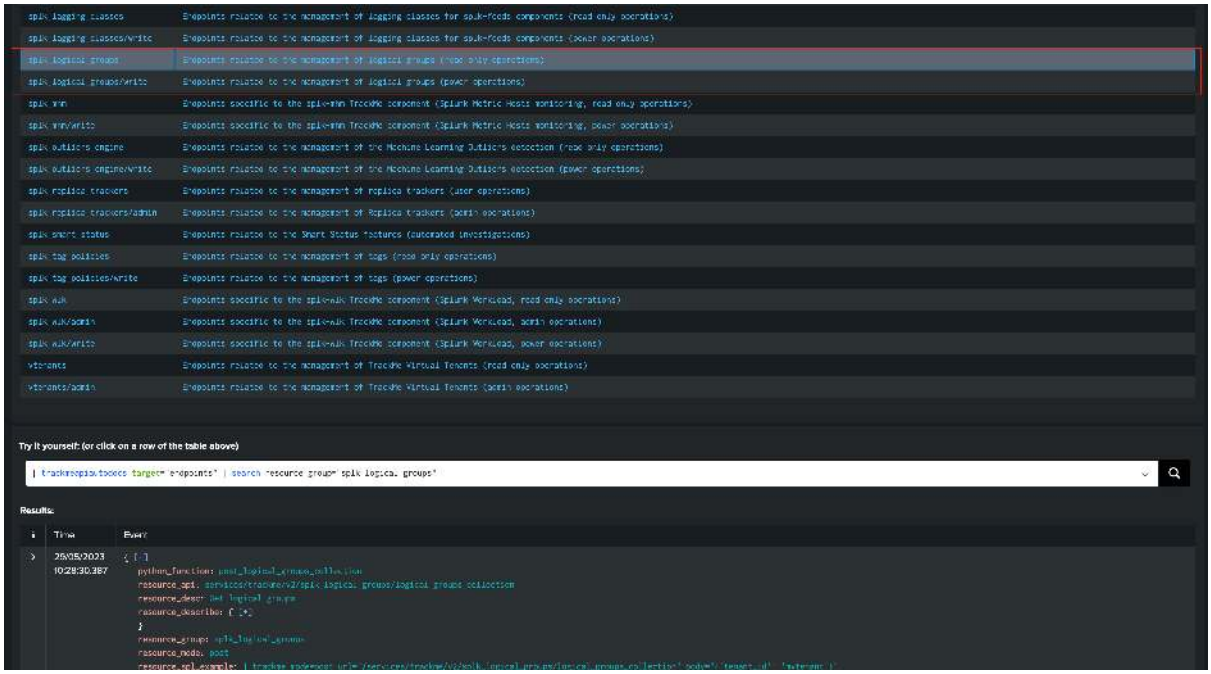
From this screen, you can:

- View the current logical group association, if any
- Associate this entity into a new logical group
- Associate this entity into an existing logical group
- Remove the association of this entity from its current logical group



Logical groups management via the REST API and the trackme command

Consult the REST API reference for the full list of available endpoints and their respective usage:



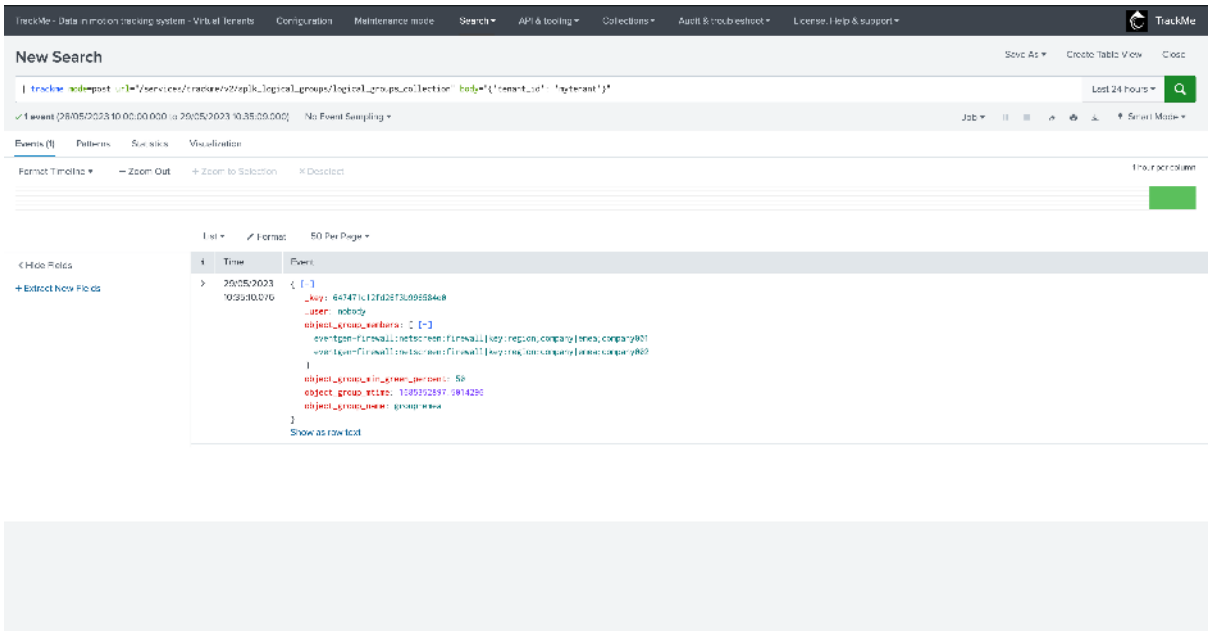
The API resource groups are:

- splk_logical_groups: for read-only operations
- splk_logical_groups/write: for write operations

For instance:

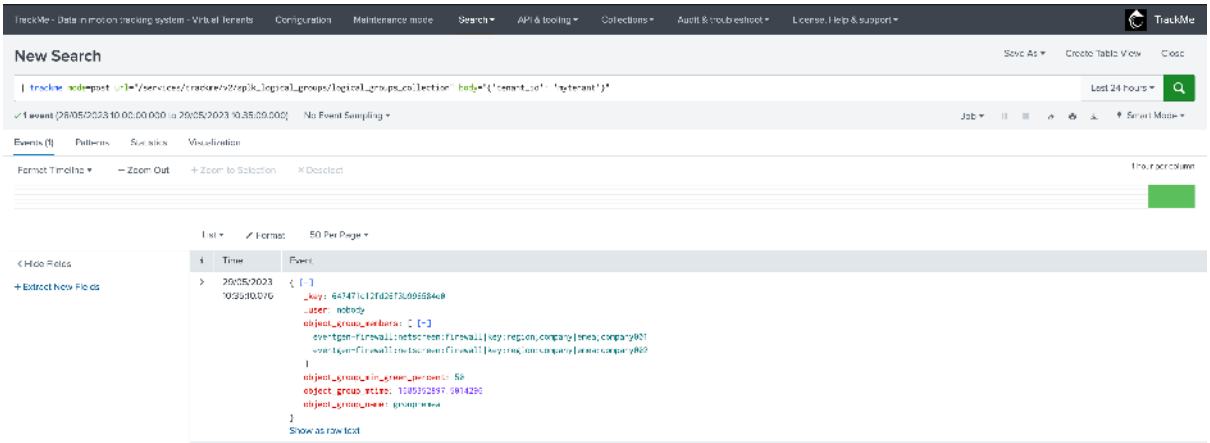
list all groups:

```
| trackme mode=post url="/services/trackme/v2/splk_logical_groups/logical_groups_
↪collection" body="{'tenant_id': 'mytenant'}"
```



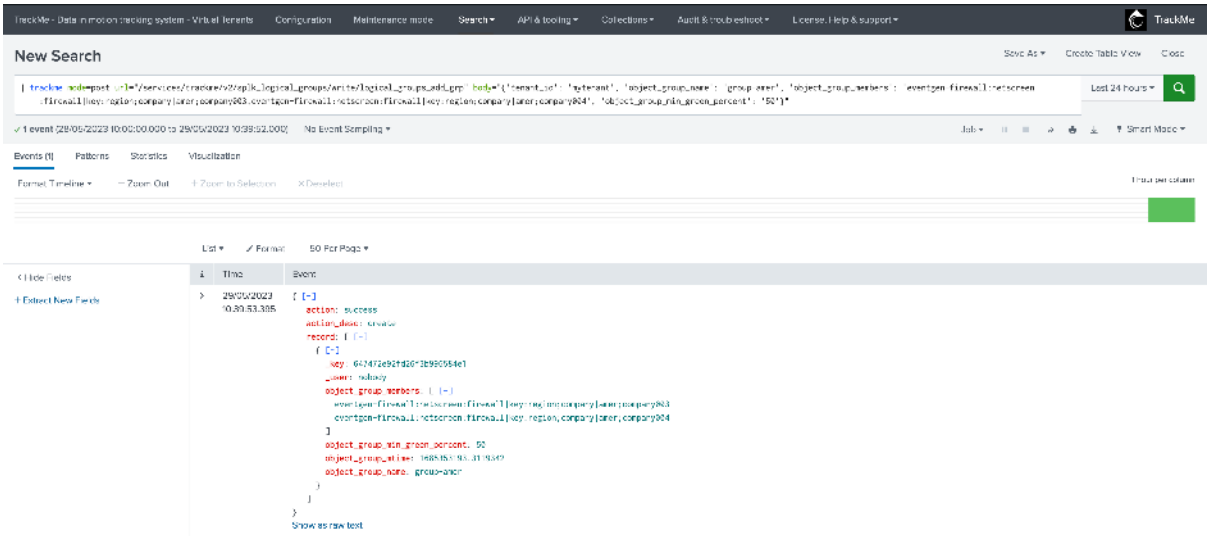
Get a specific group:

```
| trackme mode=post url="/services/trackme/v2/splk_logical_groups/logical_groups_get_
↪grp" body="{'tenant_id': 'mytenant', 'object_group_name': 'group-emea'}"
```



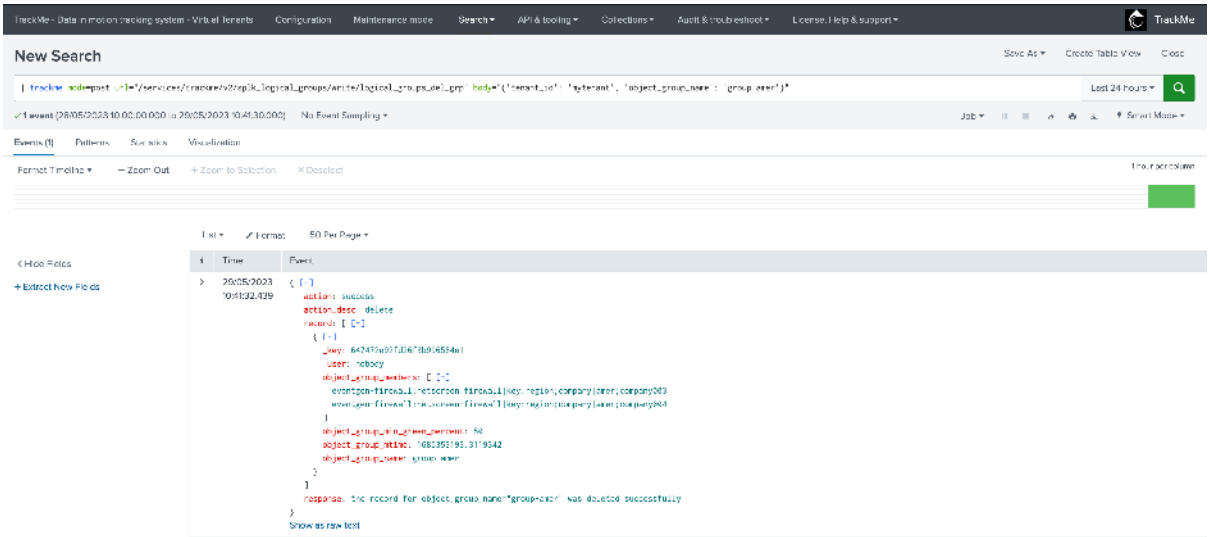
Create a new group and associate entities, update the group if it already exists:

```
| trackme mode=post url="/services/trackme/v2/splk_logical_groups/write/logical_
↪groups_add_grp" body="{'tenant_id': 'mytenant', 'object_group_name': 'group-amer',
↪'object_group_members': 'eventgen-firewall:netscreen:firewall|key:region;
↪company|amer;company003,eventgen-firewall:netscreen:firewall|key:region;
↪company|amer;company004', 'object_group_min_green_percent': '50'}"
```



Delete a group and clear all associations for its members:

```
| trackme mode=post url="/services/trackme/v2/splk_logical_groups/write/logical_
↪groups_del_grp" body="{'tenant_id': 'mytenant', 'object_group_name': 'group-amer'}"
```



Auto logical group management via the command trackmeautogroup

Hint

The command trackmeautogroup was introduced in release 2.0.36

The custom command `trackmeautogroup` is a streaming custom command utility which can be leveraged to automatically perform group creation, association and management according to your needs.

This is a streaming command which means that it will retrieve entities to be managed from an upstream SPL logic you will define, it behaves in the following way:

- The command requires two fields to be provided by your SPL logic:
  - `object_group_name`: the name of the logical group
  - `object_group_members`: the members of the logical group in a multi-value format
- Based on these upstream results, the command verifies if a group already exists and will create the group if necessary
- If the group already exists, but the members definition differs, it will update the group automatically and define the minimal percentage expected for the group (ex: 50% if two entities, 33.33% if 3 entities, etc)
- It will remove the group automatically if there is only 1 active member left, this can be controlled by the argument `purge_single_member_grp=True|False`

Example of implementation

Based on our prior example, we have a list of common criteria which define what a group definition should be:

- same index and sourcetype
- same “region”

In our example, any entity from the same "company" should be part of a group which belongs to these criteria (index / sourcetype / region), we can leverage this simple SPL logic with a regular expression to extract the relevant association:

```
| inputlookup append=t trackme_dsm_tenant_mytenant where monitored_state="enabled"
| fields object

``` Extract from the object fields which will be used for the logical group␣
↪definition ```
| rex field=object "^(?<index>[^\:]*)\:(?<sourcetype>[^\|]*)\|key\:[^\|]*\|(?<region>
↪[^\;]*)\;(?<company>.*)"
```

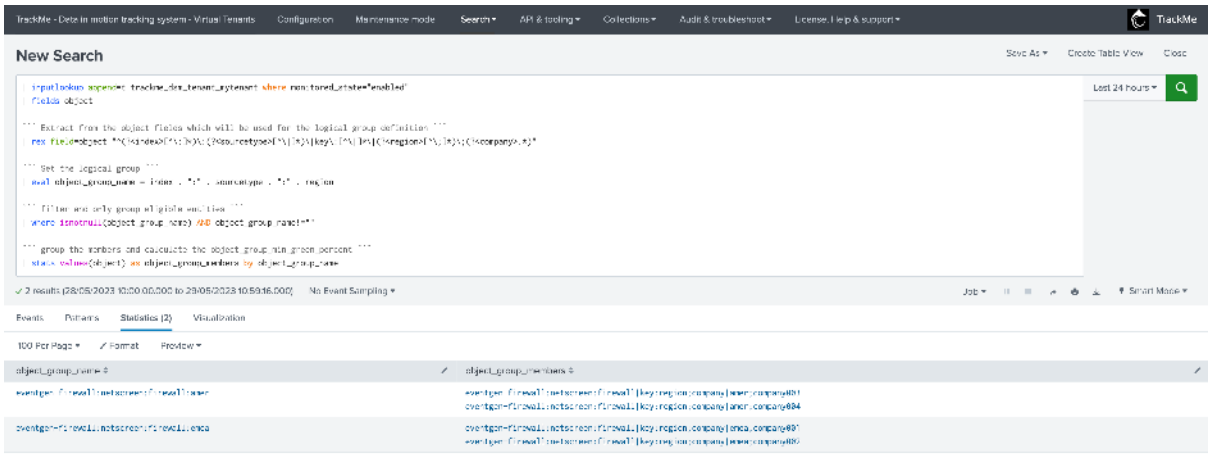
Note:

- Note that we filter on entities for which the monitored_state is enabled, as there wouldn't be any interest in maintaining groups for entities that are not monitored

Next, we want to define the logical group name, form the association and filter on entities which match our criteria:

```
| inputlookup append=t trackme_dsm_tenant_mytenant where monitored_state="enabled"
| fields object

``` Extract from the object fields which will be used for the logical group␣
↪definition ```
| rex field=object "^(?<index>[^\:]*)\:(?<sourcetype>[^\|]*)\|key\:[^\|]*\|(?<region>
↪[^\;]*)\;(?<company>.*)"

``` Set the logical group ```
| eval object_group_name = index . ":" . sourcetype . ":" . region

``` filter and only group eligible entities ```
| where isnotnull(object_group_name) AND object_group_name!=""

``` group the members and calculate the object_group_min_green_percent ```
| stats values(object) as object_group_members by object_group_name
```



This is all that we need! We can now call the custom command which takes of everything else for us:
```

```
| inputlookup append=t trackme_dsm_tenant_mytenant where monitored_state="enabled"
| fields object

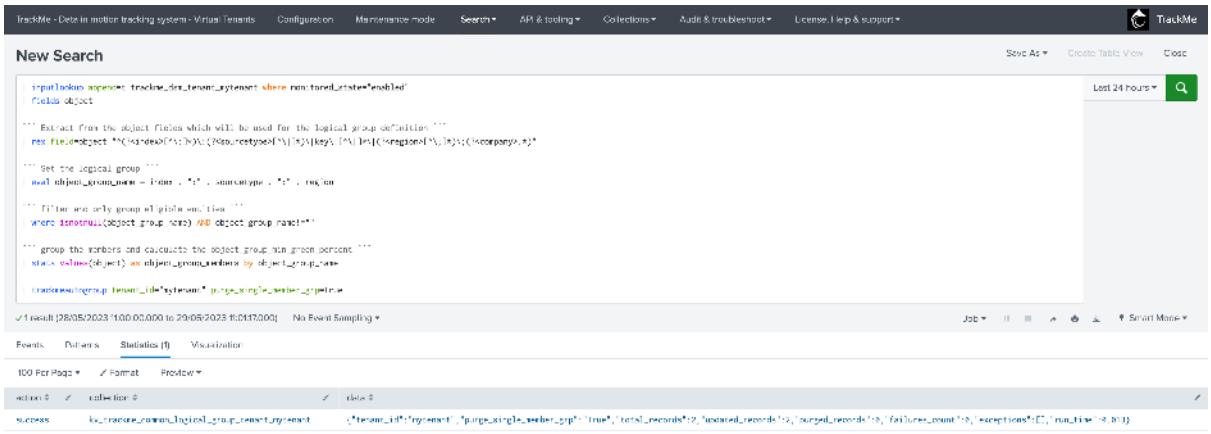
``` Extract from the object fields which will be used for the logical group␣
↪definition ```
| rex field=object "^(?<index>[^\:]*)\:(?<sourcetype>[^\|]*)\|key\:[^\|]*\|(?<region>
↪[^\;]*)\;(?<company>.*)"

``` Set the logical group ```
| eval object_group_name = index . ":" . sourcetype . ":" . region

``` filter and only group eligible entities ```
| where isnotnull(object_group_name) AND object_group_name!=""

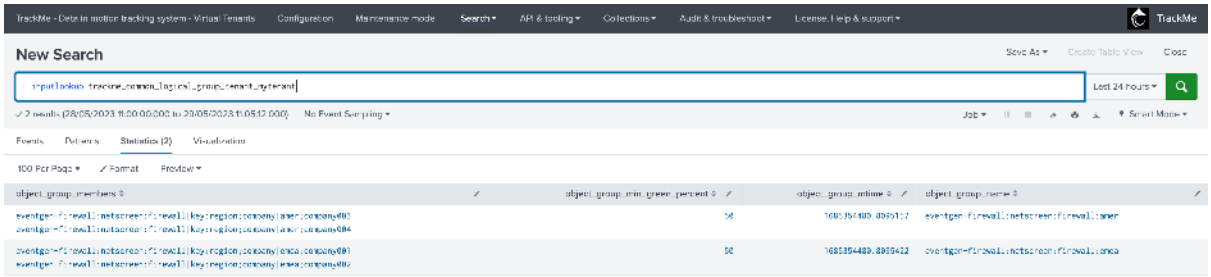
``` group the members and calculate the object_group_min_green_percent ```
| stats values(object) as object_group_members by object_group_name

| trackmeautogroup tenant_id="mytenant" purge_single_member_grp=True
```



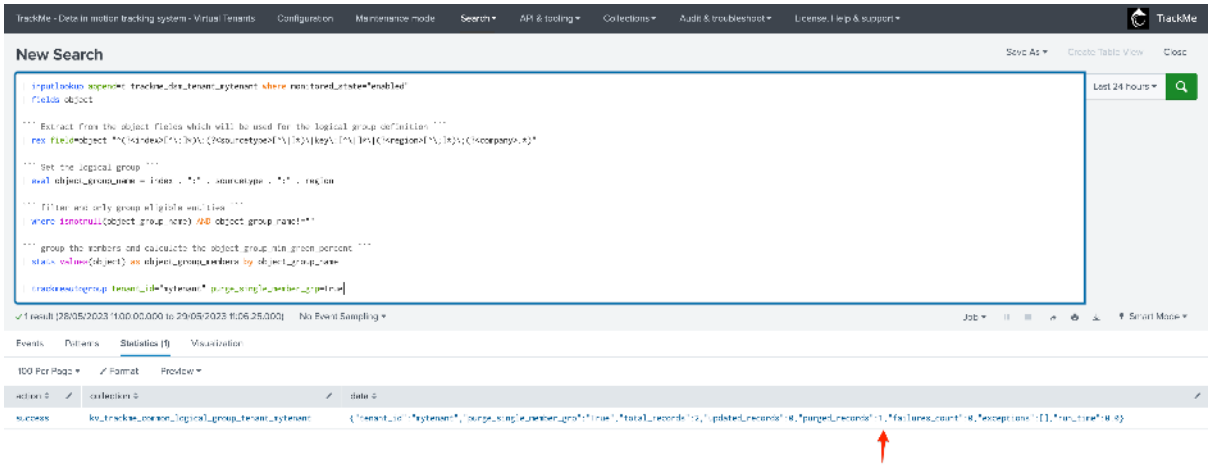
See the collection:

```
| inputlookup trackme_common_logical_group_tenant_mytenant
```



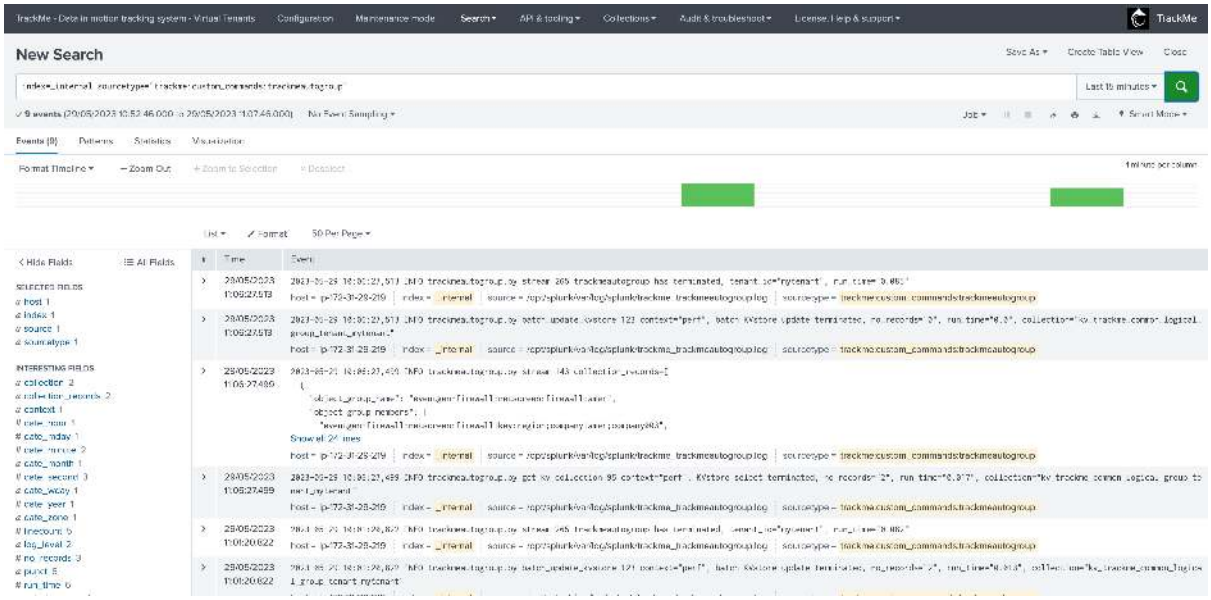
Next executions of this logic with take care of adding any new group of it has more than 1 entity, and remove any group that has only 1 active entity left, logging the output of performed actions

*Example: we remove 1 of these 4 entities:**



The custom command logs into:

```
index=_internal sourcetype="trackme:custom_commands:trackmeautogroup"
```



All that is left now is to save logic into a report, and shedule its execution according to your preference.

Note that as of now, this scheduled logic will not be orchestrated by TrackMe, therefore if you later on remove this tenant, you need to manually remove this report too.

9.11 Splunk Workload (splk-wlk)

Note

TrackMe Workload component

- This feature is part of TrackMe's restricted features and allows tracking the key value activity and KPIs of your Splunk scheduling, from alerts to scheduled reports and **Enterprise Security correlation searches**.
- We track and alert on sophisticated concepts such as the detection of **execution anomalies** or **delayed execution**, **consumption behavior** changes using **Machine Learning** as well as **version control** and changes in the searches themselves.
- The **TrackMe Workload component** is extremely powerful and provides the missing deep visibility on your critical Splunk workflow, whether you are using Splunk for security or IT operations, or any other use case.

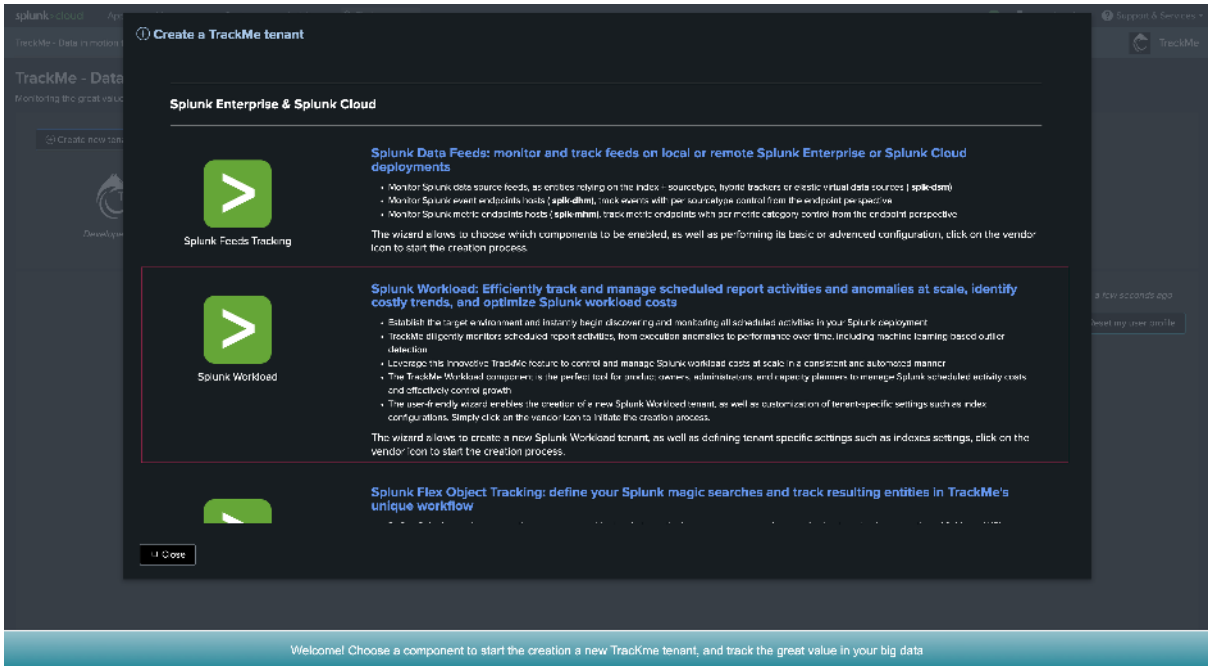
9.11.1 1. Introduction

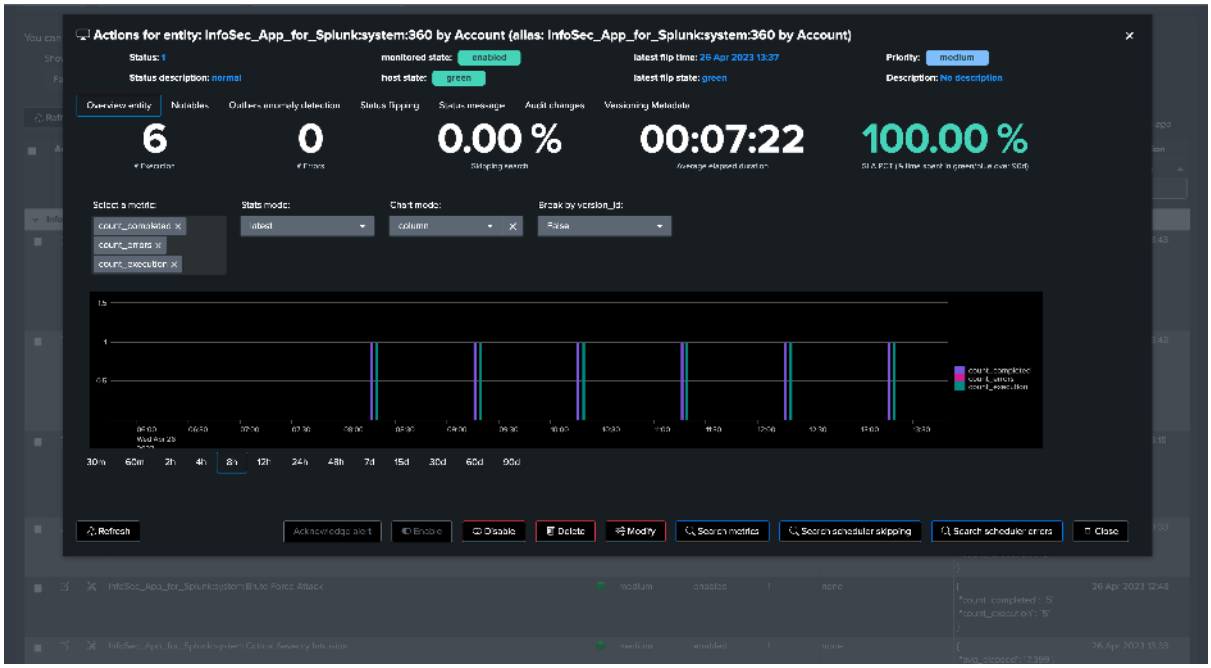
The objective of the Splunk Workload component for TrackMe is to continuously track the scheduling activity of a Splunk deployment, and perform key activity and detection:

- Discover Splunk scheduled activity across the environment; an entity will be the association of the Splunk application namespace, the owner, and the search name
- Track the scheduled activity behavior, and detect issues that can affect the well-being of Splunk scheduled reports and alerts
- Track and identify the scheduled objects' Metadata, such as their definition, to investigate and detect changes impacting searches' behaviors over time
- Generate consistent and key value metrics from Splunk introspection, scheduler and SVC usage for Splunk Cloud customers

- Use Machine Learning Outliers detection to track for abnormal behaviors and changes, such as an abnormal increase in costs, run time or consumption
- Alert when scheduled reports and alerts are affected by issues, and get benefit from the full set of TrackMe's features and workflow to handle the lifecycle of Splunk scheduled activity

The TrackMe Workload component, associated with TrackMe's unique workflow, is a game-changing companion for Splunk, allowing it to quickly detect and alert on Splunk's most critical activity, as well as providing the keys to understand scheduled related costs like never before.





Note

Grouping in the Workload component: group and overgroup

- By default, the Workload component groups entities by app which represents the Splunk appli-

cation namespace hosting the scheduled report or alert.

- When creating the tenant and also later on when creating trackers manually, since TrackMe 2.0.70, you can optionnally override this behaviour by setting up an `overgroup` value.
- By doing so, the Workload will group entities based on a custom term instead of applications, this can be useful if for instance you want to have multiple Search Head tiers in the same tenant.
- Refer to the *Creating a Workload tenant to host multiple Search Head tiers with overgroup* section for more information.

9.11.2 2. Workload entities

Once a Workload TrackMe Virtual Tenant has been configured, TrackMe starts to track for any scheduled activity and will create and maintain associated entities:

An entity name is composed of:

- `app` + ":" + `owner` + ":" + savedsearch name

Should any of this information change, TrackMe will consider this as a new entity to be created and maintained, for instead of a search reassignment, if the search is moved between different application name spaces, or if the search knowledge object identifier is changed.

9.11.3 3. Anomaly reason

3.1 Anomaly reason definition

TrackMe considers each entity individually, and will trigger a status change based on the following criteria:

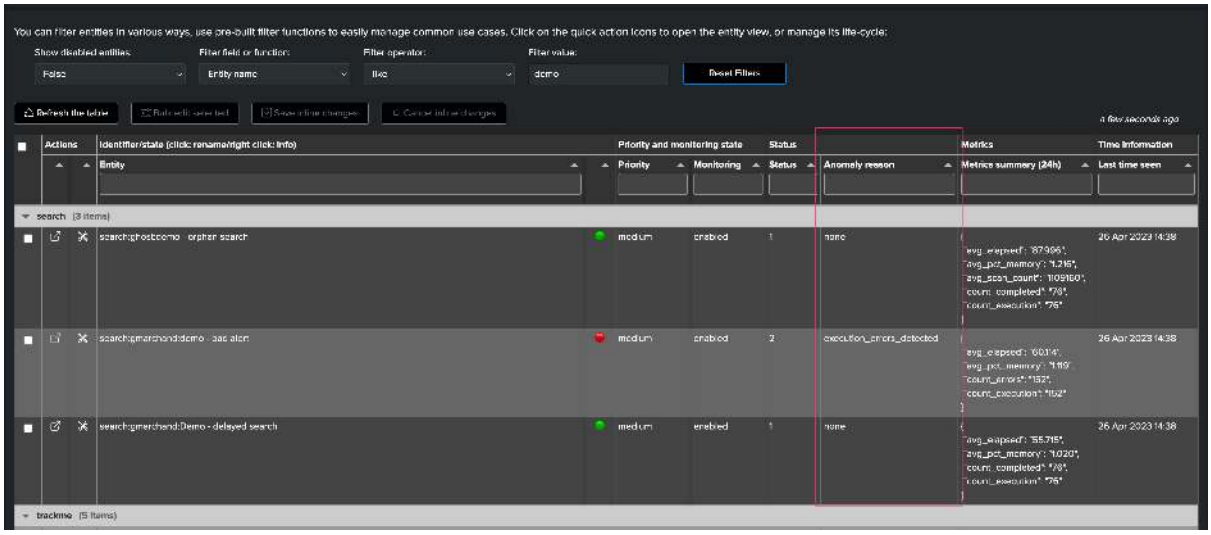
Anomaly Reason	Conditions
none	The entity is green and is healthy, there are no issues detected currently
skipping_searches_	TrackMe detected skipping executions for that scheduled, by default orange<5% and red>5%
orphan_search_d	A previously active scheduled search is now orphaned, which means the owner is not valid any longer and the search cannot be executed
execution_errors_de	Execution errors are detected and the scheduled report/alert is not working properly
anomaly_outlie	The Machine Learning outliers engine detected issues in one or more active ML models, this can be for instance an abnormal increase in the schedule runtime
execution_delayed	TrackMe uses the Metadata and the cron schedule translation to monitor if the search is delayed compared to its expected schedule (with 5 min of grace time by default)
status_not_met	The status of the entity is red, for unclassified reason
out_of_monito	The status of the entity is not healthy, however the current period is out of the monitoring window set for this entity

3.2 Checking the Anomaly Reason

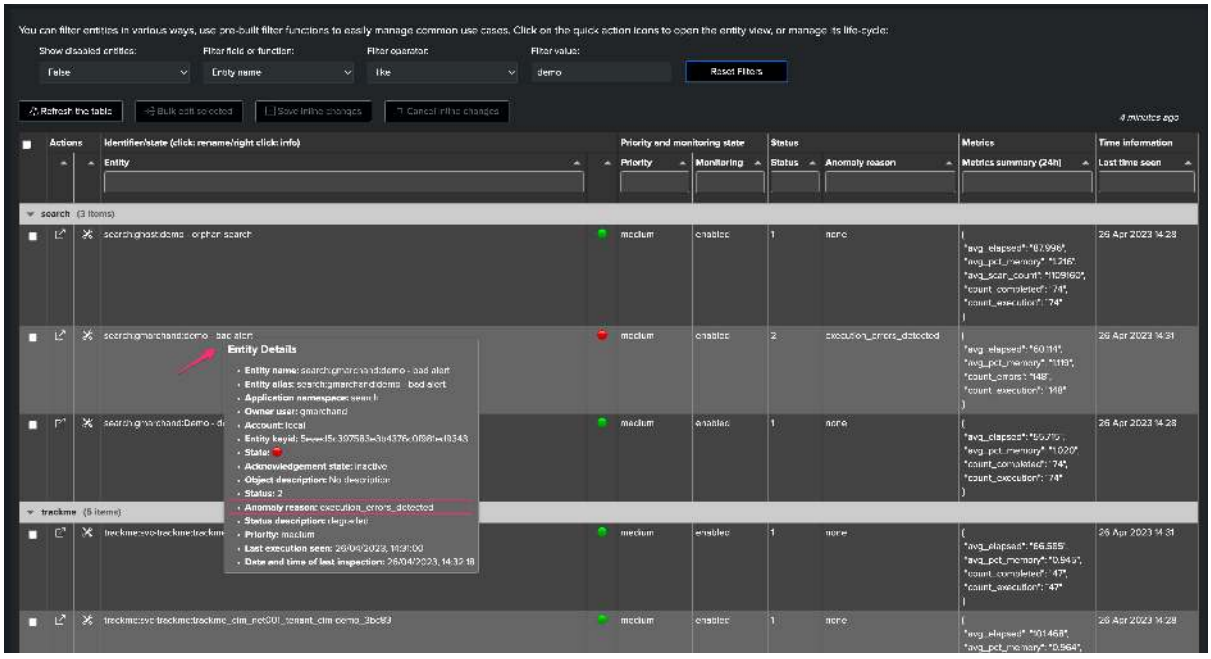
You can observe the current anomaly reason in different locations:

- In the table column called "Anomaly Reason"
- By right clicking on the entity, in the contextual menu
- When the alert is sent over, as part of the fields resulting from the alert
- When the alert fires, as part of the TrackMe notable event

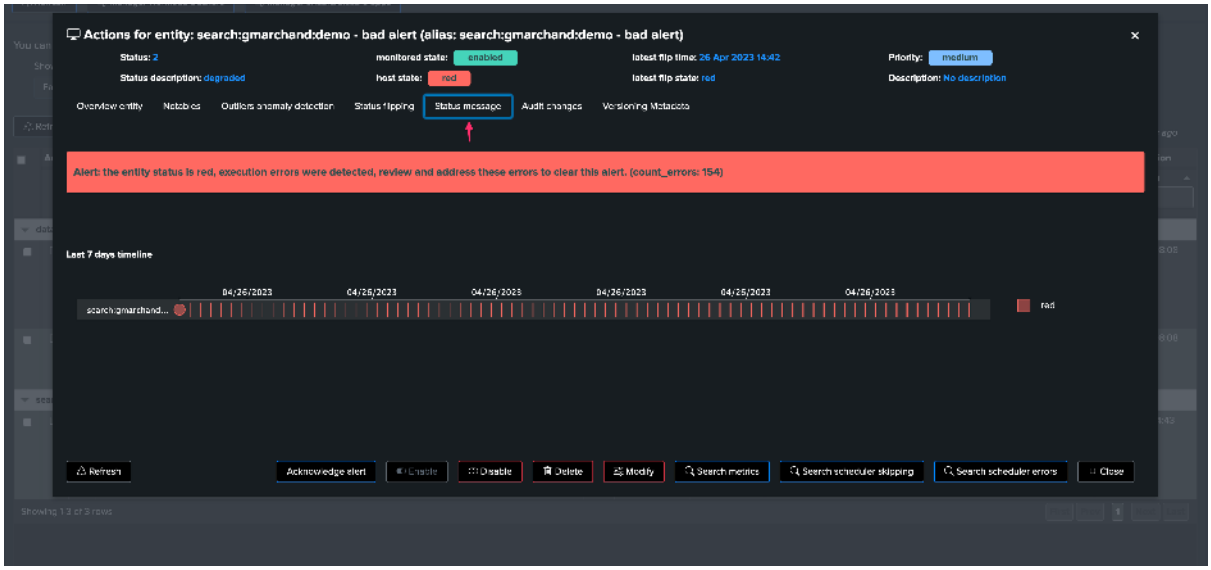
table anomaly reason column:



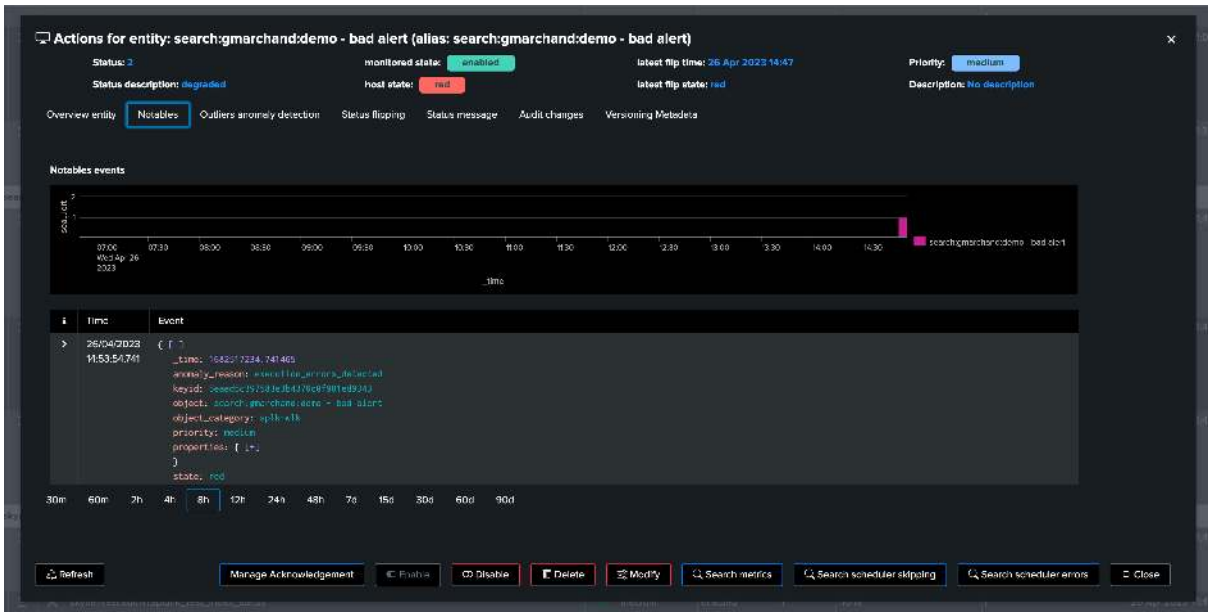
right click contextual menu on the entity name:



The status message is conditioned by the anomaly reason value and translated into a detailed message:

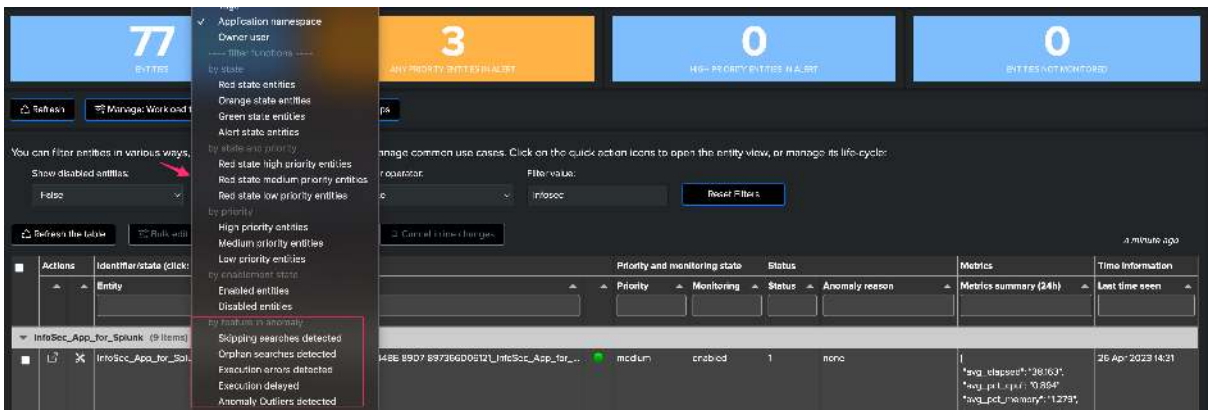


The anomaly reason is part of a notable event when an alert triggered:



3.3 Filtering on Anomaly Reasons

You can use the filter functions to check all entities with a given anomaly reason:



9.11.4 4. Workload metrics

Once a scheduled entity was discovered, TrackMe tracks its activity and generates various key metrics:

Metric name	Description
count_executi	Scheduler metric: the number of requested executions for this schedule
count_comple	Scheduler metric: the number of successfully completed executions for this schedule
count_skippe	Scheduler metric: the number of skipped executions for this schedule
count_errors	Scheduler metric: the number of executions errors detected for this schedule
elapsed	Introspection metric: the run time of the search, in seconds
pct_cpu	Introspection metric: the aggregated percentage of CPU used for this schedule (which can be more than 100%)
pct_memory	Introspection metric: the aggregated percentage of Memory used for this schedule (which can be more than 100%)
scan_count	Introspection metric: the number of events scanned for this schedule
svc_usage	Splunk Cloud metric: the SVC usage for this consumer

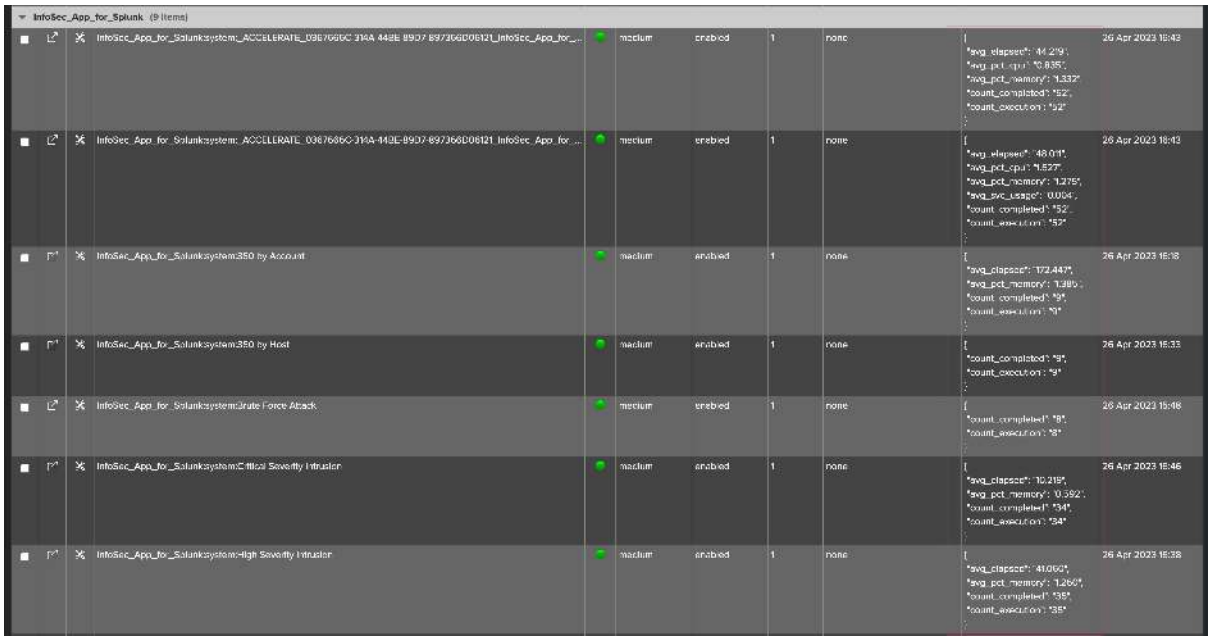
The Workload metrics are then used to condition the status of the entity, and feed the Machine Learning Outliers engine.

4.1 Accessing metrics

4.1.1 Metrics summary table

Metrics summary in the table:

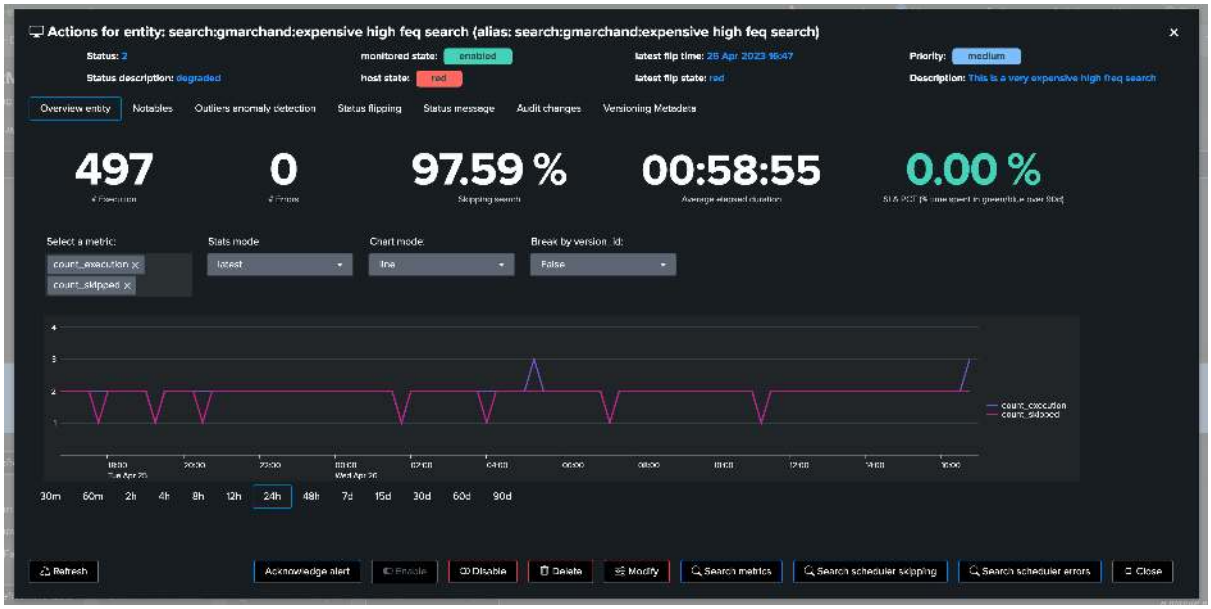
TrackMe shows a summary of the metrics of the last known 24 hours per entity:



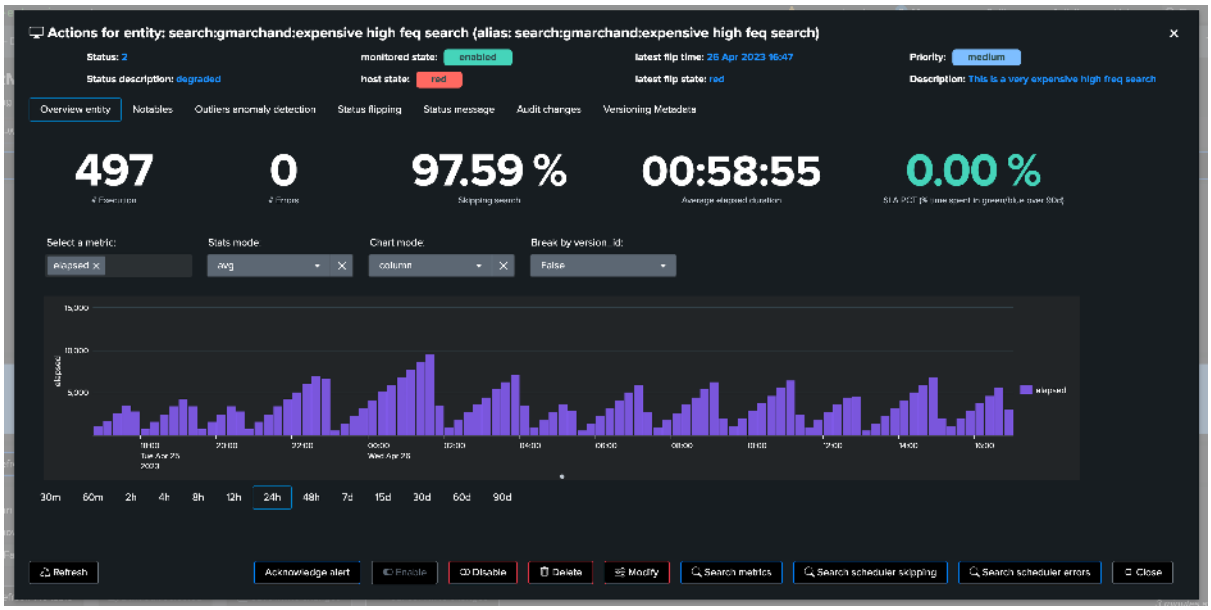
Note that metrics shown in the summary JSON vary depending on the metrics available, for instance an entity which has not been experiencing skipping searches will not show the metric as it is null.

4.2.2 Metrics chart over time

You access to the metric chart selector by opening an entity overview:



Use the metric selector, and optionally other settings according to your needs, in the following examples, we are looking at the average elapsed for that entity:



9.11.5 5. Metadata Versioning

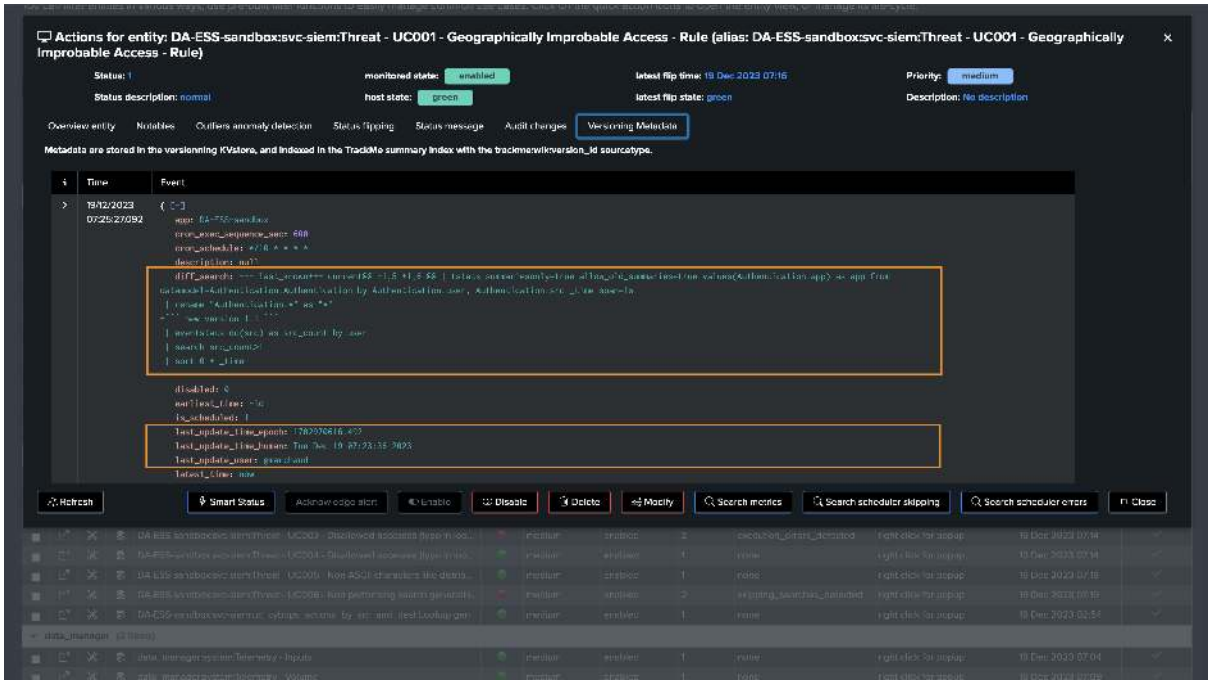
TrackMe monitors the versioning of scheduled entities, to detect changes and allow linking a knowledge object state with its performance metrics.

Hint

Diff search/earliest/latest (new in TrackMe 2.0.72)

- Since TrackMe 2.0.72, the diff of the search and/or earliest/latest quantifiers is performed and stored as part of the versioning events and records
- This means that TrackMe automatically identifies the difference of the search when a change is detected against the previously known version, stored as part of the `diff_<context>` fields
- It will also attempt to identify the user that performed the change and when this change occurred (if the change occurred via Splunk Web/Splunk API, not application configuration

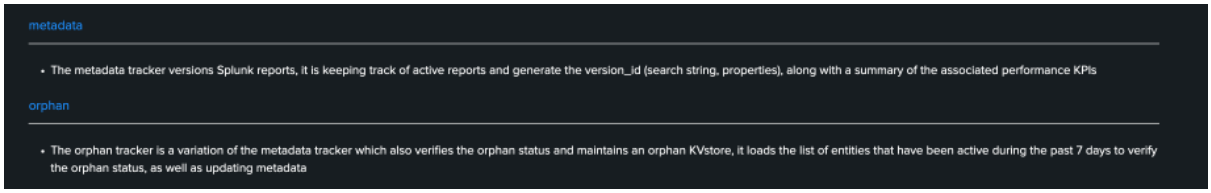
push)



5.1 How the Metadata Versioning works under the cover

This activity is performed by two associated scheduled TrackMe trackers:

- metadata
- orphan

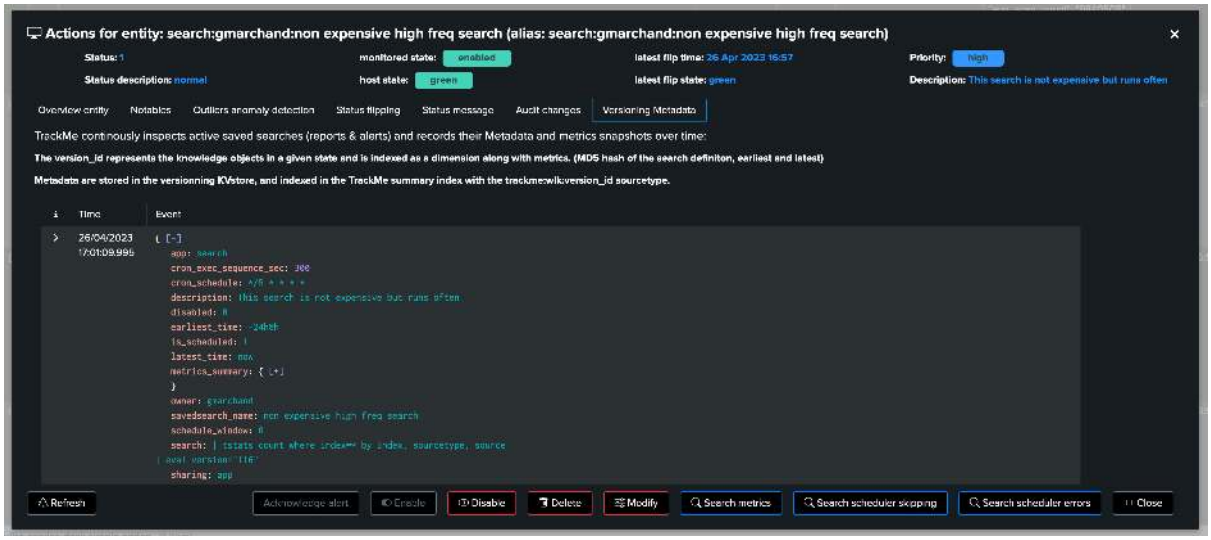


Via a Python based integration, TrackMe maintains the knowledge of scheduled entities, and identity a certain state as an MD5 hash which is composed by:

- The search code definition, basically the whole SPL statement whatever its complexity is
- The earliest time quantifier
- The latest time quantifier

5.2 Accessing Metadata information via the UI

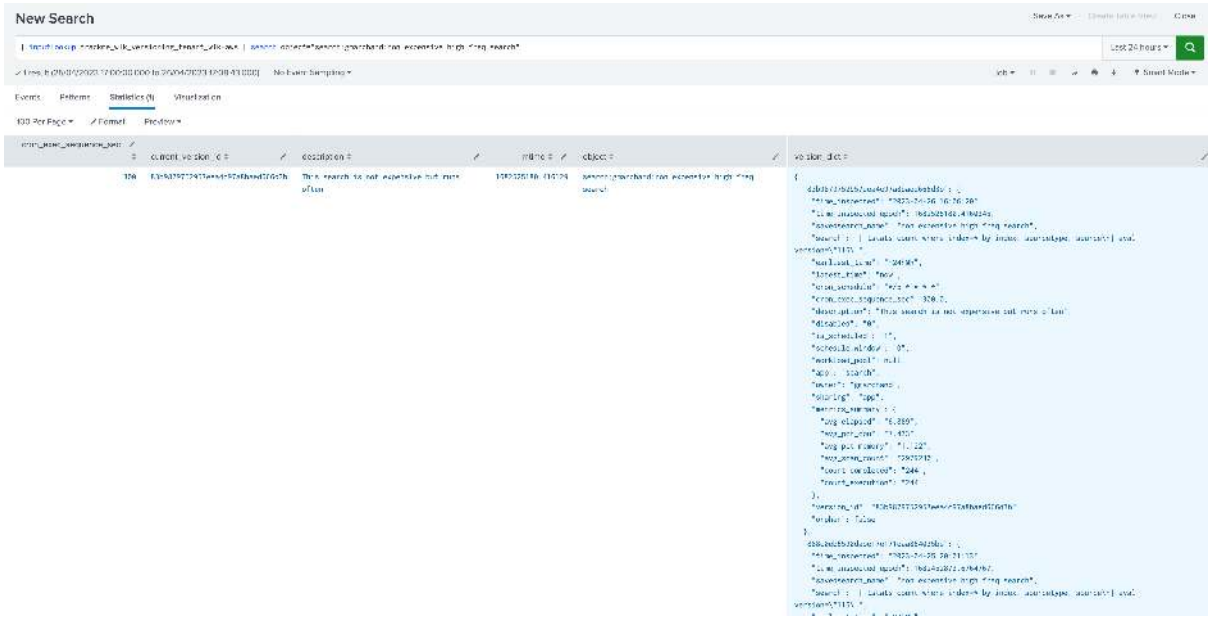
Regularly, active entities are inspected, if a change is detected, a new version_id is defined, a new set of Metadata are extracted and stored in a KVstore collection:



5.3 Accessing Metadata information via the KVstore

The Metadata version is stored in a JSON structure within the KVstore record associated with a given entity, in a persistent fashion:

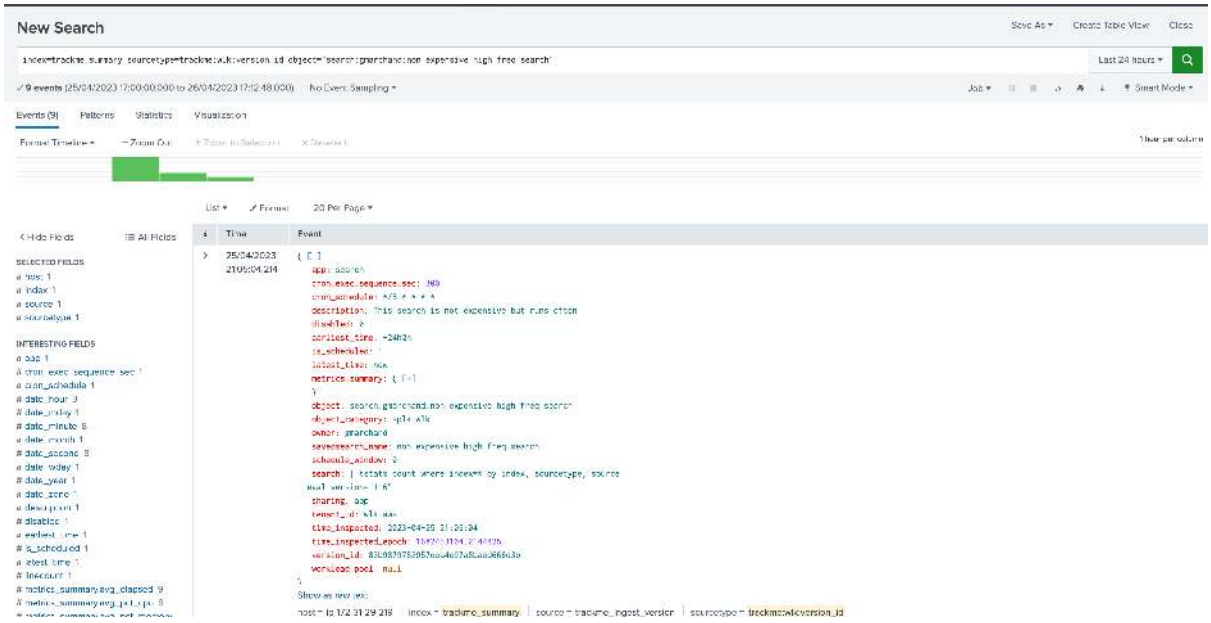
```
| inputlookup trackme_wlk_versioning_tenant_<replace with tenant_id> | search object="
↪<replace with object name>"
```



5.4 Accessing Metadata information via TrackMe indexed events

When TrackMe detects a new version of a monitored scheduled entity, it will as well generate an event in the trackme_summary index of the tenant, with the sourcetype:

```
index=trackme_summary sourcetype=trackme:wlk:version_id tenant_id="<replace with␣
↪tenant_id> object="<replace with object name>"
```

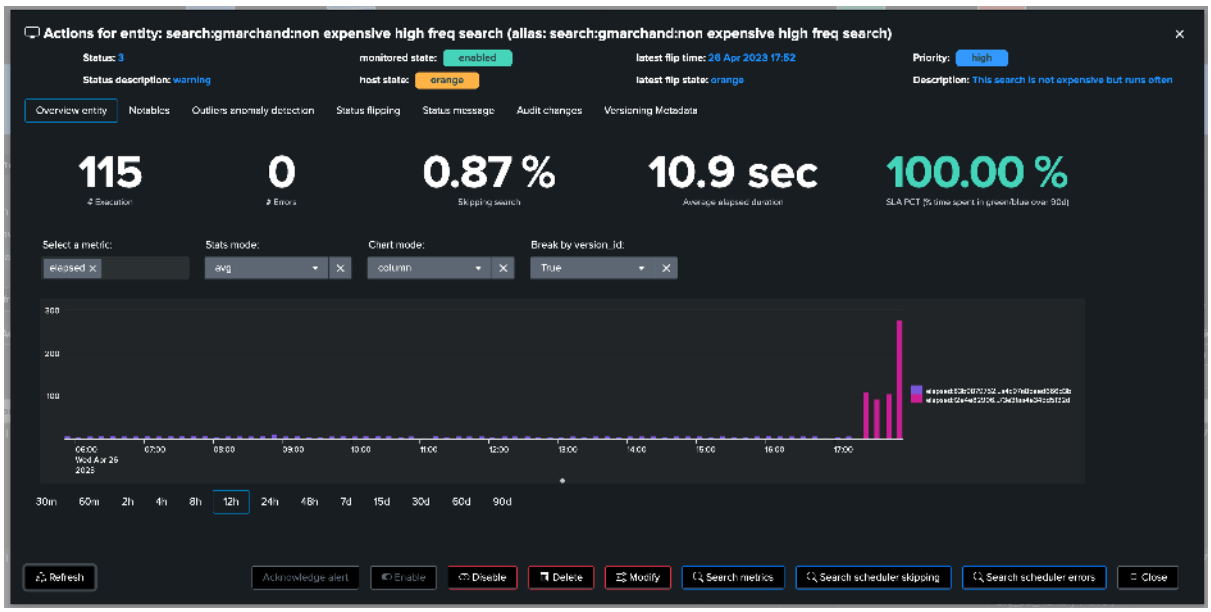


5.5 Use cases for versioning

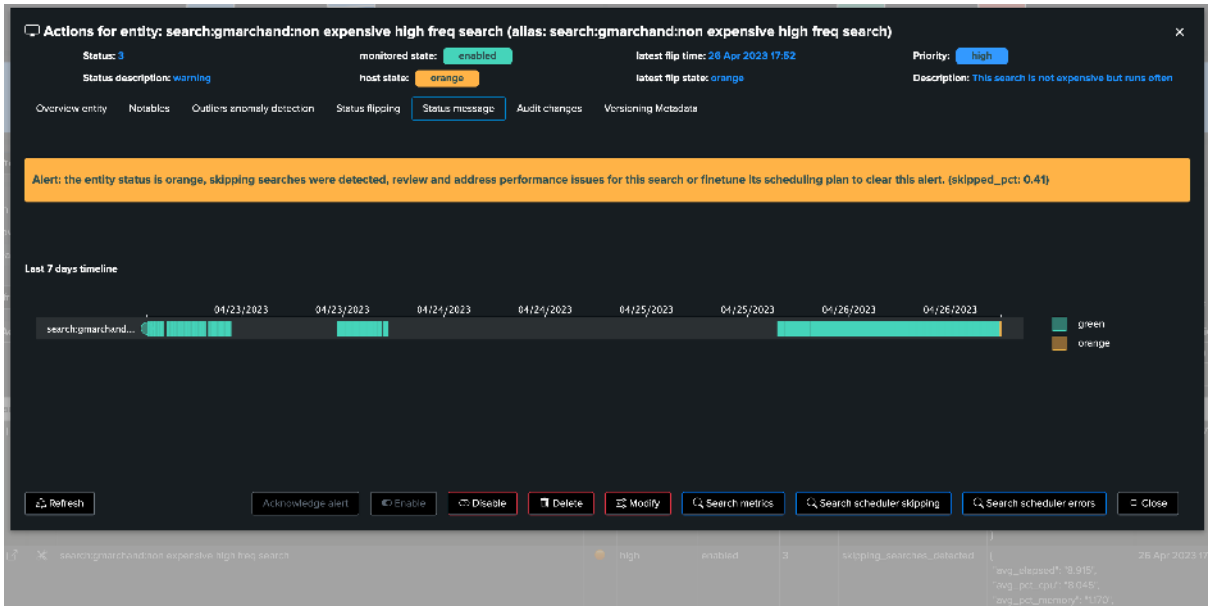
So, what are the use cases for the versioning TrackMe feature, associated with TrackMe's workflow, incident management, metrics generation and so forth? There are many valuable use cases of course.

For instance, it becomes fairly easy to link a change in the search logic to a massive increase of its run time or computing costs, potentially suddenly leading to alerts in TrackMe caused by skipping searches.

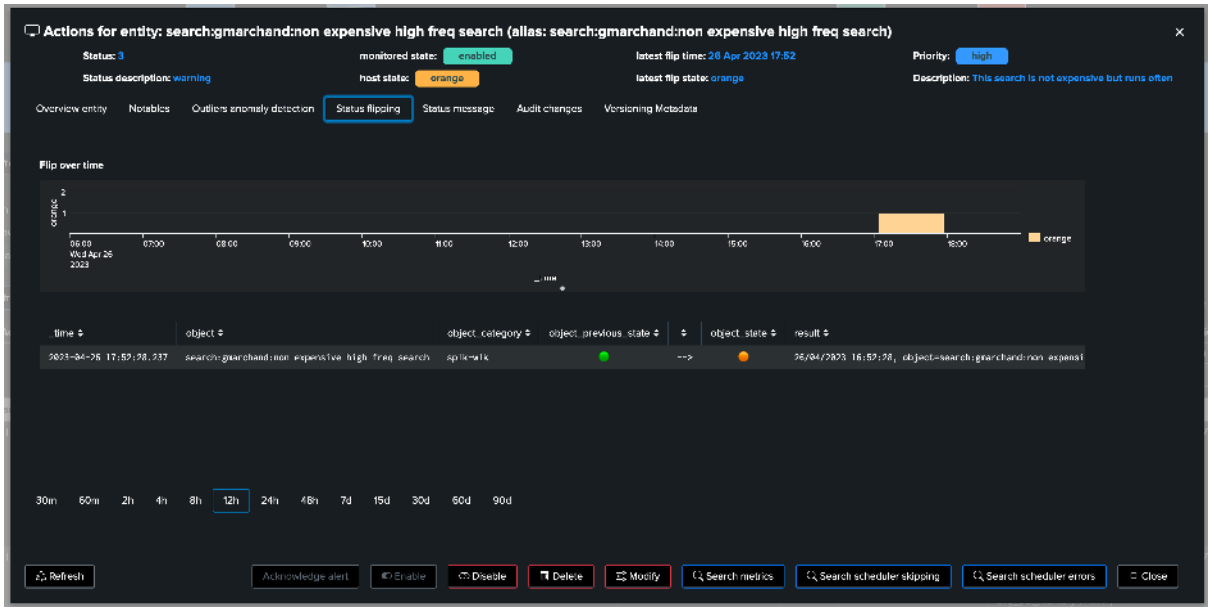
When accessing the metrics overview, use the break down selector to add the version_id as part of the break by statement:



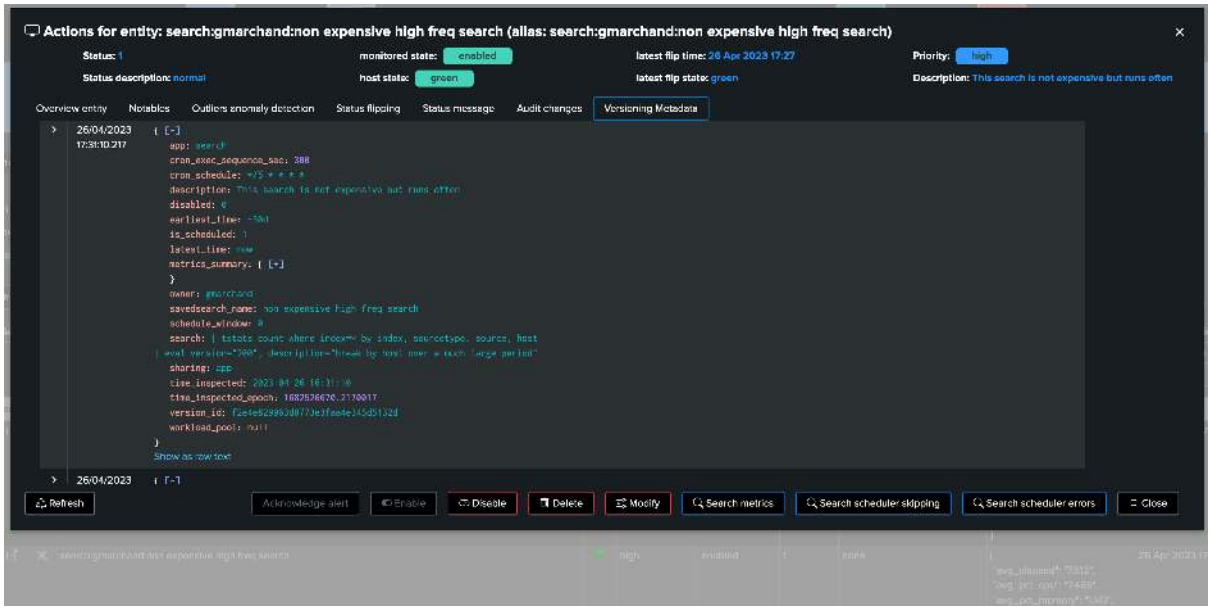
Perhaps TrackMe started to detect bad behaviour of the scheduled object, for instance starting to generate a certain level of skipping searches:



Wait what, don't we have information about flipping statuses? We also know the "when":



As we can see in the chart, a clear increase in in the elapsed (run time in seconds) became suddenly visible, which we have been able to easily associate with a change of the Splunk knowledge object itself:



A search which was very efficient suddenly starts consuming a huge amount of resources, potentially impacting the functional use case itself, perhaps even the platform and starting to cost real money to the company, with the need for additional resources and so forth!

Use cases for the Workload features and the versioning are legions, its value grows exponentially along with your deployment size, scale and maturity.

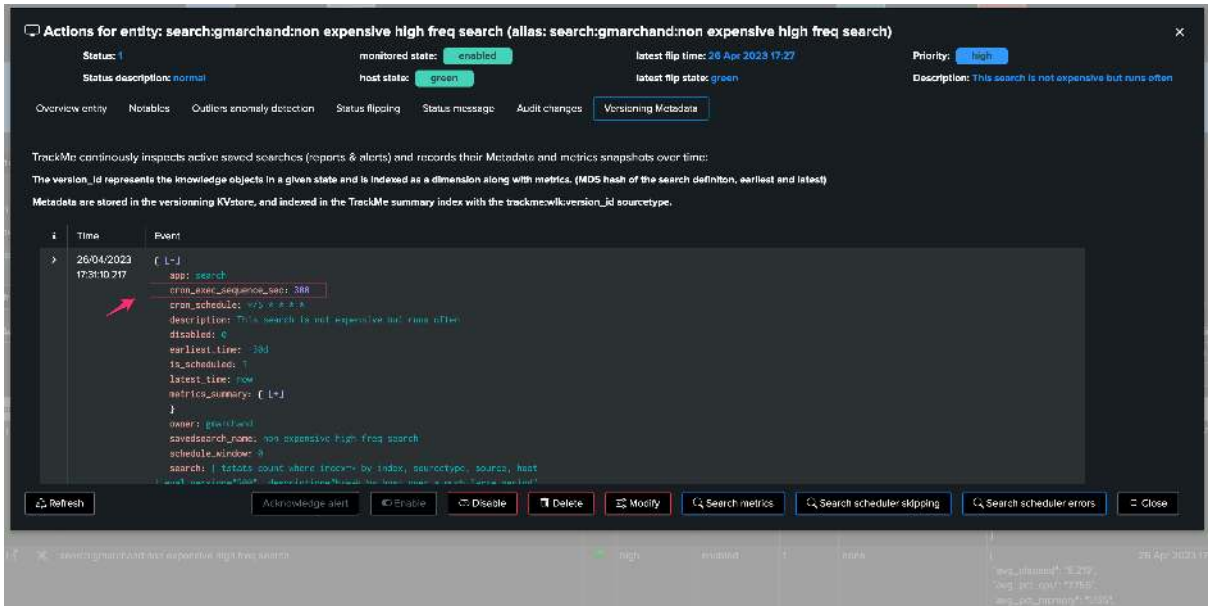
9.11.6 6. Delayed searches execution detection

As part of the Workload component, TrackMe verifies that scheduled entities have been executing properly according to their requested schedule.

A delayed search execution can have severe implications, from a functional perspective this can mean security events which are not considered properly for instance, with all the range of potential consequences.

This verification is called "execution_delayed" anomaly reason status, and works the following way:

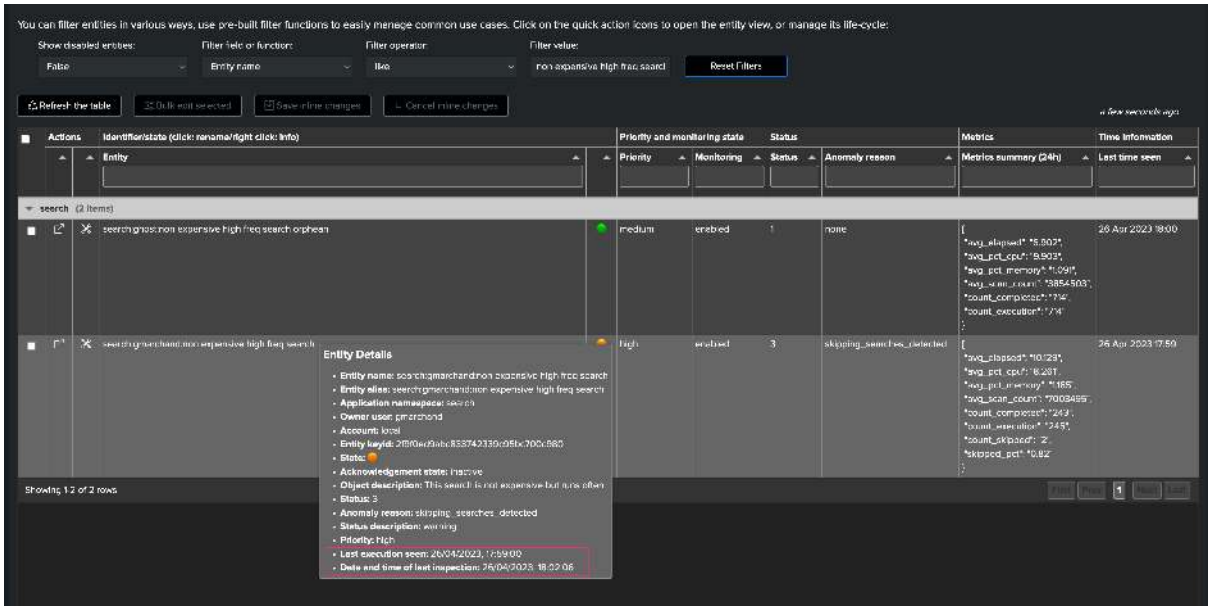
- When the versioning components inspect the Metadata of an entity, it retrieves the cron schedule expression too, which defines how often a scheduled is going to be scheduled
- This cron schedule is transformed into a per seconds value defining how often the search should have been executing (using the croniter Python library), and stored in the field `cron_exec_sequence_sec` along with the metadata record of the entity



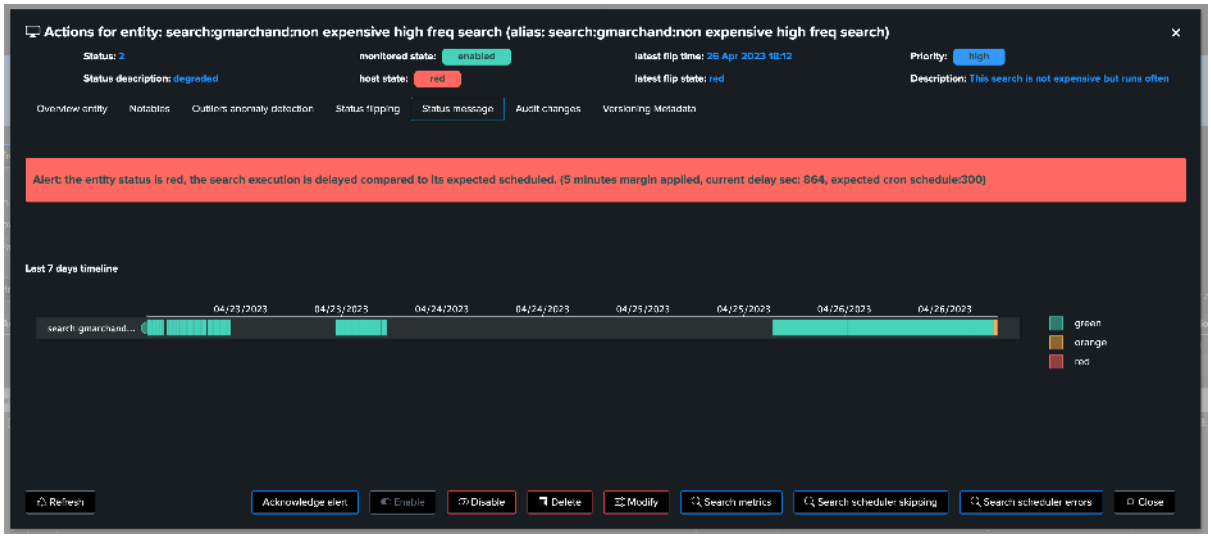
Finally, TrackMe verifies and stores the scheduled successful execution traces, if a search which should have executed at least once in the past 5 minutes has not been since active since more than 5 minutes, plus the grace period, then TrackMe can consider this entity as unhealthy and trigger and alert accordingly,

For instance, let's disable the scheduling of an active schedule (a mistake? bad behaviour?), after a short period of time, TrackMe detects the conditions which immediately impacts the entity, generates an alert and TrackMe notable!

we can observe the dates of latest inspection and latest execution in the contextual menu:



After some minutes, TrackMe detected that the scheduled search is delayed:

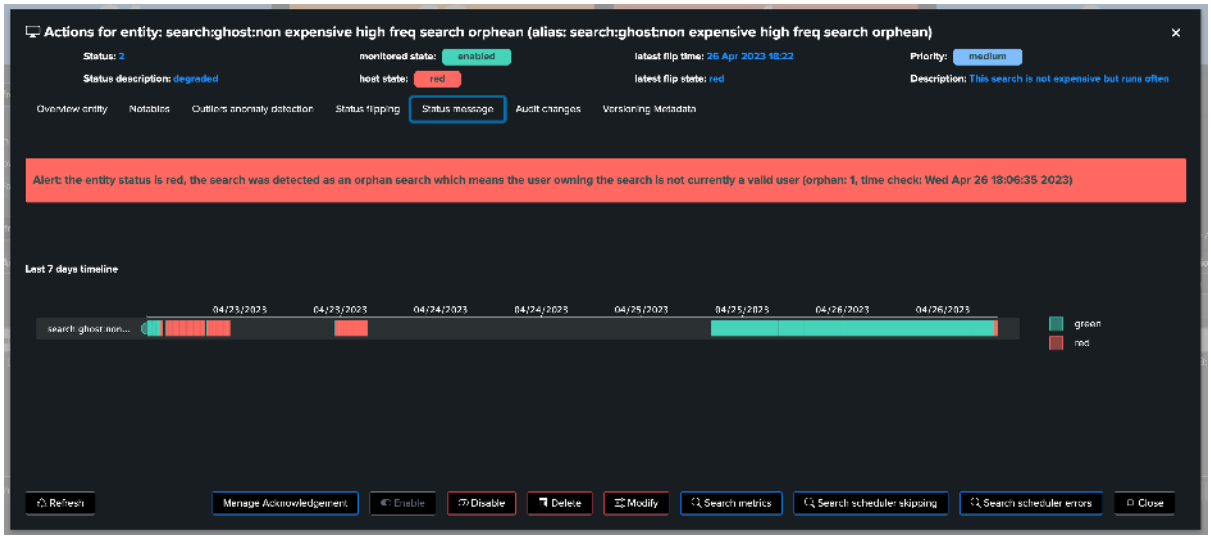


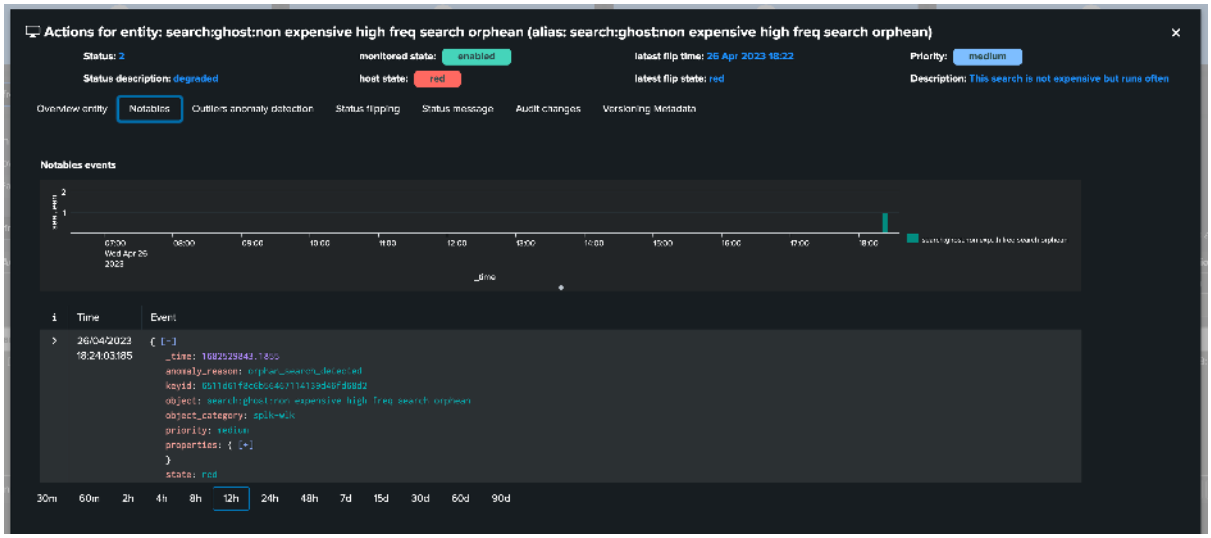
From this stage, we automatically detect an abnormal condition for our critical scheduled activity, and we can start acting accordingly before it starts having any severe consequences!

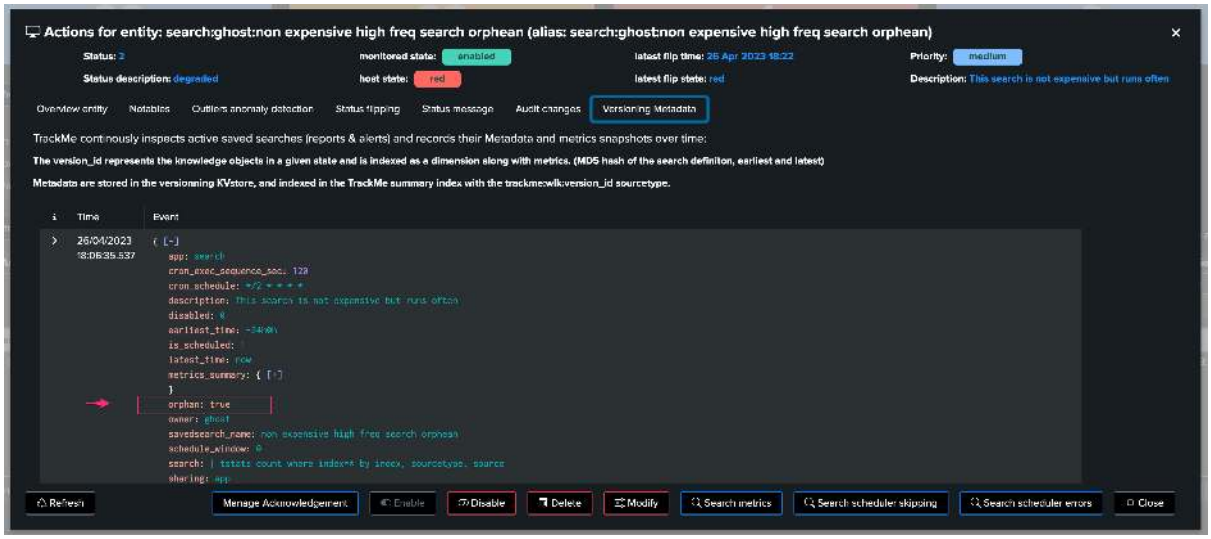
9.11.7 7. Orphan searches detection

Orphan search condition is also detected by TrackMe, for any search that has been actively discovered, in short:

- Once per hour, TrackMe verifies via the orphan tracker the current status of all active scheduled objects for the past 7 days
- It updates the Metadata information with the `orphan` boolean
- In return, TrackMe as part of its condition verifies that the search has not been detected as an orphan search, which otherwise will be considered as a critical condition for this entity



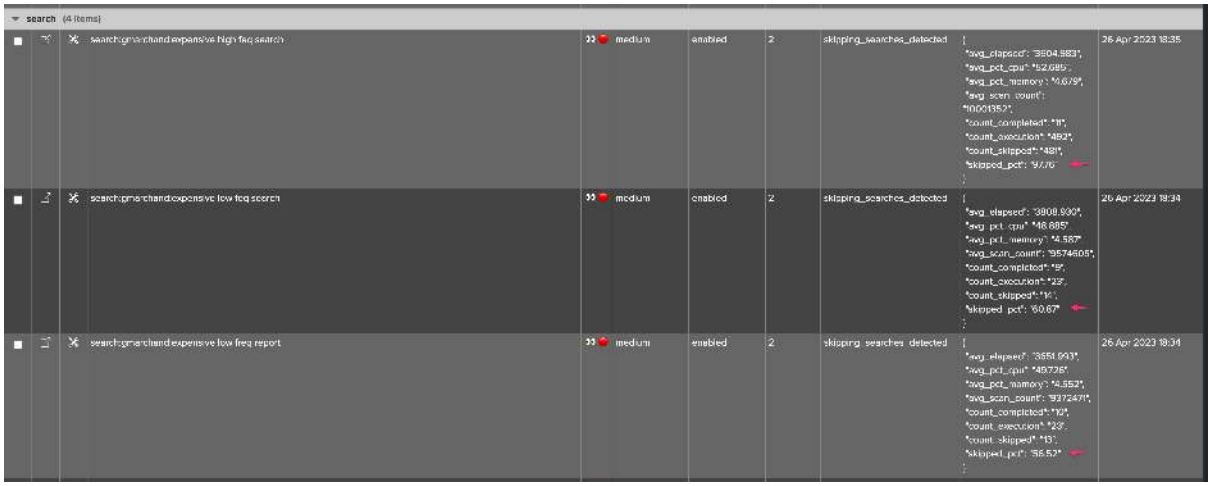


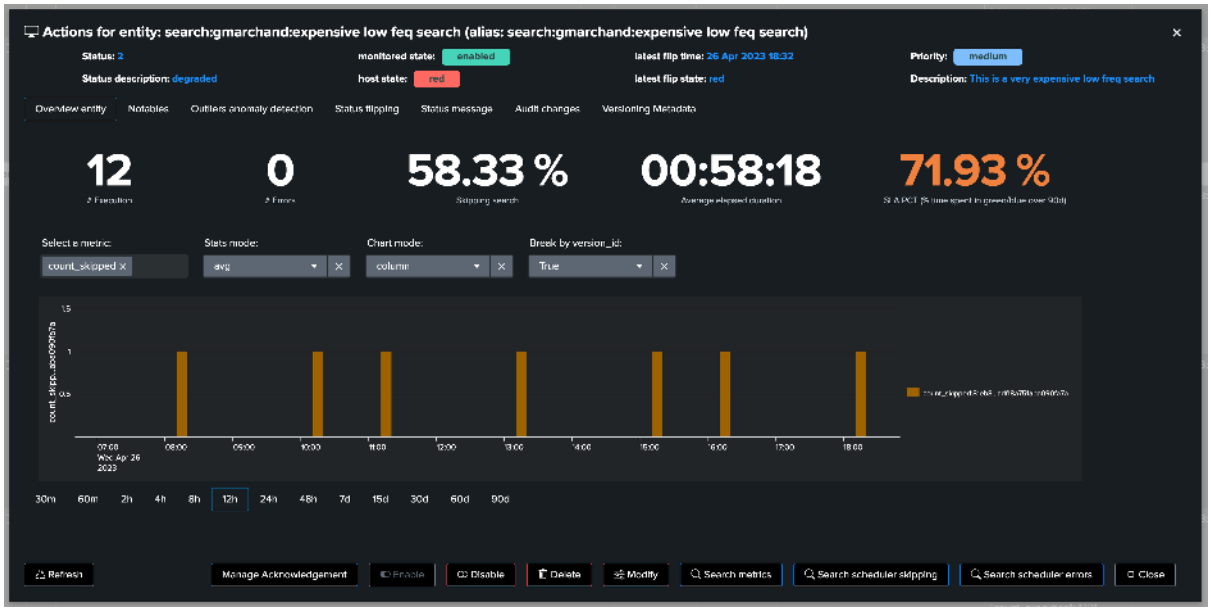


9.11.8 8. Skipping searches detection

Skipping search condition is achieved by comparing the collected count_execution and count_completed collected metrics, which gives the skipping search percentage:

- If the skipping search percentage is somewhere between 0 to less than 5%, the entity status will be **orange**
- Over 5% of skipping search percentage, the entity will turn **red**
- The skipping percentage can be seen in the summary metrics JSON in the table, or via the metric over time inspector

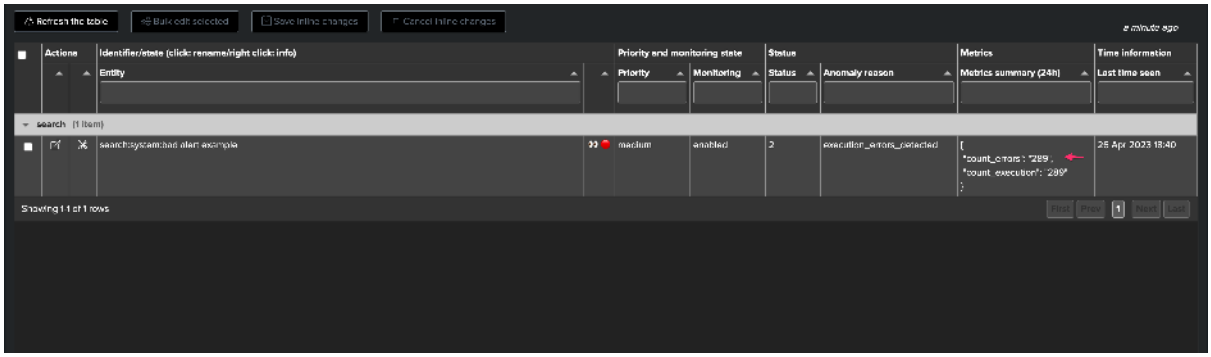


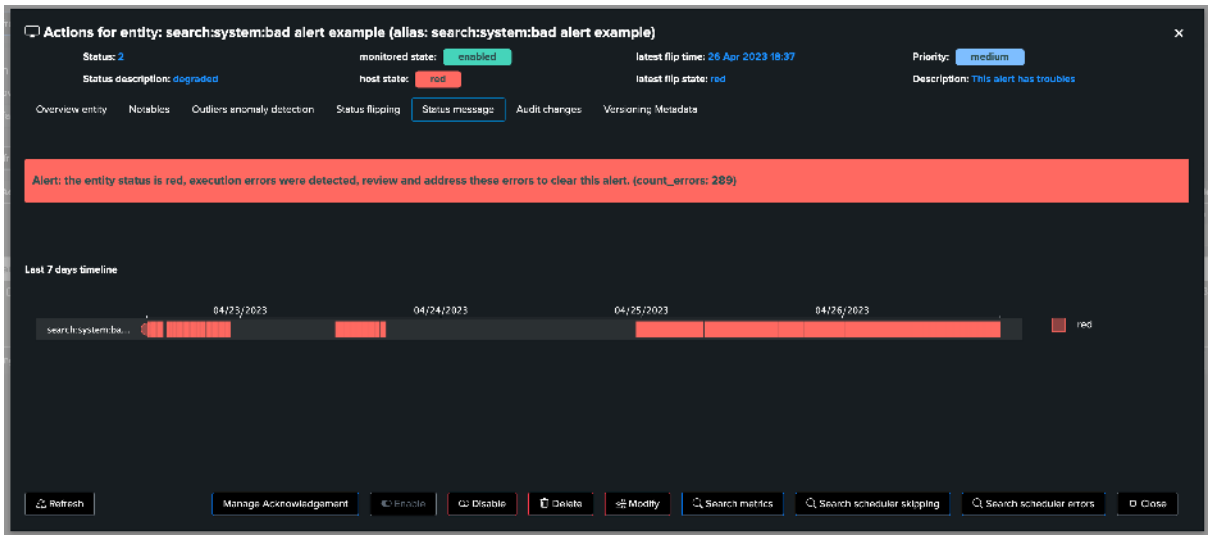


9.11.9 9. Error executions detection

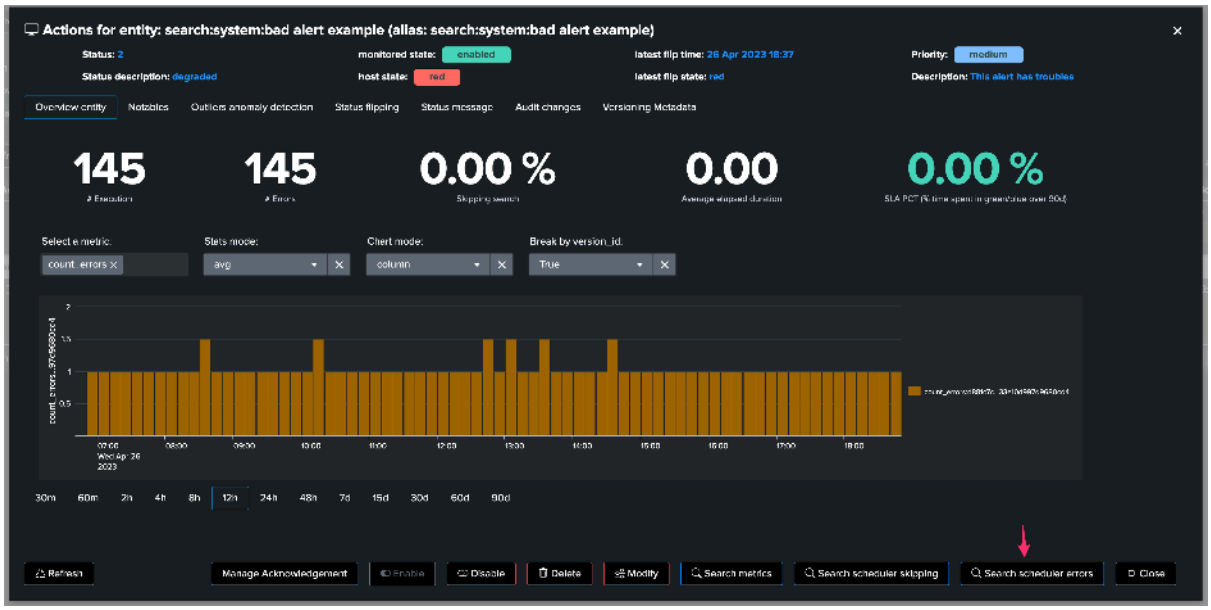
Detecting at scale and automatically execution errors is another challenge TrackMe tackles in the Workload component:

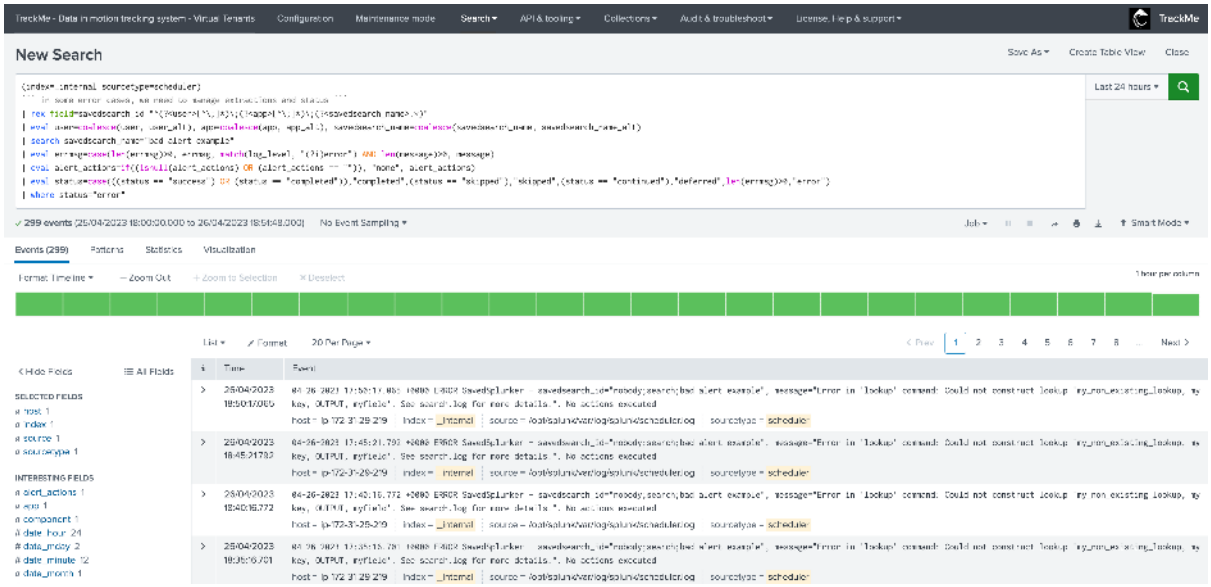
- Execution errors can happen for all sorts of reasons, for instance a related knowledge object (macro, lookup etc.) which is not available to the search logic
- Errors in the application development, lack of qualification, unexpected deployment side effect, etc. many combinations can lead to a scheduled search to be failing pretty much silently
- TrackMe handles the challenge by continuously looking at the scheduler activity, and extracting the executions errors turning these into a metric which we can take into account





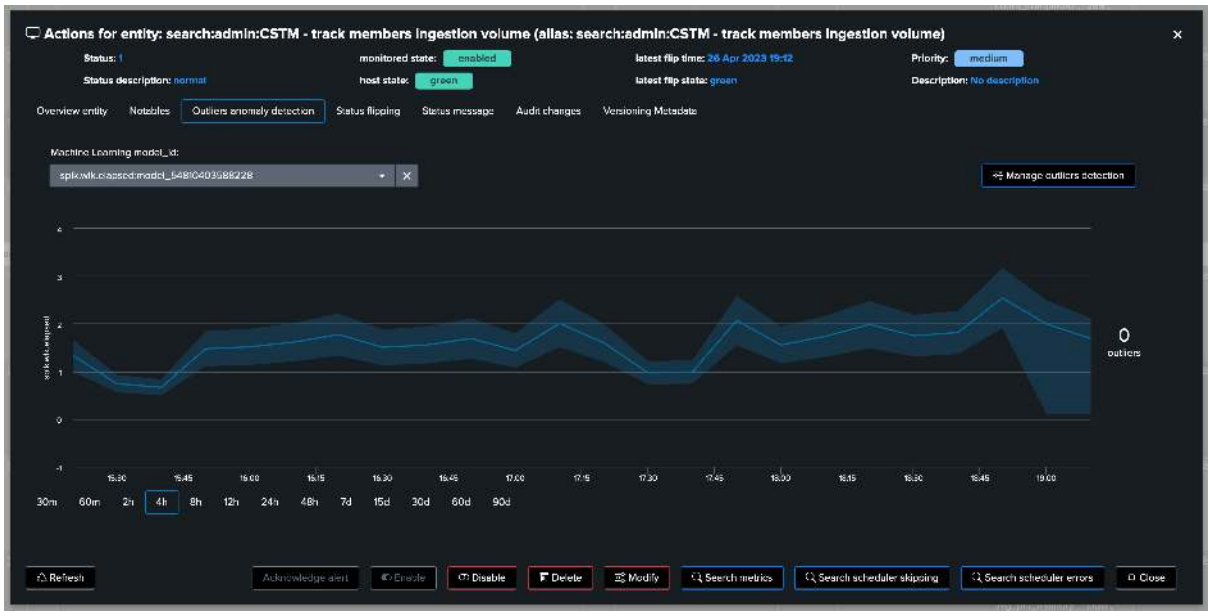
From the overview (currently looking at the metrics), you can as well directly access to the scheduler errors:



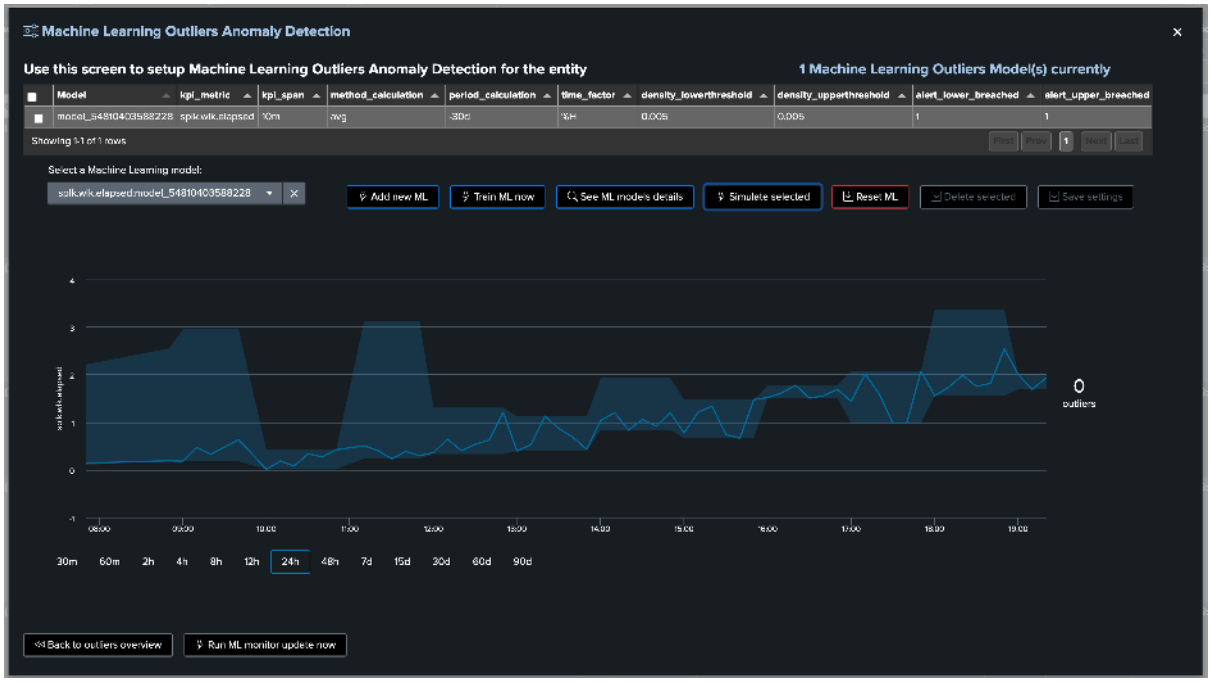


9.11.10 10. Machine Learning Outliers detection

Machine Learning Outliers detection can be used to detect abnormal trends for the scheduled entities, based on any of the available KPIs:



Machine Learning models created and trained by default depend on the Virtual Tenant configuration, additional ML models can be created on demand:



9.12 Splunk SOAR Cloud & on-premise monitoring and active actions in TrackMe

Hint

Version 2.0.46 and later

- The SOAR integration requires TrackMe version 2.0.46 and later
- SOAR High Availability management for Automation Brokers is available from TrackMe version 2.0.50 and later

Hint

New since TrackMe 2.1.10: REST lookup capabilities and review of the adhoc/playbook failure detections use cases

- In TrackMe 2.1.10, we have introduced a new REST lookup capability with the command `trackmesplksoarlookup` and fully refactored the adhoc and playbook failure detections use cases to leverage this new capability.
- With this command, TrackMe performs REST based live lookups to SOAR, allowing to retrieve the metadata associated with the SOAR objects such as the playbooks, assets, applications and brokers associated with the detection.
- These metadata information are then added to the extra_attributes field of the resulting entities, allowing to have a full context of the detection and to be able to review the details of the detection directly from the TrackMe UI.

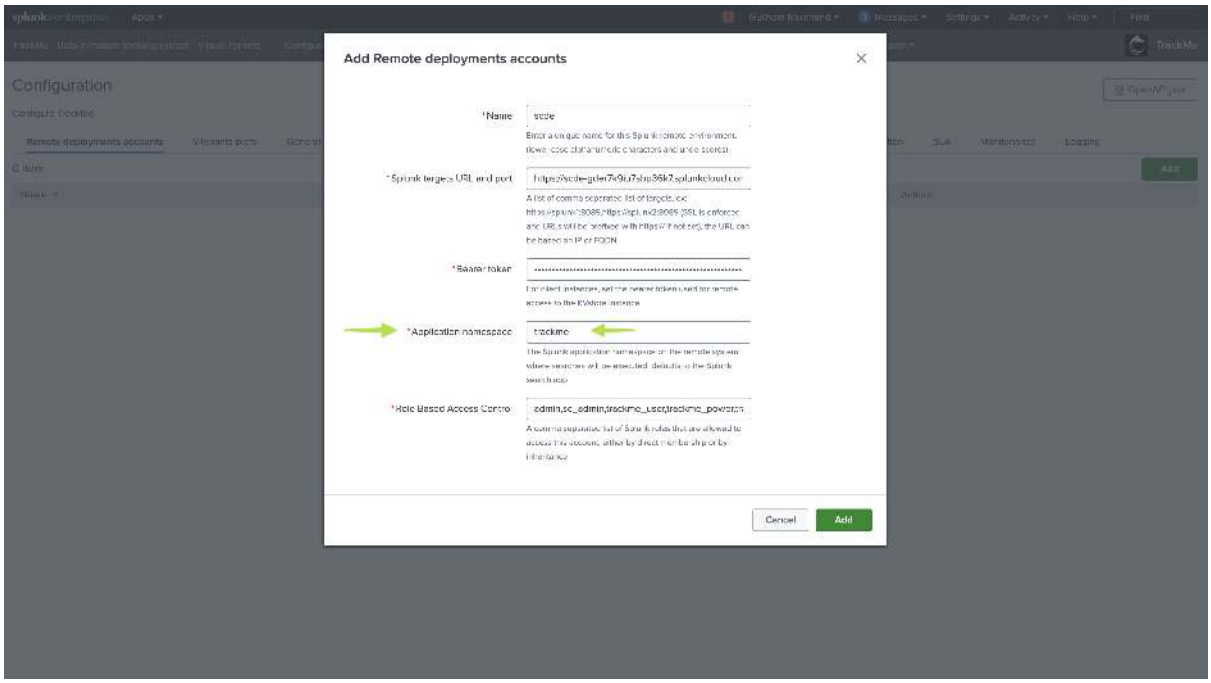
Hint

New since TrackMe 2.1.11: REST based SOAR monitoring of playbooks run concurrency and statuses

- In TrackMe 2.1.11, we have introduced a new REST based SOAR monitoring of playbooks run concurrency and statuses, leveraging the REST lookup capabilities.
- This allows to monitor the concurrency of playbooks run and their statuses, and to be alerted in case of issues.
- This is available out of the box with the use case `splk_soar_concurrent_playbooks`.

Splunk Remote Account: You must use the trackme namespace

- If you use a remote account to run the SOAR active monitoring (Automation Brokers High Availability), you must use the TrackMe namespace when configuring the Splunk remote account.
- This is required because TrackMe by default is shared at the application level, so unless you share TrackMe at the system level, your remote account would not be able to access to TrackMe custom commands on the remote partner.



Minimal permissions for TrackMe SOAR monitoring

- On the Splunk side, when setting up a dedicated Splunk user account for the purposes of SOAR monitoring, notably active monitoring such as the Automation Broker high availability management, you must honor some minimal permissions.
- This is applicable to both local monitoring, and **remote** account via TrackMe.
- The user account must be a member of the `trackme_admin` role (or have the `trackmeadminoperations` capability), the user should also be a member of the Splunk builtin `user` account, or have equivalent capabilities.
- Regarding access to Splunk indexes, you must grant access to the service account to all `phantom_*` indexes, and eventually to the `_internal` indexes depending on any custom Flex Object tracker you would create in addition with TrackMe builtin use cases.
- Finally, resources should also be sufficient to avoid hitting a max concurrent search quota, and storage quota, for instance up to 5 concurrent search and 1GB file storage quota.

9.12.1 1. Introduction to SOAR monitoring

TrackMe provides builtin use cases to actively and efficiently monitor at scale one or more Splunk SOAR environments.

SOAR Cloud & on-premise monitoring is performed via the TrackMe Flex Object component (splk-flx) which is a restricted component not available with the Free community edition of TrackMe.

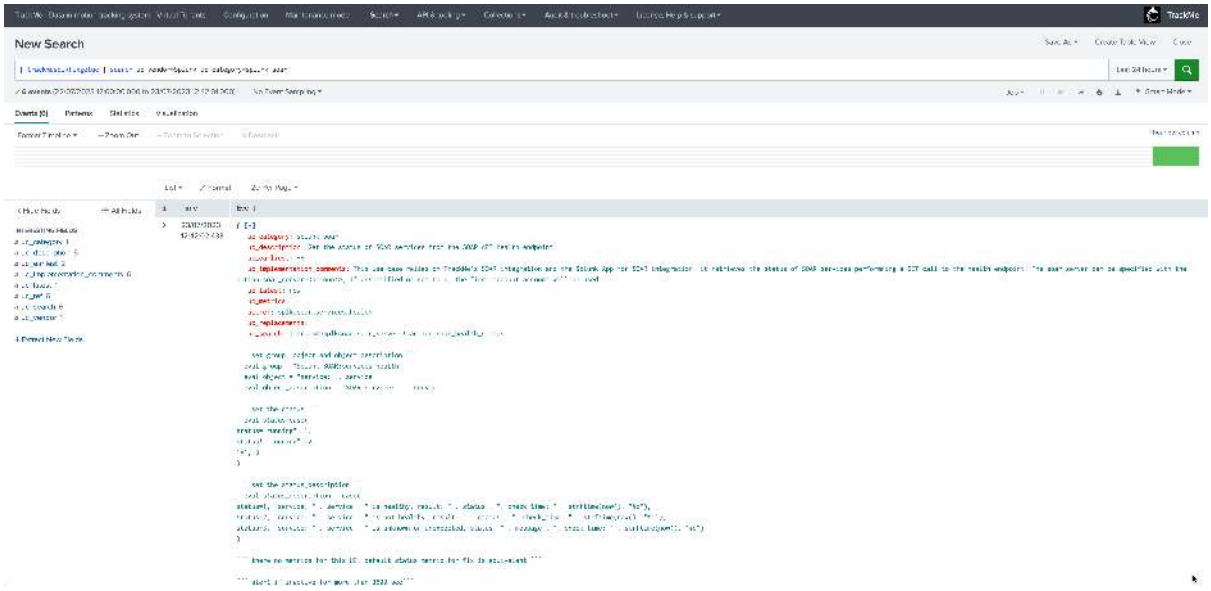
The monitoring relies on TrackMe's SOAR integration and the Splunk Application for SOAR integration, associated with TrackMe's Flex Object concepts, and provides the following use cases:

uc_r	uc_descrip	uc_implementation_comments
splk	Monitor the concurrency of playbooks run	This use case relies on SOAR events indexed in Splunk through the Splunk Application for SOAR integration. It monitors for playbooks run concurrency and triggers an alert if the concurrency limit is exceeded. This use case also leverages a live REST API lookup capability to retrieve the SOAR object context metadata.
splk	Monitor for SOAR adhoc actions failures	This use case relies on SOAR events indexed in Splunk through the Splunk Application for SOAR integration. It monitors for app run action failures and triggers shall there be any failures, the alert will clear itself once failures cannot be found in the time range. You may want to adapt the search time range according to your preferences to keep the alert active for a certain amount of time after detection happened. This use case also leverages a live REST API lookup capability to retrieve the SOAR object context metadata.
splk	Monitor for SOAR playbook actions failures	This use case relies on SOAR events indexed in Splunk through the Splunk Application for SOAR integration. It monitors for playbook action failures and triggers shall there be any failures, the alert will clear itself once failures cannot be found in the time range. You may want to adapt the search time range according to your preferences to keep the alert active for a certain amount of time after detection happened. This use case also leverages a live REST API lookup capability to retrieve the SOAR object context metadata.
splk	Runs an active assets health check, parse and render results	This use case relies on TrackMe's SOAR integration and the Splunk App for SOAR integration, if active_check is set to True, it performs an active asset health check (connectivity test) by discovering assets, running a POST call to request the connectivity test, then parses and renders results for each asset. If active_check is set to False, it simply parses assets and checks the latest connectivity test achieved by SOAR and renders results. Running the active check requires the automation user in SOAR to have the edit asset permissions, if you do not want this or cannot set it, you can disable the active check and rely on SOAR doing it automatically once per day, therefore a failure detection can take up to 24 hours, enable active check is recommended. The soar_server can be specified with the option soar_server=<account>, if unspecified or set to *, the first account will be used. You can restrict the list of assets to be taken into account using the assets_allow_list option, and/or avoid some assets to be taken into account using assets_block_list option. The SOAR automation used for the SOAR server account configuration needs to be able to update assets if active check is enabled as it performs an active test request against the SOAR REST API.
splk	Monitor SOAR CPU load from the SOAR API health endpoint	This use case relies on TrackMe's SOAR integration and the Splunk App for SOAR integration, it retrieves the 1/5/15 minutes CPU load performing a GET call to the health endpoint. The soar_server can be specified with the option soar_server=<account>, if unspecified or set to *, the first account will be used.
splk	Monitor SOAR CPU load from the SOAR API health endpoint	This use case relies on TrackMe's SOAR integration and the Splunk App for SOAR integration, it retrieves and calculates the percentage of memory usage of the SOAR instance performing a GET call to the health endpoint. The soar_server can be specified with the option soar_server=<account>, if unspecified or set to *, the first account will be used.
splk	Get the status of SOAR services from the SOAR	This use case relies on TrackMe's SOAR integration and the Splunk App for SOAR integration, retrieving the status of SOAR services performing a GET call to the health endpoint. The soar_server can be specified with the option soar_server=<account>, if unspecified or set to *, the first account will be used.

These use cases are provided via the Flex Object use cases library, but not that you can also manually implement new use cases, or customise builtin use cases as needed.

You can review the use case details ahead of their creation in TrackMe with the following command:

```
| trackmesplkflxgetuc | search uc_vendor=Splunk uc_category=splunk_soar
```



9.12.2 2. Requirements for SOAR monitoring

SOAR Cloud & SOAR on-premise

TrackMe's SOAR integration is compatible with both SOAR Cloud and SOAR on-premise deployments.

Splunk Application for SOAR integration

TrackMe's integration for SOAR relies on the Splunk SOAR integration for Splunk provided by the Splunk App for SOAR:

- https://splunkbase.splunk.com/app/6361

Hint

Splunk App for SOAR

- TrackMe performs active and bi-directional interactions with the SOAR API
- TrackMe relies on the Splunk App for SOAR integration and connectivity definition to achieve this job
- You do not need to have any additional configuration in TrackMe as long as the Splunk App for SOAR is installed and configured
- Some of the SOAR use cases rely on the SOAR events indexed in Splunk (index-phantom_*) which is also part of the Splunk App for SOAR integration
- TrackMe leverages the Splunk App for SOAR and extends its capabilities even further

TrackMe tenant with the Flex Object component for SOAR

You need a TrackMe tenant with the Flex Object component enabled, you can decide to create a dedicated tenant for the monitoring of SOAR, and use any existing tenant of your choice.

Once the Flex trackers have been created, TrackMe automatically groups the resulting entities for SOAR into the following groups:

uc_ref	grouping
splk_soar_actions_apprun_failures, splk_soar_actions_playbooks_failures	Splunk_SOAR:actions_health
splk_soar_assets_health	Splunk_SOAR:asset_health
splk_soar_infra_load, splk_soar_services_health	Splunk_SOAR:infrastructure
splk_soar_services_health	Splunk_SOAR:services_health
splk_soar_automation_brokers_manage	Splunk_SOAR:automation_broker_manage

TrackMe deployment target

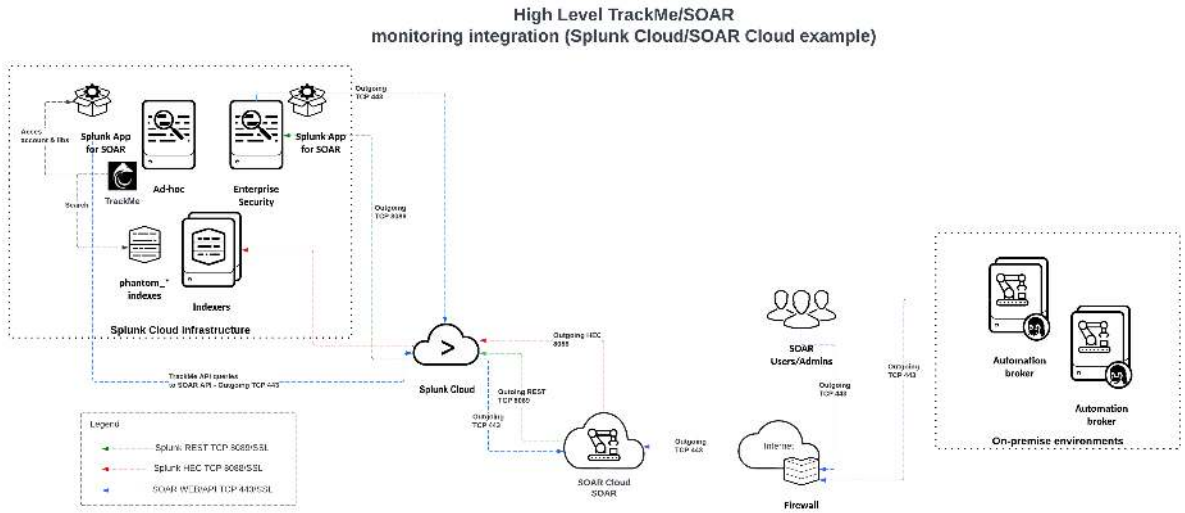
In TrackMe, you can choose where the SOAR monitoring Flex Objects use cases are executed, this can be either "local" or a remote deployment.

Note that some of the use cases would require TrackMe to be deployed on the remote target, as well as the Splunk App for SOAR configured on the remote target.

9.12.3 3. Implementation of SOAR monitoring

3.1 Integration architecture overview

The following diagram represents the integration from an high level perspective:



As a brief summary:

- REST API related use cases in TrackMe imply an active interaction with SOAR API
- TrackMe retrieves and loads the SOAR account configuration from the Splunk App for SOAR, you do not need to define anything in TrackMe
- TrackMe implements its own REST queries against the SOAR API as needed, in some cases such as the Assets connectivity check, this interaction may imply a bi-directional integration with POST and GET calls performed by TrackMe
- Some other use cases only deal with the SOAR data indexed in Splunk, and do not imply an API interaction

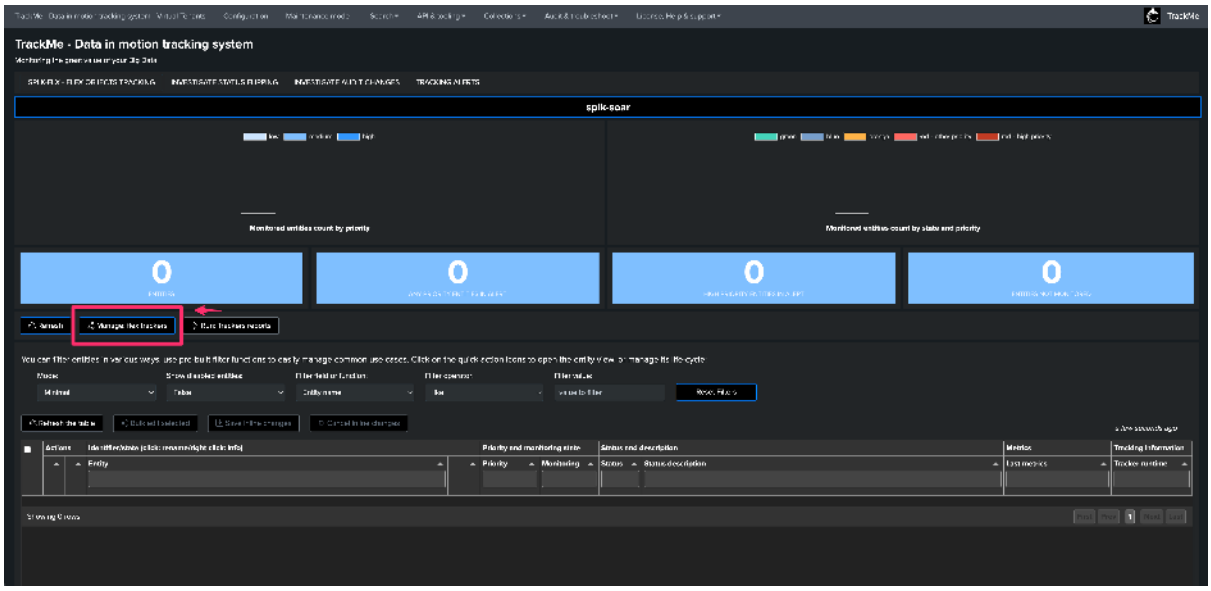
3.2 Creating SOAR Flex Object trackers

Hint

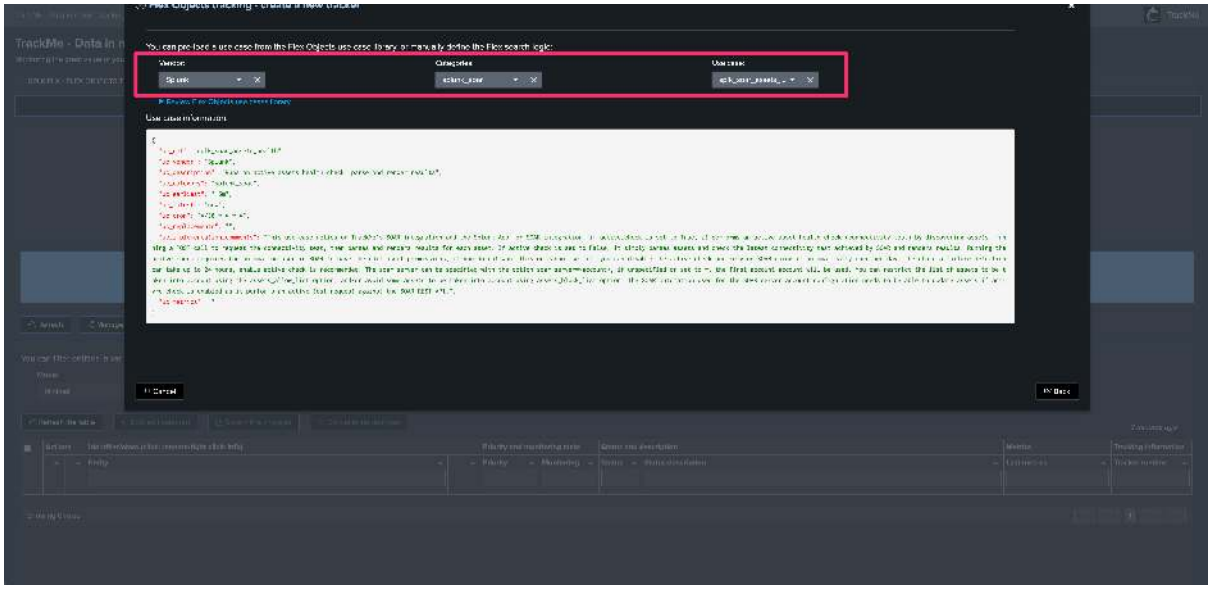
Running actions on a remote Splunk Search Head tier

- You can leverage TrackMe's splunkremotesearch to run these actions transparently on a remote Splunk Search Head tier
- In this case, the Splunk App for SOAR would be installed and configured on the target (not on TrackMe's Search Head)
- However, TrackMe needs to be be installed on the target Splunk Search Head tier, although it is unconfigured and not actively running anything (if you run in Splunk Cloud Victoria, TrackMe is installed on all Search Heads automatically)
- You need to create a TrackMe remote account which uses a token related a Splunk user on the remote Search Head
- TrackMe leverages a minimalist least privileges approach, the user on the remote Search Head tier only needs to be a member of the power role and trackme_admin role (or have equivalent capabilities)
- This requires TrackMe version 2.0.48 and later as we have addressed some issues to allow minimal permissions to be used

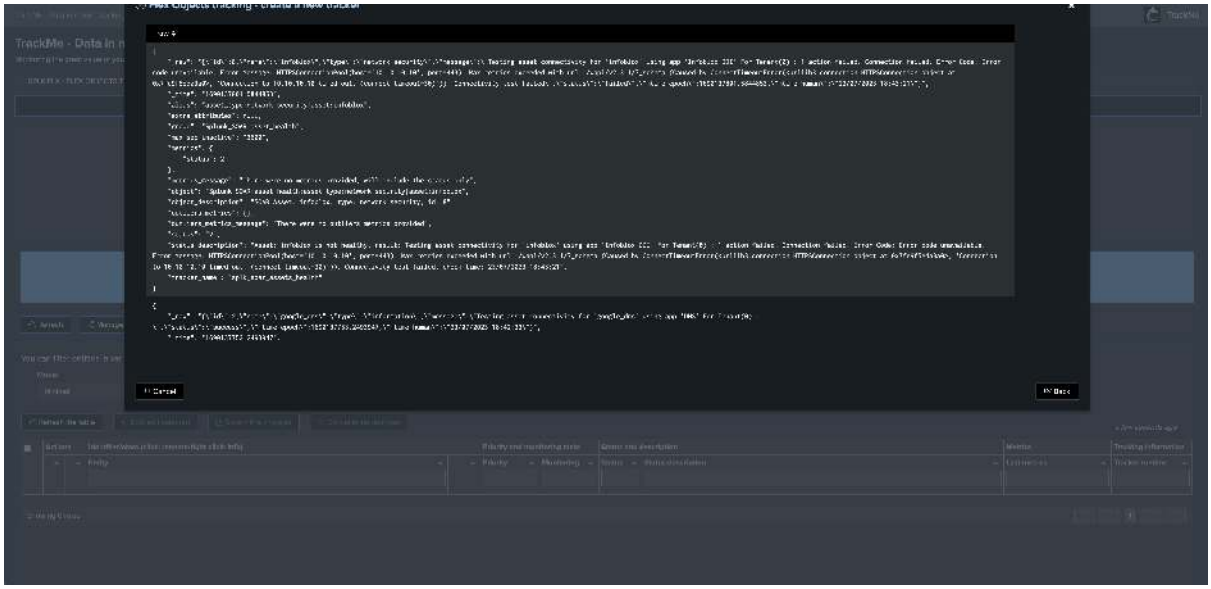
Once a TrackMe Virtual Tenant with the Flex Object (splk-flx) has been created, the setup is really straightforward and done via the UI:



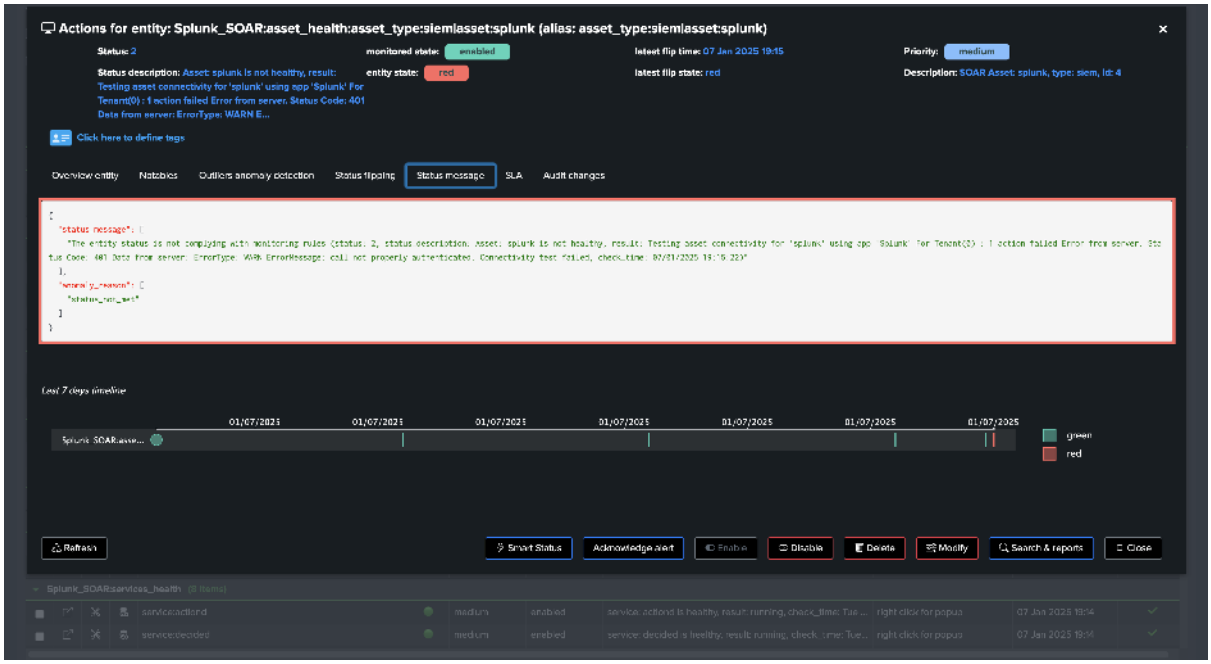
Then click on Create a new Flex Tracker, select Splunk as the vendor and splunk_soar as the category, the next step is to review and validate the results, this is all:



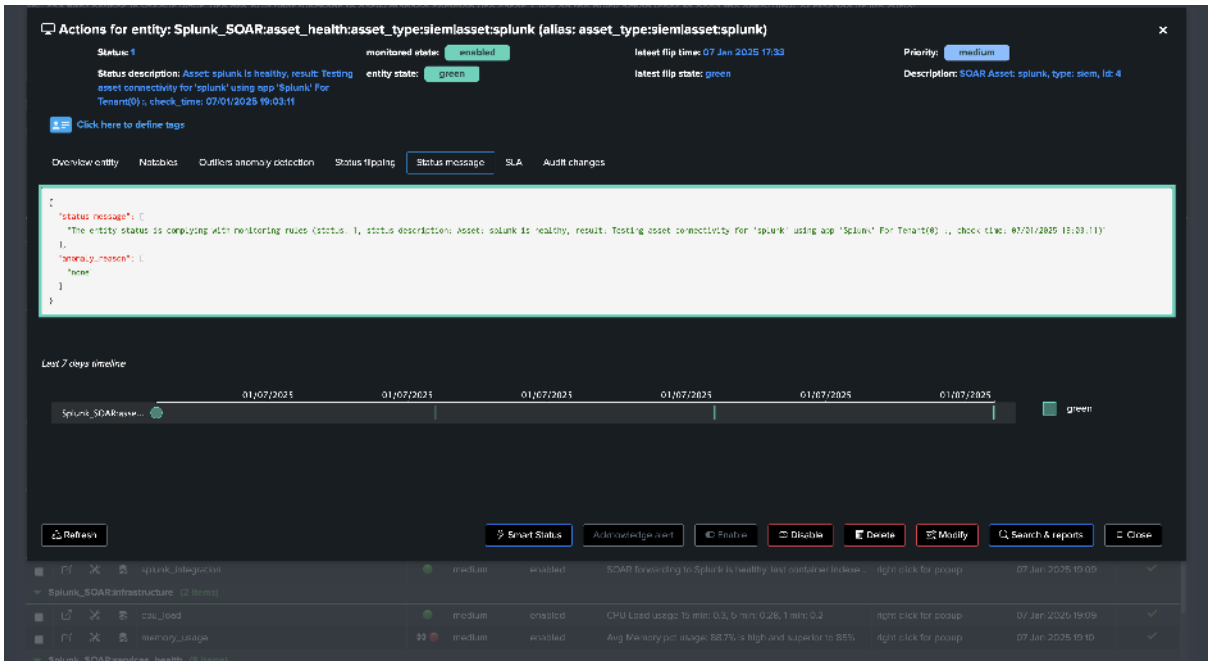
Example of results with the Assets active health check:



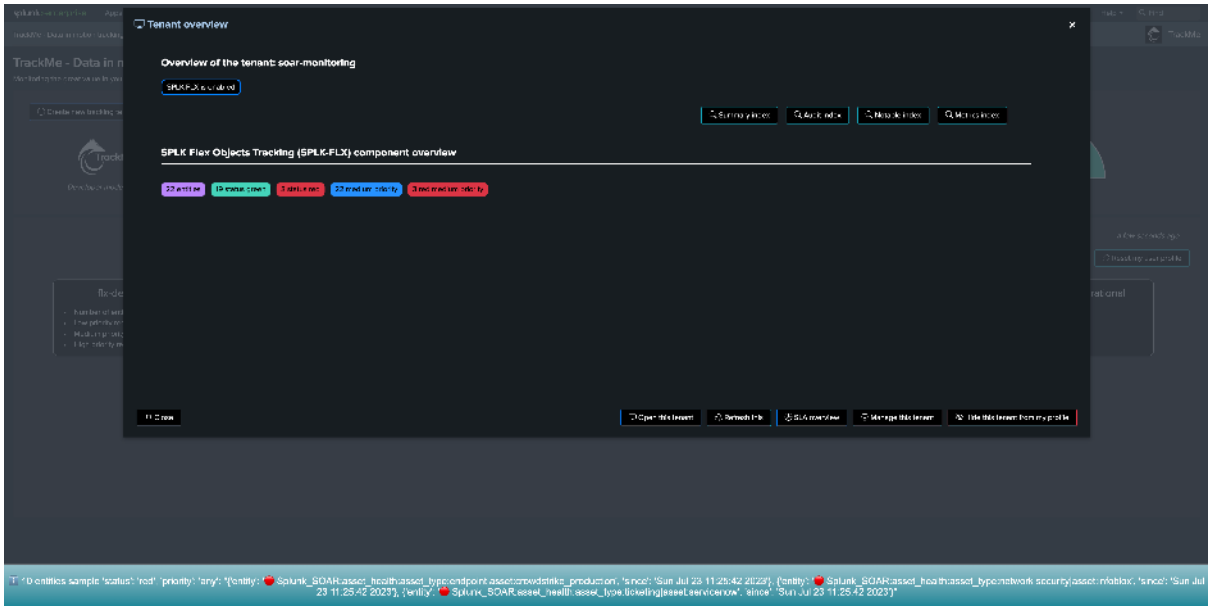
This asset is failing the connectivity check:

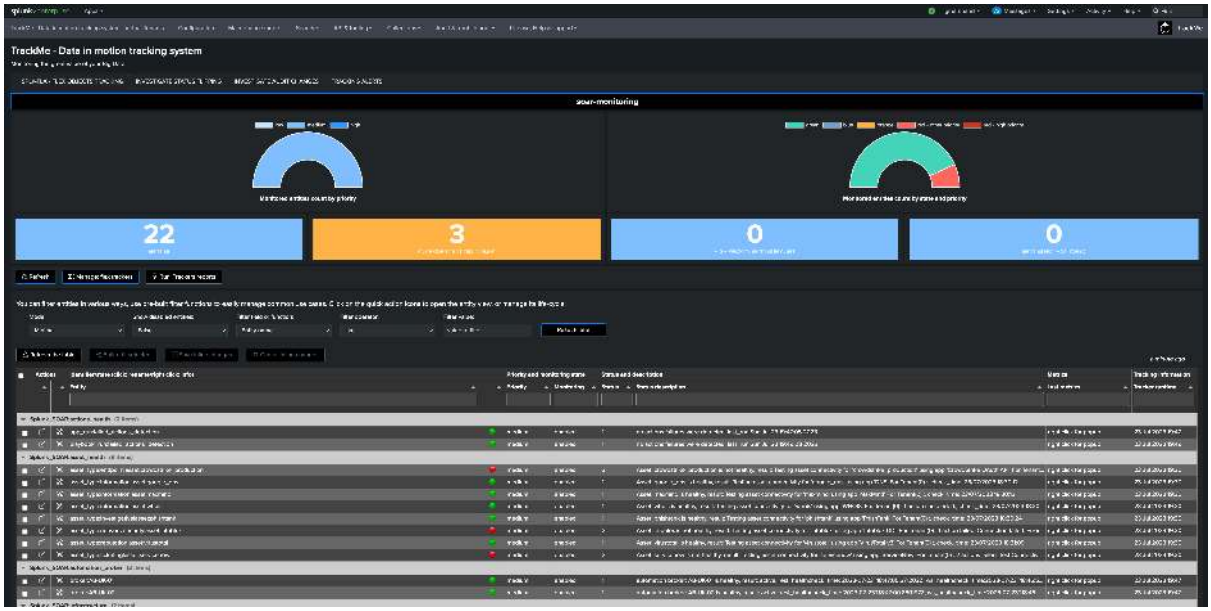


This asset passes successfully:



Setup is completed, the number of resulting entities depends on your environment:





9.12.4 4. SOAR Connectors Assets status active health check

A key monitoring capability for SOAR is to be able to detect as soon as possible when an application and its corresponding asset is experiencing an issue.

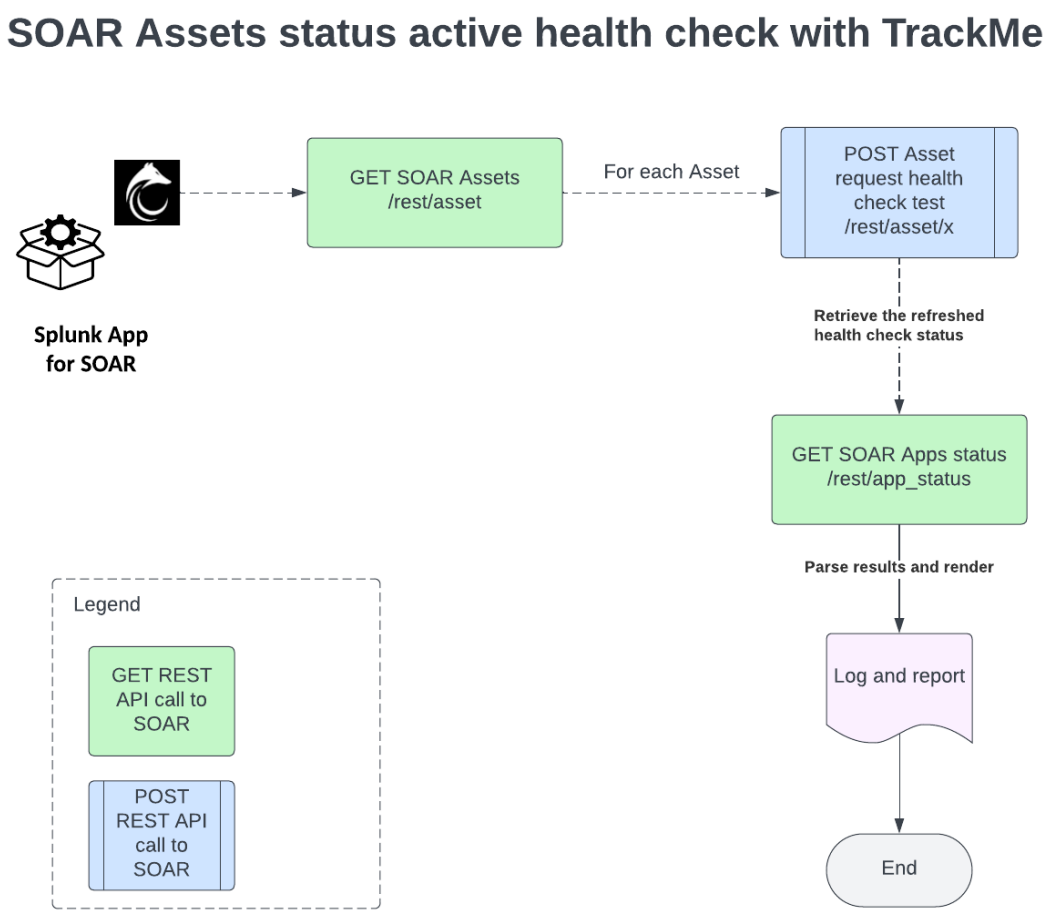
An asset that fails the status health check means that actions cannot be performed anymore, either ad-hocs actions requested by analysts or automated actions in the context of SOAR Playbooks.

Failures can happen for a variety of reasons, such as service accounts credential expiration, firewall rules changes or loss, API changes, SOAR Application upgrades, and many more.

This is an highly critical aspect of SOAR monitoring, especially true as Playbooks will only execute an action when needed, so the Application integration failure could remain undetected before it's too late.

SOAR has a builtin and mandatory connectivity feature per application, and per asset, however this health check is not continuously performed by SOAR, so only monitoring the `/rest/app_status` endpoint would result in monitoring outdated information.

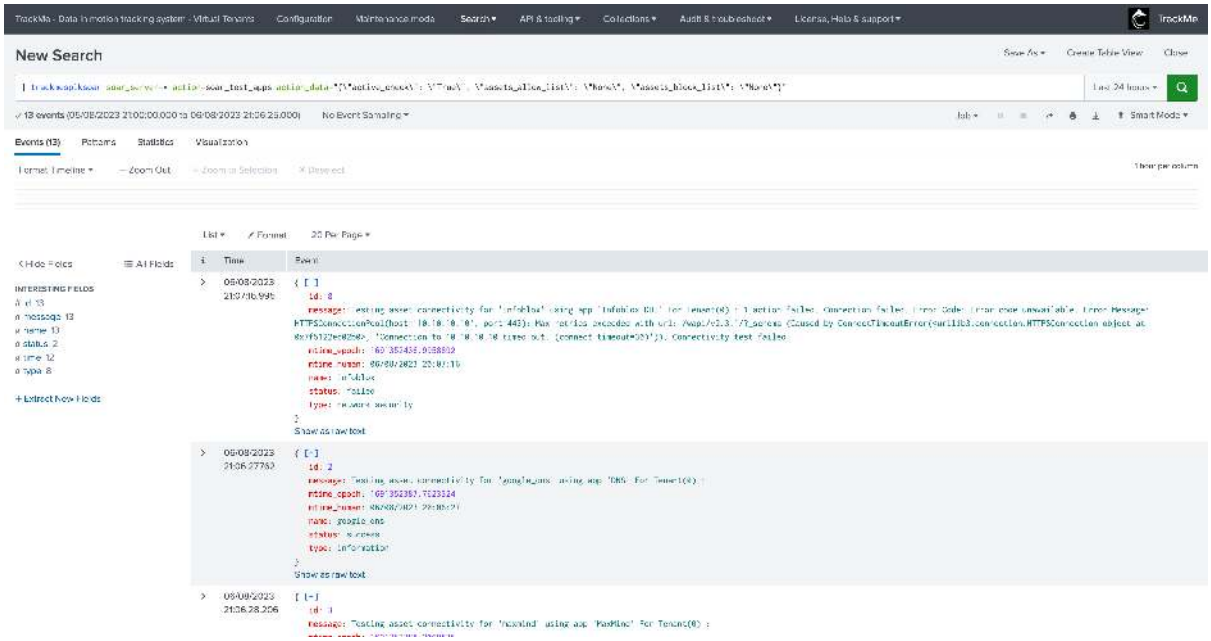
TrackMe handles this task by performing a multi-steps bi-directional integration with the SOAR REST API, summarised in the following diagram:



Advanced pre-built bi-directional interactions with Assets active check

The SOAR Assets active check in TrackMe relies on some multi-steps and bi-directional actions performed by TrackMe, the default usage of the command is the following:

```
| trackmesplksoar soar_server=* action=soar_test_apps action_data="{\"active_check\":␣
↪\"True\", \"assets_allow_list\": \"None\", \"assets_block_list\": \"None\"}"
```



In summary, the command does the following:

- Retrieve the list of assets eligible for a connection test
- For each asset, perform a POST call against the Asset API endpoint which requests an immediate application health check
- Record the response and store for later usage in the process
- Retrieve the Asset app_status endpoint result, parse and render the final results

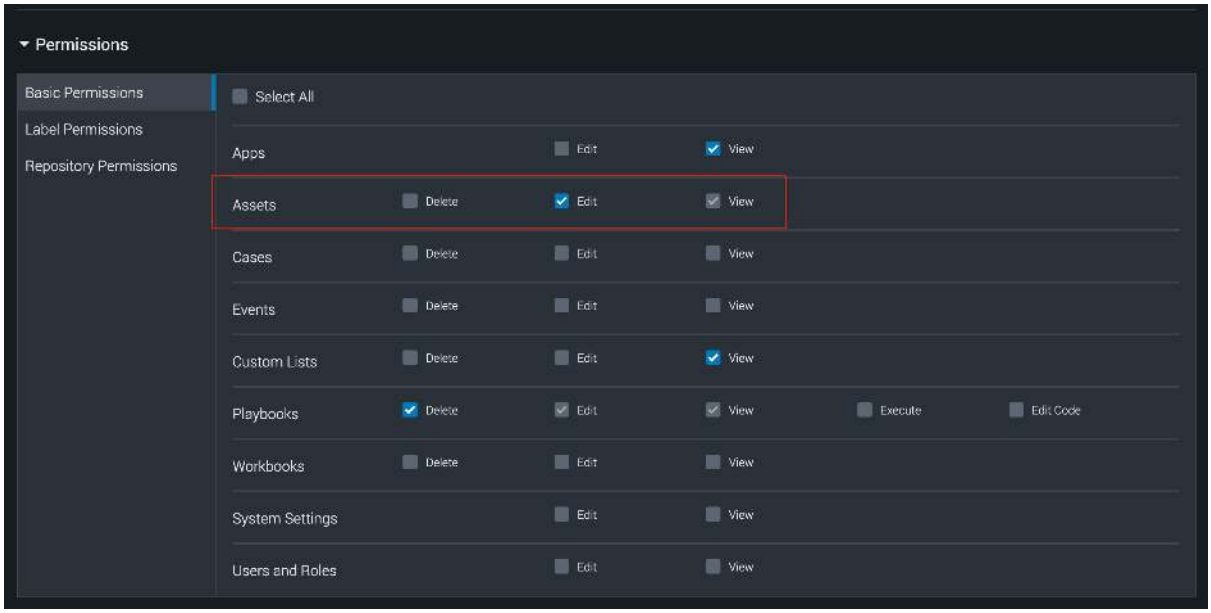
The command can accept several options:

Active checks

The default behaviour with `active_check: True` means TrackMe will actively perform the check bu running the POST call as described above:

If disabled with `active_check: False`, TrackMe does not perform the POST call and instead relies on SOAR doing it once per day, therefore it can take up to 24 hours before a failing application could be detected.

Note that `active_check: True` requires the SOAR automation user to have the edit Assets permissions:



Allow list and Block list:

You can mix the usage of asset_allow_list and asset_block_list to restrict and/or avoid some assets from being taken into account, for instance to avoid including the SMTP asset "internal_smtp":

```
| trackmesplksoar soar_server=* action=soar_test_apps action_data="{\"active_check\":␣
↪\"True\", \"assets_allow_list\": \"None\", \"assets_block_list\": \"internal_smtp\"}
↪"
```

9.12.5 5. SOAR Automation Brokers Monitoring, High Availability and Failover with TrackMe

Hint

Automation Broker Pools: Latest updates since TrackMe 2.1.8

- This feature was first introduced in TrackMe Version 2.0.50.
- Since TrackMe 2.1.8, these capabilities were enhanced so we can support independent pools of brokers that can be defined and managed in TrackMe.
- This allows you to have full flexibility, and configure multiple groups of brokers where Assets can be moved transparently depending on the brokers availability.

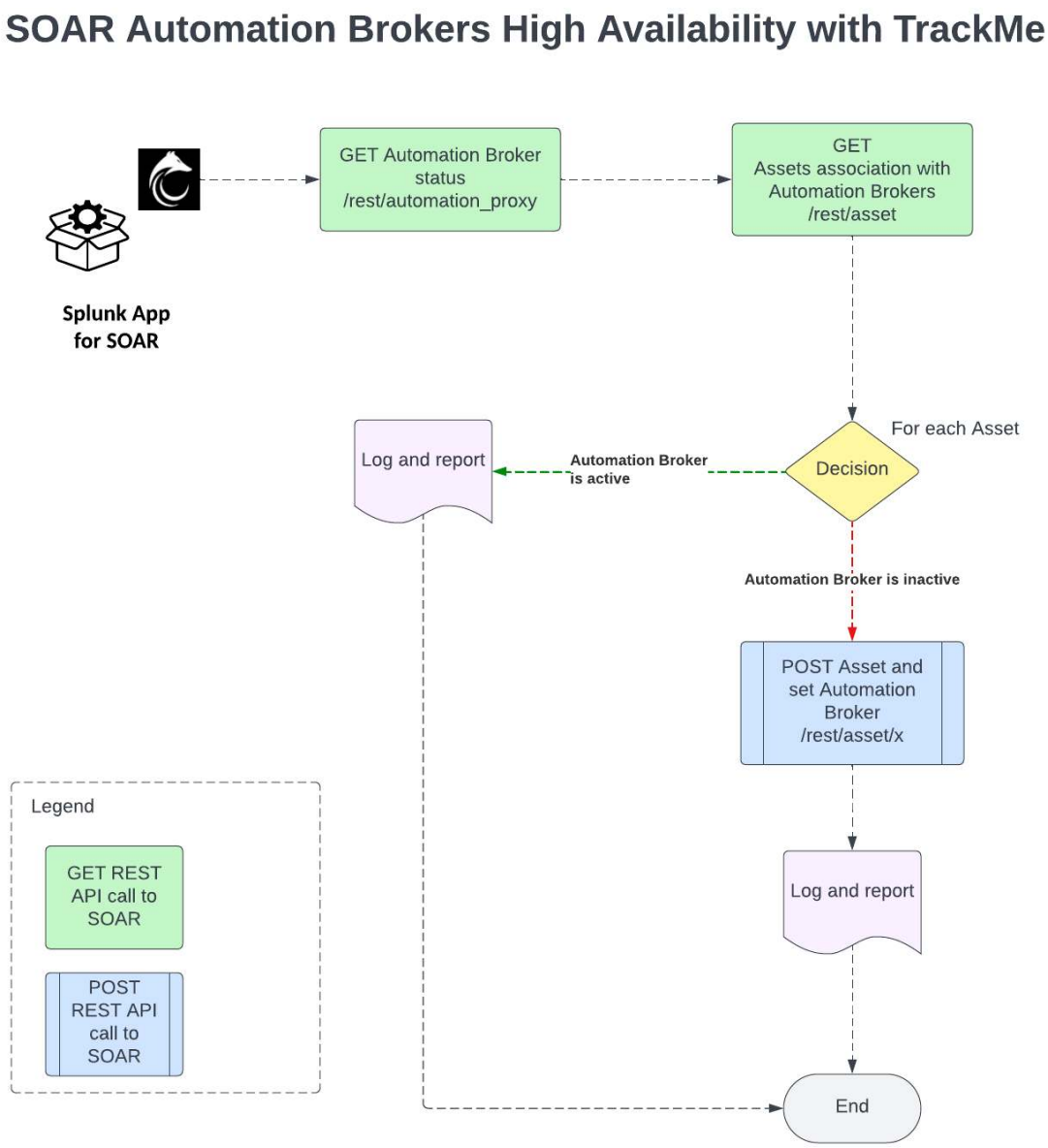
The builtin use case `splk_soar_manage_automation_brokers` allows managing SOAR Automation Brokers **High Availability** using TrackMe, based on the following workflow:

- Retrieve the list of Automation Brokers and their status from the SOAR API
- Retrieve the list of SOAR Assets and the association with Automation Brokers
- For each Automation Broker, if the broker status is inactive, perform an update of associated Assets to the next available and online Automation Broker

Automation broker status: the status is provided by the SOAR API and is based on the health check performed automatically by the SOAR Automation Broker.

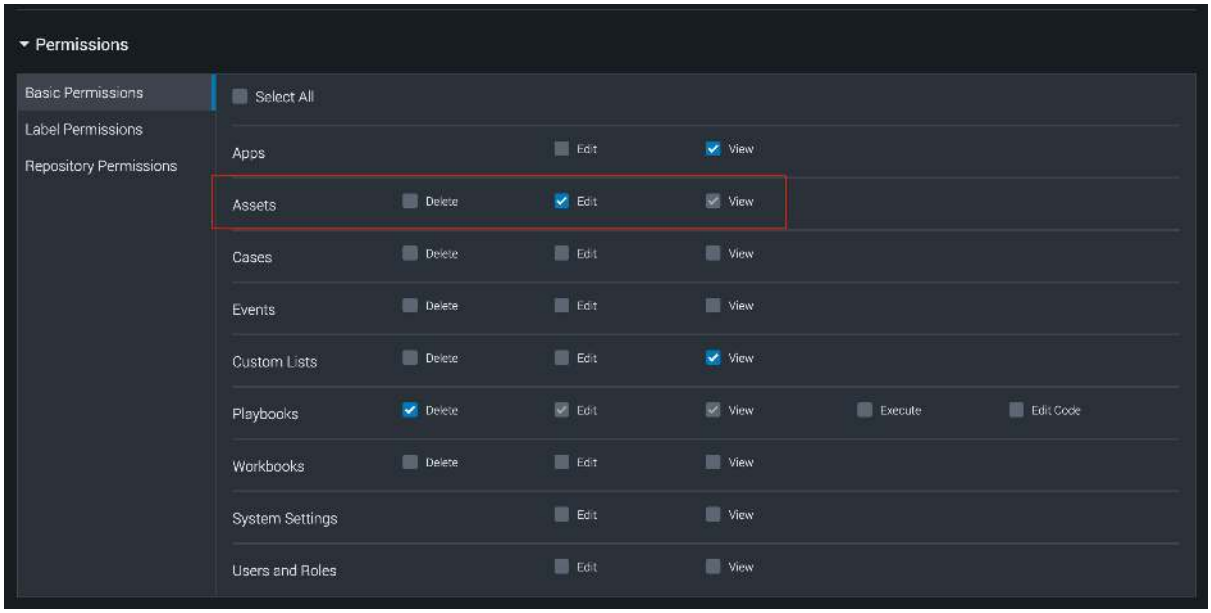
Additional options can be used to control and restrict the High Availability features, consult the examples below.

High level workflow diagram:



5.1 Requirements

The SOAR Automation User requires Assets management permissions:



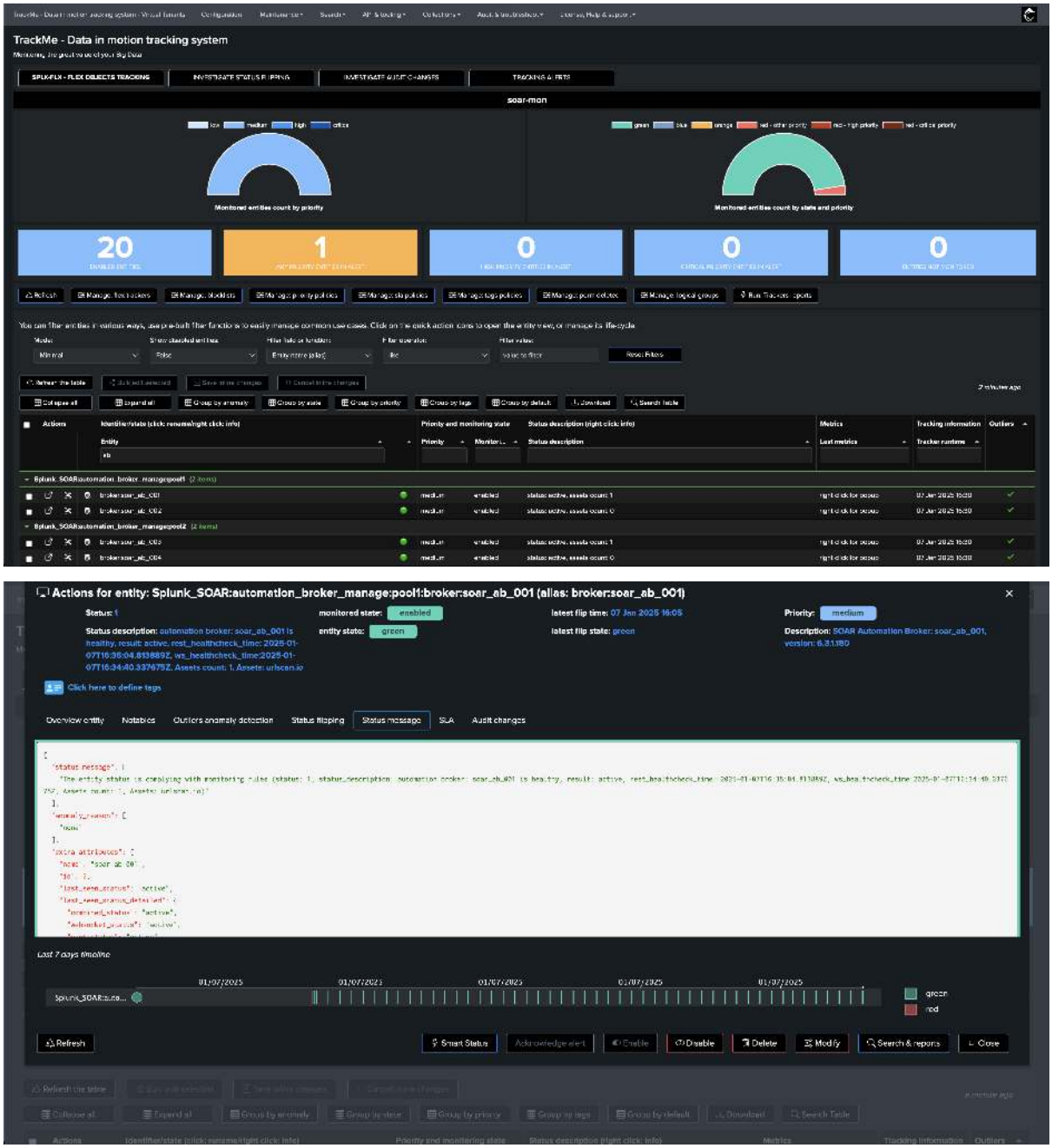
5.2 SOAR Automation Broker High Availability - failure detection and Assets updates

The tracking and management of SOAR Automation Brokers is achieved by running the following command through the builtin use case splk_soar_manage_automation_brokers:

```
| trackmesplksoar soar_server=* action=soar_automation_broker_manage
```

The command calls the TrackMe REST API endpoint `/services/trackme/v2/splk_soar/admin/soar_automation_broker_manage`, the following options are available:

argument	description
`soar_se`	The SOAR server account as defined in the Splunk App for SOAR, if unspecified or set to *, the first server in the Splunk application for SOAR configuration will be used
`mode`	Optional, the run mode, valid options are readonly | live. In read only mode, the Asset is not updated and only the message of the action to be performed is registered. In Live mode, assets are updated if the associated broker is not healthy, and if at least one active broker is available in the pool, defaults to live
`automat`	The comma separated list of Automation Brokers for the pool, if unspecified TrackMe will consider all registered SOAR Automation Brokers. When specified, the Flex Object tracker instance is tight to a certain list of Automation Brokers, and only these will be considered for the High Availability management. When creating multiple Trackers to manage different pools, you can update the group definition to clearly separate the resulting TrackMe entities, for easier management in the TrackMe UI.



Configuration example: Two pools of Automation Brokers

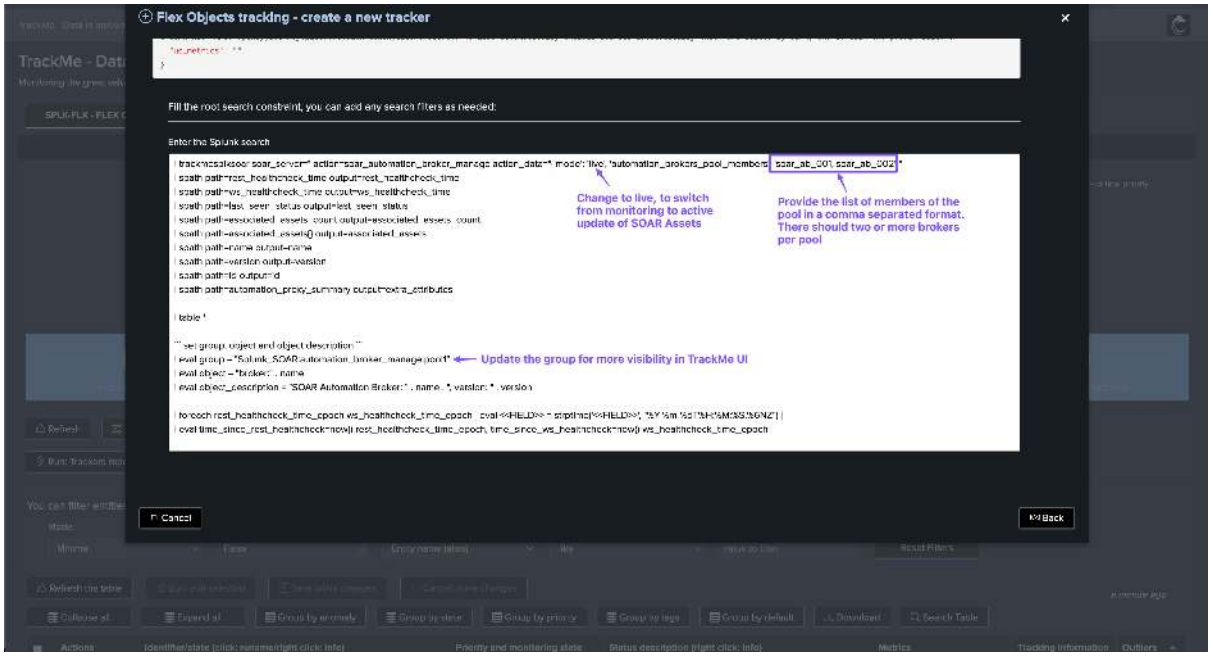
Consider the following Architecture scenario:

- Our SOAR environment has 4 Automation Brokers in total.
- Due to Network or any other technical constraints, certain assets can only use specific Automation Brokers.
- Therefore, we have 2 groups of Automation Brokers, in each of these groups, Assets cab be configured to use either broker from that pool.

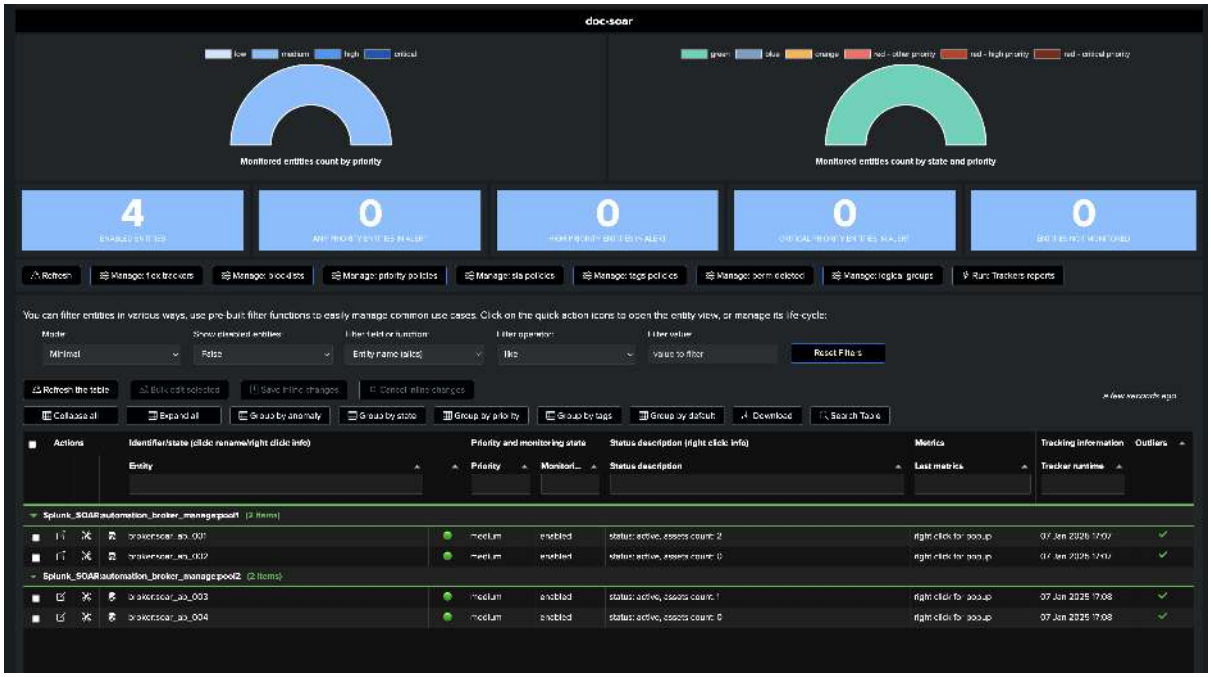
Automation Broker	Pool
soar_ab_001	Pool1
soar_ab_002	Pool1
soar_ab_003	Pool2
soar_ab_004	Pool2

Create two Flex Object trackers to manage each pool:

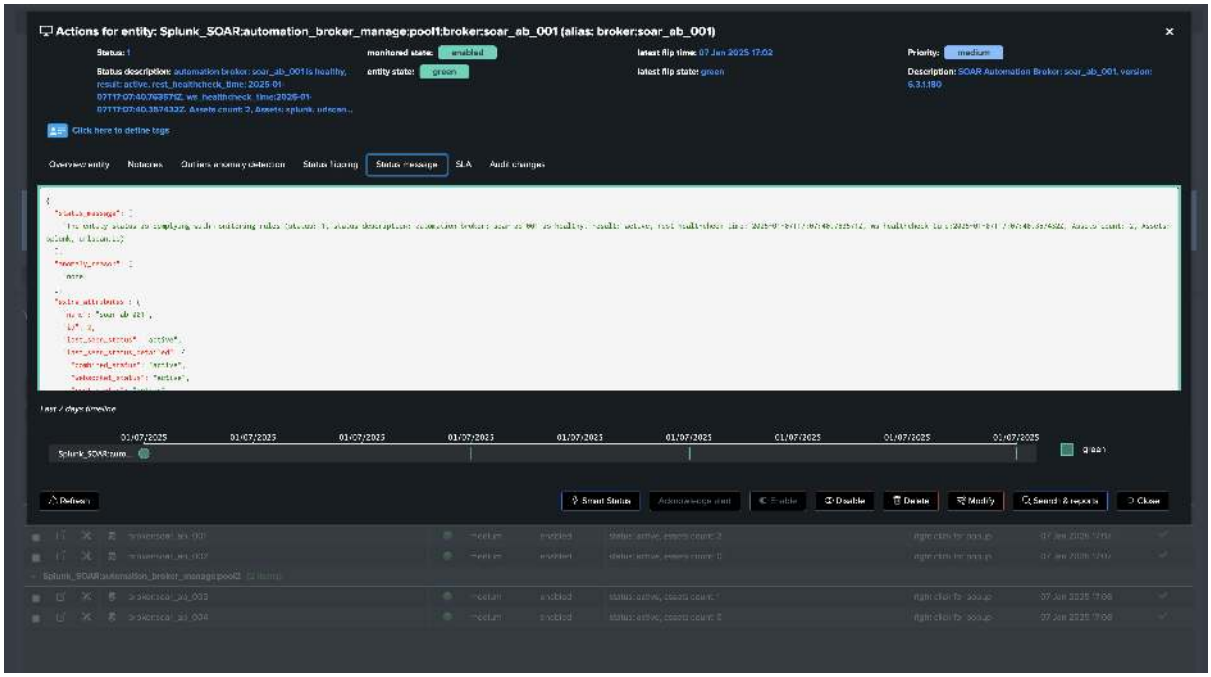
Create the first Tracker as following, and repeat for the second pool:

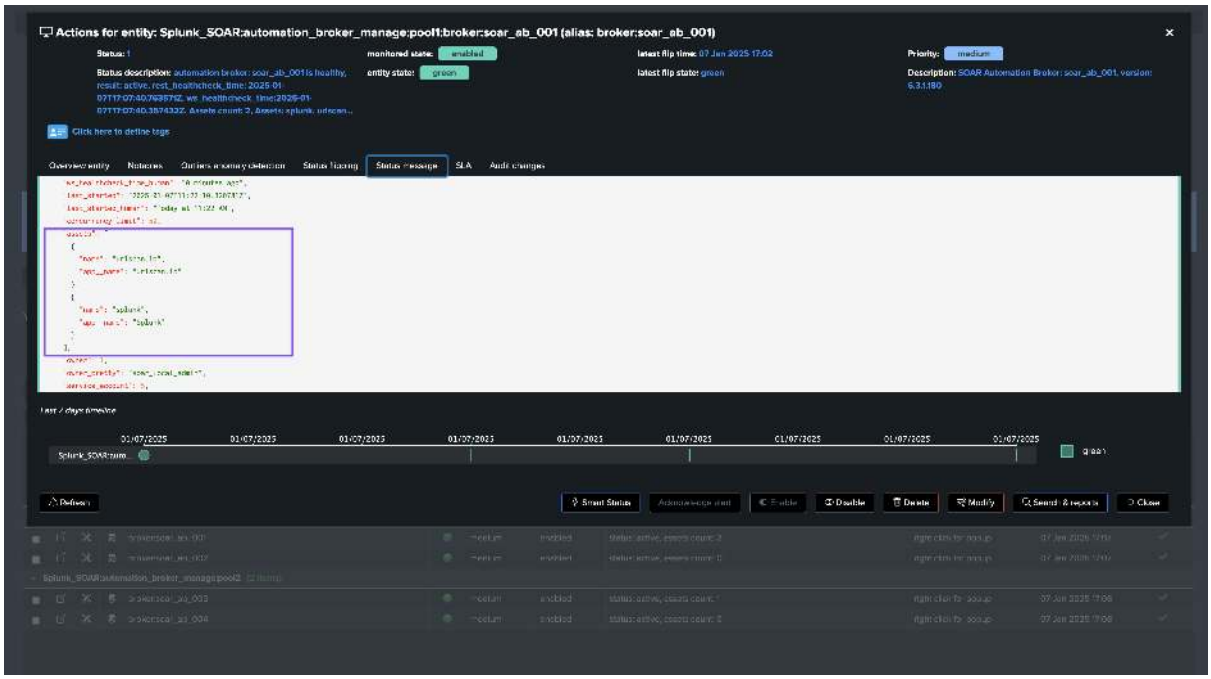


Once the trackers have been executed at least once, our setup is ready and we have 4 entities representing the 2 pools of brokers:



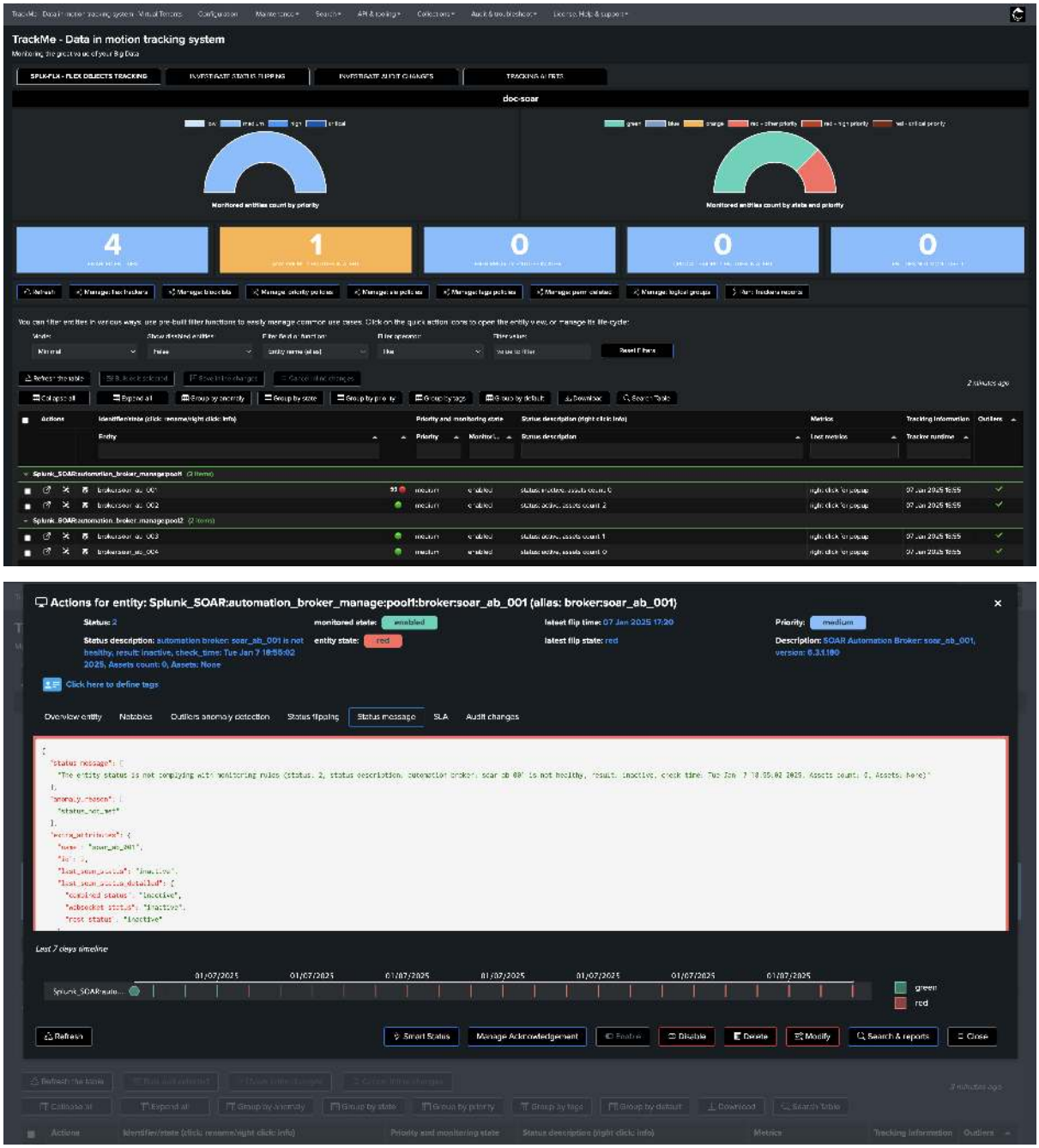
The status message shows the extended attributes, which contains all Metadata from the brokers, and various information such as the list of assets currently linked to this asset:





What happens when an Automation Broker is detected as inactive?

If an issue affects an Automation Broker, TrackMe turns the entity in red state, and will automatically update Assets so these are associated with any randomly chosen active broker from the same pool:

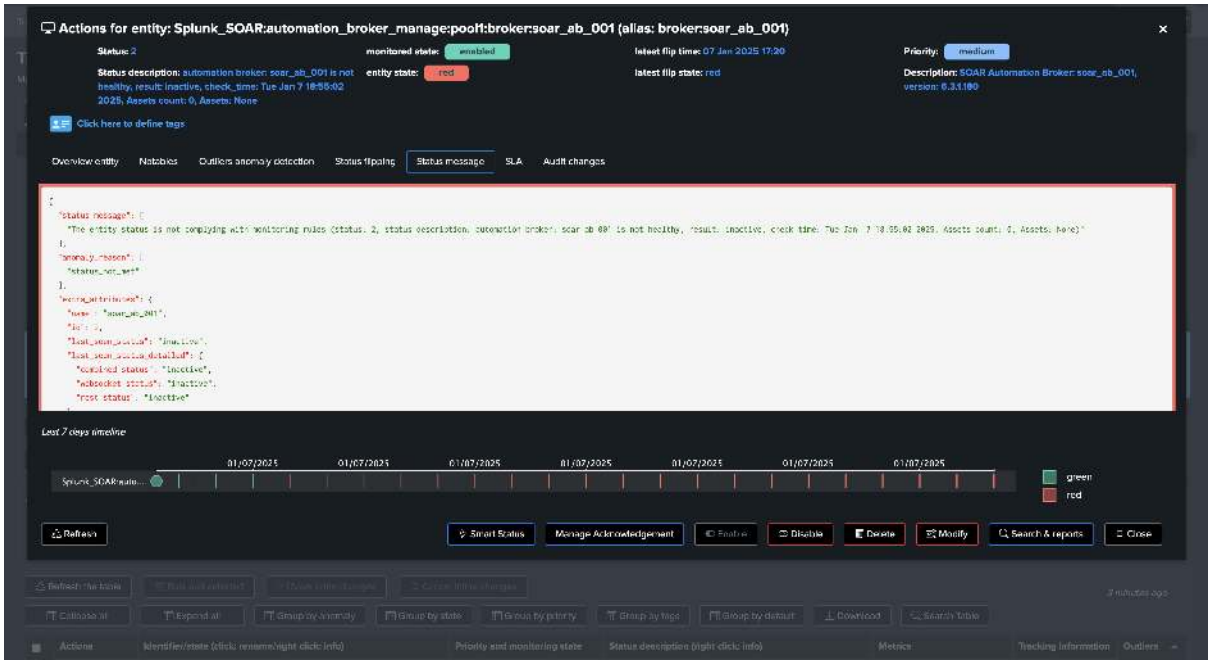


Logs can be found at:

```
index=_internal sourcetype=trackme:rest_api post_soar_automation_broker_manage
```

When updates of SOAR Assets are performed, the following search can be used to review the logs and executed actions:

```
index=_internal sourcetype=trackme:rest_api post_soar_automation_broker_manage updated
```



A typical trace for an Asset update will look the following example:

A first warning message shows that the Automation Broker associated with the Asset is currently offline, followed by the update result message from the action performed by TrackMe.

```
2025-01-07 17:20:06,002 WARNING trackme_rest_handler_splk_soar_admin.py post_soar_
↪automation_broker_manage 1081 asset=urlscan.io, id=3 is associated with automation_
↪broker=soar_ab_001, id=2, status=inactive, asset will be updated now!
2025-01-07 17:20:06,104 INFO trackme_rest_handler_splk_soar_admin.py post_soar_
↪automation_broker_manage 1138 asset=urlscan.io, id=3, asset automation broker␣
↪configuration was successfully updated from automation_broker=soar_ab_001, id=2,␣
↪status=inactive to automation_broker=soar_ab_002, id=3, status=active, response="{
↪'success': True, 'id': 3}"
```

What happens when the Automation Broker is back online?

When the Automation Broker is back online, the entity will turn back to green. However, note that the Asset definition will not be updated back to the original Automation Broker, this is a design choice to avoid a ping-pong effect, and to ensure that the Asset is always associated with an active Automation Broker.

9.12.6 6. Interacting with the SOAR API with GET and POST calls in pure SPL

The following sections show the root searches and technical details of each TrackMe's SOAR use cases.

Refer to the SOAR API reference documentation as needed:

- https://docs.splunk.com/Documentation/SOAR/current/PlatformAPI/Using

6.1 Running GET calls to SOAR API

trackmesplksoar:

TrackMe comes with a custom command (generating custom command) called `trackmesplksoar`, this SPL command does the interface with the Splunk App for SOAR and the SOAR API itself, usage:

```
| trackmesplksoar soar_server=<soar_server> action=<action> action_data=<json action␣
↪data>
```

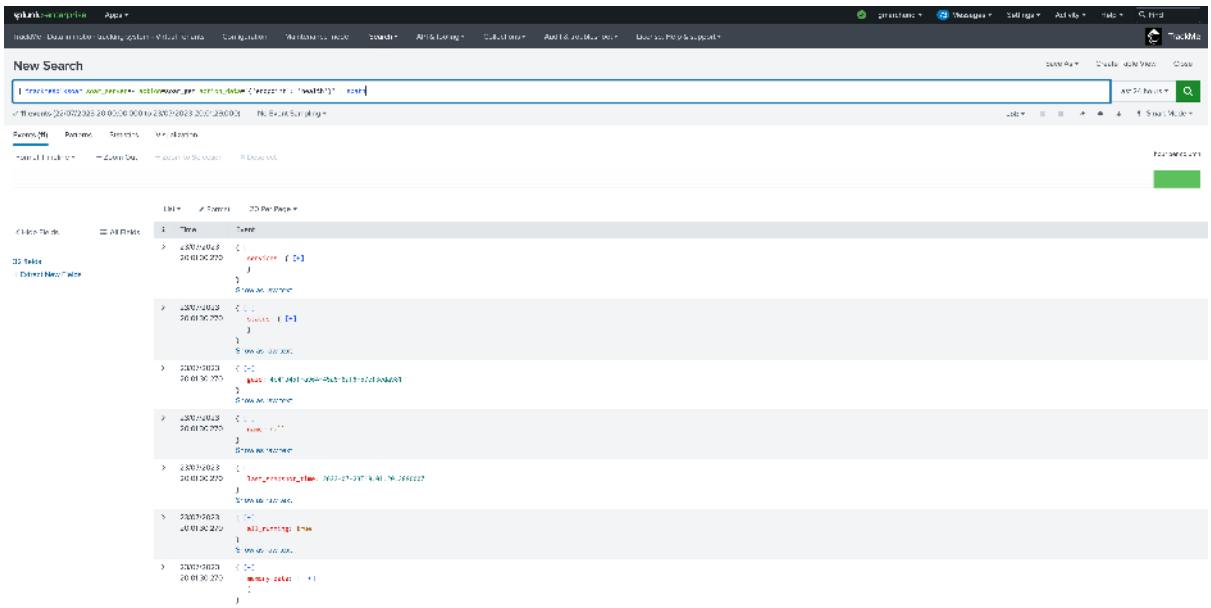
The command handles different options:

Syntax:

- soar_server: the name of the SOAR server as configured in the Splunk App for SOAR
- action: an action in the following support list: `soar_get|soar_post|soar_test_apps|soar_health_status|soar_`
- action_data: a JSON formatted object, either used by specific actions or used to perform a POST query to a SOAR endpoint

Example: the following command allows retrieving the health information from the SOAR API, it targets the endpoint rest/health_status:

```
| trackmesplksoar soar_server=* action=soar_get action_data="{'endpoint': 'health'}"␣
↪| spath
```



This action allows calling any SOAR API endpoint performing a GET call only.

Another example, to retrieve the configuration of all SOAR Assets, you would:

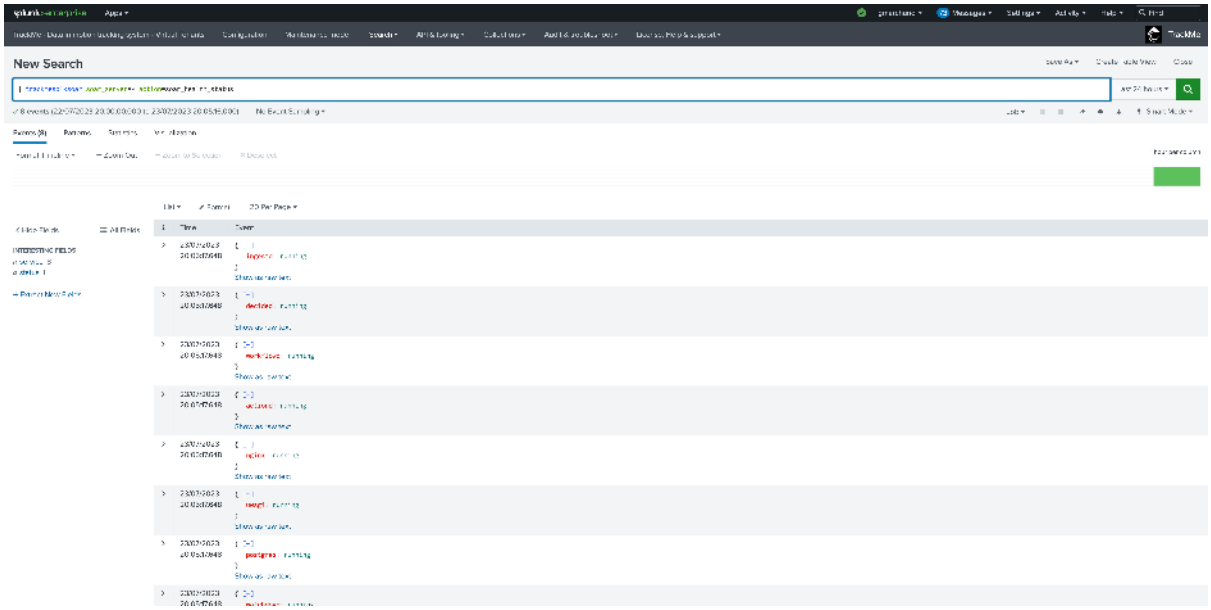
```
| trackmesplksoar soar_server=* action=soar_get action_data="{'endpoint': 'asset'}" |␣
↪spath
```

6.2 Pre-built actions and advanced parsing

In some cases, such as the one above, it may not be very straightforward to parse and use the API endpoints results, for instance when there are nested JSON structures as part of the response.

For instance, for the ease of the integration, the custom command provides builtin use cases which will parse the JSON results as needed and render the results properly, for instance the following relies on the health endpoint and extracts the status of SOAR services:

```
| trackmesplksoar soar_server=* action="soar_health_status"
```



6.3 Performing a POST call to a SOAR API endpoint via TrackMe

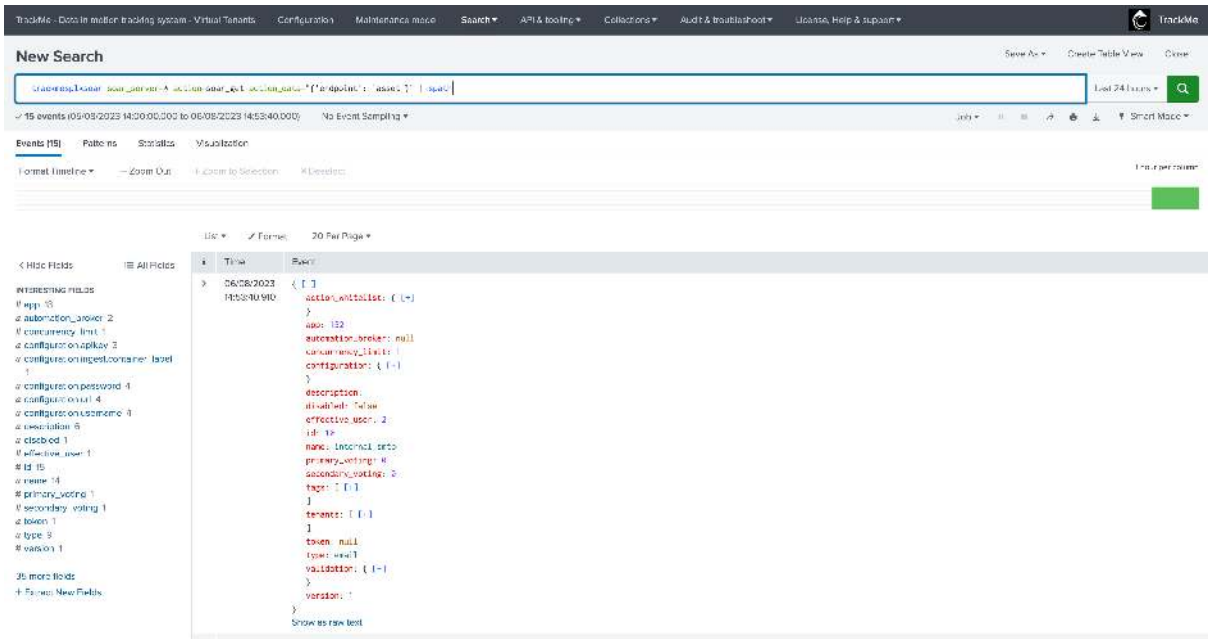
In fact, you can perform a POST to a SOAR API endpoint using TrackMe's integration, which the Splunk App for SOAR does not provide.

For instance, to run a POST against an Asset and request an immediate health check, you would first retrieve the list of assets to identify the ID of the Asset:

See:

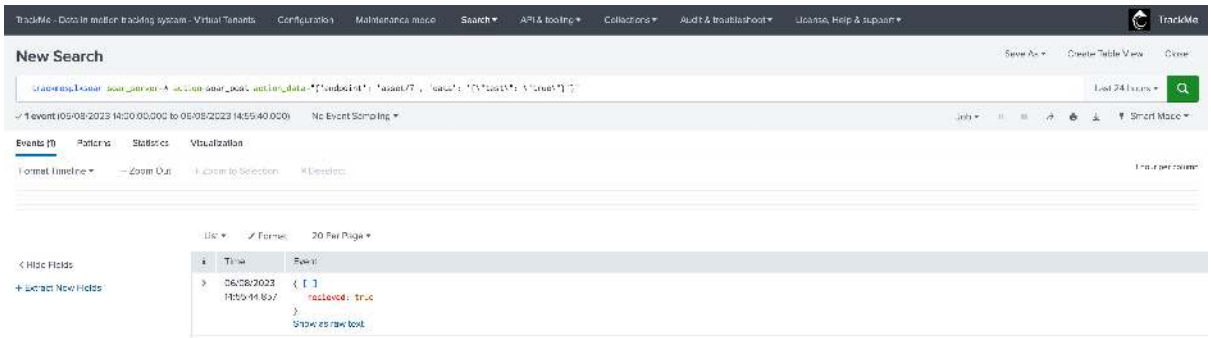
- https://docs.splunk.com/Documentation/SOAR/current/PlatformAPI/RESTInfo
- https://docs.splunk.com/Documentation/SOAR/current/PlatformAPI/RESTAssets

```
| trackmesplksoar soar_server=* action=soar_get action_data="{'endpoint': 'asset'}" |␣
↪spath
```



Then, you would:

```
| trackmesplksoar soar_server=* action=soar_post action_data="{'endpoint': 'asset/7',
↪'data': '{\"test\": \"true\"}'}"
```



Note: "received: true" is the actual response from SOAR, this is not a typo from TrackMe but a typo in the SOAR API currently for this endpoint!

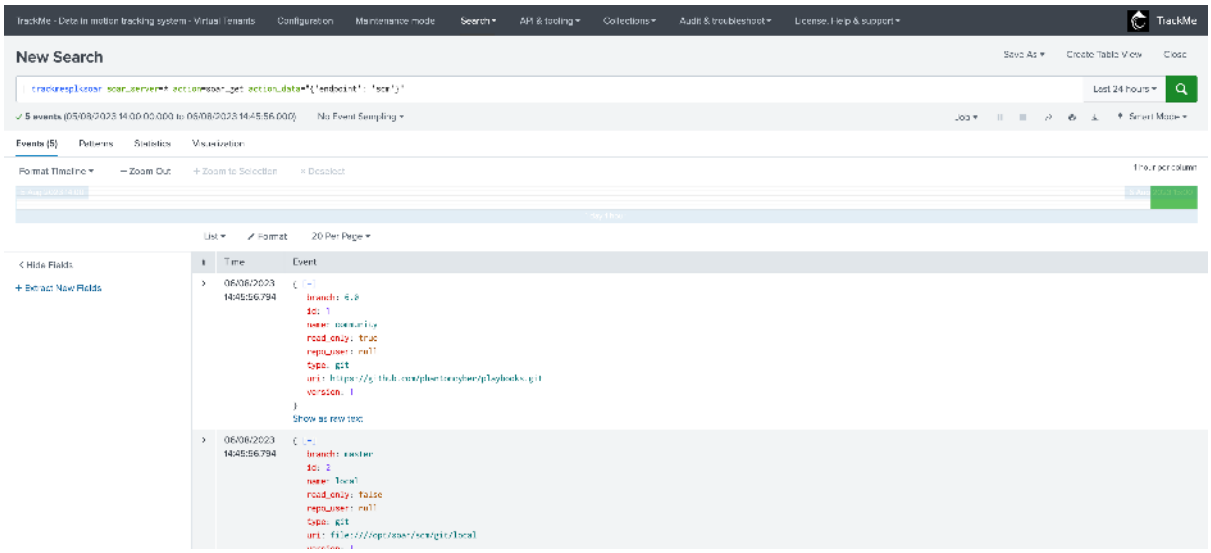
Another example, let's request the sync refresh of a Git Repository via the SCM endpoint:

See:

- https://docs.splunk.com/Documentation/SOAR/current/PlatformAPI/RESTScm

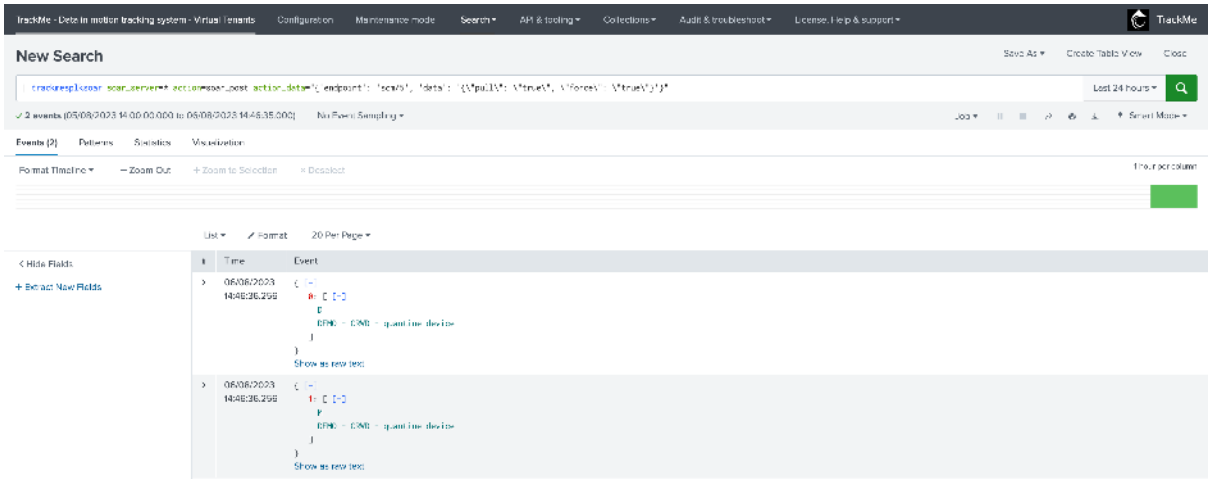
First, we run a GET call to get the ID of the SCM Git repository configuration:

```
| trackmesplksoar soar_server=* action=soar_get action_data="{'endpoint': 'scm'}"
```

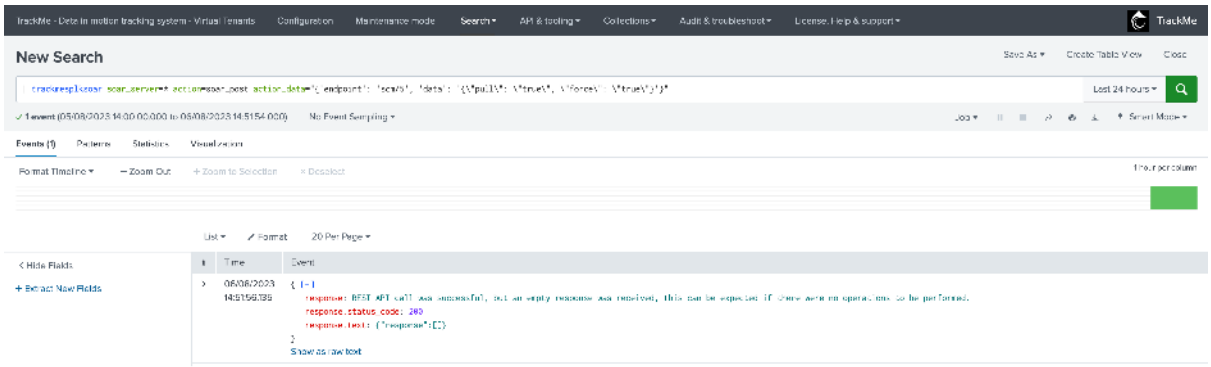


We run a POST call again to request the SCM update, we have a few changes that were actually merged to the branch:

```
| trackmesplksoar soar_server=* action=soar_post action_data="{'endpoint': 'scm/5',
↪'data': '{\"pull\": \"true\", \"force\": \"true\"}'}"
```



If we had no operations pending, we would have got the following message returned:



6.4 Performing live REST SOAR lookups in TrackMe using the command trackmesplksoarlookup

TrackMe provides a new command called `trackmesplksoarlookup` which allows to perform live REST lookups to the SOAR API, and to retrieve the metadata associated with the SOAR objects such as the playbooks, assets, applications and brokers associated with the detection.

Usage of the command:

```
[trackmesplksoarlookup-command]
syntax = | trackmesplksoarlookup soar_server=<soar_server> endpoint_target=<endpoint_
↪target> source_field=<source_field> dest_field_name=<dest_field_name> dest_field_
↪definition=<dest_field_definition> definition_filter_fields=<A comma separated list␣
↪of fields to retrieve from the definition>
description = \
   This streaming command can be used to interact with the SOAR API in a lookup way,␣
↪so that from an id of an object, its definition can be retrieved easily in native␣
↪SPL \
   Syntax: \
   - soar_server: the name of the SOAR server as configured in the Splunk App for␣
↪SOAR, \
   - endpoint_target: the endpoint target for the object to lookup\
```

(continues on next page)

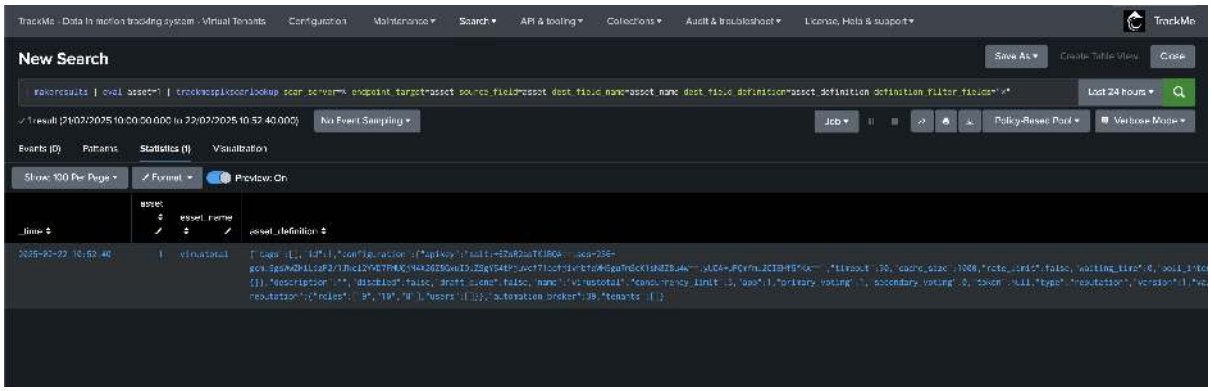
(continued from previous page)

```
   - source_field: the name of the field containing the object id, \
   - dest_field_name: the name of the field to store the logical name of the␣
↪corresponding object retrieved from this id (if any!), \
   - dest_field_definition: the name of the field to store the definition of the␣
↪corresponding object retrieved from this id (if any!) \
   - definition_filter_fields: a comma separated list of fields to retrieve from the␣
↪definition \
   | trackmesplksoarlookup soar_server=<soar_server> endpoint_target=<endpoint_target>
↪ source_field=<source_field> dest_field_name=<dest_field_name> dest_field_
↪definition=<dest_field_definition> definition_filter_fields=<A comma separated list␣
↪of fields to retrieve from the definition>
comment1 = \
   Lookup the definition of a SOAR object from its id in a streaming manner
example1 = \
   | makeresults | eval asset=1 | trackmesplksoarlookup soar_server=* endpoint_
↪target=asset source_field=asset dest_field_name=asset_name dest_field_
↪definition=asset_definition definition_filter_fields="name,description"
shortdesc = Streaming command for TrackMe's Splunk SOAR integration
```

Example:

You can for instance use the lookup command to retrieve the definition any SOAR object in pure SPL:

```
| makeresults | eval asset=1 | trackmesplksoarlookup soar_server=* endpoint_
↪target=asset source_field=asset dest_field_name=asset_name dest_field_
↪definition=asset_definition definition_filter_fields="*"
```



A full usage example can be to track SOAR actions and call the command to lookup associated objects metadata using their IDs:

```
index=phantom_action_run

``` extract target from the json ```
| rex field=_raw "targets\":\s(?<target_json>\[.*\])\,\s\"tenant_id\":\s"

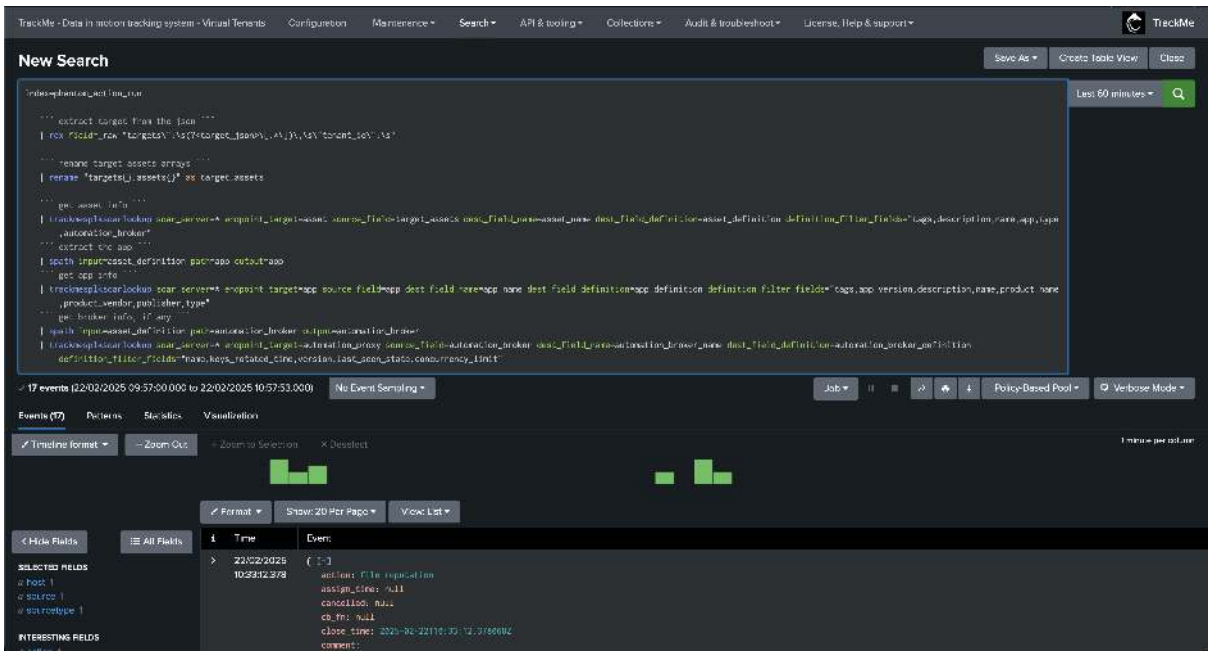
``` rename target assets arrays ```
| rename "targets{}.assets{}" as target_assets

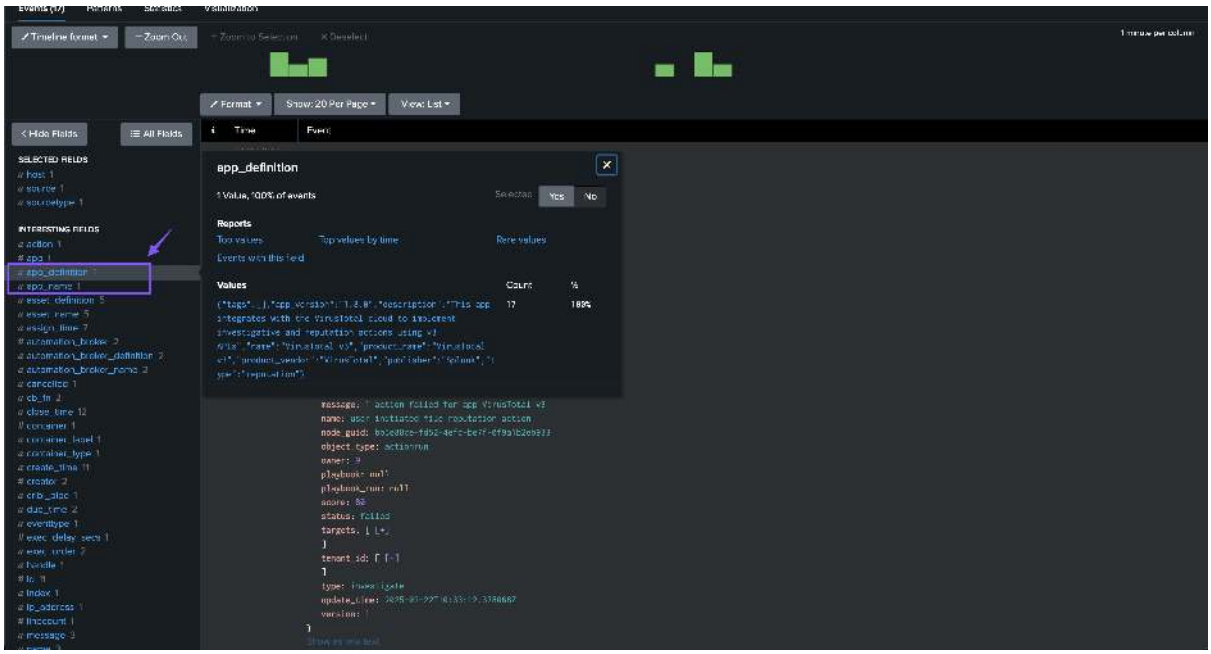
``` get playbook info ```
| trackmesplksoarlookup soar_server=* endpoint_target=playbook source_field=playbook␣
↪dest_field_name=playbook_name dest_field_definition=playbook_definition definition_
↪filter_fields="tags,active,category,comment,labels,name,metadata,playbook_type"
``` get asset info ```
| trackmesplksoarlookup soar_server=* endpoint_target=asset source_field=target_
```

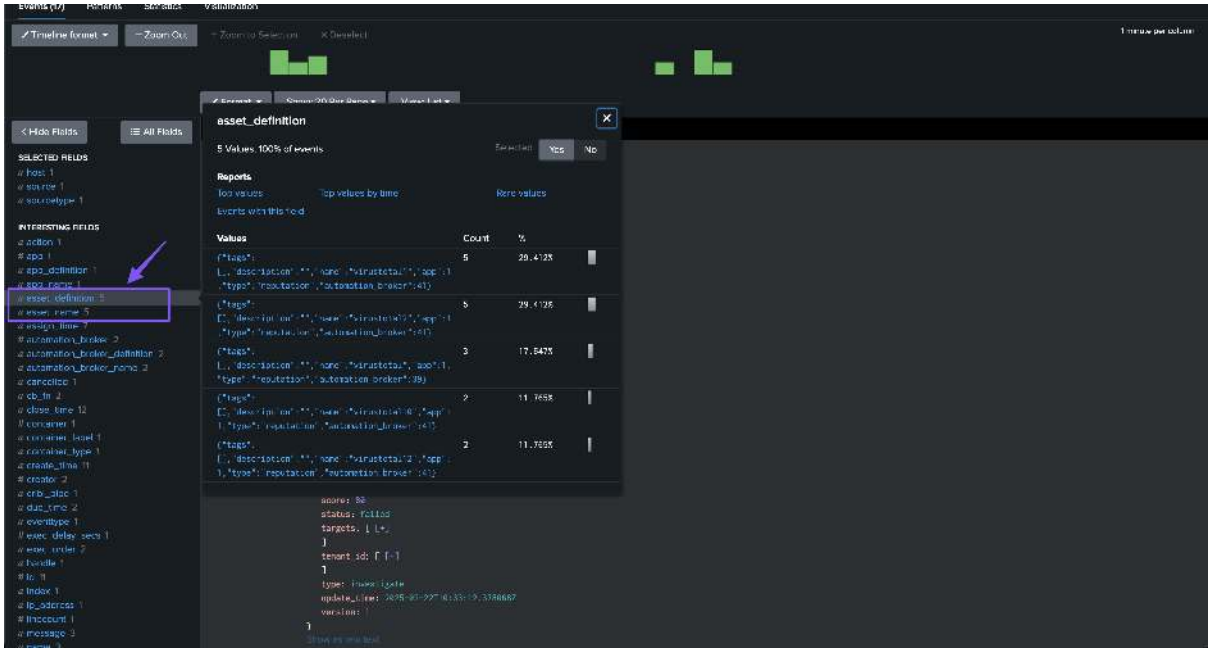
(continues on next page)

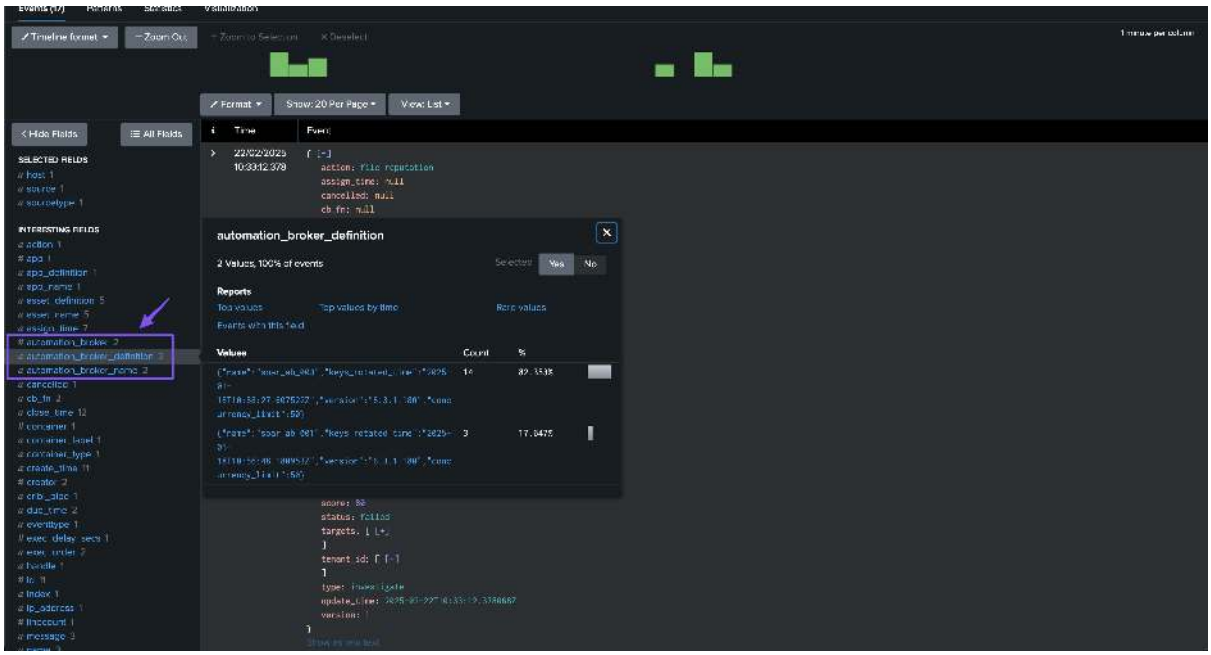
(continued from previous page)

```
↪assets dest_field_name=asset_name dest_field_definition=asset_definition definition_
↪filter_fields="tags,description,name,app,type,automation_broker"
``` extract the app ```
| spath input=asset_definition path=app output=app
``` get app info ```
| trackmesplksoarlookup soar_server=* endpoint_target=app source_field=app dest_field_
↪name=app_name dest_field_definition=app_definition definition_filter_fields="tags,
↪app_version,description,name,product_name,product_vendor,publisher,type"
``` get broker info, if any ```
| spath input=asset_definition path=automation_broker output=automation_broker
| trackmesplksoarlookup soar_server=* endpoint_target=automation_proxy source_
↪field=automation_broker dest_field_name=automation_broker_name dest_field_
↪definition=automation_broker_definition definition_filter_fields="name,keys_rotated_
↪time,version,last_seen_state,concurrency_limit"
```



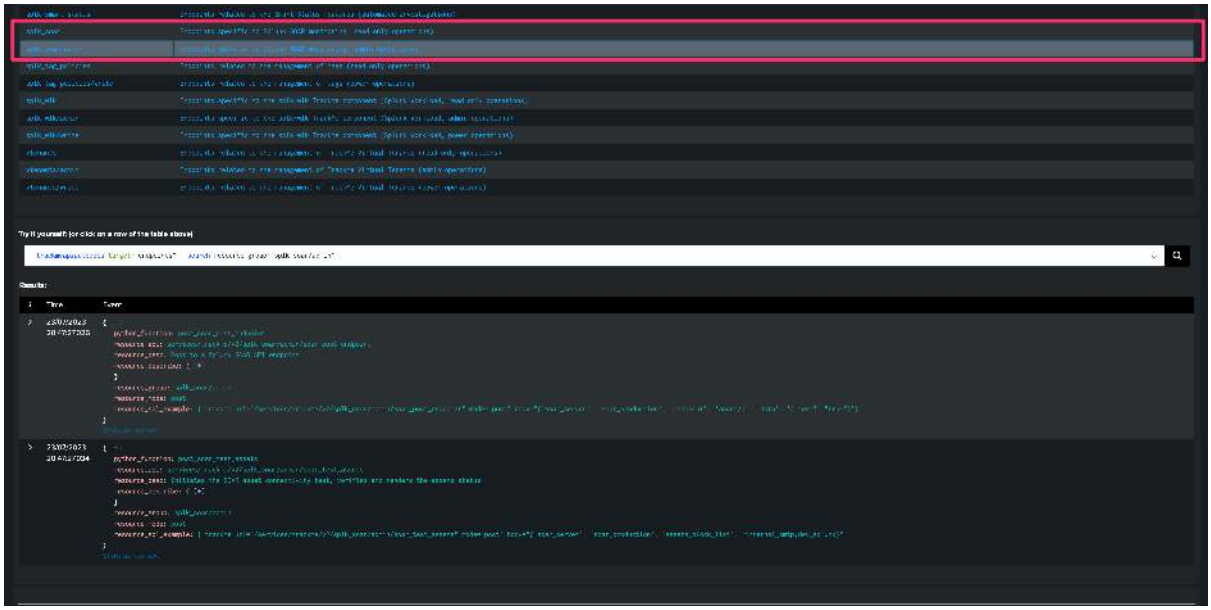

```





6.5 TrackMe's REST API splk-soar endpoints

Underneath, the `trackmesplksoar` command interacts with the TrackMe API endpoints for SOAR, you can find the endpoints references and their documentation in the API reference dashboard: (Menu API & Tooling)



9.13 Cribl Logstream monitoring in TrackMe

Hint

Version 2.0.45 and later

- The SOAR integration requires TrackMe version 2.0.45 and later

9.13.1 1. Introduction to Cribl Logstream monitoring

TrackMe provides builtin use cases to efficiently monitor at scale one or more Cribl Logstream environments.

Cribl Logstream monitoring is performed via the TrackMe Flex Object component (splk-flx) which is a restricted component not available with the Free community edition of TrackMe.

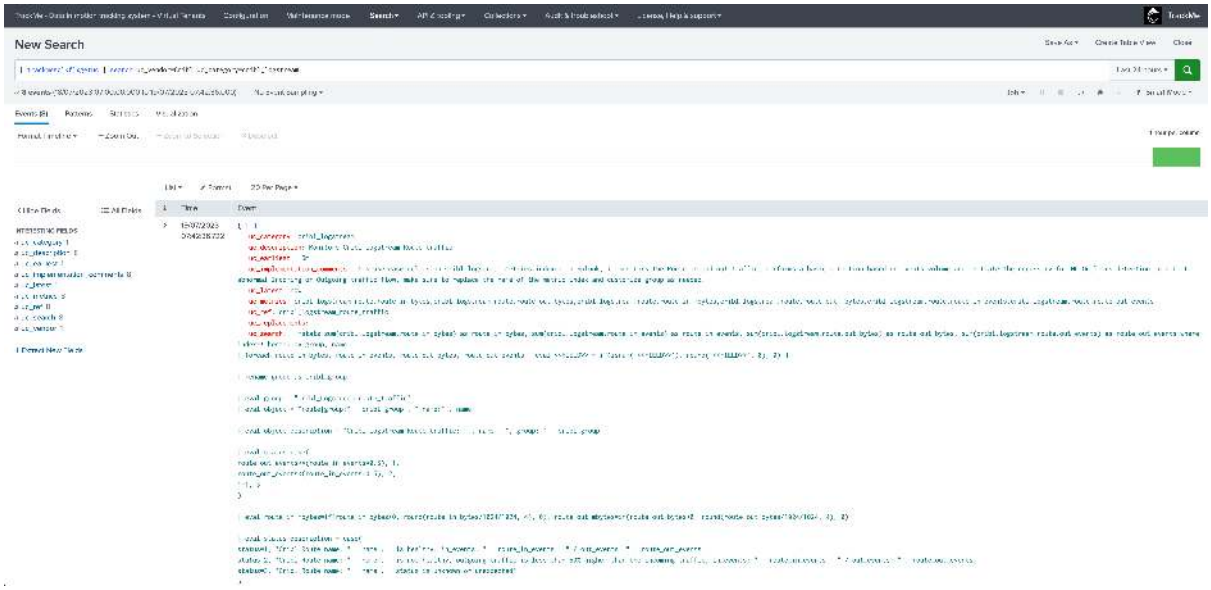
The monitoring relies on Cribl internal metrics sent to Splunk associated with TrackMe's Flex Object concepts, and provides the following use cases:

uc_ref	uc_description	uc_metrics
cribl_logstr	Monitors Cribl Logstream health inputs status	cribl_logstream.health.health_inputs
cribl_logstr	Monitors Cribl Logstream health outputs status	cribl_logstream.health.health_outputs
cribl_logstr	Monitors Cribl Logstream hosts CPU usage and triggers on high usage thresholds	cribl_logstream.avg_cpu_usage
cribl_logstr	Monitors Cribl Logstream destination output blocked and under backpressure statuses	cribl_logstream.output.blocked_outputs, cribl_logstream.output.backpressure_outputs
cribl_logstr	Monitors Cribl Logstream Pipelines	cribl_logstream.pipeline.in_events, cribl_logstream.pipeline.out_events, cribl_logstream.pipeline.dropped_events, cribl_logstream.pipeline.pct_sent_events, cribl_logstream.pipeline.pct_dropped_events
cribl_logstr	Monitors Cribl Logstream Route traffic	cribl_logstream.route.route_in_bytes, cribl_logstream.route.route_out_bytes, cribl_logstream.route.route_in_mbytes, cribl_logstream.route.route_out_mbytes, cribl_logstream.route.route_in_events, cribl_logstream.route.route_out_events
cribl_logstr	Monitors Cribl Logstream total input traffic	cribl_logstream.total.total_in_bytes, cribl_logstream.total.total_in_events, cribl_logstream.total.total_in_mbytes
cribl_logstr	Monitors Cribl Logstream total output traffic	cribl_logstream.total.total_out_bytes, cribl_logstream.total.total_out_events, cribl_logstream.total.total_out_mbytes

These use cases are provided via the Flex Object use cases library, but note that you can also manually implement new use cases, or customise builtin use cases as needed.

You can review the use case details ahead of their creation in TrackMe with the following command:

```
| trackmesplkflxgetuc | search uc_vendor=Cribl uc_category=cribl_logstream
```



9.13.2 2. Requirements for Cribl Logstream monitoring

Cribl Internal metrics

TrackMe for Cribl Logstream monitoring relies on Cribl internal metrics indexed in Splunk.

The pre-built use cases searches search in all metric indexes by default using `where index=*`, for optimisation purposes you can update the search during the creation of the Flex trackers, example:

```
| mstats sum(cribl.logstream.route.in_bytes) as route_in_bytes, sum(cribl.logstream.
↪route.in_events) as route_in_events, sum(cribl.logstream.route.out_bytes) as route_
↪out_bytes, sum(cribl.logstream.route.out_events) as route_out_events where index=*␣
↪host=* by group, name
```

For more information to Cribl Logstream metrics in Splunk:

- https://docs.cribl.io/stream/internal-metrics.

Hint

Multiple worker groups

- If your Cribl Logstream deployment is composed by multiple worker groups, there is nothing to do and you can have a single tracker managing all worker groups individually
- All searches break against the Cribl Logstream group dimension, which information is also used to create and maintain the entity

TrackMe tenant with the Flex Object component

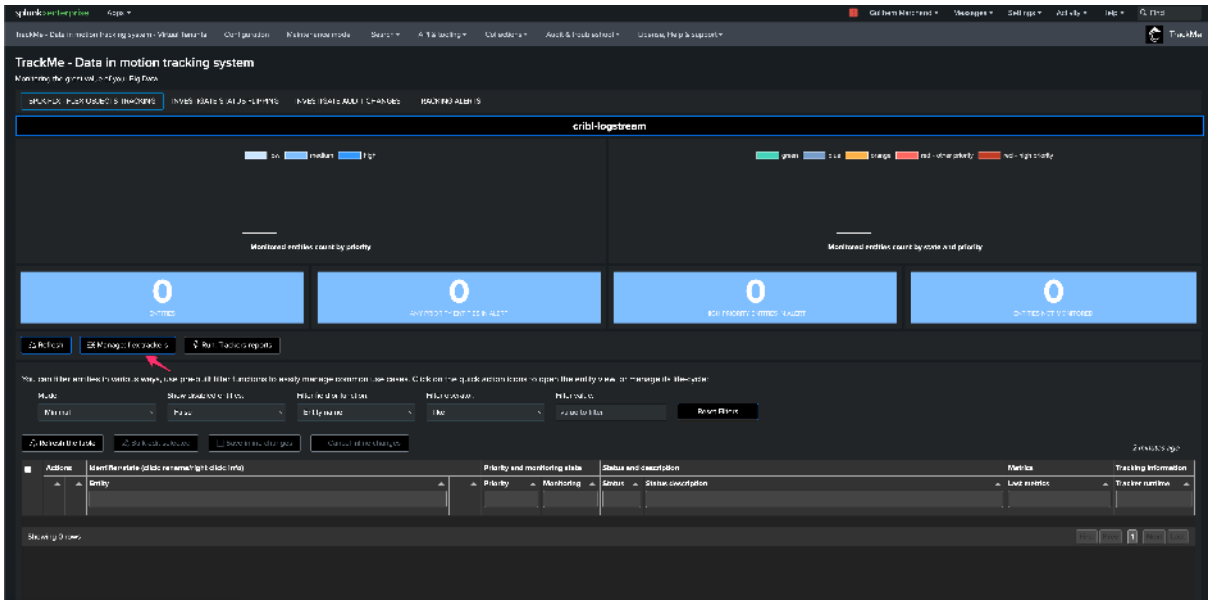
You need a TrackMe tenant with the Flex Object component enabled, you can decide to create a dedicated tenant for the monitoring of Cribl Logstream, and use any existing tenant of your choice.

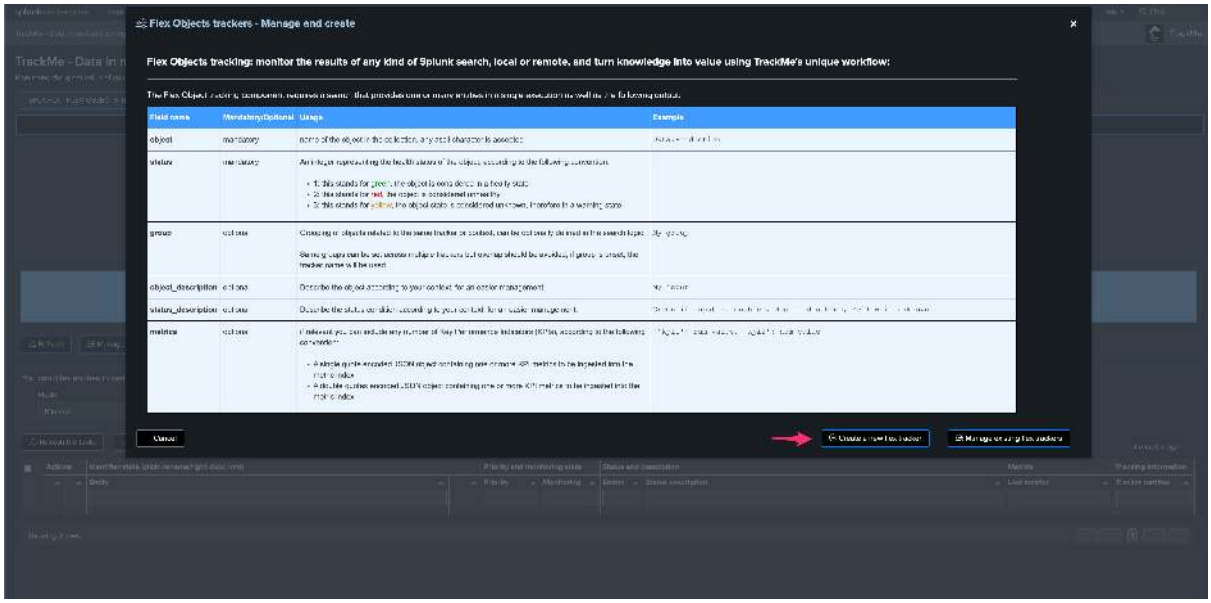
Once the Flex trackers have been created, TrackMe automatically groups the resulting entities for Cribl Logstream into the following groups:

uc_ref	grouping
cribl_logstream_health_inputs	Cribl_Logstream:health
cribl_logstream_health_outputs	Cribl_Logstream:health
cribl_logstream_hosts_cpu_usage	Cribl_Logstream:infrastructure
cribl_logstream_output_destination_pressure	Cribl_Logstream:Destination
cribl_logstream_pipeline	Cribl_Logstream:pipeline_traffic
cribl_logstream_route_traffic	Cribl_Logstream:route_traffic
cribl_logstream_total_traffic_inputs	Cribl_Logstream:traffic_in_total
cribl_logstream_total_traffic_outputs	Cribl_Logstream:traffic_out_total

9.13.3 3. Implementation for Cribl Logstream monitoring

The integration is really straightforward, in your target tenant, access the Flex tracker management screen and load Flex use cases of your choice:

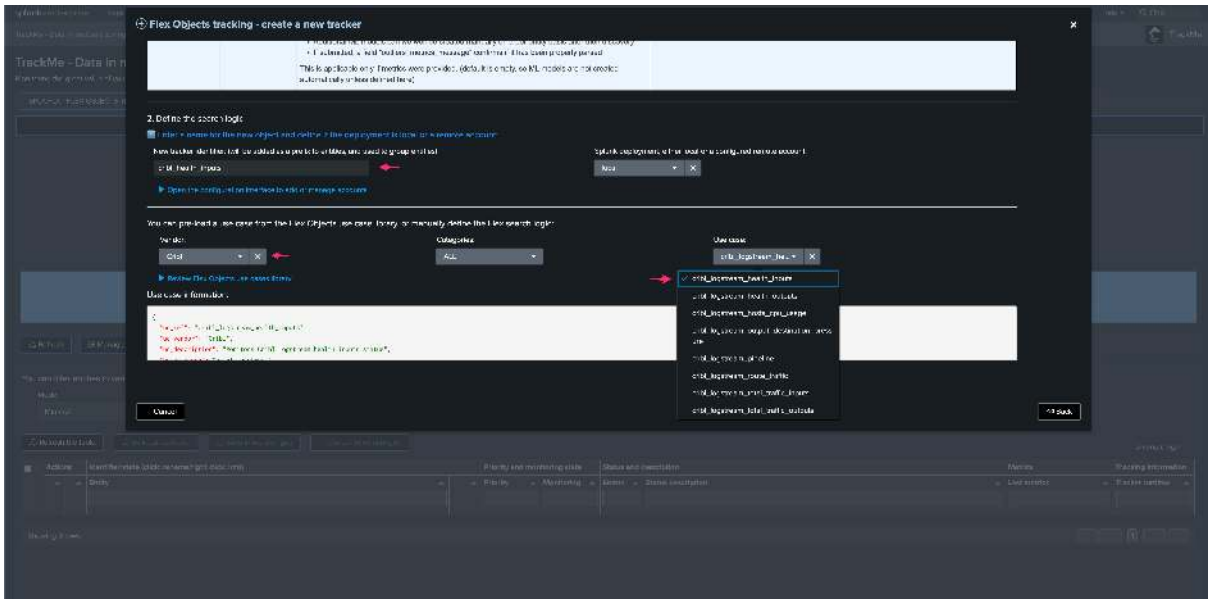




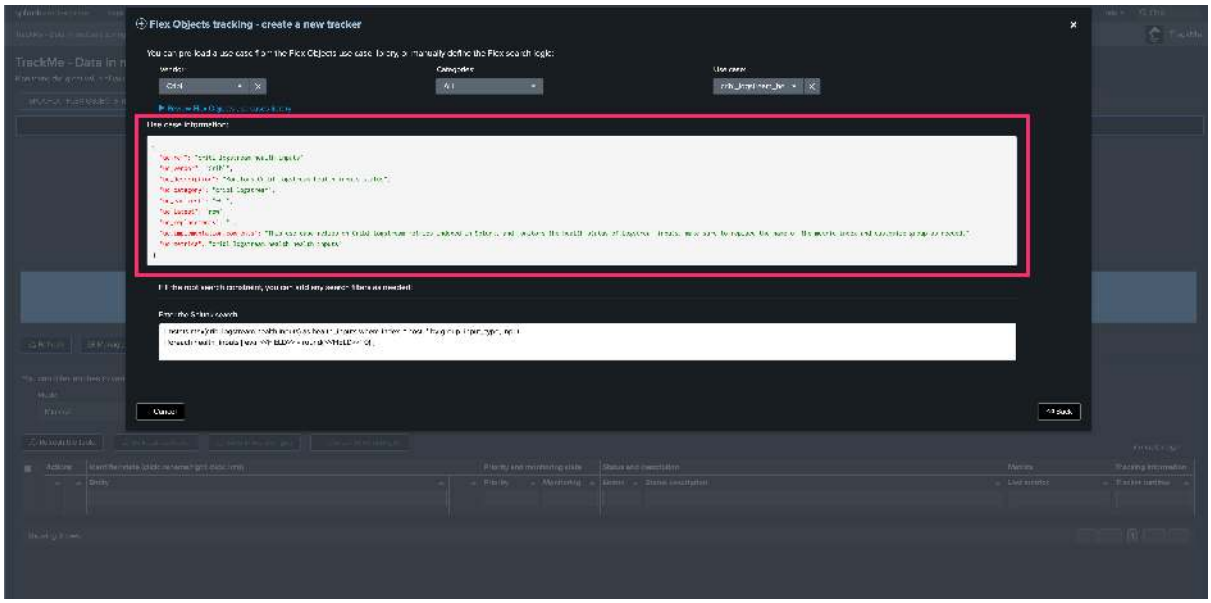
Let's load the health inputs use cases, which rely on the health metrics for Cribl Logstream inputs:

- Enter a meaningful name for the tracker

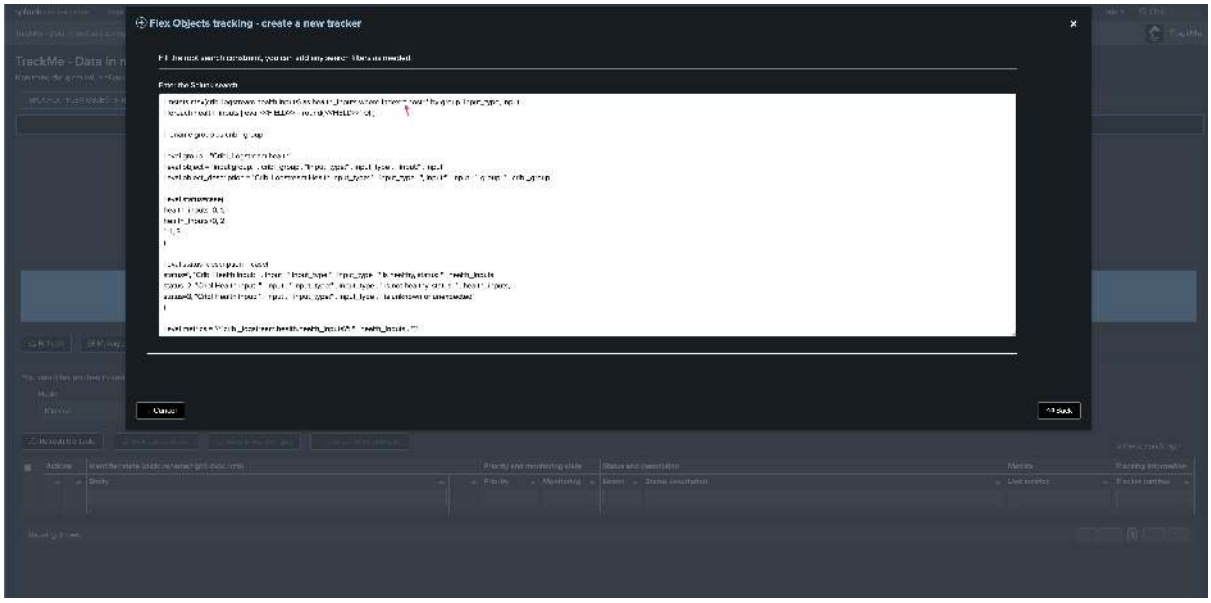
- Select Cribl as the vendor
- Select the use case reference identifier



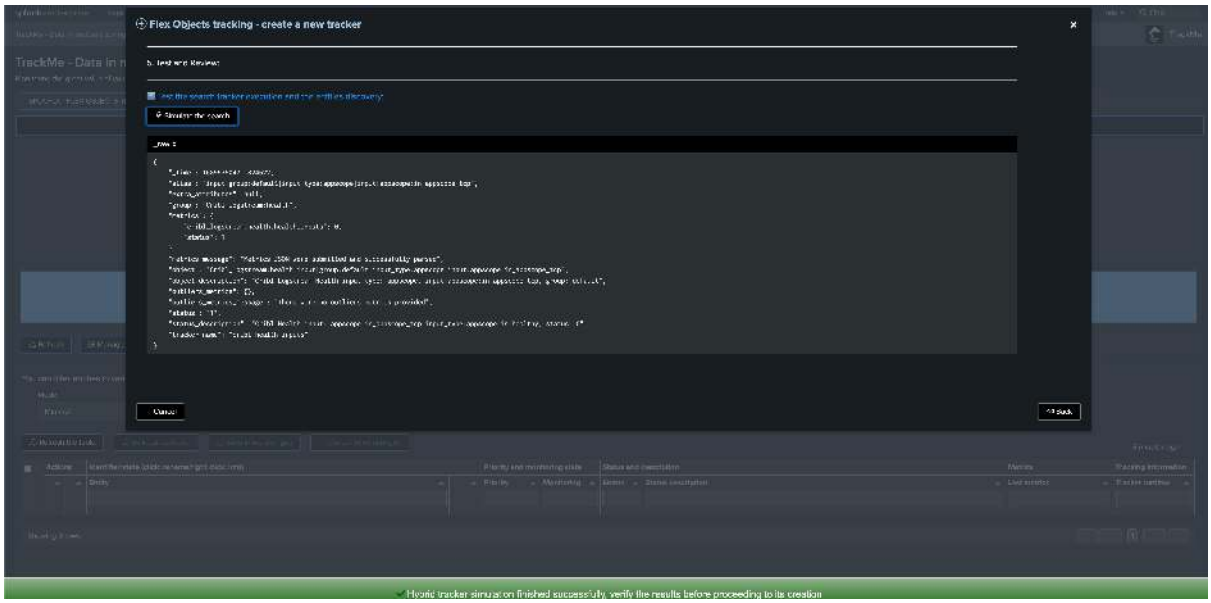
TrackMe provides high level information for the use case, such as metrics that will be generated, requirements or recommendations:

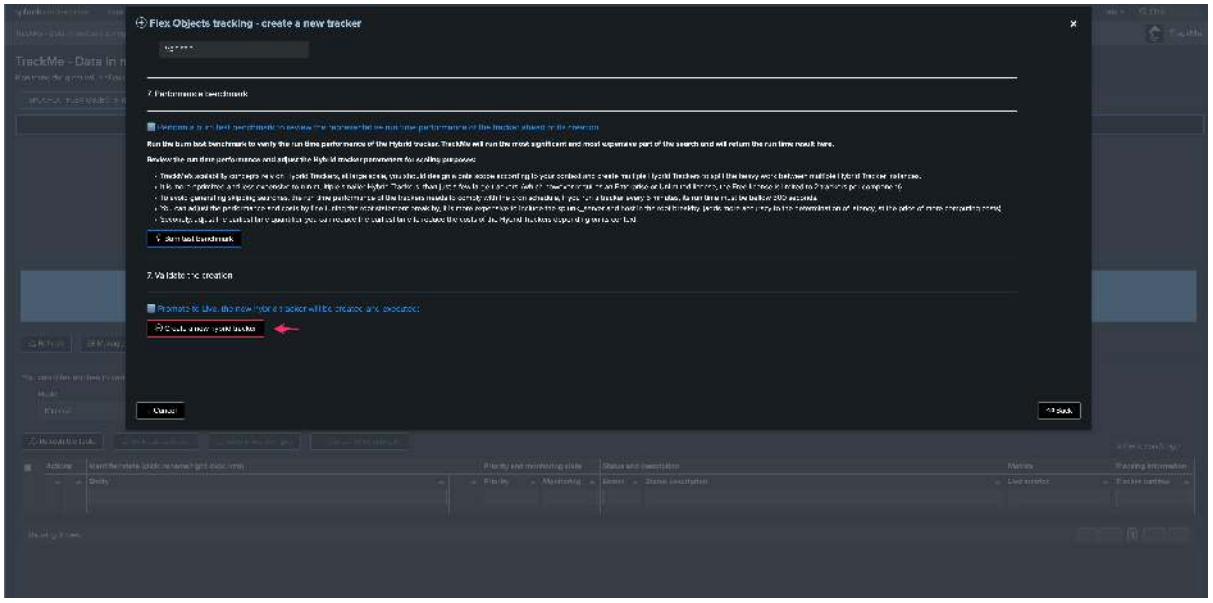


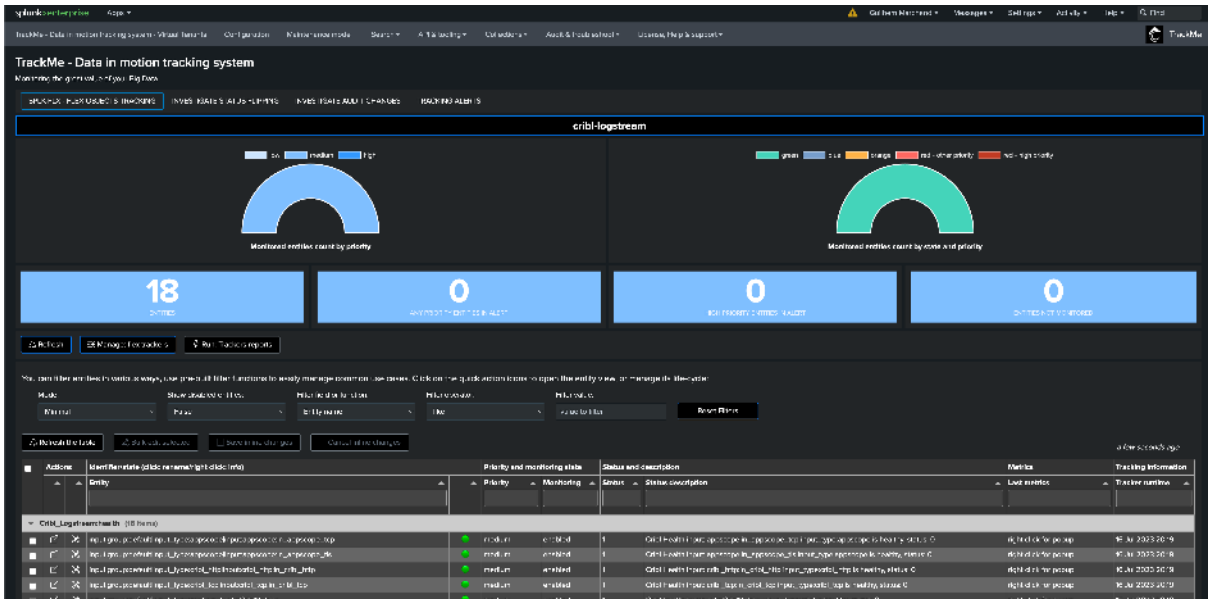
Review the search generated by TrackMe, you may want for instance to set explicitly the name of the metric index containing Cribl internal metrics:

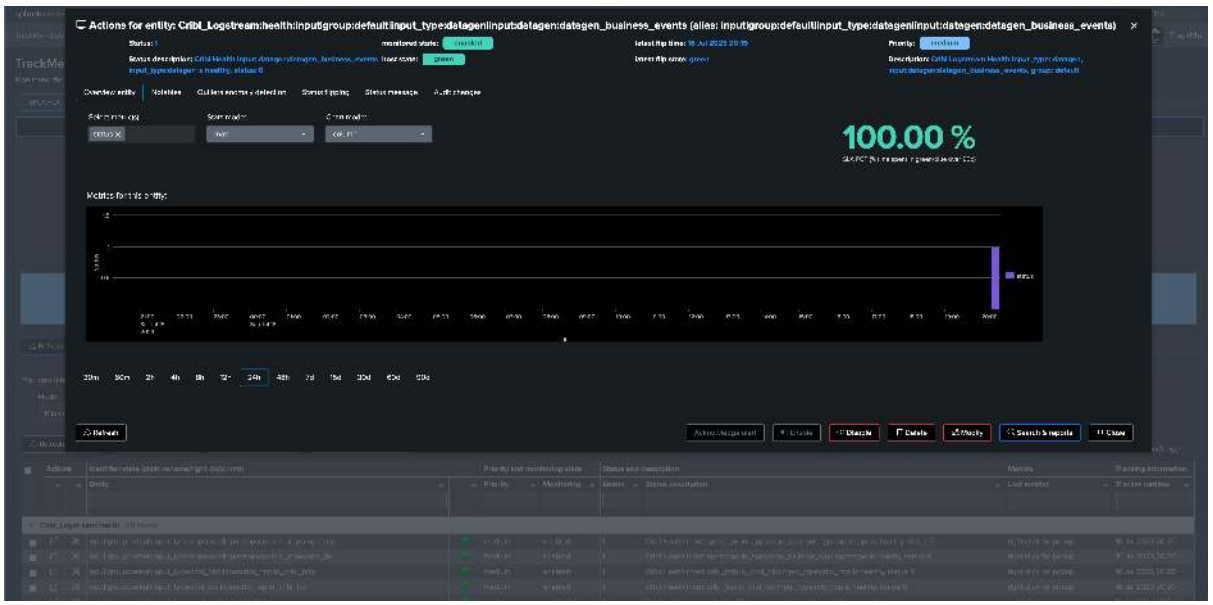


This is all you need, earliest and latest time are defined automatically, click on Simulate the search, review results and proceed to the creation of the tracker



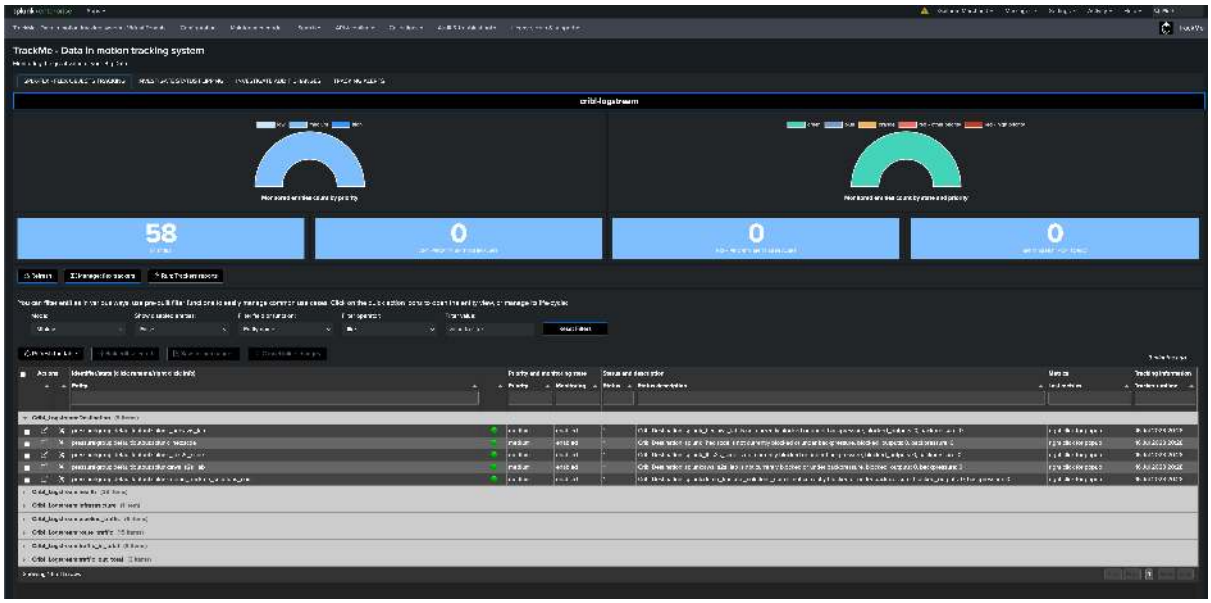

Flex Objects tracking - create a new tracker
Cancel

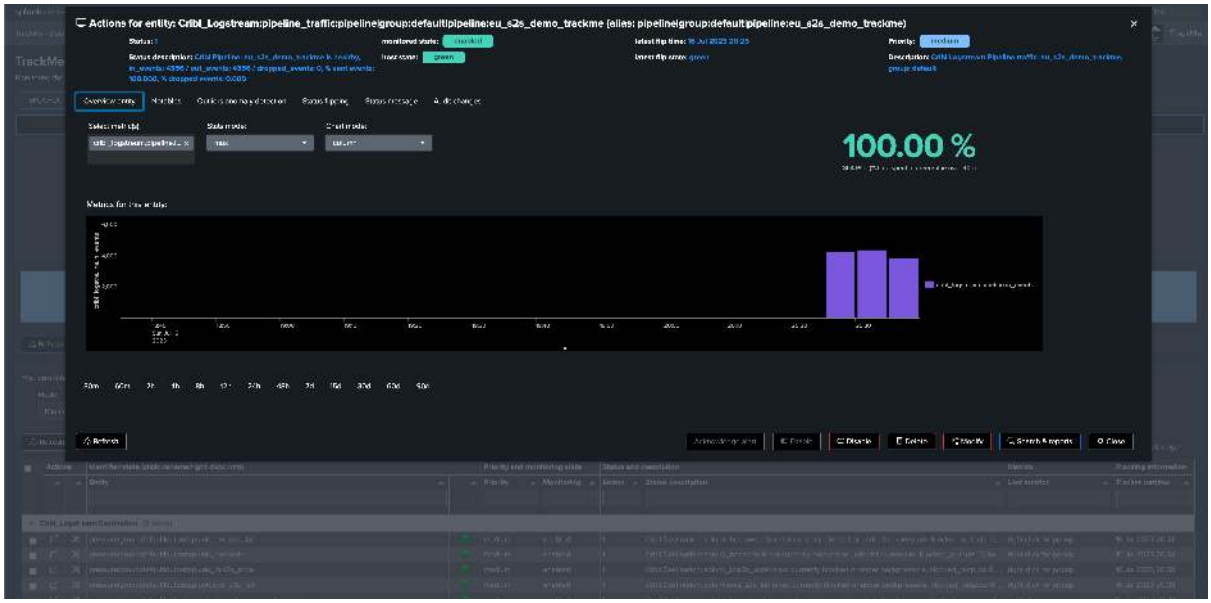

TrackMe - Data in motion tracking system
cribl-logstream
18
0
0
0

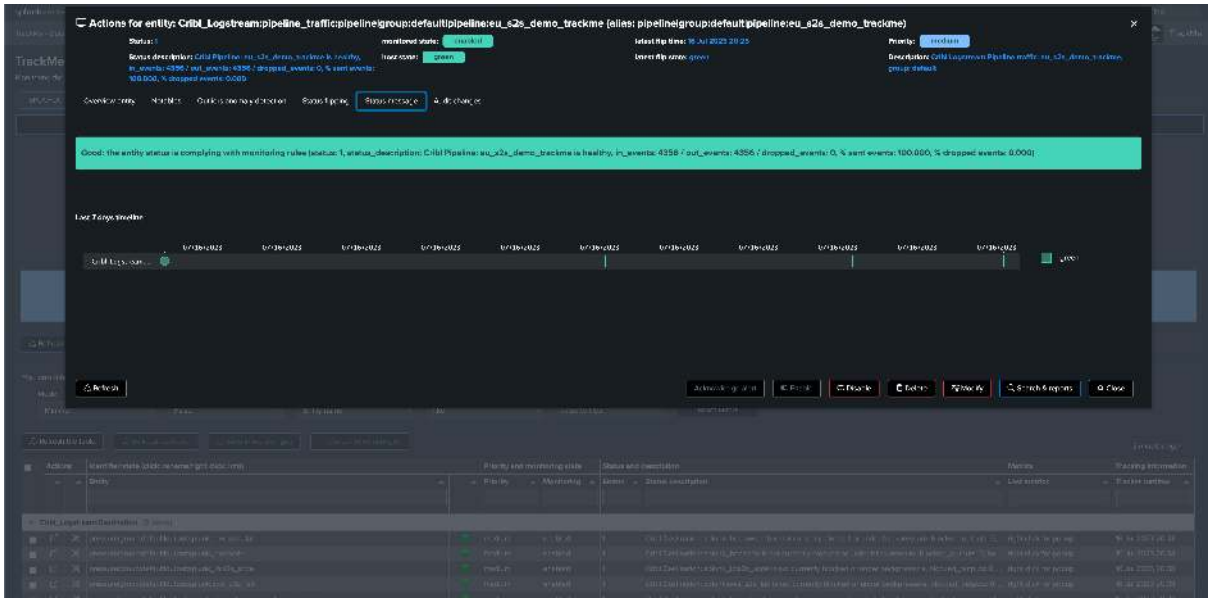

Actions for entity: Cribl_Logstream:health:inputgroup:default:input_type:datagen:inputdatagen:datagen_business_events (alias: inputgroup:default:input_type:datagen:inputdatagen:datagen_business_events)
100.00 %

9.13.4 4. Review

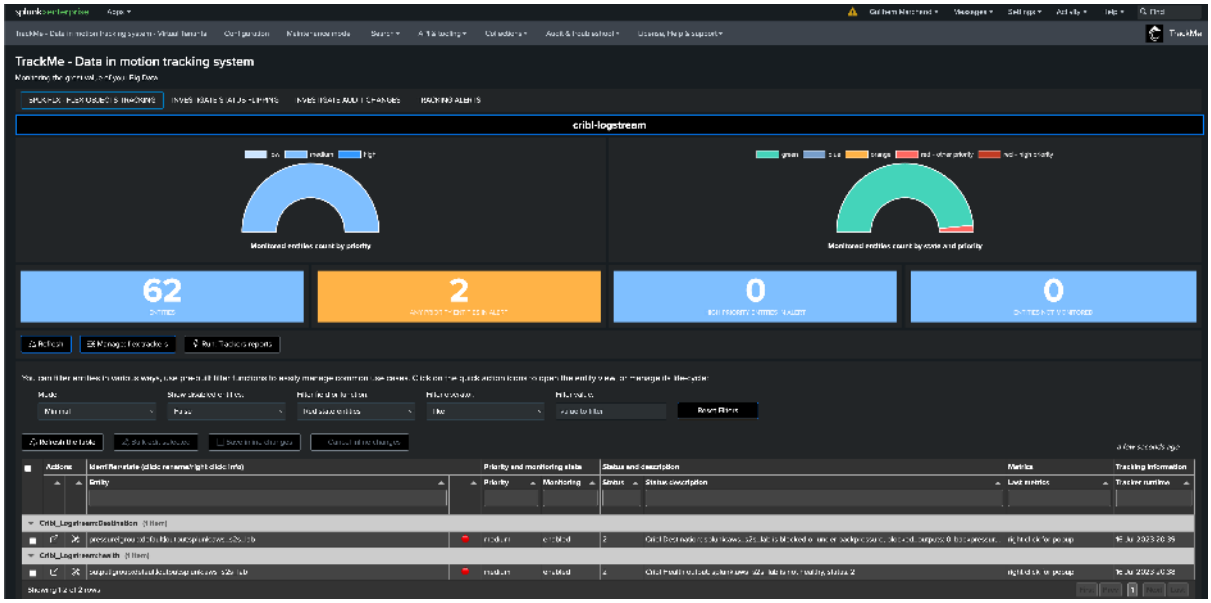
Once you have completed the integration, a full coverage of key monitoring aspect of your Cribl Logstream environment is effective:

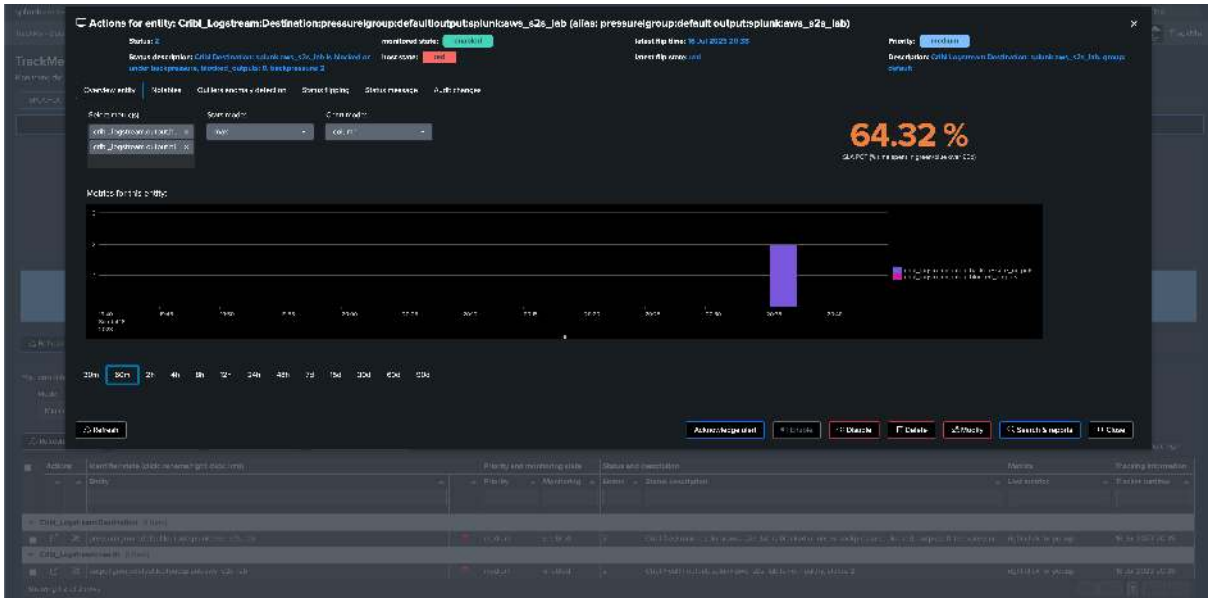


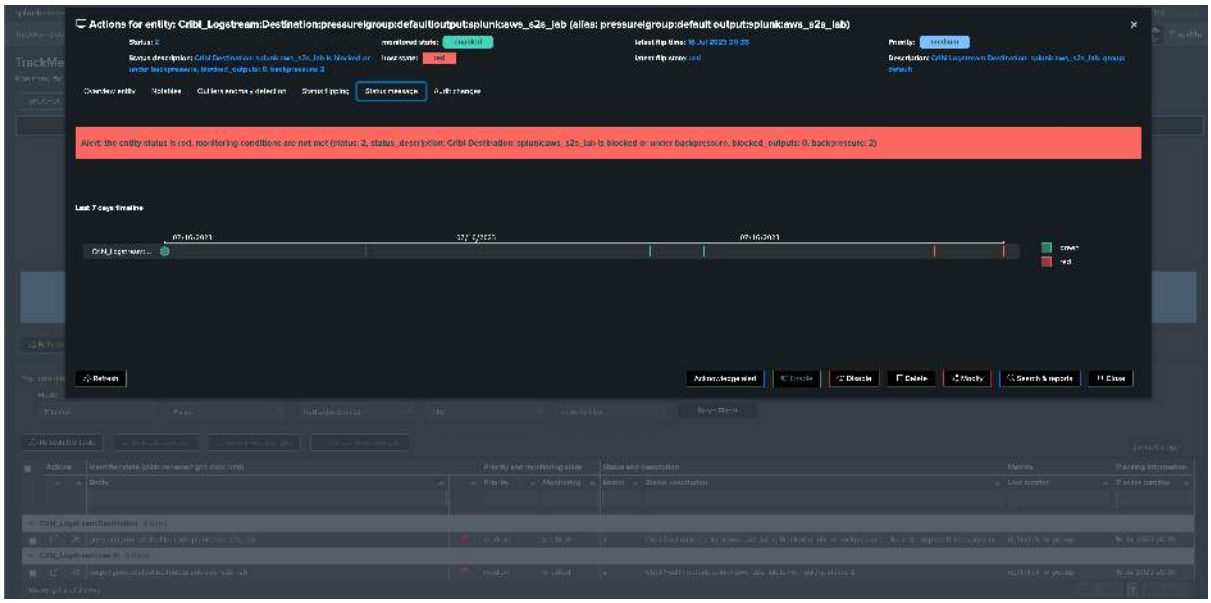




Shall an issue occur, TrackMe will automatically detect the condition with the support of Cribl internal metrics, and reflect this depending on the conditions, in the following example we simulate an outage affecting one of the Splunk destinations used in Cribl Logstream (Splunk S2S output):




Actions for entity: Cribl_Logstream:Destination:pressuregroup:default:output:splunkaws_s2s_lab (alias: pressuregroup:default:output:splunkaws_s2s_lab)
64.32 %


Actions for entity: Cribl_Logstream:Destination:pressuregroup:default:output:splunkaws_s2s_lab (alias: pressuregroup:default:output:splunkaws_s2s_lab)

CHAPTER

TEN

TROUBLESHOOT & FAQ:

10.1 Troubleshooting TrackMe

10.1.1 REST API Endpoints Logging

All TrackMe REST API handlers log events to a unique log file which is automatically indexed in Splunk, available through:

```
index=_internal sourcetype=trackme:rest_api
```

Ingest time parsing is carefully handled, so even large events wouldn't suffer from truncation.

You can rely on the logging level to review specific classes of events:

Review errors:

```
index=_internal sourcetype=trackme:rest_api log_level=ERROR
```

10.1.2 Custom Commands Logging

Each custom command backend available in TrackMe logs events to a dedicated log file, which itself ties to a specific sourcetype.

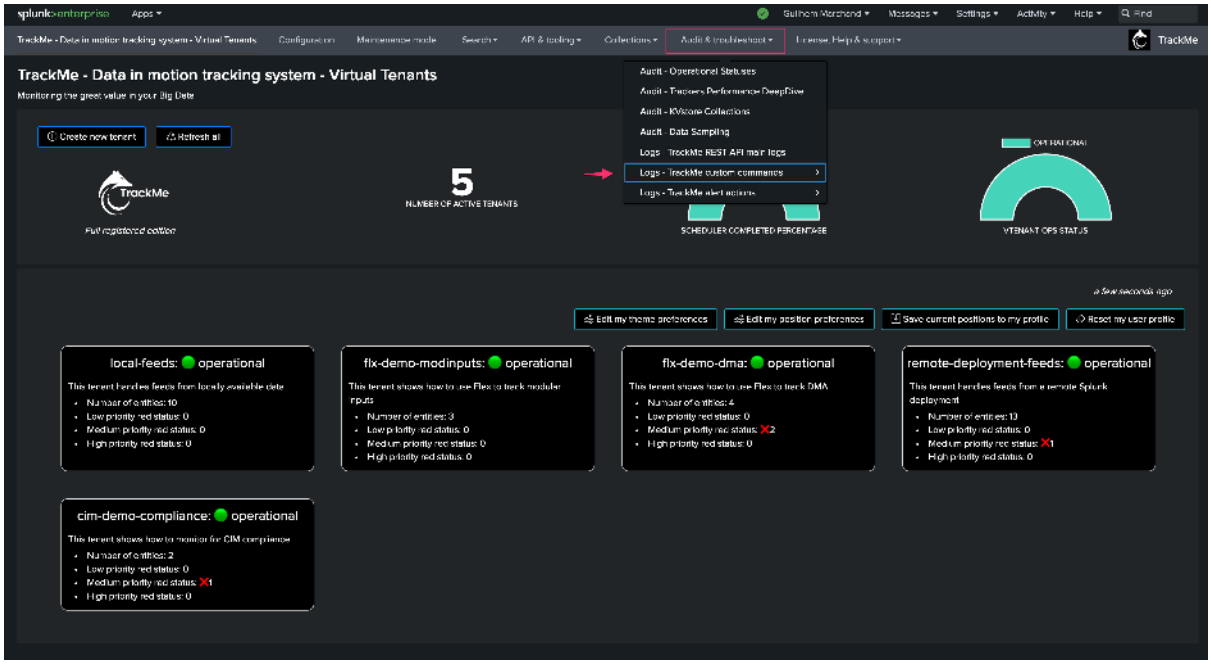
You can review all custom command logs from the following convention:

```
index=_internal sourcetype=trackme:custom_commands:*
```

Similarly, you can review any errors such as:

```
index=_internal sourcetype=trackme:custom_commands:* log_level=ERROR
```

The navigation bar provides pre-classified shortcuts per TrackMe component:



10.1.3 Alert Actions Logging

TrackMe provides multiple alert actions, such as the Notable alert action. Each alert action logs events to its dedicated log file.

You can review all modular alert actions logs from the following convention:

```
index=_internal sourcetype=modular_alerts:trackme_*
```

10.1.4 TrackMe Health Events

TrackMe produces and indexes health events for the purpose of tracking its tracker healthy status. You can review these events via the sourcetype **trackme:health**:

Assuming your TrackMe audit index(es) all start with `trackme_audit*`:

```
index=trackme_audit* sourcetype=trackme:state
```

Health events are indexed events generated from the live statuses from the following REST endpoint:

```
| trackme mode=post url=/services/trackme/v2/configuration/get_tenant_ops_status body=
↪"{'mode': 'raw'}" | trackmeopsstatusexpand
```

10.1.5 TrackMe Health Tracker

TrackMe has an important tracker that is automatically created on a per Virtual Tenant basis. Notably, this tracker is responsible for triggering upgrade procedures as needed, called schema upgrade.

Hint

TrackMe 2.1.0 improvements:

- Since TrackMe 2.1.0, the logging format was massively improved regarding this very specific component, so you can track easily the execution of every single task, its run time and so forth.

Access the logs:

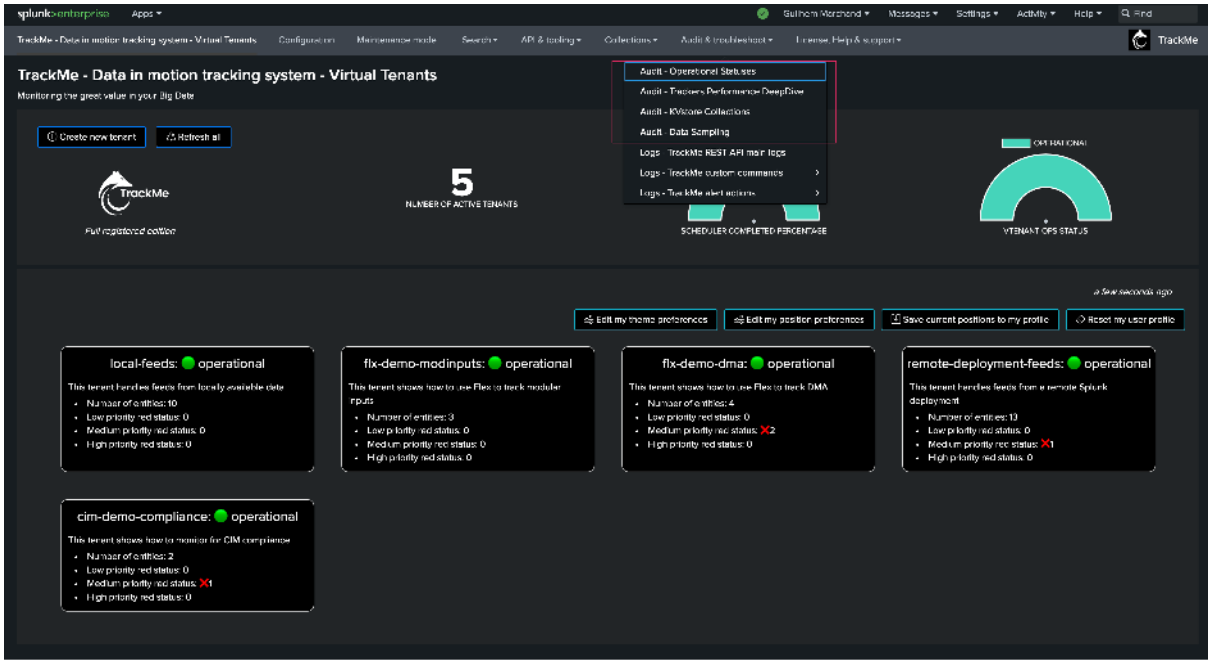
```
index=_internal sourcetype=trackme:custom_commands:trackmetrackerhealth task="schema_
↪upgrade"
```

You can track the run time of every task handled by the Health tracker using the following search example:

```
index=_internal sourcetype=trackme:custom_commands:trackmetrackerhealth instance_id=*␣
↪task_instance_id=* task=* run_time=* tenant_id=*
| table _time tenant_id instance_id task task_instance_id run_time _raw
| sort 0 - _time
```

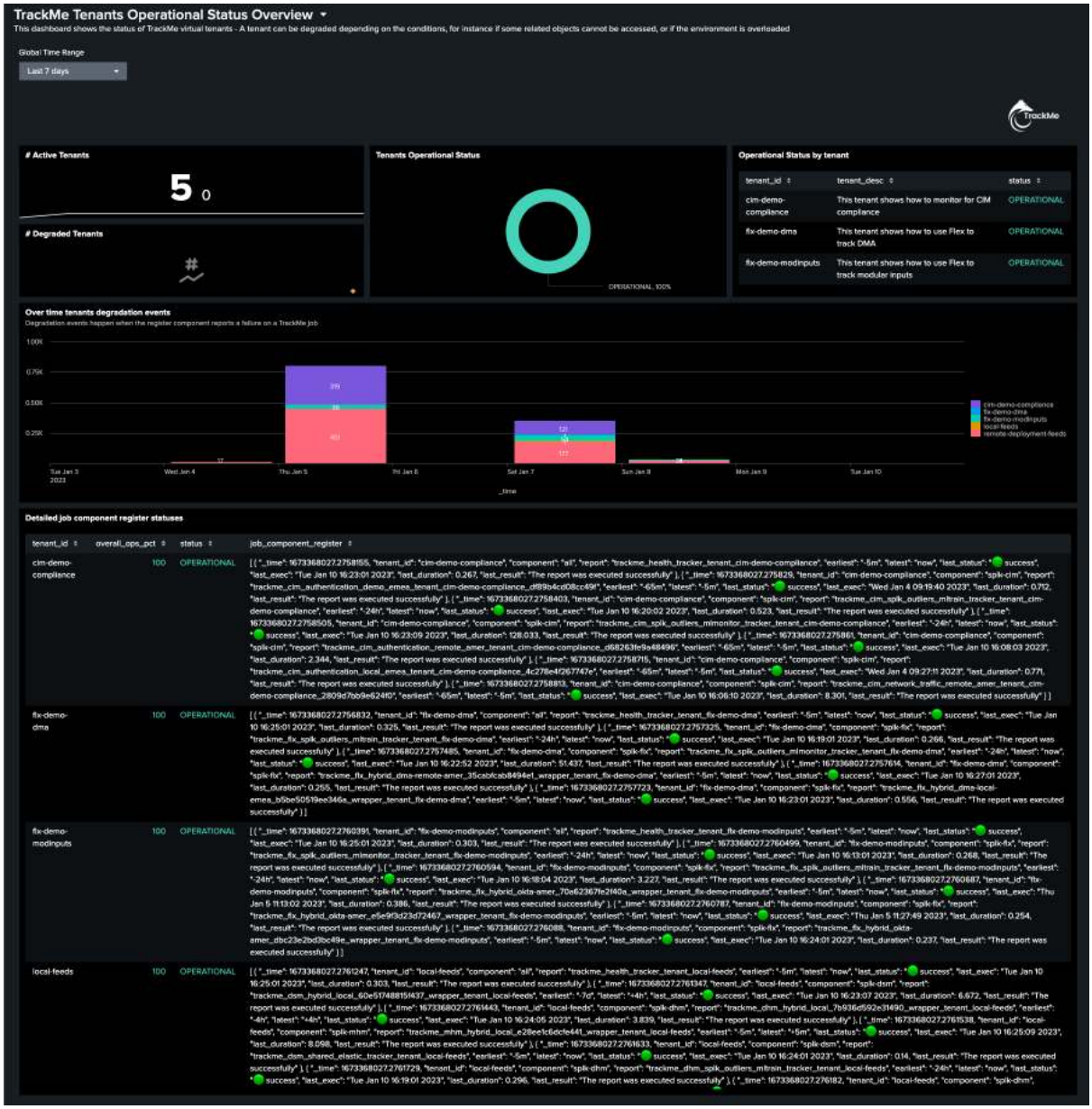
10.1.6 Audit Dashboards

Several dashboards are provided for the purposes of troubleshooting and auditing TrackMe features and behaviors:



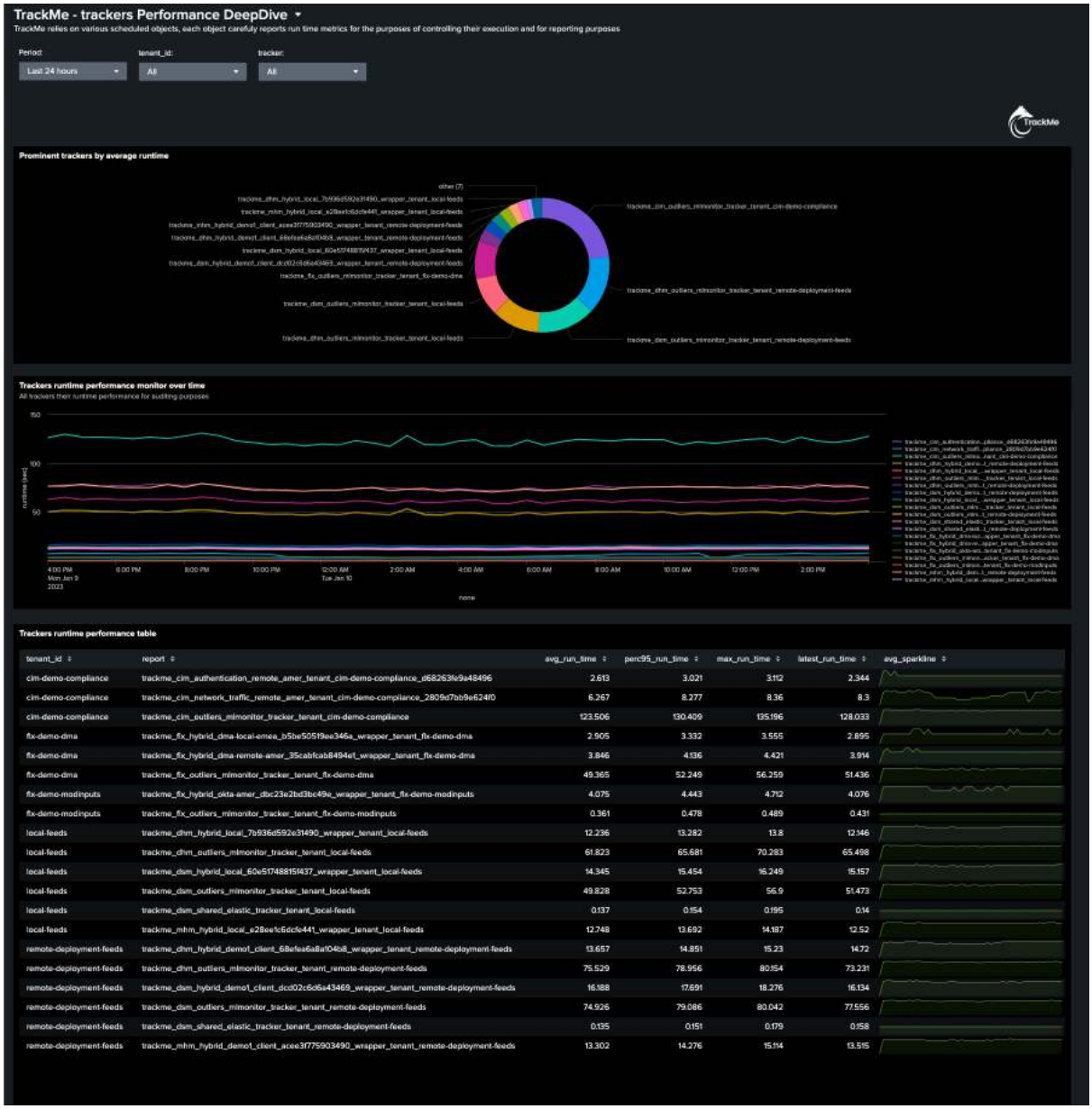
Audit - Operational Statuses

This dashboard provides a summary review of the Virtual Tenants operation statuses, which relies on the components register and the Health events:



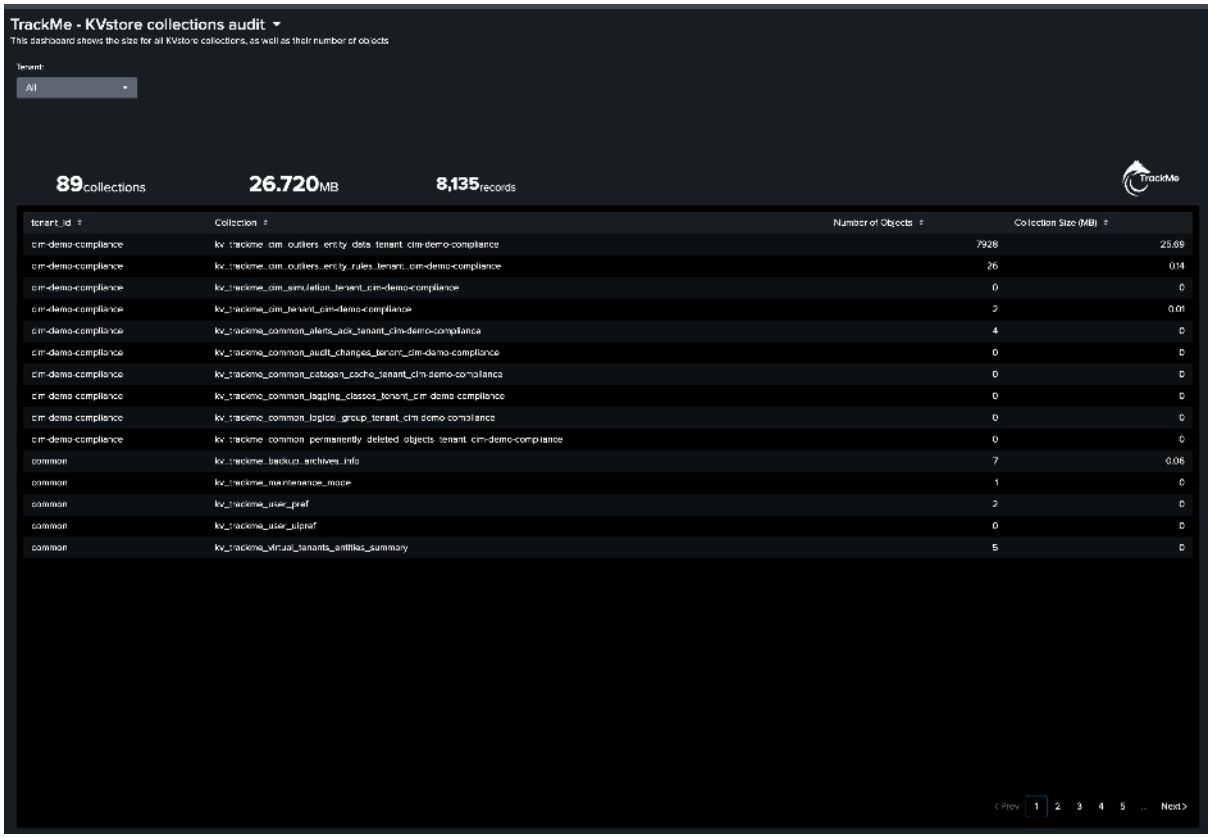
Audit - Trackers Performance Deep Dive

This dashboard provides a comprehensive review of the Trackers run time performance. This Key Performance Indicator is generated and logged when a tracker is executed:



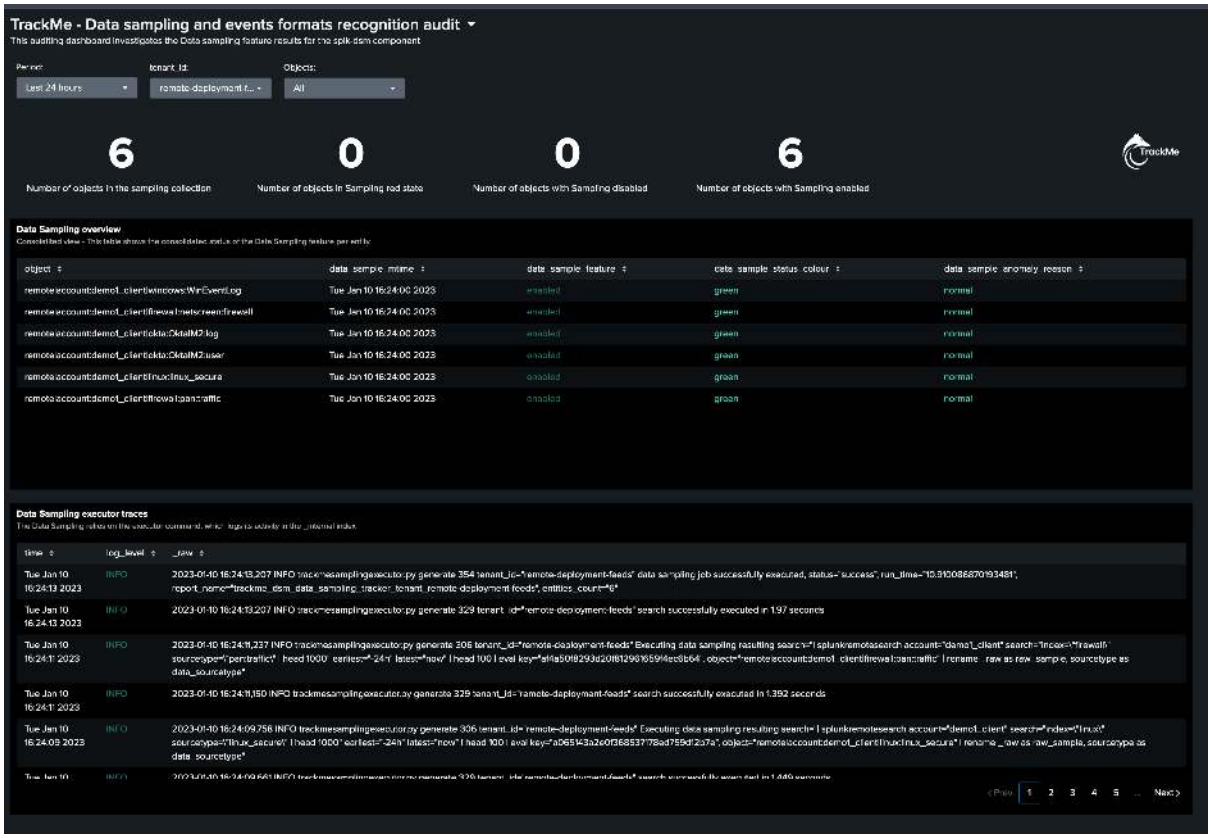
Audit - KVstore Collections

This dashboard provides a summary overview of the KVstore collections classified per tenant. This allows you to review the global size of the KVstore collections as well as the details per KVstore:



Audit - Data Sampling

This dashboard is investigating the status of the Data sampling feature for the splk-dsm component (part of splk-feeds):



10.2 Frequently asked questions (FAQ)

10.2.1 Virtual Tenants were created but the Virtual Tenants UI does not show any content

As of today, we do not support role inheritance when it comes to the RBAC configuration of the Virtual Tenants.

This means that a user needs to be a proper explicit member of either any of the administrative roles or the user roles for TrackMe to grant access to the tenant(s).

In short, you can review the workflow as follows:

trackmeload custom command

What TrackMe does is first calling the custom command called `trackmeload`, which can be called as follows:

```
| trackmeload
```

The custom command verifies the role membership of the current user and compares these with the roles defined at the level of the Virtual Tenant KVstore collection:

```
| inputlookup trackme_virtual_tenants | eval keyid=_key
| table tenant_id, tenant_owner, tenant_roles_admin, tenant_roles_user
```

If you enable the `DEBUG` logging level, the `trackmeload` custom command logs activity traces, similar to:

```
index=_internal sourcetype=trackme:custom_commands:trackmeload trackmeload.py "check␣
↪if username_role" OR "username="
```

Example:

```
2023-02-13 21:03:37,101 INFO trackmeload.py generate 105 username="admin", roles="[
↪'admin']"
2023-02-13 21:03:37,122 DEBUG trackmeload.py generate 147 check if username_role=
↪"admin" is in {'trackme_admin'} or {'trackme_user'}
```

If the current user is not an explicit member of the roles defined in the Virtual Tenant, or admin or sc_admin, the custom command will not grant access to these tenants.

10.2.2 How can I manage priority using a third-party source?

External priority management since TrackMe 2.1.10

- Since this release, TrackMe supports natively the external management of priority levels.
- This means that you can use any Splunk logic to define the levels of priority for your entities, and TrackMe will take this information into account and display it in the UI.
- Finally, you will also be able to manually update the priority level of an entity, and the external management will not override the manual update.
- See: *Managing priority externally*

10.2.3 Python error with module splunklib.results has no attribute JSONResultsReader

If you encounter this error when using TrackMe, then you have an issue in your environment which is improperly overriding the Splunk Python SDK used by TrackMe.

The symptoms lead to TrackMe various failures with the message:

```
module 'splunklib.results' has no attribute 'JSONResultsReader'
```

This happens if TrackMe is forced to use an incompatible and outdated version of the Splunk SDK for Python which does not have the JSONResultsReader module.

You can verify the version of the SDK which TrackMe uses by opening a Python interpreter and running the following command:

```
/opt/splunk/bin/splunk cmd python
```

Then paste the following code and press enter:

```
import sys, os
splunkhome = os.environ["SPLUNK_HOME"]
sys.path.append(os.path.join(splunkhome, "etc", "apps", "trackme", "lib"))

import splunklib.results as results
import pkg_resources

def get_package_version(package_name):
    try:
        return pkg_resources.get_distribution(package_name).version
    except pkg_resources.DistributionNotFound:
        return None

sdk_version = get_package_version("splunk-sdk")
if sdk_version:
    print("Splunk SDK for Python Version:", sdk_version)
else:
    print("The Splunk SDK for Python is not found or its version can't be determined.
↪")
```

The expected output should show the SDK version which should match what TrackMe is shipped with. The SDK version can be found in:

```
trackme/lib/splunklib/__init__.py
```

Example:

```
__version_info__ = (1, 7, 4)
```

This is also the case for every other Splunk application shipping the Python SDK. Once you have identified the application causing this, you can either manually upgrade the SDK or identify where a custom configuration such as a modification of environment variables (ex: PYTHONPATH) leads to this unexpected behavior.

You can also execute the following Python code in the interpreter to show the value of the PYTHONPATH from the perspective of Splunk:

```
import os
import sys

# Print the PYTHONPATH environment variable
pythonpath = os.environ.get('PYTHONPATH')
if pythonpath:
    print("PYTHONPATH:", pythonpath)
else:
    print("PYTHONPATH is not set.")
```

(continues on next page)

(continued from previous page)

```
# Print the current sys.path
print("\nsys.path:")
for path in sys.path:
    print(path)
```

CHAPTER

ELEVEN

VERSIONING AND BUILD HISTORY:

11.1 Release notes

11.1.1 Version 2.1.21 - build 1752477750 (14/07/2025)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.0.x and earlier.
- The last release compatible with Splunk 9.0.x is TrackMe 2.0.99

- SHA256: 0d10f8440707aa1939d20cbc95995f58d753b5412c4a7de47854f3d38a454077

Table 1: Fixed Issues

Issue Number	Description	Details
#1127	Flex Object (splk-flx) mvfind matches substrings	Fixes a logic error in the mvfind function used in SPL for Flex Object CIM compliance. The previous implementation matched substrings, which caused inconsistent summary status values. Now matches only exact strings.
#1134	Flex Object simulation fails if object field is missing	Ensures the object field is present in Flex simulations. If the field is missing, the parser sets status to 1 instead of raising a Python exception.
#1140	Stateful alerting fails on Python 3.7 (Splunk 9.2.x)	Resolves Python compatibility issues in stateful alerting caused by OS-dependent libraries missing support for Python 3.7 in Splunk 9.2.x.

Table 2: Enhancements and New Features

Issue Number	Description	Details
#1125	New "pending" priority level	Adds a new "pending" priority level to support qualification workflows, enabling entities to be flagged for review before being included in alerting.
#1128	Alert macro usage update	Ensures alerts use the updated macro *trackme_apply_maintenance_mode_v2* for improved behavior.
#1129	Maintenance macro matching safety	Prevents incorrect tenant ID matches due to substring logic in macros, improving reliability of maintenance targeting.
#1130	Maintenance timezone safety	Improves handling of timezone conversion when the user's profile does not explicitly define a timezone and operates outside UTC.
#1131	Schema upgrade robustness	Ensures upgrades can continue safely even if conflicting objects exist from prior rollbacks or restores.
#1132	Fields quality - Render regex per field	Improves backend and UI to track and display the regex expression applied per field in field quality analysis.
#1135	Flex Object thresholds - Compare against metrics	Allows threshold logic to reference another metric in the same record rather than being limited to static numbers.
#1136	Stateful alert - Custom chart time range	Adds support for customizing the time range used to generate embedded charts in stateful alert emails.
#1137	Stateful alert - Cooldown between alerts	Prevents creation or updates of incidents within 60 minutes of the last opened alert to avoid premature or duplicate detections.
#1138	General Health Tracker - Cleanup duplicates	Adds logic to automatically detect and remove duplicated open incidents from the stateful KVstore collection.
#1141	Fields Quality - Accept null or empty fields	Allows the field dictionary to accept empty or null field values and override regex failures using *accept_unknown* or *accept_empty*.

11.1.2 Version 2.1.20 - build 1751374209 (01/07/2025)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.0.x and earlier.
- The last release compatible with Splunk 9.0.x is TrackMe 2.0.99

- SHA256: f054ff962fac174aba5ecd6efdaf080247f8cdc9225bd3b0a95582b0027146dd

Table 3: Fixed Issues

Issue Number	Description	Details
#1124	Elastic Source tracker disabled by mistake	A hotfix regression affected Elastic Sources where the health tracker optimization maintenance task incorrectly disabled the shared elastic source tracker even when entities still required management. This release restores correct tracker enablement logic.

11.1.3 Version 2.1.19 - build 1751273959 (30/06/2025)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.0.x and earlier.
- The last release compatible with Splunk 9.0.x is TrackMe 2.0.99

- SHA256: fd492b3d6fd7a1271ac6069baf0c9f5106d6199fdb25ce4095b993ee23a01a1e

Table 4: Fixed Issues

Issue Number	Description	Details
#1093	Non existing custom command reference	A non-existing custom command (*trackmesendemail*) was referenced in *commands.conf*, potentially leading to Splunk warnings or appinspect failures. Removed for compliance and clarity.
#1098	Configuration screen rejects advanced group by	Regex validation in Virtual Tenant configuration prevented *eval* or non-CSV group by options, limiting Tabulator flexibility. Validation logic is now adapted.
#1099	Regex validation and default value issues	Fixes logic errors in regex validators and corrects invalid default values to ensure robust configuration parsing.
#1100	SmartStatus future tolerance extraction bug	Host and source extractions failed due to incorrect retrieval of future_tolerance values. Now resolved for reliable extractions in DSM/DHM.
#1101	UI freezes on large search source code	Show Trackers view loaded entire SPL source, which could freeze the browser with large searches. Optimized for performance stability.
#1104	Email formatting inconsistencies	Remaining formatting issues in some email clients, notably Outlook, resolved by simplifying HTML div structures.
#1105	Embedded chart compatibility in emails	Replaces SVG with PNG in alert emails to ensure full compatibility with all clients (Outlook, Gmail, etc.), along with supporting code enhancements.
#1111	Outliers chart UI glitch	Deleting and re-creating ML models broke the main outlier chart until full UI reload. Fixed to restore seamless chart rendering.
#1112	Flex status metric mismatch	Under some threshold breach conditions, the status metric did not reflect the decision maker logic. Now moved generation to *trackmedecisionmaker* for consistency.
#1113	Incorrect component registration in Disruption	Invalid component sent to registration function caused harmless log errors. Now corrected to send the proper component.
#1114	Ack tracker anomaly reason comparison bug	Under specific conditions, Ack could release prematurely due to incorrect comparison logic. Normalization now ensures accuracy.
#1116	Role concurrency exception handling in SDK search	*run_splunk_search* did not handle role concurrency limits gracefully. Now improved to detect and retry accordingly.
#1123	Event field wrongly indexed as metadata	*trackme_events_ingest_evals* and *trackme_audit_events_ingest_evals* transforms caused *event* fields to be indexed as metadata, creating warnings due to length. This is now prevented.

Table 5: Enhancements and New Features

Issue Number	Description	Details
#1102	Incremental schema upgrades	The schema upgrade process now updates the version incrementally after each function completes, improving resilience against upgrade interruptions.
#1103	Prevent concurrent backups post-upgrade	Health trackers now detect in-progress backups from logs rather than KVstore, preventing simultaneous backups during upgrades.
#1106	More embedded charts in alerting	Adds incident, flipping, and state charts as embedded visuals in StateFul alerting emails, enhancing diagnostic clarity.
#1107	Persistent charts across alert phases	Ensures ML Outliers and related charts are included consistently in opened, updated, and closure phases for continuity.
#1109	Outliers exclusion periods apply to all models	Period exclusions can now be applied to all ML models for an entity simultaneously via UI bulk actions.
#1110	Background ML bulk actions	*mlmonitor* and *mltrain* bulk operations now execute asynchronously (fire-and-forget), avoiding UI timeout issues.
#1116	Status metrics for inactive Flex entities	Flex objects now generate status metrics via inactive trackers with associated handler events for accurate lifecycle tracking.
#1117	Automated utility scheduling	TrackMe utilities (e.g., mltrain/mlmonitor) schedules are automatically enabled/disabled based on operational need, reducing workload.
#1118	Health tracker self-monitoring	Adds maintenance task to verify Virtual Tenant Health Tracker is active, ensuring core functionality is never unintentionally disabled.
#1119	Hide cron schedule in Alerts UI	The cron schedule field is now hidden from alert creation UI to prevent misconfiguration by untrained users.
#1120	Simplified StateFul alert creation	Provides granular priority-based actions for notifications and commands, streamlining the StateFul alert setup workflow.
#1121	Improved search backoff strategy	*run_splunk_search* now uses progressive backoff for retries, improving reliability on systems with concurrency limits.
#1122	Fields Quality major enhancements	Introduces three new SPL utilities (*trackmefieldsqualitytyextract*, *trackmefieldsqualitygensummary*, *trackmefieldsqualitygendict*) and a full rewrite of the Flex Object OOTB use case for CIM compliance and continuous monitoring.

11.1.4 Version 2.1.18 - build 1748377090 (27/05/2025)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.0.x and earlier.
- The last release compatible with Splunk 9.0.x is TrackMe 2.0.99

- SHA256: 6aab45bc762ea66ca4f06f961efdc2244a52c31c511dff5b70fbba7090602c05

Table 6: Fixed Issues

Issue Number	Description	Details
#1034	Tabulator group state not preserved	Resolves improper group state reset on manual or auto refresh in TrackMe UI Tabulator.
#1041	DHM metrics incorrect for multi-sourcetypes	Fixes macro logic to account for all sourcetypes on volume calculations in *splk-dhm*.
#1043	*email_send_update_if_ack* not respected	Ensures updates are emailed if entity is acknowledged and flag is enabled.
#1045	Unbalanced quotes in *splk-mhm*	Fixes tracker regression from 2.1.16 affecting hybrid creation.
#1049	Email chart uses *min* instead of *max*	Status now reflects highest state instead of lowest.
#1050	Outlook formatting issues in emails	Enhances HTML for Outlook client compatibility.
#1052	Virtual Tenant creation fails with limited permissions	Applies proper privilege elevation in REST headers.
#1055	StateFul alerts not restored	Includes StateFul alerts in backup and restore processes.
#1056	Scoped restore affects unrelated KV collections	Filters collections by tenant during scoped restore.
#1057	Restore fails on Splunk email actions	Excludes native Splunk email settings to prevent restore failures.
#1059	Drilldown uses API v1 instead of v2	Corrects dashboard links to target REST API v2.
#1064	Outliers default latest time not applied	Updates *trackme-settings.conf* to use *-1d* as default.
#1065	ML charts fail with past *latest_time*	Fixes chart simulation to support historic latest values.
#1066	No UI refresh after Flex logical group edit	Automatically refreshes status views post group changes.
#1069	Missing *event_id* for inactive objects	Ensures summary events include *event_id* for red-state entities.
#1070	Full-down service shows orange	Adjusts status logic to show red (status=2) if all members are down.
#1074	State event error in *splk-cim*	Resolves outlier-based Python error in CIM component.
#1076	Ack expires on empty anomaly list	Prevents misinterpretation of empty lists as state changes.
#1079	Tracker fails if *tags_manual* is native list	Adds type safety to ensure *tags_manual* is always a string.
#1086	Blue state considered down	Treats *object_state=blue* as up for Flex convergence logic.
#1089	Missing *mvexpand* for email accounts	Ensures all delivery options are listed in UI popup.
#1090	Tag audit records include full entity	Fixes audit trail to log tag list only, not entire object.

Table 7: Enhancements and New Features

Issue Number	Description	Details
#1032	Load tenants if one is corrupted	Backend resilience added to avoid full failure on partial tenant corruption.

continues on next page

Table 7 – continued from previous page

Issue Number	Description	Details
#1033	Skip disabled entities in convergence	Default behavior now excludes *monitored_state="disabled"* from convergence.
#1035	Right-click popup improvements	Enhances readability of contextual entity view.
#1036	Add *converging_status* to extras	Adds *up/down* visibility in extra_attributes.
#1037	Auto-Ack config in wizard	Enables toggling Auto-Ack at creation time for StateFul alerts.
#1038	Prevent new alert if Ack is active	Updates only if Ack exists, avoids duplicate threads.
#1039	Render gaps in alert charts	Gaps now rendered instead of misleading zeroes.
#1040	Allow closure within 5 mins	Prevents suppression of incident closure events.
#1042	Accept threshold in seconds or units	Enhances threshold entry with unit suffix support (h/d/w).
#1044	Fetch tracker definition live	UI now pulls real-time definition to avoid outdated data confusion.
#1047	New Studio dashboard template	Adds OOTB Dashboard Studio template for service health.
#1048	Support JSON dict parsing	Allows *trackmefieldsquality* to extract structured fields.
#1051	Minimum percentage up for green	Convergence logic now accepts minimum healthy percentage threshold.
#1053	Alert status in email header	Visually tags alerts as critical/informational with color code.
#1054	Retry on restore failure	Adds fallback retry if dependency isn't restored yet.
#1058	Filter API by keys	Accepts comma-separated object keys for component data.
#1060	UI/URL filter by keyid	Adds keyid param support for safer drilldown with special chars.
#1061	Use keyid in alert drilldown links	Avoids URL errors for non-ASCII object names.
#1063	Configurable thresholds per entity	Flex tracker and UI support for per-entity metric thresholds.
#1067	Pre-checks before Ack call	Ensures entity is alerting before REST-based Ack.
#1068	Larger chart height in Flex UI	Defaults to 450px for improved visual clarity.
#1072	List down entities first	Prioritizes down entities in status descriptions.
#1073	Prism.js for JSON formatting	Syntax highlights JSON in entity popups.
#1075	Skip disabled entity in alerting	Ignores processing for disabled entities in StateFul alerting.
#1077	Priority selector in flip UI	Adds dropdown in status flipper for setting entity priority.
#1078	Disruption Queue feature	Introduces time-based disruption tracking before turning entities red.
#1080	Improve audit search consistency	Enhances uniformity of audit tab searches across components.
#1081	Update OOTB use cases	Leverages new thresholds and disruption logic in bundled Flex trackers.
#1082	Push expected data from CMDB	Adds *trackmepushdatasource* to create entities from referentials.
#1083	Improve tenant wizard preview	Enhances UX with embedded table preview in creation flow.
#1084	Disable SLA events if frequency=0	Skips SLA breaches generation for disabled or configured tenants.

continues on next page

Table 7 – continued from previous page

Issue Number	Description	Details
#1085	Suppress entity list in all-green	Avoids listing all entities if they're all up.
#1088	Improve corrupted tenant error UI	Adds friendly remediation steps for tenant load failures.
#1091	Check for orphan/outdated incidents	New health tracker task verifies alert integrity.
#1092	Remove deprecated health task	*synchronize_trackers_attr* task removed as obsolete.

11.1.5 Version 2.1.17 - build 1746041932 (30/04/2025)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.0.x and earlier.
- The last release compatible with Splunk 9.0.x is TrackMe 2.0.99

- SHA256: bbc6cb524f5e4d93d70f5c9f003408f8a8b6d9a0231919ba3e966abafc1ab5a3

Table 8: Fixed Issues

Issue Number	Issue Description	Issue Details
#1031	Appinspect - prohibited characters in sourcetypes	A newly introduced Appinspect check `check_props_conf_has_no_prohibited_characters_in_sourcet` leads to Appinspect failure. This update resolves the issue.

Table 9: Enhancements and New Features

Issue Number	Enhancement Description	Enhancement Details
#1027	New streaming command `trackmefieldsquality` for fields quality auditing	Adds a new TrackMe custom command designed to support fields quality auditing in Splunk environments.
#1028	SmartStatus - improve search for data in the future	Extends the future/past time range in the UC logic to capture events where timestamps are more than 4 hours ahead, preventing search failures.
#1029	Upgrade Tabulator JS library	Upgrades the embedded Tabulator JavaScript library to version 6.3.1.
#1030	Health Tracker - new task `check_tenants_indexes_settir`	Introduces a new health task to automatically verify and fix missing index declarations in tenant configurations, which could otherwise lead to errors during event generation.

11.1.6 Version 2.1.16 - build 1745791632 (27/04/2025)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.0.x and earlier.
- The last release compatible with Splunk 9.0.x is TrackMe 2.0.99

- SHA256: 9f31ae61532da1b343e6ff1dcc5734c33e3dce61972809c5ce2dd4bd54715f9c

Table 10: Fixed Issues

Issue Number	Issue Description	Issue Details
#1024	TrackMe sub-logging regression	TrackMe 2.1.15 caused regressions in logging subsystems, resulting in missed ingestion of audit events, handler events, and others. This release fixes all associated logging issues.
#1025	TrackMe Home UI - SLA theme color issue	In the SLA tab, if the SLA is breached, the message should display in red theme. This was not happening properly in some use cases and is now corrected.

Table 11: Enhancements and New Features

Issue Number	Enhancement Description	Enhancement Details
#1018	StateFul Alerting - Execute commands based on incident state	Introduces active commands triggered when incidents are opened, updated, or closed, enabling state-aware integration with third-party systems (e.g., ServiceNow). See documentation.
#1020	SOAR Monitoring - Page size control and recent window logic	Enhances the `soar_playbook_status` action to allow page size control and reduce the concurrent playbooks search window from 1 hour to 5 minutes for better scalability.
#1021	Audit & Troubleshoot - Splunk Remote Accounts dashboard	Provides a new dashboard to review the status of Splunk Remote Deployment accounts and monitor token rotation status via the new `trackmetestremoteaccounts` custom command.
#1022	Flex Object Library - Search Head health check UC update	Updates the Search Head health check use case to immediately trigger red status if the response is anything other than success.
#1023	StateFul Alerting - Include trackme:state events for early trigger	Incorporates `trackme:state` events into alerting logic to allow faster incident triggering without waiting for new event movements. Schema upgrade will update existing alerts automatically.
#1026	StateFul Alerts - Control email updates when Acknowledged	Adds a control option at the alert level to determine whether updated email notifications should be sent when the entity has an active Acknowledgement, improving alerting behavior.

11.1.7 Version 2.1.15 - build 1745332184 (22/04/2025)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.0.x and earlier.
- The last release compatible with Splunk 9.0.x is TrackMe 2.0.99

- SHA256: 6a3df5cfd71bfdd1f841c558f5db19c486c7a627716c8993cfb8dda1420b60bf

Table 12: Fixed Issues

Issue Number	Issue Description	Issue Details
#1019	Trackers and degraded mode - remote accounts regression	A regression caused remote account-based trackers to remain in degraded mode even after the issue was resolved, preventing proper recovery after outages.

Table 13: Enhancements and New Features

Issue Number	Enhancement Description	Enhancement Details
#1016	TrackMe REST API logging - prevent external collision	Additional improvements were made to TrackMe's REST API logging to ensure no logging collision occurs with other external applications in shared environments.
#1017	StateFul Alerting - Alert wizard UI update	Enhances the alert creation wizard to dynamically display or hide "Emails only" options based on the selected alert mode, improving clarity and user experience.

11.1.8 Version 2.1.14 - build 1744885691 (17/04/2025)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.0.x and earlier.
- The last release compatible with Splunk 9.0.x is TrackMe 2.0.99

- SHA256: 0087a889a3c25510a9a6a71c86f5b0c7f696d064ac3c17a24ca9ff6f25efeb89

Table 14: Fixed Issues

Issue Number	Issue Description	Issue Details
#1013	TrackMe logging - REST API potential logging conflicts	In certain environments, TrackMe's REST API logging lacked explicit handler definitions. This could result in logging conflicts where TrackMe logs contain unrelated external logs, or TrackMe logs are leaked into external logs.
#1015	Hosts Tracking (splk-dhm) - regression in host-level latency/delay	A regression introduced in version 2.1.11 prevented proper persistence and handling of host-level latency and delay thresholds in *splk-dhm* monitoring logic.

11.1.9 Version 2.1.13 - build 1744663474 (14/04/2025)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.0.x and earlier.
- The last release compatible with Splunk 9.0.x is TrackMe 2.0.99

- SHA256: 60a9959a9779c3f62420eec025368a156a46b78692209b57957714d99193da53

Table 15: Fixed Issues

Issue Number	Issue Description	Issue Details
#1009	StateFul Alerting - ingest only does not generate stateful events	Due to a bug in the alert action, ingest-only mode does not function as intended and fails to generate stateful events.
#1010	StateFul Alerting - Alias field is missing	The *alias* field is not included in the stateful events, which may impact downstream logic or integration relying on this metadata.
#1011	StateFul Alerting - local MTA triggers configuration exception	Systems using a local MTA (mail transfer agent) for email delivery encountered an exception due to missing *trackme_emails* configuration. This update ensures the fallback to localhost config is handled properly.
#1012	StateFul Alerting - configurable options for local MTA	Added new system-level options in the General tab to support sender address and other configurations when using the localhost MTA for email delivery.

11.1.10 Version 2.1.12 - build 1744327446 (11/04/2025)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.0.x and earlier.
- The last release compatible with Splunk 9.0.x is TrackMe 2.0.99

- SHA256: 16eb3f5bc532b09860006c3faf7741939a8ae1d3680095e749e4d3d15368b269

Table 16: Fixed Issues

Issue Number	Issue Description	Issue Details
#1008	StateFul Alerting - Splunk 9.1/9.2 Python 3.7 compatibility	StateFul Alerting introduced in 2.1.11 fails in Splunk 9.1.x and 9.2.x due to import errors caused by Python 3.7. This fix ensures the StateFul feature no longer crashes and gracefully disables embedded charts if Pygal cannot be loaded. Note: Emails charts are not available in Splunk 9.1/9.2 environments.

11.1.11 Version 2.1.11 - build 1744200878 (09/04/2025)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.0.x and earlier.
- The last release compatible with Splunk 9.0.x is TrackMe 2.0.99

- SHA256: 0122ea5cd0ed73a17562f24a12cb3c8808a75501d7021d70b07c37bf8970494c

Table 17: Fixed Issues

Issue Number	Issue Description	Issue Details
#969	Minor typos in config/help descriptions	Fix various typos or grammatical issues in TrackMe.
#974	Audit Data Sampling dashboard token bug	Hard coded value in a main search for the tenant_id token replacement.
#978	Schema upgrade fails if alerts were manually deleted	Customers experienced issues upgrading to 2.1.10 if alerts were manually deleted.
#979	Restore alert actions not re-enabled	Issue in re-enabling alert actions due to improper "actions" formatting.
#981	SLA regex input issue	UCC config doesn't allow editing SLA classes due to regex validation bug.
#982	UI filters in Virtual Tenant view	KO filtering breaks navigation when no results exist.
#986	SVC metrics now in _cmc_summary	Splunk Cloud index change; update to TrackMe logic accordingly.
#988	Ack expiry bug for future_over_tolerance	Incorrectly treated as anomaly reason change, prematurely expiring Acks.
#991	0d auto-disablement disables all entities	Logic flaw disables all entities when 0d is defined.
#993	Transforms backup fails if admin renamed	Hardcoded admin username in REST call macro.
#990	Flex group rename escaping issue	Problem handling double quotes in remote contexts.
#998	Incorrect export for timeline viz	Should be "none" instead of "app" in default.meta.
#1001	Disabled ranges ignored in delayed entity inspector	Setting range to 0 doesn't disable it properly in backend.
#1002	Default entity priority dropdown ignored	Affects Flex/Workload/CIM Virtual Tenant creation.
#1003	UC large lookup file inconsistent timeout	max_sec_inactive is 1h while schedule is every 12h.
#1004	ML Outliers only renders one model	Bug prevents rendering of multiple models per entity.
#1005	ML tab missing in splk-cim	Outliers tab missing in UI even if ML is enabled.
#1007	Adaptive delay causes config loss	Auto-updates override non-delay settings like max latency.

Table 18: Enhancements and New Features

Issue Number	Feature Description	Feature Details
#963	Health Tracker alert consistency check	New task to check for manually removed alerts and purge stale entries.
#964	Schema Upgrade resilience	Prevents failure if alerts were removed outside of TrackMe.
#962	Handlers event notifications system	Track handler actions and health status in a new tab per entity.
#961	Exclude KVstores from backups	Backup exclusion list for KVstore collections.
#965	Exclude KVstore/KO from restores	Restore blocklist for collections and knowledge objects.
#966	Logging deprecation fix	Address deprecated log.setLevel usage.
#967	Virtual Tenant creation UI refresh	CSS and label improvements.
#968	Modal UI enhancements	Ensure all modals have a header close button.
#970	SOAR failure use case optimizations	Performance improvements for high-scale environments.
#971	Prevent browser grammar on Flex inputs	Disable grammar checking on code inputs.
#972	SOAR Broker status enhancements	Status red triggered if not active.
#975	Tracker health consistency tasks	Checks existence of trackers and KO references.
#976	Constraint inputs UI improvement	Use textarea for better experience.
#983	Decision Maker message clarity	Better status messages to reduce confusion.
#985	Alert modal UI refresh	Simplified and improved per-alert modal design.
#987	Host thresholds precedence	Host-level thresholds now override sourcetypes.
#980	Stateful alerting and email threading	Full-featured incident lifecycle with HTML email and embedded charts.
#989	Sticky Acks default	Sticky Ack now default based on user preference.
#992	SOAR concurrent playbooks tracking	REST API and Flex tracker to monitor playbooks live.
#994	Info/SPL/success UI consistency	Prevent UI messages from rendering off-screen.
#995	SLA breach event throttling	Configurable generation frequency for SLA breach events.
#996	Virtual Tenant auto-repair	Automatically detects and fixes upgrade failures.
#997	Update Splunk Remote Account URLs	New admin API endpoint to update *splunk_url*.
#999	Improved rendering for search/inspector menus	Better UI handling of embedded Splunk actions.
#1000	Remote Search performance metrics	Query each SH member, return response time and job count.
#1006	Reduce frequency of delayed entity inspector	Less frequent checks to reduce resource costs.

11.1.12 Version 2.1.10 - build 1741820646 (12/03/2025)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.0.x and earlier.
- The last release compatible with Splunk 9.0.x is TrackMe 2.0.99

- SHA256: 1503eff8cd5b8864a30ea9bc04c6518c2ac96d14508f192256cfc3be76fb0930

Table 19: Fixed Issues

Issue Number	Issue Description	Issue Details
#909	UI Virtual Tenant - Remove Deprecated Color Option	The *color_theme_red_default* option, replaced by *color_theme_alert_default* in 2.1.9, was still available and has now been removed.
#910	TrackMe Outliers Detection - ML Model Creation Failure	Fixed an issue where new ML models could not be created if no previous models existed for an entity.
#911	TrackMe Outliers Detection - *auto_correct* Not Defined	Resolved an issue where *auto_correct* was not being set when adding a new model for Flex Object trackers via the UI.
#915	TrackMe Health Tracker - Missing Tenant Account Handling	Fixed an issue where the health tracker would not recreate missing Virtual Tenant accounts due to a missing *force_create_missing* parameter.
#916	TrackMe REST API - Incorrect *tenant_default_priority* Check	Fixed an issue where *tenant_default_priority* was incorrectly validated as *default_priority*, causing API failures.
#918	Hybrid Trackers - Inconsistent *time_window* Scheduling	Addressed inconsistencies in how TrackMe schedules *time_window* for hybrid trackers, ensuring uniform behavior.
#919	Virtual Tenant UI - Drag-and-Drop Issue	Fixed a UI issue where dragging Virtual Tenant boxes did not work correctly when multiple rows were present.
#920	CIM Tracking - Exception Handling for ML Detection	Fixed a Python assignment error that could occur when an ML anomaly was detected in CIM tracking.
#922	Data Sampling UI - *relative_time_window_seconds* Not Honored	Fixed an issue where user-defined *relative_time_window_seconds* values were not being correctly applied in the UI.
#923	Data Sampling Backend - *relative_time_window_seconds* Not Applied	Fixed an issue where the backend (*trackmesamplingexecutor* command) ignored user-defined *relative_time_window_seconds*.
#924	Remote Accounts - Token Rotation Failure	Fixed a token rotation failure due to a hardcoded service account name in the REST API.
#933	CIM Compliance - Incorrect Outlier Field Count	Fixed an issue where green and red field counts were not displayed correctly in the UI when ML Outliers were detected.
#934	ML Outliers - Training Failure When Deleting Old Models	Resolved a failure in ML Outliers training where old models were not being properly deleted before retraining.
#935	ML Outliers - UI Chart Rendering Issue	Fixed an issue where ML Outliers charts would not load unless the page was manually refreshed.
#940	Batch Priority Updates - Missing Action Field	Ensured that *generic_batch_update* always includes an *action* field in responses for improved debugging.
#941	CIM Compliance - Duplicate JSON Download Requests	Fixed an issue where downloading a CIM JSON configuration triggered duplicate requests.
#946	Virtual Tenants UI - Status Preview for Flex Objects	Improved Flex Object preview to display *status_description_short* and allow right-click actions in the UI.
#949	Acknowledgment Tracker - Expiration Failure for Removed Components	Fixed an issue where expired acknowledgments could not be processed for components that had been removed.
#951	Virtual Tenant Component Deletion - Associated Alerts Not Removed	Ensured that when deleting a component, any associated alerts are also removed automatically.
#956	Python Backend - Improper Logging Restoration	Fixed an issue where TrackMe's Python metric generation libraries failed to restore proper logging in some conditions.
#957	jQuery Scan Failure in Splunk URA App	Addressed a jQuery scan failure in Splunk URA caused by the deprecated Splunk-built timeline app.

Table 20: Enhancements, New Features, and Changes

Issue Number	Issue Description	Issue Details
#912	Virtual Tenants and Home UI - Enhanced Look and Feel	Improved TrackMe UI with modernized visuals and better handling of modal screens, especially for low-resolution contexts.
#913	Flex Objects - Group Renaming Capabilities	Added ability to rename Flex Object groups, automatically updating associated trackers, migrating entities, and preserving historical metrics.
#917	REST API - Remove Unnecessary Logging for Global Index Settings	Prevented unnecessary error messages when a Virtual Tenant uses "global" index settings, reducing redundant log entries.
#921	Data Sampling Dashboard Enhancements	Improved out-of-the-box Data Sampling review dashboard for better usability and insights.
#925	Outliers Detection - Default *kpi_span* Control	Introduced a system-wide option to define the default *kpi_span* value for newly created ML models.
#926	*trackmegenjsonmetrics* Command - Enhancements	Improved command to support automatic recognition of numerical values and prefix-based modifications for better metric handling.
#927	Flex Objects UI - Pre-Filtering Entities in Metric Views	Added a pre-filter text input to help users refine selections when adding entities to metric charts.
#928	Flex Object - Improved Splunk Cluster Global Status Use Case	Enhanced the cluster monitoring use case to consider additional health factors such as replication, site factors, and active bucket fixing.
#929	Flex Object - Filter Valid OOTB Use Cases	Ensured only valid out-of-the-box use cases are displayed, filtering out outdated or removed configurations.
#930	SOAR with Flex Objects - Refactored Playbook and Adhoc Error Detection	Redesigned detection for SOAR playbook and adhoc failures, adding REST lookups to extract associated metadata.
#931	TrackMe Theming - Match Skipping/Degraded Colors	Adjusted color scheme to ensure consistency with TrackMe's global theming.
#932	ML Outliers Detection - Bulk Rule Updates	Introduced a REST API endpoint for bulk updates to ML outlier detection rules, integrated into the UI.
#936	Flex Objects - Split License Usage Volume by Deployment Type	Separated license usage tracking for Splunk Cloud and Splunk Enterprise due to differences in pool concepts.
#937	Priority Management - Identify Policy/External Control	Added *priority_reason* field to clarify if an entity's priority is controlled by policies or external sources.
#938	SOAR Integration - Prevent ES8 Conflict	Renamed *soar.py* to avoid conflicts with embedded libraries in Splunk Enterprise Security 8.
#939	Enhanced Priority Management - Policy and External Overrides	Improved UI messaging and per-entity override capabilities when policies or external systems manage priorities.
#942	Virtual Tenant - Control Machine Learning Activation	Introduced *mloutliers_allowlist* parameter to selectively enable ML outlier detection per component.
#943	Virtual Tenants - Improved Creation UI	Enhanced tenant creation UI with clearer labels and real-time feedback.
#945	Flex Objects - Converging Multi-Dimensional KPI	Introduced a KPI *pct_availability* to aggregate multiple Flex entities into a single service availability metric.
#947	License Verification - Prevent Incorrect Actions on KVstore Failure	Ensured TrackMe does not take license-related actions if a KVstore exception prevents verification.
#950	TrackMe Acknowledgements - Preselect *ack_type*	Defaulted *ack_type* dropdown to match the existing acknowledgment record for an entity.
#952	UI Enhancements - Improved Legend Styling	Adjusted legend design for better readability.
#954	Flex Objects - Deduplication for Metric Ingestion	Added deduplication capabilities to prevent duplicate metrics by tracking the last seen epoch in

11.1.13 Version 2.1.9 - build 1738934449 (07/02/2025)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.0.x and earlier.
- The last release compatible with Splunk 9.0.x is TrackMe 2.0.99

- SHA256: 85f83d14bb98010e9fe1f2f56989b4daf085a118367f587eae9522bf7bd03ac8

Table 21: Fixed Issues

Issue Number	Issue Description	Issue Details
issue#895	Flex Object - Ensure *status_description_short* Always Has Value	Fixed an issue where *status_description_short* could be empty under rare conditions, particularly when *max_sec_inactive* is 0 and the entity has not been handled by the inactive entities tracker.
issue#897	CIM Compliance Tracking - Flipping Event Generation Errors	Fixed a Python error causing failures in generating flipping events under specific conditions.
issue#905	SOAR Tracking - Automation Brokers HA Issues with Large Asset Volumes	Resolved REST API pagination issues that impacted SOAR Automation Brokers tracking and high availability features when a large number of assets were configured.

Table 22: Enhancements, New Features, and Changes

Issue Number	Issue Description	Issue Details
issue#894	Flex Object/Workload - Allow Disabling Inactive Entity Purging	Updated *trackmesplkflxinactiveinspector* and *trackmesplkwlkinactiveinspector* commands to allow *0* as a valid argument, disabling entity purging.
issue#896	Trackers - Inherit Earliest/Latest and Control Alert Behavior	Removed the explicit need for *earliest* and *latest* arguments, allowing them to inherit from metadata variables unless specified. Introduced an argument to control whether an empty result set should impact Virtual Tenant operational status.
issue#898	Data Source Monitoring - Real-time Entity Status Refresh	Enabled real-time updates to entity status for Data Sampling (*splk-dsm*).
issue#899	Health Tracker - Improved Logging and Performance	Enhanced logging for *trackme:health* events, including ACL details and additional metadata. Improved performance using *requests.session* to reduce REST overhead.
issue#900	Hybrid Tracker Wizard - Clarify Optional Burn Benchmark Tasks	Explicitly marked burn benchmark tasks as optional in the Hybrid Tracker creation wizard to reduce user confusion.
issue#901	Flex Objects - Multi-Entity Selection for Chart Generation	Added support for selecting multiple entities within the same Virtual Tenant when generating charts for specific metrics.
issue#902	Flex Object Trackers - UI Enhancements with Single Stats Visualization	Added single stat visualizations (Perc95, Max, Avg) to the Flex entity modal views for improved usability.
issue#904	Cribl Monitoring - Enhanced Status Descriptions	Added *status_description_short* field to all Cribl monitoring use cases for improved user experience.
issue#906	Virtual Tenant UI - Improved Alert Visualization	Updated the Virtual Tenant UI with an enhanced color scheme for better visualization of high and critical priority entities in alert states.
issue#907	Home UI - Modernized Alert Priorities Display	Improved the TrackMe Home UI with a modernized look and feel for managing single views associated with alert priorities and alert counts.

11.1.14 Version 2.1.8 - build 1738021014 (27/01/2025)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.0.x and earlier.
- The last release compatible with Splunk 9.0.x is TrackMe 2.0.99

- SHA256: fb323b1bc0c529ffcb1342dbd6c0146d8f5b20e8fcbbbf7c06719801ef759f46

Table 23: Fixed Issues

Issue Number	Issue Description	Issue Details
issue#870	Notable Events - Drilldown Link Ignores Custom Web Root Endpoint	Fixed an issue where drilldown links in notable events did not preserve additional URL path parts when using a custom Splunk Web root endpoint.
issue#872	Home UI - Refresh Button Failure with Custom Splunk Web Root	Fixed an issue where the refresh button failed due to an invalid REST endpoint definition when using a custom Splunk Web root endpoint.
issue#873	Audit System - Missing Tenant ID in Audit Events	Resolved missing *tenant_id* values in some audit events, ensuring they are properly displayed in the entity audit tab.
issue#874	AppInspect Failure - Inline Comment Parsing Issue	Addressed an issue where the AppInspect check *check_collections_conf_for_specified_name_field_type* incorrectly failed due to inline comments in configurations.
issue#875	Notable Alerts - Missing Tenant ID Filter	Fixed an issue where notable alerts did not include *tenant_id* as a search filter, preventing cross-notable events in multi-tenant environments.
issue#876	Prevent splunkd Startup Error from Timeline Viz Integration	Addressed splunkd startup error messages caused by missing README spec instructions in the timeline visualization integration.
issue#881	REST API - Missing Double Quotes in SPL Import/Export Examples	Fixed missing double quotes in resource examples for SPL import/export.
issue#882	Tags Policies - Updating Policy ID Causes 404 Error	Resolved an issue where updating the policy ID from the UI led to a 404 error due to incorrect KVstore record retrieval logic.
issue#884	SLA & Priority Policies - Policy ID Update Causes 404	Fixed a similar issue affecting SLA and Priority policy updates in the UI.
issue#886	Adaptive Delay Tracker - Default Search Time Issue	Fixed an issue where the recent activity search in the Adaptive Delay Tracker unexpectedly ran over all time due to a missing earliest time argument.
issue#889	Workload (splk-wlk) - SmartStatus searches for errors tracking	The generated search for errors (function smartstatus_investigations_uc_wlk_execution_errors) should not use indexed earliest and latest.
issue#890	TrackMe Dashboards - Errors loading TrackMe logo	Fixed an issue where TrackMe dashboards failed to load the TrackMe logo due to incorrect path resolution.
issue#892	Week days monitoring - (all components except splk-dsm) - Week days monitoring by the day selection issues	Different UI and API issues have been identified for all components, except splk-dsm.

Table 24: Enhancements, Changes, and New Features

Issue Number	Issue Description	Issue Details
issue#871	Splunk SOAR - High Availability & Monitoring Enhancements	Introduced a pool definition capability for automation brokers, improving flexibility over active/passive configurations. Decommissioned the monitor use case in favor of a unified Flex Object use case.
issue#854	Drilldown Links - Preset Time Range Support	Added support for influencing preset time ranges in entity views via drilldown links. Allows relative time formats such as *-24h* or *24h*.
issue#877	Notable Events - Per Alert Default Earliest Time	Introduced a *drilldown_earliest* option for defining a per-alert default earliest time in notable event drilldowns.
issue#878	Timechart Span Definition Enhancements	Introduced a new macro for automated span value definition in timechart calls, improving reliability over UI-based span determination.
issue#880	Outliers Detection - Default Latest Time Adjustment	Changed the default *latest* value for ML training to *-1d* instead of *now* to prevent abnormal periods from affecting training.
issue#879	Outliers Detection - Handling Feed Interruptions	Updated outliers detection for *splk-dsm/dhm* to generate *0* value metrics, ensuring feed interruptions are properly accounted for in detection processes.
issue#883	Flex Objects - Inactive State Management Improvements	Improved inactive state detection, real-time status updates, and enhanced anomaly reasoning in *splk-flx*.
issue#885	Splunk Remote Account Management - Bearer Token Auto-Rotation	Introduced automatic bearer token rotation for Splunk remote accounts. Configurable per account with a default rotation interval of 7 days.
issue#887	Flex Object use cases library - Splunk introspection CPU & Memory Monitoring	Enhancement to these use cases with the definition of the status short description field.
issue#888	Flex Object use cases library - DataModel Acceleration Monitoring	Enhancement to the use case the definition of the status short description field.
issue#891	TrackMe dashboards - Refresh for Dashboard Studio based dashboards	Syntax and source code refresh for out of date dashboard studio based dashboards.

11.1.15 Version 2.1.7 - build 1735907247 (03/01/2025)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.1.0 and earlier.
- The last release compatible with Splunk 9.0.x up to 9.1.0 is TrackMe 2.0.99

- SHA256: 3fc504463e4270fd8142ad0bff0c32c46bf33fb329fa13b9d9a6e04f35cefa90

Table 25: Fixed Issues

Issue Number	Issue Description	Issue Details
issue#826	Dependency App Issues - Timeline App EOL by Splunk	The timeline visualization app has reached EOL as of December 24, 2024, and is no longer available on Splunk Base. TrackMe now incorporates these visualization components natively, removing the external dependency.

Table 26: Enhancements, Changes, and New Features

Issue Number	Issue Description	Issue Details
issue#865	Flex Object Library - Update for Cluster Management Use Cases	Updated Flex Object library use cases to replace legacy references to the Splunk master convention with modern terminology for cluster management.
issue#861	Bulk Edit Tags	Introduced a bulk edit UI capability for managing manual tags efficiently.
issue#866	Flex Object Library - Deployment Server Clients Tracking Refactor	Refactored the Splunk deployment server clients tracking use case to utilize the *_ds** indexes introduced in Splunk 9.3.x for better tracking capabilities.
issue#868	Flex Object Trackers Library Enhancements	Enhanced the Flex Object trackers library with additional attributes and improved status descriptions, including updates to: - *splk_splunk_enterprise_cluster_status* - *splk_splunk_enterprise_cluster_peers_status*
issue#867	Flex Object Trackers - Short Status Description Field	Added a *status_description_short* field for displaying concise status descriptions in the main Tabulator table while maintaining a detailed status description for additional insights. The logic now automatically handles this new field if not present in existing configurations.

11.1.16 Version 2.1.6 - build 1734273789 (15/12/2024)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.1.0 and earlier.
- The last release compatible with Splunk 9.0.x up to 9.1.0 is TrackMe 2.0.99

- SHA256: b9236c33c34fcdfb948629de3007ce874380e20f9f63029a658fcffba7930715

Table 27: Fixed Issues

Issue Number	Issue Description	Issue Details
issue#858	Knowledge Objects and Shortcuts Issues with Custom Splunk Web Root Endpoint	Resolved issues accessing knowledge objects and built-in shortcuts when using a custom Splunk Web root endpoint. Fixed drilldown link generation for TrackMe technical alerts and addressed JS errors in license registration and maintenance knowledge database notifications.
issue#858	Maintenance Mode Enablement Fails via UI	Fixed a regression in TrackMe 2.1.5 causing a 404 error when enabling maintenance mode through the UI.
issue#857	Backup & Restore Issues with Disabled Virtual Tenants	Resolved exceptions during backup operations when disabled Virtual Tenants exist. The restore process now forces purging of disabled tenants to prevent loading issues.
issue#858	Remote Search - Default HTTPS Port Handling	Improved handling of default HTTPS/443 port in remote Splunk REST API requests to avoid failures when the port is not explicitly provided in the URL.

11.1.17 Version 2.1.5 - build 1734004063 (12/12/2024)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.1.0 and earlier.
- The last release compatible with Splunk 9.0.x up to 9.1.0 is TrackMe 2.0.99

- SHA256: 2d3ccbe77a6929fc982f645378521fa56fff97a0d865b11642918f7c8c252eb4

Table 28: Fixed Issues

Issue Number	Issue Description	Issue Details
issue#834	System Level Preferences Misconfiguration	Home UI system level preferences were not properly applied, with Virtual Tenant UI preferences taking precedence. Fixed to honor distinct system level preferences for Home and Virtual Tenant UIs.
issue#837	Virtual Tenants - Scheduled Search Count Issue	Show Knowledge Objects screen incorrectly displayed a count of 0 for scheduled searches due to misparsed *is_scheduled* field.
issue#838	Get Knowledge Objects - JSON Parsing Error	The macro *get_tenants_reports* did not escape double quotes in macro definitions, leading to invalid JSON structures.
issue#839	Data Sampling Migration Issue	Schema migration from TrackMe versions < 2.0.36 directly to 2.1.x failed for the Data Sampling Tracker. A workaround migration path is recommended until resolved in version 2.1.5.
issue#840	Schema Upgrade Errors	Various errors during schema upgrade tasks (2084, 2064, 2099) resolved for migrations from TrackMe 2.0.x to 2.1.x.
issue#845	Alerts Creation - JSON Key Duplication	Resolved an issue where duplicate keys were generated in JSON payload during alert creation.
issue#843	Data Sampling - Typo in Remediation Messages	Fixed typos in Data Sampling remediation messages.
issue#849	FIPS Transition Issues	Minor issues with SHA256 transition from MD5 in version_id metrics ingestion and flipping events identification for splk-wlk and splk-cim resolved.
issue#848	Flex Objects - Typo in Lookup Files Monitoring Use Case	Corrected typo in JSON file name and use case name for large lookup files monitoring.
issue#852	Modular Alerts Logs - Timezone Issue	Modular alerts logs now properly handle non-UTC timezone settings.
issue#853	Drilldown Error in Multi-Component Tenants	Resolved drilldown activation error for components not first in the tab list.
issue#855	Splunk Web Custom Root Endpoint Issue	Addressed compatibility issues with custom root endpoint configurations in *web.conf*.
issue#856	Workload Metadata Tracker - Minimal Privileges Issue	Improved handling of minimal privilege accounts in metadata tracker operations for splk-wlk.

Table 29: Enhancements, Changes, and New Features

Issue Number	Issue Description	Issue Details
issue#832	Splunk Python SDK Upgrade	Upgraded to the latest Splunk Python SDK version 2.1.0.
issue#833	Splunk UCC Validator Warning Messages	Ensured all UCC configurations include input validators to address warning messages introduced in UCC 5.52.0.
issue#835	Virtual Tenants Wizard - Custom Fields Enhancement	Added missing hover action for custom field configurations in the Virtual Tenants wizard.
issue#841	Splunk UCC Upgrade to 5.53.0	Upgraded Splunk UCC to version 5.53.0.
issue#842	Configuration Management Enhancement	Improved readability of TrackMe configuration screens by grouping configuration items using UCC 5.53.0 features.
issue#836	Backup & Restore Feature	Introduced capabilities to backup and restore knowledge objects and KVstore content for Virtual Tenants and components.
issue#847	Virtual Tenant API Enhancements	Improved input verification for tenant index settings to prevent misconfigurations.
issue#850	Flex Objects Tracker Name Simplification	Removed `my_` prefix from Flex Object tracker names.
issue#851	Flex Objects Tracker Name Sanitization	Enhanced sanitization for requested tracker names during simulation and creation.

11.1.18 Version 2.1.4 - build 1731085887 (08/10/2024)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.1.0 and earlier.
- The last release compatible with Splunk 9.0.x up to 9.1.0 is TrackMe 2.0.99

- SHA256: 46b4db465c4dafc67fdaffc534d629ad894a6ac47a67210d2eda7d508a427e9f

Table 30: Fixed Issues

Issue Number	Issue Description	Issue Details
issue#822	Inline Bulk Edit - Adaptive Delay Persistence Issue	The Adaptive delay setting, modified via inline bulk edit in the Tabulator, does not persist. This occurs due to the UI's restricted persistent fields, missing Adaptive delay, which should rely on TrackMe's Python library for centralized handling.
issue#823	Virtual Tenants - System Level Fallback Misconfiguration	The fallback configuration for *splk_feeds_auto_disablement_period* fails if Virtual Tenant lacks the expected definition. Issue impacts auto disablement period settings for inactive entities.
issue#825	Flex Objects - Error in Inactive Entities Tracker	Flex inactive entity tracking may fail if *max_sec_inactive* is undefined, caused by a Python code error when handling this missing value.
issue#830	Remote Search - Timeout Config Error	Remote search failures occur due to timeout values received as strings instead of integers. This fix ensures timeout values are processed correctly as integers.
issue#831	Data Sampling - Remote Entities issues	Data sampling issues with remote entities. This fix addresses various issues with Data Sampling v2 when entities are remote entities.

Table 31: Enhancements, Changes, and New Features

Issue Number	Issue Description	Issue Details
issue#818	Tabulator - JS Upgrade to v6.3.0	Upgraded the Tabulator JS component to version 6.3.0 for improved functionality and performance.
issue#819	Splunk UCC Upgrade to 5.52.0	Updated Splunk UCC from version 5.48.2 to 5.52.0.
issue#820	Virtual Tenants UI - Usability Improvements	Added quick access buttons for Scheduler and Ops Status, improved modal screens, and a more consistent layout with closing icons where needed.
issue#821	Flex/Workload Blocklist Extension	Blocklist features extended to splk-flx/wlk components, adding flexibility to block specific patterns with a schema upgrade to initialize allowlist collections.
issue#824	Virtual Tenant Account Management Enhancement	Implements a more secure approach for verifying and updating Virtual Tenant accounts, with auto-check and repair features through the Health Tracker.
issue#827	Flex Objects - Enhanced Cribl Logstream CPU Usage Detection	Improved Cribl Logstream CPU consumption use case for fewer false positives and clearer alerts.
issue#828	Flex Objects - New Use Case for Dynamic Sourcetypes	Introduces a new use case to detect and track dynamic sourcetypes in Splunk, optimizing log rotation handling.
issue#829	Data Sources Tracking - Hybrid Tracker Enhancements	Allows inclusion/exclusion of sourcetypes during hybrid tracker creation for splk-dsm, adding control over custom break-by fields.

11.1.19 Version 2.1.3 - build 1728629753 (11/10/2024)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.1.0 and earlier.
- The last release compatible with Splunk 9.0.x up to 9.1.0 is TrackMe 2.0.99

- SHA256: f33353510b450588df38976b465d87bf95f7614d9223c4a3348b08d48b95af1b

Table 32: Fixed Issues

Issue Number	Issue Description	Issue Details
issue#817	TrackMe Home UI - Regression with TrackMe 2.1.2	High priority regression due to a missed token validation for all components except splk-dsm.

11.1.20 Version 2.1.2 - build 1728540776 (10/10/2024)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.1.0 and earlier.
- The last release compatible with Splunk 9.0.x up to 9.1.0 is TrackMe 2.0.99

- SHA256: 361ea0ee5bd07584f96ebd8960af8f1ff6ac82d5f2b68b08ea45cae6f880e848

Table 33: Fixed Issues

Issue Number	Issue Description	Issue Details
issue#780	TrackMe Home UI - Workload / Flex / Metric hosts (splk-wlk/flx/mhm)	The entity screen incorrectly shows “host state” rather than “entity state” next to the state icon.
issue#781	Acknowledgement - Auto expiration of Acknowledgements based on condition changes	The Ack that auto raised can be expired by the Ack auto-management. If the anomaly reason is none, it will incorrectly expire the Ack.
issue#777	TrackMe Hybrid trackers - Missing Hybrid tracker from central collection	A new task ensures that the hybrid tracker KV collection remains consistent and fixes records if needed.
issue#782	Blocklists features - Unexpected additional dot in regex-based blocklists	Adding a regex blocklist with a wildcard results in an extra dot due to incorrect handling of non-regex blocklists.
issue#786	Data Source monitoring - Data sampling engine issues with future data	Earliest time can be after the latest, causing sampling searches to fail due to future data indexing.
issue#788	Home UI - Donut chart doesn’t show green state entities	A bug prevents green state entities from being represented in the top-right donut chart.
issue#789	Elastic Sources - Background task for entity count refresh not called	The Shared Elastic tracker does not call the method to refresh entity count, leading to incorrect data.
issue#790	Elastic Sources - mstats-based searches don’t generate expected dcount host metrics	An SPL field name prevents expected host distinct count generation.
issue#786	TrackMe Health Tracker - Incorrect warning for tags tracker	Health Tracker generates incorrect warnings due to hard-coded DSM component definition in tags tracker.
issue#802	API Documentation & Reference - Incorrect API reference examples for tags policies management	Examples in the documentation are missing the component argument.
issue#804	Persistence issue for entities with backslashes	Backslashes in entity names cause issues with persistence of settings like priority and lagging thresholds.
issue#808	Data Sampling & Events format recognition - Confusing regex simulation message	When regex does not match any events, the result message is confusing and should clearly indicate 0% match condition.

Table 34: Enhancements, Changes, and New Features

Issue Number	Issue Description	Issue Details
issue#783	Blocklist management for splk-feeds - Various improvements	Enhanced management screen, added comment storage, and background entity count updates for blocklists.
issue#784	TrackMe Virtual Tenants UI licence information & Schema Upgrade	Added clickable TrackMe version display, schema version info, and shortcut to manage licence and upgrades.
issue#785	Virtual Tenants UI - Notify messages for quick action buttons	Ensured all buttons in the Virtual Tenants screen show notify messages when hovered.
issue#787	Virtual Tenants UI - Embedded Entities Overview Tabulator	Introduced a single-pane Entities Overview in Virtual Tenants for easier navigation.
issue#791	TrackMe Home UI - Quick access to reports from Elastic Dedicated screen	Added quick Splunk Web access for reports from Elastic Sources management UI.
issue#793	Data Sources monitoring - Tenant level control of Data Sampling	Added an option to enable/disable Data Sampling for Virtual Tenants, hiding UI functions accordingly.
issue#792	TrackMe Home UI - Hiding adaptive_delay feature when disabled	Automatically hides adaptive_delay-related UI elements when the feature is disabled.
issue#795	Elastic Sources - Support for mpreview-based searches	Added mpreview-based searches as a replacement for mtstats, providing true metrics count reporting.
issue#796	Elastic Sources wizard - Presets for earliest/latest based on search type	Automatically presets recommended earliest/latest times based on the type of search.
issue#797	TrackMe Health Tracker - Auto fix duplicated entities	Automatically detects and fixes duplicated entities in all components.
issue#798	Cribl Logstream monitoring - CPU usage metrics	Added CPU usage metrics, including time spent in green/red states, to TrackMe metrics indexes.
issue#799	Virtual Tenants UI - UX enhancement for Tenants Ops view	Improved user experience with status selectors, quick access to reports, and logs in the Tenants Ops view.
issue#800	Virtual Tenants & Home UI - Tabulator sort header improvement	Removed the sort header for fields where sorting is not meaningful in Tabulator.
issue#801	Virtual Tenants UI - Scheduler overview enhancement	Replaced Splunk table with a tabulator view for the scheduler overview with quick actions.
issue#803	Splunk Remote Search - Configurable timeouts for remote accounts	Added per-account configurable timeouts for connection and search in Splunk Remote Search.
issue#805	ML Outliers - Minor log improvements	Enhanced the quality of logs generated by ML outlier detection.
issue#807	Hybrid Trackers - Cron schedule validation	Added cron schedule validity checks using croniter library for all scheduled logic.
issue#809	Data Sampling & Events format recognition - Python code improvements	Improved Python code quality and safer behavior for the Data Sampling engine.
issue#810	Data Sampling & Events format recognition - Entity settings overview	Added a dynamic entity settings overview in JSON format within the Data Sampling UI screen.
issue#811	Bulk edits & Audit logging - New audit format for bulk edits	Refactored bulk edit function to track changes per field, improving audit logging.
issue#812	TrackMe Audit subsystem - Mass audit REST call improvements	Switched to mass audit REST calls for better performance and flexibility in the Audit subsystem.
issue#813	TrackMe events - Consistent event_id convention	Standardized event_id across all TrackMe-generated events using sha256 hash.
issue#814	TrackMe Home UI - Allows selecting visible tabs and their or-	This feature adds a new parameter in the Virtual Tenant account, to control the order and visibility

11.1.21 Version 2.1.1 - build 1726614488 (18/09/2024)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.1.0 and earlier.
- The last release compatible with Splunk 9.0.x up to 9.1.0 is TrackMe 2.0.99

Introducing TrackMe's data sampling events format recognition v2!

- TrackMe 2.1.0 welcomes the introduction of a brand new version of the events format and recognition for data sampling. (Data quality inspection, PII tracking, and more)
- This is major change and improvements in the way TrackMe handles data sampling events, and will allow for slightly more flexibility and control over the data sampling process.
- For more information about the engine v2 and its capabilities, see the admin guide: *TrackMe Data Sampling - Events and Format Recognition for Quality Inspection in TrackMe*

- SHA256: eb750290ecf39e926fe7e6528de8fbfdac8f078f9a6922cca7b0167a69cc4f0d

Fixed issues:

- **trackme-limited/trackme-report-issues#769 - bug - TrackMe Data Sampling UI - links to pre-built KPI metrics search generate an incorrect earliest time**
  - In the data sampling, several quick access buttons allow generating automated KPIs metric searches in a new blank tab.
  - However, due to a Javascript bug, an incorrect pattern is unexpectedly added to the earliest time.
- **trackme-limited/trackme-report-issues#770 - bug - Virtual Tenant UI & REST API - Splunk Knowledge Objects explorer and associated endpoint generated an invalid JSON, preventing the UI formatting to work as expected**
  - The Virtual Tenant UI screen for knowledge object access should generate a JSON pretty printed of the properties field.
  - However, the macro called by the endpoint generates an invalid JSON, and the REST API endpoint should better handle the parsing too.
- **trackme-limited/trackme-report-issues#774 - bug - Flex Objects & Workload - trackmesplkflxinactiveinspector/trackmesplkwlkinactiveinspector custom commands are designed to handle inactive entities purge, however the process is not currently working as intended**
  - These commands are designed notably to purge entities which have been inactive for too long, after the configured period in days passed as an argument to the commands.
  - However, this particular action does not work as intended currently, and purge of entities is not currently effective.
  - This fix addresses these issues and also slightly improves the code quality and logging of these commands.
- **trackme-limited/trackme-report-issues#779 - bug - SLA tracking - When entities status change, the SLA status will temporary be inconsistent and will show an incorrect time until trackers have reflected the real time change in the KVstore**
  - When a given entity status changes from one to another, the SLA status is temporarily inconsistent due to the discrepancy between the real time status provided by the Decision

Maker and the fact that the KVstore object_state is yet to be updated.

- With this fix, the SLA status takes into account both values, and will show a specific SLA refresh pending message, and the SLA status and timer will wait for the KVstore to be updated accordingly to avoid any inconsistency.

Enhancements, changes and new features:

- **trackme-limited/trackme-report-issues#771 enhancement - Virtual Tenant UI - Tenants Splunk Knowledge Objects explorer screen - Add quick actions button for the Tabulator**
  - This enhancement adds various quick action buttons to filter out the Tabulator content against savedsearches only, macros only, and so forth
- **trackme-limited/trackme-report-issues#772 - enhancement - Notify bar lookup & feel - Responsive design and modern look & feel for the notification bar in TrackMe**
  - This enhancement is a major refresh of the responsive notification bar plugin in TrackMe.
  - The provides a responsive look & feel notification bar to TrackMe and a much improved and more modern appearance.
- **trackme-limited/trackme-report-issues#773 - enhancement - TrackMe Home UI - Add quick Splunk Web access to reports & macros from the Hybrid Tracker**
  - This enhancement adds quick access in Splunk Web for reports & macros through the Hybrid trackers management UI for all components
- **trackme-limited/trackme-report-issues#775 - change - Python - Addressing the deprecation of calling log.setLevel with logging.getLevelName when defining the current logging level in the various Python backends**
  - This change addresses a Python deprecation of calling the method logging.getLevelName within log.setLevel
- **trackme-limited/trackme-report-issues#778 - enhancement - In Virtual Tenants & Home UI, show the number of enabled entities rather then total number of entities**
  - This enhancement updates the number of entities primarily shown in the Virtual Tenants UI, as well as the Home UI and the left single view, so that we show the total number of enabled entities, instead of the total number entities active + inactive.
  - This provides more valuable information as well as better clarity in TrackMe.

11.1.22 Version 2.1.0 - build 1726088942 (11/09/2024)

Hint

For Splunk 9.1.1 and later

- From TrackMe 2.1.x, there is no more compatibility with Splunk 9.1.0 and earlier.
- The last release compatible with Splunk 9.0.x up to 9.1.0 is TrackMe 2.0.99

Introducing TrackMe's data sampling events format recognition v2!

- TrackMe 2.1.0 welcomes the introduction of a brand new version of the events format and recognition for data sampling. (Data quality inspection, PII tracking, and more)
- This is major change and improvements in the way TrackMe handles data sampling events, and will allow for slightly more flexibility and control over the data sampling process.

- For more information about the engine v2 and its capabilities, see the admin guide: *TrackMe Data Sampling - Events and Format Recognition for Quality Inspection in TrackMe*

- SHA256: 2a79af2a363bf1d10beb2f51f8c3f1334298d046015d4a23d1197c14ae5572c9

Fixed issues:

- **trackme-limited/trackme-report-issues#757 - bug - TrackMe schema migration from 2.0.97 and prior to 2.0.98 and latest should address the tags extension to splk-mhm**
  - In TrackMe 2.0.98, the tags features were normalised and extended to all components.
  - However, splk-mhm was not included in the list of eligible components leading to various issues in this components.
  - This change addresses the issue automatically, if the splk-mhm component was enabled, the tags extension will be managed during the schema upgrade.
- **trackme-limited/trackme-report-issues#761 - bug - TrackMe Health Tracker - subcontext="entities_auto_disablement" is attempting to perform a REST call per entity instead of a mass disablement operation, leading the tracker to eventually take an abnormal amount of time to be executed while failing to disable entities, and possibly generate skipping searches**
  - The Tenant health tracker runs a subcontext task called entities_auto_disablement, which is designed to automatically disable the monitoring state of entities that have not generated any data according to the system wide setting splk_general_feeds_auto_disablement_period.
  - However, a bug affecting this task incorrectly attempts to run a REST call per entity, instead of a mass REST call.
  - In some circumstances, this leads to an abnormal amount of run time for tracker and can cause skipping searches for the tracker
- **trackme-limited/trackme-report-issues#763 - bug - Typo in UI - Create Hybrid trackers**
  - This fixes a typo in the Home UI and the Virtual Tenant UI, and for the Hybrid tracker creation wizards
- **trackme-limited/trackme-report-issues#765 - bug - trackmesplkgetflipping can still be affected by a non expected missing object_category value in the KVstore record**
  - In some conditions, exceptions can still be encountered during the call of the streaming custom command trackmesplkgetflipping, leading to failures in properly handling the search logic. (especially in splk-dhm)
  - This update adds the object_category as an argument to the streaming custom command instead of getting this value from the records, the schema upgrade will process the update of all hybrid trackers wrapper search automatically so that the argument is called.
- **trackme-limited/trackme-report-issues#766 - bug - Transition to SHA256 based logic for FIPS compatibility mode in TrackMe 2.0.99 needs to be retro-applied on any tracker created by TrackMe**
  - Since TrackMe 2.0.99, we use sha256 instead of md5 to calculate the expected hash for TrackMe entities objects.
  - In some specific use cases, such as the Flex object for host tracking or some contexts in Workload, this change needs to be reflected on the search logic itself.
  - The schema upgrade will verify all trackers and automatically update any tracker if needed.
- **trackme-limited/trackme-report-issues#767 - bug - Virtual Tenants UI - When creating hybrid trackers during the Virtual Tenant wizard, tracker names should be a random combination rather than only the account name**

  - During the Virtual tenant creation, Hybrid trackers can be created per tenant/component, the hybrid tracker names should be a combination of "tracker-<random ID>" instead of just the account name.

- **trackme-limited/trackme-report-issues#768 - bug - TrackMe Backup REST API endpoints - avoids raising a file does not exist exception in some rase cases**
  - In some specific circumstances, the POST backup API endpoint can raise an exception if the expected file does not exist, this update simply avoids this condition.

Enhancement, changes and new features:

- **trackme-limited/trackme-report-issues#756 - feature - Data Source tracking - Introducing the Data Sampling events and format recognition engine v2**
  - TrackMe 2.1.0 welcomes the introduction of the engine v2 for the data sampling and events format recognition.
  - This is a completely rewritten engine in full Python, providing flexible and powerful capabilities for events quality inspection in TrackMe.
  - The new engine provides flexible options at the system wide level, which can be customised on a per entity basis, such as controlling the min time between sampling iterations, the number of events sampled per iteration (which is now 10K), the truncation of events when storing samples for investigation purposes, initial thresholds for min match inclusive percentage, and more!)
  - With the the engine v2, parsing of recognitions models is made against the whole event and is no longer limited due to truncation, events are no longer stored in the KVstore then processed, but processed then a sample of sampled events is stored in the KVstore per model matched and for review purposes.
  - The new engine introduces a brand new concept of major / minor models matching, allowing to tackle minor quality issues without generating non meaningful alerts, TrackMe admin can control the minimal threshold of acceptable percentage of events matching the main model.
  - Tracking Personally Identifiable Information (PII) can handle as many models as required (exclusive match)
  - The interfaces were rewritten so the data sampling feature can be controlled and reviewed more efficiently and with more capabilities.
  - KPIs generation from Data Sampling: The engine now generates KPIs in TrackMe's metrics models, so you can review over time the events matching percentage per model, the amount of events parsed and matched, as well as other KPIs such as the run time of the sampling operation per entity.
  - Many additional improvements were made in the data sampling engine v2!
- **trackme-limited/trackme-report-issues#758 - enhancement - TrackMe Schema upgrade - normalise the schema version to always use a 4 digits based logic, handling the patch version number**
  - This update ensures that TrackMe uses a consistent 4 digits based logic for the schema_version number, and handles notably the question of a new minor release and its associated patch number (ex: 2.1.0 versus 2.0.99)
- **trackme-limited/trackme-report-issues#759 - feature - TrackMe Notable events - Add a unique identifier in each TrackMe notable event**
  - Some customers may make use of a unique identifier in TrackMe notable events, especially to ensure notables have been consumed accordingly.
- **trackme-limited/trackme-report-issues#760 - feature - TrackMe Home UI & Tabulator - Add a Download button which allows downloading visible and filtered entities as a CSV file**

  - This new feature adds a Download button above the Tabulator table which allows quickly exporting visible and filtered entities as a CSV file for quick review out or data manipulation out of TrackMe

- **trackme-limited/trackme-report-issues#762 - enhancement - TrackMe Health Tracker - Logging and code improvements to allow easily monitoring the run_time taken by each task processed by the tracker**
  - This enhancement slightly improves the logging of the run_time taken by each task executed by the Health Tracker using a concept of task_instance_id associated with a task_name
  - A sample SPL:

```
` index=_internal sourcetype=trackme:custom_commands:trackmetrackerhealth
instance_id=* task_instance_id=* task=* run_time=* tenant_id=* | table _time
tenant_id instance_id task task_instance_id run_time _raw | sort 0 - _time `
```

- **trackme-limited/trackme-report-issues#764 - enhancement - TrackMe Schema Upgrade - before starting upgrade procedures, execute TrackMe's builtin backup**
  - With this enhancement, when TrackMe detects that migration procedures must be initiated, it will first query a TrackMe backup to be executed.

11.1.23 Version 2.0.99 - build 1723722498 (15/08/2024)

- SHA256: 6d7174da69a584a5dfdef160f2cb07410630db5b5bcf397aeb1956f83b037cd2

Additional notes about this release

- To address FIPS compatibility requirements, we have migrated from md5 to sha256 various TrackMe internal search logics.
- There are near no impacts to existing installations, however for customers using TrackMe Workload, you will notice that all monitored objects (Scheduled discovered) will generate a Metadata event, which normally happens only when a change in the search is detected.
- This behaviour is due to the change from md5 sum calculations to sha256 calculations for FIPS compatibility purposes, and can be safety ignored and acknowledged as part of the upgrade to TrackMe 2.0.99

Fixed issues:

- **trackme-limited/trackme-report-issues#738 - bug - TrackMe pagination - When using pagination mode = local, the pagination size is not submitted by the Tabulator and should default to size = 0 since the pagination is performed by the Tabulator rather than the server, which leads to missing records in high scale collections**
  - The default pagination mode is local rather than remote, however when using pagination = local, the default size should be 0 as the Tabulator will not submit this as an argument to the REST call to TrackMe.
  - This leads currently to missing records in the UI for high scale collections.

- **trackme-limited/trackme-report-issues#740 - bug - TrackMe Home UI - When opening an entity and if the tenant_id field of the Kvstore has an empty content unexpectedly, the UI fails to load the entity overview modal screen**
  - If in the tenant_id/component KVstore collection, the tenant_id field does not have a content, the UI fails to open the entity overview

- **trackme-limited/trackme-report-issues#741 - bug - trackmehealthtracker - logging reports the details of untracked entities for splk-dhm rather than the number of them**
  - for splk-dhm, the trackmehealthtracker should not report the detailed content of untracked entities, but how many of them were found instead.

- **trackme-limited/trackme-report-issues#743 - bug - Data Sampling for splk-dsm - Managing the Data sampling feature (enable/disable/run/reset) would fail for an entity which has not been processed at least once by the data sampling engine**
  - In the current release, managing the status of the data sampling feature can only happen if the data sampling has processed the entity at least once, and would fail otherwise.
  - This fix ensures that we properly manage the feature depending on the user request, no matter if the entity was processed already or not.
- **trackme-limited/trackme-report-issues#746 - bug - CIM compliance tracking (splk-cim) - When creating a new entity by cloning and if the CIM constraint contains one or more double quotes, the creation fails**
  - Creating a new entity by cloning for splk-cim fails if the CIM constraint contains double quotes
- **trackme-limited/trackme-report-issues#753 - bug - Outliers Anomaly detection - Workload (splk-wlk) - TrackMe should not attempt to train models for entities parts of applications that have been disabled in the Workload component**
  - In TrackMe's Workload component, entities can be enabled/disabled at the app level.
  - When an application is disabled, TrackMe should not attempt to consider training ML models.

Enhancements, changes and new features:

- **trackme-limited/trackme-report-issues#736 - feature - FIPS compatibility for TrackMe**
  - Splunk 9.3.x introduced several fixes which made Splunk really FIPS compatible, which disabled some crypto algo such as md5 which is used by TrackMe.
  - These developments allow TrackMe to be fully FIPS compatible, and FIPS validated from TrackMe Limited.
- **trackme-limited/trackme-report-issues#742 - enhancement - Data Sampling for splk-dsm - Improve the message and status returned when data sampling is disabled for a given entity to avoid any confusion**
  - When data sampling is disabled, the message shown in the UI, and returned underneath by the API endpoint, should be clearer to avoid any confusion.
- **trackme-limited/trackme-report-issues#737 - bug - Data Source tracking (splk-dsm) - Grouping issues with reduced pagination and entities for the same indexes that are split over multiple pages**
  - If using a reduced pagination size, entities relying on the same indexes can be split over multiple pages.
  - This fix may not entirely prevent this as this depends on the pagination size, but the addition of the index in the initial sort should limit the risk of this happening.
- **trackme-limited/trackme-report-issues#744 - enhancement - Health Tracker - Addition of tenant_id value verification in the record inspection steps**
  - This added step verifies that records of the main KVstore collection have a valid value for the field tenant_id, for consistency purposes.
- **trackme-limited/trackme-report-issues#745 - change - Virtual Tenant UI default settings - reducing the flex box default size from 374px to 350px**
  - This change updates the default flex box size in of the Virtual Tenants UI
- **trackme-limited/trackme-report-issues#747 - change - Data Hosts tracking (splk-dhm) - Change the default delay value from 3600 to 86400 seconds for Hosts Tracking**
  - In most use cases, it makes sense to increase the default delay value for Hosts tracking compared to Data source tracking.

  - Hosts tracking is a very different activity and most often, we need less restrictions when it comes to tracking the last time hosts have sent data as a default.

- **trackme-limited/trackme-report-issues#748 - bug - TrackMe Home UI - Flex Object creation screen can hide the bottom action buttons if the screen resolution is very low**
  - When creating a new Flex Object tracker, a very low screen resolution can prevent access to the bottom action buttons, unless zooming out from the Web Browser.

- **trackme-limited/trackme-report-issues#739 - feature - Flip events - Add a calculated disruption_time value in seconds within the flip results message when the entity switches from green to red**
  - When a given entity state changes from red to green, this changes adds a new calculated field disruption_time in seconds to ease further calculations of availability for users.
  - When the state are matching other conditions, the field has a 0 seconds value.

- **trackme-limited/trackme-report-issues#749 - enhancement - Data Sampling for Data Sources tracking - improve the message in the UI when sampling has not been processed yet**
  - When Data Sources tracking is pending and has not been processed yet, the message shown is simply N/A, and would deserve to be better explained.

- **trackme-limited/trackme-report-issues#750 - enhancement - TrackMe Health Tracker - Add a step to verify consistency regarding permanently deleted records**
  - When entities are permanently deleted, TrackMe stores records referencing these entities, so we will not discover these entities again.
  - In some circumstances such as restoring the KVstore collection, there could be a discrepancy due to the fact that entities are existing in the main KVstore collection, while in the same time listed as permanently deleted.
  - This additional verification ensures that if such a case happens, TrackMe would automatically purge associated records in the main KVstore.

- **trackme-limited/trackme-report-issues#751 - feature - Data Sources/Data Hosts tracking (splk-dsm/splk-dhm) - Add avg/max choices and additional choices with threshold in performance metrics tab**
  - This feature adds further more choices in the performing metrics screen, notably it adds choices between avg/max calculations at the time chart level, as well as options to include the current threshold for latency/delay.

- **trackme-limited/trackme-report-issues#752 - enhancement - Flex Objects library - Improve grouping for Splunk Cloud SVCs tracking use cases**
  - TrackMe has currently two builtin use cases for SVCs tracking in Splunk Cloud, the grouping should be improved so we dissociate global SVC tracking and per app SVC tracking.

- **trackme-limited/trackme-report-issues#754 - enhancement - Workload (splk-wlk)- When managing applications enablement, the tenant component summary should be refreshed**
  - In the workload component, TrackMe administrators can enable or disable entities at the application level.
  - When doing so, we should refresh the component summary as soon as possible, without waiting for the main tracker to perform it.

11.1.24 Version 2.0.98 - build 1722591315 (02/08/2024)

- SHA256: 371c327a8f492c07e57b60c8f8e505ecb8e9ba0aacca5d864ebfb31084612d26

Fixed issues:

- **trackme-limited/trackme-report-issues#700 - bug - SmartStatus alert action - Python exception in some circumstances when accessing the anomaly_reason**
  - The SmartStatus alert action can raise an exception while trying to investigate the anomaly_reason field.
- **trackme-limited/trackme-report-issues#702 - bug - Bulk edit - Critical priority button should be available in Bulk edit entities**
  - Critical priority was added in TrackMe 2.0.95, but in Bulk Edit we didn't add the associated button to mass update for the new priority.
- **trackme-limited/trackme-report-issues#705 - bug - Virtual Tenants UI - copy to clipboard TrackMe spl for Virtual Tenant creation can failed when executed due to boolean in JSON not properly handled**
  - When creating a new Virtual Tenant via the UI, one can at the end of the execution copy to clipboard the TrackMe SPL command that can be used to achieve the same creation in CLI.
  - However, an issue appears with the enablement of the component that remains in boolean and is not correctly handled in the SPL statement.
- **trackme-limited/trackme-report-issues#708 - bug - TrackMe Home UI Tabulator - regression due to trackme-limited/trackme-report-issues#697 for the management of encoded backslashes prevents the Alias to be inline editable**
  - A regression is affecting the Alias editable capability within the Tabulator due to the management of the encoded backslashes in issue#697.
  - This fix ensures the Alias is editable again within the Tabulator while still handling encoded backslashes.
- **trackme-limited/trackme-report-issues#710 - bug - Adaptive Delay (command trackmesplkadaptivedelay) - In some conditions, the backend tries to split a string into a list while already a list, raising a Python exception**
  - In the command trackmesplkadaptivedelay, we turn the anomaly_reason into a Python list from pipe separated, in some circumstances the field is already a list and the backend should check for the type of the object before applying the split.
- **trackme-limited/trackme-report-issues#719 - bug - TrackMe Notables and multi value fields in properties - mv fields should be properly handled in the properties, and stored as list within the JSON event**
  - When TrackMe generates a TrackMe notable event, the properties field contains all fields stored in the KVstore record for that entity.
  - Currently, multivalue fields are not correctly handled, and end in a pseudo multi value string structure instead.
- **trackme-limited/trackme-report-issues#725 - bug - Data Source tracking (splk-dsm) - Missing call to trackme_default_allow_adaptive_delay in the abstract macro called by the health tracker results in the field allow_adaptive to be empty in some conditions**
  - When the Health Tracker inspects offline entities (entities not actively generating data within the trackers scope), it shall call the macro that defines the default allow adaptive value.
- **trackme-limited/trackme-report-issues#727 - bug - SmartStatus - The use case search for future tolerance and the extraction of samples in the future is not consistent**
  - When SmartStatus is called and run the UC for data in the future (future over tolerance), one of the searches extracts a sample of events in the future.

  - The current search syntax is not ideally consistent and should be fixed for more meaningful results.

- **trackme-limited/trackme-report-issues#729 - bug - Feeds tracking (splk-dsm/dhm/mhm) - Auto-disablement of entities handled by the system wide configuration setting does not work as expected**
  - For feeds tracking, a system wide option was meant to allow automatically disabling the monitoring state of feeds tracking entities if the entity has not actively sent data since a certain period of time. (45 days by default)
  - However, the features is not properly working and does not influence the monitored state.
  - This change updates the process and transfer this work to the Virtual Tenant health tracker instead.
  - It fixes the action and enhances the worklfow by calling instead the associated API endpoint (rather than modifying silently the monitored_state), which also allows audting the change properly.
  - The period is also changed by default to 60 days, and the option is moved from General to splk-general for more consistency.
- **trackme-limited/trackme-report-issues#731 - bug - Machine Learning Outliers - Avoid generating an error with the command trackmesplkoutliersgetrules when dealing with Flex tracking that do not handle ML models**
  - Prevents generating an error message from this custom command and for Flex trackers that do not handle ML models.
- **trackme-limited/trackme-report-issues#732 - bug - TrackMe Home UI - Missing open in search for Notable events in the entitiy screen for all components**
  - When the mouse focus is on the Notable table in the entities screen, there should be an open in search option underneath the table.
- **trackme-limited/trackme-report-issues#734 - bug - TrackMe Home UI for splk-flx/splk-wlk - Machine Learning Outliers - In Adding model, the KPI dropdown selector does not populate properly**
  - When adding a new ML model for splk-flx/splk-wlk, the dropdown selector for the KPI selection does not populate due to a token generation issue.
- **trackme-limited/trackme-report-issues#735 - bug - TrackMe Home UI - regression in the tracking alert screen when clicking on a given alert to see the different charts, leading to none of the charts to be visible**
  - In the Home UI and the Tracking alerting tabs, a regression due due to a previous change (defining the default timerange via the Virtual Tenant account) leads to non of the charts to be visible when opening the activity of a given alert.

Enhancements, changes and new features:

- **trackme-limited/trackme-report-issues#699 - change - Splunk UCC 5.8 decommissioned placeholder in entity from globalConfig.json**
  - Splunk UCC 5.8 removed the placeholder option for entity, this change ensures compatibility for the current and future releases of Splunk UCC.
- **trackme-limited/trackme-report-issues#701 - change - Upgrade of moment.js to last version 2.30.1 - Appinspect warning**
  - Appinspect warning has raised a message regarding the moment.js lib that needs to be upgraded.
- **trackme-limited/trackme-report-issues#703 - feature - Bulk Edit - Allow handling all options for lagging policies via bulk edit for eligible components**

  - In Bulk edit, this evolution adds a section providing full control for lagging monitoring policies for splk-dsm/splk-dhm.

- **trackme-limited/trackme-report-issues#707 - feature - TrackMe Home UI - Add buttons to expand all / collapse all grouped items in the Tabulator JS**
  - This feature adds “Expand all” and “Collapse all” buttons to allow expanding or collapsing items per group in the Tabulator JS table for the TrackMe Home UI.

- **trackme-limited/trackme-report-issues#709 - feature - Flipping events - Log the previous anomaly_reason when generating flipping events**
  - When TrackMe detects a change in the status of an entity, it generates a flipping event.
  - With this evolution, TrackMe will also log the previous anomaly_reason in addition to the new anomaly_reason.

- **trackme-limited/trackme-report-issues#711 - feature - All components - Tabulator in Home UI and entities grouping - Allows controlling entities grouping by configuration**
  - This new feature allows controlling the Tabulator group at the level of the Virtual Tenant account.
  - With this control, you can define a list of custom fields of your choice for multi-level grouping, or you can use expressions to compose a custom field for the Tabulator grouping.
  - This evolution provides a number of quick actions buttons in the Home UI, so you can change the grouping temporarily, for instance such as grouping by anomaly reason or priority, and allows calls back the default grouping.

- **trackme-limited/trackme-report-issues#712 - change - Address Appinspect warning about splunk_resource_usage**
  - Appinspect reports a warning about: default/props.conf contains a [splunk_resource_usage] stanza.
  - This change addresses this warning which is due to a field alias for the purposes of the Workload component (splk-wlk).

- **trackme-limited/trackme-report-issues#713 - enhancement - Outliers Anomaly detection - Before attempting to render an Outlier model, verify the true existing of the model to avoid failing the search if the model is not yet ready**
  - This enhancement allows TrackMe to verify the true existence and readiness of a Machine Outliers model before attempting to process with the render search, avoiding to generate a failing search in the system.
  - In some circumstances, TrackMe may spawn rendering searches while the model is not yet ready, it has not been trained yet or the KPI underneath does not generate metric points, which results in the generation of a failing search from Splunk perspective.
  - This evolution prevents this situation by performing a true verification of the model readiness.

- **trackme-limited/trackme-report-issues#714 - change - Virtual Tenant creation - When creating a new Virtual Tenant and splk-dhm is the only enabled component, automatically disable ML Outliers at the Virtual Tenant account so it can be qualified for further enablement**
  - This change automatically disables ML Outliers detection feature at the Virtual Tenant account level, and when creating a new Virtual Tenant where splk-dhm is the only enabled tenant, so it can be decided later on to enable it or not.
  - The purpose of this change is to reduce system pressure for users that do not qualify enough TrackMe configuration, leading to very large ML models volume to handle.

- **trackme-limited/trackme-report-issues#715 - change - Data Hosts/Metric Hosts tracking (splk-dsm/splk-mhm) - Add a safety regarding the presence of object_category before calling the command trackmesplkgetflipping**

  - This change adds a safety feature at the Python level to ensure the presence of a valid value for the field object_category in the tracker process execution, and before it calls the streaming command trackmesplkgetflipping.
  - The objective is to avoid an unexpected condition that could lead the search to fail.

- **trackme-limited/trackme-report-issues#717 - feature - Extend and normalize the tags feature to all TrackMe components**
  - This extends the concept of tags, handled at the entity level and by policies, for all TrackMe components equally. (this was first released for splk-dsm)
  - Decommissioning the historical enrichment tag for splk-dhm/splk-mhm which were made redundant since we introduced the CMDB integration, and for consistency purposes.
  - After the upgrade, the schema upgrade will upgrade necessary objects and create new objects for newly eligible components, there are no interventions required.
- **trackme-limited/trackme-report-issues#718 - enhancement - TrackMe REST API - Improve the behavior when forcing the deletion of a Virtual Tenant via the del_tenant endpoint**
  - When calling the del_tenant API endpoint in force mode, we should systematically try to clean any report that could be associated with the Virtual Tenant as per the upstream request.
  - Avoid systematically returning the status of failure, in force mode we will always try to delete knowledge objects that may not exist.
- **trackme-limited/trackme-report-issues#720 - change - TrackMe Home UI - When creating a technical component alert, the Ack mode should be a dropdown instead of a text box selector**
  - When creating a new alert for the component, the Ack mode selector is provided as a text input rather than a more adapted dropdown selector as only two options are possible.
- **trackme-limited/trackme-report-issues#722 - enhancement - Tabulator in Home UI - Add control against the allow adaptive column in the Tabulator, Add missing column for more consistency in splk-dsm/splk-dhm**
  - This enhancement adds the allow adaptive thresholding column to the Tabulator for splk-dsm/splk-dhm for consistency purposes.
  - It also adds some missing columns regarding lagging policies features for these components, and makes column size and titles more consistent.
- **trackme-limited/trackme-report-issues#723 - change - TrackMe persistent backend - Address some inconsistency in the list of the persistent fields per component**
  - TrackMe uses a library Python file that defines the list of fields which should be considered as persistent.
  - This is used to ensure that we detect a modification of these fields while a concurrent update logic (tracker) can be running, so these changes are not lost.
  - This change addresses some inconsistency in these lists.
- **trackme-limited/trackme-report-issues#721 - enhancement - Role Based Access Control (RBAC) - Ensure vtenant main collections and vtenant summary main collections are made readable to roles added to Virtual Tenants**
  - TrackMe's built-in mains KVstore collections and transforms are by default readable to a few specific roles, admin/sc_admin and TrackMe built-in roles.
  - When handling RBAC to allow access to foreign roles, we should also check and grant read access to these collections for RBAC to work as expected without further intervention from Splunk admins.

- **trackme-limited/trackme-report-issues#724 - enhancement - Maintenance mode - Access as a non-admin should ideally show an informational message rather than a blocked error**
  - When accessing the maintenance mode dashboard as a non-TrackMe admin, we should ideally show an information message, instead of having the dashboard blocked with an insufficient permission issue.
- **trackme-limited/trackme-report-issues#726 - enhancement - Virtual Tenants creation - Safer verification and management of requested Virtual Tenant identifier**
  - When creating a new Virtual Tenant, there are some conventions that TrackMe will apply, such as forcing lowercase, using hyphens as the separator.
  - In some conditions, the current verification can be bypassed leading to issues during the Virtual Tenant creation. This enhancement ensures a safer and more consistent approach.
- **trackme-limited/trackme-report-issues#728 - feature - Allows defining a Virtual Tenant wide indexed constraint for splk-dsm/splk-dhm that automatically influences associated generated searches created by TrackMe, such as UI search action button or SmartStatus**
  - This feature allows defining at the Virtual Tenant level a custom indexed constraint, which is then automatically used while defining automated searches such as the search button in the Home UI, or searches created by the SmartStatus.
  - This can be useful in a scenario where each Virtual Tenant is associated with a custom indexed constraint, such as referring to a splunk_server_group or any other required indexed string.
- **trackme-limited/trackme-report-issues#730 - change - Moving the default future tolerance system wide option from General to splk-general for consistency purposes**
  - The system wide option Future indexing tolerance is moved from General to splk-general for more consistency in the options.
- **trackme-limited/trackme-report-issues#733 - enhancement - Flex Object library - Add dcount host to the drop detect use cases (splk_detect_drop_events_count_absolute/splk_detect_drop_events_count_rolling)**
  - Improvement to the Flex Obect library use cases related to drop events detection, adding the dictinct count host KPI.

11.1.25 Version 2.0.97 - build 1720562684 (09/07/2024)

- SHA256: 44c4a80e9584f546583f4cc43661a45c499f05b50f26e31159fa419225ced26e

Fixed Issues:

- **trackme-limited/trackme-report-issues#669 - bug - Flex Object (splk-flx) - When triggering due to Anomaly Outliers and when not actively managed by a tracker, a Flex Object entity will not return to green state if the Outliers conditions is fixed**
  - If a Flex entity turns red due to Outliers condition, and if that same entity is not actively managed by a tracker (for instance if the tracker is time conditioned for some reasons it's not active in the tracker time window), TrackMe will not update the status of the entity properly.
  - This is due to the fact that we should take into account the flag field status in addition with the object_state especially for Flex objects
- **trackme-limited/trackme-report-issues#670 - bug - Outliers Anomaly detection - Singles in the simulation screen do not honour the simulation screen time range selector and use the front page UI time range instead**
  - When performing Outliers simulation, there are different single views designed to show the key behaviours and statistics.

  - However, the searches driving the calculations underneath do not honour the simulation specific time range selector, and instead obey to the time range selector of the main entity screen.

- **trackme-limited/trackme-report-issues#672 - bug - Outliers Anomaly detection - Disabling ML detection on a per entity basis is not properly honoured by TrackMe**
  - ML Outliers Anomaly detection can be disabled on a per entity basis.
  - There is a regression in the current release of TrackMe and the Decision Maker component which prevents this setting from being properly honoured.
  - This fix addresses the issue and also provides an enhanced behaviour making this immediately reflected.

- **trackme-limited/trackme-report-issues#674 - bug - TrackMe State events - the sourcetype trackme:state should by default expect object_state and not state as the field name containing the object_state (default configuration for allow list in trackme_settings.conf)**
  - When trackers are executed, TrackMe generates state events in trackme:state
  - The fields behaviours are dicted by the TrackMe configuration, however the allow list currently expects the field "state" where we should expect "object_state" as per TrackMe's convention

- **trackme-limited/trackme-report-issues#675 - bug - TrackMe Flip events - For consistency purposes, TrackMe should also include the anomaly_reason field in the event generation**
  - When TrackMe entities experience a status change, a corresponding flipping event is generated with the sourcetype trackme:flip
  - The field anomaly_reason is a key field in TrackMe's convention, it is part of the flipping message but should also be included on its own for consistency purposes with other TrackMe concepts.

- **trackme-limited/trackme-report-issues#676 - bug - Data Source tracking - Tags manual - Creating manual tags fail if there are no tags policies created for the tenant**
  - If there are no tags policies in the Virtual Tenant, attempting to create manual tags for a given entity fails due to a Python raise condition, leading to the an error message instead.

- **trackme-limited/trackme-report-issues#685 - bug - Bulk Edit - Select all tick box in the Tabulator does not honour table filters and leads to all visible entities to be selected**
  - When performing bulk edit entities in TrackMe, there is an option "tick all" which allows selecting all entities.
  - However, there has been a regression, and TrackMe does not apply current filters from the Tabulator table, leading to all entities to be selected by the tick all checkbox, rather than only resulting entities.

- **trackme-limited/trackme-report-issues#690 - bug - Virtual Tenants UI - Avoid switching between operation and degraded on the Virtual Tenant flex box when there is an actual tracker in failure**
  - In TrackMe's Virtual Tenant UI, if an issue is affecting a tracker, in some circumstances the degraded cross information on the Virtual Tenant box will switch from degraded to operation, then at back to degraded.
  - This is due to a Python flaw in the logic of the process_exec_summary function in lib/trackme_libs_load.py.

- **trackme-limited/trackme-report-issues#691 - bug - Virtual Tenants UI - screen TrackMe Tenants Operational health statuses can have the Tabulator table hidding buttons with large number of tenants**

  - In the Virtual Tenants UI and when there are a large number of tenants, the bottom buttons of the screen TrackMe Tenants Operational health statuses may be hidden by the table.

- **trackme-limited/trackme-report-issues#693 - bug - Metric Hosts tracking (splk-mhm) - When creating a new hybrid tracker on a remote target, the break by statement is invalid leading to no results for the tracker**
  - This bug affects the creation of a new hybrid tracker for splk-mhm when the target is a remote target.
  - The break by statement generated is invalid and missing the host call in the statement.
- **trackme-limited/trackme-report-issues#696 - bug - Hybrid Tracker (all components) - Ensure to safety truncate the submitted tracker name to 40 chars at the API level and prevents from any risk of failure due to Splunk 100 chars limit**
  - Handles conditions where the submitted tracker ID could lead to a report name requested by TrackMe's API that goes beyond the 100 max chars accepted by Splunk

Enhancements, changes and new features:

- **trackme-limited/trackme-report-issues#668 - change - Tabulator - Upgrade to Tabulator 6.2.1**
  - Upgrade of Tabulator JS to release 6.2.1
- **trackme-limited/trackme-report-issues#619 - feature - Maintenance Mode - Support for enabling Maintenance Mode with selection of applicable tenants, support for Maintenance Knowledge DataBase #619**
  - This introduces support for selective Maintenance Mode on a per tenant basis, you can enable the Maintenance Mode for all tenants (default) or a list of applicable Virtual Tenants.
  - Support is also added to the Maintenance Knowledge DataBase in TrackMe, as well as automatically influencing SLA calculations depending on if the maintenance period for applicable for the entity tenant.
  - After the upgrade to 2.0.97, TrackMe's schema upgrade will automatically update TrackMe alerts to call the new macro trackme_apply_maintenance_mode
- **trackme-limited/trackme-report-issues#650 - change - Overview eventcount timechart calculations and Performance metrics tab timechart calculations for spl-dsm/splk-dhm, for consistency purposes, use a sum calculation per metrics at the timechart level instead of an avg, as it happens for the latest_eventcount_5m**
  - In the overview chart tab, when looking at increased timeranges, we should rather use a sum calculation for eventcount metrics for consistency purposes.
  - In the Performance Metrics tab, and for splk-dsm/splk-dhm. TrackMe uses a "sum(latest_eventcount_5m) as latest_eventcount_5m" where others metrics are calculted using an avg.
  - None of these are technically false and just different reading, but for users going trough a basic approach of comparing true eventcount, this can be confusing.
- **trackme-limited/trackme-report-issues#671 - change - Outliers Anomaly detection - Allow full control on the period_calculation definition**
  - This update allows complete control for the definition of the period_calculation.
  - The time quantifier period expression can be submitted without pre-defined periods, for default models generation and on per model basis.
- **trackme-limited/trackme-report-issues#677 - change - Hybrid Trackers creation screen - Automatically pre-fill the tracker name with a randomly generated ID**
  - When creating hybrid trackers, a text input is expecting a name to be choosen for this tracker.

  - To improve the global user experience, automatically prefill this input with a randomly generated identifier.

- **trackme-limited/trackme-report-issues#678 - enhancement - Flex Objects (splk-flx) and TrackMe KPI generation - Support for time definition in metrics generation at ingest time**
  - This enhancements provides support in TrackMe to generate metrics with an upstream value for the metric time stamp.
  - This allows supporting use cases where the time in the Tracker logic is not equal to when the tracker is executed, but rather part of the SPL statement.
- **trackme-limited/trackme-report-issues#679 - feature - Flex Object use cases library - New use cases splk_detect_daily_variations_volume_global / splk_detect_daily_variations_volume_index**
  - These two new use cases are designed to leverage the Splunk license indexing logs to track the daily absolute amount of data indexed globally on the license pool, and on a per index basis.
  - These KPIs then are used to train Outliers Models with the goal of detecting abnornal decrease/increase of indexing volume per entity.
- **trackme-limited/trackme-report-issues#680 - enhancement - Outliers Anomaly detection - Add %w as an option for the time_factor (time factor influenced per week day)**
  - This enhancement adds support for a time factor option per week day (%w) for the configuration of Machine Learning models via TrackMe's UI
- **trackme-limited/trackme-report-issues#681 - feature - TrackMe Home UI and Virtual Tenant account preferences - Add time range selections up to 1y and allows defining the default time range at the level of the Virtual Tenant account preferences**
  - Adds new period for 6 months (180d) and 1 year (365d) in the time ranger selector of TrackMe's main UI.
  - Allows defining on a per Virtual Tenant account the default time range to be selected when accessing entities.
- **trackme-limited/trackme-report-issues#682 - enhancement - Common Information Model compliance (splk-cim) - Improvement of the preview search functions and direct links to open searches in a new window**
  - Add the from datamodel search
  - Add direct links button for from / datamodel / tstats searches which dynamically open the search into a new window with local/remote account support
- **trackme-limited/trackme-report-issues#684 - feature - Acknowledgment management - Expire Ack on anomaly reasons changes so TrackMe can raise a new alert when conditions for alerting have changed**
  - This new feature allows TrackMe Ack to be influenced by the change of anomalies affecting entities.
  - Conditioned by system level configurable options (See Configure / General / Expire Ack on anomaly reason change behaviour, Expire Ack on anomaly reason change min time since, Expire Ack on anomaly reason only for auto ack), this new feature completes and enhances the Ack capabilities in TrackMe.
  - If an entity that turned red due to an Outliers detection for instance, and later on is also affected by an additional condition such as a lag breach, the Ack will be automatically expired so that a new alert can be raised transparently by TrackMe.
- **trackme-limited/trackme-report-issues#687 - enhancement - Data Hosts tracking (splk-dhm) - The Performance metrics tab in entity overview should include the Delay Metrics and also include dynamic explanations as with splk-dsm**

  - In Overview entity then Performance Metrics tab, we should for splk-dhm provide access to the Delay metrics, as well as explanations regarding these metrics calculation, similarly to splk-dsm

- **trackme-limited/trackme-report-issues#688 - enhancement - TrackMe Tracker executor backend - Improved detection of silently failing trackers**
  - When TrackMe executes trackers, this execution goes through a quality and review backend with the custom command `trackmetrackerexecutor`
  - Especially, this process tracks for execution failures, generates run time metrics for TrackMe's trackers and feeds the Tenant operation statuses.
  - In some circumstances, some types of execution failres can happen silently and the current version of the backend does not notice it, this fix slightly enhances and is capable of detecting any conditions leading to the failure of the tracker.
- **trackme-limited/trackme-report-issues#689 - enhancement - TrackMe Health Tracker - Add an additional safety check to identify and purge unexpected foreign records that would have been added by mistake to a main data KVstore collection**
  - Each tenant has a TrackMe Health Tracker which performs various maintenance routines, in this issue we add an additional action to check for the presence of unexpected foreign records in the main KVstore collection, and purge these records automatically if any.
  - Foreign records could have been added by mistake when manipulating KVstore collections, and would lead to be blocking many logics in TrackMe.
- **trackme-limited/trackme-report-issues#692 - feature - TrackMe Virtual Tenants - New API endpoint to clear the Virtual Tenants Operation Status actionable through the Virtual Tenants UI**
  - This new API endpoint allows TrackMe Admins to clear the Virtual Tenants Operation Status and optionally request the imediate refresh through the execution of the tenant's health tracker.
  - In the Virtual Tenants UI, this feature can be requested via the screen TrackMe Tenants Operational health statuses for all tenants, or a selection of tenants with the option to execute or not the health tracker.
  - Clearing the Virtual Tenant Operation status can be useful when dealing with a degraded Virtual Tenant which status is blocked due to some issues.
- **trackme-limited/trackme-report-issues#694 - feature - SOAR Monitoring - Adding Flex Object UC to track SOAR/Splunk forwarding integration (splk_soar_forwarding_splunk)**
  - This additional Flex Object use case for SOAR focusses on tracking the SOAR/Splunk forwarding integration
- **trackme-limited/trackme-report-issues#695 - feature - Flex Object library - New Flex Use Case splk_splunk_infra_log_level_variations which deals with Splunk logs and their logging level and use Machine Learning to detect abnormal behaviours of your Splunk instances and deployments**
  - This new Flex Object use case tracks Splunk internal log events and their associated logging level to detect suspscious trends, which are symptomatics of Splunk behaving improperly and facing or about to face serious issues.
  - To achieve this, we leverage TrackMe's Flex component and our Machine Learning implementation, we then track trends notably of errors in Splunk logs to alert when an abnormal amount of errors is detected.
- **trackme-limited/trackme-report-issues#697 - bug - All components - Entities containing backslashes generate all sorts of issues in TrackMe, this condition can notably be encountered in Workload (splk-wlk) with very bad report naming**

  - Entities ending up with backslashes can generate various issues in TrackMe, especially in advanced features such as ML Outliers or Metadata tracking for splk-wlk.
  - This issue addresses this problematic by encoding backslahes at the discovery Python phases, and decode transparently for users as needed.

- **trackme-limited/trackme-report-issues#698 - enhancement - Workload (splk-wlk) - Management of dupplicated entities at the phase of the health tracker execution**
  - In the Workload component (splk-wlk), the Health Tracker verifies for duplicated entities, and deletes automatically one of the duplicated randomly.
  - However, it can happen in some conditions that we will keep continously deleting the wrong entity which then keeps being re-created.
  - For more consistency, this fix will allow TrackMe to purge both concerned entities, so only the right one gets re-created accordingly.

11.1.26 Version 2.0.96 - build 1718623969 (17/06/2024)

Major UI filtering performance improvements

- This release introduces major performance improvements in the TrackMe main UI, especially when performing entity filtering, which is now nearly instantaneous, regardless of the collection size.
- These improvements are made possible by the switch to client-side (local) pagination and filtering in Tabulator, which can now also be controlled through general and per-tenant parameters since this release.

- SHA256: 02e835c27b2c681a7ad89f0c9e100a4a6317e824bb9fb3d21286e1108ceabee0

Fixed issues:

- **trackme-limited/trackme-report-issues#657 - bug - Outliers Anomaly Detection - TrackMe does not honour the method_calculation defined at the model level when performing training and rendering of the model #657**
  - On per model basis, a method calculation can be applied at the level of the mstats search, which will be associated with the KPI span to influence the Outliers root calculation.
  - However, currently TrackMe does not honour properly the method calculation due to a bug in TrackMe's Outliers Python library.
- **trackme-limited/trackme-report-issues#659 - bug - Outliers Anomaly Detection - Default system parameters should not require a value for static LowerBound/UpperBound #659**
  - In Configuration / splk-outliers-detection, saving parameters should not require a value for LowerBound/UpperBound
- **trackme-limited/trackme-report-issues#665 -bug - TrackMe Home User Interface - Outliers Anomaly Detection appearance remaining issues after performing training via the Manage Outliers UI**
  - Folllowing fixes from trackme-limited/trackme-report-issues#638, there are remaing issues and conditions leading to the Outliers MLTK chart to fail appearing properly after a training is made through the Manage Outliers UI screen.
  - This is due to the fact that refreshing the search underneath the MTLK Outliers charts while the chart is not visible yet leads to this issue.

Enhancements, changes and new features:

- **trackme-limited/trackme-report-issues#658 - enhancement - Outliers Anomaly Detect - Add additional options for the Outliers kpi_span #658**

  - Complete selectable options for the kpi_span per model with additional values up to 24h

- **trackme-limited/trackme-report-issues#660 - feature request - Filter functions - Add in the "By Acknowledgment state" new options to filter on Acknowledged entities per priority**
  - In TrackMe's UI, one can use filter functions to prefilter on multiple conditions at once.
  - This feature requests is to add filter functions for Acknowledged entities based on priority filters
- **trackme-limited/trackme-report-issues#661 - enhancement - TrackMe Home UI performance - client side pagination and filtering in Tabulator for largely improved performances especially when filtering**
  - By implementing client side (local) pagination and filtering, this release introduces major performance gains in TrackMe main UI, especially when performing entities filtering based on any available simple or complex conditions.
  - The pagination mode and pagination size can now also be controled at the level of the Virtual Tenant account, with the base general configuration that can be customised when creating tenants, and once the tenant has been created through the Virtual tenant account
  - These enhancements bring major performance improvements to TrackMe, slightly improving the end user experience.
- **trackme-limited/trackme-report-issues#662 - enhancement - Virtual Tenants UI - Make the results from focus searches more readable and valuable**
  - In the Virtual Tenants UI, when putting the focus on a given tenant / status by priority, a search runs and provides an high level overview of underneath entities.
  - The purpose of this issue is to simpify the approach to get more readable and valuable results, from this release the search will generate a simpler list of concerned entities ordered by their flip status. (ordered by the last time these entities have had a status changed)
- **trackme-limited/trackme-report-issues#663 - enhancement - Adaptive Thresholding - Allows to control the review period through the argument review_period_no_days at the level of the adaptive tracker**
  - In this issue, we introduce a new argument to the Adaptive delay custom command which controls the period of time used to identify entities to be reviewed over time following a change made the Adaptive Treshold backend.
  - The argument review_period_no_days accepts 3 period options: 7, 15 or 30 days for the period of review.
  - After the upgrade, TrackMe will automatically update existing and active trackers through the schema upgrade.
- **trackme-limited/trackme-report-issues#664 - change - Adaptive Thresholding - Change of the default period for review from 7 days to 30 days following the introduction of the new option review_period_no_days**
  - Associated with the new argument review_period_no_days, the default is now set to 30 days to improve the behaviour over time of the Adaptive Tresholding backend, and ensure we review for long period enoughs entities that have been modified by the backend.

11.1.27 Version 2.0.95 - build 1718137396 (11/06/2024)

New priority level with critical priority

- This release introduces an additional priority level "critical" for TrackMe entities.

- This will provide more flexibility and consistency for customers to leverage various CMDB and logics, and alert with different types of actions depending on the importance of associated entities.
- You may need to review your current alerts, and include the new priority level in your alerting logic.

- SHA256: 10f1318c0895f7cd4d648f1a9e48795858ebc5991c0c27447ace816058a9c84a

Fixed issues:

- **trackme-limited/trackme-report-issues#638 - bug - TrackMe Home User Interface - Outliers Anomaly Detection chart may not show up properly in some circumstances, as after Models modifications or attempting to handle a non valid model #638**
  - The Outliers Anomaly Detection tab triggers when actioned by the user, and will display the models statistics and the Outliers chart.
  - In some circumstances, such as after a modification of a model or after attempting to display an entity with no models, the chart fails to display properly and will not display until the UI is fully refreshed.
  - This is caused by attempting to enable / disable the containing HTML div at the CSS level which does not behave well with the MLTK viz chart.
- **trackme-limited/trackme-report-issues#641 - bug - Metrics hosts monitoring (splk-mhm) - Get component loads to fail entities due to regression since 2.0.93**
  - Entities fail to be loaded properly for splk-mhm due to the load component librairies evolutions
  - The component incorreclty attempts to load Outliers KVstore collections, which is not applicable to splk-mhm resulting in failure to load entities when opening the UI
- **trackme-limited/trackme-report-issues#642 - bug - Data Source monitoring (splk-dsm) - Data Sampling status and enablement should be immediately reflected by the DecisionMaker**
  - When modifying the Data sampling feature enablement, this should be immediately and properly reflected in the DecisionMaker results as well as the TrackMe UI screen.
- **trackme-limited/trackme-report-issues#645 - bug - REST API - Update endpoints calling the method generic_batch_update will not take into account replacements with empty values, which impacts reset actions such as in splk-dhm/mhm**
  - When calling REST API endpoints for update purposes, TrackMe implements a batch update method to update KVstore records as fast as possible.
  - This Python method is called generic_batch_update and currently ignores replacement of values by an actual empty value.
  - However, doing so causes a regression for some specific endpoints such as the reset endpoint for splk-dhm/mhm.
  - This fix updates the function to call a Python native object method instead to update the records transparently.
- **trackme-limited/trackme-report-issues#652 - bug - TrackMe logs rotation should ideally be taken into account for Splunk ingest purposes #652**
  - When TrackMe logs are rotated, our props.conf should take into account incremented log.* files

Enhancements, changes and new features:

- **trackme-limited/trackme-report-issues#636 - enhancement - Splunk SOAR monitoring of Automation Brokers - enhancement of the REST API endpoint access to retrieve**

the status of the automation brokers to avoid hitting some scenarios where the normal REST API endpoint misses some statuses errors of the brokers #636

  - The Flex Object tracker use case for SOAR Automation Brokers monitoring allow retieveing and acting when the SOAR Automation Brokers are not in active state
  - In some edge use cases, the SOAR automation_brokers REST API misses an offline status of the Automation Broker wrongly if the API endpoints is not accessed with some additional arguments in the parameter of the REST API call

- **trackme-limited/trackme-report-issues#637 - enhancement - Acknowledgements - Safer code to handle unexpected records with no object_category #637**
  - if Ack records are unexpectly created without a valid object_category, the Ack tracker would not handle this issue properly, and would attempt and fail to retrieve the corresponding record in the data collection.
  - This would lead the tracker to fail expiring non corrupted Ack records.
  - This evolution ensures that any corruputed record would be purged accordingly, and will avoid the tracker from failing to manage other valid Ack records
- **trackme-limited/trackme-report-issues#639 - feature - Bulk edit entities - For Data Sources monitoring (splk-dsm), allows to manage Data Sampling via bulk edit**
  - Manage Data sampling actions via bulk edit: enable / disable / run / reset
- **trackme-limited/trackme-report-issues#640 - enhancement - Bulk edit - Ensures scroll bar would appear if the screen resolution is too low**
  - If the screen resolution is too low, ensure to load the vertical scrolling bar to avoid truncating the bulk edit screen
- **trackme-limited/trackme-report-issues#643 - enhancement - SOAR Automation Broker active management - Add a safety layer for ignoring typical fields containing secrets in the JSON post response when updating assets, in addtion with existing automated salted fields exclusion #643**
  - When updating SOAR Assets, and when not using a Vault for password management, we must not include secrets when performing the POST call.
  - TrackMe already automatically excludes fields which have been salted by SOAR, however as an additional safety and to be retro-compatible with older Assets defined, we also exclude typical fields: apikey,api_key,password,auth_token,client_secret
  - This can be controled at the level of the POST call using the option: assets_update_forbidden_fields
- **trackme-limited/trackme-report-issues#646 - enhancement - Data Host/Metric host tracking (splk-dsm/mhm) - Behaviour improvements for the reset actions**
  - The reset actions can be used to reset the current knowledge for a given entity when it comes to indexes, sourcetypes for splk-dhm or metric categories for splk-mhm.
  - The current behaviour can be improved to better cleanup the associated fields with a more consistent approach.
  - This enhancement also avoids removing the visibility for the entity that was reset until knowledge is built again.
- **trackme-limited/trackme-report-issues#647 - enhancement - TrackMe Notable events - automatically parse the anomaly_reason and turn into a list so Splunk can extract it as an mvfield**
  - The anomaly_reason a primordial field in TrackMe which is used by the DecisionMaker to insert all conditions encountered for a given entity, at the lowest level it is a native Python list.

  - However, when generating TrackMe notable events, the field is turned into a pipe separated string.
  - To allow automated mv structure extraction in Splunk, the field should rather be turned back into a list in the JSON structure.

- **trackme-limited/trackme-report-issues#648 - enhancement - SmartStatus - smartstatus_investigations_uc_dsm_latency and smartstatus_investigations_uc_dhm_latency should rather leverage tstats based search to slightly reduce associated costs**
  - When the SmartStatus is executed and when the entity is red for latency reasons, we currently generate a raw search for a full accuracy regarding the latency calculation.
  - However, these searches can be slightly expensive at high scale for a relative value, in this issue we migrate the generated searches to a tstats based search instead.

- **trackme-limited/trackme-report-issues#649 - enhancement - SmartStatus alert action - Protect Splunk workload and prevent SmartStatus alert action from being executed more than once per 24 hours per entity**
  - In some circumstances such as if a TrackMe alert was badly setup without leveraging TrackMe's Ack concepts, or increasing the suppression period, the SmartStatus alert action could be triggered and executed more than wanted, which in turn could affect Splunk workload and generate more activity than required.
  - In this issue, we introduce a concept that keeps track of the last seen execution per entity, and we will automatically skip the SmartStatus action if the action has been executed in the past 24 hours already.

- **trackme-limited/trackme-report-issues#651 - enhancement - Add sum and min as calculation methods when missing as selectable options in Outliers configuration and other dropdown in TrackMe's UI**
  - In Outliers calculation methods configuration (default configuration and per model fine tuning), the sum and min options should be available.
  - In selectable options parts of drildown selectors such as in Flex Objects, these methods should be available.

- **trackme-limited/trackme-report-issues#653 - change - Anomaly Outliers detection - Add -15d in selector for period of calculation, change -360d to -365d for consistency when requesting 1 year for the period, reflect the same periods in the configuration screen for default assignment**
  - Add -15d in selectable options
  - Swtich -360d to -365d for consistency regarding a year of relative period of time for the calculation period
  - Reflect the same options in the configuration screen for default perod assignment for consistency

- **trackme-limited/trackme-report-issues#654 - feature - Entities priority management - Add a new priority with critical priority to provide more flexibility in TrackMe entities management**
  - This release introduces a new priority level "critical" for TrackMe entities.
  - This will provide more flexibility and consistency for customers to leverage various CMDB and logics, and alert with different types of actions depending on the importance of associated entities

- **trackme-limited/trackme-report-issues#655 - feature - Priority management - Migrate priority management from macro based to a per Virtual Tenant account option for more flexibility**

  - The priority management is being migrated from the macro trackme_default_priority to an easily configurable option per Virtual Tenant account.
  - Users can now define the default priority at the time of the creation of the Virtual Tenant, or any time in the Configure / Virtual Accounts configuration screen.
  - This provides a more flexible and more consistent approach to the priority management in TrackMe.

- **trackme-limited/trackme-report-issues#618 - Feature Request - Alert configuration "trigger on outliers" and "trigger on sampling" behaviour would lead to miss other anomaly reasons**
  - When creating a TrackMe alert, one can select to trigger or not against Outliers, and Sampling. (note: Sampling is splk-dsm only)
  - However, the current logic can be much improved to also handle use cases where we have a mutli-detection, and we have more than anomaly in addition with Outliers/Sampling.
  - With this evolution, TrackMe will parse automatically the anomaly_reason as part of the trackmegetcoll output, adds a new field for the count of anomaly_reason, and finally updates the method when creating a new alert.

- **trackme-limited/trackme-report-issues#656 - change - CIM Compliance trackers (splk-cim) - Licensing restriction increase to 32 trackers for Enterprise Edition customers**
  - We are increasing the max number of CIM Compliance trackers for Enterprise Edition customers from 16 to 32.

11.1.28 Version 2.0.94 - build 1716849152 (27/05/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 465a36c35cc9a161218a9b9c7ff14204d8fd896d72878c943277fc9b5664d4ff

Fixed issues:

- **trackme-limited/trackme-report-issues#626 - bug - SOAR Automation Broker high availability management - update of the broker will reset unexpectly any secrets of the assets for users not using a Password Vault #626**
  - When performing an active update of the automation broker via the Flex Object use case, we perform an update of the Asset configuration via the SOAR API to swtich the broker from A to B.
  - For users not using a Password Vault, SOAR handles any credential such as an API token, the token is salted in the data.
  - When performing the REST POST call to the API, we should remove any field in the JSON structure which starts with a "salt:" to avoid resetting this secret unexpectly, or the asset connectivity is lost.
  - This only applies to internal SOAR secret management, in the sense that SOAR customers using a Password Vault are not affected by this issue.

- **trackme-limited/trackme-report-issues#628 - bug - error when clicking on refresh entities in TrackMe UI when looking at a given entity: search_kv_collection() got an unexpected keyword argument #628**
  - Issue happens when clicking on refresh when looking at a given entity
  - This is a regression introduced in TrackMe 2.0.92

- **trackme-limited/trackme-report-issues#629 - bug - Adaptive Tresholding - Avoid attempting to take into account during the review a feed that was previously updated but stop indexing to Splunk in the past 7 days #629**
  - When the adaptive threshold backend updates an entity, this entity automatically enters the review phase to ensure we take into account updated behaviours, such as an outage that was resolved in the meantime.
  - However, if an entity that was previously updated stop indexing data to Splunk, we should not take it into account anymore if it didn't index any event for the past 7 days to avoid raising an exception while accessing the adaptive_delay result.
- **trackme-limited/trackme-report-issues#631 - bug - Workload (splk-wlk) - Regression in TrackMe 2.0.93 due to missing fields in lookup transforms leading to status not met instead of advanced status distinction #631**
  - In TrackMe 2.0.93 and to address some CPU & Memory pressure, we have swtiched the base logic to access KVstores to a search based approach.
  - This impacted the Workload component due to missing fields in the Lookup transforms, which cannot be access in a search unless part of the transform, this had lead to a status not met instead of the detailed statuses.
  - Once upgraded, the TrackMe health tracker schema upgrade routine will update the lookup transforms accordingly with no action required.
- **trackme-limited/trackme-report-issues#633 - bug - Outliers detection potential regression in TrackMe 2.0.93 leading to isOutlier not reported at DecisionMaker time**
  - Due to search based approach when accessing KVstore records in TrackMe 2.0.93, in some circumstances it is possible that detected Outliers do not get reported while loading entities.
  - This issue introduces a robust and consistent approach at the Python level to lookup Outliers, similarly to other phases in the TrackMe Decision Maker.

Enhancements, changes and new features:

- **trackme-limited/trackme-report-issues#627 - enhancement - trackmehealthtracker - Optimize run time, costs and behaviour of the inspect_collection phases #627**
  - The schedule job trackmehealthtracker is responsible for various maintenance routines, one of these is called "inspect_collection"
  - This maintenance routine verifies the consistency of TrackMe entities statuses (object_state) between the KVstore record view and the realtime view from TrackMe's DecisionMaker processes
  - The TrackMe DecisionMaker process is a suite of many Python functions depending on the time of component which apply conditions for monitoring, such as delay/latency rules, logical group mapping, Outliers detection and so forth
  - In this issue, we optimized this specific process with a faster and lighter search based approach to load the KVstore raw collection, load the TrackMe DecisionMaker view (using the trackmedecisionmaker streaming custom command) then performing the comparison
  - The objective of this update is therefore to reduce the costs of this step, reduce the global runtime of the trackmehealthtracker job and avoids generating skipping search
- **trackme-limited/trackme-report-issues#630 - enhancement - Adaptive Threshold - Avoid attempting to inspect entities that have not actively generated data in Splunk for a minimum period equivalent to the max_delay_sec argument given to the backend #630**
  - For optimization and costs reduction purposes, TrackMe's adaptive threshold backend should not attempt to inspect entities which are not actively sending data to Splunk.
  - The current behaviour implies that we may continue to attempt to inspect entities that are monitored actively but without any recent ingest activity (past 7 days)

  - To improve consistency while reducing TrackMe's workload, the Adpative Threshold backend should ensure to take into account entities in the initial inspection phase only if the current delay is < max_auto_delay_sec (default to 7 days)

- **trackme-limited/trackme-report-issues#632 - enhancement - Decomission ML models orphans cleanup from the trackmehealthtracker as it is also handled via the general health tracker #632**
  - In TrackMe 2.0.84 was introduced the general health tracker which is executed once per day amongst all Virtual Tenants.
  - Especially, this job handles all cleaning related to Machine Learning, such as detecting and purging Orphans models. (models which entities have been purged, or the tenant was purged)
  - Previously, this activty was handled by the tenant level health tracker, this is not required any longer and we can save from this activity to optimize and reduce TrackMe's workload.

11.1.29 Version 2.0.93 - build 1716457845 (23/05/2024)

Hint

High CPU and Memory pressure regression from TrackMe 2.0.92: this release addresses several important issues leading to extra CPU and memory pressure introduced with TrackMe 2.0.92

- SHA256: 9d6d5cd975f6f7fcbb1966b212206f77a414938b5f5b0446a0bded64233f550c

Fixed issues:

- trackme-limited/trackme-report-issues#620 - bug - Option sla_default_threshold in sla is not used on purpose and should have been removed from the configuration UI #620
- trackme-limited/trackme-report-issues#621 - bug - Virtual Tenants UI - If using legacy TrackMe load mode, this should also apply to automated refresh #621
- trackme-limited/trackme-report-issues#622 - bug - Maintenance mode management UI - Web browser over consumption over time due to resources leak with Javascript autorefresh #622
- trackme-limited/trackme-report-issues#623 - bug/enhancement - Performance and footprint reduction at high scale (More than 10k/100k collections) - changes introduced in TrackMe 2.0.92 can lead to extra CPU and memory consumption #623
- trackme-limited/trackme-report-issues#624 - bug - CIM Compliance tracking (splk-cim) - object_category should be in the collections for consistency purposes regarding all other components, its lack currently impact notables and acknowledgement #624
- trackme-limited/trackme-report-issues#625 - bug - CIM Compliance tracking - When creating a notable or SLA alert, components alert actions are wrongly added to the alert #625

11.1.30 Version 2.0.92 - build 1715771041 (15/05/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 800d2adbf600d31124558b3dff4104bc3bb41e405365c284077ddc1500e38864

Fixed issues:

- trackme-limited/trackme-report-issues#617 - bug - Regression with the usage of numpy which impacts schedule logic where we limit their run time - due to an Appinspect restriction and numpy

storing libs in a hidden directory which was removed automatically by our automation, this leads to the custom command to fail at exec time #617

11.1.31 Version 2.0.91 - build 1715725834 (14/05/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 0d41329cd50ed1531b4548ff0e7136b293dec45f25eec57993b315ad5aa0e6ee

Fixed issues:

- trackme-limited/trackme-report-issues#601 - bug - Logical Group REST API - avoid raising an exception when groups members, or green / red members are unexpectly null #601
- trackme-limited/trackme-report-issues#603 - bug - Prevents an exception in the REST API endpoint post_component_summary_update which is responsible for caching the tenant and component statistics #603
- trackme-limited/trackme-report-issues#604 - bug - Missing searchbnf providing usage syntax for the custom command trackmesplkpriority #604
- trackme-limited/trackme-report-issues#605 - bug - Priority Policies apply in TrackMe UI - incorrect variable leads to slient failure while applying policies for other components than splk-dsm #605
- trackme-limited/trackme-report-issues#608 - bug - Logical Groups - Unexpected non list structured in object_group_members / object_group_members_green / object_group_members_red can lead to Python exceptions and to the related entities not be available in the UI or from trackmegetcoll #608
- trackme-limited/trackme-report-issues#609 - bug - Data Hosts tracking (splk-dhm) - At high scale collection (more than 10k hosts), the current pagination count per count leads to incomplete rendering of entities #609

Enhancement, changes and new features:

- trackme-limited/trackme-report-issues#597 - enhancement - Adaptive Threshold tracker for Data Sources / Data Host tracking - The recent activty instrospection should take into account a change of the allow_adaptive field in case it has changed after the entity entered the cycle of adaptive review #597
- trackme-limited/trackme-report-issues#598 - feature request - Implement a per entity SLA timer and threshold concept, this would be used in a 2 tiers alerting system when a specifc alert would be sent when the SLA of entity is breached after having spent too long in a red state #598
- trackme-limited/trackme-report-issues#606 - change - Virtual Tenants UI - entities summary while double clicking on a given tenant should specify “enabled entities” rather than simply “entities” to avoid any confusion #606
- trackme-limited/trackme-report-issues#610 - change - Adaptive Treshold tracker - At the creation phase, the Adaptive Treshold tracker should be executed every 20 minutes to avoid risks of generating skipping searches at high scale #610
- trackme-limited/trackme-report-issues#611 - enhancement - Improving TrackMe logic to avoid generating skipping searches in various TrackMe scheduled logics #611
- trackme-limited/trackme-report-issues#612 - feature - TrackMe Alerting Architecture - Allows creating TrackMe Notables from TrackMe UI, Add builtin documentations and design good practices #612
- trackme-limited/trackme-report-issues#613 - change - REST API - bulk edit endpoints update to verify if json_data is submitted as a string, and if so loads it as a dict #613

- trackme-limited/trackme-report-issues#614 - enhancement - Persistent fields - centralization of per component persistent fields in collection_dict.py for more consistent and safer code #614
- trackme-limited/trackme-report-issues#615 - feature - Flex Object Library - Add a new use case to track the daily volume of data ingested per day and per index, and leverage Machine Learning for the Outlers detection #615
- trackme-limited/trackme-report-issues#616 - feature - Bulk Edit performance - Massive improvement in bulk edit performance in TrackMe, bulkd edit now runs in a fraction of seconds no matter the volume of the collection #616

11.1.32 Version 2.0.90 - build 1714432454 (30/04/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: a0d0360ae77c807bc991fa960580675860a4e75828e6b55f08bf5cd70d97e1b2

Fixed issues:

- trackme-limited/trackme-report-issues#584 - bug - New ctime field is not persistent in some components (dsm, wlk) #584
- trackme-limited/trackme-report-issues#593 - bug - Data Hosts tracking / Metric Hosts tracking - error message trackmeextractsplkmhm/trackmeextractsplkdhm when the command is executed in no metric generation mode #593
- trackme-limited/trackme-report-issues#595 - bug - Data Source / Data host tracking (splk-dsm/splk-dhm) Persistence of fields issue when the Adaptive tracker runs due to some Python level issues with batch update related code in the specific circumstances of sending a partial update #595

Enhanccements, changes and new features:

- trackme-limited/trackme-report-issues#585 - feature - priority management - provide a component wide feature for priority dynamic managements using regex based policies #585
- trackme-limited/trackme-report-issues#587 - enhancement - Virtual Tenants - Load Tenants high level statistics available when double clicking on the tenant flex box from cachedstats for consistency and better performance at high scale #587
- trackme-limited/trackme-report-issues#588 - enhancement - Virtual Tenants UI - Add a configuration choice for the trackmeload mode (REST versus legacy search driven) to address some limited compatibility issues reported by FEDRAMP Classic Splunk Cloud #588
- trackme-limited/trackme-report-issues#589 - feature - Machine Learning engine - Add capabilities to define static static_lower_threshold / static_upper_threshold per model #589
- trackme-limited/trackme-report-issues#590 - change - Data Hosts tracking (splk-dhm) - presets tstats root span to 1m by default #590
- trackme-limited/trackme-report-issues#591 - feature - Virtual Tenants creation UI - Allow in the first steps to define tenants level settings (ML Outliers features and other main Tenants level optons) #591
- trackme-limited/trackme-report-issues#592 - feature - Virtual Tenants - Allows to control the enablement of TrackMe Machine Learning Outliers Anomaly detection at the level of the Virtual Tenant #592
- trackme-limited/trackme-report-issues#596 - enhancement - Machine Learning - Avoids the error "The ML search is not yet available for rendering" when the ML model is not yet ready for rendering #596

11.1.33 Version 2.0.89 - build 1713898383 (23/04/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: d7a561e975b0ddfa4c1ee423c7e7eef7f59f339c6d1ee415112ca89cb1a2ec47

Fixed issues:

- trackme-limited/trackme-report-issues#562 - bug - REST API - Maintenance mode disable endpoint should return a native JSON response rather than a JSON dumped response #562
- trackme-limited/trackme-report-issues#563 - bug - REST API - fix various documentation errors in TrackMe's REST API endpoints #563
- trackme-limited/trackme-report-issues#566 - bug - Machine Learning - perc_min_lowerbound_deviation in repeated twice in dsm Outliers table management, min_value_for_lowerbound_breached/min_value_for_upperbound_breached are missing from dhm tables #566
- trackme-limited/trackme-report-issues#569 - bug - DecisionMaker - Prevents against various possibilities of Python exceptions in the TrackMe Decision Maker libraries and calls which can lead to Error processing record #569
- trackme-limited/trackme-report-issues#570 - bug - Logical Groups - Ensure to limit match=1 for logical grouping enrichment at search time before reaching the DecisionMaker #570
- trackme-limited/trackme-report-issues#571 - bug - Backup and Restore - Builtin TrackMe KVstore backup fails when there are disabled tenants #571
- trackme-limited/trackme-report-issues#576 - bug - CIM (splk-cim) - SLA metrics are not generated if the trackme_metric index has been customised #576
- trackme-limited/trackme-report-issues#579 - bug - Machine Learning - ML Model addition UI in some components would not render a result when simulating the addition of the model as the command should call the lightsimulation mode rather than the simulation mode since TrackMe 2.0.88 #579
- trackme-limited/trackme-report-issues#580 - bug - Machine Learning - custom command trackmesplkoutlierssetrules generates errors when dealing with Flex Object trackers with no Outliers definition #580

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#557 - feature - Flex Object library use cases - Add new UCs for detecting abnormal drop in Splunk feeds events count using Flex #557
- trackme-limited/trackme-report-issues#558 - feature - Machine Learning Outliers - Allows up to 1 year in the time range selection for the Outliers calculation by step of 30 days #558
- trackme-limited/trackme-report-issues#559 - feature - Machine Learning Outliers - Add max in calculation methods available #559
- trackme-limited/trackme-report-issues#560 - feature - Machine Learning Outliers - Flex Object - Support all settings to be defined per Flex Object tracker rule, update built in documentation #560
- trackme-limited/trackme-report-issues#564 - enhancements - REST API - When deleting entities, permanently or temporary, the API should also clean up records for Outliers and Sampling, if any. #564
- trackme-limited/trackme-report-issues#565 - feature - New immutable KVstore field called ctime in TrackMe main KVstore component collections to keep track of entities origin creation time #565

- trackme-limited/trackme-report-issues#567 - enhancement - Virtual Tenants UI - When defining custom indexes as default indexes, the new Virtual Tenant creation UI should preset indexes with corresponding default indexes #567
- trackme-limited/trackme-report-issues#568 - enhancement - Workload (splk-wlk) - SmartStatus searches code improvements, ensure to include host=* splunk_server=* in SmartStatus Workload searches, more consistent searches matching the trackers, code improvements #568
- trackme-limited/trackme-report-issues#572 - feature - Data Host tracking (splk-dhm) - Add the capability to exclude (blocklist) a list of indexes and/or sourcetypes per host #572
- trackme-limited/trackme-report-issues#573 - feature - Machine Learning Outliers - Allow predefining at the system level extra parameters for the MLTK fit command, which can also be defined on a per model basis #573
- trackme-limited/trackme-report-issues#575 - enhancement - User Interface Home - ensure the main entity modification screens use scroll bar if the screen resolution is too low #575
- trackme-limited/trackme-report-issues#577 - feature - Machine Learning Outliers - allow using a custom MLTK algorithm #577
- trackme-limited/trackme-report-issues#581 - enhancement - Add an additional numerical verification in the Python function trackme_components_register_gen_metrics to prevents from any risks of generating malformed metrics leading to Splunk notification #581

11.1.34 Version 2.0.88 - build 1712331711 (05/04/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 70b5d340687c3e45d3702b1c4ce84e8cb6edb7a866fba75915c4de3cdafff8db

Fixed issues:

- trackme-limited/trackme-report-issues#550 - feature - Home interface drilldown & notable drilldown link - Allows submitting an object or alias URL param which filters out and opens automatically the entity overview, also add a drilldown_link to TrackMe Notables #550
- trackme-limited/trackme-report-issues#552 - bug - Virtual Tenant UI - count discrepency in summarized stats due to the monitoring enablement not being taken into account #552
- trackme-limited/trackme-report-issues#553 - bug - Python shared functions - get_kv_collection function used in some backends can lead to the generation of error messages with document ID conflict #553
- trackme-limited/trackme-report-issues#553 - bug - Python shared functions - get_kv_collection function used in some backends can lead to the generation of error messages with document ID conflict #553
- trackme-limited/trackme-report-issues#554 - bug - Data Source tracking - trackmesplktags does not implement batch_save leading to potentially increased run time #554
- trackme-limited/trackme-report-issues#555 - bug - TrackMe UI - Entities filtering functions do not properly take into account the show Enabled True/False dropdown #555
- trackme-limited/trackme-report-issues#556 - bug - Flex Object UC - SOAR Services monitoring - non reachable SOAR shoudl lead to services being red immediately #556

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#550 - feature - Home interface drilldown & notable drilldown link - Allows submitting an object or alias URL param which filters out and opens automatically the entity overview, also add a drilldown_link to TrackMe Notables #550

11.1.35 Version 2.0.87 - build 1711995624 (01/04/2024)

- SHA256: 40e3bd2e52eed4c5e27e62b6e6386d13264284f65ac91a8cf61ebc6db8e9914b

High performance for high scale collections in TrackMe with pagination, server side filtering, KVstore batch_find & Tabulator theming

- This release introduces massive performance improvements in TrackMe, allowing notably high scale collections to be managed with ease.
- `REST API Pagination` - With TrackMe REST pagination capabilities and Tabulator capabilities, TrackMe can handle any number of entities in a collection without any performance degradation, allowing to deal with large collections of more than 100K entities.
- `Server side REST filtering` - TrackMe and the Tabulator now perform server side REST level filtering, this slightly optimises response time while filtering for entities with simple or complex filters even when working with very large collections.
- `Server side stats caching` - TrackMe now caches tenants and components statistics at the server level, allowing it to retrieve the stats in a fraction of the time it used to take.
- `Python native implementation for the Decision Maker and filter handling` - From this release, TrackMe handles entirely the Decision Maker phases and filtering handling in Python, without involving any Splunk searches, allowing to largely optimise the performance of these operations.
- `Background Python threading` - TrackMe also uses background side Python threading methods to maintain cached statistics, allowing to largely optimise performance run time of these operations and slightly reducing the usage of search slots in TrackMe.
- `KVstore batch_find and batch_update implementation` - This release also implements KVstore batch_find and batch_update for all user side interactions, allowing all entities update actions such as bulk edits or per entity/feature edit (priority update, etc) to take a fraction of the time it used to take in previous releases, no matters the number of entities in the collection.
- `Massive UI side performance improvements` - All these changes are reflected in TrackMe's UI by major reduction of load time, major reduction of the response time during entity updates, and globally slightly enhanced response times in TrackMe.
- `Tabulator theming` - This release also introduces new capabilities to update at the system and user level the look and feel of the Tabulator, allowing users to choose between 5 different themes, at the system and user level. (Dark Site, Dark, Light Site, Light, Light Modern)

Fixed issues:

- trackme-limited/trackme-report-issues#525 - bug - Data Hosts / Metric Hosts tracking (splk-dhm/splk-mhm) - Allow list KV transforms definitions are lacking the is_rex field, this will be corrected automatically with TrackMe's schema upgrade #525
- trackme-limited/trackme-report-issues#539 - bug - Data Source tracking (splk-dsm) - Allow Adaptive Delay field persistence is not honoured by hybrid trackers #539

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#505 - feature - Filter for acknowledgement comment content in main dashboard #505
- trackme-limited/trackme-report-issues#520 - feature - Implement systematic pagination mechanisms at the TrackMe's REST API level for high scale collections performance, implement server REST side filtering for high performance #520
- trackme-limited/trackme-report-issues#522 - change - Tabulator JS - Upgrade to version 6.1 #522
- trackme-limited/trackme-report-issues#523 - enhancement - Docs references feature for splk-dsm - Allows robust system wide default parameters, decomission related knowledge objects #523

- trackme-limited/trackme-report-issues#524 - feature - REST API TrackMe - Support for params GET based endpoints #524
- trackme-limited/trackme-report-issues#526 - enhancement - Blocklists for Feeds tracking (splk-dsm/dhm/mhm) - Allows the alias in addition with the object to choosen as the field to apply the blocklists against, code improvements #526
- trackme-limited/trackme-report-issues#534 - change - Decomission of the DataGen concepts replaced with more meaningful blocklist concepts for Feeds tracking #534
- trackme-limited/trackme-report-issues#535 - change - Splunk Python SDK 2.0.0 - deprecation explicit lib is required #535
- trackme-limited/trackme-report-issues#536 - enhancement - Dependencies verification - Add the Splunk Scientific package in dependencies verifications #536
- trackme-limited/trackme-report-issues#540 - enhancement - Data Sources tracking (splk-dsm) - Manual tags refreshed UI, new management endpoints and enhanced workflow #540
- trackme-limited/trackme-report-issues#541 - enhancement - REST API endpoints performance optimization - Implement KVstore batch_find and optimize all actions for much faster performances in REST API calls #541
- trackme-limited/trackme-report-issues#542 - enhancement - Tags policies tracker for Data Sources tracking (splk-dsm) - Immediately apply tags against the data collection in a batch_save manner for optimial performances and behaviour #542
- trackme-limited/trackme-report-issues#543 - feature - TrackMe's Vtenant UI and Home Tenants themes for Tabulator - Allow to define at the system and user level between 5 Tabulator theme (Dark Site, Dark, Light Site, Light, Light Modern) #543
- trackme-limited/trackme-report-issues#545 - change - Machine Learning models management - Ensures privately owned TrackMe ML models from the splunks-system-user are excluded from the Knowledge Bundle replication #545
- trackme-limited/trackme-report-issues#546 - change - Python and Splunk SDK 2.0.x - remove outdated or non necessary imports #546
- trackme-limited/trackme-report-issues#547 - change - trackmetenantstatus custom command - log in warning rather than error when there is not yet activity registered for a newly created tenant #547
- trackme-limited/trackme-report-issues#548 - enhancement - Maintenance mode & Maintenance Knowledge Database - Better handle user local time and show the local time information properly #548

11.1.36 Version 2.0.86 - build 1710525022 (15/03/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 30cc9a93c821b1d772b55ff8ed89aa4ba40de9394409b4792d9f8890c7d9d512

Fixed issues:

- trackme-limited/trackme-report-issues#529 - bug - Data Hosts tracking (splk-dhm) - Bulk edit for Ack enablement does not honour Ack expiration and type dropdowns (only affects this component) #529
- trackme-limited/trackme-report-issues#530 - bug - Data Sources tracking (splk-dsm) - Tags policies update through the UI breaks the policies structure #530
- trackme-limited/trackme-report-issues#531 - bug - Python function for central searching in Splunk - preview must be set to false or results may appear to be duplicated #531

- trackme-limited/trackme-report-issues#532 - bug - TrackMe performance counters for Trackers report inaccurate measures (trackmetrackerexecutor) #532

11.1.37 Version 2.0.85 - build 1710194416 (11/03/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: f79ca52d8eed0b4d8db4fad80373c4ea079aeae1ddb8cfd1bbf61cb1b5de0744

Fixed issues:

- trackme-limited/trackme-report-issues#527 - bug - splunkremovesearch - The local account should not be accounted against the license restriction (in Free Community edition, 1 remote account should be granted) #527
- trackme-limited/trackme-report-issues#528 - bug - Data Sources tracking (splk-dsm) - TrackMe REST API will not accept global_dcount_host as the min_dcount_field value #528
- trackme-limited/trackme-report-issues#521 - bug - Trackers and Licensing - If the user calls a tracker with "_tracker" part of its name, other reports (abstract, wrapper) are wrongly accounted against the license #521

11.1.38 Version 2.0.84 - build 1709505402 (03/03/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

Schema upgrade for TrackMe version 2.0.84

- This release includes a TrackMe schema upgrade which will automatically clean Outliers orphan records and orphan ML models.
- The schema upgrade is executed within the next 5 minutes after the upgrade, through the tenant's health tracker jobs.
- If there is a large amount of orphan models to be cleaned up, this can temporarily increase generate skipping searches for the health tracker as its execution would eventually take much longer than usual.
- After this, the health tracker will resume its normal execution and skipping searches for it will disappear.
- This process is fully automated, and there are no intervention required.

Schema upgrade issues for jump from old releases of TrackMe v2.0.x

- Different issues where addressed in this release to properly support migrating from very old versions of TrackMe v2.0.x. to this release.
- You can therefore safety migrate from any earlier version of TrackMe v2.0.x without expected issues.

- SHA256: 425ce2d470ec072f17289eedccbb94ce87115c5a063e92af4998a39ff4ed27da

Fixed issues:

- trackme-limited/trackme-report-issues#474 - bug - Workload (splk-wlk) - diff_search and other related deleted modification fields are not preserved in the KVstore record in other iterations of the metadata job (but preserved as indexed events, however). #474
- trackme-limited/trackme-report-issues#476 - bug - Alert action - The label is incorrect on the type of Ack for the TrackMe auto Ack action. #476
- trackme-limited/trackme-report-issues#482 - bug - Flex Library - The lastchanceindex object name should not include the current prefix. #482
- trackme-limited/trackme-report-issues#483 - bug - Flex Library - Cribl Logstream destination pressure UC should take into account yellow state metrics (value: 1) as well as green/red metrics. #483
- trackme-limited/trackme-report-issues#485 - bug - Hybrid Trackers - Creation via REST API endpoints should mirror UI default False options for break by host/splunk_server. #485
- trackme-limited/trackme-report-issues#486 - bug - Virtual Tenant UI - Overview duplicates entities in red state. #486
- trackme-limited/trackme-report-issues#489 - bug - Machine Learning models update screen - Depending on the component, the list of metrics is incorrect or incomplete, for Flex Objects, a free text update capability is required. #489
- trackme-limited/trackme-report-issues#492 - bug - Adaptive Thresholds for Data Sources (splk-dsm) - Error in the formula for review over time logic when defining the average of the 3 KPIs over 30d/7d/24h. #492
- trackme-limited/trackme-report-issues#494 - bug - Adaptive threshold (splk-dsm/splk-dhm) - The Adaptive threshold does not parse the pipe-delimited nature of anomaly_reason properly, thus it ignores entities affected by delay breached in addition to any other anomaly. #494
- trackme-limited/trackme-report-issues#497 - bug - Tenants Knowledge Objects permissions issue with Schema Upgrade - Read and Write permissions were inverted in the Schema upgrade in recent versions using standardized libs to manipulate KOs, this leads to created objects during the schema upgrade to eventually define inconsistent permissions. This update fixes it and also automatically fixes any existing tenant. #497
- trackme-limited/trackme-report-issues#500 - bug - Reject/remove special or unprintable characters when automatically adding newly discovered sources to TrackMe. #500
- trackme-limited/trackme-report-issues#501 - bug - Workload (splk-wlm) - Discrepancy and remaining issues when searches contain non-unicode or foreign characters. #501
- trackme-limited/trackme-report-issues#502 - bug - Data Host tracking (splk-dhm) - In some conditions, all sourcetypes red should be overridden by global host level thresholds (host shows red, should show green). #502
- trackme-limited/trackme-report-issues#504 - bug - Add quotes for object token in the dashboard "Adaptive delay threshold audit." #504
- trackme-limited/trackme-report-issues#508 - bug - Data Sources (splk-dsm) - Permanent entity deletion via the dedicated button through the modification screen performs a temporary deletion instead (but bulk permanent deletion works as expected). #508
- trackme-limited/trackme-report-issues#511 - bug - Virtual Tenants creation can fail during the upgrade process from an old enough version of TrackMe V2. #511
- trackme-limited/trackme-report-issues#518 - bug - REST API documentation - A few REST API endpoints incorrectly set the root uri (admin/write) for the resource_spl_example value #518

Enhancements, changes, and new features:

- trackme-limited/trackme-report-issues#472 - enhancement - Virtual Tenants - Major performance improvements in the loading time of the UI by avoiding a slot search to get TrackMe tenants in pure Python. #472

- trackme-limited/trackme-report-issues#475 - enhancement - Python backend search framework - A consistent and centralized approach to programmatic Pythonic searching in Splunk. #475
- trackme-limited/trackme-report-issues#477 - enhancement - Flex Library - Performance runtime improvements for the use case splk_license_usage_per_index. #477
- trackme-limited/trackme-report-issues#478 - bug - Flex Library - Wrong outlier metric name in OOTB use case cribl_logstream_pipeline. #478
- trackme-limited/trackme-report-issues#480 - enhancement - Flex Library - Queues filling use case set max_inactive_sec to 0, which is now allowed by splk-flx. #480
- trackme-limited/trackme-report-issues#481 - change - Alert naming default - Remove "custom on" from the alert default name in the input alert name. #481
- trackme-limited/trackme-report-issues#488 - feature request - Data Source tracking (splk-dsm) - Generate and ingest a global dcount host metrics that is not driven by the ingest and is closer to a simple dcount host. #488
- trackme-limited/trackme-report-issues#491 - feature - Flex Objects (splk-flx) - New use cases for Splunk Search Head Clusters (SHC) infrastructure monitoring. #491
- trackme-limited/trackme-report-issues#493 - feature request - Filter option for acknowledged entities. #493
- trackme-limited/trackme-report-issues#495 - enhancement - Adaptive Threshold for Feeds tracking (splk-dsm/splk-dhm) - Use max_auto_delay_sec in case the calculated threshold is higher than max_auto_delay_sec. #495
- trackme-limited/trackme-report-issues#496 - enhancement - PersistentFields command (KVstore batch update process) - For splk-dsm/splk-dhm, reject a KVstore record update request if the current KVstore value for data_last_time_seen is bigger than the upstream value from the tracker run. #496
- trackme-limited/trackme-report-issues#498 - feature - Data Sources tracking (splk-dsm) - Tags management - Major improvements to the tags policies for splk-dsm: Allow multi-match tags policies, new dedicated Python backend replacing the previous SPL native logic, enhanced UI elements for tags, enhancements tags policies management UI. #498
- trackme-limited/trackme-report-issues#499 - feature - Flex Objects / Workload (splk-flx/splk-wlk) - Allows more flexibility for charting type and mode selection in Flex Objects and Workload. #499
- trackme-limited/trackme-report-issues#506 - Feature - Entities in blue state show as alert in dashboard. #506
- trackme-limited/trackme-report-issues#509 - change - Virtual Tenants wizard - Disable splk-dhm/splk-dhm components by default unless requested. #509
- trackme-limited/trackme-report-issues#512 - feature - Outliers engine - New automated training feature, this allows automatically performing an ML model train operation when the backend attempts to render an out-of-date ML model to avoid false positives. #512
- trackme-limited/trackme-report-issues#514 - Bulk Acknowledgement unified for all components (Allows bulk Ack with expiration selection similarly to splk-dsm). #514
- trackme-limited/trackme-report-issues#515 - change - Tags for Data Sources (splk-dsm) - Include tags as part of minimal events indexed with trackme:state events by default #515
- trackme-limited/trackme-report-issues#516 - feature - Bulk actions - Provide various bulk actions capabilities for Outliers management (reset Outliers status, enable/disable Outliers detection, run mltrain / mlmonitor) #516
- trackme-limited/trackme-report-issues#517 - change - Logging - Outliers error message "The ML search is not yet available for rendering" should be rendered as warning rather than errors #517

11.1.39 Version 2.0.83 - build 1706721363 (31/01/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 7348149f074e719719bce7cad50c1861ee1c46646b03b1bd1294726c07924e92

Fixed issues:

- trackme-limited/trackme-report-issues#451 - bug - Hybrid Trackers / Flex Object trackers - latest_time is not used during tracker creation #451
- trackme-limited/trackme-report-issues#456 - bug - Logical Group - object is red even though logical group has sufficient green members #456
- trackme-limited/trackme-report-issues#459 - bug - Decision Maker - If both out of monitoring days and monitoring hours are True, a dplicated message is generated in status_message and status_message_json #459
- trackme-limited/trackme-report-issues#461 - bug - User Interface - In some conditions, the status message screen may not allow access to the footer management buttons due to the timeline component #461
- trackme-limited/trackme-report-issues#462 - bug - Data Hosts tracking (splk-dhm) - Outliers status should be looked up before the Decision Maker is called for the anomaly_reason and status_message to be reflected in the KVstore (which however has no impact on the detection) #462
- trackme-limited/trackme-report-issues#465 - bug - Data Hosts tracking (splk-dhm) - outliers_readiness is not preserved while running DHM trackers, leading the ML screen to display ML not ready message altrhough ML is actually ready #465

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#457 - feature - Virtual Tenant - Introducing a tenant alias concept, this allow assigning an alias per tenant which can be updated via the Configure UI, this value is now used in the Virtual Tenant UI rather than the tenant_id which is immutable #457
- trackme-limited/trackme-report-issues#458 - feature - Logical Groups - Extend Logical Groups to Flex Object (splk-flx) #458
- trackme-limited/trackme-report-issues#460 - enhancement - Logical Groups - Major rewrite of the backend management for Logical Groups which is now full taken in charge by the Decision Maker, we also automatically detect and purge orphans logical group members (via the health tracker), major improvements and immediate change reflection via the Decision Maker #460
- trackme-limited/trackme-report-issues#463 - enhancement - SmartStatus - Extend SmartStatus to Flex Object, various improvements to the SmartStatus backend for automatic search retry, improved search management and search use cases for all components, more consistent approach with normalized ML UC #463
- trackme-limited/trackme-report-issues#464 - enhancement - Virtual Tenants UI - show/hide spinner while loading tenant's knowledge objects until API call is over #464
- trackme-limited/trackme-report-issues#466 - change - Virtual Tenants UI - Disable by default the splk-mhm when creating a new feeds tenant, unless instructed otherwise in the wizard #466
- trackme-limited/trackme-report-issues#467 - enhancements - Flex Objects (splk-flx) - Improving inline documentation and added max_sec_inactive as well as time_factor in ML models generation #467
- trackme-limited/trackme-report-issues#468 - enhancement - Flex Object library (splk-flx) - Improving the Splunk DMA builtin use case #468
- trackme-limited/trackme-report-issues#469 - enhancement - Flex Object (splk-flx) - Allowing a max_sec_inactive = 0 to disable automated red trigger based on detected inactivity #469

- trackme-limited/trackme-report-issues#470 - feature - Logical Groups - Add new management screen allowing to add / update / delete Logical Groups with easier access and management #470
- trackme-limited/trackme-report-issues#471 - feature - Health Tracker - Implement a new context called inspect_collection which ensures that object statuses in KVstore collections are always consistent with the Decision Maker, this also addresses some specific use case where there could be an inconsistent object_state in the KVstore collection #471

11.1.40 Version 2.0.82 - build 1705991568 (23/01/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: c7b039911bf8f9506096b5b1b03f98edf9a53d6ea0d4b7f22edfc68e80b66935

*Fixed issues:**

- trackme-limited/trackme-report-issues#452 - bug - Adaptive delay audit dashboard - remaining typo and dead link in the navigation menu #452
- trackme-limited/trackme-report-issues#453 - bug - Maintenance mode & Maintenance Knowledge DataBase - Prevents failure to load the Knowledge DataBase UI when the maintenance mode was enabled through a REST call #453
- trackme-limited/trackme-report-issues#454 - bug - Maintenance mode & Maintenance Knowledge DataBase - Retro-compatbility for older version of Firefox due to issues with the datetime-local input selector #454

Enhancement, changes and new features:

- trackme-limited/trackme-report-issues#455 - enhancement - Splunk Remote Search - Improve logging and error handling when testing / configuration / using Splunk Remote Search in TrackMe #455

11.1.41 Version 2.0.81 - build 1705906378 (22/01/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 3c33e18c7fb3920523eebaf795dfb02c9f80220292353ed6fc99f8d44c5b452d

Fixed issues:

- trackme-limited/trackme-report-issues#447 - bug - Typo in the new adjustements dashboard for Adaptive audit #447

Enhancement, changes and new features:

- trackme-limited/trackme-report-issues#448 - enhancement - Adaptive delay adjustment audit dashboard user experience improvements #448
- trackme-limited/trackme-report-issues#449 - enhancement - Acknowledgment management REST API endpoints - code and behaviour enhancements, allows listing all Ack, better management and new API endpoint for the UI purposes #449
- trackme-limited/trackme-report-issues#450 - enhancement - UI Acknowledgement - Enhanced Ack management screen relying on direct REST integration for faster and richer user experience #450

11.1.42 Version 2.0.80 - build 1705650542 (19/01/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 06bba3369ad7fa358e026bcbec3bc7e604b20cc47e648308b63ad1944d9fc0b3

Fixed issues:

- trackme-limited/trackme-report-issues#446 - change - Splunk Base failure to properly initiate Appinspect vetting request #446

11.1.43 Version 2.0.79 - build 1705620290 (18/01/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: bf29cb3fe4d65e1decbb958dfe2b32c625a3950ed58d2e38fadb4dc3bb9b2cd5

Fixed issues:

- trackme-limited/trackme-report-issues#439 - bug - Logging system - missing log_level search time extraction for alert actions logs #439
- trackme-limited/trackme-report-issues#440 - bug - Bulk edit Acknowledgment - The Ack period selected is interpreted in seconds instead of days when doing Ack through Bulk editing #440
- trackme-limited/trackme-report-issues#441 - bug - Acknowledgement backend logging - Avoid improperly generating the message "no object state information could be retrieved" #441
- trackme-limited/trackme-report-issues#442 - bug/enhancement - Decision Maker for Data Hosts tracking (splk-dhm) - logic adjustementfor entity level thresholds management #442

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#437 - Feature request - Allow to define if automated acknowledgements should be sticky or unsticky within TrackMe's builtin alert action #437
- trackme-limited/trackme-report-issues#438 - enhancement - Flex Object Library - Last Chance Index use case improvements #438
- trackme-limited/trackme-report-issues#443 - feature request - Data Source monitoring (splk-dsm) - Overview chart series selection improvements to allow more choices and alertnatively hide the delay and/or latency series #443
- trackme-limited/trackme-report-issues#444 - feature - Adaptive Threshold - Adding a new Audit dashboard focusing on reviewing the adjustments made by TrackMe #444
- trackme-limited/trackme-report-issues#445 - enhancement - Logging backend - Retrieve report and macros details and log them before attempting to delete knowledge objects when requested to do so #445

11.1.44 Version 2.0.78 - build 1705310134 (14/01/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 71ae1311bc9fc6bd87b01f60da8b80c727094766d9c92fc9f0b8a8769eac7bd6

Fixed issues:

- trackme-limited/trackme-report-issues#429 - bug - Adaptive Delay backend - prevent UnboundLocalError errors when mstats returned no results in some conditions #429
- trackme-limited/trackme-report-issues#430 - bug - trackmepersistentfields (TrackMe persistent fields) - prevent exception message="could not convert string to float: " if tracker_runtime is unexpectly empty #430
- trackme-limited/trackme-report-issues#431 - bug - Cribl Logstream Flex Object use cases for inputs and outputs health check should take into account green/yellow/red returns from Cribl #431
- trackme-limited/trackme-report-issues#432 - bug - Data Hosts/Metric Hosts (splk-dsm/splk-mhm) - Avoid error "gen_metrics" failed with exception 'NoneType' object has no attribute 'get' #432
- trackme-limited/trackme-report-issues#435 - bug - Adaptive Delay (Data Sources / Data Hosts tracking - splk-dsm/splk-dhm) - TrackMe does not honour properly allow_adaptive_delay #435
- trackme-limited/trackme-report-issues#436 - enhancement - Adaptive Delay (splk-dsm/splk-dhm) - Improved logic and logging for the management of ML based adaptive delay tresholding #436

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#433 - enhancement - Flex Object Library - Splunk Queues filling use case review and improvements #433
- trackme-limited/trackme-report-issues#434 - feature - Flex Object Library - New use case for Splunk Search Heads key activity tracking #434

11.1.45 Version 2.0.77 - build 1704838956 (09/01/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: fe1d3723cd2a091781a71992b13255884275170ee8d23b5b22f9b2ca6e375706

Fixed issues:

- trackme-limited/trackme-report-issues#425 - bug - Workload / Flex Objects - muliselect dropdown should automatically refresh when the time range is changed #425
- trackme-limited/trackme-report-issues#428 - bug - Decision Maker - regression with custom wdays / hours ranges parameters not properly taken into account #428

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#392 - enhancement - Future data detection - Take into account a negative latency as a likely data in the future use case and turn entity orange as expected when future detection is operated against _time #392
- trackme-limited/trackme-report-issues#423 - enhancement - Status message improvements with a new native JSON structure and enhanced viz mode #423
- trackme-limited/trackme-report-issues#424 - enhancement - CIM compliance - extend week days & hours ranges concepts to CIM compliance tracking #424
- trackme-limited/trackme-report-issues#426 - enhancement - Cribl Logstream - Flex Object library use cases improvements, enhanced syntax and improved logic, better use ML Outliers rather than basic thresholds for some of the use cases, globally improved use cases #426
- trackme-limited/trackme-report-issues#427 - enhancement - Flex Object library - review use case splk_splunk_cloud_svc_usage_by_app and base threshold on ML Outliers #427

11.1.46 Version 2.0.76 - build 1704492296 (05/01/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: d6c01dc0e902605422c7375cc920172e01595d3f44689fb5f8cc8e03d0dc117f

Fixed issues:

- trackme-limited/trackme-report-issues#422 - bug - Decision Maker - regression when red on outliers or red on sampling is turned off on the tenant but an an actual outliers or sampling alert is active #422

11.1.47 Version 2.0.75 - build 1704475839 (05/01/2024)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 5436020d80f4cb2c7cc55ead436e3c3d4ccd0102fe6797fa87233f9903de573f

Fixed issues:

- trackme-limited/trackme-report-issues#416 - bug - Timezone offset management - properly handle time information management honoring users & system timezone offsets #416
- trackme-limited/trackme-report-issues#420 - bug - trackmesplkoutlierstrain - this command should not call directly the component register when raising an exception (leading to unexpected error logging) #420

Enhancement, changes and new features:

- trackme-limited/trackme-report-issues#410 - enhancement - Workload (splk-wlk) - Improved and safer scheduler and introspection tracking logic to avoid missing execution traces and false positive execution delayed alerts #410
- trackme-limited/trackme-report-issues#411 - enhancement - Outliers Adaptive Thresholding (splk-dsm/splk-dhm) - adjustments of the logic for enhanced behaviour #411
- trackme-limited/trackme-report-issues#398 - Feature Request: Acknowledgement overlay in Tabulator tables (right click context popover) #398
- trackme-limited/trackme-report-issues#414 - feature - Add row click popover context for Outliers and Data Sampling #414
- trackme-limited/trackme-report-issues#415 - feature - Introducing TrackMe decision maker backend, this new concepts replaces SPL based complex evaluations to define the status of TrackMe entities depending on the context and components, for a safer and more robust decision making #415
- trackme-limited/trackme-report-issues#417 - feature - Allows enabling/disabling at the tenant level the adaptive delay threshold feature (via a Virtual Tenant account switch) #417
- trackme-limited/trackme-report-issues#418 - enhancement - Flex Object - Complete popover context menu (Outliers status, status message and anomaly_reason) #418
- trackme-limited/trackme-report-issues#419 - change - Data Sources tracking (splk-dsm) - Do not include the remote account information in the definition of the alias #419
- trackme-limited/trackme-report-issues#421 - enhancement - Workload (splk-wlk) - Improved logic for detection and purge of any duplicated entities in Workload #421

11.1.48 Version 2.0.74 - build 1703259037 (22/12/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: d2a7e5c5447741cc166256589174e31eb01d9e658fcedd54f24264f9c5f92f15

Fixed issues:

- trackme-limited/trackme-report-issues¢12 - bug - Workload - Regression issue with outliers definition when performing the schema migration, leading to invalid eval and interrupting the Workload detection - #412

11.1.49 Version 2.0.73 - build 1703095950 (20/12/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: ddb21e231b5ba7d4f0fc31306ce538a9466b1801e3f3dfe67fcdccba633663f2

Fixed issues:

- trackme-limited/trackme-report-issues#408 - bug - Virtual Tenants UI - regression on the listing of reports in TrackMe Tenants Operational health statuses #408
- trackme-limited/trackme-report-issues#409 - bug - Virtual Tenants UI - Tenants Operational health statuses can show empty last_exec under some conditions #409

11.1.50 Version 2.0.72 - build 1703080417 (20/12/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 99c071e498886209e5a231f9443f96741e5ed7026fbb9aa1ded365774c6a174a

Fixed issues:

- trackme-limited/trackme-report-issues#379 - bug - Data Source tracking (splk-dsm) - regression in the simulate thresholds screen due to the migration to restricted summary state events in TrackMe 2.0.68 #379
- trackme-limited/trackme-report-issues#380 - bug - Configuration UI - title wording is not consistent for thresholds default configuration management #380
- trackme-limited/trackme-report-issues#381 - bug - Workload (splk-wlk) - Outliers are set with lower breached enabled unexpectly with elapsed KPI, shema version upgrade will address this issue automatically #381
- trackme-limited/trackme-report-issues#387 - fix - Avoid permissions issues for the Health tracker shema upgrade when handling TrackMe's knowledge upgrade #387
- trackme-limited/trackme-report-issues#394 - bug - Workload/Flex (splk-wlk/splk-flx) - Metric dropdown populating search use static -24h earliest time range #394
- trackme-limited/trackme-report-issues#395 - bug - Outliers - Permissions issues for Power users in different advanced Outliers related actions such as resetting or force training models #395

- trackme-limited/trackme-report-issues#399 - bug - Flipping status detection - Non unicode chars can lead to continuous discovery #399
- trackme-limited/trackme-report-issues#401 - bug - Elastic processing backend - error message local variable 'count_processed' referenced before assignment when no entities to be processed #401
- trackme-limited/trackme-report-issues#403 - bug - User Interface - Auto-refresh should be disabled automatically when performing bulk edition & inline edition #403

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#372 - change - Allow assigning an Ack to a blue state entity #372
- trackme-limited/trackme-report-issues#373 - enhancement - Least privileges & permissions - Some ingest related activities (health tracker & Notables) require the edit_tcp capability, which can be avoided by controled TrackMe capabilities #373
- trackme-limited/trackme-report-issues#374 - feature - Manage permanently deleted entities through a builtin UI screen from components #374
- trackme-limited/trackme-report-issues#376 - enhancement - Flex Objects - Add a group filter option in the Tabulator #376
- trackme-limited/trackme-report-issues#382 - enhancement - Workload (splk-wlk) - Take into account status delegated_remote_error as parts of scheduler excution failures, existing trackers will be updated automatically by the schema upgrade #382
- trackme-limited/trackme-report-issues#383 - change - Workload (splk-wlk) - Increase the Smart-Status earliest time from -24h to -7d for the execution error search #383
- trackme-limited/trackme-report-issues#384 - feature - Adaptive delay - Introducing the Adaptive delay feature to allow managing automatically delay threshold value for Data Sources and Hosts tracking (splk-dsm/splk-dhm) #384
- trackme-limited/trackme-report-issues#385 - enhancement - Outliers - Add more context information in the isOutlierReason field when an Outlier is triggered #385
- trackme-limited/trackme-report-issues#386 - feature - Machine Learning Outliers - Introducing the confidence concept to reduce false positive and identify low confidence models and entities #386
- trackme-limited/trackme-report-issues#388 - feature request - Overview Table: Column for human readable thresholds #388
- trackme-limited/trackme-report-issues#389 - change - User Interfaces - Increase 90% width modal screens to 96% of the screen as a basis for enhanced user experience #389
- trackme-limited/trackme-report-issues#390 - enhancement - Flex Objects / Workload / CIM compliance (splk-flx/splk-wlk/splk-cim) - Include the Outliers column in the Tabulator view #390
- trackme-limited/trackme-report-issues#391 - feature - Maintenance Knowledge DataBase - Intoducing a concept of a maintenance knowledge database, which can be used in association with the maintenance mode or independently to influence the SLA calculations by injecting knowledge of planned or unplanned operations that have lead to an impact on TrackMe entities #391
- trackme-limited/trackme-report-issues#393 - feature - Add Ack duration and Ack type as customizable options for bulk edit actions #393
- trackme-limited/trackme-report-issues#396 - feature - Introducing a new command "trackmesplk-outliersgetdata" to get easier access to Outliers results #396
- trackme-limited/trackme-report-issues#397 - change - Virtual Tenants - code improvements for the managment of boolean options when creating tenants #397
- trackme-limited/trackme-report-issues#400 - feature - Outliers - Allowing to set the time_factor to none which enables TrackMe to apply a simpler LowerBound/UpperBound with no seasonability variations #400

- trackme-limited/trackme-report-issues#402 - change - Workload (splk-wlk) - define the Outliers by default based on time factor with no seasonability for elapsed based metrics for enhanced results #402
- trackme-limited/trackme-report-issues#404 - feature - Workload (splk-wlk) - Automatically process a diff of the 3 main search Metadata (search, earliest, latest) and attempt to identify the user who performed the change and the time of the change when detecting a saved search version change #404
- trackme-limited/trackme-report-issues#406 - enhancement - Virtual Tenant - Health Status reporting - enhanced Tabulator view #406
- trackme-limited/trackme-report-issues#407 - change - Tenants & knowledge objects creation ownership - switch the default owner from admin to nobody #407

11.1.51 Version 2.0.71 - build 1700472127 (20/11/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 174868c183c78036487881576355a9bc7de228e6295811caf5ac8d3428af8fc8

Fixed issues:

- trackme-limited/trackme-report-issues#364 - bug - Typo in distinct count #364
- trackme-limited/trackme-report-issues#370 - bug - Replica tenants - Do not attempt to perform the replica tracker for a disabled tenant #370

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#365 - enhancement - Workload (splk-wlk) - Handle use cases when Splunk incorrectly logs scheduler activity with no user context, introducing a new dynamic get owner retrieval component, scheduler trackers are updated automation during the schema upgrade #365
- trackme-limited/trackme-report-issues#366 - enhancement - Review of timeout policies in TrackMe, ensures all service definition and Python request define a timeout #366
- trackme-limited/trackme-report-issues#367 - enhancement - Flex Objects library - Improvement of the splk_kvstore_size use case for Flex #367
- trackme-limited/trackme-report-issues#368 - enhancement - Feeds tracking - Improving the status message for latency & delay alerts (including durations, incude both thresholds, round to 3 decimals) #368
- trackme-limited/trackme-report-issues#369 - feature - Data Sources tracking (splk-dsm) - Allow choosing between any of the dcount metrics to define minimal distinct count host thresholds rather than the default mandatory choice (latest_dcount_host_5m) #369

11.1.52 Version 2.0.70 - build 1700087843 (15/11/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 4690b0653623f7a96b9347a493a24806c3120bf098fd95fcd2a75d939b369f24

Fixed issues:

- trackme-limited/trackme-report-issues#362 - bug - healthtracker - errors generating the expected audit events in trackme_audit for the health tracker itself due to a regression #362
- trackme-limited/trackme-report-issues#363 - bug - last_exec is reported as null in the component register audit events #363

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#360 - enhancement - When missing the right permissions and capabilities, show a clearly understandable message for admins to take actions #360
- trackme-limited/trackme-report-issues#361 - feature - Workload component (splk-wlk) - Introducing the overgroup feature, allowing to override the per application grouping and allowing to colocate multiple Search tiers in the same tenant #361

11.1.53 Version 2.0.69 - build 1699886135 (13/11/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: dd9ca1df32eb23008db8d128f7dee9665562224e93aa2fe5e384af64ffc3808e

Fixed issues:

- trackme-limited/trackme-report-issues#352 - bug - Shared Elastic - minor logging errors #352

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#353 - enhancement - Elastic Dedicated - improve the manage screen rendering #353
- trackme-limited/trackme-report-issues#354 - feature - Migrate component register tracker run time to TrackMe's metric store for faster queries, and better retention than from the _internal only #354
- trackme-limited/trackme-report-issues#355 - feature - Bootstrap icons / Emoji ascii compatibility mode - provide a configurable option for both Vtenants UI / Home UI to switch between Emoji ascii based statuses icons and Bootstrap based icons, this addresses compatibility issues for some customers on Wndows not supporting Emoji ascii fonts #355
- trackme-limited/trackme-report-issues#356 - enhancement - Flex Objects library - Enhancement search for the DMA use case #356
- trackme-limited/trackme-report-issues#357 - feature - Flex Object library - New use case for Splunk large lookup files detection #357
- trackme-limited/trackme-report-issues#359 - change - Increase minimal time betweem two ML training per entity from 24 hours to 7 days for TrackMe footprint reduction #359

11.1.54 Version 2.0.68 - build 1699407909 (08/11/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 939013c951a1884369efeb934f12cad919c1d41a22411de8fd1c09a1f3a25ee7

Note

SLA to metrics migration

- This new release introduces the migration for SLA metrics to metrics based indexes instead of the previous SLA calculations based on the state events
- This allows slightly reducing the size and volume of state events, reducing storage and licensing costs for TrackMe, as well as performing much faster queries and allowing much longer retentions
- If you wish to backfill the existing SLA knowledge after you have migrated to TrackMe 2.0.68, run the following Splunk search to backfill SLA metrics using `mcollect`
- We made the choice not to automate the SLA migration such that you can decide to do it or not, and control its execution process

*Use this search after the migration to TrackMe 2.0.68 to backfill SLA metrics (this search can takes a while, think about modifying indexes if necessary, reduce the timerange if you do not care about all metrics, and send this to the background for the best control of its excution)**

```
index=trackme_summary sourcetype="trackme:state" object_category=* object=* key=*␣
↪tenant_id=* current_state=* earliest=-90d
| fields _time, tenant_id, object_category, object, alias, current_state, monitored_
↪state, priority, key
| bucket _time span=1m
| stats latest(current_state) as object_state, latest(alias) as alias,␣
↪latest(monitored_state) as monitored_state, latest(priority) as priority by _time,␣
↪tenant_id, object_category, object, key

``` convert string status to numerical ```
| eval object_state=case(
    object_state = "green", 1,
    object_state = "red", 2,
    object_state = "orange", 3,
    object_state = "blue", 4,
    1=1, 5
    )

``` rename to the metric_name target, key is objct_id in the new metrics schema ```
| rename object_state as trackme.sla.object_state, key as object_id

``` use mcollect to backfill metrics ```
| mcollect index=trackme_metrics split=t tenant_id, object_category, object, object_
↪id, alias, monitored_state, priority
```

Note

Introducing the TrackMe stats events minimal mode

- This new release introduces a major reduction of the TrackMe state events (sourcetype=trackme:state) in terms of volume and size, as well as a consistent schema
- This change was made possible in association with the SLA to metrics migration
- You can control in the Configuration screen the mode of generation, minimal (default) or full (as prior to 2.0.68), as well as the list of fields to allow (minimal mode) or block (full mode)
- These options are available in the *General Configuration* tab (Minimal state events, allowlist fields (minimal), In full, block list fields)
- There are no actions required to benefit from this change, unless you had some custom reporting or alerting based on the state events, in which case you should review your use cases and adapt them to the new schema
```

Fixed issues:

- trackme-limited/trackme-report-issues#339 - bug - Virtual Tenant UI regression on dynamic theme system level preferences application (flex cards should turn red properly) #339
- trackme-limited/trackme-report-issues#342 - bug - Health Tracker (inactive entities tracking) - handle if tracker_runtime is null #342
- trackme-limited/trackme-report-issues#343 - bug - Health Tracker (inactive entities tracking) - offline abstract macros should not exclude permanently deleted entities #343
- trackme-limited/trackme-report-issues#344 - bug - command trackmepersistentfields - logic assignement error in persistent fields definition #344
- trackme-limited/trackme-report-issues#346 - bug - Elastic Sources - Addressing various issues in this release (eventcount not parsed with from lookups, results duplicated in simulation, code weakness) #346
- trackme-limited/trackme-report-issues#348 - bug - Virtual Tenants - Issues with underscores in tenant identifiers when created through the REST API #348
- trackme-limited/trackme-report-issues#350 - bug - Virtual Tenant - Enabling a previously tenant that has splk-dhm/wlk will report a failure on enabling some macros #350
- trackme-limited/trackme-report-issues#351 - bug - Data Sources tracking (splk-dsm) - regression on honoring not including the host in the tstats root break by fields #351

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#337 - change - Tabulator update to version 5.5.2 #337
- trackme-limited/trackme-report-issues#338 - feature - Flex Objects - Introducing the Splunk practices use cases for the Flex Objects component #338
- trackme-limited/trackme-report-issues#340 - feature / enhancements - Introducing major improvements for the Elastic Sources Shared backend with parallel muti-processing, automated job max runtime definition, ordering of execution and improved logging #340
- trackme-limited/trackme-report-issues#341 - feature/enhancement - SLA metrics - For enhanced performances and better management, SLA calculations are moving to true metrics #341
- trackme-limited/trackme-report-issues#345 - enhancement - Logging - standardize run_time logging to 3 decimals for all TrackMe backends #345
- trackme-limited/trackme-report-issues#349 - feature - State events minimal mode - Major reduction in the state events volume and size to reduce the impact on storage and license (migrates splk-dhm/mhm to full metrics, introducing the state event minimal configuration to ingest minimal state events) #349

11.1.55 Version 2.0.67 - build 1698669312 (30/10/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 4c6c90fcad4bf91dbdc17c434d19e4c00de5f18dab7860c18d4f72b9c059fb66

Fixed issues:

- trackme-limited/trackme-report-issues#329 - bug - Persistentfields - Python exception if the mtime or tracker_runtime is not in the expected format #329
- trackme-limited/trackme-report-issues#330 - bug - Workload (splk-wlk) - Non ASCII characters in knowledge objects names such as foreign accents are not properly handled #330

- trackme-limited/trackme-report-issues#331 - bug - Maintenance mode - Failure when attempting to enable the maintenance mode #331
- trackme-limited/trackme-report-issues#332 - bug - Missing arguments in searchbnf.conf for the Data Sampling tracker executor #332
- trackme-limited/trackme-report-issues#333 - bug - Flex Objects / CIM compliance - missing file-handler rotation in Python lib leads to the log file not being rotated #333
- trackme-limited/trackme-report-issues#336 - bug - Flex Objects - properly handle some problematic escaped rex sequences when running remote searches #336
- trackme-limited/trackme-report-issues#304 - bug - Virtual Tenant UI - Dropdown text search is not working (affects initial creation and RBAC update modal screens) #304

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#334 - feature - Adding the new command trackmesplkoutliersexpand to expand ML outliers results for further processing #334
- trackme-limited/trackme-report-issues#335 - feature - Adding a new expending streaming command for Flex Objects (trackmesplkflxexpandextra), its purpose is to expand the extra_attributes for new use cases management in the Flex Object library #335

11.1.56 Version 2.0.66 - build 1698184235 (24/10/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 2593a02f2a8f2a475a6e0318bddd48d94b31fc014a8441cfef10c1168dc495f6

Fixed issues:

- trackme-limited/trackme-report-issues#328 - bug - Data Sources tracking (splk-dsm) - The overview single average latency and percentile 95 incorrectly show the same metric (regression from 2.0.65) #328

11.1.57 Version 2.0.65 - build 1698103284 (24/10/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: e14d7b9e4e198cf79680c2ea6dd598ab3b2b58450077127bc3dbba4f4bedd728

Fixed issues:

- trackme-limited/trackme-report-issues#324 - bug - Data Hosts tracking (splk-dhm) - regression on alias value definition at discovery #324
- trackme-limited/trackme-report-issues#325 - bug - Ack - wrong audit message #325
- trackme-limited/trackme-report-issues#326 - bug - Flex Objects library - error in default cron schedule for lastchanceindex use case #326
- trackme-limited/trackme-report-issues#327 - bug - Data Sources tracking (splk-dsm) - If adding host in the custom break by field, the hybrid tracker incorrectly defines entities #327

11.1.58 Version 2.0.64 - build 1698044829 (23/10/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 2a0981f700bf2d3c759bb37839578e35876dd5aa7947b17aaf0f15b30d3b816e

Fixed issues:

- trackme-limited/trackme-report-issues#317 - bug - Data Sources/Data Hosts tracking (splk-dsm/splk-dhm) - fix discrepency between banner delay and single form delay as well as the Tabulator delay (ensures last delay is refreshed against now) #317
- trackme-limited/trackme-report-issues#318 - bug - Data Hosts tracking (splk-dhm) - Issue in the offline abstract macro called by the health tracker (execution fails due to missing pipe when called) #318
- trackme-limited/trackme-report-issues#320 - bug - Data Hosts tracking (splk-dhm) - Alias is not correctly persisted when the entity goes out of the trackers range #320

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#319 - change - Data Sources/Hosts tracking (splk-dsm/splk-dhm) - decomission the delayed entities tracker which features are now better handled by the health tracker #319
- trackme-limited/trackme-report-issues#321 - enhancement - Data Sources/Hosts tracking (splk-dsm/splk-dhm) - maintain the generation of the delay metric (lag_event_sec) when entities are out of the range of trackers #321
- trackme-limited/trackme-report-issues#322 - enhancement - Data Sources / Data Hosts tracking (splk-dsm/splk-dhm) - Extend the auto-lagging screen to include both ingest latency and delay concepts #322
- trackme-limited/trackme-report-issues#323 - enhancement - Data Sources/Hosts tracking - show the delay metric (lag_event_sec) in the overview timechart #323

11.1.59 Version 2.0.63 - build 1697650503 (18/10/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 1c506fe8b6535228631f8e5c72a817bb00e0a6fac7da886912e30c9932fb2ce6

Fixed issues:

- trackme-limited/trackme-report-issues#310 - bug - ML Outliers - Avoid generating an error message when attemping to load the period of exclusion if not a list (add safety) #310
- trackme-limited/trackme-report-issues#313 - bug - Workload (splk-wml) - TrackMe should not attempt to perform replacement for app stanza criterias any more if target is remote as these are now explicit in the creation process #313
- trackme-limited/trackme-report-issues#314 - bug - Ingest - Since the migration to INGEST_EVAL in 2.0.60, some expected key indexed fields in trackme:state and others are not indexed any longer #314
- trackme-limited/trackme-report-issues#315 - bug - SmartStatus - ingested alert actions are lacking the tenant_id and object_category fields, breaking the indexed key consistency scheme in TrackMe #315

- trackme-limited/trackme-report-issues#316 - bug - Fix splunkd WARN message "with request data but no Content-Type: header; not parsing POST arguments" #316

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#311 - feature - Allow defining the default sharing level (app or global) when TrackMe creates or manages Splunk Knowledge Objects #311
- trackme-limited/trackme-report-issues#312 - change - INGEST_EVAL - Add a safety fail back condition for ingest evals defining the index target #312

11.1.60 Version 2.0.62 - build 1697551318 (17/10/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: b2f8e6fb03716ce1d9950ca39be0d40c6ded740e7a64035fcf34ef2a3cc9ea24

Fixed issues:

- trackme-limited/trackme-report-issues#303 - TrackMe bug report - Hybrid Tracker cron no applied in the report schedule #303
- trackme-limited/trackme-report-issues#307 - bug - ML Outliers - Auto Correct should not allow lowerBound and upperBound to be equals #307

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#306 - change - Dark theme compatibility - Enable dark theme compatibility in app.conf #306
- trackme-limited/trackme-report-issues#305 - change - ML Outliers - Disable by default the generation of the latency based model for Feeds which is not a great candidate in most of the use cases #305
- trackme-limited/trackme-report-issues#308 - enhancement - ML Outliers - inherit earliest and latest from the time range picker rather than explicitely for the ML rendering commands #308
- trackme-limited/trackme-report-issues#309 - feature - ML Outliers - Capability to add or delete a period of time for exclusions in the ML models training #309

11.1.61 Version 2.0.61 - build 1697150459 (12/10/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

Note

Inheritance support for RBAC

- This release introduces support for roles inheritance for RBAC in TrackMe
- Virtual Tenants are Splunk Remote Accounts can be accessed, managed and administrated by inheriting roles according to your configuration

- SHA256: ad69875eba15dd7680add23d5fba72131916ea04ec862d04df3479fd9e56bf21

Fixed issues:

- trackme-limited/trackme-report-issues#294 - bug - Workload / Flex Objects - When more than a single Outliers model is in anomaly, the status_message comes back null as the macro did not expect the multivalue nature of these fields #294
- trackme-limited/trackme-report-issues#300 - bug - SLIM Packing for Splunk Cloud Classic - spec files are not instructing the partitioning properly #300
- trackme-limited/trackme-report-issues#301 - bug - Data Sources tracking (splk-dsm) - UI token manipulation related issues leads to a null search eating the user disk quota under some circumstances #301

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#290 - enhancement - Flex Objects (splk-flx) - improvement of the use case splk_splunk_enterprise_cluster_peers_status (calculate buckets inbalance deviation and alert) #290
- trackme-limited/trackme-report-issues#291 - enhancement - Flex Objects (splk-flx) - improvement of the use case splk_splunk_enterprise_cluster_status #291
- trackme-limited/trackme-report-issues#292 - enhancement - Flex Objects (splk-flx) - New use case for rolling tracking of license usage per index and pool #292
- trackme-limited/trackme-report-issues#293 - bug/enhancement - Machine Learning Outliers detection - Auto correct logic defects leads to avoid generating true positive outliers #293
- trackme-limited/trackme-report-issues#295 - enhancement - Flex Object - Cribl integration UC improvements for health inputs and outputs to remove false positive #295
- trackme-limited/trackme-report-issues#296 - enhancement - Flex Objects use cases library - UC splk_queues_filling improvement - avoid generating alerts when the queues are inactive
- trackme-limited/trackme-report-issues#297 - change - Remove owner=admin as the default in default.meta to avoid Enterprise customers with no admin users to be impacted by the default behavior of TrackMe #297
- trackme-limited/trackme-report-issues#298 - enhancement - Roles Based Access Control (RBAC) - Support inheritance globally in TrackMe #298

11.1.62 Version 2.0.60 - build 1695681952 (25/09/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 859bd778ac65750a5e4eb05cc3c11a884ddbdedd9fffcb1e33fafd54909dd71b

Fixed issues:

- trackme-limited/trackme-report-issues#289 - bug - SLIM partitioning causes ingest issues in Splunk Cloud Classic experience, requires explicit stanza placement in spec files #289

11.1.63 Version 2.0.59 - build 1695559981 (24/09/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 67d7a8466af72c68705cfeeca6504589ad732bc01c0961f8597f1e1236059d44

Fixed issues:

- trackme-limited/trackme-report-issues#283 - bug - trackmetrackerhealth (Health Tracker) - Hybrid tracker macro update in the KVstore should only happen if the currently known definition differs from system #283
- trackme-limited/trackme-report-issues#284 - bug - TrackMe alert actions (notable, SmartStatus, Ack) - failures to run actions in the context of a strict least privilege service account owning the tenant #284

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#285 - change - Health Tracker - Improve logging for inactive entities tracking for splk-dsm/splk-dhm #285
- trackme-limited/trackme-report-issues#286 - change - entity_info API endpoints - always return the object and key value in the response to recycle values as needed and ease further processing #286
- trackme-limited/trackme-report-issues#287 - change - Reduce the timerange considered by the delayed entity trackers to 24h by default, after this time inactive entities are taken into account by the health tracker #287
- trackme-limited/trackme-report-issues#288 - enhancement - Data Sources and Hosts tracking (splk-dsm/splk-dhm) - Ensures that the delayed entities tracker updates last entity Metadata information even if the target search did not return any results #288
- trackme-limited/trackme-report-issues#261 - enhancement - Provide cURL examples for each REST API endpoints in the REST API auto-documentation #261

11.1.64 Version 2.0.58 - build 1694716015 (14/09/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 90119b248d9a1a820a335254a3d994ab4b45a7839f2468c7d087d3604208a91a

Fixed issues:

- trackme-limited/trackme-report-issues#281 - bug - splunkremotesearch - Non meaningful Python exception when calling a non existing account #281

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#282 - enhancement - Workload (splk-wlk) - Workload Virtual Tenant creation wizard improvements #282

11.1.65 Version 2.0.57 - build 1694635429 (13/09/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 5febf5ab3f93abf7ce8b0218f192374bdd5e3094d6150cc98f1e1a9b6126470a

Fixed issues:

- trackme-limited/trackme-report-issues#275 - bug - Data Hosts tracking (splk-dhm) - error when deleting entity on a per entity basis (list index out of range) #275
- trackme-limited/trackme-report-issues#277 - bug - Data Hosts tracking (splk-dhm) - error when trying to update monitoring hours of a given entity due to wrong REST API endpoint path #277

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#276 - feature - Introducing the CMDB integrator feature - Allows queriying an external third data source for contextual information in TrackMe tenants #276 - See: https://docs.trackme-solutions.com/admin_guide_cmdb_integration.html
- trackme-limited/trackme-report-issues#279 - change - RBAC - Optimisation for role membership verification #279
- trackme-limited/trackme-report-issues#280 - enhancement - Workload (splk-wlk) - Virtual Tenant creation wizard improvements, split the search filters to be specific in the UI for Scheduler / Introspection / Splunk Cloud SVC #280

11.1.66 Version 2.0.56 - build 1694411312 (11/09/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: d7d5ed282cda25375216de5e47eb770c6b8bc34d5d1c89354d7e123923374879

Fixed issues:

- trackme-limited/trackme-report-issues#264 - bug - typo in RBAC ownership view #264
- trackme-limited/trackme-report-issues#266 - bug - Workload (splk-wlk) - When creating the main tracker, the SVC usage should be part of the avg_svc_usage is trackmegenjsonmetricsmissing from the calls in #266
- trackme-limited/trackme-report-issues#268 - bug/change - INGEST_EVAL migration for all summary events and metric generation workflow, this migration is performed to overcome a Splunk Cloud Classic DMC deployment bug when deploying applications using transforms to override the DEST_KEY - While this issue is Splunk Cloud responsability, this is not going to be fixed in any acceptable timeline, TrackMe therefore turns to a different approach which is not affected by this #268
- trackme-limited/trackme-report-issues#271 - bug - Audit events - When using custom indexes per tenant, audit events remain generated in the default TrackMe configured index rather than the tenant specific index #271
- trackme-limited/trackme-report-issues#273 - bug - Benchmark Burn Test tends to time out for long run queries in Splunk Cloud due to time out reach in Splunk Cloud Web reverse proxy #273

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#265 - feature - TrackMe SVC usage audit dashboard for Splunk Cloud customers #265
- trackme-limited/trackme-report-issues#267 - change - Workload - Switch the default stats mode for the dropdown to max rather than latest to ensure visibility in most use cases #267
- trackme-limited/trackme-report-issues#269 - feature - Flex Object library (splk-flx) - New use case to track SVC consumption in Splunk Cloud by application #269
- trackme-limited/trackme-report-issues#270 - change - Flex Objects (splk-flx) - Licensing restriction increase to 32 trackers for Enterprise Edition customers #270
- trackme-limited/trackme-report-issues#272 - change - Ack behaviour default system wide configuration when returning to green - enables purging Ack by default when returning to non green if non sticky #272
- trackme-limited/trackme-report-issues#274 - enhancement - Feeds tracking (splk-feeds) - synchronize macros knowledge hybrid trackers attributes when the macros are updated in Splunk #274

11.1.67 Version 2.0.55 - build 1693924977 (05/09/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: b22a72485ba6b09d0b01bb0b19c4faf265aafd3e30a41f076fdc4eba75322b2d

Fixed issues:

- trackme-limited/trackme-report-issues#263 - bug - Virtual Tenants UI for Feeds tracking - indexes discovery feature does not work as expected due to Javascript regression when configured at the Virtual Creation phase #263

11.1.68 Version 2.0.54 - build 1693744485 (03/09/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 1a745c8ae615620d3c526e94742908897a0aa1e85dfa8454b1fb48d84a5b808e

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#42 - feature - Data Sources tracking (splk-dsm) - Tags for Data Source monitoring - Remove tags linked to a tag policy when the tag policy is removed #42
- trackme-limited/trackme-report-issues#259 - bug/enhancement - Virtual Tenants UI optimizations with a new unified endpoint for a faster and safer user experience, this also addresses issues observed in Splunk Cloud classic only #259
- trackme-limited/trackme-report-issues#260 - change - Update moment.js to version 2.29.4
- trackme-limited/trackme-report-issues#262 - enhancement - Virtual Tenants UI - Alphabetically sort tenants in the UI if no positions are preset for the user profile #262

Fixed issues:

- trackme-limited/trackme-report-issues#256 - bug - Data Hosts / Metrics Hosts (splk-dhm/splk-mhm) - Cannot filter on tags within the Tabulator #256
- trackme-limited/trackme-report-issues#257 - bug - Data Hosts tracking (splk-dhm) - Max global latency & delay per entity should match the highest relevant value between all sourcetypes related to it #257
- trackme-limited/trackme-report-issues#258 - bug - logging issues when checking permissions for trackmeload/trackmetenantstatus (not logging the right user name) #258

11.1.69 Version 2.0.53 - build 1692273340 (17/08/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 42654231000a4bae75d40d4d9317babd93b8cc5e080e8d2367ebc5d45365333f

Enhancement, changes and new features:

- trackme-limited/trackme-report-issues#251 - feature - Data Hosts / Metric hosts preset the alias equal to the raw object without the key(s) addition #251

- trackme-limited/trackme-report-issues#252 - feature - Flex Objects - New use cases for CPU and Memory infrastructure tracking via Splunk introspection #252
- trackme-limited/trackme-report-issues#253 - feature - Data Hosts and Metric Hosts tracking - enhancement for tags enrichment purposes #253

11.1.70 Version 2.0.52 - build 1692002557 (14/08/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 11dc12c922f8005257c1d8bc5eccf0e8d0f3b848b0881a6eabe42ea56944850f

FIxed issues:

- trackme-limited/trackme-report-issues#247 - bug - Replica tenants - logic issues when having more than a single replica tracker with the same component leading to the incorrect purge of replica records #247
- trackme-limited/trackme-report-issues#248 - bug - Replica tenants - The Flex object inactive entities tracker should not be created for Replica tenants #248

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#249 - feature - Allow pre-defining default owner and defaults admin/power/roles in TrackMe general configuration for the Virtual Tenants user interfaces #249

11.1.71 Version 2.0.51 - build 1691618697 (09/08/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 90b21e5cffa2ec91e968def2b857d083f46eb6c0fecfe5cc4f423d3d87168617

Fixed issues:

- trackme-limited/trackme-report-issues#245 - bug - All components - In large scale scenarios with more than 50k entities on a per tenant/component basis, the Tabulator is limited to 50k entities due to the underneath oneshot SDK search #245
- trackme-limited/trackme-report-issues#246 - bug - Data Sources/Data Hosts tracking (splk-dsm/splk-dhm) - In some rare conditions, a null search can be generated and run unexpectly impacting user quota #246

11.1.72 Version 2.0.50 - build 1691356328 (06/08/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 0147f78edb580e0a67229ee7eb42699e211d1b5791e844e6eb280d52fcf66043

Fixed issues:

- trackme-limited/trackme-report-issues#242 - bug - SOAR integration custom command trackme-splksoar - issues rendering a POST response rendered as a list #242

- trackme-limited/trackme-report-issues#243 - bug - SOAR integration - pagination issues in some circumstances restricts the number of entities returned #243

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#244 - feature - SOAR integration - Manage Automation Brokers High Availability with TrackMe, update SOAR Assets automatically when an Automation Broker is inactive to an active counter part - High Availability for SOAR Automation Brokers via TrackMe #244

11.1.73 Version 2.0.49 - build 1691080561 (03/08/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 6bd3ea567f0465a4f9e388c04cd95cb839b17594f3adb84619b01d00311de1b2

Fixed issues:

- trackme-limited/trackme-report-issues#236 - bug - SLA dashboard - Dropdowns populating search is using static 24 hours range rather than timerange picker from the dashboard #236
- trackme-limited/trackme-report-issues#240 - bug - Flex Objects (splk-flx) - UC Splunk Cloud SVC usage - ensure to generate metrics of SVC usage if the licensed SVCs is null #240

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#237 - enhancement - Flex Objects (splk-flx) - Allows the priority to be defined at the phase of the Flex Tracker execution #237
- trackme-limited/trackme-report-issues#238 - change - Workload (splk-wlk) - Increase the last_seen filter to last 90m for the metadata retrieval #238
- trackme-limited/trackme-report-issues#239 - enhancement - Flex Objects (splk-flx) - Include pool_quota_gb metrics in the license pool usage tracking #239
- trackme-limited/trackme-report-issues#241 - enhancement - Flex Objects (splk-flx) - Simplification and better code for the Deployment Server tracking use case #241

11.1.74 Version 2.0.48 - build 1690973605 (02/08/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: cc03ecd66725692e332ad6604ce5c3baddd4f3336883ffb65ff7aaad7ee67a42

Fixed Issues:

- trackme-limited/trackme-report-issues#219 - bug - Feeds Tracking (splk-dsm) - The delayed entities trackers re-generates non merged entities in a hybrid context of merged / non merged and does not track merged entities properly #219
- trackme-limited/trackme-report-issues#221 - bug - Virtual Tenants UI - Addresses some issues with theming and user preferences, more consistent management of preferences
- trackme-limited/trackme-report-issues#222 - bug - Workload (splk-wlk) - error in trackmesplk-wlkgetreportsdefstream for metadata retrieval when using remote target multiple load balanced search head targets #222
- trackme-limited/trackme-report-issues#224 - bug - Workload (splk-wlk) - simulation fails for Splunk Cloud SVC when running through the UI due to incorrect quote #224

- trackme-limited/trackme-report-issues#225 - bug - Workload (splk-wlk) - Back button not working from create hybrid trackers #225
- trackme-limited/trackme-report-issues#230 - bug - incorrect report names for the mltrain reports when adding to the report state register component #230
- trackme-limited/trackme-report-issues#231 - bug - Workload (splk-wlk) - Under some circumstances an entity generating execution errors could lead to incorrect definition of the user and looping with multivalue fields gnerating bad objects #231

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#223 - enhancement - Outliers engine - When requesting reset ML, the endpoint performs a search, if the max concurrency is searched on the Search Head this can lead to an unexpected failure, ensures we attempt automated retry if it is the case before failing permanently if necessary #223
- trackme-limited/trackme-report-issues#226 - feature - Flex Object (splk-flx) - new use case for tracking KVstore collections size #226
- trackme-limited/trackme-report-issues#227 -enhancement - Allows a service account owner to be using the minimal level of permissions and capabilites to own and run properly TrackMe objects #227
- trackme-limited/trackme-report-issues#228 - enhancement - Python code sanitization, autoformatting and unit testings for automated bug identification #228
- trackme-limited/trackme-report-issues#229 - enhancement - Fix any hard coded reference to localhost for the communication with splunkd using best practice Python splunkd uri inherited URI #229
- trackme-limited/trackme-report-issues#232 - enhancement - Data Sources/Data Hosts tracking (spl-dsm/splk-dhm) - Health tracker maintains untracked entities which are out of the scope of any tracker to update and maintain state consistency #232
- trackme-limited/trackme-report-issues#233 - feature - Flex Object (splk-flx) - Use Case for Splunk Enterprise license pool usage tracking #233
- trackme-limited/trackme-report-issues#234 - enhancement - Splunk SOAR integration - Allows a least privilege approach for SOAR interactions #234
- trackme-limited/trackme-report-issues#235 - change - Feeds Tracking - delayed entities tracker switch to False for break by splunk_server and host which is the default now in TrackMe #235

11.1.75 Version 2.0.47 - build 1690295356 (25/07/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: bcdf0903d3fe531786764ff009911ade7a1a3ca779193733ea3771806d6ef0e3

fixed issues:

- trackme-limited/trackme-report-issues#220 - bug - regression in trackmeapiautodocs introduced in 2.0.46 when Splunk App for SOAR is not installed on the Search Tier #220

11.1.76 Version 2.0.46 - build 1690266086 (25/07/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: c62b857fc20638a97e3b17fd03e9cb5f6fb0d76c5027c8d95ba5cb661bc88fb0

fixed issues:

- trackme-limited/trackme-report-issues#210 - bug - Flex Objects (splk-flx) - When a given entity turns red due to inactivity, a summary state event should also be generated to properly influence the SLA percentage calculation #210
- trackme-limited/trackme-report-issues#213 - bug - Virtual Tenants - endpoint post_vtenants_accounts should not return an exception when there are no tenants yet #213
- trackme-limited/trackme-report-issues#215 - bug - Workload (splk-wlk) - status_message can come back null in some circumstances #215
- trackme-limited/trackme-report-issues#216 - bug - Virtual Tenants - deleting a component should clean up the vtenant summary record #216

Enhancements, changes & new features:

- trackme-limited/trackme-report-issues#211 - feature - Flex Objects - Splunk SOAR native integration (UCs for SOAR monitoring) #211
- trackme-limited/trackme-report-issues#214 - feature - Flex Object (splk-flx) - lastchanceindex use case for Splunk data_collection #214
- trackme-limited/trackme-report-issues#217 - change - Data Hosts tracking - automatically restrict the indexes to the main and internal indexes for splk-dhm if indexes is left unconfigured at the tenant creation phase with Hybrid tracker creation enabled (click next disease) #217

11.1.77 Version 2.0.45 - build 1689676533 (18/07/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 2b394e1617836c6e5757cac1ad9c2896d5d1340e008d23d403c47ba52c23f78d

Fixed issues:

- trackme-limited/trackme-report-issues#201 - bug - Flex UC splk_splunk_enterprise_cluster_status - wrong term Down rather than Stopped #201
- trackme-limited/trackme-report-issues#206 - bug - Flipping REST API issue (hitting Splunk CIM) #206
- trackme-limited/trackme-report-issues#207 - bug - CIM Tracking - regression in ML Outliers model generation #207
- trackme-limited/trackme-report-issues#208 - bug - CIM Tracking - deletion of entities in bulk fails since 2.0.40 #208
- trackme-limited/trackme-report-issues#209 - bug - CIM Tracking - failure to generate the initial discovered flipping event #209

Enhancements and new features:

- trackme-limited/trackme-report-issues#202 - feature - Flex Objects - Cribl Logstream use cases for deep monitoring of Cribl Logstream in TrackMe #202
- trackme-limited/trackme-report-issues#203 - enhancement - Flex Objects - allow multiselect metrics in entity overview #203
- trackme-limited/trackme-report-issues#204 - enhancement - Flex Object - preset the alias of the entity as the short value of the object (without the group) and allows defining custom values for the alias at the entity discovery phase of the tracker #204

- trackme-limited/trackme-report-issues#205 - enhancement - Flex Objects (splk-flx) - Manage inactive entities #205

11.1.78 Version 2.0.44 - build 1689362642 (14/07/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 7602e39ffcdfa299100fb33e0b25363a11ae25da6a5d3ec5051a8bad3bbb235c

Enhancement and new features:

- trackme-limited/trackme-report-issues#191 - feature - Flex Objects tracking - Introducing the Flex Objects use case library and major component features improvements #191

11.1.79 Version 2.0.43 - build 1689342033 (14/07/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

Hint

Workload upgrade:

- review the release special instructions if you are using the workload component

- SHA256: 2af481f61b93eaa3c5811856e29871742c50ea176f59446ef39948cac5075cdf

Fix issues

- trackme-limited/trackme-report-issues#195 - bug - Workload (splk-wlk) - In some circumstances the Splunk scheduler logs can lack app and user context leading to the creation of new entities in case of execution errors detected #195
- trackme-limited/trackme-report-issues#198 - bug - Data Sources (splk-dsm) - enable/disable entities in bulk fails due to regression (object not defined) #198
- trackme-limited/trackme-report-issues#199 - bug - Outliers - regression due to the ds_account field decommisioning leading to failures in generating Outliers rules for new entities #199
- trackme-limited/trackme-report-issues#200 - bug - Remove the characters length restrictions in the Vtenant configuration in UCC #200

Enhancements and new features:

- trackme-limited/trackme-report-issues#197 - enhancement - All components - Execution of TracKers via the UI and when permited via RBAC should be executed as the system user to avoid user related context to impact results consistency #197

Special intructions or notes for this release:

- To benefit from the fix of issue #195 related to the Workload, the scheduler tracker should be deleted and re-created for each Workload tenant
- This can be achieved via the UI, or via REST API

11.1.80 Version 2.0.42 - build 1688984590 (10/07/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 7d4cf2359d629d9f56dd121ab03e981efe0fb1eb2bf98225f1cce6fcb7a882db

fixed issues:

- trackme-limited/trackme-report-issues#190 - bug - Workload - the main tracker does not include the count_ess_notable metrics in the metrics summary popup #190
- trackme-limited/trackme-report-issues#192 - bug - Data Sources (splk-dsm) - Clear state & run sampling resets the entity for DSM #192
- trackme-limited/trackme-report-issues#193 - bug - The number of currently existing trackers should show up in the management UI for Flex Objects and Workloads #193
- trackme-limited/trackme-report-issues#194 - bug - Data Hosts Tracking (splk-dhm) - summary level sourcetype state does not honour properly the latency/delay independently as expected #194

11.1.81 Version 2.0.41 - build 1688538958 (05/07/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: 9ee5384747ee3d022a3a3d8aaf0ae3794dffb9a501de0ce9e9c4a4002ac593a4

Fixed issues:

- trackme-limited/trackme-report-issues#189 - bug - splk-dsm (Data Source) bulk edit regression for enable/disable monitoring via bulk edit due to change #182 #189

11.1.82 Version 2.0.40 - build 1688457335 (04/07/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: a163d0b1b0892edecfd09784b39b6ae0ba13aad275b54355d86c92ccb1fa950e

Fixed issues:

- trackme-limited/trackme-report-issues#182 - bug - All components - handle entities changes via their unique identifier rather than the object (handles bad entities with unexpected special characters) #182
- trackme-limited/trackme-report-issues#183 - bug - Performance issues at large scale of entities for Flex / Workload trackers #183
- trackme-limited/trackme-report-issues#186 - bug - splunkremotesearch - splunk-system-user and admin users should be RBAC granted for all configured accounts #186
- trackme-limited/trackme-report-issues#187 - bug - Virtual Tenants UI - count=0 is missing from some rest searches, leading to avoid returning all results from the upstream search (ex: user account selection) #187

Enhancements, changes and new features:

- trackme-limited/trackme-report-issues#184 - change - Flex Object - allows automated width for the Status description in the Tabulator #184
- trackme-limited/trackme-report-issues#185 - feature - SmartStatus for Workload entities, allows the SmartStatus to handle Workload UCs as well as capturing Splunk internal events with a least privileges approach (no need for users to be able to access to the _internal index to review internal scheduler errors through the SmartStatus control) #185
- trackme-limited/trackme-report-issues#188 - enhancement - REST API logical groups - allows updating min percent if an existing group via REST without having to have to provide the list of current members #188

11.1.83 Version 2.0.39 - build 1687757627 (26/06/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA256: d855a2c6467e7a1d97abfb783a91883a2205b0b59102bef0471aa74aacf49303

Fixed issues:

- trackme-limited/trackme-report-issues#176 - bug - User Interface - Using DSM "Show disabled entities" filter clears the "Filter field or function" dropdown #176
- trackme-limited/trackme-report-issues#177 - bug - Data Hosts Tracking (splk-dhm) - truncation in trackme:state for entities with a very large amount of related sourcetypes #177

Enhancements and new features:

- trackme-limited/trackme-report-issues#178 - enhancement - Do not allow deleting or cloning Virtual tenants accounts in the Configuration UCC UI #178
- trackme-limited/trackme-report-issues#179 - enhancement - Check the Splunk Remote account connectivity and authentication at the creation / edit step in the Configuration UI (UCC framework) #179
- trackme-limited/trackme-report-issues#181 - change - Data sources/Data hosts (splk-dsm/splk-dhm) - sets break by splunk_server/host by default to False #181

11.1.84 Version 2.0.38 - build 1687154702 (19/06/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

Hint

Metrics expansion mode and Workload upgrade:

- review the release special instructions for more information about the metrix expansion mode change in this release
- review the release special instructions if you are using the workload component

- SHA256: 90a6d51fc68b5e78b2b5a523d834fabbc2eea18cbcefb78e34f3f1ac793de04b

Fixed issues:

- trackme-limited/trackme-report-issues#151 - bug - Workload - the app filter provided as an example in the tracker search constraint can lead to the non detection of some use cases of execution errors #151
- trackme-limited/trackme-report-issues#152 - bug - failure to populate tenants dropdowns in SLA and Data Sampling Dashboard studio dashboards due to earlier changes in trackmeload output #152
- trackme-limited/trackme-report-issues#153 - bug - Workload - trackmesplkwlkgetreportsdefstream should call select url function to properly handle multiple Splunk endpoints for a remote account #153
- trackme-limited/trackme-report-issues#154 - bug - error in endpoint /splk_dsm/ds_get_dsm_sampling_obfuscation_mode due to obfuscation Virtual tenant account change #154
- trackme-limited/trackme-report-issues#155 - bug - Logical group auto group command - flow logic when adding single member groups #155
- trackme-limited/trackme-report-issues#158 - bug - Data Hosts (splk-dhm) - logic flow in trackme_dhm_tracker_abstract macro does not preserve per host max latency/delay and does therefore leads to no honouring these settings #158
- trackme-limited/trackme-report-issues#150 - bug - Elastic Sources - metrics generation fails for raw/from based Elastic Sources definition (shared and dedicated) #150
- trackme-limited/trackme-report-issues#159 - bug - Common Information Model tracking (splk-cim) - button horizontal alignment issue in TrackMe UI #159
- trackme-limited/trackme-report-issues#163 - bug - Vtenant UI - Prevents the running spinner to be removed (due to auto-refresh) before then end of the operation when executing long run operations such as tenants creation #163
- trackme-limited/trackme-report-issues#164 - enhancement - avoids running trackers during the Virtual Tenant creation phase to reduce time required for its creation (multiops endpoints) #164
- trackme-limited/trackme-report-issues#165 - bug - HTML duplicated ids, issues in label definition, various UI related issues #165
- trackme-limited/trackme-report-issues#166 - bug - Workload (splk-wlk) - indentation issues when creating Workload trackers, failures in the tracker creation UI to check remote connectivity #166
- trackme-limited/trackme-report-issues#167 - bug - Acknowledgments - typo when creating Ack manually leads to unstricky rather than unsticky status for Ack, prevent their proper expiration #167
- trackme-limited/trackme-report-issues#168 - bug - Workload (splk-wlk) - Orphan tracker enhancements from Issue#117 were lost during the transition to least privileges #168
- trackme-limited/trackme-report-issues#171 - bug - missing props definition for the command trackmeprettyjson #171

New features and enhancements:

- trackme-limited/trackme-report-issues#156 - enhancement - Logical Groups - round the percentage of current group status commitment, allows filtering on Blue entities for splk-dsm/dhm/mhm #156 enhancement - User Interface minimal mode and context popup approach to improve readibility for all eligible components #157
- trackme-limited/trackme-report-issues#160 - enhancement - Health Tracker - automatically detect when a TrackMe object no longer exists and cleanup the register knowledge #160
- trackme-limited/trackme-report-issues#161 - bug - mlmonitor reports are not registered with the right name in the component register #161
- trackme-limited/trackme-report-issues#162 - enhancement - Workload - Adding the notable type tracker to allow tracking the number of Enterprise Security notable events per correlation search #162

- trackme-limited/trackme-report-issues#169 - enhancement - Flex Objects (splk-flx) - The tracker wizard should allow trackers not returning any entities to be created, as lookling only bad conditions can be a use case #169
- trackme-limited/trackme-report-issues#170 - enhancement - splunkremotesearch - handle Splunk automated extractions when fields resuting from remote events are not consistents #170
- trackme-limited/trackme-report-issues#172 - enhancement - Workload (splk-wlk) - provides a deeper visibility with a 3 periods metrics approach of scheduled activity #172
- trackme-limited/trackme-report-issues#173 - enhancement - Tabulator component upgrade 5.5 #173
- trackme-limited/trackme-report-issues#174 - enhancement - Bulk edit - when clicking on all entities selector, ensures selected entities honour current filters including header filters and add the count number of entities to be impacted in the bulk edit screen #174
- trackme-limited/trackme-report-issues#175 - enhancements - Logs inspector dashboard - fixes and improvements for the log inspector dashboard #175

Special instructions for this release:

Default metrics expanded mode

- This new release introduces a change in the visibility of eligible components (splk-wlk/splk-cim/splk-flx/splk-dhm/splk-mhm) regarding the default expansion of the metrics column and/or JSON formatted context columns
- From 2.0.38, the column is not expanded any longer, a user would see a "right click for popup" message instead, right clicking will provide the expected information in a more context menu, providing better global readibility when dealing with many entities
- At anytime in the UI, one can switch to the expanded mode by selecting the "full" visibility in the mode selector dropdown in TrackMe
- Also, TrackMe administrators can update the default visbility mode when the tenant is loaded by editing the Vtenant preferences (Configuration / Virtual Tenant account) and defining the default mode for **UI prefs - expand metrics**

Workload (splk-wlk)

Workload notable tracking:

- If you are using Splunk Enterprise Security, you way want to track the notable activity which is a new type of Workload tracker added to this release
- The notable track will monitor the number of notable events generated per ES correlation search, and add a new metric "count_ess_notable" which can be used for context and investigations, or Outliers detection eventually.
- To add the new notable tracker, run the following command: (replace mytenant with the tenant name, define account according to your context)

```
| trackme mode=post url="/services/trackme/v2/splk_wlk/admin/wlk_tracker_create" body=
↪"{'tenant_id': 'mytenant', 'account': 'local', 'tracker_type': 'notable'}"
```

- Also, you need to add the "count_ess_notable" metric in the main tracker, you can either edit manually the wrapper main report or follow the next instructions to re-create a brand new main tracker
- TrackMe schema version update will not perform this for you as you filter preferences (app filters for instance in the root constraints) would be lost and because this can run on a remote target, this cannot be added to a local macro for persistence)

Workload behaviour enhancements:

If you are using the Workload component, you may want to perform the following actions to benefit from some specific updates:

step 1: - Go in the tenant, click on "Manage: Workload Trackers" - Locate the main tracker, and click on Delete

step 2: - Go in a search, run the following command (replace mytenant by the tenant_id, the account is not relevant for main tracker and should always be local):

```
| trackme mode=post url="/services/trackme/v2/splk_wlk/admin/wlk_tracker_create" body=
↪"{'tenant_id': 'mytenant', 'account': 'local', 'tracker_type': 'main'}"
```

step 3: - Search the following macro: "trackme_wlk_set_status_tenant_<tenant_id>" - Update its content to: (replace the occurences of <tenant_id> with the name of your tenant)

```
lookup local=t trackme_wlk_orphan_status_tenant_<tenant_id> object OUTPUT orphan,␣
↪mtime as orphan_last_check | eval orphan_last_check=case(isnotnull(orphan_last_
↪check), strftime(orphan_last_check, "%c"))
| lookup local=t trackme_wlk_versioning_tenant_<tenant_id> object OUTPUT cron_exec_
↪sequence_sec
``` init a status 1```
| eval status=1
``` If there are execution errors detected, status=2, we use periods data from 60m to␣
↪4h to 24h, the JSON metrics will not contain the metric if it equals to 0 ```
``` Therefore, if a given search generating errors if fixed and has frequent␣
↪executions, it likely will turn green in the next 60m from the deployment of the␣
↪fix ```
| eval status=case(
count_errors_last_60m=0, status,
count_errors_last_4h=0, status,
count_errors_last_24h=0, status,
count_errors_last_60m>0 OR count_errors_last_4h>0 OR count_errors_last_24h>0, 2,
1=1, status
)
``` If there are skipping searches, define two levels of alerting, less than 5% is 3␣
↪(orange), more is 2 (red) ```
``` we base the calculation over the 24 period (suffix last_24h) - this can be␣
↪customised up to your preferences if you wish to used the additional periods ```
| eval status=case(
isnum(skipped_pct_last_24h) AND skipped_pct_last_24h>0 AND skipped_pct_last_24h<5, 3,␣
↪isnum(skipped_pct_last_24h) AND skipped_pct_last_60m>0 AND skipped_pct_last_24h>=5,␣
↪2,
1=1, status
)
``` If we detected the search as an orphan search (not period related) ```
| eval status=if(orphan=1, 2, status)
``` Calculate the delta in sequence between now and the last execution compared␣
↪against the requested cron schedule sequence, add 1h of grace time, detect if the␣
↪execution has been delayed ```
| eval status=if(cron_exec_sequence_sec>0 AND ( now()-last_seen > (cron_exec_sequence_
↪sec + 3600) ), 2, status)
``` Set a brief status description, a more granular description will be provided with␣
↪the anomaly_reason and status_message fields ```
| eval status_description=case(status=1, "normal", status=2, "degraded", status=3,
↪"warning", 1=1, "unknown")
```

11.1.85 Version 2.0.37 - build 1686088225 (06/06/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

Hint

Roles Based Access Control enhancements:

- From version 2.0.34, TrackMe implements a new strict **least privilege Role Bbased Access Control**
- A new role **trackme_power** is now builtin in TrackMe and designed to allow performing updates to entities of a granted tenant
- Access to TrackMe is driven by builtin capabilities provided by TrackMe builtin roles (trackme_user, trackme_power, trackme_admin)
- The least privilege approach implemented since this release allows granular access to TrackMe without requiring problematic capabilities which have security implications (list_settings, list_storage_passwords)
- TrackMe user interfaces automatically adapt its content and provided options depending on the profile of the current user, a normal user will for instance not see write or admin related actions
- TrackMe REST API endpoints are now classified in 3 groups, user level endpoints, write level endpoints and admin level endpoints
- The TrackMe `splunkremotesearch` also supports Roles Based Access Control, a user calling a given account must be a member of any of the listed roles in the account configuration to be granted access to this account
- For retro-compability purposes, TrackMe will allow access to an existing Remote account that has no RBAC roles setup yet to typical admin users in addition with TrackMe builtin roles (admin, sc_admin, trackme_user, trackme_power, trackme_admin)
- When TrackMe is **upgraded**, the migration of existing tenant is automatically performed by the **schema version management**, *Upgrading TrackMe*
- For more **information**, see: *Role Based Access Control and ownership*

- SHA256: 5a0b110099a769abea3af34cb61f4725c686d0554fcf89a1e63ce98486a7cc23
- trackme-limited/trackme-report-issues#147 - bug - splk-dsm (Data Source) - regression when call run sampling on a particular entity due to obfuscation change in v2.0.36 #147
- trackme-limited/trackme-report-issues#148 - bug - splk-dhm (Data Hosts) - the title of the modal screen incorrectly mentiones splk-mhm #148
- trackme-limited/trackme-report-issues#145 - enhancement: Higher width for the status column (which can truncated under Ack circumstances) #145
- trackme-limited/trackme-report-issues#149 - bug - Workload / Flex (splk-wlk/splk-flx) - Truncate long description to avoid impacting the view screen #149

11.1.86 Version 2.0.36 - build 1685947587 (05/06/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

Hint

Roles Based Access Control enhancements:

- From version 2.0.34, TrackMe implements a new strict **least privilege Role Bbased Access Control**
- A new role **trackme_power** is now builtin in TrackMe and designed to allow performing updates to entities of a granted tenant
- Access to TrackMe is driven by builtin capabilities provided by TrackMe builtin roles (trackme_user, trackme_power, trackme_admin)
- The least privilege approach implemented since this release allows granular access to TrackMe without requiring problematic capabilities which have security implications (list_settings, list_storage_passwords)
- TrackMe user interfaces automatically adapt its content and provided options depending on the profile of the current user, a normal user will for instance not see write or admin related actions
- TrackMe REST API endpoints are now classified in 3 groups, user level endpoints, write level endpoints and admin level endpoints
- The TrackMe `splunkremotesearch` also supports Roles Based Access Control, a user calling a given account must be a member of any of the listed roles in the account configuration to be granted access to this account
- For retro-compability purposes, TrackMe will allow access to an existing Remote account that has no RBAC roles setup yet to typical admin users in addition with TrackMe builtin roles (admin, sc_admin, trackme_user, trackme_power, trackme_admin)
- When TrackMe is **upgraded**, the migration of existing tenant is automatically performed by the **schema version management**, *Upgrading TrackMe*
- For more **information**, see: *Role Based Access Control and ownership*

- SHA256: f0c47447023dca0daf9cb5e5e434dc077a0e8c71bfc75233d73717268eef33a3
- trackme-limited/trackme-report-issues#135 - bug - Data Sampling - Creating an mstats based Elastic Source breaks the Data Sampling query execution #135
- trackme-limited/trackme-report-issues#136 - bug - Outliers engine - When reseting Outliers models, TrackMe should also reset the data outliers records for a more consistent approach #136
- trackme-limited/trackme-report-issues#137 - bug - Acknowledgement - Updating Ack fails due to Python regression introduced in 2.0.34 #137
- trackme-limited/trackme-report-issues#138 - enhancement - Add a new command utility trackmeautogroup to allow auto management of logical group association from an upstream SPL logic #138
- trackme-limited/trackme-report-issues#139 - bug - SmartStatus - incorrect timechart search in UC delay causes no results to be found #139
- trackme-limited/trackme-report-issues#140 - enhancement - SmartStatus - rely on latest known event rather than latest - - trackme-limited/trackme-report-issues#141 - known ingest when defining the earliest for UC delay/latency for better results when looking at an offline entity #140
- trackme-limited/trackme-report-issues#141 - enhancement - vtenants accounts integration scheme for more flexible tenant level configuration management #141
- trackme-limited/trackme-report-issues#142 - enhancement - Improvements and minor fixes for user interfaces behaviours when user is a power user (capability: trackmepoweroperations) #142
- trackme-limited/trackme-report-issues#143 - bug - splk-dhm (Data Host Tracking) - TrackMe does not honor properly the per sourcetype policy due to evaluation of the state at the table loading time which avoids taking into account the status per sourcetype #143

- trackme-limited/trackme-report-issues#144 - feature - Introducing the TrackMe Configuration Manager (TCM) to provides CI/CD capabilities for TrackMe #144

Additional notes: - In version 2.0.36, the data sampling obfuscation macro is deprecated and decommissioned automatically, it is replaced by a much more flexible approach relying on the tenant account setting - To enable the obfuscation mode for a given tenant post-migration, go in Configuration / vtenant preferences and edit the tenant to enable the obfuscation mode

11.1.87 Version 2.0.35 - build 1684913150 (24/05/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

Hint

Roles Based Access Control enhancements:

- From version 2.0.34, TrackMe implements a new strict **least privilege Role Bbased Access Control**
- A new role **trackme_power** is now builtin in TrackMe and designed to allow performing updates to entities of a granted tenant
- Access to TrackMe is driven by builtin capabilities provided by TrackMe builtin roles (trackme_user, trackme_power, trackme_admin)
- The least privilege approach implemented since this release allows granular access to TrackMe without requiring problematic capabilities which have security implications (list_settings, list_storage_passwords)
- TrackMe user interfaces automatically adapt its content and provided options depending on the profile of the current user, a normal user will for instance not see write or admin related actions
- TrackMe REST API endpoints are now classified in 3 groups, user level endpoints, write level endpoints and admin level endpoints
- The TrackMe `splunkremotesearch` also supports Roles Based Access Control, a user calling a given account must be a member of any of the listed roles in the account configuration to be granted access to this account
- For retro-compability purposes, TrackMe will allow access to an existing Remote account that has no RBAC roles setup yet to typical admin users in addition with TrackMe builtin roles (admin, sc_admin, trackme_user, trackme_power, trackme_admin)
- When TrackMe is **upgraded**, the migration of existing tenant is automatically performed by the **schema version management**, *Upgrading TrackMe*
- For more **information**, see: *Role Based Access Control and ownership*

- SHA-256: 0fbba6699287c2ac6fdcbeb28d4d6ccfa3d889b351b26f1e5010bd2ba74f8fef
- trackme-limited/trackme-report-issues#133 - bug - SmartStatus - regression introduced by version 2.0.34 causes SmartStatus function failure #133
- trackme-limited/trackme-report-issues#134 - bug - bad entities containing double quotes lead trackmesplkoutlierstrainhelper and trackmesamplingexecutor to continuously fail running searches for these entities with bad request #134

11.1.88 Version 2.0.34 - build 1684860645 (23/05/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

Hint

Roles Based Access Control enhancements:

- In this release, TrackMe implements a new strict **least privilege Role Bbased Access Control**
- A new role **trackme_power** is now builtin in TrackMe and designed to allow performing updates to entities of a granted tenant
- Access to TrackMe is driven by builtin capabilities provided by TrackMe builtin roles (trackme_user, trackme_power, trackme_admin)
- The least privilege approach implemented since this release allows granular access to TrackMe without requiring problematic capabilities which have security implications (list_settings, list_storage_passwords)
- TrackMe user interfaces automatically adapt its content and provided options depending on the profile of the current user, a normal user will for instance not see write or admin related actions
- TrackMe REST API endpoints are now classified in 3 groups, user level endpoints, write level endpoints and admin level endpoints
- The TrackMe `splunkremotesearch` also supports Roles Based Access Control, a user calling a given account must be a member of any of the listed roles in the account configuration to be granted access to this account
- For retro-compability purposes, TrackMe will allow access to an existing Remote account that has no RBAC roles setup yet to typical admin users in addition with TrackMe builtin roles (admin, sc_admin, trackme_user, trackme_power, trackme_admin)
- When TrackMe is **upgraded**, the migration of existing tenant is automatically performed by the **schema version management**, *Upgrading TrackMe*
- For more **information**, see: *Role Based Access Control and ownership*

- SHA-256: ce0d5a73b314c8dc246737149962dc5bd2038f89b313429f13485e3e99e2cd35
- trackme-limited/trackme-report-issues#106 - enhancement - Least privilege implementation - TrackMe implementation of a least privileges approach to provide with minimal capabilities requirement and a best practice security implementation #106
- trackme-limited/trackme-report-issues#119 - enhancement - All components - Performance optimisations #119
- trackme-limited/trackme-report-issues#120 - bug - Compliance Tracking (splk-cim) - UI affected by a previous change (regression from #116) #120
- trackme-limited/trackme-report-issues#121 - enhancement - UI behaviours - Call spinner in a more consistent manner when actions are being performed #121
- trackme-limited/trackme-report-issues#122 - bug - Flex Object (splk-flx) - Convention for status in the docs explanation is wrong #122
- trackme-limited/trackme-report-issues#101 - enhancement - Data Source/Host (splk-dsm/dhm) - Allows managing data in the future detection on a per entity basis #101
- trackme-limited/trackme-report-issues#124 - enhancement - major performance improvements for

trackmesplkoutlierssetrules #124

- trackme-limited/trackme-report-issues#125 - enhancement/bug - major performance improvements for Trackers execution (trackmepersistentfields) #125
- trackme-limited/trackme-report-issues#126 - enhancement - major performance enhancements for bulk edit operations in TrackMe #126
- trackme-limited/trackme-report-issues#127 - bug - Remove component does not remove some knowledge objects #127
- trackme-limited/trackme-report-issues#128 - enhancement - Workload - Allow the component to be added to / deleted from an existing Virtual Tenant #128
- trackme-limited/trackme-report-issues#129 - enhancement - splunkremotesearch - Roles Based Access Control support #129
- trackme-limited/trackme-report-issues#130 - enhancement - trackmeapiautodocs - Remove redundant resource_spl_example/resource_desc from endpoint usage output #130
- trackme-limited/trackme-report-issues#131 - bug - Data sampling & events format recognition - escaped double quotes are incorrectly escaped again leading the sampling generation to fail #131
- trackme-limited/trackme-report-issues#132 - bug - Data sampling & events format recognition - Reset loses the preset number of records, sets the number of records would fail if the entity has not been processed yet #132

11.1.89 Version 2.0.33 - build 1683898726 (12/05/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA-256: b9e8494d654bc60d1f0e12afe220d10c10f87aab1dd2fd20e517511040f9f9c8
- trackme-limited/trackme-report-issues#115 - bug - splk-dsm - tags - tags policies not applied as expected due a native multivalue format when taken into account by TrackMe's REST API #115
- trackme-limited/trackme-report-issues#116 - enhancements - Acknowledgments UI behaviours consistency #116
- trackme-limited/trackme-report-issues#117 - enhancement - Workload (splk-wlk) - The Orphan check and maintain search takes too long #117
- trackme-limited/trackme-report-issues#118 - bug - Data Host Monitoring (splk-dhm) - max delay and max latency are not honoured properly #118

11.1.90 Version 2.0.32 - build 1683797653 (11/05/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA-256: b570f9e6a668cfd895832cb2812e540e8a8e263606b49ae9014900d8e0683137
- bug - Workload (splk-wlk) - false positive issues with anomaly_reason=execution_delayed under some specific conditions #113
- bug - Workload (splk-wlk) - introspection metrics generation - introduce a bucket _time span=1m to properly aggregate metrics for pct_cpu/memory, sum the scan eventcount #114

11.1.91 Version 2.0.31 - build 1683730441 (10/05/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA-256: 32d31b6b3c8eade39c27af09dbe2e5d8497a7cecbc5b374f1ba939555ae59069
- bug - ucc-framework issue with urllib3 v2.0.x - latest version of urllib3 require fresher openssl version which builtin Splunk versions do not meet causing issues in alert actions #112

11.1.92 Version 2.0.30 - build 1683715542 (10/05/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA-256: 4652676182e6271bef61bc368db1fcdc3c216a26d022d4eb54dd6f28e8ec9168
- bug - all components - Tracking Alerts UI always created splk-dsm Alert #110
- bug - all components - SLA single should turn red if the entity has never been green since it was discovered #111

11.1.93 Version 2.0.29 - build 1683576225 (08/05/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA-256: 60e8e0665f3d924d3f7b636fc372fb8f1c6d4ca9274681913ea795706ac804cb
- bug - Workload (splk-wlk) - issues in Metadata collection when using a remote account with more than one member in the account definition #107
- bug - Flex Object - demo search for deployment servers should filter for the group when doing the inputlookup back #108
- bug - Workload (splk-wlk) - mltrain should be scheduled once per hour, mlmonitor should be scheduled every 20 minutes to prevent skipping searches #109

11.1.94 Version 2.0.28 - build 1682667017 (28/04/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA-256: 198ddc37df076de98e42a530bf66aa903eff8ae87c4c7d2e601b0c6316611c5d
- bug - splk-wlk (Workload) - If running in remote, introspection and Splunk Cloud SVC queries cannot rely on app fieldaliases #105

11.1.95 Version 2.0.27 - build 1682578920 (27/04/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA-256: b226ad96a069f070b5293bfe50fab101503e56c2bdf2c2d2027ed2d06bb8bf50
- bug - splk-wlk - Missing field alias for svc-consumer causes SVC consumption not to render expected SVC metrics #104

11.1.96 Version 2.0.26 - build 1682503730 (26/04/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA-256: fe68d95983066a1f8a2fcf2a4a60271ad1ce91d457c56f76f228a68418059baa
- feature - Introducing the new Splunk Workload component for TrackMe, to monitor your Splunk scheduling activity and take the control back #102
- bug - splk-cim - avoids append=t in the very first pipe which causes issues in Splunk Cloud #103

11.1.97 Version 2.0.25 - build 1682069909 (21/04/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

Note: Hybrid Trackers need to be re-created to benefit from the latest_eventcount_5m

- SHA-256: d992c12d1bb9998bc39be0171c3721d4c3f30ecef2ee0be1bfc1ab93dac29897
- bug/enhancement - latest_eventcount_5m from TrackMe metrics should perform an aggregation to properly represent the 5m sum of eventcounts #94

11.1.98 Version 2.0.23 - build 1681985039 (20/04/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA-256: e03e25136a8803cea926721d959a2312cdbcdec70f810279de3ffdf9c3cf5043
- bug - splk-feeds - Hybrid tracker creation, if breaking by host in splk-dsm, the dcount host leads to wrongly interpreting the host value, issues with burn test in raw mode #99
- bug - Outliers detection - incorrect message statement when upperBound is breached #100

11.1.99 Version 2.0.22 - build 1681860827 (19/04/2023)

Hint

Splunk 8.1.x and later, Linux, Python3 support only

- SHA-256: 08ae4facab3c6c141f0967998562bd1440fe1e1d6fe8ee8c85cef47a0191b81a
- bug - ack tracker regression issue introduced in release 2.0.21 #97
- bug - alerts creation - incorrect statement when including orange status for entities #95
- enhancement - splunkremotesearch - accepts a list of multiple Splunk REST endpoints and address targets randomly with HA and DR #93
- bug/enhancement - avoid disabling access to the acnknowledgement if it is still active althrough the entity is back in green state #96

11.1.100 Version 2.0.21 - build 1681766136 (17/04/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: 3b15dff23199adb46b8305cda8172062e25ddc24d3610e8da3a90345e4d08077
- bug - regression in trackmecollect for splk-dhm. the field splk_dhm_st_summary is required by the UI for processing #92

11.1.101 Version 2.0.20 - build 1681751403 (17/04/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: 59f122da1acc5728f8192365adf4a8b4f83bbd5e740d87f05d62678bdfaea020
- change - disable drilldown in API ref table #78
- change - Add skipping search shortcut access in Virtual Tenant (skipping donut screen) #79
- bug - mistmatch between custom command log files and associated props stanza #80
- bug/enhancement - improve detection of latency at ingest and its sensittivity using TrackMe metrics #81
- bug - trackmepersistentfields backend would raise an exception and block the remaining updates if an unexpected error occurs in the update process #82
- enhancement - avoids TrackMe custom command to be distributed amongst indexes while it's unecessary #83
- bug/enhancement - reduce the foot print of TrackMe state events stored in the summary indexes, prevents unecessary large fields (metrics summary, etc) #84
- enhancement - Preparation for the Implementation of least privileges approach in TrackMe and advanced capabilities management #85
- enhancements - Python backend enhancements #86
- enhancement - Add or Delete components for a TrackMe Virtual Tenant after it was created #87
- bug - "Show burn test search" creates a persistent macro #88
- bug/enhancenent - splk-feeds - Maintain delayed entities running out of the scope of TrackMe trackers #89
- enhancement - massive performance gains in events generating Python backends #90
- enhancement - trackmesplkoutlierstrainhelper should implement a max run time sec mechanism to avoid generating skipping search #91

11.1.102 Version 2.0.19 - build 1680519959 (03/04/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: 7f418e954415f4bdd74e8ce685eca7dab1b160ea6706dc6a0170b8fca65b571a
- bug - splk-dsm - data_first_time_seen should be part of persistent fields in the macro trackme_dsm_lookup_persistent_fields #75
- enhancement - trackmepersistentfields command - in some circumstances, there can be an unexpected duplication of entities, this enhancement ensures that this cannot happen #76

11.1.103 Version 2.0.18 - build 1680475914 (02/04/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: 71dd7ac5314ea3826c19a323844834bad95f3f98de317edb1ea05313761667e3
- bug/enhancement - TrackMe metrics generation and vizualisation issues when suffering from latency or low frequency entities #72
- bug - Virtual Tenant UI graphical issue when testing remote connectivity #73
- bug/enhancement - Improve latency detection by taking into account TrackMe metrics at Hybrid Tracker execution time #74
- enhancement - improve consistency of wording for lagging / latency / delay concepts #10

11.1.104 Version 2.0.17 - build 1680257518 (31/03/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: c4c68dc01cf1998db95566c15dc89228478848d969a583eaa617b142ac276547
- bug - splk-dsm/splk-flx status flipping will incorrectly continue to see new entities being discovered due to regression in 2.0.15 #71

11.1.105 Version 2.0.16 - build 1680138733 (30/03/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: e07a3f909033b93089541f27b1834ef327910f9f6c50ff11eade33b7e24fbb5c
- bug - splk-dsm - bad syntax in screen auto lagging def #68
- bug - splk-dsm/splk-dhm - avoid continuing to generate TrackMe metrics for an entity which data flow is interrupted, restrict the metrics scope to the 5 last minutes against the last event of the entity #69

- enhancement - Some high scale SHC environments with a large number of entities, especially in Splunk Cloud, were reported to encounter out of sync issues due to ML models update activity, this release reduce the frequency of the ML train activity to avoid this #70

Notes:

- Regarding fix #69, Hybrid Trackers need to be re-created, or manually updated:

trackme_dsm_hybrid_abstract_<id>

the break by change may change depending on your context, the fix relies on restricting the the spantime to avoid generating new metrics while the flow is interrupted

```
| eval spantime=_time | eventstats max(data_last_time_seen) as data_last_time_seen by␣
↪index,sourcetype | eval spantime=if(spantime>=(now()-300), spantime, null())
```

11.1.106 Version 2.0.15 - build 1679995508 (28/03/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: affba63ecf9fc7a8b718d5c45894dc64f920ec6d36f1e9794ca7d76f3ca54272
- bug / enhancements - introducing the custom command trackmepersistentfields to protect KVstore collection records from conflicting updates and replace the call to outputlookup Splunk command with more control #55
- bug - Vtenant creation endpoint should set the current schema_version immediately at the creation phase #56
- enhancement - Allow splunkremotesearch command to inherit earliest and latest from the environment (time range picker) #57
- bug/enhancement - avoid skipping searches for ML train/monitor and data sampling by reducing the default cron to every 20 when creating a new tenant #58
- enhancement - Limit the tenant name identifier to 15 characters max to avoid allowing users from reaching any Splunk limitations, reduce the random digits for trackers to 5 #59
- bug/enhancement - splk-dsm and splk-flx, at large scale with large number of concurrent Hybrid Trackers, concurrent loading of whole collections lead to impacts on other entities #60
- enhancement - Store the root constraint in a macro when creating the Hybrid Trackers for splk-feeds, for easier design, update and management #61
- bug - inherit trackmer_user role in trackme_admin to avoid any non explicit read access #62
- bug - If using Federated search in the instance running TrackMe, makeresults duplicates results unexpectly #63
- enhancement - splk-feeds Hybrid Tracker creation improvements, new builtin options to control performance denominators, review Burn test search before execution #64
- bug - Outliers management issues and enhancements #65
- change - Licensing management evolutions #66
- bug - log rotation is lacking for the various trackme logs #67

11.1.107 Version 2.0.14 - build 1679295918 (20/03/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: 5cc6306228293260ee82801bbf198a65ca13aedc6bf68bc0bda983b6ba6cae8c
- bug - conflict the same object exists already error when attempting to create a lagging class for the same conditions if one exists already for another category #45
- feature - splk-flx - Allow to control grouping of entities #46
- bug - splk-cim/splk-flx - metric ingestion issues when objects have space characters #47
- bug - negative value metrics will be ignored in splk-flx #48
- bug - indexes preset by default in tenant creation dropdown regression from 2.0.13 - showing first result index rather than preset index #49
- bug/enhancement - detect and degrade a Virtual Tenant using remote splunk account that was removed later on, or if all remote accounts were removed post configuration #50
- bug - Virtual Tenant UI - copy spl button may generate trackme SPL commands that cannot be parsed properly #52
- feature - Provide a burn test performance benchmark feature while creating Hybrid Trackers to investigate the run time performance ahead of the tracker creation #53

11.1.108 Version 2.0.13 - build 1678259747 (08/03/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: cc4d34f9f54e4fce2dd4299cc4bb549974ec7395a63b6eb4159ee46f2a7b02e5
- bug/enhancement - reduce volume of logs in trackme_splk_outliers_train_helper.log #41
- bug - lagging classes does not accept splk-dsm / splk-dhm pattern, failures to apply lagging classes against object!=all, various issues affecting lagging classes for splk-dhm #40
- bug - timezone issue in REST API and custom command logging events when the user running the command is in a non UTC timezone #43

11.1.109 Version 2.0.12 - build 1678171647 (07/03/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: 001d57ab9960024fde3eabf9439e1643ee118b99626b7f46e1d7ad3797c65378
- enhancement - avoids any enabled scheduled report by default including app level management utilities (Ack tracker, backup scheduler, maintenance mode tracker) #33
- bug - merged mode for splk-dsm not behaving as expected #34
- bug - Virtual Tenants UI regression when deleting the last tenant (should refresh and show up Welcome modal screen) #35

- enhancement - reduce the default earliest to -4h instead of -7d when creating Hybrid trackers to limit design requirements for first time users #36
- enhancement - improve consistency of wording for lagging / latency / delay concepts #10
- bug - missing perc95_latency_5m and stdev_latency_5m metrics for splk-dhm #38
- enhancement - Improve global TrackMe experience for splk-feeds with Overview based on TrackMe metrics primarly rather than direct Splunk query (Allows faster query and scalability, enhance RBAC consistency) #37

11.1.110 Version 2.0.11 - build 1677767350 (01/03/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: 16f797f4140bbff976c9d7ff7fb093f5ac519f1b699ff7010aa097e8474c4e8e
- bug - Entity remains in red state due to Data sampling detection altrhough the feature has been disabled #28

11.1.111 Version 2.0.10 - build 1677707255 (01/03/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: 423dc06178dd7360ccbffa3741dd7e41ae4ad63eb8cdb9bb703f86828729a3d2
- bug - custom indexes not properly used when creating Virtual Tenants from the user interfaces for splk-dsm/dhm/mhm #30
- bug - regression from 2.0.9 preventing access to RBAC update from the Virtual Tenant UI #31

11.1.112 Version 2.0.9 - build 1677588126 (28/02/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: edd8c6d22bc6fb80c9b7c08ee46b58d05ea2970f41678c89d6cfbf8f88f3d5d4
- bug: Virtual Tenants UI fails to load properly if a Virtual Tenant is disabled and was created with value for its description #21
- bug: Virtual Tenant creation error handling issues can lead to undetected failures within the Virtual Tenant user interface #22
- bug/enhancement: Virtual Tenants objects creation - avoid and enhance detection and re-attempt if splunkd API is not ready yet to server the newly created object #23
- bug/enhancement: disable auto-refresh in Virtual Tenants UI during long run operations to avoid loosing the spinner #24
- enhancement: splk-feeds - bulk edit management for Logical groups (splk-dsm/dhm/mhm) #25
- feature: introducing the concept of TrackMe schema versioning to allow future automated updates to the Virtual Tenants & Knowledge Objects schema #27

- feature: Sticky Acknowledgements #9
- bug/enhancement: Single forms and Donut drilldown do not lead to actions (all components) #16
- feature: license model update to allow an intermediate pricing plan with the Enterprise Edition #29

11.1.113 Version 2.0.8 - build 1677163367 (23/02/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: 80d0437c355c1ab71930bbf68f6ae0739817994c087712888f65d86d074678b2
- bug/enhancement: splk-dsm Data sampling - Tabulator occasionally loads before the modal screen, optimize and avoid multiple REST calls #11
- bug/enhancement - splk-flx - simplify the regular expression used in the deploymet server example #12
- bug - splk-flx - copy to clipboard button not working for deployment server example from first level modal screen #13
- Enhancement - improving naming convention consistancy in status and anomaly_reason #20
- Feature request - logical grouping to be made available for splk-dsm component #18
- bug - splk-dhm/splk-mhm entity view host Metadata filter do not apply when hybrid tracker was created manually in a tenant (opposed to created during the Virtual Tenant creation phase) #19

11.1.114 Version 2.0.7 - build 1676377640 (14/02/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: 13bc28f5693f9e6f7391ac2f61ddd598818d372c396d4f0d53bc6f5faf4fa865
- bug: splk-dsm - dictinct count host issue inconsistency when setting up a dcount_host treshold #1
- bug: splk-dsm - Elastic source syntax issue with from datamodel sources - error in identification of remote from searches #5
- feature: splk-dsm - Feature request - Simulation of thresholds before applying #3
- enhancement: Put a clear RBAC related message in when creating Virtual Tenants regarding membership explicit management
- enhancement: TrackMe Alert Suppression/Throttling Enhancements #6
- bug/enhancement: bug Tabulator loading modal - all components - In some circumstances, the screen can load before the REST endpoint call return the Tabulator data #7
- enhancement: Feature - Disable Ack when an entity goes back to green #8 - You can now enable the option “Remove Ack behaviour” in configuration if you wish to have Ack being disabled automatically when a previously non green entity comes back to green, rather than relying only on the Ack expiration - As well, there has been enhancements on the Ack tracker backend for better reporting and auditing of its activity (generate an audit event per entity)

Notes: - Hybrid/Elastic Trackers need to be re-created to benefit from the new distinct count hosts metrics for splk-dsm (Feeds tracking for Data Sources)

11.1.115 Version 2.0.6 - build 1675851310 (08/02/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: a5bf6e9580ca9924d20ea00c029a4cd61f6bffa700a493a2a8e251934d030bdb
- issue with splk-dhm timecharts in Splunk remote deployments when data gaps occur #9
- issue with splk-dhm compact mode which should show the sourcetype in addition with the index in the JSON summary #11
- wrong label in lagging classes applies to dropdown for splk-dsm/splk-dhm #12

11.1.116 Version 2.0.5 - build 1675711433 (06/02/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: ab77d89634b3debc5d2ddd881243310bbb18b959254efc53dcf6a83a873c5427
- Fix - Some REST endpoints are unexpectedly limiting their output to the first 100 records #7

11.1.117 Version 2.0.4 - build 1675617150 (05/02/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- Optimization - function dataset_update_cache should sleep before retrying in case of max concurrent searches run Optimization - function dataset_update_cache should sleep before retrying in case of max concurrent searches run #4
- Optimization - avoid logging check license return in non debug mode Optimization - avoid logging check license return in non debug mode #3
- Optimization - reduce internal logs from datagen custom command Optimization - reduce internal logs from datagen custom command #6

11.1.118 Version 2.0.3 - build 1675586140 (05/02/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: 661069bc7dfe803c9e6c10021cb693c85e616dce13b54c708f38ddc760848df4
- Data sampling engine - syntax error leads custom rule in simulation mode to fail rendering the expected results #1

11.1.119 Version 2.0.2 - build 1675379421 (02/02/2023)

Hint

Splunk 8.2.x/9.x and Python3 support only

- SHA-256: b5edf46f5bf6a293b318d33b0e4b07c982019dae427d4ad7b7b1b6881fb74145
- This the first official release for TrackMe V2

CHAPTER

TWELVE

VARIOUS

12.1 Third-party components credits

12.1.1 Tabulator

- Description: Easy to use, simple to code, fully featured, interactive JavaScript tables and data grids
- License Description: Licensed under the MIT License
- License Link: https://github.com/olifolkerd/tabulator/blob/master/LICENSE
- Component Version: 6.3.1

```
The MIT License (MIT)

Copyright (c) 2015-2024 Oli Folkerd

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

12.1.2 MomentJS

- Description: Parse, validate, manipulate, and display dates and times in JavaScript.
- License Description: Licensed under the MIT License
- License Link: https://github.com/moment/moment/blob/develop/LICENSE
- Component Version: 2.30.1

```
Copyright (c) JS Foundation and other contributors
```

(continues on next page)

(continued from previous page)

```
Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
```

12.1.3 Boostrap Icons

- Description: TrackMe uses the boostrap icons
- License Description: Licensed under the MIT License
- License Link: https://github.com/twbs/bootstrap/blob/v4.0.0/LICENSE
- Component Version: 1.11.0

```
The MIT License (MIT)

Copyright (c) 2011-2018 Twitter, Inc.
Copyright (c) 2011-2018 The Bootstrap Authors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
```

12.1.4 licensing

- Description: TrackMe uses Cryptolens API services for the licenses management, for the integration in TrackMe, it relies on the licensing Python librairies from Cryptolens.
- License Description: Licensed under the MIT License
- License Link: https://github.com/Cryptolens/cryptolens-python/blob/master/LICENSE.txt
- Component Version: 0.51

```
MIT License

Copyright (c) Cryptolens AB and Contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

12.1.5 certifi

- Description: Certifi provides Mozilla's carefully curated collection of Root Certificates for validating the trustworthiness of SSL certificates while verifying the identity of TLS hosts. It has been extracted from the Requests project.
- License Description: Mozilla Public License 2.0 (MPL 2.0)
- License Link: https://www.mozilla.org/en-US/MPL/2.0
- Component Version: 2024.8.30

```
Mozilla Public License
Version 2.0
1. Definitions
1.1. "Contributor"
means each individual or legal entity that creates, contributes to the creation of,␣
↪or owns Covered Software.

1.2. "Contributor Version"
means the combination of the Contributions of others (if any) used by a Contributor␣
↪and that particular Contributor's Contribution.

1.3. "Contribution"
means Covered Software of a particular Contributor.

1.4. "Covered Software"
means Source Code Form to which the initial Contributor has attached the notice in␣
```

(continues on next page)

(continued from previous page)

```
↪Exhibit A, the Executable Form of such Source Code Form, and Modifications of such␣
↪Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"
means

that the initial Contributor has attached the notice described in Exhibit B to the␣
↪Covered Software; or

that the Covered Software was made available under the terms of version 1.1 or␣
↪earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"
means any form of the work other than Source Code Form.

1.7. "Larger Work"
means a work that combines Covered Software with other material, in a separate file␣
↪or files, that is not Covered Software.

1.8. "License"
means this document.

1.9. "Licensable"
means having the right to grant, to the maximum extent possible, whether at the time␣
↪of the initial grant or subsequently, any and all of the rights conveyed by this␣
↪License.

1.10. "Modifications"
means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or␣
↪modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor
means any patent claim(s), including without limitation, method, process, and␣
↪apparatus claims, in any patent Licensable by such Contributor that would be␣
↪infringed, but for the grant of the License, by the making, using, selling,␣
↪offering for sale, having made, import, or transfer of either its Contributions or␣
↪its Contributor Version.

1.12. "Secondary License"
means either the GNU General Public License, Version 2.0, the GNU Lesser General␣
↪Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or␣
↪any later versions of those licenses.

1.13. "Source Code Form"
means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
means an individual or a legal entity exercising rights under this License. For legal␣
↪entities, "You" includes any entity that controls, is controlled by, or is under␣
↪common control with You. For purposes of this definition, "control" means (a) the␣
↪power, direct or indirect, to cause the direction or management of such entity,␣
↪whether by contract or otherwise, or (b) ownership of more than fifty percent (50%)␣
```

(continues on next page)

(continued from previous page)

```
↪of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions
2.1. Grants
Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by␣
↪such Contributor to use, reproduce, make available, modify, display, perform,␣
↪distribute, and otherwise exploit its Contributions, either on an unmodified basis,␣
↪with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made,
↪ import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date
The licenses granted in Section 2.1 with respect to any Contribution become effective␣
↪for each Contribution on the date the Contributor first distributes such␣
↪Contribution.

2.3. Limitations on Grant Scope
The licenses granted in this Section 2 are the only rights granted under this License.
↪ No additional rights or licenses will be implied from the distribution or␣
↪licensing of Covered Software under this License. Notwithstanding Section 2.1(b)␣
↪above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party's modifications of␣
↪Covered Software, or (ii) the combination of its Contributions with other software␣
↪(except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of␣
↪any Contributor (except as may be necessary to comply with the notice requirements␣
↪in Section 3.4).

2.4. Subsequent Licenses
No Contributor makes additional grants as a result of Your choice to distribute the␣
↪Covered Software under a subsequent version of this License (see Section 10.2) or␣
↪under the terms of a Secondary License (if permitted under the terms of Section 3.
↪3).

2.5. Representation
Each Contributor represents that the Contributor believes its Contributions are its␣
↪original creation(s) or it has sufficient rights to grant the rights to its␣
↪Contributions conveyed by this License.

2.6. Fair Use
This License is not intended to limit any rights You have under applicable copyright␣
↪doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions
Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities
```

(continues on next page)

(continued from previous page)

```
3.1. Distribution of Source Form
All distribution of Covered Software in Source Code Form, including any Modifications␣
↪that You create or to which You contribute, must be under the terms of this License.
↪ You must inform recipients that the Source Code Form of the Covered Software is␣
↪governed by the terms of this License, and how they can obtain a copy of this␣
↪License. You may not attempt to alter or restrict the recipients' rights in the␣
↪Source Code Form.

3.2. Distribution of Executable Form
If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described␣
↪in Section 3.1, and You must inform recipients of the Executable Form how they can␣
↪obtain a copy of such Source Code Form by reasonable means in a timely manner, at a␣
↪charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or␣
↪sublicense it under different terms, provided that the license for the Executable␣
↪Form does not attempt to limit or alter the recipients' rights in the Source Code␣
↪Form under this License.

3.3. Distribution of a Larger Work
You may create and distribute a Larger Work under terms of Your choice, provided that␣
↪You also comply with the requirements of this License for the Covered Software. If␣
↪the Larger Work is a combination of Covered Software with a work governed by one or␣
↪more Secondary Licenses, and the Covered Software is not Incompatible With␣
↪Secondary Licenses, this License permits You to additionally distribute such␣
↪Covered Software under the terms of such Secondary License(s), so that the␣
↪recipient of the Larger Work may, at their option, further distribute the Covered␣
↪Software under the terms of either this License or such Secondary License(s).

3.4. Notices
You may not remove or alter the substance of any license notices (including copyright␣
↪notices, patent notices, disclaimers of warranty, or limitations of liability)␣
↪contained within the Source Code Form of the Covered Software, except that You may␣
↪alter any license notices to the extent required to remedy known factual␣
↪inaccuracies.

3.5. Application of Additional Terms
You may choose to offer, and to charge a fee for, warranty, support, indemnity or␣
↪liability obligations to one or more recipients of Covered Software. However, You␣
↪may do so only on Your own behalf, and not on behalf of any Contributor. You must␣
↪make it absolutely clear that any such warranty, support, indemnity, or liability␣
↪obligation is offered by You alone, and You hereby agree to indemnify every␣
↪Contributor for any liability incurred by such Contributor as a result of warranty,␣
↪support, indemnity or liability terms You offer. You may include additional␣
↪disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation
If it is impossible for You to comply with any of the terms of this License with␣
↪respect to some or all of the Covered Software due to statute, judicial order, or␣
↪regulation then You must: (a) comply with the terms of this License to the maximum␣
↪extent possible; and (b) describe the limitations and the code they affect. Such␣
↪description must be placed in a text file included with all distributions of the␣
↪Covered Software under this License. Except to the extent prohibited by statute or␣
↪regulation, such description must be sufficiently detailed for a recipient of␣
```

(continues on next page)

(continued from previous page)

```
↪ordinary skill to be able to understand it.

5. Termination
5.1. The rights granted under this License will terminate automatically if You fail␣
↪to comply with any of its terms. However, if You become compliant, then the rights␣
↪granted under this License from a particular Contributor are reinstated (a)␣
↪provisionally, unless and until such Contributor explicitly and finally terminates␣
↪Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You␣
↪of the non-compliance by some reasonable means prior to 60 days after You have come␣
↪back into compliance. Moreover, Your grants from a particular Contributor are␣
↪reinstated on an ongoing basis if such Contributor notifies You of the non-
↪compliance by some reasonable means, this is the first time You have received␣
↪notice of non-compliance with this License from such Contributor, and You become␣
↪compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement␣
↪claim (excluding declaratory judgment actions, counter-claims, and cross-claims)␣
↪alleging that a Contributor Version directly or indirectly infringes any patent,␣
↪then the rights granted to You by any and all Contributors for the Covered Software␣
↪under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user␣
↪license agreements (excluding distributors and resellers) which have been validly␣
↪granted by You or Your distributors under this License prior to termination shall␣
↪survive termination.

6. Disclaimer of Warranty
Covered Software is provided under this License on an "as is" basis, without warranty␣
↪of any kind, either expressed, implied, or statutory, including, without limitation,
↪ warranties that the Covered Software is free of defects, merchantable, fit for a␣
↪particular purpose or non-infringing. The entire risk as to the quality and␣
↪performance of the Covered Software is with You. Should any Covered Software prove␣
↪defective in any respect, You (not any Contributor) assume the cost of any␣
↪necessary servicing, repair, or correction. This disclaimer of warranty constitutes␣
↪an essential part of this License. No use of any Covered Software is authorized␣
↪under this License except under this disclaimer.

7. Limitation of Liability
Under no circumstances and under no legal theory, whether tort (including negligence),
↪ contract, or otherwise, shall any Contributor, or anyone who distributes Covered␣
↪Software as permitted above, be liable to You for any direct, indirect, special,␣
↪incidental, or consequential damages of any character including, without limitation,
↪ damages for lost profits, loss of goodwill, work stoppage, computer failure or␣
↪malfunction, or any and all other commercial damages or losses, even if such party␣
↪shall have been informed of the possibility of such damages. This limitation of␣
↪liability shall not apply to liability for death or personal injury resulting from␣
↪such party's negligence to the extent applicable law prohibits such limitation.␣
↪Some jurisdictions do not allow the exclusion or limitation of incidental or␣
↪consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation
Any litigation relating to this License may be brought only in the courts of a␣
↪jurisdiction where the defendant maintains its principal place of business and such␣
↪litigation shall be governed by laws of that jurisdiction, without reference to its␣
↪conflict-of-law provisions. Nothing in this Section shall prevent a party's ability␣
↪to bring cross-claims or counter-claims.
```

(continues on next page)

(continued from previous page)

```
9. Miscellaneous
This License represents the complete agreement concerning the subject matter hereof.␣
↪If any provision of this License is held to be unenforceable, such provision shall␣
↪be reformed only to the extent necessary to make it enforceable. Any law or␣
↪regulation which provides that the language of a contract shall be construed␣
↪against the drafter shall not be used to construe this License against a␣
↪Contributor.

10. Versions of the License
10.1. New Versions
Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one␣
↪other than the license steward has the right to modify or publish new versions of␣
↪this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions
You may distribute the Covered Software under the terms of the version of the License␣
↪under which You originally received the Covered Software, or under the terms of any␣
↪subsequent version published by the license steward.

10.3. Modified Versions
If you create software not governed by this License, and you want to create a new␣
↪license for such software, you may create and use a modified version of this␣
↪License if you rename the license and remove any references to the name of the␣
↪license steward (except to note that such modified license differs from this␣
↪License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses
If You choose to distribute Source Code Form that is Incompatible With Secondary␣
↪Licenses under the terms of this version of the License, the notice described in␣
↪Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice
This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0.␣
↪If a copy of the MPL was not distributed with this file, You can obtain one at␣
↪https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You␣
↪may include the notice in a location (such as a LICENSE file in a relevant␣
↪directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice
This Source Code Form is "Incompatible With Secondary Licenses", as defined by the␣
↪Mozilla Public License, v. 2.0.
```

12.1.6 charset_normalizer

- Description: A library that helps you read text from an unknown charset encoding.
- License Description: Licensed under the MIT License
- License Link: https://github.com/Ousret/charset_normalizer/blob/master/LICENSE
- Component Version: 3.4.0

```
MIT License

Copyright (c) 2019 TAHRI Ahmed R.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

12.1.7 croniter

- Description: croniter provides iteration for datetime object with cron like format
- License Description: Licensed under the MIT License
- License Link: https://github.com/kiorky/croniter/blob/master/LICENSE
- Component Version: 6.0.0

```
Copyright (C) 2010-2012 Matsumoto Taichi

Permission is hereby granted, free of charge, to any person obtaining a copy of this␣
↪software and associated documentation files (the "Software"), to deal in the␣
↪Software without restriction, including without limitation the rights to use, copy,␣
↪modify, merge, publish, distribute, sublicense, and/or sell copies of the Software,␣
↪and to permit persons to whom the Software is furnished to do so, subject to the␣
↪following conditions:

The above copyright notice and this permission notice shall be included in all copies␣
↪or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,␣
↪INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A␣
↪PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT␣
↪HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION␣
↪OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE␣
↪SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

12.1.8 python-dateutil

- Description: Various utilities for working with date and datetime objects
- License Description: Apache 2.0
- License Link: https://github.com/dateutil/dateutil/blob/master/LICENSE
- Component Version: 2.9.0.post0

```
Copyright 2017- Paul Ganssle <paul@ganssle.io>
Copyright 2017- dateutil contributors (see AUTHORS file)

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
   limitations under the License.

The above license applies to all contributions after 2017-12-01, as well as
all contributions that have been re-licensed (see AUTHORS file for the list of
contributors who have re-licensed their code).
--------------------------------------------------------------------------------
dateutil - Extensions to the standard Python datetime module.

Copyright (c) 2003-2011 - Gustavo Niemeyer <gustavo@niemeyer.net>
Copyright (c) 2012-2014 - Tomi Pieviläinen <tomi.pievilainen@iki.fi>
Copyright (c) 2014-2016 - Yaron de Leeuw <me@jarondl.net>
Copyright (c) 2015-     - Paul Ganssle <paul@ganssle.io>
Copyright (c) 2015-     - dateutil contributors (see AUTHORS file)

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

   * Redistributions of source code must retain the above copyright notice,
      this list of conditions and the following disclaimer.
   * Redistributions in binary form must reproduce the above copyright notice,
      this list of conditions and the following disclaimer in the documentation
      and/or other materials provided with the distribution.
   * Neither the name of the copyright holder nor the names of its
      contributors may be used to endorse or promote products derived from
      this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The above BSD License Applies to all code, even that also covered by Apache 2.0.
```

12.1.9 defusedxml

- Description: XML bomb protection for Python stdlib modules
- License Description: Python Software Foundation License (PSFL)
- License Link: https://github.com/tiran/defusedxml/blob/main/LICENSE
- Component Version: 0.7.1

```
PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2
--------------------------------------------

1. This LICENSE AGREEMENT is between the Python Software Foundation
("PSF"), and the Individual or Organization ("Licensee") accessing and
otherwise using this software ("Python") in source or binary form and
its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF
hereby grants Licensee a nonexclusive, royalty-free, world-wide
license to reproduce, analyze, test, perform and/or display publicly,
prepare derivative works, distribute, and otherwise use Python
alone or in any derivative version, provided, however, that PSF's
License Agreement and PSF's notice of copyright, i.e., "Copyright (c)
2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 Python Software Foundation;
All Rights Reserved" are retained in Python alone or in any derivative
version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on
or incorporates Python or any part thereof, and wants to make
the derivative work available to others as provided herein, then
Licensee hereby agrees to include in any such work a brief summary of
the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS"
basis.  PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR
IMPLIED.  BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND
DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS
FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT
INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON
FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS
A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON,
OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material
breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any
relationship of agency, partnership, or joint venture between PSF and
Licensee.  This License Agreement does not grant permission to use PSF
trademarks or trade name in a trademark sense to endorse or promote
products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee
agrees to be bound by the terms and conditions of this License
Agreement.
```

12.1.10 idna

- Description: Internationalized Domain Names in Applications (IDNA)
- License Description: BSD 3-Clause 'New' or 'Revised' License
- License Link: https://github.com/kjd/idna/blob/master/LICENSE.md
- Component Version: 3.10

```
BSD 3-Clause License

Copyright (c) 2013-2022, Kim Davies and contributors.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its
   contributors may be used to endorse or promote products derived from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED
TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

12.1.11 requests

- Description: Python HTTP for Humans.
- License Description: Apache Software License (Apache 2.0)
- License Link: https://github.com/psf/requests/blob/main/LICENSE
- Component Version: 2.32.3

```
                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
```

(continues on next page)

(continued from previous page)

```
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.
```

(continues on next page)

(continued from previous page)

```
2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.
```

(continues on next page)

(continued from previous page)

```
   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions
   for use, reproduction, or distribution of Your modifications, or
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.
```

12.1.12 six

- Description: Python 2 and 3 compatibility utilities
- License Description: MIT License
- License Link: https://github.com/benjaminp/six/blob/master/LICENSE
- Component Version: 1.17.0

```
Copyright (c) 2010-2020 Benjamin Peterson

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

12.1.13 solnlib

- Description: The Splunk Software Development Kit for Splunk Solutions
- License Description: Apache License 2.0
- License Link: https://github.com/splunk/addonfactory-solutions-library-python/blob/main/LICENSE
- Component Version: 6.3.0

```
                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership of such entity.
```

(continues on next page)

(continued from previous page)

```
   "You" (or "Your") shall mean an individual or Legal Entity
   exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation
   source, and configuration files.

   "Object" form shall mean any form resulting from mechanical
   transformation or translation of a Source form, including but
   not limited to compiled object code, generated documentation,
   and conversions to other media types.

   "Work" shall mean the work of authorship, whether in Source or
   Object form, made available under the License, as indicated by a
   copyright notice that is included in or attached to the work
   (an example is provided in the Appendix below).

   "Derivative Works" shall mean any work, whether in Source or Object
   form, that is based on (or derived from) the Work and for which the
   editorial revisions, annotations, elaborations, or other modifications
   represent, as a whole, an original work of authorship. For the purposes
   of this License, Derivative Works shall not include works that remain
   separable from, or merely link (or bind by name) to the interfaces of,
   the Work and Derivative Works thereof.

   "Contribution" shall mean any work of authorship, including
   the original version of the Work and any modifications or additions
   to that Work or Derivative Works thereof, that is intentionally
   submitted to Licensor for inclusion in the Work by the copyright owner
   or by an individual or Legal Entity authorized to submit on behalf of
   the copyright owner. For the purposes of this definition, "submitted"
   means any form of electronic, verbal, or written communication sent
   to the Licensor or its representatives, including but not limited to
   communication on electronic mailing lists, source code control systems,
   and issue tracking systems that are managed by, or on behalf of, the
   Licensor for the purpose of discussing and improving the Work, but
   excluding communication that is conspicuously marked or otherwise
   designated in writing by the copyright owner as "Not a Contribution."

   "Contributor" shall mean Licensor and any individual or Legal Entity
   on behalf of whom a Contribution has been received by Licensor and
   subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
```

(continues on next page)

(continued from previous page)

```
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions
   for use, reproduction, or distribution of Your modifications, or
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
```

(continues on next page)

(continued from previous page)

```
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "{}"
   replaced with your own identifying information. (Don't include
   the brackets!)  The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright 2021 Splunk Inc.
```

(continues on next page)

(continued from previous page)

```
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

12.1.14 sortedcontainers

- Description: Sorted Containers – Sorted List, Sorted Dict, Sorted Set
- License Description: Apache Software License (Apache 2.0)
- License Link: http://www.apache.org/licenses/LICENSE-2.0
- Component Version: 2.4.0

```
                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity
   exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation
   source, and configuration files.

   "Object" form shall mean any form resulting from mechanical
   transformation or translation of a Source form, including but
   not limited to compiled object code, generated documentation,
   and conversions to other media types.
```

(continues on next page)

(continued from previous page)

```
   "Work" shall mean the work of authorship, whether in Source or
   Object form, made available under the License, as indicated by a
   copyright notice that is included in or attached to the work
   (an example is provided in the Appendix below).

   "Derivative Works" shall mean any work, whether in Source or Object
   form, that is based on (or derived from) the Work and for which the
   editorial revisions, annotations, elaborations, or other modifications
   represent, as a whole, an original work of authorship. For the purposes
   of this License, Derivative Works shall not include works that remain
   separable from, or merely link (or bind by name) to the interfaces of,
   the Work and Derivative Works thereof.

   "Contribution" shall mean any work of authorship, including
   the original version of the Work and any modifications or additions
   to that Work or Derivative Works thereof, that is intentionally
   submitted to Licensor for inclusion in the Work by the copyright owner
   or by an individual or Legal Entity authorized to submit on behalf of
   the copyright owner. For the purposes of this definition, "submitted"
   means any form of electronic, verbal, or written communication sent
   to the Licensor or its representatives, including but not limited to
   communication on electronic mailing lists, source code control systems,
   and issue tracking systems that are managed by, or on behalf of, the
   Licensor for the purpose of discussing and improving the Work, but
   excluding communication that is conspicuously marked or otherwise
   designated in writing by the copyright owner as "Not a Contribution."

   "Contributor" shall mean Licensor and any individual or Legal Entity
   on behalf of whom a Contribution has been received by Licensor and
   subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
```

(continues on next page)

(continued from previous page)

```
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions
   for use, reproduction, or distribution of Your modifications, or
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
```

(continues on next page)

(continued from previous page)

```
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!)  The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

12.1.15 sparklines

- Description: Generate sparklines for numbers using Unicode characters only.
- License Description: GNU General Public License (GPL)
- License Link: https://www.gnu.org/licenses/gpl-3.0.txt
- Component Version: 0.5.0


```
                    GNU GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <https://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The GNU General Public License is a free, copyleft license for
software and other kinds of works.

  The licenses for most software and other practical works are designed
to take away your freedom to share and change the works.  By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.  We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors.  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

  To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights.  Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received.  You must make sure that they, too, receive
or can get the source code.  And you must show them these terms so they
know their rights.

  Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

  For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software.  For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to
authors of previous versions.

```


(continues on next page)

(continued from previous page)

```
  Some devices are designed to deny users access to install or run
modified versions of the software inside them, although the manufacturer
can do so.  This is fundamentally incompatible with the aim of
protecting users' freedom to change the software.  The systematic
pattern of such abuse occurs in the area of products for individuals to
use, which is precisely where it is most unacceptable.  Therefore, we
have designed this version of the GPL to prohibit the practice for those
products.  If such problems arise substantially in other domains, we
stand ready to extend this provision to those domains in future versions
of the GPL, as needed to protect the freedom of users.

  Finally, every program is threatened constantly by software patents.
States should not allow patents to restrict development and use of
software on general-purpose computers, but in those that do, we wish to
avoid the special danger that patents applied to a free program could
make it effectively proprietary.  To prevent this, the GPL assures that
patents cannot be used to render the program non-free.

  The precise terms and conditions for copying, distribution and
modification follow.

                       TERMS AND CONDITIONS

  0. Definitions.

  "This License" refers to version 3 of the GNU General Public License.

  "Copyright" also means copyright-like laws that apply to other kinds of
works, such as semiconductor masks.

  "The Program" refers to any copyrightable work licensed under this
License.  Each licensee is addressed as "you".  "Licensees" and
"recipients" may be individuals or organizations.

  To "modify" a work means to copy from or adapt all or part of the work
in a fashion requiring copyright permission, other than the making of an
exact copy.  The resulting work is called a "modified version" of the
earlier work or a work "based on" the earlier work.

  A "covered work" means either the unmodified Program or a work based
on the Program.

  To "propagate" a work means to do anything with it that, without
permission, would make you directly or secondarily liable for
infringement under applicable copyright law, except executing it on a
computer or modifying a private copy.  Propagation includes copying,
distribution (with or without modification), making available to the
public, and in some countries other activities as well.

  To "convey" a work means any kind of propagation that enables other
parties to make or receive copies.  Mere interaction with a user through
a computer network, with no transfer of a copy, is not conveying.

  An interactive user interface displays "Appropriate Legal Notices"
to the extent that it includes a convenient and prominently visible
feature that (1) displays an appropriate copyright notice, and (2)
```

(continues on next page)

(continued from previous page)

```
tells the user that there is no warranty for the work (except to the
extent that warranties are provided), that licensees may convey the
work under this License, and how to view a copy of this License.  If
the interface presents a list of user commands or options, such as a
menu, a prominent item in the list meets this criterion.

  1. Source Code.

  The "source code" for a work means the preferred form of the work
for making modifications to it.  "Object code" means any non-source
form of a work.

  A "Standard Interface" means an interface that either is an official
standard defined by a recognized standards body, or, in the case of
interfaces specified for a particular programming language, one that
is widely used among developers working in that language.

  The "System Libraries" of an executable work include anything, other
than the work as a whole, that (a) is included in the normal form of
packaging a Major Component, but which is not part of that Major
Component, and (b) serves only to enable use of the work with that
Major Component, or to implement a Standard Interface for which an
implementation is available to the public in source code form.  A
"Major Component", in this context, means a major essential component
(kernel, window system, and so on) of the specific operating system
(if any) on which the executable work runs, or a compiler used to
produce the work, or an object code interpreter used to run it.

  The "Corresponding Source" for a work in object code form means all
the source code needed to generate, install, and (for an executable
work) run the object code and to modify the work, including scripts to
control those activities.  However, it does not include the work's
System Libraries, or general-purpose tools or generally available free
programs which are used unmodified in performing those activities but
which are not part of the work.  For example, Corresponding Source
includes interface definition files associated with source files for
the work, and the source code for shared libraries and dynamically
linked subprograms that the work is specifically designed to require,
such as by intimate data communication or control flow between those
subprograms and other parts of the work.

  The Corresponding Source need not include anything that users
can regenerate automatically from other parts of the Corresponding
Source.

  The Corresponding Source for a work in source code form is that
same work.

  2. Basic Permissions.

  All rights granted under this License are granted for the term of
copyright on the Program, and are irrevocable provided the stated
conditions are met.  This License explicitly affirms your unlimited
permission to run the unmodified Program.  The output from running a
covered work is covered by this License only if the output, given its
content, constitutes a covered work.  This License acknowledges your
```

(continues on next page)

(continued from previous page)

```
rights of fair use or other equivalent, as provided by copyright law.

  You may make, run and propagate covered works that you do not
convey, without conditions so long as your license otherwise remains
in force.  You may convey covered works to others for the sole purpose
of having them make modifications exclusively for you, or provide you
with facilities for running those works, provided that you comply with
the terms of this License in conveying all material for which you do
not control copyright.  Those thus making or running the covered works
for you must do so exclusively on your behalf, under your direction
and control, on terms that prohibit them from making any copies of
your copyrighted material outside their relationship with you.

  Conveying under any other circumstances is permitted solely under
the conditions stated below.  Sublicensing is not allowed; section 10
makes it unnecessary.

  3. Protecting Users' Legal Rights From Anti-Circumvention Law.

  No covered work shall be deemed part of an effective technological
measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

  When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

  4. Conveying Verbatim Copies.

  You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

  You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

  5. Conveying Modified Source Versions.

  You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

    a) The work must carry prominent notices stating that you modified
    it, and giving a relevant date.

    b) The work must carry prominent notices stating that it is
```

(continues on next page)

(continued from previous page)

```
    released under this License and any conditions added under section
    7.  This requirement modifies the requirement in section 4 to
    "keep intact all notices".

    c) You must license the entire work, as a whole, under this
    License to anyone who comes into possession of a copy.  This
    License will therefore apply, along with any applicable section 7
    additional terms, to the whole of the work, and all its parts,
    regardless of how they are packaged.  This License gives no
    permission to license the work in any other way, but it does not
    invalidate such permission if you have separately received it.

    d) If the work has interactive user interfaces, each must display
    Appropriate Legal Notices; however, if the Program has interactive
    interfaces that do not display Appropriate Legal Notices, your
    work need not make them do so.

  A compilation of a covered work with other separate and independent
works, which are not by their nature extensions of the covered work,
and which are not combined with it such as to form a larger program,
in or on a volume of a storage or distribution medium, is called an
"aggregate" if the compilation and its resulting copyright are not
used to limit the access or legal rights of the compilation's users
beyond what the individual works permit.  Inclusion of a covered work
in an aggregate does not cause this License to apply to the other
parts of the aggregate.

  6. Conveying Non-Source Forms.

  You may convey a covered work in object code form under the terms
of sections 4 and 5, provided that you also convey the
machine-readable Corresponding Source under the terms of this License,
in one of these ways:

    a) Convey the object code in, or embodied in, a physical product
    (including a physical distribution medium), accompanied by the
    Corresponding Source fixed on a durable physical medium
    customarily used for software interchange.

    b) Convey the object code in, or embodied in, a physical product
    (including a physical distribution medium), accompanied by a
    written offer, valid for at least three years and valid for as
    long as you offer spare parts or customer support for that product
    model, to give anyone who possesses the object code either (1) a
    copy of the Corresponding Source for all the software in the
    product that is covered by this License, on a durable physical
    medium customarily used for software interchange, for a price no
    more than your reasonable cost of physically performing this
    conveying of source, or (2) access to copy the
    Corresponding Source from a network server at no charge.

    c) Convey individual copies of the object code with a copy of the
    written offer to provide the Corresponding Source.  This
    alternative is allowed only occasionally and noncommercially, and
    only if you received the object code with such an offer, in accord
    with subsection 6b.
```

(continues on next page)

(continued from previous page)

```
    d) Convey the object code by offering access from a designated
    place (gratis or for a charge), and offer equivalent access to the
    Corresponding Source in the same way through the same place at no
    further charge.  You need not require recipients to copy the
    Corresponding Source along with the object code.  If the place to
    copy the object code is a network server, the Corresponding Source
    may be on a different server (operated by you or a third party)
    that supports equivalent copying facilities, provided you maintain
    clear directions next to the object code saying where to find the
    Corresponding Source.  Regardless of what server hosts the
    Corresponding Source, you remain obligated to ensure that it is
    available for as long as needed to satisfy these requirements.

    e) Convey the object code using peer-to-peer transmission, provided
    you inform other peers where the object code and Corresponding
    Source of the work are being offered to the general public at no
    charge under subsection 6d.

  A separable portion of the object code, whose source code is excluded
from the Corresponding Source as a System Library, need not be
included in conveying the object code work.

  A "User Product" is either (1) a "consumer product", which means any
tangible personal property which is normally used for personal, family,
or household purposes, or (2) anything designed or sold for incorporation
into a dwelling.  In determining whether a product is a consumer product,
doubtful cases shall be resolved in favor of coverage.  For a particular
product received by a particular user, "normally used" refers to a
typical or common use of that class of product, regardless of the status
of the particular user or of the way in which the particular user
actually uses, or expects or is expected to use, the product.  A product
is a consumer product regardless of whether the product has substantial
commercial, industrial or non-consumer uses, unless such uses represent
the only significant mode of use of the product.

  "Installation Information" for a User Product means any methods,
procedures, authorization keys, or other information required to install
and execute modified versions of a covered work in that User Product from
a modified version of its Corresponding Source.  The information must
suffice to ensure that the continued functioning of the modified object
code is in no case prevented or interfered with solely because
modification has been made.

  If you convey an object code work under this section in, or with, or
specifically for use in, a User Product, and the conveying occurs as
part of a transaction in which the right of possession and use of the
User Product is transferred to the recipient in perpetuity or for a
fixed term (regardless of how the transaction is characterized), the
Corresponding Source conveyed under this section must be accompanied
by the Installation Information.  But this requirement does not apply
if neither you nor any third party retains the ability to install
modified object code on the User Product (for example, the work has
been installed in ROM).

  The requirement to provide Installation Information does not include a
```

(continues on next page)

(continued from previous page)

```
requirement to continue to provide support service, warranty, or updates
for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed.  Access to a
network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and
protocols for communication across the network.

  Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
unpacking, reading or copying.

  7. Additional Terms.

  "Additional permissions" are terms that supplement the terms of this
License by making exceptions from one or more of its conditions.
Additional permissions that are applicable to the entire Program shall
be treated as though they were included in this License, to the extent
that they are valid under applicable law.  If additional permissions
apply only to part of the Program, that part may be used separately
under those permissions, but the entire Program remains governed by
this License without regard to the additional permissions.

  When you convey a copy of a covered work, you may at your option
remove any additional permissions from that copy, or from any part of
it.  (Additional permissions may be written to require their own
removal in certain cases when you modify the work.)  You may place
additional permissions on material, added by you to a covered work,
for which you have or can give appropriate copyright permission.

  Notwithstanding any other provision of this License, for material you
add to a covered work, you may (if authorized by the copyright holders of
that material) supplement the terms of this License with terms:

    a) Disclaiming warranty or limiting liability differently from the
    terms of sections 15 and 16 of this License; or

    b) Requiring preservation of specified reasonable legal notices or
    author attributions in that material or in the Appropriate Legal
    Notices displayed by works containing it; or

    c) Prohibiting misrepresentation of the origin of that material, or
    requiring that modified versions of such material be marked in
    reasonable ways as different from the original version; or

    d) Limiting the use for publicity purposes of names of licensors or
    authors of the material; or

    e) Declining to grant rights under trademark law for use of some
    trade names, trademarks, or service marks; or

    f) Requiring indemnification of licensors and authors of that
    material by anyone who conveys the material (or modified versions of
    it) with contractual assumptions of liability to the recipient, for
    any liability that these contractual assumptions directly impose on
```

(continues on next page)

(continued from previous page)

```
    those licensors and authors.

  All other non-permissive additional terms are considered "further
restrictions" within the meaning of section 10.  If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term.  If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.

  If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating
where to find the applicable terms.

  Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above requirements apply either way.

  8. Termination.

  You may not propagate or modify a covered work except as expressly
provided under this License.  Any attempt otherwise to propagate or
modify it is void, and will automatically terminate your rights under
this License (including any patent licenses granted under the third
paragraph of section 11).

  However, if you cease all violation of this License, then your
license from a particular copyright holder is reinstated (a)
provisionally, unless and until the copyright holder explicitly and
finally terminates your license, and (b) permanently, if the copyright
holder fails to notify you of the violation by some reasonable means
prior to 60 days after the cessation.

  Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.

  Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License.  If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same
material under section 10.

  9. Acceptance Not Required for Having Copies.

  You are not required to accept this License in order to receive or
run a copy of the Program.  Ancillary propagation of a covered work
occurring solely as a consequence of using peer-to-peer transmission
to receive a copy likewise does not require acceptance.  However,
nothing other than this License grants you permission to propagate or
```

(continues on next page)

(continued from previous page)

```
modify any covered work.  These actions infringe copyright if you do
not accept this License.  Therefore, by modifying or propagating a
covered work, you indicate your acceptance of this License to do so.

  10. Automatic Licensing of Downstream Recipients.

  Each time you convey a covered work, the recipient automatically
receives a license from the original licensors, to run, modify and
propagate that work, subject to this License.  You are not responsible
for enforcing compliance by third parties with this License.

  An "entity transaction" is a transaction transferring control of an
organization, or substantially all assets of one, or subdividing an
organization, or merging organizations.  If propagation of a covered
work results from an entity transaction, each party to that
transaction who receives a copy of the work also receives whatever
licenses to the work the party's predecessor in interest had or could
give under the previous paragraph, plus a right to possession of the
Corresponding Source of the work from the predecessor in interest, if
the predecessor has it or can get it with reasonable efforts.

  You may not impose any further restrictions on the exercise of the
rights granted or affirmed under this License.  For example, you may
not impose a license fee, royalty, or other charge for exercise of
rights granted under this License, and you may not initiate litigation
(including a cross-claim or counterclaim in a lawsuit) alleging that
any patent claim is infringed by making, using, selling, offering for
sale, or importing the Program or any portion of it.

  11. Patents.

  A "contributor" is a copyright holder who authorizes use under this
License of the Program or a work on which the Program is based.  The
work thus licensed is called the contributor's "contributor version".

  A contributor's "essential patent claims" are all patent claims
owned or controlled by the contributor, whether already acquired or
hereafter acquired, that would be infringed by some manner, permitted
by this License, of making, using, or selling its contributor version,
but do not include claims that would be infringed only as a
consequence of further modification of the contributor version.  For
purposes of this definition, "control" includes the right to grant
patent sublicenses in a manner consistent with the requirements of
this License.

  Each contributor grants you a non-exclusive, worldwide, royalty-free
patent license under the contributor's essential patent claims, to
make, use, sell, offer for sale, import and otherwise run, modify and
propagate the contents of its contributor version.

  In the following three paragraphs, a "patent license" is any express
agreement or commitment, however denominated, not to enforce a patent
(such as an express permission to practice a patent or covenant not to
sue for patent infringement).  To "grant" such a patent license to a
party means to make such an agreement or commitment not to enforce a
patent against the party.
```

(continues on next page)

(continued from previous page)

```
  If you convey a covered work, knowingly relying on a patent license,
and the Corresponding Source of the work is not available for anyone
to copy, free of charge and under the terms of this License, through a
publicly available network server or other readily accessible means,
then you must either (1) cause the Corresponding Source to be so
available, or (2) arrange to deprive yourself of the benefit of the
patent license for this particular work, or (3) arrange, in a manner
consistent with the requirements of this License, to extend the patent
license to downstream recipients.  "Knowingly relying" means you have
actual knowledge that, but for the patent license, your conveying the
covered work in a country, or your recipient's use of the covered work
in a country, would infringe one or more identifiable patents in that
country that you have reason to believe are valid.

  If, pursuant to or in connection with a single transaction or
arrangement, you convey, or propagate by procuring conveyance of, a
covered work, and grant a patent license to some of the parties
receiving the covered work authorizing them to use, propagate, modify
or convey a specific copy of the covered work, then the patent license
you grant is automatically extended to all recipients of the covered
work and works based on it.

  A patent license is "discriminatory" if it does not include within
the scope of its coverage, prohibits the exercise of, or is
conditioned on the non-exercise of one or more of the rights that are
specifically granted under this License.  You may not convey a covered
work if you are a party to an arrangement with a third party that is
in the business of distributing software, under which you make payment
to the third party based on the extent of your activity of conveying
the work, and under which the third party grants, to any of the
parties who would receive the covered work from you, a discriminatory
patent license (a) in connection with copies of the covered work
conveyed by you (or copies made from those copies), or (b) primarily
for and in connection with specific products or compilations that
contain the covered work, unless you entered into that arrangement,
or that patent license was granted, prior to 28 March 2007.

  Nothing in this License shall be construed as excluding or limiting
any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.

  12. No Surrender of Others' Freedom.

  If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License.  If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may
not convey it at all.  For example, if you agree to terms that obligate you
to collect a royalty for further conveying from those to whom you convey
the Program, the only way you could satisfy both those terms and this
License would be to refrain entirely from conveying the Program.

  13. Use with the GNU Affero General Public License.
```

(continues on next page)

(continued from previous page)

```
  Notwithstanding any other provision of this License, you have
permission to link or combine any covered work with a work licensed
under version 3 of the GNU Affero General Public License into a single
combined work, and to convey the resulting work.  The terms of this
License will continue to apply to the part which is the covered work,
but the special requirements of the GNU Affero General Public License,
section 13, concerning interaction through a network will apply to the
combination as such.

  14. Revised Versions of this License.

  The Free Software Foundation may publish revised and/or new versions of
the GNU General Public License from time to time.  Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

  Each version is given a distinguishing version number.  If the
Program specifies that a certain numbered version of the GNU General
Public License "or any later version" applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation.  If the Program does not specify a version number of the
GNU General Public License, you may choose any version ever published
by the Free Software Foundation.

  If the Program specifies that a proxy can decide which future
versions of the GNU General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

  Later license versions may give you additional or different
permissions.  However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

  15. Disclaimer of Warranty.

  THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW.  EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU.  SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. Limitation of Liability.

  IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
```

(continues on next page)

(continued from previous page)

```
SUCH DAMAGES.

  17. Interpretation of Sections 15 and 16.

  If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program.  It is safest
to attach them to the start of each source file to most effectively
state the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <https://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

  If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

  You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
```

(continues on next page)

(continued from previous page)

```
For more information on this, and how to apply and follow the GNU GPL, see
<https://www.gnu.org/licenses/>.

  The GNU General Public License does not permit incorporating your program
into proprietary programs.  If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.  But first, please read
<https://www.gnu.org/licenses/why-not-lgpl.html>.
```

12.1.16 splunklib

- Description: The Splunk Enterprise Software Development Kit for Python
- License Description: Apache License 2.0
- License Link: https://github.com/splunk/splunk-sdk-python/blob/master/LICENSE
- Component Version: 2.1.0

```
                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity
   exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation
   source, and configuration files.

   "Object" form shall mean any form resulting from mechanical
   transformation or translation of a Source form, including but
   not limited to compiled object code, generated documentation,
   and conversions to other media types.

   "Work" shall mean the work of authorship, whether in Source or
   Object form, made available under the License, as indicated by a
   copyright notice that is included in or attached to the work
```

(continues on next page)

(continued from previous page)

```
   (an example is provided in the Appendix below).

   "Derivative Works" shall mean any work, whether in Source or Object
   form, that is based on (or derived from) the Work and for which the
   editorial revisions, annotations, elaborations, or other modifications
   represent, as a whole, an original work of authorship. For the purposes
   of this License, Derivative Works shall not include works that remain
   separable from, or merely link (or bind by name) to the interfaces of,
   the Work and Derivative Works thereof.

   "Contribution" shall mean any work of authorship, including
   the original version of the Work and any modifications or additions
   to that Work or Derivative Works thereof, that is intentionally
   submitted to Licensor for inclusion in the Work by the copyright owner
   or by an individual or Legal Entity authorized to submit on behalf of
   the copyright owner. For the purposes of this definition, "submitted"
   means any form of electronic, verbal, or written communication sent
   to the Licensor or its representatives, including but not limited to
   communication on electronic mailing lists, source code control systems,
   and issue tracking systems that are managed by, or on behalf of, the
   Licensor for the purpose of discussing and improving the Work, but
   excluding communication that is conspicuously marked or otherwise
   designated in writing by the copyright owner as "Not a Contribution."

   "Contributor" shall mean Licensor and any individual or Legal Entity
   on behalf of whom a Contribution has been received by Licensor and
   subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:
```

(continues on next page)

(continued from previous page)

```
   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions
   for use, reproduction, or distribution of Your modifications, or
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
```

(continues on next page)

(continued from previous page)

```
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!)  The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

12.1.17 splunktalib

- Description: Supporting library for Splunk Add-ons
- License Description: Apache Software License (Apache-2.0)
- License Link: https://github.com/splunk/addonfactory-ta-library-python/blob/main/LICENSE
- Component Version: 3.0.5

```
                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity
   exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation
   source, and configuration files.

   "Object" form shall mean any form resulting from mechanical
   transformation or translation of a Source form, including but
   not limited to compiled object code, generated documentation,
   and conversions to other media types.

   "Work" shall mean the work of authorship, whether in Source or
   Object form, made available under the License, as indicated by a
   copyright notice that is included in or attached to the work
   (an example is provided in the Appendix below).

   "Derivative Works" shall mean any work, whether in Source or Object
   form, that is based on (or derived from) the Work and for which the
   editorial revisions, annotations, elaborations, or other modifications
   represent, as a whole, an original work of authorship. For the purposes
   of this License, Derivative Works shall not include works that remain
   separable from, or merely link (or bind by name) to the interfaces of,
   the Work and Derivative Works thereof.

   "Contribution" shall mean any work of authorship, including
   the original version of the Work and any modifications or additions
```

(continues on next page)

(continued from previous page)

```
   to that Work or Derivative Works thereof, that is intentionally
   submitted to Licensor for inclusion in the Work by the copyright owner
   or by an individual or Legal Entity authorized to submit on behalf of
   the copyright owner. For the purposes of this definition, "submitted"
   means any form of electronic, verbal, or written communication sent
   to the Licensor or its representatives, including but not limited to
   communication on electronic mailing lists, source code control systems,
   and issue tracking systems that are managed by, or on behalf of, the
   Licensor for the purpose of discussing and improving the Work, but
   excluding communication that is conspicuously marked or otherwise
   designated in writing by the copyright owner as "Not a Contribution."

   "Contributor" shall mean Licensor and any individual or Legal Entity
   on behalf of whom a Contribution has been received by Licensor and
   subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and
```

(continues on next page)

(continued from previous page)

```
   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions
   for use, reproduction, or distribution of Your modifications, or
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
```

(continues on next page)

(continued from previous page)

```
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!)  The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright 2021 Splunk Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

12.1.18 splunktaucclib

- Description: Splunk Add-on UCC factory library
- License Description: Apache 2.0
- License Link: https://github.com/splunk/addonfactory-ucc-library/blob/main/LICENSE
- Component Version: 8.0.0

```
                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
```

(continues on next page)

(continued from previous page)

```
1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity
   exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation
   source, and configuration files.

   "Object" form shall mean any form resulting from mechanical
   transformation or translation of a Source form, including but
   not limited to compiled object code, generated documentation,
   and conversions to other media types.

   "Work" shall mean the work of authorship, whether in Source or
   Object form, made available under the License, as indicated by a
   copyright notice that is included in or attached to the work
   (an example is provided in the Appendix below).

   "Derivative Works" shall mean any work, whether in Source or Object
   form, that is based on (or derived from) the Work and for which the
   editorial revisions, annotations, elaborations, or other modifications
   represent, as a whole, an original work of authorship. For the purposes
   of this License, Derivative Works shall not include works that remain
   separable from, or merely link (or bind by name) to the interfaces of,
   the Work and Derivative Works thereof.

   "Contribution" shall mean any work of authorship, including
   the original version of the Work and any modifications or additions
   to that Work or Derivative Works thereof, that is intentionally
   submitted to Licensor for inclusion in the Work by the copyright owner
   or by an individual or Legal Entity authorized to submit on behalf of
   the copyright owner. For the purposes of this definition, "submitted"
   means any form of electronic, verbal, or written communication sent
   to the Licensor or its representatives, including but not limited to
   communication on electronic mailing lists, source code control systems,
   and issue tracking systems that are managed by, or on behalf of, the
   Licensor for the purpose of discussing and improving the Work, but
   excluding communication that is conspicuously marked or otherwise
   designated in writing by the copyright owner as "Not a Contribution."

   "Contributor" shall mean Licensor and any individual or Legal Entity
```

(continues on next page)

(continued from previous page)

```
   on behalf of whom a Contribution has been received by Licensor and
   subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
```

(continues on next page)

(continued from previous page)

```
      or as an addendum to the NOTICE text from the Work, provided
      that such additional attribution notices cannot be construed
      as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions
   for use, reproduction, or distribution of Your modifications, or
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.
```

(continues on next page)

(continued from previous page)

```
END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!)  The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright 2021 Splunk Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

12.1.19 urllib3

- Description: Python urllib3
- License Description: Licensed under the MIT License
- License Link: https://github.com/urllib3/urllib3/blob/main/LICENSE.txt
- Component Version: 1.26.20

```
MIT License

Copyright (c) 2008-2020 Andrey Petrov and contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
```

(continues on next page)

(continued from previous page)

```
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

12.1.20 pytz

- Description: World timezone definitions, modern and historical
- License Description: Licensed under the MIT License
- License Link: https://pythonhosted.org/pytz/#license
- Component Version: 2025.2

```
MIT License

Copyright (c) 2008-2020 Andrey Petrov and contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

12.1.21 packaging

- Description: Core utilities for Python packages
- License Description: Apache 2.0 / BSD
- License Link: https://github.com/pypa/packaging/blob/main/LICENSE.APACHE - https://github.com/pypa/packaging/blob/main/LICENSE.BSD
- Component Version: 25.0

```
                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.
```

(continues on next page)

(continued from previous page)

```
   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity
   exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation
   source, and configuration files.

   "Object" form shall mean any form resulting from mechanical
   transformation or translation of a Source form, including but
   not limited to compiled object code, generated documentation,
   and conversions to other media types.

   "Work" shall mean the work of authorship, whether in Source or
   Object form, made available under the License, as indicated by a
   copyright notice that is included in or attached to the work
   (an example is provided in the Appendix below).

   "Derivative Works" shall mean any work, whether in Source or Object
   form, that is based on (or derived from) the Work and for which the
   editorial revisions, annotations, elaborations, or other modifications
   represent, as a whole, an original work of authorship. For the purposes
   of this License, Derivative Works shall not include works that remain
   separable from, or merely link (or bind by name) to the interfaces of,
   the Work and Derivative Works thereof.

   "Contribution" shall mean any work of authorship, including
   the original version of the Work and any modifications or additions
   to that Work or Derivative Works thereof, that is intentionally
   submitted to Licensor for inclusion in the Work by the copyright owner
   or by an individual or Legal Entity authorized to submit on behalf of
   the copyright owner. For the purposes of this definition, "submitted"
   means any form of electronic, verbal, or written communication sent
   to the Licensor or its representatives, including but not limited to
   communication on electronic mailing lists, source code control systems,
   and issue tracking systems that are managed by, or on behalf of, the
   Licensor for the purpose of discussing and improving the Work, but
   excluding communication that is conspicuously marked or otherwise
   designated in writing by the copyright owner as "Not a Contribution."

   "Contributor" shall mean Licensor and any individual or Legal Entity
   on behalf of whom a Contribution has been received by Licensor and
   subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
```

(continues on next page)

(continued from previous page)

```
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
      Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
      stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
      that You distribute, all copyright, patent, trademark, and
      attribution notices from the Source form of the Work,
      excluding those notices that do not pertain to any part of
      the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
      distribution, then any Derivative Works that You distribute must
      include a readable copy of the attribution notices contained
      within such NOTICE file, excluding those notices that do not
      pertain to any part of the Derivative Works, in at least one
      of the following places: within a NOTICE text file distributed
      as part of the Derivative Works; within the Source form or
      documentation, if provided along with the Derivative Works; or,
      within a display generated by the Derivative Works, if and
      wherever such third-party notices normally appear. The contents
      of the NOTICE file are for informational purposes only and
      do not modify the License. You may add Your own attribution
      notices within Derivative Works that You distribute, alongside
      or as an addendum to the NOTICE text from the Work, provided
      that such additional attribution notices cannot be construed
      as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions
   for use, reproduction, or distribution of Your modifications, or
```

(continues on next page)

(continued from previous page)

```
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
```

```
Copyright (c) Donald Stufft and individual contributors.
All rights reserved.

Redistribution and use in source and binary forms, with or without
```

(continues on next page)

(continued from previous page)

```
modification, are permitted provided that the following conditions are met:

   1. Redistributions of source code must retain the above copyright notice,
      this list of conditions and the following disclaimer.

   2. Redistributions in binary form must reproduce the above copyright
      notice, this list of conditions and the following disclaimer in the
      documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

12.1.22 prism

- Description: Prism is a lightweight, robust, and elegant syntax highlighting library
- License Description: MIT
- License Link: https://raw.githubusercontent.com/PrismJS/prism/refs/heads/v2/LICENSE
- Component Version: 1.30.0

```
MIT LICENSE

Copyright (c) 2012 Lea Verou

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
```

12.1.23 pygal

- Description: pygal is a dynamic SVG charting library written in python
- License: GNU Lesser General Public License v3 or later (LGPLv3+) (GNU LGPL v3+)
- License Link: https://www.gnu.org/licenses/lgpl-3.0.en.html#license-text

- Component Version: 3.0.5

```
GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <https://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document,
↪ but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and␣
↪conditions of version 3 of the GNU General Public License, supplemented by the␣
↪additional permissions listed below.

0. Additional Definitions.
As used herein, "this License" refers to version 3 of the GNU Lesser General Public␣
↪License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an␣
↪Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library,␣
↪but which is not otherwise based on the Library. Defining a subclass of a class␣
↪defined by the Library is deemed a mode of using an interface provided by the␣
↪Library.

A "Combined Work" is a work produced by combining or linking an Application with the␣
↪Library. The particular version of the Library with which the Combined Work was␣
↪made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source␣
↪for the Combined Work, excluding any source code for portions of the Combined Work␣
↪that, considered in isolation, are based on the Application, and not on the Linked␣
↪Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or␣
↪source code for the Application, including any data and utility programs needed for␣
↪reproducing the Combined Work from the Application, but excluding the System␣
↪Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.
You may convey a covered work under sections 3 and 4 of this License without being␣
↪bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.
If you modify a copy of the Library, and, in your modifications, a facility refers to␣
↪a function or data to be supplied by an Application that uses the facility (other␣
↪than as an argument passed when the facility is invoked), then you may convey a␣
↪copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in␣
↪the event an Application does not supply the function or data, the facility still␣
↪operates, and performs whatever part of its purpose remains meaningful, or
b) under the GNU GPL, with none of the additional permissions of this License␣
↪applicable to that copy.
3. Object Code Incorporating Material from Library Header Files.
The object code form of an Application may incorporate material from a header file␣
```

(continues on next page)

(continued from previous page)

```
↪that is part of the Library. You may convey such object code under terms of your␣
↪choice, provided that, if the incorporated material is not limited to numerical␣
↪parameters, data structure layouts and accessors, or small macros, inline functions␣
↪and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used␣
↪in it and that the Library and its use are covered by this License.
b) Accompany the object code with a copy of the GNU GPL and this license document.
4. Combined Works.
You may convey a Combined Work under terms of your choice that, taken together,␣
↪effectively do not restrict modification of the portions of the Library contained␣
↪in the Combined Work and reverse engineering for debugging such modifications, if␣
↪you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used␣
↪in it and that the Library and its use are covered by this License.
b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
c) For a Combined Work that displays copyright notices during execution, include the␣
↪copyright notice for the Library among these notices, as well as a reference␣
↪directing the user to the copies of the GNU GPL and this license document.
d) Do one of the following:
0) Convey the Minimal Corresponding Source under the terms of this License, and the␣
↪Corresponding Application Code in a form suitable for, and under terms that permit,␣
↪the user to recombine or relink the Application with a modified version of the␣
↪Linked Version to produce a modified Combined Work, in the manner specified by␣
↪section 6 of the GNU GPL for conveying Corresponding Source.
1) Use a suitable shared library mechanism for linking with the Library. A suitable␣
↪mechanism is one that (a) uses at run time a copy of the Library already present on␣
↪the user's computer system, and (b) will operate properly with a modified version␣
↪of the Library that is interface-compatible with the Linked Version.
e) Provide Installation Information, but only if you would otherwise be required to␣
↪provide such information under section 6 of the GNU GPL, and only to the extent␣
↪that such information is necessary to install and execute a modified version of the␣
↪Combined Work produced by recombining or relinking the Application with a modified␣
↪version of the Linked Version. (If you use option 4d0, the Installation Information␣
↪must accompany the Minimal Corresponding Source and Corresponding Application Code.␣
↪If you use option 4d1, you must provide the Installation Information in the manner␣
↪specified by section 6 of the GNU GPL for conveying Corresponding Source.)
5. Combined Libraries.
You may place library facilities that are a work based on the Library side by side in␣
↪a single library together with other library facilities that are not Applications␣
↪and are not covered by this License, and convey such a combined library under terms␣
↪of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library,␣
↪uncombined with any other library facilities, conveyed under the terms of this␣
↪License.
b) Give prominent notice with the combined library that part of it is a work based on␣
↪the Library, and explaining where to find the accompanying uncombined form of the␣
↪same work.
6. Revised Versions of the GNU Lesser General Public License.
The Free Software Foundation may publish revised and/or new versions of the GNU␣
↪Lesser General Public License from time to time. Such new versions will be similar␣
↪in spirit to the present version, but may differ in detail to address new problems␣
↪or concerns.
```

(continues on next page)

(continued from previous page)

```
Each version is given a distinguishing version number. If the Library as you received␣
↪it specifies that a certain numbered version of the GNU Lesser General Public␣
↪License "or any later version" applies to it, you have the option of following the␣
↪terms and conditions either of that published version or of any later version␣
↪published by the Free Software Foundation. If the Library as you received it does␣
↪not specify a version number of the GNU Lesser General Public License, you may␣
↪choose any version of the GNU Lesser General Public License ever published by the␣
↪Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future␣
↪versions of the GNU Lesser General Public License shall apply, that proxy's public␣
↪statement of acceptance of any version is permanent authorization for you to choose␣
↪that version for the Library.
```