TA-trackme-cribl: Cribl API integration for TrackMe

Introduction

This application provides API integration between Splunk and Cribl API:

- Configure one or more Cribl API targets (account)
- The Cribl target can be an on-premise Cribl deployment, or Cribl Cloud
- Access and perform API calls to Cribl in pure Splunk SPL via the cribl builtin command
- Run predefined built in functions to retrieve metrics for complex actions (example: get pipelines metrics), this allows charting real time metrics in Splunk from the Cribl API
- Roles Based Access Control and least privilege approach

Get pipelines example:

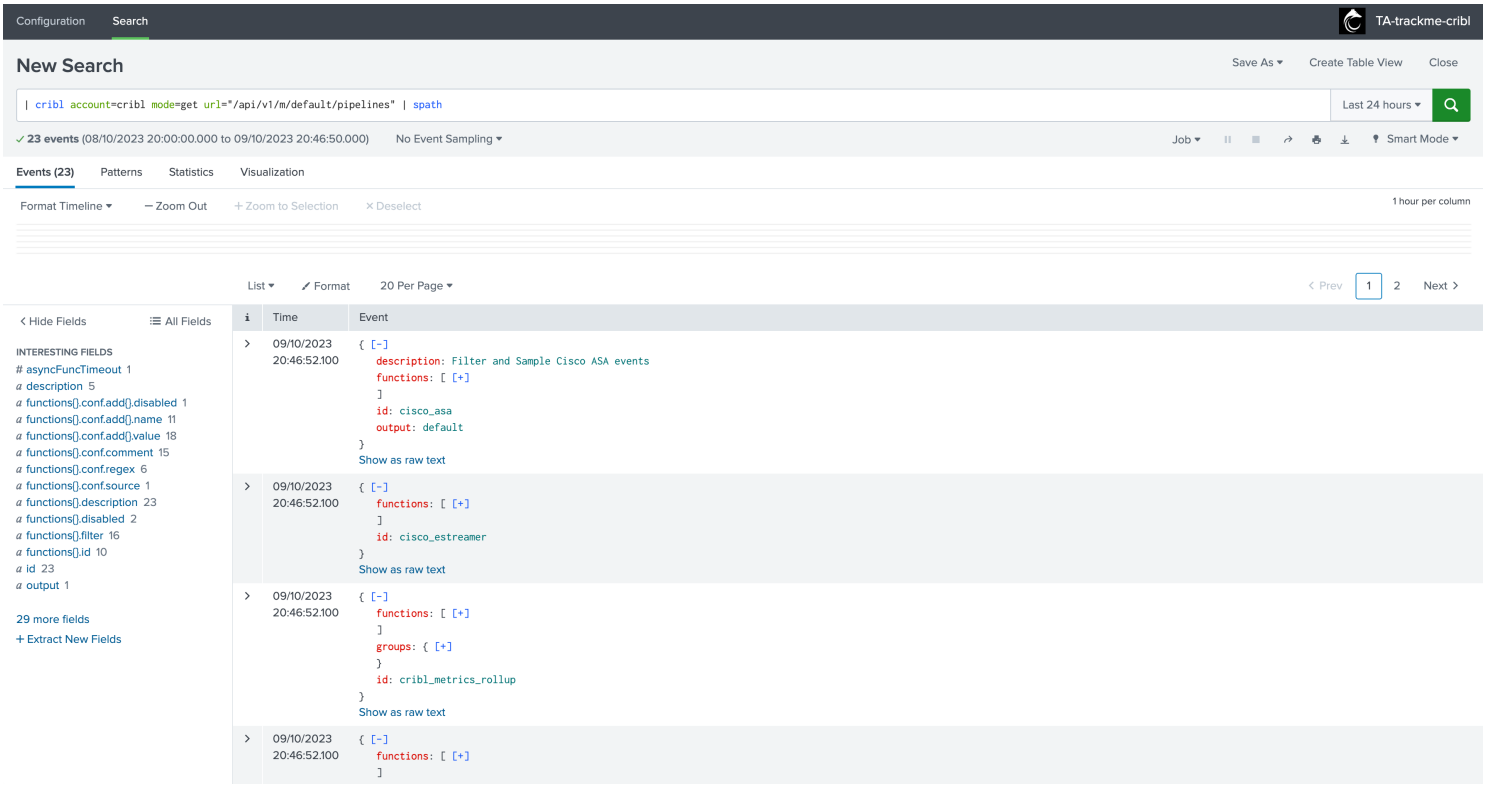


Figure 1: img

Real Time metrics charting:

License

This application is licensed upon the TrackMe EULA:

- https://docs.trackme-solutions.com/license.html

Terms & Conditions

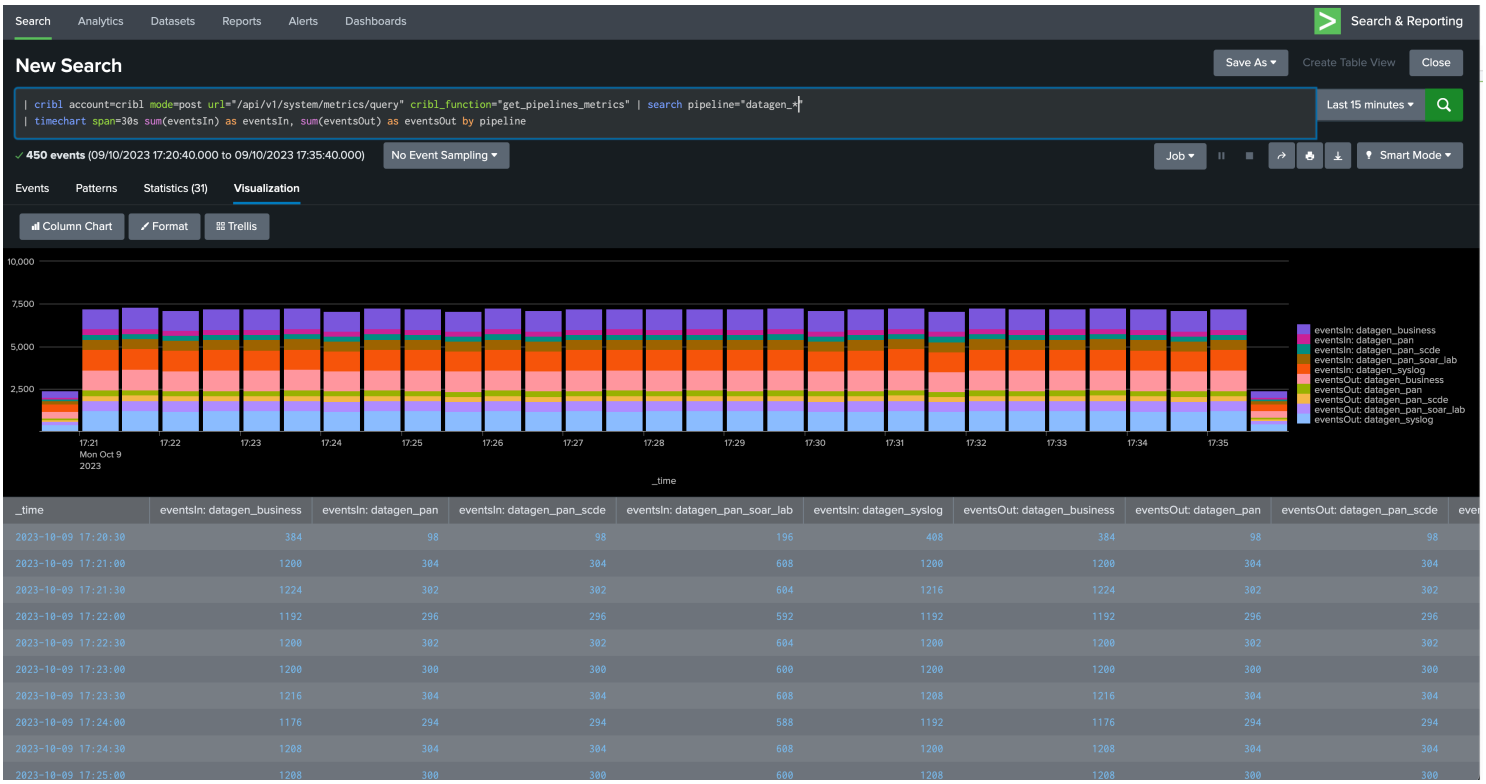
Our terms and conditions can be found at the following address:

- https://docs.trackme-solutions.com/terms_and_conditions.html

Support

Support conditions can be found at the following address:

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



_time	eventsIn: datagen_business	eventsIn: datagen_pan	eventsIn: datagen_pan_scde	eventsIn: datagen_pan_soar_lab	eventsIn: datagen_syslog	eventsOut: datagen_business	eventsOut: datagen_pan	eventsOut: datagen_pan_scde
2023-10-09 17:20:30	384	98	98	196	408	384	98	98
2023-10-09 17:21:00	1200	304	304	608	1200	1200	304	304
2023-10-09 17:21:30	1224	302	302	604	1216	1224	302	302
2023-10-09 17:22:00	1192	296	296	592	1192	1192	296	296
2023-10-09 17:22:30	1200	302	302	604	1200	1200	302	302
2023-10-09 17:23:00	1200	300	300	600	1200	1200	300	300
2023-10-09 17:23:30	1216	304	304	608	1208	1216	304	304
2023-10-09 17:24:00	1176	294	294	588	1192	1176	294	294
2023-10-09 17:24:30	1208	304	304	608	1200	1208	304	304
2023-10-09 17:25:00	1208	300	300	600	1208	1208	300	300

Figure 2: img

Requirements

Applications dependencies

This application has no dependencies regarding other Splunk applications.

Network requirements

The Splunk Search Head where this application is deployed needs to be able to reach the Cribl URL target on the associated port. (typically HTTPs over 443)

Installation & deployment matrix

Install this Splunk application on the following target:

- Search Head(s), standalone or Search Head Cluster

Configuration

Once deployed, access the Configuration UI and create a first Cribl API account:

Parameter	Description
Name	Enter a unique name for this account. (default: cribl)
Cribl deployment type	The type of Cribl deployment, Cloud or on-premise
cribl_cloud_organization_id	If Cloud, specify the organization ID
cribl_onprem_leader_url	If on-premise, specify the leader URL in the format `https://<hostname>:<port>`
cribl_client_id	If on-premise, the username for the API connection, if using Cloud, the client_id.
cribl_client_secret	If on-premise, the password for the API connection, if using Cloud, the client_secret.
rbac_roles	A comma separated list of Splunk roles a user must be a member of to be allowed to used this account, the role must be a true membership
cribl_ssl_verify	Enable or disable SSL verification for Cribl on-prem only, for testing and development purposes. (mandatory for Cloud)
cribl_ssl_certificate_path	To verify a self-signed or internal PKI certificate, you can specify the local path to the PEM file

Account example:

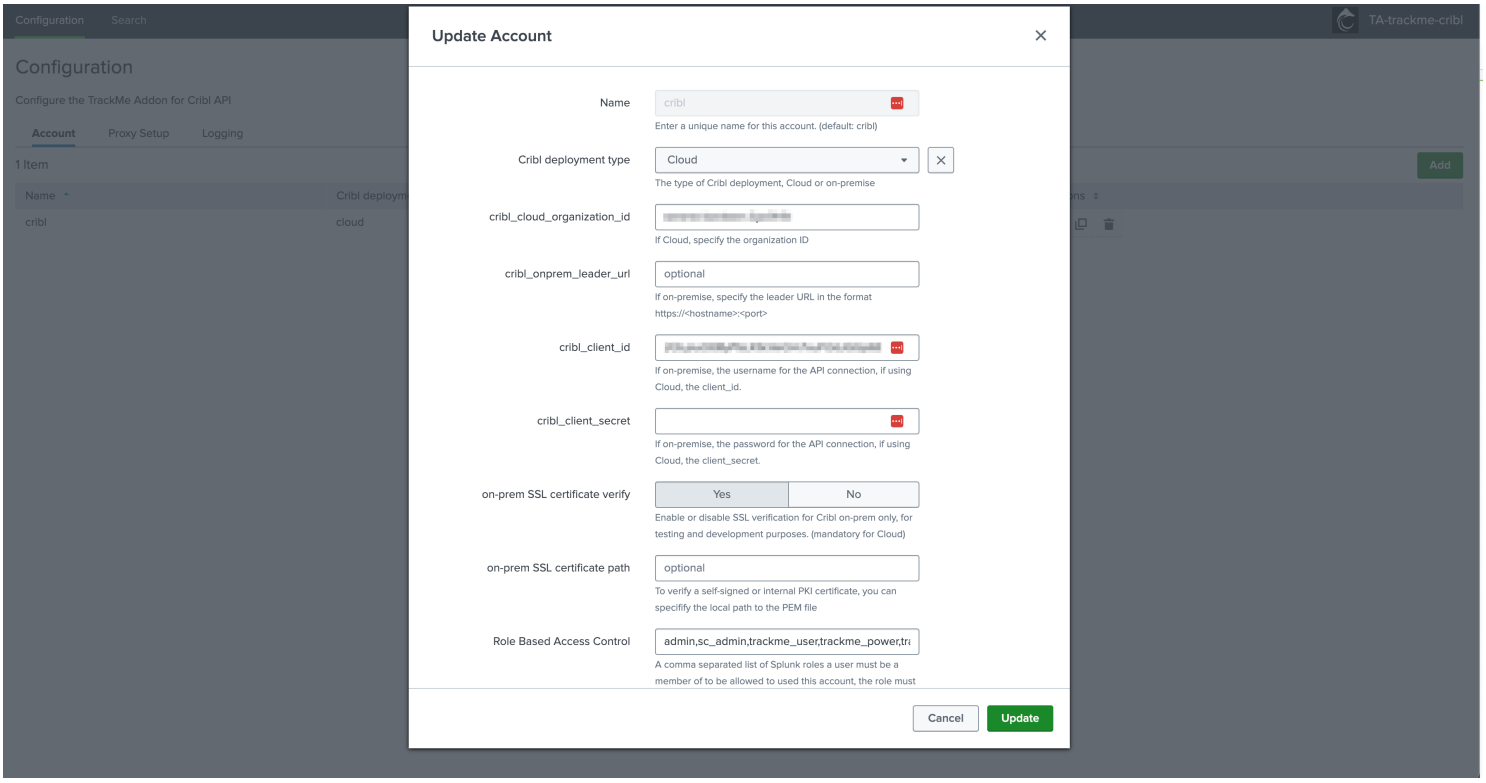


Figure 3: img

When creating the Account, the application will automatically attempt to access and authenticate against the Cribl API, if there are any failures, the UI will refuse the account creation and will return the exception.

You can also test the connectivity prior to the configuration of the account, in fact the UI calls this same endpoint to verify the connectivity for the account configuration:

Cribl Cloud example:

```
| cribl mode=post url="/services/cribl/v1/test_cribl_connectivity" body="{'cribl_deployment_type': 'cloud', 'cri
```

Capabilities & Roles Based Access Control

Capabilities & builtin roles

This application implements a least privilege approach, this means you do not require high privilege capabilities to be able to use the application and the underneath secrets, instead a capability can be added to your roles, or you can inherit from the builtin roles:

- capability: `criblapi`
- builtin roles: `cribl_api`

Roles Based Access Control

On a per Cribl account basis, you can define the list of Splunk roles a user must be a member, this setting is defined at the level of the account:

- `rbac_roles`, defaults to: admin,sc_admin,trackme_user,trackme_power,trackme_admin

This is a true membership, the user must be a member of the roles. (not inheriting one of these roles)

Logs & Troubleshooting

Internal REST API

The Application implements an API for its internal processing, logs are available at:

```
index=_internal sourcetype="cribl:rest_api"
```

Custom command

The Application provides a builtin custom command `cribl` allowing to interact with the Cribl API, logs can be found here:

`index=_internal sourcetype="cribl:custom_commands:cribl"`

Usage

Cribl API interactions

The following command sheet shows various useful usage to interact with the Cribl API:

- When working with metrics, you can use the Splunk time range filter which will automatically be taken into account, the period will be reflected when calling the API
- The Cribl API allows up to the last 3 hours of metric with a granular 10 seconds time window
- Beyond 3 hours, Cribl API rolls metrics with a 10 minutes span
- The `cribl` custom command automatically takes this into account
- When working with configuration, use `spath` to automatically extracted the fields from the JSON API response
- You can also perform `POST` calls using `mode=post` and the `body` option

Get workers count & information

```
| cribl account=cribl mode=get url="/api/v1/master/groups?fields=workerCount&product=stream" | spath
```

Get extended workers info & stats

```
| cribl account=cribl mode=get url="/api/v1/master/workers?filterExp=info.cribl.distMode%3D%3D%22worker%22" | sp
```

Get worker count

```
| cribl account=cribl mode=get url="/api/v1/master/summary/workers?filterExp=info.cribl.distMode%3D%3D%22worker%
```

Get groups

```
| cribl account=cribl mode=get url="/api/v1/master/groups" | spath
```

Get groups (Cribl Stream only)

```
| cribl account=cribl mode=get url="/api/v1/master/groups?product=stream" | spath
```

Get groups (only stream)

```
| cribl account=cribl mode=get url="/api/v1/master/groups?product=stream" | spath
```

Get inputs

```
| cribl account=cribl mode=get url="/api/v1/m/default/system/inputs" | spath
```

Get outputs

```
| cribl account=cribl mode=get url="/api/v1/m/default/system/outputs" | spath
```

Get routes

```
| cribl account=cribl mode=get url="/api/v1/m/default/routes" | spath
```

Get pipelines

```
| cribl account=cribl mode=get url="/api/v1/m/default/pipelines" | spath
```

Get packs

```
| cribl account=cribl mode=get url="/api/v1/m/default/packs" | spath
```

Get system metrics

```
| cribl account=cribl mode=post url="/api/v1/system/metrics/query" cribl_function="get_global_metrics"
```

Get destination metrics

```
| cribl account=cribl mode=post url="/api/v1/system/metrics/query" cribl_function="get_destinations_metrics"
```

Get pipelines metrics

```
| cribl account=cribl mode=post url="/api/v1/system/metrics/query" cribl_function="get_pipelines_metrics"
```

Get routes metrics

```
| cribl account=cribl mode=post url="/api/v1/system/metrics/query" cribl_function="get_routes_metrics"
```

Get system messages (there can be nothing here)

```
| cribl account=cribl mode=get url="/api/v1/system/messages" | spath
```

Get banner (there can be nothing here)

```
| cribl account=cribl mode=get url="/api/v1/system/banners" | spath
```

Get recent actions

```
| cribl account=cribl mode=get url="/api/v1/ui/recentActions" | spath
```

Internal API

The following sections shows internal API calls:

Get request_info

```
| cribl mode=get url="/services/cribl/v1/request_info"
```

Test connectivity to Cribl before creating an account

```
| cribl mode=post url="/services/cribl/v1/test_cribl_connectivity" body="{'cribl_deployment_type': 'cloud', 'cri
```

Access the account (for least privileges access) *This is used internally to access credentials with a least privileges approach.*

```
| cribl mode=post url="/services/cribl/v1/get_account" body="{'account': 'cribl'}"
```