trackme Documentation

Release 1

Guilhem Marchand

Aug 08, 2022

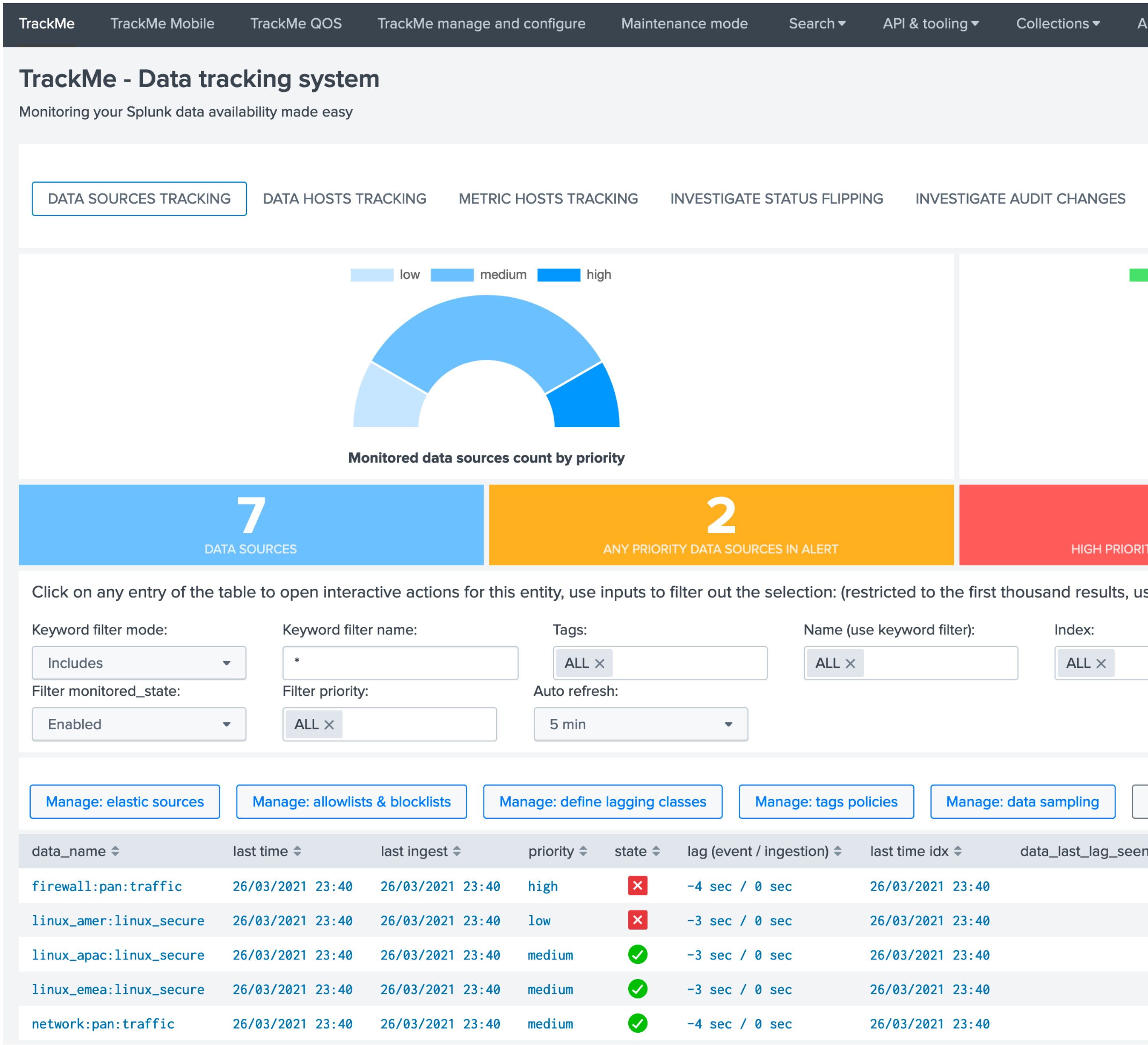
Overview

Important: !Major changes and announcements!

- Major changes to TrackMe are coming, very soon!
- TrackMe v2, the next generation of the solution, is expected to be GA within a couple of months
- This totally new version keeps all that made the success of TrackMe, and adds powerful key features and a brand new user experience
- TrackMe v2 addresses as well the deprecation of HTML dashboard which affects the current version of TrackMe
- Keep on track, this is coming

TrackMe provides automated monitoring and visibility insight of your data sources, with a powerful user interface and workflow for Splunk product owners to detect and alert on lack of availability, abnormal latency, volume outliers detection and quality issues:

- Discover and store key states information of data sources, data hosts and metric hosts availability
- Provides a powerful user interface to manage activation states, configuration and quickly identify data availability failures
- Analyse and detect lack of data and performance lagging of data sources and hosts within your Splunk deployment
- Behaviour analytic with outlier detection based on machine learning outliers calculations
- Behaviour analytic with data sampling and event format recognition, monitor and detect anomalies in raw events to detect event format changes or misbehaviour based on builtin rules and extended with your own custom rules
- Create elastic sources for any kind of custom monitoring requirements based on tstats / raw / mstats / from searches to fullfill any requirements
- Record and investigate historical changes of statuses, as well as administrators changes (audit flipping and changes)
- Easy administration via graphical human interface from A to Z
- No matters the purpose of your Splunk deployment, trackMe will become an essential piece of your deployment, providing key value for PCI or compliance requirements
- Keep things under your control and be the first to know when data is not available, get alerted before your users get back to you!



TrackMe - Data tracking system

Monitoring your Splunk data availability made easy

Monitored data sources count by priority

7 DATA SOURCES

2 ANY PRIORITY DATA SOURCES IN ALERT

HIGH PRIORIT

Click on any entry of the table to open interactive actions for this entity, use inputs to filter out the selection: (restricted to the first thousand results, us

Keyword filter mode: Includes

Keyword filter name: *

Tags: ALL ×

Name (use keyword filter): ALL ×

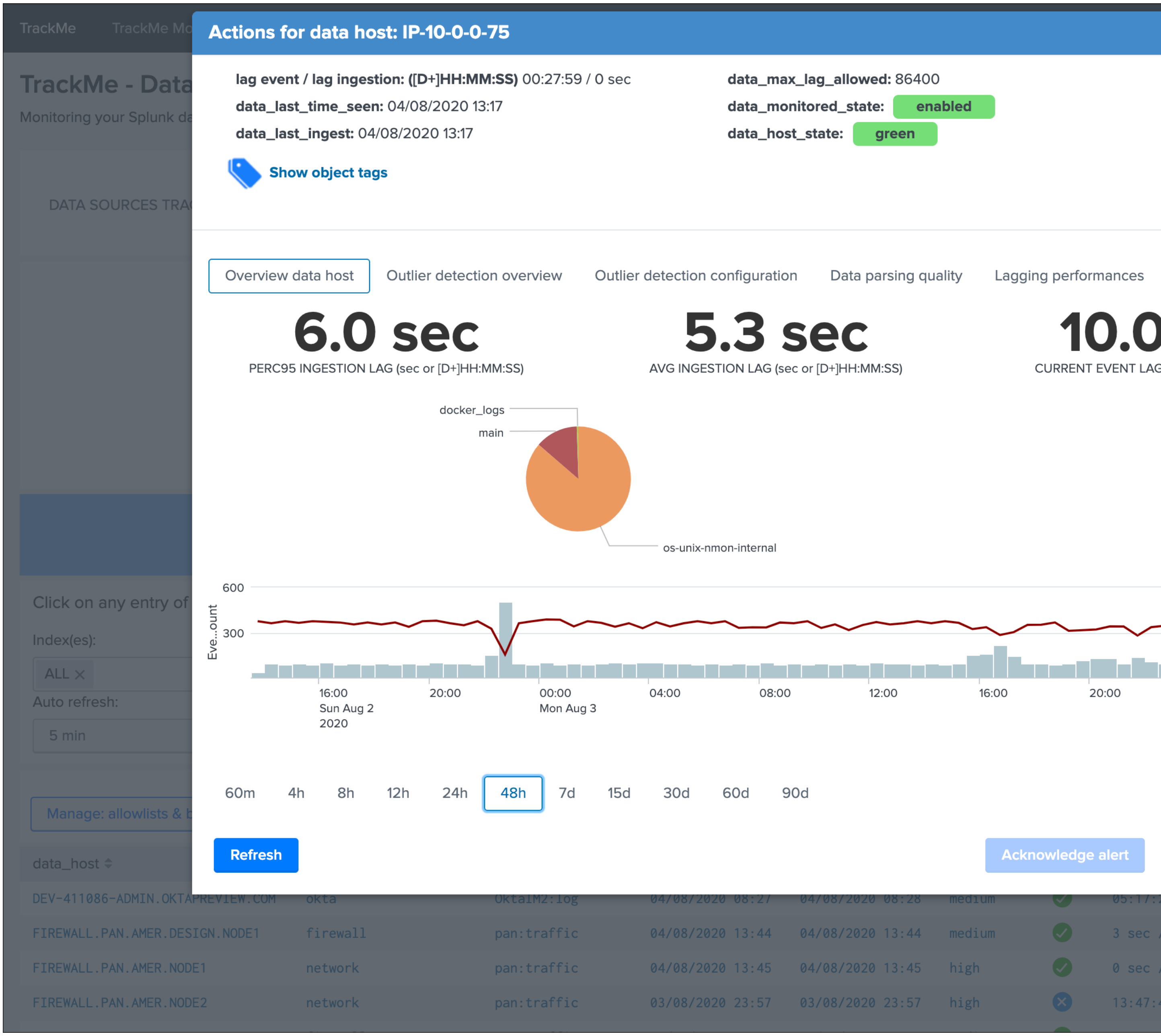
Index: ALL ×

Filter monitored_state: Enabled

Filter priority: ALL ×

Auto refresh: 5 min

data_name	last time	last ingest	priority	state	lag (event / ingestion)	last time idx	data_last_lag_seen
firewall:pan:traffic	26/03/2021 23:40	26/03/2021 23:40	high	✗	-4 sec / 0 sec	26/03/2021 23:40	
linux_amer:linux_secure	26/03/2021 23:40	26/03/2021 23:40	low	✗	-3 sec / 0 sec	26/03/2021 23:40	
linux_apac:linux_secure	26/03/2021 23:40	26/03/2021 23:40	medium	✓	-3 sec / 0 sec	26/03/2021 23:40	
linux_emea:linux_secure	26/03/2021 23:40	26/03/2021 23:40	medium	✓	-3 sec / 0 sec	26/03/2021 23:40	
network:pan:traffic	26/03/2021 23:40	26/03/2021 23:40	medium	✓	-4 sec / 0 sec	26/03/2021 23:40	



Actions for data host: IP-10-0-0-75

lag event / lag ingestion: ([D+]HH:MM:SS) 00:27:59 / 0 sec

data_last_time_seen: 04/08/2020 13:17

data_last_ingest: 04/08/2020 13:17

data_max_lag_allowed: 86400

data_monitored_state: enabled

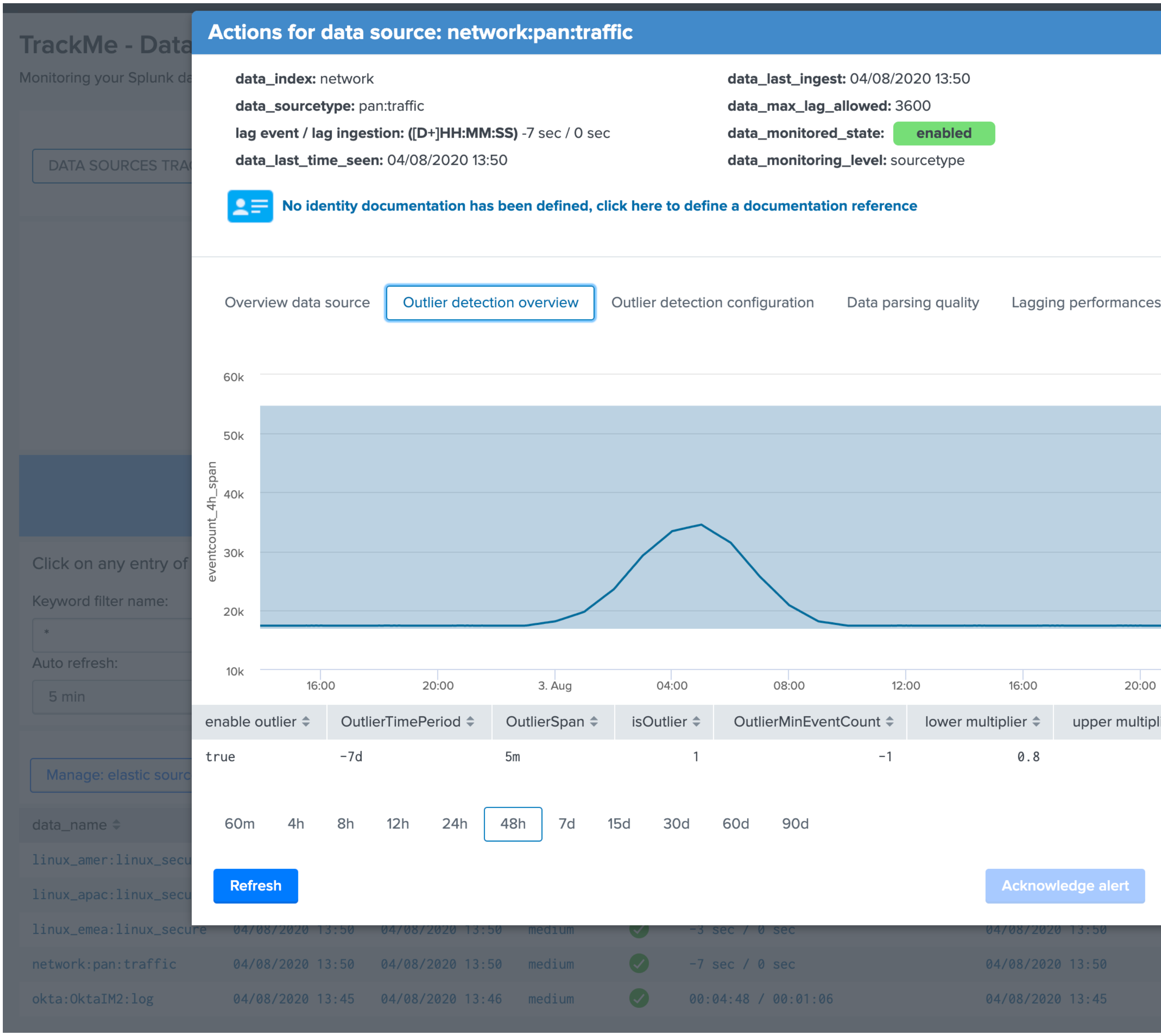
data_host_state: green

6.0 sec
PERC95 INGESTION LAG (sec or [D+]HH:MM:SS)

5.3 sec
AVG INGESTION LAG (sec or [D+]HH:MM:SS)

10.0
CURRENT EVENT LAG

```
DEV-411086-ADMIN.OKTAPREVIEW.COM   okta       OktaIM2:log   04/08/2020 08:27   04/08/2020 08:28   medium   05:17:
FIREWALL.PAN.AMER.DESIGN.NODE1     firewall   pan:traffic   04/08/2020 13:44   04/08/2020 13:44   medium   3 sec
FIREWALL.PAN.AMER.NODE1            network    pan:traffic   04/08/2020 13:45   04/08/2020 13:45   high     0 sec
FIREWALL.PAN.AMER.NODE2            network    pan:traffic   03/08/2020 23:57   03/08/2020 23:57   high     13:47:
```



Actions for data source: network:pan:traffic

data_index: network

data_sourcetype: pan:traffic

lag event / lag ingestion: ([D+]HH:MM:SS) -7 sec / 0 sec

data_last_time_seen: 04/08/2020 13:50

data_last_ingest: 04/08/2020 13:50

data_max_lag_allowed: 3600

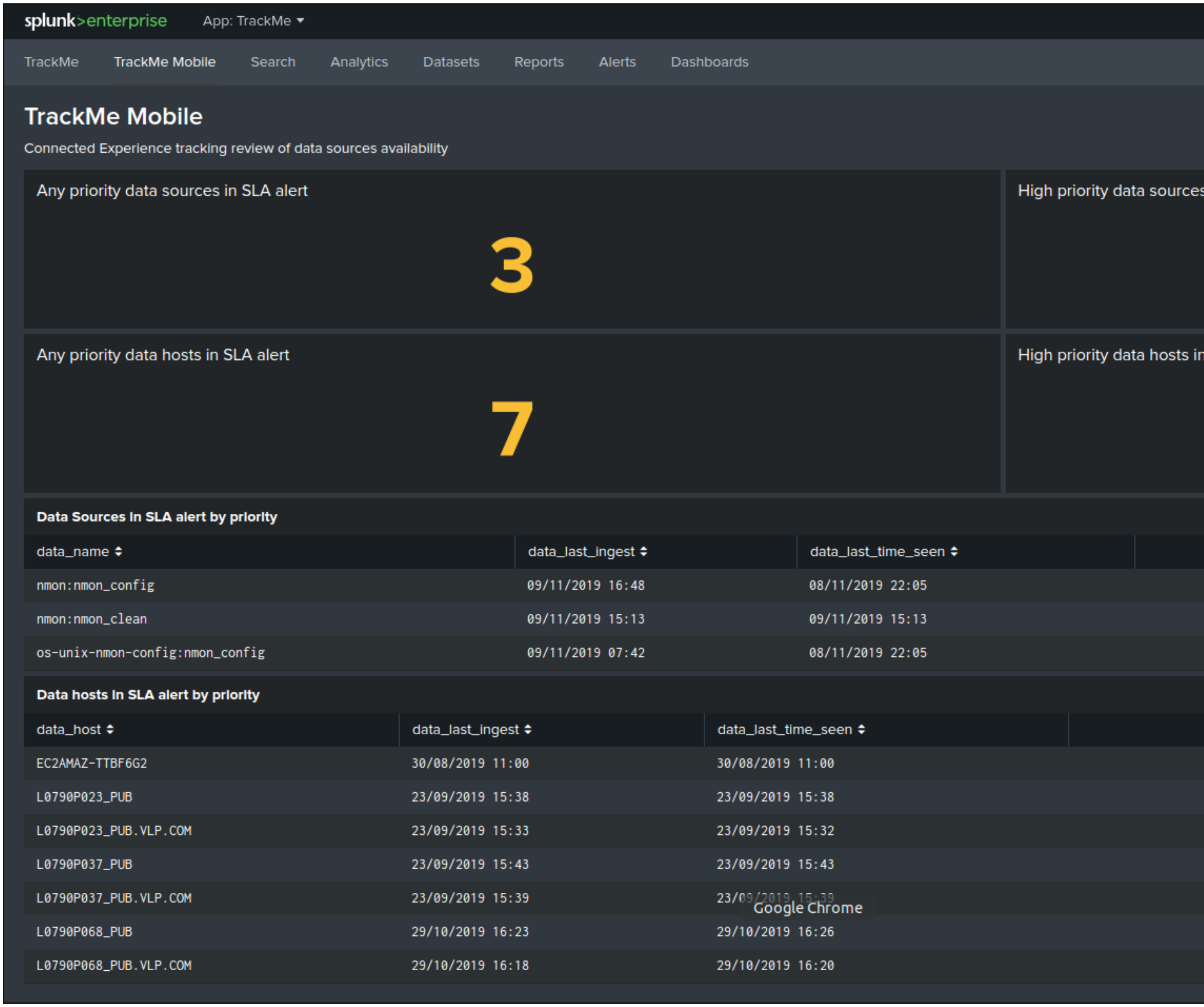
data_monitored_state: enabled

data_monitoring_level: sourcetype

No identity documentation has been defined, click here to define a documentation reference

enable outlier	OutlierTimePeriod	OutlierSpan	isOutlier	OutlierMinEventCount	lower multiplier	upper multipl
true	-7d	5m	1	-1	0.8	

data_name						
linux_amer:linux_secu						
linux_apac:linux_secu						
linux_emea:linux_secure	04/08/2020 13:50	04/08/2020 13:50	medium	✓	-3 sec / 0 sec	04/08/2020 13:50
network:pan:traffic	04/08/2020 13:50	04/08/2020 13:50	medium	✓	-7 sec / 0 sec	04/08/2020 13:50
okta:OktaIM2:log	04/08/2020 13:45	04/08/2020 13:46	medium	✓	00:04:48 / 00:01:06	04/08/2020 13:45



TrackMe Mobile

Connected Experience tracking review of data sources availability

Any priority data sources in SLA alert

3

High priority data sources

Any priority data hosts in SLA alert

7

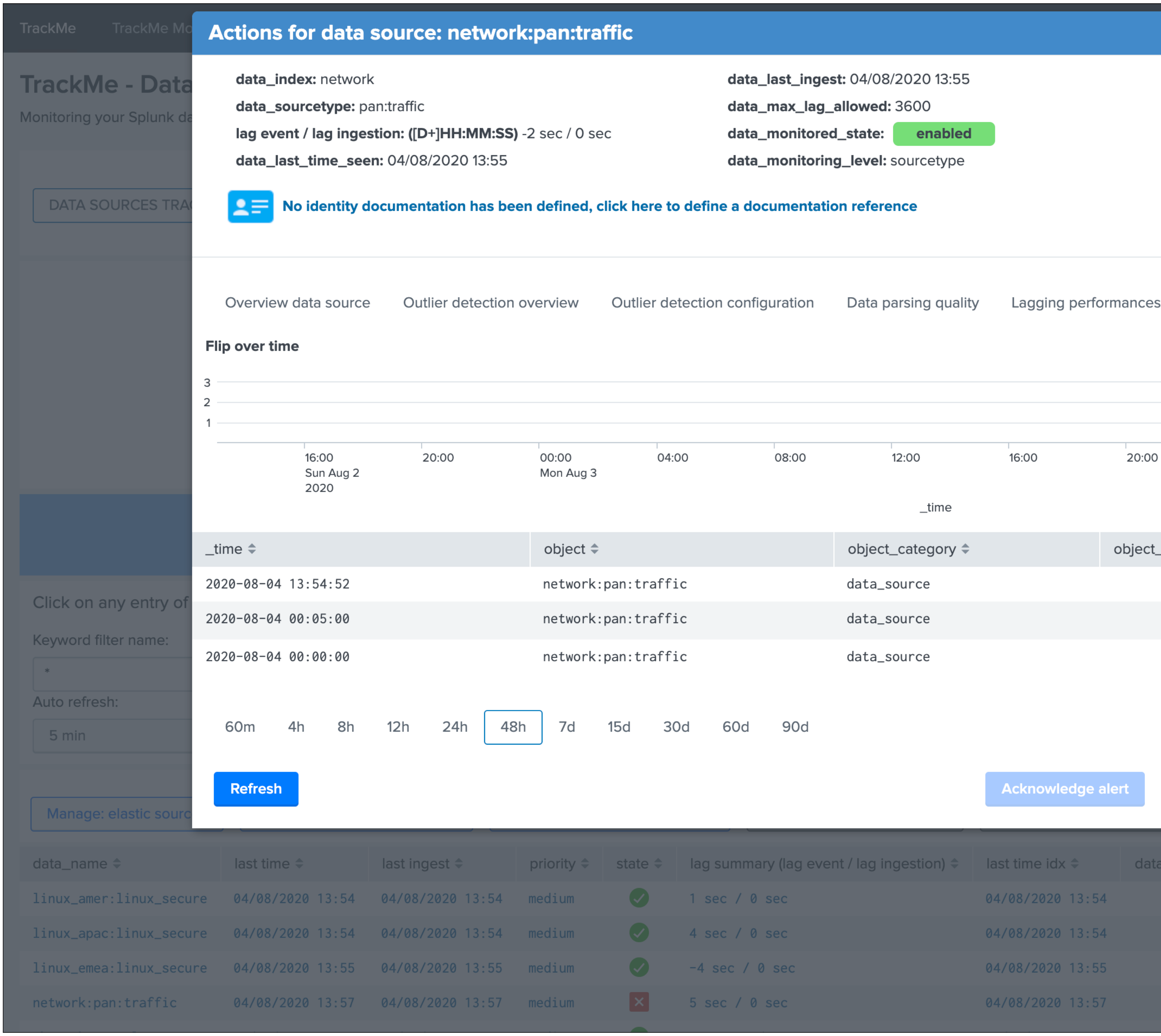
High priority data hosts in

Data Sources in SLA alert by priority

data_name	data_last_ingest	data_last_time_seen
nmon:nmon_config	09/11/2019 16:48	08/11/2019 22:05
nmon:nmon_clean	09/11/2019 15:13	09/11/2019 15:13
os-unix-nmon-config:nmon_config	09/11/2019 07:42	08/11/2019 22:05

Data hosts in SLA alert by priority

data_host	data_last_ingest	data_last_time_seen
EC2AMAZ-TTBF6G2	30/08/2019 11:00	30/08/2019 11:00
L0790P023_PUB	23/09/2019 15:38	23/09/2019 15:38
L0790P023_PUB.VLP.COM	23/09/2019 15:33	23/09/2019 15:32
L0790P037_PUB	23/09/2019 15:43	23/09/2019 15:43
L0790P037_PUB.VLP.COM	23/09/2019 15:39	23/09/2019 15:39
L0790P068_PUB	29/10/2019 16:23	29/10/2019 16:26
L0790P068_PUB.VLP.COM	29/10/2019 16:18	29/10/2019 16:20



Actions for data source: network:pan:traffic

data_index: network
data_sourcetype: pan:traffic
lag event / lag ingestion: ([D+]HH:MM:SS) -2 sec / 0 sec
data_last_time_seen: 04/08/2020 13:55

data_last_ingest: 04/08/2020 13:55
data_max_lag_allowed: 3600
data_monitored_state: enabled
data_monitoring_level: sourcetype

No identity documentation has been defined, click here to define a documentation reference

_time	object	object_category	object_
2020-08-04 13:54:52	network:pan:traffic	data_source	
2020-08-04 00:05:00	network:pan:traffic	data_source	
2020-08-04 00:00:00	network:pan:traffic	data_source	

data_name	last time	last ingest	priority	state	lag summary (lag event / lag ingestion)	last time idx	data
linux_amer:linux_secure	04/08/2020 13:54	04/08/2020 13:54	medium	✓	1 sec / 0 sec	04/08/2020 13:54	
linux_apac:linux_secure	04/08/2020 13:54	04/08/2020 13:54	medium	✓	4 sec / 0 sec	04/08/2020 13:54	
linux_emea:linux_secure	04/08/2020 13:55	04/08/2020 13:55	medium	✓	-4 sec / 0 sec	04/08/2020 13:55	
network:pan:traffic	04/08/2020 13:57	04/08/2020 13:57	medium	✗	5 sec / 0 sec	04/08/2020 13:57	

Why this application?

Splunk administrators and engineers have to spend a good amount of time and energy to on-board and monitor data sources, which becomes more and more complex and time consuming with the explosion of volume and variety of data.

However, it is very frequent to realise after math that something went wrong, for some reason the sender stopped sending, an upgrade broke a configuration, a network rule was lost, an unexpected side effect of a change occurred, parsing issues are not detected. . .

No administrator should be informed of an issue in the data flow by the customer or the end users, this is why you need pro-activity, costless and scalable availability monitoring.

with the massive amount and variety of data sources, this becomes easily a painful and problematic activity, this application aims to drastically help you in these daily tasks.

TrackMe provides a handy user interface associated with an efficient data discovery, state and alerting workflow.

Made by Splunk admins for Splunk admins, the TrackMe application provides builtin powerful features to monitor and administer you data source monitoring the easy way!

Use cases for TrackMe?

No matters the purpose of your Splunk deployment, trackMe will easily become an essential and positive piece of your Splunk journey:

- Security Operation Centers (SOC) with or without Enterprise Security compliance: detect lack of data, abnormal latency potentially impacting your security posture
- PCI and compliance: deliver, alert and action
- Monitoring and insight visibility about your indexes, sourcetypes, events and metrics
- General data activity monitoring and detection of Zombie data
- Continous and automated data quality assessment
- PII data detection with custom regular expression based rules and data sampling
- many more!

CHAPTER 1

Overview:

1.1 About

- Author: Guilhem Marchand, Splunk certified consultant and part of Splunk Professional Services
- First release published in July 2019
- License: Apache License 2.0

1.2 Compatibility

1.2.1 Splunk core compatibility

Splunk core

- TrackMe is compatible with Splunk 8.x and later. (Python3 only support starting from release 1.2.52)

The previous main branch of TrackMe (V1.1.x) was compatible with Splunk versions starting from Splunk 7.0.x, which changed from 7.2.x due to the usage of the mcollect command.

The latest release compatible with Splunk 7.2.x/7.3.x and Python2 is the release 1.2.51.

1.2.2 Splunk Cloud compatibility

Splunk Cloud vetting

- TrackMe is vetted for Splunk Cloud deployments
- When a new release is pubished, there can be some time before the last release is vetted

- Even if the latest release would not be vetted yet, open a request to Cloud Ops and the vetting process will be achieved

Splunk Cloud notes:

- When a new release of TrackMe is published, there is a certain amount of time required for the vetting process to be performed
- As such, in Splunk Base the current release might appear as not vetted yet
- To get TrackMe deployed in Splunk Cloud, you can submit a case to Cloud Ops, if the latest release is not vetted yet, you can request its deployment and vetting will be performed, or you can check what the latest version is in Splunk Base and request its deployment
- TrackMe currently requires an "assisted" deployment, which means that it is not currently deployable by Splunk Cloud Self Services and you need to submit a request

Checkout the latest vetted version in Splunk Base:

- In Splunk Base, if the latest version is not yet vetted the following message is displayed "This version is not yet available for Splunk Cloud."
- Use the version dropdown to select an earlier version to see "Splunk Cloud" listed in the Products, which confirms that this version has been vetted

1.2.3 Python compatibility

Python 3 compatibility

- TrackMe supports Python 3 exclusively
- Python2 support was dropped in release 1.2.51, starting from release 1.2.52 TrackMe only supports Python3

1.2.4 Web Browser compatibility

The application can be used with any of the supported Web Browser by Splunk:

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

1.3 Support

1.3.1 Splunk community

Open a question in Splunk community:

- https://community.splunk.com

1.3.2 Splunk community slack

Contact me on Splunk community slack, and even better, ask the community!

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

1.3.3 Open a issue in Git

To report an issue, request a feature change or improvement, please open an issue in Github:

- https://github.com/guilhemmarchand/trackme/issues

1.3.4 Email support

- support@trackme-solutions.com

1.4 Download

The application can be downloaded from:

1.4.1 Splunk base

- https://splunkbase.splunk.com/app/4621

1.4.2 GitHub

- https://github.com/guilhemmarchand/trackme

CHAPTER 2

Deployment and configuration:

2.1 Deployment & Upgrades

2.1.1 Deployment matrix

Splunk roles	required
Search head	yes
Indexer tiers	no

If Splunk search heads are running in Search Head Cluster (SHC), the Splunk application must be deployed by the SHC deployer.

2.1.2 Dependencies

Hint: Since TrackMe 1.2.0, there are several application dependencies

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120
- Splunk SA CIM - Splunk Common Information Model, Splunk Base: https://splunkbase.splunk.com/app/1621 (require for alert actions and result ingestion purposes)

2.1.3 Indexes

Hint: Since TrackMe 1.2.0, the application requires the creation of an event index and a metric index

- summary event index defaults to `trackme_summary`, handled by the macro `trackme_idx`
- metric index defaults to `trackme_metrics`, handled by the macro `trackme_metrics_idx`

Customise these macros via the UI or via a local/macros.conf file if you wish to use a different index naming convention.

2.1.4 Initial deployment

The deployment of the Splunk application is very straight forward:

- Using the application manager in Splunk Web (Settings / Manages apps)
- Extracting the content of the tgz archive in the "apps" directory of Splunk
- For SHC configurations (Search Head Cluster), extract the tgz content in the SHC deployer and publish the SHC bundle

2.1.5 Upgrades

Upgrading the Splunk application is pretty much the same operation than the initial deployment.

All of TrackMe components and configuration items are upgraded resilient, in respects with Splunk configuration good practices.

2.2 Step by step installation and configuration

2.2.1 Step 1: Deploy TrackMe

Where to deploy TrackMe

The first question you need an answer when you look at deploying TrackMe the very first time is generally where to deploy TrackMe?

To answer this question in a nutshell:

- TrackMe is deployed exclusively on a search head layer, there are no components running on forwarders (Universal Forwarders, Heavy Forwarders) or Splunk indexers
- The search head layer targets depends on your preference, it can be standalone search head (SH) you are using to run monitoring tools, the monitoring console host (MC) or a Search Head Cluster (SHC)
- The essential part of the content TrackMe is generated in dedicated indexes (summary events and metrics) and non-replicated KVstore collections which have near zero impacts on the search knowledge bundle size that is replicated automatically to your indexers

Configure indexes

Once you decided which search head layer will host TrackMe, the next step is to configure its indexes.

TrackMe requires the creation of two indexes, one for the summary events and one for the metrics, the second is a metric type of index opposed to events indexes, TrackMe includes the following indexes.conf:

`default/indexes.conf`

```
[trackme_summary]
coldPath = $SPLUNK_DB/trackme_summary/colddb
homePath = $SPLUNK_DB/trackme_summary/db
thawedPath = $SPLUNK_DB/trackme_summary/thaweddb

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

Hint: Indexes definition on Search Heads and Indexers

- before deploying TrackMe, ensure to declare these indexes on your indexers, in clustering mode this means updating on your indexes on the master-apps and pushing the cluster bundle
- indexes need to be declared on the search head layer hosting TrackMe too (and other search heads as a good practice), data will not be stored on the search head but this allows autocompletion and collect/mcollect features
- given that TrackMe comes with a built-in definition as part of the package, you do not need to handle this normally on the search head but on the indexers (unless the default/indexes.conf is trashed via some automation)

In well-designed Splunk environments, you will most likely use volumes on the indexers, you would translate this within your indexer configuration to the following configuration potentially:

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

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

To be adapted depending on your volume configuration!

Using a different naming convention for indexes

In some cases you may need to use a different naming convention for the two TrackMe indexes, this is not an issue and the only thing you will need to update in the application will be defining the custom configuration in the following two macros:

- `trackme_idx`
- `trackme_metrics_idx`

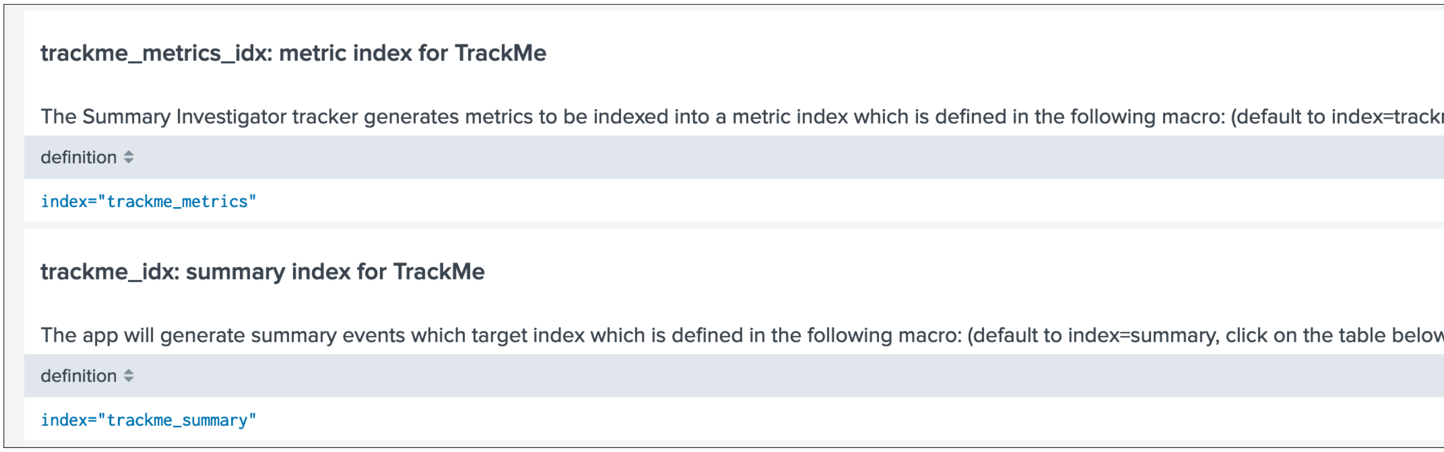
The out of the box definition is:

`default/macros.conf`

```
[trackme_idx]
definition = index="trackme_summary"
iseval = 0

[trackme_metrics_idx]
definition = index="trackme_metrics"
iseval = 0
```

Up to your choice, you can do this manually in the same time you deploy TrackMe (in a local/macros.conf) or you can update this within the UI once the application has been deployed:

`TrackMe manage and configure`



trackme_metrics_idx: metric index for TrackMe

The Summary Investigator tracker generates metrics to be indexed into a metric index which is defined in the following macro: (default to index=trackı

definition ⇕
index="trackme_metrics"

trackme_idx: summary index for TrackMe

The app will generate summary events which target index which is defined in the following macro: (default to index=summary, click on the table below

definition ⇕
index="trackme_summary"

2.2.2 Step 2: Configure TrackMe to match your needs

TrackMe strategy for data access - What TrackMe will be looking at

The first thing to consider once your deployed TrackMe is to design your strategy for which data TrackMe will be monitoring.

By default, TrackMe will search efficiently (tstats based queries for events) against any index the search head can access, you can choose between **two** main strategies:

- Either you use `allow listing` features to restrict access to explicit list of indexes
- Either you use `block listing` features to be looking at everything **but** specific items you exclude explicitly (indexes, sourcetypes, hosts and so forth)

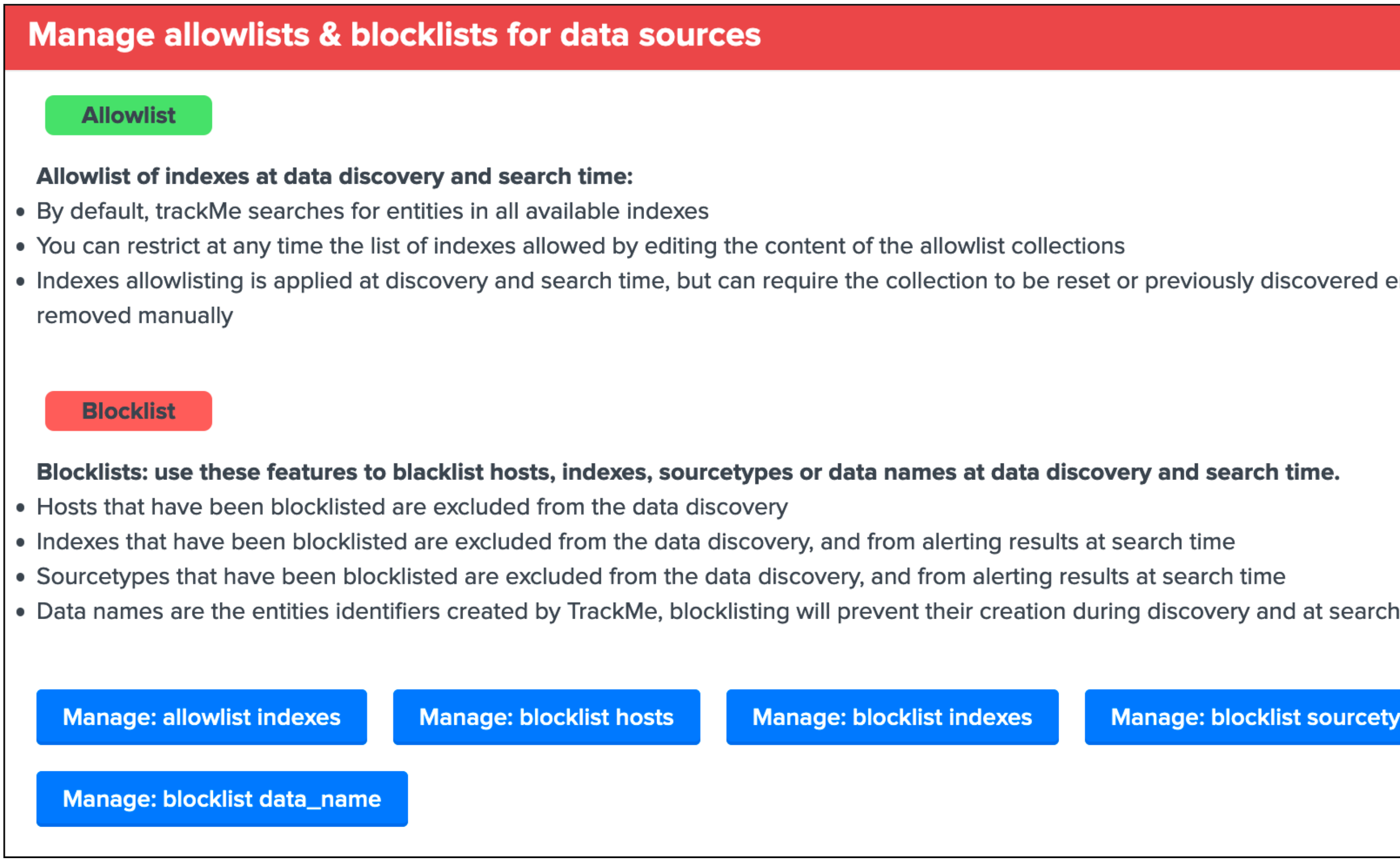
Both approaches are configurable via the TrackMe UI, and both approaches have its advantages and disadvantages:

- Allow listing is the cleaner and more efficient way but requires that you have a deep knowledge of your environment
- Allow listing can lead to be missing things you should have been tracking if not configured properly nor maintained over time
- Block listing can require more work over time as you need to exclude the bad things you do not want to consider

The two approaches are not exclusive, you can use allow listing AND block listing! This means you can restrict the basic index access scope AND block list certain things you do not want to consider.

See *Allowlisting & Blocklisting* in the User guide.

Interface to allow listing and block listing definitions:



Manage allowlists & blocklists for data sources

Allowlist

Allowlist of indexes at data discovery and search time:

- By default, trackMe searches for entities in all available indexes
- You can restrict at any time the list of indexes allowed by editing the content of the allowlist collections
- Indexes allowlisting is applied at discovery and search time, but can require the collection to be reset or previously discovered e removed manually

Blocklist

Blocklists: use these features to blacklist hosts, indexes, sourcetypes or data names at data discovery and search time.

- Hosts that have been blocklisted are excluded from the data discovery
- Indexes that have been blocklisted are excluded from the data discovery, and from alerting results at search time
- Sourcetypes that have been blocklisted are excluded from the data discovery, and from alerting results at search time
- Data names are the entities identifiers created by TrackMe, blocklisting will prevent their creation during discovery and at search

Manage: allowlist indexes
Manage: blocklist hosts
Manage: blocklist indexes
Manage: blocklist sourcety
Manage: blocklist data_name

Hint: Each main TrackMe categories have their own definitions for allow and block listing: `Data sources`, `Data hosts` and `Metric hosts`

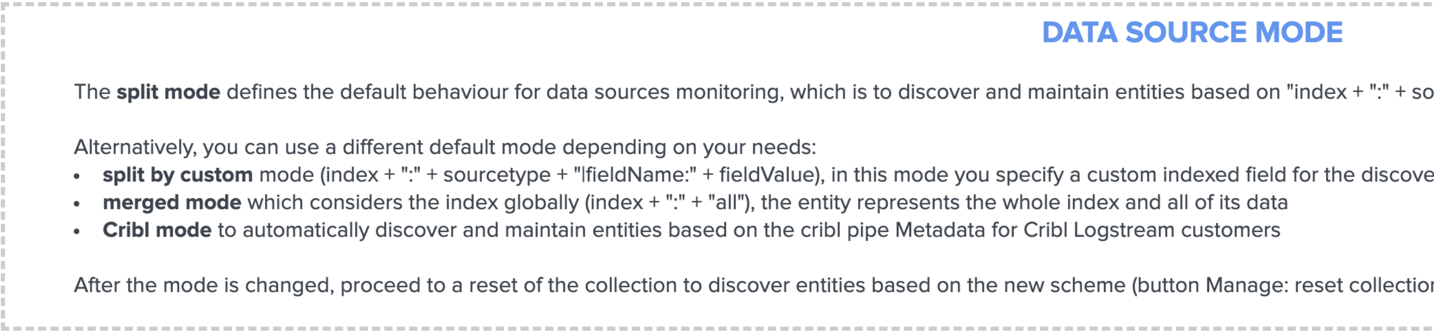
You can define the strategy while you are starting to use TrackMe, and gradually configure what TrackMe accesses to depending on your environment and requirements.

TrackMe Data Sources - Define what works for you

The primary concept of TrackMe is called **data sources**, See *Data Sources tracking and features* in the User guide for more explanations.

For the purposes of defining the best strategy that works for you, let's explain the different modes available, which you can configure via the `Trackme manage and configure` interface:

- Split mode (default)
- Split custom mode
- Merged mode
- Cribl mode



DATA SOURCE MODE

The **split mode** defines the default behaviour for data sources monitoring, which is to discover and maintain entities based on "index + ":" + so

Alternatively, you can use a different default mode depending on your needs:

- **split by custom** mode (index + ":" + sourcetype + "|fieldName:" + fieldValue), in this mode you specify a custom indexed field for the discove
- **merged mode** which considers the index globally (index + ":" + "all"), the entity represents the whole index and all of its data
- **Cribl mode** to automatically discover and maintain entities based on the cribl pipe Metadata for Cribl Logstream customers

After the mode is changed, proceed to a reset of the collection to discover entities based on the new scheme (button Manage: reset collectio

See *Your first steps with TrackMe* for more details in the *User guide* to start with tracking concepts

Trackme Data Sources - Split mode

The Split mode is the default mode that TrackMe uses, in this mode, the application discovers, classifies and creates entities based on:

```
index + ":" + sourcetype
```

Let's take the following simple example, we index Windows Events logs Application, System and Security WinEventLogs each WinEventLog in a specific index, we would endup with 3 entities, for instance:

- oswinsec:XmlWinEventLog
- oswinapp:XmlWinEventLog
- oswinsys:XmlWinEventLog

On the other hand, would we index these 3 WinEventLogs into a unique index, we would end up with 1 entity only, which covers (meaning TrackMe is looking at) all of the logs:

- oswin:XmlWinEventLog

Don't worry, TrackMe has plenty of features that allow you to cover any use cases (Elastic Sources, allow and block listing, etc), the Split mode is generally what covers most use cases, but this is very depending to your context.

Trackme Data Sources - split custom mode

The Split custom mode allows you to define an additional indexed field to be used when discovering and maintaining the data sources.

Once you define the indexed field, entities are going to be created as following:

```
index + ":" + sourcetype + "|<keyName>:<keyValue>
```

Where `keyName` is the name of the indexed field, `keyValue` the value.

Hint: Once enabled, any data source that does not include the indexed field will not be discovered any longer, you can handle any additional use cases as *Elastic sources* or create custom trackers in hybrid mode.

Trackme Data Sources - Merged mode

The Merged mode removes the concept of sourcetype and basically creates 1 entity per index, no matters what sourcetypes are indexed in it, entities are created as:

`index + ":all"`

This mode can potentially be interesting for you if you dedicate each index to a specific data flow, and you know by design that this is what you care about.

Trackme Data Sources - Cribl mode

If you are using Cribl, you can integrate TrackMe transparently and get benefits from the Cribl design very easily, in the Cribl mode, we create Data sources based on:

`index + ":" + sourcetype + "|cribl:" + cribl_pipe`

For a complete review of the Cribl mode, see *Cribl LogStream and TrackMe integration*

Finally, note that if you enable the Cribl mode, TrackMe will only discover automatically data sources coming via Cribl.

TrackMe Data Hosts - Define what works for you

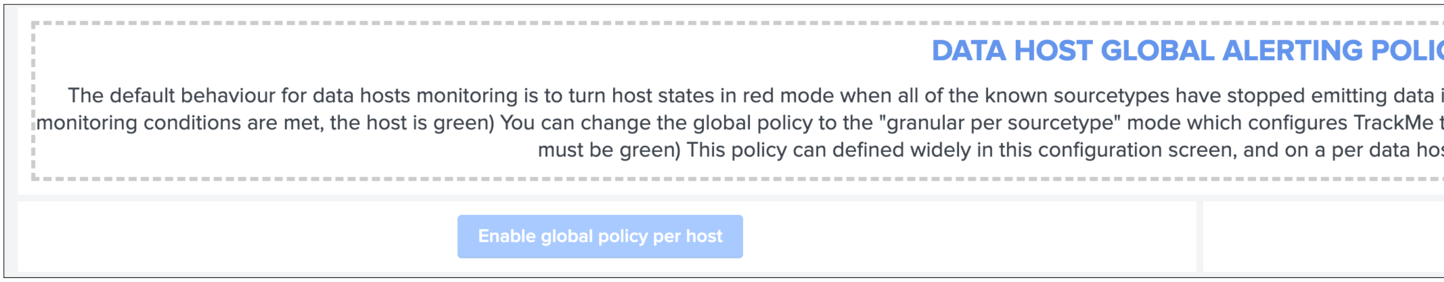
The second big concept in TrackMe is called `data hosts`, this basically means tracking the activity of host sending data to Splunk, from the `host Splunk Metadata` point of view.

There are two modes available, called `Data hosts global alerting policy`:

- `granular by host`: instructs TrackMe to consider turning a host red only if there are no more sourcetypes emitting data for that hosts according to the various configuration items
- `granular by sourcetype`: instructs TrackMe to consider each sourcetype individually by host, including their own max lagging rules, to determine if a host is having issues or not

Hint: This defines the global policy applied by default on all data hosts, this can be overridden on a per host basis if needed

Configuration of the global policy mode in the management UI:



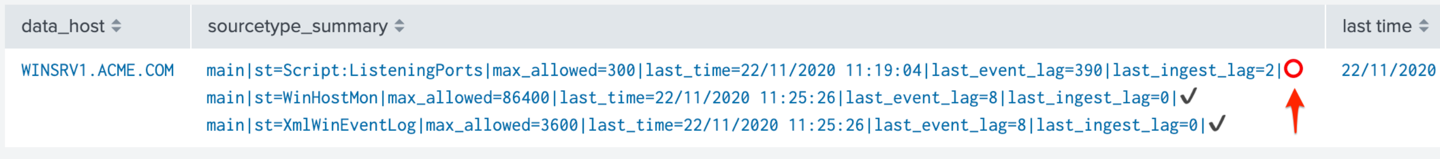
DATA HOST GLOBAL ALERTING POLIC

The default behaviour for data hosts monitoring is to turn host states in red mode when all of the known sourcetypes have stopped emitting data i
monitoring conditions are met, the host is green) You can change the global policy to the "granular per sourcetype" mode which configures TrackMe t
must be green) This policy can defined widely in this configuration screen, and on a per data hos

See *Alerting policy for data hosts* for more details in the *User guide* to start with data hosts tracking

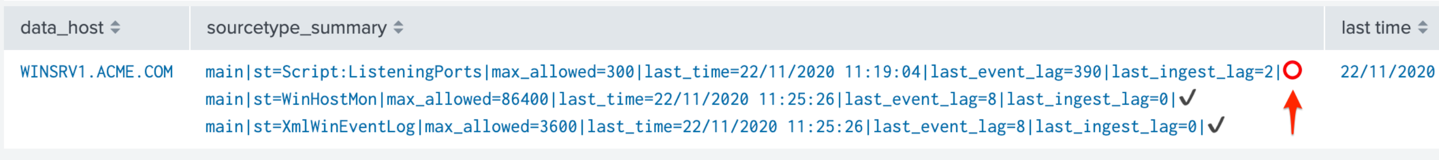
Behaviour examples:

Alerting policy track per sourcetype:



data_host	sourcetype_summary	last time
WINSRV1.ACME.COM	main|st=Script:ListeningPorts|max_allowed=300|last_time=22/11/2020 11:19:04|last_event_lag=390|last_ingest_lag=2|○ main|st=WinHostMon|max_allowed=86400|last_time=22/11/2020 11:25:26|last_event_lag=8|last_ingest_lag=0|✓ main|st=XmlWinEventLog|max_allowed=3600|last_time=22/11/2020 11:25:26|last_event_lag=8|last_ingest_lag=0|✓	22/11/2020

Alerting policy track per host:



data_host	sourcetype_summary	last time
WINSRV1.ACME.COM	main|st=Script:ListeningPorts|max_allowed=300|last_time=22/11/2020 11:19:04|last_event_lag=390|last_ingest_lag=2|○ main|st=WinHostMon|max_allowed=86400|last_time=22/11/2020 11:25:26|last_event_lag=8|last_ingest_lag=0|✓ main|st=XmlWinEventLog|max_allowed=3600|last_time=22/11/2020 11:25:26|last_event_lag=8|last_ingest_lag=0|✓	22/11/2020

Choosing which mode complies with your requirements all depends on how deep and how granular you need to be monitoring data hosts, many users will be happy with the default mode and would use the granular mode for specific entities, others will need to ensure to track hosts in a very detailed way, your choice!

TrackMe Metric Hosts - Define what works for you

The last big concept is called `metric hosts` tracking, this basically monitors all hosts (from the Splunk Metadata point of view) sending metrics to the metric store indexes.

There are specific configuration or mode to choose for metric hosts, your configuration will essentially be based on:

- Allow and Block listing to define which indexes and metric categories you want to track
- Defining threshold policies to configure what delay is acceptable or not on per metric category basis

See *Metric Hosts tracking and features* in the *User guide* to start with metric hosts tracking

2.2.3 Step 3: RBAC and access policies

Roles and permissions

TrackMe can be used by different populations of users, depending on the size of your Splunk implementation its maturity, essentially:

- Splunk administrators that responsible for the daily monitoring and maintenance of the Splunk deployment
- Ingestion teams responsible for that ingestion data flow from the providers to Splunk (could be the Splunk administrators, or not)
- Department teams that care about their own data sources and need to be able to understand what is available to them and the data source states
- Management
- maybe more!

From the application point of view, this essentially means two types of profiles:

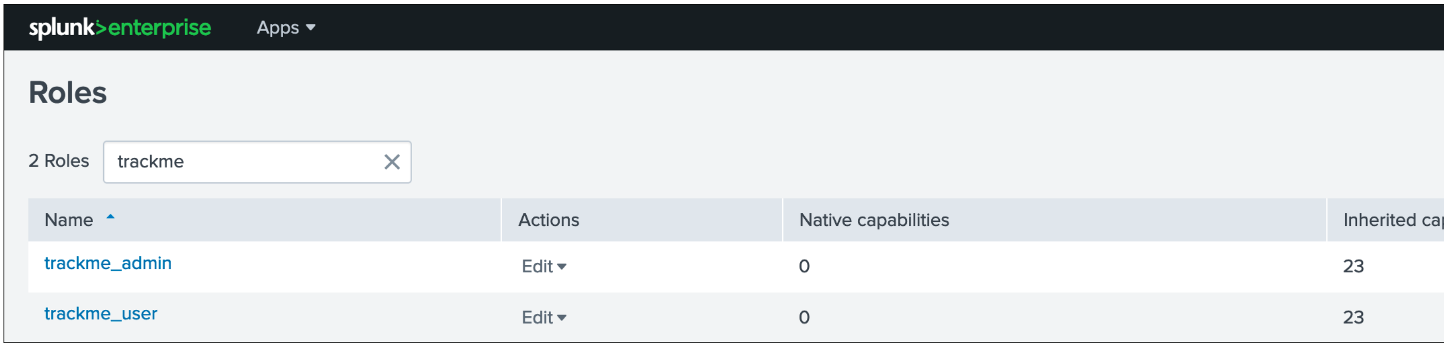
- `trackme admins` that can achieve modifications of what is tracked, and how
- `trackme users` that are looking at entities, without being allowed to perform changes

Fortunately, TrackMe handles this for you, and provides two types of roles you can use or import to properly define the level of permissions needed:

- `trackme_admin` role
- `trackme_user` role

These roles define write or read only permissions on the various objects TrackMe depends on, essentially stored in many KVstore collections.

Make sure to inherit, or make user member of these roles accordingly.



splunk>enterprise Apps ▾

Roles

2 Roles trackme

Name ▴	Actions	Native capabilities	Inherited cap
trackme_admin	Edit ▾	0	23
trackme_user	Edit ▾	0	23

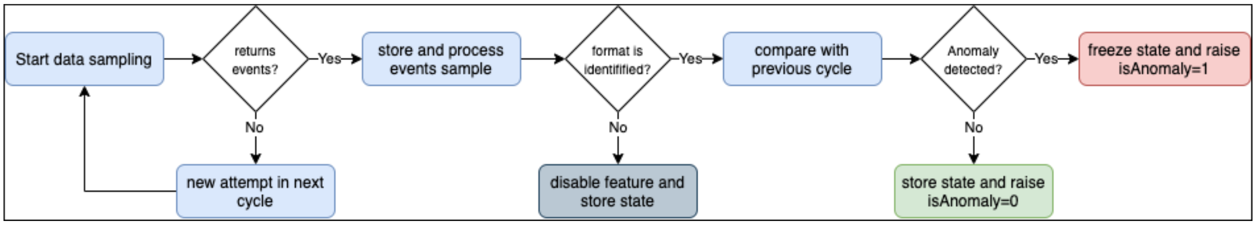
Tip: capabilities for trackme_admin:

- the capability `list_settings` is required for trackme admins that are not privileged users, to be able to run actions doing updates via the TrackMe rest endpoints

Data privacy

While TrackMe's job is monitoring data, it does generate its own data as well, and especially it is tracking and performing data quality assessments in the scope of a very powerful feature called *Data sampling and event formats recognition*.

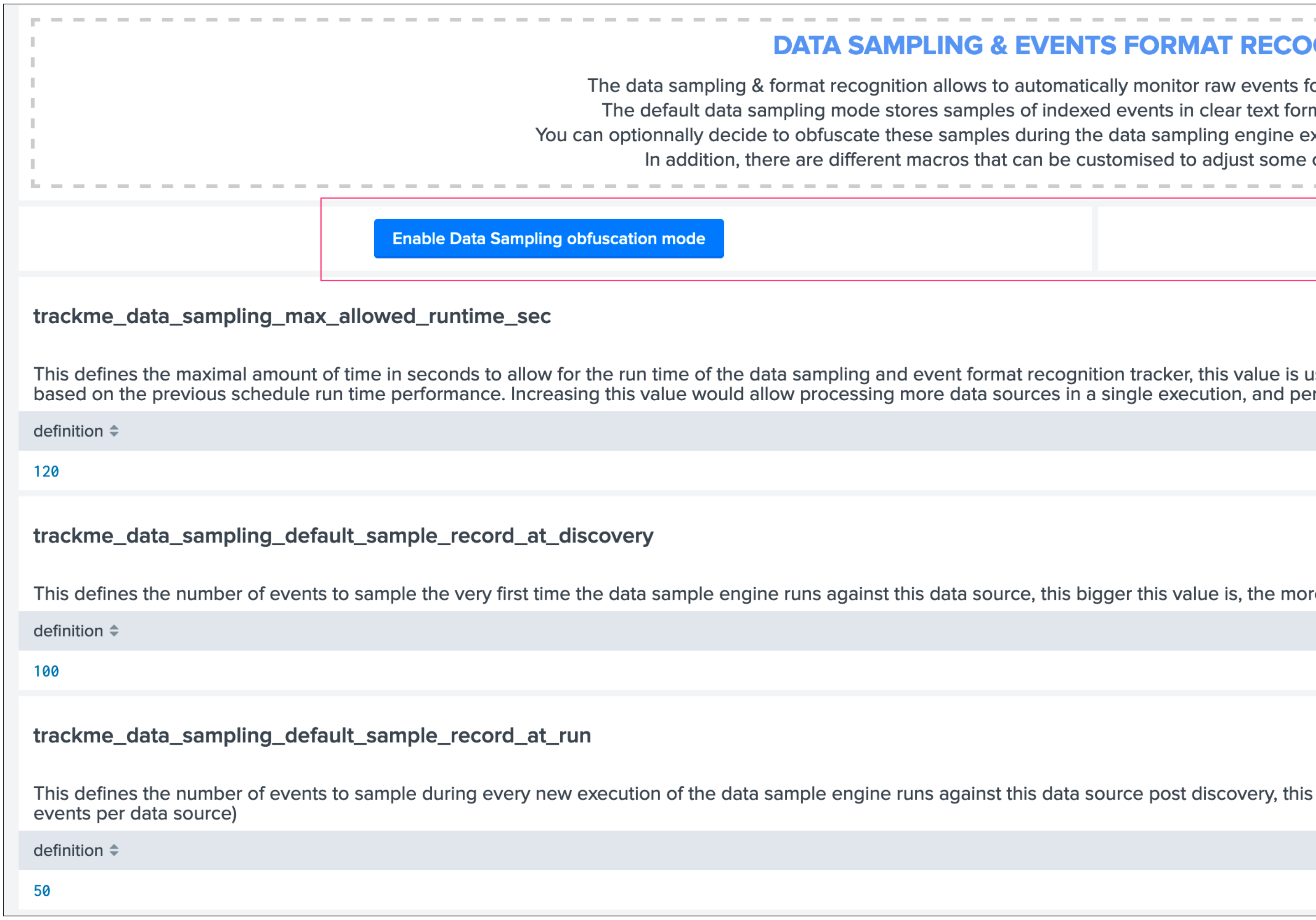
This results in samples of real events being stored in a dedicated KVstore collection `trackme_data_sampling`, managed via the data sampling workflow:


Start data sampling
returns events?
Yes
No
new attempt in next cycle
store and process events sample
format is identitified?
Yes
No
disable feature and store state
compare with previous cycle
Anomaly detected?
Yes
No
store state and raise isAnomaly=0
freeze state and raise isAnomaly=1

By default, the `trackme_data_sampling` is only available in read mode to users member of the `trackme_user` and `trackme_admin` roles, bellow is the default.meta stanzas:

```
[transforms/trackme_data_sampling]
access = read : [ admin, trackme_admin, trackme_user ], write : [ admin, trackme_
↪admin ]

[collections/kv_trackme_data_sampling]
access = read : [ admin, trackme_admin, trackme_user ], write : [ admin, trackme_
↪admin ]
```

If you are concerned about this activity, if for some reasons TrackMe users (and even admins) are not supposed to be able to see samples of real events that TrackMe is looking at, you can enable the *Data Sampling obfuscation mode*:



DATA SAMPLING & EVENTS FORMAT RECO

The data sampling & format recognition allows to automatically monitor raw events fo
The default data sampling mode stores samples of indexed events in clear text form
You can optionnally decide to obfuscate these samples during the data sampling engine ex
In addition, there are different macros that can be customised to adjust some o

trackme_data_sampling_max_allowed_runtime_sec

This defines the maximal amount of time in seconds to allow for the run time of the data sampling and event format recognition tracker, this value is u
based on the previous schedule run time performance. Increasing this value would allow processing more data sources in a single execution, and per

definition ⇕
120

trackme_data_sampling_default_sample_record_at_discovery

This defines the number of events to sample the very first time the data sample engine runs against this data source, this bigger this value is, the more

definition ⇕
100

trackme_data_sampling_default_sample_record_at_run

This defines the number of events to sample during every new execution of the data sample engine runs against this data source post discovery, this
events per data source)

definition ⇕
50

- In the default mode, that is `Disable Data Sampling obfuscation mode`, events that are sampled are stored in the data sampling KVstore collection and can be used to review the results from the latest sampling operation
- In the `Enable Data Sampling obfuscation mode`, events are not stored anymore and replaced by an admin message, the sampling processing still happens the same way but events cannot be reviewed anymore using the latest sample traces
- In such a case, when then obfuscation mode is enabled, users will need to either run the rules manually to locate the messages that were captured to the conditions being met (bad format, PII data, etc) or use the Smart *Smart Status* feature to have TrackMe run this operation on demand

2.2.4 Step 4: Indexers macro definition

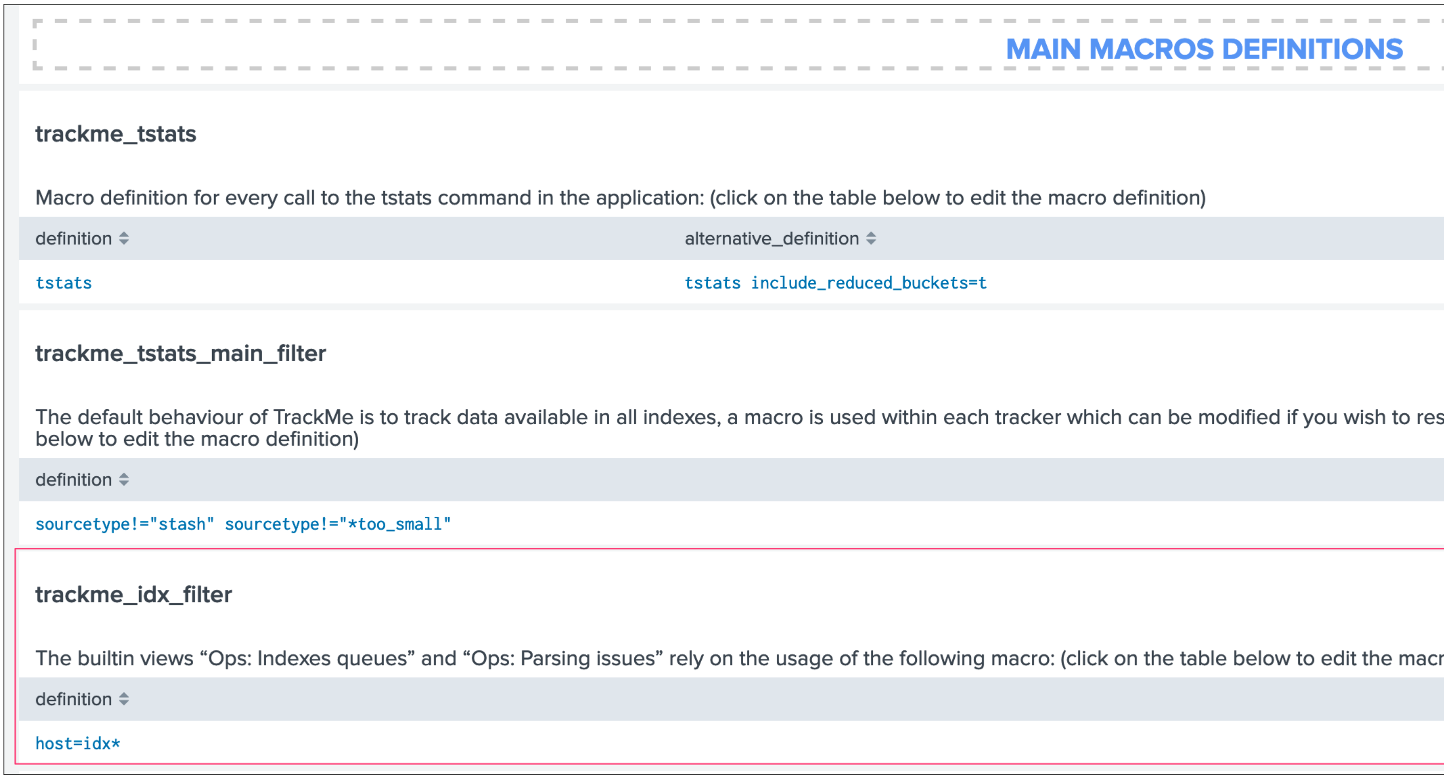
TrackMe provides different views that are related to the Splunk pipelines and queues, such as:

- `Ops: Queues Center`
- `Ops: Parsing Issues`
- In entities tab `Data Parsing Quality`

All searches underneath rely on the definition of a macro:

```
# defined pattern filter for indexers
[trackme_idx_filter]
definition = host=idx*
iseval = 0
```

In TrackMe manage and configure:



MAIN MACROS DEFINITIONS

trackme_tstats

Macro definition for every call to the tstats command in the application: (click on the table below to edit the macro definition)

definition	alternative_definition
tstats	tstats include_reduced_buckets=t

trackme_tstats_main_filter

The default behaviour of TrackMe is to track data available in all indexes, a macro is used within each tracker which can be modified if you wish to res
below to edit the macro definition)

definition
sourcetype!="stash" sourcetype!="*too_small"

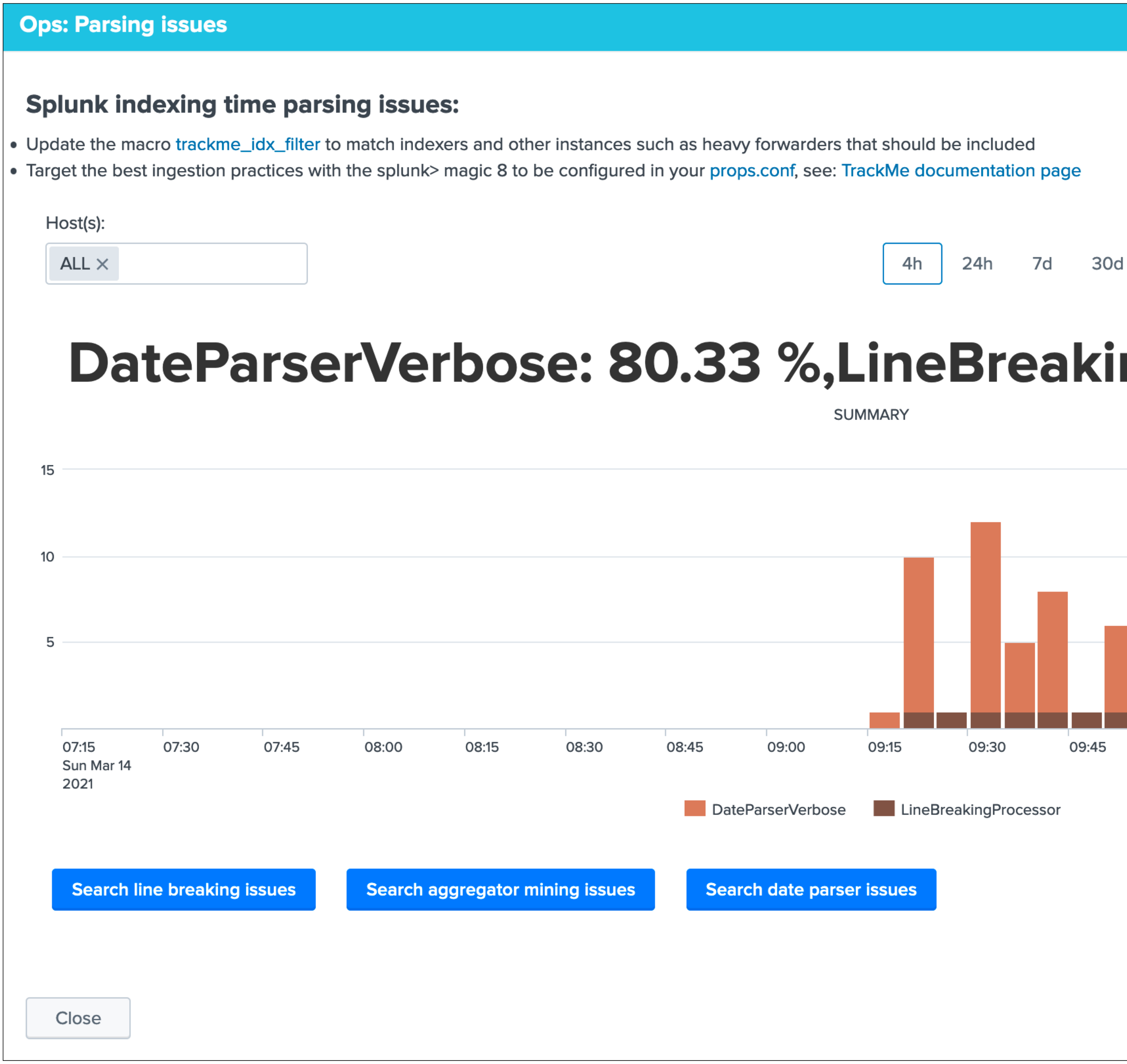
trackme_idx_filter

The builtin views "Ops: Indexes queues" and "Ops: Parsing issues" rely on the usage of the following macro: (click on the table below to edit the macr

definition
host=idx*

Make sure to update this definition accordingly to match your indexers and potentially Heavy Forwarders naming convention.

view example:



Ops: Parsing issues

Splunk indexing time parsing issues:

- Update the macro trackme_idx_filter to match indexers and other instances such as heavy forwarders that should be included
- Target the best ingestion practices with the splunk> magic 8 to be configured in your props.conf, see: TrackMe documentation page

Host(s): ALL ×

DateParserVerbose: 80.33 %,LineBreaki

SUMMARY

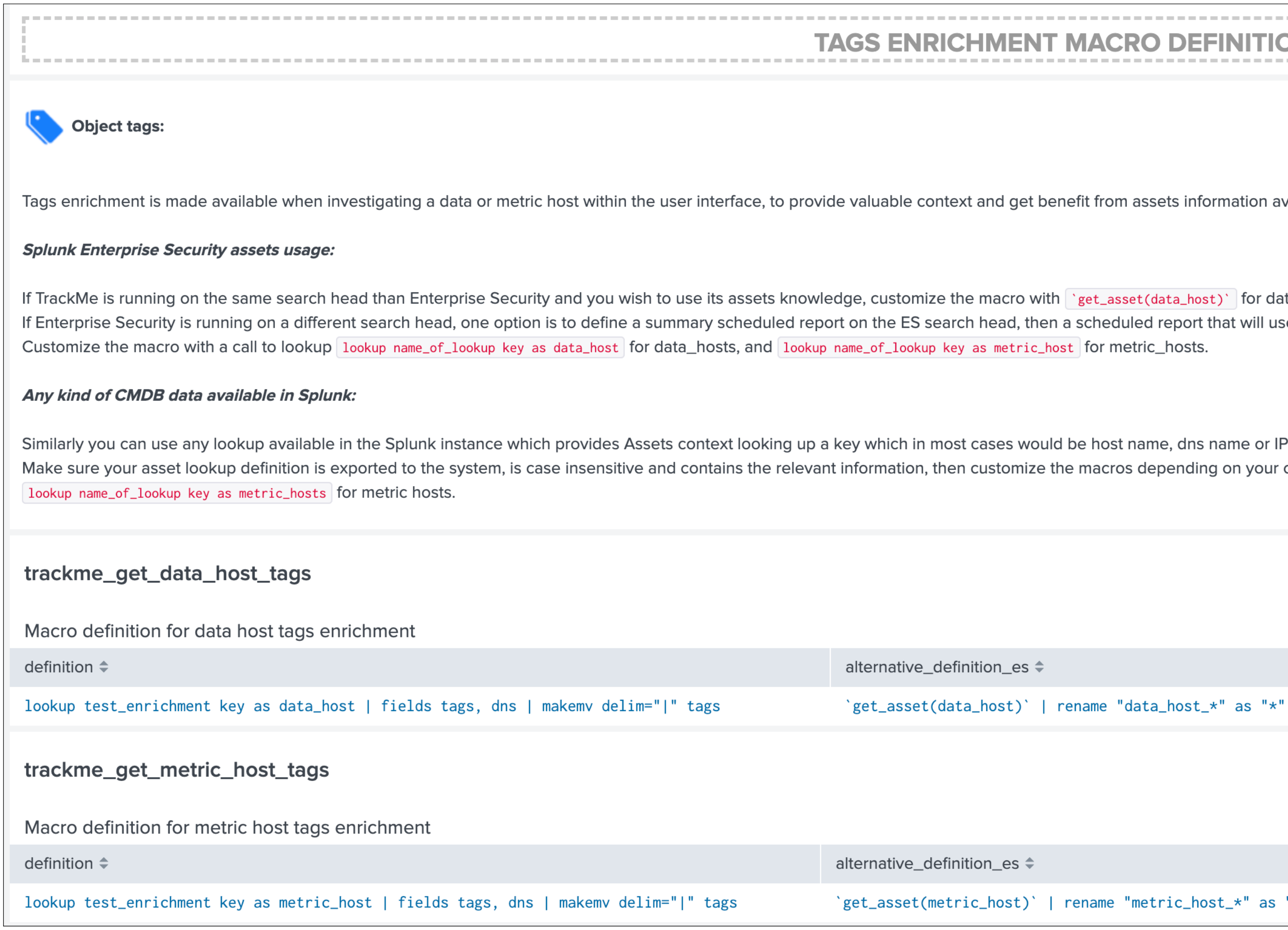
2.2.5 Step 5: host tags enrichment

OPTIONAL: tags enrichment for data and metric hosts

This step is optional and depends on your context:

Tags enrichment feature

Tags enrichment is made available when investigating a data or metric host within the user interface, to provide valuable context and get benefit from assets information available in the Splunk deployment.



TAGS ENRICHMENT MACRO DEFINITIC

Object tags:

Tags enrichment is made available when investigating a data or metric host within the user interface, to provide valuable context and get benefit from assets information av

Splunk Enterprise Security assets usage:

If TrackMe is running on the same search head than Enterprise Security and you wish to use its assets knowledge, customize the macro with `` `get_asset(data_host)` `` for dat
If Enterprise Security is running on a different search head, one option is to define a summary scheduled report on the ES search head, then a scheduled report that will us
Customize the macro with a call to lookup `lookup name_of_lookup key as data_host` for data_hosts, and `lookup name_of_lookup key as metric_host` for metric_hosts.

Any kind of CMDB data available in Splunk:

Similarly you can use any lookup available in the Splunk instance which provides Assets context looking up a key which in most cases would be host name, dns name or IP
Make sure your asset lookup definition is exported to the system, is case insensitive and contains the relevant information, then customize the macros depending on your c
`lookup name_of_lookup key as metric_hosts` for metric hosts.

trackme_get_data_host_tags

Macro definition for data host tags enrichment

definition ⇕	alternative_definition_es ⇕
`lookup test_enrichment key as data_host | fields tags, dns | makemv delim="|" tags`	`` `get_asset(data_host)` | rename "data_host_*" as "*" ``

trackme_get_metric_host_tags

Macro definition for metric host tags enrichment

definition ⇕	alternative_definition_es ⇕
`lookup test_enrichment key as metric_host | fields tags, dns | makemv delim="|" tags`	`` `get_asset(metric_host)` | rename "metric_host_*" as ' ``

Splunk Enterprise Security assets usage:

If TrackMe is running on the same search head than Enterprise Security and you wish to use its assets knowledge, customize the macro with `` `get_asset(data_host)` `` for data hosts, and `` `get_asset(metric_host)` `` for metric hosts.

If Enterprise Security is running on a different search head, one option is to define a summary scheduled report on the ES search head, then a scheduled report that will use the summary data to automatically build a copy of Enterprise Security assets lookup. (asset_lookup_by_str) Customize the macro with a call to `lookup lookup name_of_lookup key as data_host` for data_hosts, and `lookup name_of_lookup key as metric_host` for metric_hosts.

Any kind of CMDB data available in Splunk:

Similarly, you can use any lookup available in the Splunk instance which provides Assets context looking up a key which in most cases would be host name, dns name or IP address.

Make sure your asset lookup definition is exported to the system, is case insensitive and contains the relevant information, then customize the macros depending on your configuration, example: `lookup name_of_lookup key as data_hosts` for data hosts, `lookup name_of_lookup key as metric_hosts` for metric hosts.

2.2.6 Step 6: entities priority management

OTIONAL: third party priority definition

When TrackMe discovers a new entity, a level of priority is defined by default:

- by default, entities are added as `medium` priority
- this is controlled via the macro `trackme_default_priority`
- TrackMe accepts 3 levels of priorities: `low` / `medium` / `high`
- The UIs will threat differently `high` priority entities to highlight top critical issues in the Splunk environments

See *Priority management* in the *User guide* for more details.

Hint: How TrackMe manages the priority value

- Once a priority is defined for an entity in its collection, this value is always preserved upon iterations of TrackMe jobs or update operations
- If a collection is reset by an admin, the priority value that was assigned is lost and will be replaced by the system affected priority value

---

TrackMe does not provide third party integration to define the priority, especially because this would be very likely highly depending on every single user context..

However, because TrackMe relies on KVstore based lookups, it is very straightforward to create your own workflow to enrich and define the entities priority level from any other data you have in Splunk such as a CMDB lookup or Enterprise Security Assets.

For example, you could define the following scheduled report that updates the priority based on third party enrichment:

```
| inputlookup trackme_host_monitoring | eval key=_key
| lookup asset_lookup_by_str key as data_host OUTPUT priority as es_priority
| eval priority=case(
  isnull(es_priority) OR es_priority="", priority,
  es_priority="low", es_priority,
  es_priority="medium", es_priority,
  es_priority="high" OR es_priority="critical", es_priority
)
| fields - es_priority
| outputlookup trackme_host_monitoring append=t key_field=key
| stats c
```

Such a report would be scheduled to run daily or so, and would automatically maintain the priority definition based on an external integration.

2.2.7 Step 7: enabling out of the box alerts or create your own custom alerts

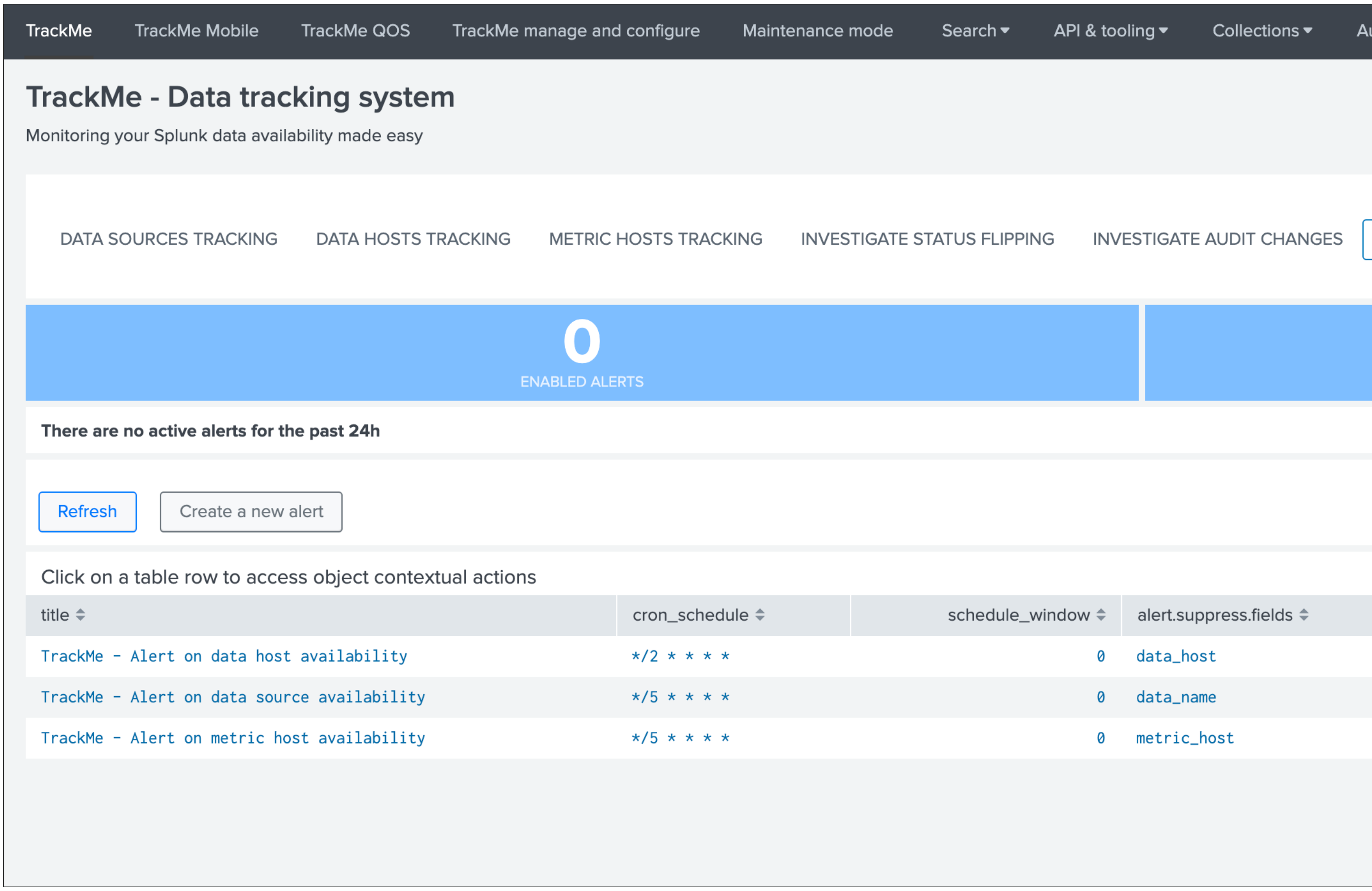
Since TrackMe 1.2.39, a dedicated screen allows to manage alerts within TrackMe, and create your own alert in assisted mode:

Using out of the box alerts

TrackMe provides out of the box alerts that can be used to deliver alerting when a monitored component reaches a red state:

- TrackMe - Alert on data source availability
- TrackMe - Alert on data host availability
- TrackMe - Alert on metric host availability

In TrackMe main screen, go to the tracking alerts tab:



TrackMe - Data tracking system

Monitoring your Splunk data availability made easy

There are no active alerts for the past 24h

Click on a table row to access object contextual actions

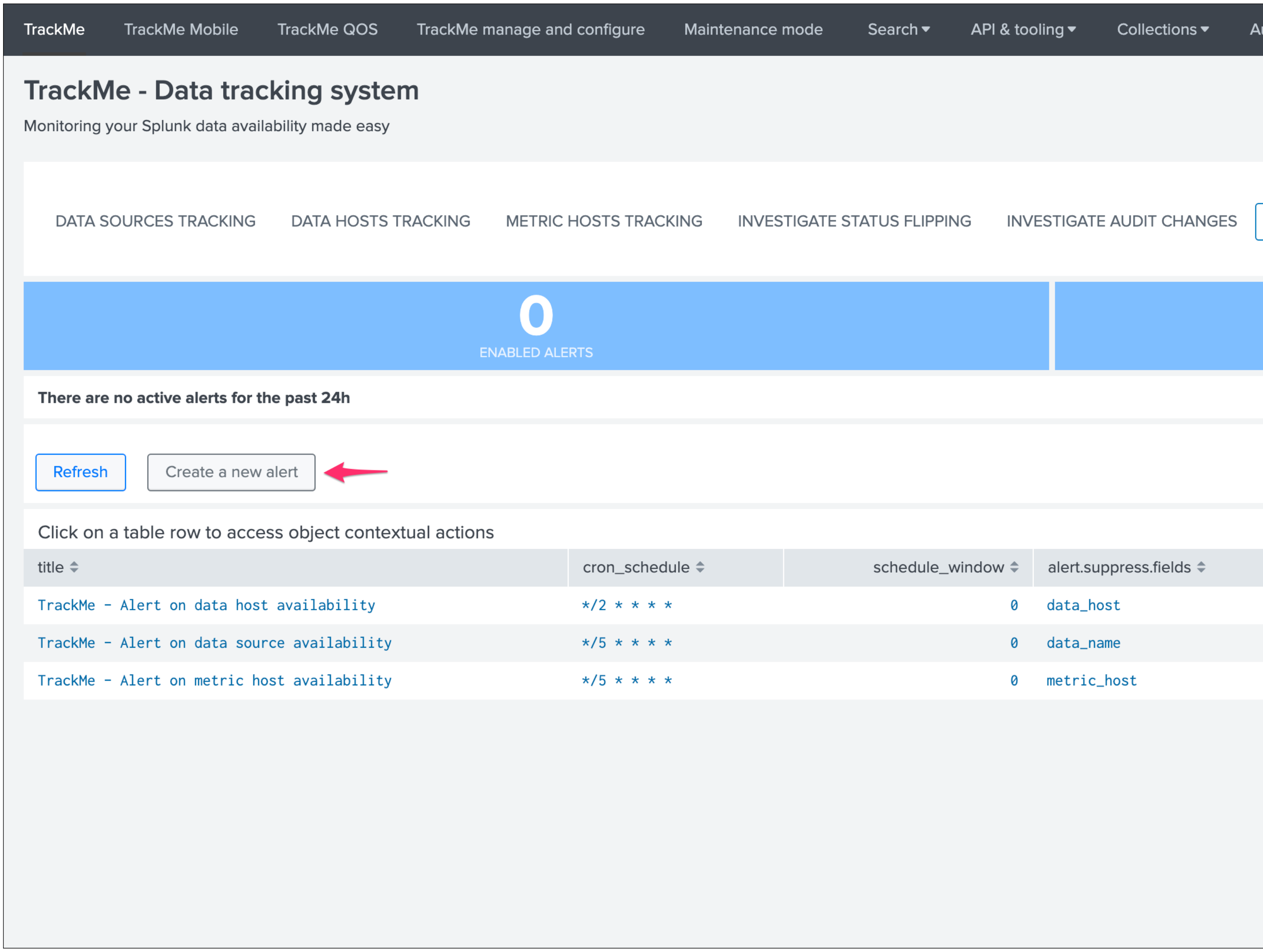
title	cron_schedule	schedule_window	alert.suppress.fields
TrackMe - Alert on data host availability	*/2 * * * *	0	data_host
TrackMe - Alert on data source availability	*/5 * * * *	0	data_name
TrackMe - Alert on metric host availability	*/5 * * * *	0	metric_host

Hint: Out of the box alerts

- Out of the box alerts are disabled by default, you need to enable alerts to start using them
- Alerts will trigger by default on `high priority` entities only, this is controlled via the macro definition `trackme_alerts_priority`
- Edit the alert to perform your third party integration, for example `sending emails` or creating `JIRA issues` based on Splunk alert actions capabilities
- Out of the box alert enable by default two TrackMe alert actions, `automatic acknowledgement` and the `Smart Status` alert actions
- The results of the `Smart Status` alert action are automatically indexed in the TrackMe summary index within the sourcetype `trackme_smart_status` and can be used for investigation purposes

Creating custom alerts in assisted mode

You can use this interface to a create one or more custom alerts:



TrackMe - Data tracking system

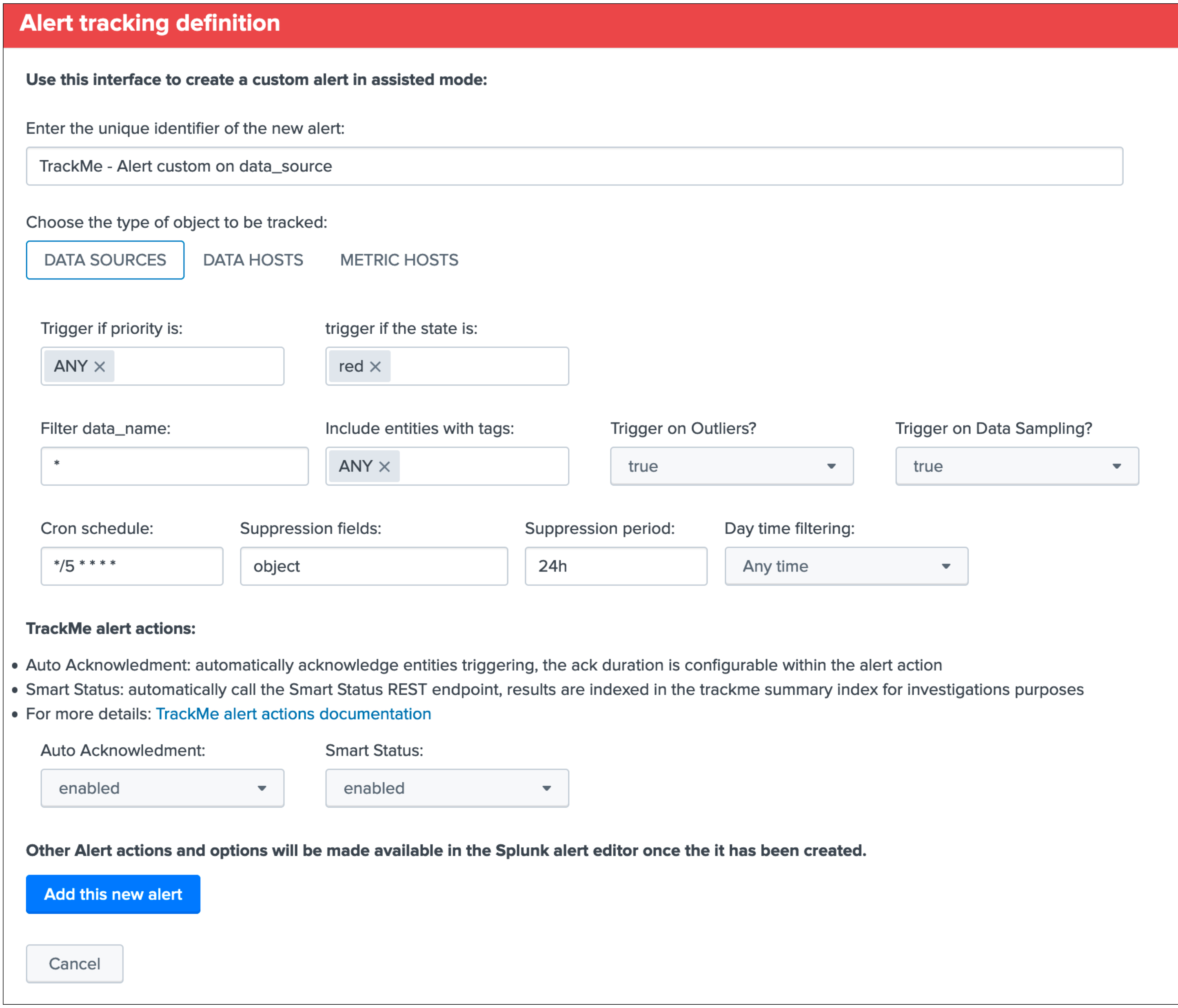
Monitoring your Splunk data availability made easy

There are no active alerts for the past 24h

Click on a table row to access object contextual actions

title	cron_schedule	schedule_window	alert.suppress.fields
TrackMe - Alert on data host availability	*/2 * * * *	0	data_host
TrackMe - Alert on data source availability	*/5 * * * *	0	data_name
TrackMe - Alert on metric host availability	*/5 * * * *	0	metric_host

This opens the assistant where you can choose between different builtin options depending on the type of entities to be monitoring:



Alert tracking definition

Use this interface to create a custom alert in assisted mode:

Enter the unique identifier of the new alert: TrackMe - Alert custom on data_source

Choose the type of object to be tracked: DATA SOURCES | DATA HOSTS | METRIC HOSTS

Trigger if priority is: ANY ×

trigger if the state is: red ×

Filter data_name: *

Include entities with tags: ANY ×

Trigger on Outliers? true

Trigger on Data Sampling? true

Cron schedule: */5 * * * *

Suppression fields: object

Suppression period: 24h

Day time filtering: Any time

TrackMe alert actions:

- Auto Acknowledment: automatically acknowledge entities triggering, the ack duration is configurable within the alert action
- Smart Status: automatically call the Smart Status REST endpoint, results are indexed in the trackme summary index for investigations purposes
- For more details: TrackMe alert actions documentation

Auto Acknowledment: enabled

Smart Status: enabled

Other Alert actions and options will be made available in the Splunk alert editor once the it has been created.

Once you have created a new alert, it will be immediately visible in the tracking alerts UI, and you can use the Splunk built alert editor to modify the alert to up to your needs such as enabling third party actions, emails actions and so forth.

Hint: Custom alert features

- Creating custom alerts provide several layers of flexibility depending on your choices and preferences
- You may for example have alerts handling lowest level of prority with a specific type of alert action, and have a specific alert for highly critical entities
- Advanced setup can easily be performed such as getting benefits from the tags features and multiple alerts using tag policies to associate data sources and different types of alerts, recipients, actions. . .
- You may decide if you wish to enable or disable the TrackMe `auto acknowledgement` and

`Smart Status` alert actions while creating alerts through the assistant

2.2.8 Final: Read the docs and start using TrackMe

TrackMe is a large, powerful and rich in features Splunk application that goes way beyond these initial configuration steps, there are many more features to discover and handle.

When you start the integration of TrackMe especially in large environments, it is generally a good approach to:

- Focus progressively on highly valuable pieces of data, such as data sources used to feed the SOC use cases, the NOC alerts, etc
- Use the priority level and tag policies to qualify and immediately get incredible value from TrackMe
- Use policies for lagging definition rather per entity definition (then you can reset collections if you need!)
- Use tag policies to identify and define data context for even better filtering and value
- Use Identity cards to provide context for TrackMe admins and users, and document or refer to your very own documentations

Reviewing these simple steps should put you on track easily, continue with reading the *User guide* for a full coverage!

CHAPTER 3

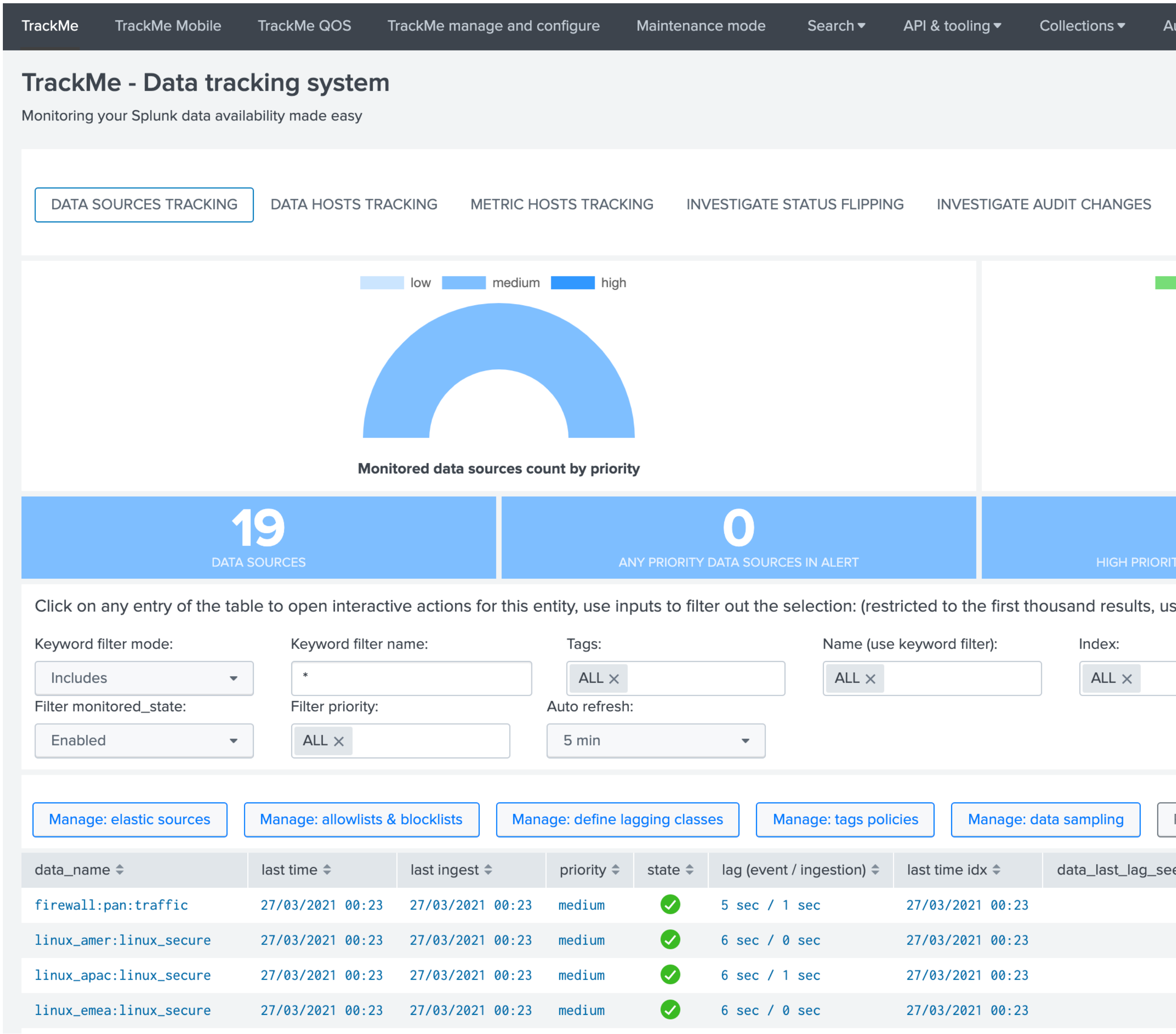
User guide:

3.1 User guide

3.1.1 Your first steps with TrackMe

Access TrackMe main interface

When you open the application, you access by default to the main TrackMe UI and especially to the data sources tracking tab, if the tracker reports have already been executed at least once, the application will expose the data that was discovered in your environment:



TrackMe - Data tracking system

Monitoring your Splunk data availability made easy

Click on any entry of the table to open interactive actions for this entity, use inputs to filter out the selection: (restricted to the first thousand results, us

Keyword filter mode: Includes

Keyword filter name: *

Tags: ALL

Name (use keyword filter): ALL

Index: ALL

Filter monitored_state: Enabled

Filter priority: ALL

Auto refresh: 5 min

data_name	last time	last ingest	priority	state	lag (event / ingestion)	last time idx	data_last_lag_see
firewall:pan:traffic	27/03/2021 00:23	27/03/2021 00:23	medium	✓	5 sec / 1 sec	27/03/2021 00:23	
linux_amer:linux_secure	27/03/2021 00:23	27/03/2021 00:23	medium	✓	6 sec / 0 sec	27/03/2021 00:23	
linux_apac:linux_secure	27/03/2021 00:23	27/03/2021 00:23	medium	✓	6 sec / 1 sec	27/03/2021 00:23	
linux_emea:linux_secure	27/03/2021 00:23	27/03/2021 00:23	medium	✓	6 sec / 0 sec	27/03/2021 00:23	

Tip: If the UI is empty and no data sources are showing up:

- You can wait for the short term trackers execution which are scheduled to run every 5 minutes
- Or manually run the data sources tracker by clicking on the button "Run: short term tracker now" (we will come back to the tracker notion later in this guide)

Main navigation tabs

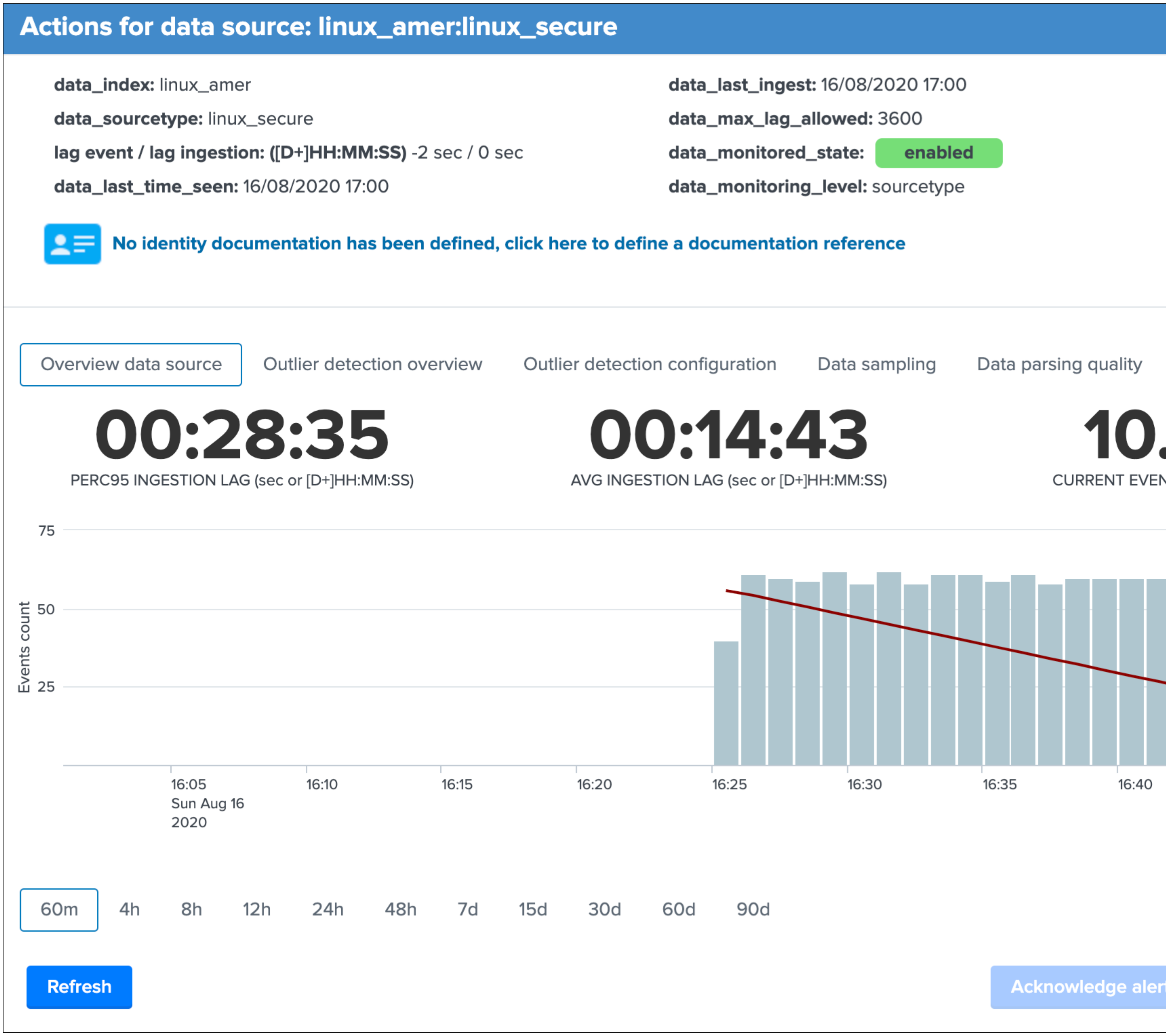
Now that TrackMe is deployed, and it discovered data available in your environment, let's review the main tabs provided in the UI:

- `DATA SOURCES TRACKING` shows the tracking of data sources, by default a data source is a breakdown of your data on a per `index + ":" + sourcetype`
- `DATA HOSTS TRACKING` shows data discovered for each `host sending events` to Splunk
- `METRIC HOSTS TRACKING` shows metrics discovered for each `host sending metrics` to Splunk
- `INVESTIGATE STATUS FLIPPING` shows the detection of an entity switching from a state, example green, to another state like red
- `INVESITAGE AUDIT CHANGES` shows all changes performed within the UI for auditing and review purposes
- `TRACKING ALERTS` shows alerts activity, allows managing and creating new custom alerts adapated to TrackMe context

Data Sources tracking and features

Data Source main screen

Let's click on any entry in the table:



Actions for data source: linux_amer:linux_secure

data_index: linux_amer

data_sourcetype: linux_secure

lag event / lag ingestion: ([D+]HH:MM:SS) -2 sec / 0 sec

data_last_time_seen: 16/08/2020 17:00

data_last_ingest: 16/08/2020 17:00

data_max_lag_allowed: 3600

data_monitored_state: enabled

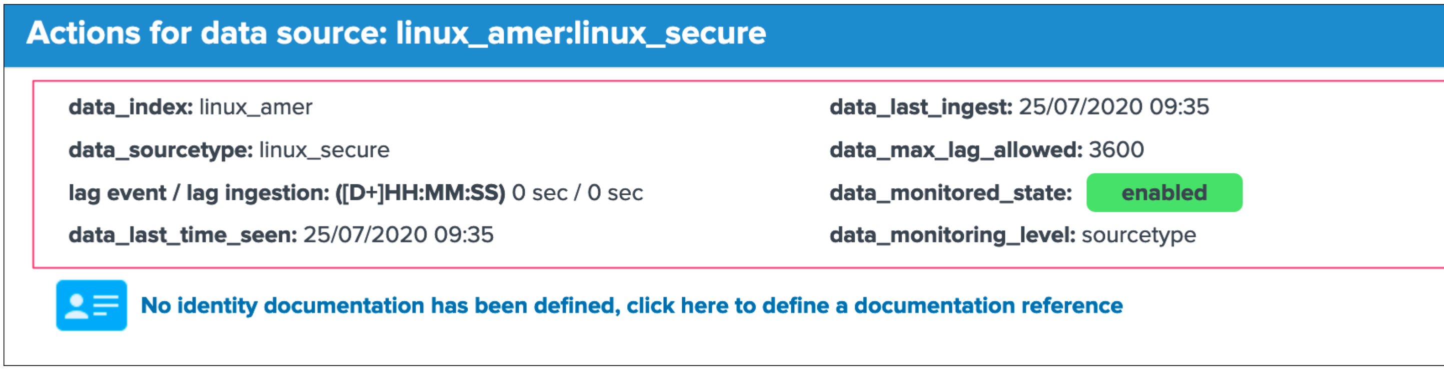
data_monitoring_level: sourcetype

No identity documentation has been defined, click here to define a documentation reference

> **Warning:** If you do not see the full window (called modal window), review your screen resolution settings, TrackMe requires a minimal high enough resolution when navigating through the app*

The modal window "open-up" is the user main interaction with TrackMe, depending on the context different information, charts, calculations and options are provided.

In the context of the data sources tracking, let's have a deeper look at top part of the window:



Actions for data source: linux_amer:linux_secure

data_index: linux_amer

data_sourcetype: linux_secure

lag event / lag ingestion: ([D+]HH:MM:SS) 0 sec / 0 sec

data_last_time_seen: 25/07/2020 09:35

data_last_ingest: 25/07/2020 09:35

data_max_lag_allowed: 3600

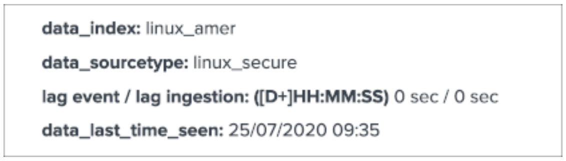
data_monitored_state: enabled

data_monitoring_level: sourcetype

No identity documentation has been defined, click here to define a documentation reference

Let's review these information:

group 1 left screen



data_index: linux_amer

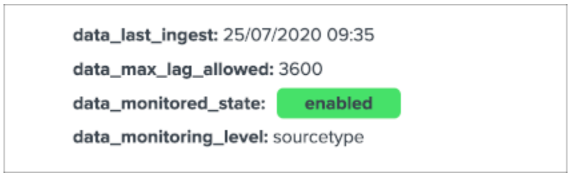
data_sourcetype: linux_secure

lag event / lag ingestion: ([D+]HH:MM:SS) 0 sec / 0 sec

data_last_time_seen: 25/07/2020 09:35

- `data_index` is the name of the Splunk index where the data resides
- `data_sourcetype` is the Splunk sourcetype for this entity
- `lag event / lag ingestion: ([D+]HH:MM:SS)` exposes the two main lagging metrics handled by TrackMe, the lag from the event point of view, and the lag from the ingestion point of view, we will come back to that very soon
- `data_last_time_seen` is the last date time TrackMe has detected data available for this data source, from the event time stamp point of view

group 2 middle screen



data_last_ingest: 25/07/2020 09:35

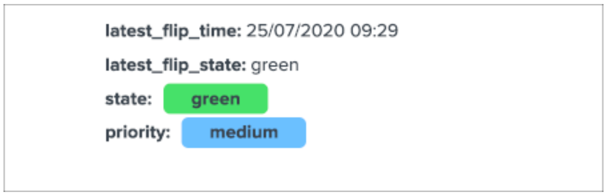
data_max_lag_allowed: 3600

data_monitored_state: enabled

data_monitoring_level: sourcetype

- `data_last_ingest` is the last date time TrackMe has detected data ingested by Splunk for the data source, this can differ from the very last event available in the data source (more after)
- `data_max_lag_allowed` is the value in seconds that TrackMe will use as the main information to define the status of the data source, by default it is defined to 1 hour (3600 seconds)
- `data_monitored_state` is a flag which tells TrackMe that this data source should be actively monitored, this is "enabled" by default and be defined within the UI to "disabled" (the red "Disable" button in the entity window)
- `data_monitoring_level` is a flag which tells TrackMe how to take into account other sourcetypes available in that same index when defining the current status of the entity

group 3 right screen



latest_flip_time: 25/07/2020 09:29

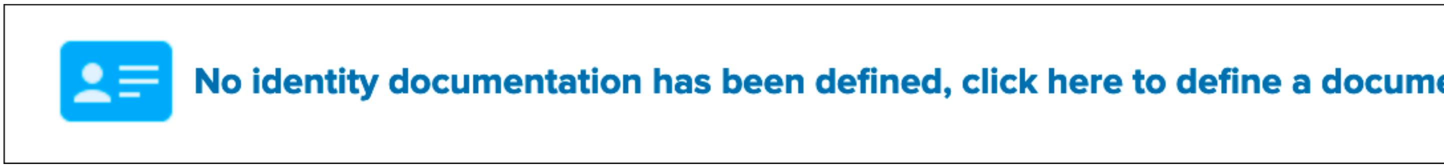
latest_flip_state: green

state: green

priority: medium

- `latest_flip_time` is the latest date time a change was detected in the state of the entity
- `latest_flip_states` is the state to which it moved at that time
- `state` is the current state, there are different states: green / orange / blue / grey / red (more explanations to come)
- `priority` represents the priority of the entity, by default all entities are added as "medium", priority is used in different parts of the app and alerts, there are 3 level of priority: low / medium / high

group 4 bottom



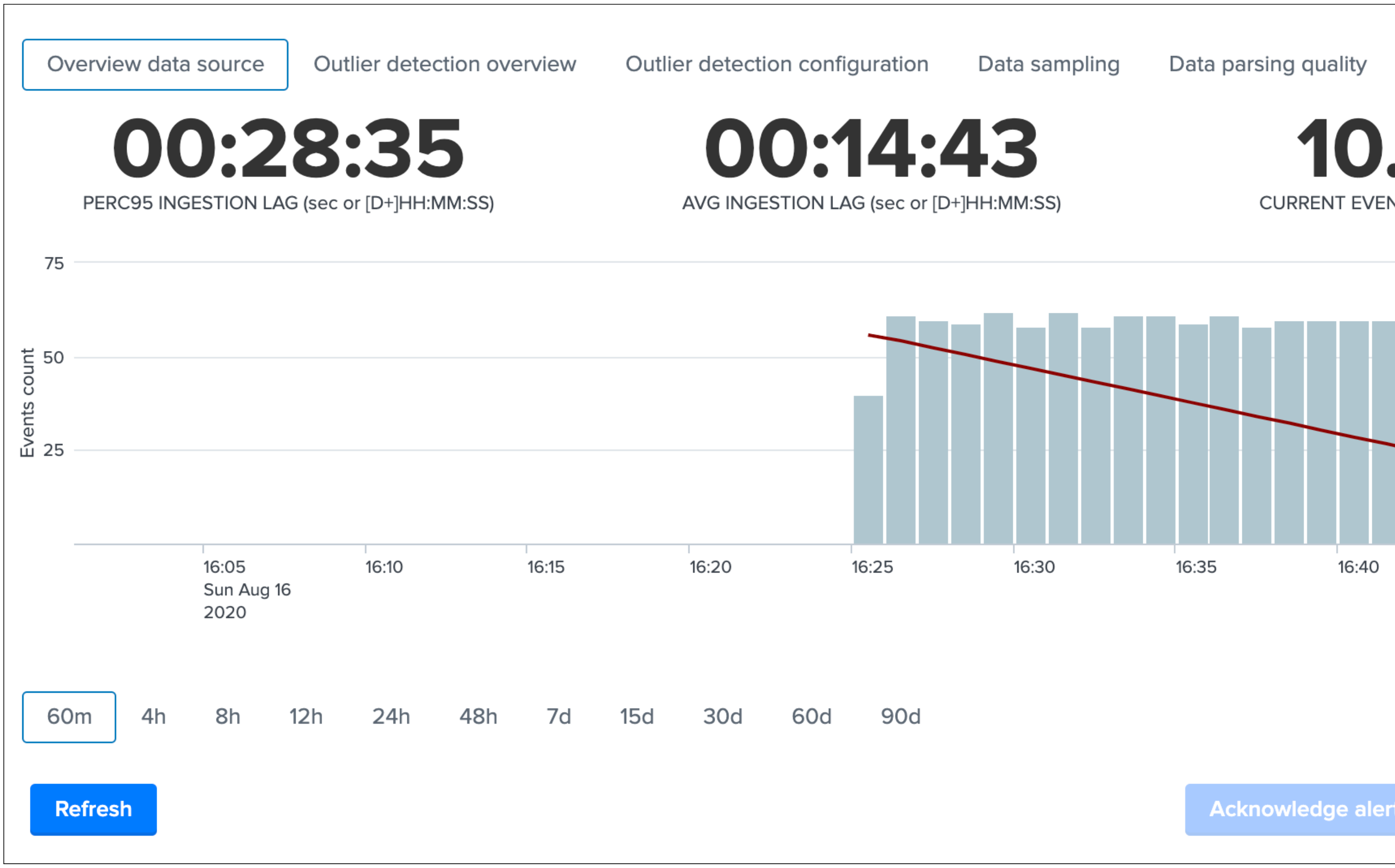
No identity documentation has been defined, click here to define a docume

- `Identity documentation card` is a feature that allows you create an information card (hyperlink and a text note), and link that card to any number of data sources.
- By default, no identity card is defined which is exposed by this message, if an identity card is created and linked to the entity, the message will turn into a link that once clicked exposes in a new window the context of the card
- Use this feature to quickly reference the main information for someone accessing to TrackMe and when there is an issue on the data source, which would provide a link to whatever you want (your Confluence, etc) and a quick help text. (at least a hyperlink or a text note must be defined)

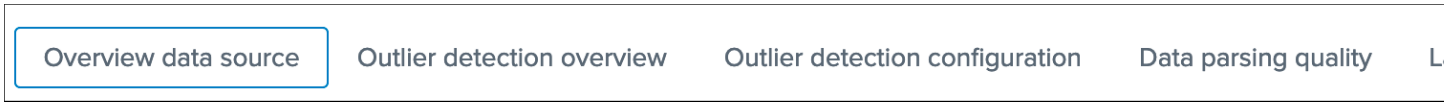
See *Data identity card* for more details about the feature.

Data source screen tabs

Let's have a look now at next part of the modal window:

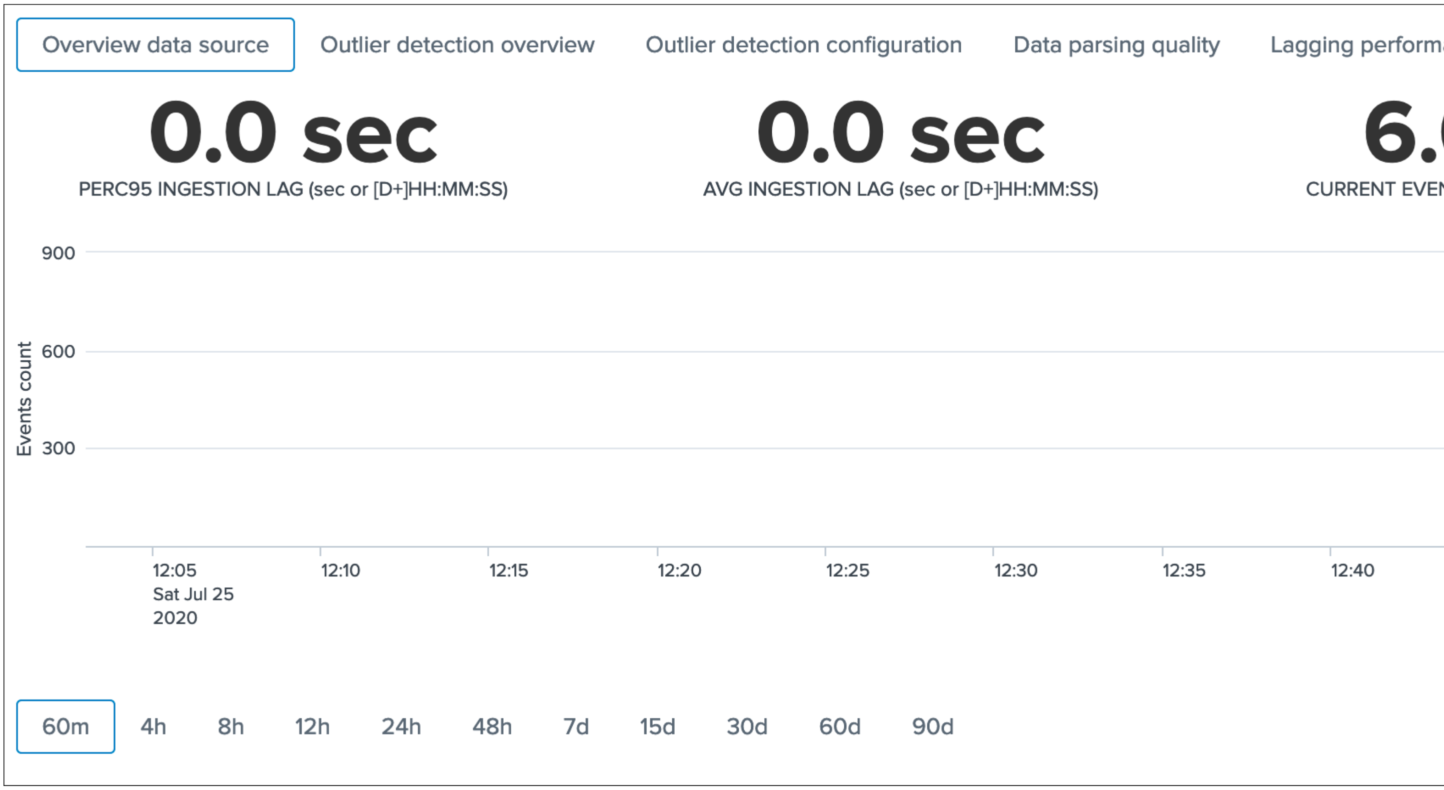


Starting by describing the tabs available in this window:



- `Overview data source` is the current view that exposes the main information and metrics for this entity
- `Outlier detection overview` exposes the event outliers detection chart
- `Outlier detection configuration` provides different options to configure the outliers detection
- `Data sampling` shows the results from the data sampling & event format recognition engine
- `Data parsing quality` exposes indexing time parsing issues such as truncation issues for this sourcetype, if any.
- `Lagging performances` exposes the event lag and ingestion lag recorded metrics in the metric index
- `Status flipping` exposes all status flipping events that were stored in the summary index
- `Status message` exposes the current status of the data source in a human friendly manner
- `Audit changes` exposes all changes recorded in the audit KVstore for that entity

Overview data source tab

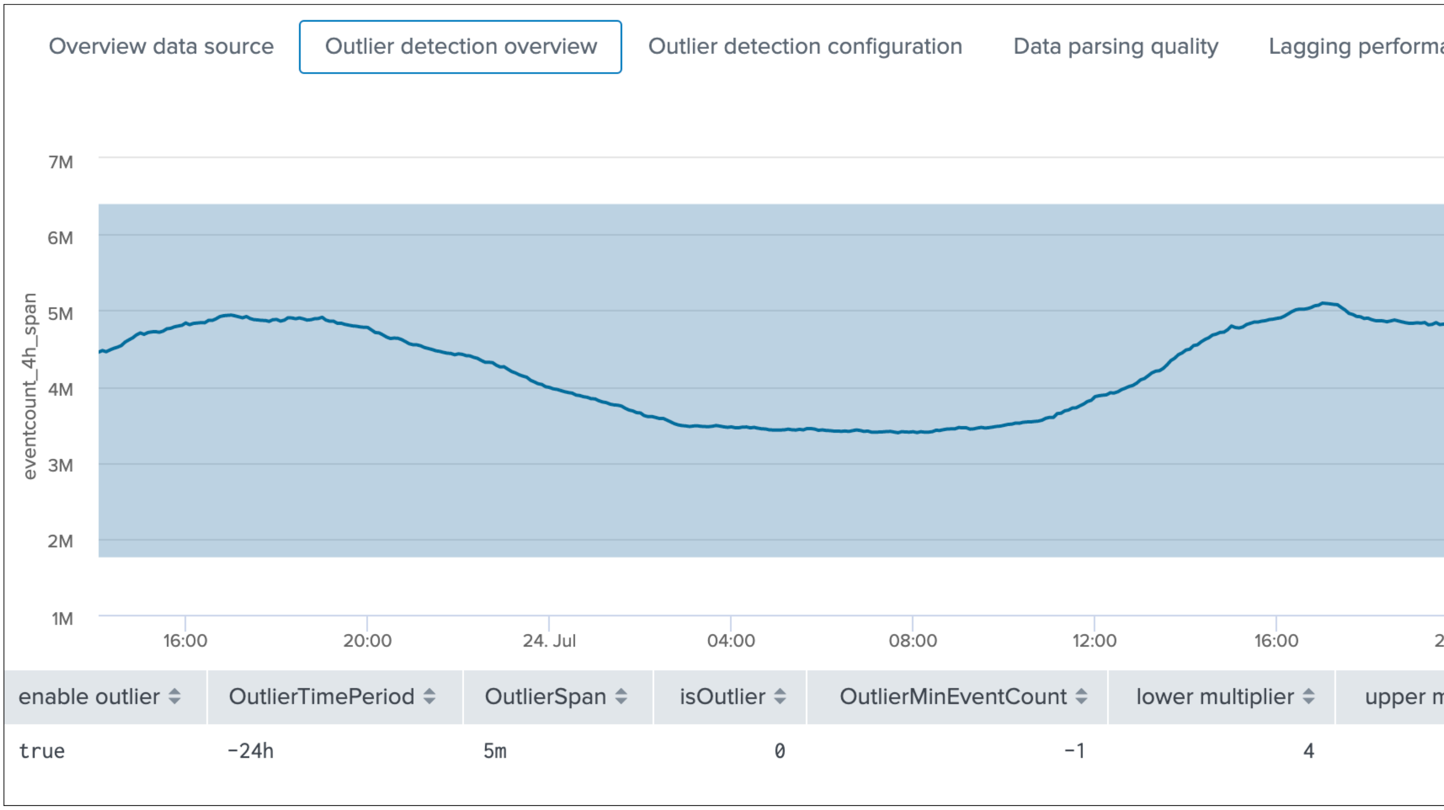


This screen exposes several single forms with the following calculations:

- `PERC95 INGESTION LAG` is the percentile 95 of the lag ingestion determined for this entity (`_indextime - _time`)
- `AVG INGESTION LAG` is the average lag ingestion for that entity
- `CURRENT EVENT LAG` is the current event lag calculated for this entity (`now() - _time`), this basically exposes how late this data source compared between now and the very last event in the entity
- `SLA PCT` is the SLA percentage which basically exposes the percent of time that entity has spent in a not green / blue state

Finally, a chart over time exposes the event count and the ingestion lag for that entity.

Outlier detection overview



enable outlier	OutlierTimePeriod	OutlierSpan	isOutlier	OutlierMinEventCount	lower multiplier	upper m
true	-24h	5m	0	-1	4	

This screen exposes the events outliers detection results over time, the purpose of the outliers detection is to provide advanced capabilities to detect when the number of events produced in the scope of an entity goes below or above a certain level, which level gets automatically defined upon the historical behaviour of the data.

For this purpose, every time the short term tracker runs, it records different metrics which includes the number of events on per 4 hours time window. (which matches the time frame scope of the short term tracker)

Then in short, a scheduled report runs every hour to perform lower bound and upper bound calculations depending on different configurable factors.

Assuming the outliers detection is enabled, if the workflow detects a significant gap in the event count, and optionally an increase too, the state of the entity will be affected and potentially turn red.

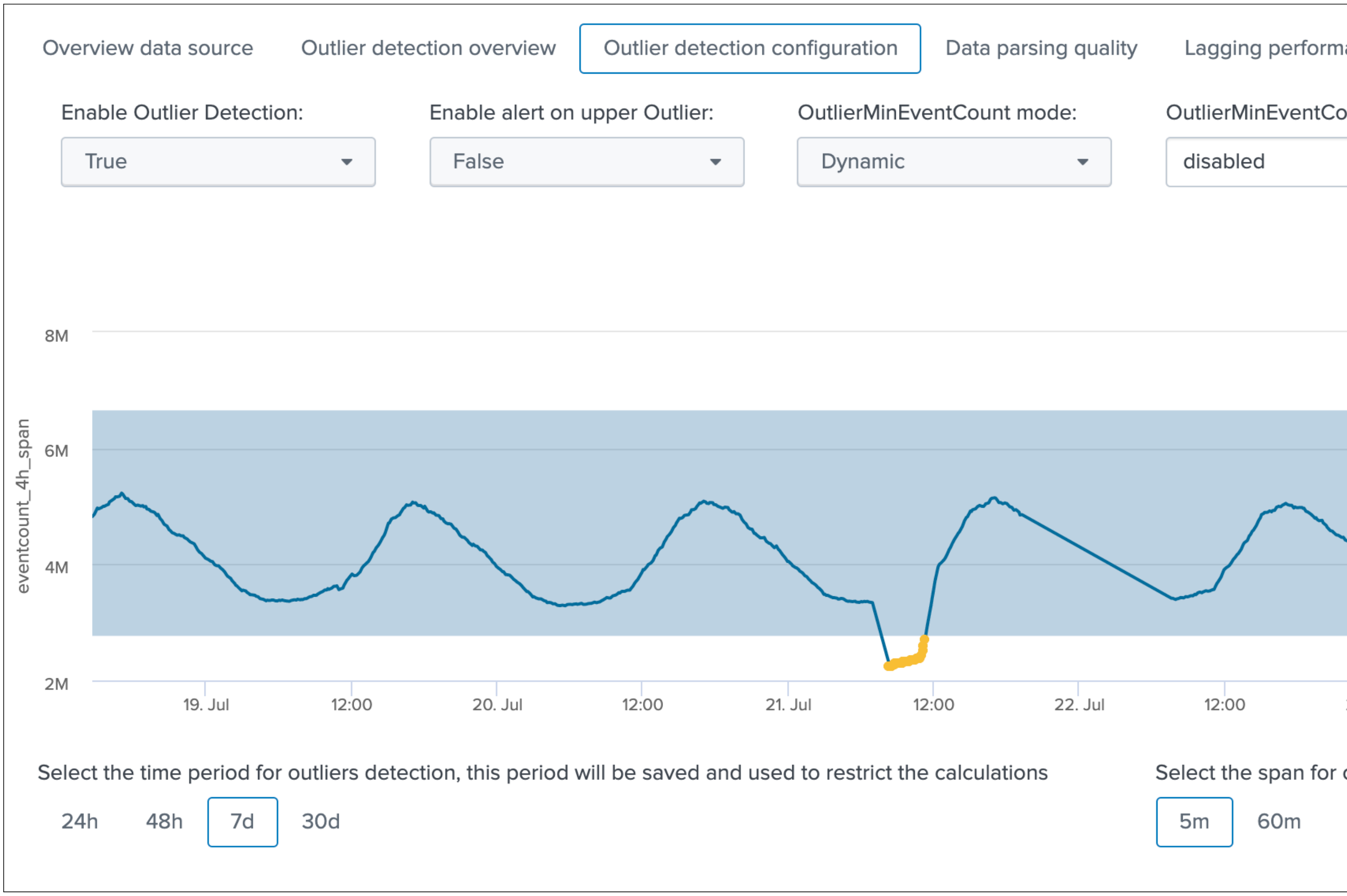
The table at the bottom of the screen provides additional information:

- `enable outlier` can be true or false and defines if the outliers detection is taken into account for the state definition of that entity
- `OutlierTimePeriod` is a time frame period between a list of restricted values, which defines the time period the backend will be looking at during for the lower bound, upper bound and standard deviation calculation
- `OutlierSpan` is used when rendering the outliers over time chart and does not influence the detection (for example if a data source emits data every 30 minutes you will want to apply a more relevant value for a better rendering)
- `isOutlier` is the current status, a value of 0 indicates that no outliers are currently active for this entity, a value of 1 indicates TrackMe detected outliers currently
- `OutlierMinEventCount` is an optional static value that can be defined for the lower bound, this is useful if you want to statically specific the minimal per 4 hours event count to be accepted

- `lower multiplier` is a multiplier that is used for the automatic definition of the lower bound, decreasing or increasing will impact the value of the lower bound definition
- `upper multiplier` is a multiplier that is used for the automatic definition of the upper bound, decreasing or increasing will impact the value of the upper bound definition
- `alert on upper` defines if upper outliers should be taken into account and affect the state if an abnormal number of events is coming in, default is false
- `lowerBound` is the lower threshold, an event count below this value will be considered as outliers
- `upperBound` is the upper threshold, an event count above this value will be considered as outlier, but will only impact the state if the alert on upper is true
- `stdev` is the standard deviation calculated by the workflow for this entity, and is used as the reference for the lower and upper bound calculation associated with the lower and upper multipliers
- `avg` represents the average 4 hours amount of event count for this entity

See *Outliers detection and behaviour analytic* for more details about the feature.

Outlier detection configuration



Enable Outlier Detection: True

Enable alert on upper Outlier: False

OutlierMinEventCount mode: Dynamic

OutlierMinEventCo disabled

Select the time period for outliers detection, this period will be saved and used to restrict the calculations

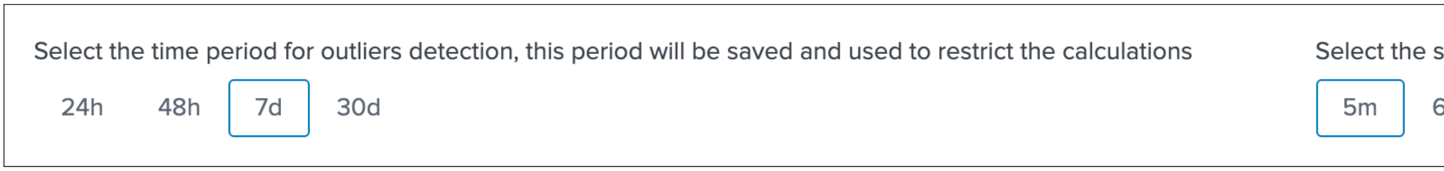
Select the span for c

This is the screen provided to configure the outliers detection for a given entity, which exposes a simulation of the results over time, allowing you to train your settings before they are applied.

On the top part of the screen you will interact with the settings exposes in the previous section:

- `Enable Outlier Detection:` you can choose to disable the Outliers detection for a given entity, default is enabled
- `Enable alert on upper Outlier:` you can choose to alert on upper outliers detection, default is false
- `OutlierMinEventCount mode:` you can choose to let the workflow defining dynamically the lower bound value, or define yourself a static threshold if you need it
- `OutlierMinEventCount:` static lower bound value if static threshold is used
- `Lower threshold multiplier:` the multiplier for the lower band calculation, must be a numerical value which will impact the lower bound calculation (the lower the multiplier is, the closer to the actual standard deviation the calculation will be)
- `Upper threshold multiplier:` the multiplier for the upper band calculation, must be a numerical value which will impact the upper bound calculation (the lower the multiplier is, the closer to the actual standard deviation the calculation will be)

Finally, there are two time related settings to interact with:



Select the time period for outliers detection, this period will be saved and used to restrict the calculations

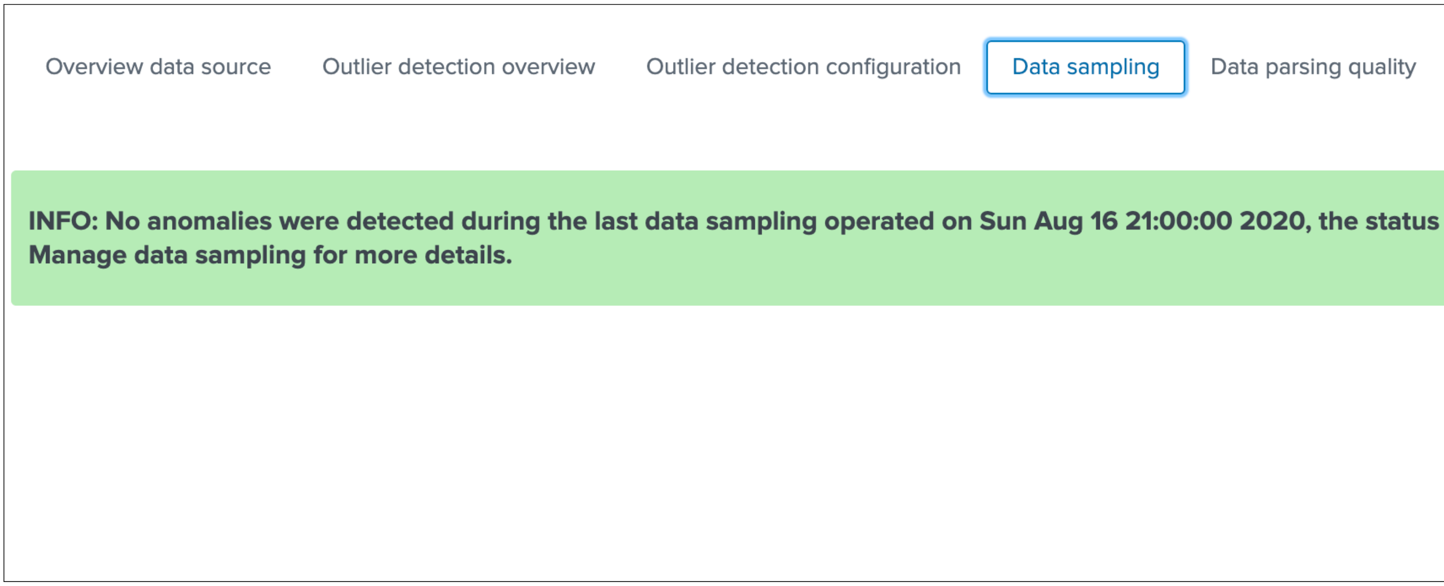
Select the s

- `time period for outliers detection` defines the time frame TrackMe will be looking at for the outliers calculations (lower/upper bands etc) which is using the recorded metrics every time the short term trackers ran
- `span for outliers rendering` is an additional setting which impact the graphical rendering within the outliers screen, but not the results of the outliers detection itself

See *Outliers detection and behaviour analytic* for more details about the feature.

Data sampling

The data sampling tab exposes the status of the data sampling and format recognition engine:

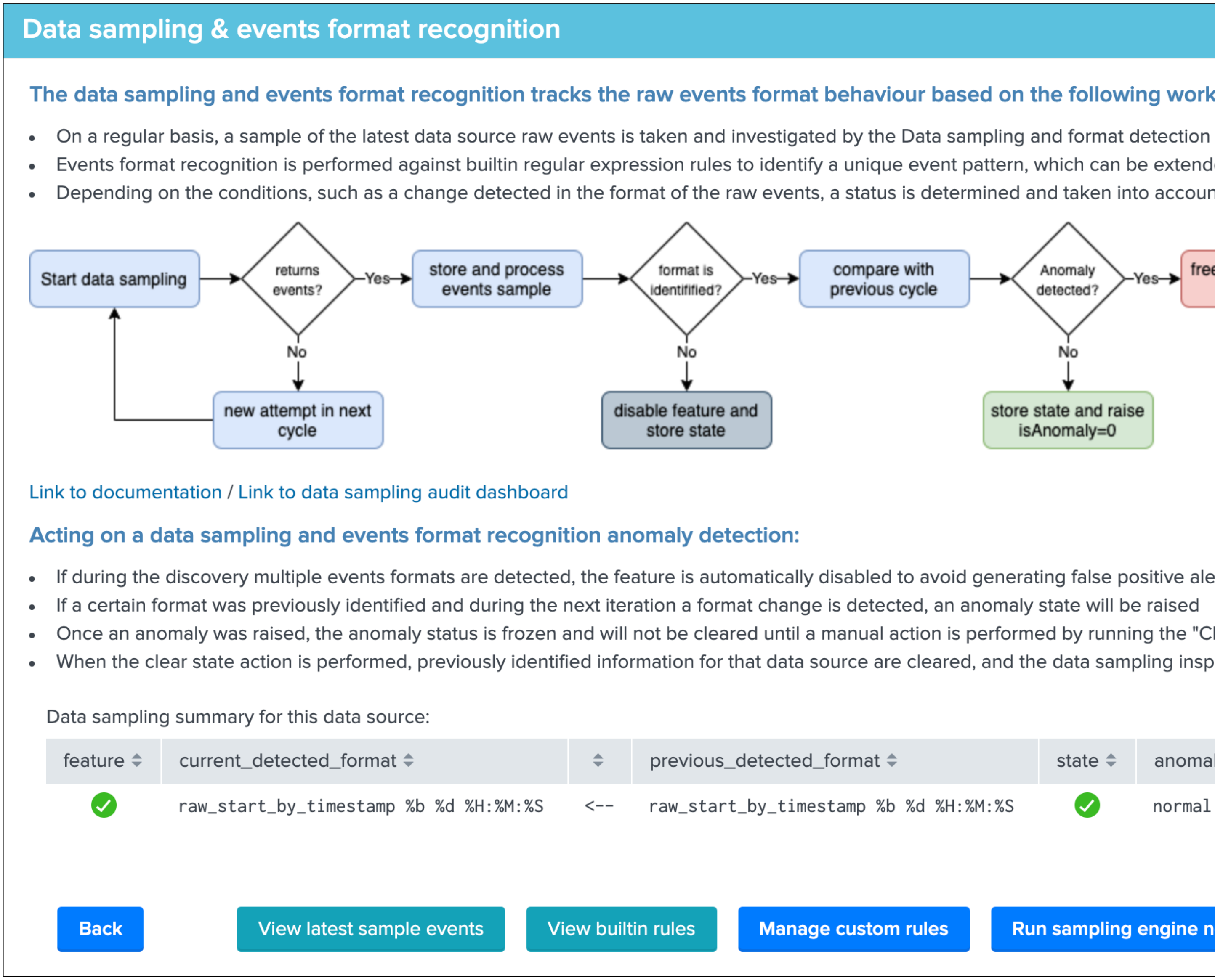


INFO: No anomalies were detected during the last data sampling operated on Sun Aug 16 21:00:00 2020, the status
Manage data sampling for more details.

The data sampling message can be:

- `green:` if no anomalies were detected
- `blue:` if the data sampling did not handle this data source yet
- `orange:` if conditions do not allow to handle this data source, which can be multi-format detected at discovery, or no identifiable event formats (data sampling will be deactivated automatically)
- `red:` if anomalies were detected by the data engine, anomalies can be due to a change in the event format, or multiple events formats detected post discovery

The button **Manage data sampling** provides summary information about the data samping status and access to data sampling related features:



Data sampling & events format recognition

The data sampling and events format recognition tracks the raw events format behaviour based on the following work

- On a regular basis, a sample of the latest data source raw events is taken and investigated by the Data sampling and format detection
- Events format recognition is performed against builtin regular expression rules to identify a unique event pattern, which can be extend
- Depending on the conditions, such as a change detected in the format of the raw events, a status is determined and taken into accoun

Start data sampling
returns events?
Yes
No
new attempt in next cycle
store and process events sample
format is identifified?
Yes
No
disable feature and store state
compare with previous cycle
Anomaly detected?
Yes
No
store state and raise isAnomaly=0
free

Link to documentation / Link to data sampling audit dashboard

Acting on a data sampling and events format recognition anomaly detection:

- If during the discovery multiple events formats are detected, the feature is automatically disabled to avoid generating false positive ale
- If a certain format was previously identified and during the next iteration a format change is detected, an anomaly state will be raised
- Once an anomaly was raised, the anomaly status is frozen and will not be cleared until a manual action is performed by running the "Cl
- When the clear state action is performed, previously identified information for that data source are cleared, and the data sampling insp

Data sampling summary for this data source:

feature ⬍	current_detected_format ⬍	⬍	previous_detected_format ⬍	state ⬍	anomal
✓	`raw_start_by_timestamp %b %d %H:%M:%S`	`<--`	`raw_start_by_timestamp %b %d %H:%M:%S`	✓	`normal`

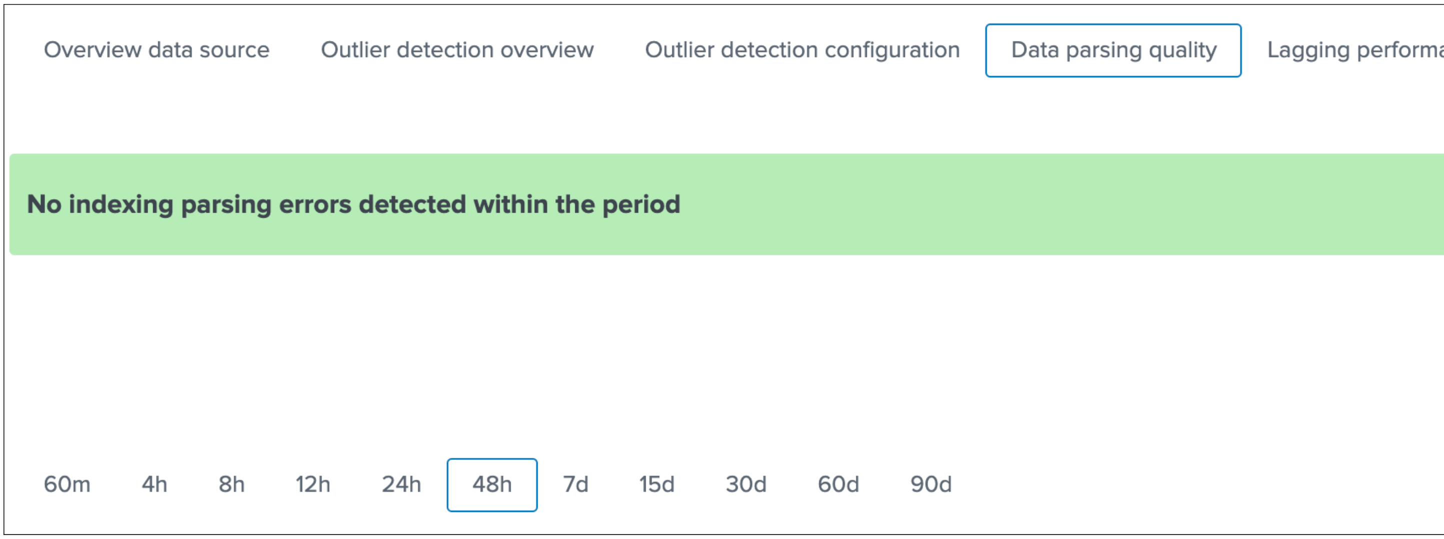
Quick button access:

- `View latest sample events`: open in search access to the last sample of raw events that were processed (raw events and identified format)
- `View builtin rules`: view the builtin rules (builtin rules are regular expressions rules provided by default)
- `Manage custom rules`: view, create and delete custom rules to handle any format that would not be recognized by the builtin rules
- `Run sampling engine now`: runs the sampling engine now for this data source
- `Clear state and run sampling`: clears the previously known states and run the sampling engine as it was the first time the engine handles this data source

See *Data sampling and event formats recognition* for more details about the feature.

Data parsing quality

The data parsing quality screen exposes if there are any indexing time parsing issues found for this sourcetype:



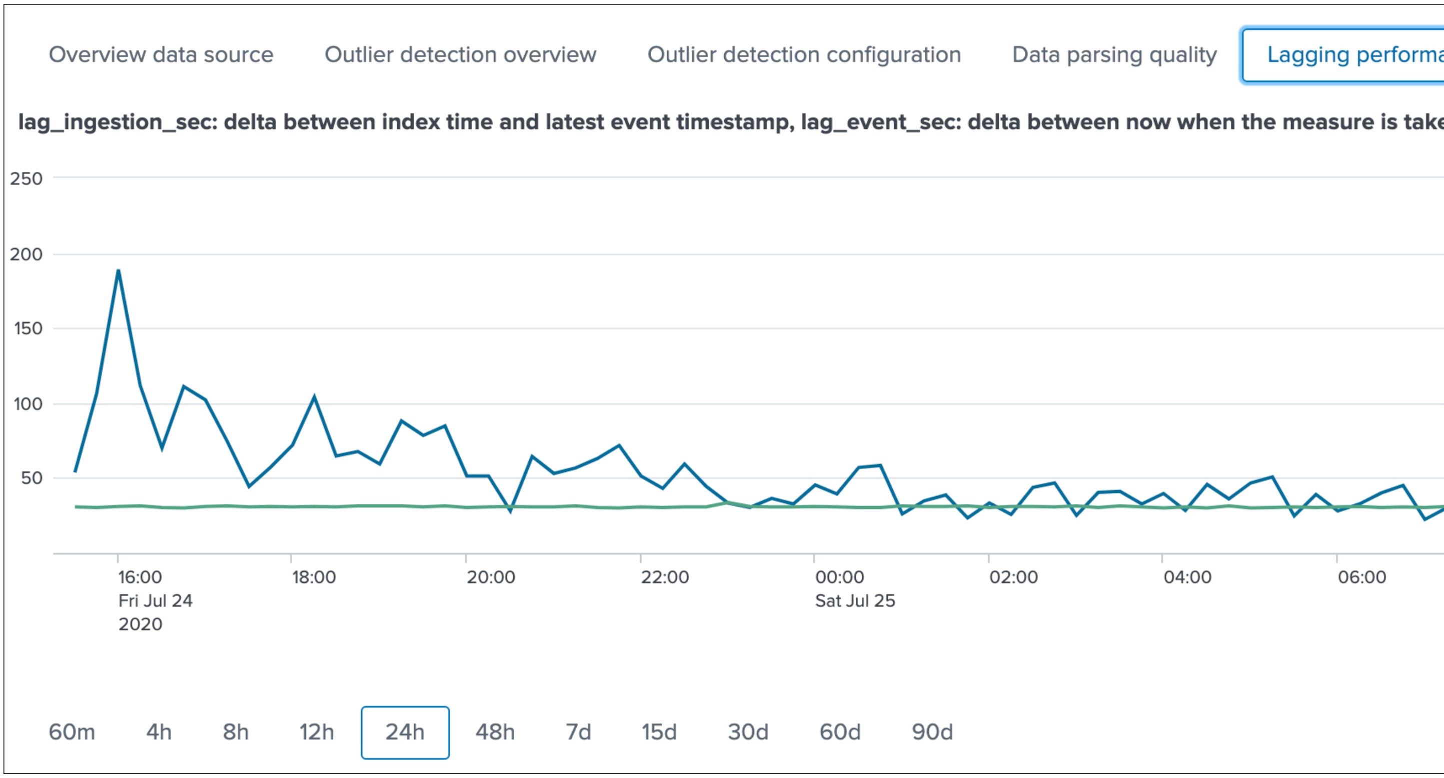
No indexing parsing errors detected within the period

Note: for data sources, the scope of indexing time parsing issues happens on the sourcetype level from a Splunk point of view, this means that if there are any parsing issues found for this sourcetype, this can be linked to this data source but as well with any other data source that looks at the same sourcetype.

Under normal conditions, this screen should not show any parsing errors, if there are any, these should be fixed.

Lagging performances

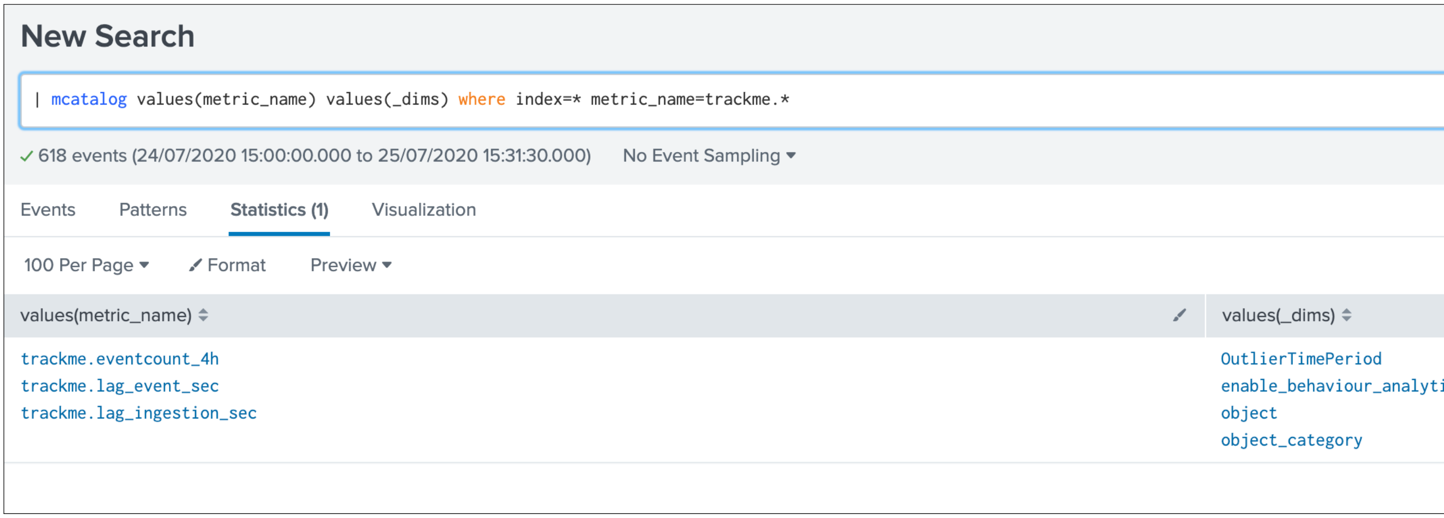
This screen exposes the event and ingestion lagging metrics that have been recorded each time the short trackers ran, these metrics are stored via a call to the mcollect command and stored into a metric store index:



lag_ingestion_sec: delta between index time and latest event timestamp, lag_event_sec: delta between now when the measure is take

The following mcatalog search can be used to expose the metrics stored in the metric store and the dimensions:

```
| mcatalog values(metric_name) values(_dims) where index=* metric_name=trackme.*
```



```
| mcatalog values(metric_name) values(_dims) where index=* metric_name=trackme.*
```

✓ 618 events (24/07/2020 15:00:00.000 to 25/07/2020 15:31:30.000)

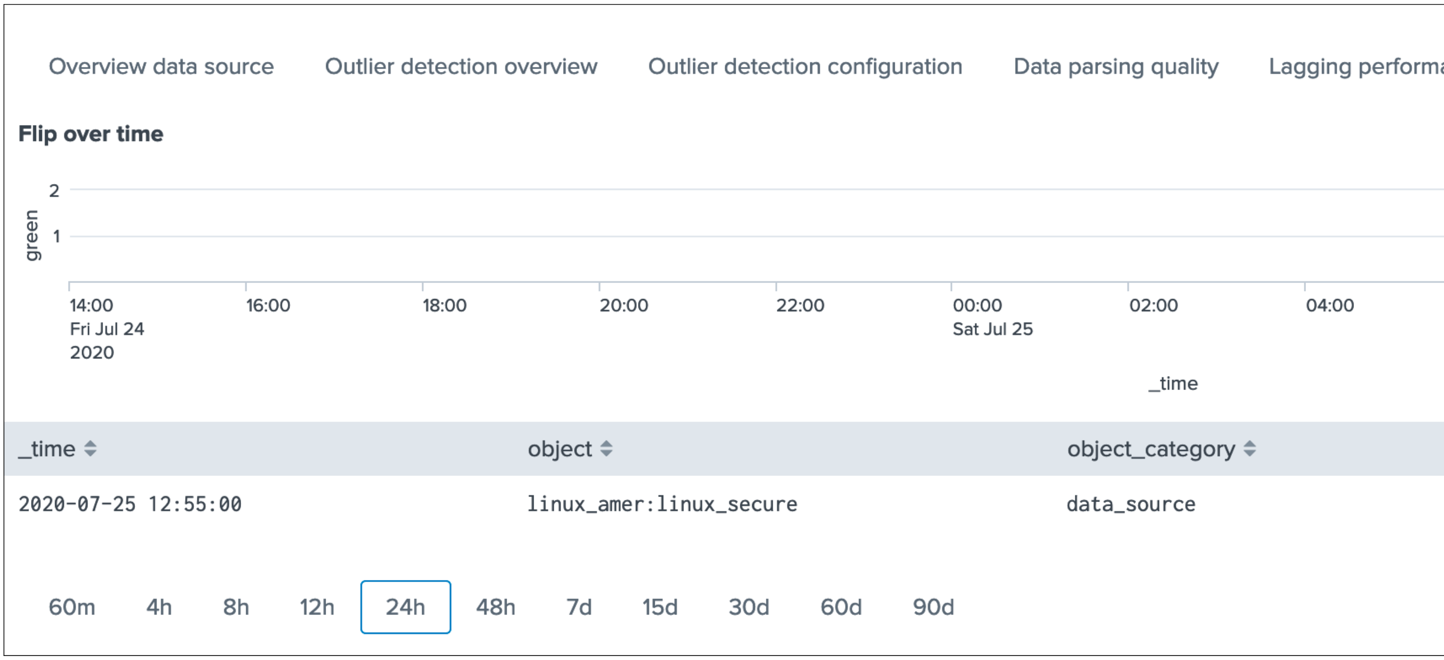
values(metric_name)	values(_dims)
trackme.eventcount_4h trackme.lag_event_sec trackme.lag_ingestion_sec	OutlierTimePeriod enable_behaviour_analyti object object_category

The main dimensions are:

- `object_category` which represents the type of entities, being data_source or data_host
- `object` which is the entity unique identifier, data_name for data sources, data_host for data hosts

Status flipping

This screen exposes all the flipping status events that were recorded for that entity during the time period that is selected:



_time	object	object_category
2020-07-25 12:55:00	linux_amer:linux_secure	data_source

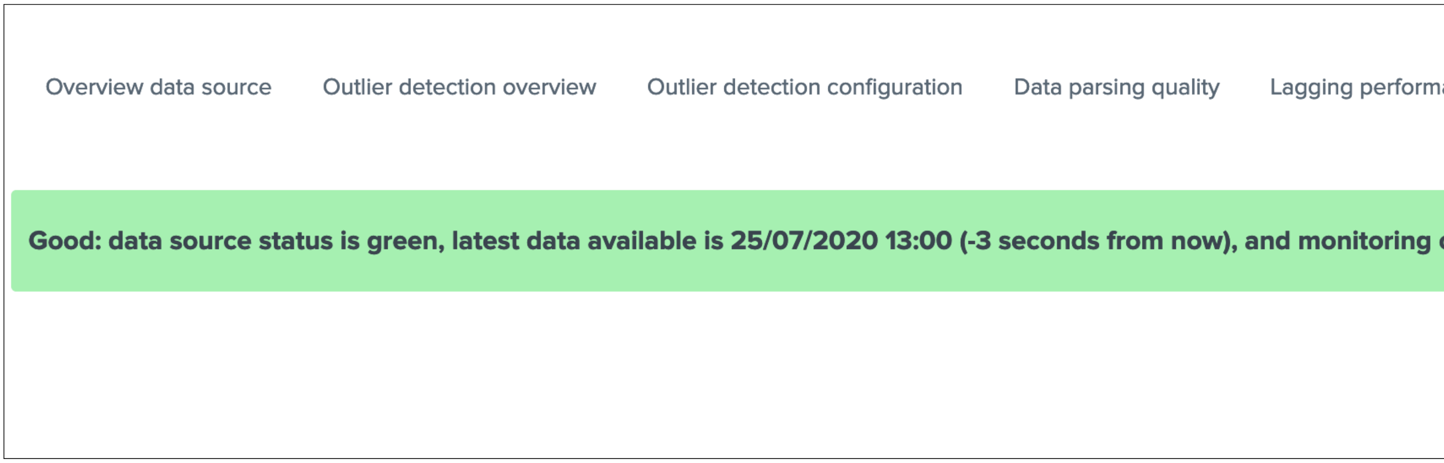
Key information:

- Anytime an entity changes from a state to another, a record is generated and indexed in the summary index
- When an entity is first added to the collection during its discovery, the origin state will be discovered
- The target state is the state (green / red and so forth) that the entity has switched to

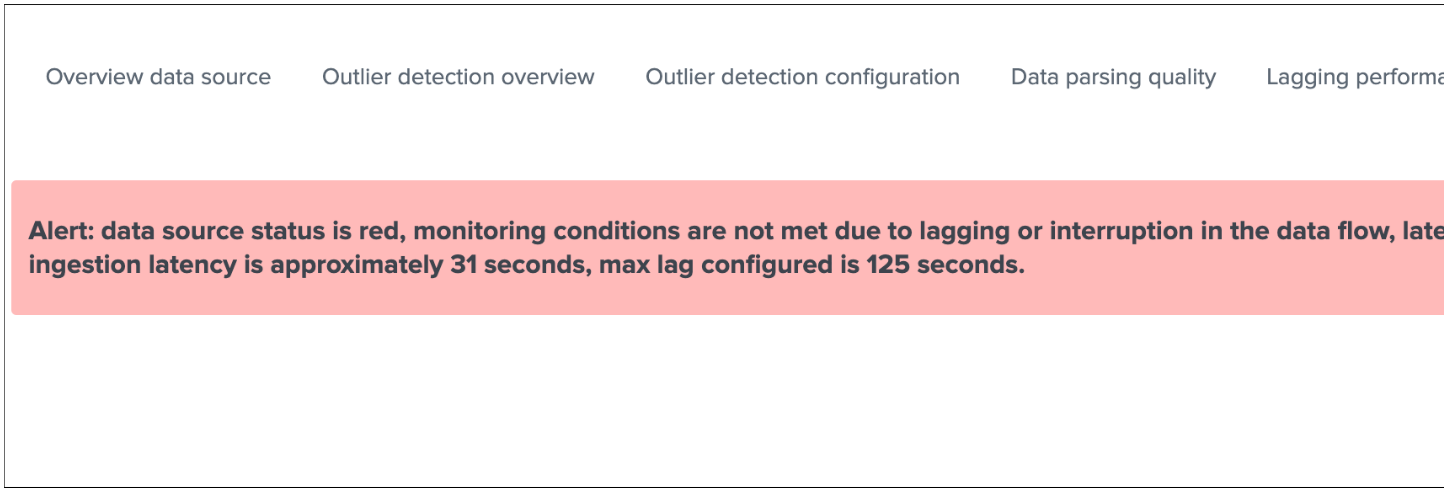
Status message

This screen exposes a human friendly message describing the current state of the entity, depending on the conditions the message will appear as green, red, orange or blue:

example of a green state:

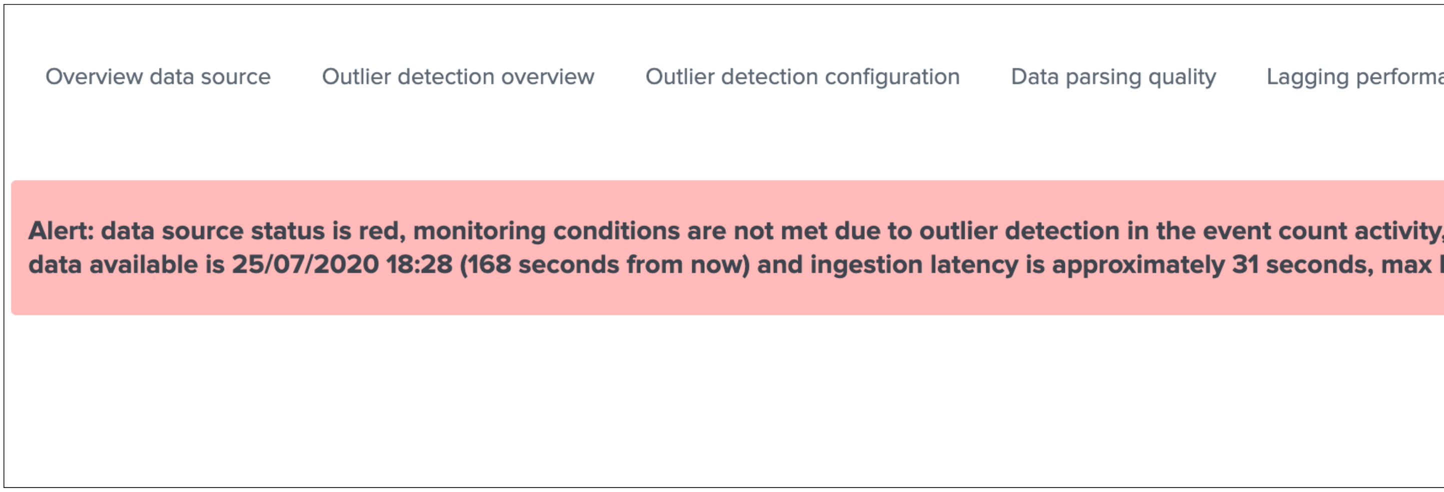


Good: data source status is green, latest data available is 25/07/2020 13:00 (-3 seconds from now), and monitoring c *example of a red state due to lagging conditions not met:*



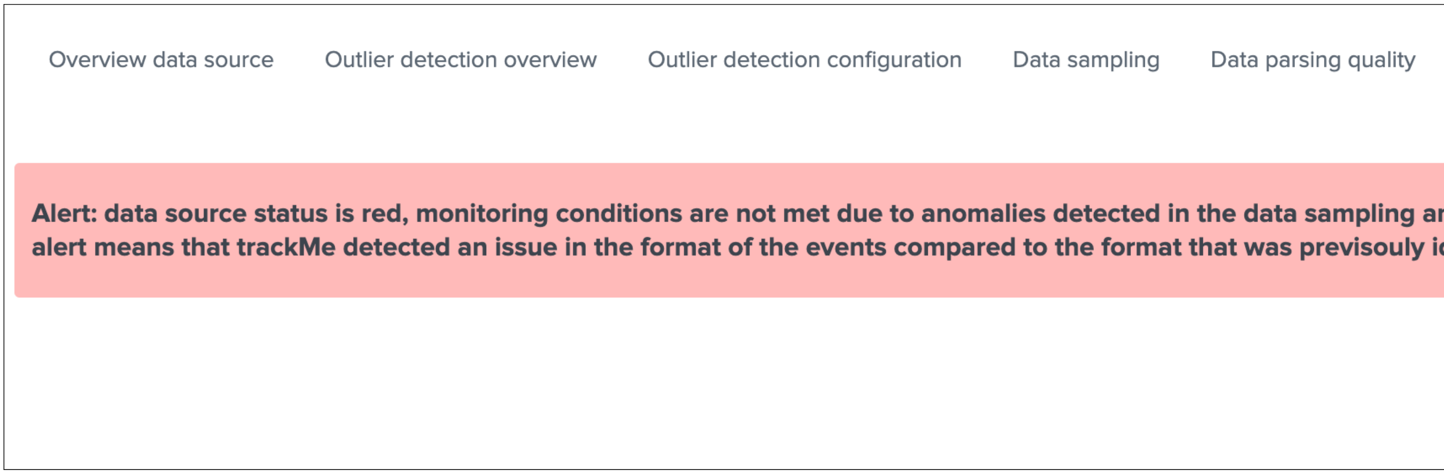
Alert: data source status is red, monitoring conditions are not met due to lagging or interruption in the data flow, late
ingestion latency is approximately 31 seconds, max lag configured is 125 seconds.

example of a red state due to outliers detection:



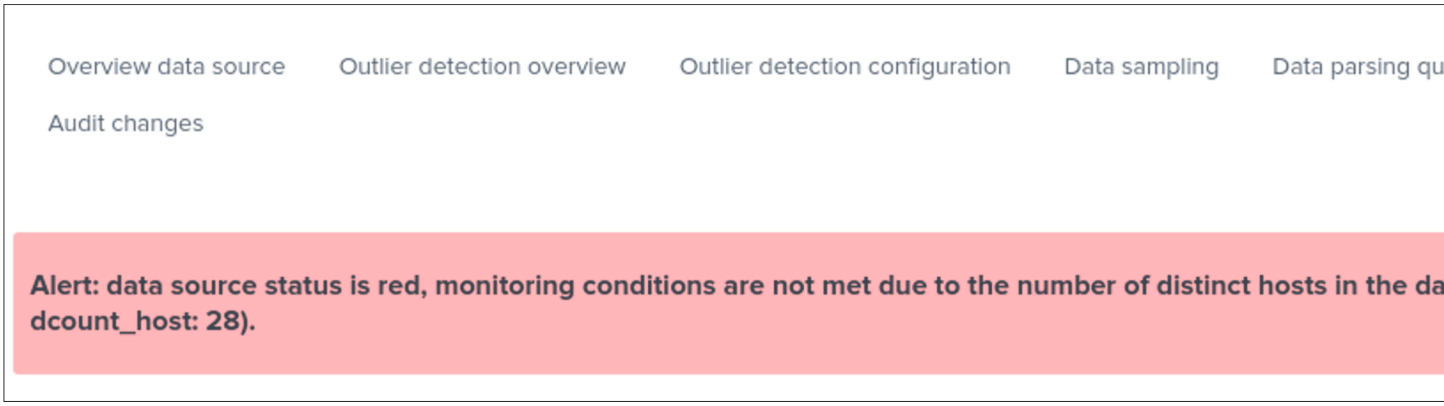
Alert: data source status is red, monitoring conditions are not met due to outlier detection in the event count activity,
data available is 25/07/2020 18:28 (168 seconds from now) and ingestion latency is approximately 31 seconds, max l

example of a red state due to data sampling anomalies detected:



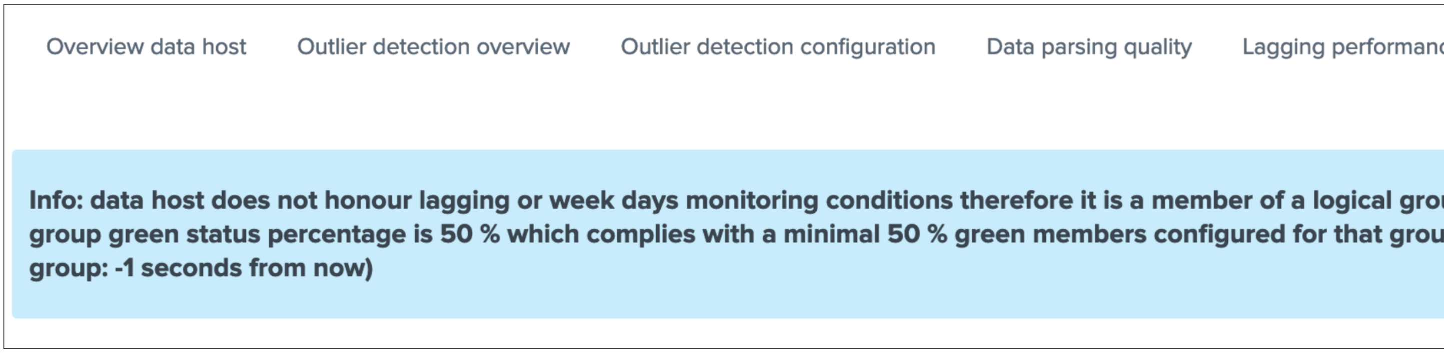
Alert: data source status is red, monitoring conditions are not met due to anomalies detected in the data sampling ar
alert means that trackMe detected an issue in the format of the events compared to the format that was previsouly ic

example of a red state due to hosts dcount threshold not reached:



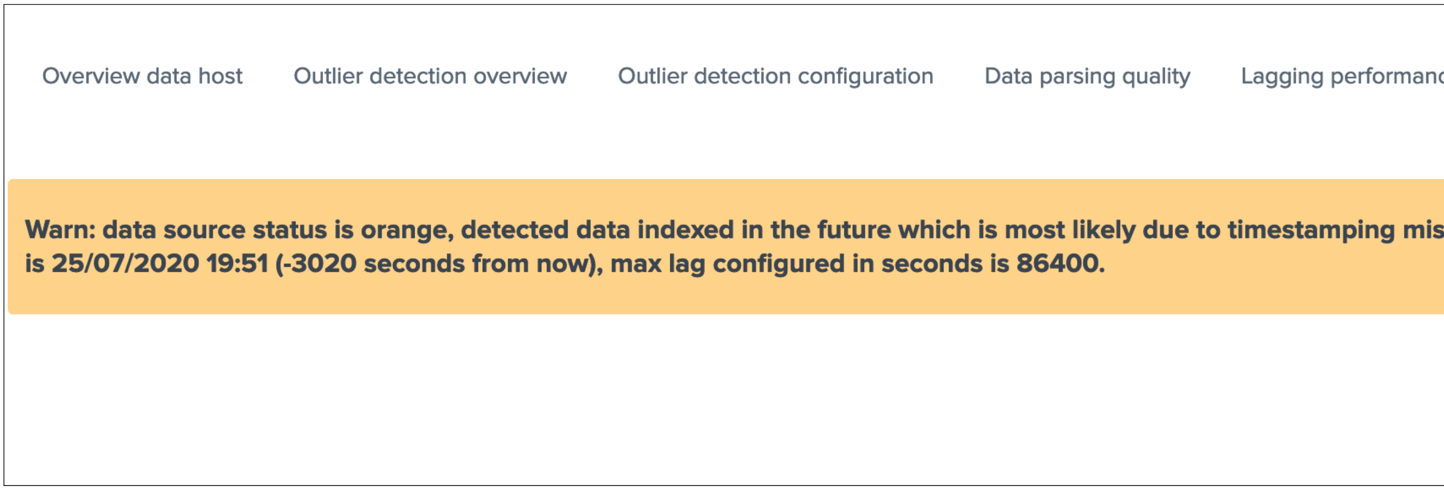
Alert: data source status is red, monitoring conditions are not met due to the number of distinct hosts in the da
dcount_host: 28).

example of a blue state due to logical groups monitoring conditions not met (applies to data hosts and metrics hosts only):



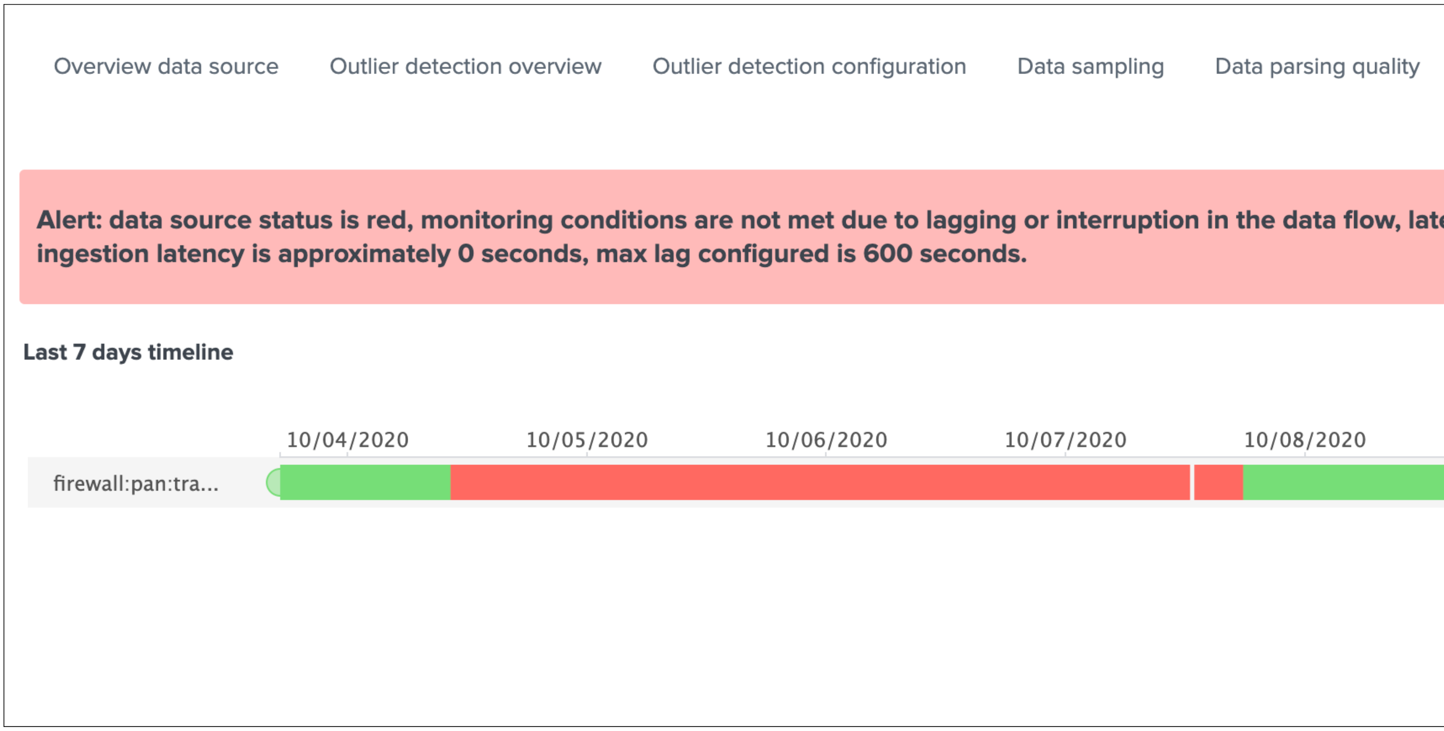
Info: data host does not honour lagging or week days monitoring conditions therefore it is a member of a logical grou
group green status percentage is 50 % which complies with a minimal 50 % green members configured for that grou
group: -1 seconds from now)

example of an orange state due to data indexed in the future:



Warn: data source status is orange, detected data indexed in the future which is most likely due to timestamping mis
is 25/07/2020 19:51 (-3020 seconds from now), max lag configured in seconds is 86400.

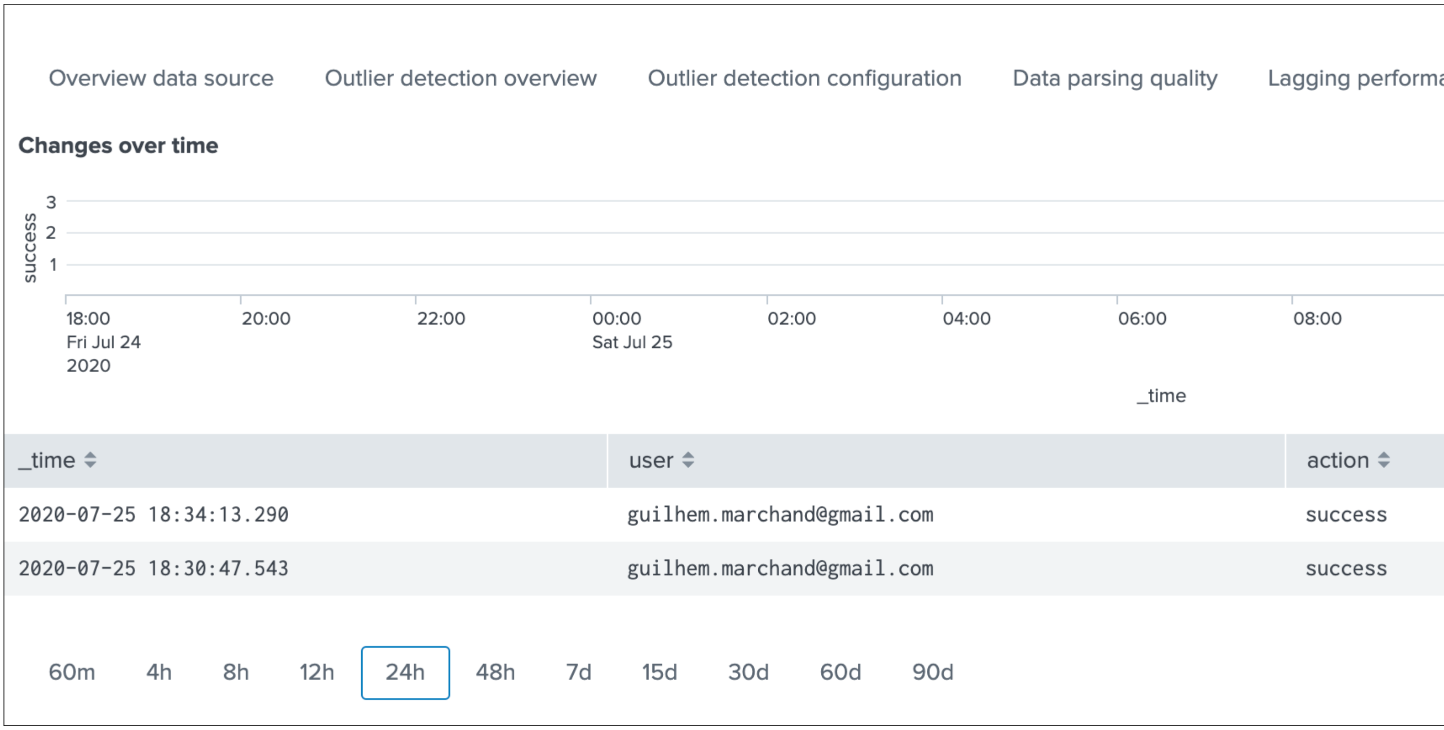
In addition, an integration using the timeline custom view provides an enhanced overview of the entity status over time:



Alert: data source status is red, monitoring conditions are not met due to lagging or interruption in the data flow, late ingestion latency is approximately 0 seconds, max lag configured is 600 seconds.

Audit changes

This final screen exposes all changes that were applied within the UI to that entity which are systematically recorded in the audit KVstore:

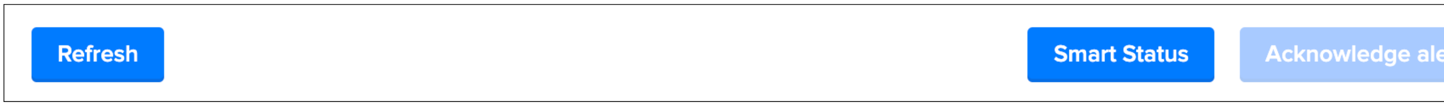


_time	user	action
2020-07-25 18:34:13.290	guilhem.marchand@gmail.com	success
2020-07-25 18:30:47.543	guilhem.marchand@gmail.com	success

See *Auditing changes* for more details about the feature.

Action buttons

Finally, the bottom part of the screen provides different buttons which lead to different actions:



Actions:

- `Refresh` will refresh all values related to this entity, it will actually run a specific version of the tracker and update the KVstore record of this data source. Charts and other calculations are refreshed as well.
- `Smart Status` is a powerful TrackMe REST API endpoint that does automated analysis and conditional correlations to provide an advanced status of the entity, and fast the investigaton of an issue root cause.
- `Acknowledge alert` can only be clicked if the data source is effectively in a red state, acknowledging an alert prevent the out of the box alerts from triggering a new alert for this entity until the acknowledgment expires.
- `Enable` can only be clicked if the monitoring state is disabled, if clicked and confirmed, the value of the field `data_monitored_state` will switch from disabled to enabled
- `Disable` opposite of the previous
- `Modify` provides access to the unified modification window which allows interacting with different settings related to this entity
- `Search` opens a search window in a new tab for that entity

See *Alerts tracking* for more details about the acknowledgment feature and alert related configurations

See *Data source unified update* for more details about the unified update UI for data sources

Data Hosts tracking and features

Rather than duplicating all the previous explanations, let's expose the differences between the data sources and data hosts tracking.

Data host monitoring

Data hosts monitoring does data discovery on a per host basis, relying on the `Splunk host Metadata`.

To achieve this, TrackMe uses tstats based queries to retrieve and record valuable Metadata information, in a simplistic form this is very similar to the following query:

```
| tstats count, values(sourcetype) where index=* by host
```

Particularities of data hosts monitoring

The features are almost equivalents between data sources and data hosts, with a few exceptions:

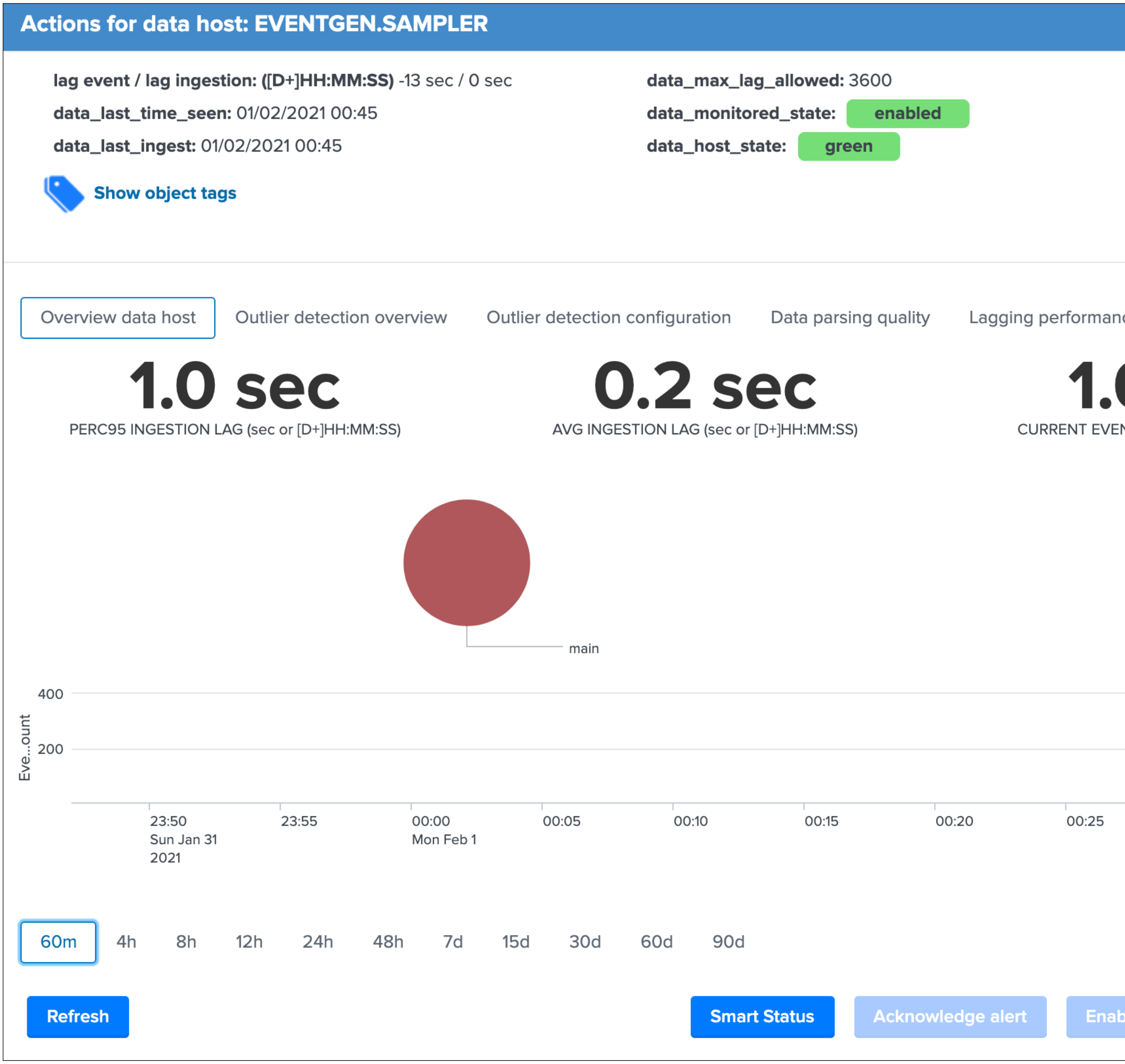
- `state condition:` the data host entity state depends on the global data host alerting policy (which is defined globally and can be overriden on a per host basis)
- Depending on the policy, he host state will turn red if either no more sourcetypes are generating data (track per host policy), or any of the sourcetypes monitored for the host has turned red (track per sourcetype policy)

- Using `allowlists and blocklists` provide additional granularity to define what data has to be included or is excluded during the searches
- `Outliers detection` is available for data hosts too and would help detecting significant changes such as a major sourcetype that is not ingested anymore
- `logical group`: a data host can be part of a logical group, this feature is useful for example to handle a couple of active / passive entities (example with firewalls) where the passive entity will not be generating any data actively
- `object tags`: this is an additional feature to data hosts and metric hosts that allows looking against a third party lookup, such as your CMDB data stored in Splunk, or the Splunk Enterprise Security assets knowledge, to provide an active link and access quickly these enrichment information

See *Logical groups (clusters)* for more details on this feature

See *Enrichment tags* for more details om this feature

Additionally, if there has been indexes migrations, or if one or more sourcetypes have been decomissioned, this will affect the state of a given host if the alert policy is defined to track per sourcetype, you can reset the knowledge of indexes and sourcetypes on a per host basis via the reset button:



Actions for data host: EVENTGEN.SAMPLER

lag event / lag ingestion: ([D+]HH:MM:SS) -13 sec / 0 sec

data_last_time_seen: 01/02/2021 00:45

data_last_ingest: 01/02/2021 00:45

data_max_lag_allowed: 3600

data_monitored_state: enabled

data_host_state: green

Metric Hosts tracking and features

Metric hosts tracking is the third main notion in TrackMe, and deals with tracking hosts sending metrics to the Splunk metric store, let’s expose the feature particularities.

Metric host monitoring

The metric hosts feature tracks all metrics send to the Splunk metric store on a per host basis.

In a very simplistic form, the notion is similar to performing a search looking at all metrics with mstats on a per host basis and within a short time frame:

```
| mstats latest(_value) as value where index=* metric_name="*" by metric_name, index,␣
↪host span=1s
```

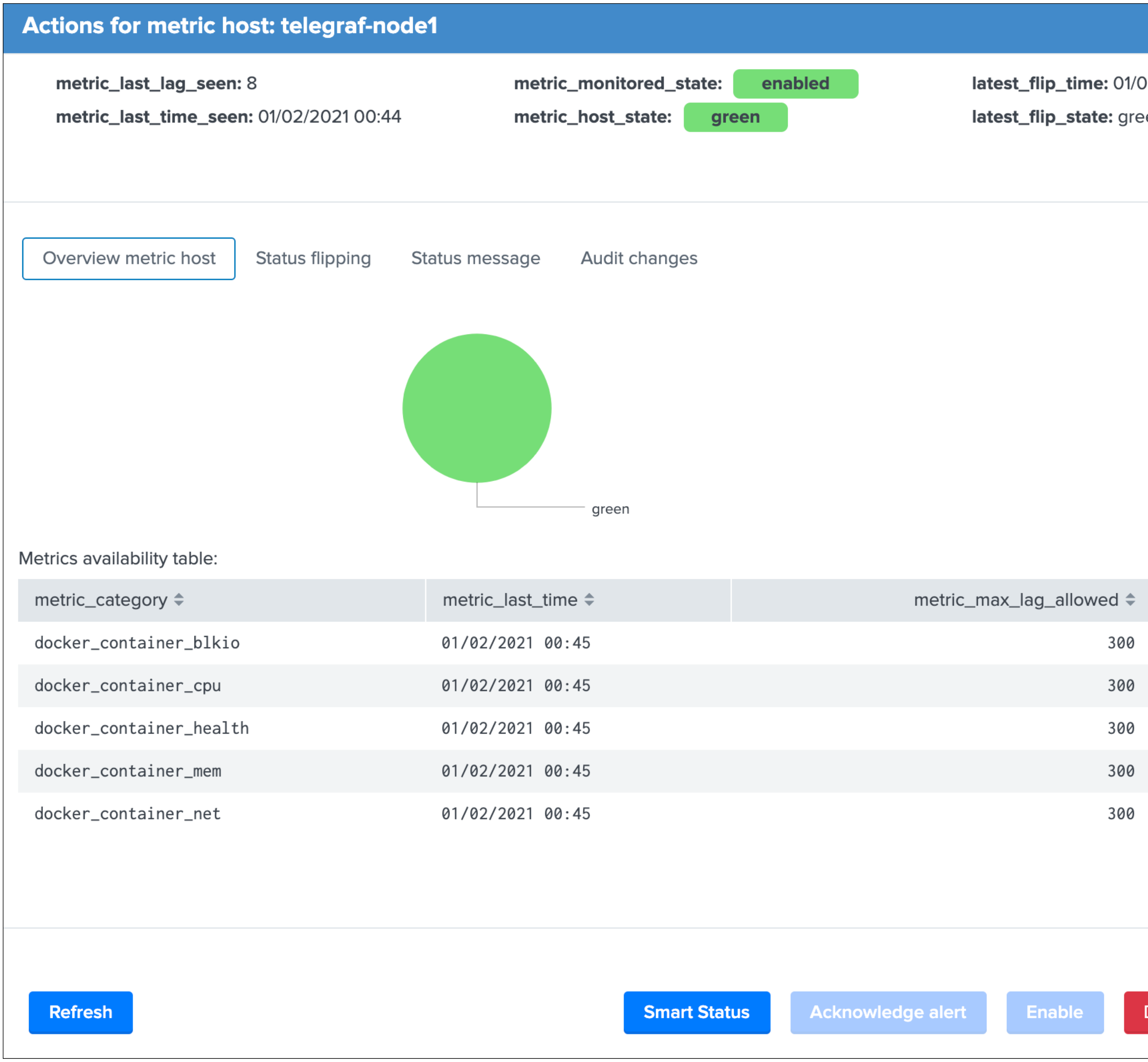
Then, the application groups all metrics on per metric metric category (the first metric name segment) and a per host basis.

Particularities of metric hosts monitoring

Compared to data sources and data hosts tracking, metric hosts tracking provides a similar level of features, with a few exceptions:

- `state condition:` the metric host state is conditioned by the availability of each metric category that was discovered for that entity
- Shall a metric category stop from being emitted, the state will be affected accordingly
- Using `allowlists and blocklists` provide additional granularity to define the include and exclude conditions of the metric discovery
- `Outliers detection` is not available for metrics hosts
- `logical group`: a metric host can be part of a logical group, this feature is useful for example to handle a couple of active / passive entities (example with firewalls) where the passive entity will not be generating any metrics actively
- `object tags`: this is an additional feature to data hosts and metric hosts that allows looking against a third party lookup, such as your CMDB data stored in Splunk, or the Splunk Enterprise Security assets knowledge, to provide an active link and access quickly these enrichment information
- Metric hosts tracking relies on the `default max lag allowed` per `metric category` which is defined by default to 5 minutes (300 seconds) and can be managed by creating `metric SLA policies`
- The entity screen provides some metric specific search options to provide insights against these specific entities and their metrics

Additionally, if a metric category stops being emitted this affects the global status of the entity, if these metrics are decomissioned you can reset the host metrics knowledge:



Actions for metric host: telegraf-node1

metric_last_lag_seen: 8

metric_last_time_seen: 01/02/2021 00:44

metric_monitored_state: enabled

metric_host_state: green

latest_flip_time: 01/0

latest_flip_state: gre

Metrics availability table:

metric_category	metric_last_time	metric_max_lag_allowed
docker_container_blkio	01/02/2021 00:45	300
docker_container_cpu	01/02/2021 00:45	300
docker_container_health	01/02/2021 00:45	300
docker_container_mem	01/02/2021 00:45	300
docker_container_net	01/02/2021 00:45	300

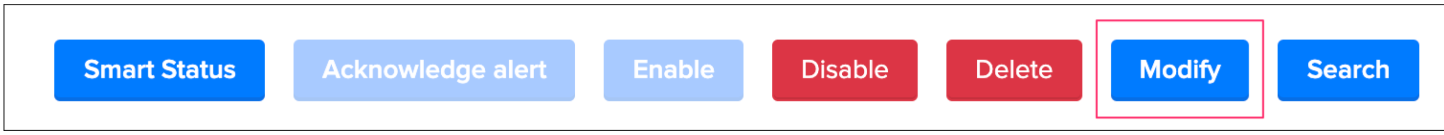
Triggering this action will remove the current knowledge of metric categories for this entity only and trigger a fresh discovery without losing additional settings like the priority.

See *Logical groups (clusters)* for more details on this feature

See *Enrichment tags* for more details om this feature

3.1.2 Unified update interface

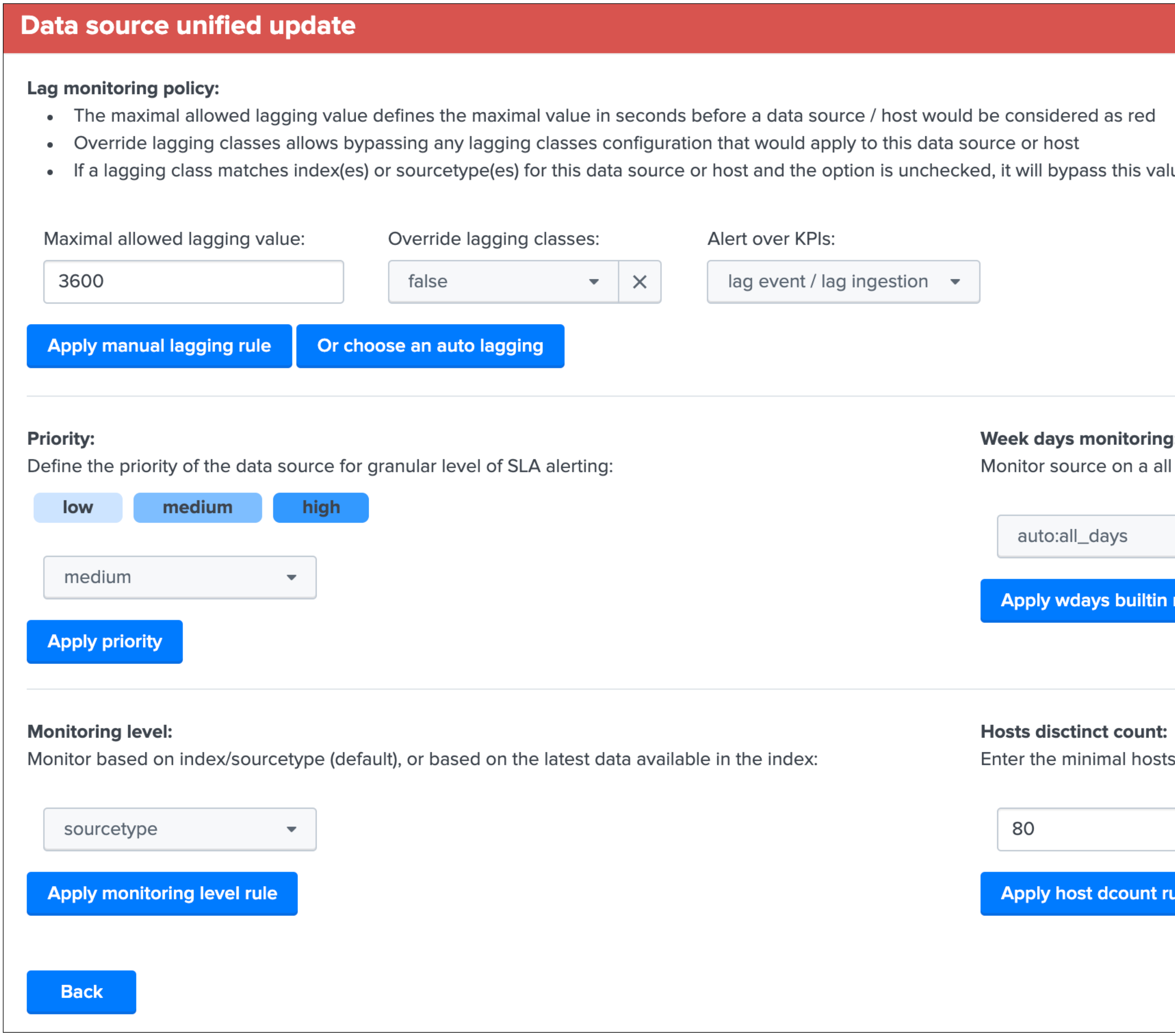
For each type of tracking, a unified update screen is available by clicking on the modify button when looking at a specific entity:



These interfaces are called unified as their main purpose is to provide a central place in the UI where the modification of the main key parameters would be achieved.

In this screens, you will define the priority level assignment, modify the lagging policy, manage logical groups, etc.

Data source unified update


Data source unified update

Lag monitoring policy:

- The maximal allowed lagging value defines the maximal value in seconds before a data source / host would be considered as red
- Override lagging classes allows bypassing any lagging classes configuration that would apply to this data source or host
- If a lagging class matches index(es) or sourcetype(es) for this data source or host and the option is unchecked, it will bypass this valu

Maximal allowed lagging value: 3600

Override lagging classes: false

Alert over KPIs: lag event / lag ingestion

Priority:

Define the priority of the data source for granular level of SLA alerting:

medium

Week days monitoring

Monitor source on a all

auto:all_days

Monitoring level:

Monitor based on index/sourcetype (default), or based on the latest data available in the index:

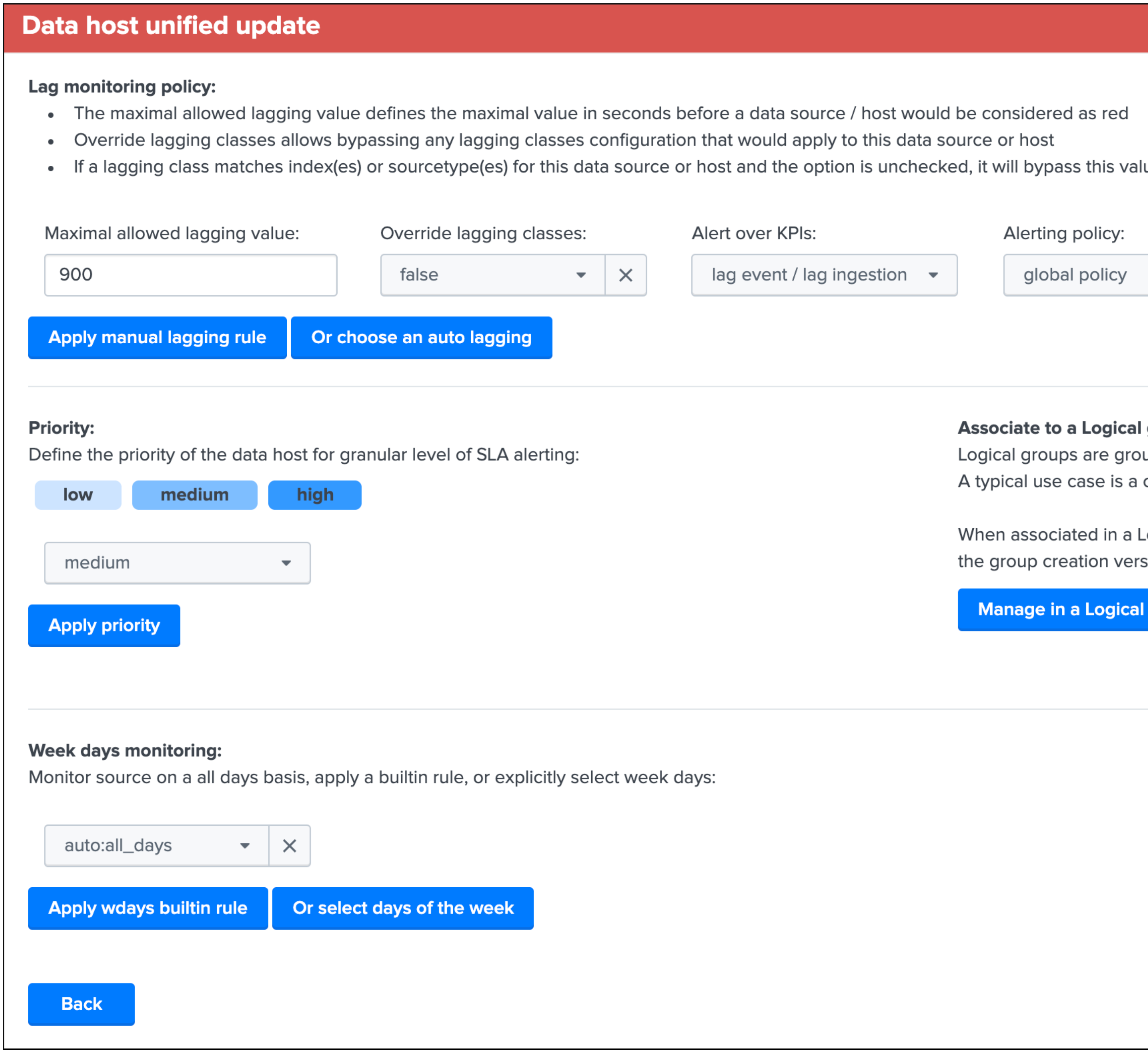
sourcetype

Hosts disctinct count:

Enter the minimal hosts

80

Data hosts unified update



Data host unified update

Lag monitoring policy:

- The maximal allowed lagging value defines the maximal value in seconds before a data source / host would be considered as red
- Override lagging classes allows bypassing any lagging classes configuration that would apply to this data source or host
- If a lagging class matches index(es) or sourcetype(es) for this data source or host and the option is unchecked, it will bypass this valu

Maximal allowed lagging value: 900

Override lagging classes: false

Alert over KPIs: lag event / lag ingestion

Alerting policy: global policy

Priority:

Define the priority of the data host for granular level of SLA alerting:

medium

Associate to a Logical

Logical groups are grou

A typical use case is a

When associated in a L

the group creation vers

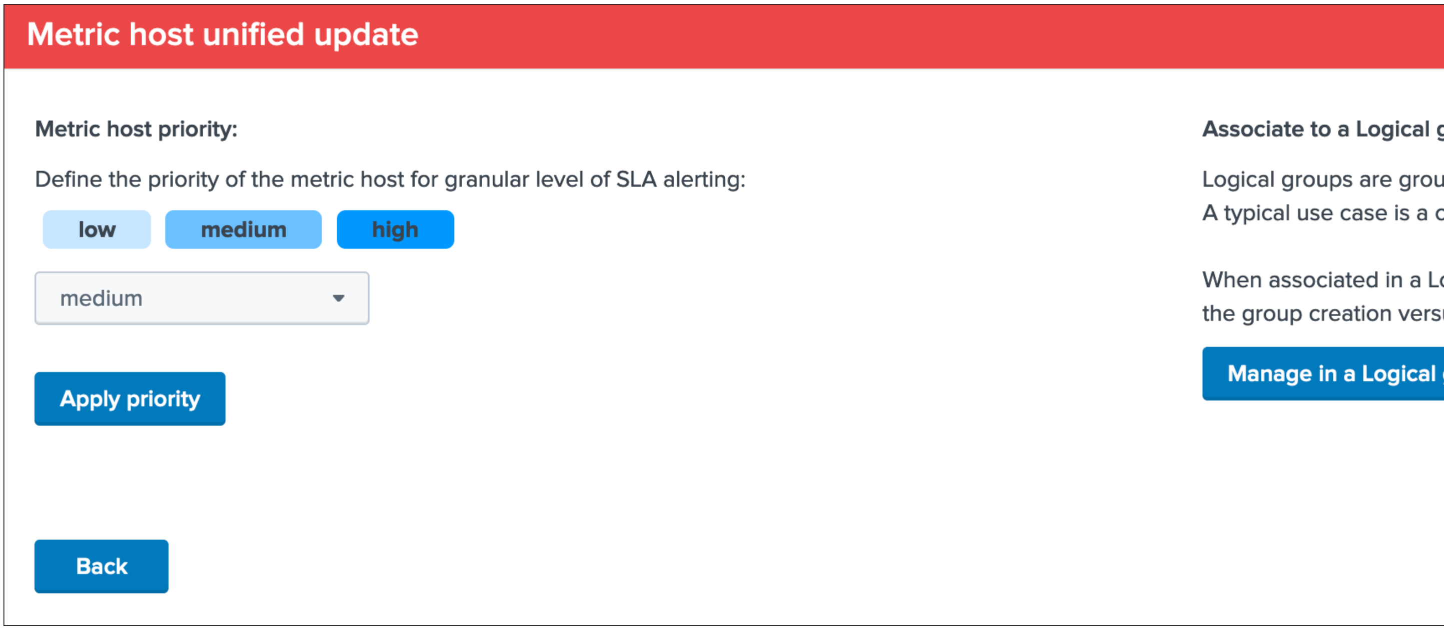
Week days monitoring:

Monitor source on a all days basis, apply a builtin rule, or explicitly select week days:

auto:all_days

Apply manual lagging rule
Or choose an auto lagging
low
medium
high
Apply priority
Manage in a Logical
Apply wdays builtin rule
Or select days of the week
Back

Metric hosts unified update



Metric host unified update

Metric host priority:

Define the priority of the metric host for granular level of SLA alerting:

medium

Associate to a Logical g

Logical groups are grou
A typical use case is a c

When associated in a Lo
the group creation versu

Unified update interface features

Lag monitoring policy:

In this part of the screen you will define:

- The `max lag allowed` value that conditions the state definition of the entity depending on the circumstances
- This value is in `seconds` and will be taken into account by the trackers to determine the colour of the state
- `Override lagging classes` allows bypassing any lagging class that would have defined and could be matching the conditions (index, sourcetype) of this entity
- You can choose which `KPIs` will be taken into account to determine the state regarding the `max lag allowed` and the two main lagging performance indicators
- For data hosts, the `alerting policy` allows controlling how to consider the green/red state assignment in regards with the state of each sourcetype indexed by the host

See *Lagging classes* for more details about the lagging classes feature.

See *Alerting policy for data hosts* for more details about the alerting policy feature.

Priority:

This is where you can define the priority of this entity. The priority is by default set to medium can by any of:

- `low`
- `medium`
- `high`

Using the priority allows granular alerting and improves the global situation visibility of the environment within the main screens.

See *Priority management* for more details about this feature

Week days monitoring:

Week days monitoring allows using specific rules for data sources and data hosts regarding the day of the week, by default monitoring rules are always applied, therefore using week days rules allow influencing the `red` state depending on the current day of the week. (which would switch to `orange` accordingly)

See *Week days monitoring* for more details about this feature

Monitoring level:

This option allows you to ask TrackMe to consider the very last events available at the index level rather than the specific sourcetype related to the entity.

This influences the state definition:

- If a data source or host is set to `sourcetype`, what conditions the state is meeting the monitoring rules for that sourcetype only (default behaviour)
- If it is set to `index`, instead of defining a red state because the monitoring conditions are not met, we will consider if there are events available at the index level according to the monitoring rules
- The purpose of this feature is to allow interacting with this data source (in that context let's talk about sourcetypes) without generating an alert as long as data is actively sent to that index

Associate to a logical group:

This option allows grouping data hosts and metric hosts into logical groups which are taken in consideration by groups rather than per entity.

See *Logical groups (clusters)* for more details about this feature.

Alerting policy: (data hosts only)

This option allows controlling on a per host basis the behaviour regarding the sourcetypes monitoring per host.

See *Alerting policy for data hosts* for more details about this feature.

Host distinct count threshold: (data sources only)

In some cases, you may want to be alerted when the number of distinct count hosts underneath a data source goes below a certain threshold.

Expected values are:

- "any" (default) which disables any verification against the hosts distinct count number
- A positive integer representing the minimal threshold for the dcount of hosts, if the current dcount goes below this value, the data source turns red

3.1.3 Elastic sources

Introduction to Elastic sources

Elastic sources feature

- The Elastic sources feature provides a builtin workflow to create virtual data sources based on any constraints and any Splunk language
- This extends TrackMe builtin features to allow dealing with any use case that the default data source concept does not cover by design
- Elastic Sources can be based on `tstats`, `raw`, `from (datamodel and lookup)` and `mstats` searches

- In addition, Elastic Sources can be executed over a `rest` remote query which allows tracking data that the search head(s) hosting TrackMe cannot access otherwise (such as a lookup that is only available to a Search Head Cluster while you run TrackMe on a monitoring utility search head)

As we have exposed the main notions of TrackMe data discovery and tracking in *Main navigation tabs*, there can be various use cases that these concepts do not address properly, considering some facts:

- Breaking by index and sourcetype is not enough, for instance your data pipeline can be distinguished in the same sourcetype by breaking on the `Splunk source Metadata`
- In a similar context, enrichment is performed either at indexing time (ideally indexed fields which allow the usage of tstats) or search time fields (evaluations, lookups, etc), these fields represent the keys you need to break on to address your requirements
- With the default `data sources` tracking, this data flow will appear as one main entity and you cannot `distinguish` a specific part of your data covered by the standard data source feature
- Specific `custom indexed fields` provide `knowledge` of the data in your context, such as `company`, `business unit` etc and these pipelines cannot be distinguished by relying on the `index` and `sourcetype` only
- You need to address any use case that the default main features do not allow you to

Hint: The Elastic source feature allows you to fulfil any type of requirements from the data identification and search perspective, and transparenly integrate these virtual entities in the normal TrackMe workflow with the exact same features.

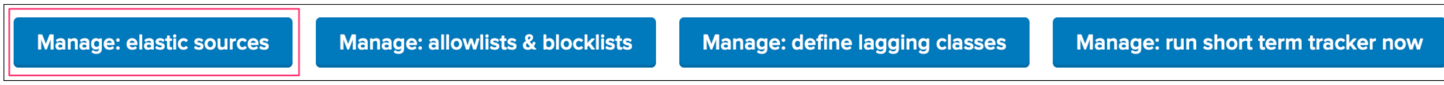
The concept of "Elastic Sources" is proper to TrackMe, and is linked to the complete level of flexibility the feature provides you to address any kind of use cases you might need to deal with.

In a nutshell:

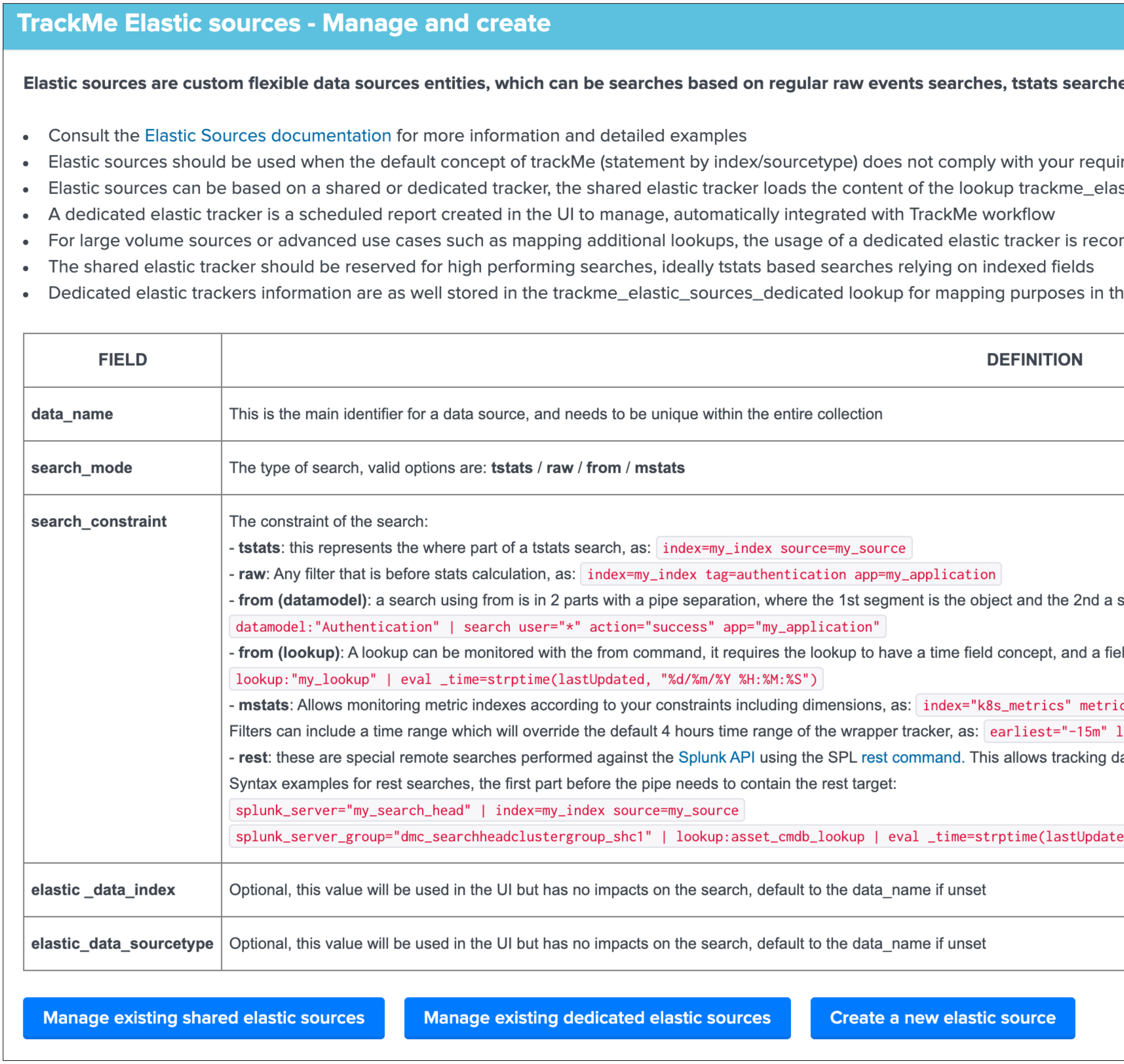
- An Elastic source can be added to the `shared tracker`, or created as an `independent tracker`
- The search language can be based on `| tstats`, `raw` searches, `| from` and `| mstats` commands
- Additionally, these searches can be run remotely over the Splunk rest API to address use cases where the data is not accessible to the search head(s) hosting TrackMe
- The shared tracker is a specific scheduled report named `TrackMe - Elastic sources shared tracker` that tracks in a single schedule execution all the entities that have been declared as shared Elastic sources via the UI
- Because the `shared tracker` performs a `single execution`, there are performance considerations to take into account and the shared tracker should be restricted to very efficient searches in term of run time
- In addition, `Elastic sources shared` have time frame restrictions which are the earliest and latest values of the tracker, you can restrict a shared entity time scope below these values but not beyond
- A `dedicated Elastic source` is created via the UI which generates a new tracker especially for it
- As the dedicated Elastic source has its `own schedule report`, this provides more capabilities to handle fewer performing searches and as well more freedom to address basically any kind of customisation
- `Dedicated Elastic sources` can be configured to address any time scope you need, and any search that is required including any advanced customisation you would need

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:


Manage: elastic sources
Manage: allowlists & blocklists
Manage: define lagging classes
Manage: run short term tracker now

The following screen appears:



TrackMe Elastic sources - Manage and create

Elastic sources are custom flexible data sources entities, which can be searches based on regular raw events searches, tstats searche

- Consult the Elastic Sources documentation for more information and detailed examples
- Elastic sources should be used when the default concept of trackMe (statement by index/sourcetype) does not comply with your requir
- Elastic sources can be based on a shared or dedicated tracker, the shared elastic tracker loads the content of the lookup trackme_elas
- A dedicated elastic tracker is a scheduled report created in the UI to manage, automatically integrated with TrackMe workflow
- For large volume sources or advanced use cases such as mapping additional lookups, the usage of a dedicated elastic tracker is recor
- The shared elastic tracker should be reserved for high performing searches, ideally tstats based searches relying on indexed fields
- Dedicated elastic trackers information are as well stored in the trackme_elastic_sources_dedicated lookup for mapping purposes in th

<table>
<tr><th>FIELD</th><th>DEFINITION</th></tr>
<tr><td><b>data_name</b></td><td>This is the main identifier for a data source, and needs to be unique within the entire collection</td></tr>
<tr><td><b>search_mode</b></td><td>The type of search, valid options are: <b>tstats</b> / <b>raw</b> / <b>from</b> / <b>mstats</b></td></tr>
<tr><td><b>search_constraint</b></td><td>The constraint of the search:
- <b>tstats</b>: this represents the where part of a tstats search, as: <code>index=my_index source=my_source</code>
- <b>raw</b>: Any filter that is before stats calculation, as: <code>index=my_index tag=authentication app=my_application</code>
- <b>from (datamodel)</b>: a search using from is in 2 parts with a pipe separation, where the 1st segment is the object and the 2nd a s
<code>datamodel:"Authentication" | search user="*" action="success" app="my_application"</code>
- <b>from (lookup)</b>: A lookup can be monitored with the from command, it requires the lookup to have a time field concept, and a fiel
<code>lookup:"my_lookup" | eval _time=strptime(lastUpdated, "%d/%m/%Y %H:%M:%S")</code>
- <b>mstats</b>: Allows monitoring metric indexes according to your constraints including dimensions, as: <code>index="k8s_metrics" metric</code>
Filters can include a time range which will override the default 4 hours time range of the wrapper tracker, as: <code>earliest="-15m" l</code>
- <b>rest</b>: these are special remote searches performed against the Splunk API using the SPL rest command. This allows tracking da
Syntax examples for rest searches, the first part before the pipe needs to contain the rest target:
<code>splunk_server="my_search_head" | index=my_index source=my_source</code>
<code>splunk_server_group="dmc_searchheadclustergroup_shc1" | lookup:asset_cmdb_lookup | eval _time=strptime(lastUpdate</code></td></tr>
<tr><td><b>elastic _data_index</b></td><td>Optional, this value will be used in the UI but has no impacts on the search, default to the data_name if unset</td></tr>
<tr><td><b>elastic_data_sourcetype</b></td><td>Optional, this value will be used in the UI but has no impacts on the search, default to the data_name if unset</td></tr>
</table>

Manage existing shared elastic sources
Manage existing dedicated elastic sources
Create a new elastic source

Elastic source example 1: source Metadata

Let's take our first example, assuming we are indexing the following events:

data flow1 : firewall traffic for the region AMER

```
index="network" sourcetype="pan:traffic" source="network:pan:amer"
```

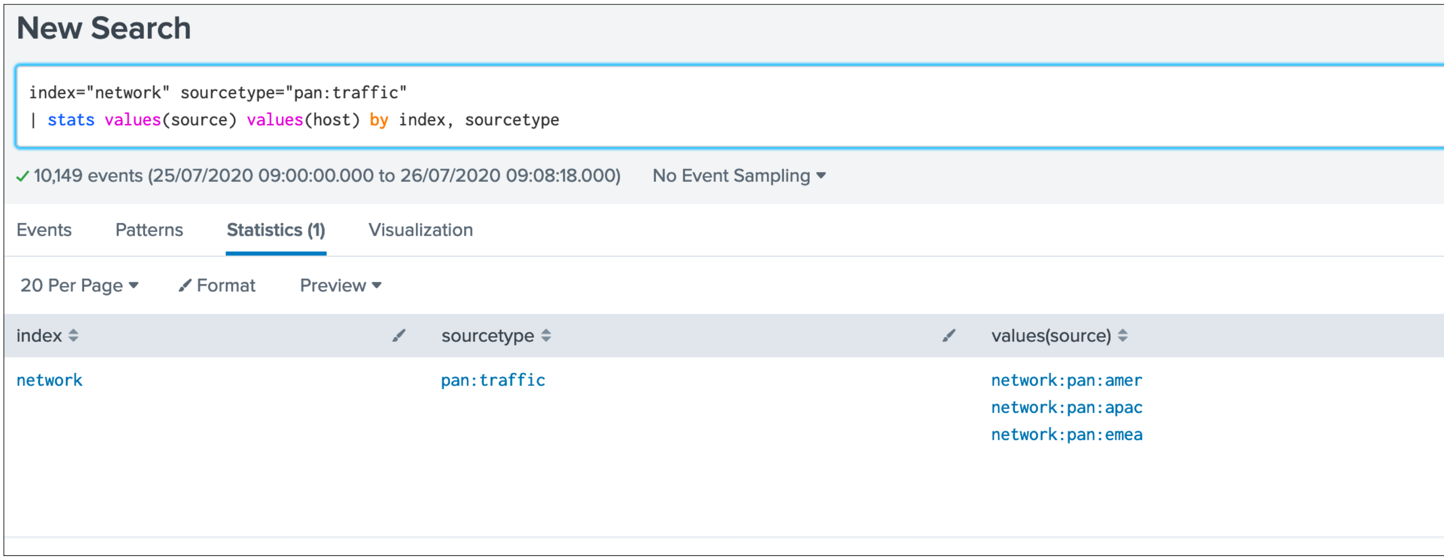
data flow2 : firewall traffic for the region APAC

```
index="network" sourcetype="pan:traffic" source="network:pan:apac"
```

data flow3 : firewall traffic for the region EMEA

```
index="network" sourcetype="pan:traffic" source="network:pan:emea"
```

It is easy to understand that the default standard for data source `index + ":" + sourcetype` does not allow us to distinguish which region is generating events properly, and which region would not:



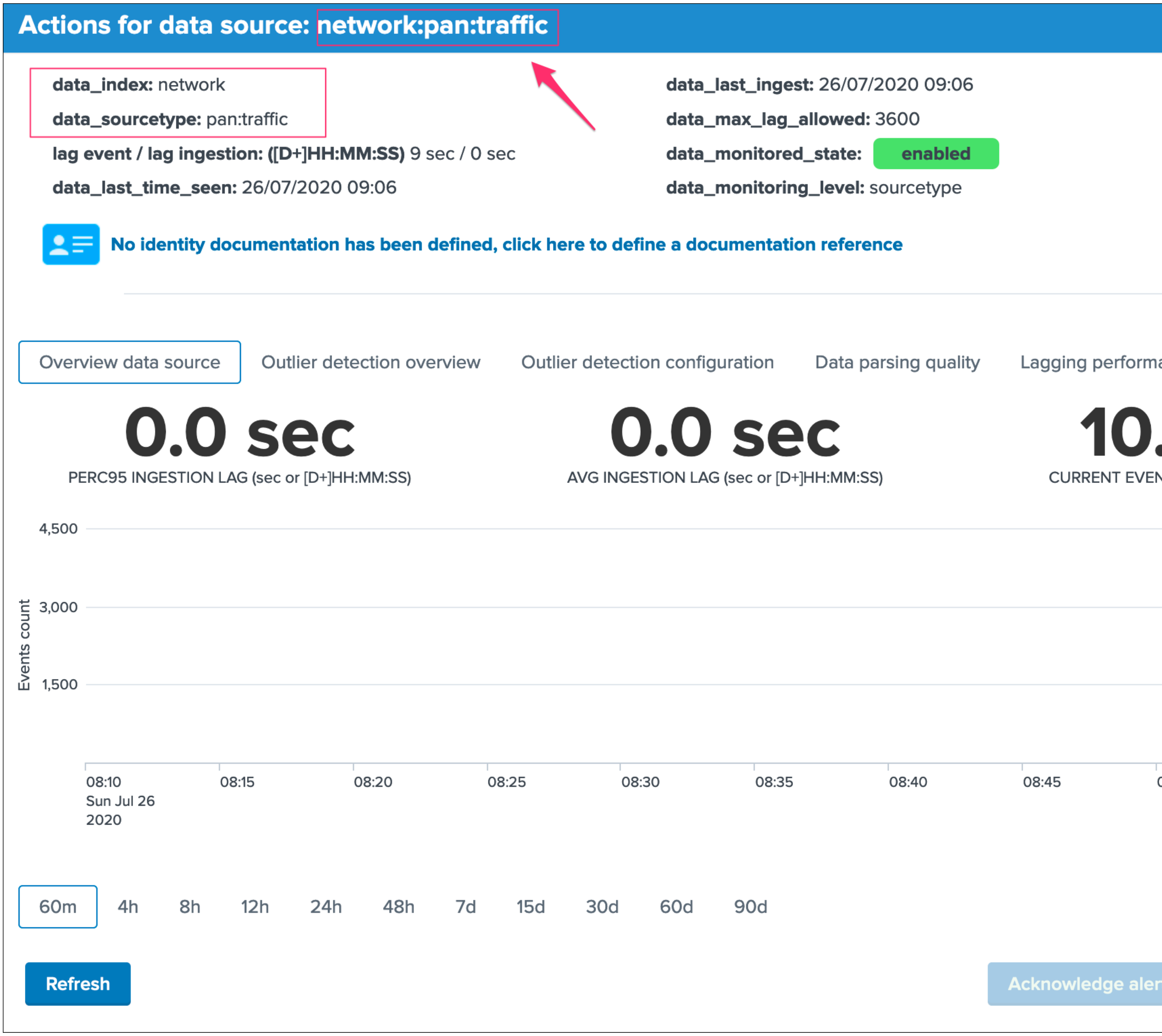
New Search

```
index="network" sourcetype="pan:traffic"
| stats values(source) values(host) by index, sourcetype
```

✓ 10,149 events (25/07/2020 09:00:00.000 to 26/07/2020 09:08:18.000)   No Event Sampling ▾

index	sourcetype	values(source)
network	pan:traffic	network:pan:amer network:pan:apac network:pan:emea

In TrackMe data sources, this would appear as one entity and this is not helping me covering that use case:



Actions for data source: network:pan:traffic

data_index: network

data_sourcetype: pan:traffic

lag event / lag ingestion: ([D+]HH:MM:SS) 9 sec / 0 sec

data_last_time_seen: 26/07/2020 09:06

data_last_ingest: 26/07/2020 09:06

data_max_lag_allowed: 3600

data_monitored_state: enabled

data_monitoring_level: sourcetype

No identity documentation has been defined, click here to define a documentation reference

What if I want to be monitoring the fact that the EMEA region continues to be indexed properly ? and other regions ?

Elastic Sources is the TrackMe answer which allows you to extend the default features with agility and address easily any kind of requirement transparently in TrackMe.

Elastic source example 2: custom indexed fields

Let’s extend a bit more the first example, and this time in addition with the region we have a company notion.

At indexing time, two custom indexed fields are created representing the “region” and the “company”.

Custon indexed fields can be created in many ways in Splunk, it is a great and powerful feature as long as it is properly implemented and restricted to the right use cases.

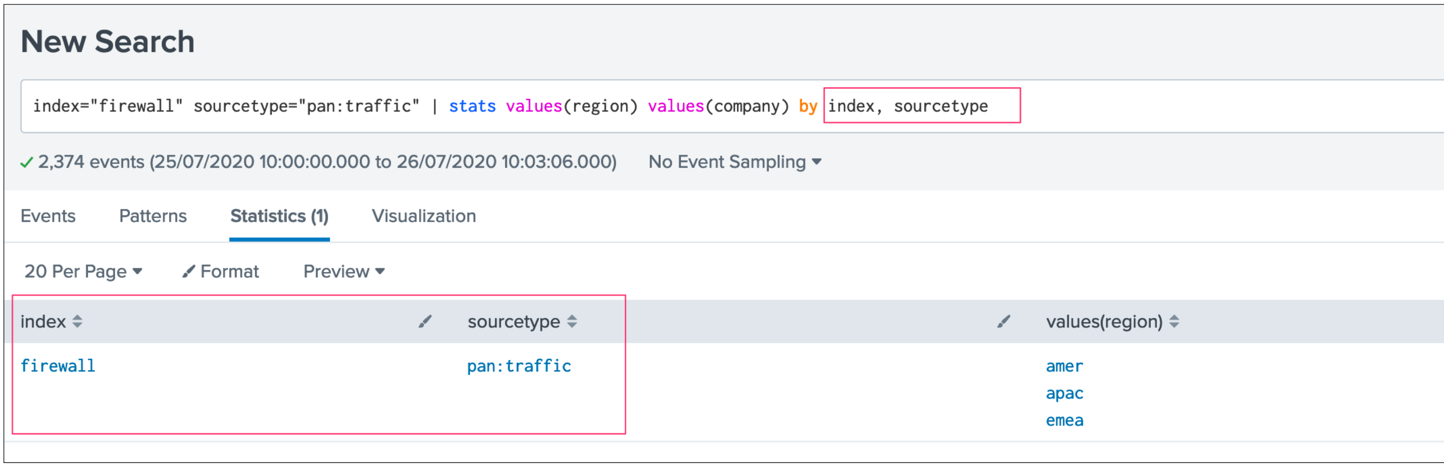
This example of excellence allows our virtual customer to work at scale with performing searches against their two major enrichment fields.

Assuming we have 3 regions (AMER / EMEA / APAC) and per region we have two companies (design / retail), to get the data of each region / company I need several searches:

```
index="firewall" sourcetype="pan:traffic" region::amer company::design
index="firewall" sourcetype="pan:traffic" region::amer company::retail
index="firewall" sourcetype="pan:traffic" region::apac company::design
index="firewall" sourcetype="pan:traffic" region::apac company::retail
index="firewall" sourcetype="pan:traffic" region::emea company::design
index="firewall" sourcetype="pan:traffic" region::emea company::retail
```

Note the usage of "::" rather than "=" which indicates to Splunk that we are explicitly looking at an indexed field rather a field potentially extracted at search time.

Indeed, it is clear enough that the default data source feature does not me with the answer I need for this use case:



New Search

```
index="firewall" sourcetype="pan:traffic" | stats values(region) values(company) by index, sourcetype
```

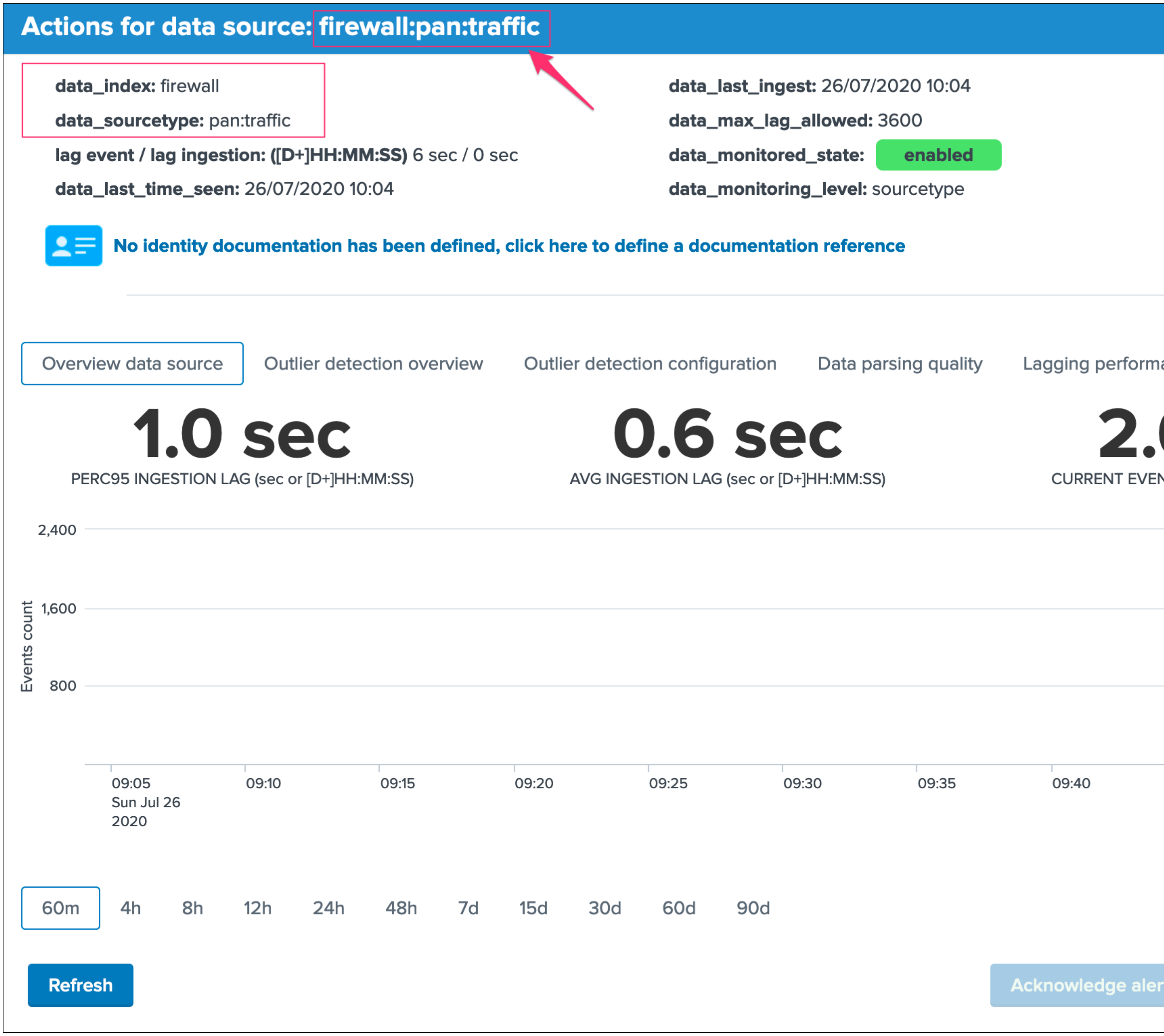
✓ 2,374 events (25/07/2020 10:00:00.000 to 26/07/2020 10:03:06.000)

index	sourcetype	values(region)
firewall	pan:traffic	amer apac emea

Rather than one data source that covers the index/sourcetype, the requirement is to have 6 data sources that cover each couple of region/company.

Any failure on the flow level which is represented by these new data sources will be detected. On the opposite, the default data source breaking on on the sourcetype would need a total failure of all pipelines to be detected.

By default, the data source would show up with a unique entity which is not filling my requirements:



Actions for data source: firewall:pan:traffic

data_index: firewall
data_sourcetype: pan:traffic
lag event / lag ingestion: ([D+]HH:MM:SS) 6 sec / 0 sec
data_last_time_seen: 26/07/2020 10:04

data_last_ingest: 26/07/2020 10:04
data_max_lag_allowed: 3600
data_monitored_state: enabled
data_monitoring_level: sourcetype

No identity documentation has been defined, click here to define a documentation reference

The default concept while powerful does not cover my need, but ok there we go and let's extend it easily with Elastic sources!

Elastic source example 3: tracking lookups update and number of records

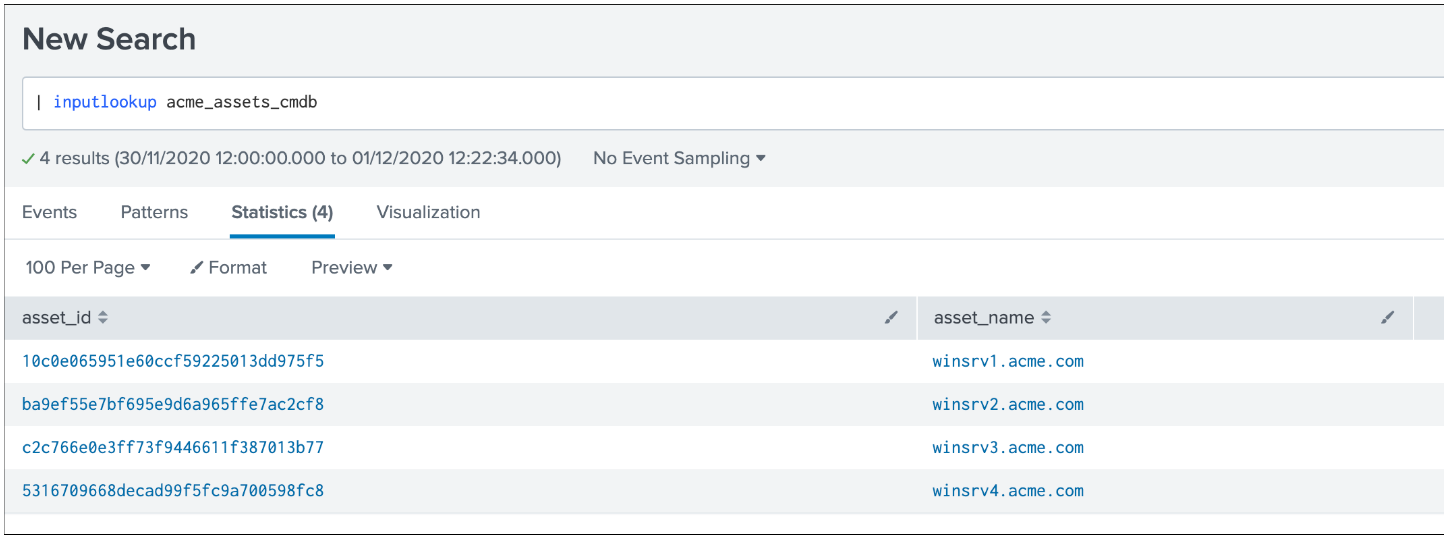
It is a very common and powerful practice to generate and maintain lookups in Splunk for numbers of purposes, which can be file based lookups (CSV files) or KVstore based lookups.

Starting with TrackMe 1.2.28, it is possible to define an Elastic Source and monitor if the lookup is being updated as expected.

A common caveheat with lookups is that their update is driven by Splunk searches, there are plenty of reasons why a lookup could stop being populated and maintained, such as scheduling issues, permissions, related knowledge objects updates, lack or changes in the data, and many more.

The purpose of this example is to provide a builtin and efficient way of tracking Splunk lookup updates at scale in the easy way, and get alerted if an update issue is detected in the lookup according to the policies defined in TrackMe.

Let's consider the simplistic following example, the lookup acme_assets_cmdb contains our ACME assets and is updated every day, we record in the field "lookupLastUpdated" the date and time of the execution of the Lookup gen report in Splunk. (in epoch time format)



New Search

```
| inputlookup acme_assets_cmdb
```

✓ 4 results (30/11/2020 12:00:00.000 to 01/12/2020 12:22:34.000)

asset_id	asset_name
10c0e065951e60ccf59225013dd975f5	winsrv1.acme.com
ba9ef55e7bf695e9d6a965ffe7ac2cf8	winsrv2.acme.com
c2c766e0e3ff73f9446611f387013b77	winsrv3.acme.com
5316709668decad99f5fc9a700598fc8	winsrv4.acme.com

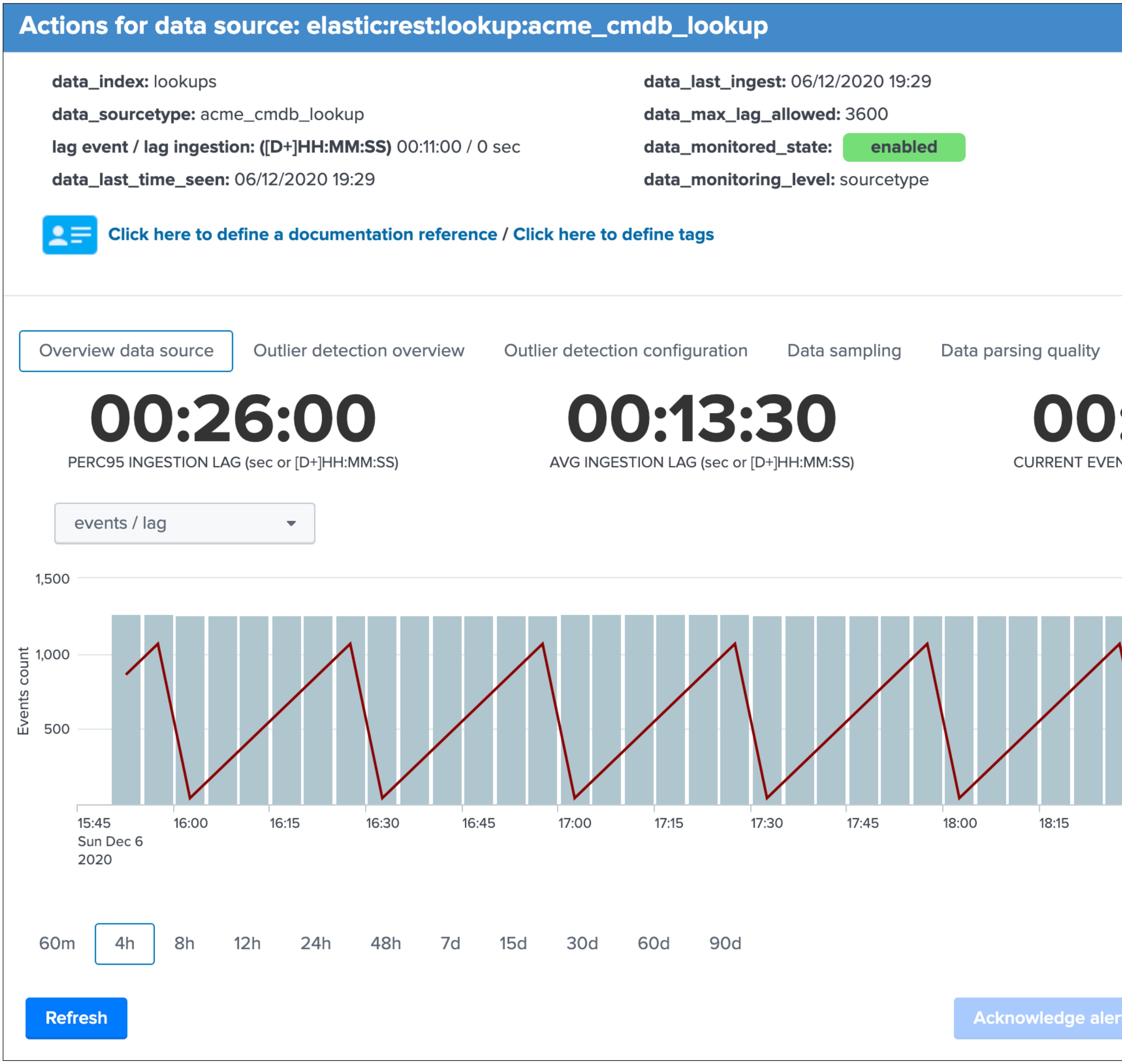
The unique requirement for TrackMe to be able to monitor a lookup is to have a time concept which can use to define as the `_time` field which TrackMe will rely on.

Lookups have no such thing of a concept of `_indextime` (time of ingestion in Splunk), therefore TrackMe will by default make the index time equivalent to the latest _time from the lookup, unless the Splunk search that will be set in the Elastic Source defines a value based on information from the lookup.

Monitoring lookups with TrackMe allow you to:

- Get automatically alerted when the last update of the lookup is older than a given amount of time (which could indicate an issue on the execution side, such as an error introduced in the SPL code maintaining the lookup, a knowledge object that is missing, etc)
- Monitor and track the number of records, the outliers detection will automatically monitor the number of records in the lookup (which outliers settings can be fine tuned up to your needs, you could even gets alerted if the number of records goes beyond a certain limit)

The following example shows the behaviour with a lookup that is updated every 30 minutes:



Actions for data source: elastic:rest:lookup:acme_cmdb_lookup

data_index: lookups

data_sourcetype: acme_cmdb_lookup

lag event / lag ingestion: ([D+]HH:MM:SS) 00:11:00 / 0 sec

data_last_time_seen: 06/12/2020 19:29

data_last_ingest: 06/12/2020 19:29

data_max_lag_allowed: 3600

data_monitored_state: enabled

data_monitoring_level: sourcetype

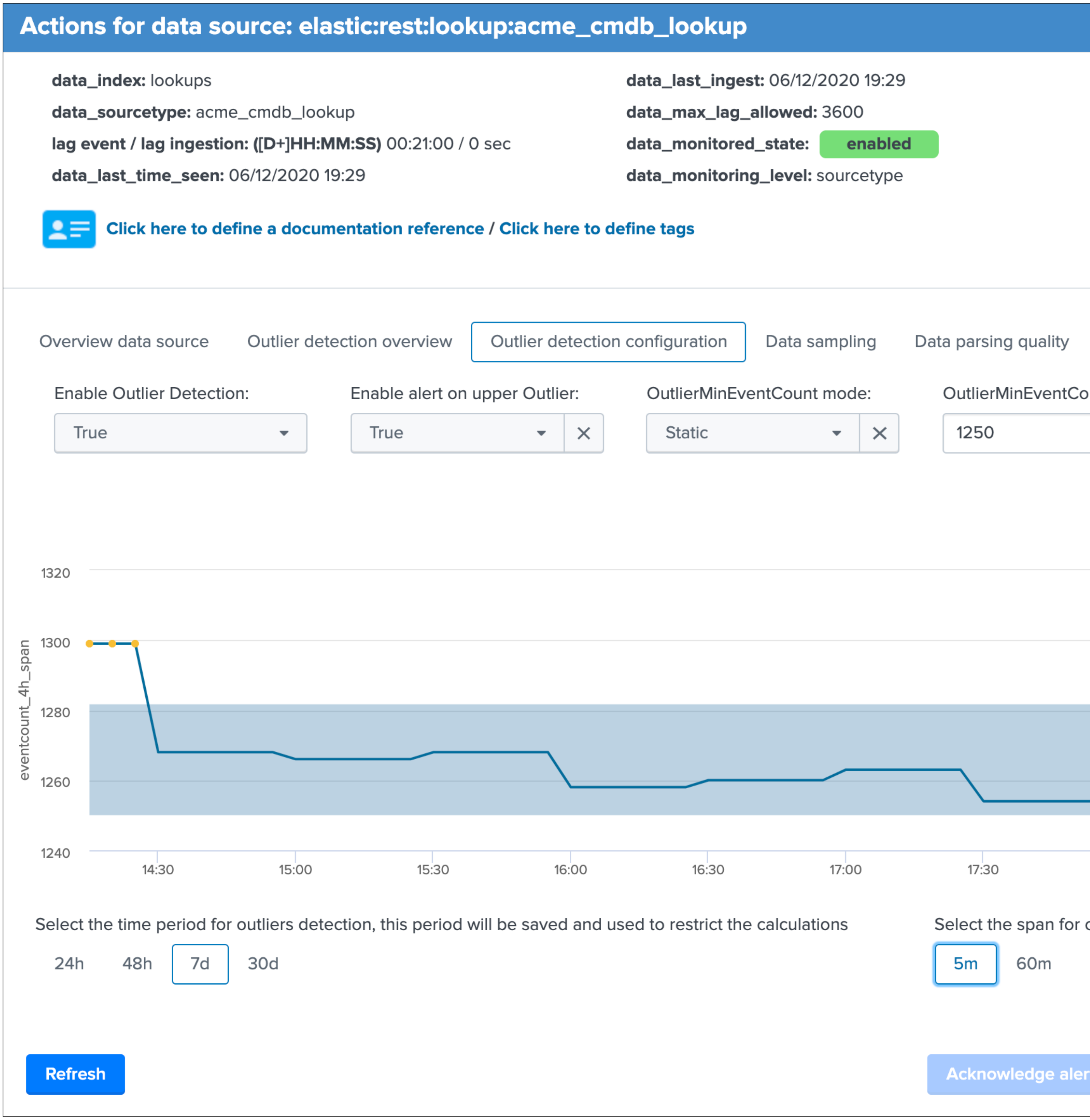
Click here to define a documentation reference / Click here to define tags

00:26:00 PERC95 INGESTION LAG (sec or [D+]HH:MM:SS)

00:13:30 AVG INGESTION LAG (sec or [D+]HH:MM:SS)

00: CURRENT EVEN

Number of records are monitored automatically by the outliers detection, setting can be fined tuned to alert if the number of records goes below, and/or beyond a certain amount of records:



Actions for data source: elastic:rest:lookup:acme_cmdb_lookup

data_index: lookups

data_sourcetype: acme_cmdb_lookup

lag event / lag ingestion: ([D+]HH:MM:SS) 00:21:00 / 0 sec

data_last_time_seen: 06/12/2020 19:29

data_last_ingest: 06/12/2020 19:29

data_max_lag_allowed: 3600

data_monitored_state: enabled

data_monitoring_level: sourcetype

Click here to define a documentation reference / Click here to define tags

Enable Outlier Detection: True

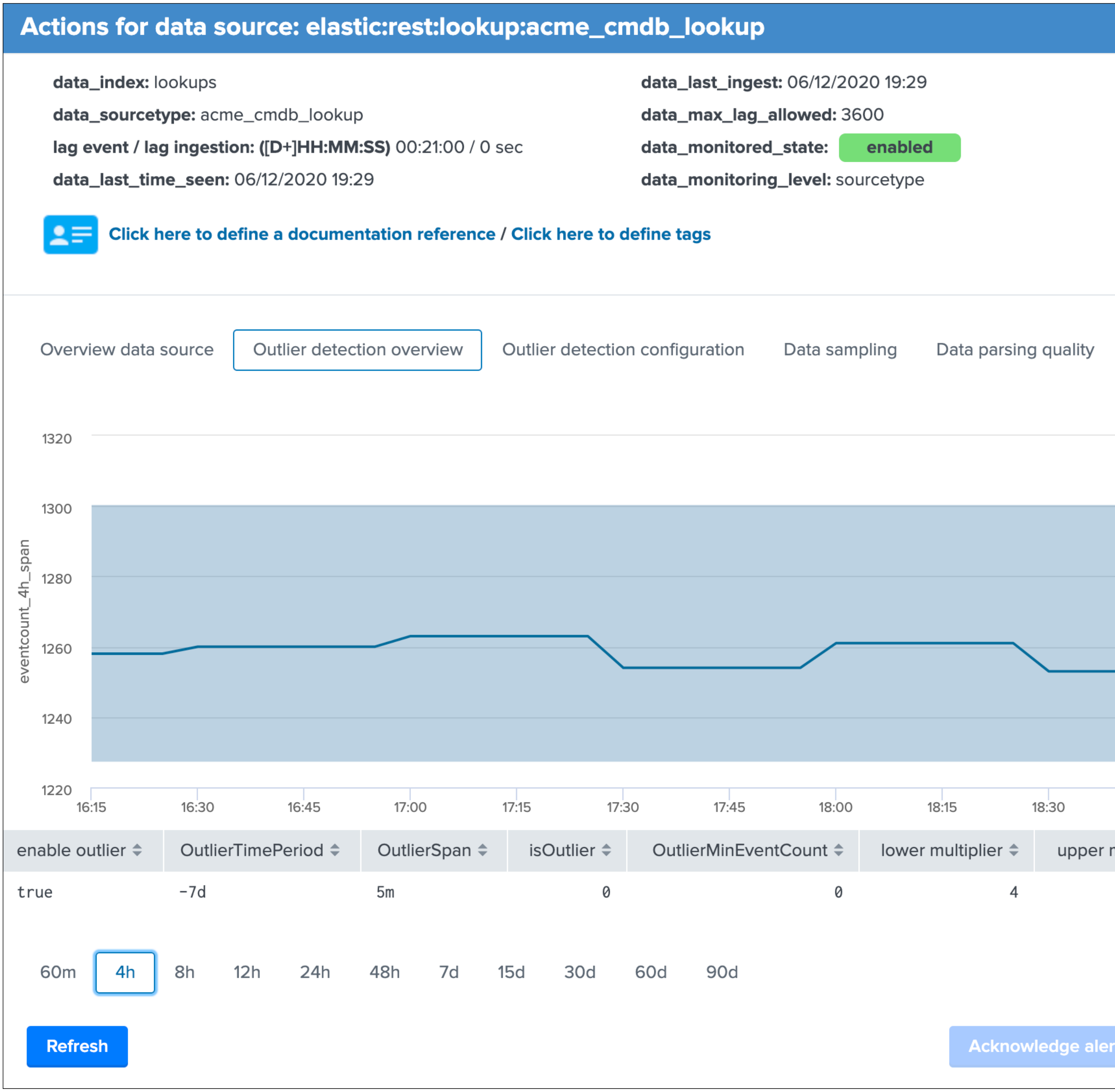
Enable alert on upper Outlier: True

OutlierMinEventCount mode: Static

OutlierMinEventCo 1250

Select the time period for outliers detection, this period will be saved and used to restrict the calculations

Select the span for c



Actions for data source: elastic:rest:lookup:acme_cmdb_lookup

data_index: lookups

data_sourcetype: acme_cmdb_lookup

lag event / lag ingestion: ([D+]HH:MM:SS) 00:21:00 / 0 sec

data_last_time_seen: 06/12/2020 19:29

data_last_ingest: 06/12/2020 19:29

data_max_lag_allowed: 3600

data_monitored_state: enabled

data_monitoring_level: sourcetype

Click here to define a documentation reference / Click here to define tags

enable outlier	OutlierTimePeriod	OutlierSpan	isOutlier	OutlierMinEventCount	lower multiplier	upper r
true	-7d	5m	0	0	4	

Elastic source example 4: rest searches

In some cases, the Splunk instance that hosts the TrackMe application may not not be able to access to a data you wish to monitor.

A very simple to understand use case would be:

- You have a Splunk Search Head Cluster, hosting for example your premium application for ITSI or Enterprise Security

- In addition, you either use your monitoring console host or a dedicated standalone search head for your Splunk environment monitoring, which is where TrackMe is deployed
- A lookup exists in the SHC which is the object you need to monitor, this lookup is only available to the SHC members and TrackMe cannot access to its content transparently

Using a `rest` command, you can hit a Splunk API search endpoint remotely, and use the builtin Elastic Source feature to monitor and track the lookup just as if it was available directly on the TrackMe search head.

In short, on the SHC you can run:

```
| inputlookup acme_assets_cmdb
```

On the TrackMe Splunk instance, we will use a search looking like:

```
| rest splunk_server_group="dmc_searchheadclustergroup_shc1" /servicesNS/admin/search/
↪search/jobs/export search="| from lookup:acme_assets_cmdb | eval _
↪time=strftime(lookupLastUpdated, \"%s\") | 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 data_name=\
↪"rest:from:lookup:example\", data_index=\"pseudo_index\", data_sourcetype=\
↪"lookup:acme_assets_cmdb\", data_last_ingestion_lag_seen=data_last_ingest-data_last_
↪time_seen" output_mode="csv"
```

Notes and technical details:

- See https://docs.splunk.com/Documentation/Splunk/latest/RESTTUT/RESTsearches for more information about running searches over rest
- See https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/Rest for more information about the rest search command
- `rest` based searches support all forms of searches supported by Elastic Sources: `tstats`, `raw`, `from:datamodel`, `from:lookup`, `mstats`
- Search Heads you wish to target need to be configured as distributed search peers in Splunk, same requirement as for the Splunk Monitoring Console host (MC, previously named DMC)
- Most of the calculation part is executed on the target search head size, TrackMe will not attempt to retrieve the raw data first before performing the calculation for obvious performance gain purposes
- You can target a search head explicity using the `splunk_server` argument, or you can target a group of search heads (such as your SHC) using the `splunk_server_group` argument
- When targeting a group of search heads, the query is executed on every search that is matched by the splunk_server_group, therefore you should limit using a target group to very efficient and low cost searches such as a from lookup for example
- TrackMe in anycase will only consider the first result from the rest command (so only one search head answer during the rest execution, assuming search heads from the same group have the same data access), and will discard other search head replies
- The search needs to be properly performing, and should complete in a acceptable time window (use timeout argument which defaults to 60 seconds)
- Each result from the rest command, during the tracker execution or within the UI, passes through a Python based custom command to parse the CSV structure resulting from the rest command, to finally create the Splunk events during the search time execution

- Except for `| from lookup:` rest searches, other types of searches automatically append the configured earliest and latest as arguments to the rest command (earliest_time, latest_time)
- Earliest and Latest arguments are configurable for dedicated trackers only, shared trackers will use earliest:"-4h" and latest:"+4h" statically
- Additional parameters to the rest command can be added within the first pipe of the search constraint during the Elastic Source creation (such as timeout, count etc)

> **Warning:** Currently the rest command generates a warning message "Unable to determine response format from HTTP Header", this message can be safety ignored as it does not impact the results in anyway, but cannot unfortunately be removed at the moment, until it is fixed by Splunk.

Examples for each type of search:

tstats over rest:

```
splunk_server="my_search_head" | index=* sourcetype=pan:traffic
```

raw search over rest:

```
splunk_server="my_search_head" | index=* sourcetype=pan:traffic
```

from datamodel over rest:

```
splunk_server="my_search_head" | datamodel:"Authentication" action=*
```

from lookup over rest:

```
splunk_server="my_search_head" | from lookup:acme_assets_cmdb | eval _
↪time=strftime(lookupLastUpdated, "%s")
```

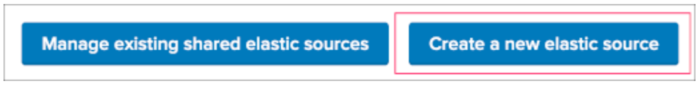
mstats over rest:

```
splunk_server="my_search_head" | index=* metric_name=docker*
```

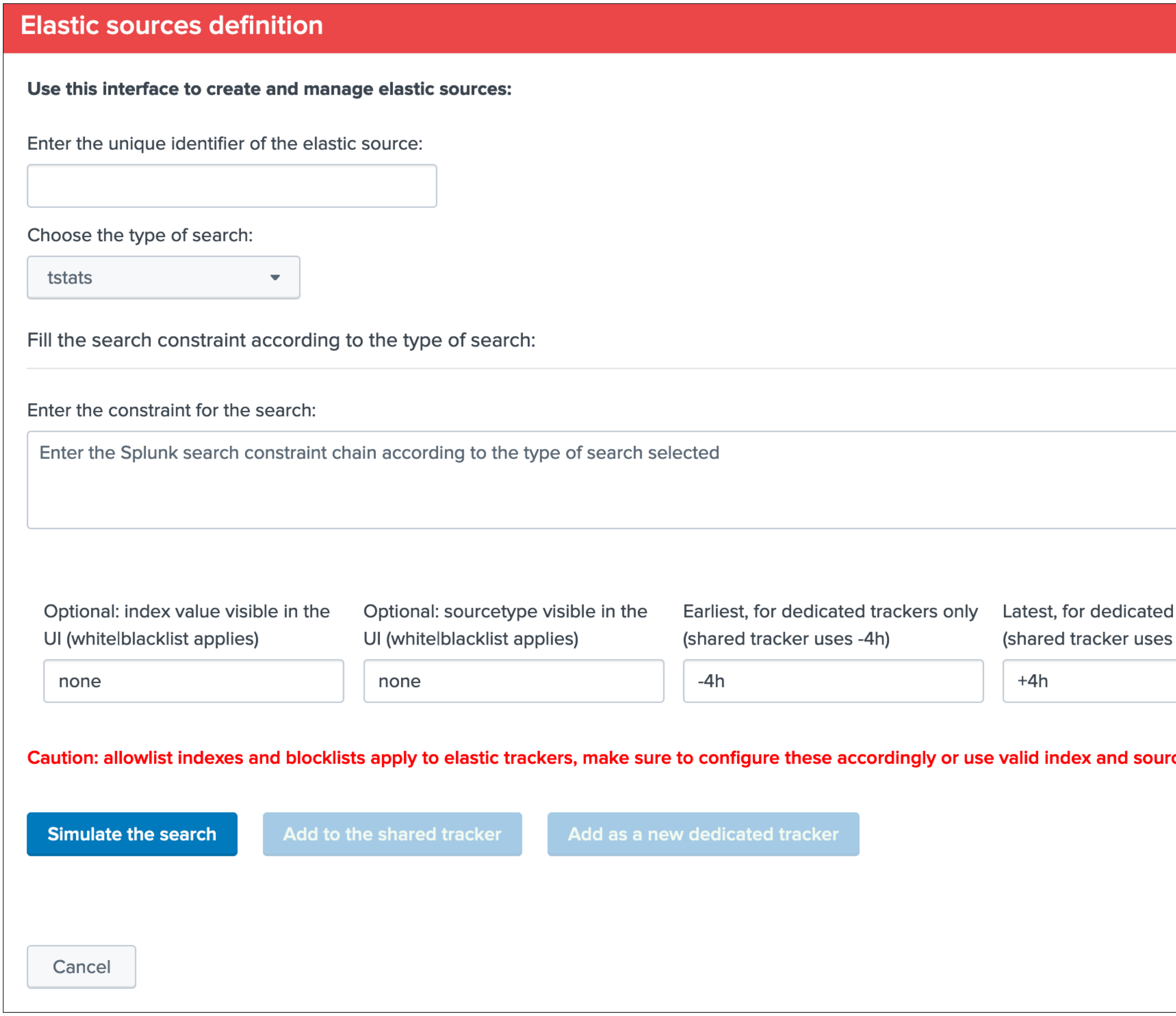
As a conclusion, using the rest based searches features successfully completes the Elastic Sources level of features, such that every single use case can be handled in TrackMe, whenever the Splunk instance cam access or not to the data you need to track!

Elastic source example 1: creation

Now, let's create our first Elastic Source which will meet our requirement to rely on the Splunk source Metadata, click on create a new Elastic source:


Manage existing shared elastic sources
Create a new elastic source

Which opens the following screen:



Elastic sources definition

Use this interface to create and manage elastic sources:

Enter the unique identifier of the elastic source:

Choose the type of search: tstats

Fill the search constraint according to the type of search:

Enter the constraint for the search: Enter the Splunk search constraint chain according to the type of search selected

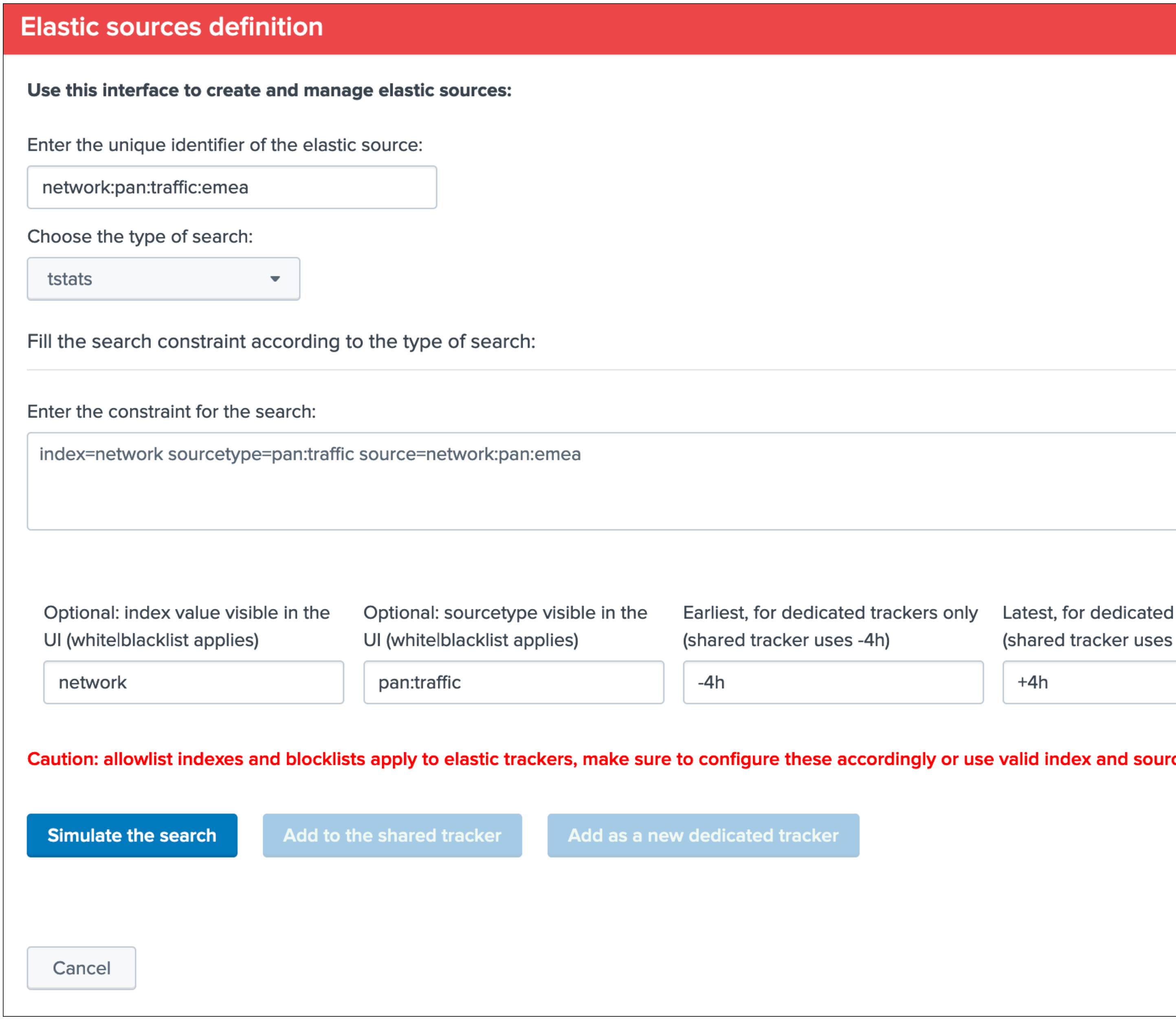
Optional: index value visible in the UI (white|blacklist applies)	Optional: sourcetype visible in the UI (white|blacklist applies)	Earliest, for dedicated trackers only (shared tracker uses -4h)	Latest, for dedicated (shared tracker uses
none	none	-4h	+4h

Caution: allowlist indexes and blocklists apply to elastic trackers, make sure to configure these accordingly or use valid index and sourc

Summary:

- Define a name for the entity, this name is the value of the field `data_name` and needs to be unique in TrackMe
- Shall that name you provide not be unique, a little red cross and a message will indicate the issue when we run the simulation
- We choose a `search language`, because the source field is a Metadata, this is an indexed field and we can use the tstats command which is very efficient by looking at the tsdidx files rather than the raw events
- We define our search constraint for the first entity, in our case `index=network sourcetype=pan:traffic source=network:pan:emea`
- We choose a value for the index, this is having `no influence` on the search itself and its result but determines how the entity is classified and filtered in the main UI
- Same for the sourcetype, which does `not influence` the search results

- Finally, we can optionally decide to define the earliest and latest time range, in our example we can leave that empty and rely on the default behaviour



Elastic sources definition

Use this interface to create and manage elastic sources:

Enter the unique identifier of the elastic source:

network:pan:traffic:emea

Choose the type of search:

tstats

Fill the search constraint according to the type of search:

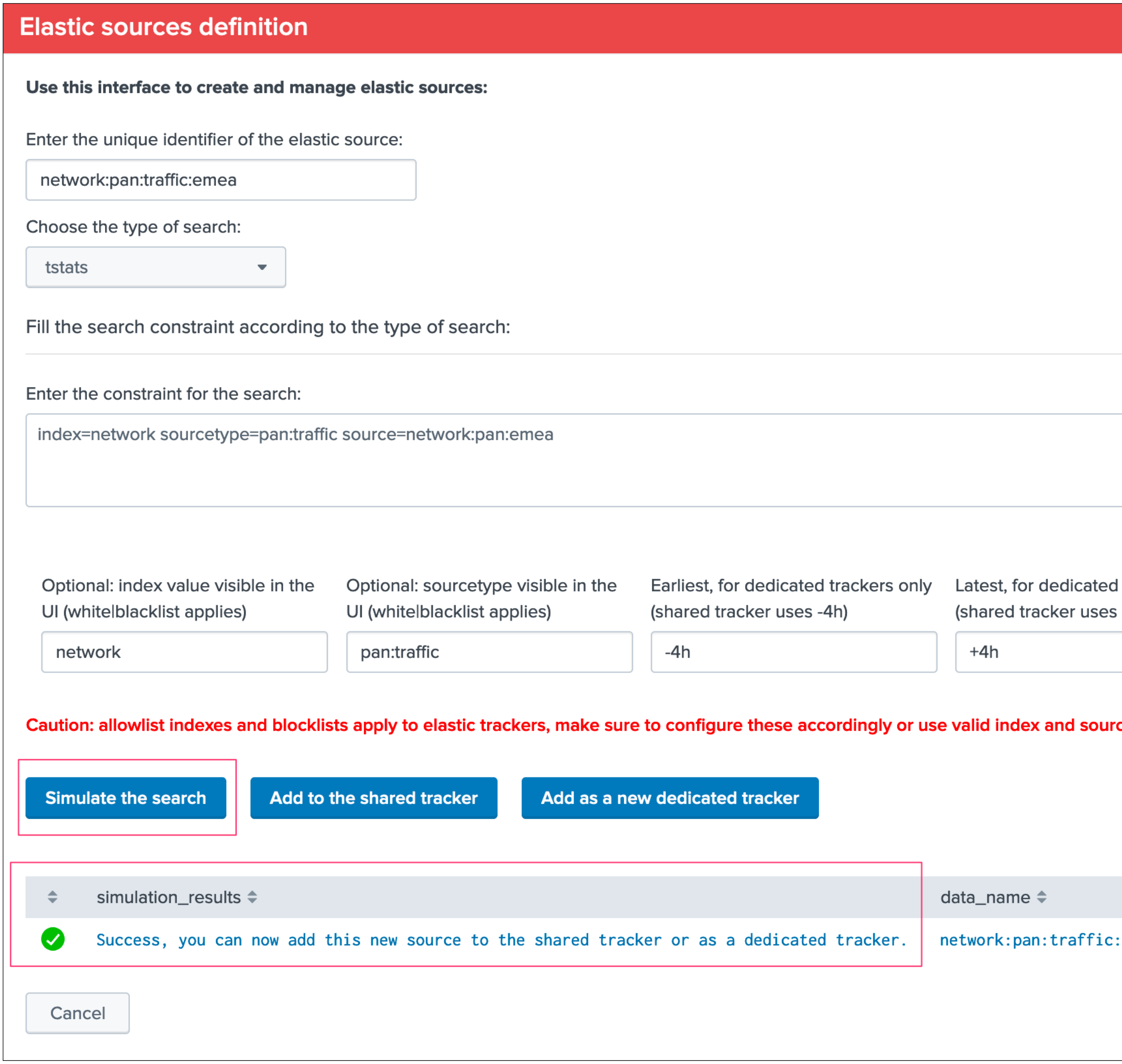
Enter the constraint for the search:

index=network sourcetype=pan:traffic source=network:pan:emea

Optional: index value visible in the UI (white|blacklist applies)	Optional: sourcetype visible in the UI (white|blacklist applies)	Earliest, for dedicated trackers only (shared tracker uses -4h)	Latest, for dedicated (shared tracker uses
network	pan:traffic	-4h	+4h

Caution: allowlist indexes and blocklists apply to elastic trackers, make sure to configure these accordingly or use valid index and sourc

Let's click on this nice button!



Elastic sources definition

Use this interface to create and manage elastic sources:

Enter the unique identifier of the elastic source: network:pan:traffic:emea

Choose the type of search: tstats

Fill the search constraint according to the type of search:

Enter the constraint for the search: index=network sourcetype=pan:traffic source=network:pan:emea

Optional: index value visible in the UI (white|blacklist applies): network

Optional: sourcetype visible in the UI (white|blacklist applies): pan:traffic

Earliest, for dedicated trackers only (shared tracker uses -4h): -4h

Latest, for dedicated (shared tracker uses: +4h

Caution: allowlist indexes and blocklists apply to elastic trackers, make sure to configure these accordingly or use valid index and sourc

	simulation_results	data_name
✓	Success, you can now add this new source to the shared tracker or as a dedicated tracker.	network:pan:traffic:

This looks good isn't it?

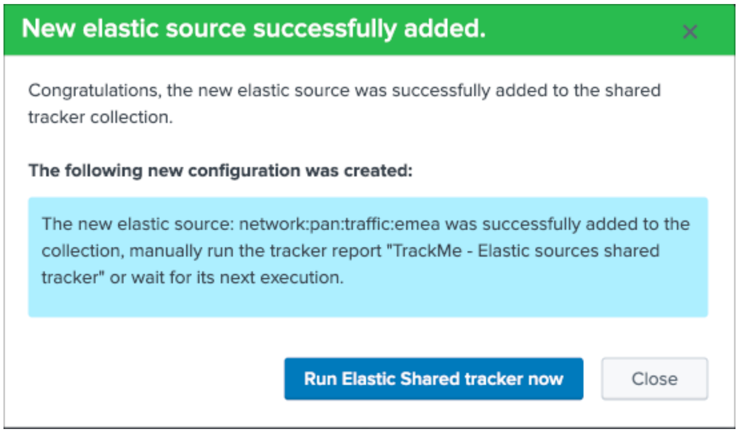
Shared tracker versus dedicated tracker:

In this context:

- Because this is a very efficient search that relies on tstats, creating it as a shared tracker is perfectly fair
- Shall I want to increase the earliest or the latest values beyond the shared tracker default of -4h / +4h, this would be reason to create a dedicated tracker
- While tstats searches are very efficient, a very high volume of events might mean a certain run time for the search, in such a case a dedicated tracker shall be used

- If you have to achieve any additional work, such as third party lookup enrichment, this would be a reason to create a dedicated tracker too

Fine? Let's cover both, and let's click on "Add to the shared tracker" button:



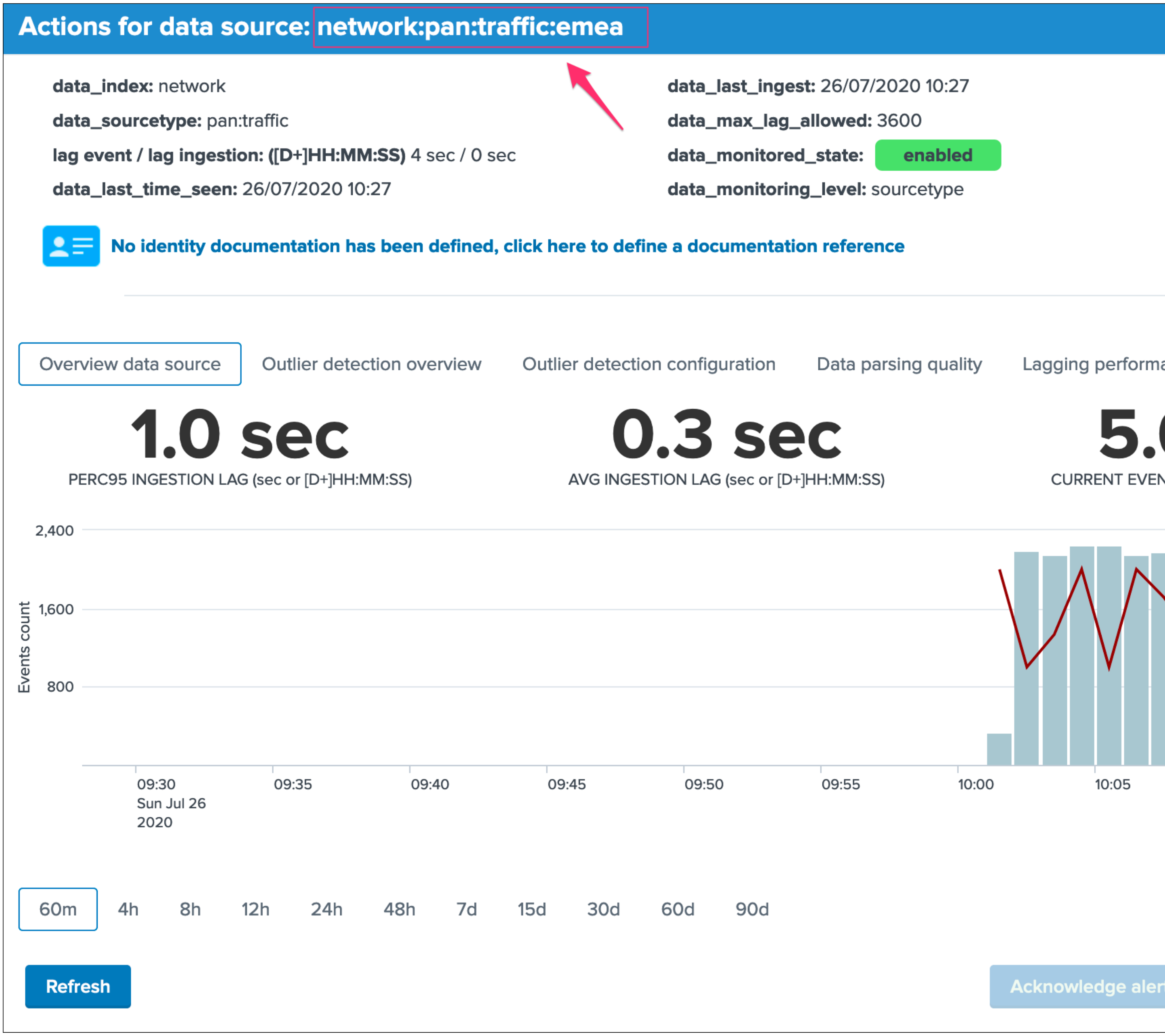
New elastic source successfully added.

Congratulations, the new elastic source was successfully added to the shared tracker collection.

The following new configuration was created:

The new elastic source: network:pan:traffic:emea was successfully added to the collection, manually run the tracker report "TrackMe - Elastic sources shared tracker" or wait for its next execution.

Nice! Let's click on that button and immediately run the shared tracker, upon its execution we can see an all brand new data source entity that matches what we created:



Actions for data source: network:pan:traffic:emea

data_index: network

data_sourcetype: pan:traffic

lag event / lag ingestion: ([D+]HH:MM:SS) 4 sec / 0 sec

data_last_time_seen: 26/07/2020 10:27

data_last_ingest: 26/07/2020 10:27

data_max_lag_allowed: 3600

data_monitored_state: enabled

data_monitoring_level: sourcetype

No identity documentation has been defined, click here to define a documentation reference

1.0 sec
PERC95 INGESTION LAG (sec or [D+]HH:MM:SS)

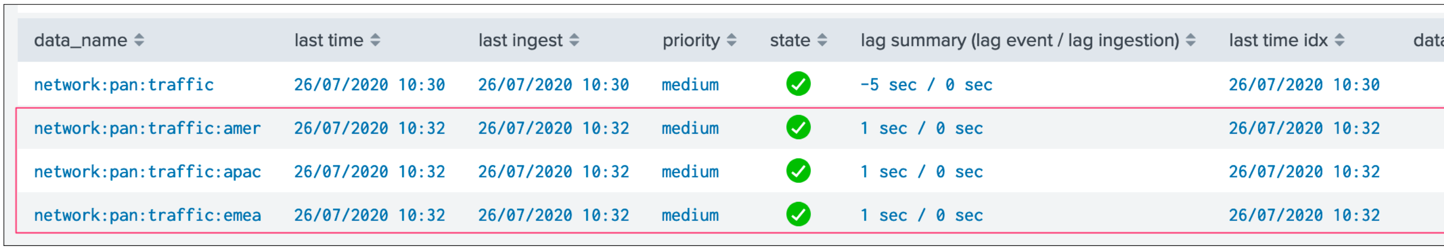
0.3 sec
AVG INGESTION LAG (sec or [D+]HH:MM:SS)

5.(
CURRENT EVEN

Ok that's cool!

Note: if you disagree with this statement, you are free to leave this site, free to uninstall TrackMe and create all of your own things we are not friends anymore that's it.

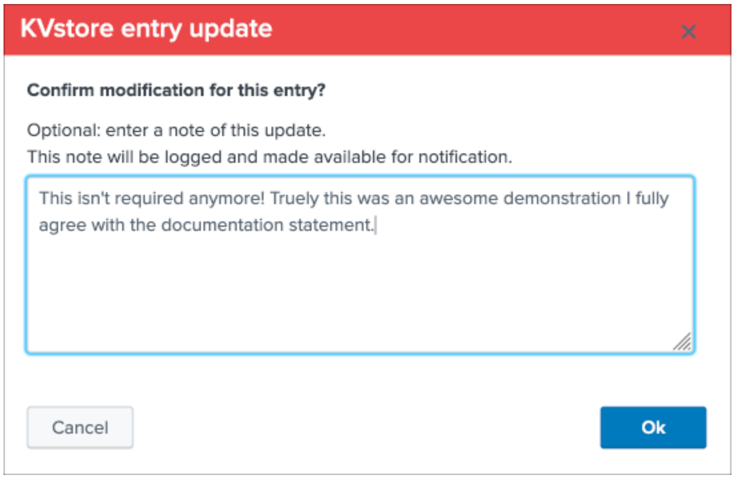
repeat the operation, which results in 3 new entities in TrackMe, one for each region:



data_name ⬍	last time ⬍	last ingest ⬍	priority ⬍	state ⬍	lag summary (lag event / lag ingestion) ⬍	last time idx ⬍	data
network:pan:traffic	26/07/2020 10:30	26/07/2020 10:30	medium	✓	-5 sec / 0 sec	26/07/2020 10:30	
network:pan:traffic:amer	26/07/2020 10:32	26/07/2020 10:32	medium	✓	1 sec / 0 sec	26/07/2020 10:32	
network:pan:traffic:apac	26/07/2020 10:32	26/07/2020 10:32	medium	✓	1 sec / 0 sec	26/07/2020 10:32	
network:pan:traffic:emea	26/07/2020 10:32	26/07/2020 10:32	medium	✓	1 sec / 0 sec	26/07/2020 10:32	

"What about the original data source that created automatically?".

We can simply disable the monitoring state via the disable button et voila!



KVstore entry update

Confirm modification for this entry?

Optional: enter a note of this update.
This note will be logged and made available for notification.

This isn't required anymore! Truely this was an awesome demonstration I fully agree with the documentation statement.

Elastic source example 2: creation

Now that we had so much fun with the example 1, let's have a look at the second example which relies on custom indexed fields.

```
source="network:pan:[region]:[company]"
```

For the purposes of the demonstration, we will this time create Elastic dedicated sources.

Let's create our first entity:

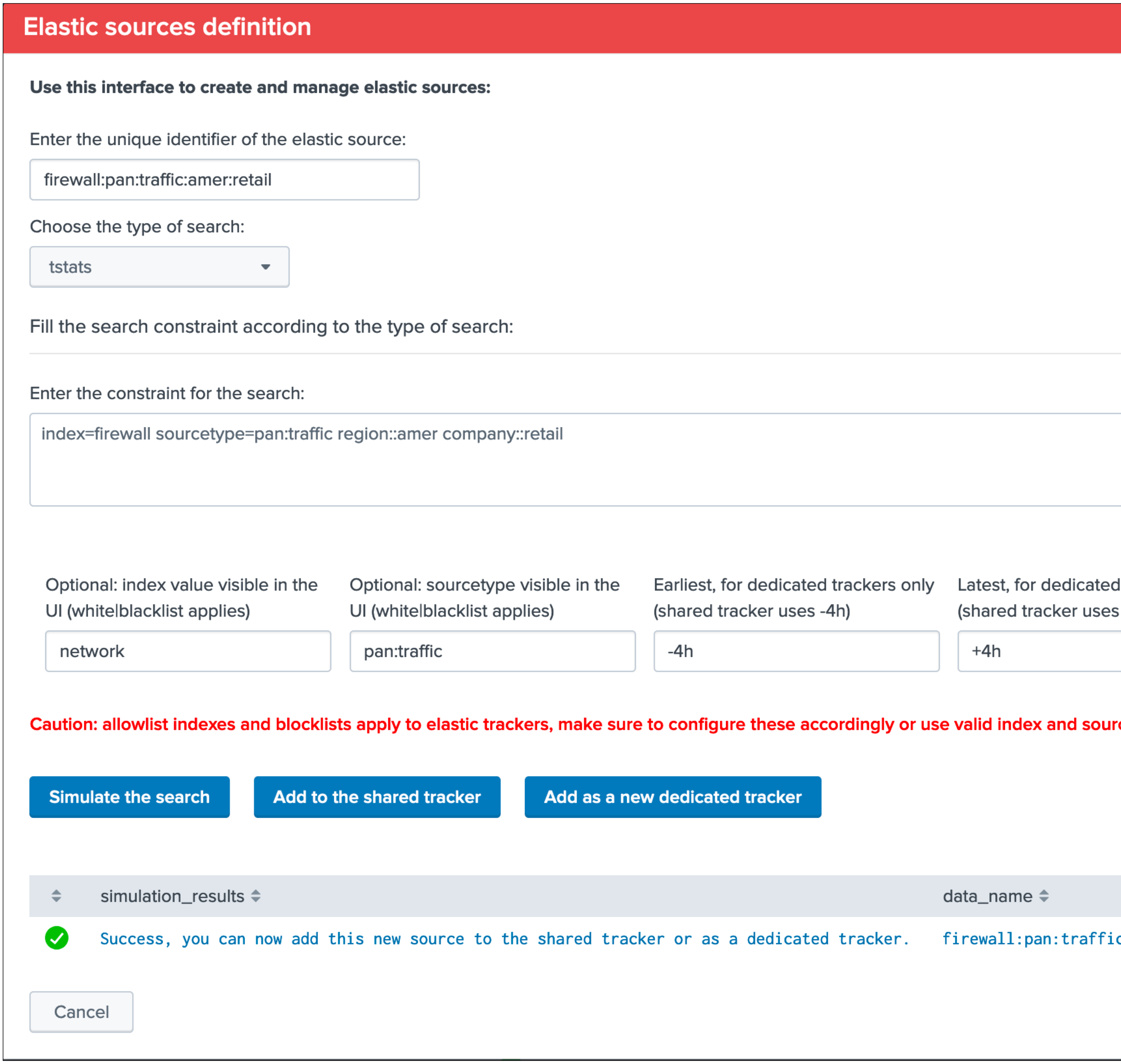
Summary:

- Define a name for the entity, this name is the value of the field `data_name` and needs to be unique in TrackMe
- Shall that name you provide not be unique, a little red cross and a message will indicate the issue when we run the simulation
- We choose a `search language`, because the source field is a Metadata, this is an indexed field and we can use the tstats command which is very efficient by looking at the tsdidx files rather than the raw events
- We define our search constraint for the first entity, in our case `index=firewall sourcetype=pan:traffic region::emea company::retail`
- We choose a value for the index and the sourcetype, this is having `no impacts` on the search itself and its result but determines how the entity is classified and filtered in the main UI
- Finally, we can optionally decide to define the earliest and latest time range, in our example we can leave that empty and rely on the default behaviour

Note about the search syntax:

- We use `"::"` as the delimiter rather than `"="` because these are indexed fields, and this indicates Splunk to treat them as such

Let's create our first entity:



Elastic sources definition

Use this interface to create and manage elastic sources:

Enter the unique identifier of the elastic source:

firewall:pan:traffic:amer:retail

Choose the type of search:

tstats

Fill the search constraint according to the type of search:

Enter the constraint for the search:

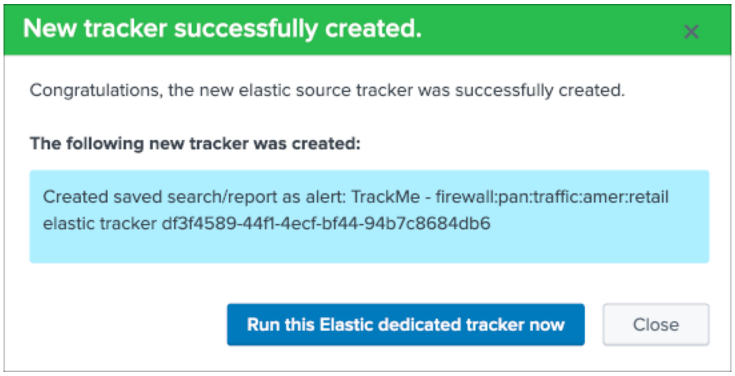
index=firewall sourcetype=pan:traffic region::amer company::retail

Optional: index value visible in the UI (white|blacklist applies)	Optional: sourcetype visible in the UI (white|blacklist applies)	Earliest, for dedicated trackers only (shared tracker uses -4h)	Latest, for dedicated (shared tracker uses
network	pan:traffic	-4h	+4h

Caution: allowlist indexes and blocklists apply to elastic trackers, make sure to configure these accordingly or use valid index and sourc

	simulation_results	data_name
✓	Success, you can now add this new source to the shared tracker or as a dedicated tracker.	firewall:pan:traffic

Once again this is looking perfectly good, this time we will create a dedicated tracker:



New tracker successfully created.

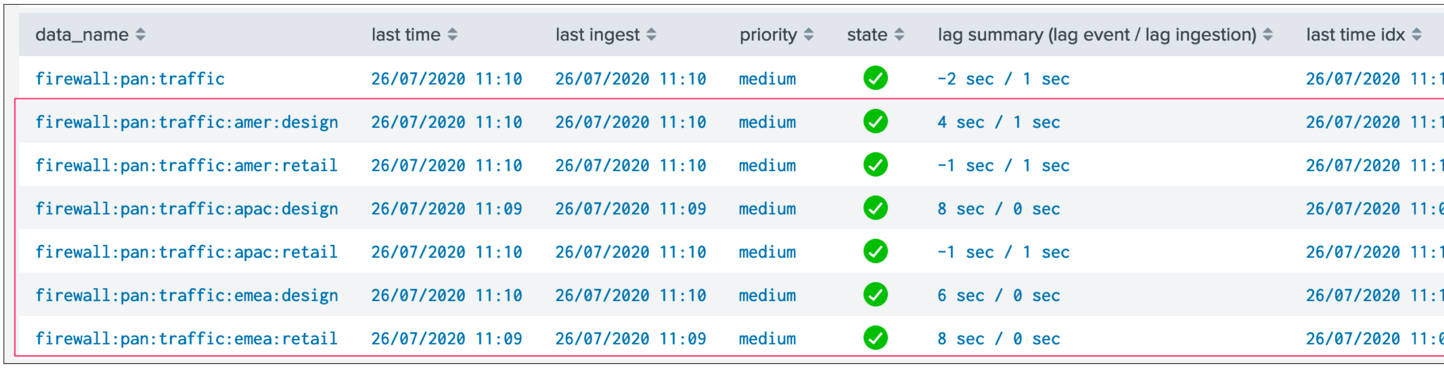
Congratulations, the new elastic source tracker was successfully created.

The following new tracker was created:

Created saved search/report as alert: TrackMe - firewall:pan:traffic:amer:retail elastic tracker df3f4589-44f1-4ecf-bf44-94b7c8684db6

Nice, let's click on the run button now, and repeat the operation for all entities!

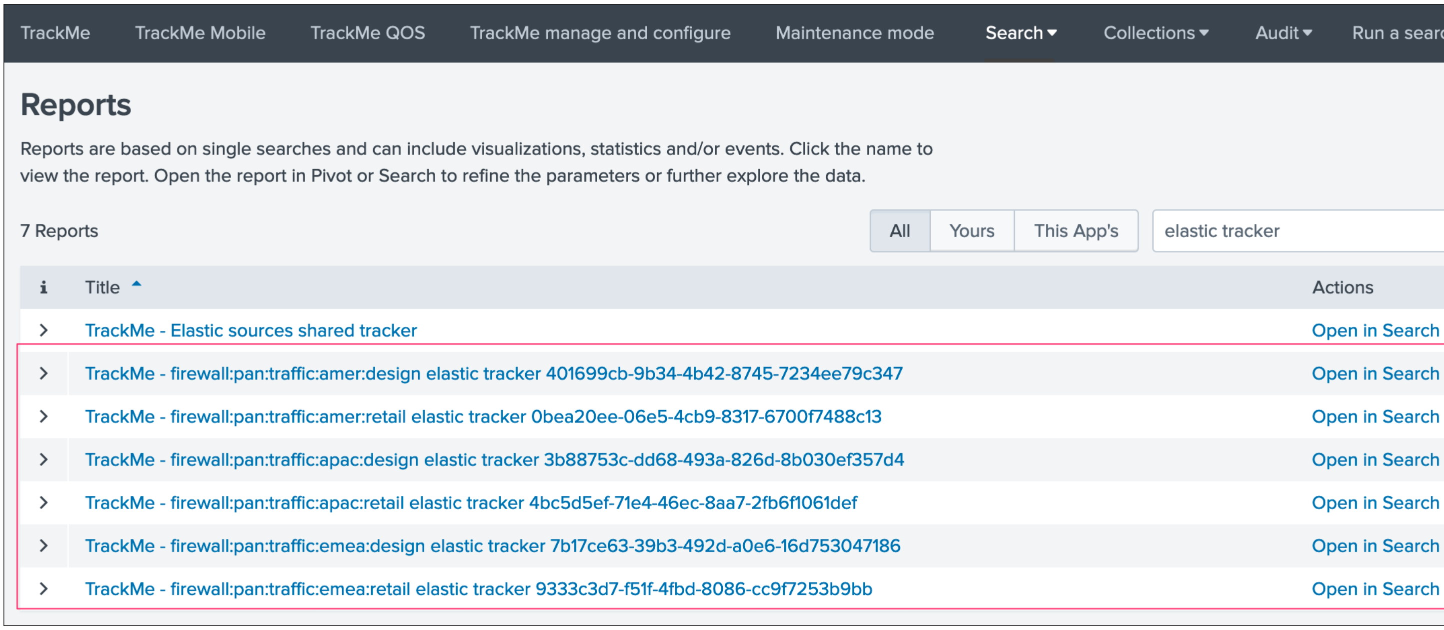
Once we did and created all the six entities, we can see the following in the data sources tab:



data_name	last time	last ingest	priority	state	lag summary (lag event / lag ingestion)	last time idx
firewall:pan:traffic	26/07/2020 11:10	26/07/2020 11:10	medium	✓	-2 sec / 1 sec	26/07/2020 11:1
firewall:pan:traffic:amer:design	26/07/2020 11:10	26/07/2020 11:10	medium	✓	4 sec / 1 sec	26/07/2020 11:1
firewall:pan:traffic:amer:retail	26/07/2020 11:10	26/07/2020 11:10	medium	✓	-1 sec / 1 sec	26/07/2020 11:1
firewall:pan:traffic:apac:design	26/07/2020 11:09	26/07/2020 11:09	medium	✓	8 sec / 0 sec	26/07/2020 11:0
firewall:pan:traffic:apac:retail	26/07/2020 11:10	26/07/2020 11:10	medium	✓	-1 sec / 1 sec	26/07/2020 11:1
firewall:pan:traffic:emea:design	26/07/2020 11:10	26/07/2020 11:10	medium	✓	6 sec / 0 sec	26/07/2020 11:1
firewall:pan:traffic:emea:retail	26/07/2020 11:09	26/07/2020 11:09	medium	✓	8 sec / 0 sec	26/07/2020 11:0

As we did earlier in the example 1, we will simply disable the original data source which is not required anymore.

Finally, because we created dedicated trackers, let's have a look at the reports:



Reports

Reports are based on single searches and can include visualizations, statistics and/or events. Click the name to view the report. Open the report in Pivot or Search to refine the parameters or further explore the data.

7 Reports

i	Title ▴	Actions
>	TrackMe - Elastic sources shared tracker	Open in Search
>	TrackMe - firewall:pan:traffic:amer:design elastic tracker 401699cb-9b34-4b42-8745-7234ee79c347	Open in Search
>	TrackMe - firewall:pan:traffic:amer:retail elastic tracker 0bea20ee-06e5-4cb9-8317-6700f7488c13	Open in Search
>	TrackMe - firewall:pan:traffic:apac:design elastic tracker 3b88753c-dd68-493a-826d-8b030ef357d4	Open in Search
>	TrackMe - firewall:pan:traffic:apac:retail elastic tracker 4bc5d5ef-71e4-46ec-8aa7-2fb6f1061def	Open in Search
>	TrackMe - firewall:pan:traffic:emea:design elastic tracker 7b17ce63-39b3-492d-a0e6-16d753047186	Open in Search
>	TrackMe - firewall:pan:traffic:emea:retail elastic tracker 9333c3d7-f51f-4fbd-8086-cc9f7253b9bb	Open in Search

We can see that TrackMe has created a new scheduled report for each entity we created, it is perfectly possible to edit these reports up to your needs.

Voila, we have now covered two complete examples of how and why creating Elastic Sources, there are many more use cases obviously and each can be very specific to your context, therefore we covered the essential part of the feature.

Elastic source example 3: creation

Let's create our lookup based Elastic Source, for this we rely on the Splunk from search command capabilities to handle lookup, and we potentially define additional statements to set the _time and _indextime (if any)

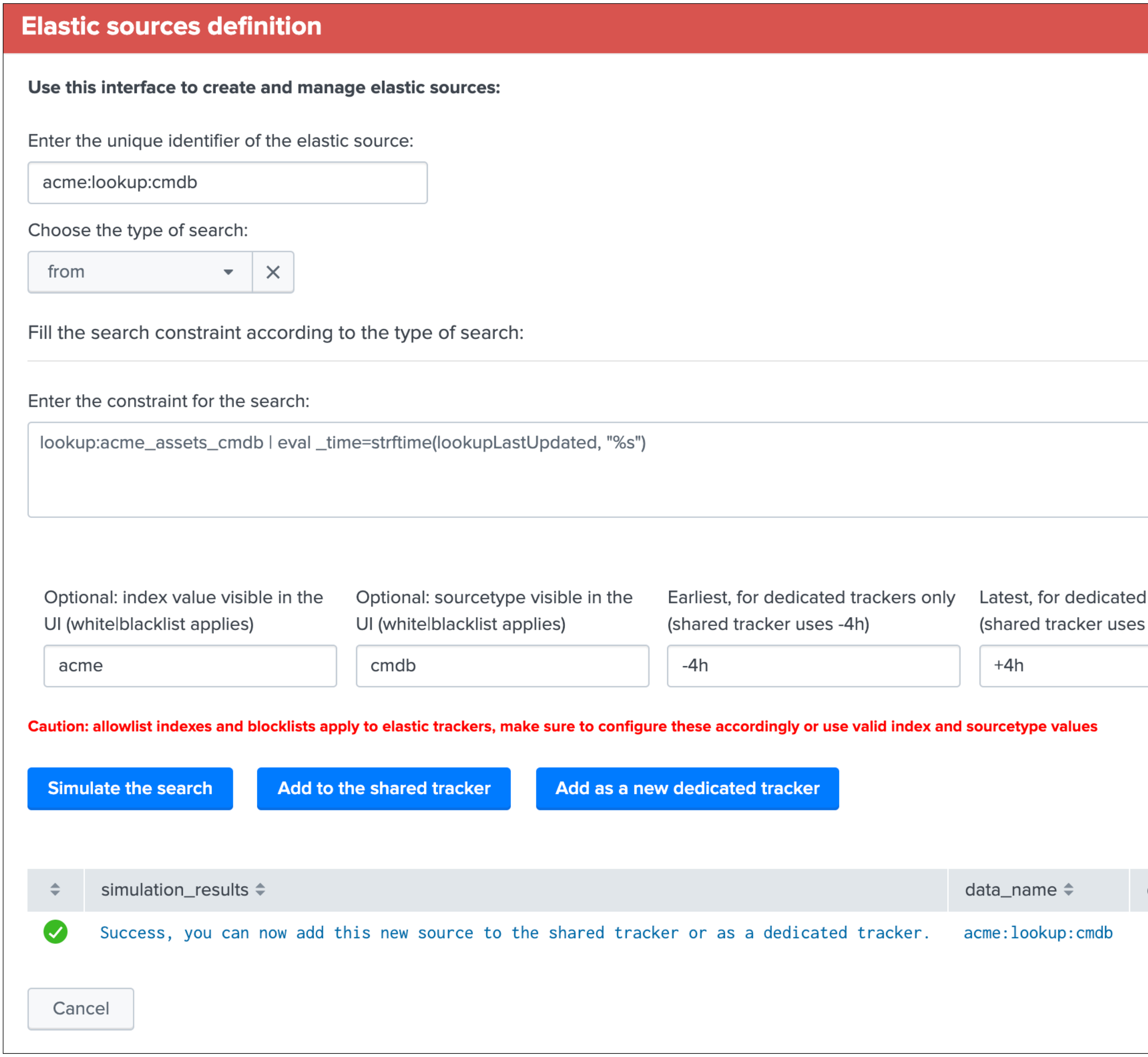
Litteraly, we are going to use the following SPL search to achieve our target:

```
| from lookup:acme_assets_cmdb | eval _time=strftime(lookupLastUpdated, "%s")
```

If our lookupLastUpdated would have been in a human readable format, we could have used the stptime function to convert it into an epoch time, for example:

```
| from lookup:acme_assets_cmdb | eval _time=strptime(lookupLastUpdated, "%d/%m/%Y %H:
↪%M:%S")
```

Applied to TrackMe in the Elastic Sources UI creation:



Elastic sources definition

Use this interface to create and manage elastic sources:

Enter the unique identifier of the elastic source:

acme:lookup:cmdb

Choose the type of search:

from

Fill the search constraint according to the type of search:

Enter the constraint for the search:

lookup:acme_assets_cmdb | eval _time=strftime(lookupLastUpdated, "%s")

Optional: index value visible in the UI (white|blacklist applies)	Optional: sourcetype visible in the UI (white|blacklist applies)	Earliest, for dedicated trackers only (shared tracker uses -4h)	Latest, for dedicated (shared tracker uses
acme	cmdb	-4h	+4h

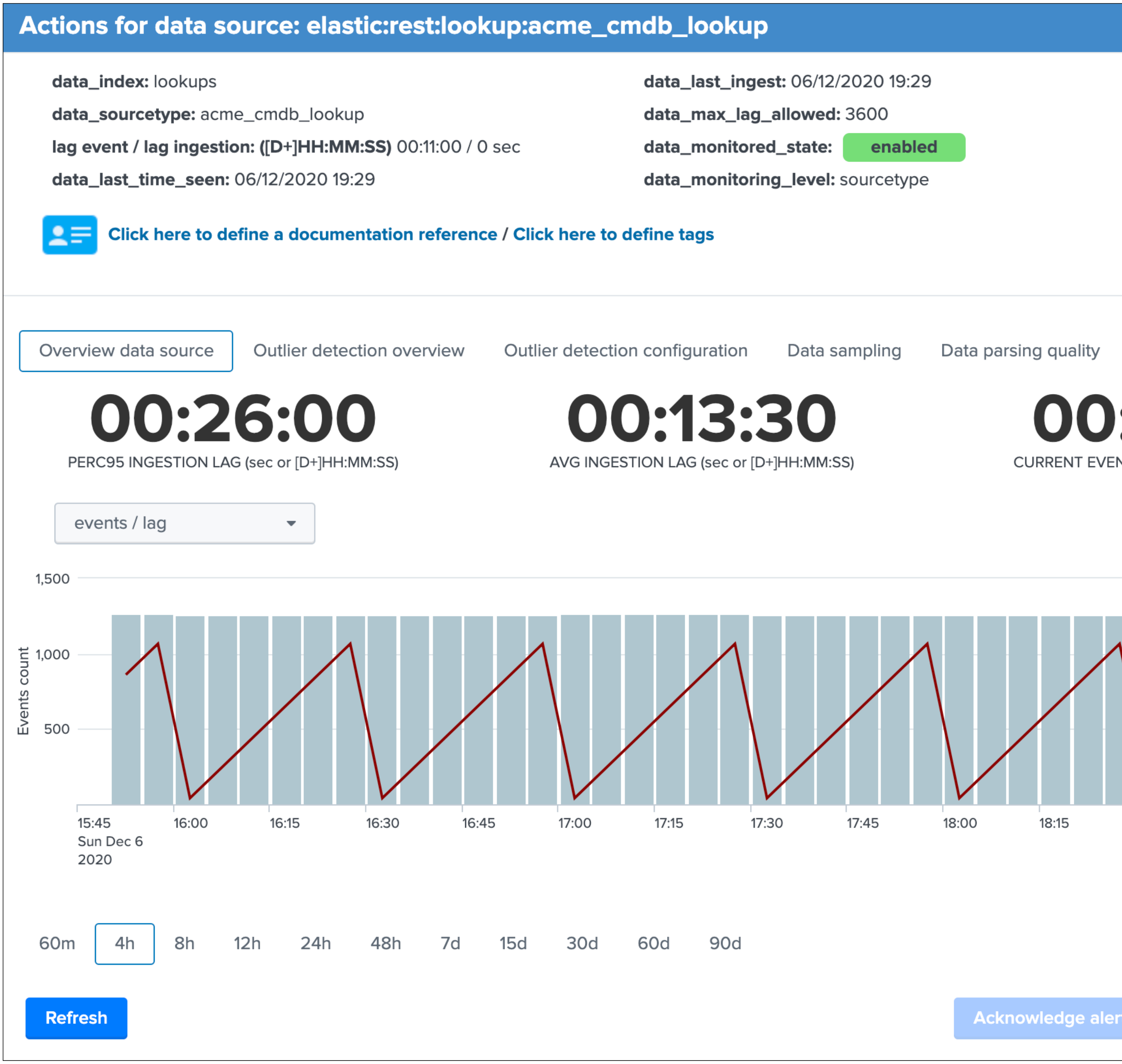
Caution: allowlist indexes and blocklists apply to elastic trackers, make sure to configure these accordingly or use valid index and sourcetype values

	simulation_results	data_name
✓	Success, you can now add this new source to the shared tracker or as a dedicated tracker.	acme:lookup:cmdb

Notes:

- The "from " key word is not required and will be substituted by TrackMe automatically (once you selected from in the dropdown)
- earliest and latest do not matter for a lookup, so you can leave these with their default values
- The index and sourcetype are only used for UI filtering purposes, so you can define the values up to your preference
- Depending on the volume of records in the lookup and the time taken by Splunk to load its content, you may consider using the shared tracker mode, or a dedicated tracker for longer execution run times

Once the Elastic Source has been created, and we ran the tracker:



Actions for data source: elastic:rest:lookup:acme_cmdb_lookup

data_index: lookups

data_sourcetype: acme_cmdb_lookup

lag event / lag ingestion: ([D+]HH:MM:SS) 00:11:00 / 0 sec

data_last_time_seen: 06/12/2020 19:29

data_last_ingest: 06/12/2020 19:29

data_max_lag_allowed: 3600

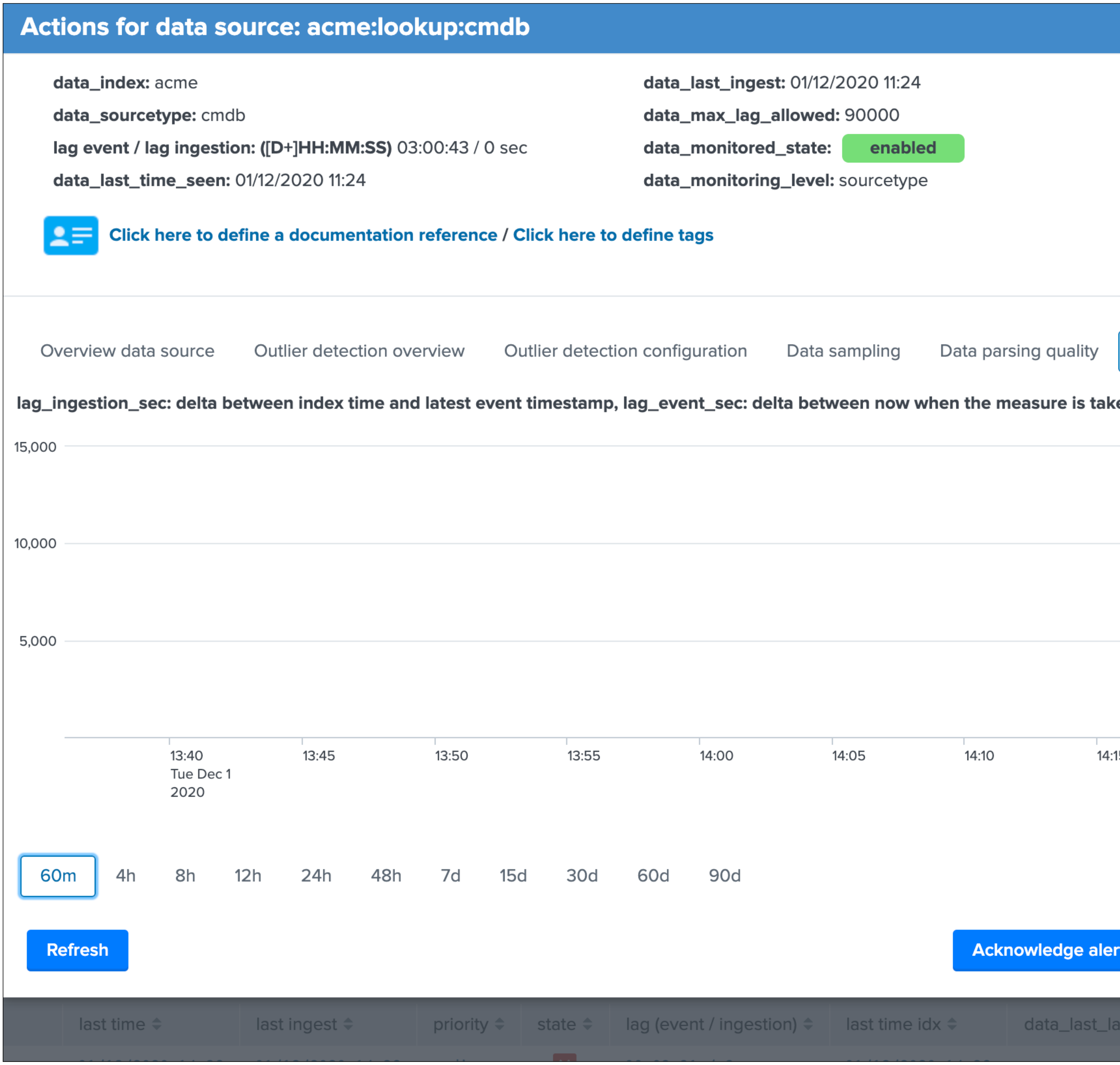
data_monitored_state: enabled

data_monitoring_level: sourcetype

Click here to define a documentation reference / Click here to define tags

As we can see, the current lagging corresponds to the difference between now and the latest update of the lookup, TrackMe will immediately starts to compute all metrics, the event count corresponds to the number of records (which allows the usage of outliers detection too), etc.

When TrackMe detects that the data source is a based on a lookup, the statistics are returned from the trackme metrics automatically.



Actions for data source: acme:lookup:cmdb

data_index: acme

data_sourcetype: cmdb

lag event / lag ingestion: ([D+]HH:MM:SS) 03:00:43 / 0 sec

data_last_time_seen: 01/12/2020 11:24

data_last_ingest: 01/12/2020 11:24

data_max_lag_allowed: 90000

data_monitored_state: enabled

data_monitoring_level: sourcetype

Click here to define a documentation reference / Click here to define tags

lag_ingestion_sec: delta between index time and latest event timestamp, lag_event_sec: delta between now when the measure is take

Elastic source example 4: creation

As explained in the example 4 description, we can use a rest based search to monitor any data that is not available to the search head host TrackMe, let's consider the example a lookup hosted on a different search head.

On the search head that owns the lookup, we can use the following query:

```
| from lookup:acme_assets_cmdb | eval _time=strftime(lookupLastUpdated, "%s")
```

Using a rest search, we will achieve the same job but this time remotely via a rest call to a search endpoint of the

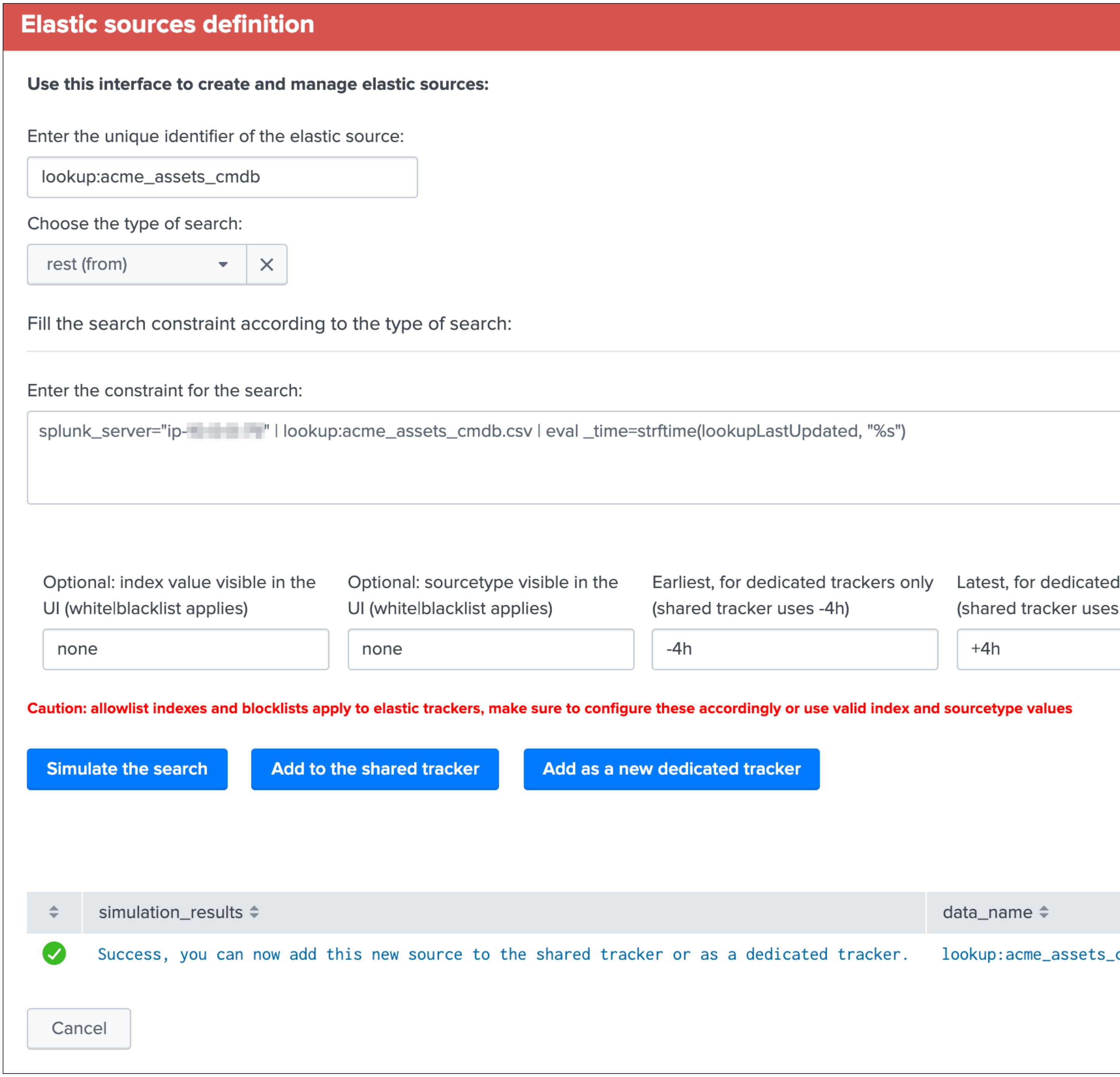
Splunk API using the rest command, the Elastic Source search syntax will be the following:

```
splunk_server="my_search_head" | from lookup:acme_assets_cmdb | eval _
↪time=strftime(lookupLastUpdated, "%s")
```

The first pipe needs to contain the arguments passed to the rest command, the only mandatory argument is either `splunk_server` to target a unique Splunk instance, or `splunk_server_group` to target a group of search heads. As well, any additional agrument can be given to the rest command by ading these in the first pipe of the search constraint. (timeout, count, etc)

Tip:

- The Splunk server name needs to be between double quotes, ex: splunk_server="my_search_head"
- In this example of a lookup, the knowledge objects needs to be shared properly such that it is available to be accessed via the rest API



Elastic sources definition

Use this interface to create and manage elastic sources:

Enter the unique identifier of the elastic source:

lookup:acme_assets_cmdb

Choose the type of search:

rest (from)

Fill the search constraint according to the type of search:

Enter the constraint for the search:

splunk_server="ip-[redacted]" | lookup:acme_assets_cmdb.csv | eval _time=strftime(lookupLastUpdated, "%s")

Optional: index value visible in the UI (white|blacklist applies): none

Optional: sourcetype visible in the UI (white|blacklist applies): none

Earliest, for dedicated trackers only (shared tracker uses -4h): -4h

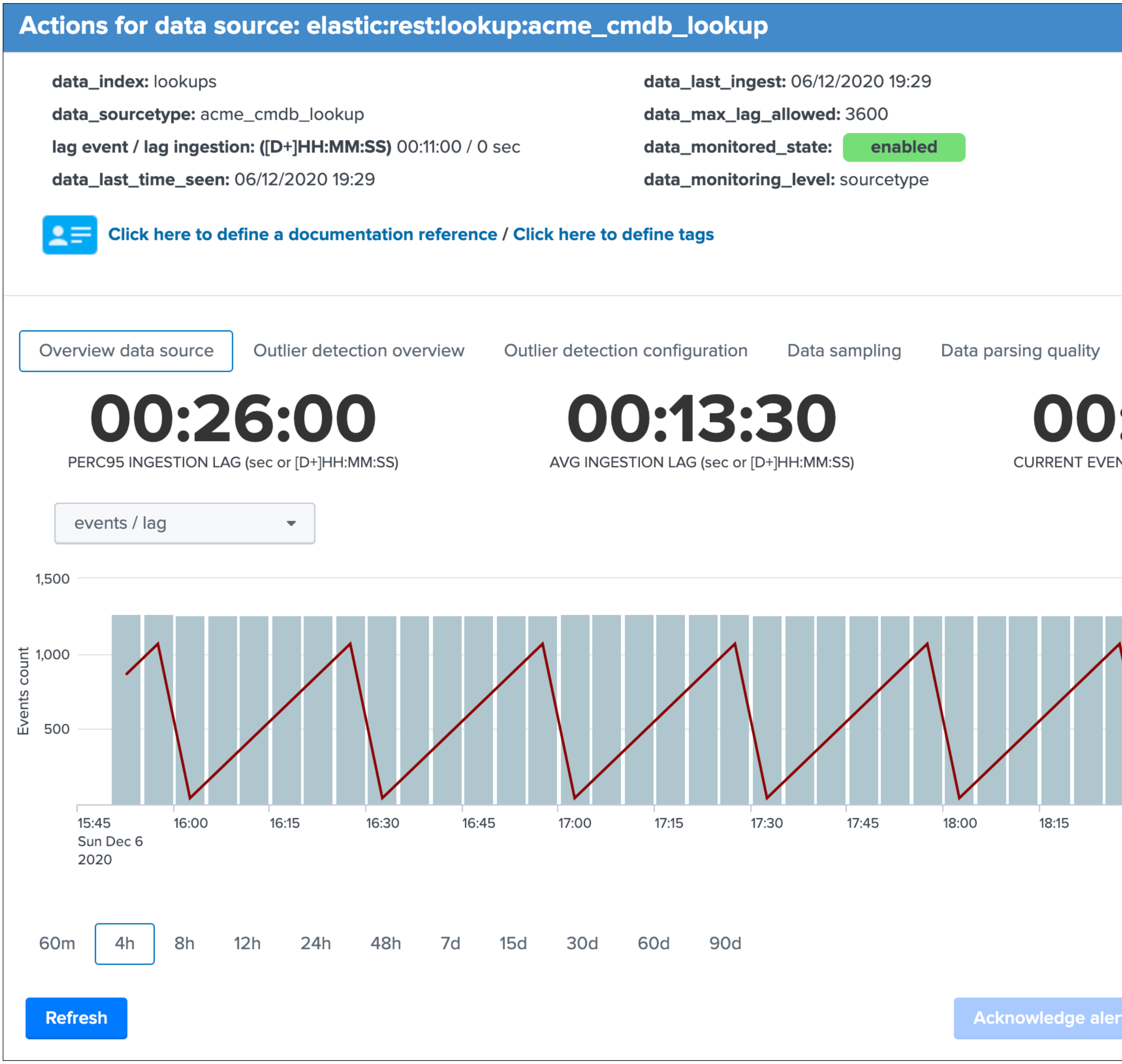
Latest, for dedicated (shared tracker uses: +4h

Caution: allowlist indexes and blocklists apply to elastic trackers, make sure to configure these accordingly or use valid index and sourcetype values

	simulation_results	data_name
✓	Success, you can now add this new source to the shared tracker or as a dedicated tracker.	lookup:acme_assets_c

Warning: Currently the rest command generates a warning message "Unable to determine response format from HTTP Header", this message can be safety ignored as it does not impact the results in anyway, but cannot unfortunately be removed at the moment, until it is fixed by Splunk.

Once created, the new data source appears in the UI automatically, the following example shows the behaviour with a lookup that is updated every 30 minutes:



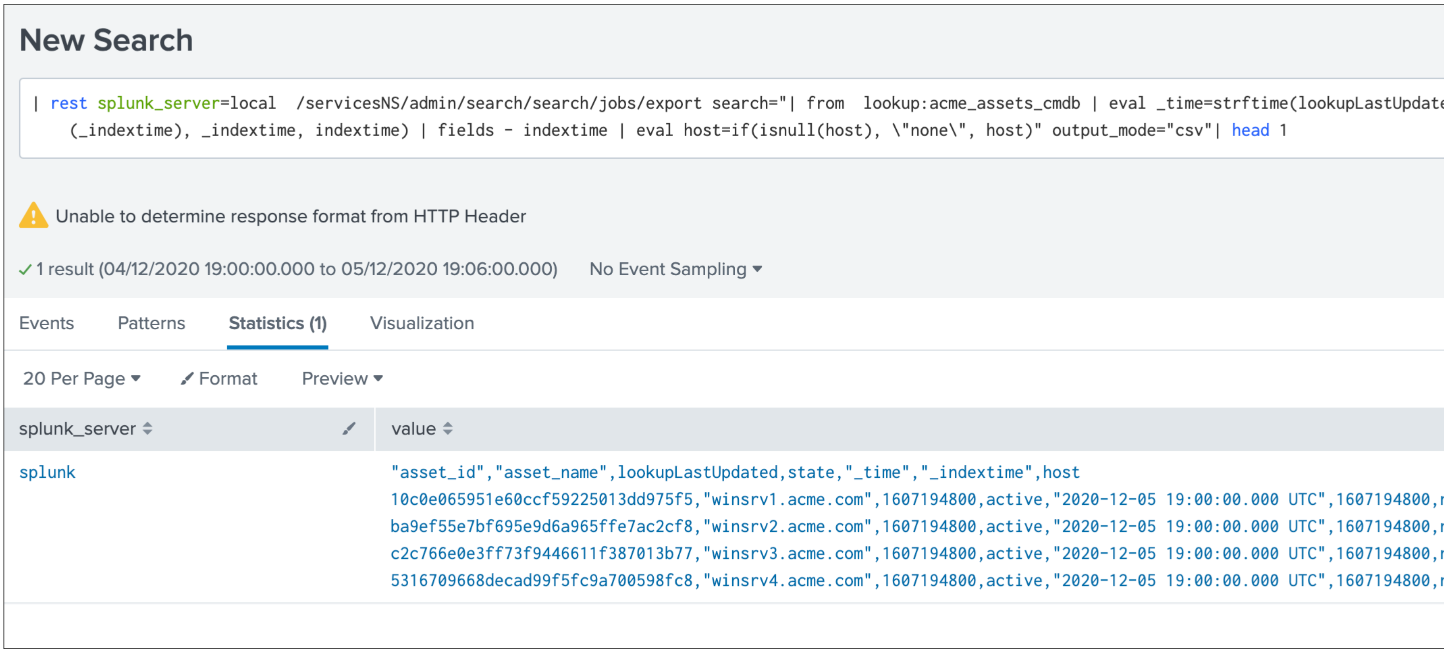
Actions for data source: elastic:rest:lookup:acme_cmdb_lookup

data_index: lookups
data_sourcetype: acme_cmdb_lookup
lag event / lag ingestion: ([D+]HH:MM:SS) 00:11:00 / 0 sec
data_last_time_seen: 06/12/2020 19:29

data_last_ingest: 06/12/2020 19:29
data_max_lag_allowed: 3600
data_monitored_state: enabled
data_monitoring_level: sourcetype

Click here to define a documentation reference / Click here to define tags

In the example of a lookup, the Search button would result in the following:



```
| rest splunk_server=local  /servicesNS/admin/search/search/jobs/export search="| from  lookup:acme_assets_cmdb | eval _time=strftime(lookupLastUpdate
   (_indextime), _indextime, indextime) | fields - indextime | eval host=if(isnull(host), \"none\", host)" output_mode="csv"| head 1
```

Unable to determine response format from HTTP Header

✓ 1 result (04/12/2020 19:00:00.000 to 05/12/2020 19:06:00.000)

splunk_server	value
splunk	"asset_id","asset_name",lookupLastUpdated,state,"_time","_indextime",host 10c0e065951e60ccf59225013dd975f5,"winsrv1.acme.com",1607194800,active,"2020-12-05 19:00:00.000 UTC",1607194800, ba9ef55e7bf695e9d6a965ffe7ac2cf8,"winsrv2.acme.com",1607194800,active,"2020-12-05 19:00:00.000 UTC",1607194800, c2c766e0e3ff73f9446611f387013b77,"winsrv3.acme.com",1607194800,active,"2020-12-05 19:00:00.000 UTC",1607194800, 5316709668decad99f5fc9a700598fc8,"winsrv4.acme.com",1607194800,active,"2020-12-05 19:00:00.000 UTC",1607194800,

Elastic sources under the hood

Some additional more technical details:

Elastic sources shared

Each elastic source definition is stored in the following KVstore based lookup:

`trackme_elastic_sources`

Specially, we have the following fields:

- `data_name` is the unique identifier
- `search_constraint` is the search constraint
- `search_mode` is the search command to be used
- `elastic_data_index` is the value for the index to be shown in the UI
- `elastic_data_sourcetype` is the value for the sourcetype to be show in the UI

When the Elastic Source shared tracker runs:

`TrackMe - Elastic sources shared tracker`

It calls a special saved search `| savedsearch runSPL` which expects in argument any number of SPL searches to be performed.

The tracker loads each record stored in the collection, and uses different evaluations to compose the final SPL search for each record.

Finally, it calls different shared knowledge objects that are commonly used by the trackers:

- Apply the TrackMe different macros and functions to calculate things like the lagging metrics, etc
- Calls all knowledge objects from TrackMe which insert and update the KVstore lookup, generate flipping status events, generate and records the metrics in the metric store

Besides the fact that Elastic sources appears in the data sources tab, there are no interactions between the data source trackers and the shared Elastic source trackers, there are independents.

In addition, the collection is used automatically by the main interface if you click on the `Search` button to generate the relevant search to access the events related to that entity.

Elastic sources dedicated

Each elastic source definition is stored in the following KVstore based lookup:

`trackme_elastic_sources_dedicated`

Specially, we have the following fields:

- `data_name` is the unique identifier
- `search_constraint` is the search constraint
- `search_mode` is the search command to be used
- `elastic_data_index` is the value for the index to be shown in the UI
- `elastic_data_sourcetype` is the value for the sourcetype to be show in the UI

When the dedicated Elastic source tracker runs, the following applies:

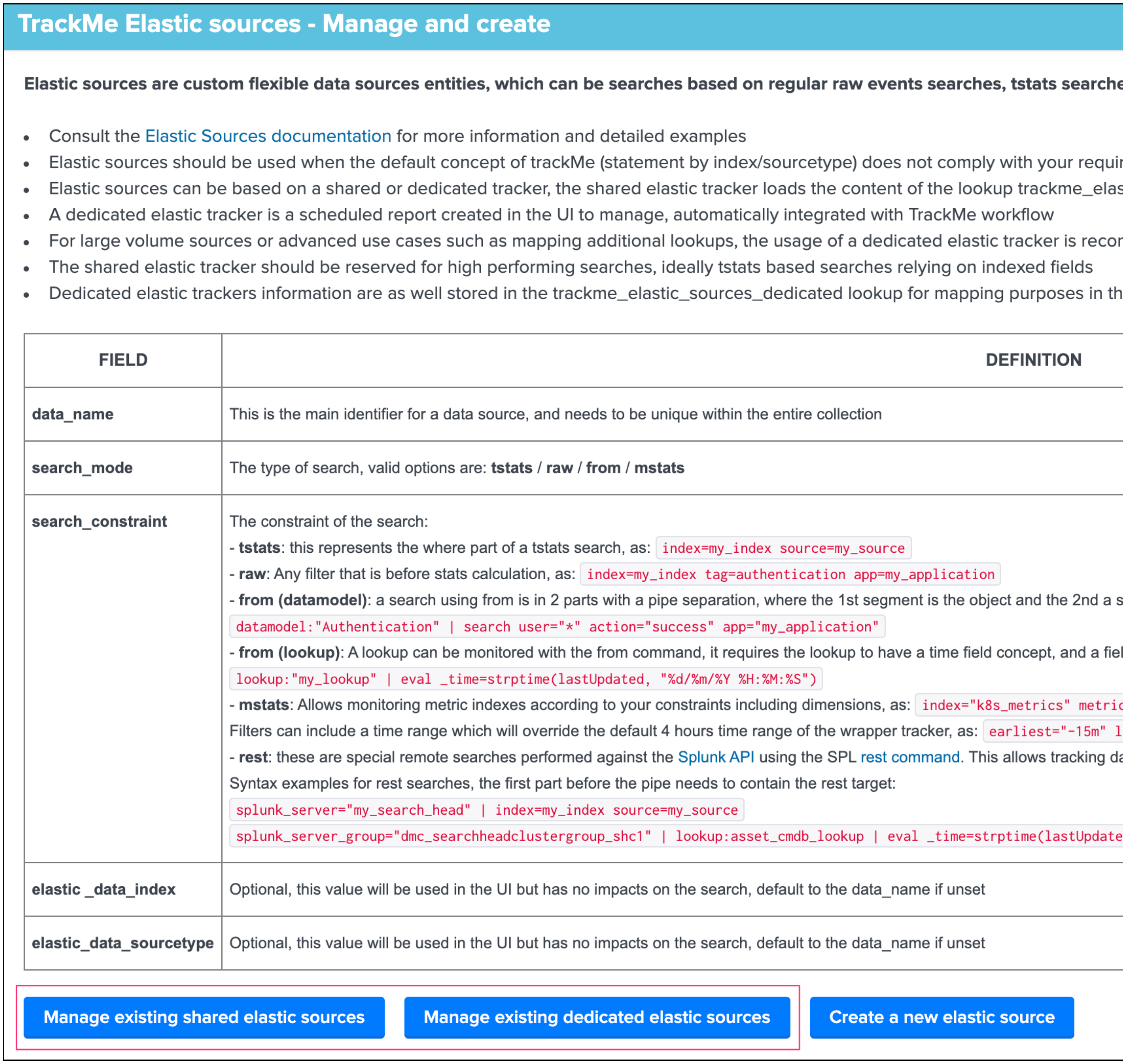
- The report contains the structured search syntax that was automatically built by the UI when it was created
- The report calls different knowledge objects that are common to the trackers to insert and update records in the KVstore, generate flipping status records if any and generate the lagging metrics to be stored into the metric store

Besides the fact that Elastic sources appears in the data sources tab, there are no interactions between the data source trackers and the dedicated Elastic source trackers, there are independents.

In addition, the collection is used automatically by the main interface if you click on the `Search` button to generate the relevant search to access the events related to that entity.

Remove Elastic Sources

You can delete one or more Elastic Sources, shared or dedicated, within the UI main screen:



TrackMe Elastic sources - Manage and create

Elastic sources are custom flexible data sources entities, which can be searches based on regular raw events searches, tstats searche

- Consult the Elastic Sources documentation for more information and detailed examples
- Elastic sources should be used when the default concept of trackMe (statement by index/sourcetype) does not comply with your requir
- Elastic sources can be based on a shared or dedicated tracker, the shared elastic tracker loads the content of the lookup trackme_elas
- A dedicated elastic tracker is a scheduled report created in the UI to manage, automatically integrated with TrackMe workflow
- For large volume sources or advanced use cases such as mapping additional lookups, the usage of a dedicated elastic tracker is recor
- The shared elastic tracker should be reserved for high performing searches, ideally tstats based searches relying on indexed fields
- Dedicated elastic trackers information are as well stored in the trackme_elastic_sources_dedicated lookup for mapping purposes in th

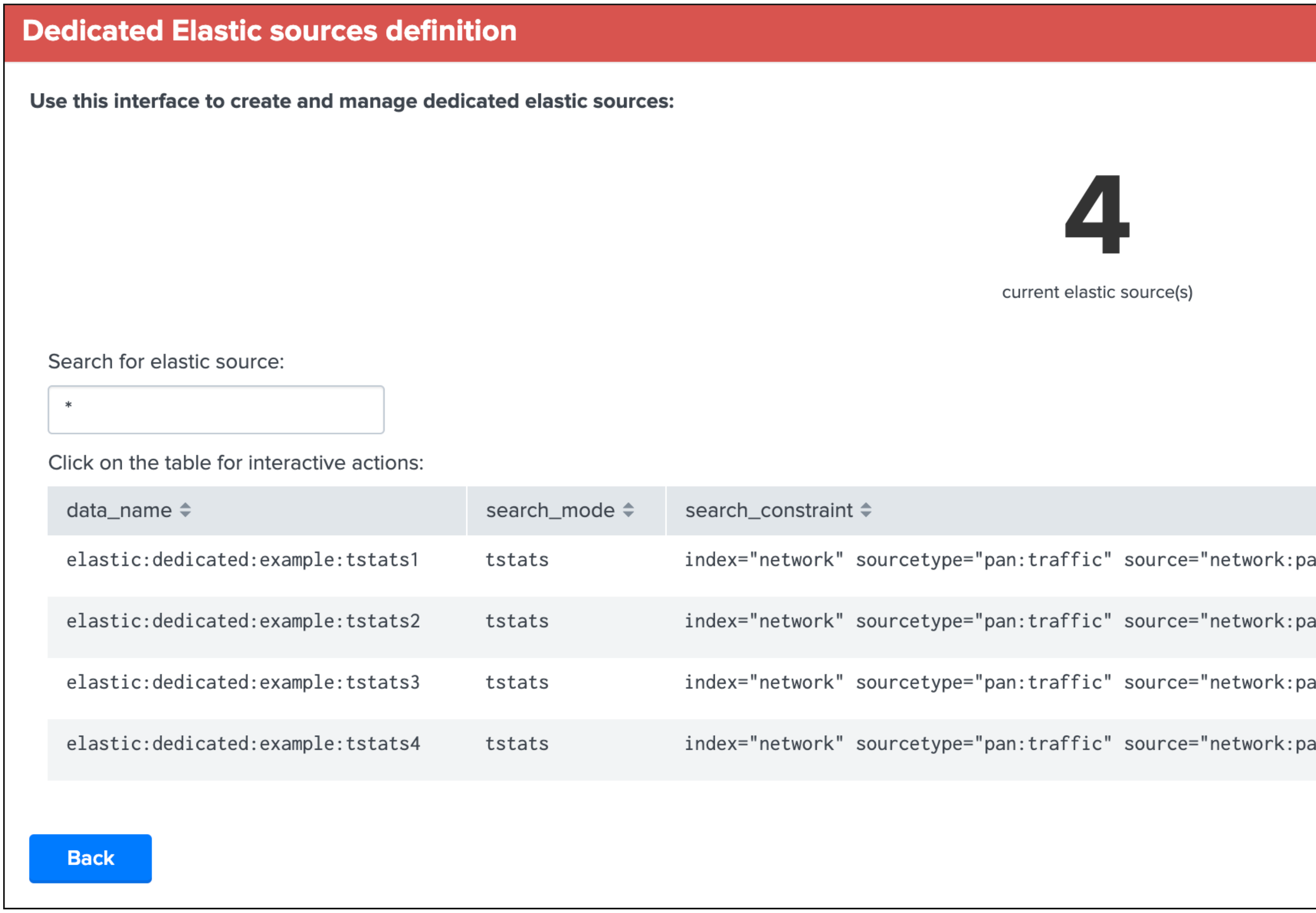
FIELD	DEFINITION
data_name	This is the main identifier for a data source, and needs to be unique within the entire collection
search_mode	The type of search, valid options are: **tstats** / **raw** / **from** / **mstats**

search_constraint	The constraint of the search:
	- **tstats**: this represents the where part of a tstats search, as: `index=my_index source=my_source`
	- **raw**: Any filter that is before stats calculation, as: `index=my_index tag=authentication app=my_application`
	- **from (datamodel)**: a search using from is in 2 parts with a pipe separation, where the 1st segment is the object and the 2nd a s
	`datamodel:"Authentication" | search user="*" action="success" app="my_application"`
	- **from (lookup)**: A lookup can be monitored with the from command, it requires the lookup to have a time field concept, and a fiel
	`lookup:"my_lookup" | eval _time=strptime(lastUpdated, "%d/%m/%Y %H:%M:%S")`
	- **mstats**: Allows monitoring metric indexes according to your constraints including dimensions, as: `index="k8s_metrics" metric`
	Filters can include a time range which will override the default 4 hours time range of the wrapper tracker, as: `earliest="-15m" l`
	- **rest**: these are special remote searches performed against the Splunk API using the SPL rest command. This allows tracking da
	Syntax examples for rest searches, the first part before the pipe needs to contain the rest target:
	`splunk_server="my_search_head" | index=my_index source=my_source`
	`splunk_server_group="dmc_searchheadclustergroup_shc1" | lookup:asset_cmdb_lookup | eval _time=strptime(lastUpdate`

elastic _data_index	Optional, this value will be used in the UI but has no impacts on the search, default to the data_name if unset
elastic_data_sourcetype	Optional, this value will be used in the UI but has no impacts on the search, default to the data_name if unset

Manage existing shared elastic sources
Manage existing dedicated elastic sources
Create a new elastic source

Example with dedicated Elastic Sources:



Dedicated Elastic sources definition

Use this interface to create and manage dedicated elastic sources:

4

current elastic source(s)

Search for elastic source:

*

Click on the table for interactive actions:

data_name ⇕	search_mode ⇕	search_constraint ⇕
elastic:dedicated:example:tstats1	tstats	index="network" sourcetype="pan:traffic" source="network:pa
elastic:dedicated:example:tstats2	tstats	index="network" sourcetype="pan:traffic" source="network:pa
elastic:dedicated:example:tstats3	tstats	index="network" sourcetype="pan:traffic" source="network:pa
elastic:dedicated:example:tstats4	tstats	index="network" sourcetype="pan:traffic" source="network:pa

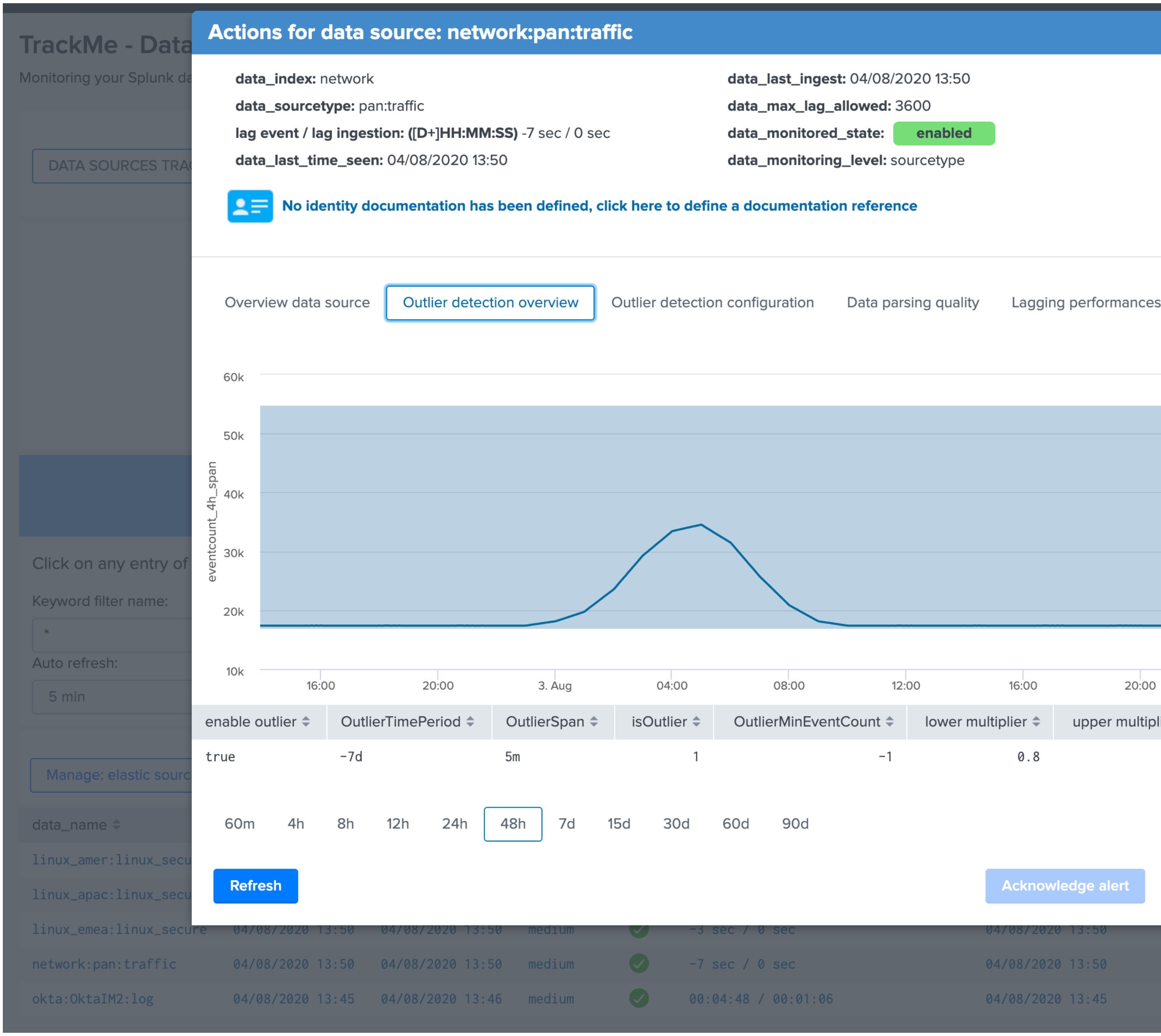
When deleting Elastic Sources via the UI, the following actions are occurring:

- The UI calls a REST API endpoint via the *REST API trackme SPL command*
- API endpoints are *elastic_shared_del / Delete a new shared Elastic Source* and *elastic_dedicated_del / Delete a new shared Elastic Source*
- All related objects are suppressed automatically, this includes the Elastic Sources KVstore collections, the entities in the main Data sources collection, and the scheduled reports for dedicated Elastic Sources
- Actions and content are logges in the audit collection before their suppression

3.1.4 Outliers detection and behaviour analytic

Outliers detection feature

Outliers detection provides a workflow to automatically detect and alert when the volume of events generated by a source goes beyond or over a usual volume determined by analysing the historical behaviour.



TrackMe - Data

Monitoring your Splunk da

DATA SOURCES TRA

Click on any entry of

Keyword filter name:

Auto refresh:

5 min

Manage: elastic sourc

Actions for data source: network:pan:traffic

data_index: network

data_sourcetype: pan:traffic

lag event / lag ingestion: ([D+]HH:MM:SS) -7 sec / 0 sec

data_last_time_seen: 04/08/2020 13:50

data_last_ingest: 04/08/2020 13:50

data_max_lag_allowed: 3600

data_monitored_state: enabled

data_monitoring_level: sourcetype

No identity documentation has been defined, click here to define a documentation reference

Overview data source | Outlier detection overview | Outlier detection configuration | Data parsing quality | Lagging performances

enable outlier	OutlierTimePeriod	OutlierSpan	isOutlier	OutlierMinEventCount	lower multiplier	upper multipl
true	-7d	5m	1	-1	0.8	

60m 4h 8h 12h 24h 48h 7d 15d 30d 60d 90d

Refresh

Acknowledge alert

data_name						
linux_amer:linux_secu						
linux_apac:linux_secu						
linux_emea:linux_secure	04/08/2020 13:50	04/08/2020 13:50	medium	✓	-3 sec / 0 sec	04/08/2020 13:50
network:pan:traffic	04/08/2020 13:50	04/08/2020 13:50	medium	✓	-7 sec / 0 sec	04/08/2020 13:50
okta:OktaIM2:log	04/08/2020 13:45	04/08/2020 13:46	medium	✓	00:04:48 / 00:01:06	04/08/2020 13:45

How things work:

- Each execution of the data trackers generates summary events which are indexed as summary data in the same time that the KVstore collections are updated
- These events are processed by the Summary Investigator tracker which uses a standard deviation calculation based approach from the Machine Learning toolkit
- We process standard deviation calculations based on a 4 hours event count reported during each execution of the data trackers
- The Summary Investigator maintains a KVstore lookup which content is used as a source of enrichment by the trackers to define essentially an "isOutlier" flag
- Should outliers be detected based on the policy, which is customisable om a per source basis, the source will be reported in alert

- Different options are provided to control the quality of the outliers calculation, as controlling lower and upper threshold multipliers, or even switching to a static lower bond definition
- Built-in views provide the key feature to quickly investigate the source in alert and proceed to further investigations if required

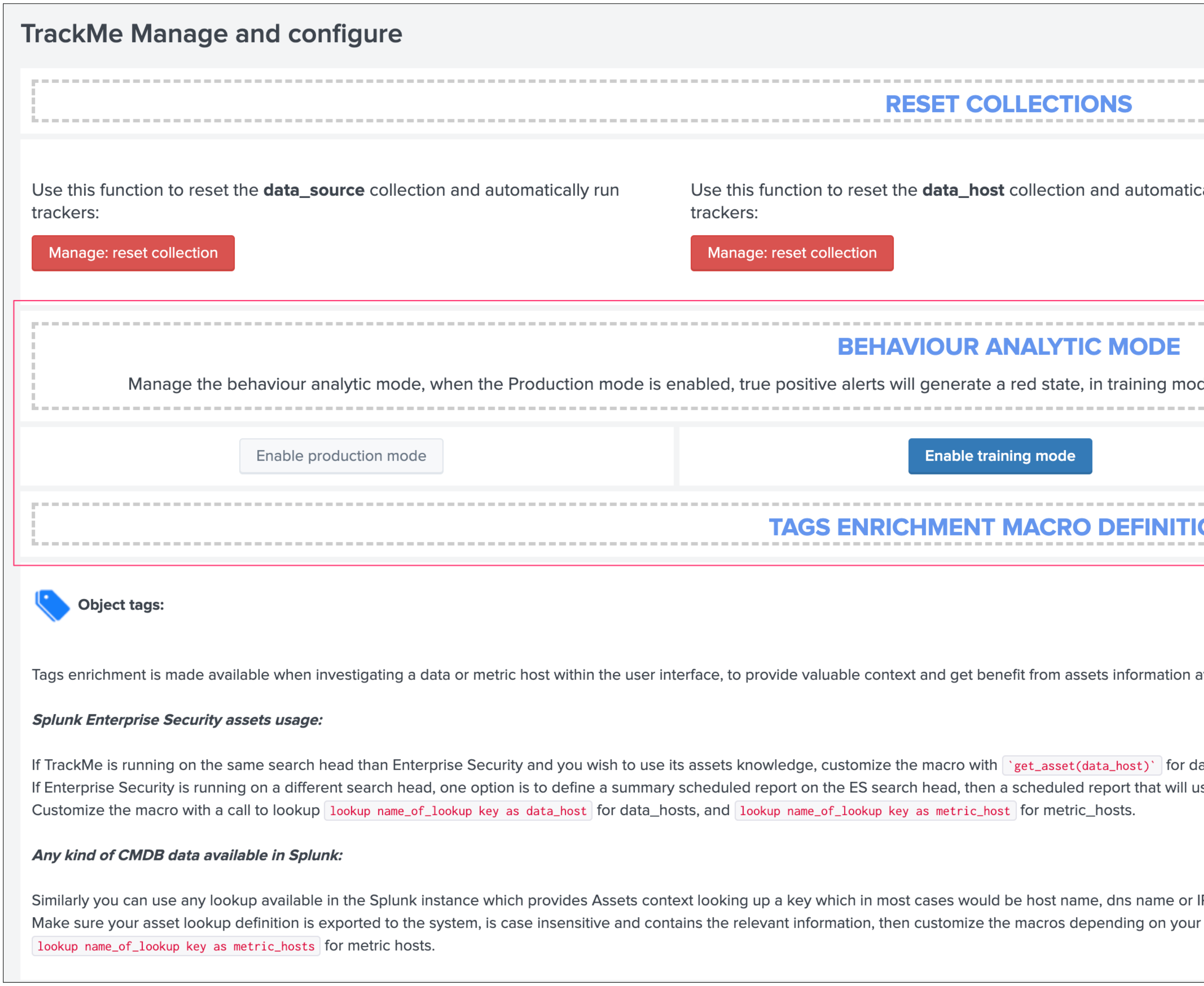
Behaviour Analytic Mode

By default, the application operates in Production mode, which means that an outlier detection occurring over a data source or host will influence its state effectively.

The behaviour analytic mode can be switched to the following status:

- production: affects objects status to the red state
- training : affects objects status to the orange state
- disabled: does nothing

The mode can be configured via UI in the "TrackMe manage and configure" link in the navigation bar:



TrackMe Manage and configure

RESET COLLECTIONS

Use this function to reset the **data_source** collection and automatically run trackers:

Use this function to reset the **data_host** collection and automatica trackers:

BEHAVIOUR ANALYTIC MODE

Manage the behaviour analytic mode, when the Production mode is enabled, true positive alerts will generate a red state, in training moc

TAGS ENRICHMENT MACRO DEFINITIC

Object tags:

Tags enrichment is made available when investigating a data or metric host within the user interface, to provide valuable context and get benefit from assets information a

Splunk Enterprise Security assets usage:

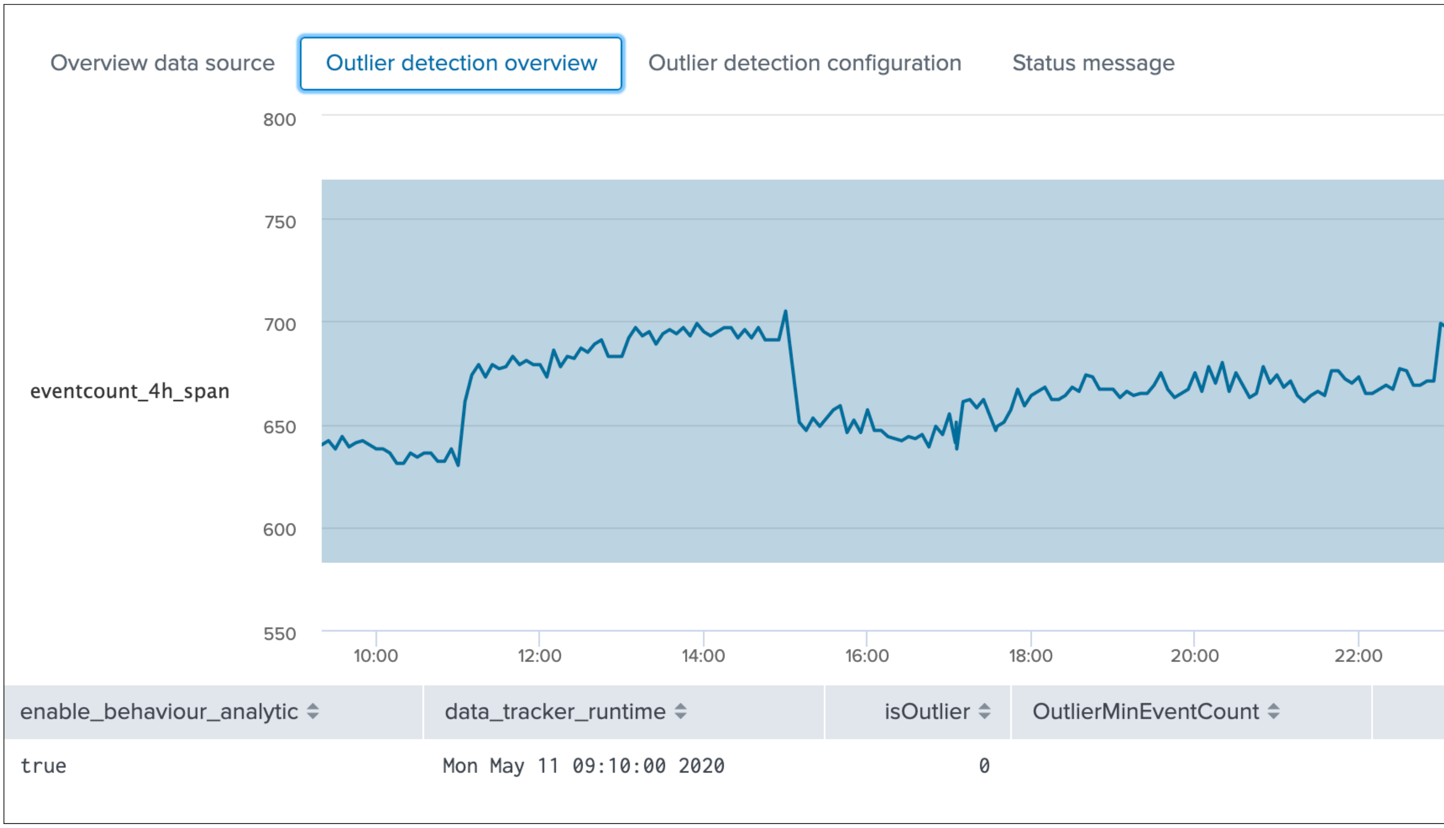
If TrackMe is running on the same search head than Enterprise Security and you wish to use its assets knowledge, customize the macro with `` `get_asset(data_host)` `` for da
If Enterprise Security is running on a different search head, one option is to define a summary scheduled report on the ES search head, then a scheduled report that will us
Customize the macro with a call to lookup `lookup name_of_lookup key as data_host` for data_hosts, and `lookup name_of_lookup key as metric_host` for metric_hosts.

Any kind of CMDB data available in Splunk:

Similarly you can use any lookup available in the Splunk instance which provides Assets context looking up a key which in most cases would be host name, dns name or IF
Make sure your asset lookup definition is exported to the system, is case insensitive and contains the relevant information, then customize the macros depending on your
`lookup name_of_lookup key as metric_hosts` for metric hosts.

Using Outliers detection

By default, the outlier detection is automatically activated for each data source and host, use the Outliers Overview tab to visualize the status of the Outliers detection:



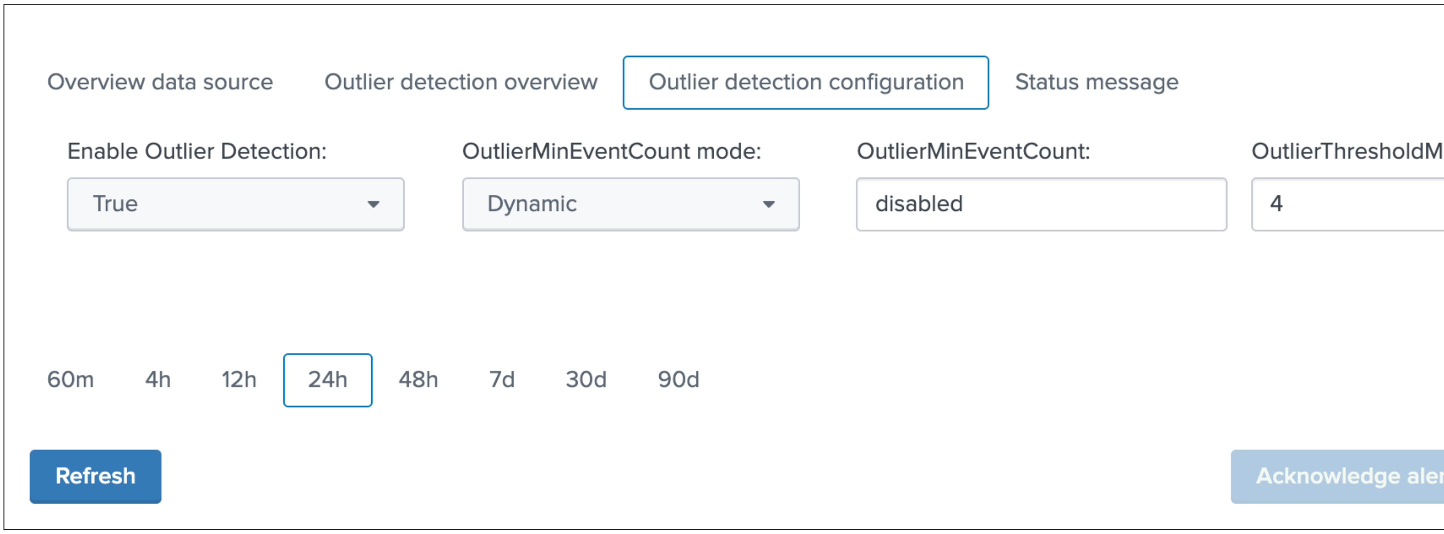
enable_behaviour_analytic	data_tracker_runtime	isOutlier	OutlierMinEventCount
true	Mon May 11 09:10:00 2020	0	

The table exposes the very last result from the analysis:

field	Purpose
enable outlier	defines if behaviour analytic should be enabled or disabled for that source (default to true)
alert on upper	defines if outliers detection going over the upper calculations (default to false)
data_tracker_runtime	last run time of the Summary Investigator tracker which defines the statuses of Outliers detection
isOutlier	main flag for Outlier detection, 0=no Outliers detected, 1=Outliers detected
OutlierMinEventCount	static lower bound value used with static mode, in dynamic mode this is not set
lower multiplier	default to 4, modifying the value influences the lower bound calculations based on the data
upper multiplier	default to 4, modifying the value influences the upper bound calculations based on the data
lowerBound/upperBound	exposes latest values for the lower and upper bound
stddev	exposes the latest value for the standard deviation calculated for that source

Simulating and adjusting Outliers detection

Use the Outliers detection configuration tab to run simulations and proceed to configuration adjustments:



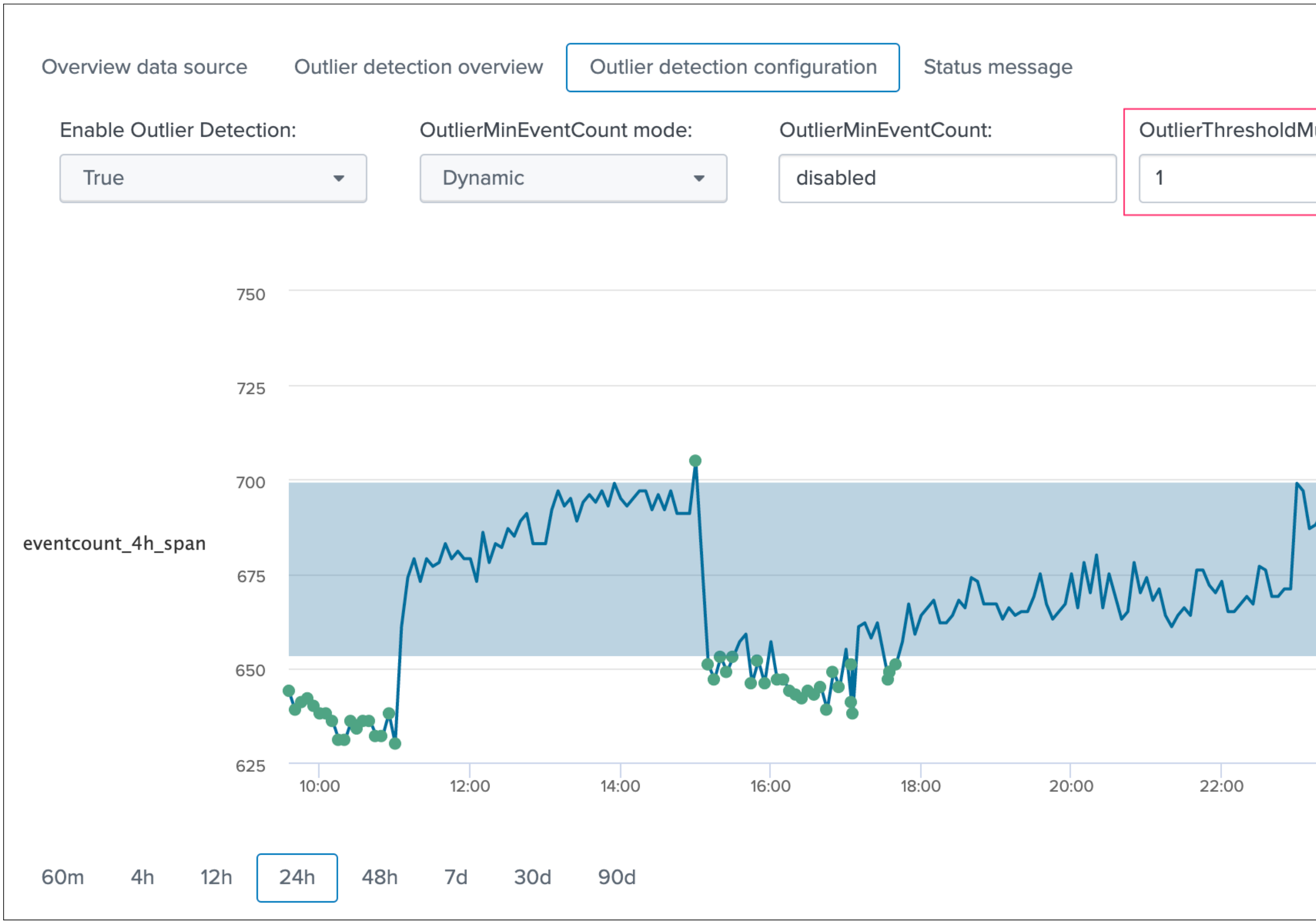
Enable Outlier Detection: True

OutlierMinEventCount mode: Dynamic

OutlierMinEventCount: disabled

OutlierThresholdM 4

For example, you can increase the value of the threshold multiplier to improve the outliers detection in regard with your knowledge of this data, or how its distribution behaves over time:



Enable Outlier Detection: True

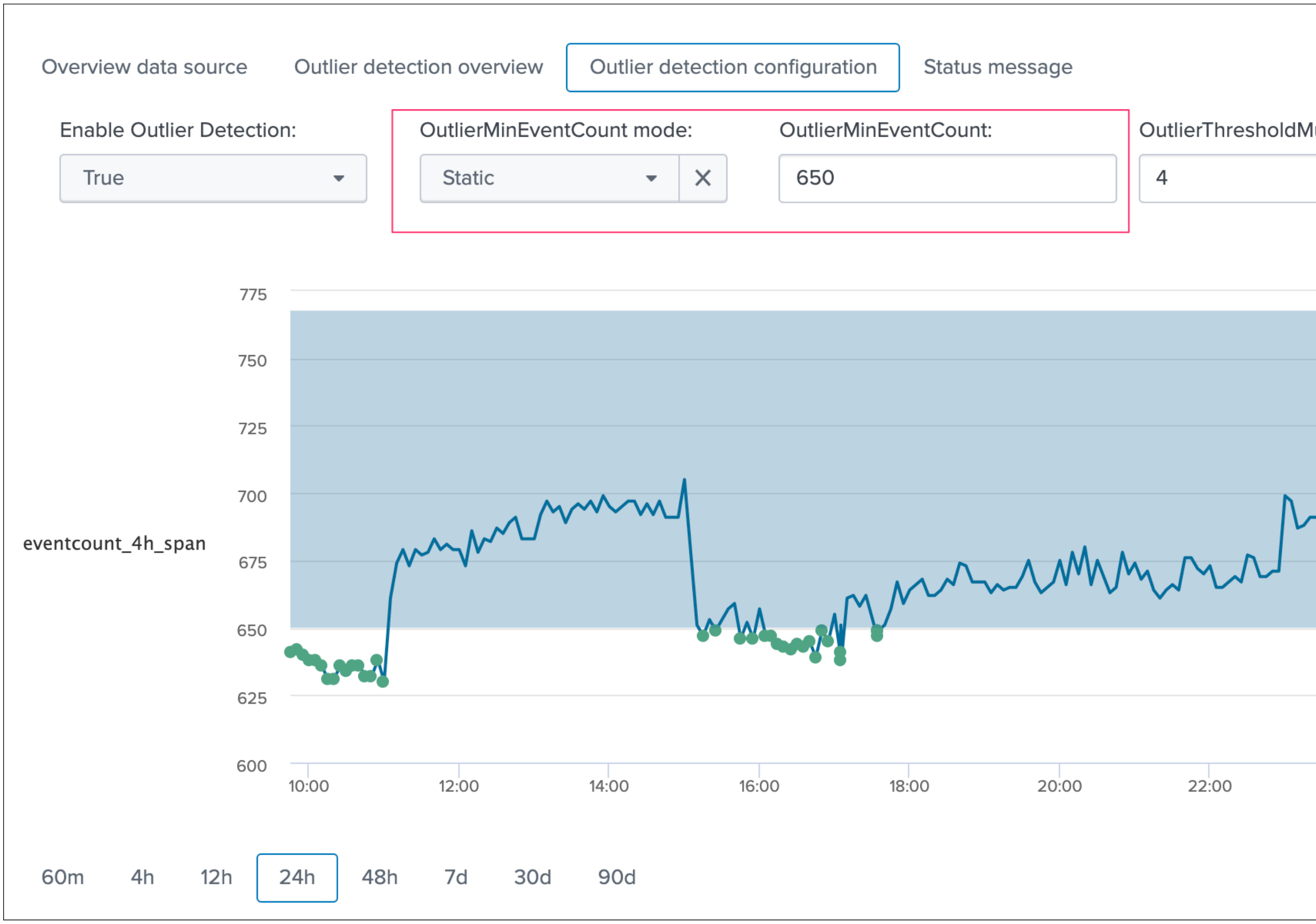
OutlierMinEventCount mode: Dynamic

OutlierMinEventCount: disabled

OutlierThresholdM 1

As well, in some cases you may wish to use a static lower bound value, if you use the static mode, then the outlier detection for the lower band is not used anymore and replaced by this static value as the minimal number of

events:



Overview data source | Outlier detection overview | Outlier detection configuration | Status message

Enable Outlier Detection: True

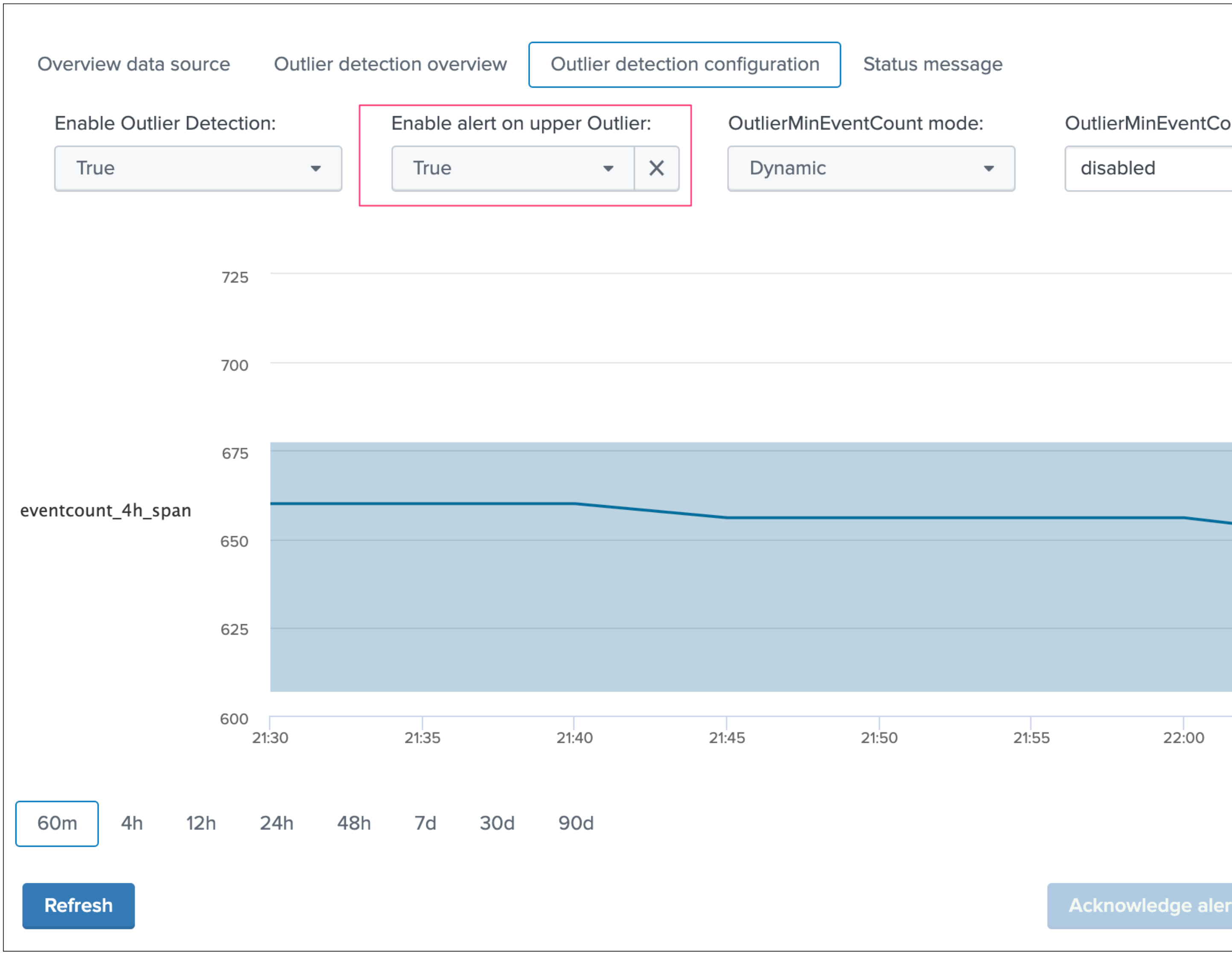
OutlierMinEventCount mode: Static

OutlierMinEventCount: 650

OutlierThresholdM

4

Upper bound outliers detection does not affect the alert status by default, however this option can be enabled and the threshold multiplier be customised if you need to detect a large increase in the volume of data generated by this source:



Enable Outlier Detection: True

Enable alert on upper Outlier: True

OutlierMinEventCount mode: Dynamic

OutlierMinEventCo disabled

Saving the configuration

Once you have validated the results from the simulation, click on the save button to immediately record the values to the KVstore collection.

When the save action is executed, you might need to wait a few minutes for it to be reported during the next execution of the Summary Investigator report.

3.1.5 Data sampling and event formats recognition

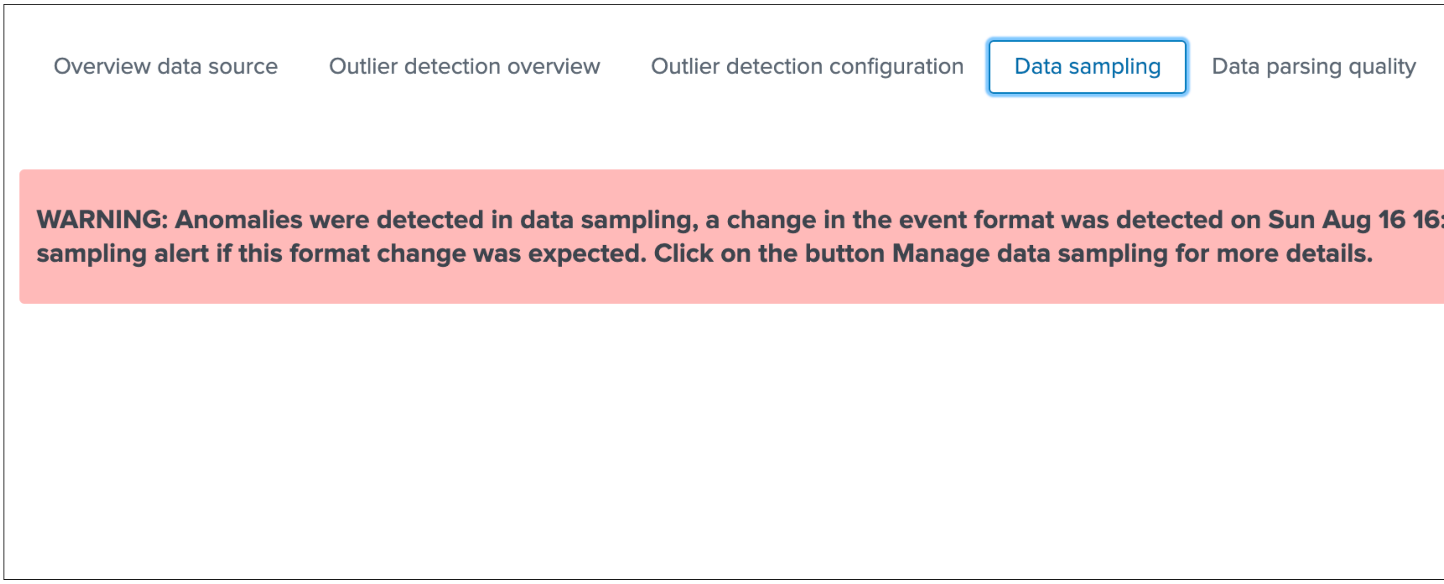
Data sampling and event format recognition

The Data sampling and event format recognition feature is a powerful automated workflow that provides the capabilities to monitor the raw events formats to automatically detect anomalies and misbehaviour at scale:

- TrackMe automatically picks a sample of from every data source on a scheduled basis, and runs regular expression based rules to find "good" and "bad" things

- builtin rules are provided to identify commonly used formats of data, such as syslog, json, xml, and so forth
- custom rules can be created to extend the feature up to your needs
- rules can be created as rules that need to be matched (looking for a format or specific patterns), or as rules that must not be matched (for example looking for PII data)
- rules that must not match (exclusive rules) are always proceeded before rules that must match (inclusive), this guarantes that if any a same data source would match multiple rules, any first rule matching "bad" things will proceed before a rule matching "good" things (as the engine will stop at the first match for a given event)
- The number of events sampled during each execution can be configured per data source, and otherwise defaults to 100 events at the first sampling, and 50 events for each new execution
- checkout custom rule example creation in the present documentation
- since the version 1.2.35, you can choose to obfuscate the sampled events that are normally stored in the collection, this might be required to avoid unwanted data accesses if you have a population of users in TrackMe who need to have limited access

You access to the data sample feature on a per data source basis via the data sample tab when looking at a specific data source:

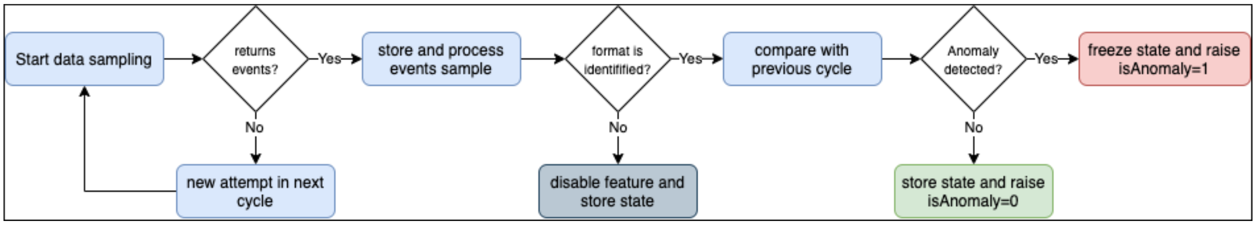


WARNING: Anomalies were detected in data sampling, a change in the event format was detected on Sun Aug 16 16:
sampling alert if this format change was expected. Click on the button Manage data sampling for more details.

How things work:

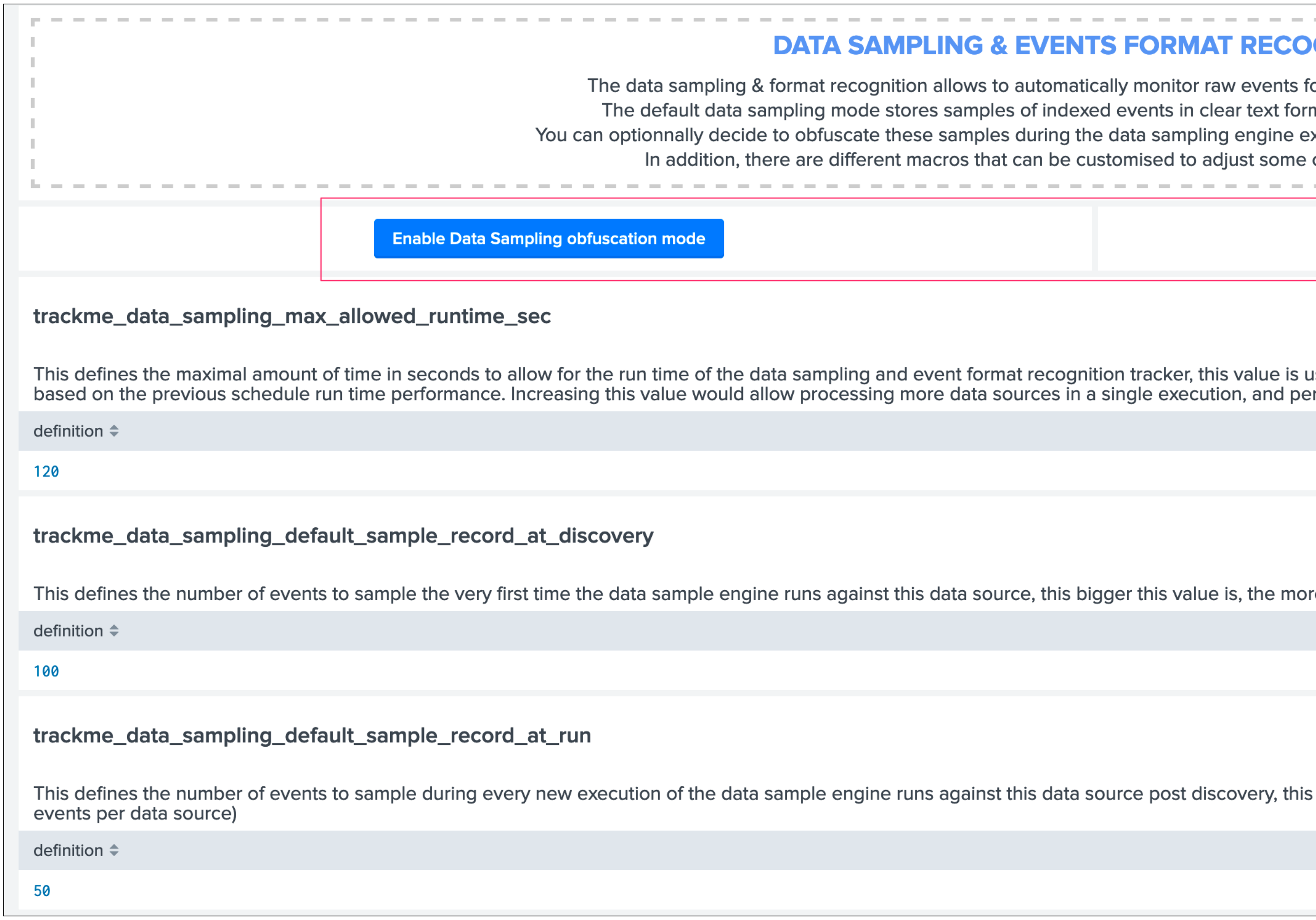
- The scheduled report named `TrackMe - Data sampling and format detection tracker` runs by default every 15 minutes
- The report uses a builtin function to determine an ideal number of data sources to be processed according to the total number of data sources to be processed, and the historical performance of the search (generates a rate per second extrapolated to limit the number of sources to be processed)
- For each data source to be processed, a given number of raw events is sampled and stored in a KVstore collection named `trackme_data_sampling`
- The number of raw events to be sampled depends on wether the data source is handled for the first time (discovery), or if it is a normal run
- On each sample per data source, the engine processes the events and applies custom rules if any, then builtin rules are processed

- Depending on the conditions, a status and additional informational fields are determined and stored in the lookup collection
- The status stored as the field `isAnomaly` is loaded by the data sources trackers and taken into account for the global data source state analysis


Start data sampling
returns events?
Yes
No
new attempt in next cycle
store and process events sample
format is identifified?
Yes
No
disable feature and store state
compare with previous cycle
Anomaly detected?
Yes
No
store state and raise isAnomaly=0
freeze state and raise isAnomaly=1

Data Sampling obfuscation mode

Access the configuration page from the navigation bar in TrackMe, "TrackMe manage and configure":



DATA SAMPLING & EVENTS FORMAT RECO

The data sampling & format recognition allows to automatically monitor raw events fo
The default data sampling mode stores samples of indexed events in clear text form
You can optionnally decide to obfuscate these samples during the data sampling engine ex
In addition, there are different macros that can be customised to adjust some c

trackme_data_sampling_max_allowed_runtime_sec

This defines the maximal amount of time in seconds to allow for the run time of the data sampling and event format recognition tracker, this value is u
based on the previous schedule run time performance. Increasing this value would allow processing more data sources in a single execution, and pe

definition ⇕
120

trackme_data_sampling_default_sample_record_at_discovery

This defines the number of events to sample the very first time the data sample engine runs against this data source, this bigger this value is, the more

definition ⇕
100

trackme_data_sampling_default_sample_record_at_run

This defines the number of events to sample during every new execution of the data sample engine runs against this data source post discovery, this
events per data source)

definition ⇕
50

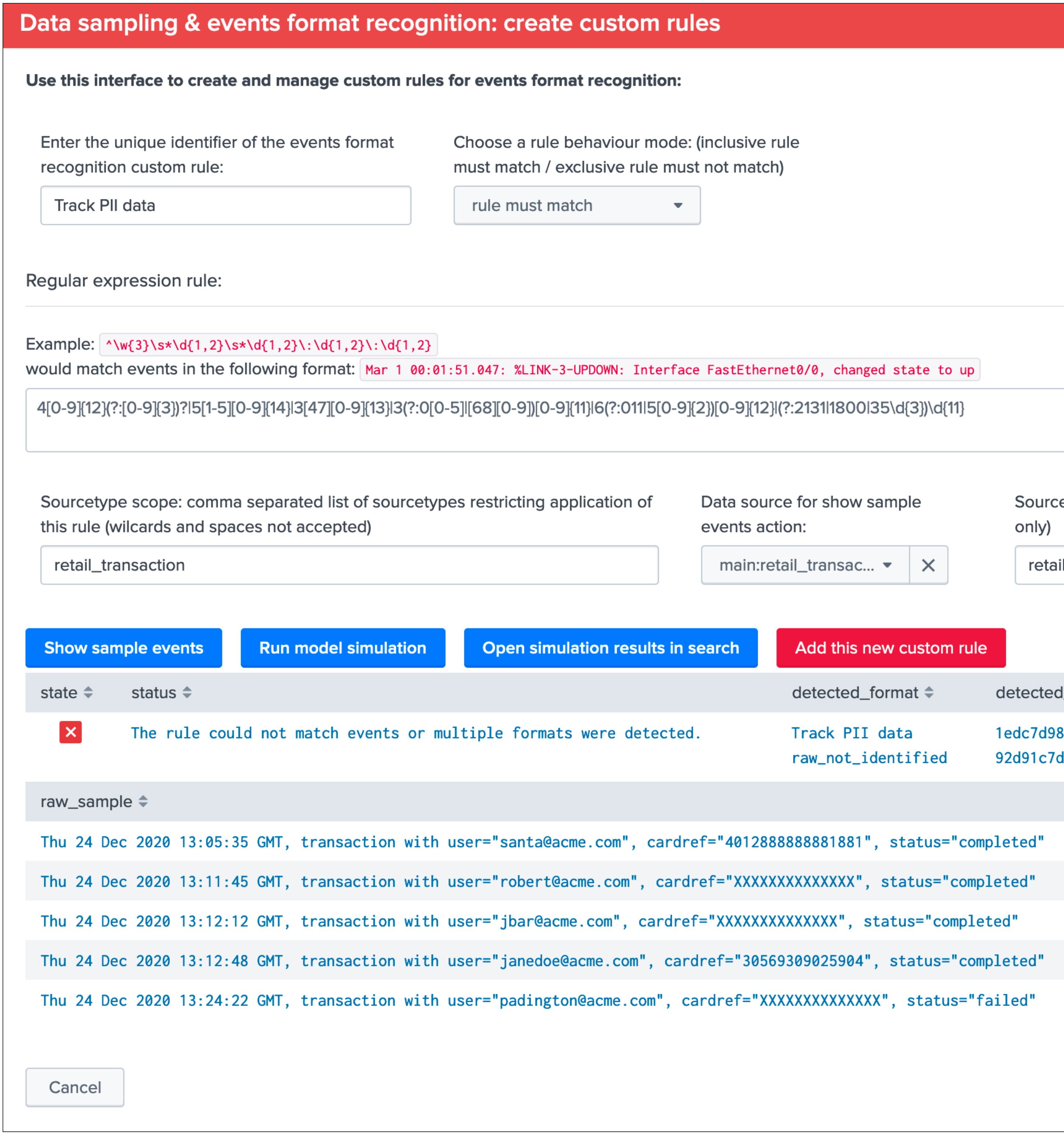
- In the default mode, that is `Disable Data Sampling obfuscation mode`, events that are sampled are stored in the data sampling KVstore collection and can be used to review the results from the latest sampling operation
- In the `Enable Data Sampling obfuscation mode`, events are not stored anymore and replaced by

an admin message, the sampling processing still happens the same way but events cannot be reviewed anymore using the latest sample traces

- In such a case, when then obfuscation mode is enabled, users will need to either run the rules manually to locate the messages that were captured to the conditions being met (bad format, PII data, etc) or use the Smart *Smart Status* feature to have TrackMe run this operation on demand

As a summary, you can enable the obfuscation mode if you have for instance a population of non admin users in TrackMe and you need to prevent them from accessing events they are not supposed to be able to accesss according to your RBAC policies in Splunk.

When a user attempts to create a new custom Data Sampling rule, the UI provides event sampling extracts:



Data sampling & events format recognition: create custom rules

Use this interface to create and manage custom rules for events format recognition:

Enter the unique identifier of the events format recognition custom rule: Track PII data

Choose a rule behaviour mode: (inclusive rule must match / exclusive rule must not match) rule must match

Regular expression rule:

Example: `^\w{3}\s*\d{1,2}\s*\d{1,2}\:\d{1,2}\:\d{1,2}`
would match events in the following format: `Mar 1 00:01:51.047: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up`

```
4[0-9]{12}(?:[0-9]{3})?|5[1-5][0-9]{14}|3[47][0-9]{13}|3(?:0[0-5]|[68][0-9])[0-9]{11}|6(?:011|5[0-9]{2})[0-9]{12}|(?:2131|1800|35\d{3})\d{11}
```

Sourcetype scope: comma separated list of sourcetypes restricting application of this rule (wilcards and spaces not accepted): retail_transaction

Data source for show sample events action: main:retail_transac...

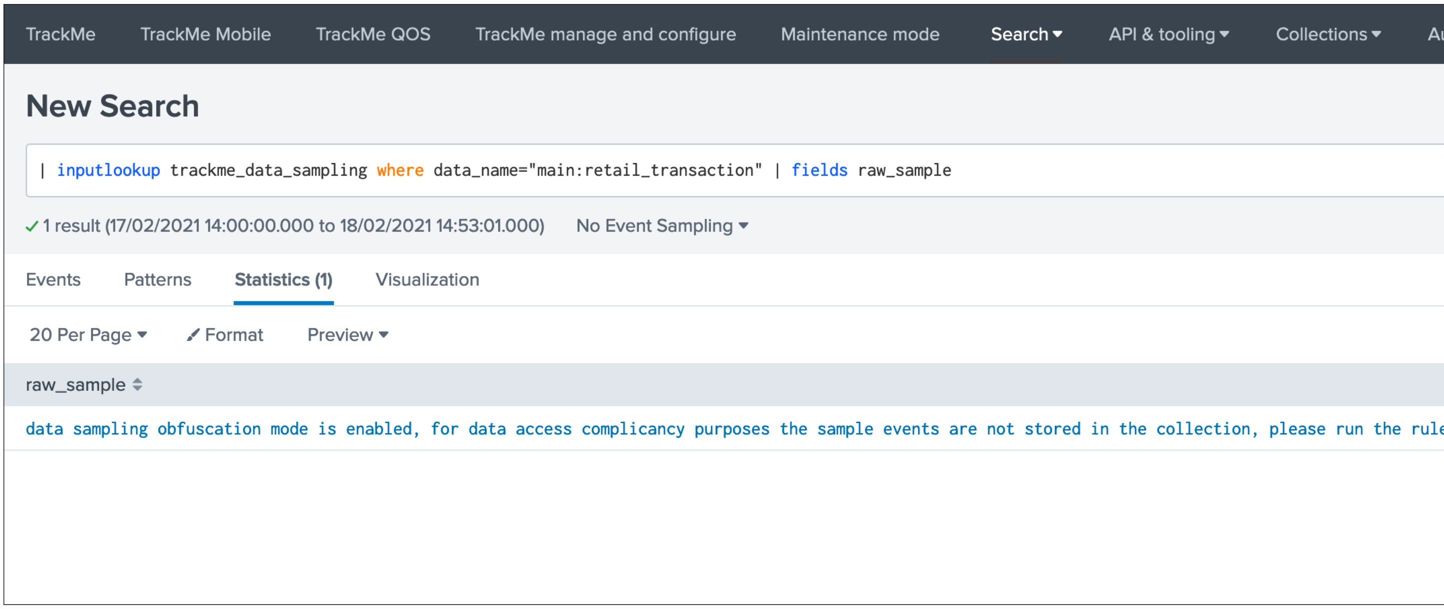
Source only) retail

state	status	detected_format	detected
☒	The rule could not match events or multiple formats were detected.	Track PII data raw_not_identified	1edc7d98 92d91c7d

raw_sample
Thu 24 Dec 2020 13:05:35 GMT, transaction with user="santa@acme.com", cardref="4012888888881881", status="completed"
Thu 24 Dec 2020 13:11:45 GMT, transaction with user="robert@acme.com", cardref="XXXXXXXXXXXXXX", status="completed"
Thu 24 Dec 2020 13:12:12 GMT, transaction with user="jbar@acme.com", cardref="XXXXXXXXXXXXXX", status="completed"
Thu 24 Dec 2020 13:12:48 GMT, transaction with user="janedoe@acme.com", cardref="30569309025904", status="completed"
Thu 24 Dec 2020 13:24:22 GMT, transaction with user="padington@acme.com", cardref="XXXXXXXXXXXXXX", status="failed"

These searches are performed on behalf on the user as normal Splunk searches, as such if the user cannot access to these data, there would be no results accessible.

When the obfuscation mode is enabled, trying to access to the latest sample events via the UI (or directly via access to the collection) would result in the following content:



```
| inputlookup trackme_data_sampling where data_name="main:retail_transaction" | fields raw_sample
```

✓ 1 result (17/02/2021 14:00:00.000 to 18/02/2021 14:53:01.000)

raw_sample
data sampling obfuscation mode is enabled, for data access complicancy purposes the sample events are not stored in the collection, please run the rule

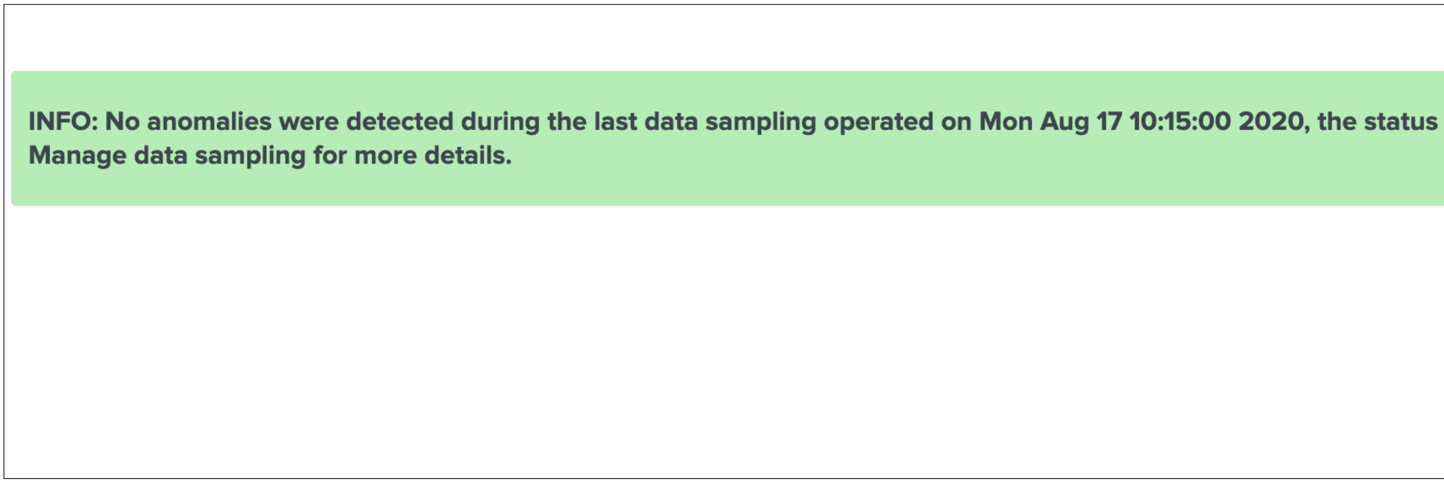
As a conclusion, enable the data sampling obfuscation mode if you are concerned about having users able to access to events they are not supposed to, when it is enabled, the collection cannot contain amymore any potentially sensitive information while the main and more valuable features are preserved.

Summary statuses

The data sampling message can be:

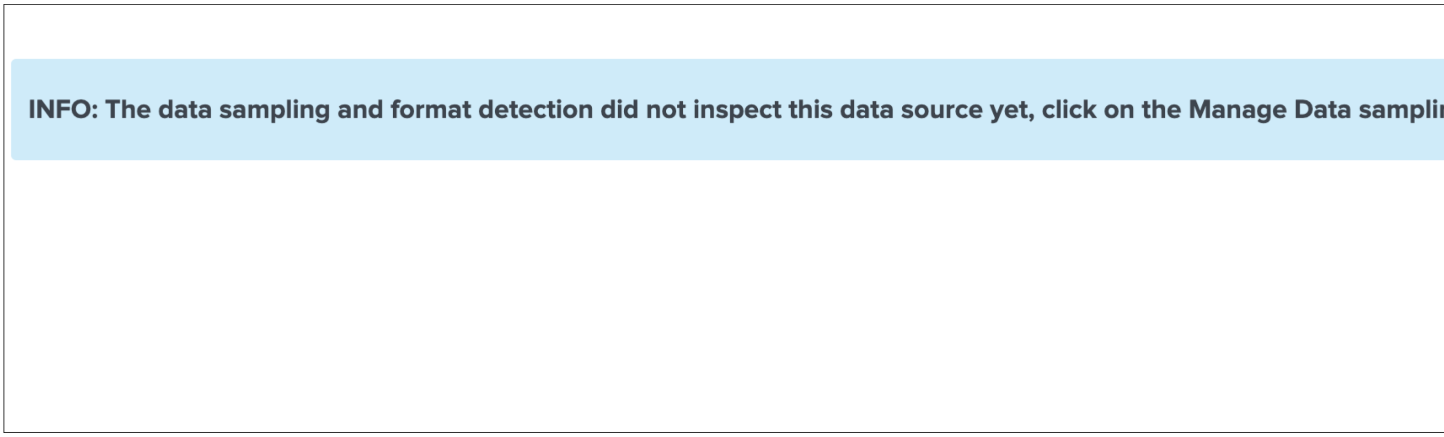
- `green`: if no anomalies were detected
- `blue`: if the data sampling did not handle this data source yet
- `orange`: if conditions do not allow to handle this data source, which can be multi-format detected at discovery, or no identifiable event formats (data sampling will be deactivated automatically)
- `red`: if anomalies were detected by the data engine, anomalies can be due to a change in the event format, or multiple events formats detected post discovery

Green state: no anomalies were detected, data sampling ran and is enabled



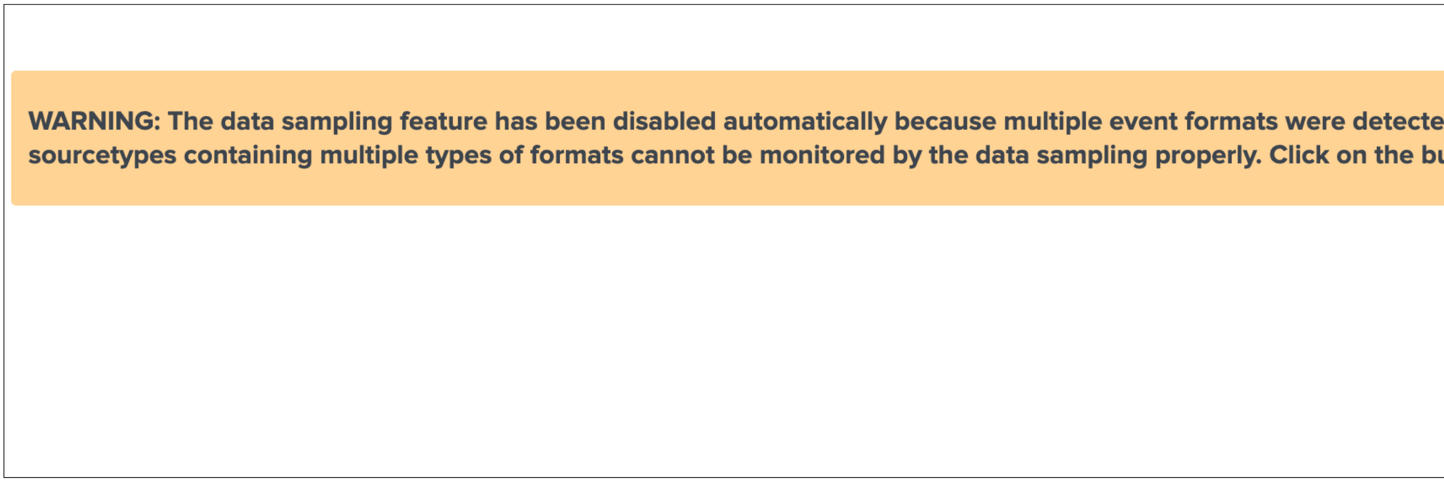
INFO: No anomalies were detected during the last data sampling operated on Mon Aug 17 10:15:00 2020, the status
Manage data sampling for more details.

Blue state: data sampling engine did not inspect this data source yet

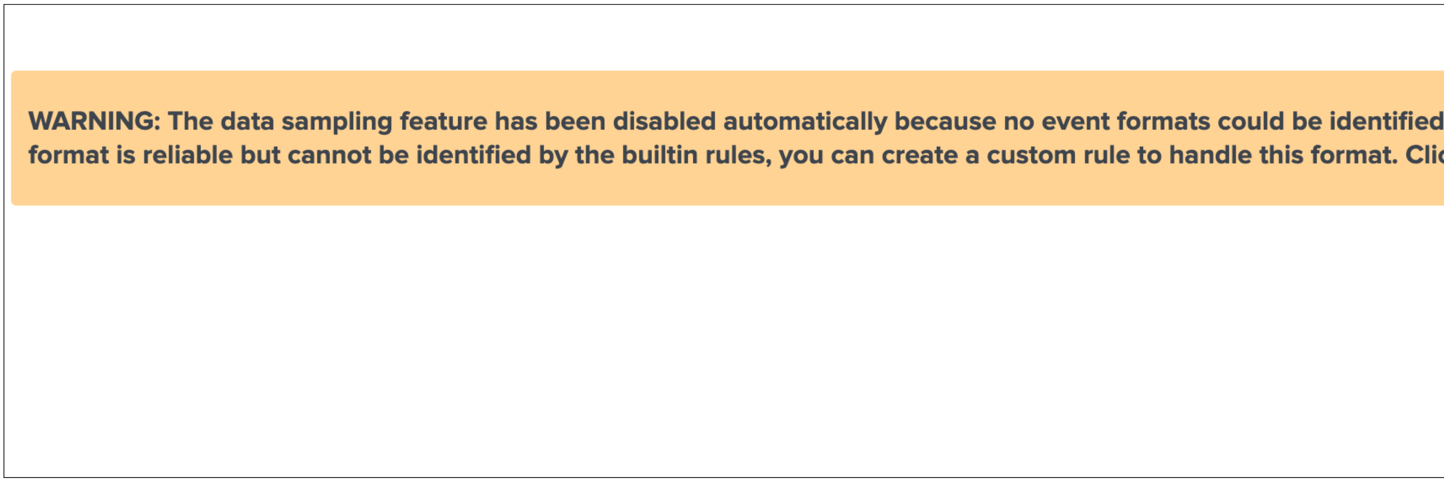


INFO: The data sampling and format detection did not inspect this data source yet, click on the Manage Data sampli

Orange state: data sampling was disabled due to events format recognition conditions that would not allow to manage this data properly (multiformat, no event formats identification possible)

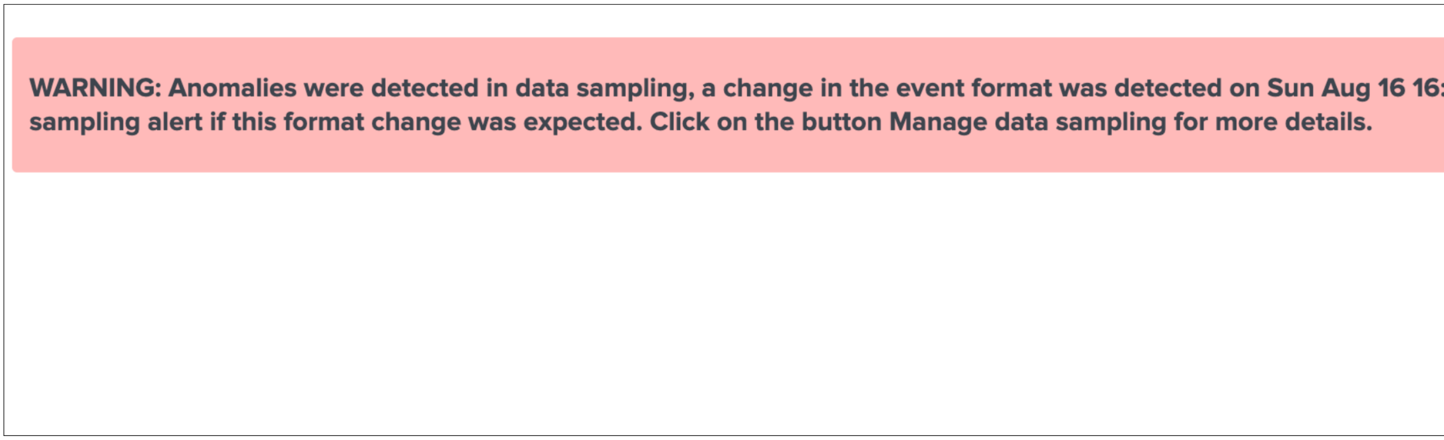


WARNING: The data sampling feature has been disabled automatically because multiple event formats were detecte
sourcetypes containing multiple types of formats cannot be monitored by the data sampling properly. Click on the bu



WARNING: The data sampling feature has been disabled automatically because no event formats could be identified
format is reliable but cannot be identified by the builtin rules, you can create a custom rule to handle this format. Clic

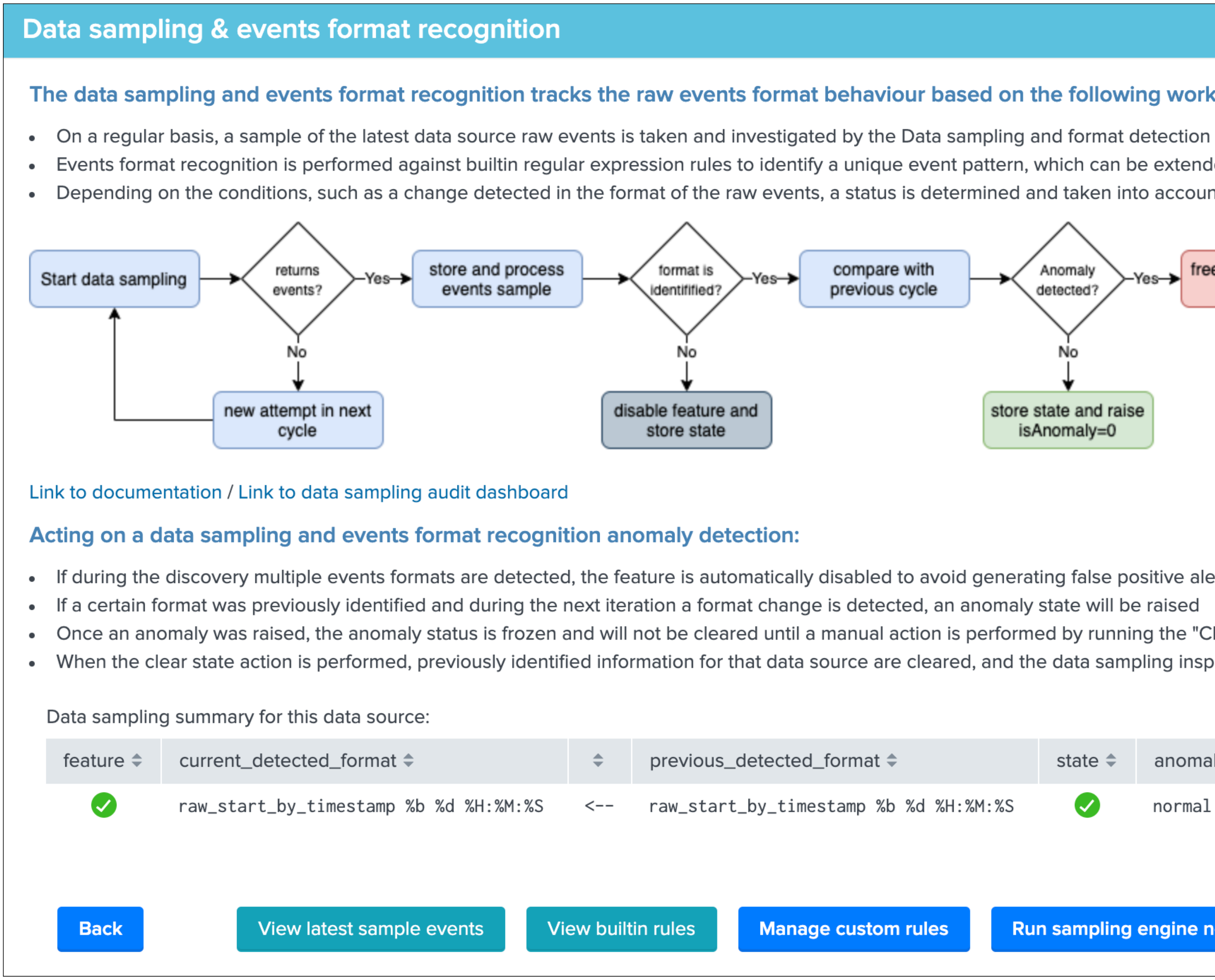
Red state: anomalies were detected



WARNING: Anomalies were detected in data sampling, a change in the event format was detected on Sun Aug 16 16:
sampling alert if this format change was expected. Click on the button Manage data sampling for more details.

Manage data sampling

The Manage data sampling button provides access to functions to review and configure the feature:



Data sampling & events format recognition

The data sampling and events format recognition tracks the raw events format behaviour based on the following work

- On a regular basis, a sample of the latest data source raw events is taken and investigated by the Data sampling and format detection
- Events format recognition is performed against builtin regular expression rules to identify a unique event pattern, which can be extende
- Depending on the conditions, such as a change detected in the format of the raw events, a status is determined and taken into accoun

Start data sampling
returns events?
Yes
No
new attempt in next cycle
store and process events sample
format is identifified?
Yes
No
disable feature and store state
compare with previous cycle
Anomaly detected?
Yes
No
store state and raise isAnomaly=0
free

Link to documentation / Link to data sampling audit dashboard

Acting on a data sampling and events format recognition anomaly detection:

- If during the discovery multiple events formats are detected, the feature is automatically disabled to avoid generating false positive ale
- If a certain format was previously identified and during the next iteration a format change is detected, an anomaly state will be raised
- Once an anomaly was raised, the anomaly status is frozen and will not be cleared until a manual action is performed by running the "Cl
- When the clear state action is performed, previously identified information for that data source are cleared, and the data sampling insp

Data sampling summary for this data source:

feature	current_detected_format		previous_detected_format	state	anomal
✓	raw_start_by_timestamp %b %d %H:%M:%S	<--	raw_start_by_timestamp %b %d %H:%M:%S	✓	normal

The summary table shows the main key information:

- `data_sample_feature`: is the data sampling feature enabled or disabled for that data source, rendered as an icon
- `current_detected_format`: the event format that has been detected during the last sampling
- `previous_detected_format`: the event format that was detected in the previous sampling
- `state`: the state of the data sampling rendered as an icon
- `anomaly_reason`: the reason why an anomaly is raised, or "normal" if there are no anomalies
- `multiformat`: shall more than one format of events be detected (true / false)
- `mtime`: the latest time data sampling was processed for this data source
- `data_sampling_nr`: the number of events taken per sampling operation, defaults to 100 events at discovery then 50 events for each new sampling (can be configured via the action Update records/sample)

View latest sample events

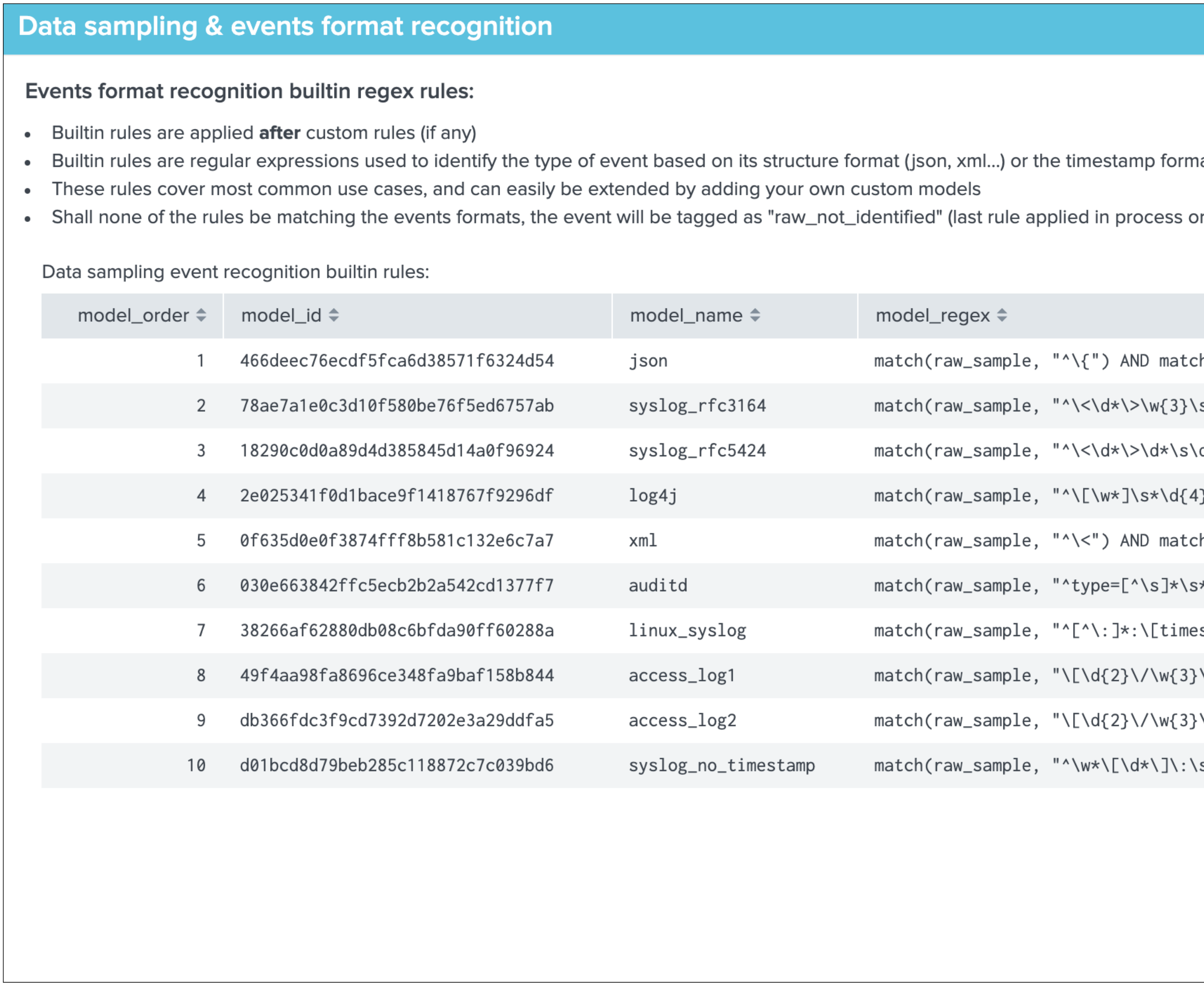
This button opens in the search UI the last sample of raw events that were processed for this data source, the search calls a macro which runs the events format recognitions rules as:

```
| inputlookup trackme_data_sampling where data_name="<data_name>" | fields raw_sample␣
↪| mvexpand raw_sample | `trackme_data_sampling_abstract_detect_events_format`
```

This view can be useful for trouble shooting purposes to determine why an anomaly was raised for a given data source.

View builtin rules

This button opens a new view that exposes the builtin rules used by TrackMe, and the order in which rules are processed:



Data sampling & events format recognition

Events format recognition builtin regex rules:

- Builtin rules are applied **after** custom rules (if any)
- Builtin rules are regular expressions used to identify the type of event based on its structure format (json, xml...) or the timestamp forma
- These rules cover most common use cases, and can easily be extended by adding your own custom models
- Shall none of the rules be matching the events formats, the event will be tagged as "raw_not_identified" (last rule applied in process or

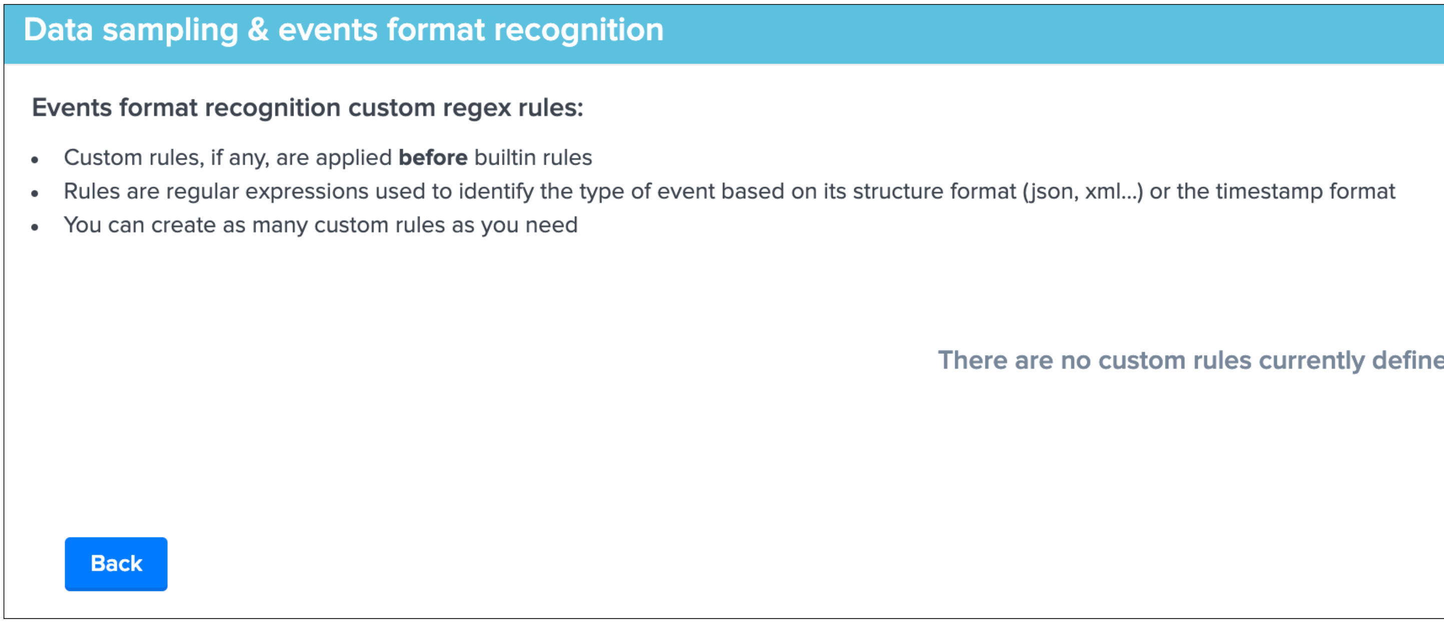
Data sampling event recognition builtin rules:

model_order	model_id	model_name	model_regex
1	466deec76ecdf5fca6d38571f6324d54	json	match(raw_sample, "^\{") AND matc
2	78ae7a1e0c3d10f580be76f5ed6757ab	syslog_rfc3164	match(raw_sample, "^\<\d*\>\w{3}\s
3	18290c0d0a89d4d385845d14a0f96924	syslog_rfc5424	match(raw_sample, "^\<\d*\>\d*\s\
4	2e025341f0d1bace9f1418767f9296df	log4j	match(raw_sample, "^\[\w*]\s*\d{4}
5	0f635d0e0f3874fff8b581c132e6c7a7	xml	match(raw_sample, "^\<") AND matc
6	030e663842ffc5ecb2b2a542cd1377f7	auditd	match(raw_sample, "^type=[^\s]*\s
7	38266af62880db08c6bfda90ff60288a	linux_syslog	match(raw_sample, "^[^\:]*:\[times
8	49f4aa98fa8696ce348fa9baf158b844	access_log1	match(raw_sample, "\[\d{2}\/\w{3}\
9	db366fdc3f9cd7392d7202e3a29ddfa5	access_log2	match(raw_sample, "\[\d{2}\/\w{3}\
10	d01bcd8d79beb285c118872c7c039bd6	syslog_no_timestamp	match(raw_sample, "^\w*\[\d*\]\:\s

Builtin rules should not be modified, instead use custom rules to handle event formats that would not be properly identified by the builtin regular expression rules.

Manage custom rules

Custom rules provides a workflow to handle any custom sourcetypes and event formats that would not be identified by TrackMe, or patterns that must not be matched, by default there are no custom rules and the following screen would appear:



Data sampling & events format recognition

Events format recognition custom regex rules:

- Custom rules, if any, are applied **before** builtin rules
- Rules are regular expressions used to identify the type of event based on its structure format (json, xml...) or the timestamp format
- You can create as many custom rules as you need

There are no custom rules currently define

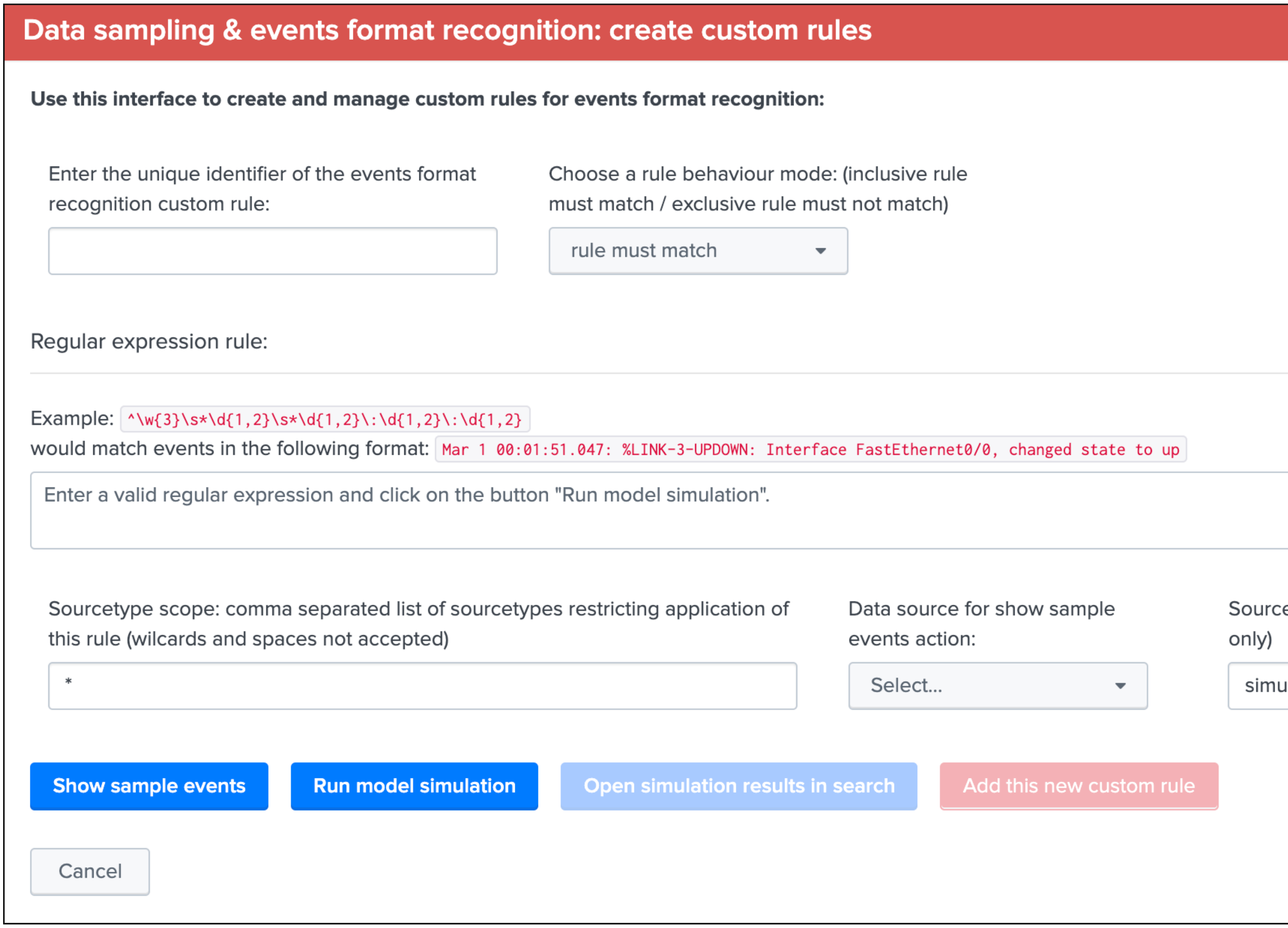
This view allows you to create a new custom rule (button Create custom rules) or remove any existing custom rules that would not be required anymore. (button Remove selected)

Tip: Each custom rule can be restricted to a given list of explicit sourcetypes, or applied against any sourcetype. (default)

Create custom rules

This screen alows to test and create a new custom rule based on the current data source:

Note: While you create a new custom rule via a specific data source, custom rules are applied to all data sources



Data sampling & events format recognition: create custom rules

Use this interface to create and manage custom rules for events format recognition:

Enter the unique identifier of the events format recognition custom rule:

Choose a rule behaviour mode: (inclusive rule must match / exclusive rule must not match)

rule must match

Regular expression rule:

Example: `^\w{3}\s*\d{1,2}\s*\d{1,2}\:\d{1,2}\:\d{1,2}`

would match events in the following format: `Mar 1 00:01:51.047: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up`

Enter a valid regular expression and click on the button "Run model simulation".

Sourcetype scope: comma separated list of sourcetypes restricting application of this rule (wilcards and spaces not accepted)

*

Data source for show sample events action:

Select...

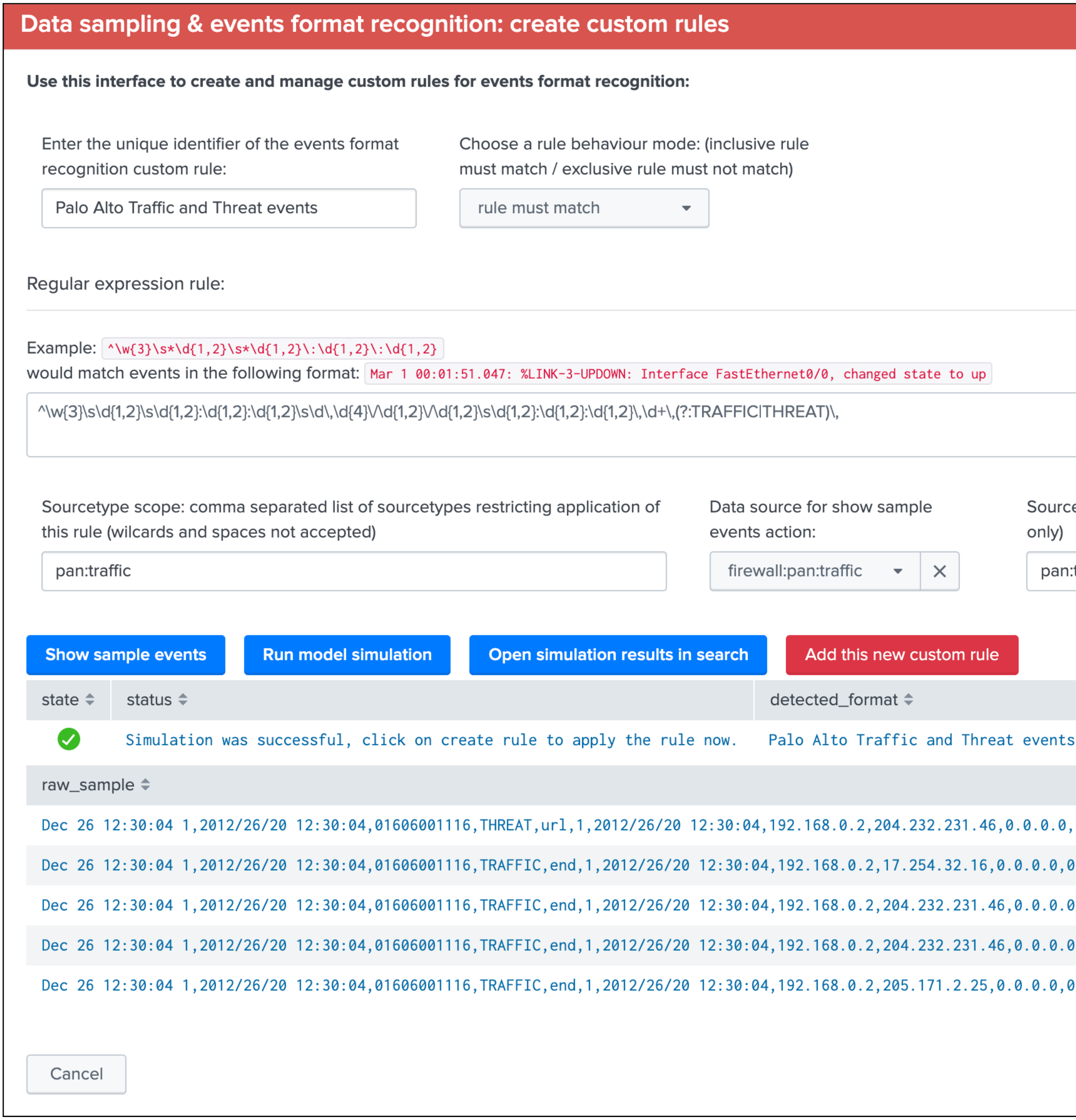
Source only)

simu

To create a new custom rule:

- Enter a name for the rule, this value is a string of your choice that will be used to idenfity the match, it needs to be unique for the entire custom source collection and will be converted into an md5 hash automatically
- Choose if the rule is a "rule must match" or "rule must not match" type of rule, this will drive the match behaviour to define the state of the data sampling results
- Enter a valid regular expression that uniquely identifies the events format
- Optionally restrict the scope of application by sourcetype, you can specify one or more sourcetypes under the form of a comma separated list of values
- Click on "Run model simulation" to simulate the exectution of the new models
- Optionnaly click on "Show sample events" to view a mini sample of the events within the screen
- Optionnaly click on ""Open simulation results in search" to open the details of the rules processing per event in the search UI
- Finally if the status of the simulation is valid, click on "Add this new custom rule" to permanently add this new custom rule

Example:



Data sampling & events format recognition: create custom rules

Use this interface to create and manage custom rules for events format recognition:

Enter the unique identifier of the events format recognition custom rule: Palo Alto Traffic and Threat events

Choose a rule behaviour mode: (inclusive rule must match / exclusive rule must not match) rule must match

Regular expression rule:

Example: `^\w{3}\s*\d{1,2}\s*\d{1,2}\:\d{1,2}\:\d{1,2}`
would match events in the following format: `Mar 1 00:01:51.047: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up`

```
^\w{3}\s\d{1,2}\s\d{1,2}:\d{1,2}:\d{1,2}\s\d\,\d{4}\/\d{1,2}\/\d{1,2}\s\d{1,2}:\d{1,2}:\d{1,2}\,\d+\,(?:TRAFFIC|THREAT)\,
```

Sourcetype scope: comma separated list of sourcetypes restricting application of this rule (wilcards and spaces not accepted): pan:traffic

Data source for show sample events action: firewall:pan:traffic

Source
only)
pan:t

state	status	detected_format
✓	Simulation was successful, click on create rule to apply the rule now.	Palo Alto Traffic and Threat events

```
raw_sample
Dec 26 12:30:04 1,2012/26/20 12:30:04,01606001116,THREAT,url,1,2012/26/20 12:30:04,192.168.0.2,204.232.231.46,0.0.0.0,
Dec 26 12:30:04 1,2012/26/20 12:30:04,01606001116,TRAFFIC,end,1,2012/26/20 12:30:04,192.168.0.2,17.254.32.16,0.0.0.0,0
Dec 26 12:30:04 1,2012/26/20 12:30:04,01606001116,TRAFFIC,end,1,2012/26/20 12:30:04,192.168.0.2,204.232.231.46,0.0.0.0
Dec 26 12:30:04 1,2012/26/20 12:30:04,01606001116,TRAFFIC,end,1,2012/26/20 12:30:04,192.168.0.2,204.232.231.46,0.0.0.0
Dec 26 12:30:04 1,2012/26/20 12:30:04,01606001116,TRAFFIC,end,1,2012/26/20 12:30:04,192.168.0.2,205.171.2.25,0.0.0.0,0
```

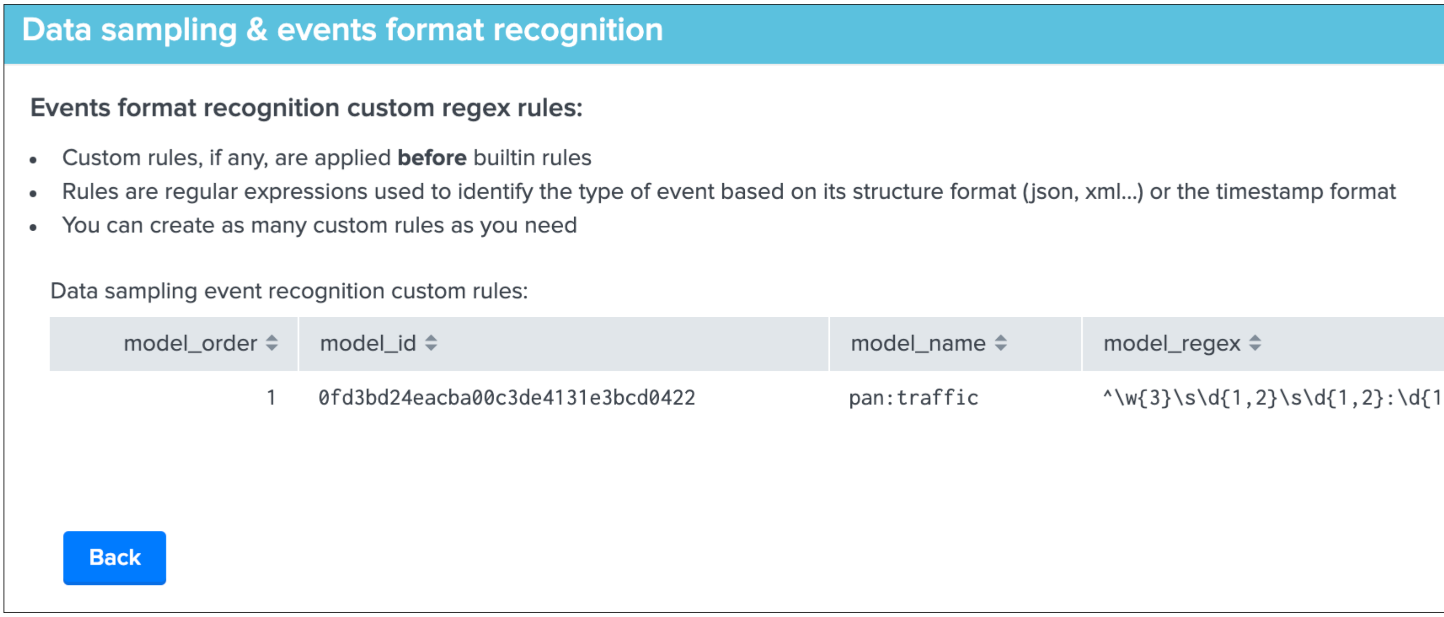
Once you have created a new custom rule, this rule will be applied automatically to future executions of the data sampling engine:

- If the format switches from a format idenfitied by the the builtin rules to a format identified by a custom rule, it will not appear in anomaly
- You can optionally clear the state of the data sampling for that data source to clean any previous states and force

a new discovery

Remove custom rules

Once there is at least one custom rule defined, the list of custom rules appears in the table and can be selected for suppression:



Data sampling & events format recognition

Events format recognition custom regex rules:

- Custom rules, if any, are applied **before** builtin rules
- Rules are regular expressions used to identify the type of event based on its structure format (json, xml...) or the timestamp format
- You can create as many custom rules as you need

Data sampling event recognition custom rules:

model_order	model_id	model_name	model_regex
1	0fd3bd24eacba00c3de4131e3bcd0422	pan:traffic	^\w{3}\s\d{1,2}\s\d{1,2}:\d{1

When a custom rule is removed, future executions of the data sampling engine will not consider the rule deleted anymore, optionally you can run the data sampling engine now or clear the state for a data source.

Custom rules are stored in a KVstore collection which can as well be manually edited if you need to update an exising rule, or modify the order in which rules are processed:

```
trackme_data_sampling_custom_models
```

Run sampling engine now

Use this function to force running the data sampling engine now against this data source, this will not force a new discovery and will run the data sampling engine normally. (the current status is preserved)

When to use the run sampling engine now?

- You can can run this action at anytime and as often as you need, the action runs the data sampling engine for that data source only
- This action will have no effect if an anomaly was raised for the data source already, when an anomaly is detected the status is frozen (see Clear state and run sampling)

Update records/sample

You can define a custom number of events to be taken per sample using this action button within the UI.

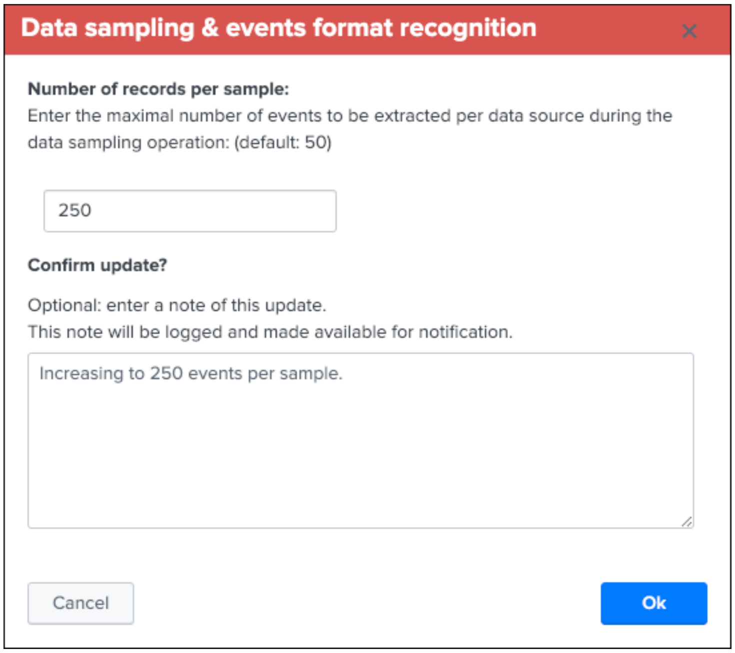
By default, the Data sampling proceeds as following:

- When the first iteration for a given data source is processed, TrackMe picks a sample of 100 events
- During every new iteration, a sample of 50 events is taken

In addition, these values are defined globally for the application via the following macros:

- trackme_data_sampling_default_sample_record_at_discovery
- trackme_data_sampling_default_sample_record_at_run

Use this UI to choose a different value, increasing the number of events per sample improves the sampling process accuracy, at the cost of more processing and more memory and storage costs for the KVstore collection:



Data sampling & events format recognition

Number of records per sample:
Enter the maximal number of events to be extracted per data source during the data sampling operation: (default: 50)

250

Confirm update?

Optional: enter a note of this update.
This note will be logged and made available for notification.

Increasing to 250 events per sample.

Clear state and run sampling

Use this function to clear any state previously determined, this forces the data source to be considered as it was the first time it was investigated by the data sampling engine. (a full sampling is processed and there are no prior status taken into account)

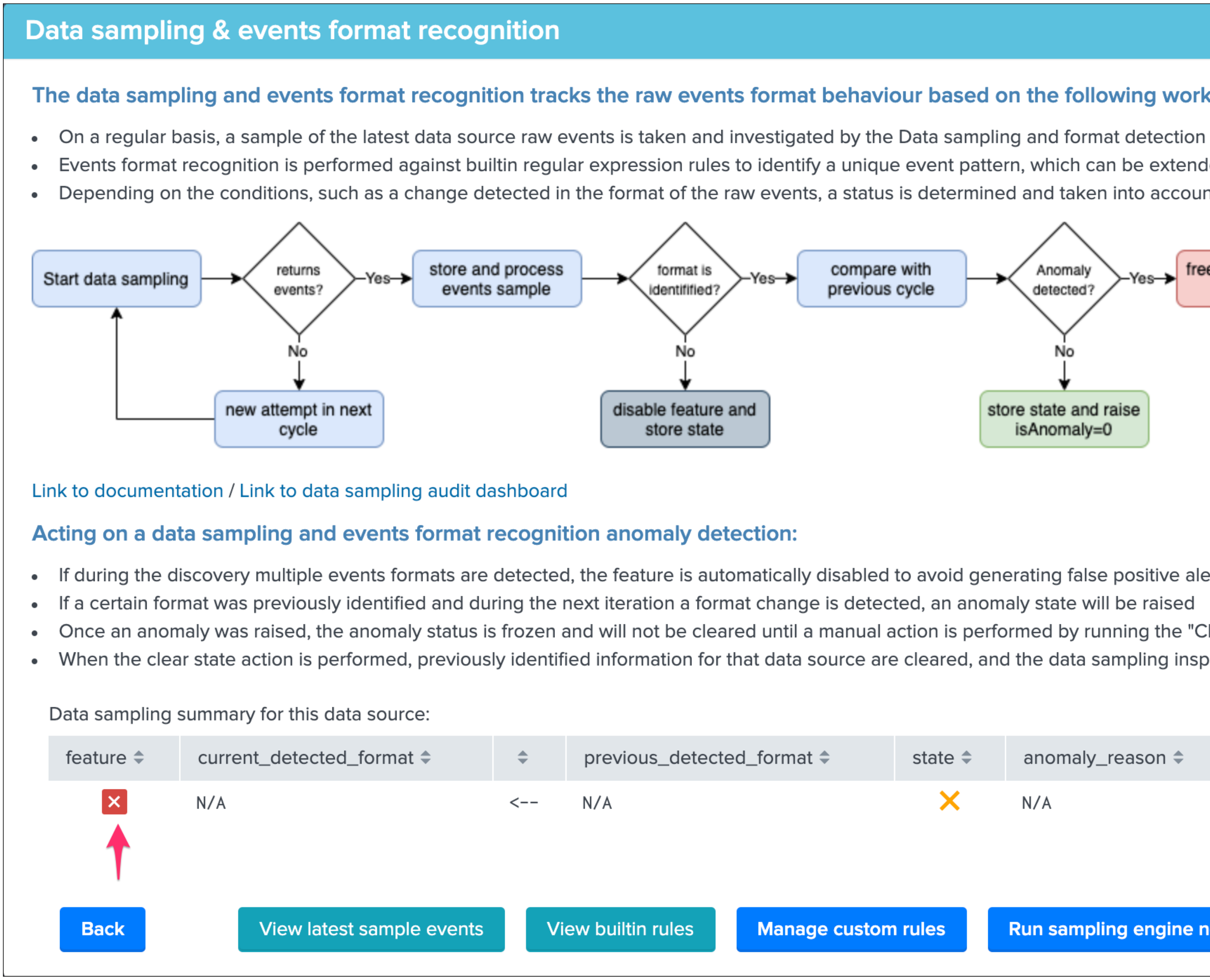
When to use the clear state and run sampling?

- Use this action to clear any known states for this data source and run the inspection from zero, just as if it was discovered for the first time
- You can use this action to clear an anomaly that was raised, when an alert is raised by the data sampling, the state is frozen until this anomaly is reviewed, once the issue is understood and fixed, run the action to clear the state and restart the inspection workflow for this data source

Disable Data sampling for a give data source

Use this function to disable data sampling for a given data source, there can be cases where you would need to disable this feature if for example there is a lack of quality which cannot be fixed, and some random formats are introduced out of your control.

Disabling the feature means defining the value of the field **data_sample_feature** to **disabled** in the collection trackme_data_sampling, once disabled the UI would show:



Data sampling & events format recognition

The data sampling and events format recognition tracks the raw events format behaviour based on the following work

- On a regular basis, a sample of the latest data source raw events is taken and investigated by the Data sampling and format detection
- Events format recognition is performed against builtin regular expression rules to identify a unique event pattern, which can be extend
- Depending on the conditions, such as a change detected in the format of the raw events, a status is determined and taken into accoun

Start data sampling
returns events?
Yes
No
new attempt in next cycle
store and process events sample
format is identifified?
Yes
No
disable feature and store state
compare with previous cycle
Anomaly detected?
Yes
No
store state and raise isAnomaly=0
free

Link to documentation / Link to data sampling audit dashboard

Acting on a data sampling and events format recognition anomaly detection:

- If during the discovery multiple events formats are detected, the feature is automatically disabled to avoid generating false positive ale
- If a certain format was previously identified and during the next iteration a format change is detected, an anomaly state will be raised
- Once an anomaly was raised, the anomaly status is frozen and will not be cleared until a manual action is performed by running the "Cl
- When the clear state action is performed, previously identified information for that data source are cleared, and the data sampling insp

Data sampling summary for this data source:

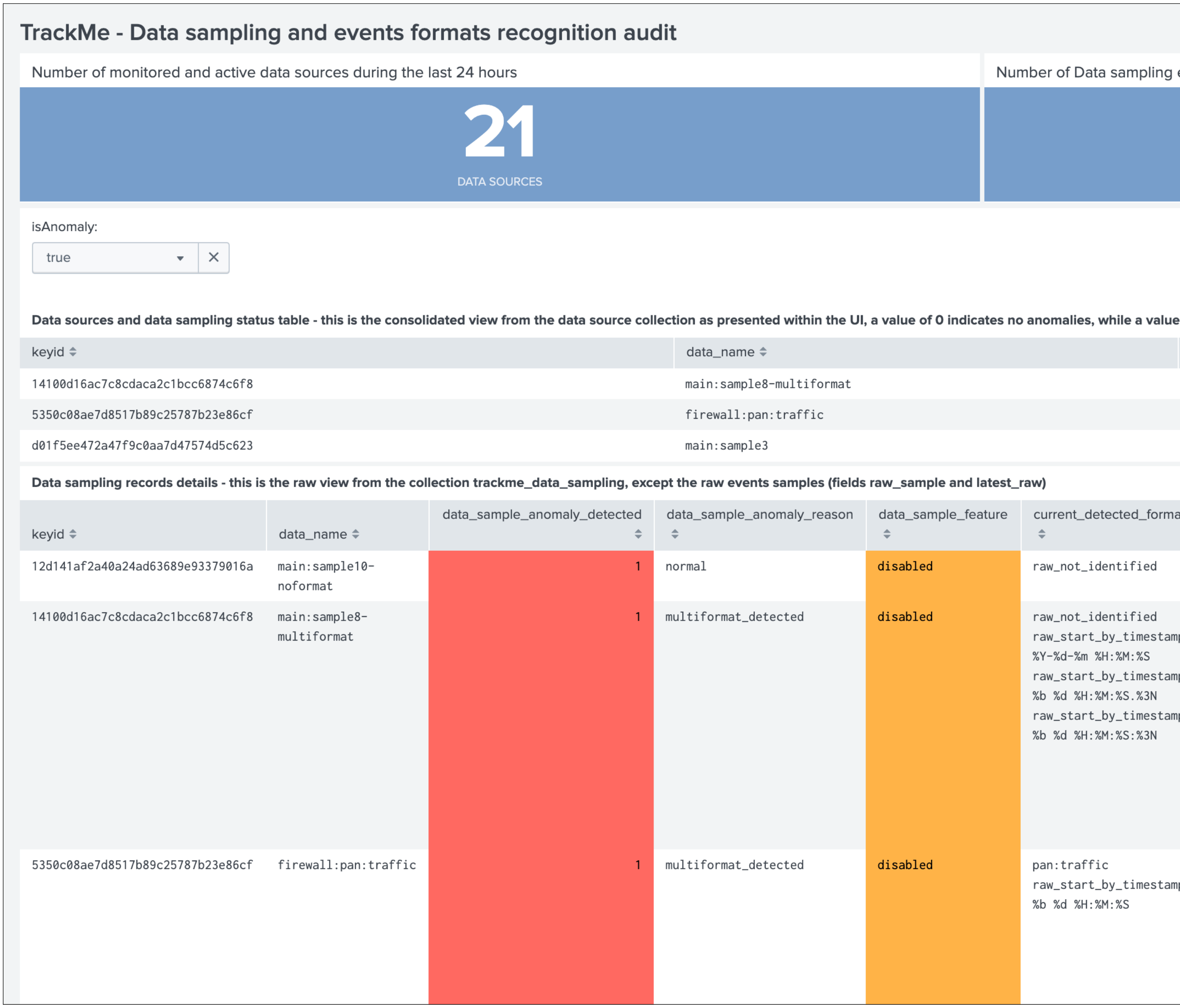
feature ⬍	current_detected_format ⬍	⬍	previous_detected_format ⬍	state ⬍	anomaly_reason ⬍
☒	N/A	<--	N/A	✗	N/A

The Data sampling feature can be enabled / disabled at any point in time, as soon as a data source is disabled, TrackMe stops considering it during the sampling operations.

Data sampling Audit dashboard

An audit dashboard is provided in the audit navigation menu, this dashboard provides insight related to the data sampling feature and workflow:

Menu Audit / TrackMe - Data sampling and events formats recognition audit



TrackMe - Data sampling and events formats recognition audit

Number of monitored and active data sources during the last 24 hours

21

DATA SOURCES

Number of Data sampling

isAnomaly: true

Data sources and data sampling status table - this is the consolidated view from the data source collection as presented within the UI, a value of 0 indicates no anomalies, while a value

keyid	data_name
14100d16ac7c8cdaca2c1bcc6874c6f8	main:sample8-multiformat
5350c08ae7d8517b89c25787b23e86cf	firewall:pan:traffic
d01f5ee472a47f9c0aa7d47574d5c623	main:sample3

Data sampling records details - this is the raw view from the collection trackme_data_sampling, except the raw events samples (fields raw_sample and latest_raw)

keyid	data_name	data_sample_anomaly_detected	data_sample_anomaly_reason	data_sample_feature	current_detected_forma
12d141af2a40a24ad63689e93379016a	main:sample10-noformat	1	normal	disabled	raw_not_identified
14100d16ac7c8cdaca2c1bcc6874c6f8	main:sample8-multiformat	1	multiformat_detected	disabled	raw_not_identified raw_start_by_timestamp %Y-%d-%m %H:%M:%S raw_start_by_timestamp %b %d %H:%M:%S.%3N raw_start_by_timestamp %b %d %H:%M:%S:%3N
5350c08ae7d8517b89c25787b23e86cf	firewall:pan:traffic	1	multiformat_detected	disabled	pan:traffic raw_start_by_timestamp %b %d %H:%M:%S

Data sampling example 1: monitor a specific format

Let's assume the following use case, we are ingesting Palo Alto firewall data and we want to monitor that our data is stricly respecting a specific expected format, any event that would not match this format would most likely be resulting from malformed events or issues in our ingestion pipeline:

Within the custom rules UI, we proceed to the creation of a new custom rule, in short our events look like:

```
Dec 26 12:15:01 1,2012/26/20 12:15:01,01606001116,TRAFFIC,start,1,2012/26/20 12:15:01,
↪192.168.0.2,204.232.231.46,0.0.0.0,0.0.0.0,
Dec 26 12:15:02 1,2012/26/20 12:15:02,01606001116,THREAT,url,1,2012/26/20 12:15:02,
↪192.168.0.2,204.232.231.46,0.0.0.0,0.0.0.0,
```

We could use the following regular expression to stricly match the format, the data sampling is similar to a where match SPL statement:

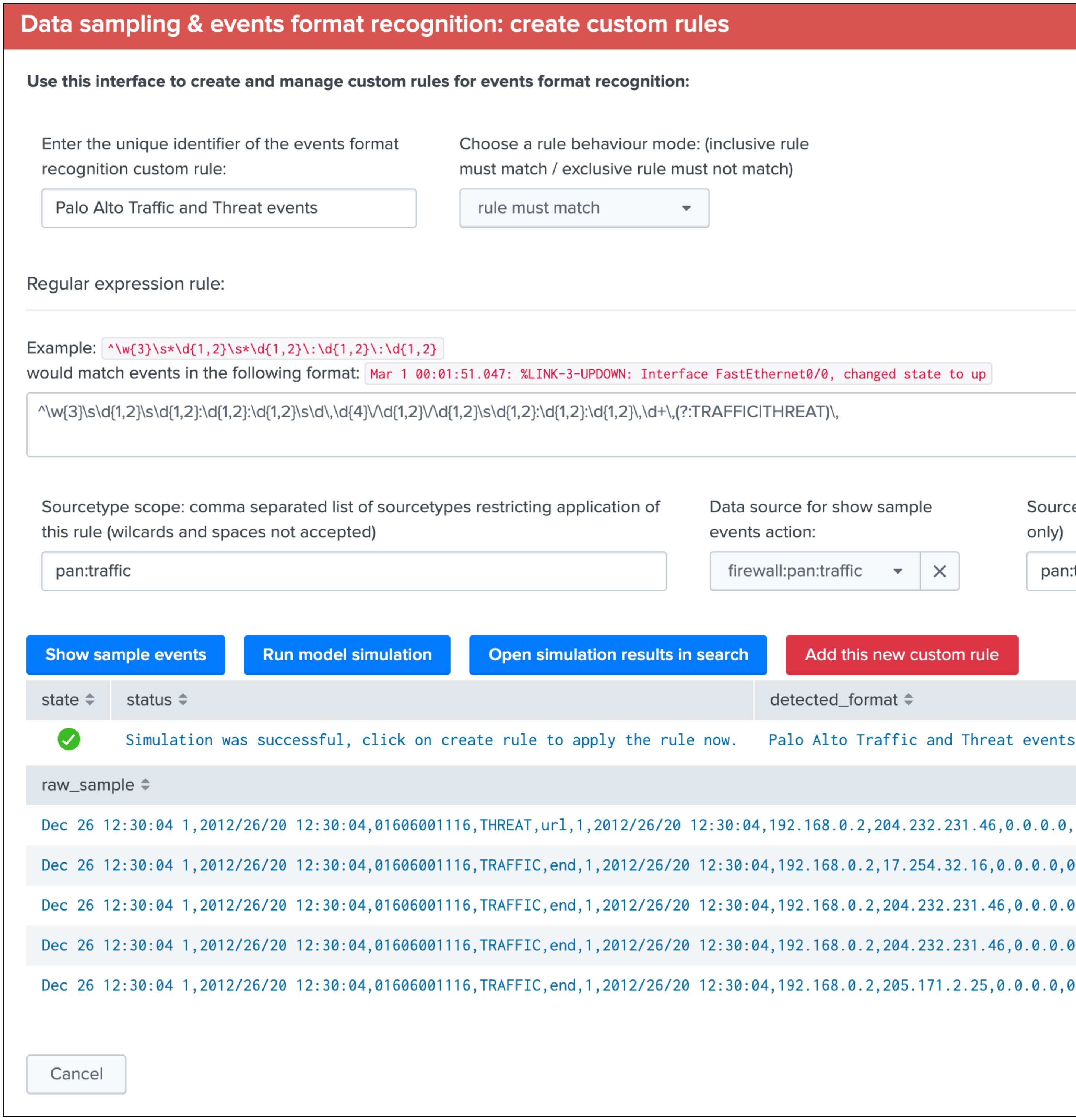
```
^\w{3}\s*\d{1,2}\s*\d{1,2}:\d{1,2}:\d{1,2}\s*\d\,\d{4}\/\d{1,2}\/\d{1,2}\s*\d{1,2}:\d
↪{1,2}:\d{1,2}\,\d+\,(?:TRAFFIC|THREAT)\,
```

Note: the regular expression doesn't have to be complex, it is up to your decide how strict it should be depending on your use case

Tip: The data sampling engine will stop at the first regular expression match, to handle advanced or more complex configuration, use the sourcetype scope to restrict the custom rule to sourcetypes that should be considered

We create a `rule must match` type of rule, which means that in normal circumstances we expect all events to be matched by our custom rule, otherwise this would be considered as an anomaly.

Once the rule has been created:



Data sampling & events format recognition: create custom rules

Use this interface to create and manage custom rules for events format recognition:

Enter the unique identifier of the events format recognition custom rule: Palo Alto Traffic and Threat events

Choose a rule behaviour mode: (inclusive rule must match / exclusive rule must not match) rule must match

Regular expression rule:

Example: `^\w{3}\s*\d{1,2}\s*\d{1,2}\:\d{1,2}\:\d{1,2}`
would match events in the following format: `Mar 1 00:01:51.047: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up`

```
^\w{3}\s\d{1,2}\s\d{1,2}:\d{1,2}:\d{1,2}\s\d\,\d{4}\/\d{1,2}\/\d{1,2}\s\d{1,2}:\d{1,2}:\d{1,2}\,\d+\,(?:TRAFFIC|THREAT)\,
```

Sourcetype scope: comma separated list of sourcetypes restricting application of this rule (wilcards and spaces not accepted) pan:traffic

Data source for show sample events action: firewall:pan:traffic

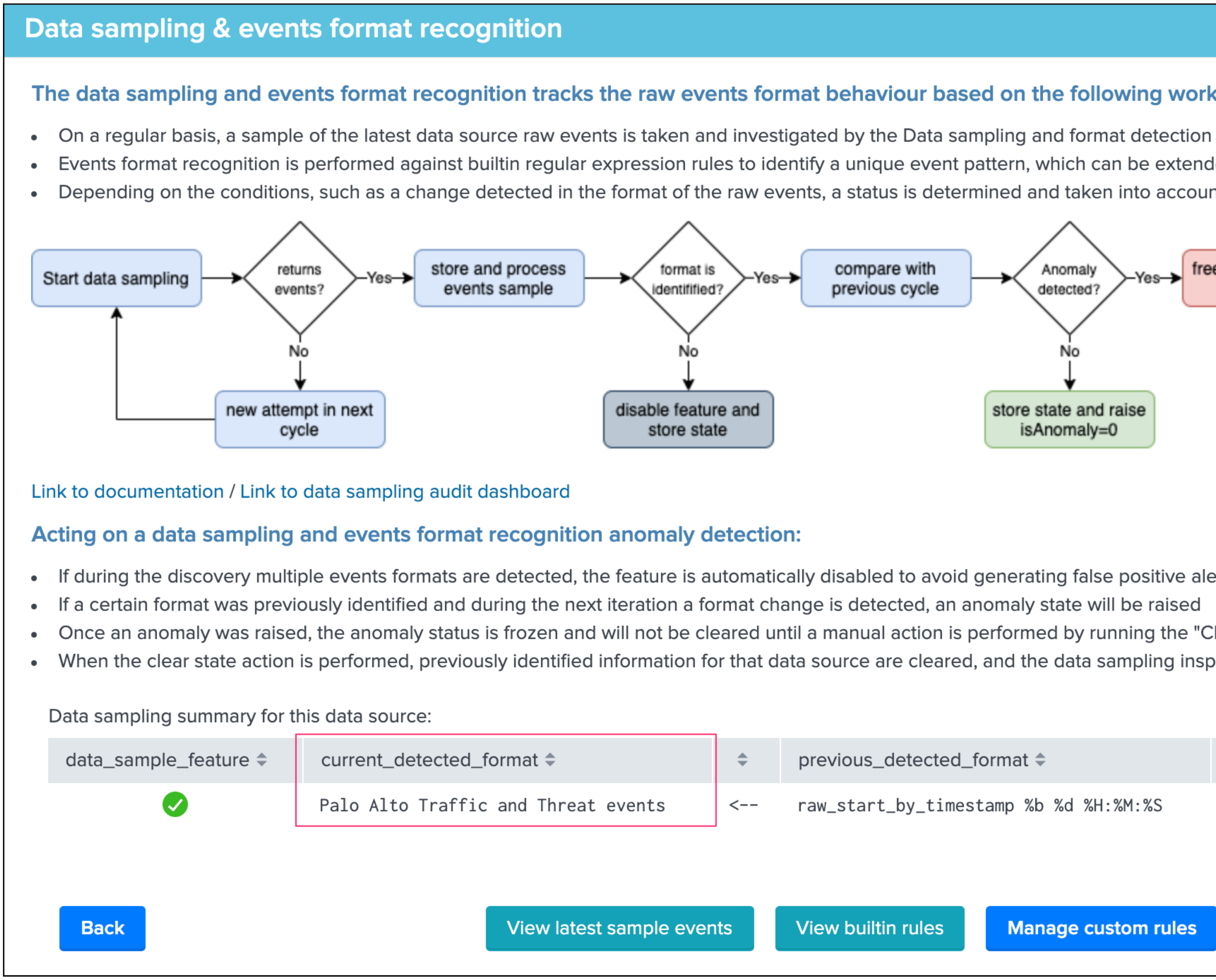
Source only) pan:t

state	status	detected_format
✓	Simulation was successful, click on create rule to apply the rule now.	Palo Alto Traffic and Threat events

raw_sample

```
Dec 26 12:30:04 1,2012/26/20 12:30:04,01606001116,THREAT,url,1,2012/26/20 12:30:04,192.168.0.2,204.232.231.46,0.0.0.0,
Dec 26 12:30:04 1,2012/26/20 12:30:04,01606001116,TRAFFIC,end,1,2012/26/20 12:30:04,192.168.0.2,17.254.32.16,0.0.0.0,0
Dec 26 12:30:04 1,2012/26/20 12:30:04,01606001116,TRAFFIC,end,1,2012/26/20 12:30:04,192.168.0.2,204.232.231.46,0.0.0.0
Dec 26 12:30:04 1,2012/26/20 12:30:04,01606001116,TRAFFIC,end,1,2012/26/20 12:30:04,192.168.0.2,204.232.231.46,0.0.0.0
Dec 26 12:30:04 1,2012/26/20 12:30:04,01606001116,TRAFFIC,end,1,2012/26/20 12:30:04,192.168.0.2,205.171.2.25,0.0.0.0,0
```

The next execution of the data sampling will report the name of the rule for each data source that is matching our conditions:



Data sampling & events format recognition

The data sampling and events format recognition tracks the raw events format behaviour based on the following work

- On a regular basis, a sample of the latest data source raw events is taken and investigated by the Data sampling and format detection
- Events format recognition is performed against builtin regular expression rules to identify a unique event pattern, which can be extend
- Depending on the conditions, such as a change detected in the format of the raw events, a status is determined and taken into accour

Start data sampling
returns events?
Yes
No
new attempt in next cycle
store and process events sample
format is identifified?
Yes
No
disable feature and store state
compare with previous cycle
Anomaly detected?
Yes
No
store state and raise isAnomaly=0
free

Link to documentation / Link to data sampling audit dashboard

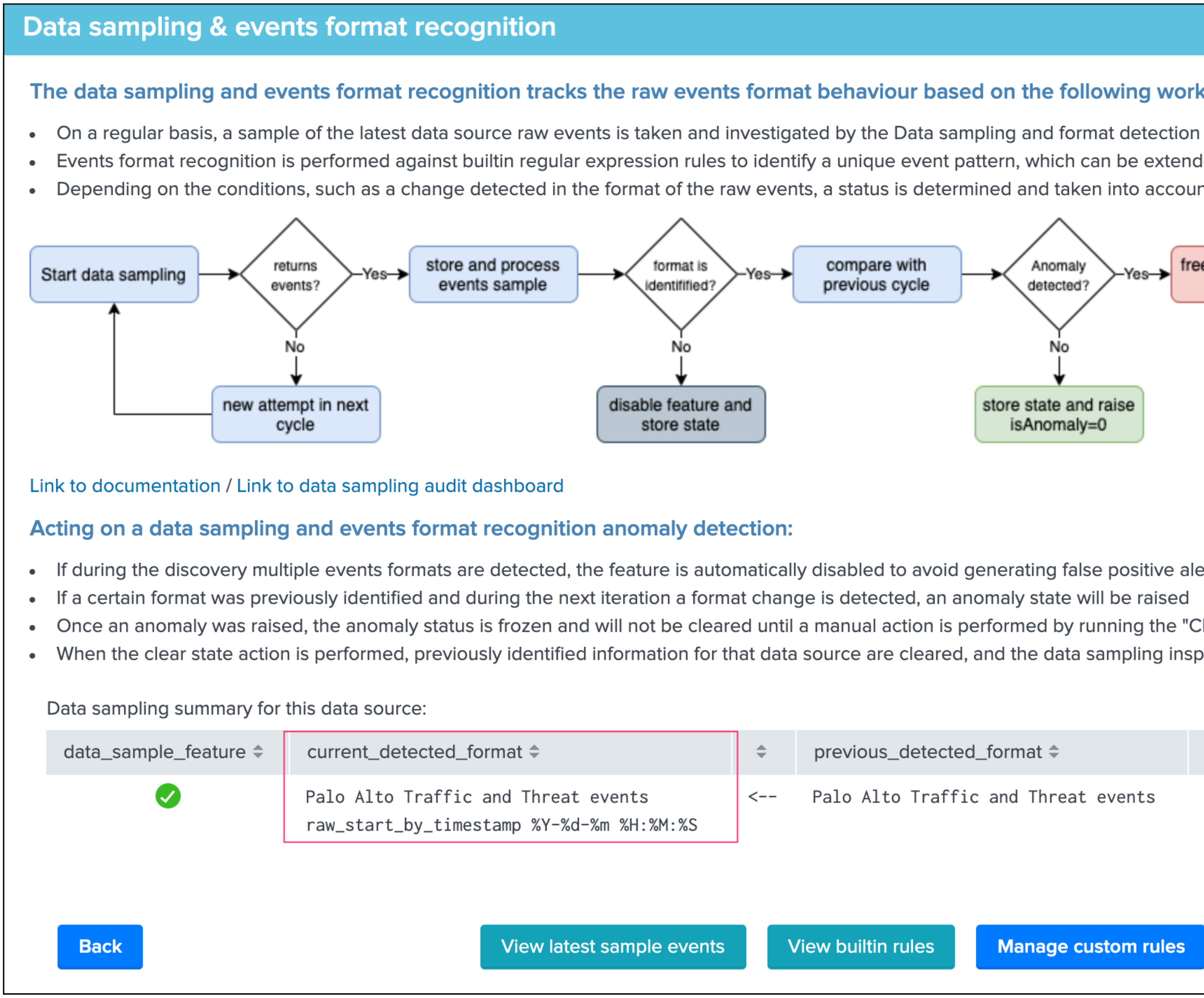
Acting on a data sampling and events format recognition anomaly detection:

- If during the discovery multiple events formats are detected, the feature is automatically disabled to avoid generating false positive ale
- If a certain format was previously identified and during the next iteration a format change is detected, an anomaly state will be raised
- Once an anomaly was raised, the anomaly status is frozen and will not be cleared until a manual action is performed by running the "Cl
- When the clear state action is performed, previously identified information for that data source are cleared, and the data sampling insp

Data sampling summary for this data source:

data_sample_feature	current_detected_format		previous_detected_format
✓	Palo Alto Traffic and Threat events	<--	raw_start_by_timestamp %b %d %H:%M:%S

Should a change in the events format happen, such as malformed events happening for any reason, the data sampling rule would match these exceptions and render a status error to be reviewed.



Data sampling & events format recognition

The data sampling and events format recognition tracks the raw events format behaviour based on the following work

- On a regular basis, a sample of the latest data source raw events is taken and investigated by the Data sampling and format detection
- Events format recognition is performed against builtin regular expression rules to identify a unique event pattern, which can be extend
- Depending on the conditions, such as a change detected in the format of the raw events, a status is determined and taken into accoun

Start data sampling
returns events?
Yes
No
new attempt in next cycle
store and process events sample
format is identifitied?
Yes
No
disable feature and store state
compare with previous cycle
Anomaly detected?
Yes
No
store state and raise isAnomaly=0
free

Link to documentation / Link to data sampling audit dashboard

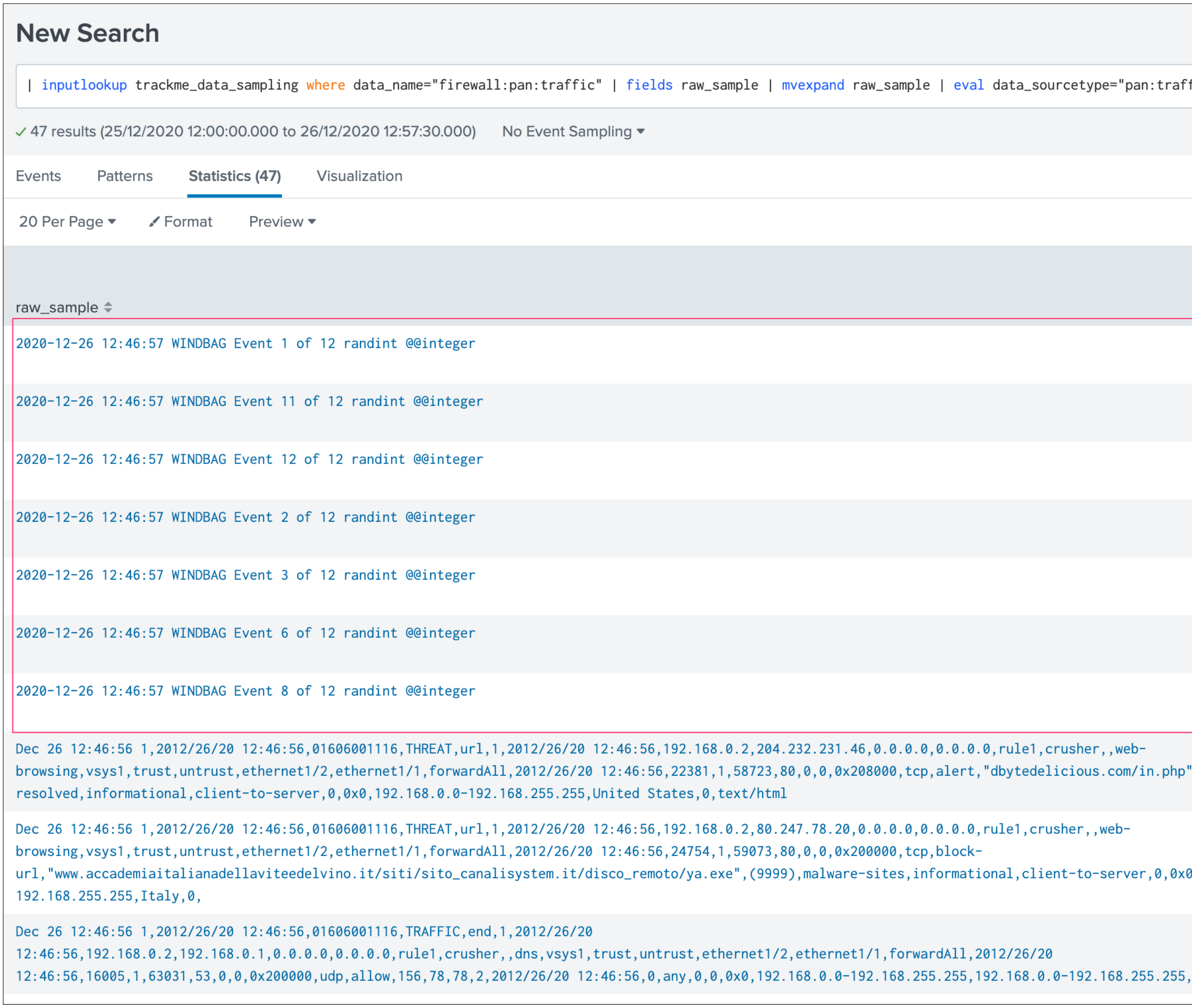
Acting on a data sampling and events format recognition anomaly detection:

- If during the discovery multiple events formats are detected, the feature is automatically disabled to avoid generating false positive ale
- If a certain format was previously identified and during the next iteration a format change is detected, an anomaly state will be raised
- Once an anomaly was raised, the anomaly status is frozen and will not be cleared until a manual action is performed by running the "Cl
- When the clear state action is performed, previously identified information for that data source are cleared, and the data sampling insp

Data sampling summary for this data source:

data_sample_feature	current_detected_format		previous_detected_format
✓	Palo Alto Traffic and Threat events raw_start_by_timestamp %Y-%d-%m %H:%M:%S	<--	Palo Alto Traffic and Threat events

Review of the latest events sample would clearly show the root cause of the issue: (button **View latest sample events**):



```
| inputlookup trackme_data_sampling where data_name="firewall:pan:traffic" | fields raw_sample | mvexpand raw_sample | eval data_sourcetype="pan:traff
```

✓ 47 results (25/12/2020 12:00:00.000 to 26/12/2020 12:57:30.000)   No Event Sampling ▾

```
2020-12-26 12:46:57 WINDBAG Event 1 of 12 randint @@integer

2020-12-26 12:46:57 WINDBAG Event 11 of 12 randint @@integer

2020-12-26 12:46:57 WINDBAG Event 12 of 12 randint @@integer

2020-12-26 12:46:57 WINDBAG Event 2 of 12 randint @@integer

2020-12-26 12:46:57 WINDBAG Event 3 of 12 randint @@integer

2020-12-26 12:46:57 WINDBAG Event 6 of 12 randint @@integer

2020-12-26 12:46:57 WINDBAG Event 8 of 12 randint @@integer

Dec 26 12:46:56 1,2012/26/20 12:46:56,01606001116,THREAT,url,1,2012/26/20 12:46:56,192.168.0.2,204.232.231.46,0.0.0.0,0.0.0.0,rule1,crusher,,web-
browsing,vsys1,trust,untrust,ethernet1/2,ethernet1/1,forwardAll,2012/26/20 12:46:56,22381,1,58723,80,0,0,0x208000,tcp,alert,"dbytedelicious.com/in.php"
resolved,informational,client-to-server,0,0x0,192.168.0.0-192.168.255.255,United States,0,text/html

Dec 26 12:46:56 1,2012/26/20 12:46:56,01606001116,THREAT,url,1,2012/26/20 12:46:56,192.168.0.2,80.247.78.20,0.0.0.0,0.0.0.0,rule1,crusher,,web-
browsing,vsys1,trust,untrust,ethernet1/2,ethernet1/1,forwardAll,2012/26/20 12:46:56,24754,1,59073,80,0,0,0x200000,tcp,block-
url,"www.accademiaitalianadellaviteedelvino.it/siti/sito_canalisystem.it/disco_remoto/ya.exe",(9999),malware-sites,informational,client-to-server,0,0x0
192.168.255.255,Italy,0,

Dec 26 12:46:56 1,2012/26/20 12:46:56,01606001116,TRAFFIC,end,1,2012/26/20
12:46:56,192.168.0.2,192.168.0.1,0.0.0.0,0.0.0.0,rule1,crusher,,dns,vsys1,trust,untrust,ethernet1/2,ethernet1/1,forwardAll,2012/26/20
12:46:56,16005,1,63031,53,0,0,0x200000,udp,allow,156,78,78,2,2012/26/20 12:46:56,0,any,0,0,0x0,192.168.0.0-192.168.255.255,192.168.0.0-192.168.255.255,
```

As the data sampling engine stops proceeding a data source as soon as an issue was detected, these events are the exact events that have caused the anomaly exception at the exact time it happened.

Once investigations have been performed, the root cause was identified and ideally fixed, a TrackMe admin would clear the data sampling state to free the current state and allow the workflow to proceed again in further executions.

Data sampling example 2: track PII data card holders

Let’s consider the following use case, we ingest retail transaction logs which are not supposed to contain PII data (Personally Identifiable Information) because the events are anonymised during the indexing phase. (this obviously is a simplitic example for the demonstration purposes)

In our example, we will consider credit card references which are replaced by the according number of “X” characters:

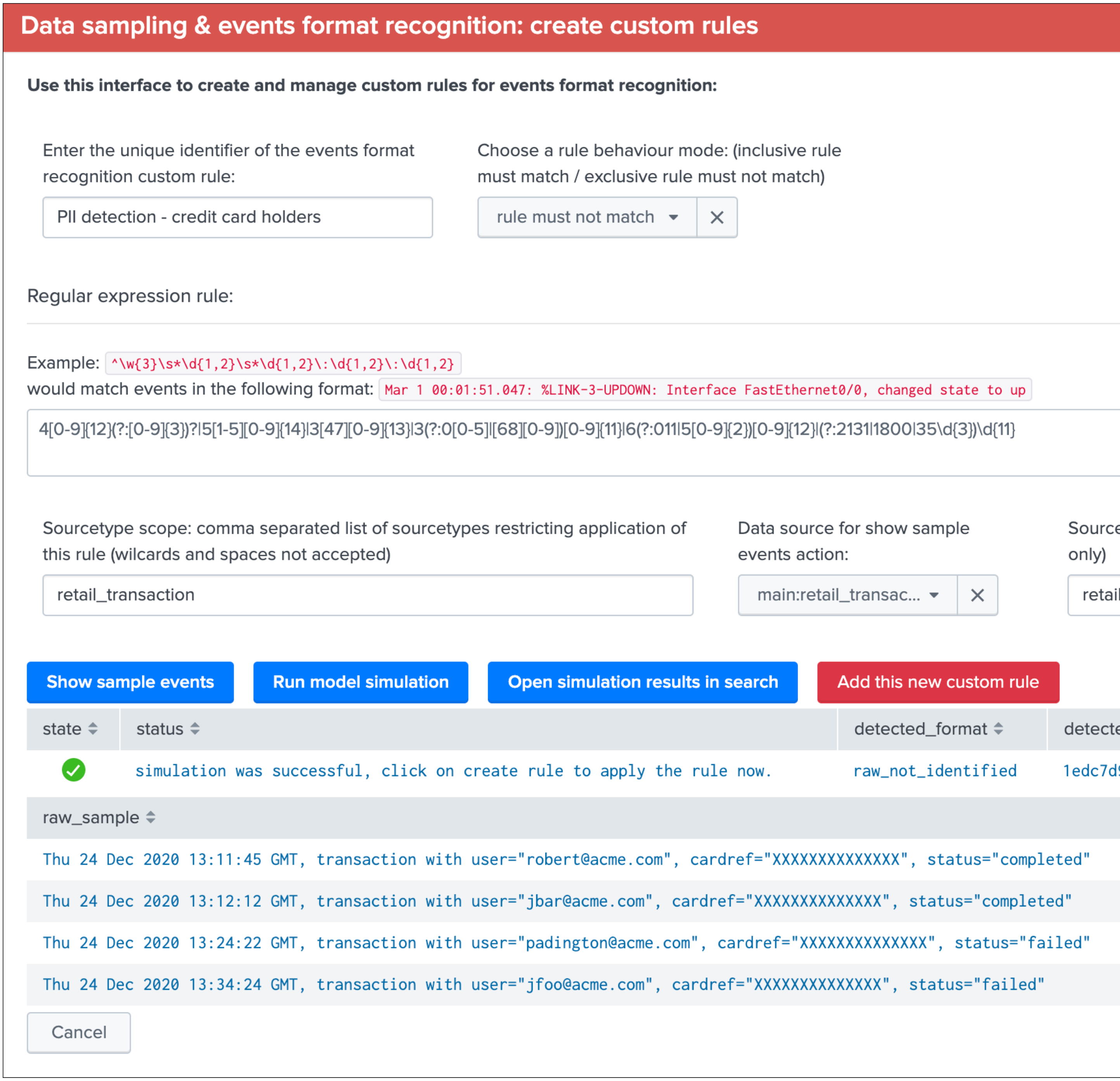
```
Thu 24 Dec 2020 13:12:12 GMT, transaction with user="jbar@acme.com", cardref=
↪"XXXXXXXXXXXXXX", status="completed"
Thu 24 Dec 2020 13:34:24 GMT, transaction with user="jfoo@acme.com", cardref=
↪"XXXXXXXXXXXXXX", status="failed"
Thu 24 Dec 2020 13:11:45 GMT, transaction with user="robert@acme.com", cardref=
↪"XXXXXXXXXXXXXX", status="completed"
Thu 24 Dec 2020 13:24:22 GMT, transaction with user="padington@acme.com", cardref=
↪"XXXXXXXXXXXXXX", status="failed"
```

To track for an anomaly in the process that normally anonymises the data, we could rely on a regular expression that targets valid credit card numbers:

See: https://www.regextester.com/93608

```
4[0-9]{12}(?:[0-9]{3})?|5[1-5][0-9]{14}|3[47][0-9]{13}|3(?:0[0-5]|[68][0-9])[0-9]{11}
↪|6(?:011|5[0-9]{2})[0-9]{12}|(?:2131|1800|35\d{3})\d{11}
```

Should any event be matching this regular expression, we would most likely face a situation where we have indexed a clear text information that is very problematic, let's create a new custom rule of a `rule must not match` type to track this use case automatically, to avoid false positive detection we will restrict this custom rule to a given list of sourcetypes:



Data sampling & events format recognition: create custom rules

Use this interface to create and manage custom rules for events format recognition:

Enter the unique identifier of the events format recognition custom rule: PII detection - credit card holders

Choose a rule behaviour mode: (inclusive rule must match / exclusive rule must not match): rule must not match

Regular expression rule:

Example: `^\w{3}\s*\d{1,2}\s*\d{1,2}\:\d{1,2}\:\d{1,2}`
would match events in the following format: `Mar 1 00:01:51.047: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up`

```
4[0-9]{12}(?:[0-9]{3})?|5[1-5][0-9]{14}|3[47][0-9]{13}|3(?:0[0-5]|[68][0-9])[0-9]{11}|6(?:011|5[0-9]{2})[0-9]{12}|(?:2131|1800|35\d{3})\d{11}
```

Sourcetype scope: comma separated list of sourcetypes restricting application of this rule (wilcards and spaces not accepted): retail_transaction

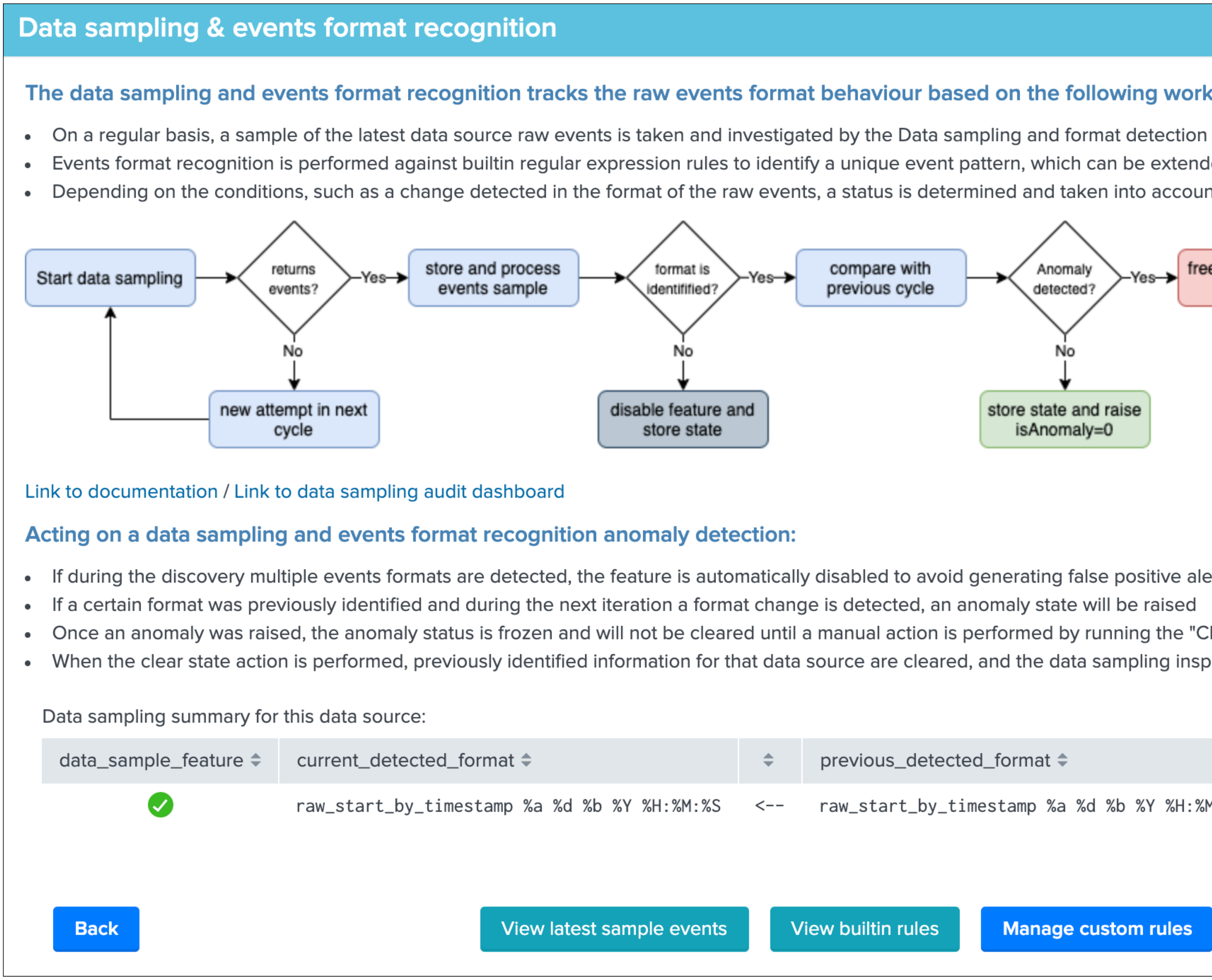
Data source for show sample events action: main:retail_transac...

Source only): retail

state	status	detected_format	detecte
✓	simulation was successful, click on create rule to apply the rule now.	raw_not_identified	1edc7d9

raw_sample
Thu 24 Dec 2020 13:11:45 GMT, transaction with user="robert@acme.com", cardref="XXXXXXXXXXXXXX", status="completed"
Thu 24 Dec 2020 13:12:12 GMT, transaction with user="jbar@acme.com", cardref="XXXXXXXXXXXXXX", status="completed"
Thu 24 Dec 2020 13:24:22 GMT, transaction with user="padington@acme.com", cardref="XXXXXXXXXXXXXX", status="failed"
Thu 24 Dec 2020 13:34:24 GMT, transaction with user="jfoo@acme.com", cardref="XXXXXXXXXXXXXX", status="failed"

Our data uses a format that is recognized automatically by builtin rules, and would appears as following in normal circumstances:



Data sampling & events format recognition

The data sampling and events format recognition tracks the raw events format behaviour based on the following work

- On a regular basis, a sample of the latest data source raw events is taken and investigated by the Data sampling and format detection
- Events format recognition is performed against builtin regular expression rules to identify a unique event pattern, which can be extende
- Depending on the conditions, such as a change detected in the format of the raw events, a status is determined and taken into accoun

Start data sampling
returns events?
Yes
No
new attempt in next cycle
store and process events sample
format is identified?
Yes
No
disable feature and store state
compare with previous cycle
Anomaly detected?
Yes
No
store state and raise isAnomaly=0
free

Link to documentation / Link to data sampling audit dashboard

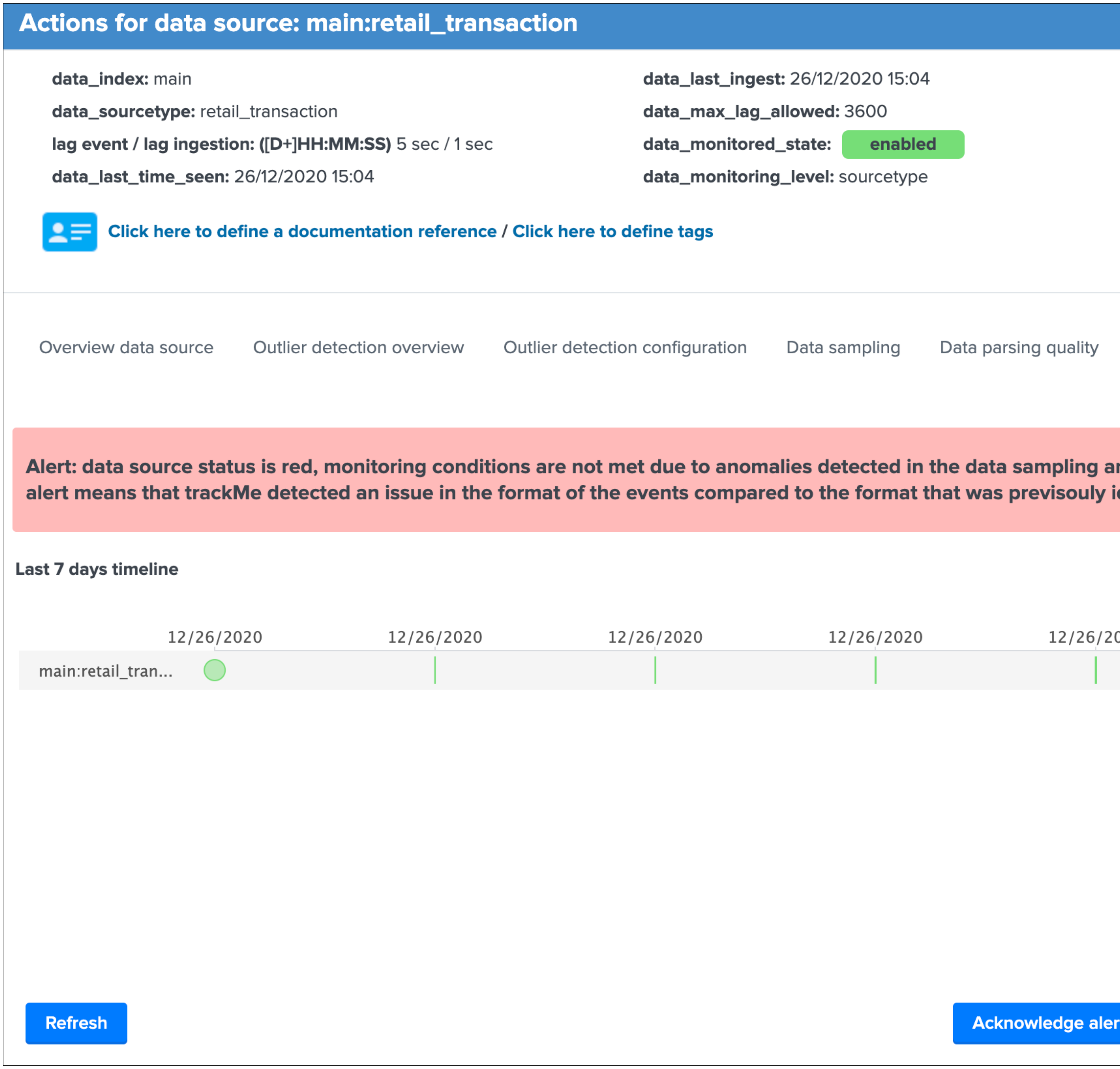
Acting on a data sampling and events format recognition anomaly detection:

- If during the discovery multiple events formats are detected, the feature is automatically disabled to avoid generating false positive ale
- If a certain format was previously identified and during the next iteration a format change is detected, an anomaly state will be raised
- Once an anomaly was raised, the anomaly status is frozen and will not be cleared until a manual action is performed by running the "Cl
- When the clear state action is performed, previously identified information for that data source are cleared, and the data sampling insp

Data sampling summary for this data source:

data_sample_feature ⬍	current_detected_format ⬍	⬍	previous_detected_format ⬍
✓	raw_start_by_timestamp %a %d %b %Y %H:%M:%S	<--	raw_start_by_timestamp %a %d %b %Y %H:%M

After some time, we introduce events containing real clear text credit card numbers, eventually our custom rule will automatically detect it and state an alert on the data source:



Actions for data source: main:retail_transaction

data_index: main

data_sourcetype: retail_transaction

lag event / lag ingestion: ([D+]HH:MM:SS) 5 sec / 1 sec

data_last_time_seen: 26/12/2020 15:04

data_last_ingest: 26/12/2020 15:04

data_max_lag_allowed: 3600

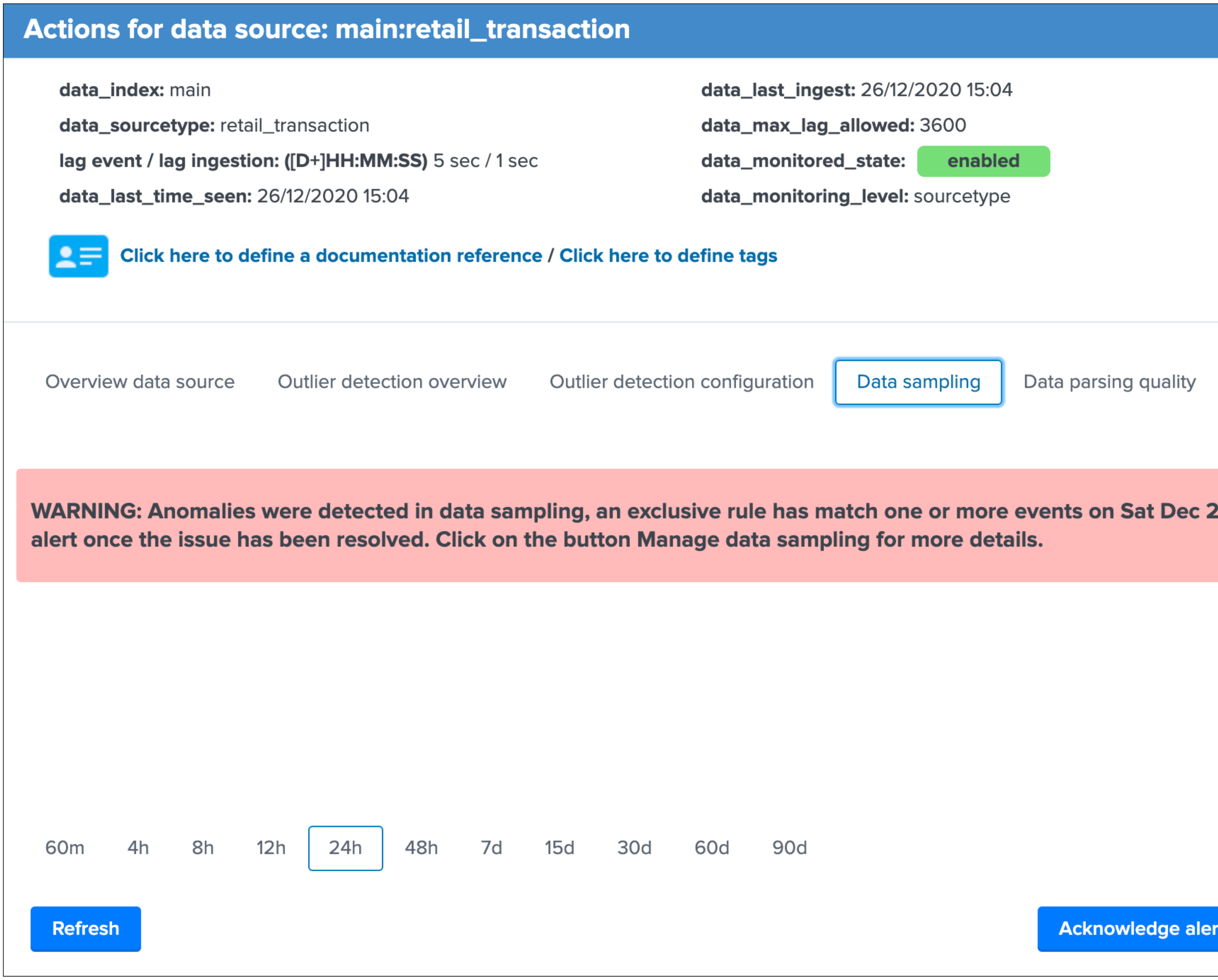
data_monitored_state: enabled

data_monitoring_level: sourcetype

Click here to define a documentation reference / Click here to define tags

Alert: data source status is red, monitoring conditions are not met due to anomalies detected in the data sampling ar alert means that trackMe detected an issue in the format of the events compared to the format that was previsouly ic

Last 7 days timeline



Actions for data source: main:retail_transaction

data_index: main

data_sourcetype: retail_transaction

lag event / lag ingestion: ([D+]HH:MM:SS) 5 sec / 1 sec

data_last_time_seen: 26/12/2020 15:04

data_last_ingest: 26/12/2020 15:04

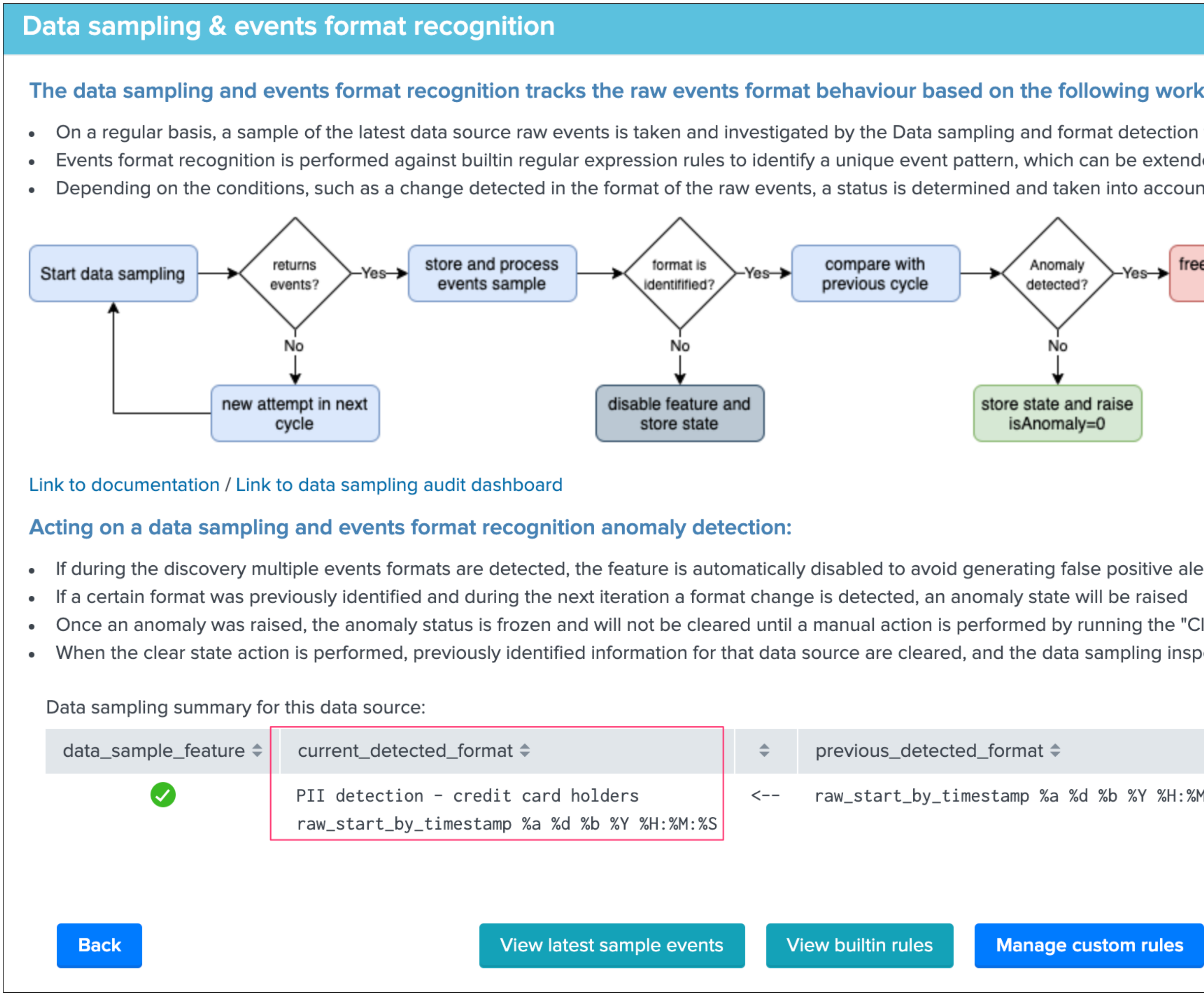
data_max_lag_allowed: 3600

data_monitored_state: enabled

data_monitoring_level: sourcetype

Click here to define a documentation reference / Click here to define tags

WARNING: Anomalies were detected in data sampling, an exclusive rule has match one or more events on Sat Dec 2
alert once the issue has been resolved. Click on the button Manage data sampling for more details.



Data sampling & events format recognition

The data sampling and events format recognition tracks the raw events format behaviour based on the following work

- On a regular basis, a sample of the latest data source raw events is taken and investigated by the Data sampling and format detection
- Events format recognition is performed against builtin regular expression rules to identify a unique event pattern, which can be extende
- Depending on the conditions, such as a change detected in the format of the raw events, a status is determined and taken into accoun

Start data sampling
returns events?
Yes
No
new attempt in next cycle
store and process events sample
format is identifified?
Yes
No
disable feature and store state
compare with previous cycle
Anomaly detected?
Yes
No
store state and raise isAnomaly=0
free

Link to documentation / Link to data sampling audit dashboard

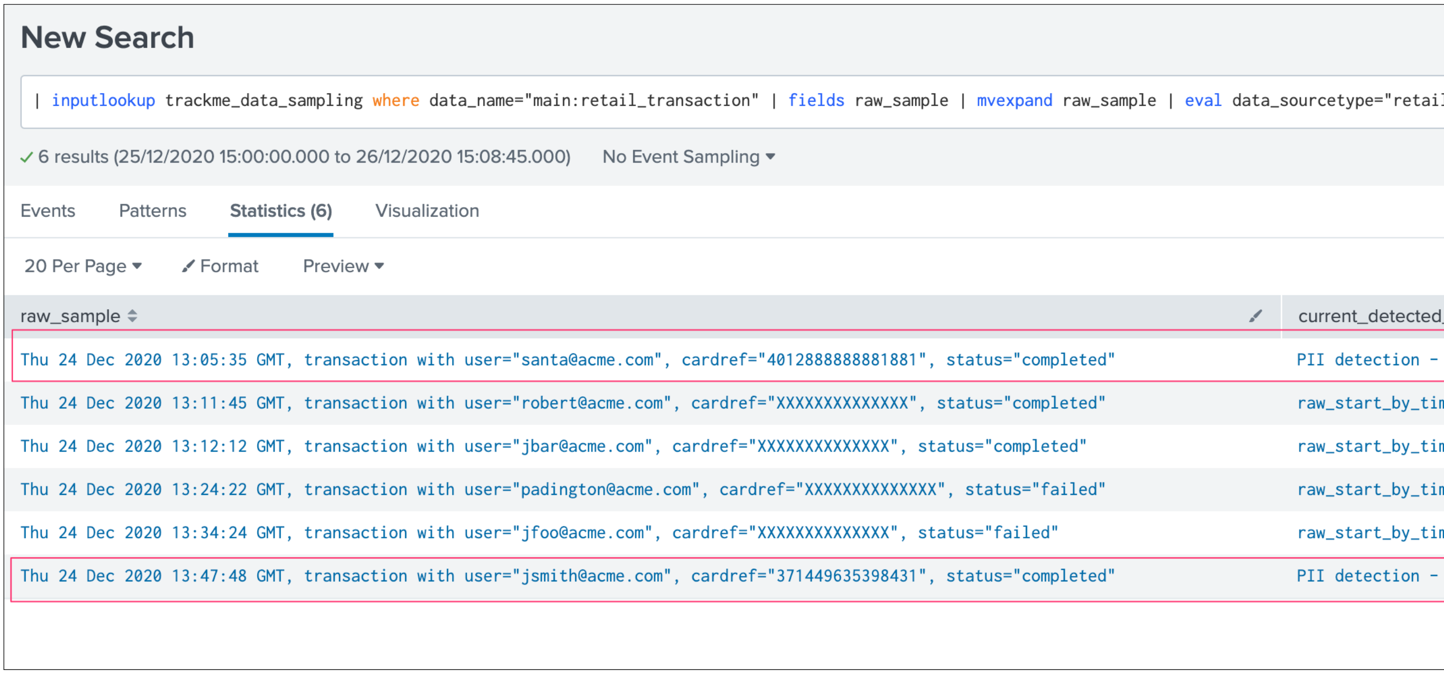
Acting on a data sampling and events format recognition anomaly detection:

- If during the discovery multiple events formats are detected, the feature is automatically disabled to avoid generating false positive ale
- If a certain format was previously identified and during the next iteration a format change is detected, an anomaly state will be raised
- Once an anomaly was raised, the anomaly status is frozen and will not be cleared until a manual action is performed by running the "Cl
- When the clear state action is performed, previously identified information for that data source are cleared, and the data sampling insp

Data sampling summary for this data source:

data_sample_feature ⬍	current_detected_format ⬍	⬍	previous_detected_format ⬍
✓	PII detection - credit card holders raw_start_by_timestamp %a %d %b %Y %H:%M:%S	<--	raw_start_by_timestamp %a %d %b %Y %H:%M

We can clearly understand the root cause of the issue reported by TrackMe, shall we investigate further (button **View latest sample events**):



```
| inputlookup trackme_data_sampling where data_name="main:retail_transaction" | fields raw_sample | mvexpand raw_sample | eval data_sourcetype="retail
```

✓ 6 results (25/12/2020 15:00:00.000 to 26/12/2020 15:08:45.000)

raw_sample	current_detected_
Thu 24 Dec 2020 13:05:35 GMT, transaction with user="santa@acme.com", cardref="4012888888881881", status="completed"	PII detection -
Thu 24 Dec 2020 13:11:45 GMT, transaction with user="robert@acme.com", cardref="XXXXXXXXXXXXXX", status="completed"	raw_start_by_tin
Thu 24 Dec 2020 13:12:12 GMT, transaction with user="jbar@acme.com", cardref="XXXXXXXXXXXXXX", status="completed"	raw_start_by_tin
Thu 24 Dec 2020 13:24:22 GMT, transaction with user="padington@acme.com", cardref="XXXXXXXXXXXXXX", status="failed"	raw_start_by_tin
Thu 24 Dec 2020 13:34:24 GMT, transaction with user="jfoo@acme.com", cardref="XXXXXXXXXXXXXX", status="failed"	raw_start_by_tin
Thu 24 Dec 2020 13:47:48 GMT, transaction with user="jsmith@acme.com", cardref="371449635398431", status="completed"	PII detection -

Thanks to the data sampling feature, we are able to get an automated tracking that is working at any scale, keep in mind that TrackMe will proceed by picking up samples, which means a very rare condition will potentially not be detected.

However, there is statistically a very high level of chance that if this is happening on a regular basis, this will be detected without having to generate very expensive searches that would look at the entire subset of data. (which would be very expensive and potentially not doable at scale)

3.1.6 Smart Status

Smart Status Introduction

The Smart Status is a powerful feature that runs automated investigations and correlations.

Under the cover, the Smart Status is a Python based backend exposed via a REST API endpoint, it is available in the TrackMe UI via the *REST API trackme SPL command* and any third party integration via the *Smart Status endpoints*.

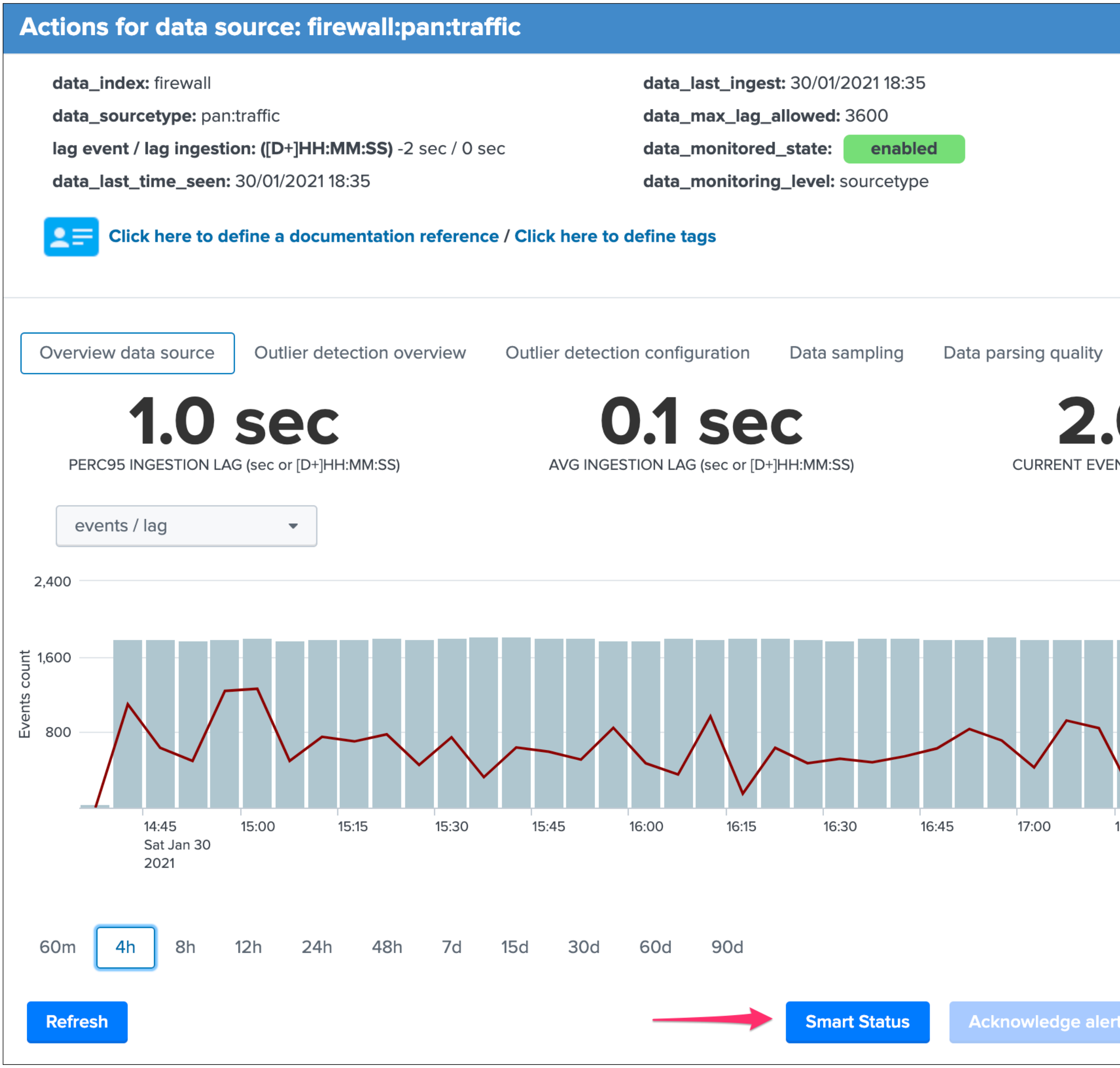
The feature uses the Python SDK for Splunk and Python capabilities to perform various conditional operations depending on the status of the entity, for instance in short for a data source it does:

- retrieve the current state of the entity
- perform a correlation over the flipping events to determine if the rate of flipping events is abnormal
- if the status is not green, determine the reason for the status and conditionally perform correlations and provide a report highlting the findings
- finally generate a JSON response with a status code depending on the investigations to ease and fast the understanding of the failure root cause

In short, the purpose of the feature is to quickly and automatically investigate the entity status, and provide a short path for investigations.

Smart Status within the UI

In the UI, access the Smart Status the open-up screen for a given entity, for data sources, hosts and metric hosts:



Actions for data source: firewall:pan:traffic

data_index: firewall

data_sourcetype: pan:traffic

lag event / lag ingestion: ([D+]HH:MM:SS) -2 sec / 0 sec

data_last_time_seen: 30/01/2021 18:35

data_last_ingest: 30/01/2021 18:35

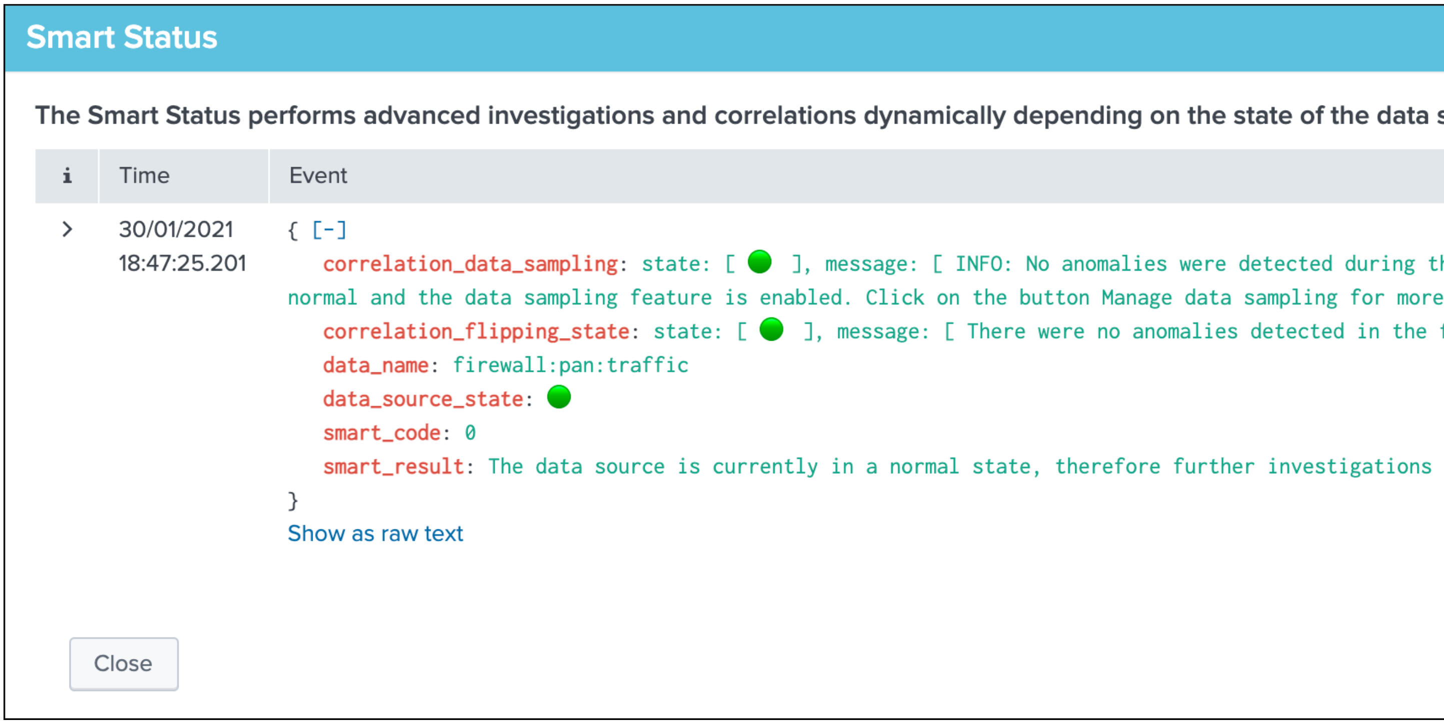
data_max_lag_allowed: 3600

data_monitored_state: enabled

data_monitoring_level: sourcetype

Click here to define a documentation reference / Click here to define tags

Smart Status example: (normal state entity)



Smart Status

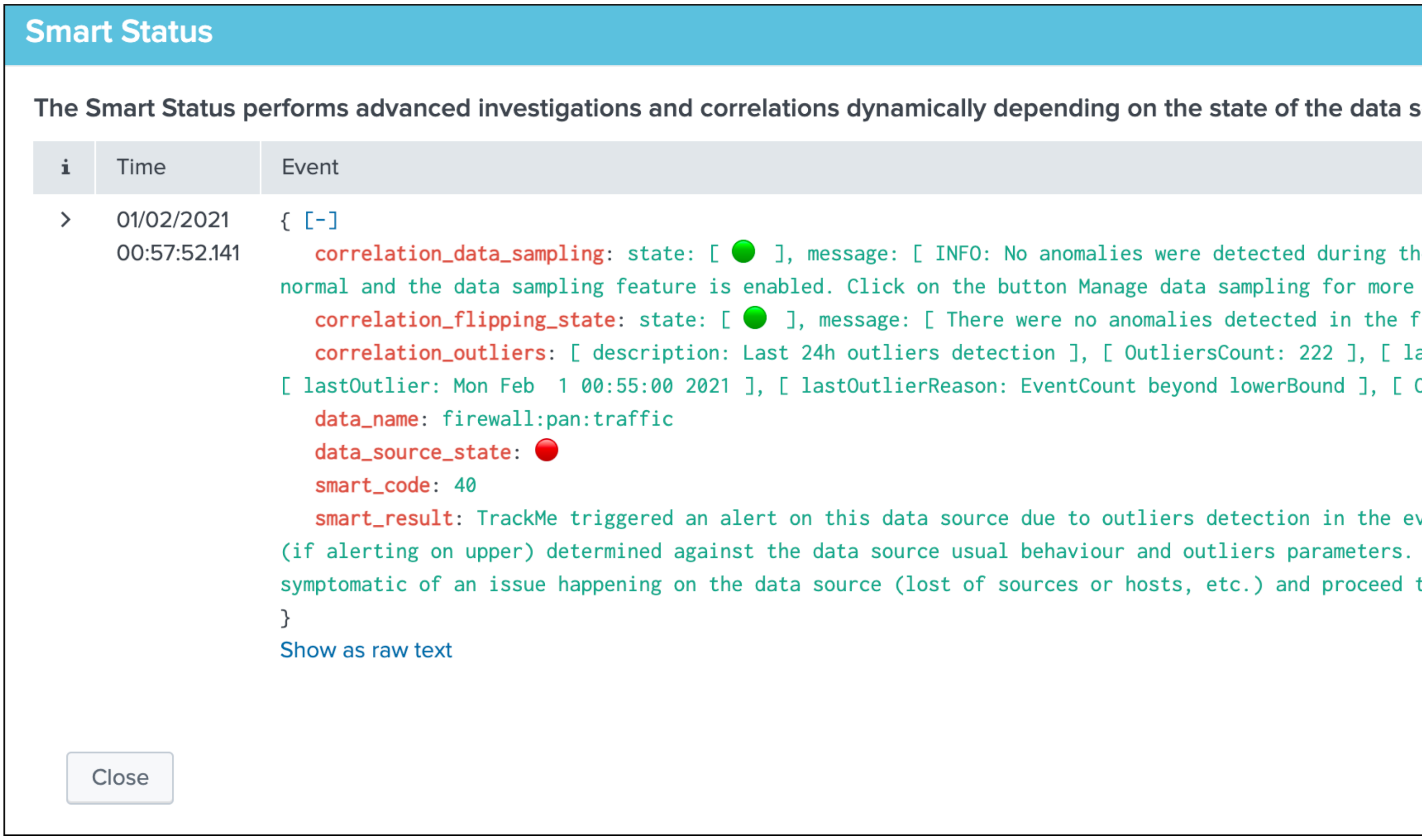
The Smart Status performs advanced investigations and correlations dynamically depending on the state of the data s

i	Time	Event
>	30/01/2021 18:47:25.201	see below

```
{ [-]
   correlation_data_sampling: state: [ 🟢 ], message: [ INFO: No anomalies were detected during th
normal and the data sampling feature is enabled. Click on the button Manage data sampling for more
   correlation_flipping_state: state: [ 🟢 ], message: [ There were no anomalies detected in the f
   data_name: firewall:pan:traffic
   data_source_state: 🟢
   smart_code: 0
   smart_result: The data source is currently in a normal state, therefore further investigations
}
```

Show as raw text

Smart Status example: (alert state entity due to outliers)



Smart Status

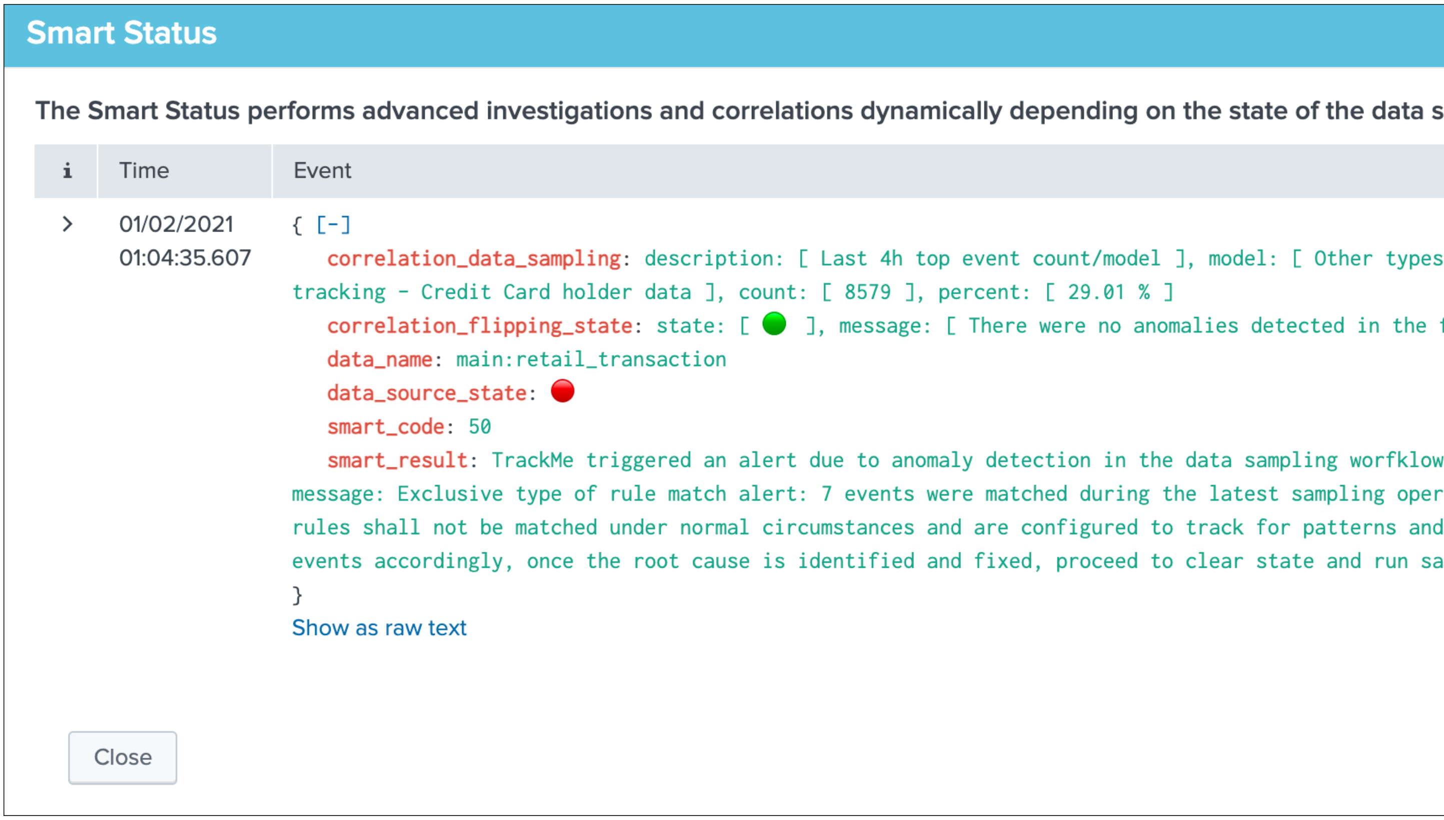
The Smart Status performs advanced investigations and correlations dynamically depending on the state of the data s

i	Time	Event
>	01/02/2021 00:57:52.141	see below

```
{ [-]
   correlation_data_sampling: state: [ 🟢 ], message: [ INFO: No anomalies were detected during th
normal and the data sampling feature is enabled. Click on the button Manage data sampling for more
   correlation_flipping_state: state: [ 🟢 ], message: [ There were no anomalies detected in the f
   correlation_outliers: [ description: Last 24h outliers detection ], [ OutliersCount: 222 ], [ la
[ lastOutlier: Mon Feb  1 00:55:00 2021 ], [ lastOutlierReason: EventCount beyond lowerBound ], [ O
   data_name: firewall:pan:traffic
   data_source_state: 🔴
   smart_code: 40
   smart_result: TrackMe triggered an alert on this data source due to outliers detection in the ev
(if alerting on upper) determined against the data source usual behaviour and outliers parameters.
symptomatic of an issue happening on the data source (lost of sources or hosts, etc.) and proceed t
}
```

Show as raw text

Smart Status example: (alert state entity due to data sampling exclusive rule matching PII data)

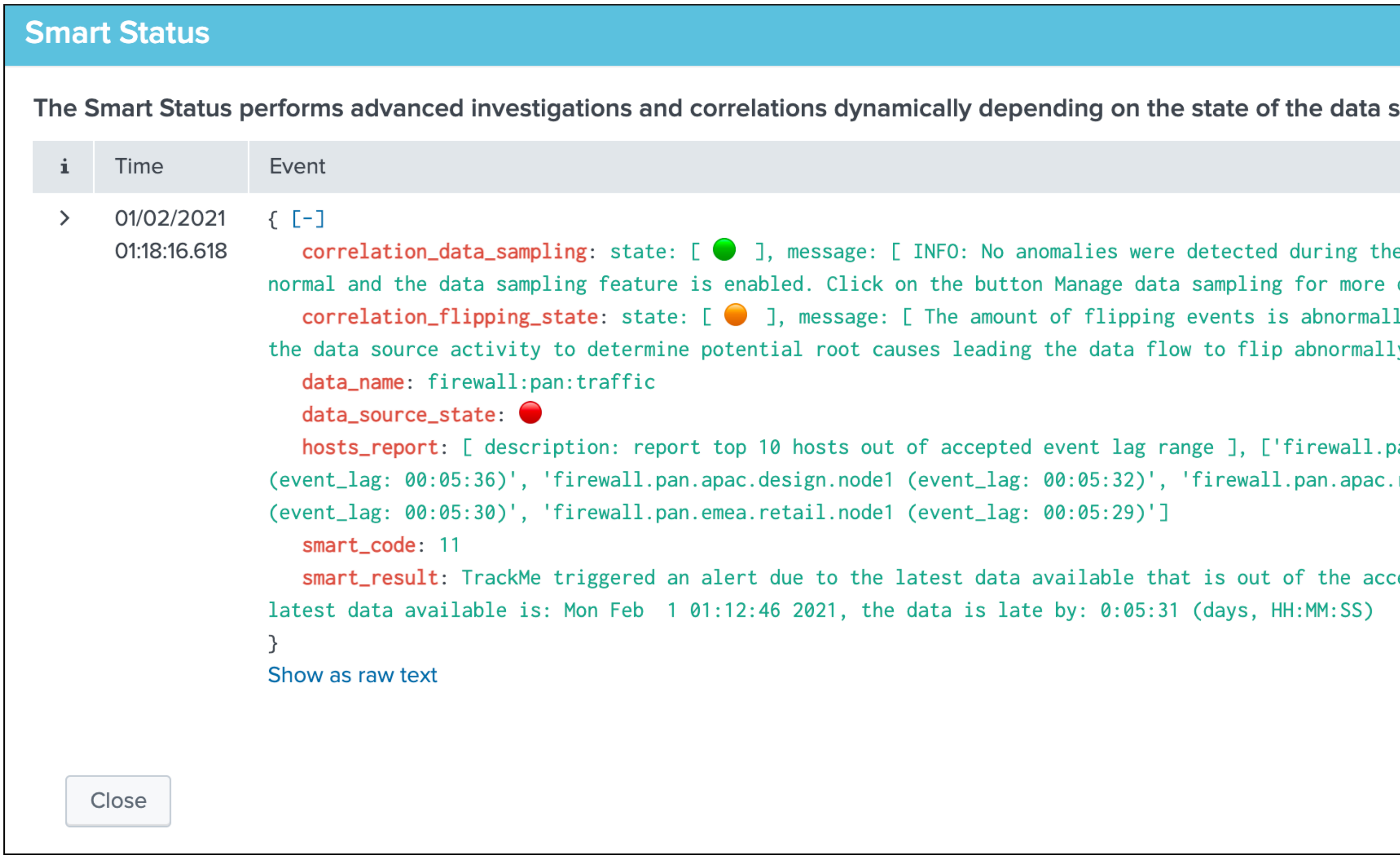


Smart Status

The Smart Status performs advanced investigations and correlations dynamically depending on the state of the data s

i	Time	Event
>	01/02/2021 01:04:35.607	{ [-] correlation_data_sampling: description: [Last 4h top event count/model], model: [Other types tracking - Credit Card holder data], count: [8579], percent: [29.01 %] correlation_flipping_state: state: [🟢], message: [There were no anomalies detected in the f data_name: main:retail_transaction data_source_state: 🔴 smart_code: 50 smart_result: TrackMe triggered an alert due to anomaly detection in the data sampling worfklow message: Exclusive type of rule match alert: 7 events were matched during the latest sampling oper rules shall not be matched under normal circumstances and are configured to track for patterns and events accordingly, once the root cause is identified and fixed, proceed to clear state and run sa } Show as raw text

Smart Status example: (alert state entity due to lagging)



Smart Status

The Smart Status performs advanced investigations and correlations dynamically depending on the state of the data s

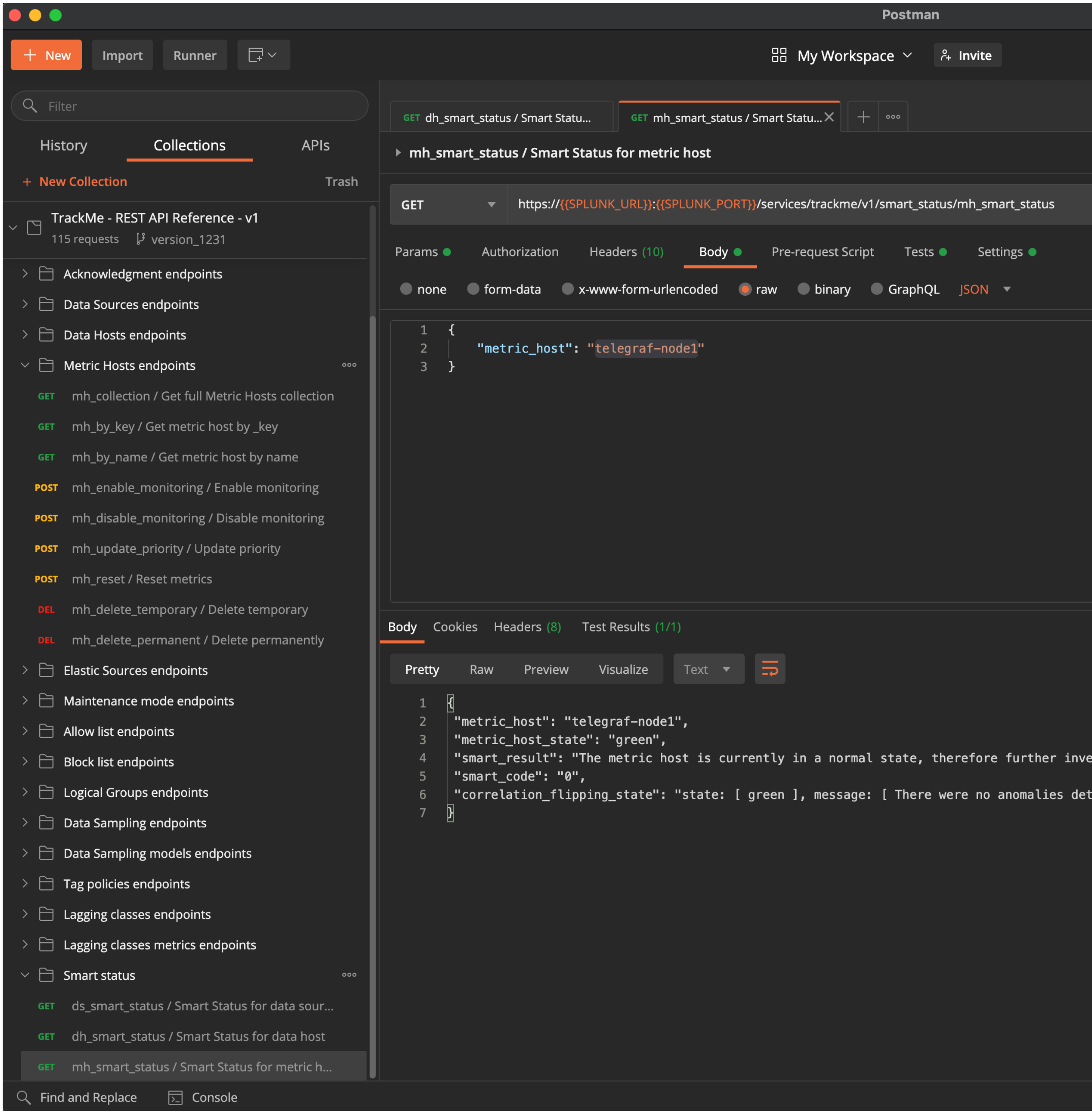
i	Time	Event
>	01/02/2021 01:18:16.618	see below

```
{ [-]
   correlation_data_sampling: state: [ 🟢 ], message: [ INFO: No anomalies were detected during the
normal and the data sampling feature is enabled. Click on the button Manage data sampling for more d
   correlation_flipping_state: state: [ 🟠 ], message: [ The amount of flipping events is abnormall
the data source activity to determine potential root causes leading the data flow to flip abnormally
   data_name: firewall:pan:traffic
   data_source_state: 🔴
   hosts_report: [ description: report top 10 hosts out of accepted event lag range ], ['firewall.pa
(event_lag: 00:05:36)', 'firewall.pan.apac.design.node1 (event_lag: 00:05:32)', 'firewall.pan.apac.r
(event_lag: 00:05:30)', 'firewall.pan.emea.retail.node1 (event_lag: 00:05:29)']
   smart_code: 11
   smart_result: TrackMe triggered an alert due to the latest data available that is out of the acce
latest data available is: Mon Feb  1 01:12:46 2021, the data is late by: 0:05:31 (days, HH:MM:SS)
}
```

Smart Status from external third party

The Smart Status feature is serviced by a REST API endpoint, as such it can be requested via any external system, such as Splunk Phantom or any other automation plateforns:

Smart Status example via Postman:

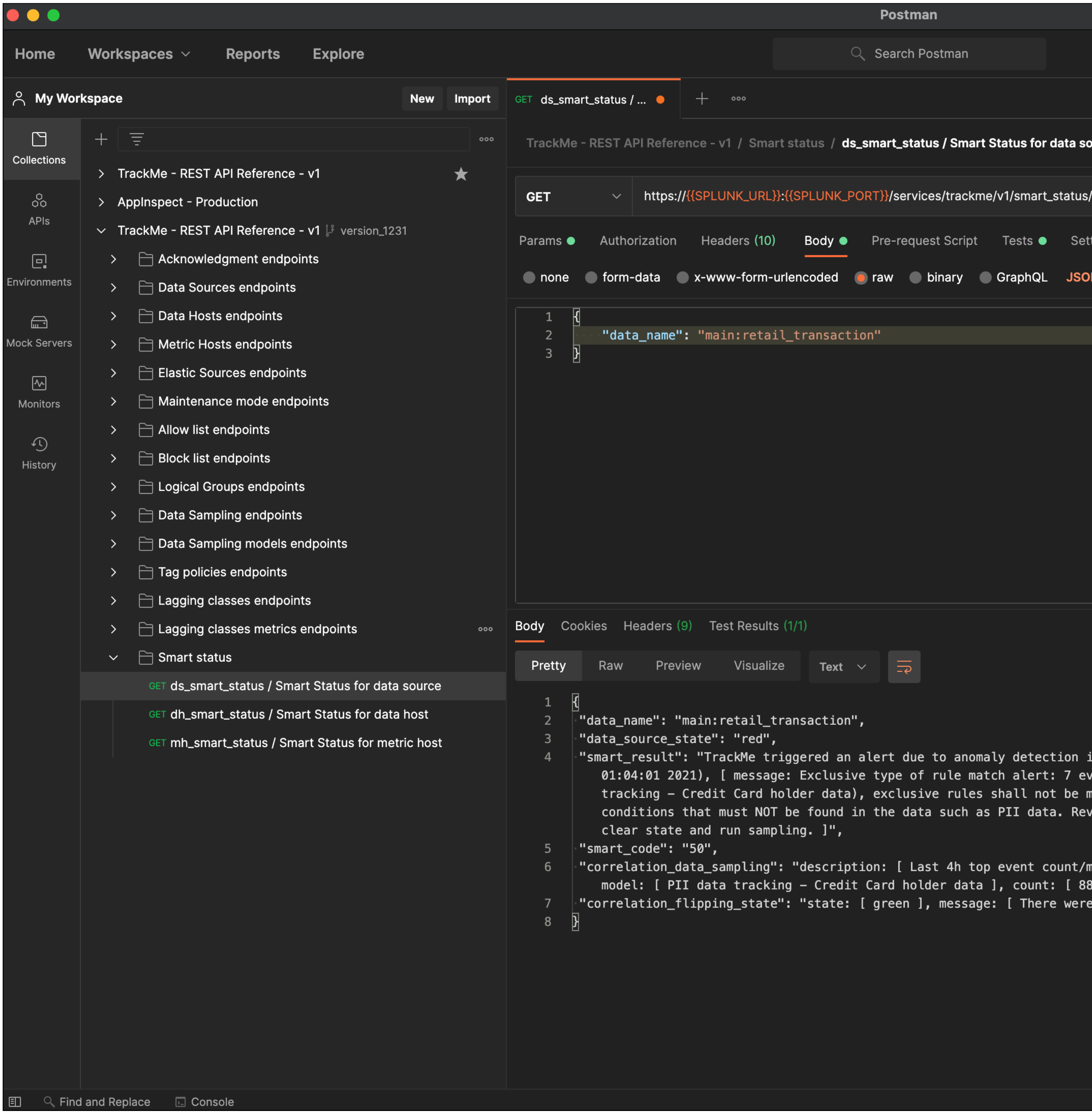

Postman
New
Import
Runner
My Workspace
Invite
Filter
History
Collections
APIs
New Collection
Trash
TrackMe - REST API Reference - v1
115 requests
version_1231
Acknowledgment endpoints
Data Sources endpoints
Data Hosts endpoints
Metric Hosts endpoints
GET mh_collection / Get full Metric Hosts collection
GET mh_by_key / Get metric host by _key
GET mh_by_name / Get metric host by name
POST mh_enable_monitoring / Enable monitoring
POST mh_disable_monitoring / Disable monitoring
POST mh_update_priority / Update priority
POST mh_reset / Reset metrics
DEL mh_delete_temporary / Delete temporary
DEL mh_delete_permanent / Delete permanently
Elastic Sources endpoints
Maintenance mode endpoints
Allow list endpoints
Block list endpoints
Logical Groups endpoints
Data Sampling endpoints
Data Sampling models endpoints
Tag policies endpoints
Lagging classes endpoints
Lagging classes metrics endpoints
Smart status
GET ds_smart_status / Smart Status for data sour...
GET dh_smart_status / Smart Status for data host
GET mh_smart_status / Smart Status for metric h...
Find and Replace
Console
GET dh_smart_status / Smart Statu...
GET mh_smart_status / Smart Statu...
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Body
Cookies
Headers (8)
Test Results (1/1)
Pretty
Raw
Preview
Visualize
Text

mh_smart_status / Smart Status for metric host

GET https://{{SPLUNK_URL}}:{{SPLUNK_PORT}}/services/trackme/v1/smart_status/mh_smart_status

```
{
    "metric_host": "telegraf-node1"
}
```

```
{
 "metric_host": "telegraf-node1",
 "metric_host_state": "green",
 "smart_result": "The metric host is currently in a normal state, therefore further inve
 "smart_code": "0",
 "correlation_flipping_state": "state: [ green ], message: [ There were no anomalies det
}
```



```
GET https://{{SPLUNK_URL}}:{{SPLUNK_PORT}}/services/trackme/v1/smart_status/
```

```
{
  "data_name": "main:retail_transaction"
}
```

```
{
"data_name": "main:retail_transaction",
"data_source_state": "red",
"smart_result": "TrackMe triggered an alert due to anomaly detection i
   01:04:01 2021), [ message: Exclusive type of rule match alert: 7 ev
   tracking - Credit Card holder data), exclusive rules shall not be m
   conditions that must NOT be found in the data such as PII data. Rev
   clear state and run sampling. ]",
"smart_code": "50",
"correlation_data_sampling": "description: [ Last 4h top event count/m
   model: [ PII data tracking - Credit Card holder data ], count: [ 88
"correlation_flipping_state": "state: [ green ], message: [ There were
}
```

See: *Smart Status endpoints*

3.1.7 Alerts tracking

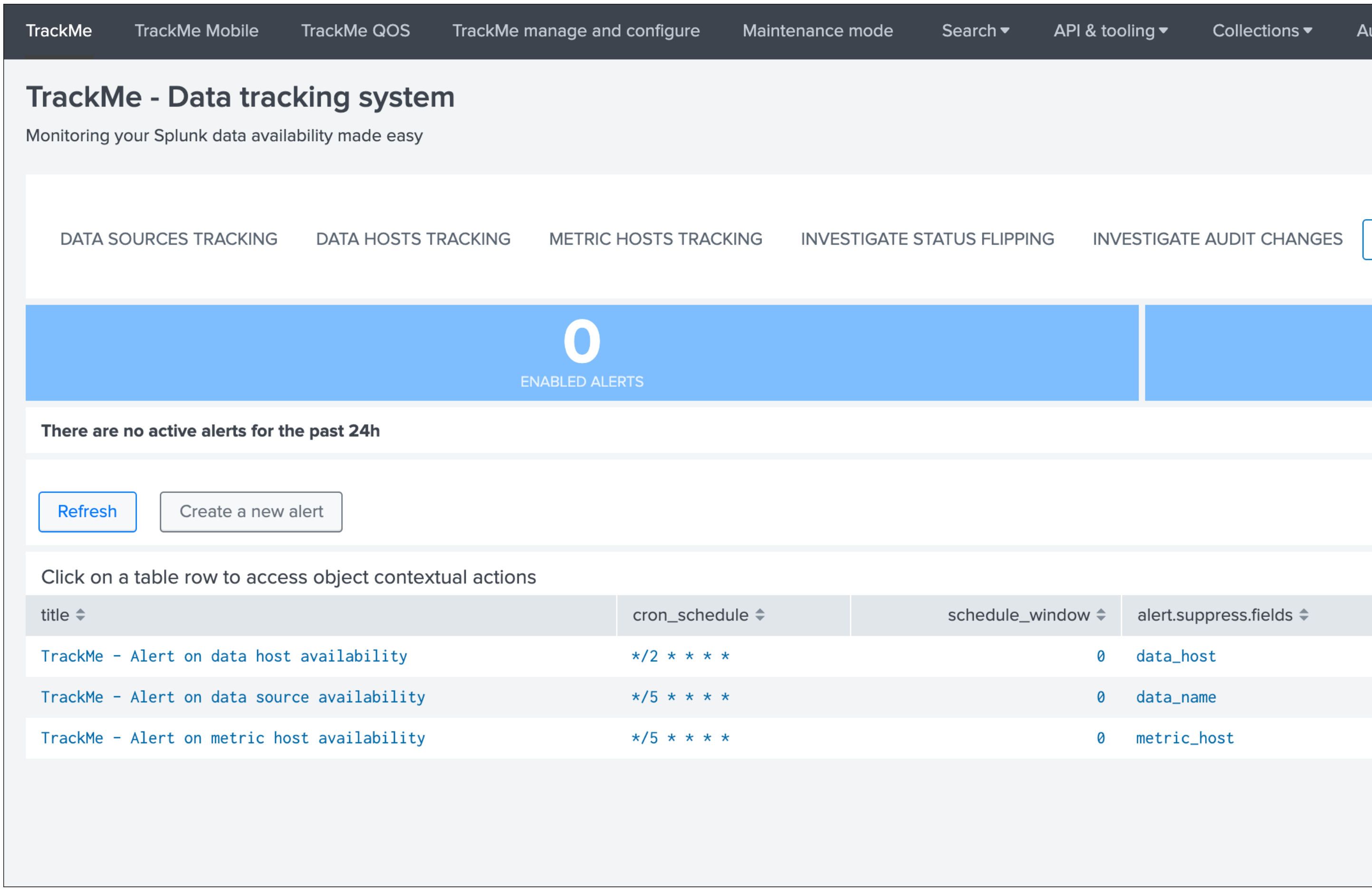
Alerts tracking

- TrackMe relies on Splunk alerts to provide automated results based on your preferences and usage
- One template alert is provided per type of entities (data sources / data hosts / metric hosts) which you can decide to enable and start using straight away
- As well, you can create custom alerts via an assistant which templates a TrackMe alert based on your preferences and choices
- Finally, TrackMe provides builtin alert actions that are used to extend the application functionalities

The alert topic is as well discussed at the configuration step: *Step 7: enabling out of the box alerts or create your own custom alerts*

Alerts tracking main screen

Within the main TrackMe UI, the alerts tracking screen is available as a selectable tab:



TrackMe - Data tracking system

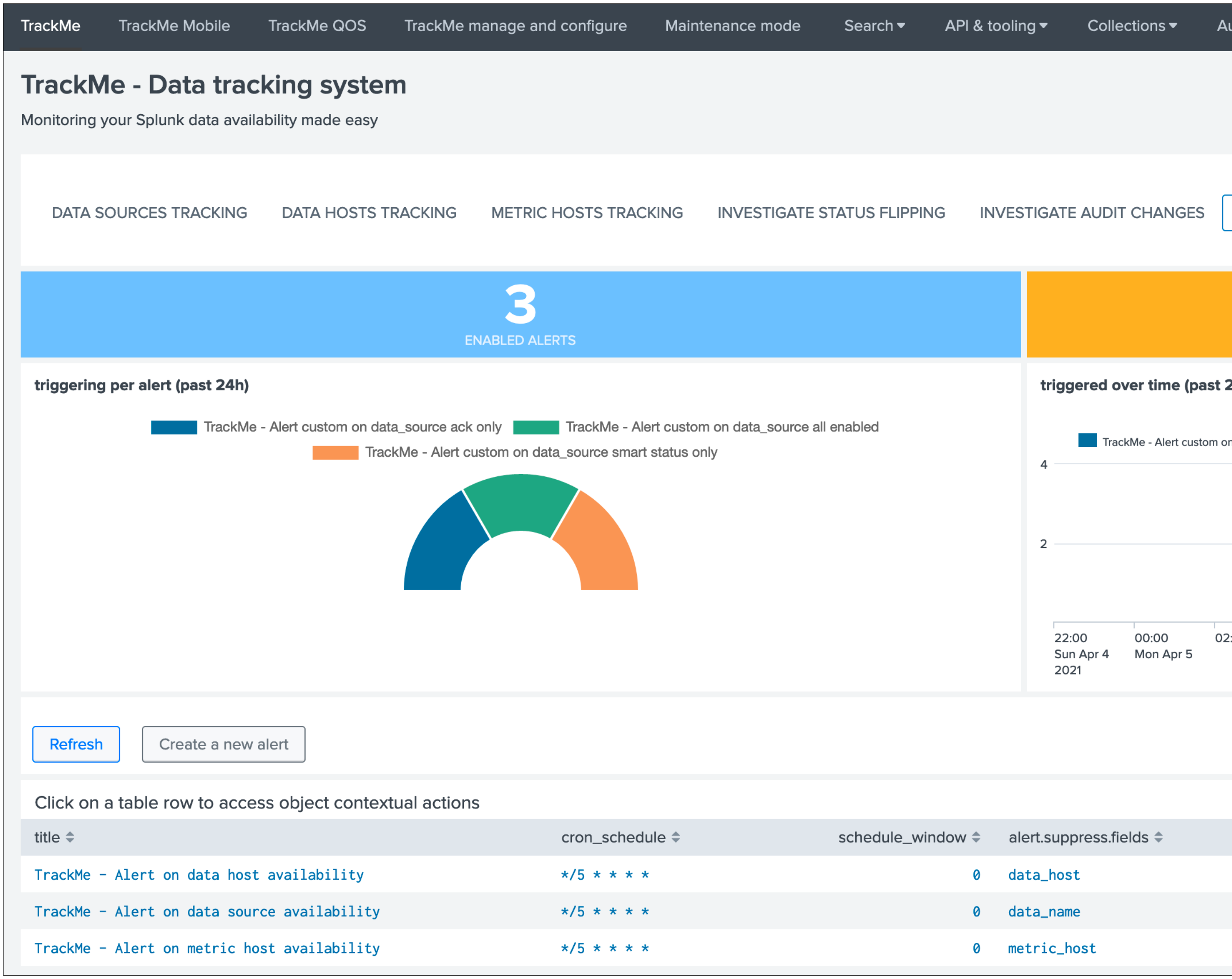
Monitoring your Splunk data availability made easy

There are no active alerts for the past 24h

Click on a table row to access object contextual actions

title	cron_schedule	schedule_window	alert.suppress.fields
TrackMe - Alert on data host availability	*/2 * * * *	0	data_host
TrackMe - Alert on data source availability	*/5 * * * *	0	data_name
TrackMe - Alert on metric host availability	*/5 * * * *	0	metric_host

Depending on the alerts that were enabled, and the actiity of the environment, the screen shows a 24 hours overview of the alerts activity:


TrackMe
TrackMe Mobile
TrackMe QOS
TrackMe manage and configure
Maintenance mode
Search ▾
API & tooling ▾
Collections ▾
Au
DATA SOURCES TRACKING
DATA HOSTS TRACKING
METRIC HOSTS TRACKING
INVESTIGATE STATUS FLIPPING
INVESTIGATE AUDIT CHANGES
3
ENABLED ALERTS
triggering per alert (past 24h)
TrackMe - Alert custom on data_source ack only
TrackMe - Alert custom on data_source all enabled
TrackMe - Alert custom on data_source smart status only
triggered over time (past 2
TrackMe - Alert custom or
4
2
22:00
Sun Apr 4
2021
00:00
Mon Apr 5
02:
Refresh
Create a new alert

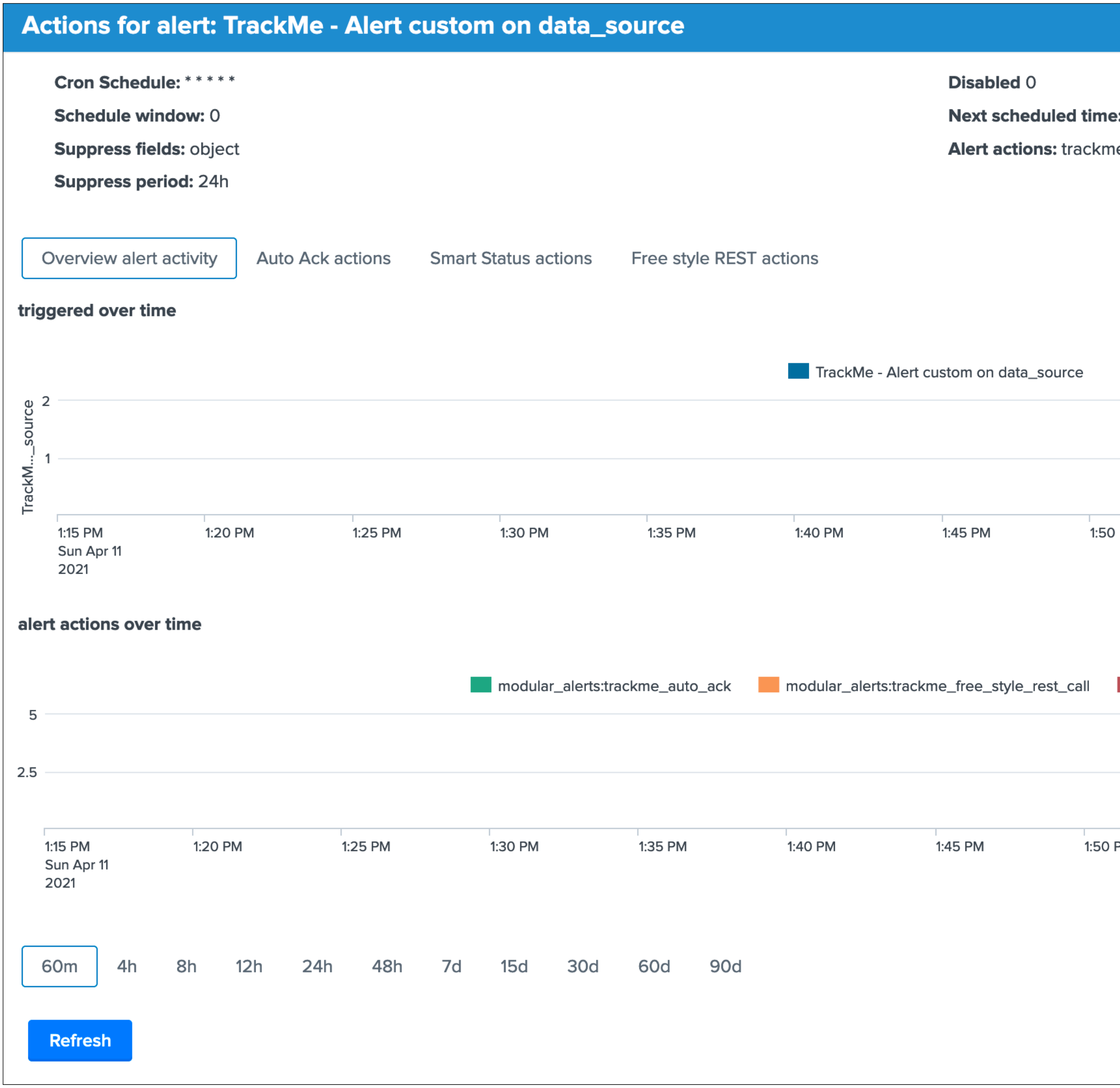
TrackMe - Data tracking system

Monitoring your Splunk data availability made easy

Click on a table row to access object contextual actions

title	cron_schedule	schedule_window	alert.suppress.fields
TrackMe - Alert on data host availability	*/5 * * * *	0	data_host
TrackMe - Alert on data source availability	*/5 * * * *	0	data_name
TrackMe - Alert on metric host availability	*/5 * * * *	0	metric_host

Clicking on any alert opens an overview window for this alert with shortcut to the Splunk alert editor and other functions:



Actions for alert: TrackMe - Alert custom on data_source

Cron Schedule: * * * * *

Schedule window: 0

Suppress fields: object

Suppress period: 24h

Disabled 0

Next scheduled time:

Alert actions: trackme

Alerts tracking: out of the box alerts

Alerts are provided out of the box that cover the basic alerting for all TrackMe entities:

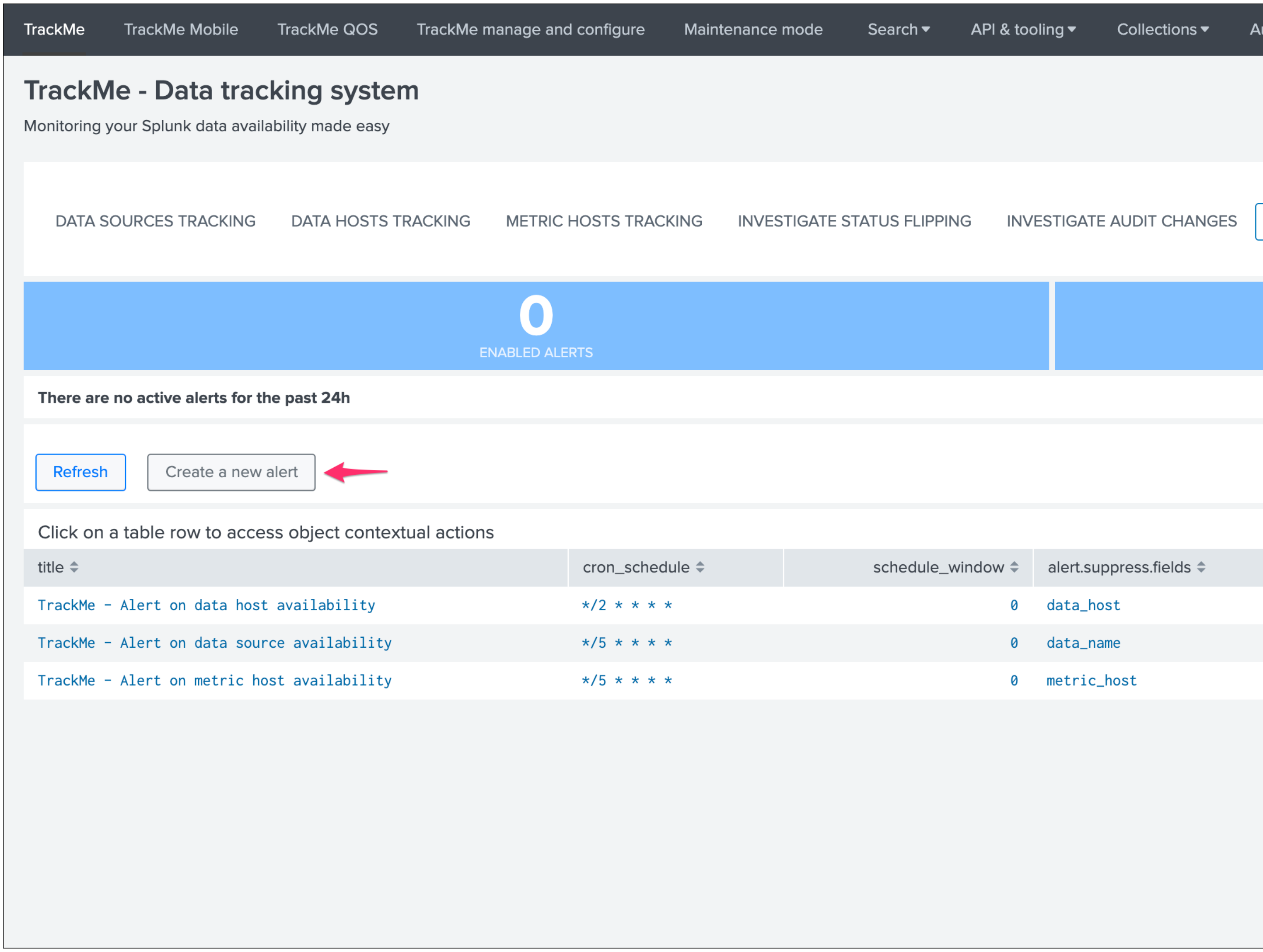
- `TrackMe - Alert on data source availability`
- `TrackMe - Alert on data host availability`
- `TrackMe - Alert on metric host availability`

Hint: Out of the box alerts

- Out of the box alerts are disabled by default, you need to enable alerts to start using them
- Alerts will trigger by default on `high priority` entities only, this is controlled via the macro definition `trackme_alerts_priority`
- Edit the alert to perform your third party integration, for example `sending emails` or creating `JIRA issues` based on Splunk alert actions capabilities
- Out of the box alert enable by default two TrackMe alert actions, `automatic acknowledgement` and the `Smart Status` alert actions
- The results of the `Smart Status` alert action are automatically indexed in the TrackMe summary index within the sourcetype `trackme_smart_status` and can be used for investigation purposes

Alerts tracking: custom alerts

You can use this interface to a create one or more custom alerts:



TrackMe - Data tracking system

Monitoring your Splunk data availability made easy

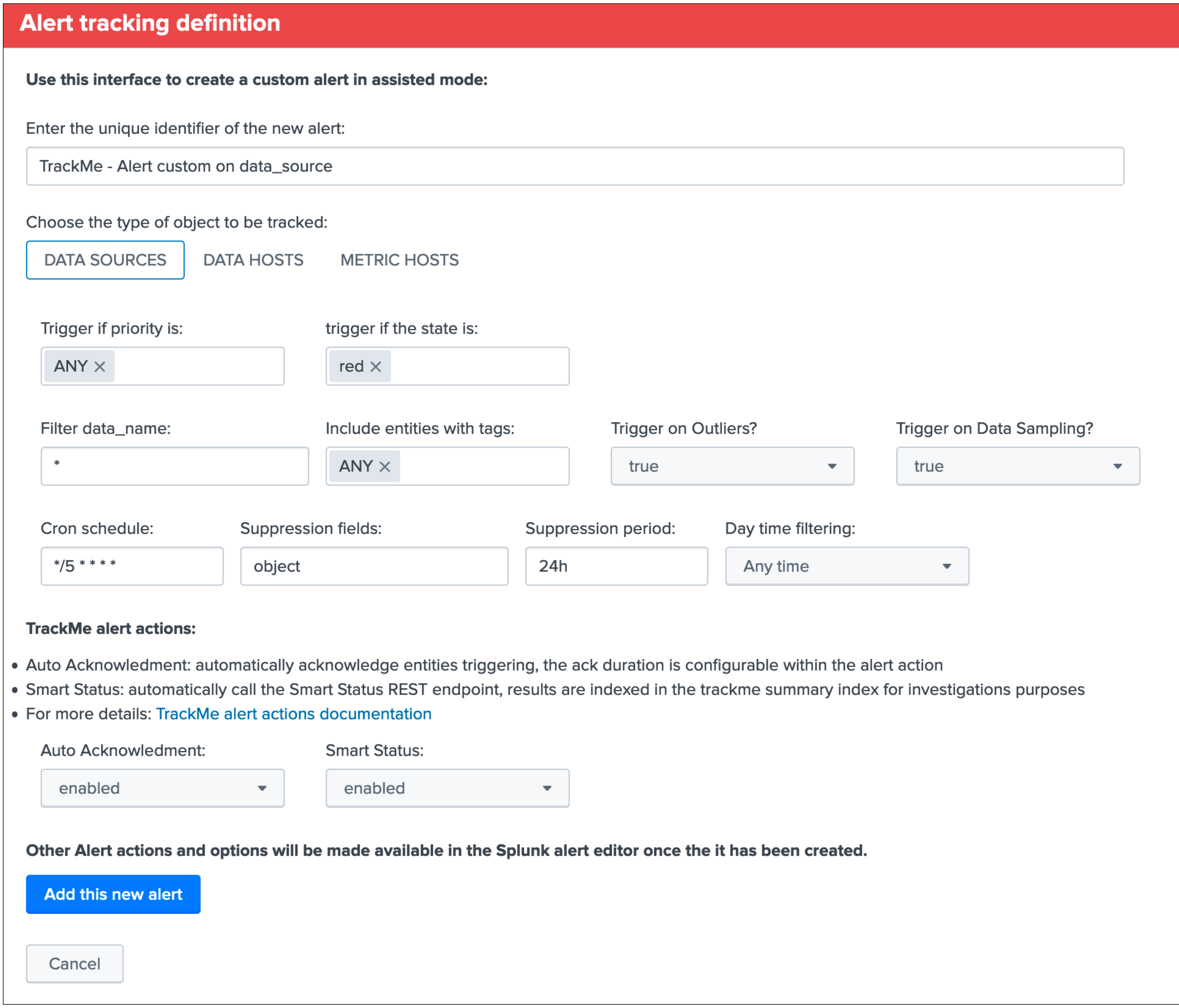
0
ENABLED ALERTS

There are no active alerts for the past 24h

Click on a table row to access object contextual actions

title	cron_schedule	schedule_window	alert.suppress.fields
TrackMe - Alert on data host availability	*/2 * * * *	0	data_host
TrackMe - Alert on data source availability	*/5 * * * *	0	data_name
TrackMe - Alert on metric host availability	*/5 * * * *	0	metric_host

This opens the assistant where you can choose between different builtin options depending on the type of entities to be monitoring:



Alert tracking definition

Use this interface to create a custom alert in assisted mode:

Enter the unique identifier of the new alert: TrackMe - Alert custom on data_source

Choose the type of object to be tracked:

Trigger if priority is: ANY ✕

trigger if the state is: red ✕

Filter data_name: *

Include entities with tags: ANY ✕

Trigger on Outliers? true

Trigger on Data Sampling? true

Cron schedule: */5 * * * *

Suppression fields: object

Suppression period: 24h

Day time filtering: Any time

TrackMe alert actions:

- Auto Acknowledment: automatically acknowledge entities triggering, the ack duration is configurable within the alert action
- Smart Status: automatically call the Smart Status REST endpoint, results are indexed in the trackme summary index for investigations purposes
- For more details: TrackMe alert actions documentation

Auto Acknowledment: enabled

Smart Status: enabled

Other Alert actions and options will be made available in the Splunk alert editor once the it has been created.

Once you have created a new alert, it will be immediately visible in the tracking alerts UI, and you can use the Splunk built alert editor to modify the alert to up to your needs such as enabling third party actions, emails actions and so forth.

Hint: Custom alert features

- Creating custom alerts provide several layers of flexibility depending on your choices and preferences
- You may for example have alerts handling lowest level of prority with a specific type of alert action, and have a specific alert for highly critical entities
- Advanced setup can easily be performed such as getting benefits from the tags features and multiple

alerts using tag policies to associate data sources and different types of alerts, recipients, actions. . .

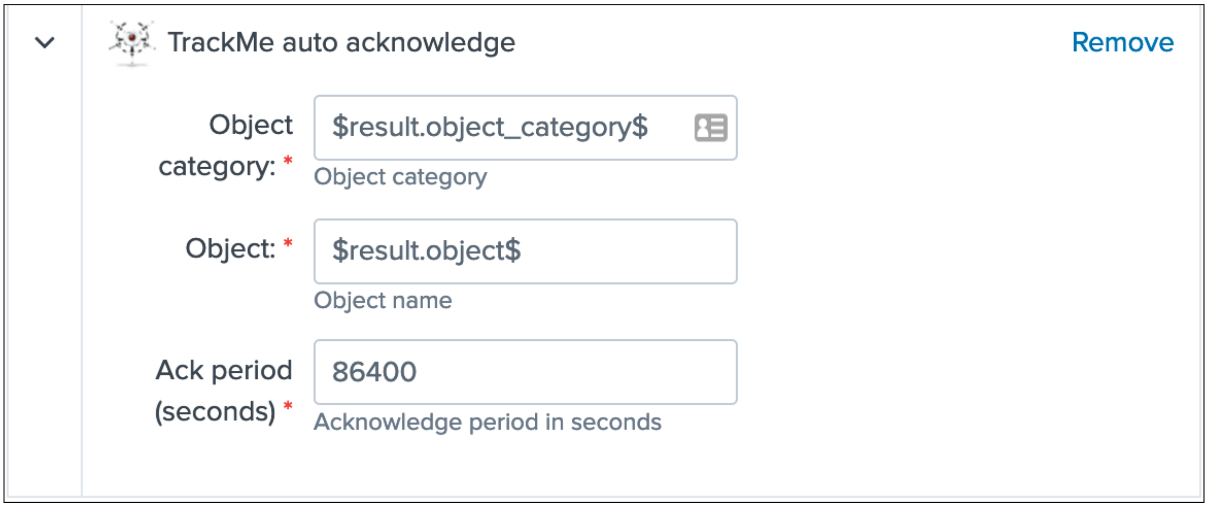
- You may decide if you wish to enable or disable the TrackMe `auto acknowledgement` and `Smart Status` alert actions while creating alerts through the assistant

Alerts tracking: TrackMe alert actions

TrackMe provides 3 builtin alert actions that help getting even more value from the application by performing easily some levels of automisation:

- `TrackMe auto acknowledge`
- `Trackme Smart Status`
- `TrackMe free style rest call`

Alert action: TrackMe auto acknowledge



TrackMe auto acknowledge

Object category: * $result.object_category$
Object category

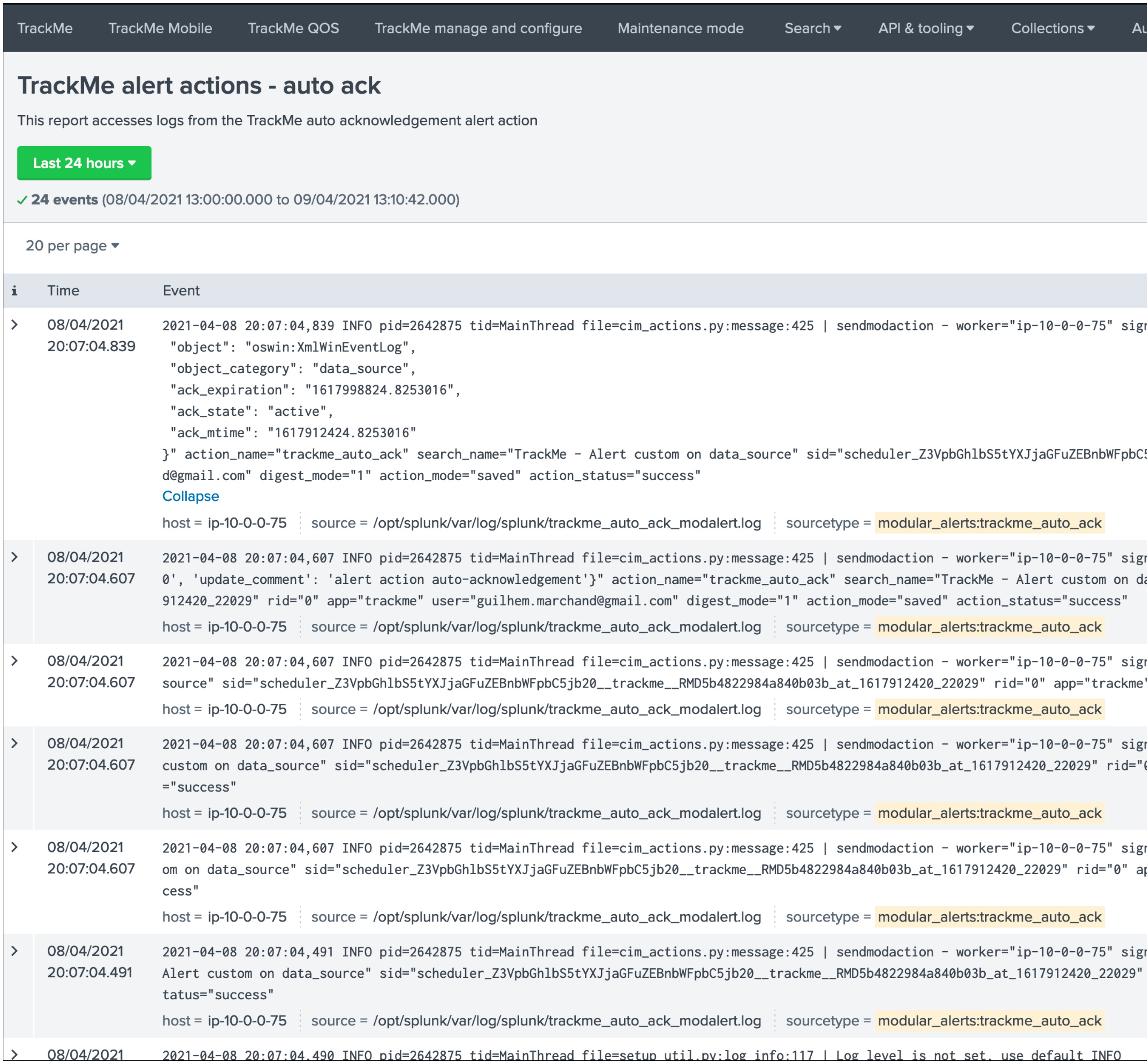
Object: * $result.object$
Object name

Ack period (seconds) * 86400
Acknowledge period in seconds

Auto acknowledgement

- This alert action allows automatically performing an acknowledgement of an entity that enters into a non green state.
- When an acknowledgement is enabled, the entity appears with a specific icon in the UI, you can control and extend the acknowledgement at any time.
- As long as an acknowledgement is enabled for a given entity, there will be no more alerts generated for it, which leaves time enough for the investigations, performing fine tuning if required or fixing the root cause of the issue.
- The alert action activity is logged in `(index="_internal" OR index="cim_modactions") sourcetype="modular_alerts:trackme_auto_ack"`
- A quick access report to the alert execution logs is available in the navigation application menu `API & tooling/TrackMe alert actions - auto ack`

Example of an auto acknowledge processing logs, at the end of the process the API endpoint JSON result is logged:



TrackMe alert actions - auto ack

This report accesses logs from the TrackMe auto acknowledgement alert action

✓ **24 events** (08/04/2021 13:00:00.000 to 09/04/2021 13:10:42.000)

i	Time	Event

> 08/04/2021 20:07:04.839

```
2021-04-08 20:07:04,839 INFO pid=2642875 tid=MainThread file=cim_actions.py:message:425 | sendmodaction - worker="ip-10-0-0-75" sig
 "object": "oswin:XmlWinEventLog",
 "object_category": "data_source",
 "ack_expiration": "1617998824.8253016",
 "ack_state": "active",
 "ack_mtime": "1617912424.8253016"
}" action_name="trackme_auto_ack" search_name="TrackMe - Alert custom on data_source" sid="scheduler_Z3VpbGhlbS5tYXJjaGFuZEBnbWFpbC5
d@gmail.com" digest_mode="1" action_mode="saved" action_status="success"
```

Collapse

host = ip-10-0-0-75 | source = /opt/splunk/var/log/splunk/trackme_auto_ack_modalert.log | sourcetype = modular_alerts:trackme_auto_ack

> 08/04/2021 20:07:04.607

```
2021-04-08 20:07:04,607 INFO pid=2642875 tid=MainThread file=cim_actions.py:message:425 | sendmodaction - worker="ip-10-0-0-75" sig
0', 'update_comment': 'alert action auto-acknowledgement'}" action_name="trackme_auto_ack" search_name="TrackMe - Alert custom on da
912420_22029" rid="0" app="trackme" user="guilhem.marchand@gmail.com" digest_mode="1" action_mode="saved" action_status="success"
```

host = ip-10-0-0-75 | source = /opt/splunk/var/log/splunk/trackme_auto_ack_modalert.log | sourcetype = modular_alerts:trackme_auto_ack

> 08/04/2021 20:07:04.607

```
2021-04-08 20:07:04,607 INFO pid=2642875 tid=MainThread file=cim_actions.py:message:425 | sendmodaction - worker="ip-10-0-0-75" sig
source" sid="scheduler_Z3VpbGhlbS5tYXJjaGFuZEBnbWFpbC5jb20__trackme__RMD5b4822984a840b03b_at_1617912420_22029" rid="0" app="trackme
```

host = ip-10-0-0-75 | source = /opt/splunk/var/log/splunk/trackme_auto_ack_modalert.log | sourcetype = modular_alerts:trackme_auto_ack

> 08/04/2021 20:07:04.607

```
2021-04-08 20:07:04,607 INFO pid=2642875 tid=MainThread file=cim_actions.py:message:425 | sendmodaction - worker="ip-10-0-0-75" sig
custom on data_source" sid="scheduler_Z3VpbGhlbS5tYXJjaGFuZEBnbWFpbC5jb20__trackme__RMD5b4822984a840b03b_at_1617912420_22029" rid="
="success"
```

host = ip-10-0-0-75 | source = /opt/splunk/var/log/splunk/trackme_auto_ack_modalert.log | sourcetype = modular_alerts:trackme_auto_ack

> 08/04/2021 20:07:04.607

```
2021-04-08 20:07:04,607 INFO pid=2642875 tid=MainThread file=cim_actions.py:message:425 | sendmodaction - worker="ip-10-0-0-75" sig
om on data_source" sid="scheduler_Z3VpbGhlbS5tYXJjaGFuZEBnbWFpbC5jb20__trackme__RMD5b4822984a840b03b_at_1617912420_22029" rid="0" ap
cess"
```

host = ip-10-0-0-75 | source = /opt/splunk/var/log/splunk/trackme_auto_ack_modalert.log | sourcetype = modular_alerts:trackme_auto_ack

> 08/04/2021 20:07:04.491

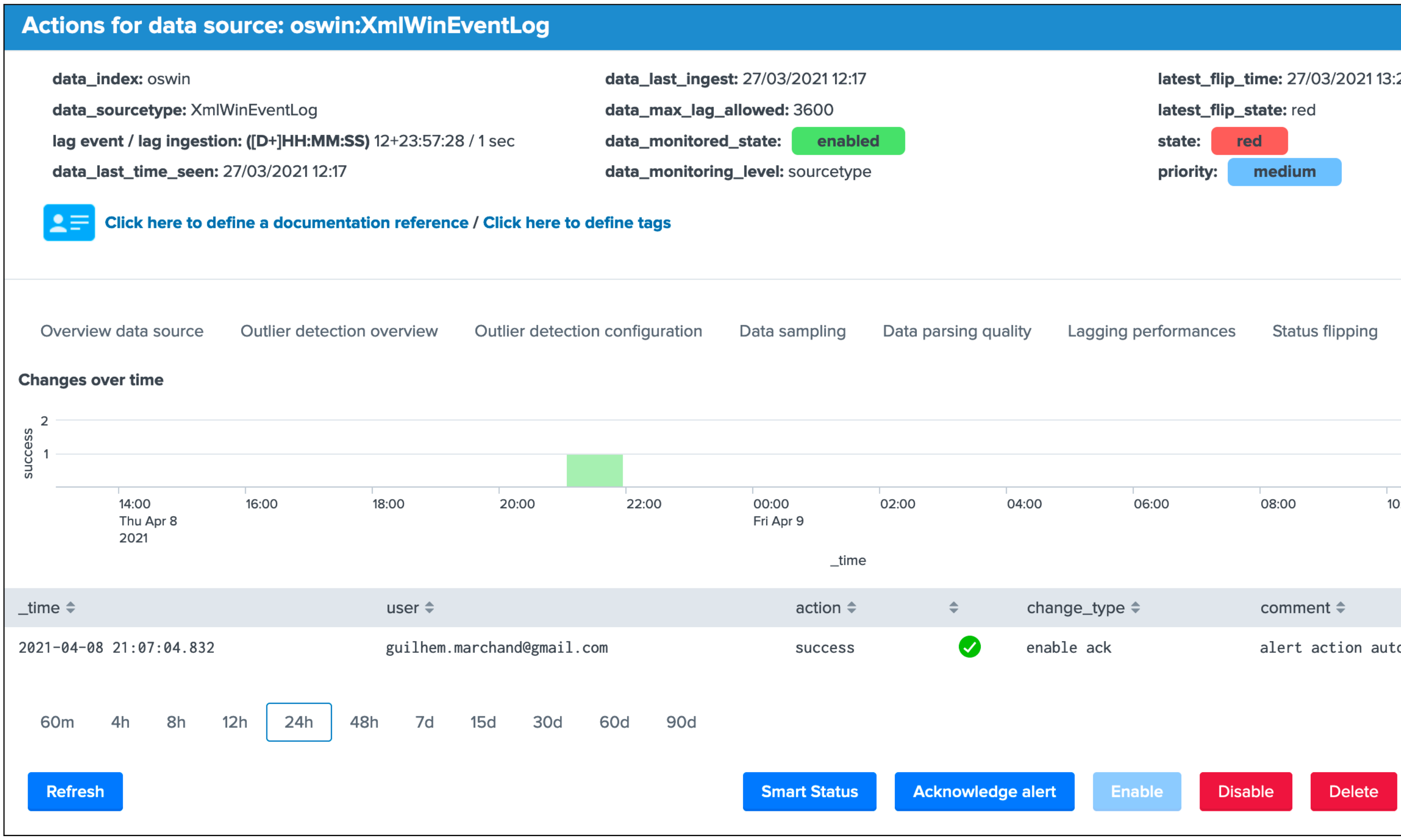
```
2021-04-08 20:07:04,491 INFO pid=2642875 tid=MainThread file=cim_actions.py:message:425 | sendmodaction - worker="ip-10-0-0-75" sig
Alert custom on data_source" sid="scheduler_Z3VpbGhlbS5tYXJjaGFuZEBnbWFpbC5jb20__trackme__RMD5b4822984a840b03b_at_1617912420_22029"
tatus="success"
```

host = ip-10-0-0-75 | source = /opt/splunk/var/log/splunk/trackme_auto_ack_modalert.log | sourcetype = modular_alerts:trackme_auto_ack

> 08/04/2021

```
2021-04-08 20:07:04.490 INFO pid=2642875 tid=MainThread file=setup_util.py:log_info:117 | Log level is not set, use default INFO
```

*An audit change event is automatically logged and visible in the UI:**



Actions for data source: oswin:XmlWinEventLog

data_index: oswin

data_sourcetype: XmlWinEventLog

lag event / lag ingestion: ([D+]HH:MM:SS) 12+23:57:28 / 1 sec

data_last_time_seen: 27/03/2021 12:17

data_last_ingest: 27/03/2021 12:17

data_max_lag_allowed: 3600

data_monitored_state: enabled

data_monitoring_level: sourcetype

latest_flip_time: 27/03/2021 13:2

latest_flip_state: red

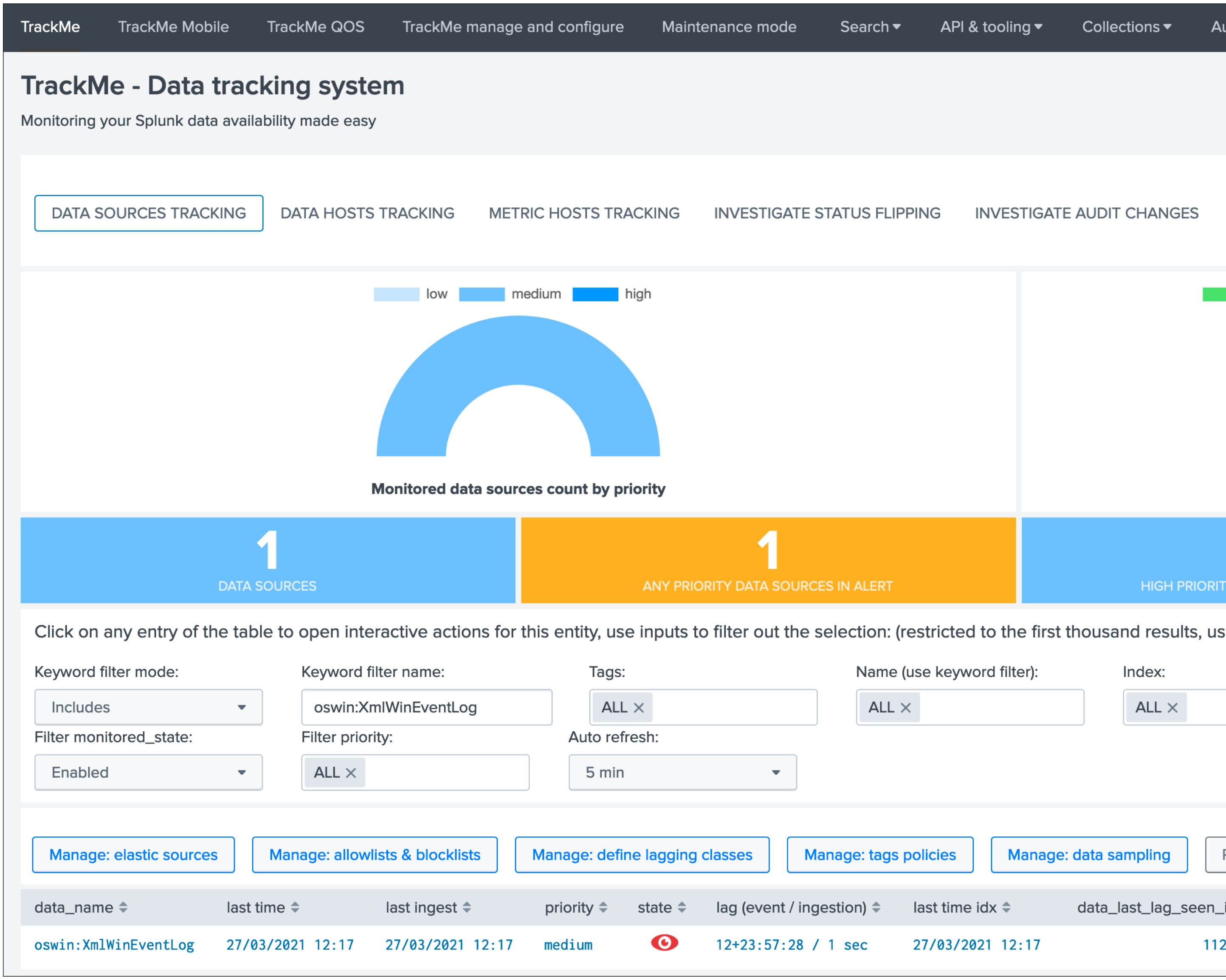
state: red

priority: medium

Click here to define a documentation reference / Click here to define tags

_time	user	action		change_type	comment
2021-04-08 21:07:04.832	guilhem.marchand@gmail.com	success	✓	enable ack	alert action auto

The entity has the acknowledged icon visible in the main UI screen:



TrackMe - Data tracking system

Monitoring your Splunk data availability made easy

Click on any entry of the table to open interactive actions for this entity, use inputs to filter out the selection: (restricted to the first thousand results, us

Keyword filter mode: Includes

Keyword filter name: oswin:XmlWinEventLog

Tags: ALL ×

Name (use keyword filter): ALL ×

Index: ALL ×

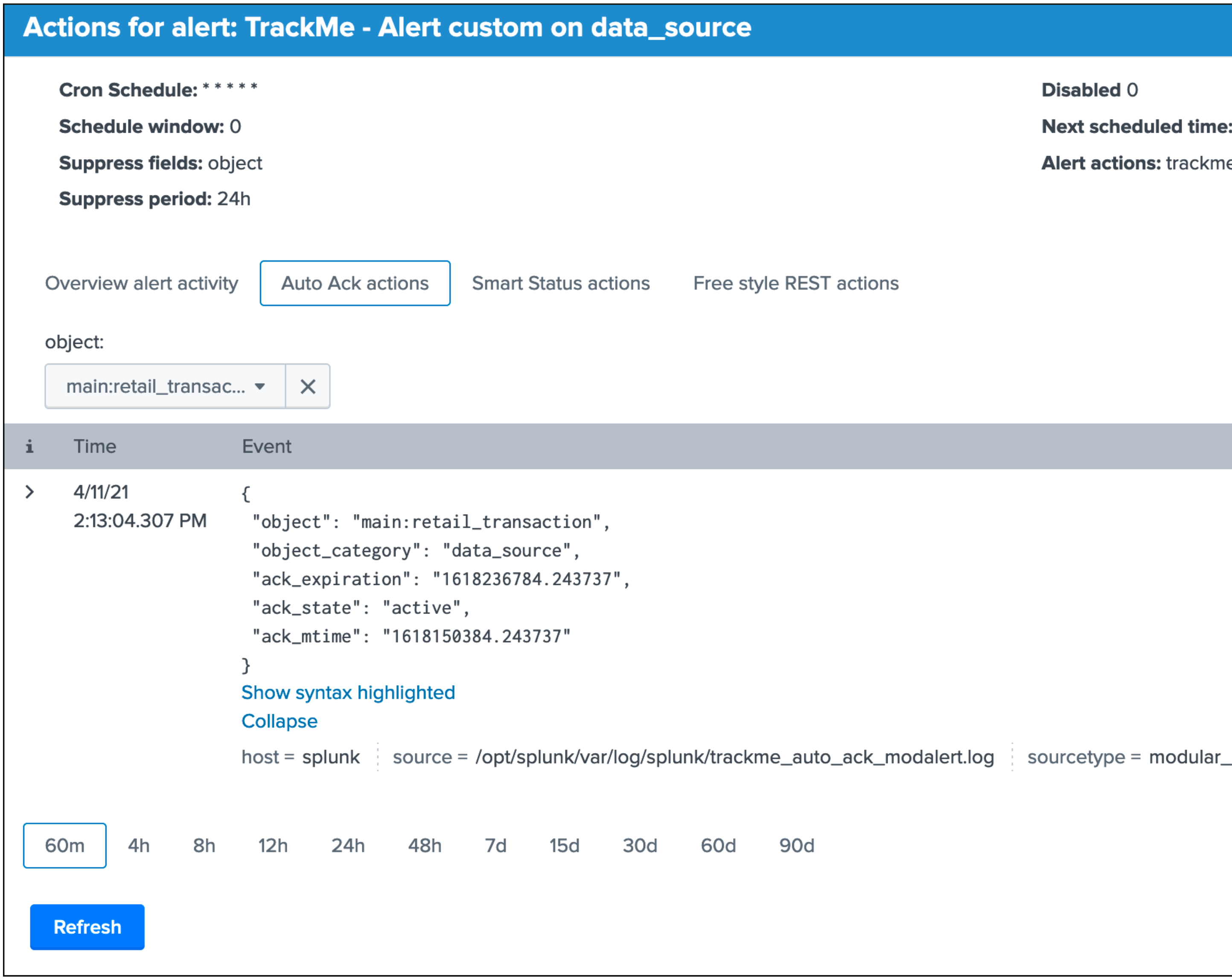
Filter monitored_state: Enabled

Filter priority: ALL ×

Auto refresh: 5 min

data_name	last time	last ingest	priority	state	lag (event / ingestion)	last time idx	data_last_lag_seen_i
oswin:XmlWinEventLog	27/03/2021 12:17	27/03/2021 12:17	medium	👁	12+23:57:28 / 1 sec	27/03/2021 12:17	112

The result from the Ack endpoint call can be accessed within the UI in the alert actions screen of the alert that generated the call:



Actions for alert: TrackMe - Alert custom on data_source

Cron Schedule: * * * * *

Schedule window: 0

Suppress fields: object

Suppress period: 24h

Disabled 0

Next scheduled time:

Alert actions: trackme

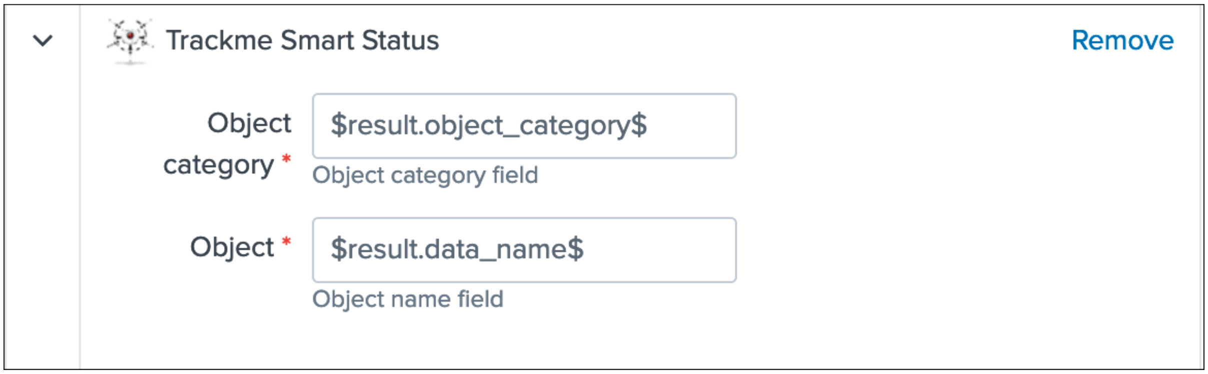
object: main:retail_transac...

i	Time	Event
>	4/11/21 2:13:04.307 PM	{ "object": "main:retail_transaction", "object_category": "data_source", "ack_expiration": "1618236784.243737", "ack_state": "active", "ack_mtime": "1618150384.243737" }

```
{
 "object": "main:retail_transaction",
 "object_category": "data_source",
 "ack_expiration": "1618236784.243737",
 "ack_state": "active",
 "ack_mtime": "1618150384.243737"
}
```

host = splunk | source = /opt/splunk/var/log/splunk/trackme_auto_ack_modalert.log | sourcetype = modular_

Alert action: Trackme Smart Status



Object category * $result.object_category$
Object category field

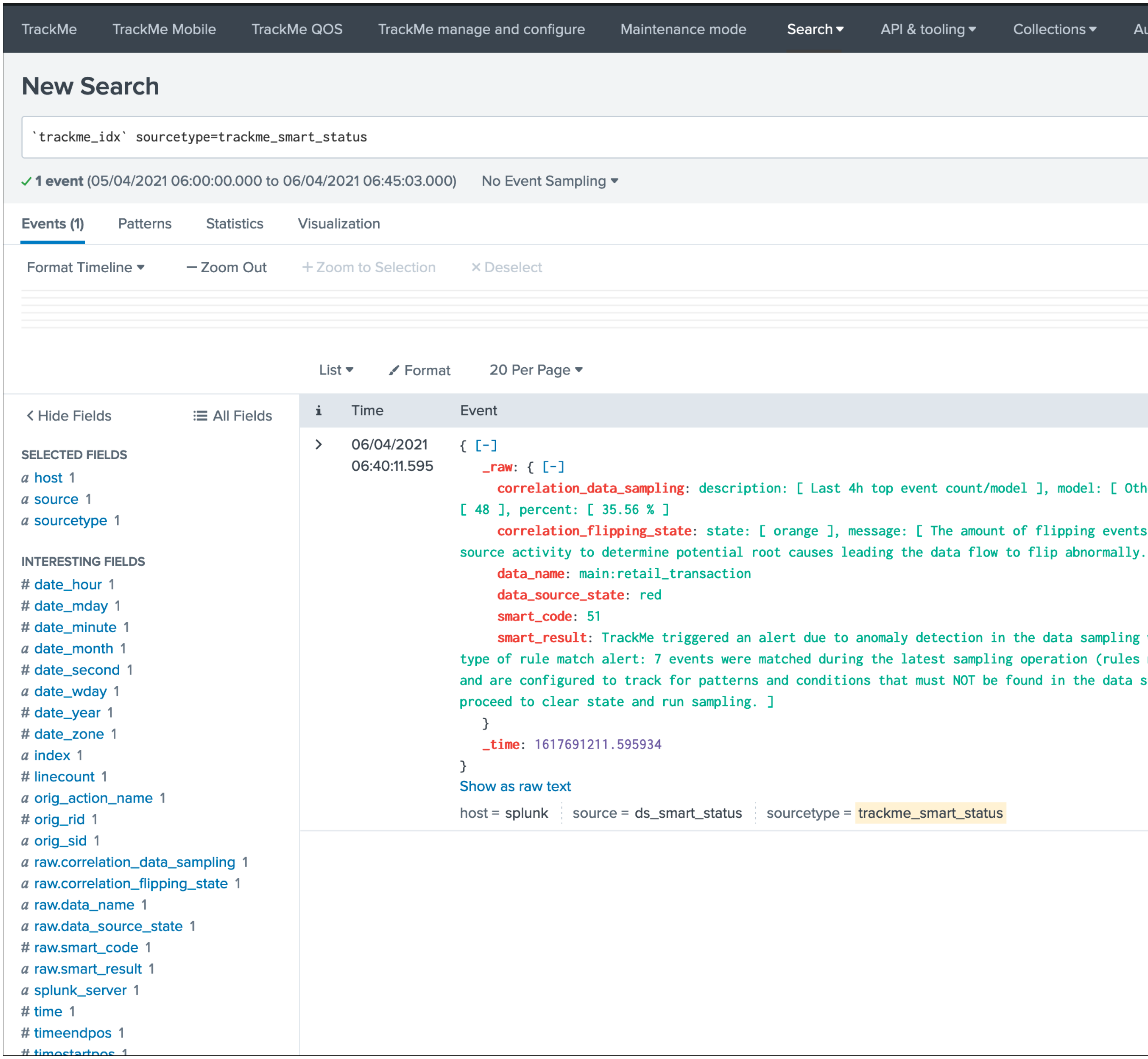
Object * $result.data_name$
Object name field

Smart Status alert action

- The Smart Status is a very advanced feature of TrackMe which performs automated investigations conditioned by the context of the entity
- In normal circumstances, you run the Smart Status action by performing a call to the TrackMe Smart Status API endpoint, or using the Smart Status functions builtin in the TrackMe UI, for more details see: *Smart Status*
- Using the alert action, the Smart Status action is performed automatically immediately when the entity triggers, and its result is indexed in the TrackMe summary event index defined in the macro `trackme_idx`
- The alert action activity is logged in `(index="_internal" OR index="cim_modactions") sourcetype="modular_alerts:trackme_smart_status"`
- the alert action result (the server response) is indexed in `` `trackme_idx` sourcetype=trackme_smart_status ``
- A quick access report to the alert execution logs is available in the navigation application menu `API & tooling/TrackMe alert actions - Smart Status`
- A quick access report fo the Smart Status results indexes is available in the navigation application menu `API & tooling/TrackMe events - Alert actions results`

Example: the alert triggers for a data source, the Smart Status action is executed and its result is indexed

```
`trackme_idx` sourcetype=trackme_smart_status
```



```
`trackme_idx` sourcetype=trackme_smart_status
```

✓ 1 event (05/04/2021 06:00:00.000 to 06/04/2021 06:45:03.000)

SELECTED FIELDS
a host 1
a source 1
a sourcetype 1

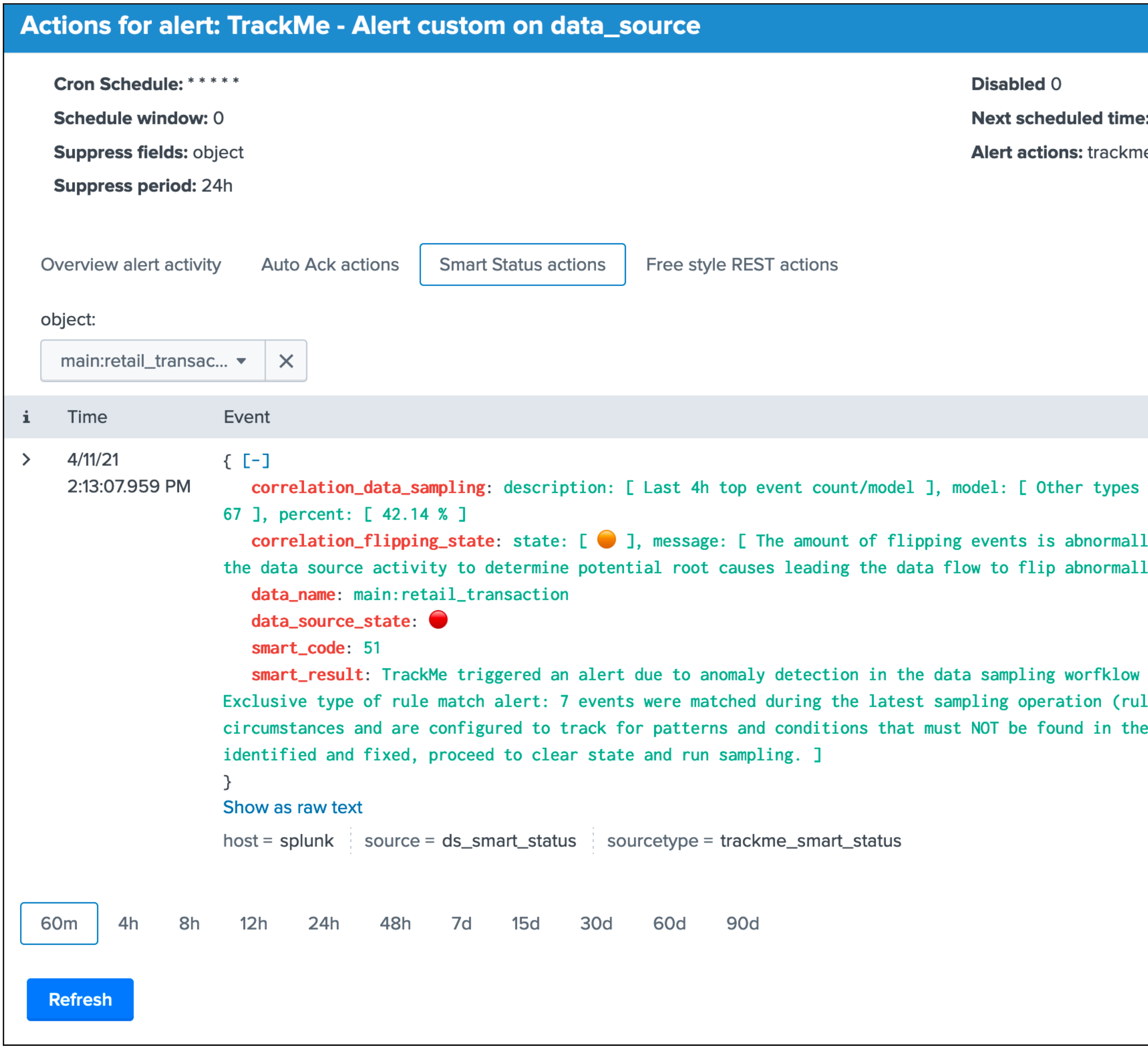
INTERESTING FIELDS
# date_hour 1
# date_mday 1
# date_minute 1
a date_month 1
# date_second 1
a date_wday 1
# date_year 1
# date_zone 1
a index 1
# linecount 1
a orig_action_name 1
# orig_rid 1
a orig_sid 1
a raw.correlation_data_sampling 1
a raw.correlation_flipping_state 1
a raw.data_name 1
a raw.data_source_state 1
# raw.smart_code 1
a raw.smart_result 1
a splunk_server 1
# time 1
# timeendpos 1
# timestartpos 1

06/04/2021
06:40:11.595

```
{ [-]
   _raw: { [-]
     correlation_data_sampling: description: [ Last 4h top event count/model ], model: [ Oth
[ 48 ], percent: [ 35.56 % ]
     correlation_flipping_state: state: [ orange ], message: [ The amount of flipping events
source activity to determine potential root causes leading the data flow to flip abnormally.
     data_name: main:retail_transaction
     data_source_state: red
     smart_code: 51
     smart_result: TrackMe triggered an alert due to anomaly detection in the data sampling 
type of rule match alert: 7 events were matched during the latest sampling operation (rules 
and are configured to track for patterns and conditions that must NOT be found in the data s
proceed to clear state and run sampling. ]
   }
   _time: 1617691211.595934
}
```

host = splunk | source = ds_smart_status | sourcetype = trackme_smart_status

The result from the Smart Status endpoint call can be accessed within the UI in the alert actions screen of the alert that generated the call:



Actions for alert: TrackMe - Alert custom on data_source

Cron Schedule: * * * * *

Schedule window: 0

Suppress fields: object

Suppress period: 24h

Disabled 0

Next scheduled time:

Alert actions: trackme

object: main:retail_transac...

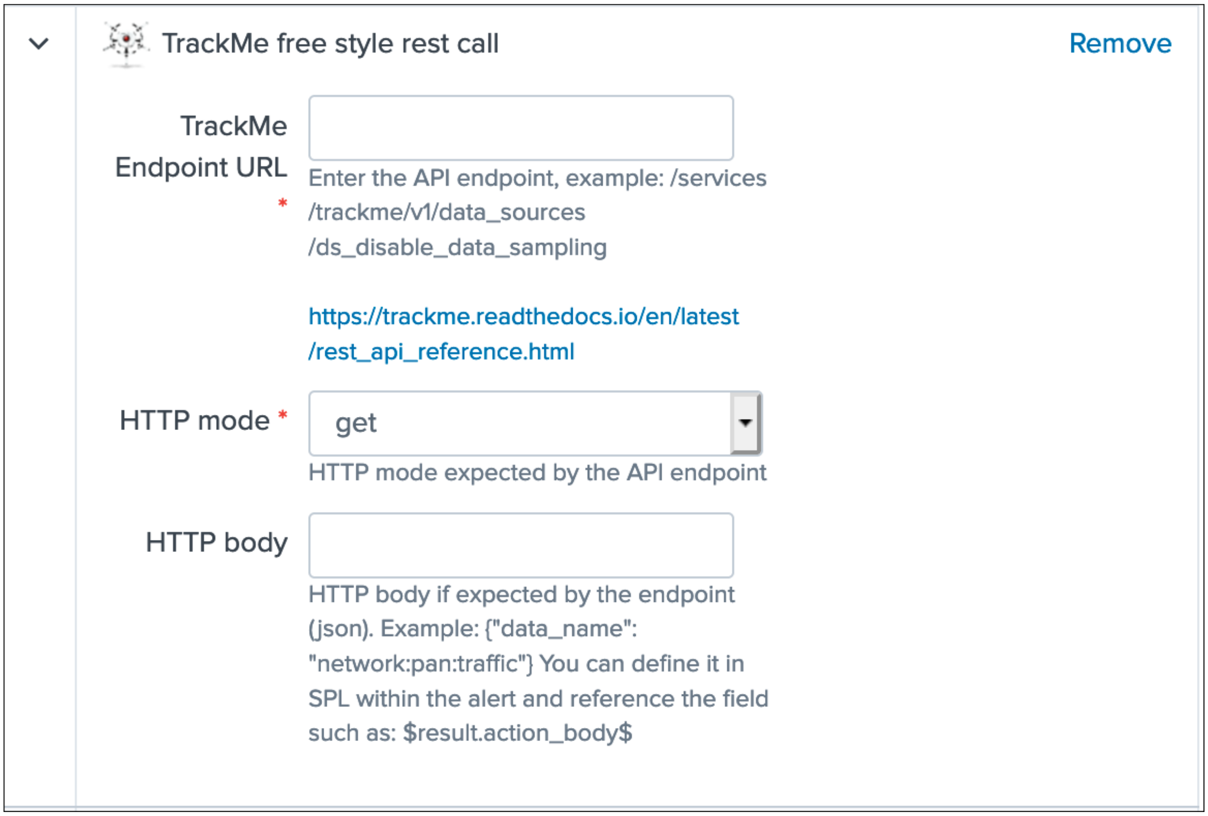
i	Time	Event
>	4/11/21 2:13:07.959 PM	

```
{ [-]
   correlation_data_sampling: description: [ Last 4h top event count/model ], model: [ Other types
67 ], percent: [ 42.14 % ]
   correlation_flipping_state: state: [ 🟠 ], message: [ The amount of flipping events is abnormall
the data source activity to determine potential root causes leading the data flow to flip abnormall
   data_name: main:retail_transaction
   data_source_state: 🔴
   smart_code: 51
   smart_result: TrackMe triggered an alert due to anomaly detection in the data sampling worfklow
Exclusive type of rule match alert: 7 events were matched during the latest sampling operation (rul
circumstances and are configured to track for patterns and conditions that must NOT be found in the
identified and fixed, proceed to clear state and run sampling. ]
}
```

Show as raw text

host = splunk | source = ds_smart_status | sourcetype = trackme_smart_status

Alert action: TrackMe free style rest call



TrackMe free style rest call

Remove

TrackMe Endpoint URL *

Enter the API endpoint, example: /services/trackme/v1/data_sources/ds_disable_data_sampling

https://trackme.readthedocs.io/en/latest/rest_api_reference.html

HTTP mode * get

HTTP mode expected by the API endpoint

HTTP body

HTTP body if expected by the endpoint (json). Example: {"data_name": "network:pan:traffic"} You can define it in SPL within the alert and reference the field such as: $result.action_body$

Free style alert action

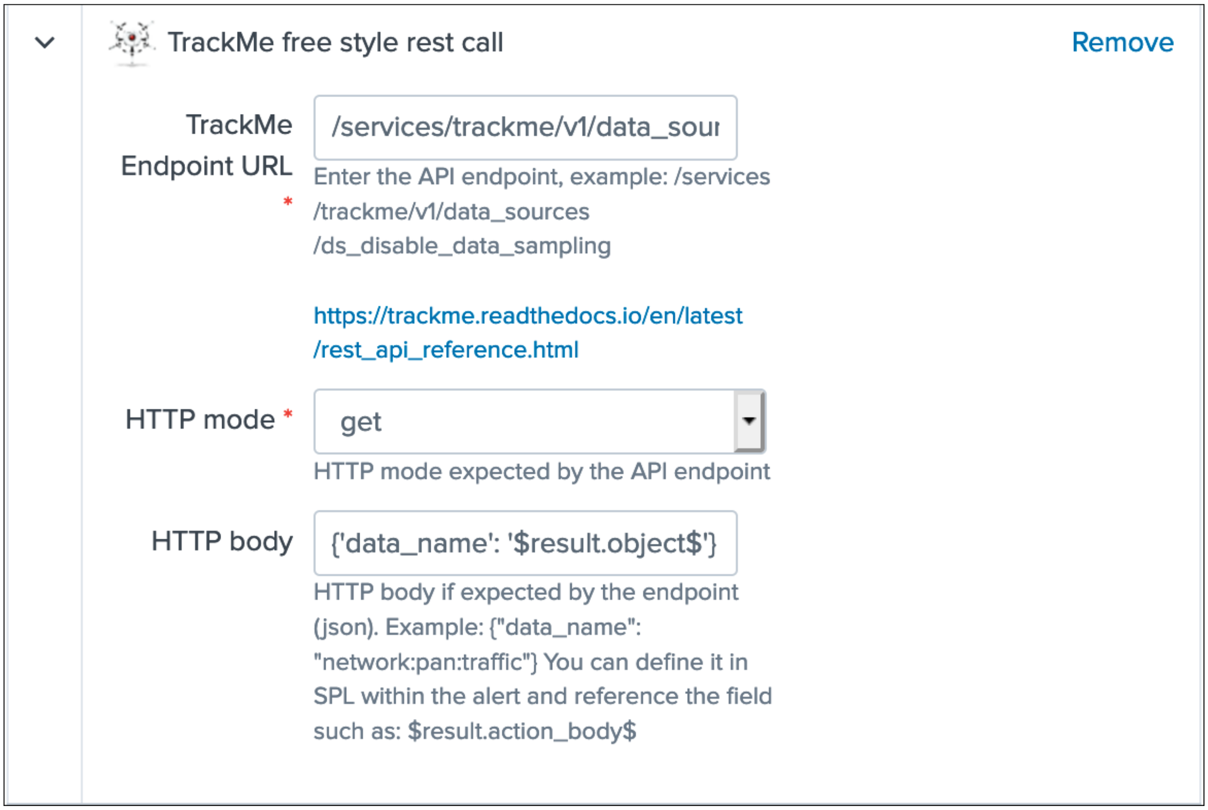
- The free style alert action allows you to call any of the TrackMe REST API endpoint to perform an automated action when the alert triggers
- The endpoint and its HTTP mode are configured in the alert action, if a body is expected by the endpoint, you can specify it statistically or recycle a field containing its value that you would define in SPL
- This alert action allows you to setup easily a custom workflow when the alert triggers dependending on your preference and context
- The alert action activity is logged in `(index="_internal" OR index="cim_modactions") sourcetype="modular_alerts:trackme_free_style_rest_call"`
- the alert action result (the server response) is indexed in `` `trackme_idx` sourcetype=trackme_alert_action ``
- A quick access report to the alert execution logs is available in the navigation application menu `TrackMe alert actions - free style`
- A quick access report fo the Smart Status results indexes is available in the navigation application menu `API & tooling/TrackMe events - Alert actions results`

The following example will generate an event of the full data source record as it is when the alert triggers:

- `TrackMe Endpoint URL`: /services/trackme/v1/data_sources/ds_by_name
- `HTTP mode`: get

- `HTTP body:`

```
{'data_name': '$result.object$'}
```



TrackMe free style rest call

TrackMe Endpoint URL *: /services/trackme/v1/data_sour

Enter the API endpoint, example: /services/trackme/v1/data_sources/ds_disable_data_sampling

https://trackme.readthedocs.io/en/latest/rest_api_reference.html

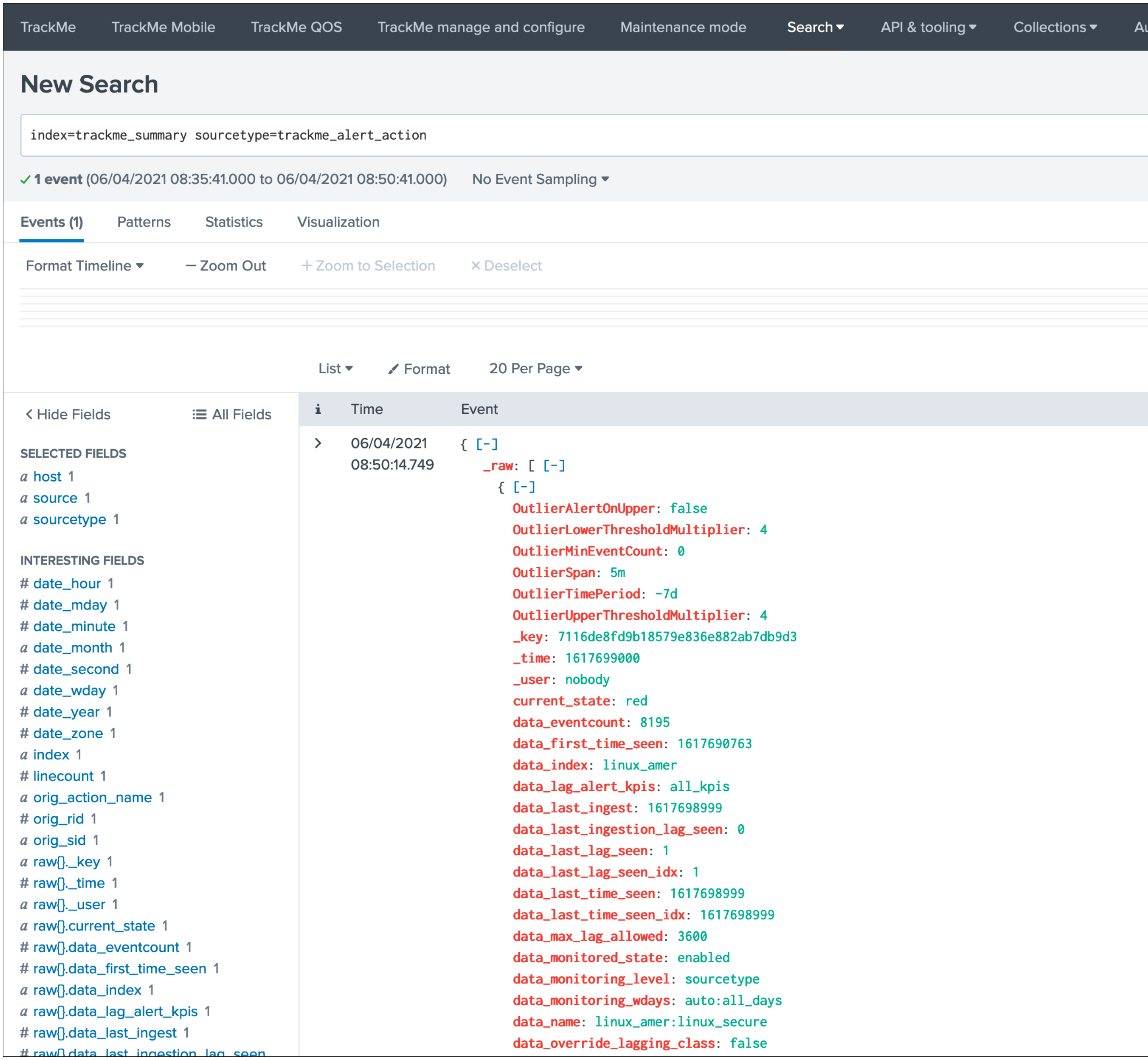
HTTP mode *: get

HTTP mode expected by the API endpoint

HTTP body: {'data_name': '$result.object$'}

HTTP body if expected by the endpoint (json). Example: {"data_name": "network:pan:traffic"} You can define it in SPL within the alert and reference the field such as: $result.action_body$

When the alert triggers:



```
index=trackme_summary sourcetype=trackme_alert_action
```

✓ 1 event (06/04/2021 08:35:41.000 to 06/04/2021 08:50:41.000)

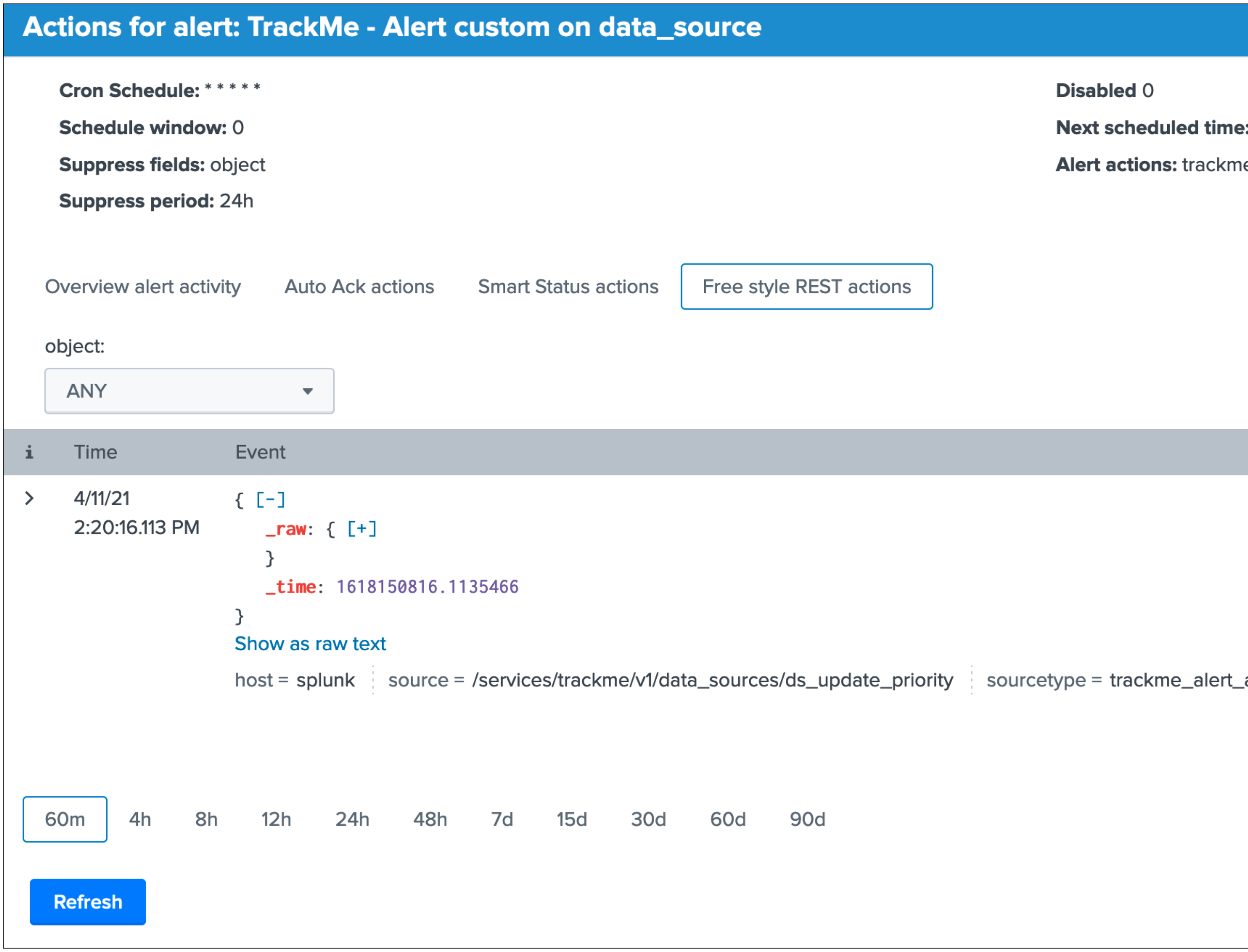
SELECTED FIELDS
- *a* host 1
- *a* source 1
- *a* sourcetype 1

INTERESTING FIELDS
- # date_hour 1
- # date_mday 1
- # date_minute 1
- *a* date_month 1
- # date_second 1
- *a* date_wday 1
- # date_year 1
- # date_zone 1
- *a* index 1
- # linecount 1
- *a* orig_action_name 1
- # orig_rid 1
- *a* orig_sid 1
- *a* raw{}._key 1
- # raw{}._time 1
- *a* raw{}._user 1
- *a* raw{}.current_state 1
- # raw{}.data_eventcount 1
- # raw{}.data_first_time_seen 1
- *a* raw{}.data_index 1
- *a* raw{}.data_lag_alert_kpis 1
- # raw{}.data_last_ingest 1
- # raw{}.data_last_ingestion_lag_seen

06/04/2021 08:50:14.749

```
{ [-]
  _raw: [ [-]
    { [-]
      OutlierAlertOnUpper: false
      OutlierLowerThresholdMultiplier: 4
      OutlierMinEventCount: 0
      OutlierSpan: 5m
      OutlierTimePeriod: -7d
      OutlierUpperThresholdMultiplier: 4
      _key: 7116de8fd9b18579e836e882ab7db9d3
      _time: 1617699000
      _user: nobody
      current_state: red
      data_eventcount: 8195
      data_first_time_seen: 1617690763
      data_index: linux_amer
      data_lag_alert_kpis: all_kpis
      data_last_ingest: 1617698999
      data_last_ingestion_lag_seen: 0
      data_last_lag_seen: 1
      data_last_lag_seen_idx: 1
      data_last_time_seen: 1617698999
      data_last_time_seen_idx: 1617698999
      data_max_lag_allowed: 3600
      data_monitored_state: enabled
      data_monitoring_level: sourcetype
      data_monitoring_wdays: auto:all_days
      data_name: linux_amer:linux_secure
      data_override_lagging_class: false
```

The result from the Smart Status endpoint call can be accessed within the UI in the alert actions screen of the alert that generated the call:



Actions for alert: TrackMe - Alert custom on data_source

Cron Schedule: * * * * *

Schedule window: 0

Suppress fields: object

Suppress period: 24h

Disabled 0

Next scheduled time:

Alert actions: trackme

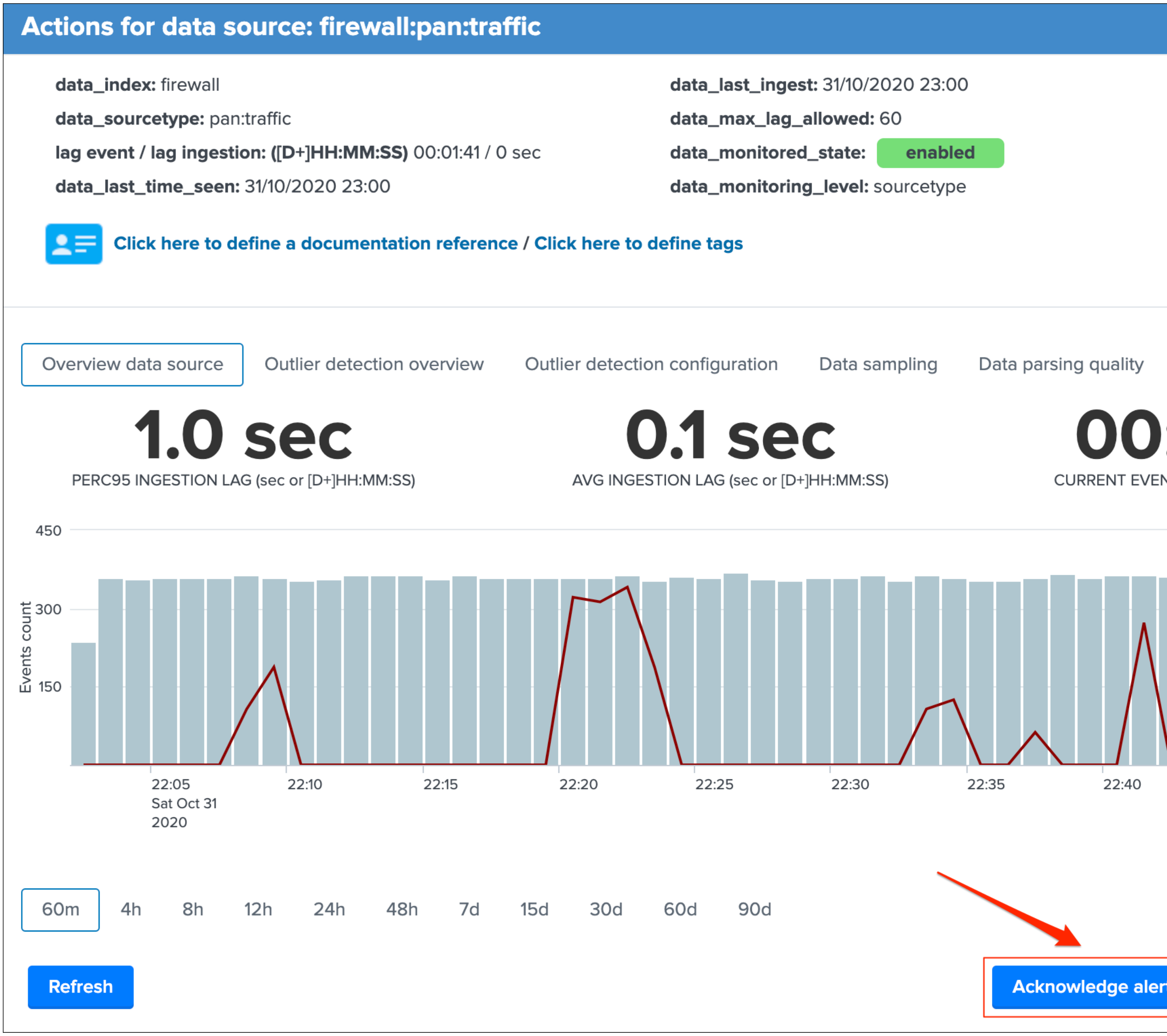
object: ANY

i	Time	Event
>	4/11/21 2:20:16.113 PM	{ [-] _raw: { [+] } _time: 1618150816.1135466 } Show as raw text host = splunk source = /services/trackme/v1/data_sources/ds_update_priority sourcetype = trackme_alert_a

Alerts acknowledgment within the UI

Acknowledgement

When using built-in alerts, you can leverage alert acknowledgments within the UI to silent an active alert during a given period.



Actions for data source: firewall:pan:traffic

data_index: firewall

data_sourcetype: pan:traffic

lag event / lag ingestion: ([D+]HH:MM:SS) 00:01:41 / 0 sec

data_last_time_seen: 31/10/2020 23:00

data_last_ingest: 31/10/2020 23:00

data_max_lag_allowed: 60

data_monitored_state: enabled

data_monitoring_level: sourcetype

Click here to define a documentation reference / Click here to define tags

1.0 sec
PERC95 INGESTION LAG (sec or [D+]HH:MM:SS)

0.1 sec
AVG INGESTION LAG (sec or [D+]HH:MM:SS)

00
CURRENT EVEN

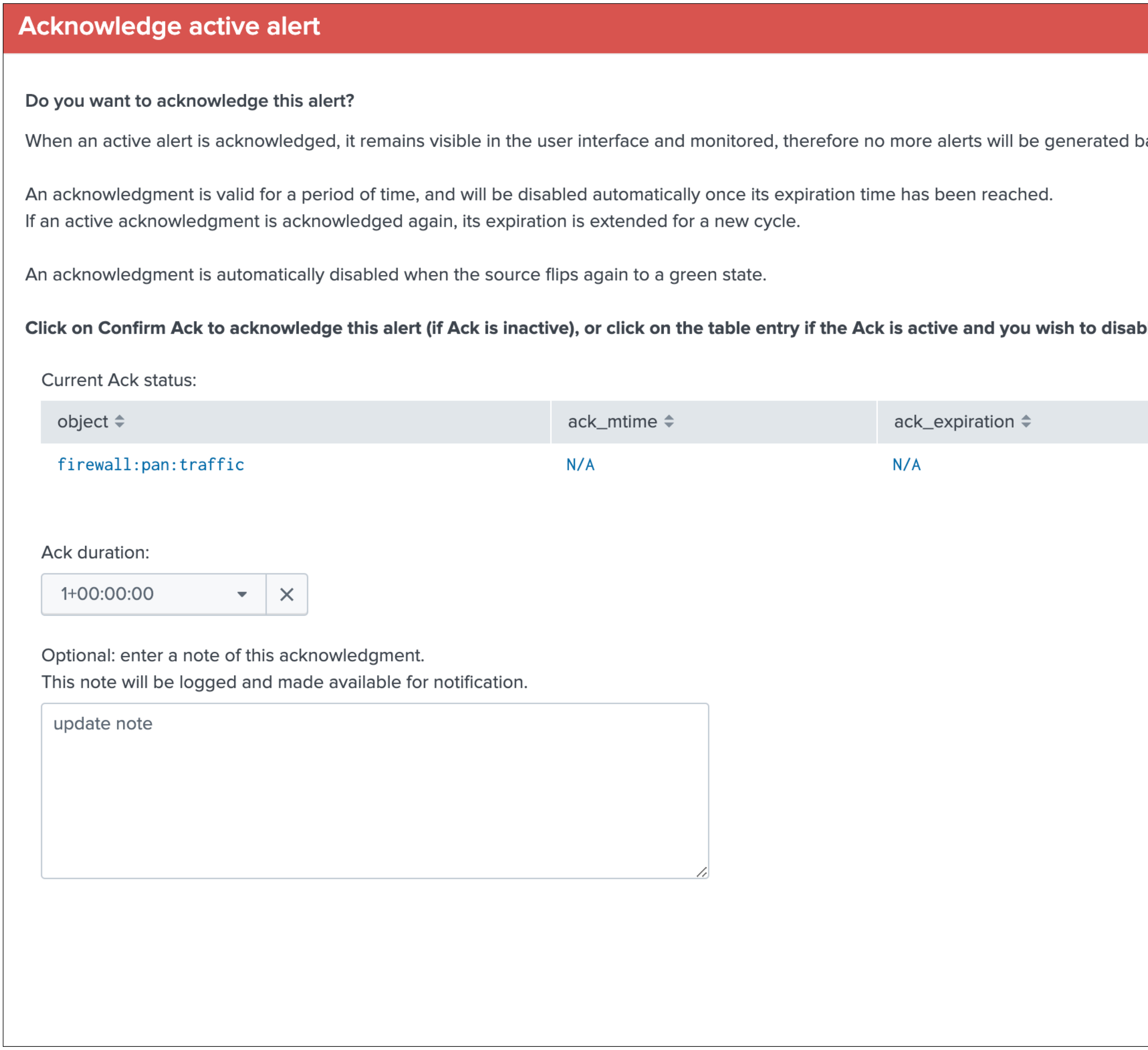
Acknowledgments provides a way to:

- Via the user interface, acknowledge an active alert
- Once acknowledged, the entity remains visible in the UI and monitored, but no more alerts will be generated during the time of the acknowledge
- An entity (data source, etc) that is in active alert and has been acknowledged will not generate any new alert for the next 24 hours by default, which value can be increased via the input selector
- Therefore, if the entity flips to a state green again, the acknowledge is automatically disabled
- If the entity flips later on to a red state, a new acknowledge should be created

Acknowledgment workflow:

- Via the UI, if the entity is in red state, the “Acknowledgment” button becomes active, otherwise it is inactive and cannot be clicked
- If the acknowledge is confirmed by the user, an active entry is created in the KVstore collection named “kv_trackme_alerts_ack”. (lookup definition trackme_alerts_ack)
- The default duration of acknowledges is define by the macro named “trackme_ack_default_duration”
- Every 5 minutes, the tracker scheduled report named “TrackMe - Ack tracker” verifies if an acknowledge has reached its expiration and will update its status if required
- The tracker as well verifies the current state of the entity, if the entity has flipped again to a green state, the acknowledge is disabled
- An acknowledge can be acknowledged again within the UI, which will extend its expiration for another cycle

Acknowledge for an active alert is inactive:



Acknowledge active alert

Do you want to acknowledge this alert?

When an active alert is acknowledged, it remains visible in the user interface and monitored, therefore no more alerts will be generated ba

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 acknowledgment is automatically disabled when the source flips again to a green state.

Click on Confirm Ack to acknowledge this alert (if Ack is inactive), or click on the table entry if the Ack is active and you wish to disab

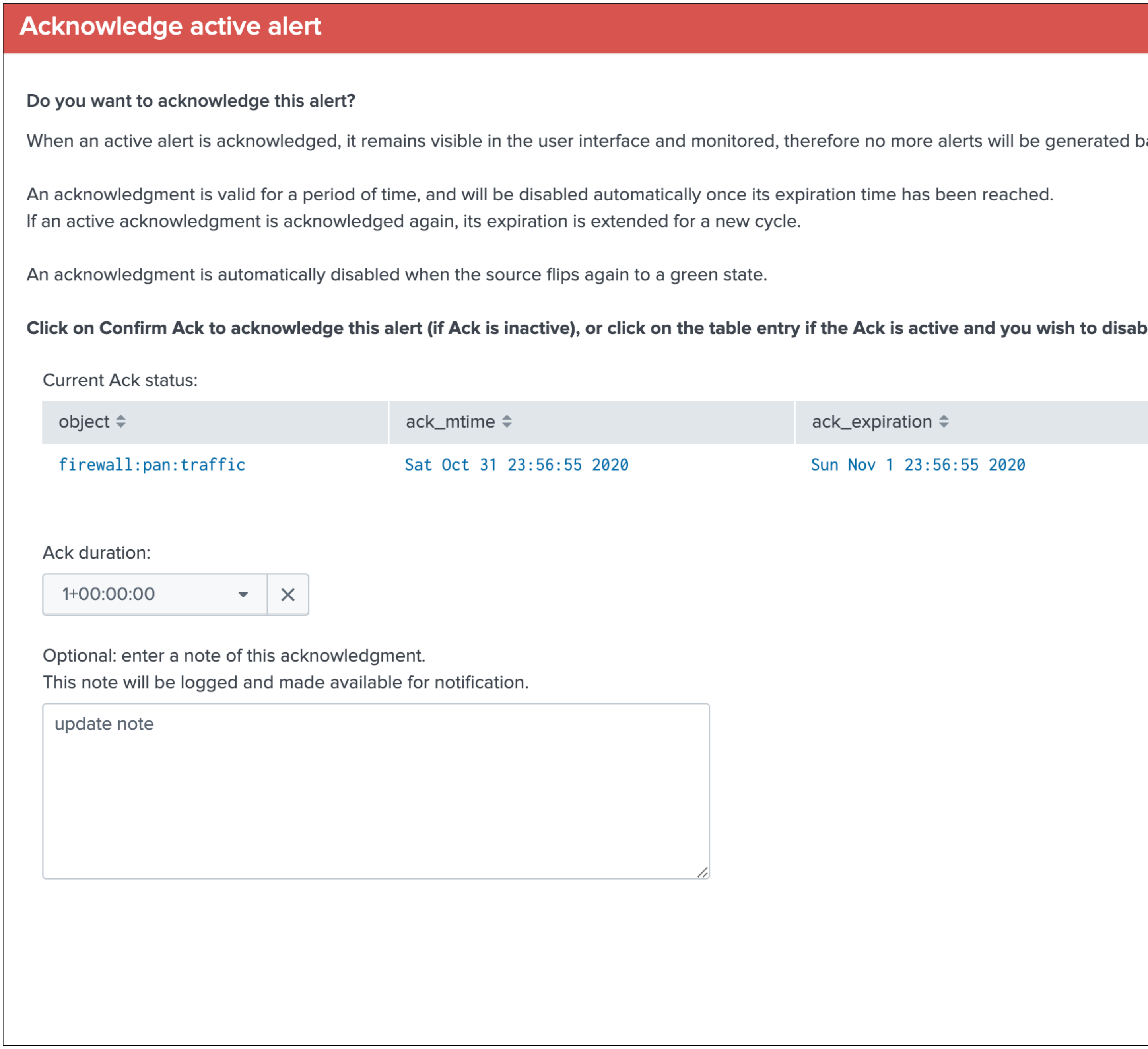
Current Ack status:

object	ack_mtime	ack_expiration
firewall:pan:traffic	N/A	N/A

Ack duration: 1+00:00:00

Optional: enter a note of this acknowledgment.
This note will be logged and made available for notification.

Acknowledge for an active alert is active:



Acknowledge active alert

Do you want to acknowledge this alert?

When an active alert is acknowledged, it remains visible in the user interface and monitored, therefore no more alerts will be generated ba

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 acknowledgment is automatically disabled when the source flips again to a green state.

Click on Confirm Ack to acknowledge this alert (if Ack is inactive), or click on the table entry if the Ack is active and you wish to disab

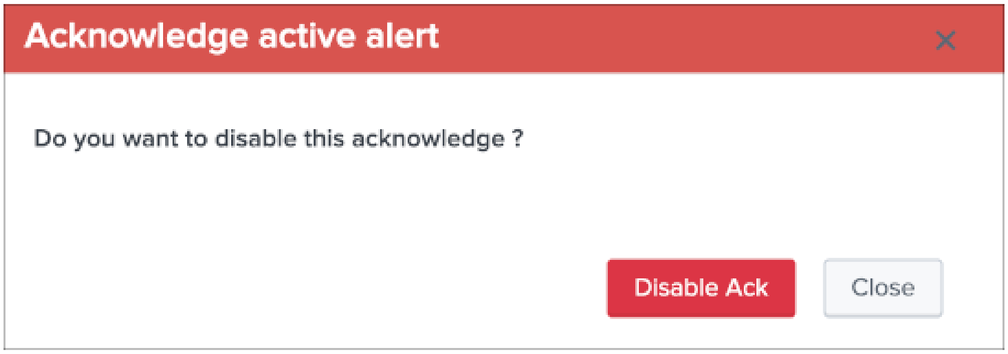
Current Ack status:

object	ack_mtime	ack_expiration
firewall:pan:traffic	Sat Oct 31 23:56:55 2020	Sun Nov 1 23:56:55 2020

Ack duration: 1+00:00:00

Optional: enter a note of this acknowledgment.
This note will be logged and made available for notification.

Once active, an acknowledge can be disabled on demand by clicking on the Ack table:

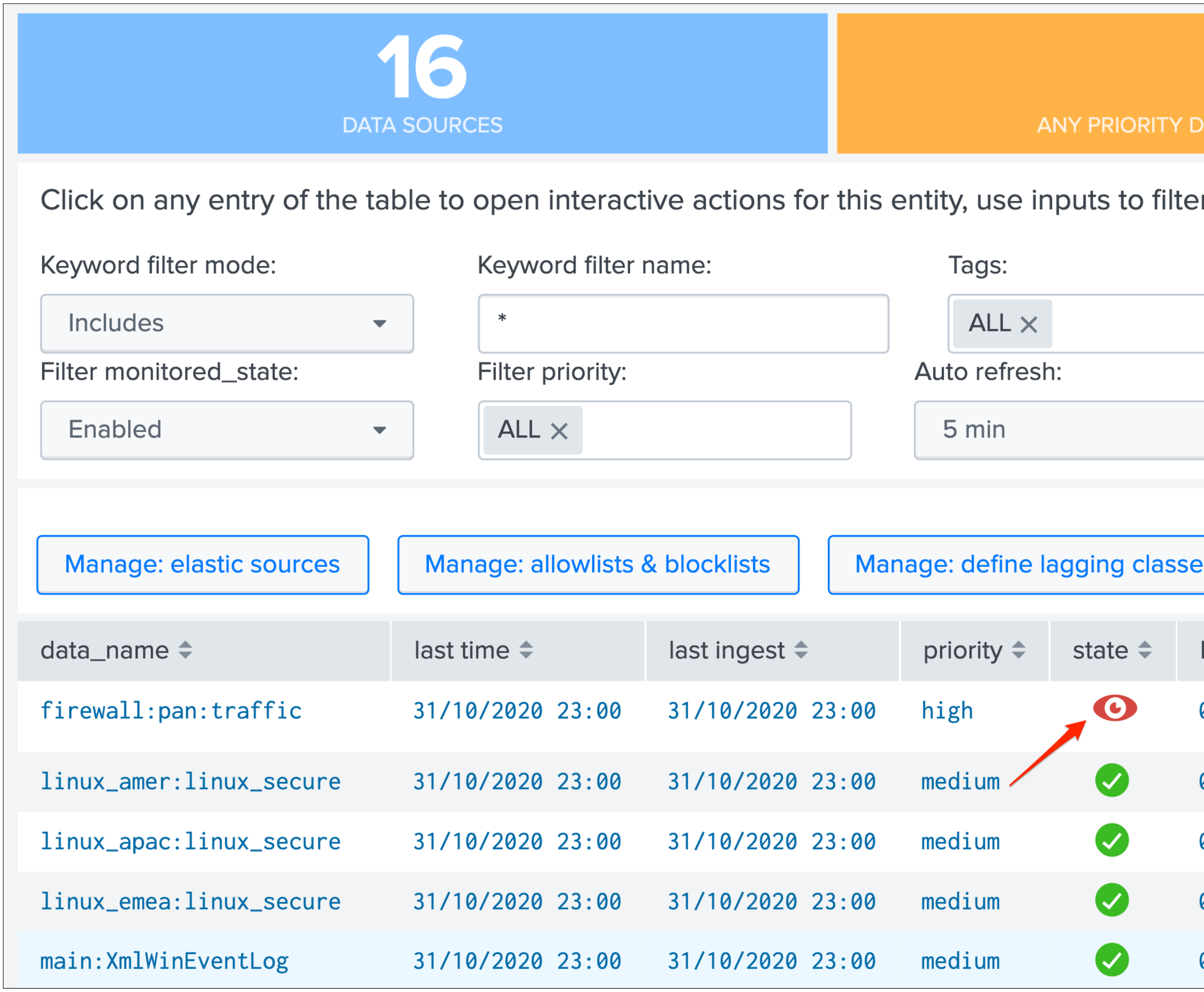


Acknowledge active alert

Do you want to disable this acknowledge ?

All acknowledgement related actions are recorded in the audit collection and report.

Tip: When an acknowledgment is active, a specific icon replaces the red state icon which easily indicates that an acknowledgement is currently active for that object.


16
DATA SOURCES
ANY PRIORITY D
Manage: elastic sources
Manage: allowlists & blocklists
Manage: define lagging classe

Click on any entry of the table to open interactive actions for this entity, use inputs to filte

Keyword filter mode: Includes

Keyword filter name: *

Tags: ALL ×

Filter monitored_state: Enabled

Filter priority: ALL ×

Auto refresh: 5 min

data_name	last time	last ingest	priority	state
firewall:pan:traffic	31/10/2020 23:00	31/10/2020 23:00	high	👁
linux_amer:linux_secure	31/10/2020 23:00	31/10/2020 23:00	medium	✓
linux_apac:linux_secure	31/10/2020 23:00	31/10/2020 23:00	medium	✓
linux_emea:linux_secure	31/10/2020 23:00	31/10/2020 23:00	medium	✓
main:XmlWinEventLog	31/10/2020 23:00	31/10/2020 23:00	medium	✓

3.1.8 Priority management

Priority levels

Priority

TrackMe has a notion of priority for each entity, you can view the priority value in any of the tables from the main interface, in the header when you click on a given entity, and you can modify it via the unified modification UI.

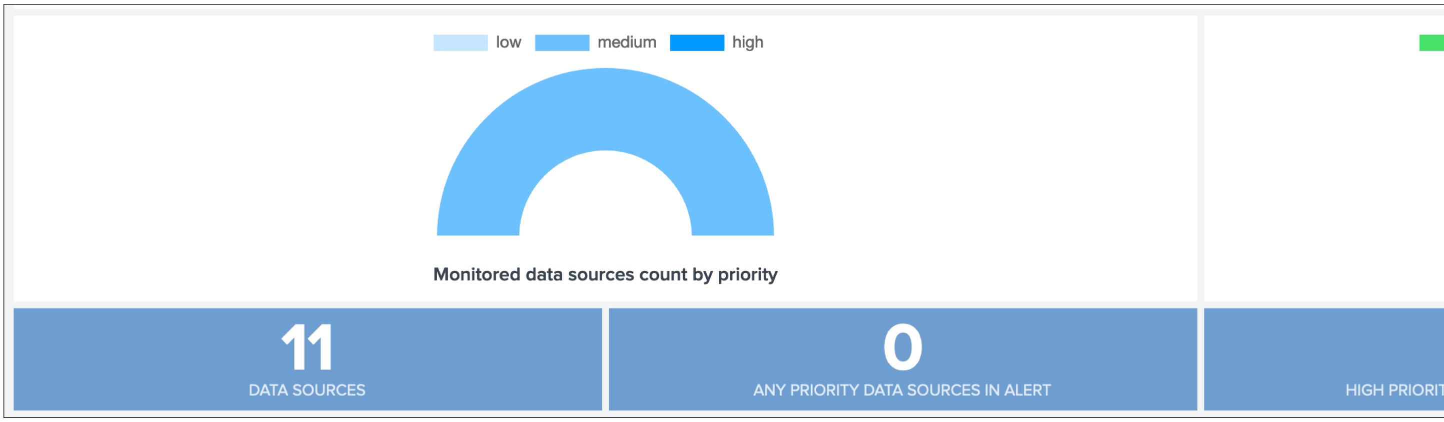
There 3 level of priorities that can be applied:

- `low`
- `medium`
- `high`

Priority feature

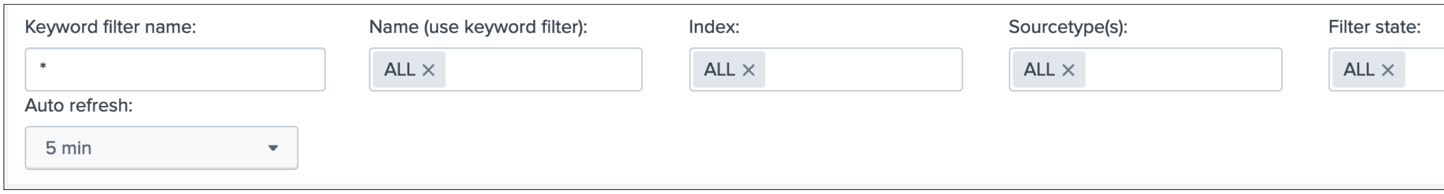
The purpose of the priority is to provide more granularity in the way you can manage entities.

First, the UI exposes the current status depending on the priority of the entities:



low
medium
high
Monitored data sources count by priority
11
DATA SOURCES
0
ANY PRIORITY DATA SOURCES IN ALERT
HIGH PRIORIT

As well, the priority can be easily filtered:



Keyword filter name: *

Name (use keyword filter): ALL ×

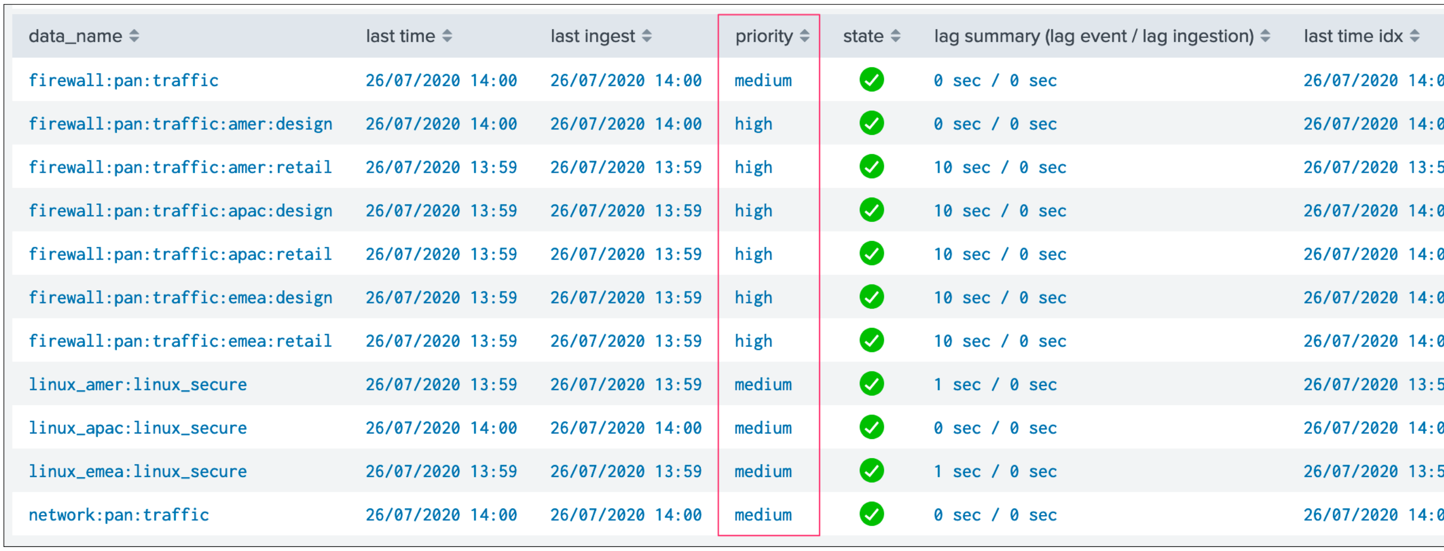
Index: ALL ×

Sourcetype(s): ALL ×

Filter state: ALL ×

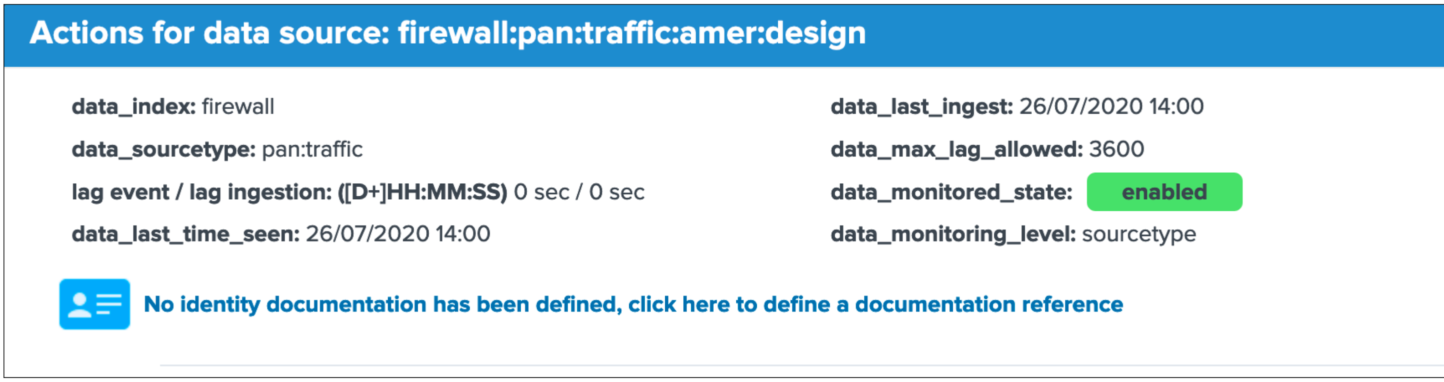
Auto refresh: 5 min

The priority is visible in the table too:



data_name ⇕	last time ⇕	last ingest ⇕	priority ⇕	state ⇕	lag summary (lag event / lag ingestion) ⇕	last time idx ⇕
firewall:pan:traffic	26/07/2020 14:00	26/07/2020 14:00	medium	✓	0 sec / 0 sec	26/07/2020 14:0
firewall:pan:traffic:amer:design	26/07/2020 14:00	26/07/2020 14:00	high	✓	0 sec / 0 sec	26/07/2020 14:0
firewall:pan:traffic:amer:retail	26/07/2020 13:59	26/07/2020 13:59	high	✓	10 sec / 0 sec	26/07/2020 13:5
firewall:pan:traffic:apac:design	26/07/2020 13:59	26/07/2020 13:59	high	✓	10 sec / 0 sec	26/07/2020 14:0
firewall:pan:traffic:apac:retail	26/07/2020 13:59	26/07/2020 13:59	high	✓	10 sec / 0 sec	26/07/2020 14:0
firewall:pan:traffic:emea:design	26/07/2020 13:59	26/07/2020 13:59	high	✓	10 sec / 0 sec	26/07/2020 14:0
firewall:pan:traffic:emea:retail	26/07/2020 13:59	26/07/2020 13:59	high	✓	10 sec / 0 sec	26/07/2020 14:0
linux_amer:linux_secure	26/07/2020 13:59	26/07/2020 13:59	medium	✓	1 sec / 0 sec	26/07/2020 13:5
linux_apac:linux_secure	26/07/2020 14:00	26/07/2020 14:00	medium	✓	0 sec / 0 sec	26/07/2020 14:0
linux_emea:linux_secure	26/07/2020 13:59	26/07/2020 13:59	medium	✓	1 sec / 0 sec	26/07/2020 13:5
network:pan:traffic	26/07/2020 14:00	26/07/2020 14:00	medium	✓	0 sec / 0 sec	26/07/2020 14:0

When clicking on an entity, the priority is shown on top with a blue colour scheme that starts from light blue for low, blue for medium and darker blue for high:



Actions for data source: firewall:pan:traffic:amer:design

data_index: firewall

data_sourcetype: pan:traffic

lag event / lag ingestion: ([D+]HH:MM:SS) 0 sec / 0 sec

data_last_time_seen: 26/07/2020 14:00

data_last_ingest: 26/07/2020 14:00

data_max_lag_allowed: 3600

data_monitored_state: enabled

data_monitoring_level: sourcetype

No identity documentation has been defined, click here to define a documentation reference

The default priority assigned is "medium" and managed by the following macro:

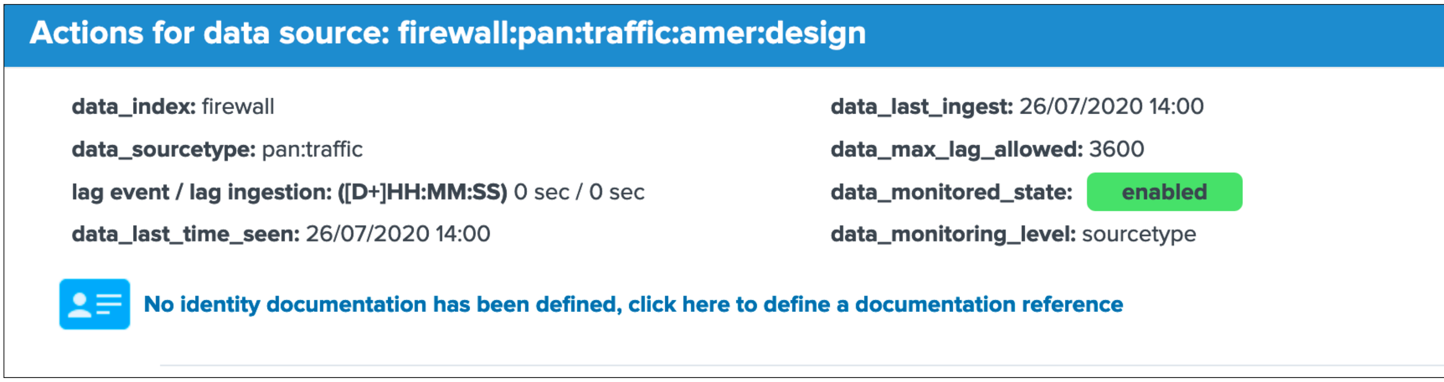
- `trackme_default_priority`

Out of the box alerts filter automatically on certain types of priorities, by default `medium` and `high`, which is managed by the following macro:

- `trackme_alerts_priority`

Modify the priority

The priority of an entity can be modified in the UI via the unified modification window:



Actions for data source: firewall:pan:traffic:amer:design

data_index: firewall

data_sourcetype: pan:traffic

lag event / lag ingestion: ([D+]HH:MM:SS) 0 sec / 0 sec

data_last_time_seen: 26/07/2020 14:00

data_last_ingest: 26/07/2020 14:00

data_max_lag_allowed: 3600

data_monitored_state: enabled

data_monitoring_level: sourcetype

No identity documentation has been defined, click here to define a documentation reference

Bulk update the priority

If you wish or need to bulk update or maintain the priority of entities such as the data hosts against a third party lookup, such a thing could be easily performed in a single search.

Example:

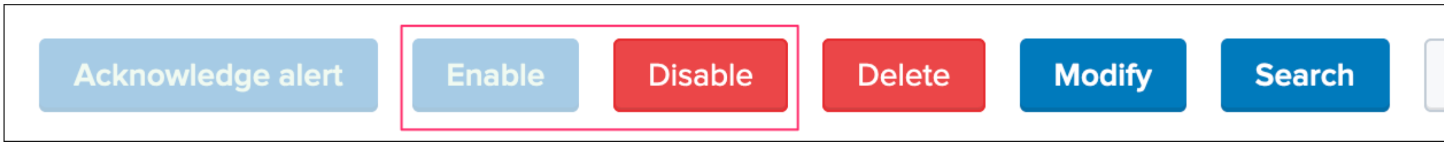
```
| inputlookup trackme_host_monitoring | eval key=_key
| lookup <the third party lookup> data_host as host OUTPUT priority as new_priority |␣
↪eval priority=if(isnotnull(new_priority), new_priority, priority)
| outputlookup trackme_host_monitoring append=t key_field=key
```

This search above for instance would bulk update all matched entities.

3.1.9 Monitored state (enable / disable buttons)

Monitored state

- Entities have a so called “monitored state”, which can be `enabled` or `disabled`.
- When disabled, an entity disappears from TrackMe UI, will stop being considered for any alert or data generation purposes.



If an entity is set to `disabled`, it will not appear anymore in the main screens, will not be part of any alert results, and no more metrics will be collected for it.

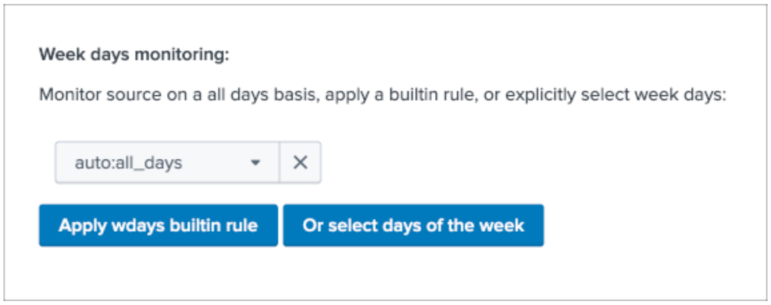
The purpose of this flag is to allow disabling an entity that is discovered automatically because the scope of the data discovery (allowlist / blocklist) allow it.

3.1.10 Week days monitoring

Week days monitoring

You can modify the rules for days of week monitoring, which means specifying for which days of the week an entity will be monitored actively.

Week days monitoring rules apply to event data only (data sources and hosts)



Week days monitoring:

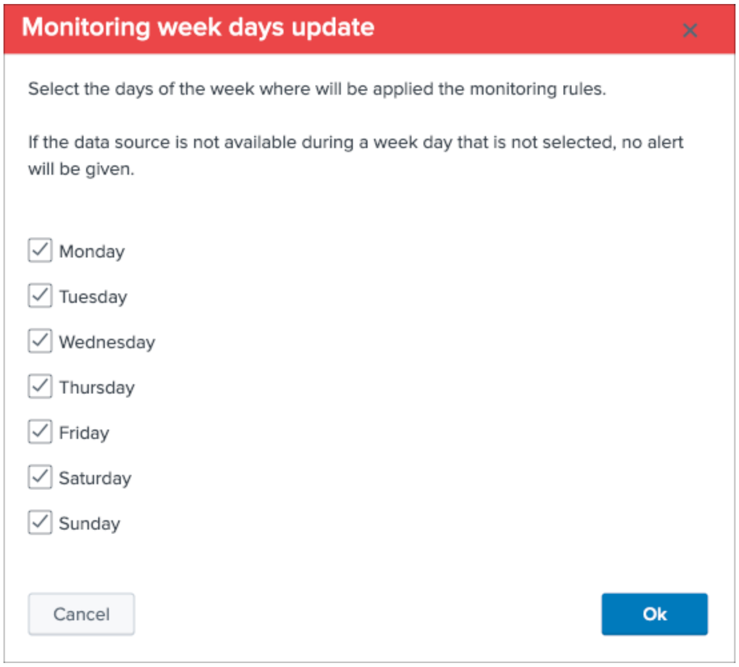
Monitor source on a all days basis, apply a builtin rule, or explicitly select week days:

auto:all_days

Several built-in rules are available:

- manual:all_days
- manual:monday-to-friday
- manual:monday-to-saturday

Or you can select explicitly which days of the week:



Monitoring week days update

Select the days of the week where will be applied the monitoring rules.

If the data source is not available during a week day that is not selected, no alert will be given.

☑ Monday

☑ Tuesday

☑ Wednesday

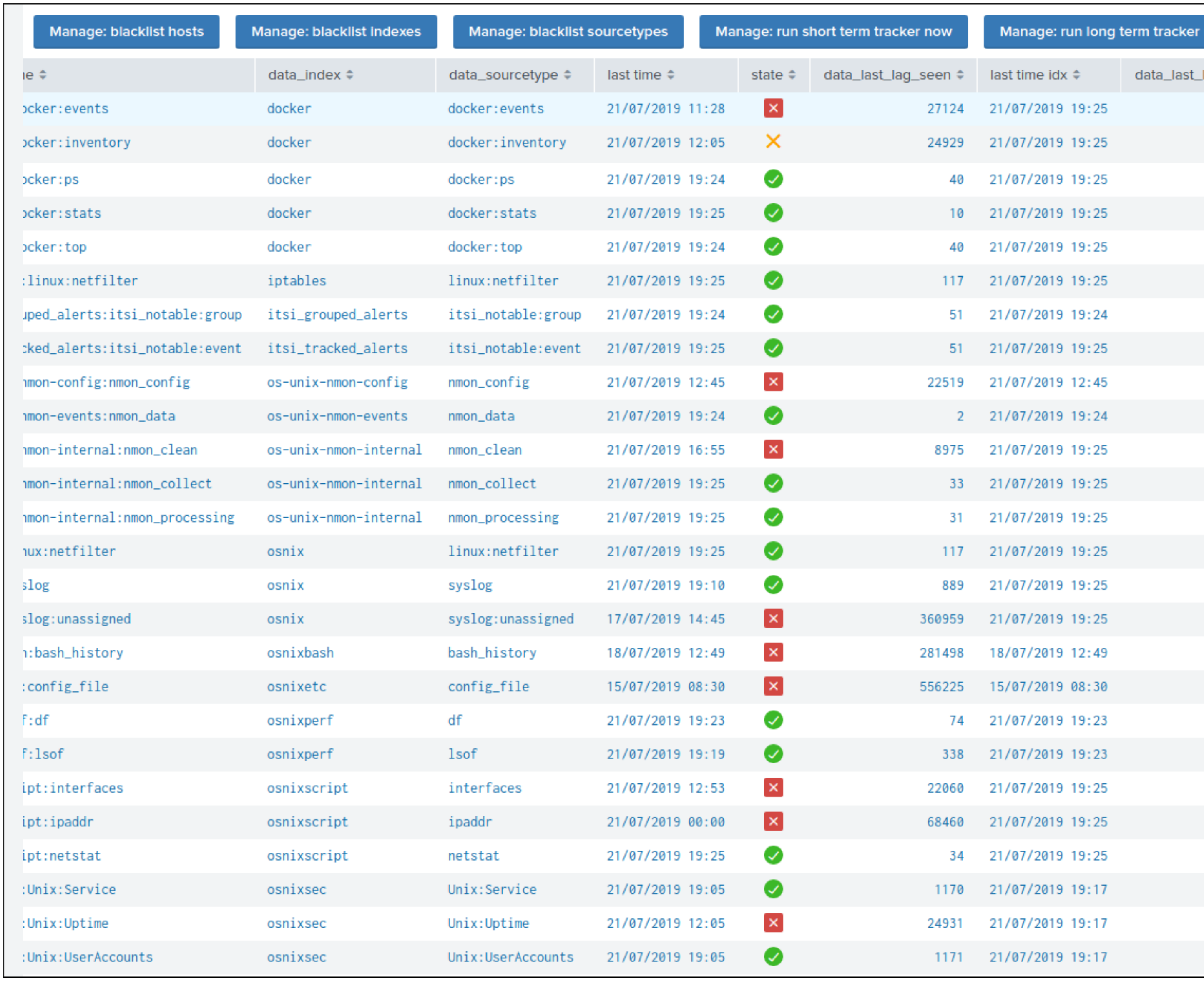
☑ Thursday

☑ Friday

☑ Saturday

☑ Sunday

Which is visible in the table:


Manage: blacklist hosts
Manage: blacklist indexes
Manage: blacklist sourcetypes
Manage: run short term tracker now
Manage: run long term tracker

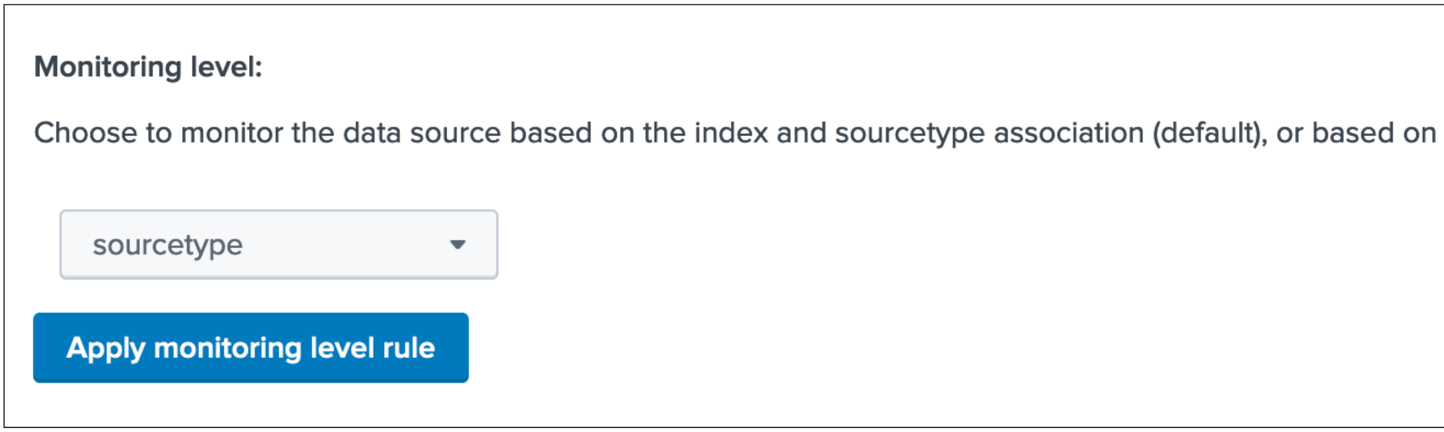
ıe ⇕	data_index ⇕	data_sourcetype ⇕	last time ⇕	state ⇕	data_last_lag_seen ⇕	last time idx ⇕	data_last_
ɔcker:events	docker	docker:events	21/07/2019 11:28	☒	27124	21/07/2019 19:25	
ɔcker:inventory	docker	docker:inventory	21/07/2019 12:05	✗	24929	21/07/2019 19:25	
ɔcker:ps	docker	docker:ps	21/07/2019 19:24	✓	40	21/07/2019 19:25	
ɔcker:stats	docker	docker:stats	21/07/2019 19:25	✓	10	21/07/2019 19:25	
ɔcker:top	docker	docker:top	21/07/2019 19:24	✓	40	21/07/2019 19:25	
:linux:netfilter	iptables	linux:netfilter	21/07/2019 19:25	✓	117	21/07/2019 19:25	
ɹped_alerts:itsi_notable:group	itsi_grouped_alerts	itsi_notable:group	21/07/2019 19:24	✓	51	21/07/2019 19:24	
:ked_alerts:itsi_notable:event	itsi_tracked_alerts	itsi_notable:event	21/07/2019 19:25	✓	51	21/07/2019 19:25	
ımon-config:nmon_config	os-unix-nmon-config	nmon_config	21/07/2019 12:45	☒	22519	21/07/2019 12:45	
ımon-events:nmon_data	os-unix-nmon-events	nmon_data	21/07/2019 19:24	✓	2	21/07/2019 19:24	
ımon-internal:nmon_clean	os-unix-nmon-internal	nmon_clean	21/07/2019 16:55	☒	8975	21/07/2019 19:25	
ımon-internal:nmon_collect	os-unix-nmon-internal	nmon_collect	21/07/2019 19:25	✓	33	21/07/2019 19:25	
ımon-internal:nmon_processing	os-unix-nmon-internal	nmon_processing	21/07/2019 19:25	✓	31	21/07/2019 19:25	
ıux:netfilter	osnix	linux:netfilter	21/07/2019 19:25	✓	117	21/07/2019 19:25	
slog	osnix	syslog	21/07/2019 19:10	✓	889	21/07/2019 19:25	
slog:unassigned	osnix	syslog:unassigned	17/07/2019 14:45	☒	360959	21/07/2019 19:25	
ı:bash_history	osnixbash	bash_history	18/07/2019 12:49	☒	281498	18/07/2019 12:49	
:config_file	osnixetc	config_file	15/07/2019 08:30	☒	556225	15/07/2019 08:30	
f:df	osnixperf	df	21/07/2019 19:23	✓	74	21/07/2019 19:23	
f:lsof	osnixperf	lsof	21/07/2019 19:19	✓	338	21/07/2019 19:23	
ipt:interfaces	osnixscript	interfaces	21/07/2019 12:53	☒	22060	21/07/2019 19:25	
ipt:ipaddr	osnixscript	ipaddr	21/07/2019 00:00	☒	68460	21/07/2019 19:25	
ipt:netstat	osnixscript	netstat	21/07/2019 19:25	✓	34	21/07/2019 19:25	
:Unix:Service	osnixsec	Unix:Service	21/07/2019 19:05	✓	1170	21/07/2019 19:17	
:Unix:Uptime	osnixsec	Unix:Uptime	21/07/2019 12:05	☒	24931	21/07/2019 19:17	
:Unix:UserAccounts	osnixsec	Unix:UserAccounts	21/07/2019 19:05	✓	1171	21/07/2019 19:17	

3.1.11 Monitoring level

For data sources, you can define if the monitoring applies on the sourcetype level (default) or the index level:

Monitoring level

- The monitoring level can be defined for a data source to either the `sourcetype` level (default) or `index` level.
- When defined against the index, the data source will be considered live until no more data sources generate data in the enitre index hosting the data source.



Monitoring level:

Choose to monitor the data source based on the index and sourcetype association (default), or based on

sourcetype

Feature behaviour:

- When the monitoring of the data source applies on the sourcetype level, if that combination of index / sourcetype data does not respect the monitoring rule, it will trigger.
- When the monitoring of the data source applies on the index level, we take in consideration what the latest data available is in this index, no matter what the sourcetype is.

This option is useful for instance if you have multiple sourcetypes in a single index, however some of these sourcetypes are not critical enough to justify raising any alert on their own but these need to remain visible in Trackme for context and troubleshooting purposes.

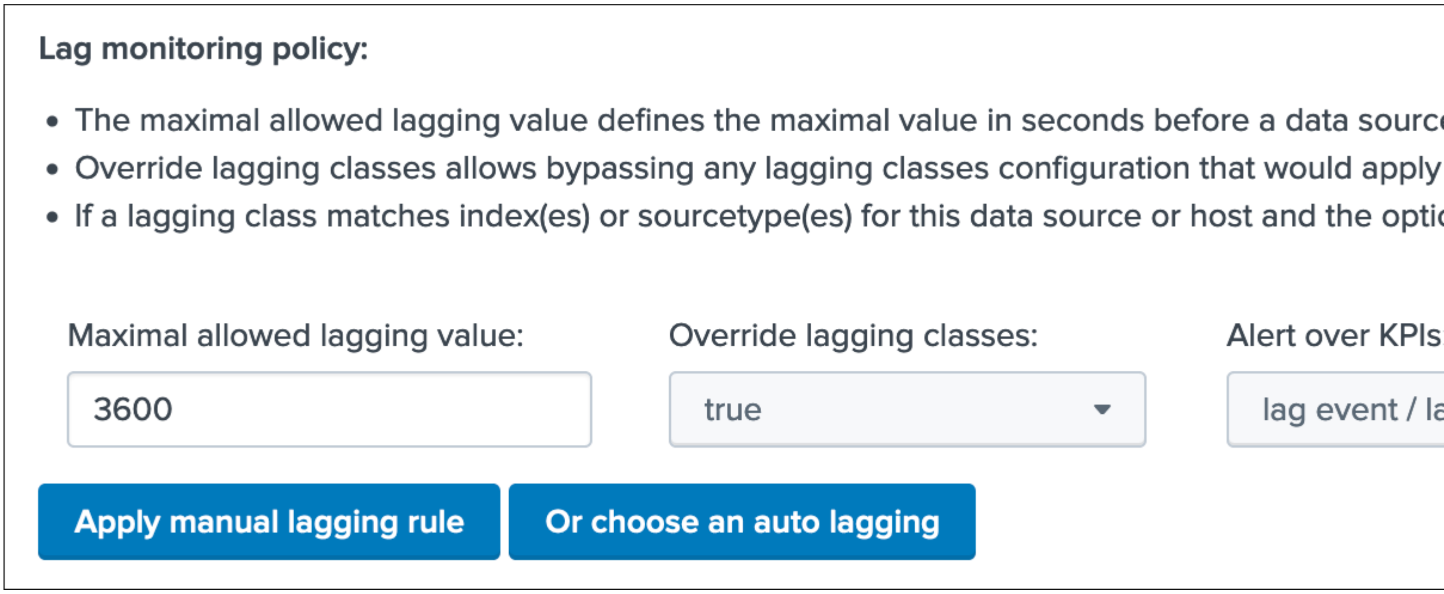
For example:

- An index contains the sourcetype “mybusiness:critical” and the sourcetype “mybusiness:informational”
- “mybusiness:critical” is set to sourcetype level
- “mybusiness:informational” is set to index level
- “mybusiness:critical” will generate an alert if lagging conditions are not met for that data source
- “mybusiness:informational” will generate an alert **only** if “mybusiness:critical” monitoring conditions are not met either
- The fact the informational data is not available in the same time than “mybusiness:critical” is a useful information that lets the engineer know that the problem is global for that specific data flow
- Using the index monitoring level for “mybusiness:informational” allows it to be visible in TrackMe without generating alerts on its own as long as “mybusiness:critical” meets the monitoring conditions

3.1.12 Maximal lagging value

Lagging value

The maximal lagging value defines the threshold to be used for alerting when a given entity goes beyond a certain value in seconds, against both lagging KPIs, or since the version 1.2.19 you can choose between different options.



Lag monitoring policy:

- The maximal allowed lagging value defines the maximal value in seconds before a data sourc
- Override lagging classes allows bypassing any lagging classes configuration that would apply
- If a lagging class matches index(es) or sourcetype(es) for this data source or host and the optio

Maximal allowed lagging value: 3600

Override lagging classes: true

Alert over KPIs: lag event / la

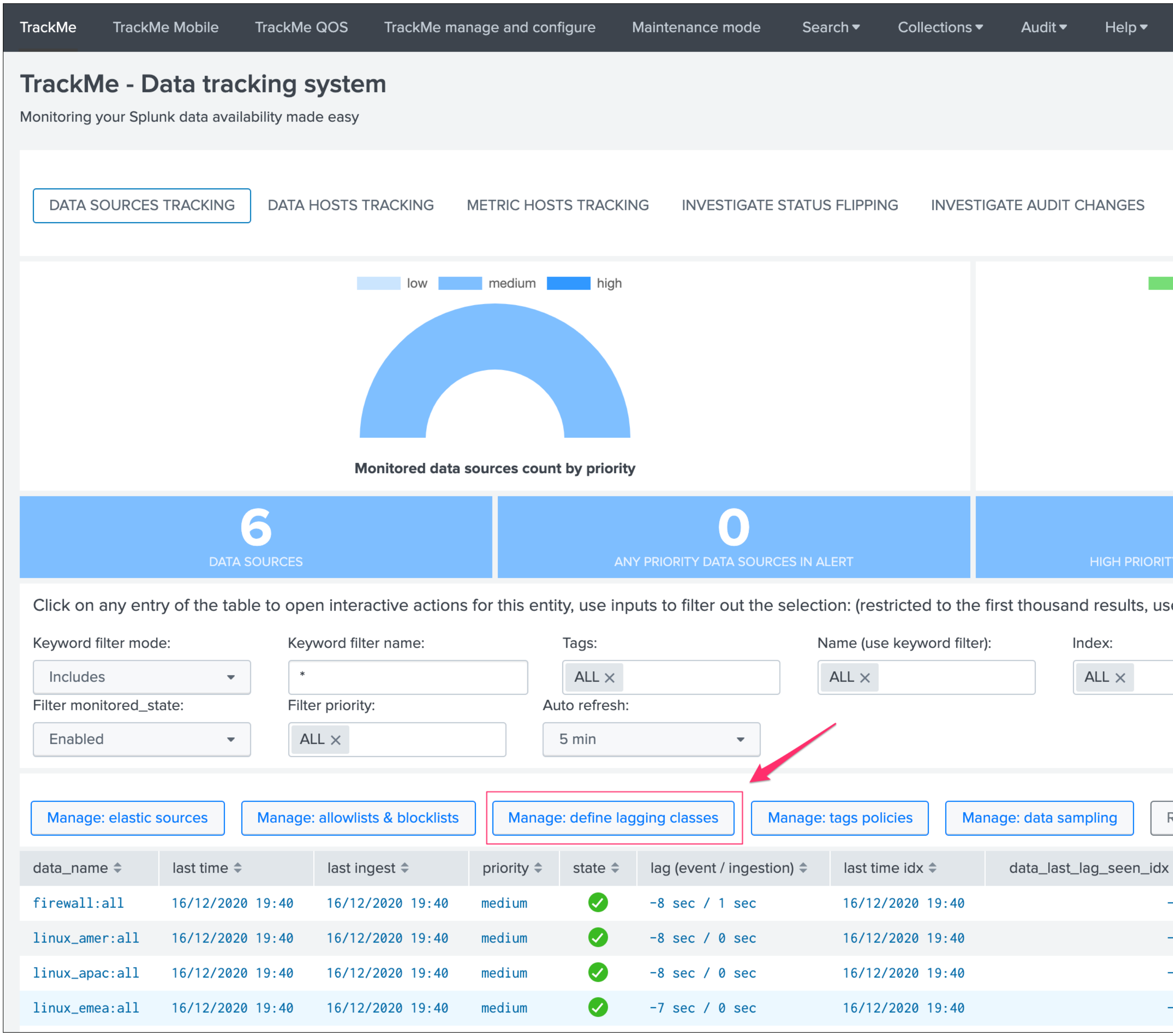
This topic is covered in details in first steps guide *Main navigation tabs* and *Unified update interface*.

3.1.13 Lagging classes

Lagging classes

- The Lagging classes feature provides capabilities to manage and configure the maximal lagging values allowed in a centralised and automated fashion, based on different factors.
- A lagging class can be configured based on index names, sourcetype values and the entities priority level.
- Lagging classes apply on data sources and hosts, and classes can be created matching either both types of object, data sources or data hosts only.

Lagging classes are configurable in the main TrackMe UI:



TrackMe - Data tracking system

Monitoring your Splunk data availability made easy

Monitored data sources count by priority

6 DATA SOURCES

0 ANY PRIORITY DATA SOURCES IN ALERT

HIGH PRIORIT

Click on any entry of the table to open interactive actions for this entity, use inputs to filter out the selection: (restricted to the first thousand results, us

Keyword filter mode: Includes

Keyword filter name: *

Tags: ALL ×

Name (use keyword filter): ALL ×

Index: ALL ×

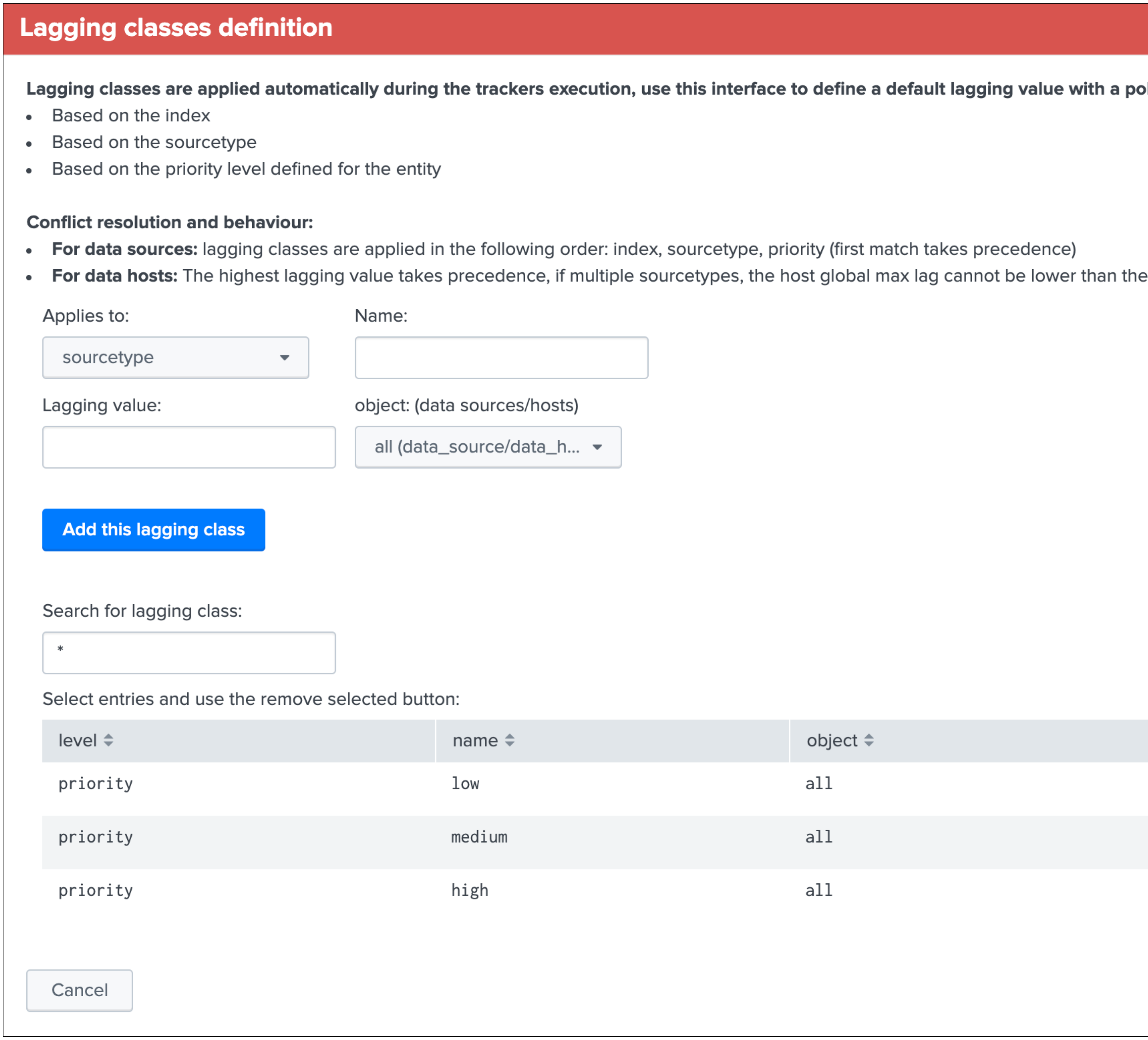
Filter monitored_state: Enabled

Filter priority: ALL ×

Auto refresh: 5 min

data_name	last time	last ingest	priority	state	lag (event / ingestion)	last time idx	data_last_lag_seen_idx
firewall:all	16/12/2020 19:40	16/12/2020 19:40	medium	✓	-8 sec / 1 sec	16/12/2020 19:40	-
linux_amer:all	16/12/2020 19:40	16/12/2020 19:40	medium	✓	-8 sec / 0 sec	16/12/2020 19:40	-
linux_apac:all	16/12/2020 19:40	16/12/2020 19:40	medium	✓	-8 sec / 0 sec	16/12/2020 19:40	-
linux_emea:all	16/12/2020 19:40	16/12/2020 19:40	medium	✓	-7 sec / 0 sec	16/12/2020 19:40	-

Which lets you access to the following UI:



Lagging classes definition

Lagging classes are applied automatically during the trackers execution, use this interface to define a default lagging value with a po

- Based on the index
- Based on the sourcetype
- Based on the priority level defined for the entity

Conflict resolution and behaviour:

- **For data sources:** lagging classes are applied in the following order: index, sourcetype, priority (first match takes precedence)
- **For data hosts:** The highest lagging value takes precedence, if multiple sourcetypes, the host global max lag cannot be lower than the

Applies to: sourcetype

Name:

Lagging value:

object: (data sources/hosts) all (data_source/data_h...

Search for lagging class: *

Select entries and use the remove selected button:

level	name	object
priority	low	all
priority	medium	all
priority	high	all

Lagging classes are controlled by the following main rules:

- For data sources: lagging classes are applied in the following order: index, sourcetype, priority (first match takes precedence)
- For data hosts: The highest lagging value takes precedence, if multiple sourcetypes, the host global max lag cannot be lower than the highest value between all sourcetypes

Lagging classes override

When a lagging class is defined and is matched for a data source or a data host, you can as well override this policy based lagging value by defining a lagging value on the object within the UI and enabling the override option.

Lagging classes behaviour for data sources

When a lagging class is configured and defined to apply on data sources (or all), the tracker reports retrieve the lagging class information via enrichment (lookup) and proceed to different conditional operations.

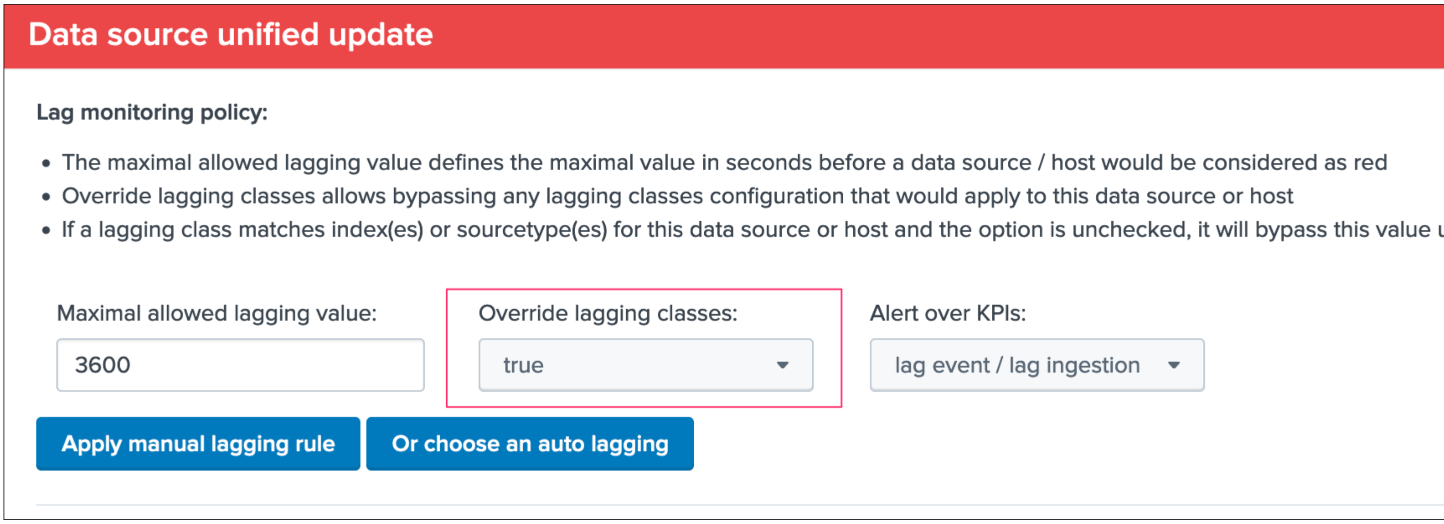
These operations in the case of data sources are proceeded in a specific order as follows:

- 1. index
- 2. sourcetype
- 3. priority

The first operation that matches a value takes precedence over any other value.

For instance, if a lagging class matches the index "network", every data source linked to this index will retrieve the maximal lagging value from the lagging class no matters if any other lagging classes would have matched. (priority for example)

As well, it is possible to override this behaviour and manually control the maximal lagging value for a given data source independently from any lagging class matching, this is configurable by modifying the data source configuration: (Modify button)



Data source unified update

Lag monitoring policy:

- The maximal allowed lagging value defines the maximal value in seconds before a data source / host would be considered as red
- Override lagging classes allows bypassing any lagging classes configuration that would apply to this data source or host
- If a lagging class matches index(es) or sourcetype(es) for this data source or host and the option is unchecked, it will bypass this value u

Maximal allowed lagging value: 3600

Override lagging classes: true

Alert over KPIs: lag event / lag ingestion

Lagging classes behaviour for data hosts

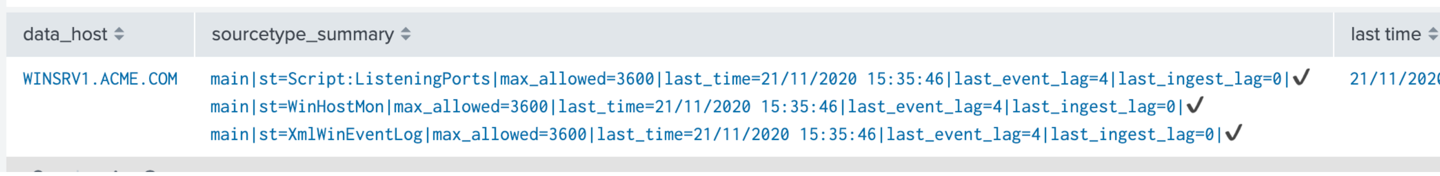
By definition, the data hosts monitoring is a more complex task which involves for a given entity (host) the monitoring of potentially numbers of sub-entities (sourcetypes).

Main rules for data hosts lagging classes:

- At first, TrackMe attempts to perform lagging class matching per host and per sourcetype
- For a given sourcetype, the higest lagging value between index based policies and sourcetype based policies is recorded per sourcetype
- Finally, the highest lagging value between all sourcetypes for the host is saved as the general maximal lagging value for the host

Let's take the following example:

- host: winsrv1.acme.com
- 3 sourcetypes indexed: XmlWinEventLog, Script:ListeningPorts, WinHostMon

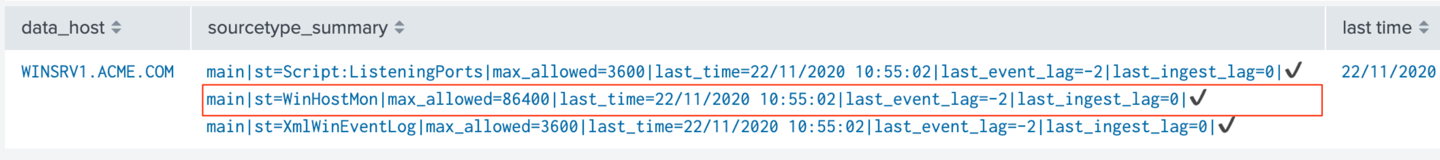


data_host	sourcetype_summary	last time
WINSRV1.ACME.COM	main|st=Script:ListeningPorts|max_allowed=3600|last_time=21/11/2020 15:35:46|last_event_lag=4|last_ingest_lag=0|✓ main|st=WinHostMon|max_allowed=3600|last_time=21/11/2020 15:35:46|last_event_lag=4|last_ingest_lag=0|✓ main|st=XmlWinEventLog|max_allowed=3600|last_time=21/11/2020 15:35:46|last_event_lag=4|last_ingest_lag=0|✓	21/11/202

by default, TrackMe applies a 3600 max lagging value per sourcetype and for the overall host

- A new lagging class is created to match the sourcetype `WinHostMon` to define a max lagging value of 86400 seconds

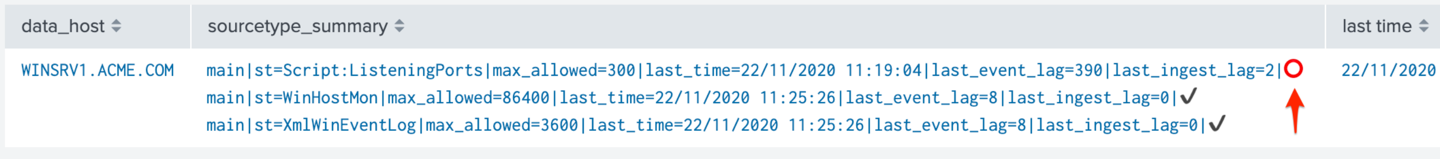
Once the tracker report has been executed, the sourcetype maximal laggging value is defined accordingly, and the overall max lagging value of the host is set to the highest value between all sourcetypes monitored:



data_host	sourcetype_summary	last time
WINSRV1.ACME.COM	main|st=Script:ListeningPorts|max_allowed=3600|last_time=22/11/2020 10:55:02|last_event_lag=-2|last_ingest_lag=0|✓ main|st=WinHostMon|max_allowed=86400|last_time=22/11/2020 10:55:02|last_event_lag=-2|last_ingest_lag=0|✓ main|st=XmlWinEventLog|max_allowed=3600|last_time=22/11/2020 10:55:02|last_event_lag=-2|last_ingest_lag=0|✓	22/11/2020

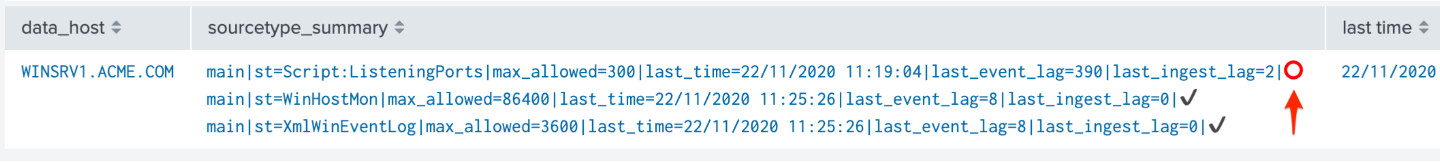
- Now let's create a new lagging class matching the sourcetype `Script:ListeningPorts` with a short max lagging class of 300 seconds
- The provider is stopped for the demonstration purposes
- After 5 minutes, the sourcetype appears in anomaly
- If the data hosts alerting policy is defined to track per sourcetype, the host turns red
- If the data hosts alerting policy is defined to track per host, the host remains green until none of the sourcetype have been indexing for at least the overall max lag of the host

Alerting policy track per sourcetype:



data_host	sourcetype_summary	last time
WINSRV1.ACME.COM	main|st=Script:ListeningPorts|max_allowed=300|last_time=22/11/2020 11:19:04|last_event_lag=390|last_ingest_lag=2|○ main|st=WinHostMon|max_allowed=86400|last_time=22/11/2020 11:25:26|last_event_lag=8|last_ingest_lag=0|✓ main|st=XmlWinEventLog|max_allowed=3600|last_time=22/11/2020 11:25:26|last_event_lag=8|last_ingest_lag=0|✓	22/11/2020

Alerting policy track per host:



data_host	sourcetype_summary	last time
WINSRV1.ACME.COM	main|st=Script:ListeningPorts|max_allowed=300|last_time=22/11/2020 11:19:04|last_event_lag=390|last_ingest_lag=2|○ main|st=WinHostMon|max_allowed=86400|last_time=22/11/2020 11:25:26|last_event_lag=8|last_ingest_lag=0|✓ main|st=XmlWinEventLog|max_allowed=3600|last_time=22/11/2020 11:25:26|last_event_lag=8|last_ingest_lag=0|✓	22/11/2020

Lagging classes override

- TrackMe will use the higher value between all sourcetypes to define the max overall lagging value of the host
- This value can as well be overriden on a per host basis in the host modification screen, but should ideally be controlled by automated policies based on indexes or sourcetypes

Lagging classes example based on the priority

A common use case, especially for data hosts, is to define lagging values based on the priority.

Let's assume the following use case:

- if the priority is `low`, assign a lagging value of `432000` seconds (5 days)
- if the priority is `medium`, assign a lagging value of `86400` seconds (1 day)
- if the priority is `high`, assign a lagging value of `14400` seconds (4 hours)

Updating priority from third party sources

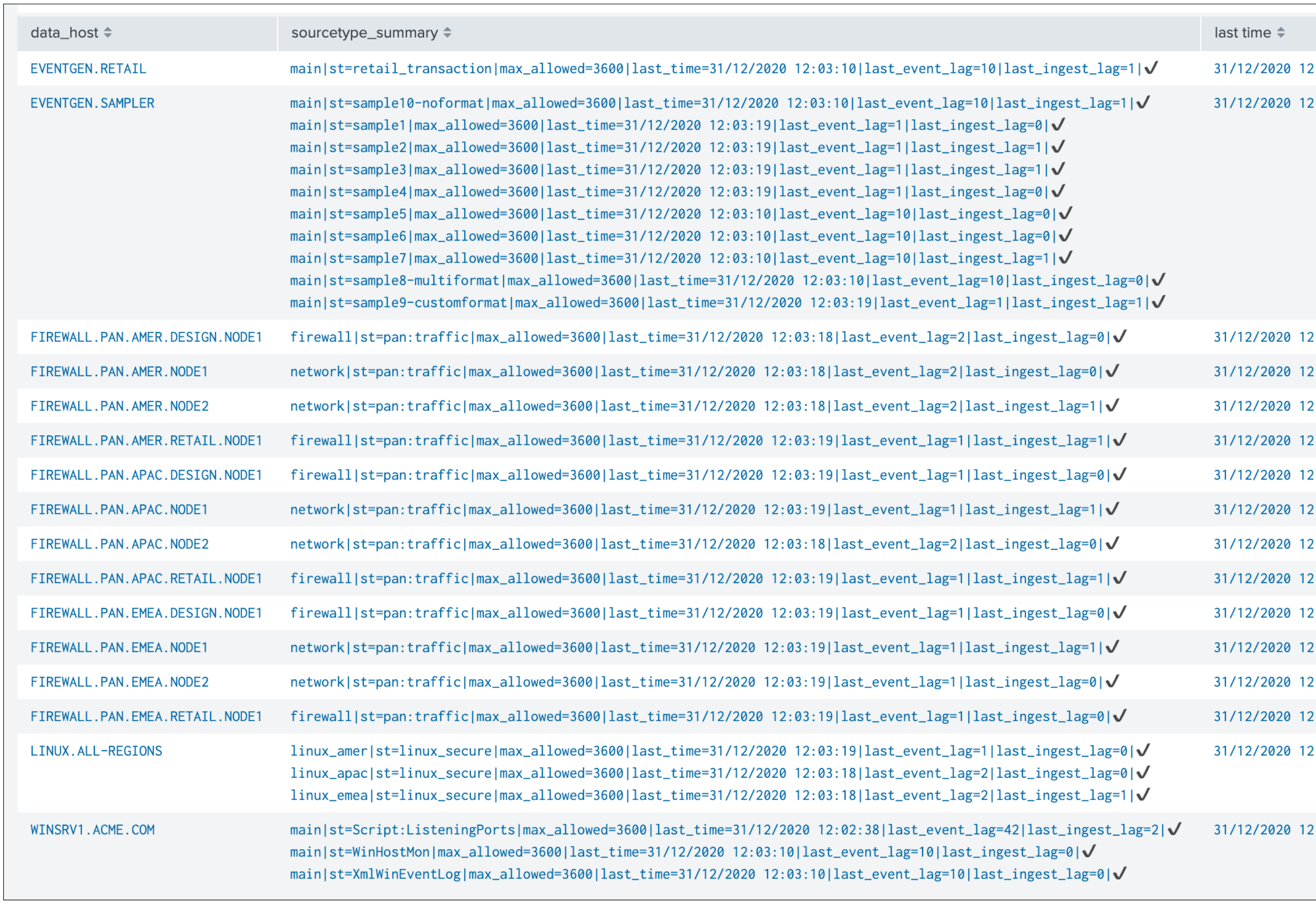
- In KVstore context, it is easy enough to update and maintain specific information such as the priority using third party sources such as any CMDB data that is available to Splunk
- To achieve this, you can simply create your own custom scheduled report that loads the TrackMe collection, enriches with the third party source, and finally updates the values in the TrackMe collection
- The priority value is preserved automatically when the tracker run, as soon as the value has been updated between low / medium / high, it will be preserved

example: assuming your CMDB data is available in the lookup acme_assets_cmdb:

```
| inputlookup trackme_host_monitoring | eval key=_key
| lookup acme_assets_cmdb.csv nt_host as data_host OUTPUTNEW priority as cmdb_priority
| eval priority=if(isnotnull(cmdb_priority), cmdb_priority, priority)
| outputlookup append=t key_field=key trackme_host_monitoring
```

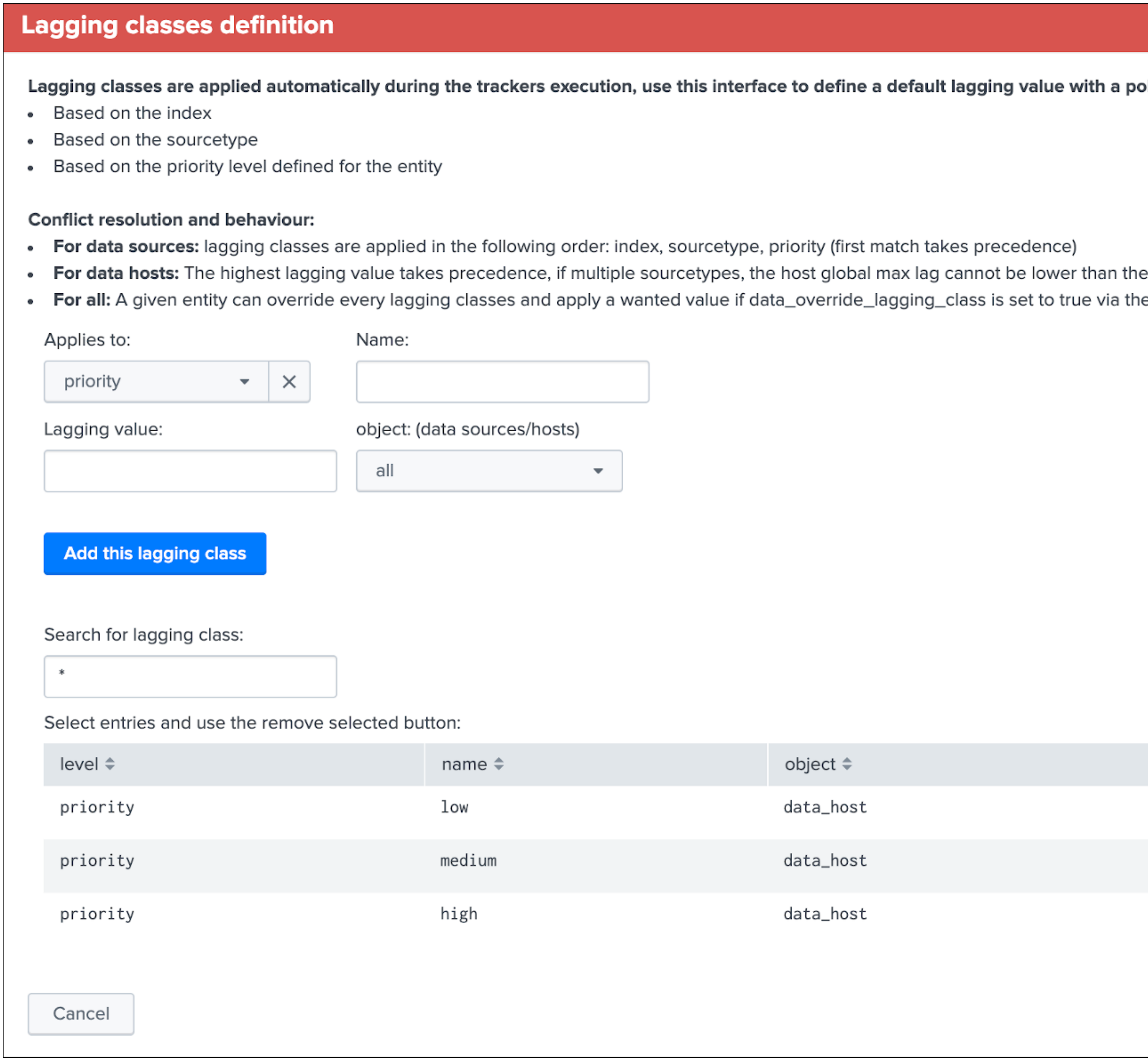
This report would be scheduled, daily for instance, any existing host having a match in the CMDB lookup will get the priority from the CMDB, newly discovered hosts would get the priority updated as soon as the job runs.

Before we apply any lagging classes, our assignment uses the default values:



<table>
<tr><th>data_host</th><th>sourcetype_summary</th><th>last time</th></tr>
<tr><td>EVENTGEN.RETAIL</td><td>main|st=retail_transaction|max_allowed=3600|last_time=31/12/2020 12:03:10|last_event_lag=10|last_ingest_lag=1|✓</td><td>31/12/2020 12</td></tr>
<tr><td>EVENTGEN.SAMPLER</td><td>main|st=sample10-noformat|max_allowed=3600|last_time=31/12/2020 12:03:10|last_event_lag=10|last_ingest_lag=1|✓
main|st=sample1|max_allowed=3600|last_time=31/12/2020 12:03:19|last_event_lag=1|last_ingest_lag=0|✓
main|st=sample2|max_allowed=3600|last_time=31/12/2020 12:03:19|last_event_lag=1|last_ingest_lag=1|✓
main|st=sample3|max_allowed=3600|last_time=31/12/2020 12:03:19|last_event_lag=1|last_ingest_lag=1|✓
main|st=sample4|max_allowed=3600|last_time=31/12/2020 12:03:19|last_event_lag=1|last_ingest_lag=0|✓
main|st=sample5|max_allowed=3600|last_time=31/12/2020 12:03:10|last_event_lag=10|last_ingest_lag=0|✓
main|st=sample6|max_allowed=3600|last_time=31/12/2020 12:03:10|last_event_lag=10|last_ingest_lag=0|✓
main|st=sample7|max_allowed=3600|last_time=31/12/2020 12:03:10|last_event_lag=10|last_ingest_lag=1|✓
main|st=sample8-multiformat|max_allowed=3600|last_time=31/12/2020 12:03:10|last_event_lag=10|last_ingest_lag=0|✓
main|st=sample9-customformat|max_allowed=3600|last_time=31/12/2020 12:03:19|last_event_lag=1|last_ingest_lag=1|✓</td><td>31/12/2020 12</td></tr>
<tr><td>FIREWALL.PAN.AMER.DESIGN.NODE1</td><td>firewall|st=pan:traffic|max_allowed=3600|last_time=31/12/2020 12:03:18|last_event_lag=2|last_ingest_lag=0|✓</td><td>31/12/2020 12</td></tr>
<tr><td>FIREWALL.PAN.AMER.NODE1</td><td>network|st=pan:traffic|max_allowed=3600|last_time=31/12/2020 12:03:18|last_event_lag=2|last_ingest_lag=0|✓</td><td>31/12/2020 12</td></tr>
<tr><td>FIREWALL.PAN.AMER.NODE2</td><td>network|st=pan:traffic|max_allowed=3600|last_time=31/12/2020 12:03:18|last_event_lag=2|last_ingest_lag=1|✓</td><td>31/12/2020 12</td></tr>
<tr><td>FIREWALL.PAN.AMER.RETAIL.NODE1</td><td>firewall|st=pan:traffic|max_allowed=3600|last_time=31/12/2020 12:03:19|last_event_lag=1|last_ingest_lag=1|✓</td><td>31/12/2020 12</td></tr>
<tr><td>FIREWALL.PAN.APAC.DESIGN.NODE1</td><td>firewall|st=pan:traffic|max_allowed=3600|last_time=31/12/2020 12:03:19|last_event_lag=1|last_ingest_lag=0|✓</td><td>31/12/2020 12</td></tr>
<tr><td>FIREWALL.PAN.APAC.NODE1</td><td>network|st=pan:traffic|max_allowed=3600|last_time=31/12/2020 12:03:19|last_event_lag=1|last_ingest_lag=1|✓</td><td>31/12/2020 12</td></tr>
<tr><td>FIREWALL.PAN.APAC.NODE2</td><td>network|st=pan:traffic|max_allowed=3600|last_time=31/12/2020 12:03:18|last_event_lag=2|last_ingest_lag=0|✓</td><td>31/12/2020 12</td></tr>
<tr><td>FIREWALL.PAN.APAC.RETAIL.NODE1</td><td>firewall|st=pan:traffic|max_allowed=3600|last_time=31/12/2020 12:03:19|last_event_lag=1|last_ingest_lag=1|✓</td><td>31/12/2020 12</td></tr>
<tr><td>FIREWALL.PAN.EMEA.DESIGN.NODE1</td><td>firewall|st=pan:traffic|max_allowed=3600|last_time=31/12/2020 12:03:19|last_event_lag=1|last_ingest_lag=0|✓</td><td>31/12/2020 12</td></tr>
<tr><td>FIREWALL.PAN.EMEA.NODE1</td><td>network|st=pan:traffic|max_allowed=3600|last_time=31/12/2020 12:03:19|last_event_lag=1|last_ingest_lag=1|✓</td><td>31/12/2020 12</td></tr>
<tr><td>FIREWALL.PAN.EMEA.NODE2</td><td>network|st=pan:traffic|max_allowed=3600|last_time=31/12/2020 12:03:19|last_event_lag=1|last_ingest_lag=0|✓</td><td>31/12/2020 12</td></tr>
<tr><td>FIREWALL.PAN.EMEA.RETAIL.NODE1</td><td>firewall|st=pan:traffic|max_allowed=3600|last_time=31/12/2020 12:03:19|last_event_lag=1|last_ingest_lag=0|✓</td><td>31/12/2020 12</td></tr>
<tr><td>LINUX.ALL-REGIONS</td><td>linux_amer|st=linux_secure|max_allowed=3600|last_time=31/12/2020 12:03:19|last_event_lag=1|last_ingest_lag=0|✓
linux_apac|st=linux_secure|max_allowed=3600|last_time=31/12/2020 12:03:18|last_event_lag=2|last_ingest_lag=0|✓
linux_emea|st=linux_secure|max_allowed=3600|last_time=31/12/2020 12:03:18|last_event_lag=2|last_ingest_lag=1|✓</td><td>31/12/2020 12</td></tr>
<tr><td>WINSRV1.ACME.COM</td><td>main|st=Script:ListeningPorts|max_allowed=3600|last_time=31/12/2020 12:02:38|last_event_lag=42|last_ingest_lag=2|✓
main|st=WinHostMon|max_allowed=3600|last_time=31/12/2020 12:03:10|last_event_lag=10|last_ingest_lag=0|✓
main|st=XmlWinEventLog|max_allowed=3600|last_time=31/12/2020 12:03:10|last_event_lag=10|last_ingest_lag=0|✓</td><td>31/12/2020 12</td></tr>
</table>

Let’s create our 3 lagging classes via the UI, in our example we will want to apply these policies to data hosts only:



Lagging classes definition

Lagging classes are applied automatically during the trackers execution, use this interface to define a default lagging value with a po

- Based on the index
- Based on the sourcetype
- Based on the priority level defined for the entity

Conflict resolution and behaviour:

- **For data sources:** lagging classes are applied in the following order: index, sourcetype, priority (first match takes precedence)
- **For data hosts:** The highest lagging value takes precedence, if multiple sourcetypes, the host global max lag cannot be lower than the
- **For all:** A given entity can override every lagging classes and apply a wanted value if data_override_lagging_class is set to true via the

Applies to: priority

Name:

Lagging value:

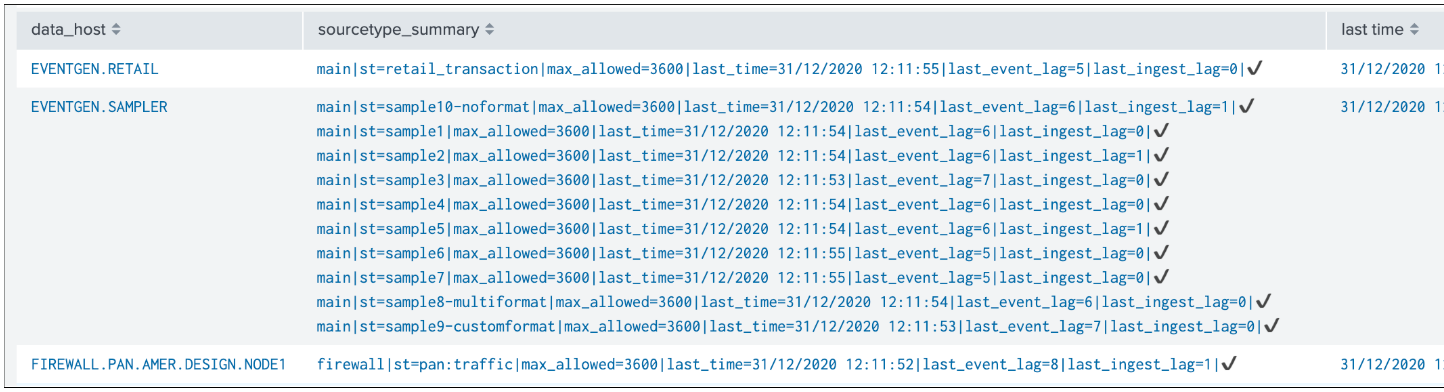
object: (data sources/hosts) all

Search for lagging class: *

Select entries and use the remove selected button:

level	name	object
priority	low	data_host
priority	medium	data_host
priority	high	data_host

Once the policies have been created, we can run the Data hosts trackers manually or wait for the next automatic execution, policies are applied successfully:



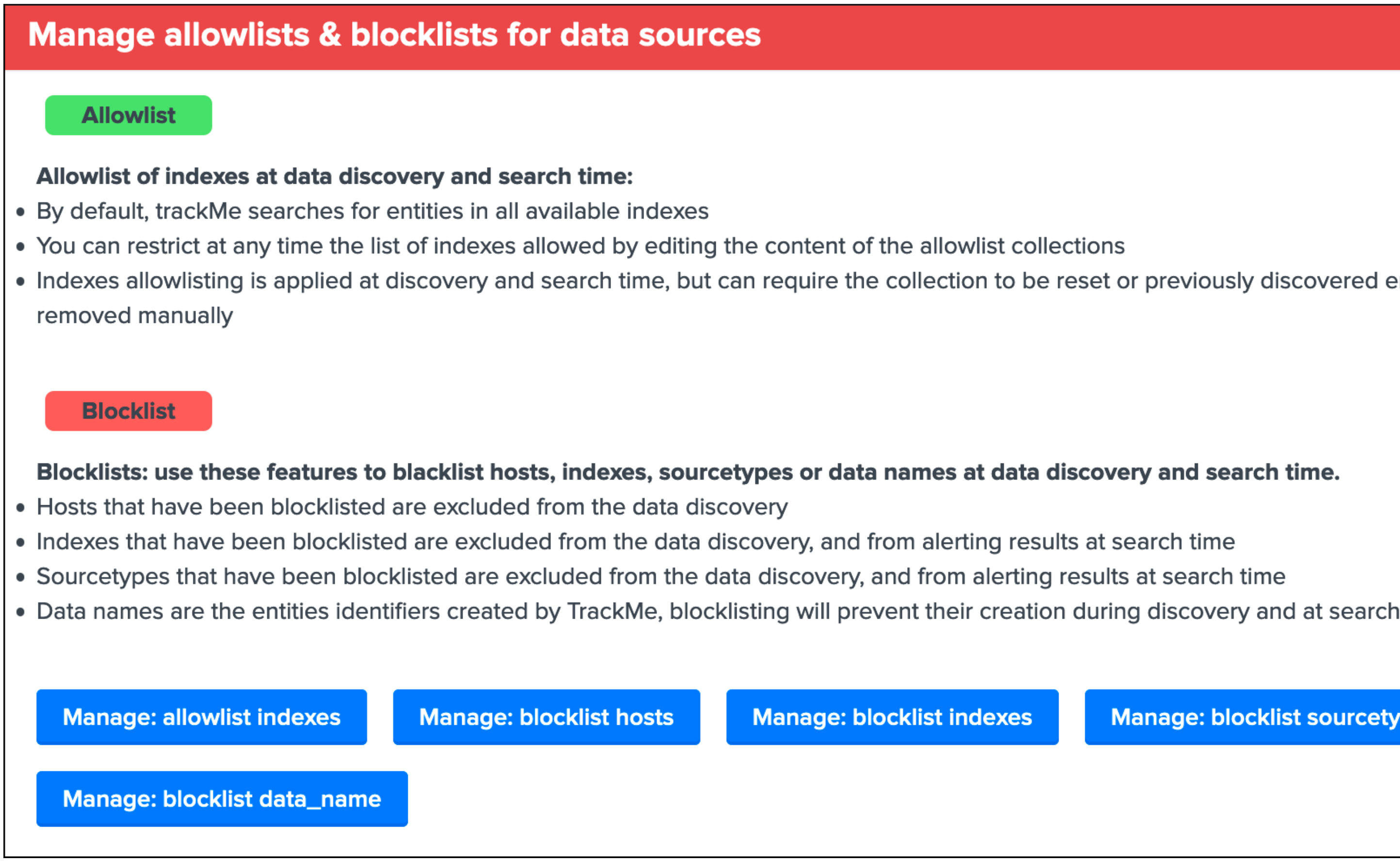
<table>
<tr><th>data_host ⇕</th><th>sourcetype_summary ⇕</th><th>last time ⇕</th></tr>
<tr><td>EVENTGEN.RETAIL</td><td>main|st=retail_transaction|max_allowed=3600|last_time=31/12/2020 12:11:55|last_event_lag=5|last_ingest_lag=0|✓</td><td>31/12/2020 1</td></tr>
<tr><td>EVENTGEN.SAMPLER</td><td>main|st=sample10-noformat|max_allowed=3600|last_time=31/12/2020 12:11:54|last_event_lag=6|last_ingest_lag=1|✓
main|st=sample1|max_allowed=3600|last_time=31/12/2020 12:11:54|last_event_lag=6|last_ingest_lag=0|✓
main|st=sample2|max_allowed=3600|last_time=31/12/2020 12:11:54|last_event_lag=6|last_ingest_lag=1|✓
main|st=sample3|max_allowed=3600|last_time=31/12/2020 12:11:53|last_event_lag=7|last_ingest_lag=0|✓
main|st=sample4|max_allowed=3600|last_time=31/12/2020 12:11:54|last_event_lag=6|last_ingest_lag=0|✓
main|st=sample5|max_allowed=3600|last_time=31/12/2020 12:11:54|last_event_lag=6|last_ingest_lag=1|✓
main|st=sample6|max_allowed=3600|last_time=31/12/2020 12:11:55|last_event_lag=5|last_ingest_lag=0|✓
main|st=sample7|max_allowed=3600|last_time=31/12/2020 12:11:55|last_event_lag=5|last_ingest_lag=0|✓
main|st=sample8-multiformat|max_allowed=3600|last_time=31/12/2020 12:11:54|last_event_lag=6|last_ingest_lag=0|✓
main|st=sample9-customformat|max_allowed=3600|last_time=31/12/2020 12:11:53|last_event_lag=7|last_ingest_lag=0|✓</td><td>31/12/2020 1</td></tr>
<tr><td>FIREWALL.PAN.AMER.DESIGN.NODE1</td><td>firewall|st=pan:traffic|max_allowed=3600|last_time=31/12/2020 12:11:52|last_event_lag=8|last_ingest_lag=1|✓</td><td>31/12/2020 1</td></tr>
</table>

Note: The lagging value that will be inherited from the policy cannot be lower than the highest lagging value between the sourcetypes of a given host, shall this be the case, TrackMe will automatically use the highest lagging value between all sourcetypes linked to that host.

3.1.14 Allowlisting & Blocklisting

Allowlisting & Blocklisting

- TrackMe supports allowlisting and blocklisting to configure the scope of the data discovery.
- Allowlisting provides a framework to easily restrict the entire scope of TracKme to an explicit list of allowed indexes.
- Blocklisting provides the opposite feature on a per index / sourcetype / host / data_name basis.



Manage allowlists & blocklists for data sources

Allowlist of indexes at data discovery and search time:

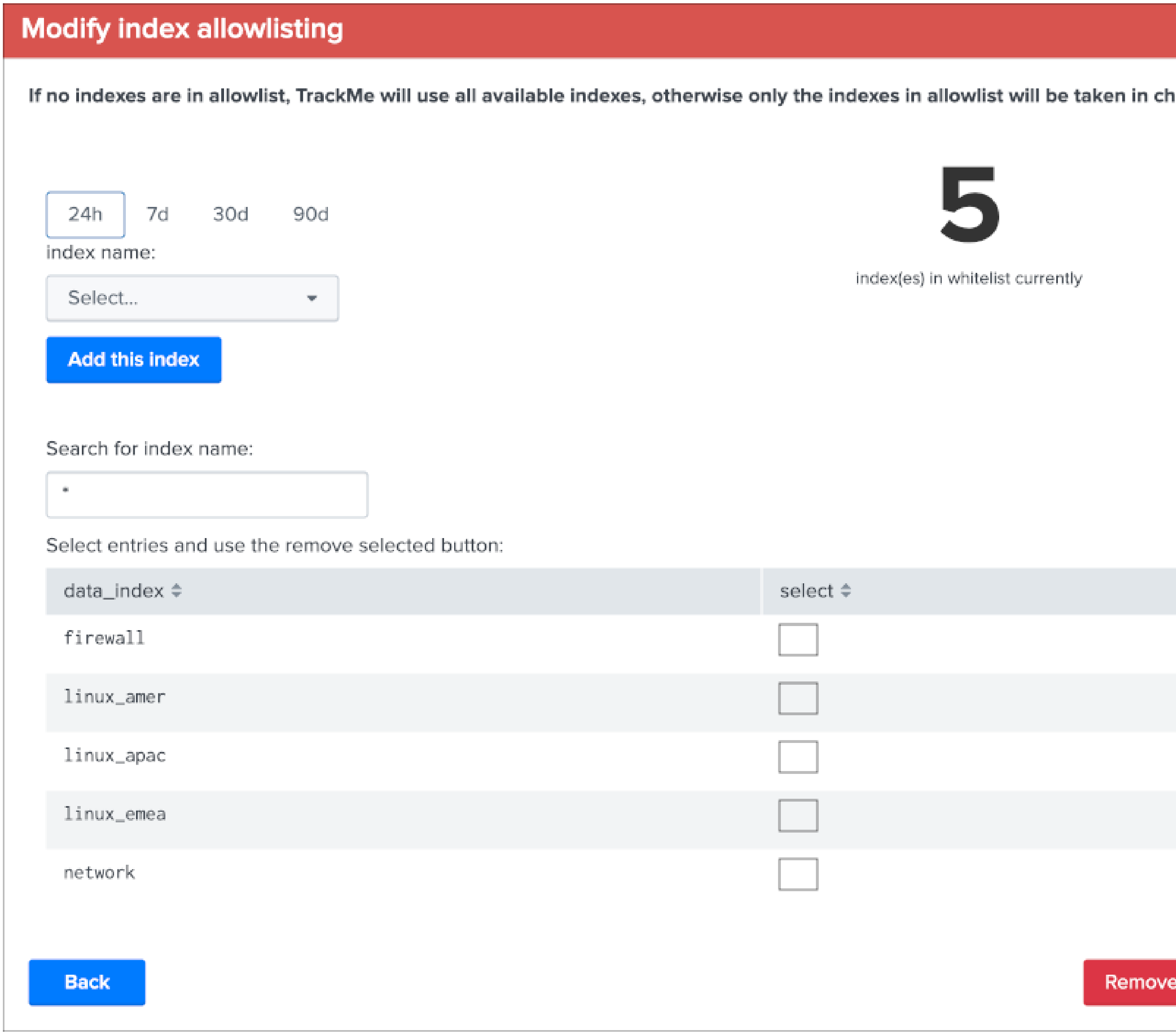
- By default, trackMe searches for entities in all available indexes
- You can restrict at any time the list of indexes allowed by editing the content of the allowlist collections
- Indexes allowlisting is applied at discovery and search time, but can require the collection to be reset or previously discovered e removed manually

Blocklists: use these features to blacklist hosts, indexes, sourcetypes or data names at data discovery and search time.

- Hosts that have been blocklisted are excluded from the data discovery
- Indexes that have been blocklisted are excluded from the data discovery, and from alerting results at search time
- Sourcetypes that have been blocklisted are excluded from the data discovery, and from alerting results at search time
- Data names are the entities identifiers created by TrackMe, blocklisting will prevent their creation during discovery and at search

Manage: allowlist indexes
Manage: blocklist hosts
Manage: blocklist indexes
Manage: blocklist sourcety
Manage: blocklist data_name

The default behaviour of TrackMe is to track data available in all indexes, which changes if allowlisting has been defined:



Modify index allowlisting

If no indexes are in allowlist, TrackMe will use all available indexes, otherwise only the indexes in allowlist will be taken in ch

5

index(es) in whitelist currently

index name: Select...

Search for index name: *

Select entries and use the remove selected button:

data_index	select
firewall	☐
linux_amer	☐
linux_apac	☐
linux_emea	☐
network	☐

Different level of blocklisting features are provided out of the box, which features can be used to avoid taking in consideration indexes, sourcetypes, hosts and data sources based on the data_name generated by TrackMe.

*The following type of blocklisting entries are supported:**

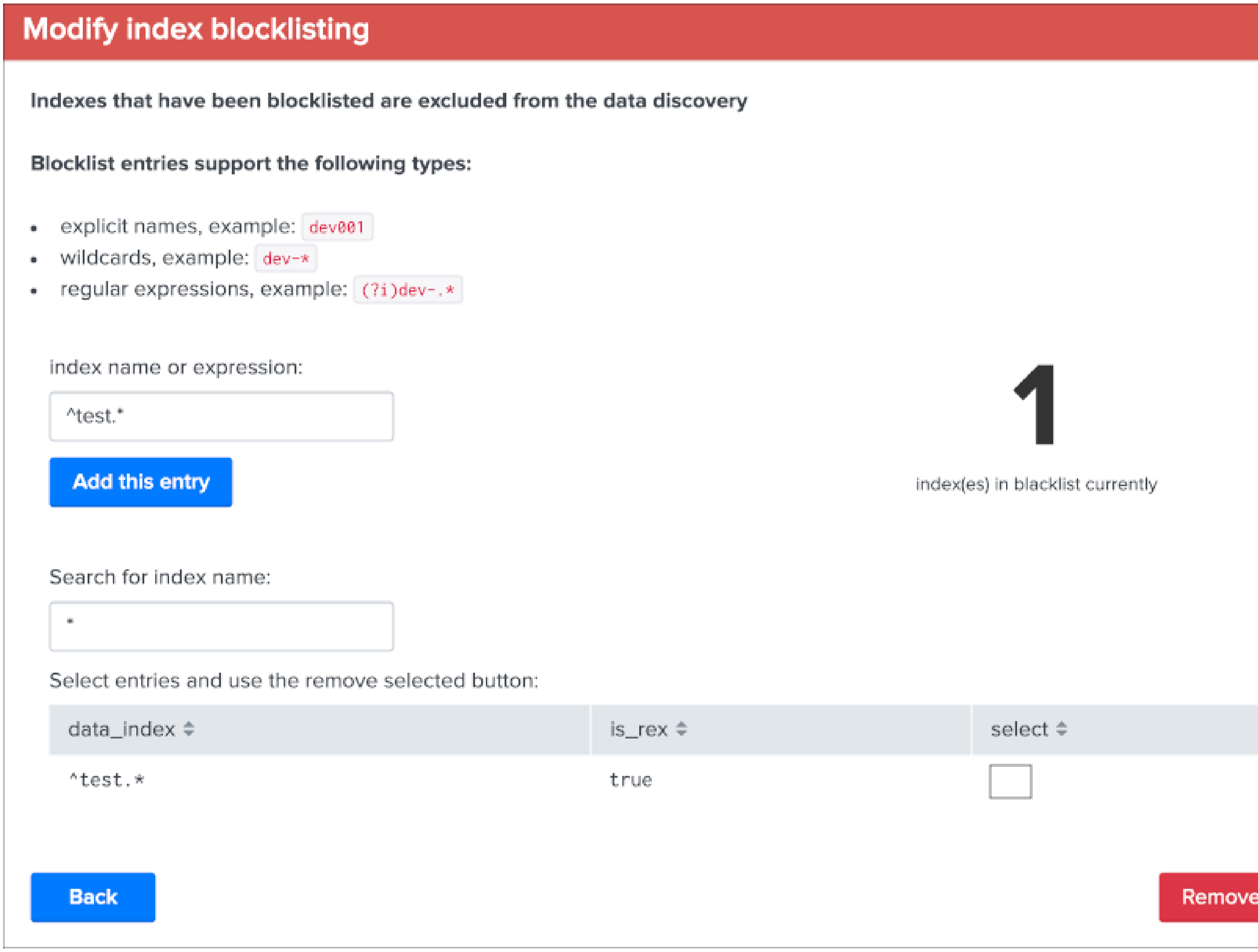
- explicit names, example: `dev001`
- wildcards, example: `dev-*`
- regular expressions, example: `(?i)dev-.*`

regular expressions are supported starting version 1.1.6.

metric_category blocklisting for metric hosts supports explicit blacklist only.

Adding or removing a blocklist item if performed entirely and easily within the UI:

]



Modify index blocklisting

Indexes that have been blocklisted are excluded from the data discovery

Blocklist entries support the following types:

- explicit names, example: `dev001`
- wildcards, example: `dev-*`
- regular expressions, example: `(?i)dev-.*`

index name or expression: ^test.*

1

index(es) in blacklist currently

Search for index name: *

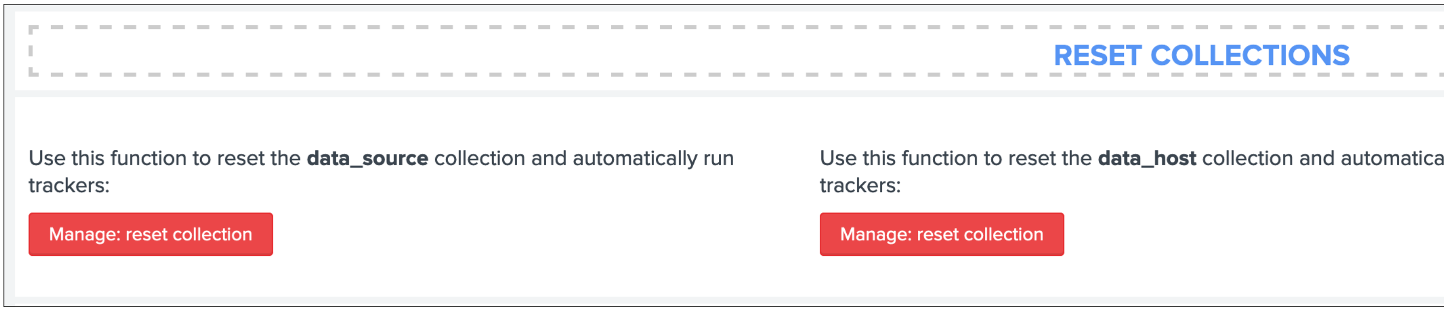
Select entries and use the remove selected button:

data_index	is_rex	select
^test.*	true	□

3.1.15 Resetting collections to factory defaults

> **Warning:** Resetting the collections will entirely flush the content of the data sources / hosts / metric hosts collections, which includes any custom setting that will be have been configured as such as the maximal lagging value.

The TrackMe Manage and Configure UI provides way to reset the full content of the collections:

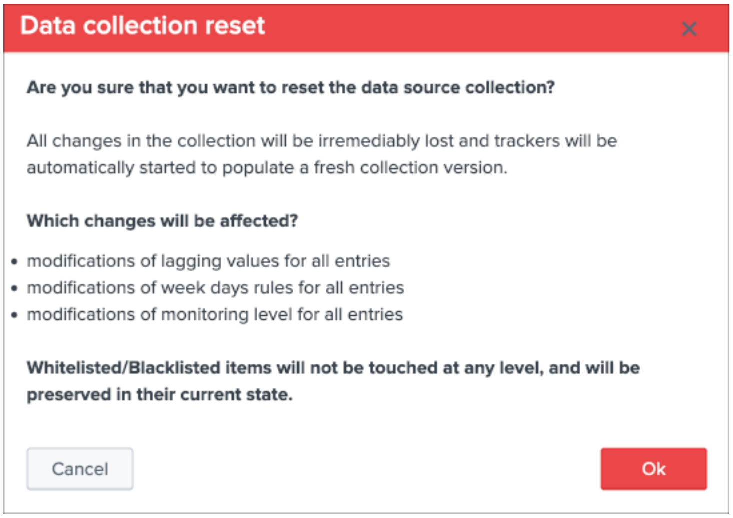


RESET COLLECTIONS

Use this function to reset the **data_source** collection and automatically run trackers:

Use this function to reset the **data_host** collection and automatica trackers:

If you validate the operation, all configuration changes will be lost (like week days monitoring rules changes, etc) and the long term tracker will be run automatically:



Data collection reset

Are you sure that you want to reset the data source collection?

All changes in the collection will be irremediably lost and trackers will be automatically started to populate a fresh collection version.

Which changes will be affected?

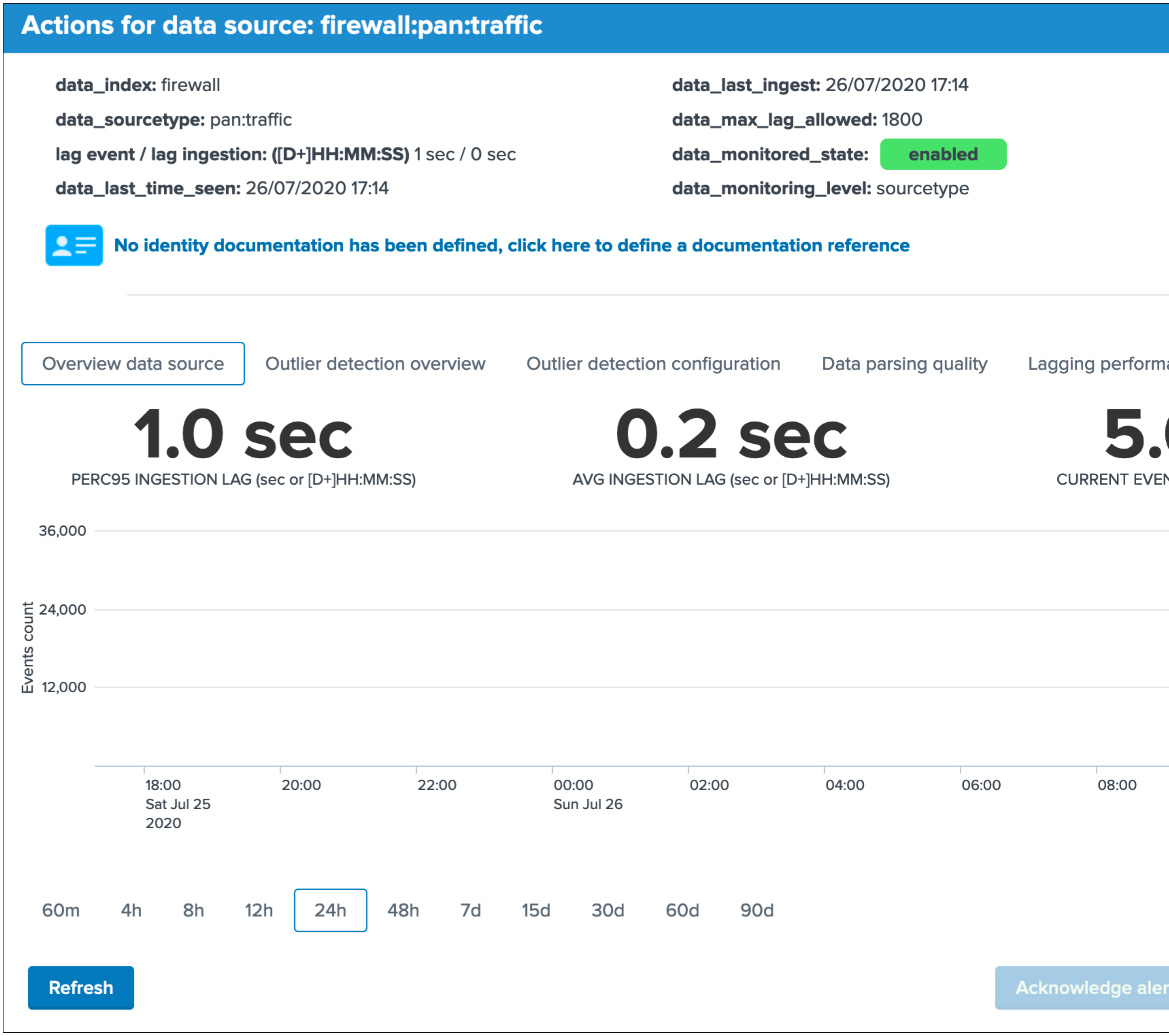
- modifications of lagging values for all entries
- modifications of week days rules for all entries
- modifications of monitoring level for all entries

Whitelisted/Blacklisted items will not be touched at any level, and will be preserved in their current state.

Once the collection has been cleared, you can simply wait for the trackers next executions, or manually perform a run of the short term and/or long term trackers.

3.1.16 Deletion of entities

You can delete a data source or a data host that was discovered automatically by using the built-in delete function:



Actions for data source: firewall:pan:traffic

data_index: firewall

data_sourcetype: pan:traffic

lag event / lag ingestion: ([D+]HH:MM:SS) 1 sec / 0 sec

data_last_time_seen: 26/07/2020 17:14

data_last_ingest: 26/07/2020 17:14

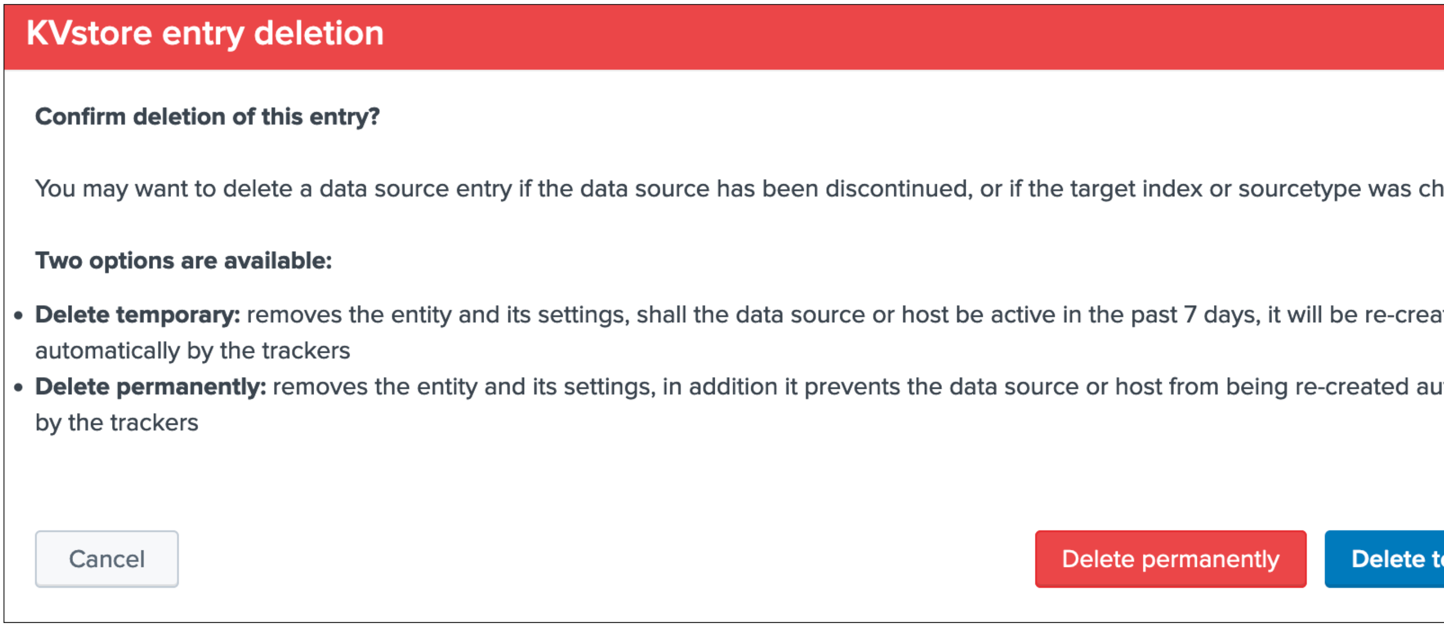
data_max_lag_allowed: 1800

data_monitored_state: enabled

data_monitoring_level: sourcetype

No identity documentation has been defined, click here to define a documentation reference

Two options are available:



KVstore entry deletion

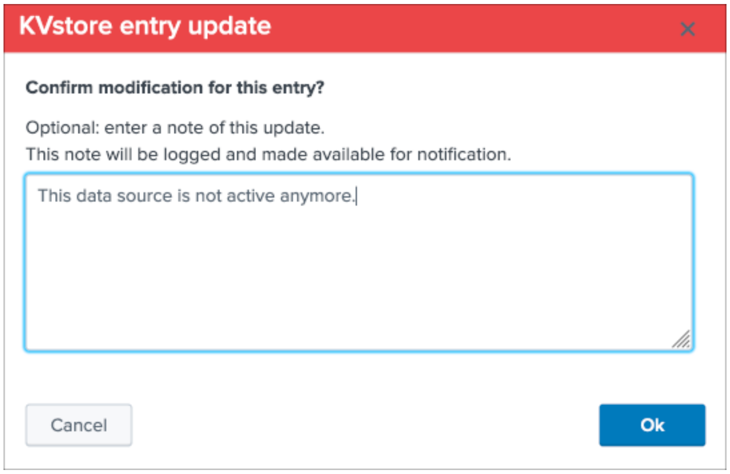
Confirm deletion of this entry?

You may want to delete a data source entry if the data source has been discontinued, or if the target index or sourcetype was ch

Two options are available:

- **Delete temporary:** removes the entity and its settings, shall the data source or host be active in the past 7 days, it will be re-crea automatically by the trackers
- **Delete permanently:** removes the entity and its settings, in addition it prevents the data source or host from being re-created au by the trackers

- When the data source or host is temporary removed, it will be automatically re-created if it has been active during the time range scope of the trackers.
- When the data source or host is permanently removed, a record of the operation is stored in the audit changes KVstore collection, which we automatically use to prevent the source from being re-created effectively.



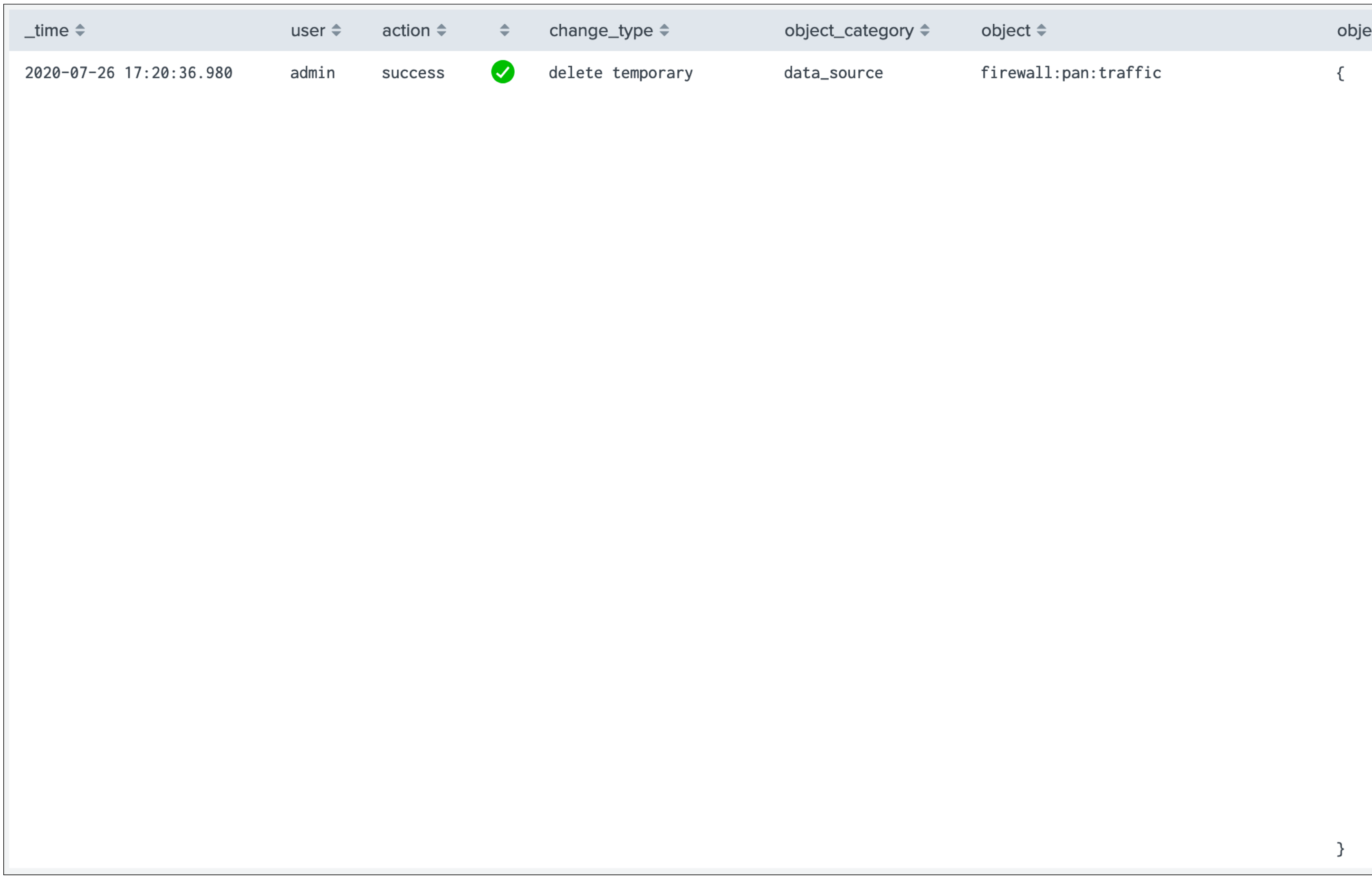
KVstore entry update

Confirm modification for this entry?

Optional: enter a note of this update.
This note will be logged and made available for notification.

This data source is not active anymore.

When an entity is deleted via the UI, the audit record exposes the full content of the entity as it was at the time of the deletion:



_time	user	action		change_type	object_category	object	obje

2020-07-26 17:20:36.980	admin	success	✓	delete temporary	data_source	firewall:pan:traffic	{
							}

It is not possible at the moment to `restore` an entity that was previously deleted, however an active entity can be recreated automatically depending on the scope of the data discovery (the data must be available to TrackMe), and with the help of the audit record you could easily re-apply any settings that would be required.

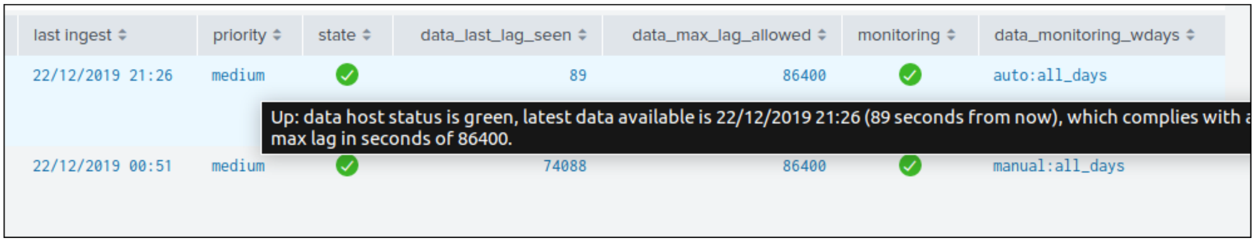
If an entity was `deleted permanently` and you wish to get it recreated, the entity must first be actively sending data, TrackMe must be able to see the data (`allowlist` and `blocklist`) and you would need to remove the audit record in the following collection:

- `trackme_audit_changes`

Once the record has been deleted, the entity will be recreated automatically during the execution of the trackers.

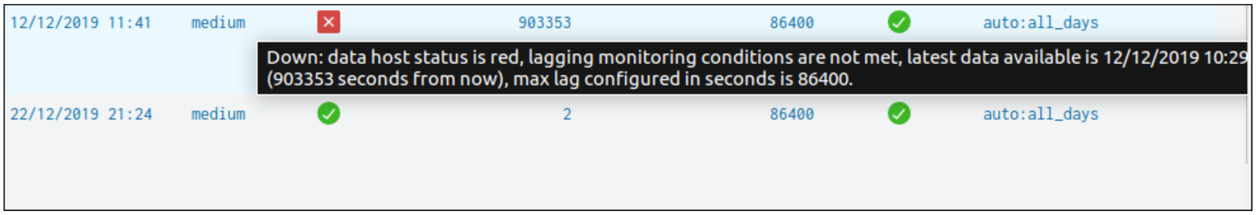
3.1.17 Icon dynamic messages

For each type object (data sources / data hosts / metric hosts) the UI shows a status icon which describes the reason for the status with dynamic information:



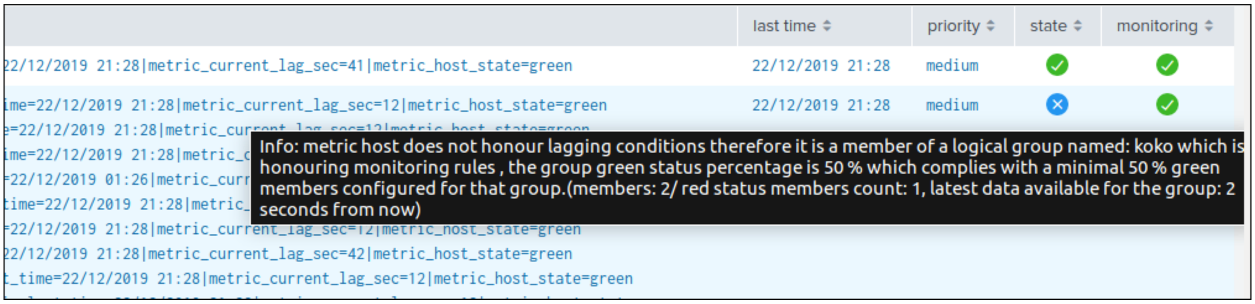
last ingest	priority	state	data_last_lag_seen	data_max_lag_allowed	monitoring	data_monitoring_wdays
22/12/2019 21:26	medium	✓	89	86400	✓	auto:all_days
22/12/2019 00:51	medium	✓	74088	86400	✓	manual:all_days

Up: data host status is green, latest data available is 22/12/2019 21:26 (89 seconds from now), which complies with a max lag in seconds of 86400.



12/12/2019 11:41	medium	☒	903353	86400	✓	auto:all_days
22/12/2019 21:24	medium	✓	2	86400	✓	auto:all_days

Down: data host status is red, lagging monitoring conditions are not met, latest data available is 12/12/2019 10:29 (903353 seconds from now), max lag configured in seconds is 86400.



	last time	priority	state	monitoring
22/12/2019 21:28|metric_current_lag_sec=41|metric_host_state=green	22/12/2019 21:28	medium	✓	✓
ime=22/12/2019 21:28|metric_current_lag_sec=12|metric_host_state=green	22/12/2019 21:28	medium	☒	✓

Info: metric host does not honour lagging conditions therefore it is a member of a logical group named: koko which is honouring monitoring rules , the group green status percentage is 50 % which complies with a minimal 50 % green members configured for that group.(members: 2/ red status members count: 1, latest data available for the group: 2 seconds from now)

To access to the dynamic message, simply focus over the icon in the relevant table cell, and the Web browser will automatically display the message for that entity.

3.1.18 Logical groups (clusters)

Logical groups feature

Logical groups

Logical groups are groups of entities that will be considered as an ensemble for monitoring purposes.

A typical use case is a couple of active / passive appliances, where only the active member generates data.

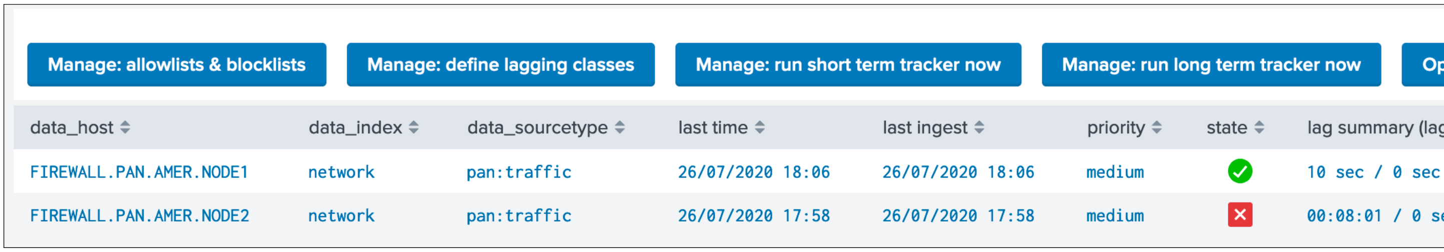
When associated in a Logical group, the entity status relies on the minimal green percentage configured during the group creation versus the current green percentage of the group. (percentages of members green)

Notes: Logical groups are available to data hosts and metric hosts monitoring objects.

Logical group example

Let's have a look at a simple example of an active / passive firewall, we have two entities which form together a cluster.

Because the passive node might not generate data, we only want to alert if both the active and the passive are not actively sending data.



Manage: allowlists & blocklists
Manage: define lagging classes
Manage: run short term tracker now
Manage: run long term tracker now
Op

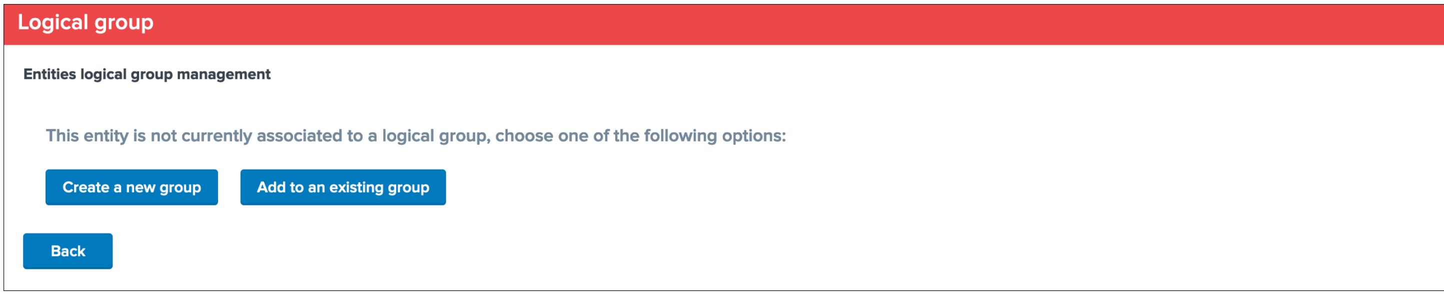
data_host	data_index	data_sourcetype	last time	last ingest	priority	state	lag summary (lag
FIREWALL.PAN.AMER.NODE1	network	pan:traffic	26/07/2020 18:06	26/07/2020 18:06	medium	✓	10 sec / 0 sec
FIREWALL.PAN.AMER.NODE2	network	pan:traffic	26/07/2020 17:58	26/07/2020 17:58	medium	☒	00:08:01 / 0 se

In our example, we have two hosts:

- `FIREWALL.PAN.AMER.NODE1` which is the active node, and green in TrackMe
- `FIREWALL.PAN.AMER.NODE2` which is the passive node, and hasn't sent data recently enough in TrackMe to be considered as green

Let's create a logical group:

For this, we click on the first host, then Modify and finally we click on the Logical groups button:

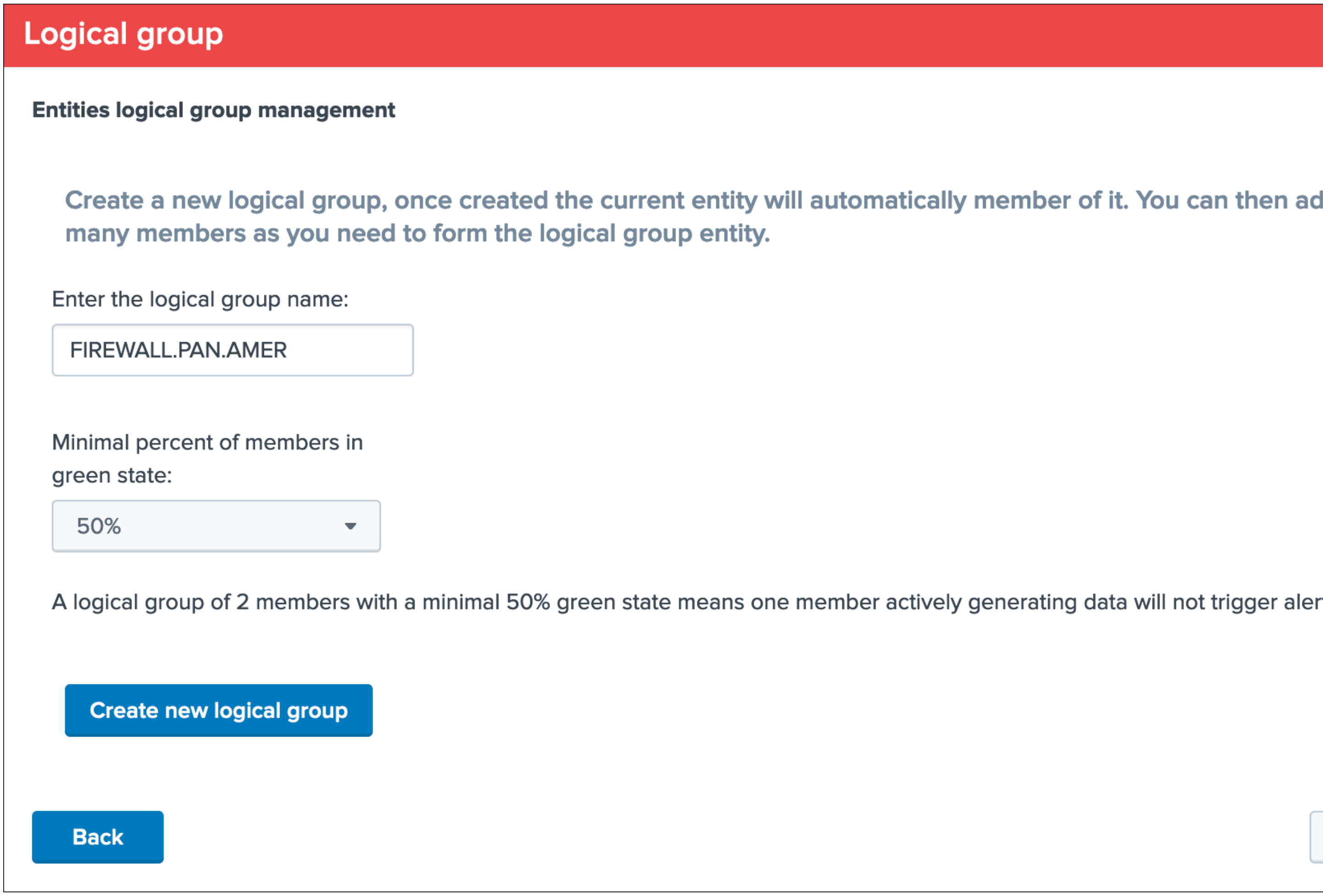


Logical group

Entities logical group management

This entity is not currently associated to a logical group, choose one of the following options:

Since we don't have yet a group, let's create a new group:



Logical group

Entities logical group management

Create a new logical group, once created the current entity will automatically member of it. You can then ad

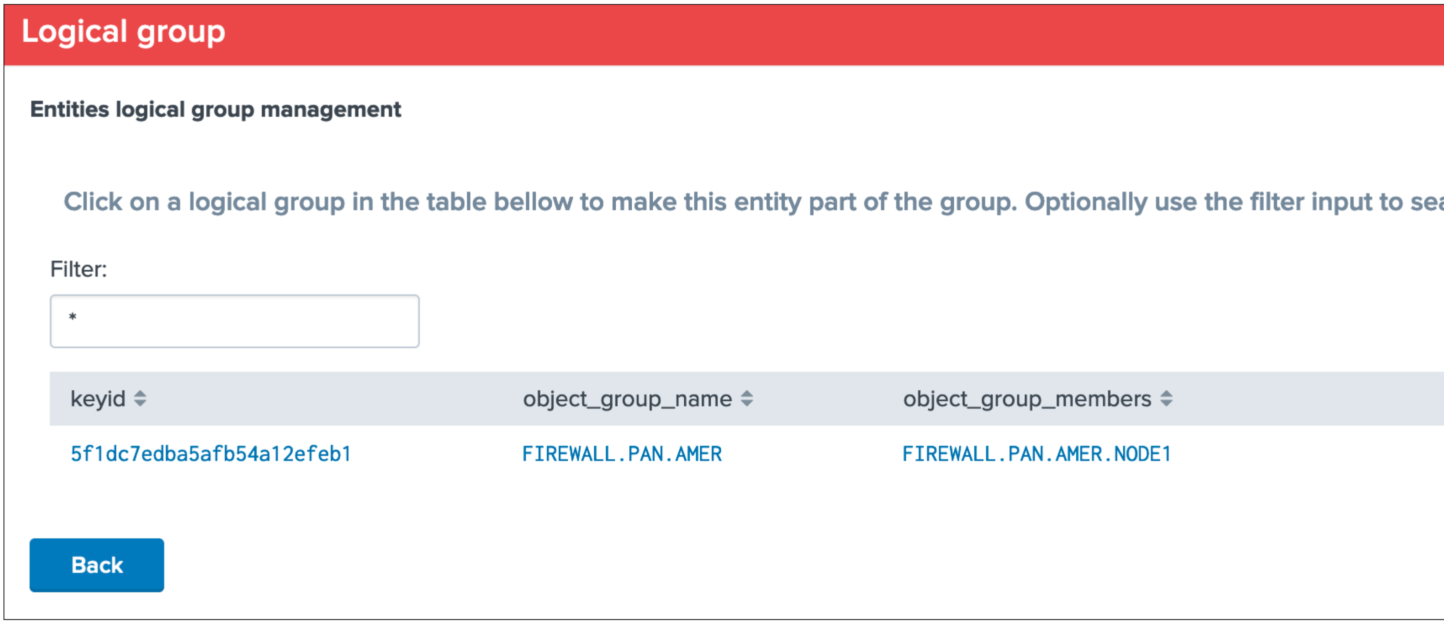
many members as you need to form the logical group entity.

Enter the logical group name: FIREWALL.PAN.AMER

Minimal percent of members in green state: 50%

A logical group of 2 members with a minimal 50% green state means one member actively generating data will not trigger aler

Once the group is created, the first node is automatically associated with the group, let's click on the second node and associate it with our new group:



Logical group

Entities logical group management

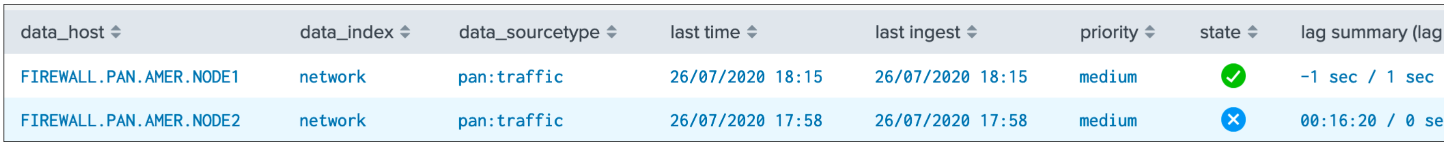
Click on a logical group in the table bellow to make this entity part of the group. Optionally use the filter input to sea

Filter:

*

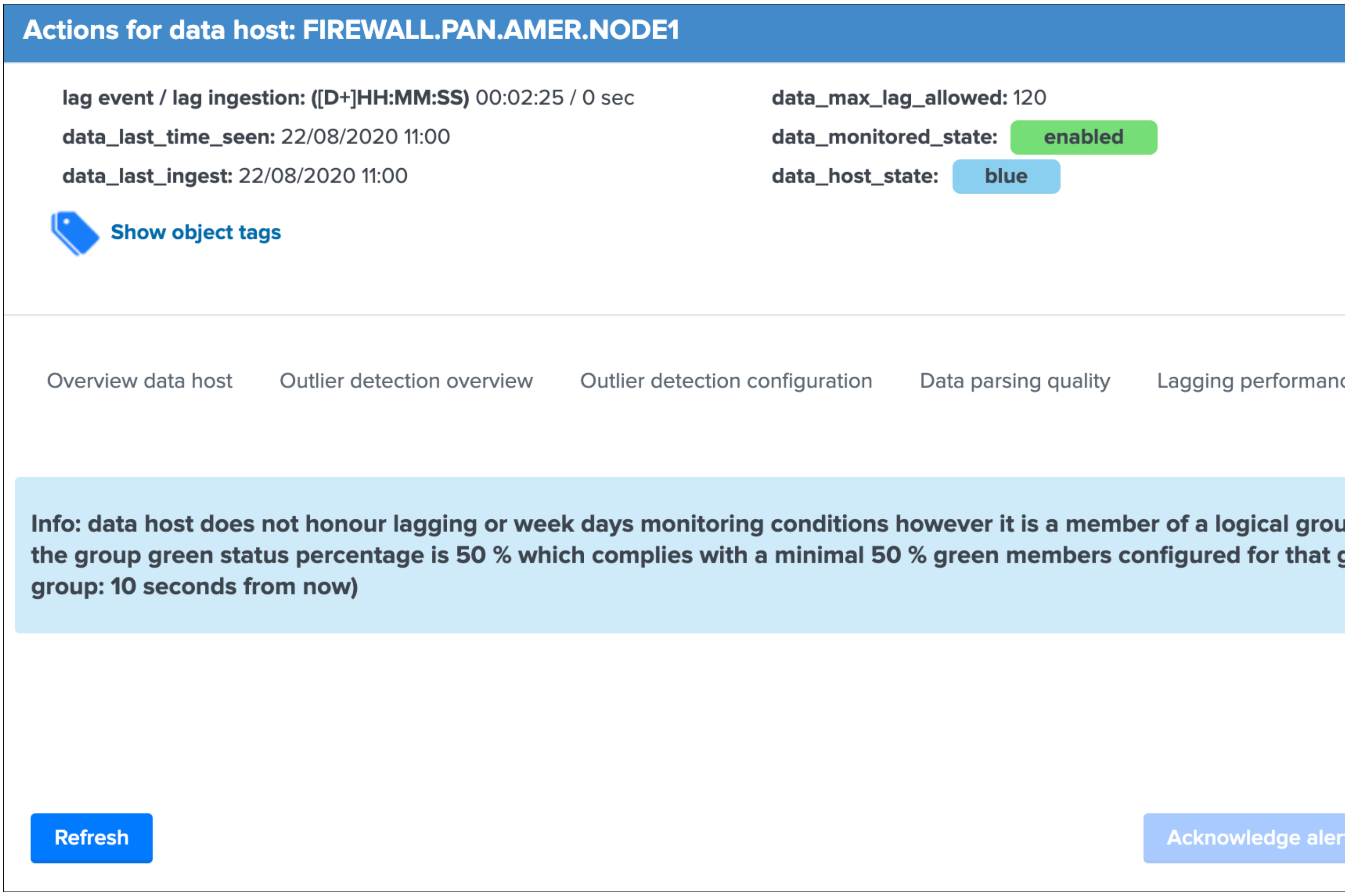
keyid	object_group_name	object_group_members
5f1dc7edba5afb54a12efeb1	FIREWALL.PAN.AMER	FIREWALL.PAN.AMER.NODE1

We clicked on the group which we want to associate the entity with, which performs the association automatically, finally we can see the state of the second host has changed from `red` to `blue`:



data_host	data_index	data_sourcetype	last time	last ingest	priority	state	lag summary (lag
FIREWALL.PAN.AMER.NODE1	network	pan:traffic	26/07/2020 18:15	26/07/2020 18:15	medium	✓	-1 sec / 1 sec
FIREWALL.PAN.AMER.NODE2	network	pan:traffic	26/07/2020 17:58	26/07/2020 17:58	medium	⊗	00:16:20 / 0 se

If we click on the entity and check the status message tab, we can observe a clear message indicating the reason of the state including the name of the logical group this entity is part of:



Actions for data host: FIREWALL.PAN.AMER.NODE1

lag event / lag ingestion: ([D+]HH:MM:SS) 00:02:25 / 0 sec

data_last_time_seen: 22/08/2020 11:00

data_last_ingest: 22/08/2020 11:00

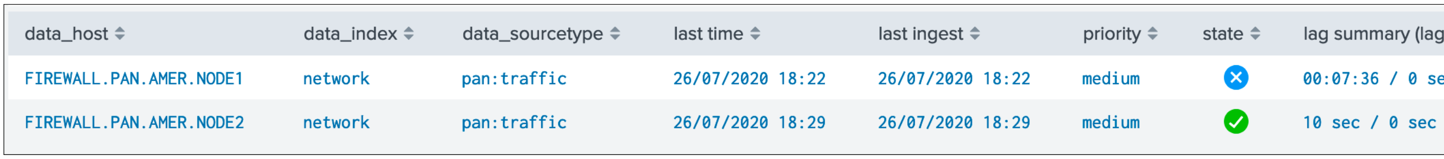
data_max_lag_allowed: 120

data_monitored_state: enabled

data_host_state: blue

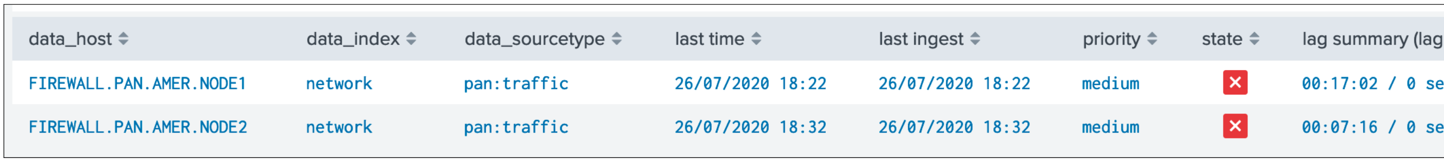
Info: data host does not honour lagging or week days monitoring conditions however it is a member of a logical grou the group green status percentage is 50 % which complies with a minimal 50 % green members configured for that g group: 10 seconds from now)

Shall later on the situation be inversed, the active node became passive and the passive became passive, the states will be reversed, since the logical group monitoring rules (50% active) are respected there will not be any alert generated:



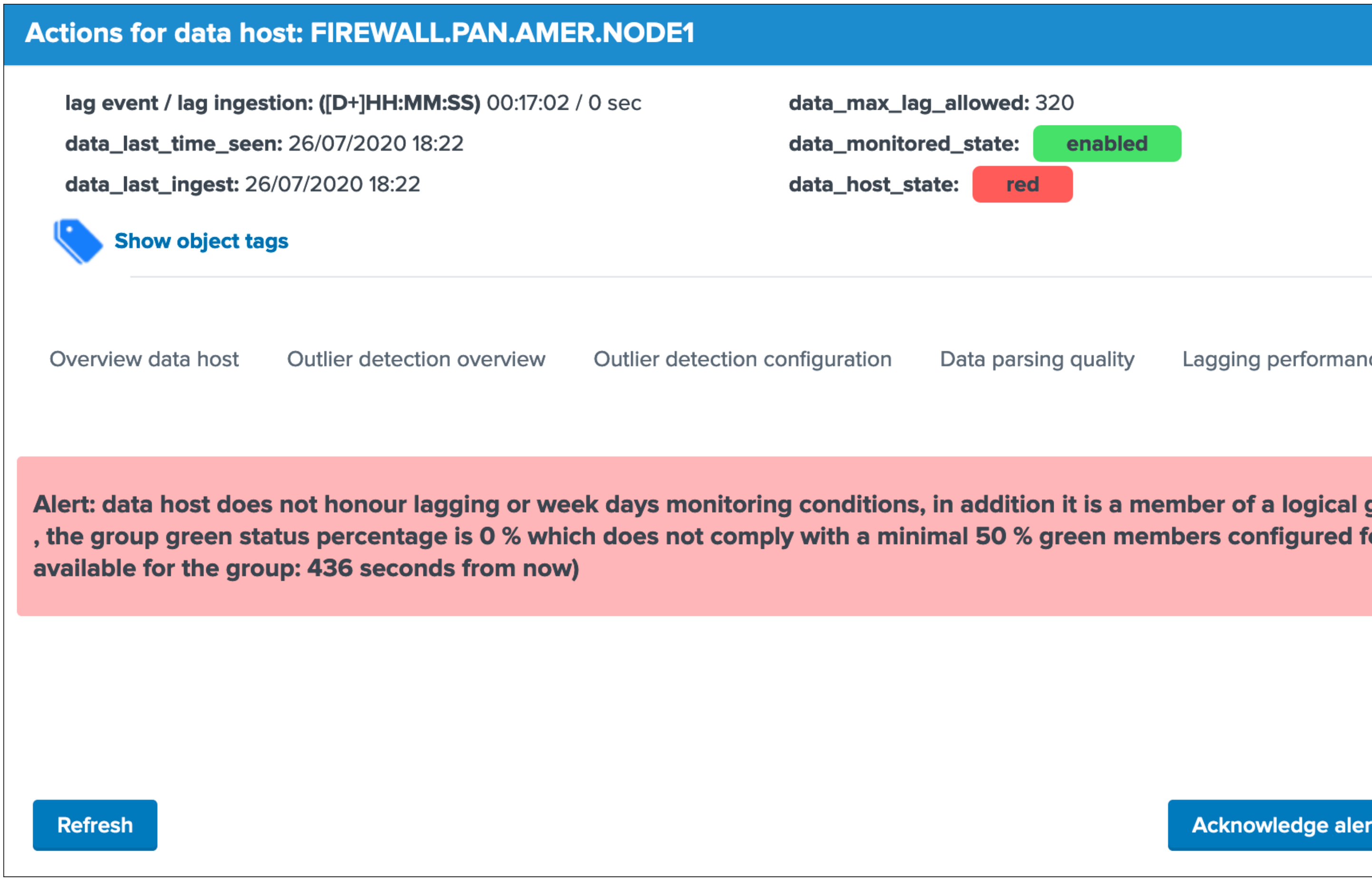
data_host ⇕	data_index ⇕	data_sourcetype ⇕	last time ⇕	last ingest ⇕	priority ⇕	state ⇕	lag summary (lag
FIREWALL.PAN.AMER.NODE1	network	pan:traffic	26/07/2020 18:22	26/07/2020 18:22	medium	✗	00:07:36 / 0 se
FIREWALL.PAN.AMER.NODE2	network	pan:traffic	26/07/2020 18:29	26/07/2020 18:29	medium	✓	10 sec / 0 sec

Finally, shall both entities be inactive, their status will be `red` and alerts will be emitted as none of these are meeting the logical group monitoring rules:



data_host ⇕	data_index ⇕	data_sourcetype ⇕	last time ⇕	last ingest ⇕	priority ⇕	state ⇕	lag summary (lag
FIREWALL.PAN.AMER.NODE1	network	pan:traffic	26/07/2020 18:22	26/07/2020 18:22	medium	☒	00:17:02 / 0 se
FIREWALL.PAN.AMER.NODE2	network	pan:traffic	26/07/2020 18:32	26/07/2020 18:32	medium	☒	00:07:16 / 0 se

The status message tab would expose clearly the reason of the `red` status:



Actions for data host: FIREWALL.PAN.AMER.NODE1

lag event / lag ingestion: ([D+]HH:MM:SS) 00:17:02 / 0 sec

data_last_time_seen: 26/07/2020 18:22

data_last_ingest: 26/07/2020 18:22

data_max_lag_allowed: 320

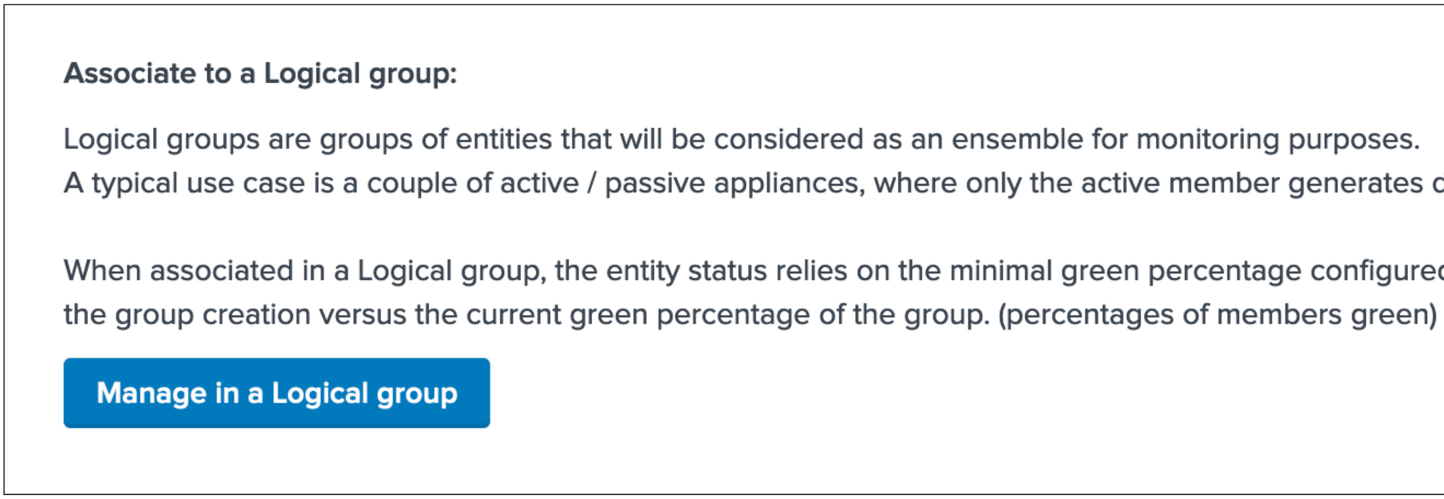
data_monitored_state: enabled

data_host_state: red

Alert: data host does not honour lagging or week days monitoring conditions, in addition it is a member of a logical g , the group green status percentage is 0 % which does not comply with a minimal 50 % green members configured f available for the group: 436 seconds from now)

Create a new logical group

To create a new logical group and associate a first member, enter the unified modification window (click on an entity and modify button), then click on the “Manage in a Logical group” button:

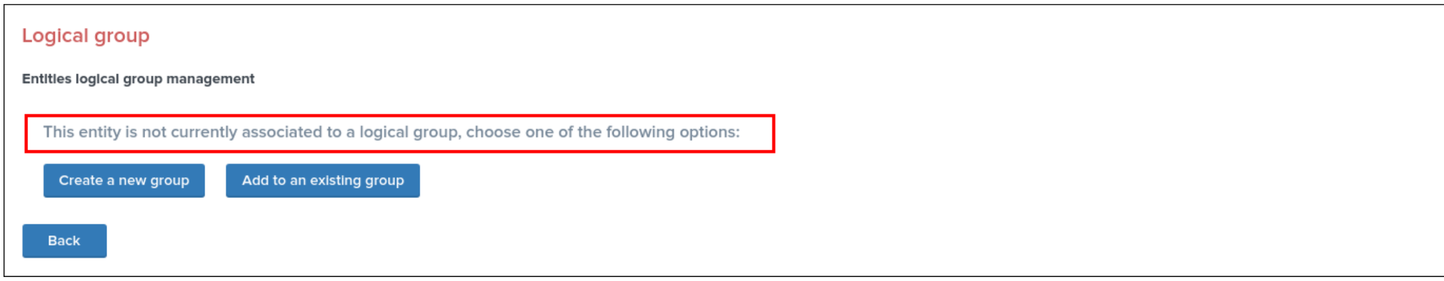


Associate to a Logical group:

Logical groups are groups of entities that will be considered as an ensemble for monitoring purposes.
A typical use case is a couple of active / passive appliances, where only the active member generates c

When associated in a Logical group, the entity status relies on the minimal green percentage configurec the group creation versus the current green percentage of the group. (percentages of members green)

If the entity is not yet associated with a logical group (an entity cannot be associated with more than one group), the following message is displayed:

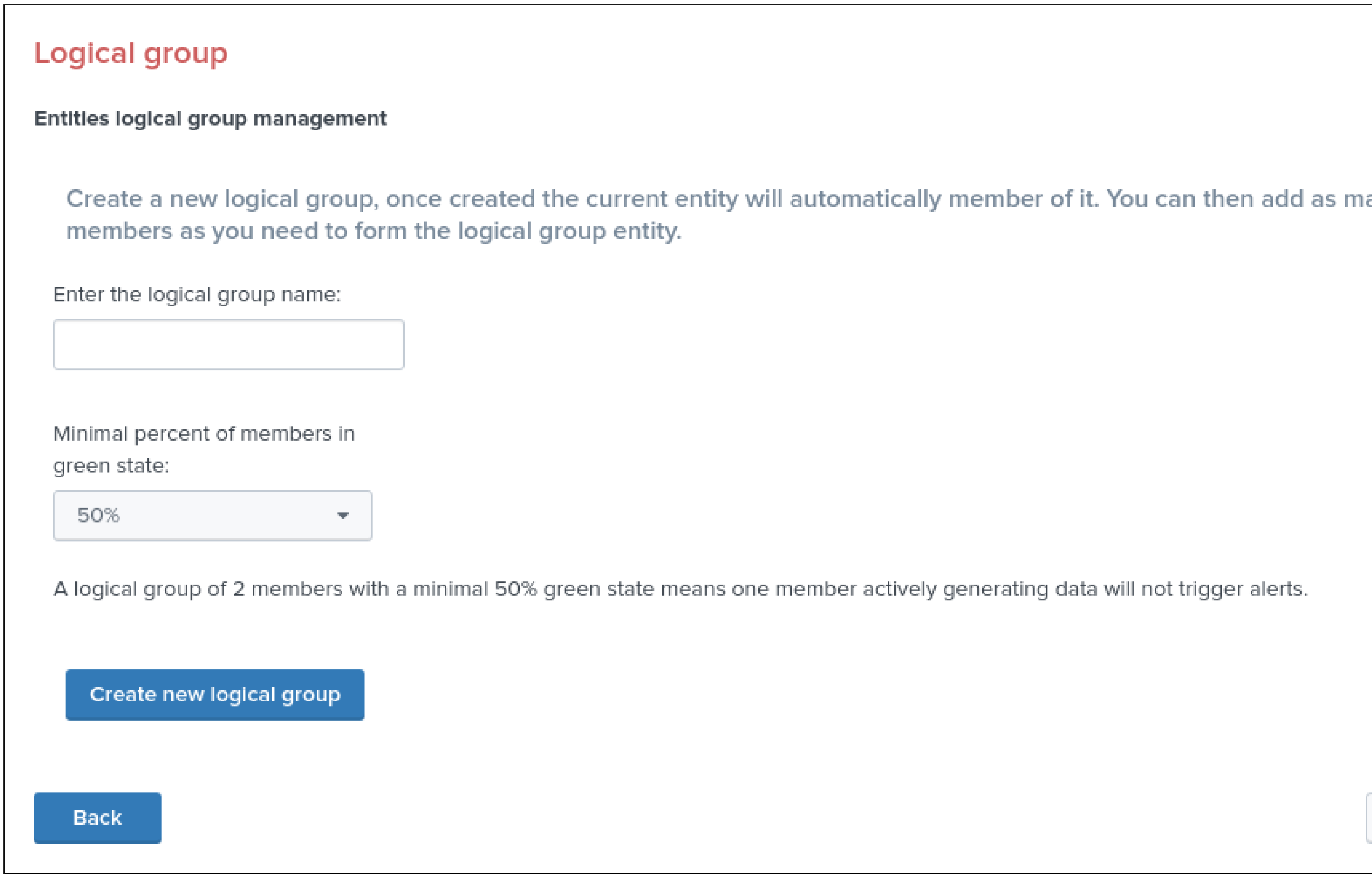


Logical group

Entities logical group management

This entity is not currently associated to a logical group, choose one of the following options:

Click on the button "Create a new group" which opens the following configuration window:



Logical group

Entities logical group management

Create a new logical group, once created the current entity will automatically member of it. You can then add as ma
members as you need to form the logical group entity.

Enter the logical group name:

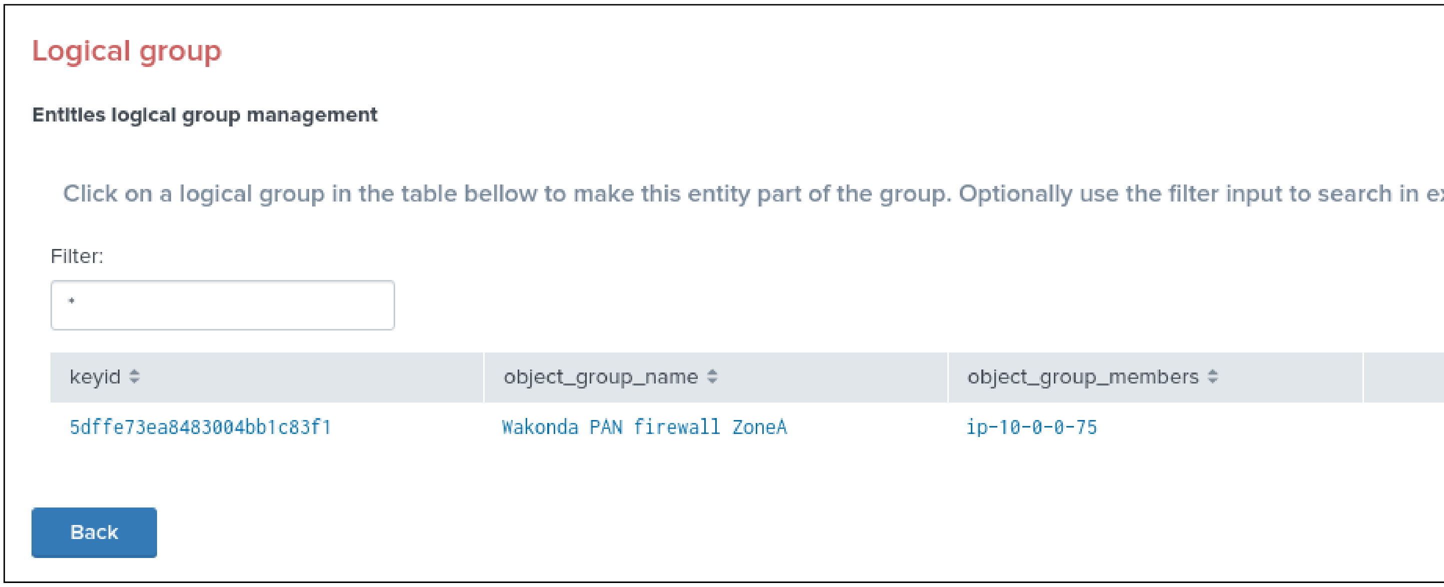
Minimal percent of members in green state: 50%

A logical group of 2 members with a minimal 50% green state means one member actively generating data will not trigger alerts.

- Enter a name for the logical group (names do not need to be unique and can accept any ascii characters)
- Choose a minimal green percentage for the group, this defines the alerting factor for that group, for example when using 50% (default), a minimal 50% or more of the members need to be green for the logical group status to be green

Associate to an existing logical group

If a logical group already exists and you wish to associate this entity to this group, following the same path (Modify entity) and select the button "Add to an existing group":



Logical group

Entities logical group management

Click on a logical group in the table bellow to make this entity part of the group. Optionally use the filter input to search in e

Filter:

*

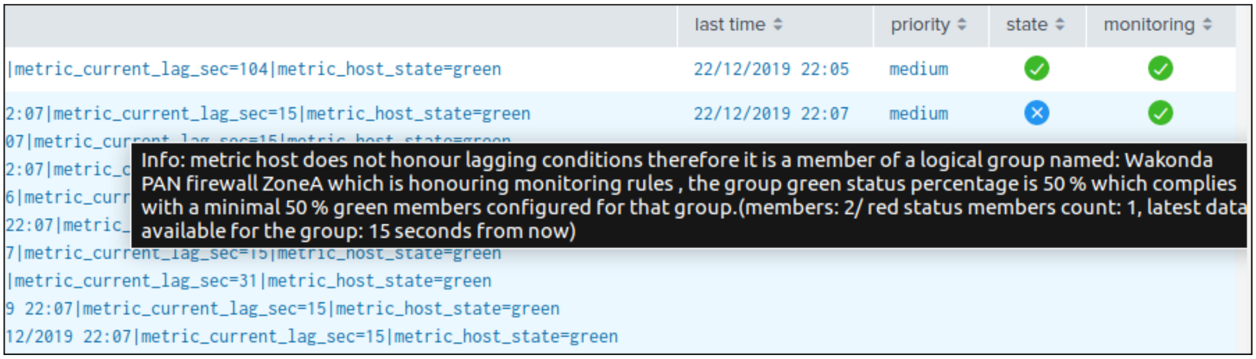
keyid	object_group_name	object_group_members
5dffe73ea8483004bb1c83f1	Wakonda PAN firewall ZoneA	ip-10-0-0-75

- Optionally use the filter input box to search for a logical group
- Click on then logical group entity table, and confirm association to automatically the entity in this logical group

How alerting is handled once the logical group is created with enough members

Member of logical group is red but logical group is green

When an entity is associated to a logical group and if this entity is in red status, but the logical group complies with the monitoring rules, the UI will show a blue icon message which dynamically provides logical group information:



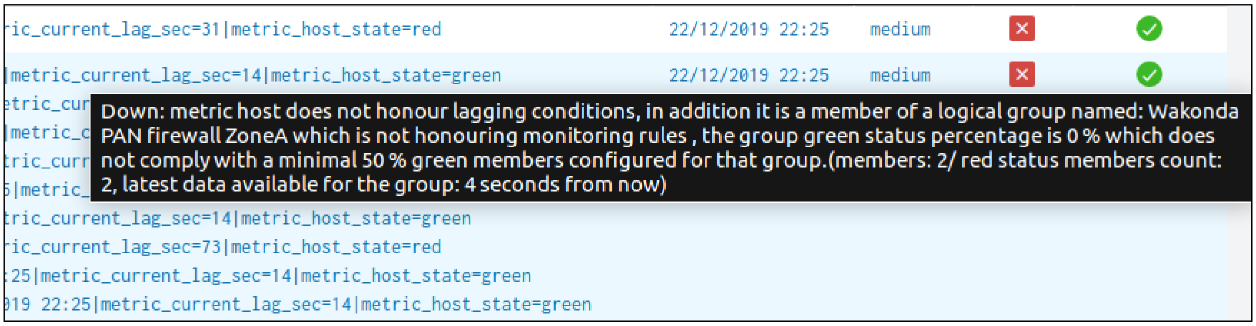
<table>
<tr><th></th><th>last time</th><th>priority</th><th>state</th><th>monitoring</th></tr>
<tr><td>|metric_current_lag_sec=104|metric_host_state=green</td><td>22/12/2019 22:05</td><td>medium</td><td>✓</td><td>✓</td></tr>
<tr><td>2:07|metric_current_lag_sec=15|metric_host_state=green
07|metric_current_lag_sec=15|metric_host_state=green
2:07|metric_c
6|metric_curr
22:07|metric_
7|metric_current_lag_sec=15|metric_host_state=green
|metric_current_lag_sec=31|metric_host_state=green
9 22:07|metric_current_lag_sec=15|metric_host_state=green
12/2019 22:07|metric_current_lag_sec=15|metric_host_state=green</td><td>22/12/2019 22:07</td><td>medium</td><td>✕</td><td>✓</td></tr>
</table>

Info: metric host does not honour lagging conditions therefore it is a member of a logical group named: Wakonda PAN firewall ZoneA which is honouring monitoring rules , the group green status percentage is 50 % which complies with a minimal 50 % green members configured for that group.(members: 2/ red status members count: 1, latest data available for the group: 15 seconds from now)

In addition, the entity will not be eligible to trigger any alert as long as the logical group honours the monitoring rules.(minimal green percentage of the logical group)

Member of logical group is red and logical group is red

When an entity associated to a logical group is red, and the logical group is red as well (for example in a logical group of 2 nodes where both nodes are down), the UI shows the following:



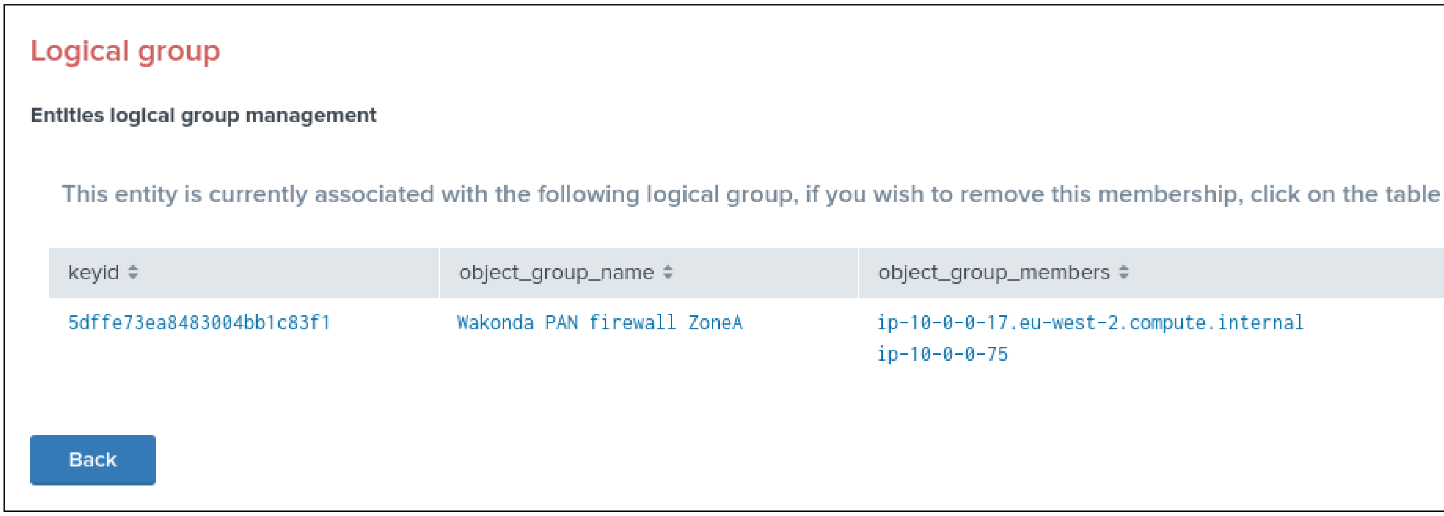
```
ic_current_lag_sec=31|metric_host_state=red                22/12/2019 22:25   medium   ☒   ✓
|metric_current_lag_sec=14|metric_host_state=green         22/12/2019 22:25   medium   ☒   ✓
etric_cur
|metric_c
tric_curr
5|metric_
tric_current_lag_sec=14|metric_host_state=green
ic_current_lag_sec=73|metric_host_state=red
:25|metric_current_lag_sec=14|metric_host_state=green
019 22:25|metric_current_lag_sec=14|metric_host_state=green
```

Down: metric host does not honour lagging conditions, in addition it is a member of a logical group named: Wakonda PAN firewall ZoneA which is not honouring monitoring rules , the group green status percentage is 0 % which does not comply with a minimal 50 % green members configured for that group.(members: 2/ red status members count: 2, latest data available for the group: 4 seconds from now)

Alerts will be generated for any entities part of the logical groups which are in red status, and where the monitoring state is enabled.

Remove association from a logical group

To remove an association from a logical group, click on the entry table in the initial logical group screen for that entity:



Logical group

Entities logical group management

This entity is currently associated with the following logical group, if you wish to remove this membership, click on the table

keyid	object_group_name	object_group_members
5dffe73ea8483004bb1c83f1	Wakonda PAN firewall ZoneA	ip-10-0-0-17.eu-west-2.compute.internal ip-10-0-0-75

Once the action is confirmed, the association is immediately removed and the entity acts as any other independent entities.

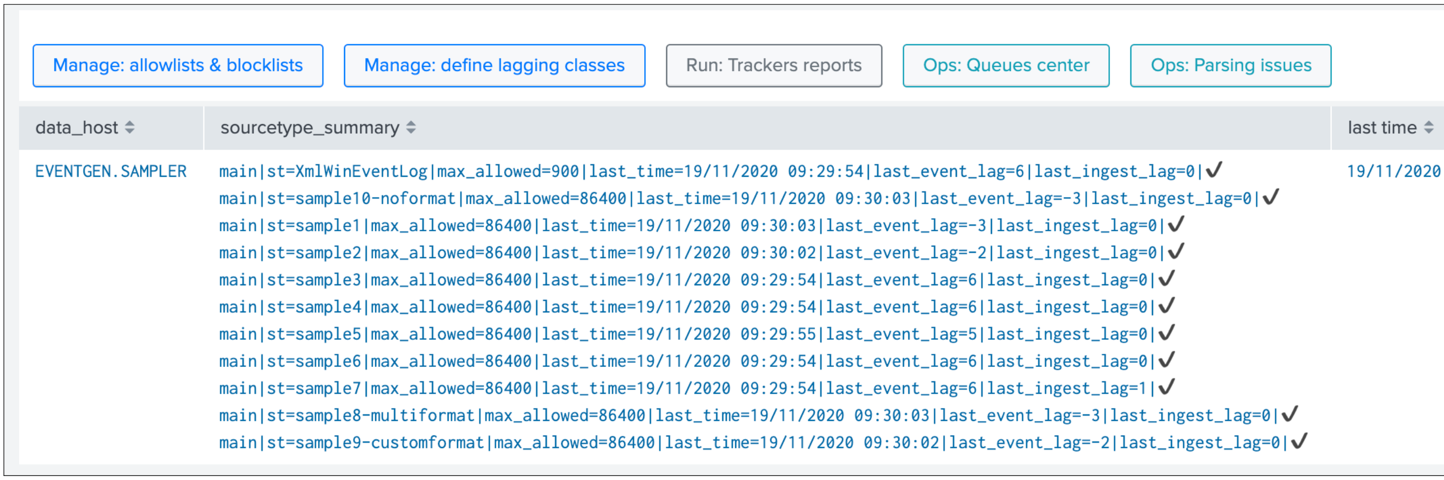
3.1.19 Alerting policy for data hosts

Data hosts alerting policy management

- The alerting policy controls how the state of a data host gets defined depending on the sourcetypes that are emitting data
- The global default mode named “track per host” instructs TrackMe to turn an host to red only if no sourcetypes are being indexed and respecting monitoring rules
- The global alternative mode named “track per sourcetype” instructs TrackMe to consider sourcetypes and their monitoring rules individually on a per host basis, to finally define the overall state of the host
- This global mode can optionally be overriden on a per host basis via the configuration screen of the data host

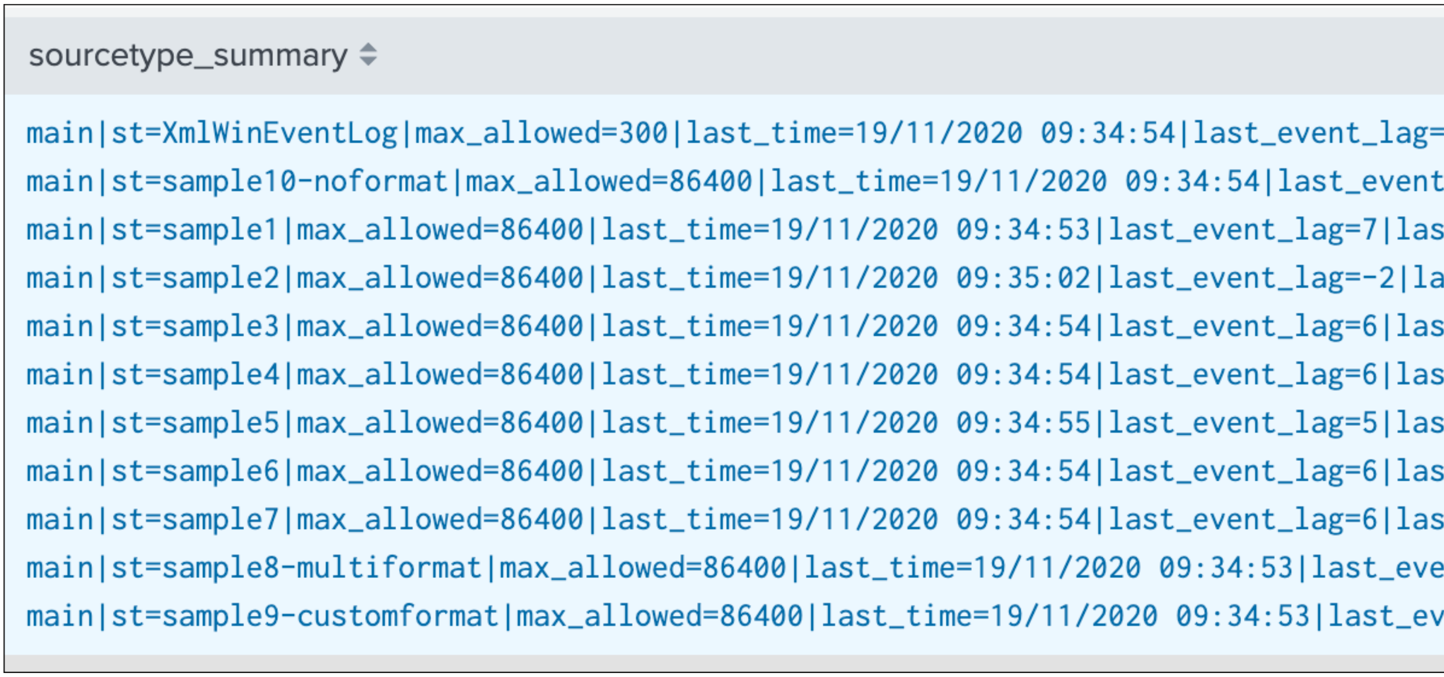
See *Data Hosts alerting policy* to control the global policy settings.

An host emitting multiple sourcetypes will appear in the UI with a multi value summary field describing the state and main information of sourcetypes:


Manage: allowlists & blocklists
Manage: define lagging classes
Run: Trackers reports
Ops: Queues center
Ops: Parsing issues

data_host	sourcetype_summary	last time
EVENTGEN.SAMPLER	main|st=XmlWinEventLog|max_allowed=900|last_time=19/11/2020 09:29:54|last_event_lag=6|last_ingest_lag=0|✓ main|st=sample10-noformat|max_allowed=86400|last_time=19/11/2020 09:30:03|last_event_lag=-3|last_ingest_lag=0|✓ main|st=sample1|max_allowed=86400|last_time=19/11/2020 09:30:03|last_event_lag=-3|last_ingest_lag=0|✓ main|st=sample2|max_allowed=86400|last_time=19/11/2020 09:30:02|last_event_lag=-2|last_ingest_lag=0|✓ main|st=sample3|max_allowed=86400|last_time=19/11/2020 09:29:54|last_event_lag=6|last_ingest_lag=0|✓ main|st=sample4|max_allowed=86400|last_time=19/11/2020 09:29:54|last_event_lag=6|last_ingest_lag=0|✓ main|st=sample5|max_allowed=86400|last_time=19/11/2020 09:29:55|last_event_lag=5|last_ingest_lag=0|✓ main|st=sample6|max_allowed=86400|last_time=19/11/2020 09:29:54|last_event_lag=6|last_ingest_lag=0|✓ main|st=sample7|max_allowed=86400|last_time=19/11/2020 09:29:54|last_event_lag=6|last_ingest_lag=1|✓ main|st=sample8-multiformat|max_allowed=86400|last_time=19/11/2020 09:30:03|last_event_lag=-3|last_ingest_lag=0|✓ main|st=sample9-customformat|max_allowed=86400|last_time=19/11/2020 09:30:02|last_event_lag=-2|last_ingest_lag=0|✓	19/11/2020

Zooming on the summary sourcetype field:



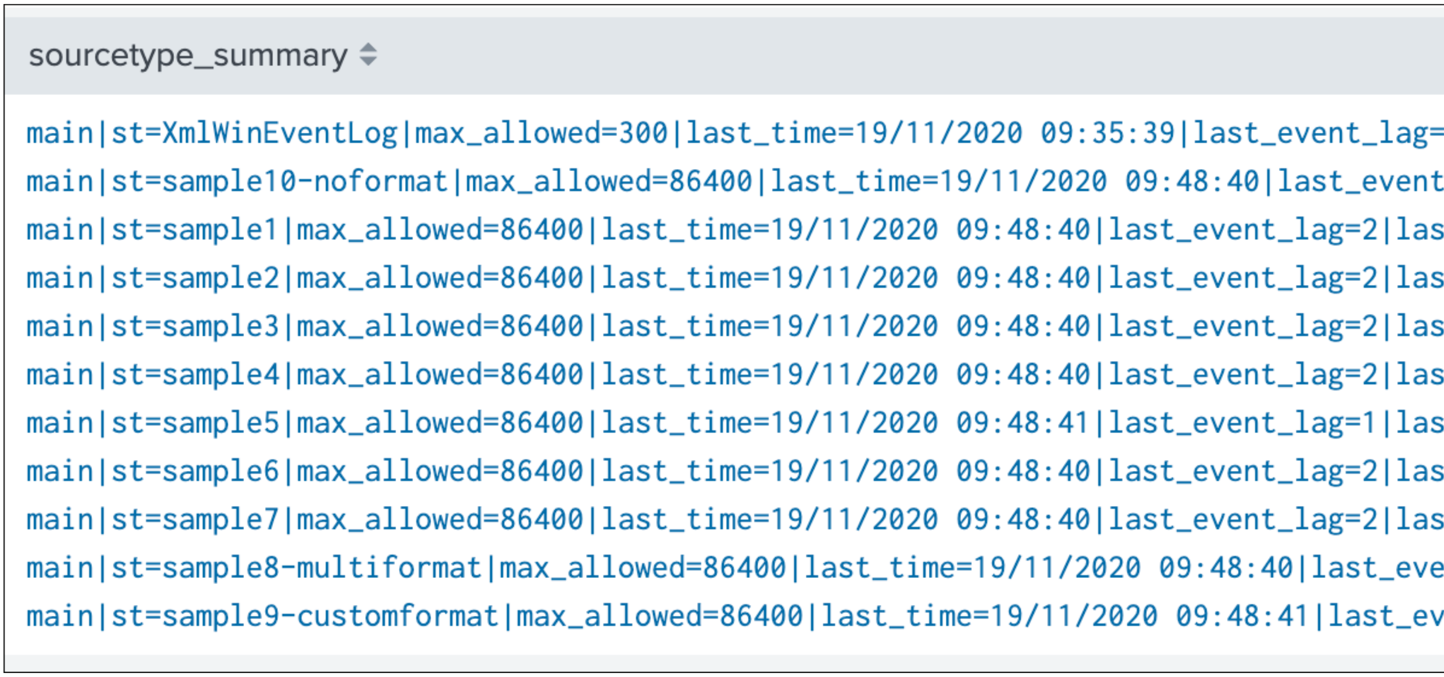
sourcetype_summary
main|st=XmlWinEventLog|max_allowed=300|last_time=19/11/2020 09:34:54|last_event_lag= main|st=sample10-noformat|max_allowed=86400|last_time=19/11/2020 09:34:54|last_event main|st=sample1|max_allowed=86400|last_time=19/11/2020 09:34:53|last_event_lag=7|las main|st=sample2|max_allowed=86400|last_time=19/11/2020 09:35:02|last_event_lag=-2|la main|st=sample3|max_allowed=86400|last_time=19/11/2020 09:34:54|last_event_lag=6|las main|st=sample4|max_allowed=86400|last_time=19/11/2020 09:34:54|last_event_lag=6|las main|st=sample5|max_allowed=86400|last_time=19/11/2020 09:34:55|last_event_lag=5|las main|st=sample6|max_allowed=86400|last_time=19/11/2020 09:34:54|last_event_lag=6|las main|st=sample7|max_allowed=86400|last_time=19/11/2020 09:34:54|last_event_lag=6|las main|st=sample8-multiformat|max_allowed=86400|last_time=19/11/2020 09:34:53|last_eve main|st=sample9-customformat|max_allowed=86400|last_time=19/11/2020 09:34:53|last_ev

The field provides visibility against each sourcetype known to the host, a main state (red / green) represented by an ASCII emoji and the KPI main information about the sourcetypes:

- `max_allowed`: the maximal laggging value allowed for this sourcetype according to the monitoring rules (lagging classes, default lagging)
- `last_time`: A human readable format of the latest events available for that host from the event timestamp point of view (_time)
- `last_event_lag`: The current event lag value in seconds (difference between now and the latest _time available for this host/sourcetype)
- `last_ingest_lag`: The current indexing lag value in seconds (difference between the event timestamp and the indexing time)

- `state`: for readability purposes, the state green/red is represented as an ASCII emoji

Should any sourcetype not being indexed or not respecting the monitoring rules, the state icon will turn red:



sourcetype_summary

```
main|st=XmlWinEventLog|max_allowed=300|last_time=19/11/2020 09:35:39|last_event_lag=
main|st=sample10-noformat|max_allowed=86400|last_time=19/11/2020 09:48:40|last_event
main|st=sample1|max_allowed=86400|last_time=19/11/2020 09:48:40|last_event_lag=2|las
main|st=sample2|max_allowed=86400|last_time=19/11/2020 09:48:40|last_event_lag=2|las
main|st=sample3|max_allowed=86400|last_time=19/11/2020 09:48:40|last_event_lag=2|las
main|st=sample4|max_allowed=86400|last_time=19/11/2020 09:48:40|last_event_lag=2|las
main|st=sample5|max_allowed=86400|last_time=19/11/2020 09:48:41|last_event_lag=1|las
main|st=sample6|max_allowed=86400|last_time=19/11/2020 09:48:40|last_event_lag=2|las
main|st=sample7|max_allowed=86400|last_time=19/11/2020 09:48:40|last_event_lag=2|las
main|st=sample8-multiformat|max_allowed=86400|last_time=19/11/2020 09:48:40|last_eve
main|st=sample9-customformat|max_allowed=86400|last_time=19/11/2020 09:48:41|last_ev
```

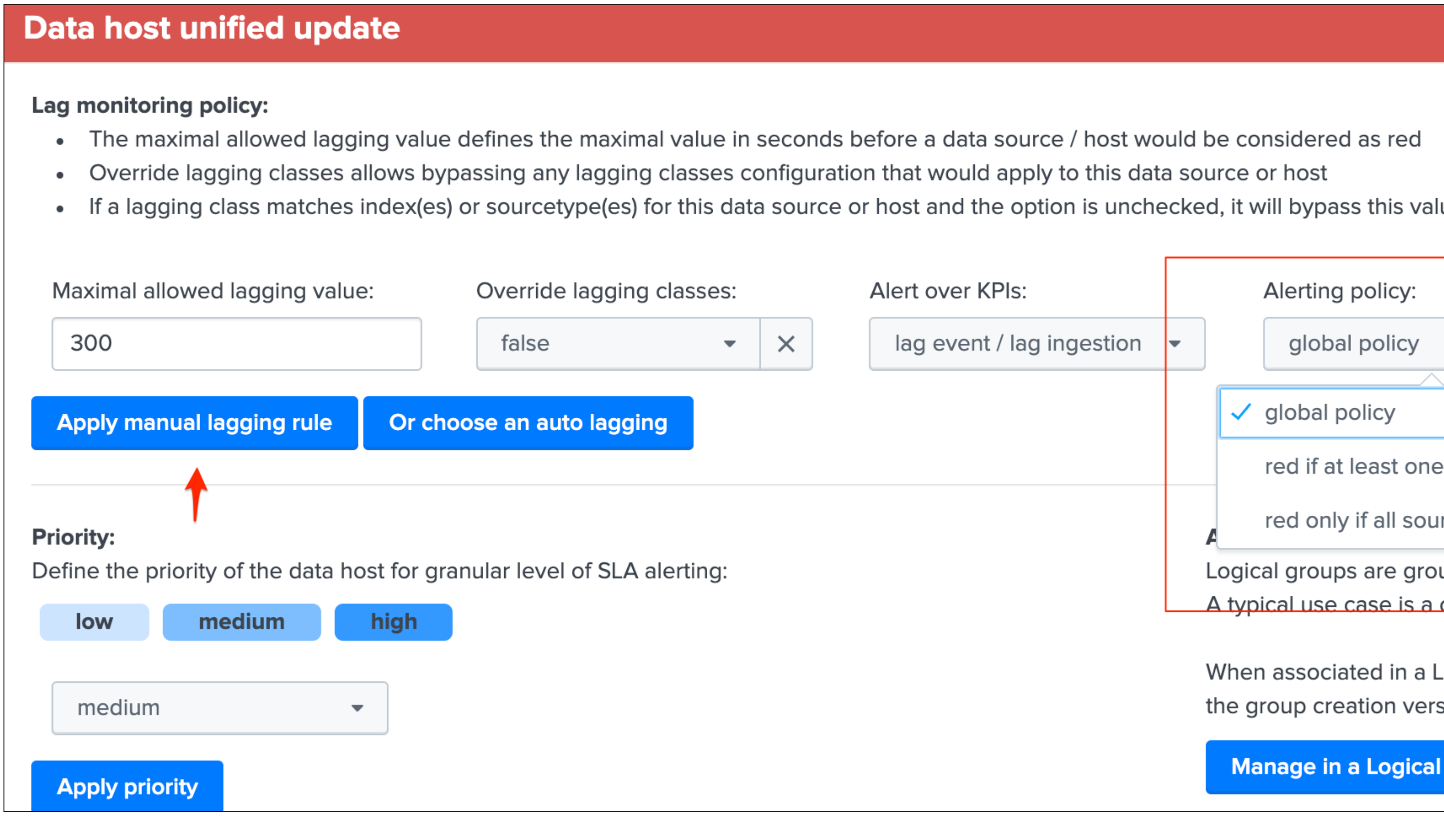
Hint: If a sourcetypes turns `red`, this will NOT impact the state of the host unless the global policy is set to `track per sourcetype`, or the host policy is defined for that host especially

To configure sourcetypes to be taken into account individually, you can either:

- Define the global policy accordingly (note: this applies by default to all hosts), See *Data Hosts alerting policy*
- Define the alerting policy for that host especially in the data host configuration screen

Defining a policy per host:

In the data host UI, click on the modify button to access to the alerting policy dropdown:



Data host unified update

Lag monitoring policy:

- The maximal allowed lagging value defines the maximal value in seconds before a data source / host would be considered as red
- Override lagging classes allows bypassing any lagging classes configuration that would apply to this data source or host
- If a lagging class matches index(es) or sourcetype(es) for this data source or host and the option is unchecked, it will bypass this valu

Maximal allowed lagging value: 300

Override lagging classes: false

Alert over KPIs: lag event / lag ingestion

Alerting policy: global policy

Priority:

Define the priority of the data host for granular level of SLA alerting:

medium

A

Logical groups are grou
A typical use case is a c

When associated in a L
the group creation vers

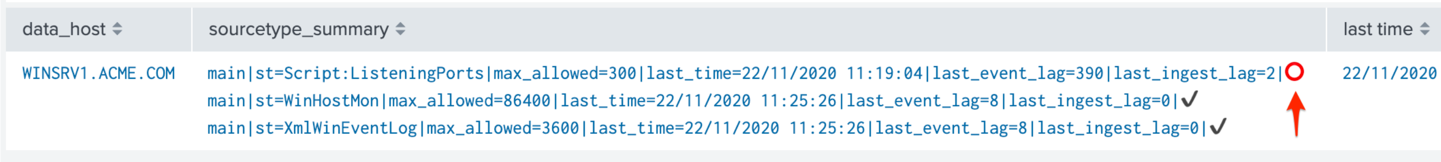
Three options are available:

- `global policy`: instructs the data host settings to rely on the global alerting policy
- `red if at least one sourcetype is red`: instructs TrackMe to turn the host red if at least one sourcetype is in a red state (track per sourcetype)
- `red only if all sourcetypes are red`: instructs TrackMe to turn the host red only if none of the sourcetypes are respecting monitoring rules (track per host)

When a mode is defined for a given host that is not equal to the global policy, then the global alerting policy is ignored and replaced by the setting defined for that host.

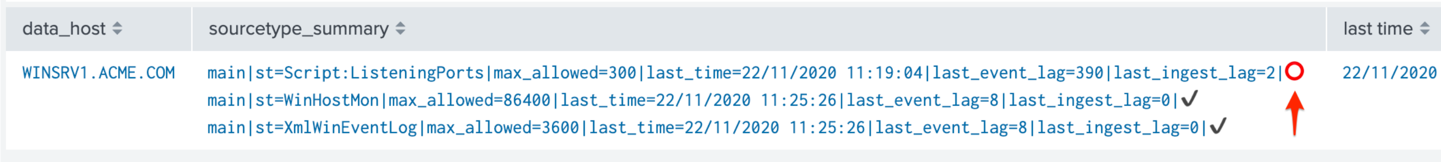
Behaviour examples:

Alerting policy track per sourcetype:



data_host	sourcetype_summary	last time
WINSRV1.ACME.COM	main|st=Script:ListeningPorts|max_allowed=300|last_time=22/11/2020 11:19:04|last_event_lag=390|last_ingest_lag=2|○ main|st=WinHostMon|max_allowed=86400|last_time=22/11/2020 11:25:26|last_event_lag=8|last_ingest_lag=0|✓ main|st=XmlWinEventLog|max_allowed=3600|last_time=22/11/2020 11:25:26|last_event_lag=8|last_ingest_lag=0|✓	22/11/2020

Alerting policy track per host:



data_host	sourcetype_summary	last time
WINSRV1.ACME.COM	main|st=Script:ListeningPorts|max_allowed=300|last_time=22/11/2020 11:19:04|last_event_lag=390|last_ingest_lag=2|○ main|st=WinHostMon|max_allowed=86400|last_time=22/11/2020 11:25:26|last_event_lag=8|last_ingest_lag=0|✓ main|st=XmlWinEventLog|max_allowed=3600|last_time=22/11/2020 11:25:26|last_event_lag=8|last_ingest_lag=0|✓	22/11/2020

3.1.20 Tags

Tags feature

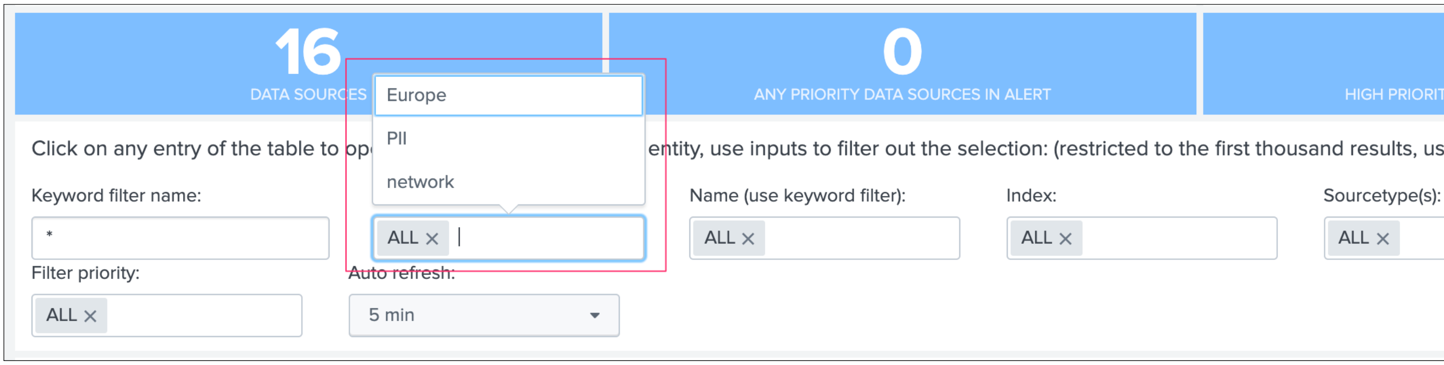
- Tags are keywords that can be defined per data source, this feature provides additional filtering options to group multiple data sources based on any custom criterias.
- Tags are available for data sources monitoring only.

Tags can be defined using:

- Tags policies, which are regular expressions rules that you can define to automatically apply tags conditionally
- Manual tags, which you can define manually via the Tags UI on a per data source basis

Tags feature purpose:

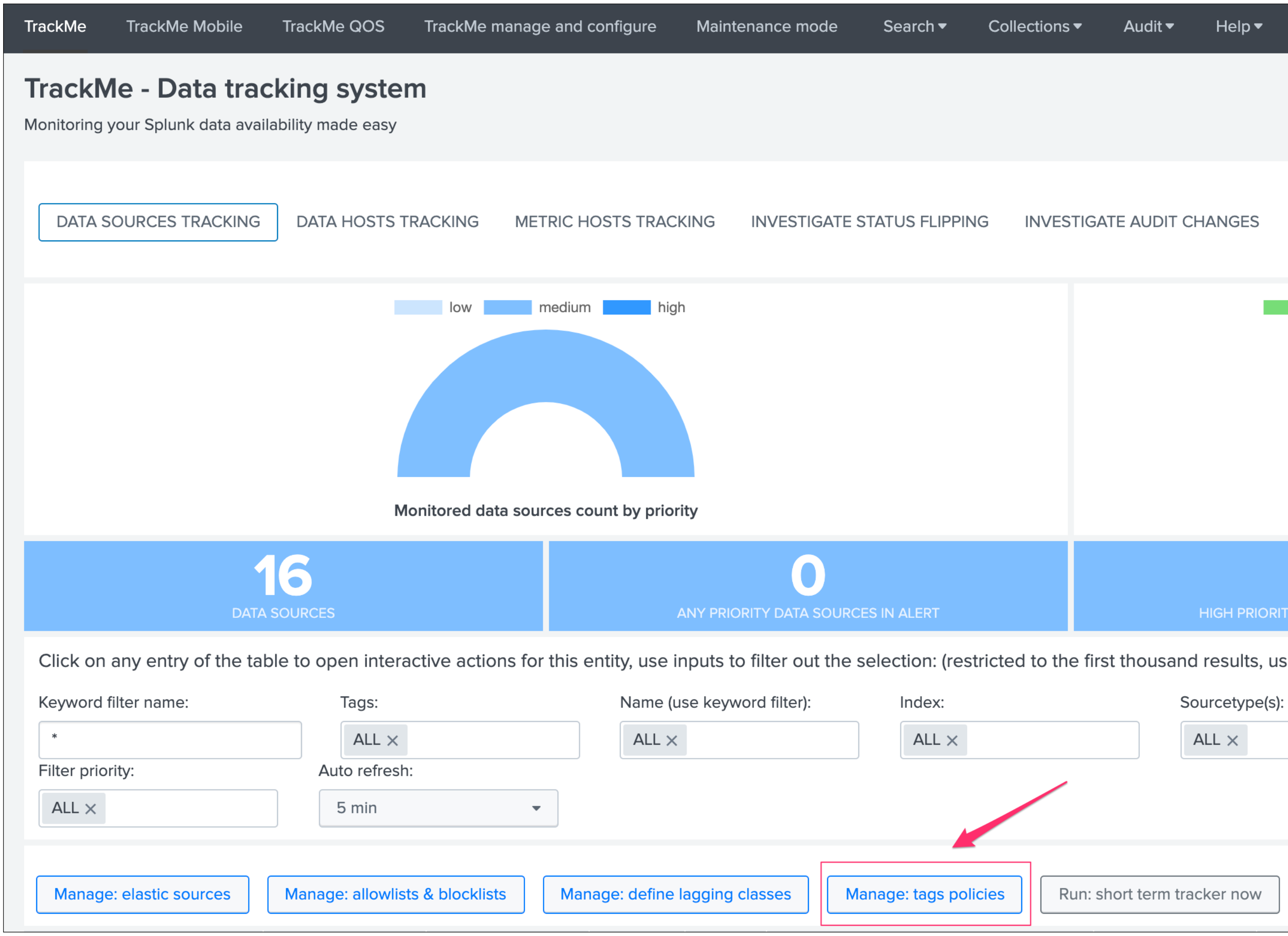
For instance, you may want to tag data sources containing PII data, such that data sources matching this criteria can be filtered on easily in the main TrackMe UI:



Click on any entry of the table to op entity, use inputs to filter out the selection: (restricted to the first thousand results, us

Tags policies

The tags policies editor can be opened via the data sources main screen tab, and the button Tags policies:


TrackMe
TrackMe Mobile
TrackMe QOS
TrackMe manage and configure
Maintenance mode
Search ▾
Collections ▾
Audit ▾
Help ▾
DATA SOURCES TRACKING
DATA HOSTS TRACKING
METRIC HOSTS TRACKING
INVESTIGATE STATUS FLIPPING
INVESTIGATE AUDIT CHANGES
low
medium
high
Monitored data sources count by priority
16
DATA SOURCES
0
ANY PRIORITY DATA SOURCES IN ALERT
HIGH PRIORIT
Manage: elastic sources
Manage: allowlists & blocklists
Manage: define lagging classes
Manage: tags policies
Run: short term tracker now

TrackMe - Data tracking system

Monitoring your Splunk data availability made easy

Click on any entry of the table to open interactive actions for this entity, use inputs to filter out the selection: (restricted to the first thousand results, us

Keyword filter name: *

Tags: ALL ×

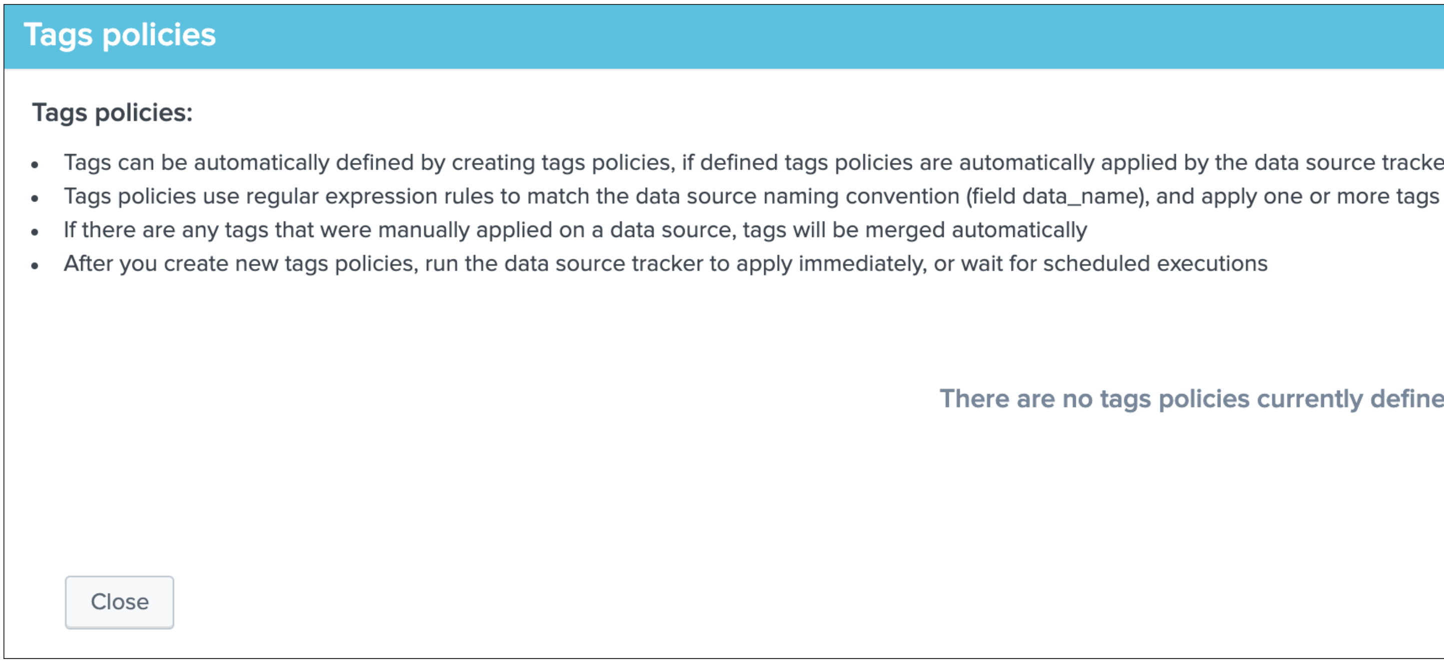
Name (use keyword filter): ALL ×

Index: ALL ×

Sourcetype(s): ALL ×

Filter priority: ALL ×

Auto refresh: 5 min



Tags policies

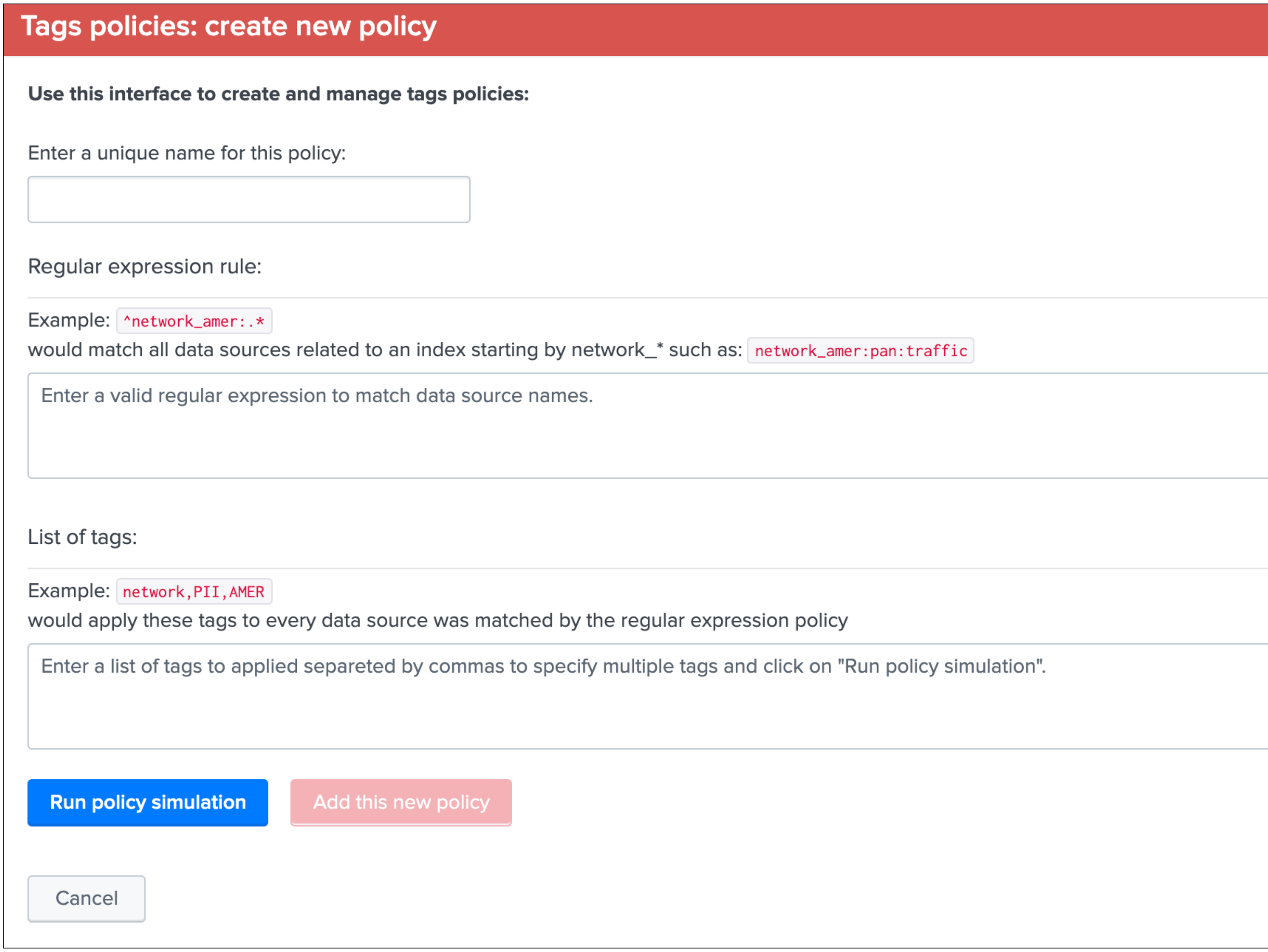
Tags policies:

- Tags can be automatically defined by creating tags policies, if defined tags policies are automatically applied by the data source tracke
- Tags policies use regular expression rules to match the data source naming convention (field data_name), and apply one or more tags
- If there are any tags that were manually applied on a data source, tags will be merged automatically
- After you create new tags policies, run the data source tracker to apply immediately, or wait for scheduled executions

There are no tags policies currently define

Create a new tags policy

To create a new tags policy, click on the Create policy button:



Tags policies: create new policy

Use this interface to create and manage tags policies:

Enter a unique name for this policy:

Regular expression rule:

Example: `^network_amer:.*`
would match all data sources related to an index starting by network_* such as: `network_amer:pan:traffic`

Enter a valid regular expression to match data source names.

List of tags:

Example: `network,PII,AMER`
would apply these tags to every data source was matched by the regular expression policy

Enter a list of tags to applied separeted by commas to specify multiple tags and click on "Run policy simulation".

Fill the UI with the required information:

- **Enter a unique name for this policy:** this id will be used and stored as the value for the field tags_policy_id in the KVstore collection
- **Regular expression rule:** this is the regular expression that will be used to conditionally apply the tags against the data_name field for every data source
- **List of tags:** the tags to be applied when the regular expression matches, multiple tags can be specified in a comma separated fashion

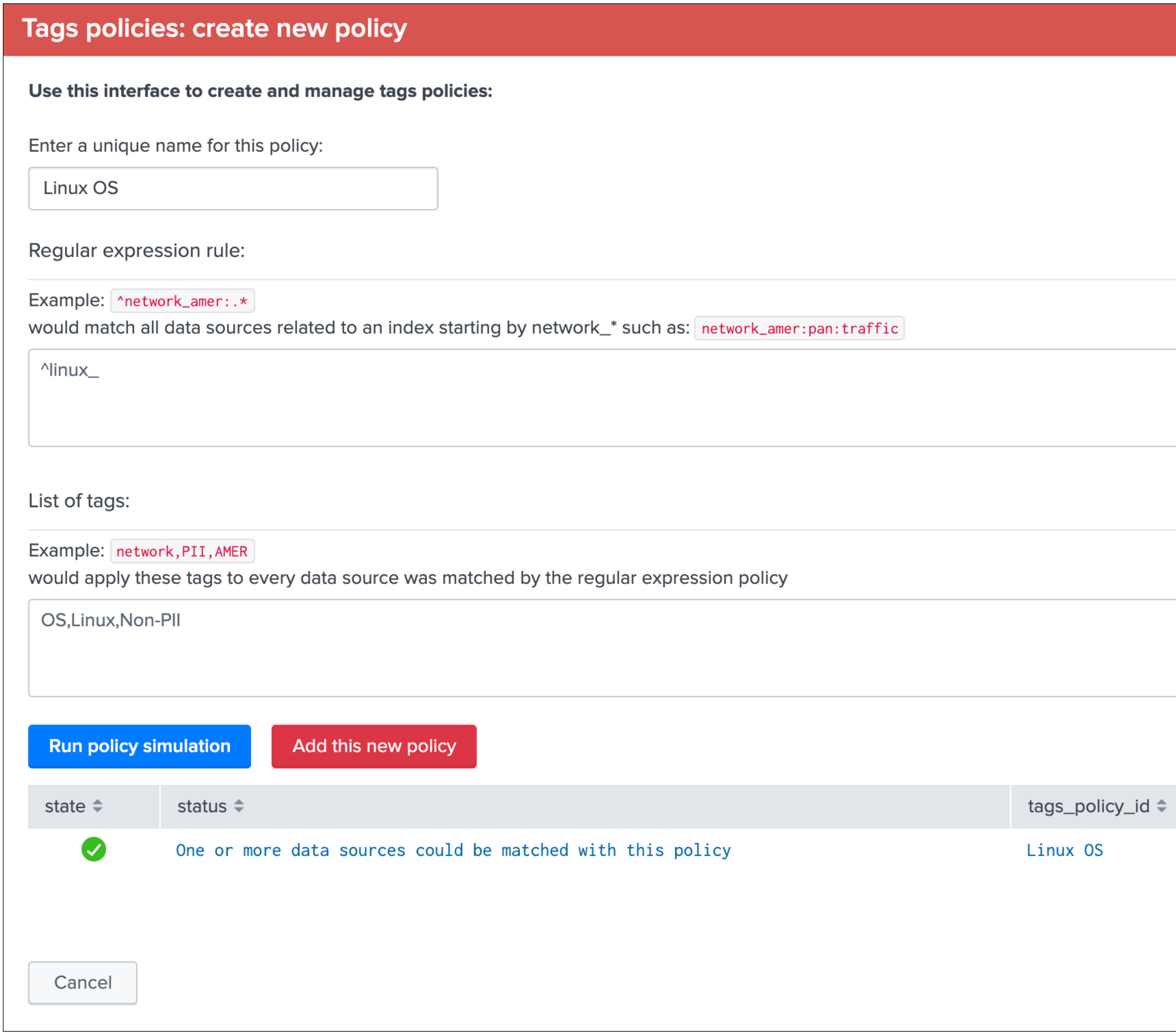
Tags policies are applied sequentially in the order the entries are stored in the KVstore collection, should a regular expression match, the execution for this specific data source stops at the first match.

Example:

- Assuming you have a naming convention for indexes, where all indexes starting by “linux_” contain OS logs of Linux based OS

- Automatically, the following tags will be defined for every data source that matches the regular expression rule, “OS,Linux,Non-PII”

The following policy would be defined:



Tags policies: create new policy

Use this interface to create and manage tags policies:

Enter a unique name for this policy:

Linux OS

Regular expression rule:

Example: `^network_amer:.*`
would match all data sources related to an index starting by network_* such as: `network_amer:pan:traffic`

^linux_

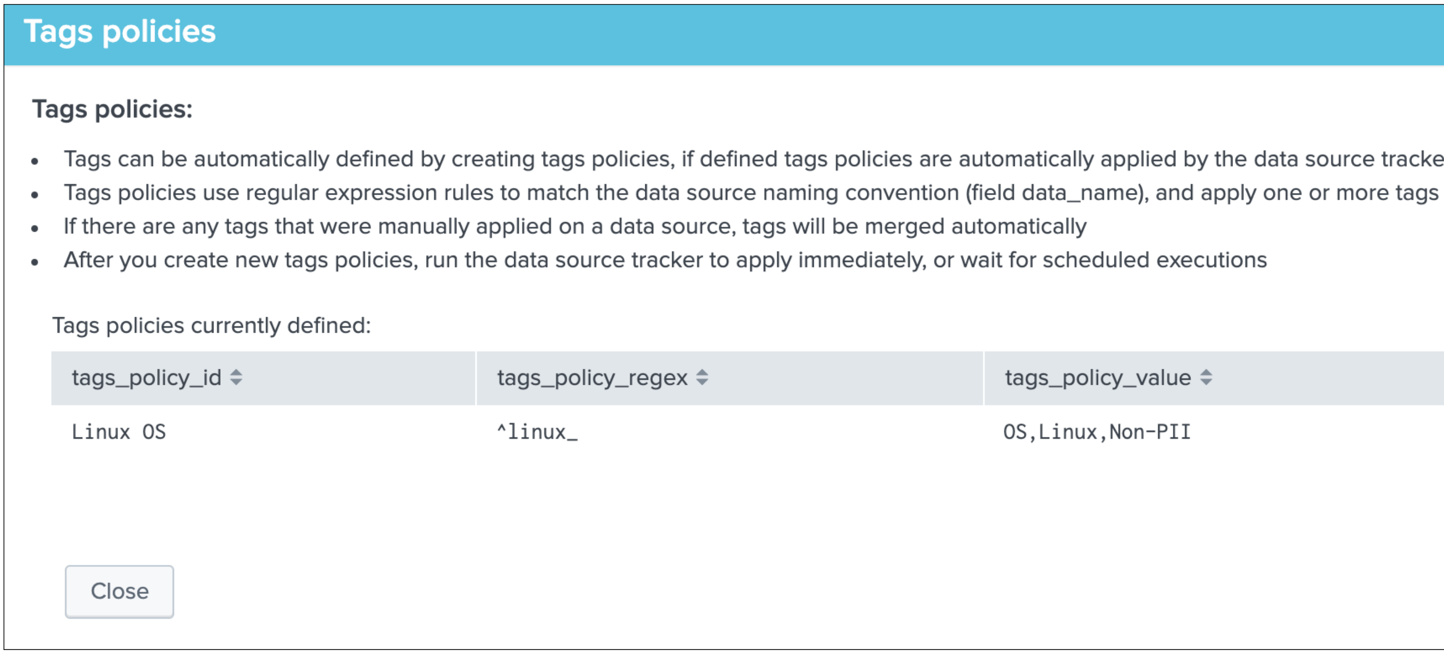
List of tags:

Example: `network,PII,AMER`
would apply these tags to every data source was matched by the regular expression policy

OS,Linux,Non-PII

state	status	tags_policy_id
✓	One or more data sources could be matched with this policy	Linux OS

Once the simulation was executed, click on the red button “Add this new policy”:



Tags policies

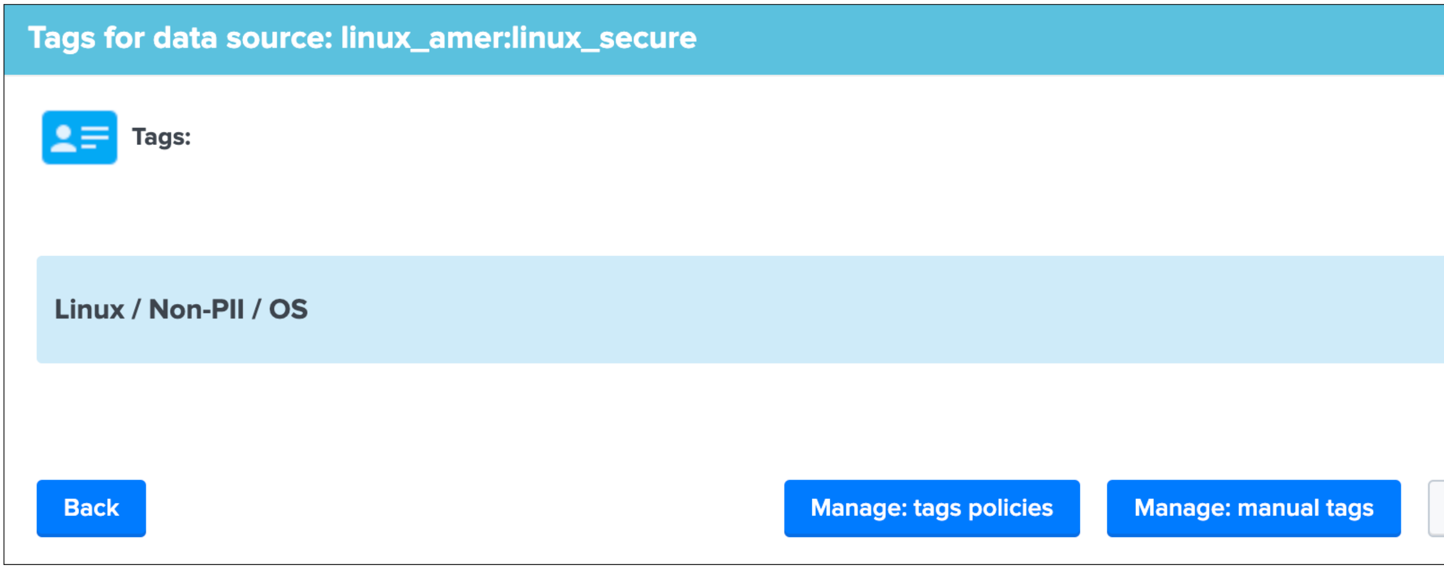
Tags policies:

- Tags can be automatically defined by creating tags policies, if defined tags policies are automatically applied by the data source tracke
- Tags policies use regular expression rules to match the data source naming convention (field data_name), and apply one or more tags
- If there are any tags that were manually applied on a data source, tags will be merged automatically
- After you create new tags policies, run the data source tracker to apply immediately, or wait for scheduled executions

Tags policies currently defined:

tags_policy_id	tags_policy_regex	tags_policy_value
Linux OS	^linux_	OS,Linux,Non-PII

Tags policies are applied automatically by the data source trackers, you can wait for scheduled executions or manually run the tracker (short term or long term, or both) to immediately assign the tags:



Tags for data source: linux_amer:linux_secure

Tags:

Linux / Non-PII / OS

Tag policy multiple matching

Tag policies are based on regular expressions, you can match multiple cases in a single policy relying on regex capabilities.

Say you want to match entities:

- containing “network” at the beginning of the data source name
- containing “firewall” at the beginning of the data source name
- containing “proxy” at the beginning of the data source name

A very simple regular expression could be:

```
^(network|firewall?proxy).*
```

Which you can complete with as many conditions as needed.

You can obvisouly be even more specific, say we want to match:

- entities that are starting by "linux_" as the index prefix
- in these entities, only those matching either "amer", "emea" or "apac"
- terminate properly the entities naming convention, such that there can be no risk of unexpectly matching other entities

Our entities look like: (note that in this example we use the merging mode, therefore all entities are suffixed by ":all")

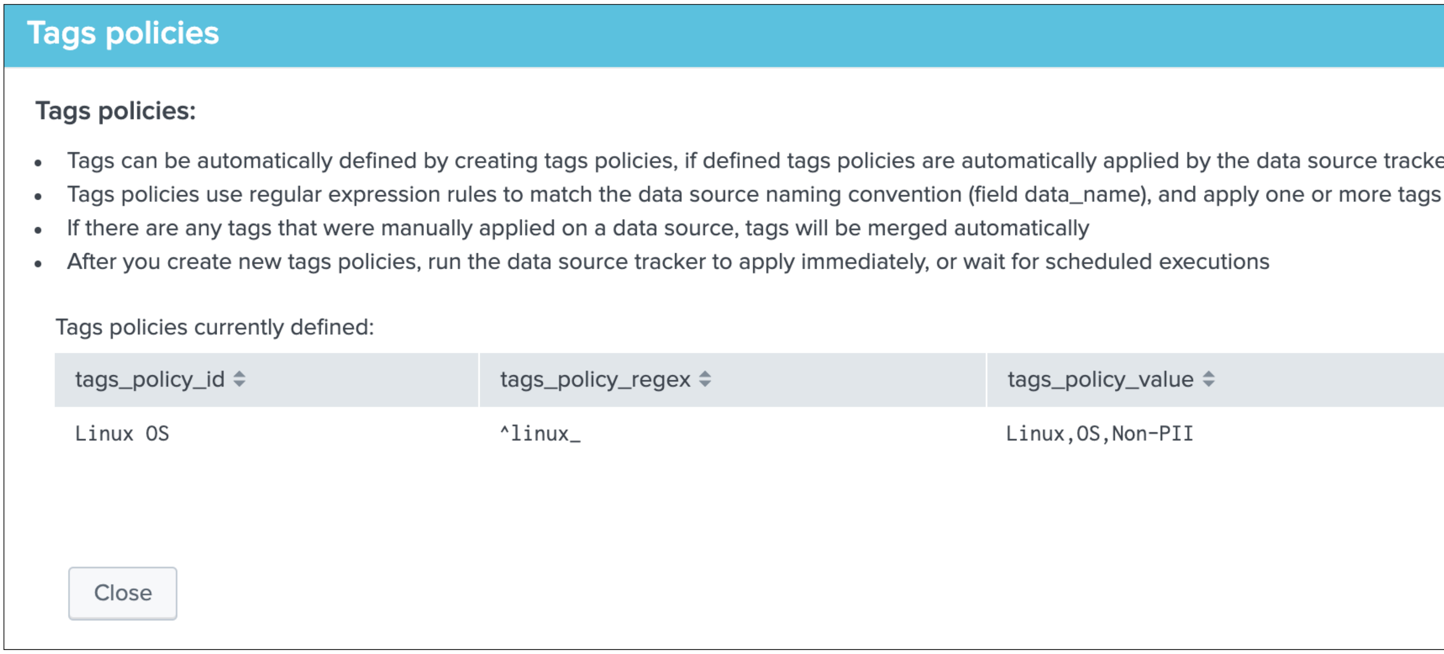
- "linux_amer:all"
- "linux_emea:all"
- "linux_apac:all"

Our strict matching tag policy regular expression could be:

```
^linux_(amer|apac|emea):all$
```

Update and delete tags policies

You cannot update tags policies via the UI, if you need to change a tags policy, you have to delete and re-create the policy using the UI:



Tags policies

Tags policies:

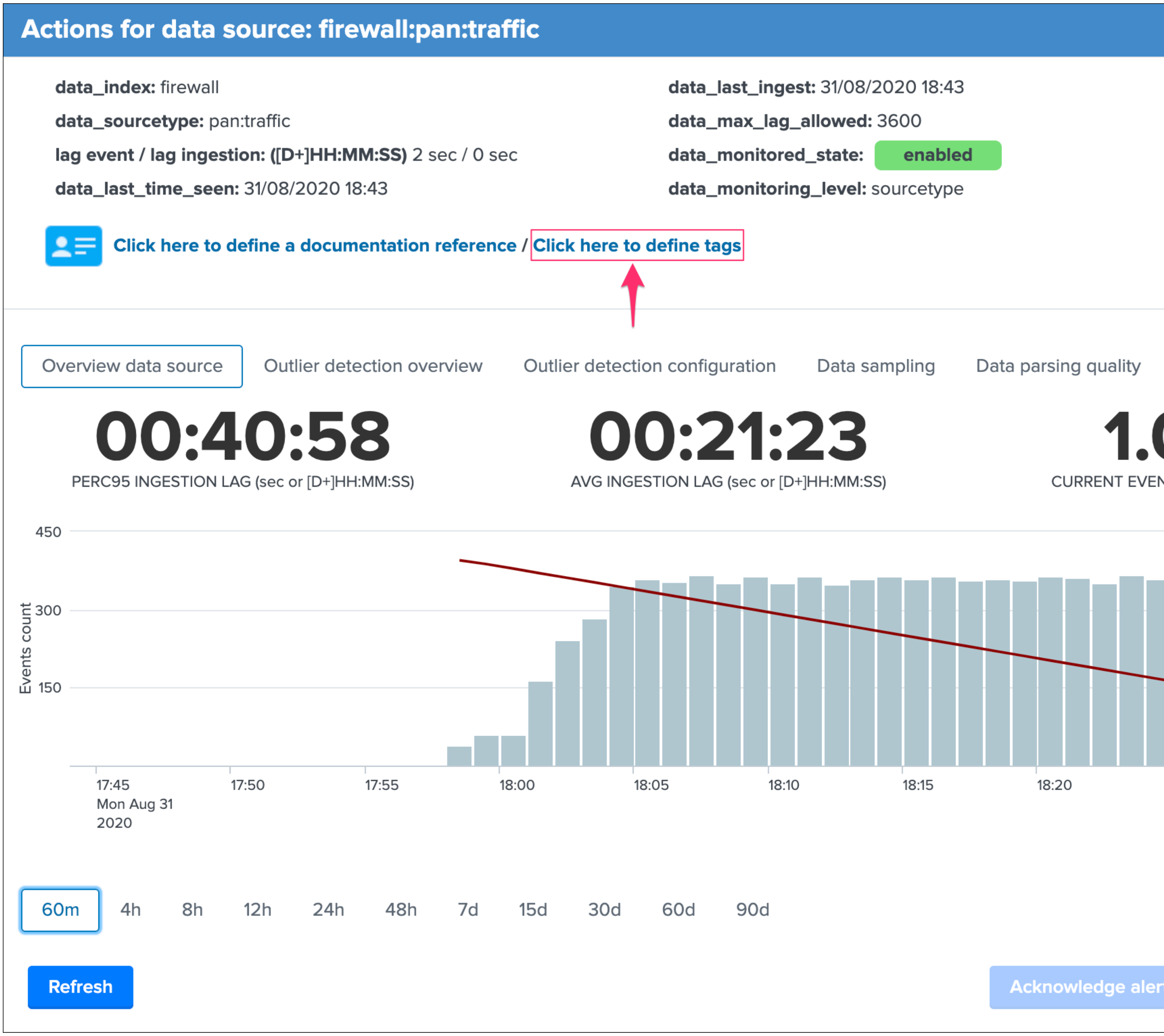
- Tags can be automatically defined by creating tags policies, if defined tags policies are automatically applied by the data source tracke
- Tags policies use regular expression rules to match the data source naming convention (field data_name), and apply one or more tags
- If there are any tags that were manually applied on a data source, tags will be merged automatically
- After you create new tags policies, run the data source tracker to apply immediately, or wait for scheduled executions

Tags policies currently defined:

tags_policy_id	tags_policy_regex	tags_policy_value
Linux OS	^linux_	Linux,OS,Non-PII

Manual tags

Manual tags are available per data source, and allows manually defining a list of tags via the UI:



Actions for data source: firewall:pan:traffic

data_index: firewall

data_sourcetype: pan:traffic

lag event / lag ingestion: ([D+]HH:MM:SS) 2 sec / 0 sec

data_last_time_seen: 31/08/2020 18:43

data_last_ingest: 31/08/2020 18:43

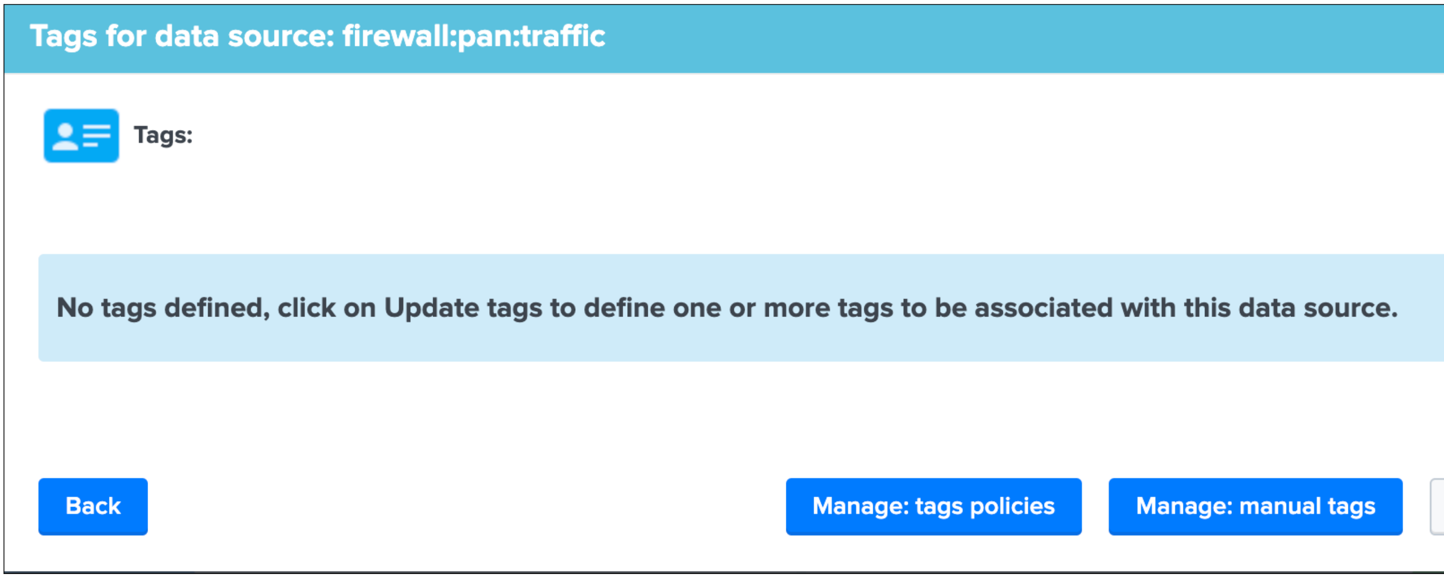
data_max_lag_allowed: 3600

data_monitored_state: enabled

data_monitoring_level: sourcetype

Click here to define a documentation reference / Click here to define tags

When no tags have been defined yet for a data source, the following screen would appear:

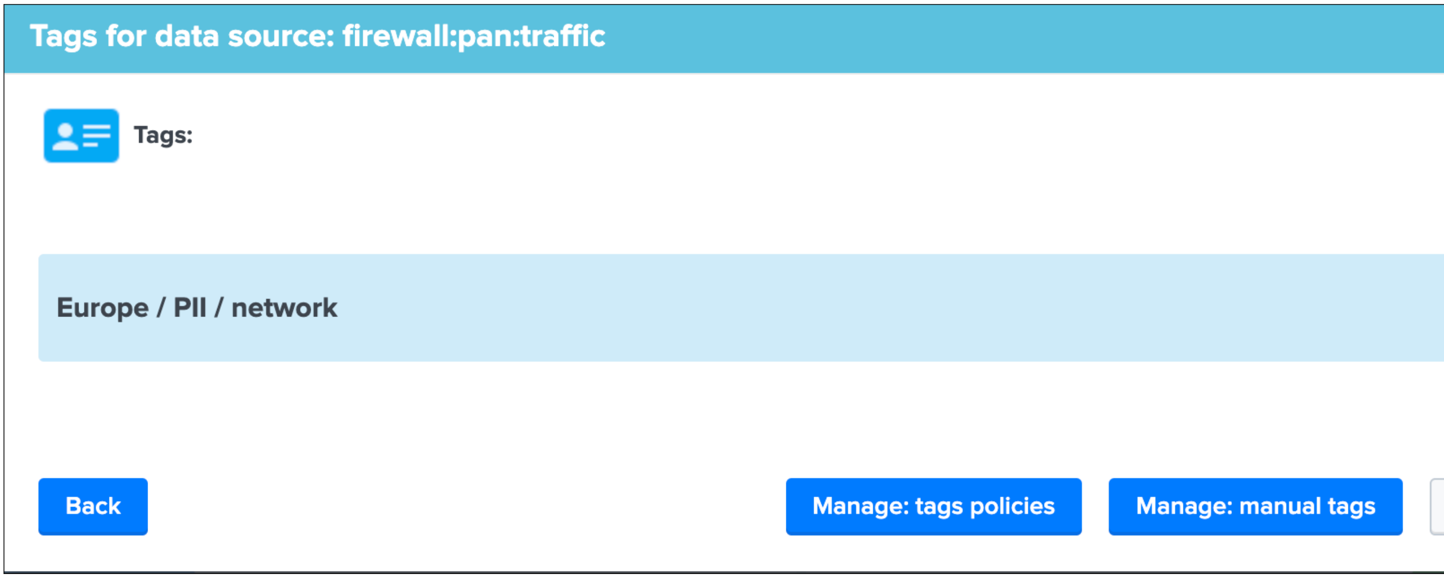


Tags for data source: firewall:pan:traffic

Tags:

No tags defined, click on Update tags to define one or more tags to be associated with this data source.

When tags have been defined for a data source, the following screen would appear:

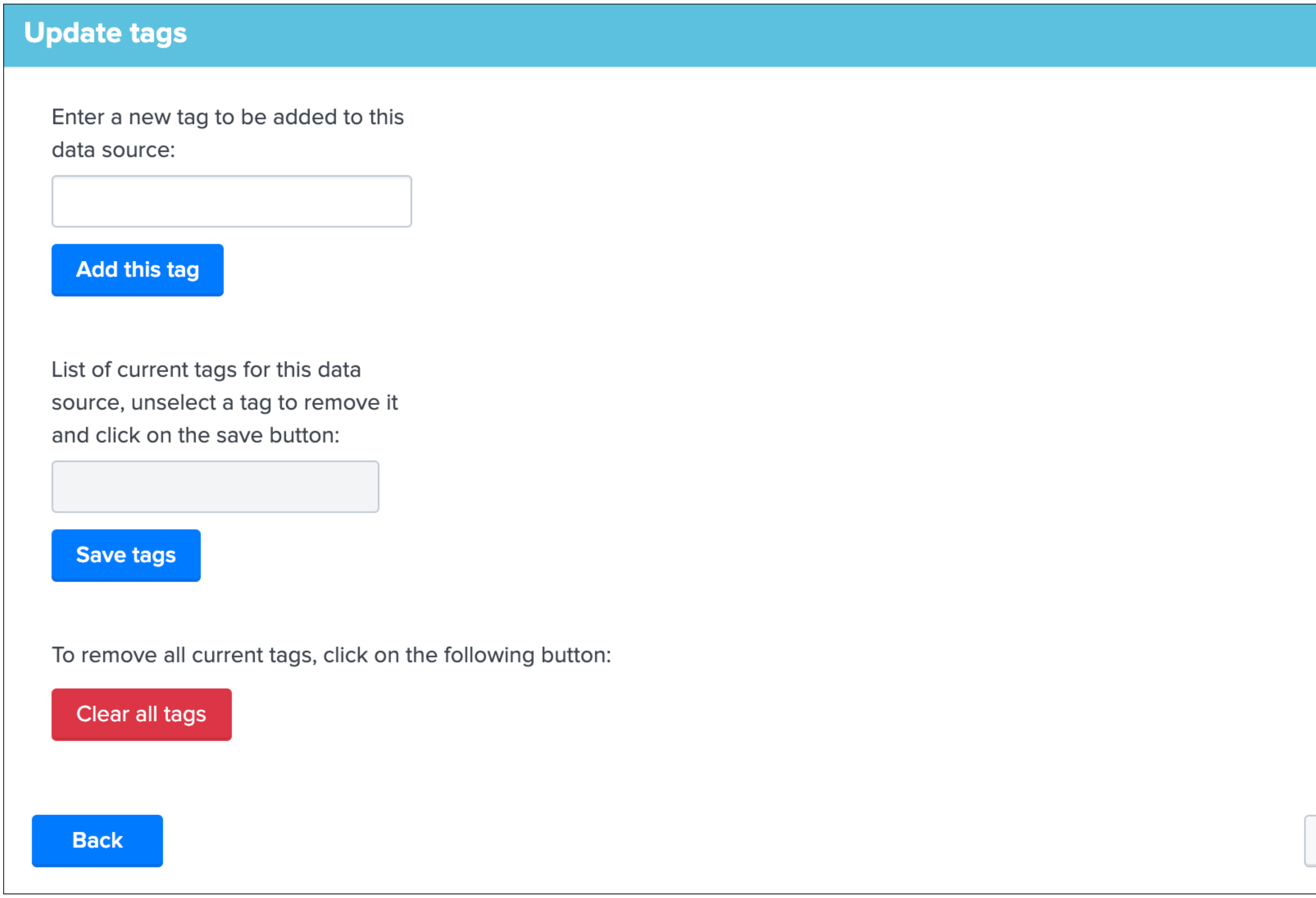


Tags for data source: firewall:pan:traffic

Tags:

Europe / PII / network

You can click on the "Manage: manual tags" button to define one or more tags for a given data source:



Update tags

Enter a new tag to be added to this data source:

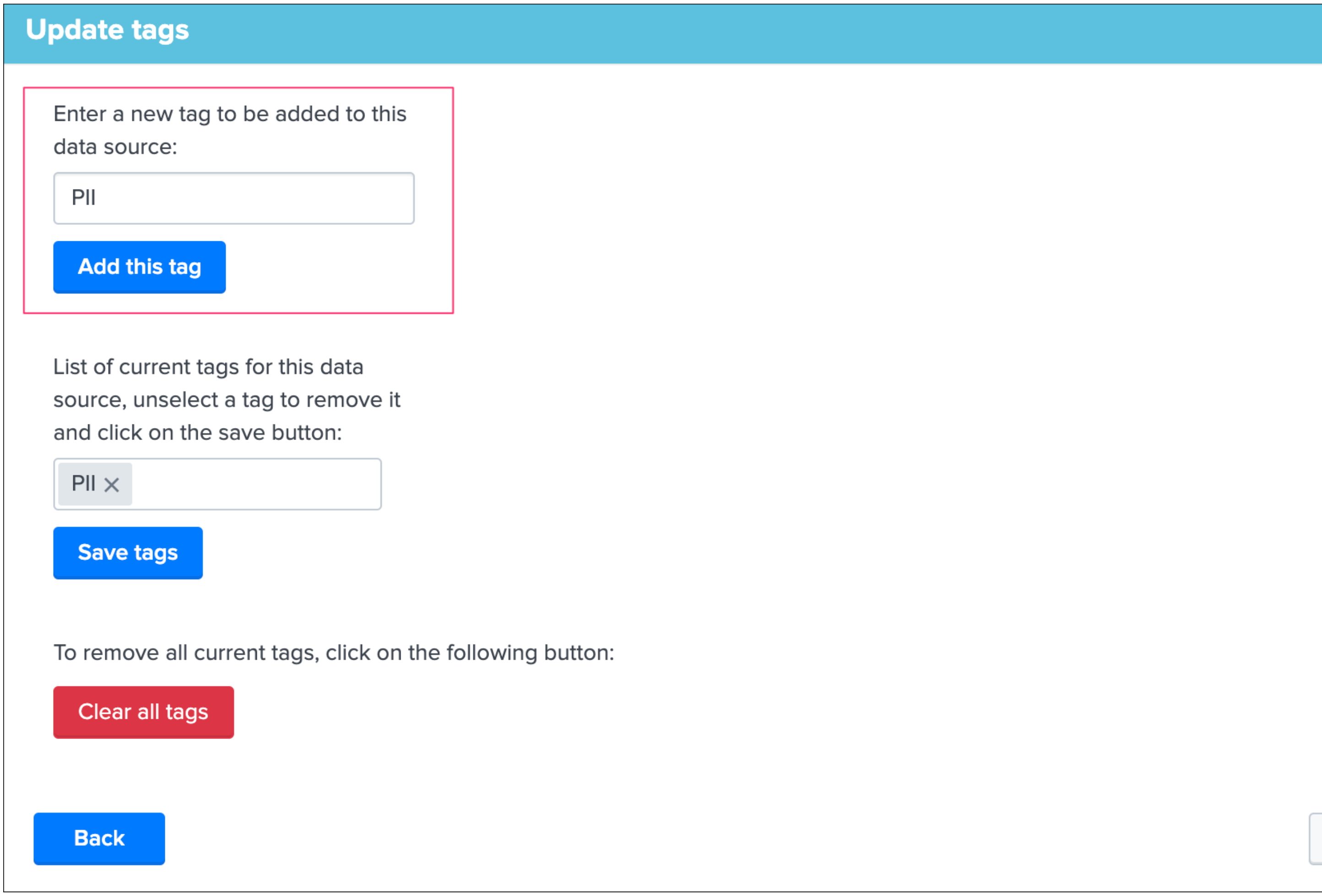
List of current tags for this data source, unselect a tag to remove it and click on the save button:

To remove all current tags, click on the following button:

Tags are stored in the data sources KVstore collection in a field called "tags", when multiple tags are defined, the list of tags is defined as a comma separated list of values.

Adding new tags

You can add a new tag by using the Add tag input and button, the tag format is free, can contain spaces or special characters, however for reliability purposes you should keep things clear and simple.



Update tags

Enter a new tag to be added to this data source:

PII

List of current tags for this data source, unselect a tag to remove it and click on the save button:

PII ×

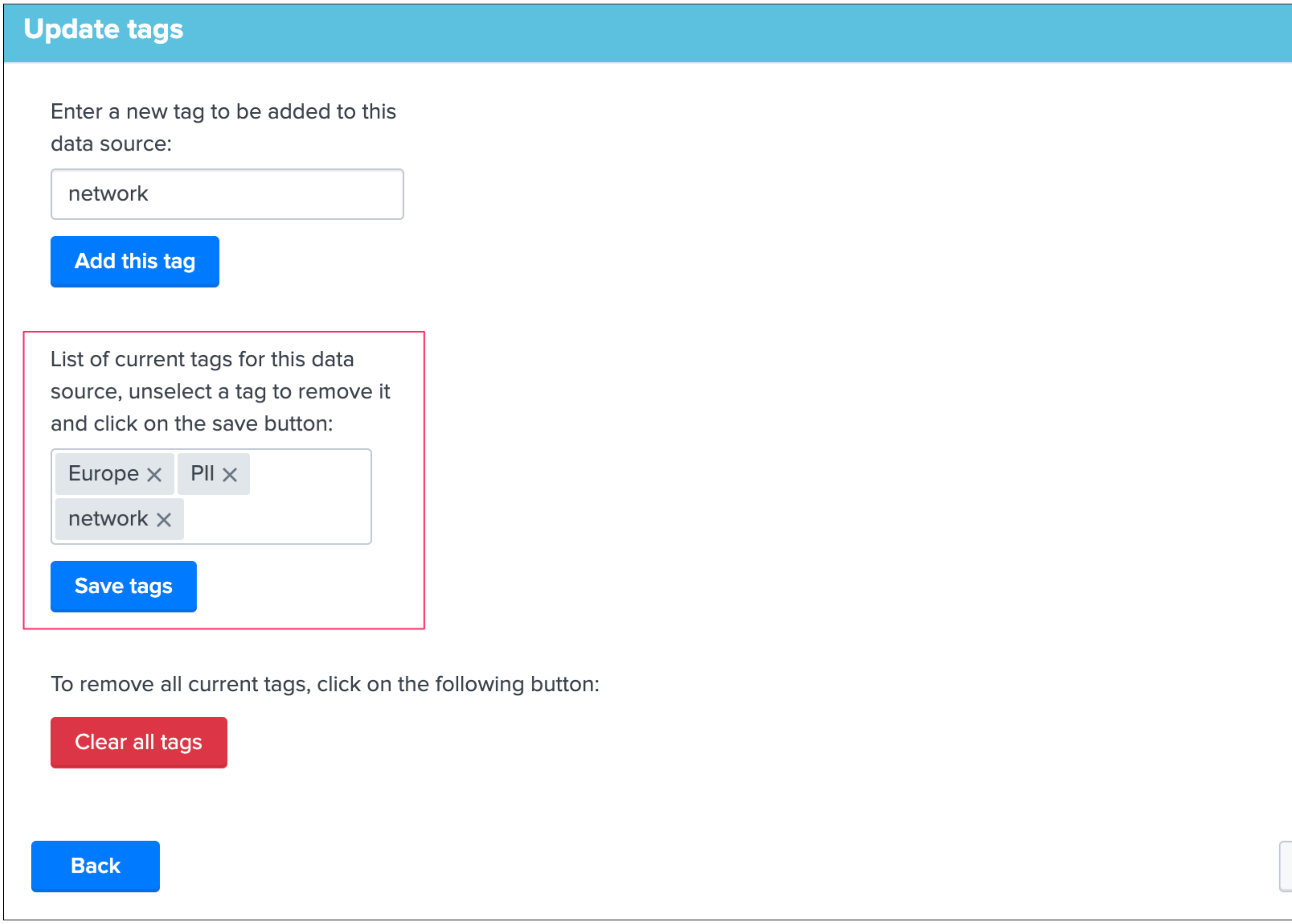
To remove all current tags, click on the following button:

Once a new tag is added, it is made available automatically in the tag filter from the main Trackme data source screen.

Updating tags

Note: Tags that have been defined by a tags policies will be defined again as long as the policy applies, to update tags applied by policies, the policy has to be updated

You can update tags using the multi-select dropdown input, by update we mean that you can clear one or more tags that are currently affected to a given data source, which updates immediately the list of tags in the main screen tags filter form.



Enter a new tag to be added to this data source:

network

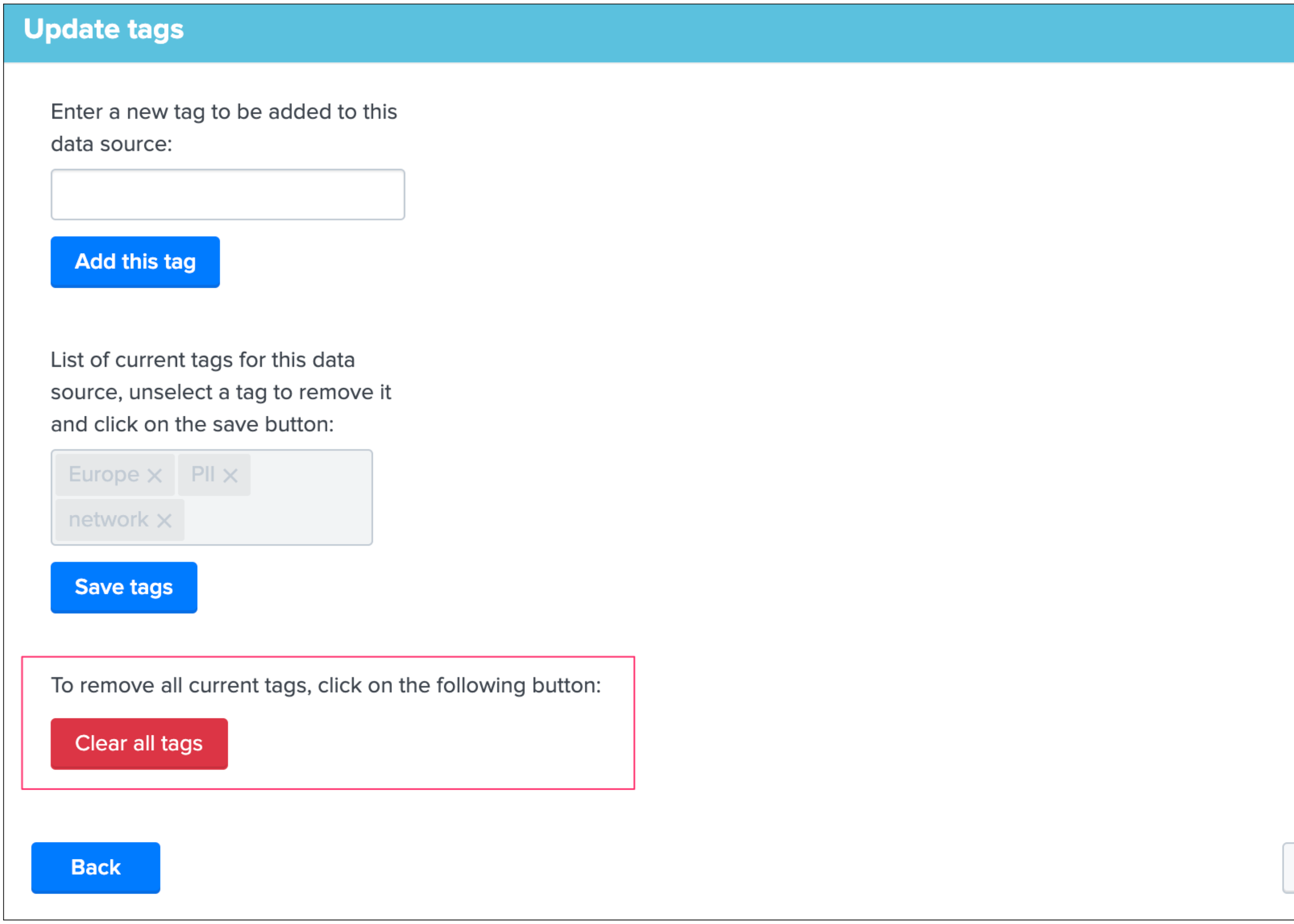
List of current tags for this data source, unselect a tag to remove it and click on the save button:

To remove all current tags, click on the following button:

Clearing tags

Note: Tags that have been defined by a tags policies will be defined again as long as the policy applies, to update tags applied by policies, the policy has to be updated

You can clear all tags that are currently affected to a data source, by clicking on the Clear tags button, you remove all tags for this data source.



Enter a new tag to be added to this data source:

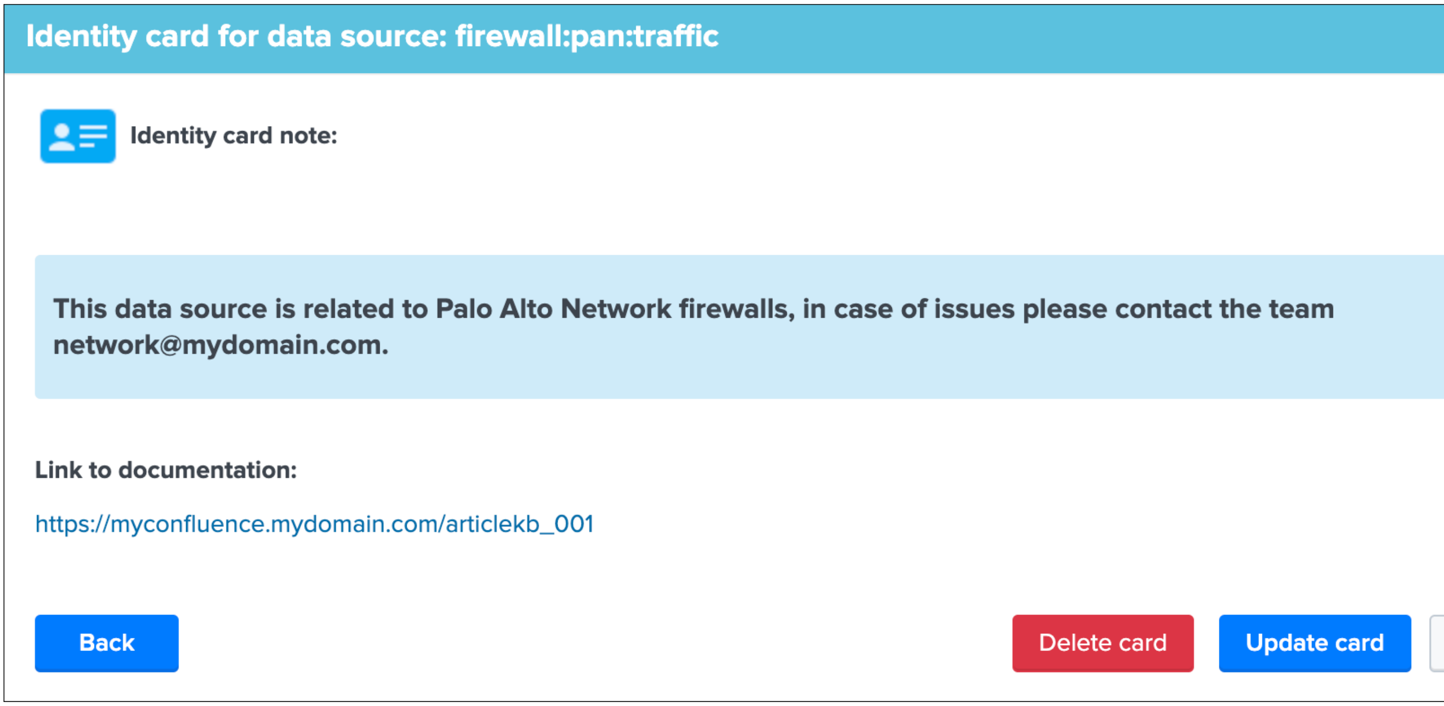
List of current tags for this data source, unselect a tag to remove it and click on the save button:

To remove all current tags, click on the following button:

3.1.21 Data identity card

Data identity card

- Data identity cards allow you to define a Web link and a documentation note that will be stored in a KVstore collection, and made available automatically via the UI and the out of the box alert.
- Data identity cards are managed via the UI, when no card has been defined yet for a data source, a message indicating it is shown.
- Data identity cards are available for data sources monitoring only.
- You can define a global idendity card that will be used by default to provide a link and a note, and you can still create specific identity cards and associations.
- You can define wildcard matching identity cards using the API endpoint and the trackme SPL command.



Identity card for data source: firewall:pan:traffic

Identity card note:

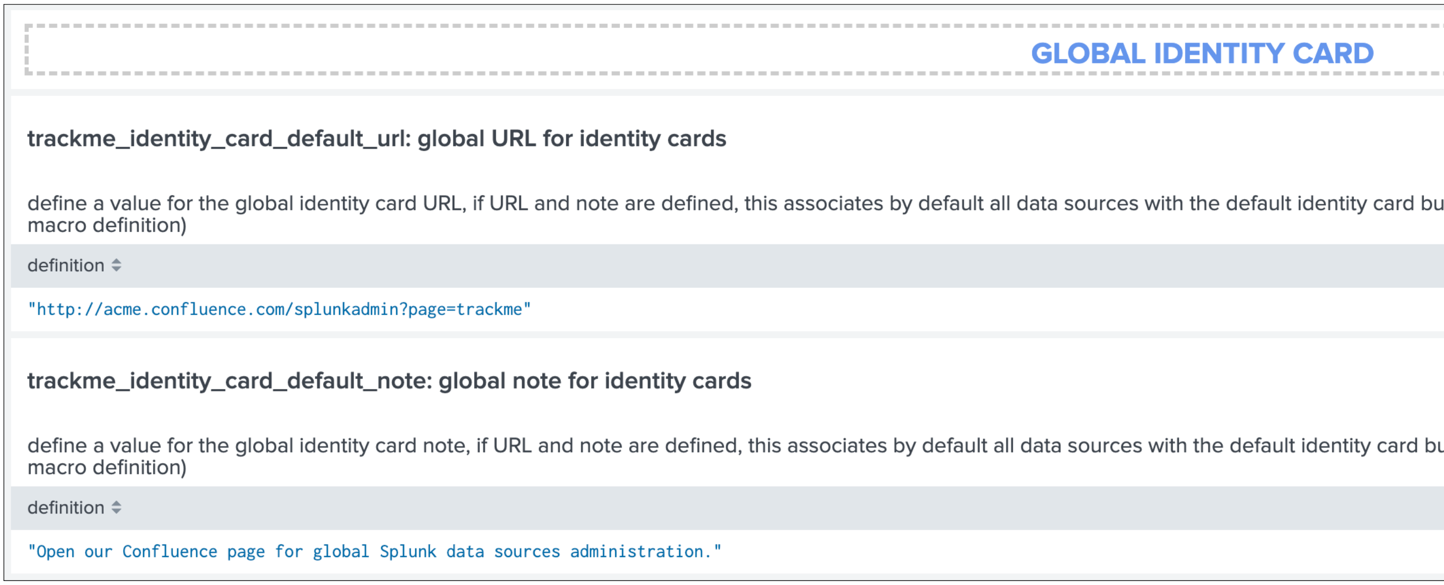
This data source is related to Palo Alto Network firewalls, in case of issues please contact the team network@mydomain.com.

Link to documentation:

https://myconfluence.mydomain.com/articlekb_001

Data identity: global identity card

As a TrackMe administrator, define a value for the global URL and the global note macros, you can quickly access these macros in the **TrackMe Manage and configure** UI:



GLOBAL IDENTITY CARD

trackme_identity_card_default_url: global URL for identity cards

define a value for the global identity card URL, if URL and note are defined, this associates by default all data sources with the default identity card bu macro definition)

definition ⇕
"http://acme.confluence.com/splunkadmin?page=trackme"

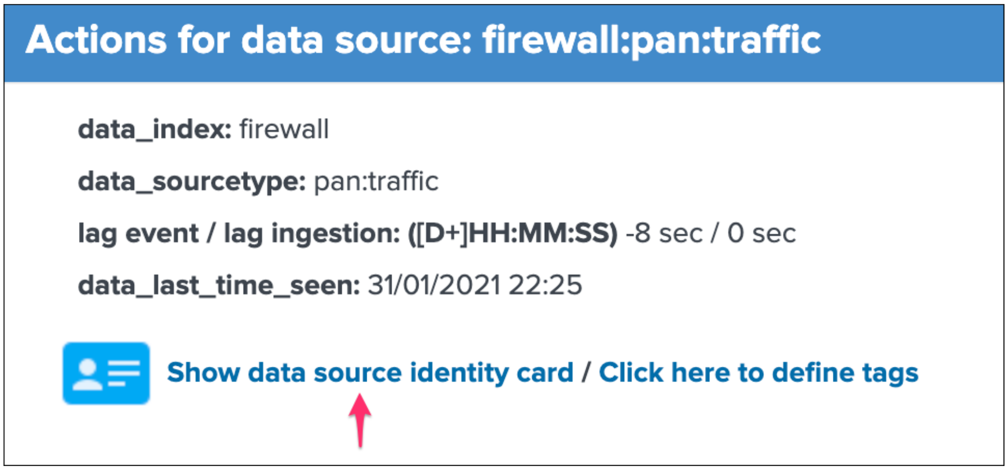
trackme_identity_card_default_note: global note for identity cards

define a value for the global identity card note, if URL and note are defined, this associates by default all data sources with the default identity card bu macro definition)

definition ⇕
"Open our Confluence page for global Splunk data sources administration."

Warning: The global identity card is enabled only if a value was defined for **both** the URL and the note

Once defined, the global identity card shows an active link:



Actions for data source: firewall:pan:traffic

data_index: firewall

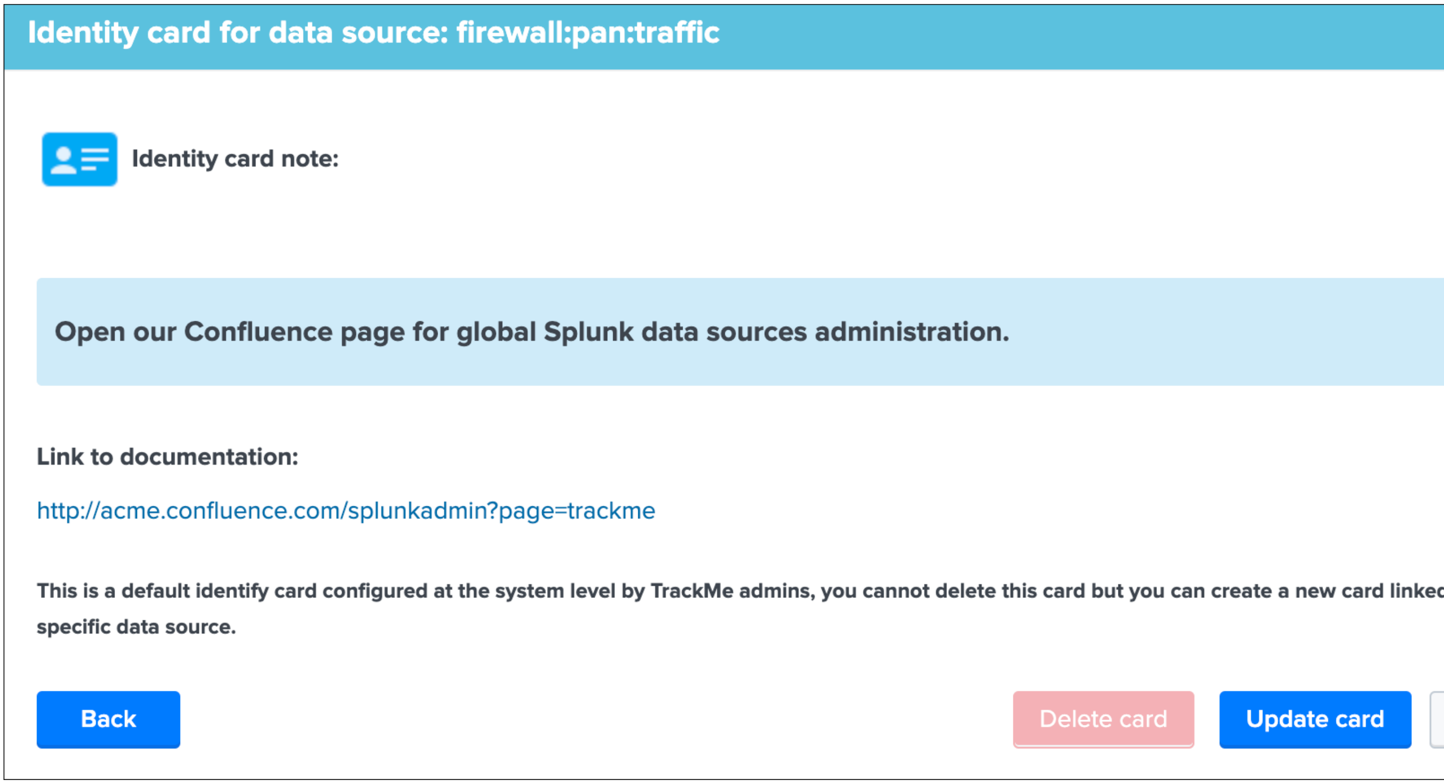
data_sourcetype: pan:traffic

lag event / lag ingestion: ([D+]HH:MM:SS) -8 sec / 0 sec

data_last_time_seen: 31/01/2021 22:25

Show data source identity card / Click here to define tags

Following the link opens the identity card UI:



Identity card for data source: firewall:pan:traffic

Identity card note:

Open our Confluence page for global Splunk data sources administration.

Link to documentation:

http://acme.confluence.com/splunkadmin?page=trackme

This is a default identify card configured at the system level by TrackMe admins, you cannot delete this card but you can create a new card linke
specific data source.

Given that this is a global identity card, the "Delete card" is disabled automatically, however it is still possible to create a new identity card to be associated with this data source which will replace the global card automatically.

Note: if you create a global card while existing cards have defined already, there will be no impacs for existing cards, custom cards take precedence over the default card if any.

Data identity: wildcard matching

In some cases, you will want to have a few ID cards that cover the whole picture relying on your naming convention, you can use wildcard matching for this purpose without having to manually associate each entity with an ID card:

Assume the following example:

- All data sources related to linux_secure are stored in indexes that uses a naming convention starting by `linux_`
- We want to create one ID card wich provides a quick informational note, and the link to our documentation
- We can to create a an ID card and use wildcard matching to automatically associate any `linux_` entity with it
- In addition, we add an additional wildcard matching for anything that starts by `windows_`

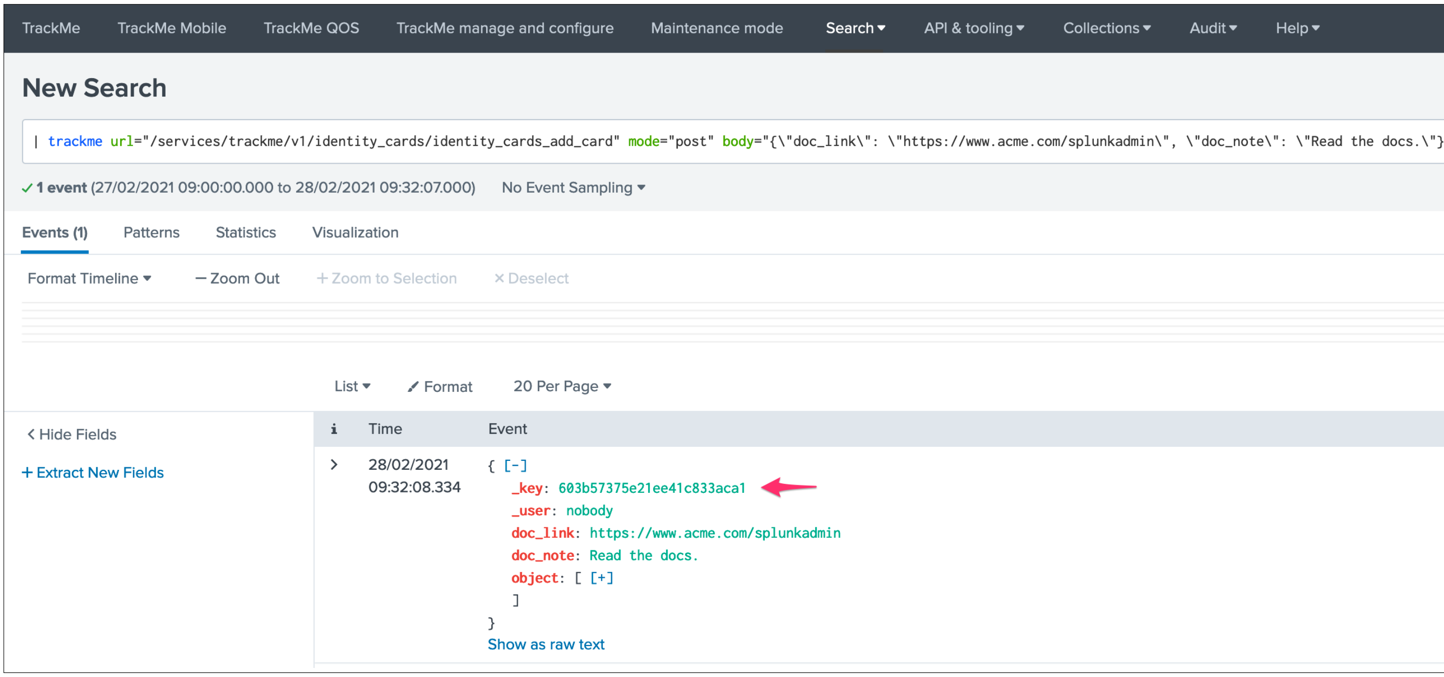
Step 1: Create the Identity card using the `trackme` SPL command

Run the following `trackme` SPL command to create a new ID card:

```
| trackme url="/services/trackme/v1/identity_cards/identity_cards_add_card" mode="post
↪" body="{\"doc_link\": \"https://www.acme.com/splunkadmin\", \"doc_note\": \"Read␣
↪the docs.\"}"
```

At this stage, the ID card is not yet associated with any entities, if the card exists already for the same documentation link, it would be updated with these information.

This command returns the ID card as a JSON object, note the `key` value which you need for the steps 2:



```
| trackme url="/services/trackme/v1/identity_cards/identity_cards_add_card" mode="post" body="{\"doc_link\": \"https://www.acme.com/splunkadmin\", \"doc_note\": \"Read the docs.\"}
```

28/02/2021 09:32:08.334

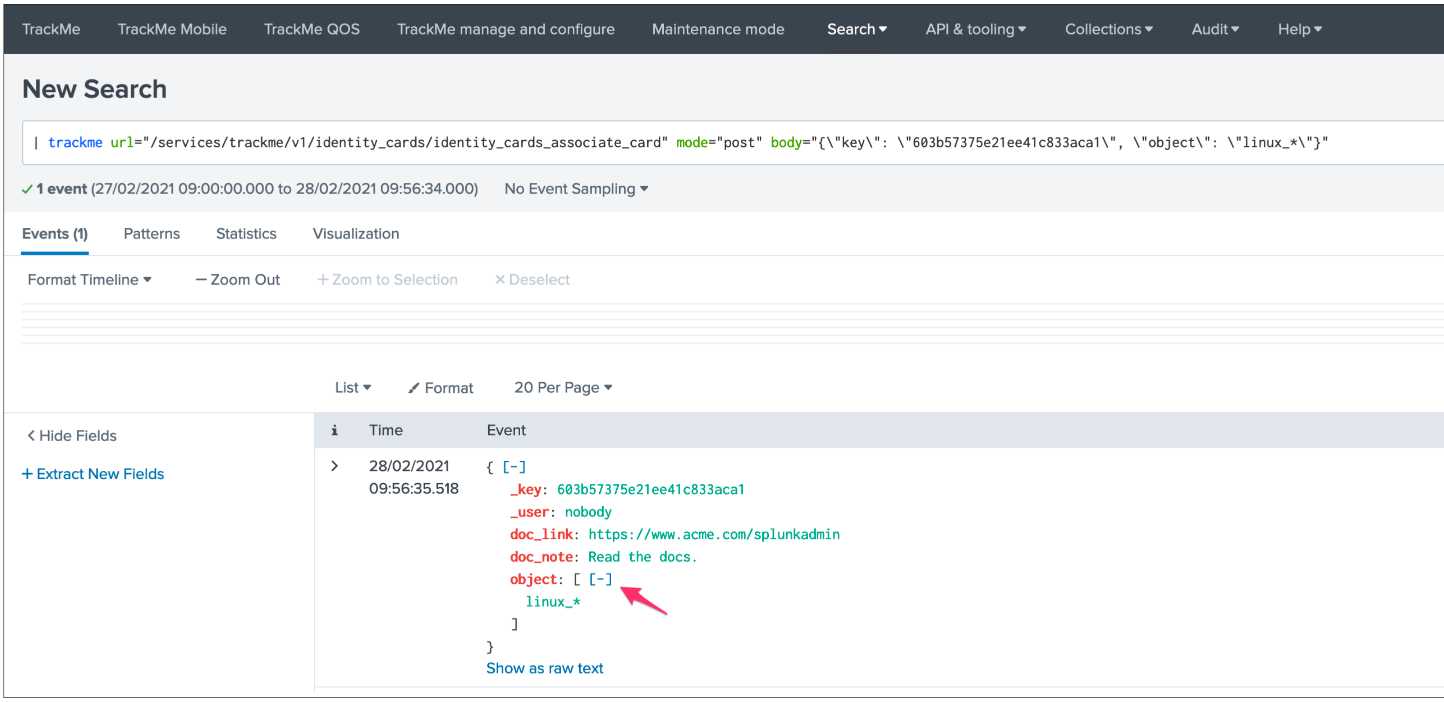
```
{ [-]
   _key: 603b57375e21ee41c833aca1
   _user: nobody
   doc_link: https://www.acme.com/splunkadmin
   doc_note: Read the docs.
   object: [ [+]
   ]
}
```

Step 2: Associate the Identity card using the `trackme` SPL command

Run the following `trackme` SPL command to create the wildcard matching association, say for `linux_*`:

```
| trackme url="/services/trackme/v1/identity_cards/identity_cards_associate_card"␣
↪mode="post" body="{\"key\": \"60327fd8af39041f28403191\", \"object\": \"linux_*\"}"
```

This command returns the ID card as a JSON object, develop the object JSON key to observe the new association:



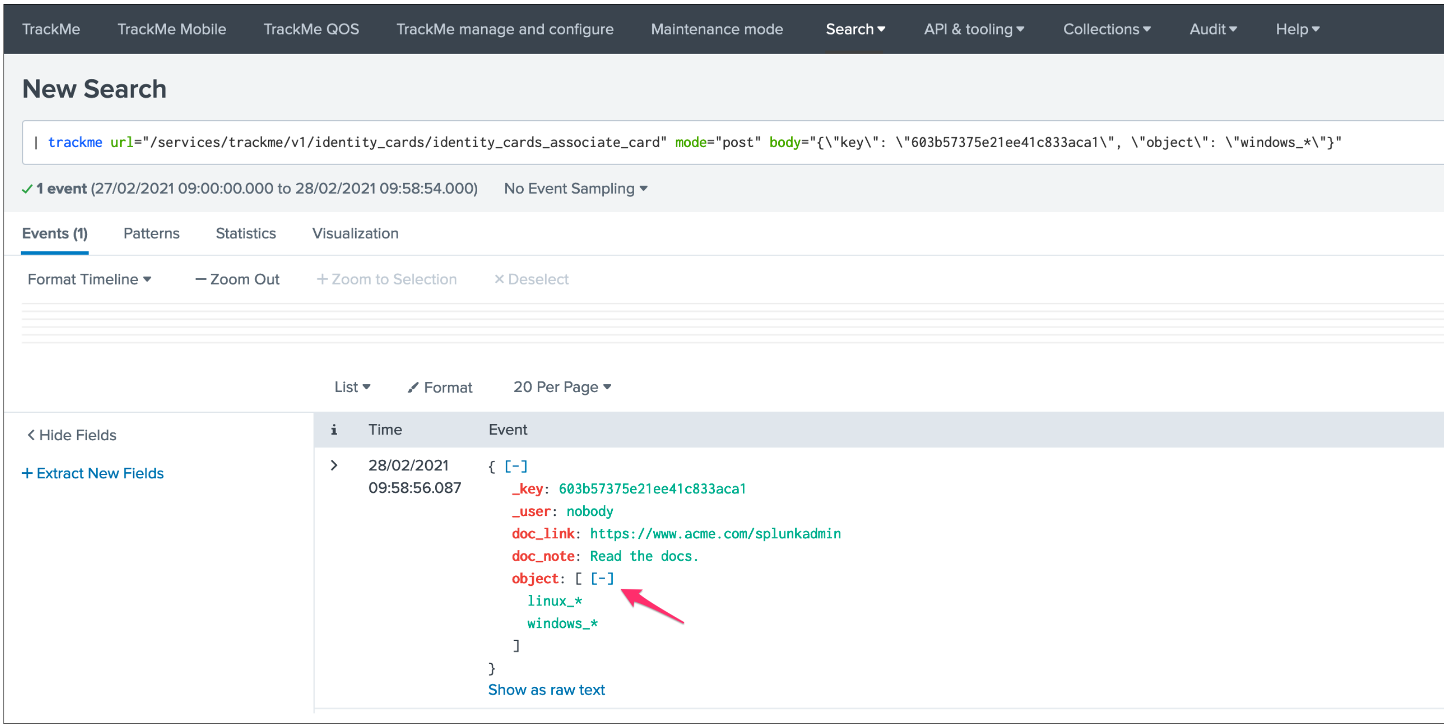
```
| trackme url="/services/trackme/v1/identity_cards/identity_cards_associate_card" mode="post" body="{\"key\": \"603b57375e21ee41c833aca1\", \"object\": \"linux_*\"}"
```

✓ 1 event (27/02/2021 09:00:00.000 to 28/02/2021 09:56:34.000)

```
28/02/2021
09:56:35.518
{ [-]
   _key: 603b57375e21ee41c833aca1
   _user: nobody
   doc_link: https://www.acme.com/splunkadmin
   doc_note: Read the docs.
   object: [ [-]
     linux_*
   ]
}
```

Any entity matching this wildcard criteria will now be associated with this ID card, shall you want to associate the same card with another matching wildcard, say `windows_*`:

```
| trackme url="/services/trackme/v1/identity_cards/identity_cards_associate_card"␣
↪mode="post" body="{\"key\": \"60327fd8af39041f28403191\", \"object\": \"windows_*\"}
↪"
```

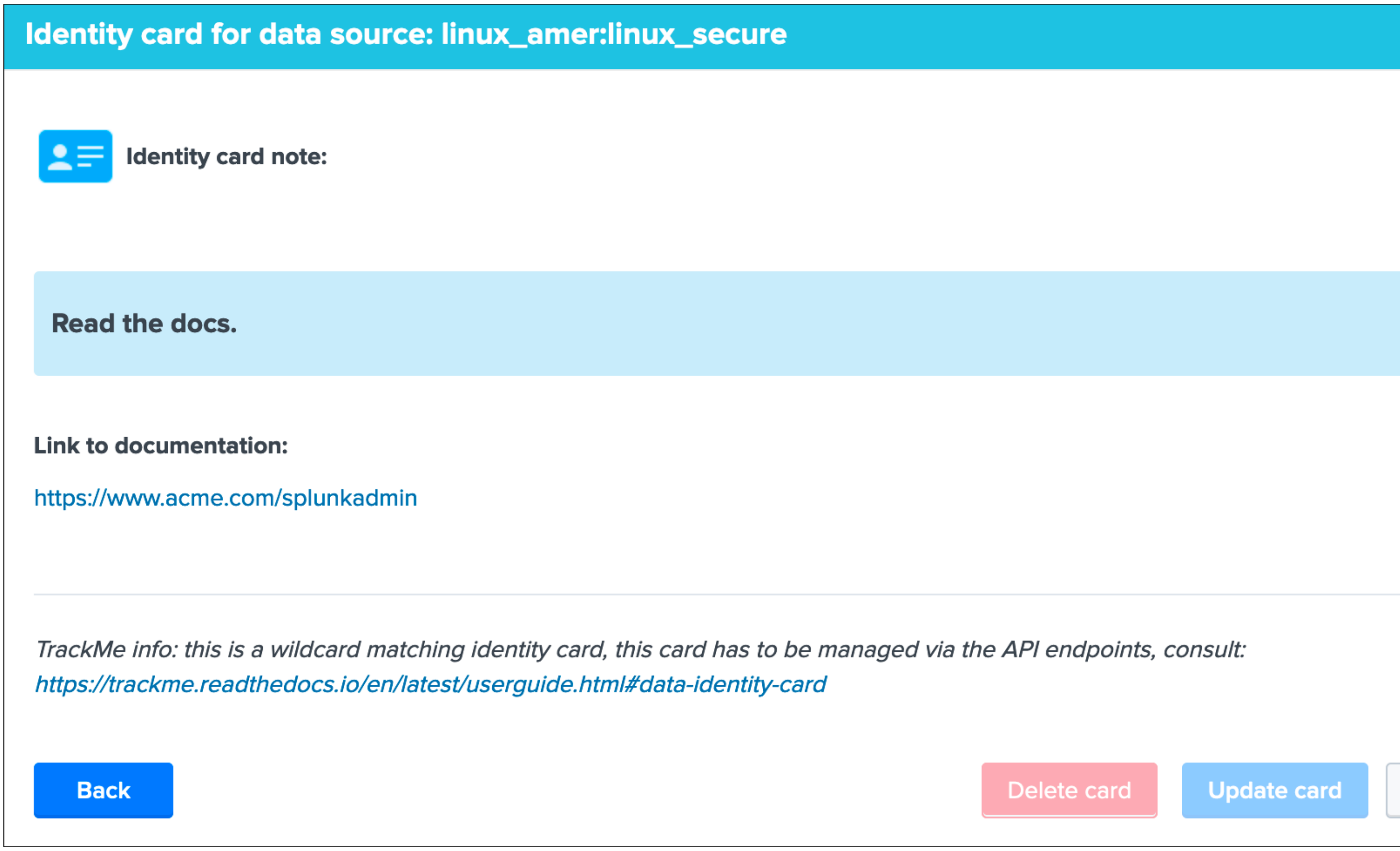


```
| trackme url="/services/trackme/v1/identity_cards/identity_cards_associate_card" mode="post" body="{\"key\": \"603b57375e21ee41c833aca1\", \"object\": \"windows_*\"}"
```

✓ 1 event (27/02/2021 09:00:00.000 to 28/02/2021 09:58:54.000)

```
28/02/2021
09:58:56.087
{ [-]
   _key: 603b57375e21ee41c833aca1
   _user: nobody
   doc_link: https://www.acme.com/splunkadmin
   doc_note: Read the docs.
   object: [ [-]
     linux_*
     windows_*
   ]
}
```

Make sure to reload the TrackMe UI, the following ID card will be associated automatically with any entity that matches your criterias:



Identity card for data source: linux_amer:linux_secure

Identity card note:

Read the docs.

Link to documentation:

https://www.acme.com/splunkadmin

TrackMe info: this is a wildcard matching identity card, this card has to be managed via the API endpoints, consult: https://trackme.readthedocs.io/en/latest/userguide.html#data-identity-card

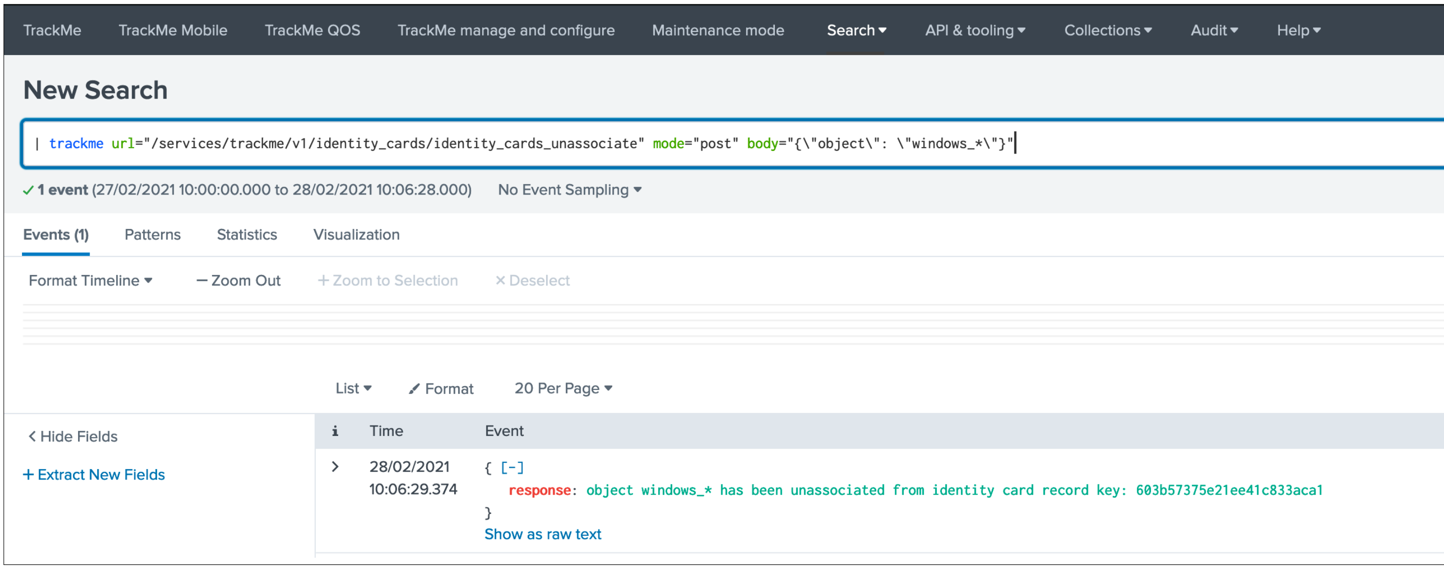
And so forth for any additional wildcard matching you may need.

Hint: A message appears at the end of the ID card screen indicating that this is a wildcard matching card that cannot be managed via the UI but with the trackme SPL command and the relevant API endpoints

Removing a wildcard association using the `trackme` SPL command

An association can be removed easily, the following `trackme` SPL command removes the association with the `windows_*` wildcard match:

```
| trackme url="/services/trackme/v1/identity_cards/identity_cards_unassociate" mode=
↪"post" body="{\"object\": \"windows_*\"}"
```



New Search

```
| trackme url="/services/trackme/v1/identity_cards/identity_cards_unassociate" mode="post" body="{\"object\": \"windows_*\"}"
```

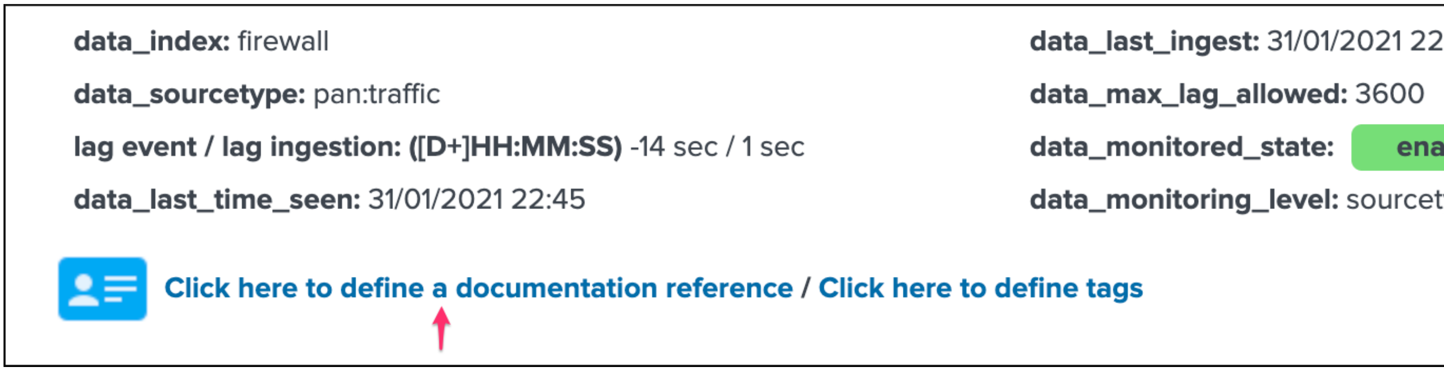
✓ 1 event (27/02/2021 10:00:00.000 to 28/02/2021 10:06:28.000) No Event Sampling

Time	Event
28/02/2021 10:06:29.374	{ [-] response: object windows_* has been unassociated from identity card record key: 603b57375e21ee41c833aca1 }

For additional options or more details, consult the *Identity Cards endpoints* documentation.

Data identity: workflow

If the data source has not been associated to a card yet (or no global card was defined), the UI shows a link to define the a documentation reference:



data_index: firewall

data_sourcetype: pan:traffic

lag event / lag ingestion: ([D+]HH:MM:SS) -14 sec / 1 sec

data_last_time_seen: 31/01/2021 22:45

data_last_ingest: 31/01/2021 22

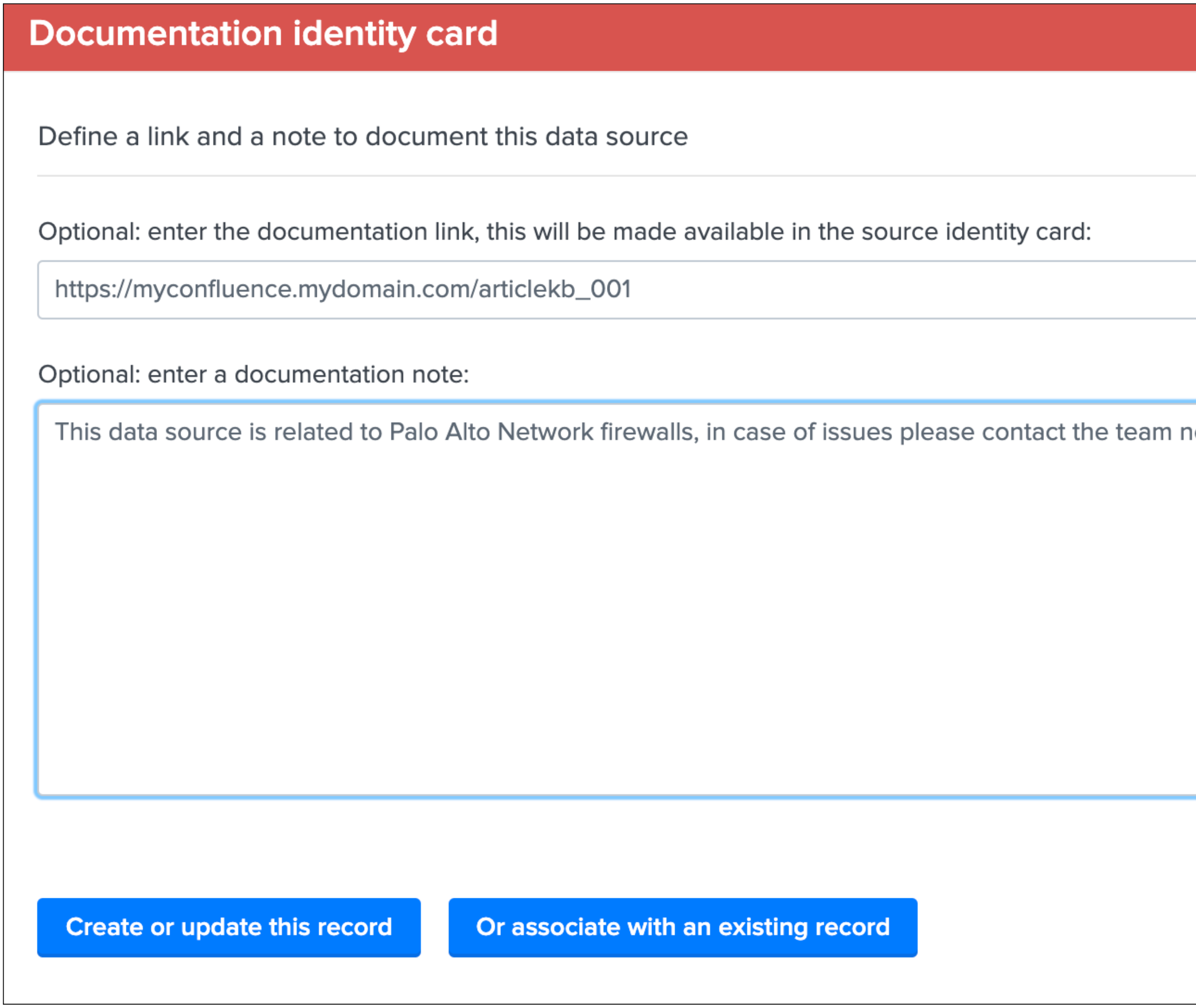
data_max_lag_allowed: 3600

data_monitored_state: ena

data_monitoring_level: sourcet

Click here to define a documentation reference / Click here to define tags

You can click on the link to create a new identity card:



Documentation identity card

Define a link and a note to document this data source

Optional: enter the documentation link, this will be made available in the source identity card:

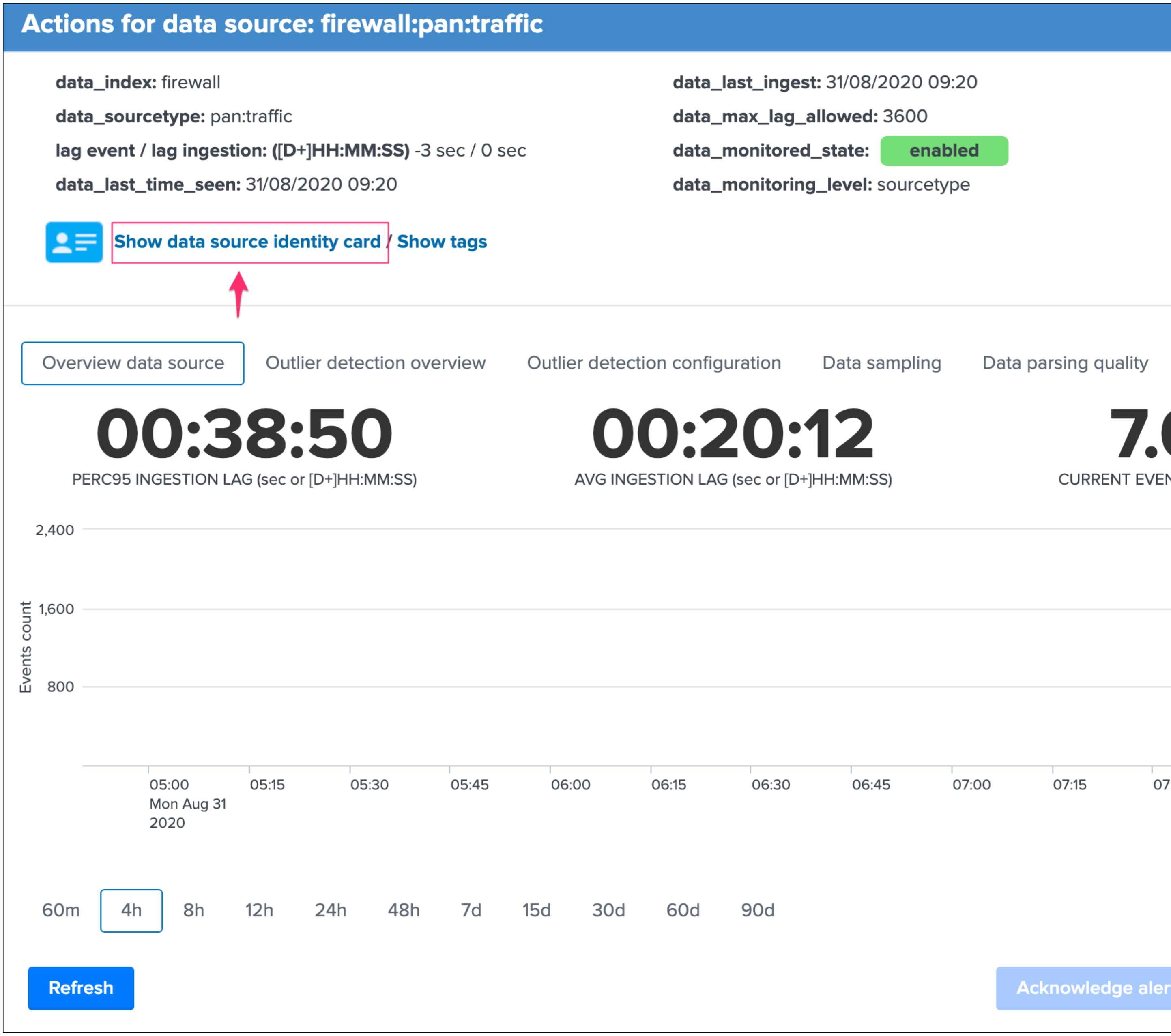
https://myconfluence.mydomain.com/articlekb_001

Optional: enter a documentation note:

This data source is related to Palo Alto Network firewalls, in case of issues please contact the team n

Create or update this record
Or associate with an existing record

Once the identity card has been created, the following message link is shown:



Actions for data source: firewall:pan:traffic

data_index: firewall

data_sourcetype: pan:traffic

lag event / lag ingestion: ([D+]HH:MM:SS) -3 sec / 0 sec

data_last_time_seen: 31/08/2020 09:20

data_last_ingest: 31/08/2020 09:20

data_max_lag_allowed: 3600

data_monitored_state: enabled

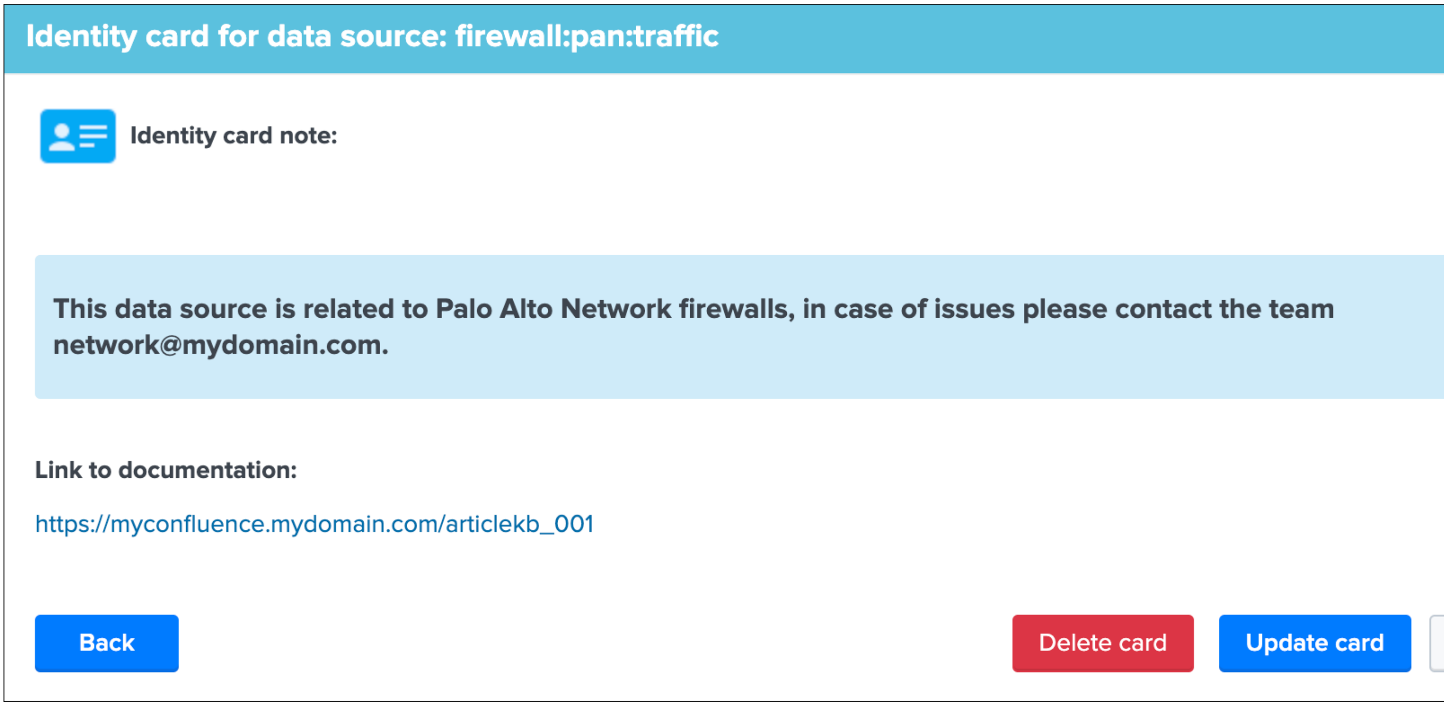
data_monitoring_level: sourcetype

00:38:50 PERC95 INGESTION LAG (sec or [D+]HH:MM:SS)

00:20:12 AVG INGESTION LAG (sec or [D+]HH:MM:SS)

7.0 CURRENT EVEN

Which automatically provides a view with the identity card content:



Identity card for data source: firewall:pan:traffic

Identity card note:

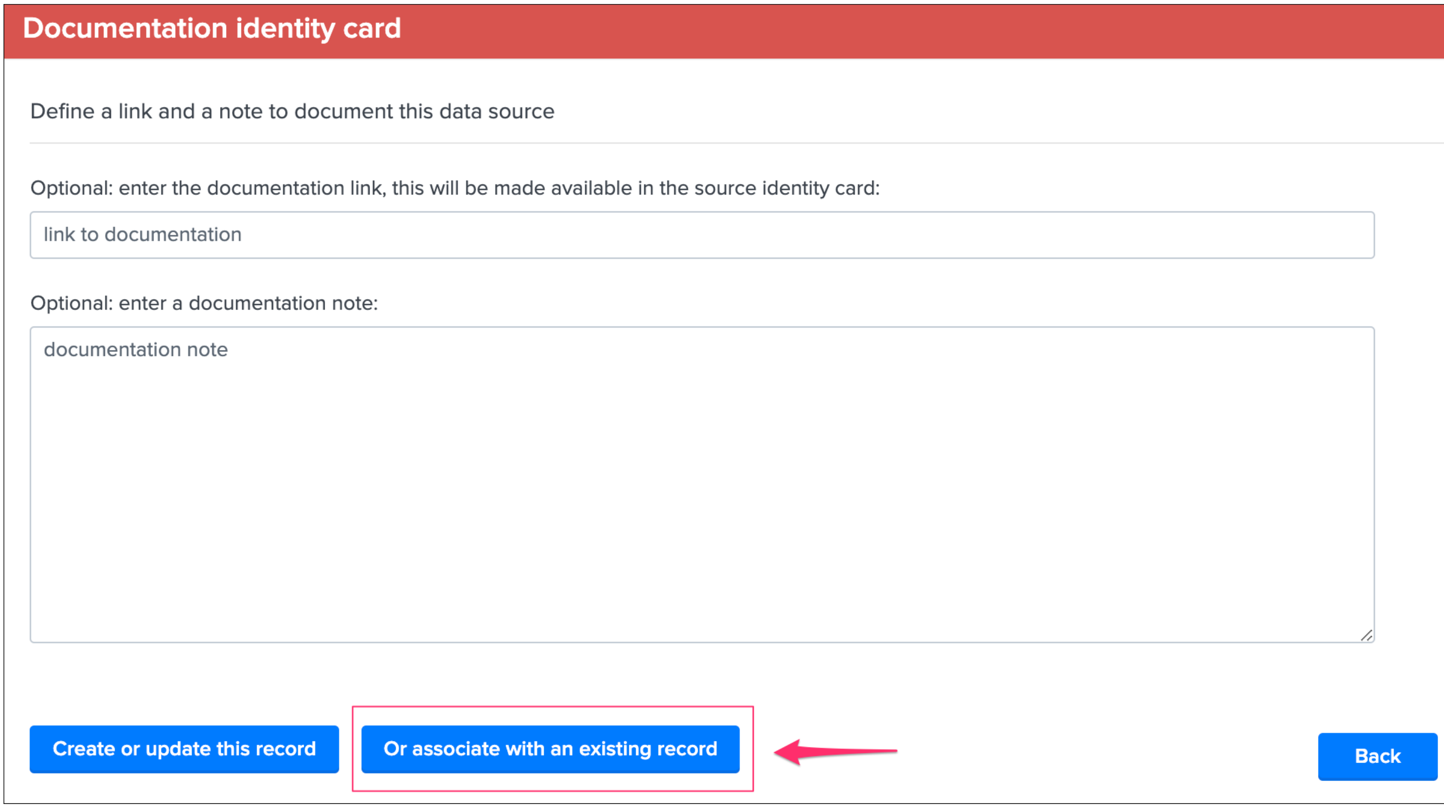
This data source is related to Palo Alto Network firewalls, in case of issues please contact the team network@mydomain.com.

Link to documentation:

https://myconfluence.mydomain.com/articlekb_001

In addition, the fields “doc_link” and “doc_note” are part of the default output of the default alert, which can be recycled eventually to enrich a ticketing system incident.

Finally, multiple entities can share the same identity record via the identity card association feature and button:



Documentation identity card

Define a link and a note to document this data source

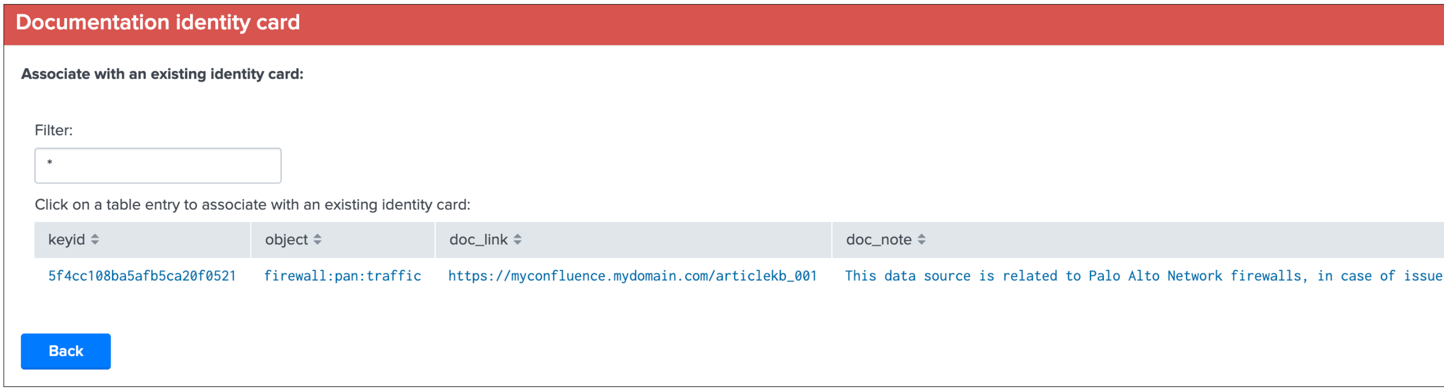
Optional: enter the documentation link, this will be made available in the source identity card:

link to documentation

Optional: enter a documentation note:

documentation note

Create or update this record
Or associate with an existing record
Back



Documentation identity card

Associate with an existing identity card:

Filter: *

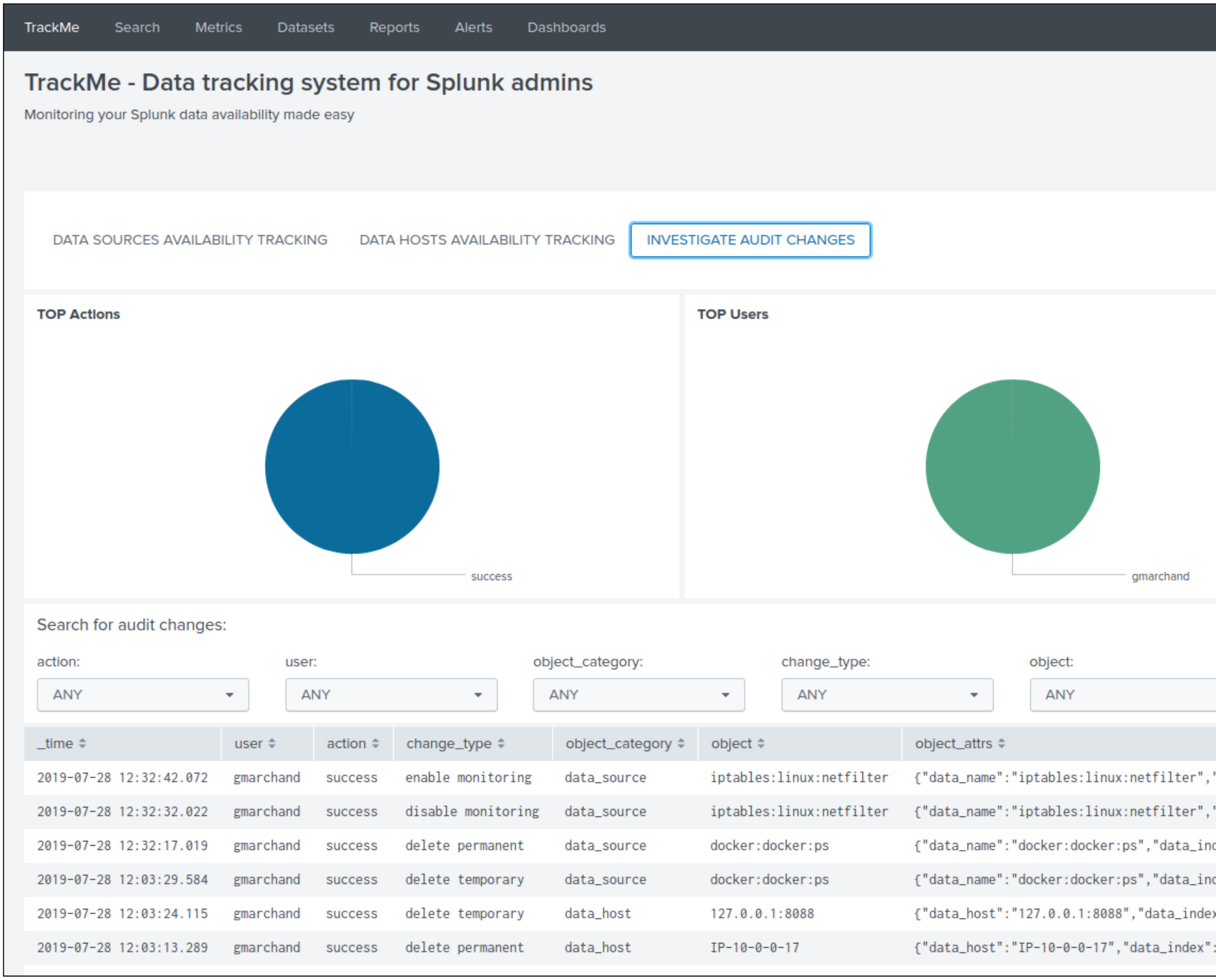
Click on a table entry to associate with an existing identity card:

keyid	object	doc_link	doc_note
5f4cc108ba5afb5ca20f0521	firewall:pan:traffic	https://myconfluence.mydomain.com/articlekb_001	This data source is related to Palo Alto Network firewalls, in case of issue

3.1.22 Auditing changes

Auditing

Every action that involves a modification of an object via the UI is stored in a KVstore collection to be used for auditing and investigation purposes.


TrackMe
Search
Metrics
Datasets
Reports
Alerts
Dashboards
DATA SOURCES AVAILABILITY TRACKING
DATA HOSTS AVAILABILITY TRACKING
INVESTIGATE AUDIT CHANGES
TOP Actions
success
TOP Users
gmarchand

TrackMe - Data tracking system for Splunk admins

Monitoring your Splunk data availability made easy

Search for audit changes:

action: ANY

user: ANY

object_category: ANY

change_type: ANY

object: ANY

_time	user	action	change_type	object_category	object	object_attrs
2019-07-28 12:32:42.072	gmarchand	success	enable monitoring	data_source	iptables:linux:netfilter	{"data_name":"iptables:linux:netfilter","
2019-07-28 12:32:32.022	gmarchand	success	disable monitoring	data_source	iptables:linux:netfilter	{"data_name":"iptables:linux:netfilter","
2019-07-28 12:32:17.019	gmarchand	success	delete permanent	data_source	docker:docker:ps	{"data_name":"docker:docker:ps","data_inc
2019-07-28 12:03:29.584	gmarchand	success	delete temporary	data_source	docker:docker:ps	{"data_name":"docker:docker:ps","data_inc
2019-07-28 12:03:24.115	gmarchand	success	delete temporary	data_host	127.0.0.1:8088	{"data_host":"127.0.0.1:8088","data_index
2019-07-28 12:03:13.289	gmarchand	success	delete permanent	data_host	IP-10-0-0-17	{"data_host":"IP-10-0-0-17","data_index":

Different information related to the change performed are stored in the collection, such as the user that performed the change, the type of object, the existing state before the change is performed, and so forth.

In addition, each audit change record has a time stamp information stored, which we use to purge old records automatically, via the scheduled report:

- `TrackMe - Audit changes night purge`

The purge is performed in a daily fashion executed during the night, by default every record older than 90 days will be purged.

You can customize this value using the following macro definition:

- `trackme_audit_changes_retention`

Finally, the auditing change collection is automatically used by the trackers reports when a permanent deletion of an object has been requested.

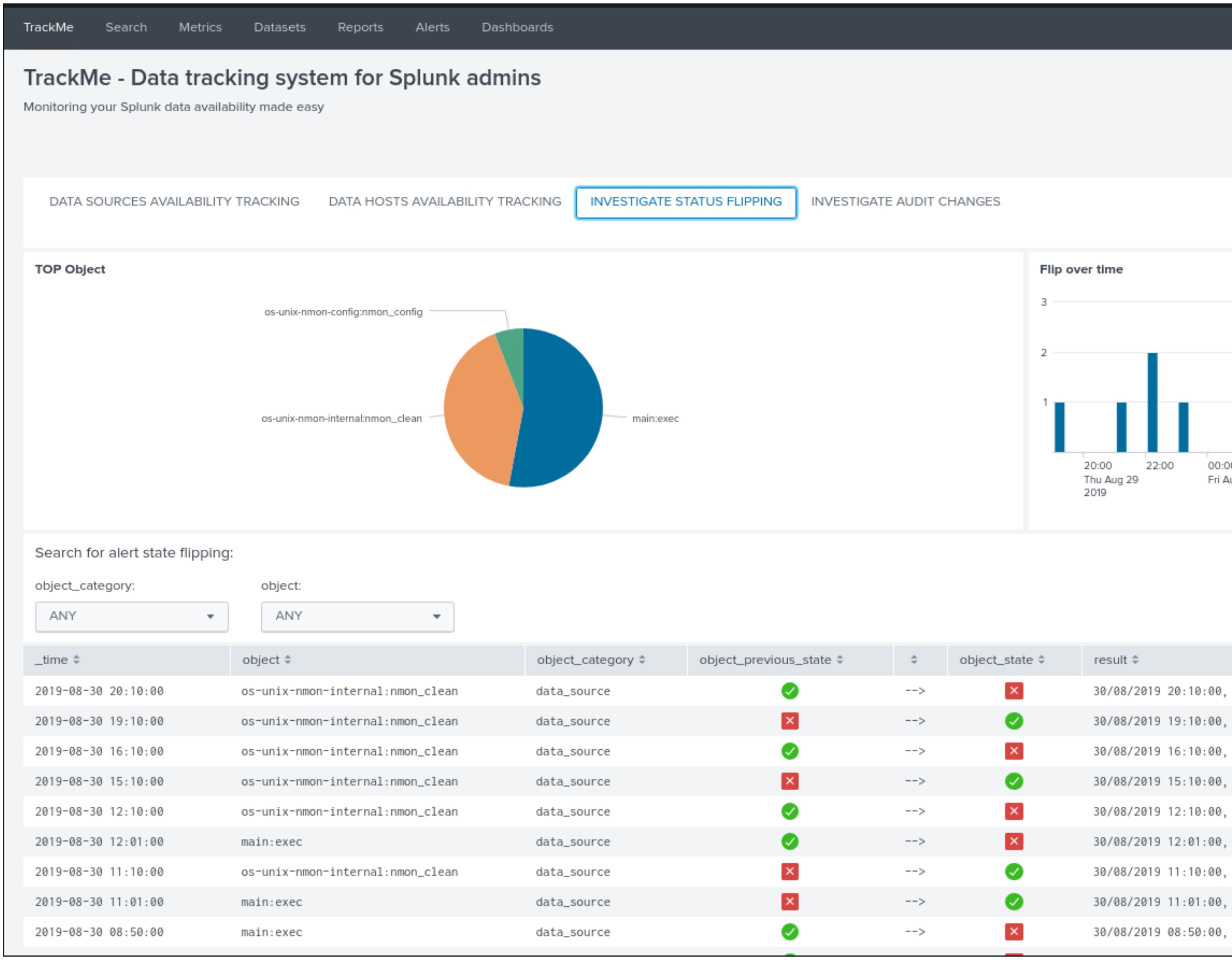
3.1.23 Flipping statuses auditing

Flipping statuses

Every time an entity status changes, for example from green to red, a record of that event is stored as a summary flipping status event.

```
`trackme_idx` source="flip_state_change_tracking"`
```

Using the UI, you can easily monitor and investigate the historical changes of a given a data source or host over time:


TrackMe
Search
Metrics
Datasets
Reports
Alerts
Dashboards
DATA SOURCES AVAILABILITY TRACKING
DATA HOSTS AVAILABILITY TRACKING
INVESTIGATE STATUS FLIPPING
INVESTIGATE AUDIT CHANGES
TOP Object
os-unix-nmon-config:nmon_config
os-unix-nmon-internal:nmon_clean
main:exec
Flip over time
3
2
1
20:00
Thu Aug 29
2019
22:00
00:0
Fri A
ANY
ANY

TrackMe - Data tracking system for Splunk admins

Monitoring your Splunk data availability made easy

Search for alert state flipping:

object_category: ANY

object: ANY

_time	object	object_category	object_previous_state		object_state	result
2019-08-30 20:10:00	os-unix-nmon-internal:nmon_clean	data_source	✓	-->	✗	30/08/2019 20:10:00,
2019-08-30 19:10:00	os-unix-nmon-internal:nmon_clean	data_source	✗	-->	✓	30/08/2019 19:10:00,
2019-08-30 16:10:00	os-unix-nmon-internal:nmon_clean	data_source	✓	-->	✗	30/08/2019 16:10:00,
2019-08-30 15:10:00	os-unix-nmon-internal:nmon_clean	data_source	✗	-->	✓	30/08/2019 15:10:00,
2019-08-30 12:10:00	os-unix-nmon-internal:nmon_clean	data_source	✓	-->	✗	30/08/2019 12:10:00,
2019-08-30 12:01:00	main:exec	data_source	✓	-->	✗	30/08/2019 12:01:00,
2019-08-30 11:10:00	os-unix-nmon-internal:nmon_clean	data_source	✗	-->	✓	30/08/2019 11:10:00,
2019-08-30 11:01:00	main:exec	data_source	✗	-->	✓	30/08/2019 11:01:00,
2019-08-30 08:50:00	main:exec	data_source	✓	-->	✗	30/08/2019 08:50:00,

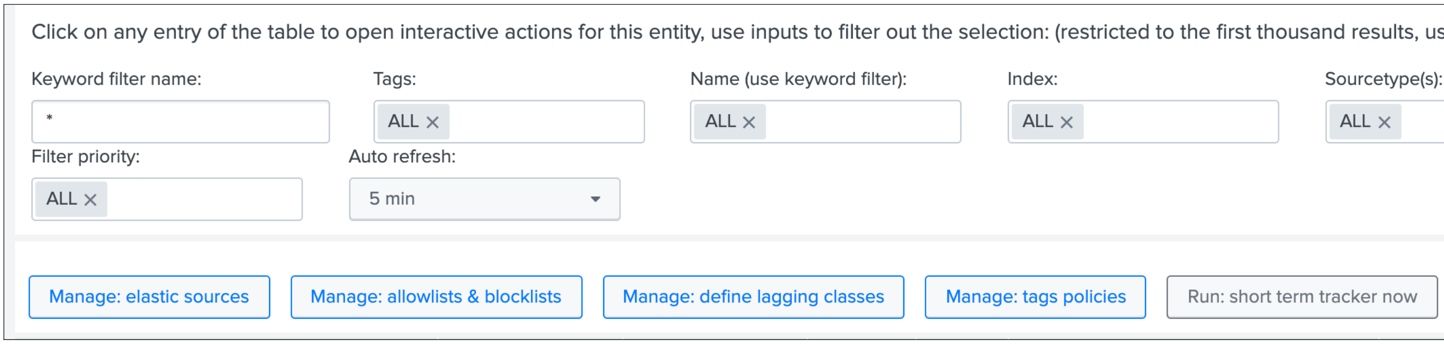
These events are automatically generated by the tracker reports, and are as well used for SLA calculation purposes.

3.1.24 Ops: Queues center

Splunk queues usage

The Queue center provides quick access to the main Splunk queues statistics.

The Ops view for Splunk indexing queues is accessible from the "Ops: Queues center" button in the main Trackme screen:



Click on any entry of the table to open interactive actions for this entity, use inputs to filter out the selection: (restricted to the first thousand results, us

Keyword filter name: *

Tags: ALL ×

Name (use keyword filter): ALL ×

Index: ALL ×

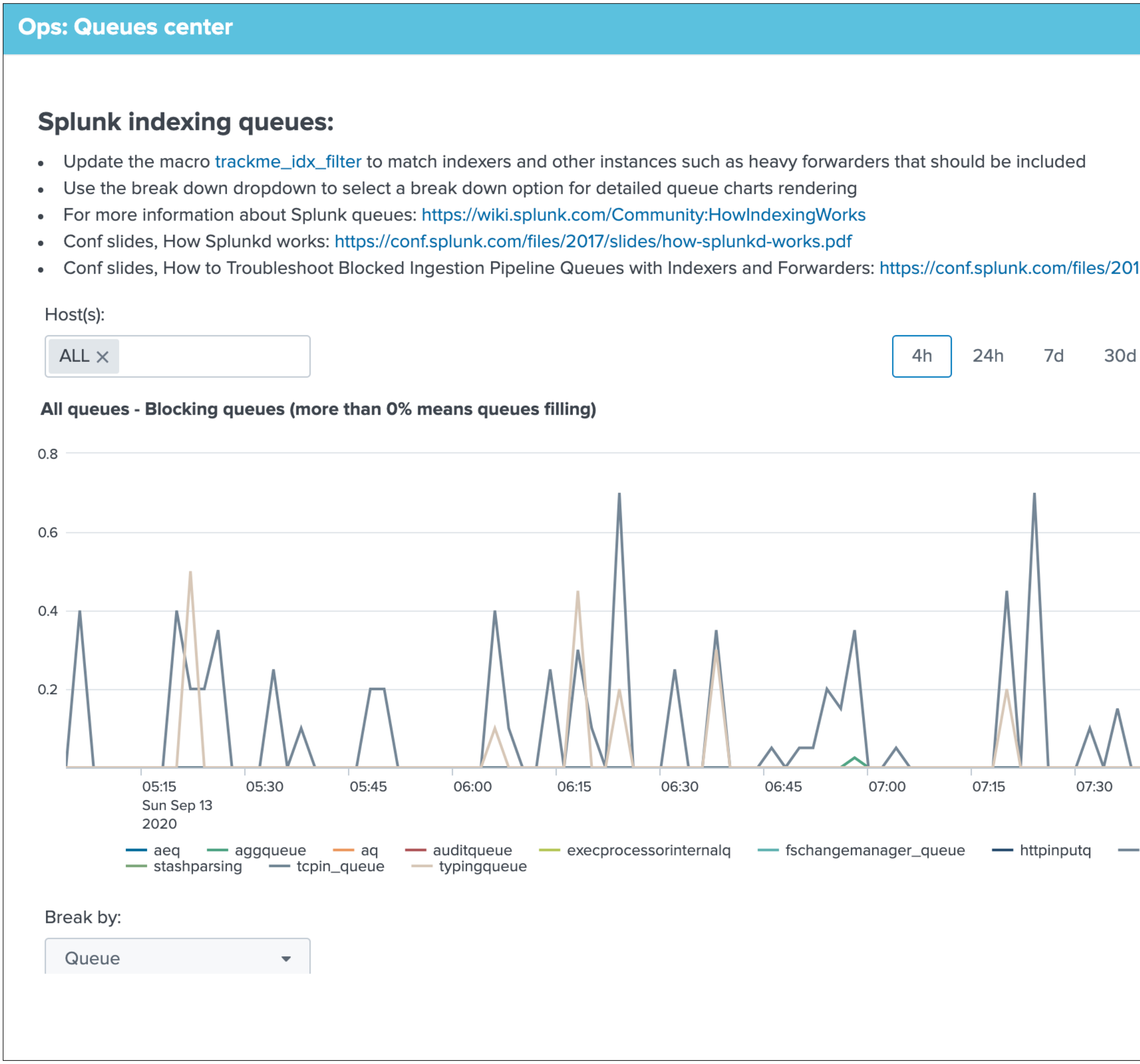
Sourcetype(s): ALL ×

Filter priority: ALL ×

Auto refresh: 5 min

Manage: elastic sources
Manage: allowlists & blocklists
Manage: define lagging classes
Manage: tags policies
Run: short term tracker now

This view shows Splunk pipeline queues usage in your environment, using the filtering results from the macro trackme_idx_filter, make sure this macro is configured to filter on indexers and heavy forwarders:



Ops: Queues center

Splunk indexing queues:

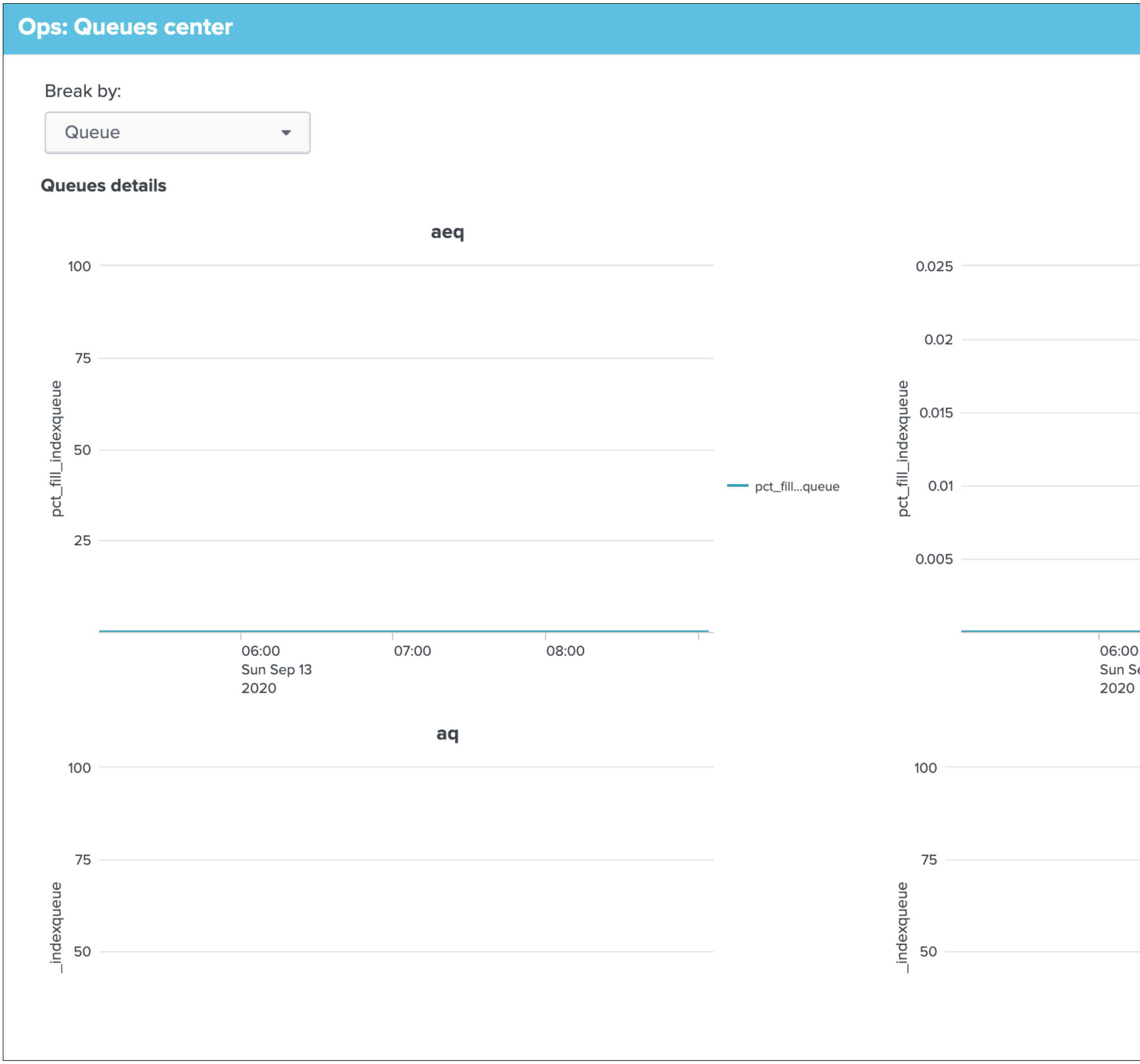
- Update the macro trackme_idx_filter to match indexers and other instances such as heavy forwarders that should be included
- Use the break down dropdown to select a break down option for detailed queue charts rendering
- For more information about Splunk queues: https://wiki.splunk.com/Community:HowIndexingWorks
- Conf slides, How Splunkd works: https://conf.splunk.com/files/2017/slides/how-splunkd-works.pdf
- Conf slides, How to Troubleshoot Blocked Ingestion Pipeline Queues with Indexers and Forwarders: https://conf.splunk.com/files/201

Host(s): ALL ×

4h 24h 7d 30d

All queues - Blocking queues (more than 0% means queues filling)

Break by: Queue



Ops: Queues center

Break by:

Queues details

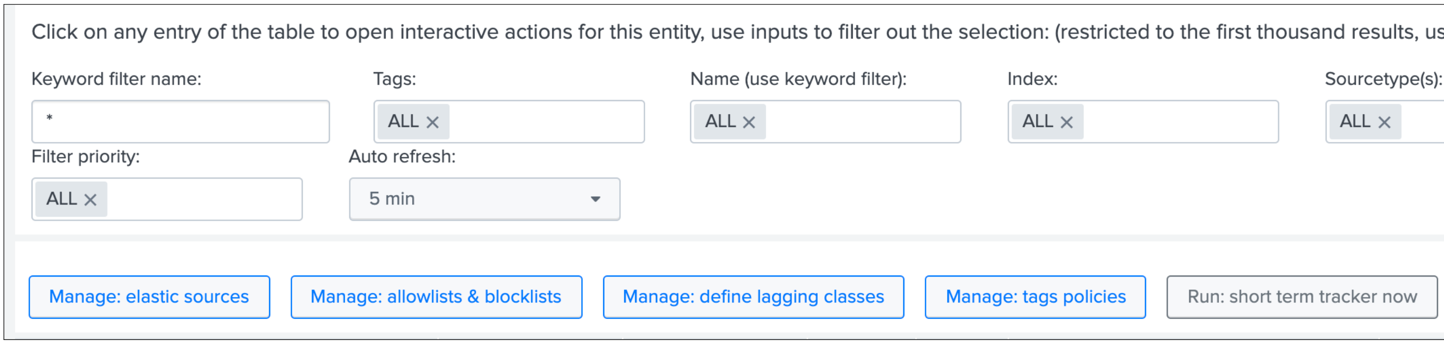
Options in the view:

- You can use the multiselect form to choose instances to be considered
- You can select a time range between the provided options
- Scroll down within the window, and choose different break down options in the detailed queue usage treillis charts dependending on your needs

3.1.25 Ops: Parsing view

Splunk parsing errors

- The Ops view for Splunk indexing time parsing failures and warnings is available from the TrackMe main screen via the "Ops: Parsing view" button.
- This UI shows the different types of parsing error happening in Splunk at the ingestion time.



Click on any entry of the table to open interactive actions for this entity, use inputs to filter out the selection: (restricted to the first thousand results, us

Keyword filter name: *

Tags: ALL ×

Name (use keyword filter): ALL ×

Index: ALL ×

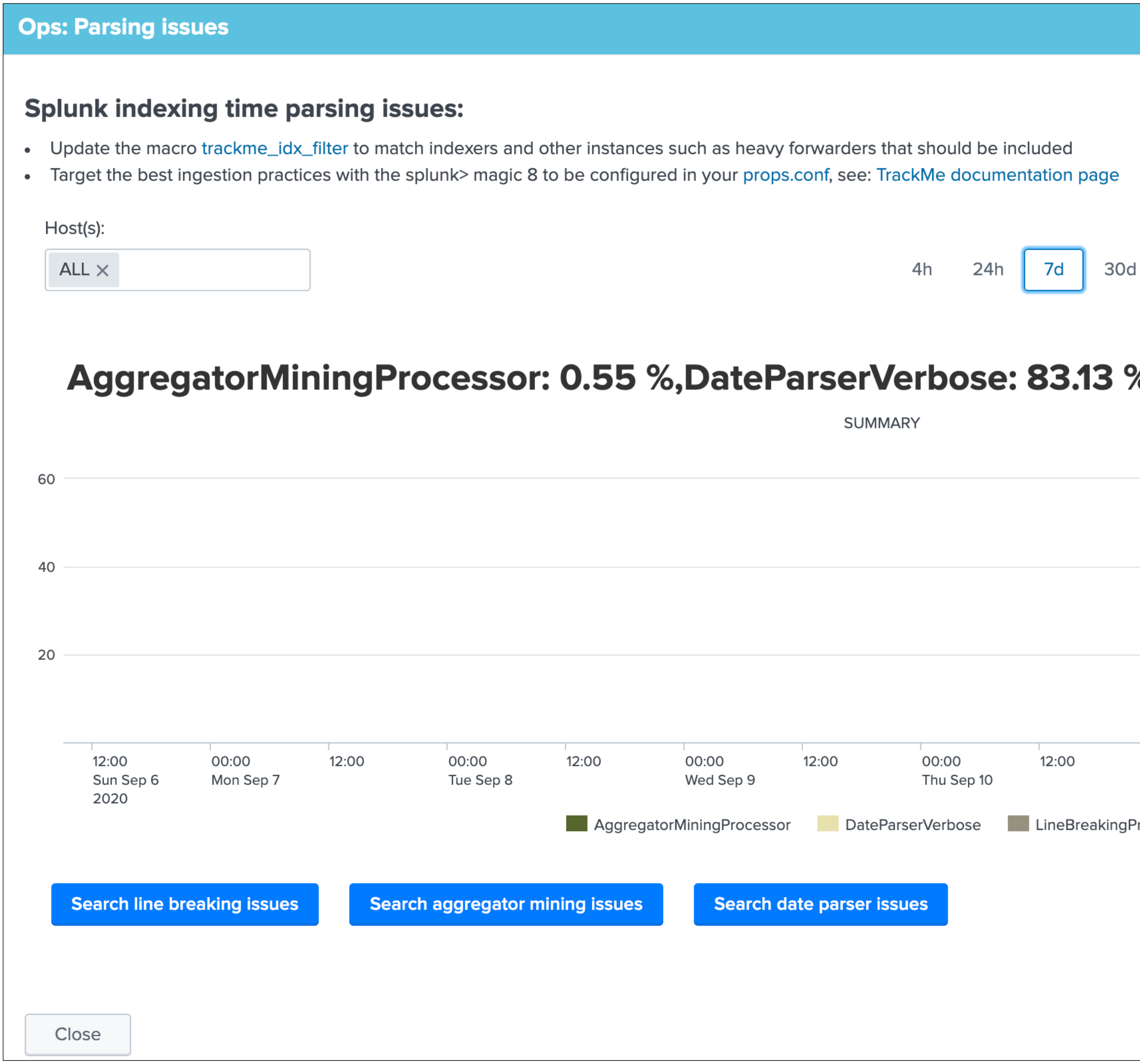
Sourcetype(s): ALL ×

Filter priority: ALL ×

Auto refresh: 5 min

Manage: elastic sources
Manage: allowlists & blocklists
Manage: define lagging classes
Manage: tags policies
Run: short term tracker now

This view shows parsing errors happening in your environment, using the filtering results from the macro trackme_idx_filter, make sure this macro is configured to filter on indexers and heavy forwarders:



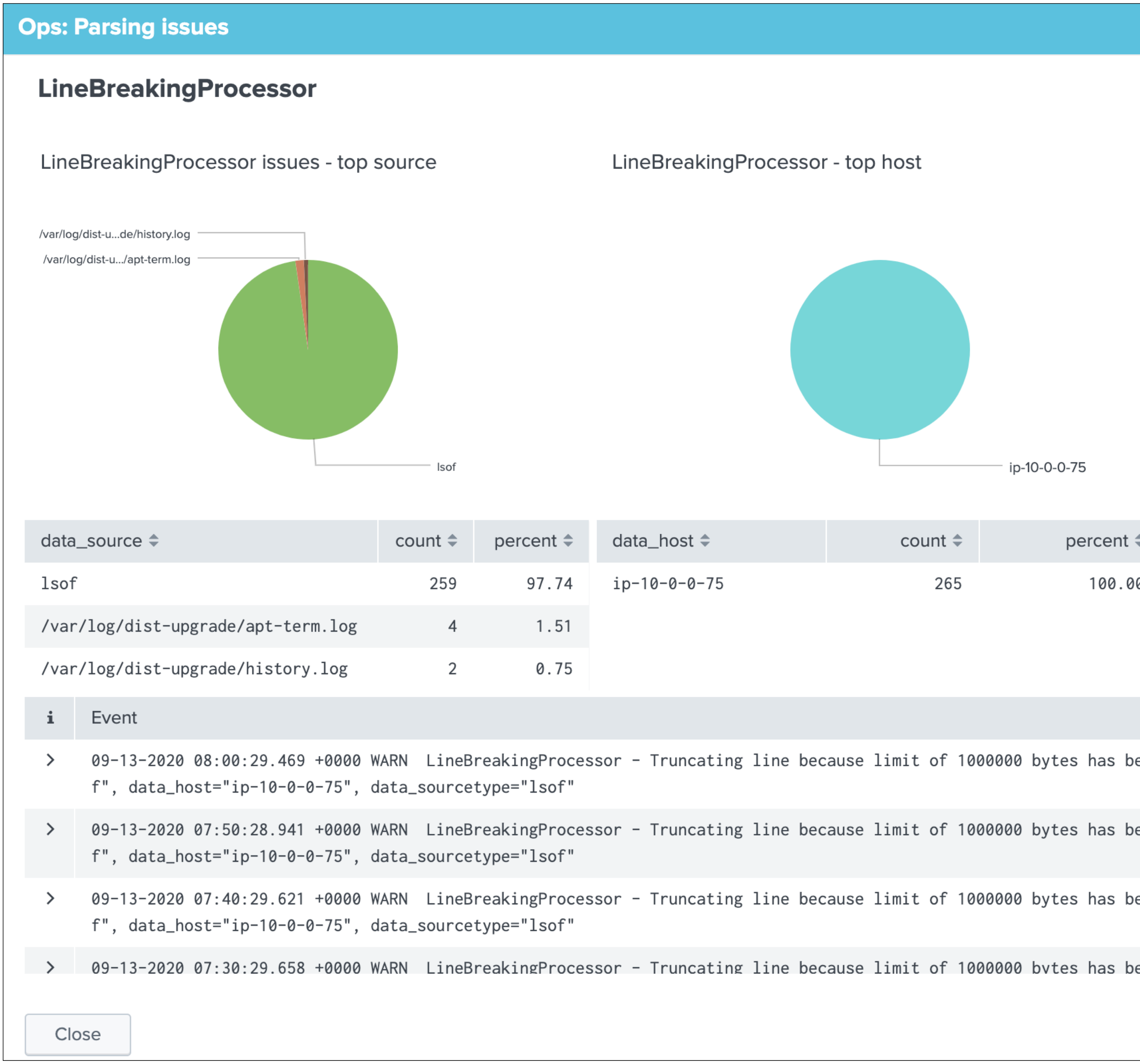
Ops: Parsing issues

Splunk indexing time parsing issues:

- Update the macro trackme_idx_filter to match indexers and other instances such as heavy forwarders that should be included
- Target the best ingestion practices with the splunk> magic 8 to be configured in your props.conf, see: TrackMe documentation page

Host(s): ALL ×

AggregatorMiningProcessor: 0.55 %,DateParserVerbose: 83.13 %


/var/log/dist-u...de/history.log
/var/log/dist-u.../apt-term.log
lsof
ip-10-0-0-75
Close

Ops: Parsing issues

LineBreakingProcessor

LineBreakingProcessor issues - top source

LineBreakingProcessor - top host

data_source	count	percent
lsof	259	97.74
/var/log/dist-upgrade/apt-term.log	4	1.51
/var/log/dist-upgrade/history.log	2	0.75

data_host	count	percent
ip-10-0-0-75	265	100.0

i	Event
>	09-13-2020 08:00:29.469 +0000 WARN LineBreakingProcessor - Truncating line because limit of 1000000 bytes has be f", data_host="ip-10-0-0-75", data_sourcetype="lsof"
>	09-13-2020 07:50:28.941 +0000 WARN LineBreakingProcessor - Truncating line because limit of 1000000 bytes has be f", data_host="ip-10-0-0-75", data_sourcetype="lsof"
>	09-13-2020 07:40:29.621 +0000 WARN LineBreakingProcessor - Truncating line because limit of 1000000 bytes has be f", data_host="ip-10-0-0-75", data_sourcetype="lsof"
>	09-13-2020 07:30:29.658 +0000 WARN LineBreakingProcessor - Truncating line because limit of 1000000 bytes has be

Options in the view:

- You can use the multiselect form to choose instances to be considered
- You can select a time range between the provided options
- Scroll down within the window to review the top root causes of the parsing issues

Splunk 8 magic props configuration

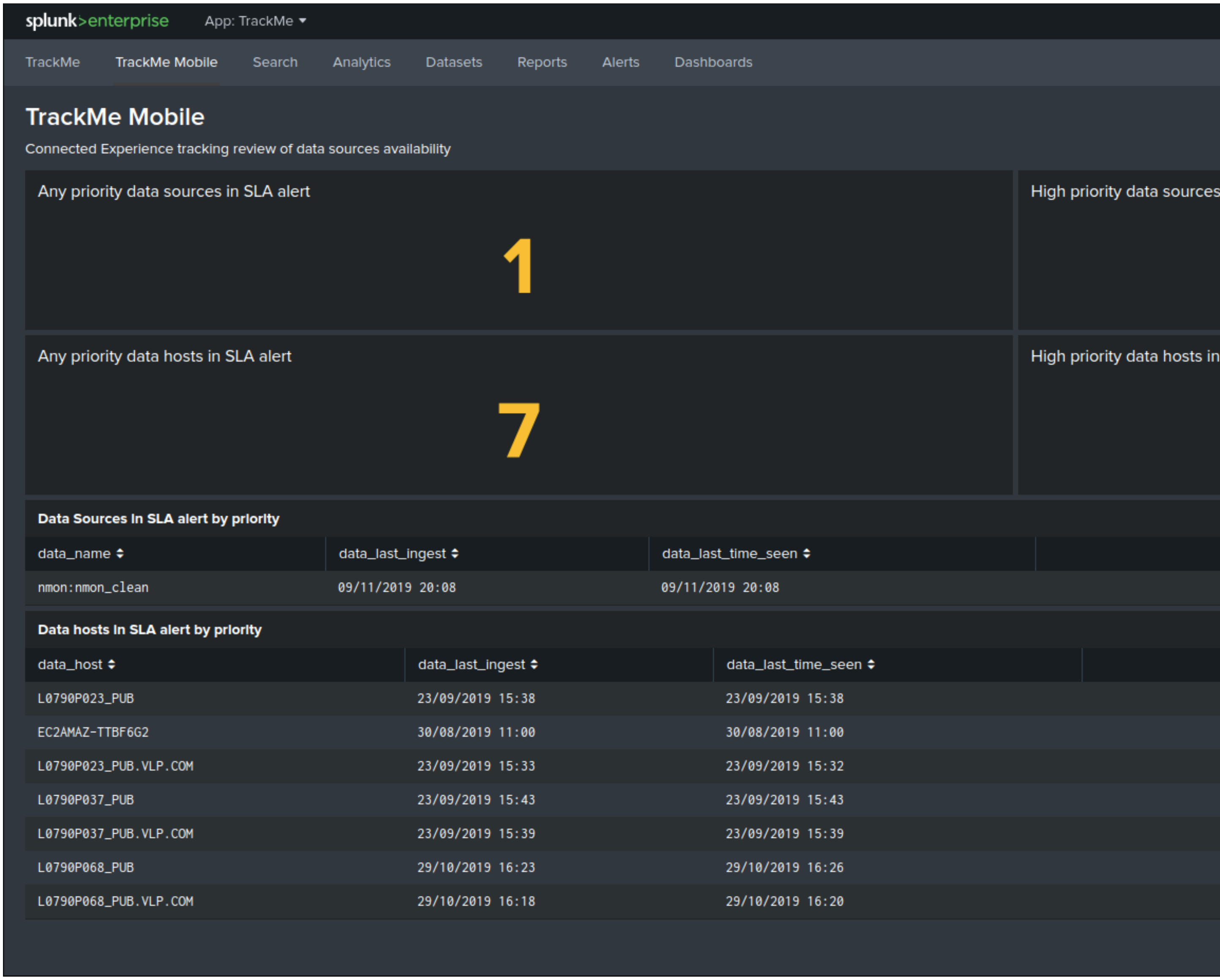
The "Splunk> magic 8" are good practice configuration items to be configured in your props.conf for the best performing and the best quality sourcetype definition:

```
[mySourcetype]

TIME_PREFIX = regex of the text that leads up to the timestamp

MAX_TIMESTAMP_LOOKAHEAD = how many characters for the timestamp

TIME_FORMAT = strftime format of the timestamp
# for multiline events: SHOULD_LINEMERGE should always be set to false as LINE_
↪BREAKER will speed up multiline events

SHOULD_LINEMERGE = false
# Wherever the LINE_BREAKER regex matches, Splunk considers the start
# of the first capturing group to be the end of the previous event
# and considers the end of the first capturing group to be the start of the next␣
↪event.
# Defaults to ([\r\n]+), meaning data is broken into an event for each line

LINE_BREAKER = regular expression for event breaks

TRUNCATE = 0
# Use the following attributes to handle better load balancing from UF.
# Please note the EVENT_BREAKER properties are applicable for Splunk Universal
# Forwarder instances only. Valid with forwarders > 6.5.0

EVENT_BREAKER_ENABLE = true

EVENT_BREAKER = regular expression for event breaks
```

This configuration represents the ideal sourcetype definition for Splunk, combining an explicit and controled definition for a reliable event breaking and time stamp recognition, as much as it is possible you should always target this configuration.

3.1.26 Connected experience dashboard for Splunk Mobile & Apple TV

TrackMe provides a connected experience dashboard for Splunk Cloud Gateway, that can be displayed on Mobile applications & Apple TV:



TrackMe Mobile

Connected Experience tracking review of data sources availability

Any priority data sources in SLA alert

1

High priority data sources

Any priority data hosts in SLA alert

7

High priority data hosts in

Data Sources in SLA alert by priority

data_name	data_last_ingest	data_last_time_seen
nmon:nmon_clean	09/11/2019 20:08	09/11/2019 20:08

Data hosts in SLA alert by priority

data_host	data_last_ingest	data_last_time_seen
L0790P023_PUB	23/09/2019 15:38	23/09/2019 15:38
EC2AMAZ-TTBF6G2	30/08/2019 11:00	30/08/2019 11:00
L0790P023_PUB.VLP.COM	23/09/2019 15:33	23/09/2019 15:32
L0790P037_PUB	23/09/2019 15:43	23/09/2019 15:43
L0790P037_PUB.VLP.COM	23/09/2019 15:39	23/09/2019 15:39
L0790P068_PUB	29/10/2019 16:23	29/10/2019 16:26
L0790P068_PUB.VLP.COM	29/10/2019 16:18	29/10/2019 16:20

This dashboard is exported to the system, to be made available to Splunk Cloud Gateway.

3.1.27 Team working with trackMe alerts and audit changes flow tracker

Nowadays it is very convenient to have team workspaces (Slack, Webex Teams, MS-Teams…) where people and applications can interact.

Fortunately, Splunk with alert actions and addon extensions allows interacting with any kind of platform, TrackMe makes it very handy with the following alerts:

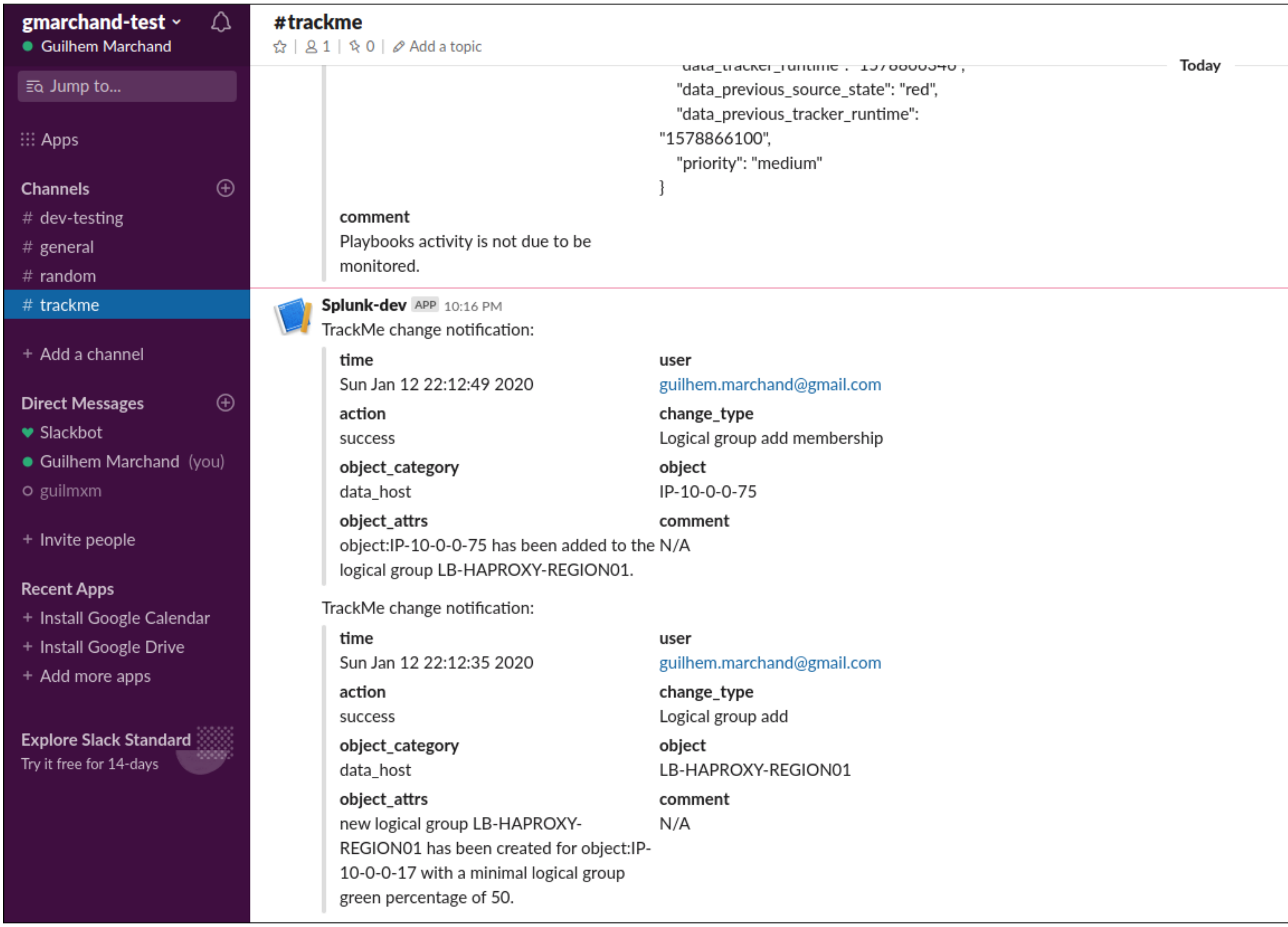
Out of the box alerts can be communicating when potential issues data sources, hosts or metric hosts are detected:

- `TrackMe - Alert on data source availability`
- `TrackMe - Alert on data host availability`
- `TrackMe - Alert on metric host availability`

In addition, the notification change tracker allows sharing automatically updates performed by administrators, which could be sent to a dedicated channel:

- TrackMe - Audit change notification tracker

Example in a Slack channel:



```
    "data_tracker_runtime": "1578866340",
    "data_previous_source_state": "red",
    "data_previous_tracker_runtime":
"1578866100",
    "priority": "medium"
}
```

comment
Playbooks activity is not due to be monitored.

Splunk-dev APP 10:16 PM
TrackMe change notification:

time Sun Jan 12 22:12:49 2020	**user** guilhem.marchand@gmail.com
action success	**change_type** Logical group add membership
object_category data_host	**object** IP-10-0-0-75
object_attrs object:IP-10-0-0-75 has been added to the logical group LB-HAPROXY-REGION01.	**comment** N/A

TrackMe change notification:

time Sun Jan 12 22:12:35 2020	**user** guilhem.marchand@gmail.com
action success	**change_type** Logical group add
object_category data_host	**object** LB-HAPROXY-REGION01
object_attrs new logical group LB-HAPROXY-REGION01 has been created for object:IP-10-0-0-17 with a minimal logical group green percentage of 50.	**comment** N/A

For Slack integration, see

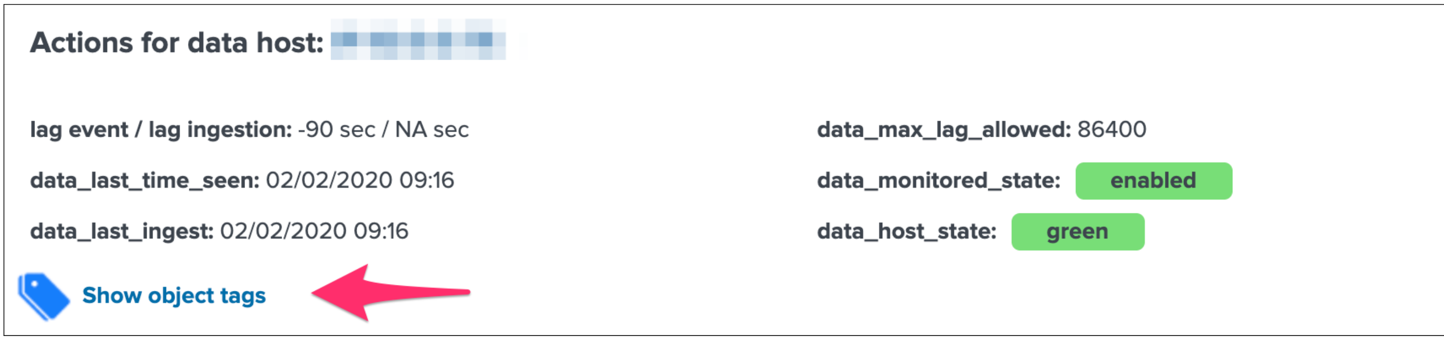
- https://splunkbase.splunk.com/app/2878

Many more integration are available on Splunk Base.

3.1.28 Enrichment tags

Enrichment tags

Enrichment tags are available for data and metric hosts to provide context for your assets based on the assets data available in your Splunk deployment.



Actions for data host: [redacted]

lag event / lag ingestion: -90 sec / NA sec

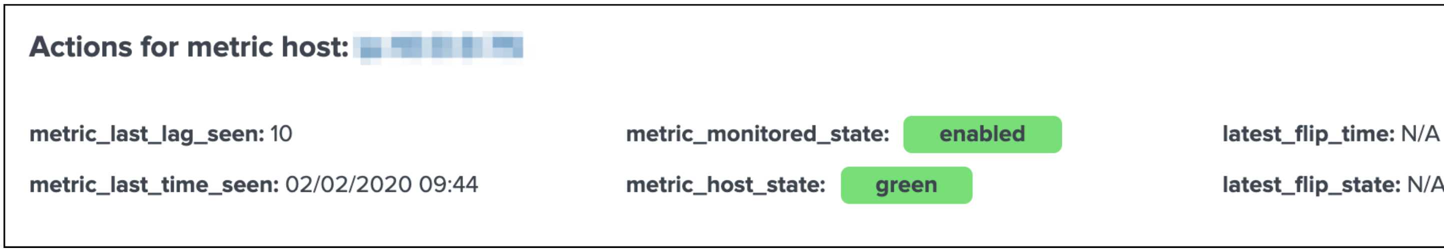
data_last_time_seen: 02/02/2020 09:16

data_last_ingest: 02/02/2020 09:16

data_max_lag_allowed: 86400

data_monitored_state: enabled

data_host_state: green



Actions for metric host: [redacted]

metric_last_lag_seen: 10

metric_last_time_seen: 02/02/2020 09:44

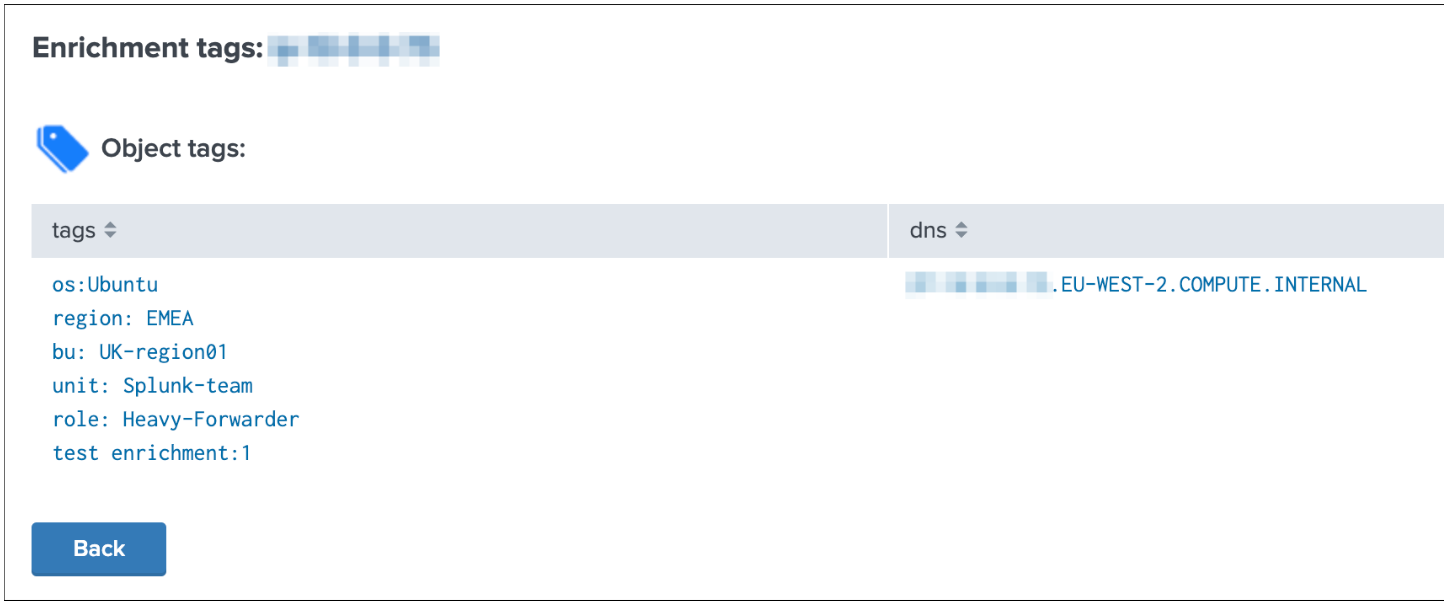
metric_monitored_state: enabled

metric_host_state: green

latest_flip_time: N/A

latest_flip_state: N/A

Once configured, enrichment tags provides access to your assets information to help analyst identifying the entities in alert and facilitate further investigations:



Enrichment tags: [redacted]

Object tags:

tags	dns
os:Ubuntu region: EMEA bu: UK-region01 unit: Splunk-team role: Heavy-Forwarder test enrichment:1	[redacted].EU-WEST-2.COMPUTE.INTERNAL

3.1.29 Maintenance mode

Maintenance mode

The maintenance mode feature provides a builtin workflow to temporary silent all alerts from TrackMe for a given period of time, which can be scheduled in advance.

All alerts are by default driven by the status of the maintenance mode stored in a KVstore collection.

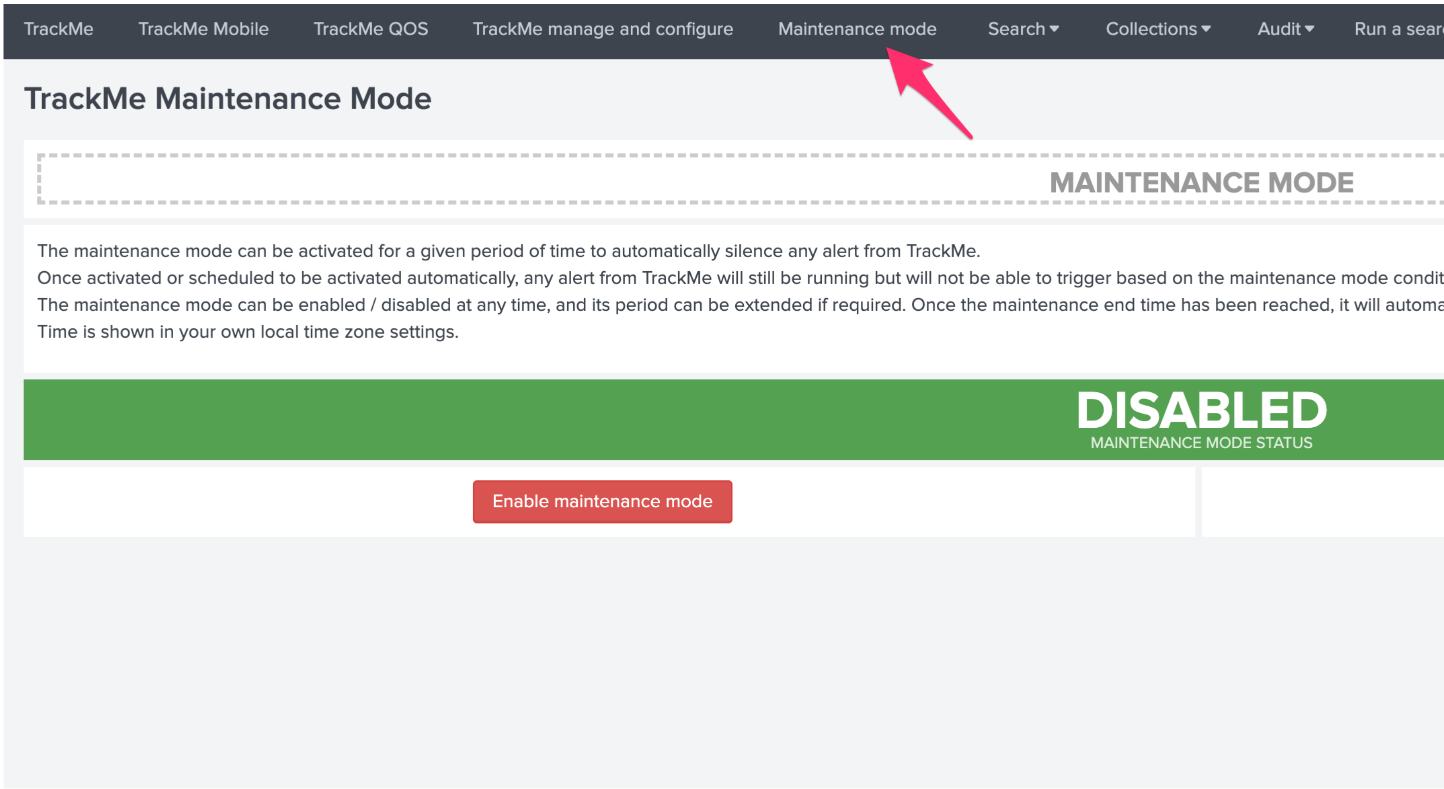
Shall the maintenance be enabled by an administrator, Splunk will continue to run the schedule alerts but none of them will be able to trigger during the maintenance time window.

When the end of maintenance time window is reached, its state will be automatically disabled and alerts will be able to trigger again.

A maintenance time window can start immediately, or be can be scheduled according to your selection.

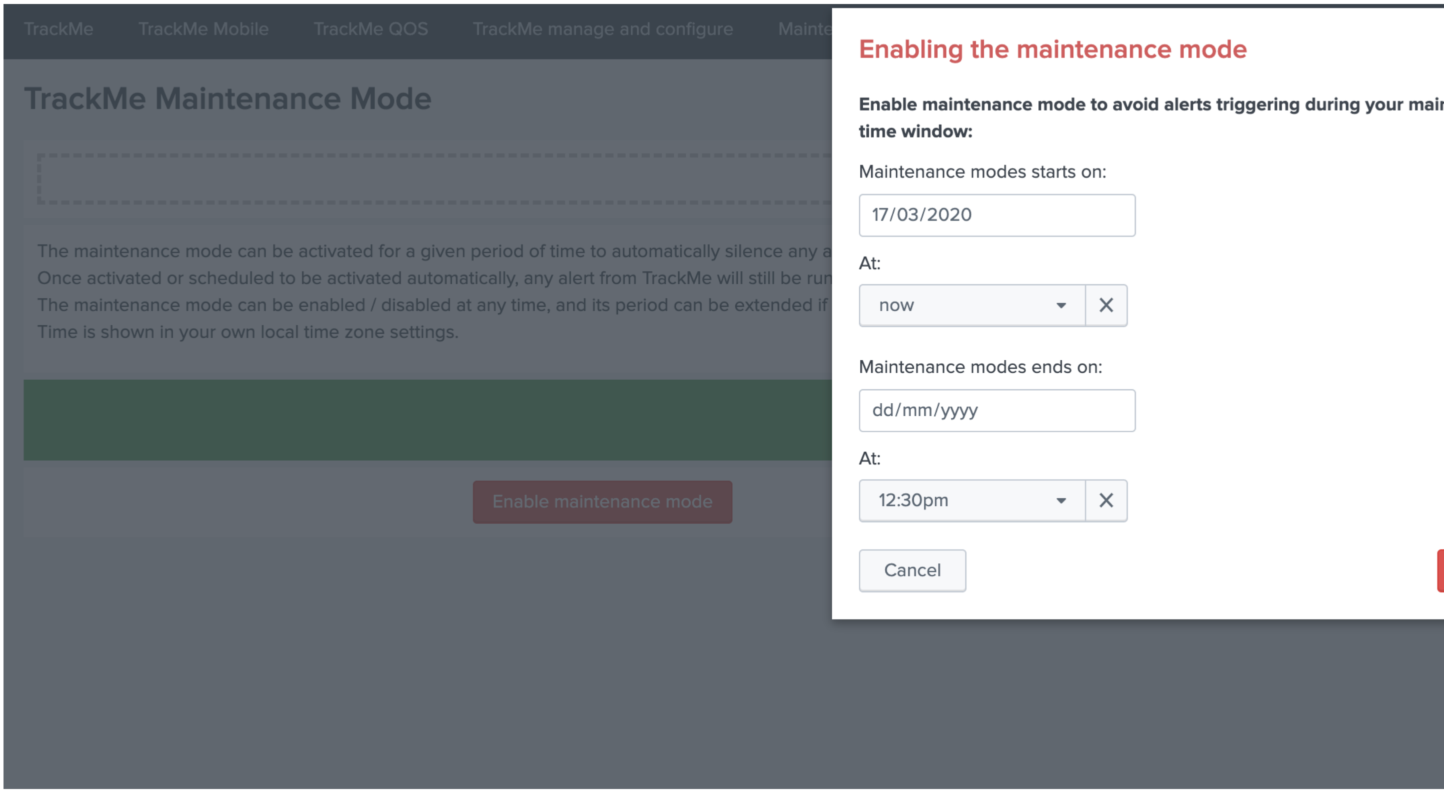
Enabling or extending the maintenance mode

- Click on the enable maintenance mode button:



The maintenance mode can be activated for a given period of time to automatically silence any alert from TrackMe.
Once activated or scheduled to be activated automatically, any alert from TrackMe will still be running but will not be able to trigger based on the maintenance mode condit
The maintenance mode can be enabled / disabled at any time, and its period can be extended if required. Once the maintenance end time has been reached, it will automa
Time is shown in your own local time zone settings.

- Within the modal configuration window, enter the date and hours of the end of the maintenance time window:



TrackMe Maintenance Mode

The maintenance mode can be activated for a given period of time to automatically silence any a
Once activated or scheduled to be activated automatically, any alert from TrackMe will still be run
The maintenance mode can be enabled / disabled at any time, and its period can be extended if
Time is shown in your own local time zone settings.

Enabling the maintenance mode

Enable maintenance mode to avoid alerts triggering during your mai time window:

Maintenance modes starts on: 17/03/2020

At: now

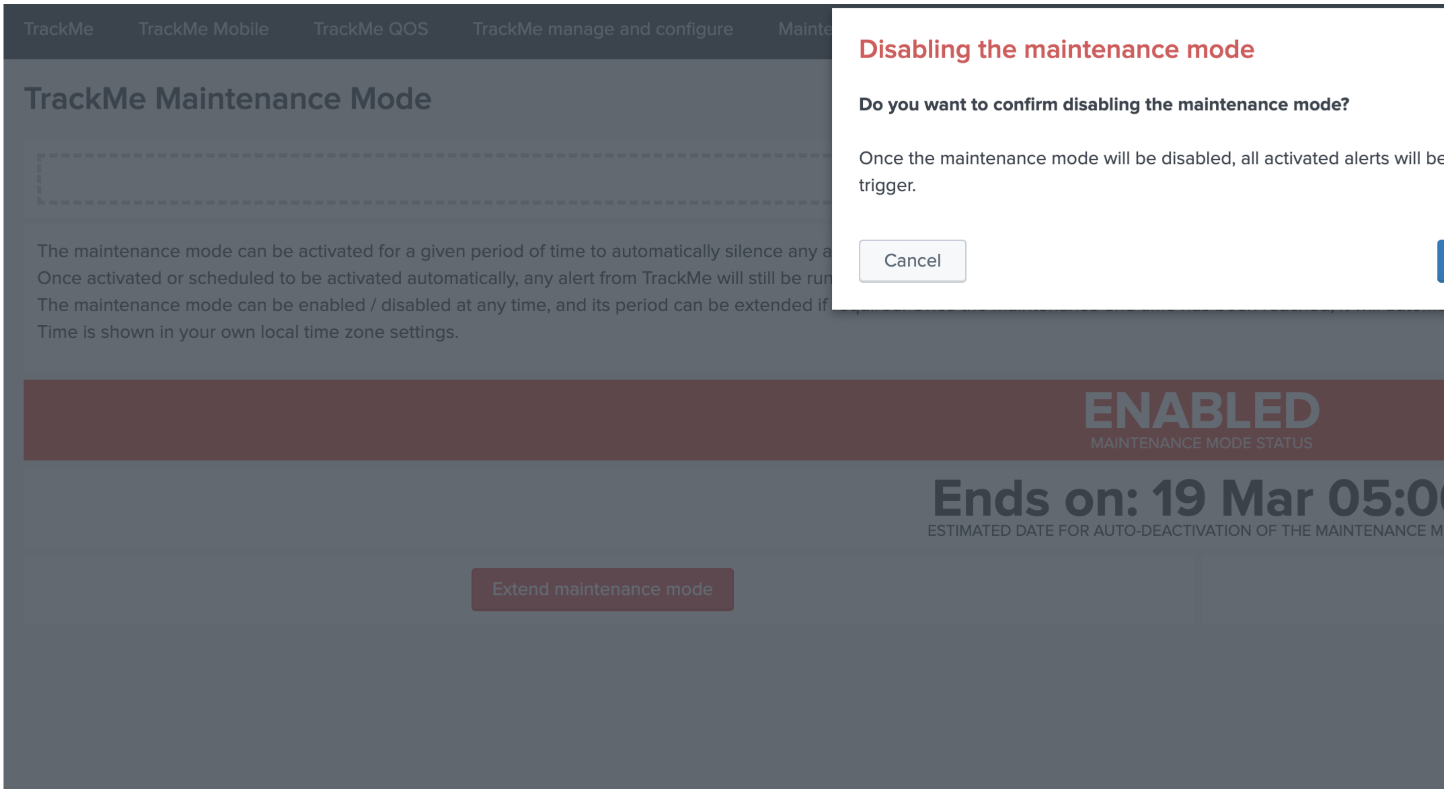
Maintenance modes ends on: dd/mm/yyyy

At: 12:30pm

- When the date and hours of the maintenance time window are reached, the scheduled report "Verify Kafka alerting maintenance status" will automatically disable the maintenance mode.
- If a start date time different than the current time is selected (default), this action will automatically schedule the maintenance time window.

Disabling the maintenance mode

During any time of the maintenance time window, an administrator can decide to disable the maintenance mode:



TrackMe Maintenance Mode

The maintenance mode can be activated for a given period of time to automatically silence any a
Once activated or scheduled to be activated automatically, any alert from TrackMe will still be run
The maintenance mode can be enabled / disabled at any time, and its period can be extended if
Time is shown in your own local time zone settings.

Disabling the maintenance mode

Do you want to confirm disabling the maintenance mode?

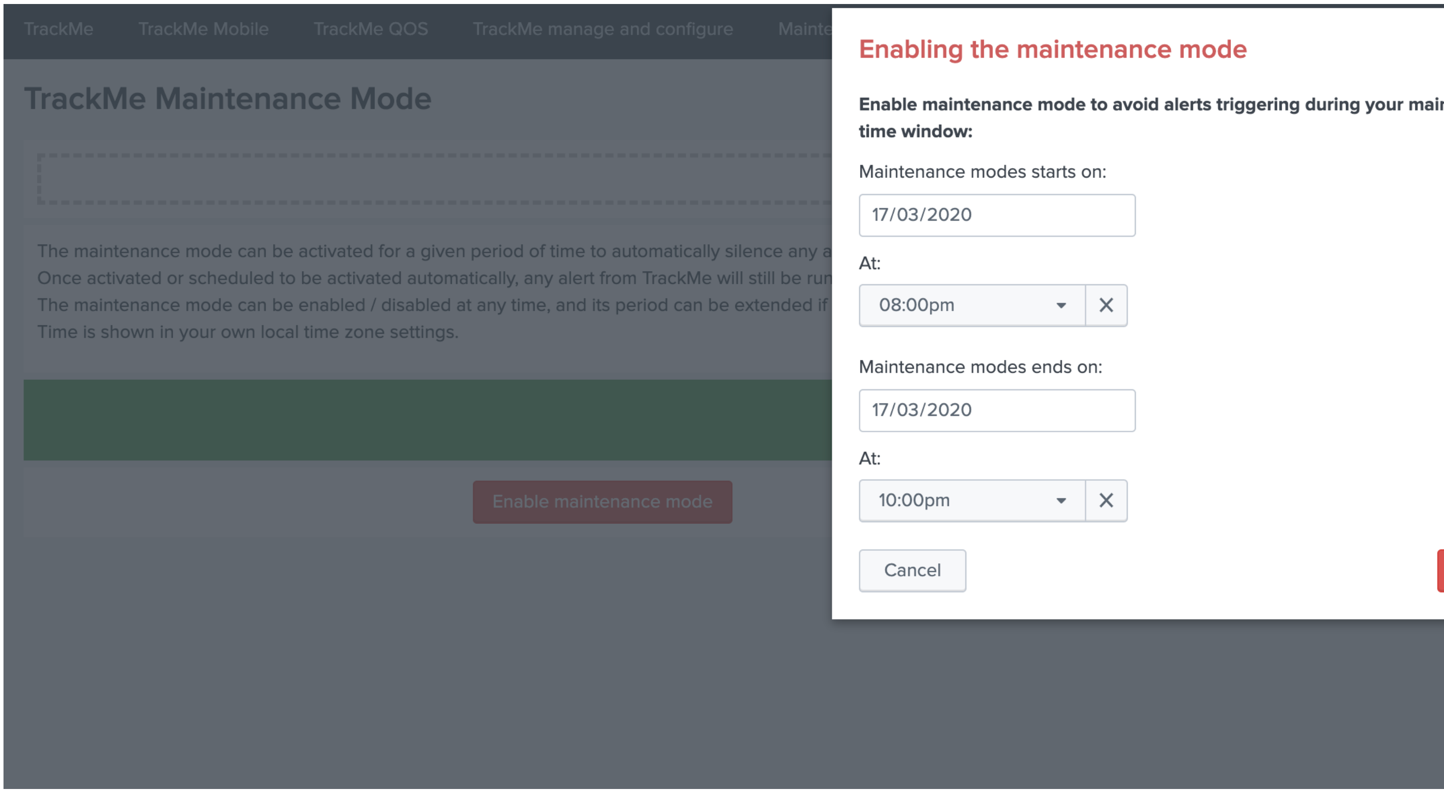
Once the maintenance mode will be disabled, all activated alerts will be
trigger.

Ends on: 19 Mar 05:0
ESTIMATED DATE FOR AUTO-DEACTIVATION OF THE MAINTENANCE M

Scheduling a maintenance window

You can configure the maintenance mode to be automatically enabled between a specific date time that you enter in the UI.

- When the end time is reached, the maintenance mode will automatically be disable, and the alerting will return to normal operations.



TrackMe Maintenance Mode

The maintenance mode can be activated for a given period of time to automatically silence any a
Once activated or scheduled to be activated automatically, any alert from TrackMe will still be run
The maintenance mode can be enabled / disabled at any time, and its period can be extended if
Time is shown in your own local time zone settings.

Enabling the maintenance mode

Enable maintenance mode to avoid alerts triggering during your mai time window:

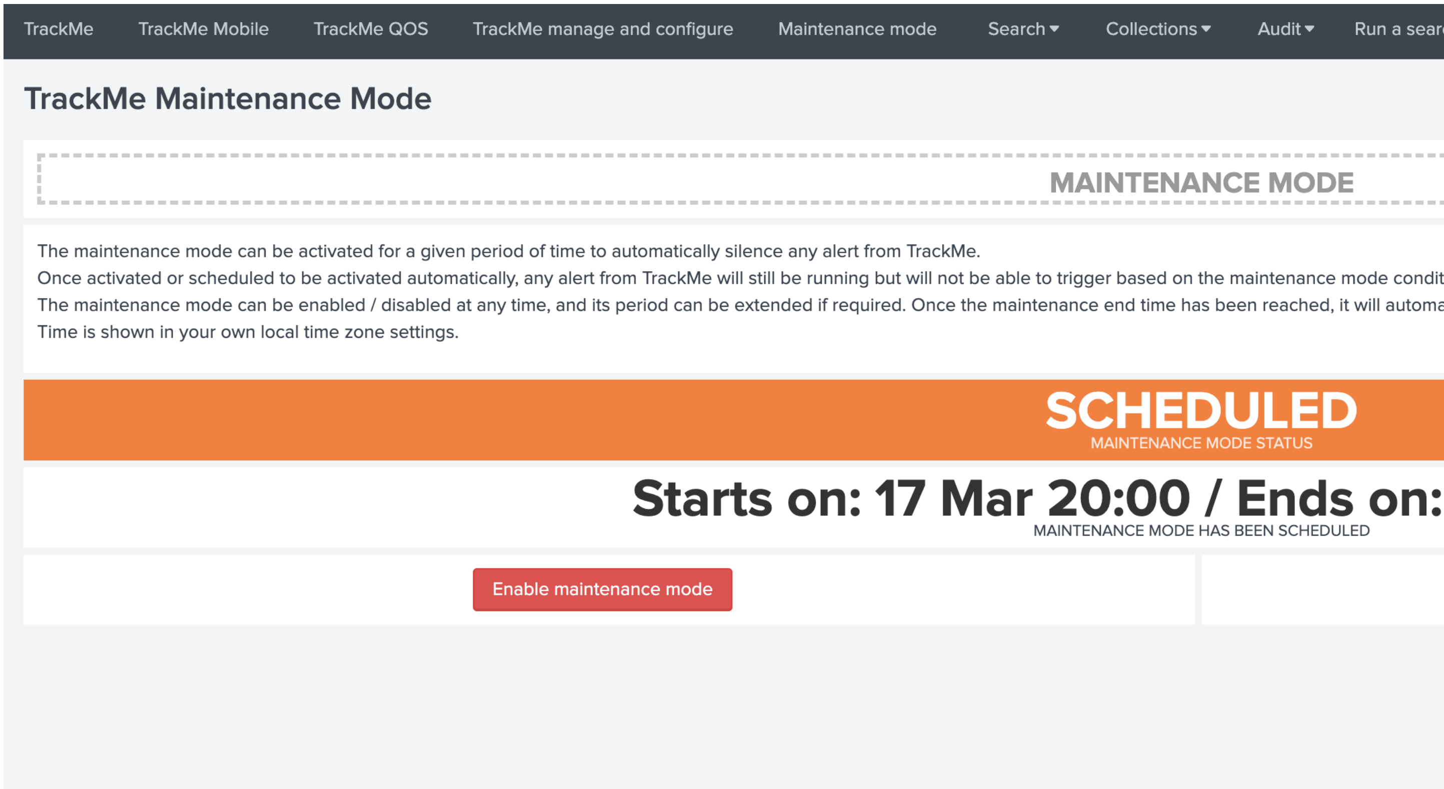
Maintenance modes starts on: 17/03/2020

At: 08:00pm

Maintenance modes ends on: 17/03/2020

At: 10:00pm

- When a maintenance mode window has been scheduled, the UI shows a specific message with the starts / ends on dates:



TrackMe Maintenance Mode

The maintenance mode can be activated for a given period of time to automatically silence any alert from TrackMe.
Once activated or scheduled to be activated automatically, any alert from TrackMe will still be running but will not be able to trigger based on the maintenance mode condit
The maintenance mode can be enabled / disabled at any time, and its period can be extended if required. Once the maintenance end time has been reached, it will automa
Time is shown in your own local time zone settings.

SCHEDULED
MAINTENANCE MODE STATUS

Starts on: 17 Mar 20:00 / Ends on:
MAINTENANCE MODE HAS BEEN SCHEDULED

3.1.30 Backup and restore

TrackMe stores the vaste majority of its content in multiple KVstore collections.

Using the *Backup and Restore endpoints* from the API, backups are taken automatically on a scheduled basis, can be taken on demand and restored if necessary.

Backups are stored in compressed tarball archives, located in the "backup" directory of the TrackMe application on the search head(s):

Example:

```
/opt/splunk/etc/apps/trackme/backup/trackme-backup-20210205-142635.tgz
```

Each archive contains a JSON file corresponding to the entire content of the KVstore collection when the backup was taken, empty collections are not backed up.

To perform a restore operation (see the documentation following), the relevant tarball archive needs to be located in the same directory.

When a backup is taken, a record with Metadata is added in a dedicated KVstore collection (kv_trackme_backup_archives_info), records are automatically purged when the archive is deleted due to retention. (any missing archive record is as well added if discovered on a search head when a get backups command runs)

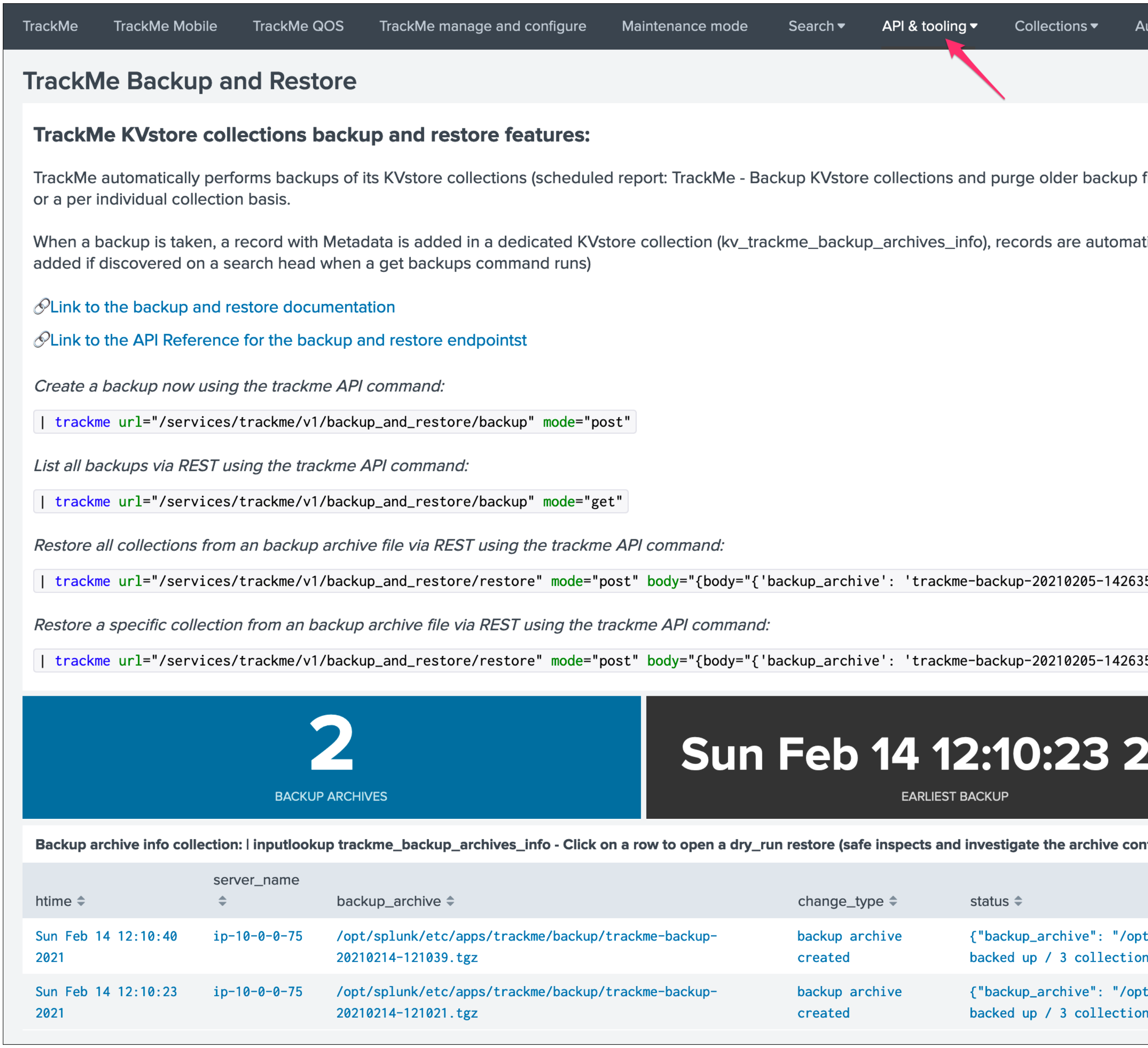
For Splunk Cloud certification purposes, the application will never attempt to write or access a directory ouf of the application name space level.

notes about Search Head Clustering (SHC)

- If TrackMe is deployed in a Search Head Cluster, the scheduled report is executed on a single search head, randomly
- As such, the archive file is created on this specific instance, but not replicated to other members
- Restoring requires to locate the server hosting the archive file using the audit dashboard or manually in the Metadata collection, and running the restore command from this node especially
- The restore operation does not mandatory requires to be executed from the SHC / KVstore captain
- in a SHC context, the purging part of schedule report happens only on the member running the report, therefore archive files can exist longer than the retention on other members

Backup and Restore dashboard

An auditing dashboard is provided in the app navigation menu "API & Tooling" that provides an overview of the backup archives knowledge and statuses:



TrackMe Backup and Restore

TrackMe KVstore collections backup and restore features:

TrackMe automatically performs backups of its KVstore collections (scheduled report: TrackMe - Backup KVstore collections and purge older backup f or a per individual collection basis.

When a backup is taken, a record with Metadata is added in a dedicated KVstore collection (kv_trackme_backup_archives_info), records are automati added if discovered on a search head when a get backups command runs)

🔗Link to the backup and restore documentation

🔗Link to the API Reference for the backup and restore endpointst

Create a backup now using the trackme API command:

```
| trackme url="/services/trackme/v1/backup_and_restore/backup" mode="post"
```

List all backups via REST using the trackme API command:

```
| trackme url="/services/trackme/v1/backup_and_restore/backup" mode="get"
```

Restore all collections from an backup archive file via REST using the trackme API command:

```
| trackme url="/services/trackme/v1/backup_and_restore/restore" mode="post" body="{body="{'backup_archive': 'trackme-backup-20210205-14263
```

Restore a specific collection from an backup archive file via REST using the trackme API command:

```
| trackme url="/services/trackme/v1/backup_and_restore/restore" mode="post" body="{body="{'backup_archive': 'trackme-backup-20210205-14263
```

2
BACKUP ARCHIVES

Sun Feb 14 12:10:23 2
EARLIEST BACKUP

Backup archive info collection: | inputlookup trackme_backup_archives_info - Click on a row to open a dry_run restore (safe inspects and investigate the archive con

htime	server_name	backup_archive	change_type	status
Sun Feb 14 12:10:40 2021	ip-10-0-0-75	/opt/splunk/etc/apps/trackme/backup/trackme-backup-20210214-121039.tgz	backup archive created	{"backup_archive": "/opt backed up / 3 collection
Sun Feb 14 12:10:23 2021	ip-10-0-0-75	/opt/splunk/etc/apps/trackme/backup/trackme-backup-20210214-121021.tgz	backup archive created	{"backup_archive": "/opt backed up / 3 collection

This dashboard uses the backup archives Metadata stores in the KVstore collection **trackme_backup_archives_info** to show the list of backups that were taken over time per instance.

Automatic backup

A Splunk report is scheduled by default to run every day at 2h AM:

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

This report does the following operations:

- call the trackme custom command API wrapper to take a backup of all non empty KVstore collections, generating an archive file in the search head the report is executed

- call the trackme custom command API wrapper to purge backup files older than 7 days (by default) in the search head the report is executed
- call the trackme custom command API wrapper to list backup files, and automatically discover any missing files in the knowledge collection

In SPL:

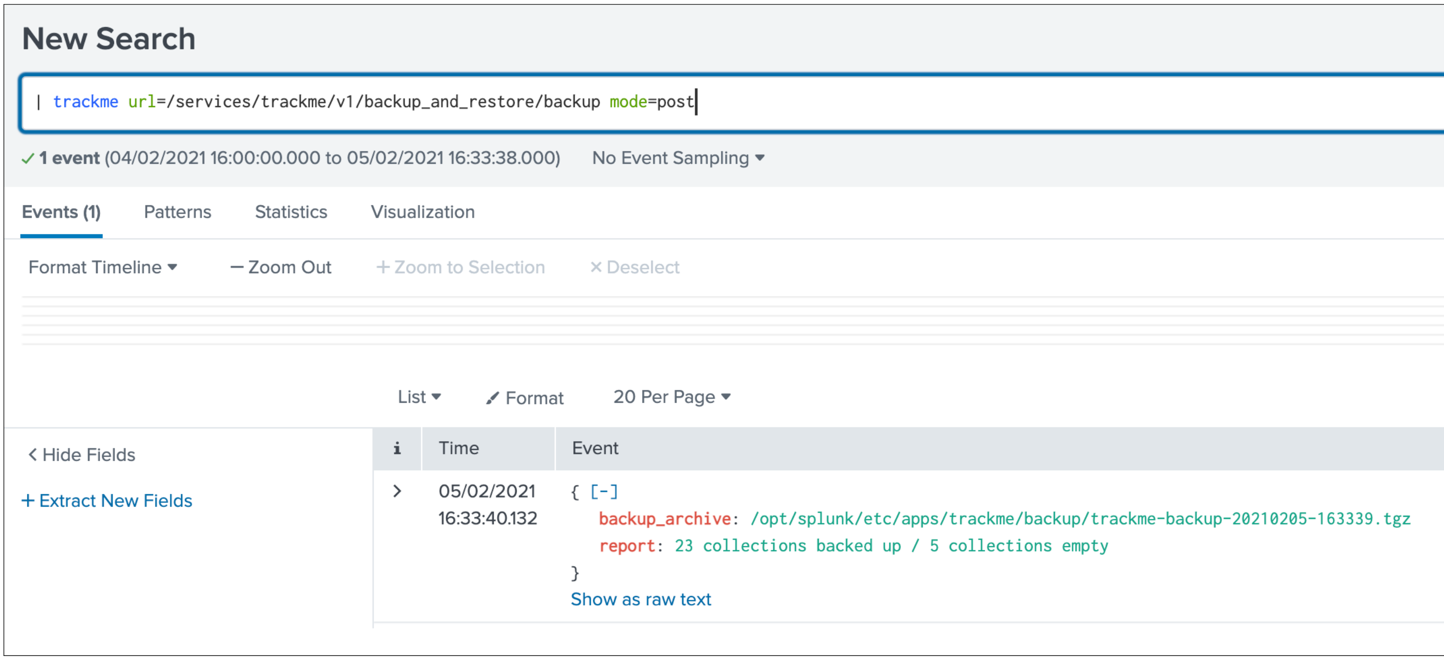
```
| trackme url=/services/trackme/v1/backup_and_restore/backup mode=post
| append [ | trackme url=/services/trackme/v1/backup_and_restore/backup mode=delete␣
↪body="{'retention_days': '7'}" ]
| append [ | trackme url=/services/trackme/v1/backup_and_restore/backup mode=get |␣
↪spath | eventstats dc({}.backup_archive) as backup_count, values({}.backup_archive)␣
↪as backup_files
| eval backup_count=if(isnull(backup_count), 0, backup_count), backup_
↪files=if(isnull(backup_files), "none", backup_files)
| eval report="List of identified or known backup files (" . backup_count . ")"
| eval _raw="{\"report\": \"" . report . "\", \"backup_files\": \" [ " .␣
↪mvjoin(backup_files, ",") . " ]\"}" ]
```

On demand backup

You can at anytime perform a backup of the KVstore collections by running the following SPL command:

```
| trackme url=/services/trackme/v1/backup_and_restore/backup mode=post
```

This command calls the *backup / Run backup KVstore collections* API endpoint, and produces the following output:



```
| trackme url=/services/trackme/v1/backup_and_restore/backup mode=post
```

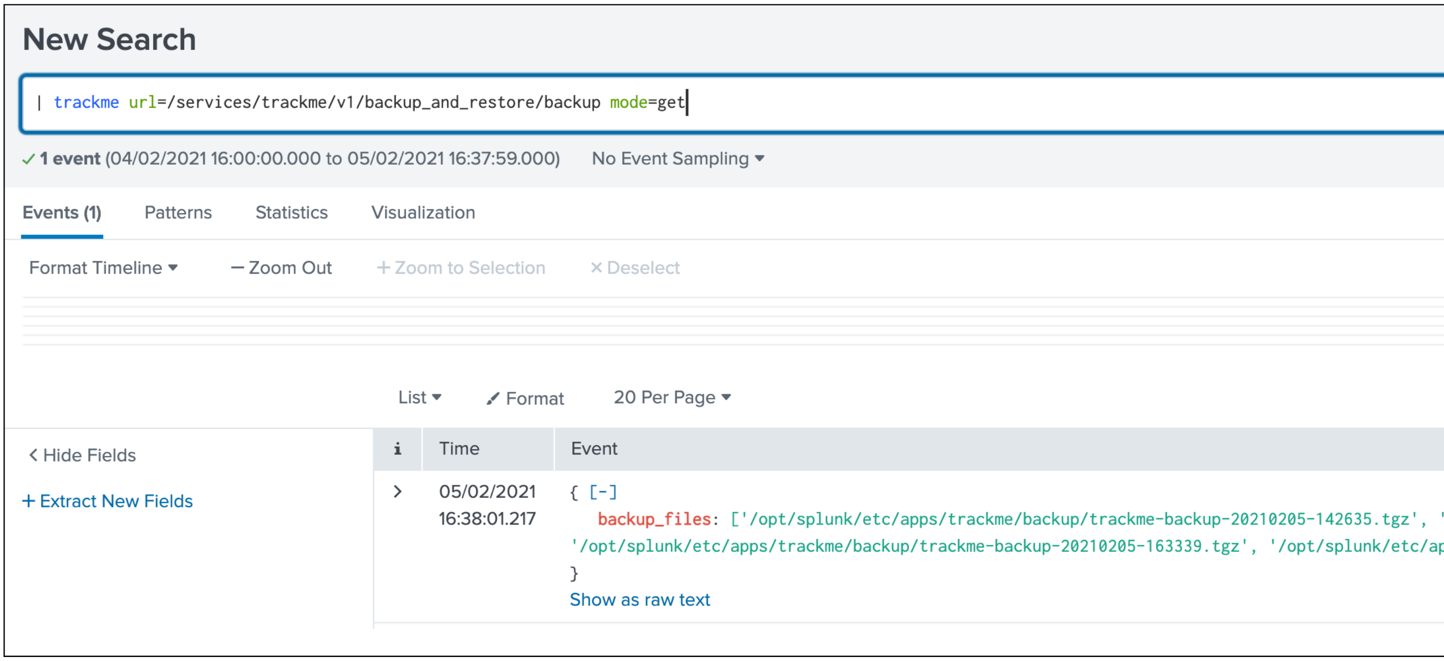
```
05/02/2021 16:33:40.132
{ [-]
   backup_archive: /opt/splunk/etc/apps/trackme/backup/trackme-backup-20210205-163339.tgz
   report: 23 collections backed up / 5 collections empty
}
```

List backup archives available

You can list the archive files available on the search head running the command using the following SPL command:

```
| trackme url=/services/trackme/v1/backup_and_restore/backup mode=get
```

This command calls the *backup / Purge older backup archive files* API endpoint, and produces the following output:



```
| trackme url=/services/trackme/v1/backup_and_restore/backup mode=get
```

✓ 1 event (04/02/2021 16:00:00.000 to 05/02/2021 16:37:59.000) No Event Sampling ▾

i	Time	Event
>	05/02/2021 16:38:01.217	{ [-] backup_files: ['/opt/splunk/etc/apps/trackme/backup/trackme-backup-20210205-142635.tgz', ' '/opt/splunk/etc/apps/trackme/backup/trackme-backup-20210205-163339.tgz', '/opt/splunk/etc/ap }

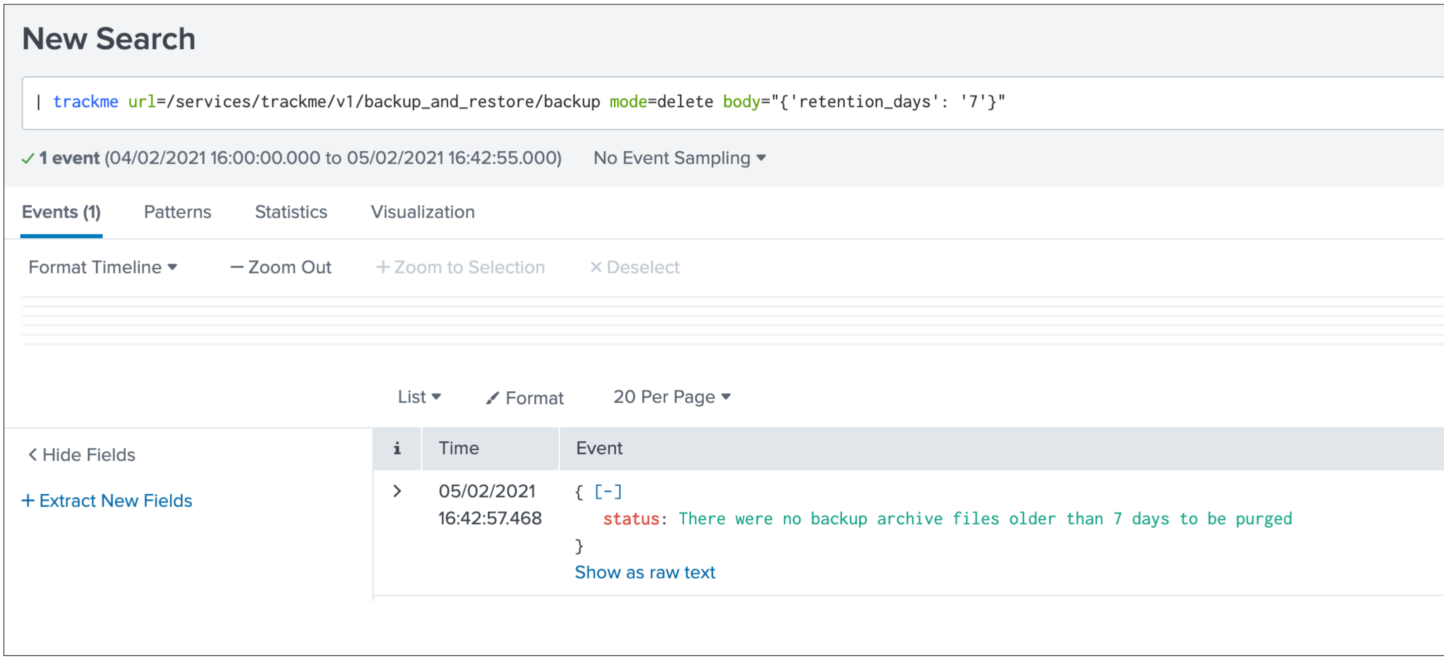
All archive files available on the search head the command is executed are listed with their full path on the file system.

Purge older backup archive

You can purge older archive files based on their creation time on the search head running the command using the following SPL command:

```
| trackme url=/services/trackme/v1/backup_and_restore/backup mode=delete body="{
↪'retention_days': '7'}"
```

This command calls the *backup / Purge older backup archive files* API endpoint, and produces the following output:



```
| trackme url=/services/trackme/v1/backup_and_restore/backup mode=delete body="{'retention_days': '7'}"
```

```
{ [-]
   status: There were no backup archive files older than 7 days to be purged
}
```

Depending on either there are no eligible archives, the response above would appear, or the list of archives that were purged will be rendered.

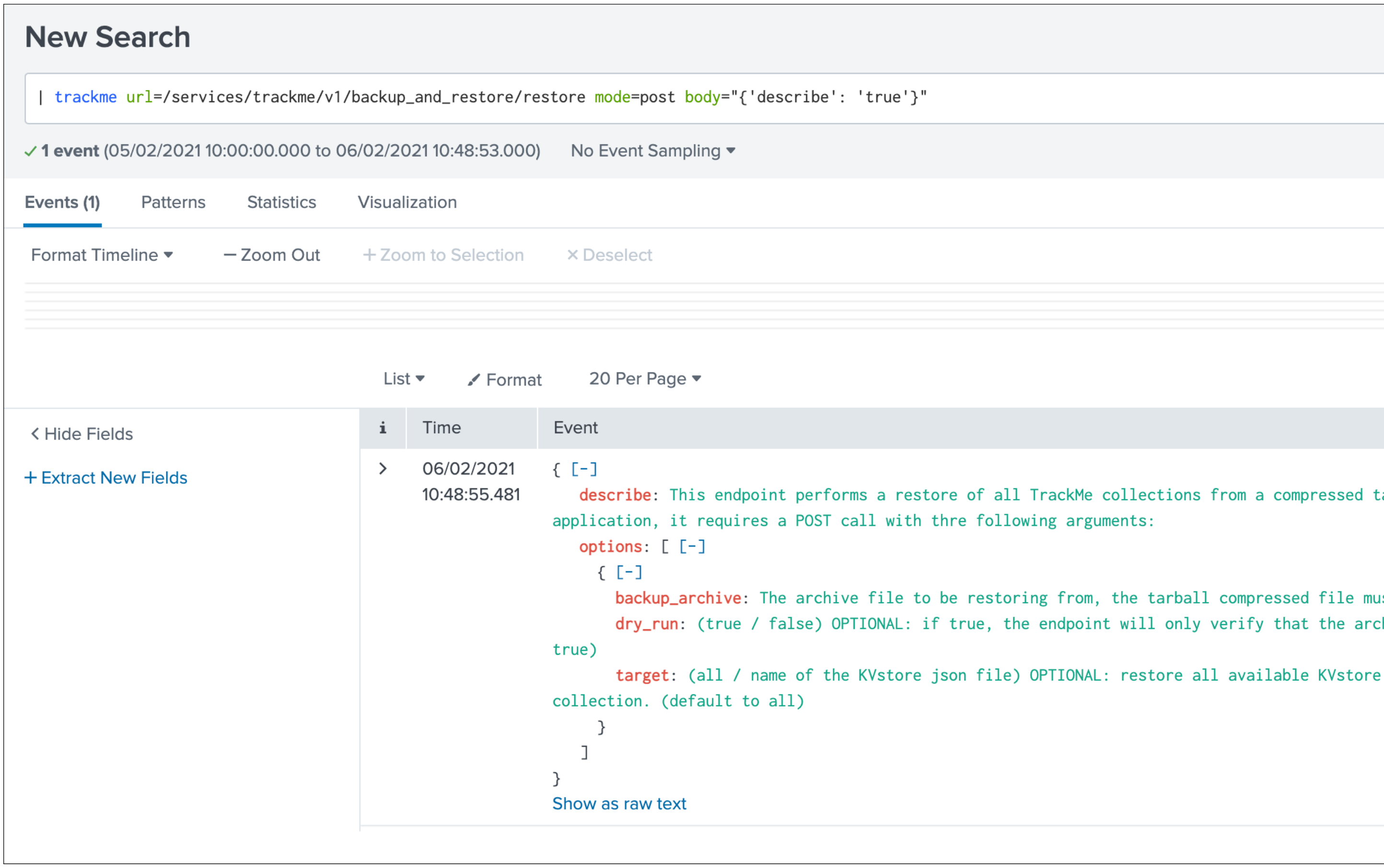
Restoring a backup

Warning: Restoring means the content of all KVstore collections will be permanently lost and replaced by the backup, use with precautions!

- Splunk API limits by default the max number of document per batch to 1000
- trackMe uses a chunk approach that limits to 500 document per API call
- To be able to perform a restore operation, ensure that limits.conf / kvstore / max_documents_per_batch_save is equal or superior to 500

Restoring relies on the *restore / Perform a restore of KVstore collections* API endpoint, which can be actionned via the `trackme` command, you can list the options:

```
| trackme url=/services/trackme/v1/backup_and_restore/restore mode=post body="{
↪'describe': 'true'}"
```



```
| trackme url=/services/trackme/v1/backup_and_restore/restore mode=post body="{'describe': 'true'}"
```

✓ 1 event (05/02/2021 10:00:00.000 to 06/02/2021 10:48:53.000)

06/02/2021 10:48:55.481

```
{ [-]
   describe: This endpoint performs a restore of all TrackMe collections from a compressed ta
application, it requires a POST call with thre following arguments:
   options: [ [-]
     { [-]
       backup_archive: The archive file to be restoring from, the tarball compressed file mus
       dry_run: (true / false) OPTIONAL: if true, the endpoint will only verify that the arch
true)
       target: (all / name of the KVstore json file) OPTIONAL: restore all available KVstore
collection. (default to all)
     }
   ]
}
```

dry_run mode

By default, the restore endpoint acts in dry_run mode, this means that the backend performs verifications **without applying any kind of modifications**:

- verify that the submitted archive tarball exists on the file system
- verify that the archive can be uncompressed effectively

It is actioned via the argument `dry_run` to be set to `true` (which is the default), or `false` which invovles performing the restore operation for real.

target for restore

By default, the restore operation clears every KVstore collection, and restore collections from the JSON files contained in the backup archive.

This is driven by the argument `target` which accepts the following options:

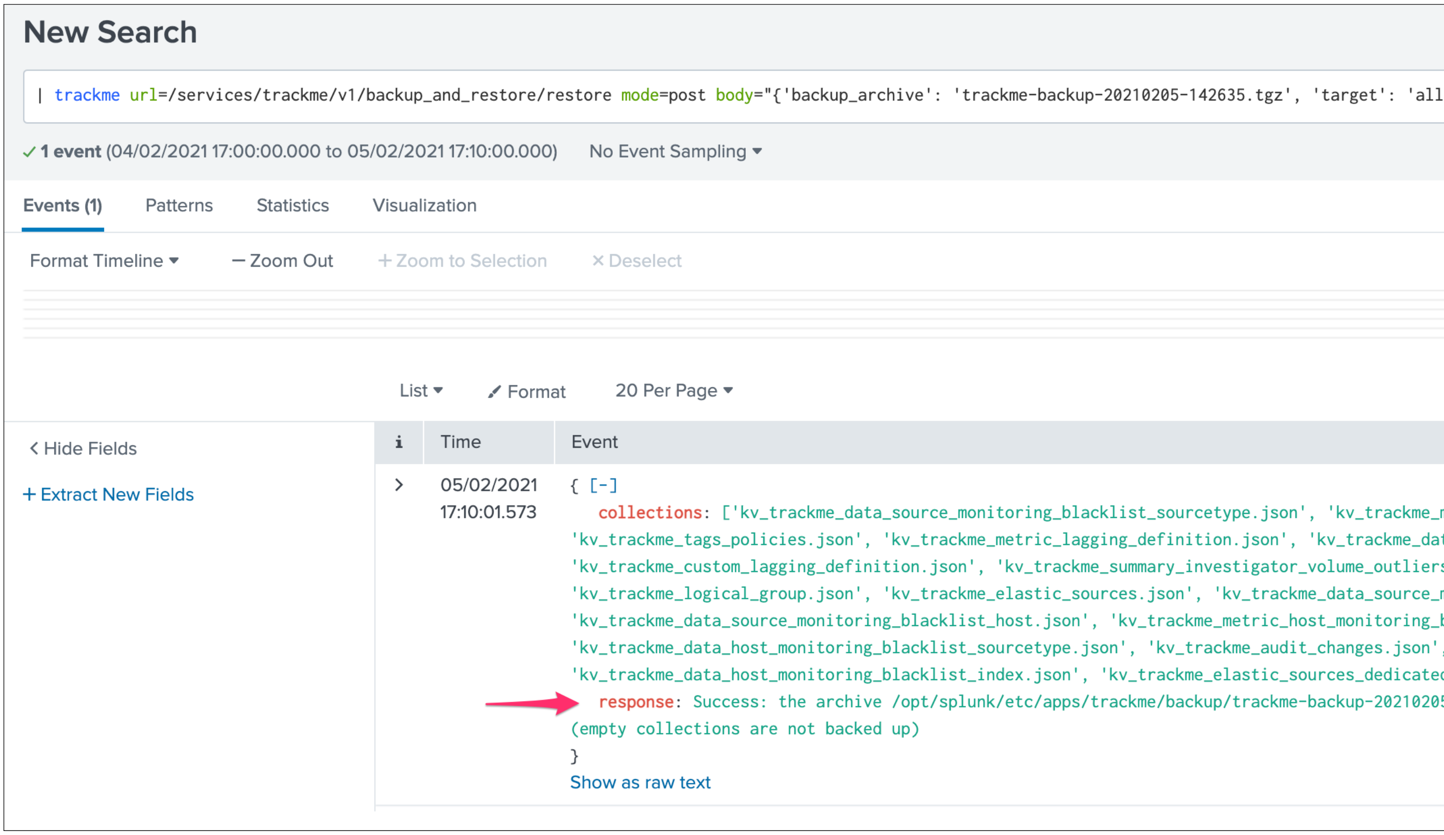
- `all` which is the default and means restoring all collections
- `<name of the JSON file corresponding to the KVstore collection` to restore a specific KVstore collection only

Use the *dry_run mode* true to list the JSON file available in a given archive file.

Restoring everything

The following SPL command will first perform a dry run to verify the archive, without modifying anything:

```
| trackme url=/services/trackme/v1/backup_and_restore/restore mode=post body="{
↪'backup_archive': 'trackme-backup-20210205-142635.tgz', 'target': 'all', 'dry_run':
↪'true'}"
```



```
| trackme url=/services/trackme/v1/backup_and_restore/restore mode=post body="{'backup_archive': 'trackme-backup-20210205-142635.tgz', 'target': 'all
```

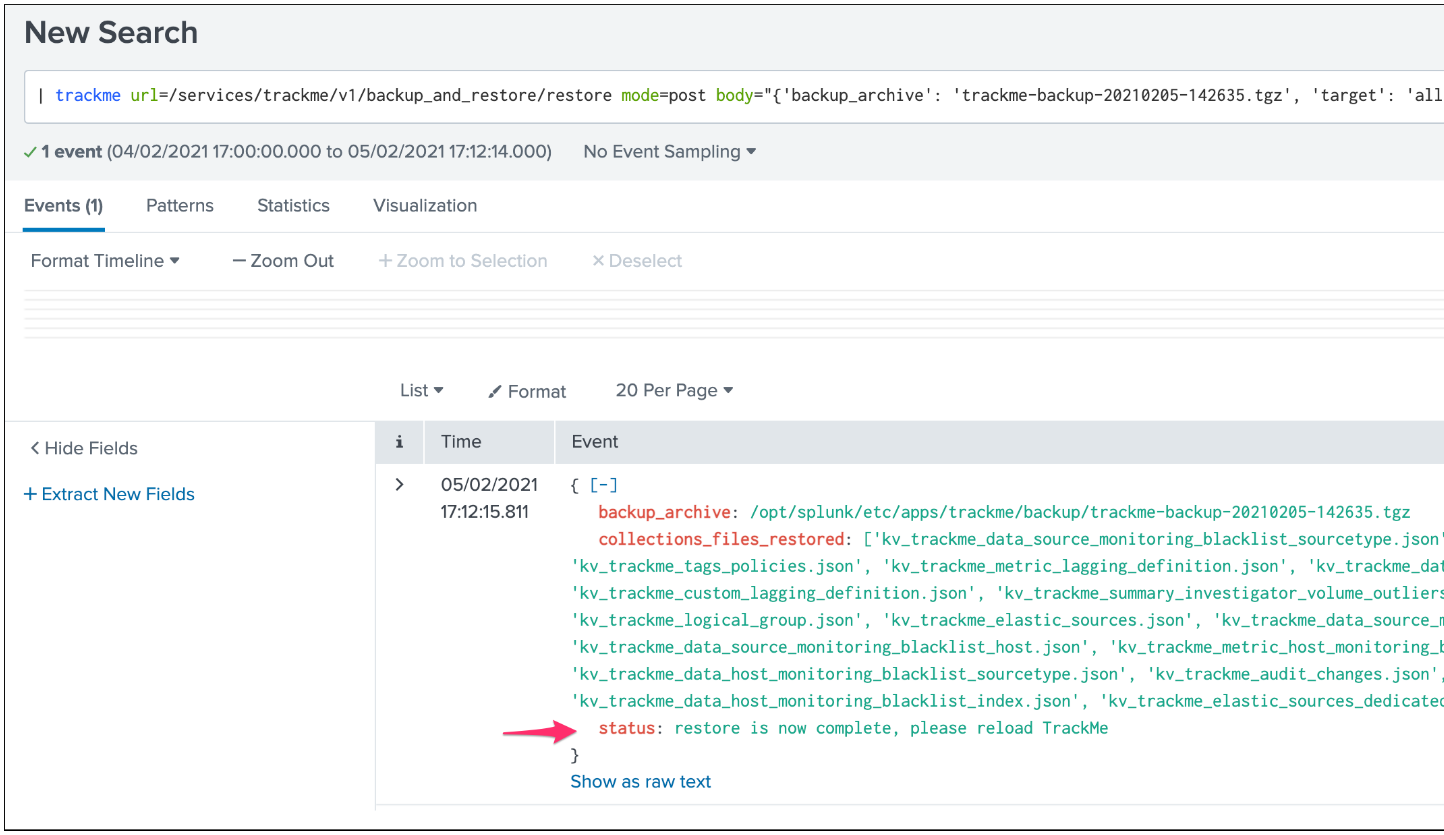
✓ 1 event (04/02/2021 17:00:00.000 to 05/02/2021 17:10:00.000)   No Event Sampling ▾

05/02/2021 17:10:01.573

```
{ [-]
   collections: ['kv_trackme_data_source_monitoring_blacklist_sourcetype.json', 'kv_trackme_r
'kv_trackme_tags_policies.json', 'kv_trackme_metric_lagging_definition.json', 'kv_trackme_dat
'kv_trackme_custom_lagging_definition.json', 'kv_trackme_summary_investigator_volume_outliers
'kv_trackme_logical_group.json', 'kv_trackme_elastic_sources.json', 'kv_trackme_data_source_r
'kv_trackme_data_source_monitoring_blacklist_host.json', 'kv_trackme_metric_host_monitoring_b
'kv_trackme_data_host_monitoring_blacklist_sourcetype.json', 'kv_trackme_audit_changes.json',
'kv_trackme_data_host_monitoring_blacklist_index.json', 'kv_trackme_elastic_sources_dedicated
   response: Success: the archive /opt/splunk/etc/apps/trackme/backup/trackme-backup-2021020
(empty collections are not backed up)
}
```

The following SPL command will restore all KVstore collections to a given state according to the content of that backup:

```
| trackme url=/services/trackme/v1/backup_and_restore/restore mode=post body="{
↪'backup_archive': 'trackme-backup-20210205-142635.tgz', 'target': 'all', 'dry_run':
↪'false'}"
```

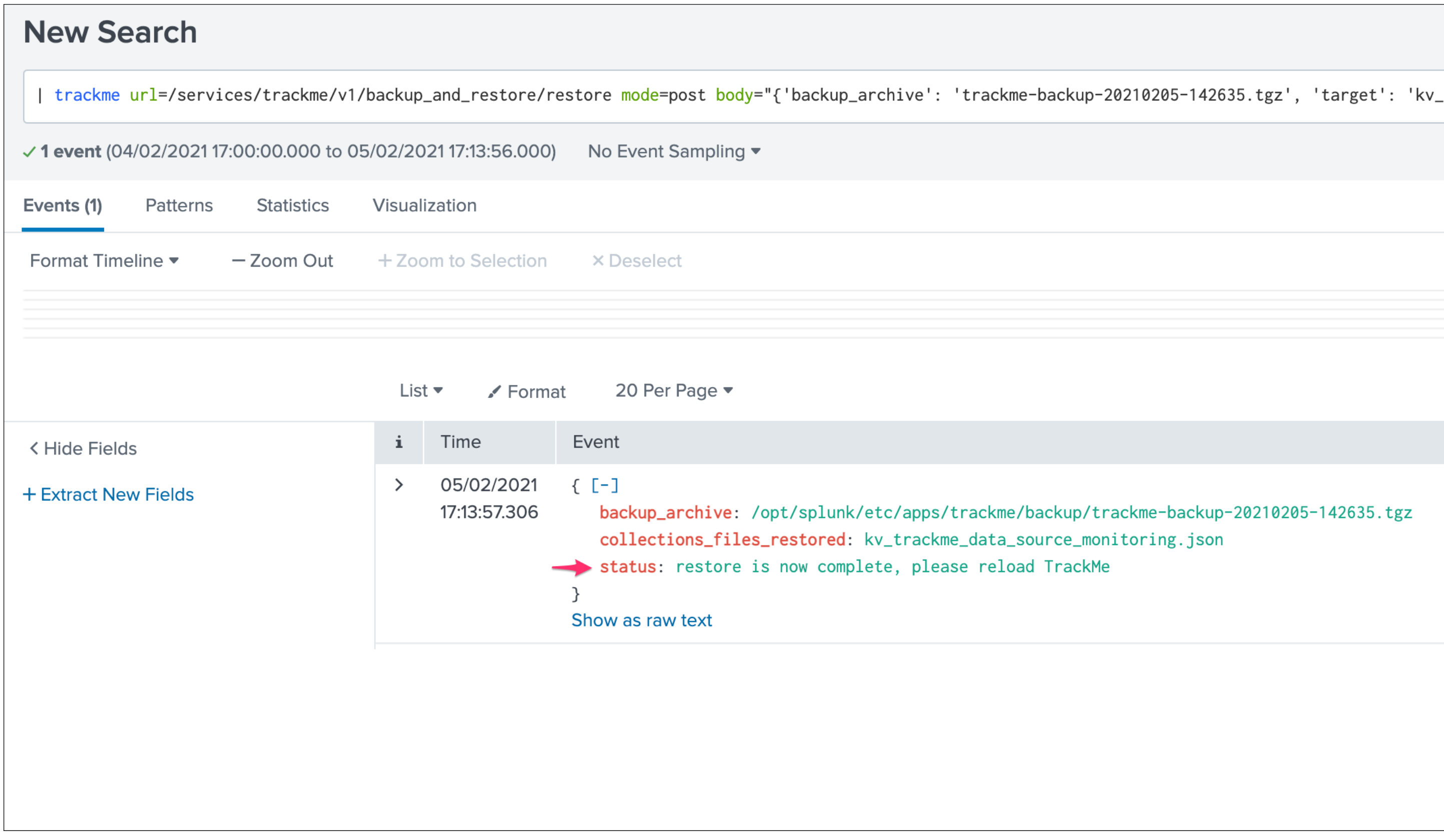


```
| trackme url=/services/trackme/v1/backup_and_restore/restore mode=post body="{'backup_archive': 'trackme-backup-20210205-142635.tgz', 'target': 'all
```

```
05/02/2021
17:12:15.811
{ [-]
   backup_archive: /opt/splunk/etc/apps/trackme/backup/trackme-backup-20210205-142635.tgz
   collections_files_restored: ['kv_trackme_data_source_monitoring_blacklist_sourcetype.json'
'kv_trackme_tags_policies.json', 'kv_trackme_metric_lagging_definition.json', 'kv_trackme_dat
'kv_trackme_custom_lagging_definition.json', 'kv_trackme_summary_investigator_volume_outliers
'kv_trackme_logical_group.json', 'kv_trackme_elastic_sources.json', 'kv_trackme_data_source_m
'kv_trackme_data_source_monitoring_blacklist_host.json', 'kv_trackme_metric_host_monitoring_b
'kv_trackme_data_host_monitoring_blacklist_sourcetype.json', 'kv_trackme_audit_changes.json',
'kv_trackme_data_host_monitoring_blacklist_index.json', 'kv_trackme_elastic_sources_dedicated
   status: restore is now complete, please reload TrackMe
}
```

The following SPL command will restore a specific collection only:

```
| trackme url=/services/trackme/v1/backup_and_restore/restore mode=post body="{
↪'backup_archive': 'trackme-backup-20210205-142635.tgz', 'target': 'kv_trackme_data_
↪source_monitoring.json', 'dry_run': 'false'}"
```



```
| trackme url=/services/trackme/v1/backup_and_restore/restore mode=post body="{'backup_archive': 'trackme-backup-20210205-142635.tgz', 'target': 'kv_
```

✓ 1 event (04/02/2021 17:00:00.000 to 05/02/2021 17:13:56.000) No Event Sampling ▾

05/02/2021 17:13:57.306

```
{ [-]
   backup_archive: /opt/splunk/etc/apps/trackme/backup/trackme-backup-20210205-142635.tgz
   collections_files_restored: kv_trackme_data_source_monitoring.json
   status: restore is now complete, please reload TrackMe
}
```

Once the restore operation is finished, please reload the application, restarting the Splunk Search head(s) is not required.

3.2 Splunk ITSI and TrackMe integration

If you are an ITSI customer, you can integrate trackMe concepts in ITSI and use the product capabilities to extend your monitoring.

Key information:

- TrackMe generates metrics in the metric index which can directly be turned into KPIs in a KPI base search
- Dimensions are indexed within the metrics which describe the entity and provide the by statement keys required to compute the KPIs
- Additionally, if TrackMe is running on the same search header layer than ITSI, search time lookups can be used to enrich the entities
- Every time a tracker runs, TrackMe records the statuses as summary events in the TrackMe summary index, which are used by ITSI to track and detect changes in the entity statuses

3.2.1 Step 1: entity search and creation

Create entity search(es) to generate entities in ITSI:

- If ITSI and TrackMe are running on the same search head layer, you can enrich the entities results with information from TrackMe such as the priority or the monitored stated

- You can include any kind of filter required, for example you might want to monitoring within ITSI only data sources, that are monitored and a medium or high priority
- Create a single entity search per TrackMe category to avoid issues such as duplicated ITSI entities
- You do not need a long time range period, the last 15 minutes is enough since all TrackMe entities are available in the summary data for each execution of the trackers

ITSI entities generation search definition

Data source entity gen search

```
index=trackme_summary source=current_state_tracking:data_source (data_monitored_state=
↪"enabled")
| eval trackme_monitored_state=coalesce(data_monitored_state, metric_monitored_state),
↪ object=lower(object)
| stats latest(priority) as trackme_priority, latest(trackme_monitored_state) as␣
↪trackme_monitored_state by object_category, object
| fields object_category, object, trackme_monitored_state, trackme_priority
| rename object as trackme_object, object_category as trackme_object_category
| eval itsi_role = "trackme", itsi_entity = trackme_object, itsi_entity_type =
↪"trackme:" . trackme_object_category
| fields itsi_entity, itsi_role, itsi_entity_type, trackme_*
```

Data host entity gen search

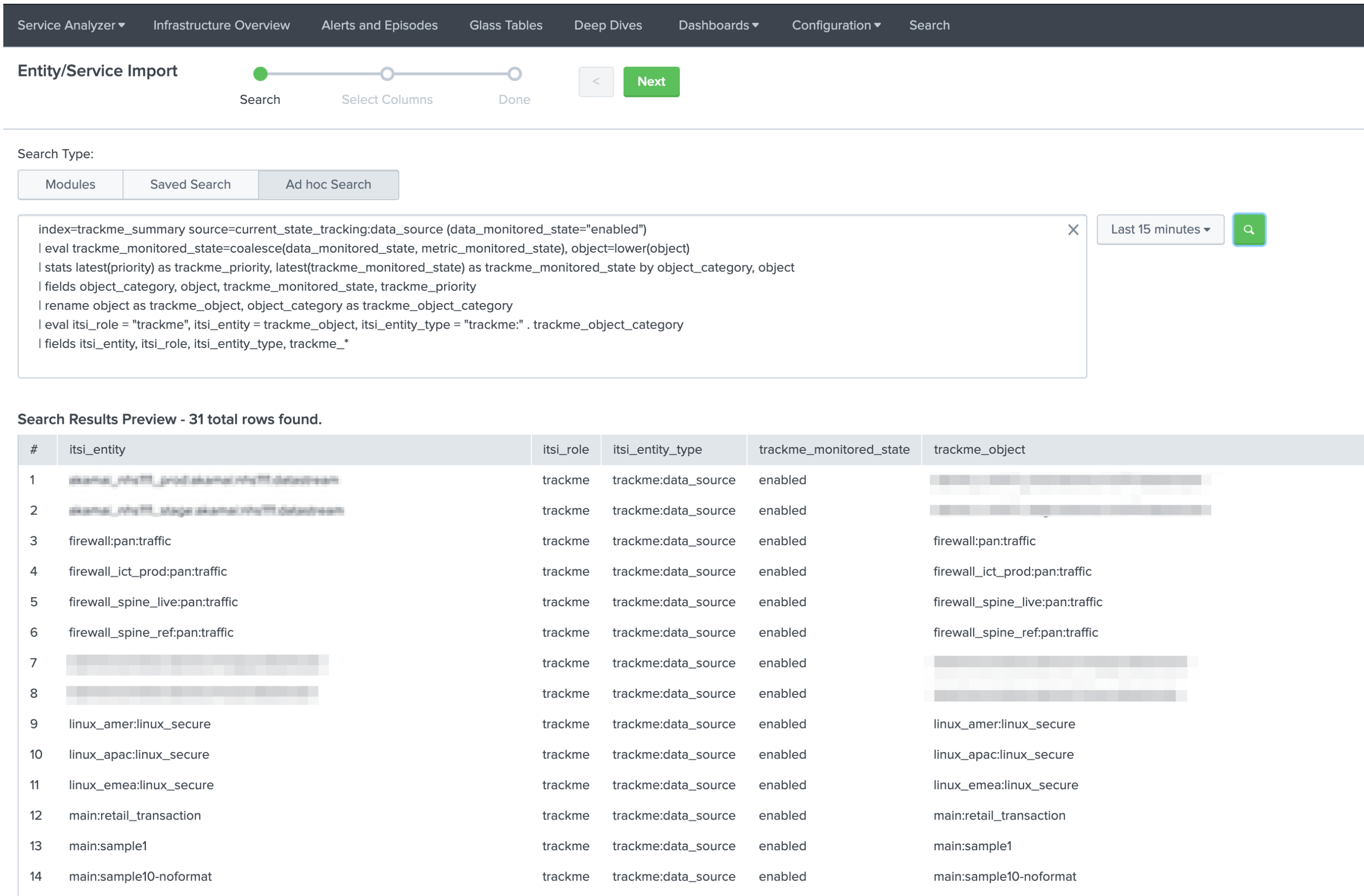
```
index=trackme_summary source=current_state_tracking:data_host (data_monitored_state=
↪"enabled")
| eval trackme_monitored_state=coalesce(data_monitored_state, metric_monitored_state),
↪ object=lower(object)
| stats latest(priority) as trackme_priority, latest(trackme_monitored_state) as␣
↪trackme_monitored_state by object_category, object
| fields object_category, object, trackme_monitored_state, trackme_priority
| rename object as trackme_object, object_category as trackme_object_category
| eval itsi_role = "trackme", itsi_entity = trackme_object, itsi_entity_type =
↪"trackme:" . trackme_object_category
| fields itsi_entity, itsi_role, itsi_entity_type, trackme_*
```

Metric host entity gen search if on the same search head(s)

```
index=trackme_summary source=current_state_tracking:metric_host (metric_monitored_
↪state="enabled")
| eval trackme_monitored_state=coalesce(data_monitored_state, metric_monitored_state),
↪ object=lower(object)
| stats latest(priority) as trackme_priority, latest(trackme_monitored_state) as␣
↪trackme_monitored_state by object_category, object
| fields object_category, object, trackme_monitored_state, trackme_priority
| rename object as trackme_object, object_category as trackme_object_category
| eval itsi_role = "trackme", itsi_entity = trackme_object, itsi_entity_type =
↪"trackme:" . trackme_object_category
| fields itsi_entity, itsi_role, itsi_entity_type, trackme_*
```

ITSI entities import

Go in ITSI / Configuration / Entities then click on the button Create Entity / Inport from Search



Search Type:

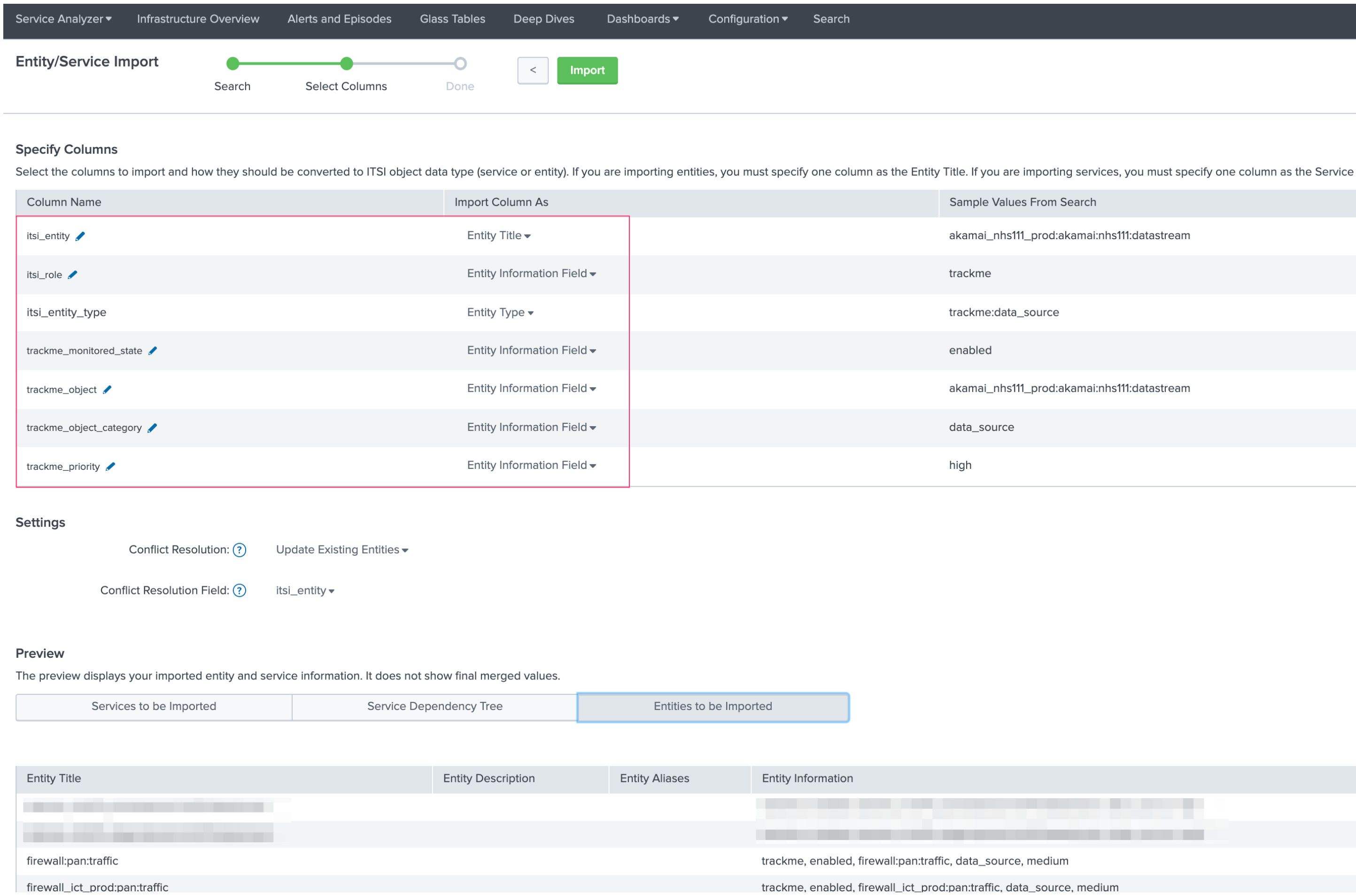
```
index=trackme_summary source=current_state_tracking:data_source (data_monitored_state="enabled")
| eval trackme_monitored_state=coalesce(data_monitored_state, metric_monitored_state), object=lower(object)
| stats latest(priority) as trackme_priority, latest(trackme_monitored_state) as trackme_monitored_state by object_category, object
| fields object_category, object, trackme_monitored_state, trackme_priority
| rename object as trackme_object, object_category as trackme_object_category
| eval itsi_role = "trackme", itsi_entity = trackme_object, itsi_entity_type = "trackme:" . trackme_object_category
| fields itsi_entity, itsi_role, itsi_entity_type, trackme_*
```

Search Results Preview - 31 total rows found.

#	itsi_entity	itsi_role	itsi_entity_type	trackme_monitored_state	trackme_object
1	[redacted]	trackme	trackme:data_source	enabled	[redacted]
2	[redacted]	trackme	trackme:data_source	enabled	[redacted]
3	firewall:pan:traffic	trackme	trackme:data_source	enabled	firewall:pan:traffic
4	firewall_ict_prod:pan:traffic	trackme	trackme:data_source	enabled	firewall_ict_prod:pan:traffic
5	firewall_spine_live:pan:traffic	trackme	trackme:data_source	enabled	firewall_spine_live:pan:traffic
6	firewall_spine_ref:pan:traffic	trackme	trackme:data_source	enabled	firewall_spine_ref:pan:traffic
7	[redacted]	trackme	trackme:data_source	enabled	[redacted]
8	[redacted]	trackme	trackme:data_source	enabled	[redacted]
9	linux_amer:linux_secure	trackme	trackme:data_source	enabled	linux_amer:linux_secure
10	linux_apac:linux_secure	trackme	trackme:data_source	enabled	linux_apac:linux_secure
11	linux_emea:linux_secure	trackme	trackme:data_source	enabled	linux_emea:linux_secure
12	main:retail_transaction	trackme	trackme:data_source	enabled	main:retail_transaction
13	main:sample1	trackme	trackme:data_source	enabled	main:sample1
14	main:sample10-noformat	trackme	trackme:data_source	enabled	main:sample10-noformat

Click next and define the entities fields import:

- **itsi_entity:** import as Entity Title
- **itsi_role:** import as Entity information field
- **itsi_entity_type:** import as Entity Type
- **trackme_monitored_state:** import as Entity information field
- **trackme_object:** import as Entity Alias
- **trackme_object_category:** import as Entity information field
- **trackme_priority:** import as Entity information field



Entity/Service Import

Specify Columns

Select the columns to import and how they should be converted to ITSI object data type (service or entity). If you are importing entities, you must specify one column as the Entity Title. If you are importing services, you must specify one column as the Service

Column Name	Import Column As	Sample Values From Search
itsi_entity	Entity Title ▾	akamai_nhs111_prod:akamai:nhs111:datastream
itsi_role	Entity Information Field ▾	trackme
itsi_entity_type	Entity Type ▾	trackme:data_source
trackme_monitored_state	Entity Information Field ▾	enabled
trackme_object	Entity Information Field ▾	akamai_nhs111_prod:akamai:nhs111:datastream
trackme_object_category	Entity Information Field ▾	data_source
trackme_priority	Entity Information Field ▾	high

Settings

Conflict Resolution: ? Update Existing Entities ▾

Conflict Resolution Field: ? itsi_entity ▾

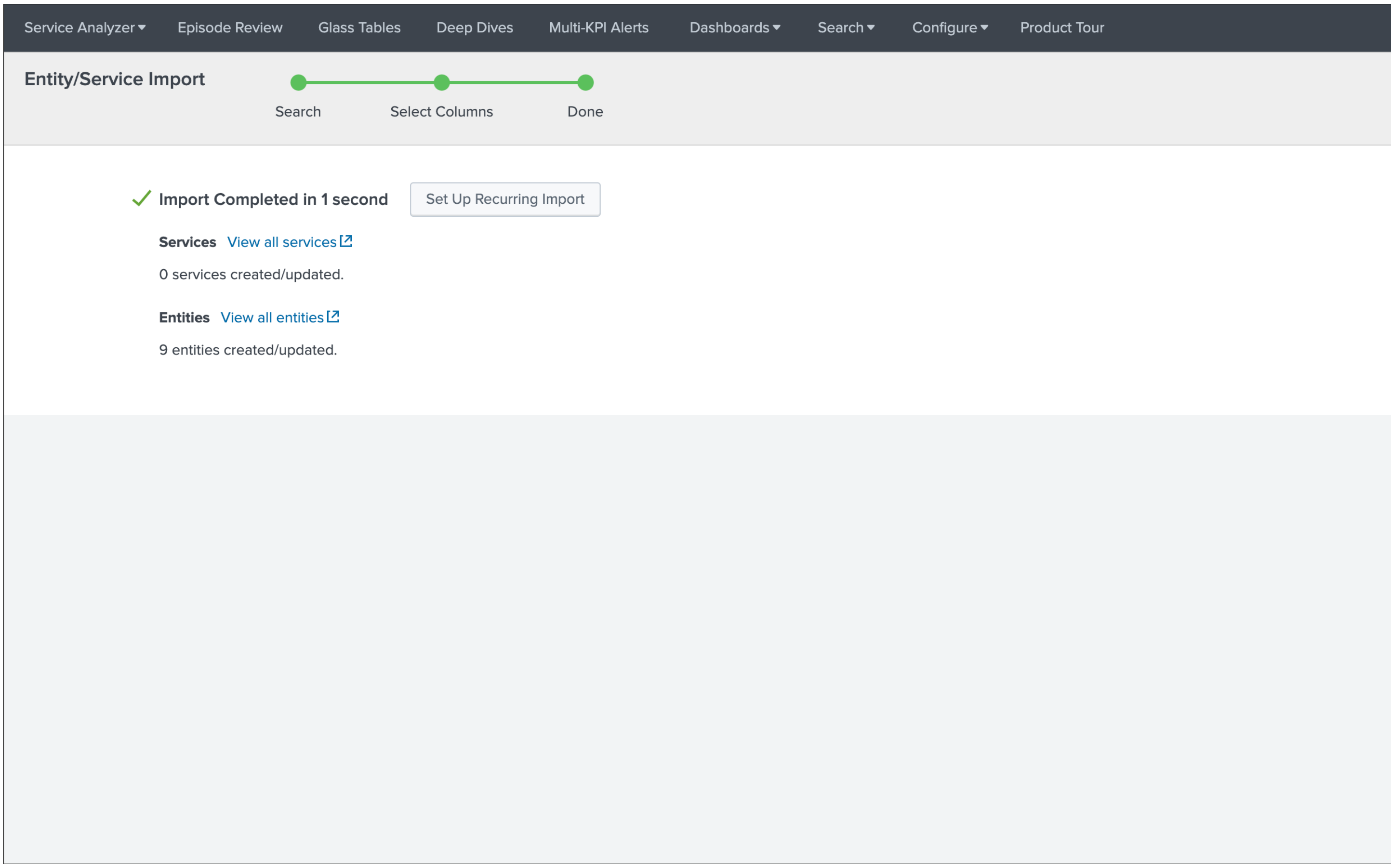
Preview

The preview displays your imported entity and service information. It does not show final merged values.

Services to be Imported | Service Dependency Tree | Entities to be Imported

Entity Title	Entity Description	Entity Aliases	Entity Information
[redacted]			[redacted]
[redacted]			[redacted]
firewall:pan:traffic			trackme, enabled, firewall:pan:traffic, data_source, medium
firewall_ict_prod:pan:traffic			trackme, enabled, firewall_ict_prod:pan:traffic, data_source, medium

Click next to generate the entities, and setup a recurrent import job:



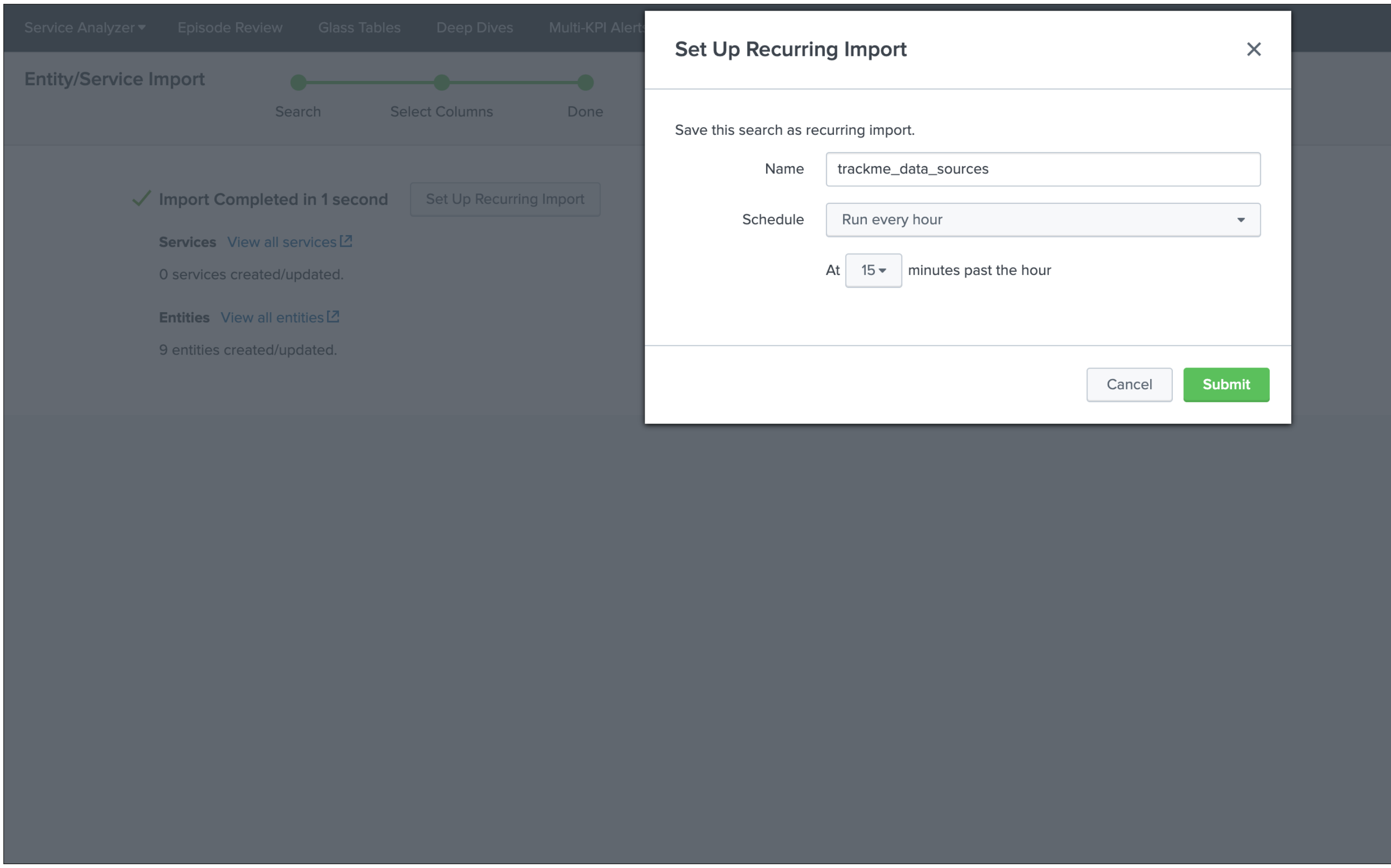
✓ Import Completed in 1 second

Services

0 services created/updated.

Entities

9 entities created/updated.



Entity/Service Import

✓ Import Completed in 1 second

Services

0 services created/updated.

Entities

9 entities created/updated.

Set Up Recurring Import

Save this search as recurring import.

Name: trackme_data_sources

Schedule: Run every hour

At 15 minutes past the hour

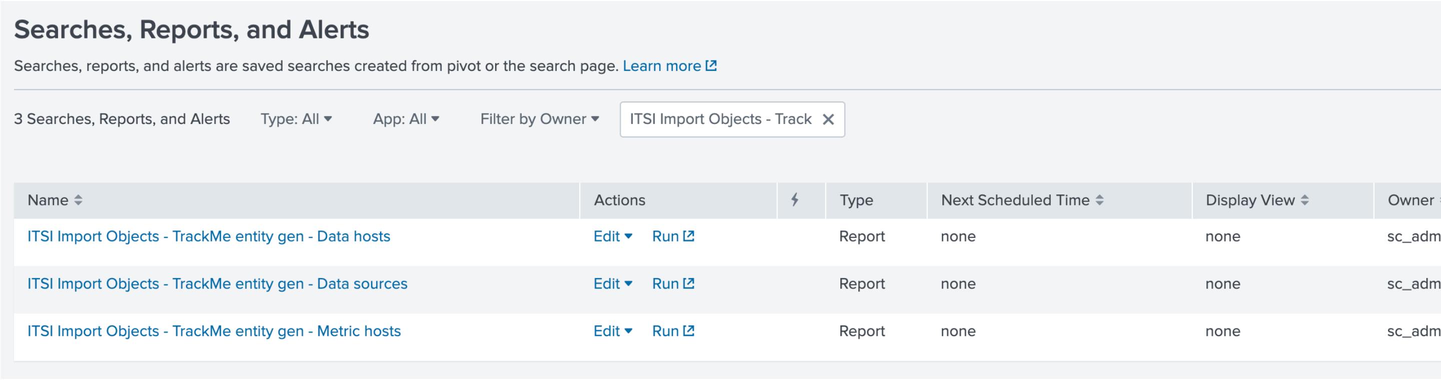
Any new data source discovered and configured in TrackMe will be created in ITSI, and existing entities will be maintained automatically.

Set up a recurring import, for reference:

https://docs.splunk.com/Documentation/ITSI/latest/Entity/ImportRecurring

Once you have setup the recurring import, you can access to the savedsearches:

ITSI Import Objects - TrackMe:



Searches, Reports, and Alerts

Searches, reports, and alerts are saved searches created from pivot or the search page.

3 Searches, Reports, and Alerts

Name	Actions		Type	Next Scheduled Time	Display View	Owner
ITSI Import Objects - TrackMe entity gen - Data hosts	Edit ▾ Run		Report	none	none	sc_adm
ITSI Import Objects - TrackMe entity gen - Data sources	Edit ▾ Run		Report	none	none	sc_adm
ITSI Import Objects - TrackMe entity gen - Metric hosts	Edit ▾ Run		Report	none	none	sc_adm

3.2.2 Step 2: create the KPI base search for metrics

The next step is to create a KPI base saarch that will turn the metrics into ITSI KPIs, within the KPI base search editor, create a new base search:

KPI base search title and description:

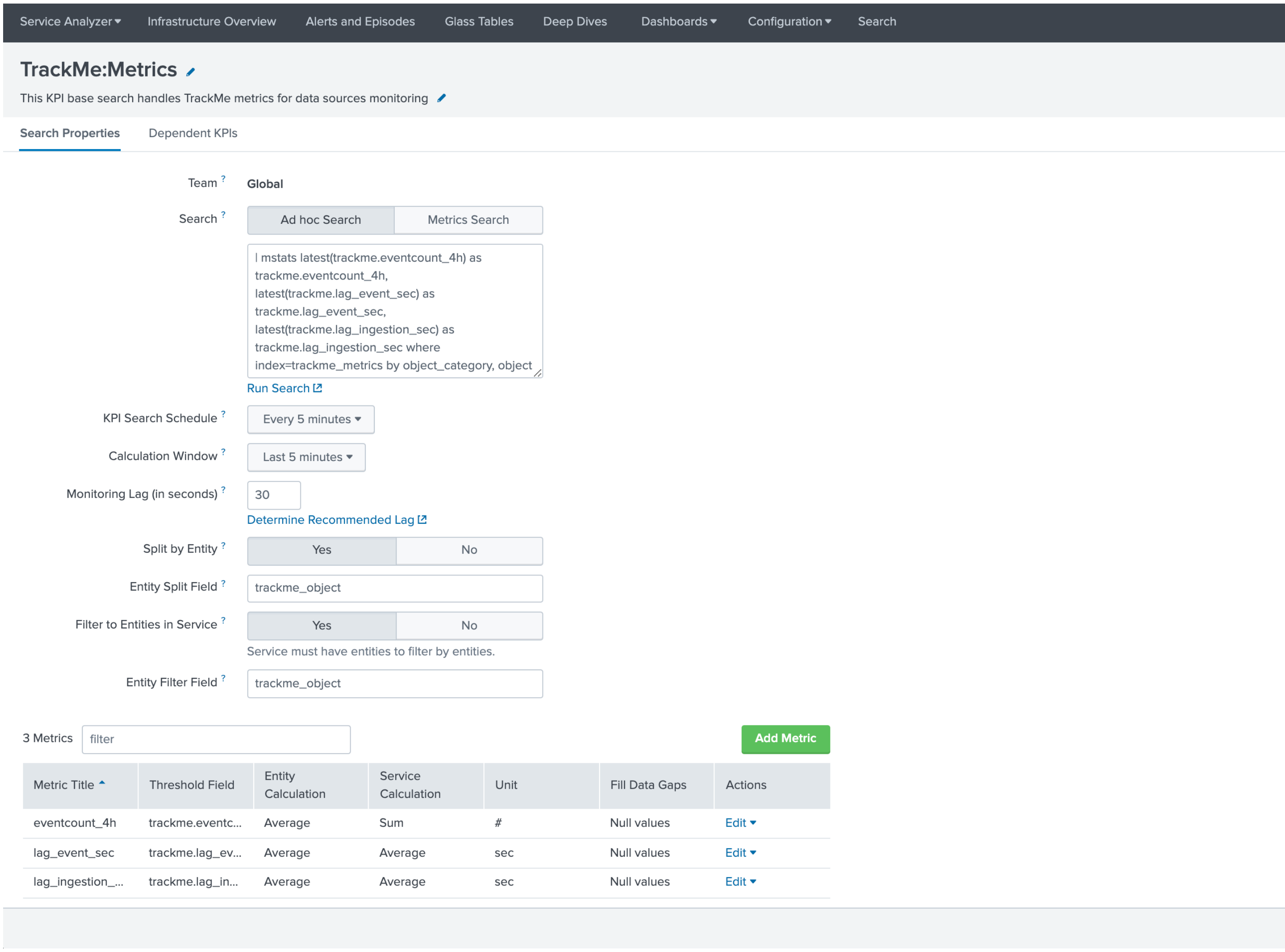
- Title: **TrackMe:Metrics**
- Description: **This KPI base search handles TrackMe metrics for data sources monitoring**

KPI base search:

- Type: **adhoc**
- search:

```
index=trackme_summary source="current_state_tracking:data_source"
| stats latest(data_eventcount) as trackme.eventcount_4h, latest(data_last_lag_seen)␣
↪as trackme.lag_event_sec, latest(data_last_ingestion_lag_seen) as trackme.lag_
↪ingestion_sec by _time, object_category, object
| rename object_category as trackme_object_category, object as trackme_object
```

- KPI Search Schedule: **Every 5 minutes**
- Calculation Window: **Last 5 minutes**
- Monitoring Lag (in seconds): **30**
- Split by Entity: **yes**
- Entity Split Field: **trackme_object**
- Filter to Entities in Service: **yes**
- Entity Filter Field: **trackme_object**



TrackMe:Metrics

This KPI base search handles TrackMe metrics for data sources monitoring

Team: Global

Search: Ad hoc Search | Metrics Search

```
| mstats latest(trackme.eventcount_4h) as
trackme.eventcount_4h,
latest(trackme.lag_event_sec) as
trackme.lag_event_sec,
latest(trackme.lag_ingestion_sec) as
trackme.lag_ingestion_sec where
index=trackme_metrics by object_category, object
```

KPI Search Schedule: Every 5 minutes

Calculation Window: Last 5 minutes

Monitoring Lag (in seconds): 30

Split by Entity: Yes | No

Entity Split Field: trackme_object

Filter to Entities in Service: Yes | No

Service must have entities to filter by entities.

Entity Filter Field: trackme_object

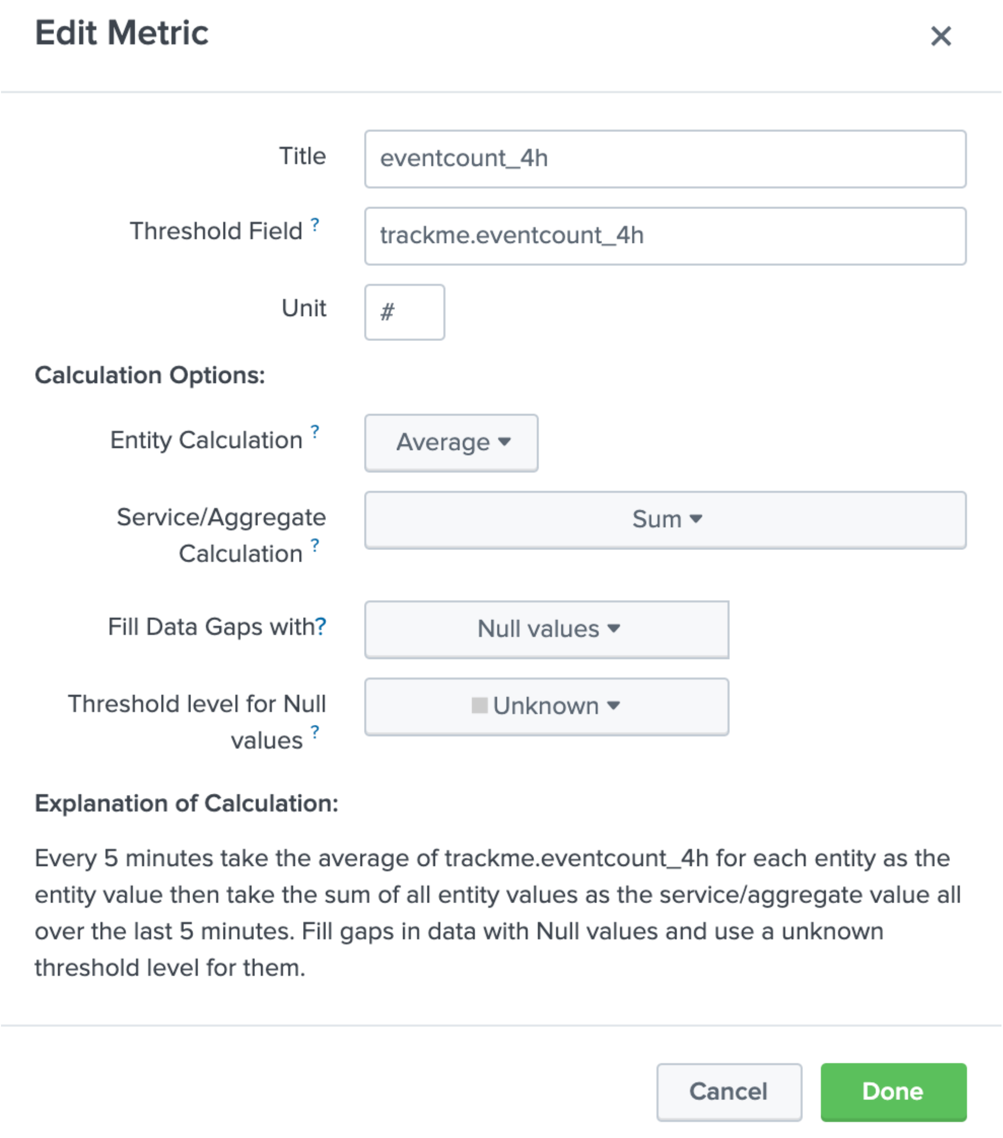
3 Metrics

Metric Title ▴	Threshold Field	Entity Calculation	Service Calculation	Unit	Fill Data Gaps	Actions
eventcount_4h	trackme.eventc...	Average	Sum	#	Null values	Edit ▾
lag_event_sec	trackme.lag_ev...	Average	Average	sec	Null values	Edit ▾
lag_ingestion_...	trackme.lag_in...	Average	Average	sec	Null values	Edit ▾

Then, create the metrics as follows:

Metric: trackme.eventcount_4h

- Title: **eventcount_4h**
- threshold field: **trackme.eventcount_4h**
- Unit: **#**
- Entity calculation: **Average**
- Service / Aggregate calculation: **Sum**
- Fill Data Gaps with: **Null values**
- Threshold level for Null values: **Unknown**



Edit Metric

Title: eventcount_4h

Threshold Field ?: trackme.eventcount_4h

Unit: #

Calculation Options:

Entity Calculation ?: Average

Service/Aggregate Calculation ?: Sum

Fill Data Gaps with?: Null values

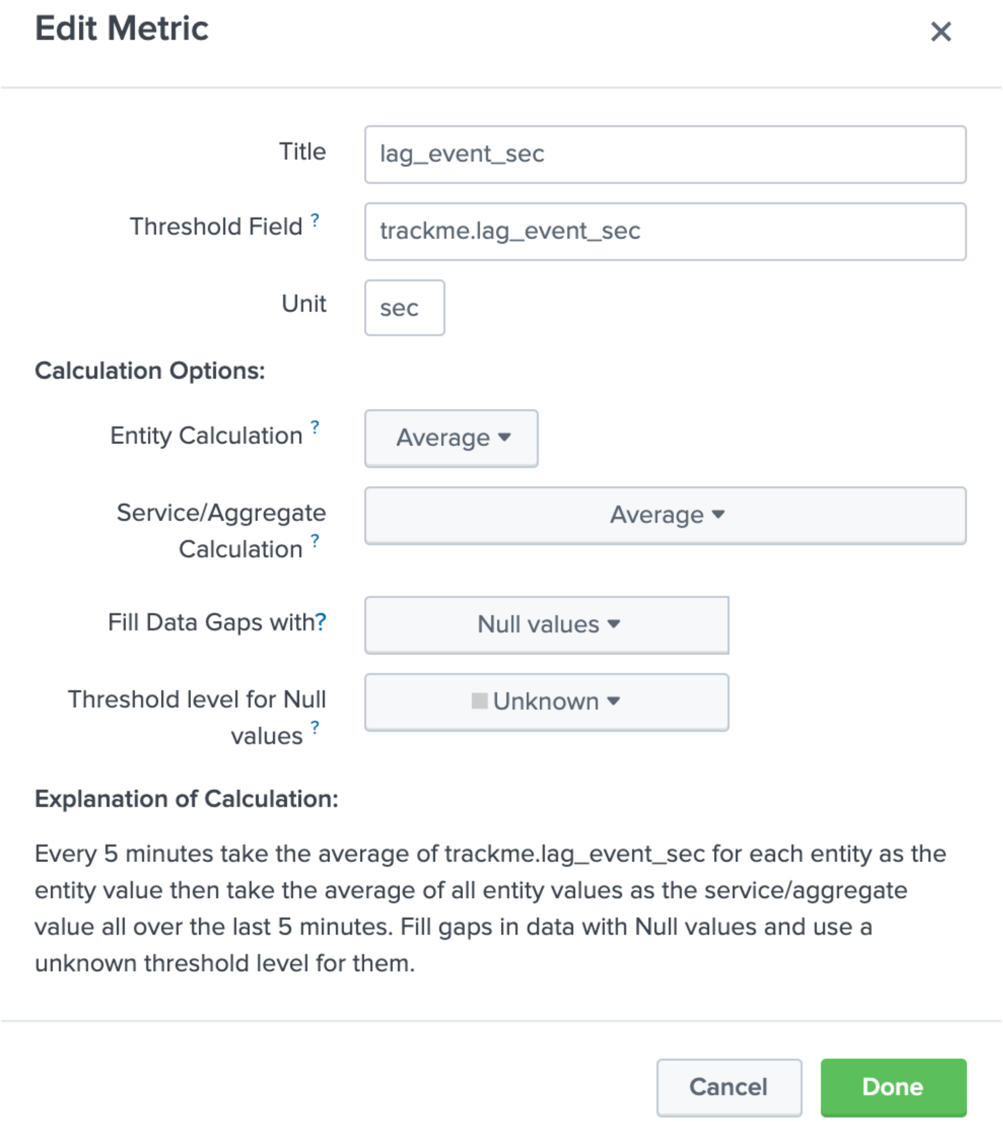
Threshold level for Null values ?: Unknown

Explanation of Calculation:

Every 5 minutes take the average of trackme.eventcount_4h for each entity as the entity value then take the sum of all entity values as the service/aggregate value all over the last 5 minutes. Fill gaps in data with Null values and use a unknown threshold level for them.

Metric: trackme.lag_event_sec

- Title: **lag_event_sec**
- threshold field: **trackme.lag_event_sec**
- Unit: **sec**
- Entity calculation: **Average**
- Service / Aggregate calculation: **Average**
- Fill Data Gaps with: **Null values**
- Threshold level for Null values: **Unknown**



Edit Metric

Title: lag_event_sec

Threshold Field ?: trackme.lag_event_sec

Unit: sec

Calculation Options:

Entity Calculation ?: Average

Service/Aggregate Calculation ?: Average

Fill Data Gaps with?: Null values

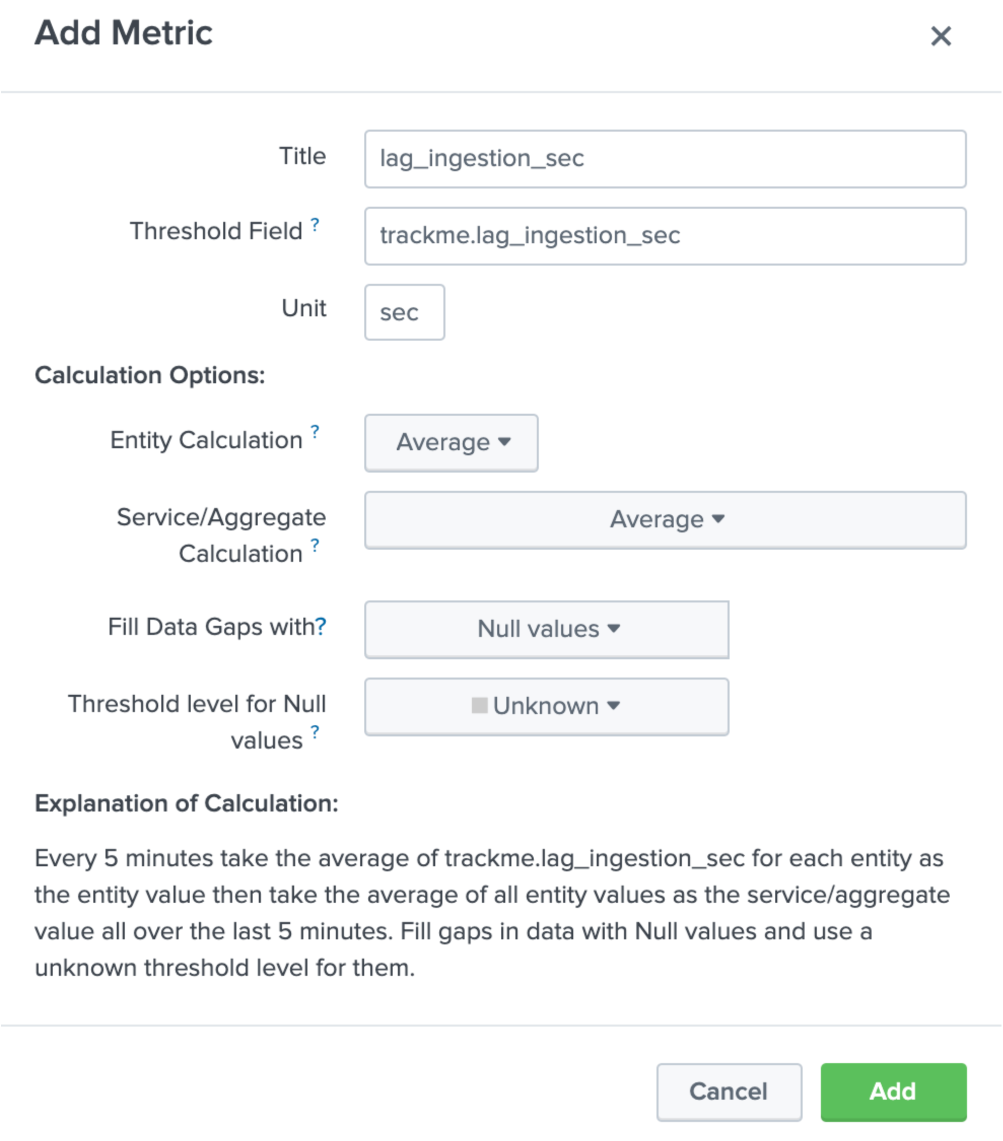
Threshold level for Null values ?: Unknown

Explanation of Calculation:

Every 5 minutes take the average of trackme.lag_event_sec for each entity as the entity value then take the average of all entity values as the service/aggregate value all over the last 5 minutes. Fill gaps in data with Null values and use a unknown threshold level for them.

Metric: trackme.lag_ingestion_sec

- Title: **lag_ingestion_sec**
- threshold field: **trackme.lag_ingestion_sec**
- Unit: **sec**
- Entity calculation: **Average**
- Service / Aggregate calculation: **Average**
- Fill Data Gaps with: **Null values**
- Threshold level for Null values: **Unknown**



Title: lag_ingestion_sec

Threshold Field ?: trackme.lag_ingestion_sec

Unit: sec

Calculation Options:

Entity Calculation ?: Average

Service/Aggregate Calculation ?: Average

Fill Data Gaps with?: Null values

Threshold level for Null values ?: Unknown

Explanation of Calculation:

Every 5 minutes take the average of trackme.lag_ingestion_sec for each entity as the entity value then take the average of all entity values as the service/aggregate value all over the last 5 minutes. Fill gaps in data with Null values and use a unknown threshold level for them.

3.2.3 Step 3: create the KPI base searches for summary statuses events

KPI base search title and description:

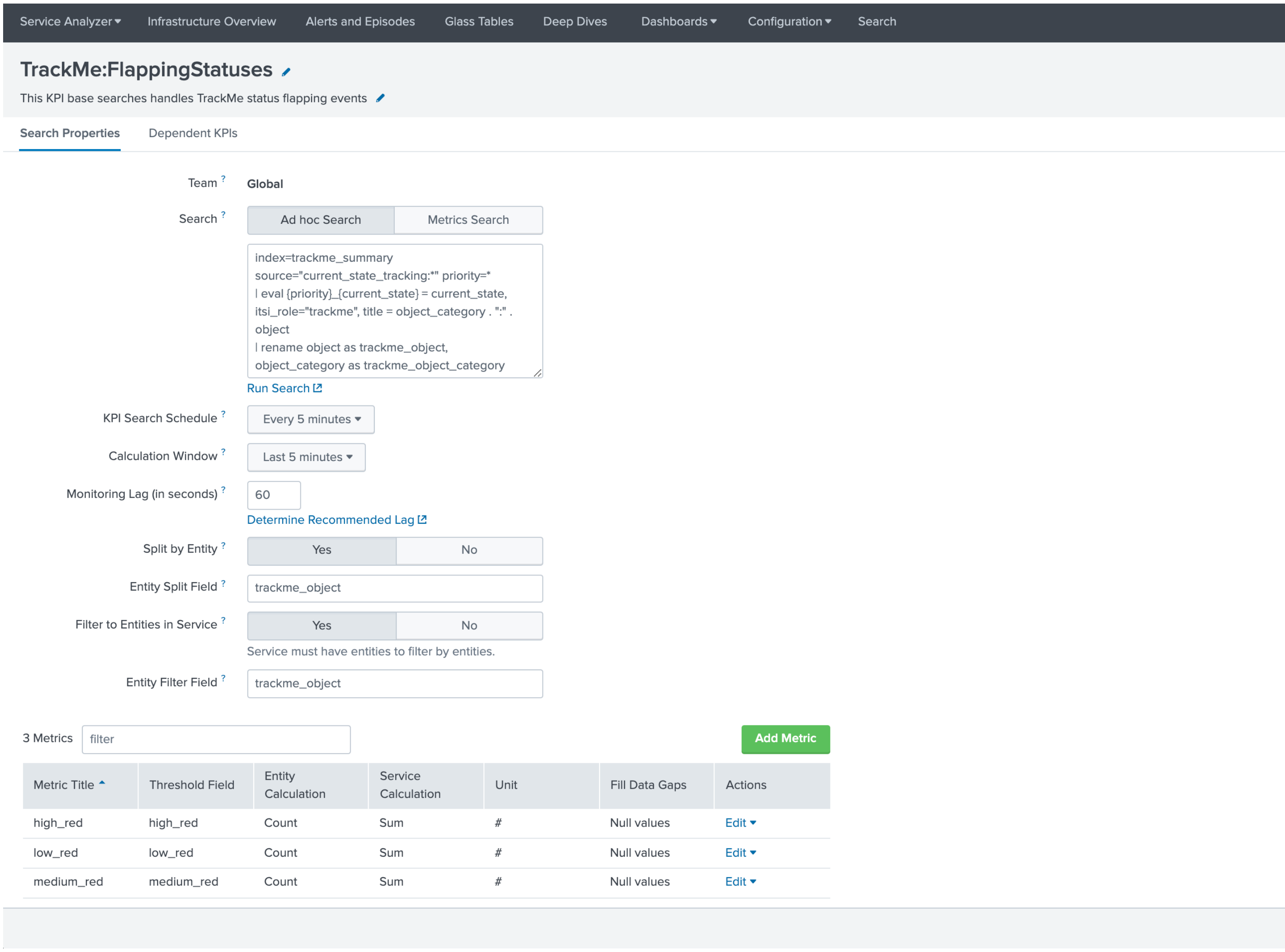
- Title: **TrackMe:FlappingStatuses**
- Description: **This KPI base searches handles TrackMe status flapping events**

KPI base search:

- Type: **adhoc**
- search:

```
`trackme_idx` source="current_state_tracking:*" priority=*
| eval {priority}_{current_state} = current_state
| rename object_category as trackme_object_category, object as trackme_object
```

- KPI Search Schedule: **Every 5 minutes**
- Calculation Window: **Last 5 minutes**
- Monitoring Lag (in seconds): **60**
- Split by Entity: **yes**
- Entity Split Field: **trackme_object**
- Filter to Entities in Service: **yes**
- Entity Filter Field: **trackme_object**



TrackMe:FlappingStatuses

This KPI base searches handles TrackMe status flapping events

Team: Global

Search: Ad hoc Search | Metrics Search

```
index=trackme_summary
source="current_state_tracking:*" priority=*
| eval {priority}_{current_state} = current_state,
itsi_role="trackme", title = object_category . ":" .
object
| rename object as trackme_object,
object_category as trackme_object_category
```

KPI Search Schedule: Every 5 minutes

Calculation Window: Last 5 minutes

Monitoring Lag (in seconds): 60

Split by Entity: Yes | No

Entity Split Field: trackme_object

Filter to Entities in Service: Yes | No

Service must have entities to filter by entities.

Entity Filter Field: trackme_object

3 Metrics

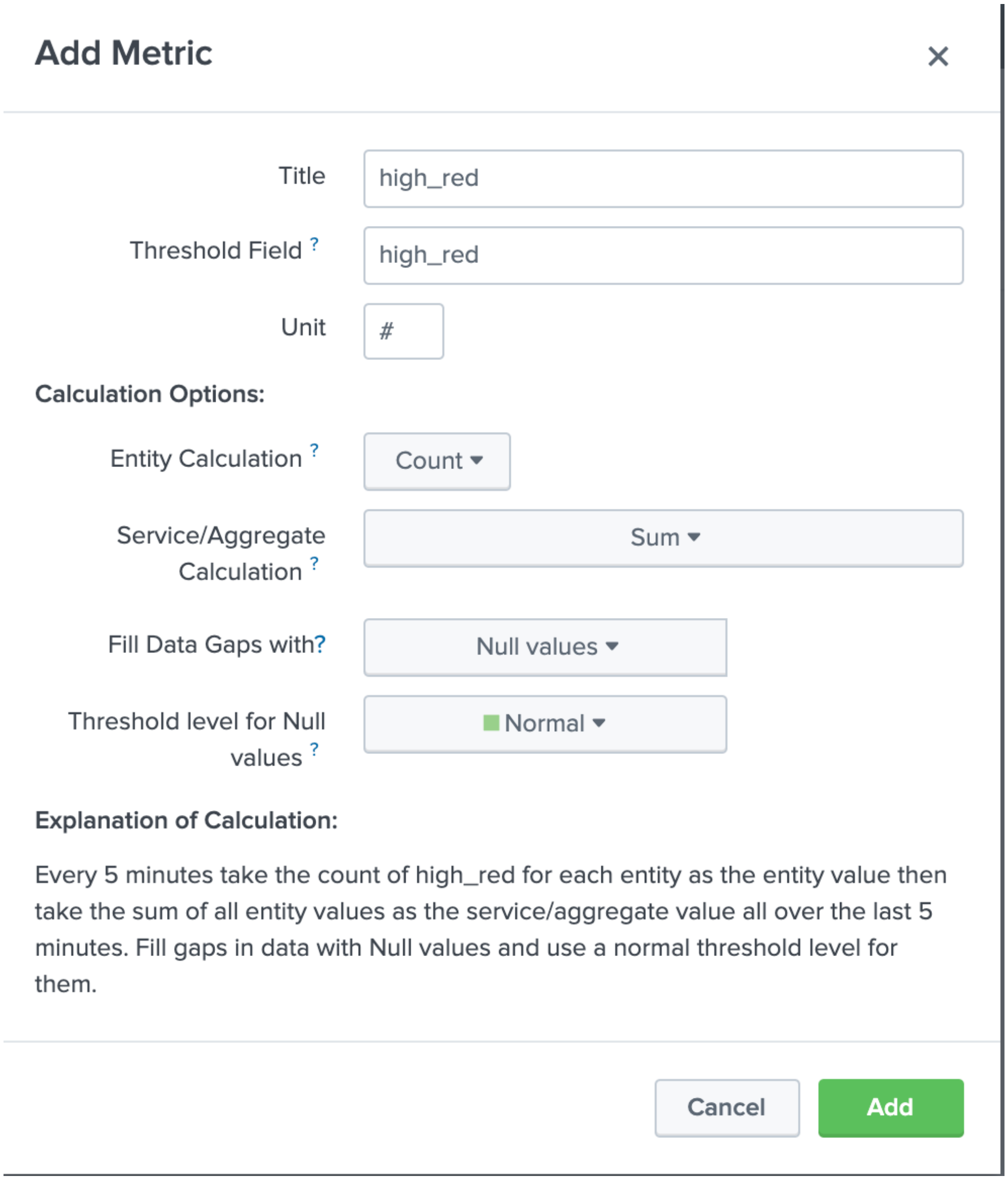
Metric Title ▴	Threshold Field	Entity Calculation	Service Calculation	Unit	Fill Data Gaps	Actions
high_red	high_red	Count	Sum	#	Null values	Edit ▾
low_red	low_red	Count	Sum	#	Null values	Edit ▾
medium_red	medium_red	Count	Sum	#	Null values	Edit ▾

Then, create the metrics as follows:

Metric: high_red

- Title: **high_red**
- threshold field: **high_red**
- Unit: **#**
- Entity calculation: **Count**

- Service / Aggregate calculation: **Sum**
- Fill Data Gaps with: **Null values**
- Threshold level for Null values: **Normal**



Add Metric

Title: high_red

Threshold Field ?: high_red

Unit: #

Calculation Options:

Entity Calculation ?: Count

Service/Aggregate Calculation ?: Sum

Fill Data Gaps with?: Null values

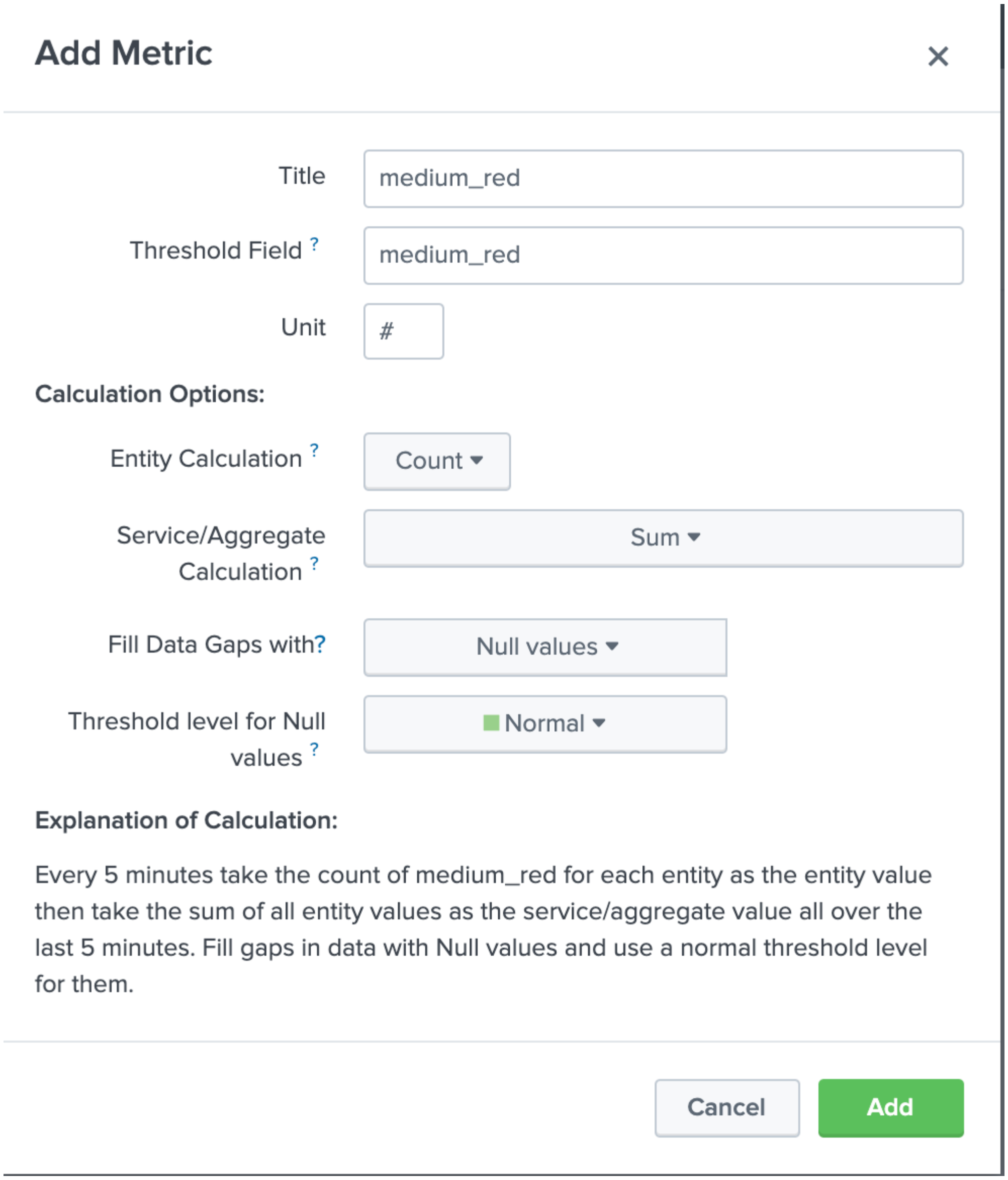
Threshold level for Null values ?: 🟩 Normal

Explanation of Calculation:

Every 5 minutes take the count of high_red for each entity as the entity value then take the sum of all entity values as the service/aggregate value all over the last 5 minutes. Fill gaps in data with Null values and use a normal threshold level for them.

Metric: medium_red

- Title: **medium_red**
- threshold field: **medium_red**
- Unit: **#**
- Entity calculation: **Count**
- Service / Aggregate calculation: **Sum**
- Fill Data Gaps with: **Null values**
- Threshold level for Null values: **Normal**



Add Metric

Title: medium_red

Threshold Field ?: medium_red

Unit: #

Calculation Options:

Entity Calculation ?: Count ▾

Service/Aggregate Calculation ?: Sum ▾

Fill Data Gaps with?: Null values ▾

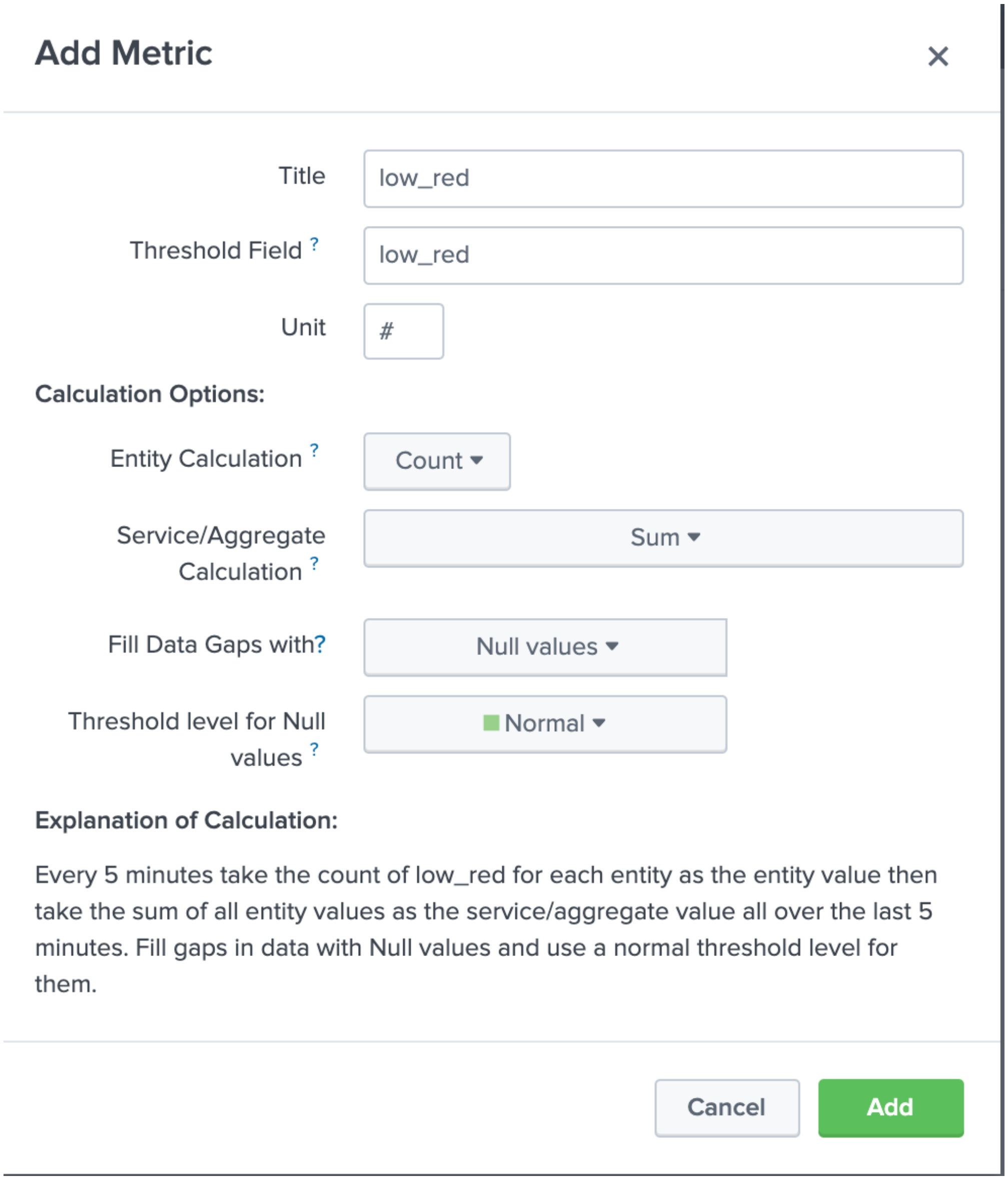
Threshold level for Null values ?: 🟩 Normal ▾

Explanation of Calculation:

Every 5 minutes take the count of medium_red for each entity as the entity value then take the sum of all entity values as the service/aggregate value all over the last 5 minutes. Fill gaps in data with Null values and use a normal threshold level for them.

Metric: low_red

- Title: **low_red**
- threshold field: **low_red**
- Unit: **#**
- Entity calculation: **Count**
- Service / Aggregate calculation: **Sum**
- Fill Data Gaps with: **Null values**
- Threshold level for Null values: **Normal**



Add Metric

Title: low_red

Threshold Field ?: low_red

Unit: #

Calculation Options:

Entity Calculation ?: Count

Service/Aggregate Calculation ?: Sum

Fill Data Gaps with?: Null values

Threshold level for Null values ?: 🟩 Normal

Explanation of Calculation:

Every 5 minutes take the count of low_red for each entity as the entity value then take the sum of all entity values as the service/aggregate value all over the last 5 minutes. Fill gaps in data with Null values and use a normal threshold level for them.

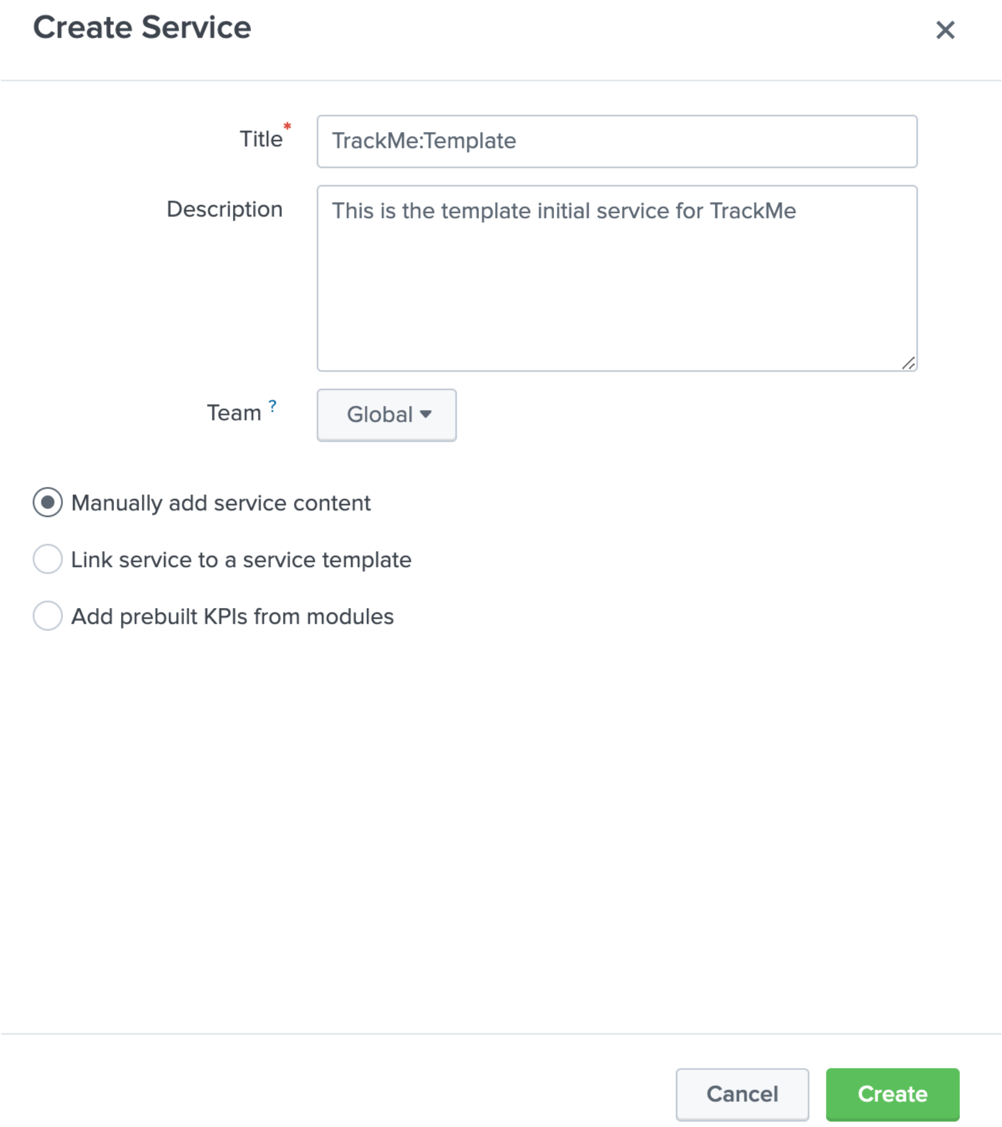
Notes:

- the technique `{priority}_{current_state} = current_state` allows you to track different levels of priorities easily without any conditional operations

3.2.4 Step 4: create a service that will be used for the service template definition

Go in Services / Create Service:

- Title: **TrackMe:Template**
- Description: **This is the template initial service for TrackMe**
- Manually add service content



Create Service

Title*: TrackMe:Template

Description: This is the template initial service for TrackMe

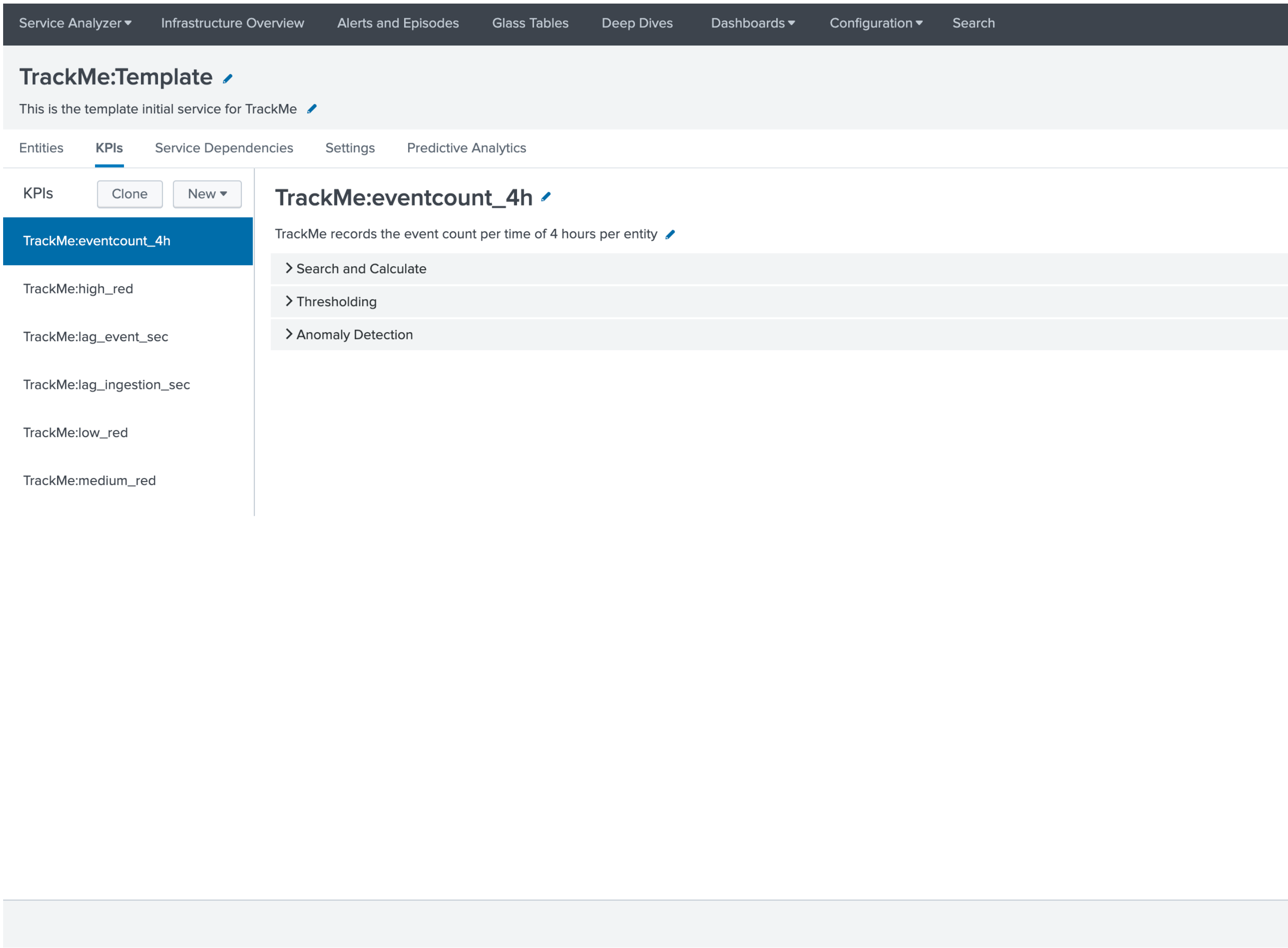
Team: Global

● Manually add service content

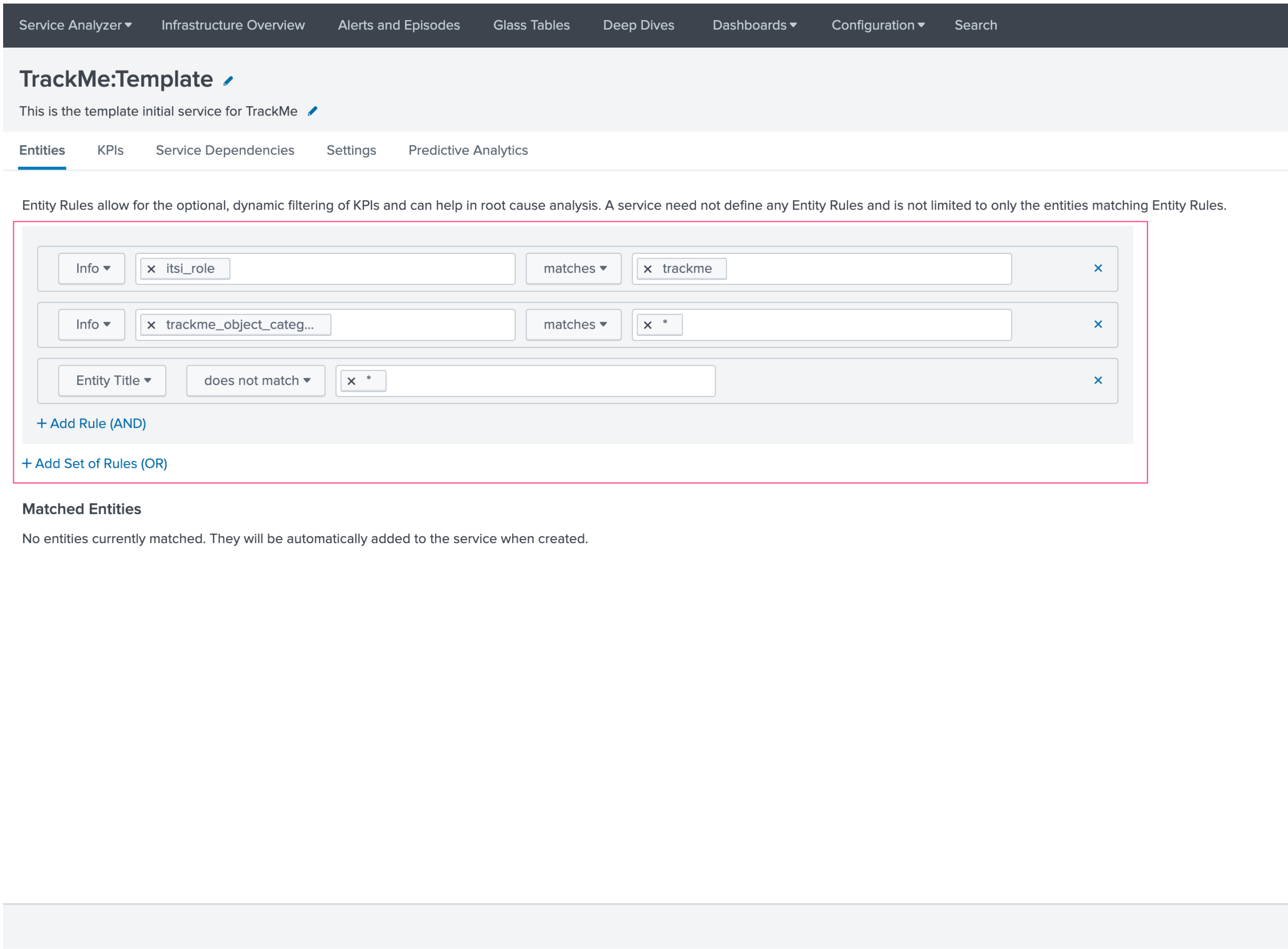
○ Link service to a service template

○ Add prebuilt KPIs from modules

Define the entities rules as follows:



- Info: itsi_role matches “trackme”
- Info: trackme_object_category matches “*”
- Entity Tile: does not match “*”



TrackMe:Template

This is the template initial service for TrackMe

Entity Rules allow for the optional, dynamic filtering of KPIs and can help in root cause analysis. A service need not define any Entity Rules and is not limited to only the entities matching Entity Rules.

Info ▾ | × itsi_role | matches ▾ | × trackme

Info ▾ | × trackme_object_categ... | matches ▾ | × *

Entity Title ▾ | does not match ▾ | × *

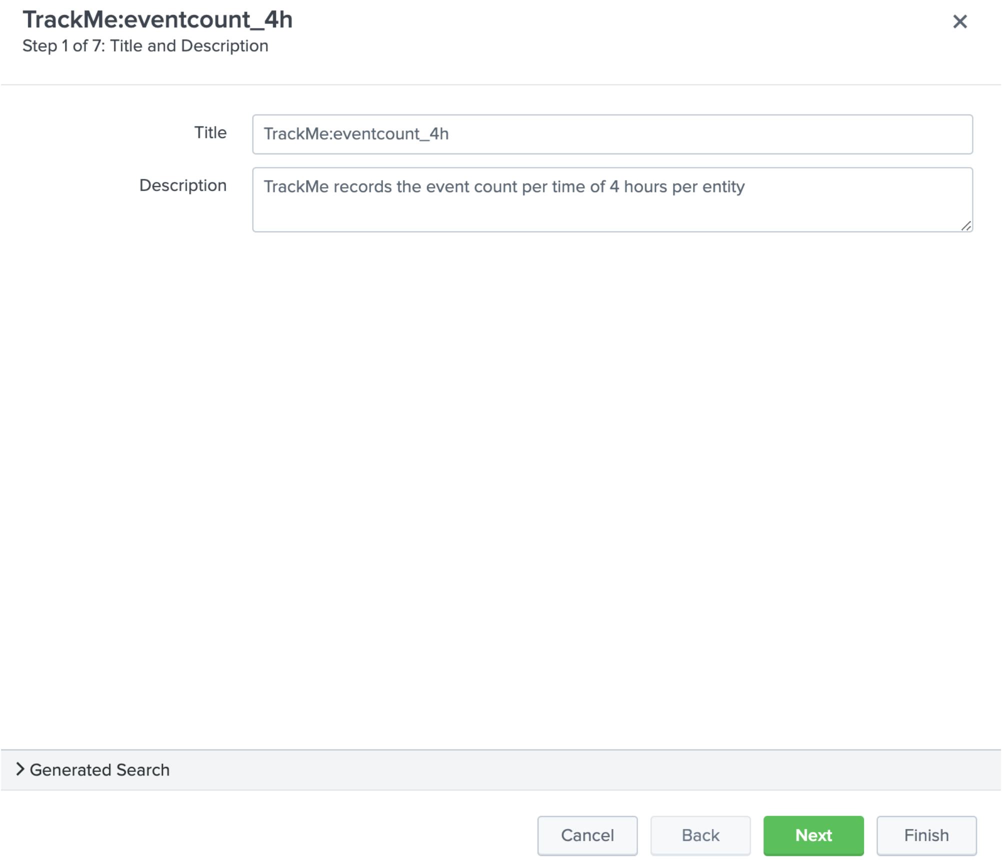
Matched Entities

No entities currently matched. They will be automatically added to the service when created.

Add the KPIs to the service, as follows:

TrackMe:eventcount_4h

- Title: TrackMe:eventcount_4h
- Description: TrackMe records the event count per time of 4 hours per entity

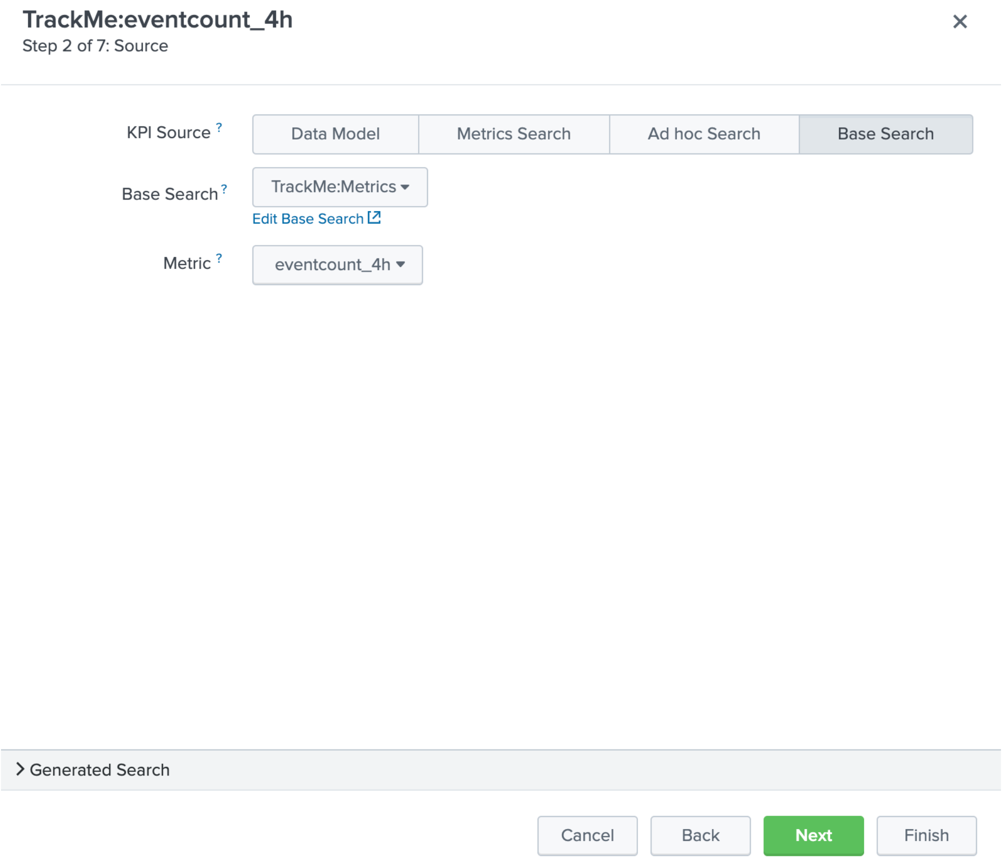


TrackMe:eventcount_4h

Step 1 of 7: Title and Description

Title: TrackMe:eventcount_4h

Description: TrackMe records the event count per time of 4 hours per entity



TrackMe:eventcount_4h

Step 2 of 7: Source

KPI Source: Base Search

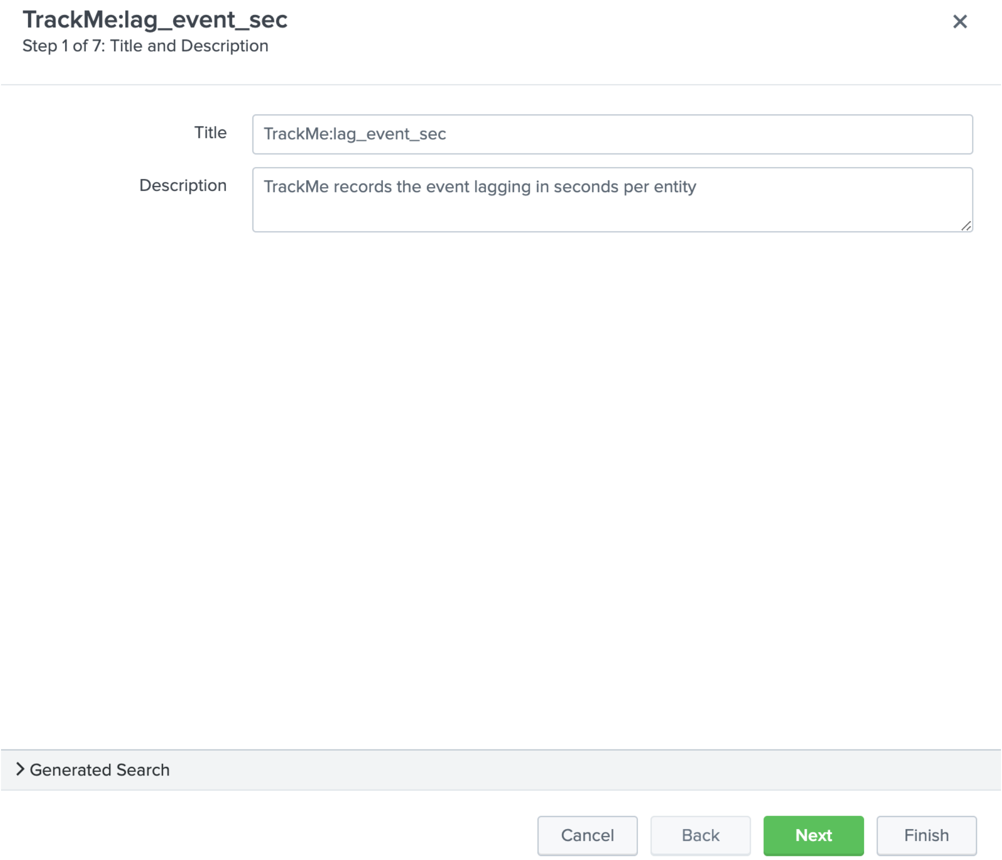
Base Search: TrackMe:Metrics

Metric: eventcount_4h

Click next until you can hit the finish button.

TrackMe:lag_event_sec

- Title: TrackMe:lag_event_sec
- Description: TrackMe records the event lagging in seconds per entity

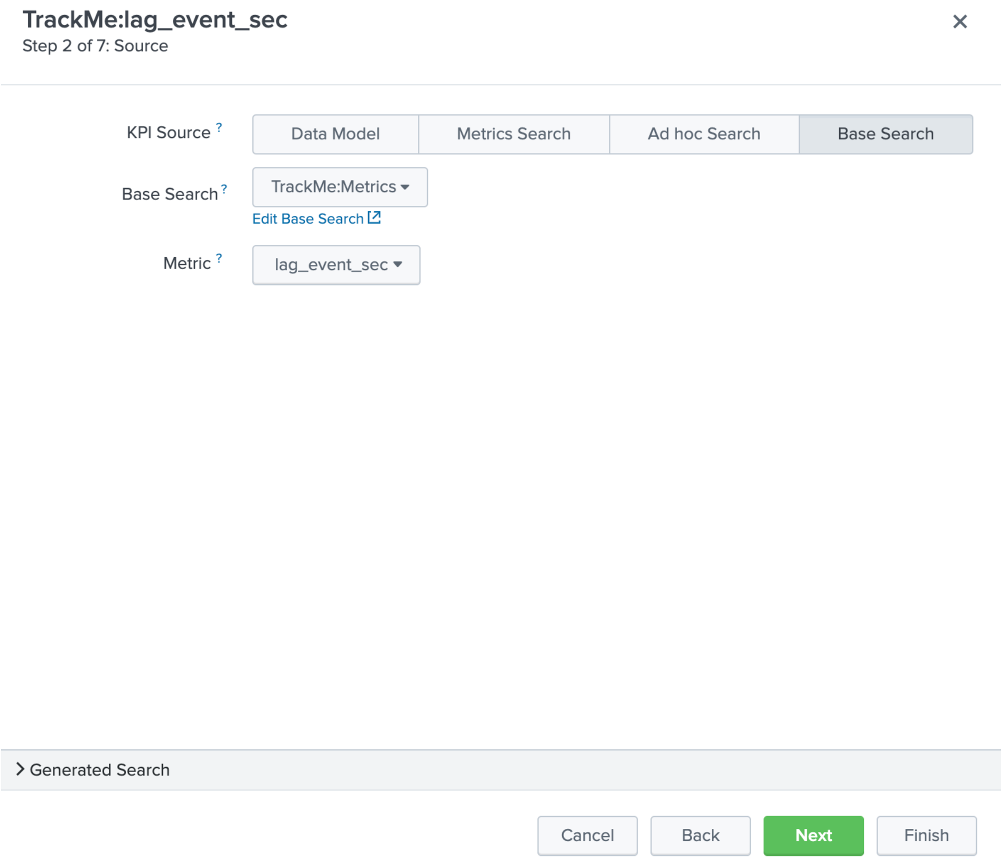


TrackMe:lag_event_sec

Step 1 of 7: Title and Description

Title: TrackMe:lag_event_sec

Description: TrackMe records the event lagging in seconds per entity



TrackMe:lag_event_sec
Step 2 of 7: Source

KPI Source: Base Search

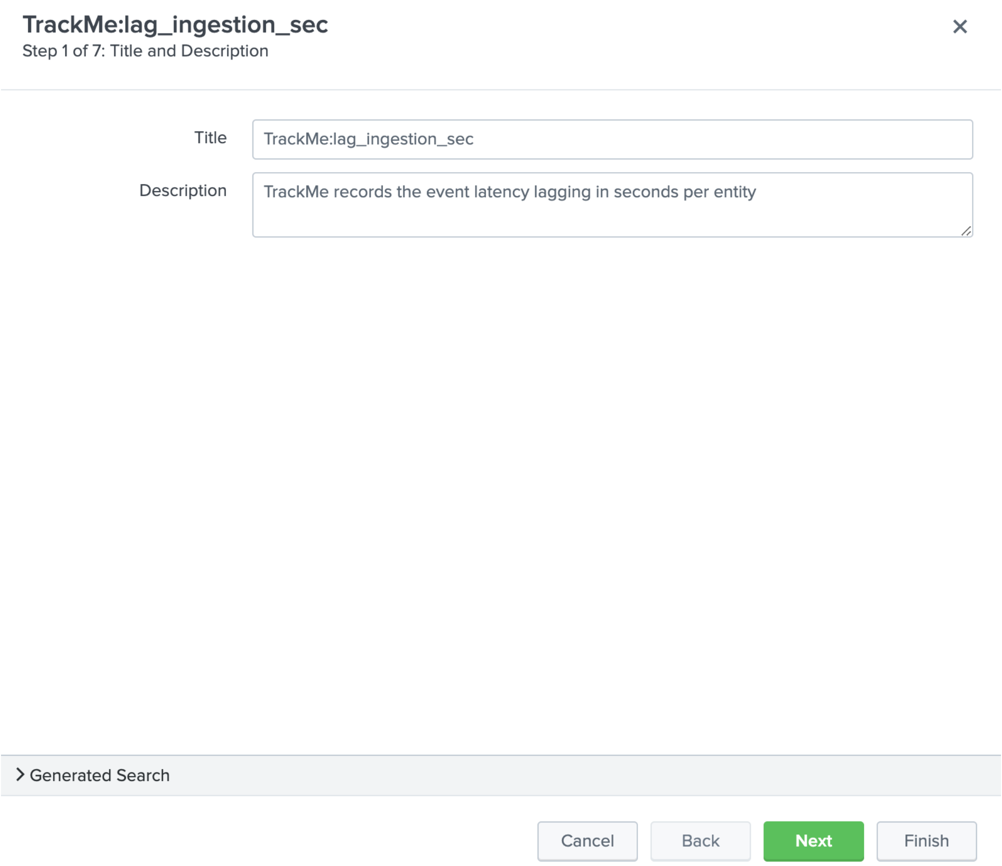
Base Search: TrackMe:Metrics

Metric: lag_event_sec

Click next until you can hit the finish button.

TrackMe:lag_ingestion_sec

- Title: TrackMe:lag_ingestion_sec
- Description: TrackMe records the event latency lagging in seconds per entity

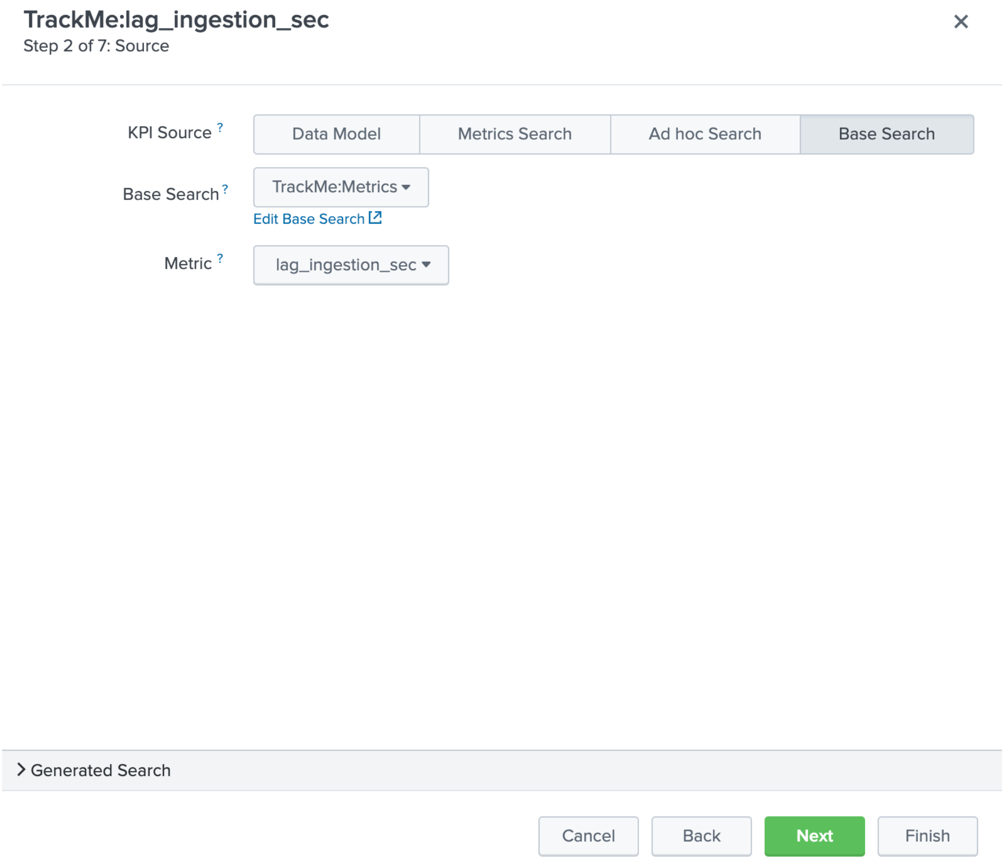


TrackMe:lag_ingestion_sec

Step 1 of 7: Title and Description

Title: TrackMe:lag_ingestion_sec

Description: TrackMe records the event latency lagging in seconds per entity



TrackMe:lag_ingestion_sec

Step 2 of 7: Source

KPI Source: Base Search

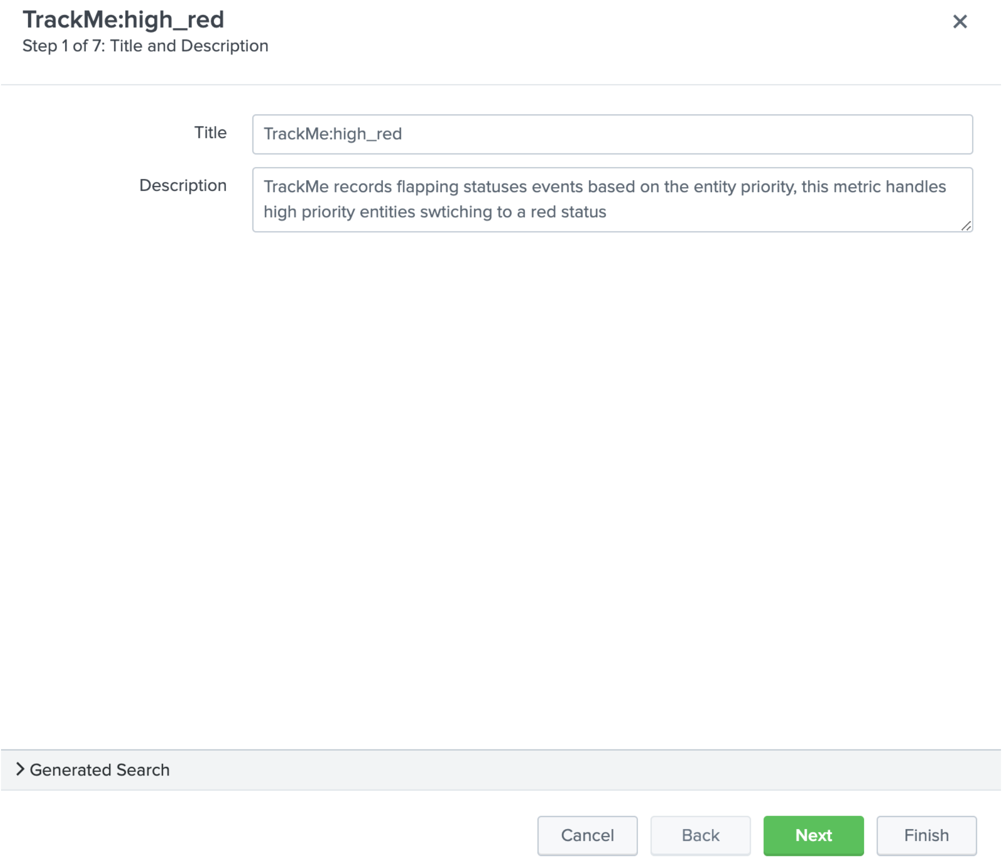
Base Search: TrackMe:Metrics

Metric: lag_ingestion_sec

Click next until you can hit the finish button.

TrackMe:medium_red

- Title: TrackMe:high_red
- Description: TrackMe records flapping statuses events based on the entity priority, this metric handles high priority entities swtiching to a red status

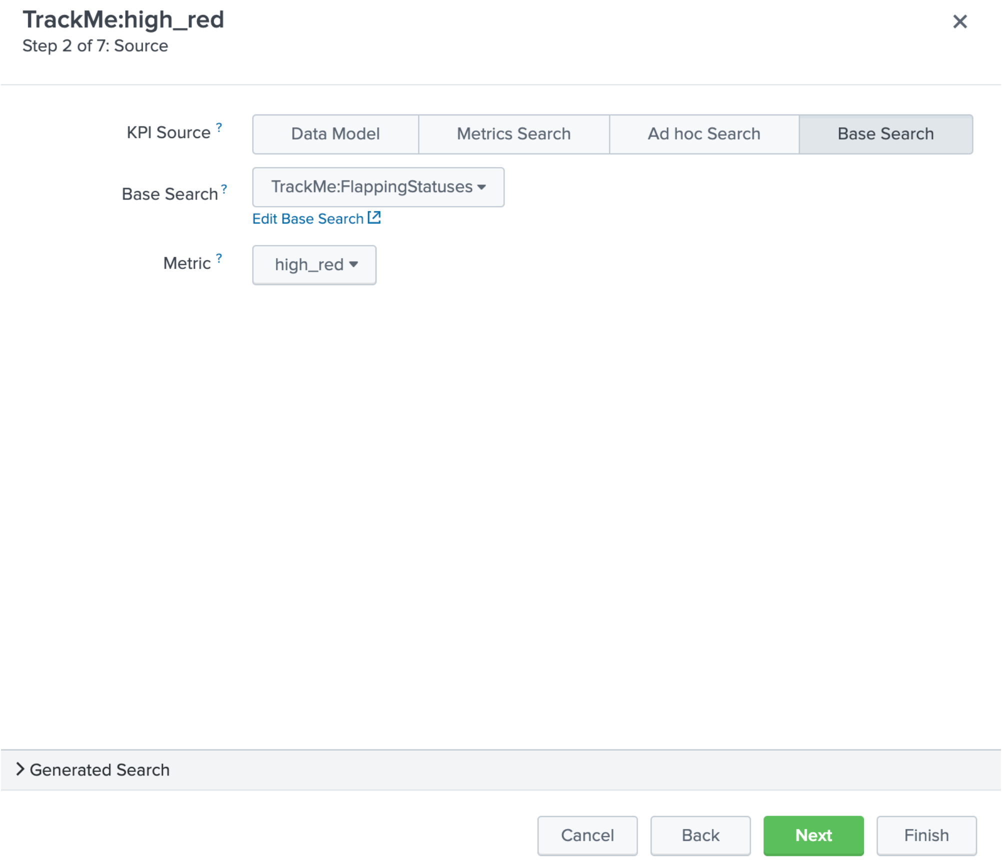


TrackMe:high_red

Step 1 of 7: Title and Description

Title: TrackMe:high_red

Description: TrackMe records flapping statuses events based on the entity priority, this metric handles high priority entities swtiching to a red status



TrackMe:high_red

Step 2 of 7: Source

KPI Source: Base Search

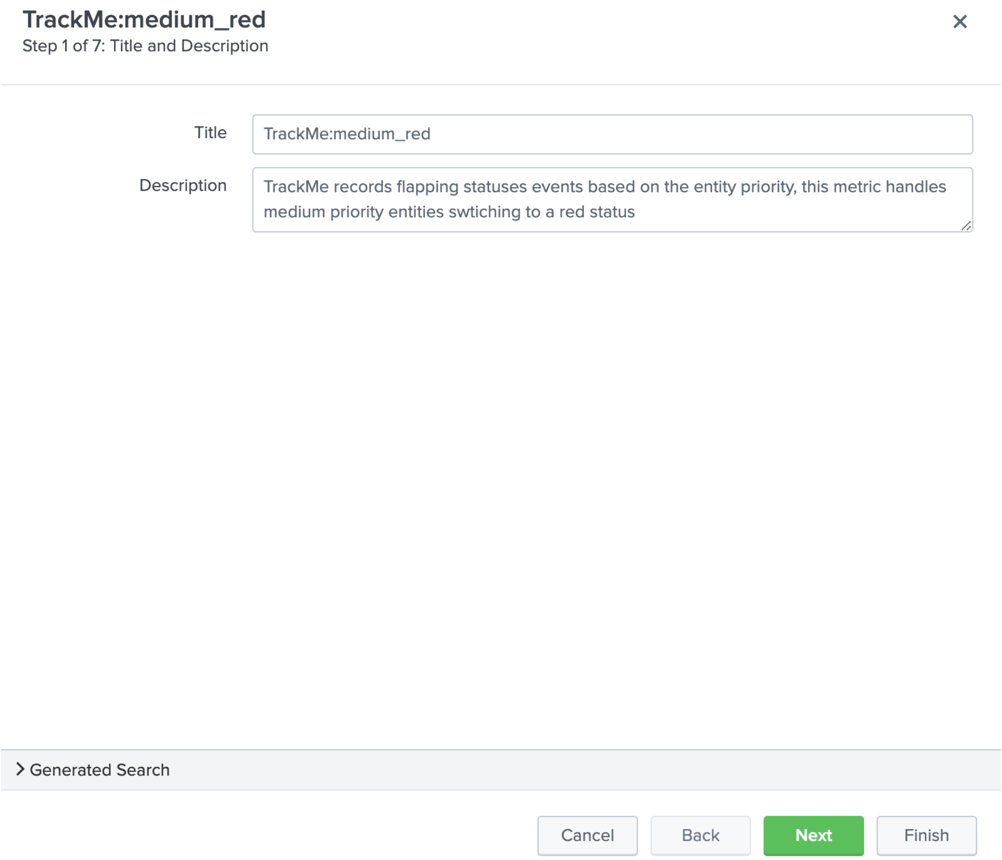
Base Search: TrackMe:FlappingStatuses

Metric: high_red

Click next until you can hit the finish button.

TrackMe:high_red

- Title: TrackMe:medium_red
- Description: TrackMe records flapping statuses events based on the entity priority, this metric handles medium priority entities swtiching to a red status

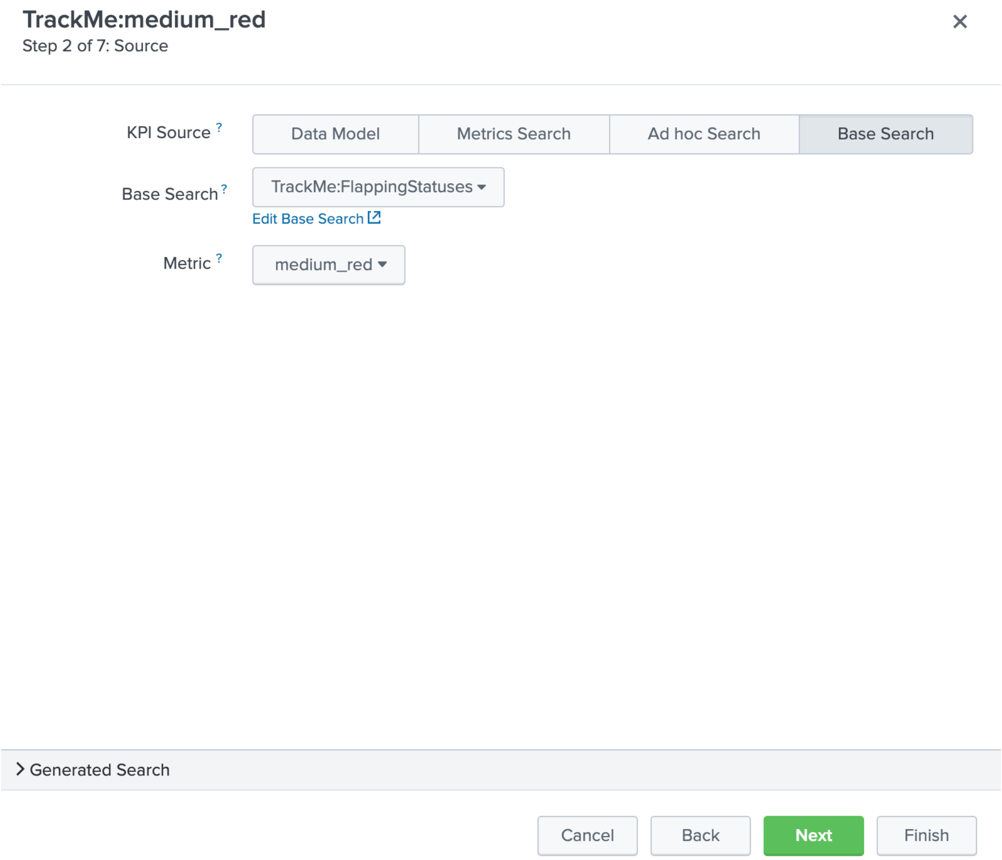


TrackMe:medium_red

Step 1 of 7: Title and Description

Title: TrackMe:medium_red

Description: TrackMe records flapping statuses events based on the entity priority, this metric handles medium priority entities swtiching to a red status



TrackMe:medium_red

Step 2 of 7: Source

KPI Source: Base Search

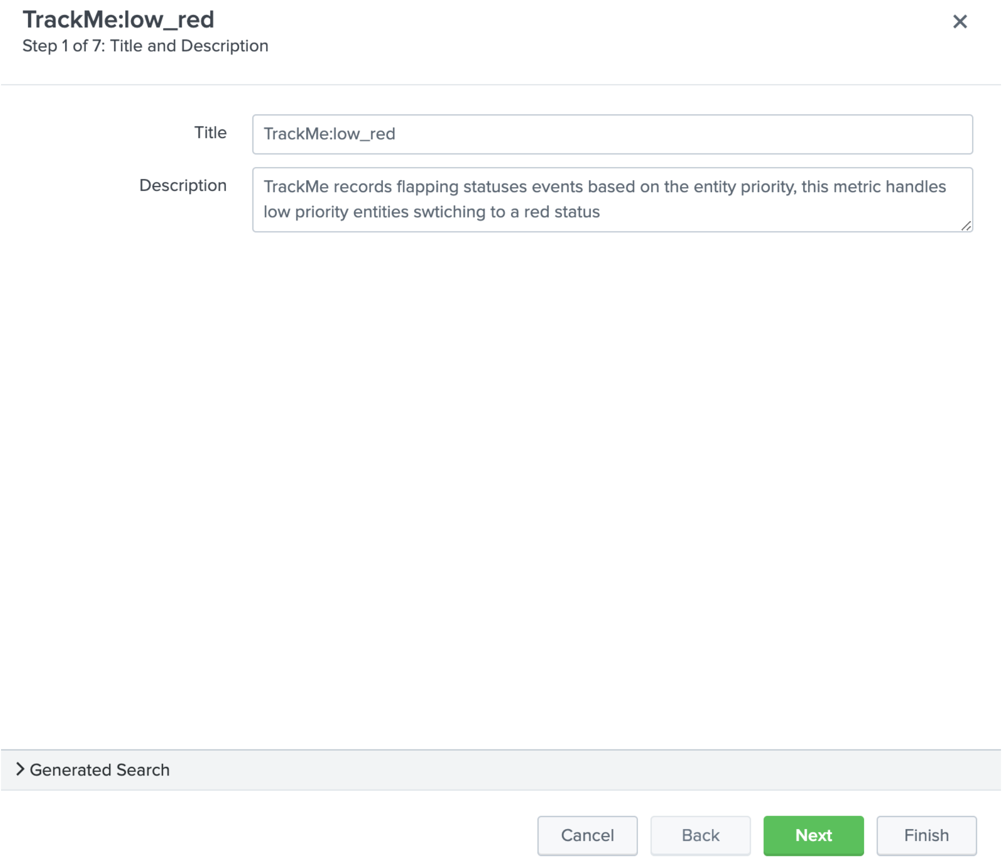
Base Search: TrackMe:FlappingStatuses

Metric: medium_red

Click next until you can hit the finish button.

TrackMe:low_red

- Title: TrackMe:low_red
- Description: TrackMe records flapping statuses events based on the entity priority, this metric handles low priority entities swtiching to a red status

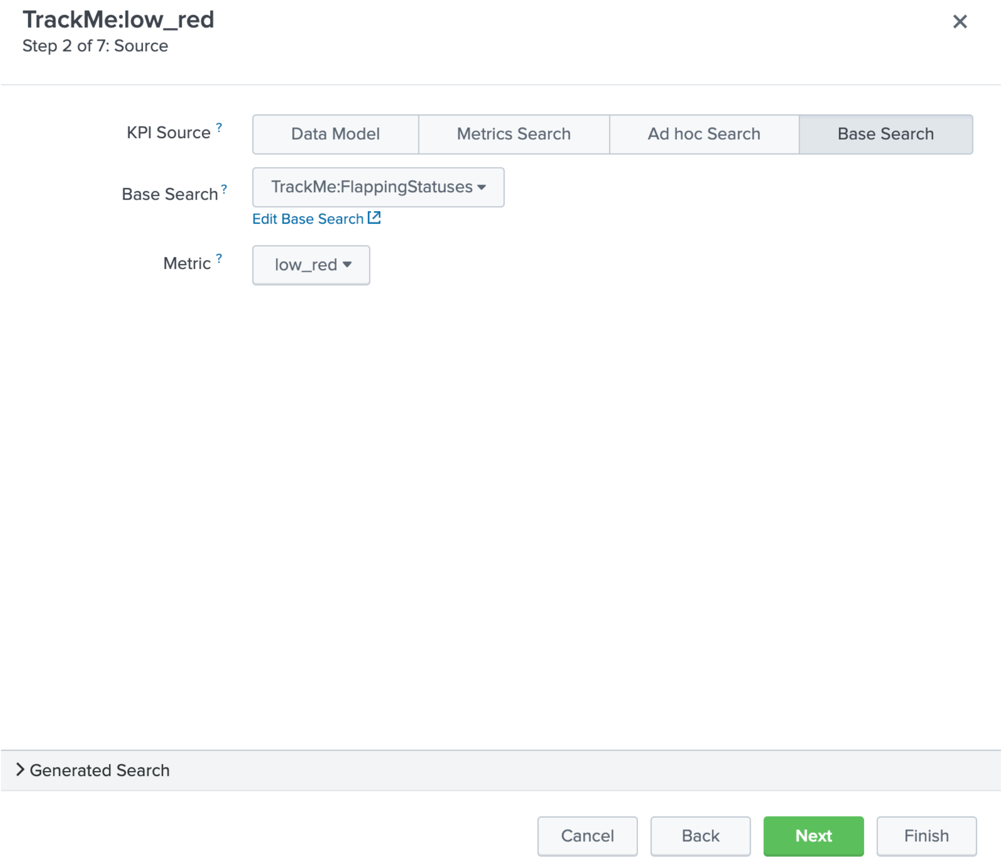


TrackMe:low_red

Step 1 of 7: Title and Description

Title: TrackMe:low_red

Description: TrackMe records flapping statuses events based on the entity priority, this metric handles low priority entities swtiching to a red status



TrackMe:low_red

Step 2 of 7: Source

KPI Source: Base Search

Base Search: TrackMe:FlappingStatuses

Metric: low_red

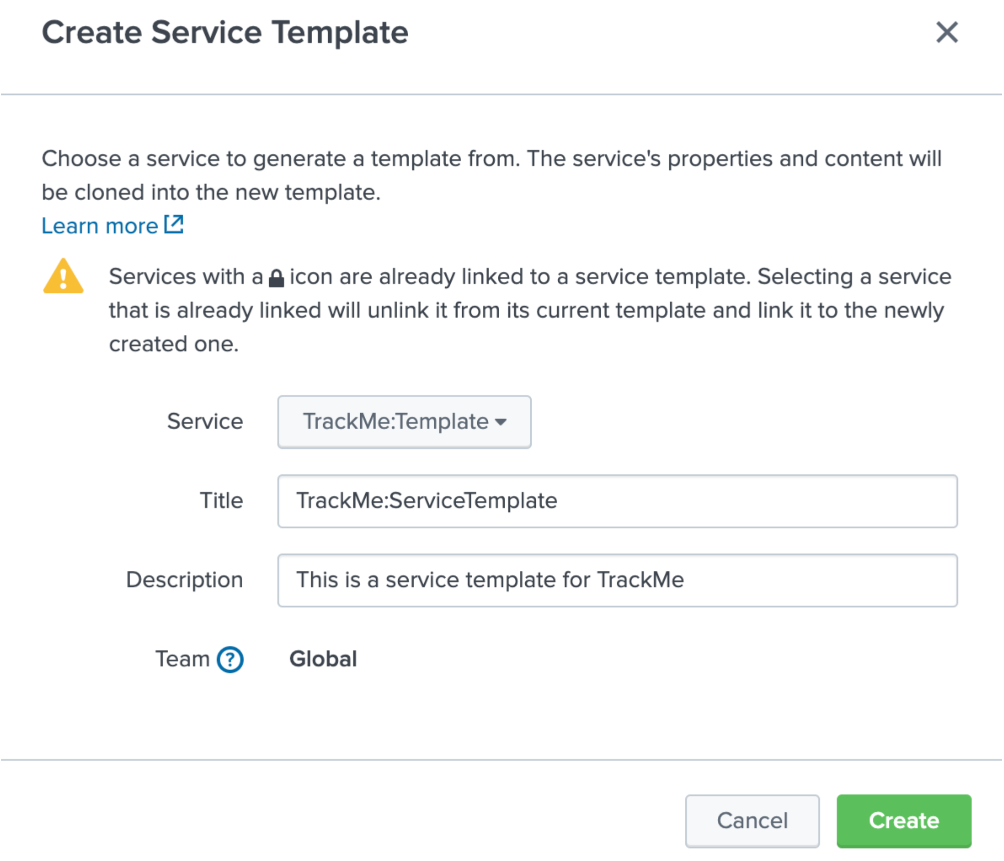
Click next until you can hit the finish button.

Note: This pseudo service can optionally be deleted post service template creation, but you can as well keep it to allow future service creation based on this service rather using the service template feature.

3.2.5 Step 5: create a service template

Now that we have a pseudo service, we can create a service template based on it, the service template would be used to create and link new services:

- Click on Configure / Services Templates
- Use the previously created pseudo service to create a new service template
- Any future customization of this service template will be reflected to all linked services (which can be controlled when modifications on the template are made)



Create Service Template

Choose a service to generate a template from. The service's properties and content will be cloned into the new template.

Learn more

⚠ Services with a 🔒 icon are already linked to a service template. Selecting a service that is already linked will unlink it from its current template and link it to the newly created one.

Service: TrackMe:Template

Title: TrackMe:ServiceTemplate

Description: This is a service template for TrackMe

Team: Global

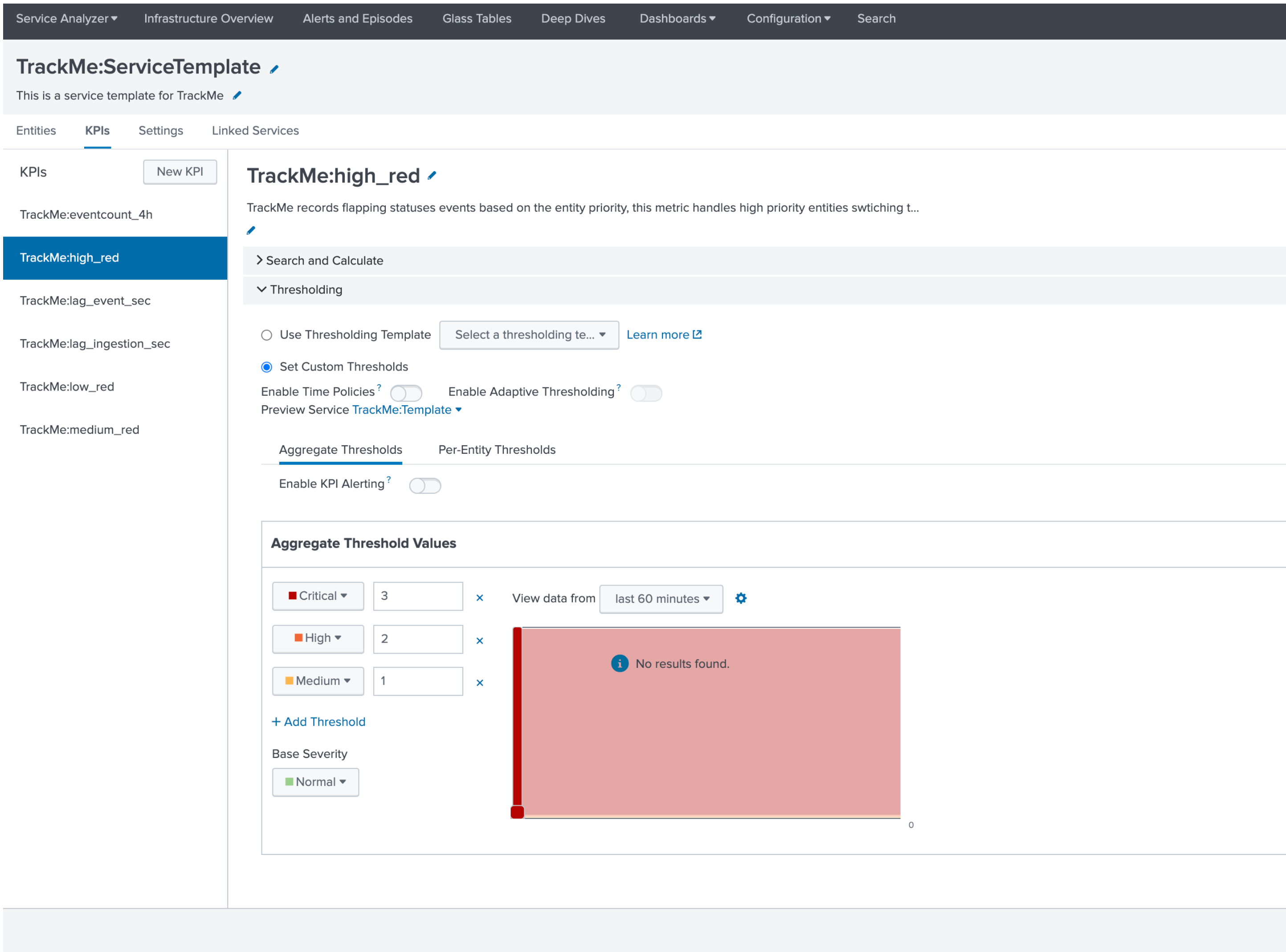
3.2.6 Step 6: fine tune thresholds

On the service template, you can fine tune some of the thresholds, essentially regarding the status flapping metrics.

The thresholds related to the maximal lagging values and evencount would be fine tuned on a per service basis.

Fine tuning the flapping statuses:

At the minimum, if TrackMe detects an issue with the entity, the ITSI service should reflect the issue on the TrackMe notation, such as:



TrackMe:ServiceTemplate

This is a service template for TrackMe

TrackMe:high_red

TrackMe records flapping statuses events based on the entity priority, this metric handles high priority entities swtiching t...

○ Use Thresholding Template: Select a thresholding te... ▾

● Set Custom Thresholds

Enable Time Policies: off

Enable Adaptive Thresholding: off

Preview Service: TrackMe:Template ▾

Enable KPI Alerting: off

Aggregate Threshold Values

🟥 Critical ▾	3
🟧 High ▾	2
🟨 Medium ▾	1

View data from: last 60 minutes ▾

Base Severity: 🟩 Normal ▾

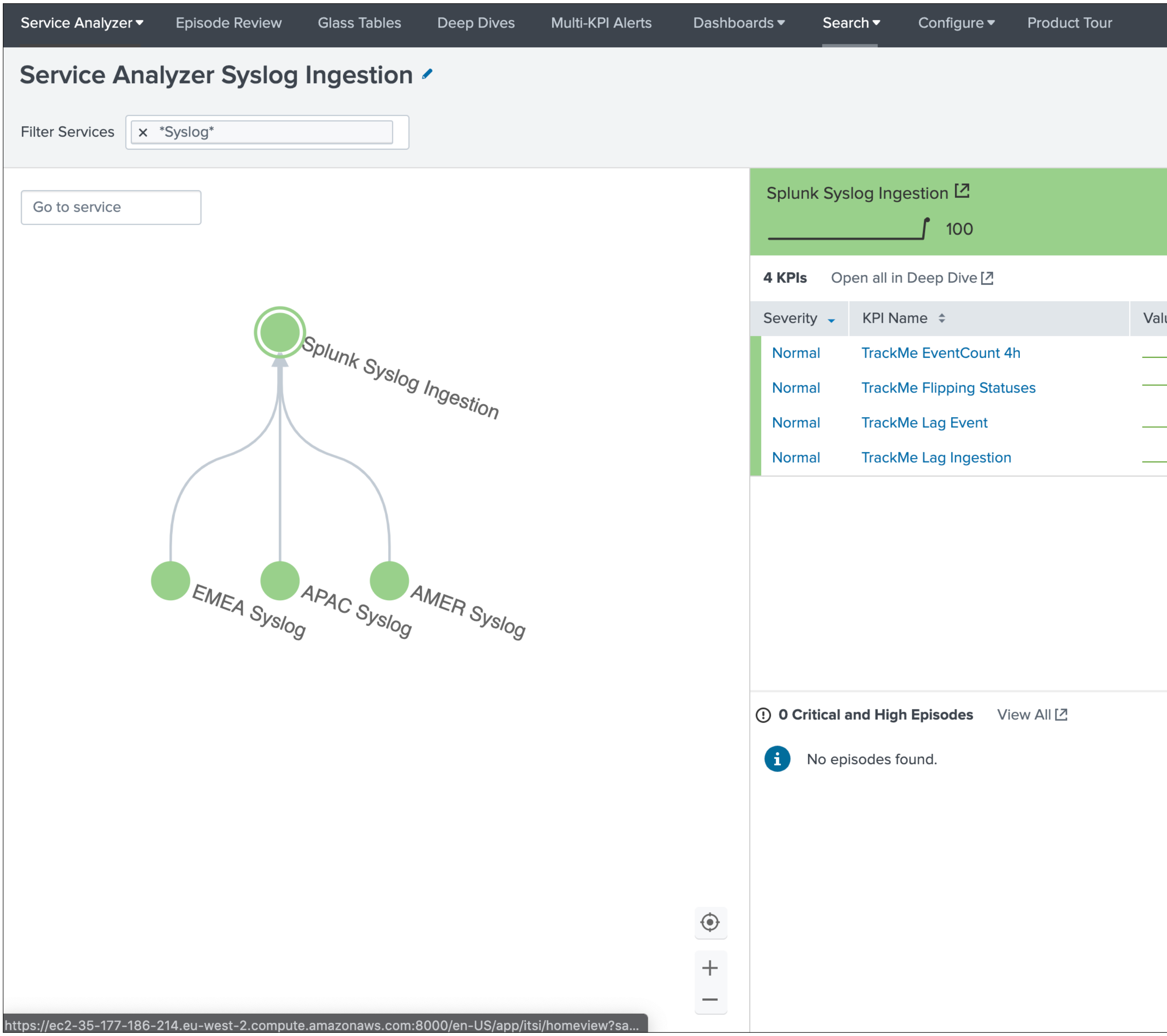
Make sure to apply the same thresholds per entity, and reflect the same change on medium_red and low_red metrics. (with potentially different values if necessary)

When business and technical services are created, you potentially can fine tune the other metrics up to the requirements, note that TrackMe settings for that or these entities composing the service are reflected in anyway using the flapping statuses metrics.

3.2.7 Final: Create services business and technical services using TrackMe KPIs

Finally, the ITSI integration is ready and you can create new services using the template service or manually cloning the pseudo service we created earlier.

Once you created and activated a new service, ITSI will start to report TrackMe KPIs after a short moment: (metrics are generated every 5 minutes)



Service Analyzer Syslog Ingestion

Filter Services: × *Syslog*

Splunk Syslog Ingestion 100

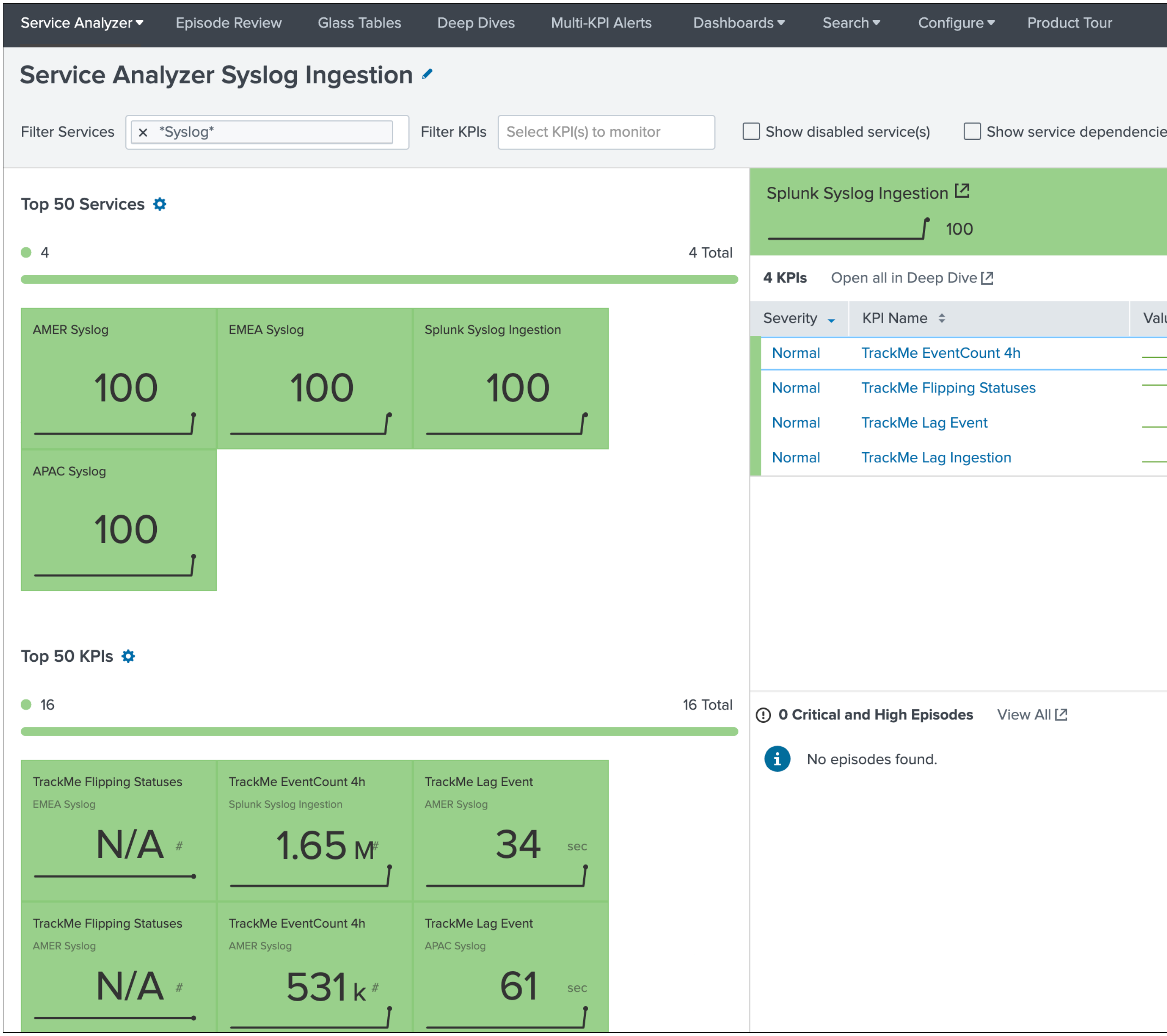
4 KPIs   Open all in Deep Dive

Severity	KPI Name	Valu
Normal	TrackMe EventCount 4h	
Normal	TrackMe Flipping Statuses	
Normal	TrackMe Lag Event	
Normal	TrackMe Lag Ingestion	

0 Critical and High Episodes   View All

No episodes found.

https://ec2-35-177-186-214.eu-west-2.compute.amazonaws.com:8000/en-US/app/itsi/homeview?sa...



Service Analyzer Syslog Ingestion

Filter Services × *Syslog*   Filter KPIs Select KPI(s) to monitor   □ Show disabled service(s)   □ Show service dependencie

Top 50 Services

● 4 — 4 Total

AMER Syslog	EMEA Syslog	Splunk Syslog Ingestion
100	100	100
APAC Syslog		
100		

Top 50 KPIs

● 16 — 16 Total

TrackMe Flipping Statuses EMEA Syslog	TrackMe EventCount 4h Splunk Syslog Ingestion	TrackMe Lag Event AMER Syslog
N/A #	1.65 M #	34 sec
TrackMe Flipping Statuses AMER Syslog	TrackMe EventCount 4h AMER Syslog	TrackMe Lag Event APAC Syslog
N/A #	531 k #	61 sec

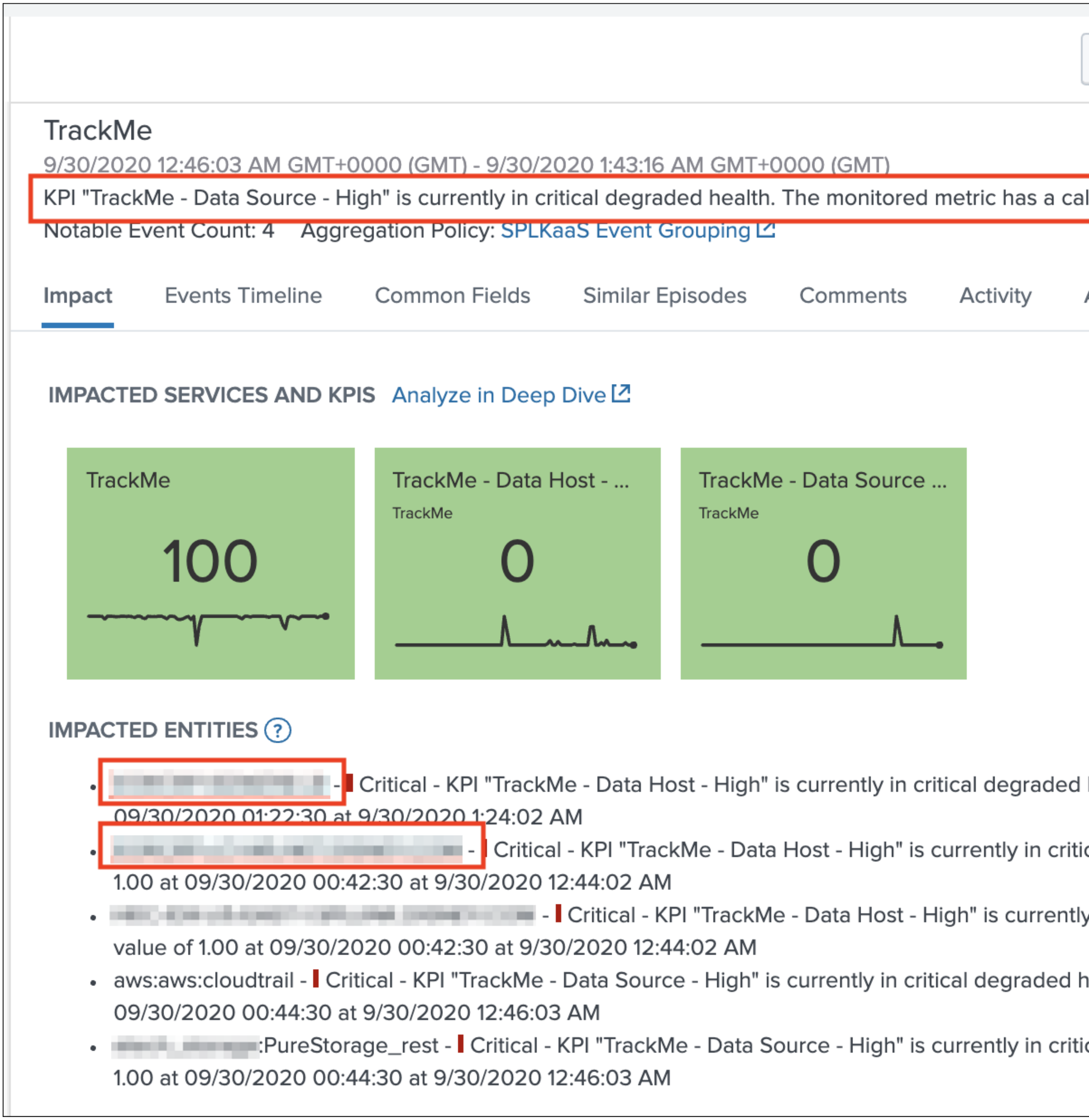
Splunk Syslog Ingestion 100

4 KPIs   Open all in Deep Dive

Severity	KPI Name	Valu
Normal	TrackMe EventCount 4h	
Normal	TrackMe Flipping Statuses	
Normal	TrackMe Lag Event	
Normal	TrackMe Lag Ingestion	

0 Critical and High Episodes   View All

No episodes found.



TrackMe

9/30/2020 12:46:03 AM GMT+0000 (GMT) - 9/30/2020 1:43:16 AM GMT+0000 (GMT)

KPI "TrackMe - Data Source - High" is currently in critical degraded health. The monitored metric has a cal

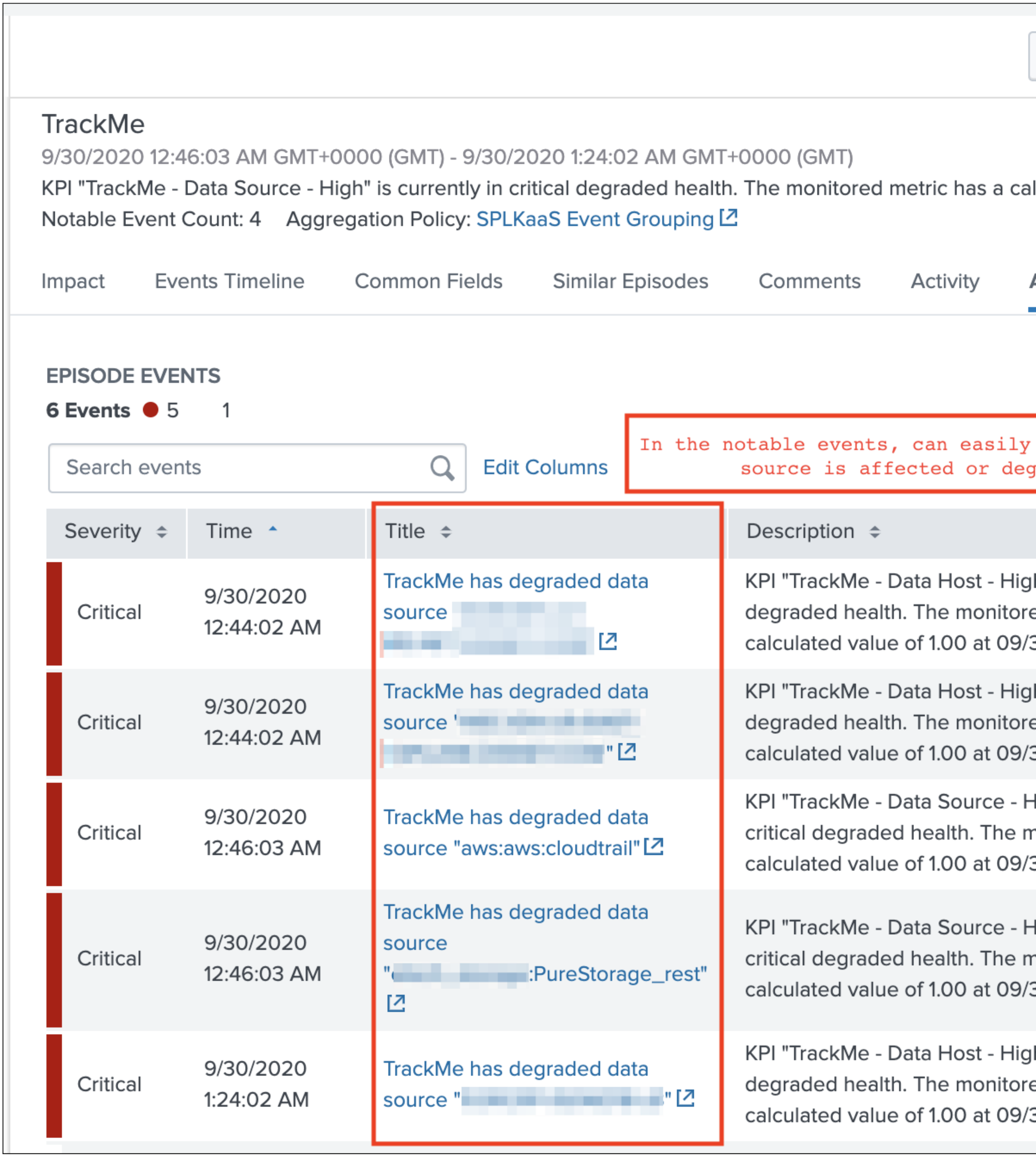
Notable Event Count: 4   Aggregation Policy: SPLKaaS Event Grouping

IMPACTED SERVICES AND KPIS Analyze in Deep Dive

TrackMe	TrackMe - Data Host - ... TrackMe	TrackMe - Data Source ... TrackMe
100	0	0

IMPACTED ENTITIES

- [redacted] - Critical - KPI "TrackMe - Data Host - High" is currently in critical degraded 09/30/2020 01:22:30 at 9/30/2020 1:24:02 AM
- [redacted] - Critical - KPI "TrackMe - Data Host - High" is currently in critic 1.00 at 09/30/2020 00:42:30 at 9/30/2020 12:44:02 AM
- [redacted] - Critical - KPI "TrackMe - Data Host - High" is currently value of 1.00 at 09/30/2020 00:42:30 at 9/30/2020 12:44:02 AM
- aws:aws:cloudtrail - Critical - KPI "TrackMe - Data Source - High" is currently in critical degraded h 09/30/2020 00:44:30 at 9/30/2020 12:46:03 AM
- [redacted]:PureStorage_rest - Critical - KPI "TrackMe - Data Source - High" is currently in critic 1.00 at 09/30/2020 00:44:30 at 9/30/2020 12:46:03 AM



TrackMe

9/30/2020 12:46:03 AM GMT+0000 (GMT) - 9/30/2020 1:24:02 AM GMT+0000 (GMT)

KPI "TrackMe - Data Source - High" is currently in critical degraded health. The monitored metric has a cal

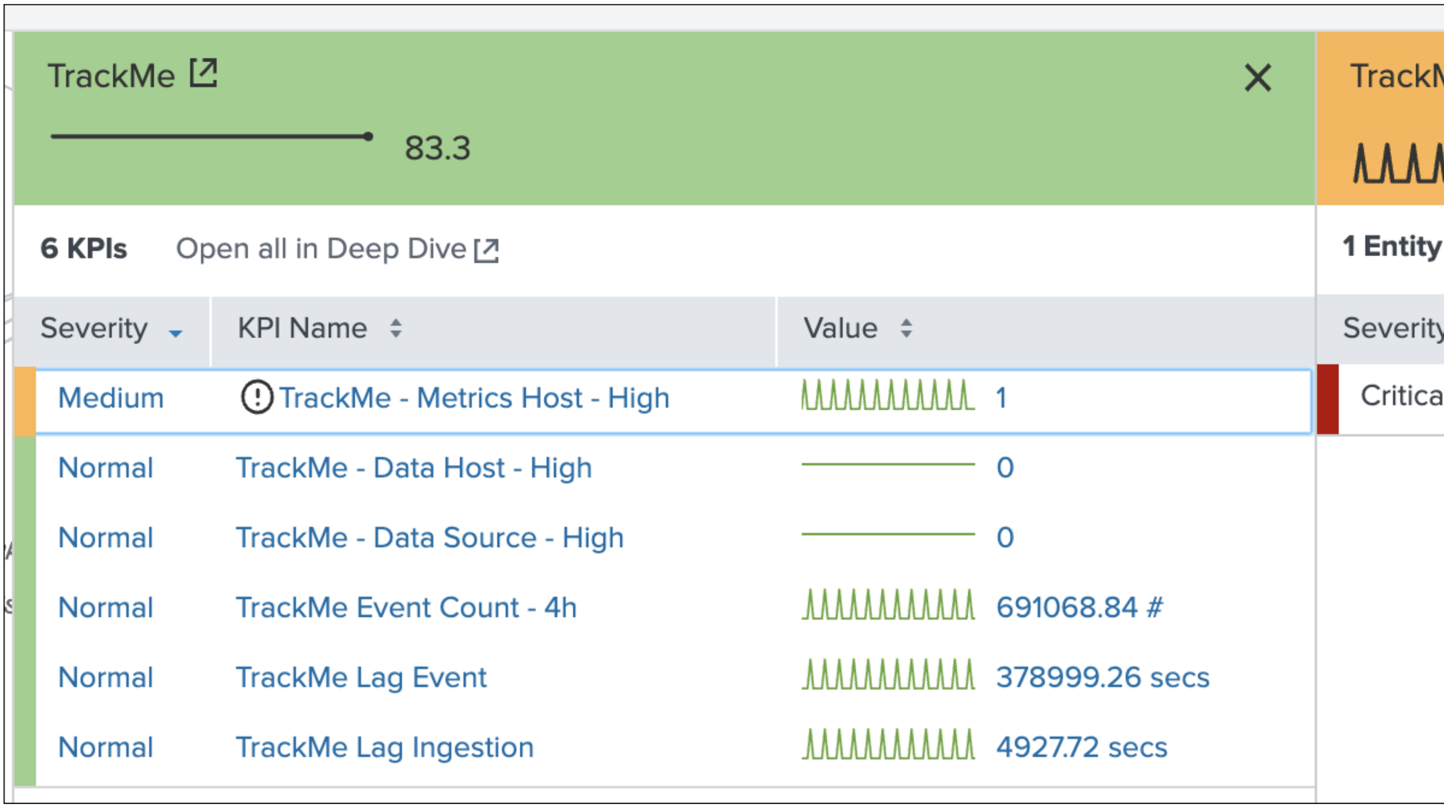
Notable Event Count: 4 Aggregation Policy: SPLKaaS Event Grouping

EPISODE EVENTS

6 Events ● 5 1

In the notable events, can easily
source is affected or deg

Severity	Time	Title	Description
Critical	9/30/2020 12:44:02 AM	TrackMe has degraded data source [redacted]	KPI "TrackMe - Data Host - Higl degraded health. The monitore calculated value of 1.00 at 09/3
Critical	9/30/2020 12:44:02 AM	TrackMe has degraded data source '[redacted]"	KPI "TrackMe - Data Host - Higl degraded health. The monitore calculated value of 1.00 at 09/3
Critical	9/30/2020 12:46:03 AM	TrackMe has degraded data source "aws:aws:cloudtrail"	KPI "TrackMe - Data Source - H critical degraded health. The m calculated value of 1.00 at 09/3
Critical	9/30/2020 12:46:03 AM	TrackMe has degraded data source "[redacted]:PureStorage_rest"	KPI "TrackMe - Data Source - H critical degraded health. The m calculated value of 1.00 at 09/3
Critical	9/30/2020 1:24:02 AM	TrackMe has degraded data source "[redacted]"	KPI "TrackMe - Data Host - Higl degraded health. The monitore calculated value of 1.00 at 09/3



TrackMe

83.3

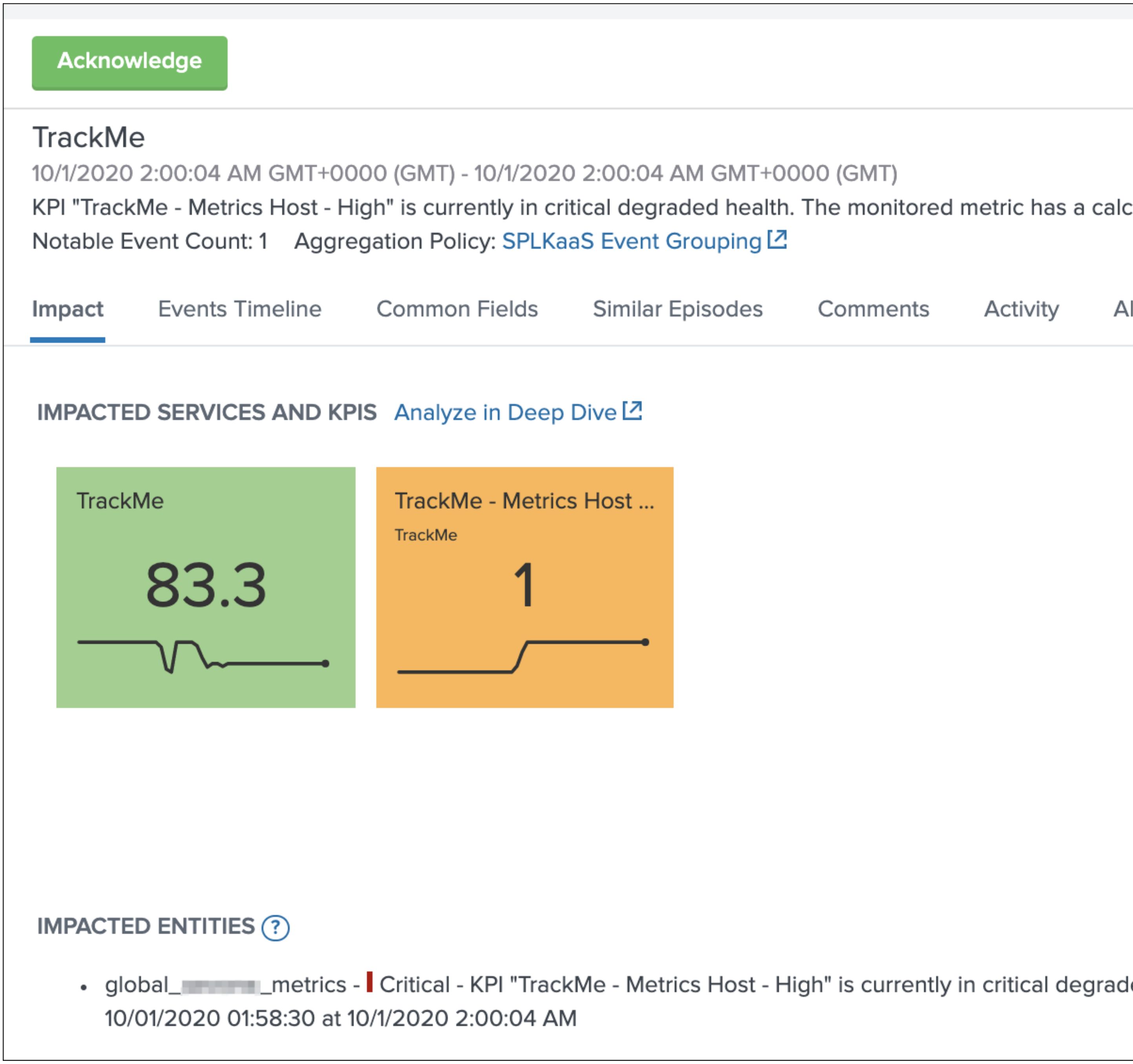
6 KPIs Open all in Deep Dive

Severity	KPI Name	Value
Medium	TrackMe - Metrics Host - High	1
Normal	TrackMe - Data Host - High	0
Normal	TrackMe - Data Source - High	0
Normal	TrackMe Event Count - 4h	691068.84 #
Normal	TrackMe Lag Event	378999.26 secs
Normal	TrackMe Lag Ingestion	4927.72 secs

TrackM

1 Entity

Severity
Critica



TrackMe

10/1/2020 2:00:04 AM GMT+0000 (GMT) - 10/1/2020 2:00:04 AM GMT+0000 (GMT)

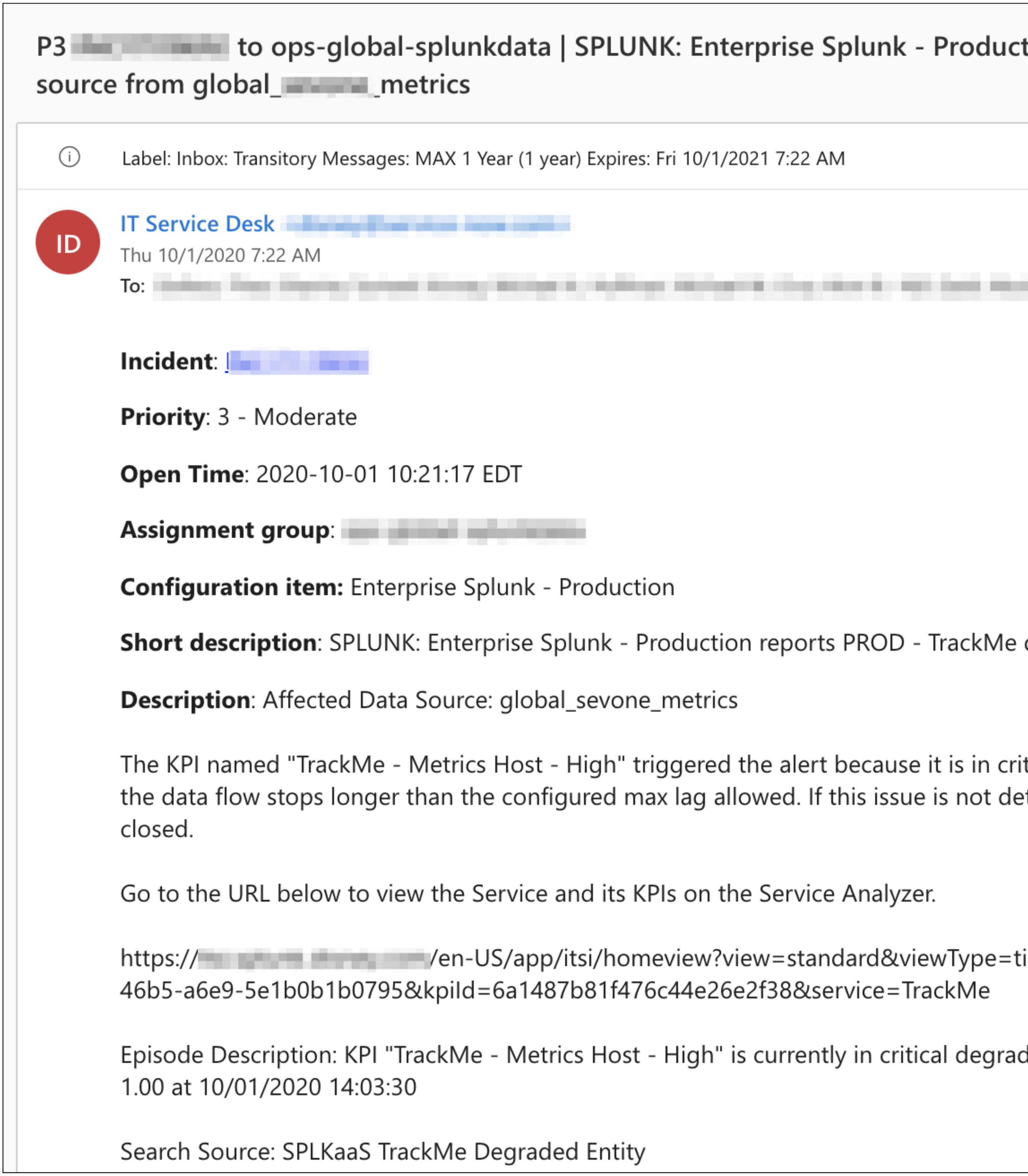
KPI "TrackMe - Metrics Host - High" is currently in critical degraded health. The monitored metric has a calc

Notable Event Count: 1   Aggregation Policy: SPLKaaS Event Grouping

IMPACTED SERVICES AND KPIS Analyze in Deep Dive

IMPACTED ENTITIES

- global_[redacted]_metrics - Critical - KPI "TrackMe - Metrics Host - High" is currently in critical degrad
  10/01/2020 01:58:30 at 10/1/2020 2:00:04 AM



P3 [redacted] to ops-global-splunkdata | SPLUNK: Enterprise Splunk - Product
source from global_[redacted]_metrics

Label: Inbox: Transitory Messages: MAX 1 Year (1 year) Expires: Fri 10/1/2021 7:22 AM

IT Service Desk [redacted]
Thu 10/1/2020 7:22 AM
To: [redacted]

Incident: [redacted]

Priority: 3 - Moderate

Open Time: 2020-10-01 10:21:17 EDT

Assignment group: [redacted]

Configuration item: Enterprise Splunk - Production

Short description: SPLUNK: Enterprise Splunk - Production reports PROD - TrackMe

Description: Affected Data Source: global_sevone_metrics

The KPI named "TrackMe - Metrics Host - High" triggered the alert because it is in crit
the data flow stops longer than the configured max lag allowed. If this issue is not de
closed.

Go to the URL below to view the Service and its KPIs on the Service Analyzer.

https://[redacted]/en-US/app/itsi/homeview?view=standard&viewType=ti
46b5-a6e9-5e1b0b1b0795&kpiId=6a1487b81f476c44e26e2f38&service=TrackMe

Episode Description: KPI "TrackMe - Metrics Host - High" is currently in critical degrad
1.00 at 10/01/2020 14:03:30

Search Source: SPLKaaS TrackMe Degraded Entity

TrackMe acts now transparently as a companion of ITSI, you will continue to manage data sources in TrackMe, create Elastic sources, manage states and max lagging values which are reflected naturally in ITSI.

3.3 Cribl LogStream and TrackMe integration

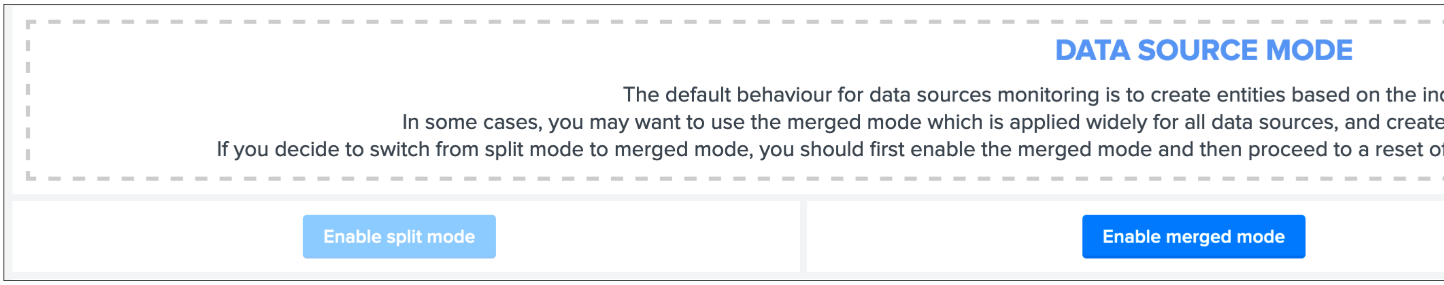
If you are using Cribl LogStream, you can easily integrate TrackMe in a just a few steps, using the excellent native Cribl LogStream design, TrackMe will take into account the concept of pipelines to create, monitor and render the data sources automatically.

In a nutshell:

- A configuration parameter is available in TrackMe to enable the Cribl mode
- Once activated, the Cribl mode updates the way TrackMe is identifying and breaking the data sources
- To achieve this, TrackMe relies on the **cribl_pipe** indexed field automatically created by LogStream when data is indexed in Splunk
- Related searches transparenly use the **cribl_pipe** information, that accurately represents the data pipeline as it should be monitored, from LogStream to Splunk

3.3.1 Enable the Cribl mode

To enable the Cribl mode, go in "TrackMe manage and configure" and click on the enable Cribl mode:



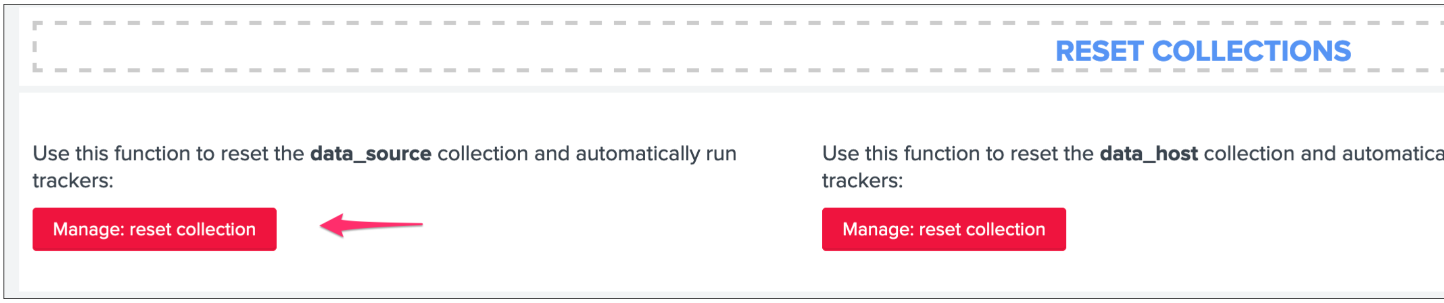
DATA SOURCE MODE

The default behaviour for data sources monitoring is to create entities based on the ind

In some cases, you may want to use the merged mode which is applied widely for all data sources, and create

If you decide to switch from split mode to merged mode, you should first enable the merged mode and then proceed to a reset of

Once the Cribl mode is enabled, perform a reset of the data source collection:



RESET COLLECTIONS

Use this function to reset the **data_source** collection and automatically run trackers:

Use this function to reset the **data_host** collection and automatica trackers:

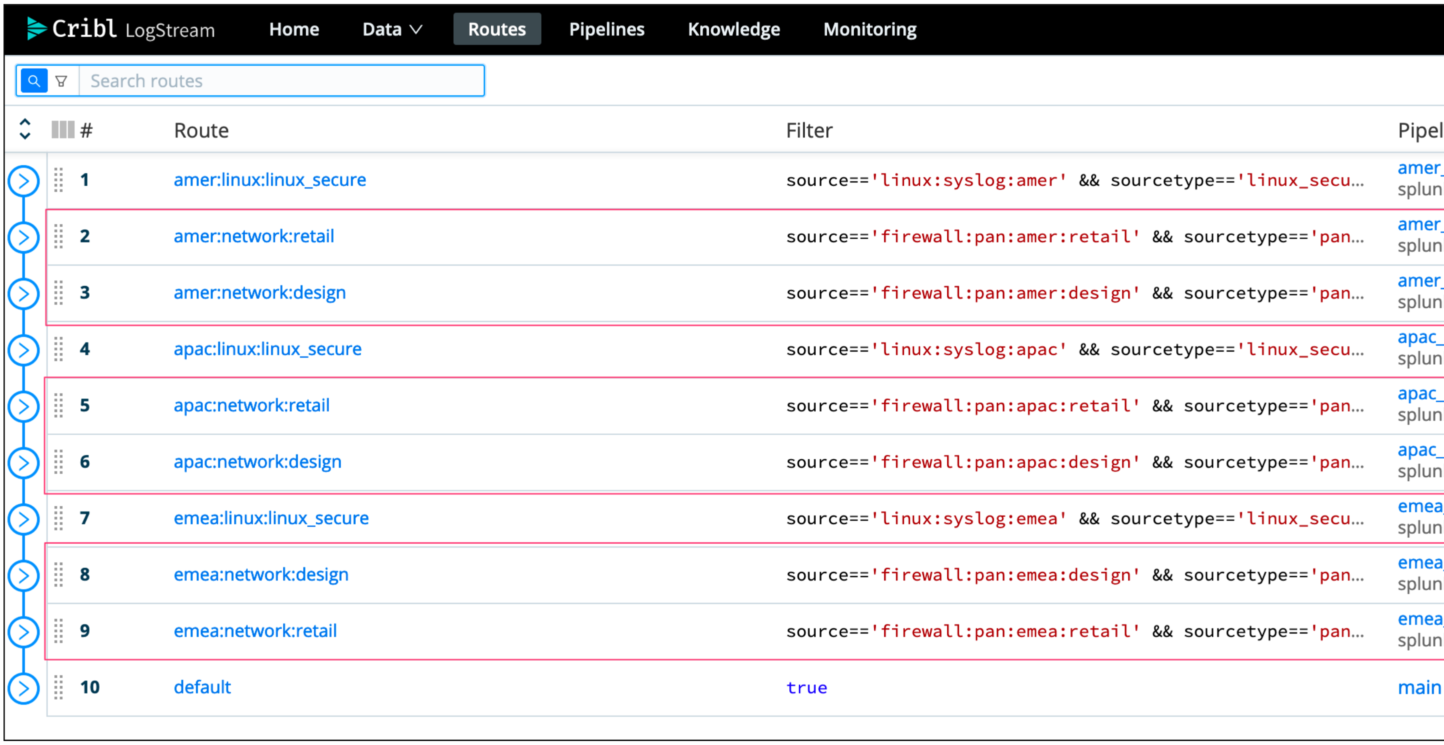
3.3.2 Cribl mode data sources

Let's assume the following simple scenario:

- Cribl LogStream receives incoming data from any kind of sources, and streams to Splunk with associated pipelines

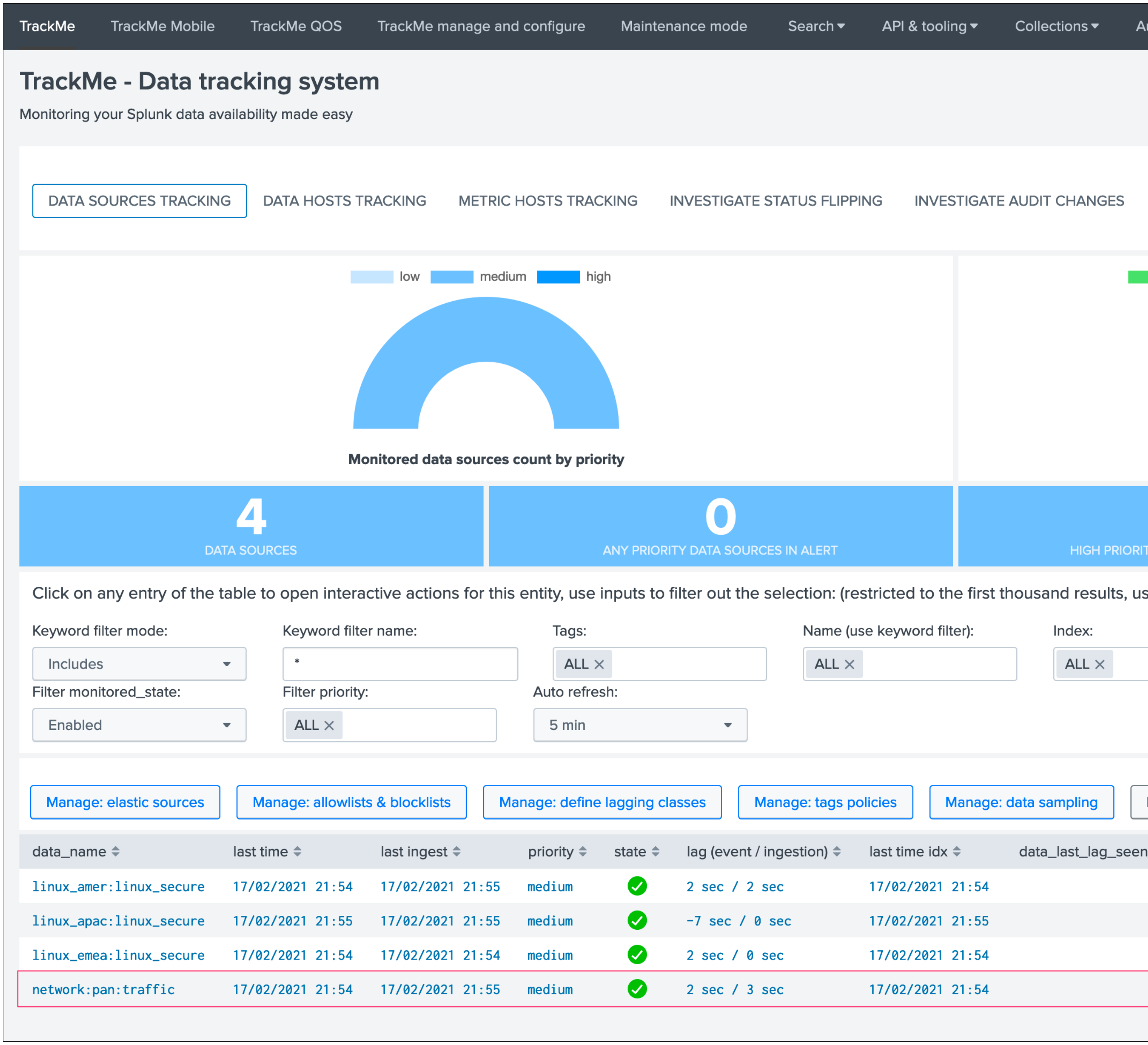
- In our example, we instruct LogStream to index data in Splunk into a few indexes, but we have many more pipelines since we perform various operations on LogStream, indexes and sourcetypes are likely fed by much more than just one pipeline
- In regular TrackMe mode, TrackMe would represent the data sources broken by indexes and sourcetypes, however, this does not represent what the incoming data flow is underneath, and does not provide the valuable information and monitoring layer we need
- Once we enable the Cribl mode, TrackMe relies on the `cribl_pipe` pipeline information to properly distinguish the real data flow as it is from the data provider (Cribl LogStream) perspective

Cribl LogStream pipeline examples:



#	Route	Filter	Pipel
1	amer:linux:linux_secure	source=='linux:syslog:amer' && sourcetype=='linux_secu...	amer_ splun
2	amer:network:retail	source=='firewall:pan:amer:retail' && sourcetype=='pan...	amer_ splun
3	amer:network:design	source=='firewall:pan:amer:design' && sourcetype=='pan...	amer_ splun
4	apac:linux:linux_secure	source=='linux:syslog:apac' && sourcetype=='linux_secu...	apac_ splun
5	apac:network:retail	source=='firewall:pan:apac:retail' && sourcetype=='pan...	apac_ splun
6	apac:network:design	source=='firewall:pan:apac:design' && sourcetype=='pan...	apac_ splun
7	emea:linux:linux_secure	source=='linux:syslog:emea' && sourcetype=='linux_secu...	emea splun
8	emea:network:design	source=='firewall:pan:emea:design' && sourcetype=='pan...	emea splun
9	emea:network:retail	source=='firewall:pan:emea:retail' && sourcetype=='pan...	emea splun
10	default	true	main

In this example, the default TrackMe mode has different issues, we stream data to an index called "network", however we have different pipelines that are potentially linked to multiple sources and from the LogStream point of view could be affected independently in case of an issue or misconfiguration:



TrackMe - Data tracking system

Monitoring your Splunk data availability made easy

Click on any entry of the table to open interactive actions for this entity, use inputs to filter out the selection: (restricted to the first thousand results, us

Keyword filter mode: Includes

Keyword filter name: *

Tags: ALL ✕

Name (use keyword filter): ALL ✕

Index: ALL ✕

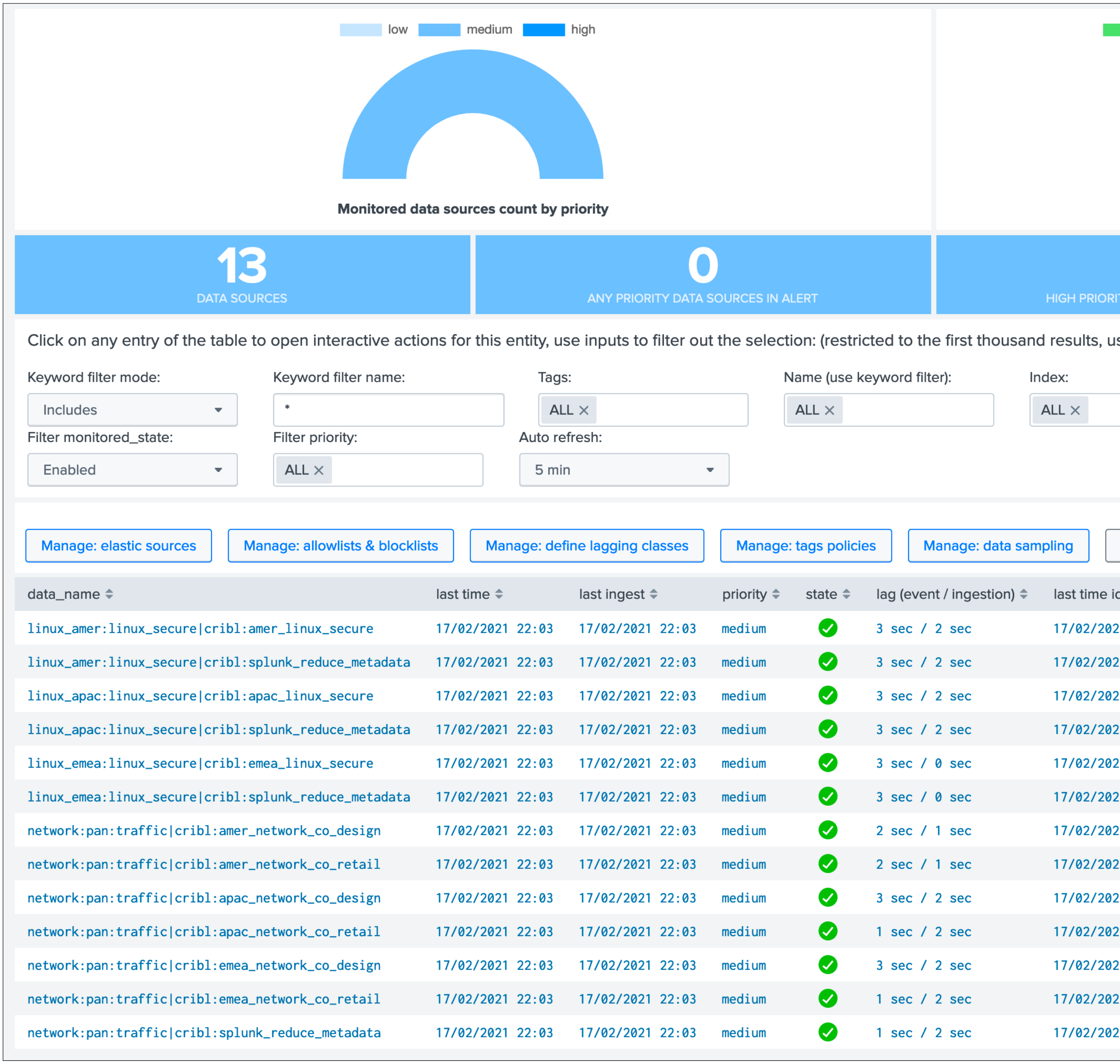
Filter monitored_state: Enabled

Filter priority: ALL ✕

Auto refresh: 5 min

data_name	last time	last ingest	priority	state	lag (event / ingestion)	last time idx	data_last_lag_seen
linux_amer:linux_secure	17/02/2021 21:54	17/02/2021 21:55	medium	✓	2 sec / 2 sec	17/02/2021 21:54	
linux_apac:linux_secure	17/02/2021 21:55	17/02/2021 21:55	medium	✓	-7 sec / 0 sec	17/02/2021 21:55	
linux_emea:linux_secure	17/02/2021 21:54	17/02/2021 21:54	medium	✓	2 sec / 0 sec	17/02/2021 21:54	
network:pan:traffic	17/02/2021 21:54	17/02/2021 21:55	medium	✓	2 sec / 3 sec	17/02/2021 21:54	

Once we enable the Cribl mode, we see a very different picture, TrackMe automatically creates data sources broken by index, sourcetype and cribl_pipe:



13
DATA SOURCES

0
ANY PRIORITY DATA SOURCES IN ALERT

HIGH PRIORI

Click on any entry of the table to open interactive actions for this entity, use inputs to filter out the selection: (restricted to the first thousand results, us

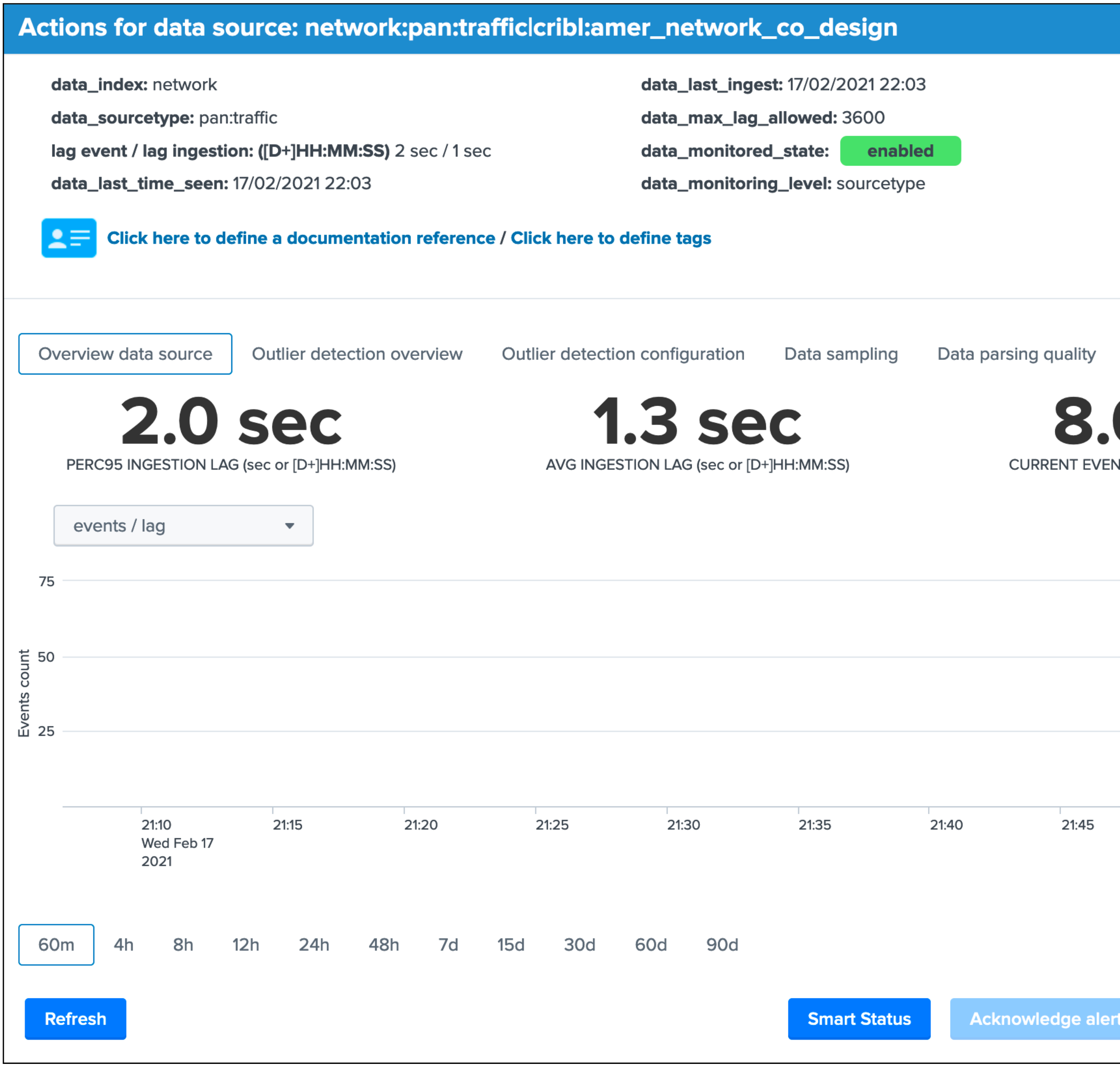
Keyword filter mode: Includes
Keyword filter name: *
Tags: ALL ×
Name (use keyword filter): ALL ×
Index: ALL ×
Filter monitored_state: Enabled
Filter priority: ALL ×
Auto refresh: 5 min

Manage: elastic sources | Manage: allowlists & blocklists | Manage: define lagging classes | Manage: tags policies | Manage: data sampling

data_name	last time	last ingest	priority	state	lag (event / ingestion)	last time ic
linux_amer:linux_secure|cribl:amer_linux_secure	17/02/2021 22:03	17/02/2021 22:03	medium	✓	3 sec / 2 sec	17/02/202
linux_amer:linux_secure|cribl:splunk_reduce_metadata	17/02/2021 22:03	17/02/2021 22:03	medium	✓	3 sec / 2 sec	17/02/202
linux_apac:linux_secure|cribl:apac_linux_secure	17/02/2021 22:03	17/02/2021 22:03	medium	✓	3 sec / 2 sec	17/02/202
linux_apac:linux_secure|cribl:splunk_reduce_metadata	17/02/2021 22:03	17/02/2021 22:03	medium	✓	3 sec / 2 sec	17/02/202
linux_emea:linux_secure|cribl:emea_linux_secure	17/02/2021 22:03	17/02/2021 22:03	medium	✓	3 sec / 0 sec	17/02/202
linux_emea:linux_secure|cribl:splunk_reduce_metadata	17/02/2021 22:03	17/02/2021 22:03	medium	✓	3 sec / 0 sec	17/02/202
network:pan:traffic|cribl:amer_network_co_design	17/02/2021 22:03	17/02/2021 22:03	medium	✓	2 sec / 1 sec	17/02/202
network:pan:traffic|cribl:amer_network_co_retail	17/02/2021 22:03	17/02/2021 22:03	medium	✓	2 sec / 1 sec	17/02/202
network:pan:traffic|cribl:apac_network_co_design	17/02/2021 22:03	17/02/2021 22:03	medium	✓	3 sec / 2 sec	17/02/202
network:pan:traffic|cribl:apac_network_co_retail	17/02/2021 22:03	17/02/2021 22:03	medium	✓	1 sec / 2 sec	17/02/202
network:pan:traffic|cribl:emea_network_co_design	17/02/2021 22:03	17/02/2021 22:03	medium	✓	3 sec / 2 sec	17/02/202
network:pan:traffic|cribl:emea_network_co_retail	17/02/2021 22:03	17/02/2021 22:03	medium	✓	1 sec / 2 sec	17/02/202
network:pan:traffic|cribl:splunk_reduce_metadata	17/02/2021 22:03	17/02/2021 22:03	medium	✓	1 sec / 2 sec	17/02/202

Data sources are created as `index + ":" + sourcetype + ":" + cribl_pipe`, this represents the data flow from Cribl LogStream to Splunk.

Every search actioned by trackMe now automatically recycles the cribl_pipe information naturally, such as latency tracking, data sampling, open in search buttons, etc:



Actions for data source: network:pan:traffic|cribl:amer_network_co_design

data_index: network

data_sourcetype: pan:traffic

lag event / lag ingestion: ([D+]HH:MM:SS) 2 sec / 1 sec

data_last_time_seen: 17/02/2021 22:03

data_last_ingest: 17/02/2021 22:03

data_max_lag_allowed: 3600

data_monitored_state: enabled

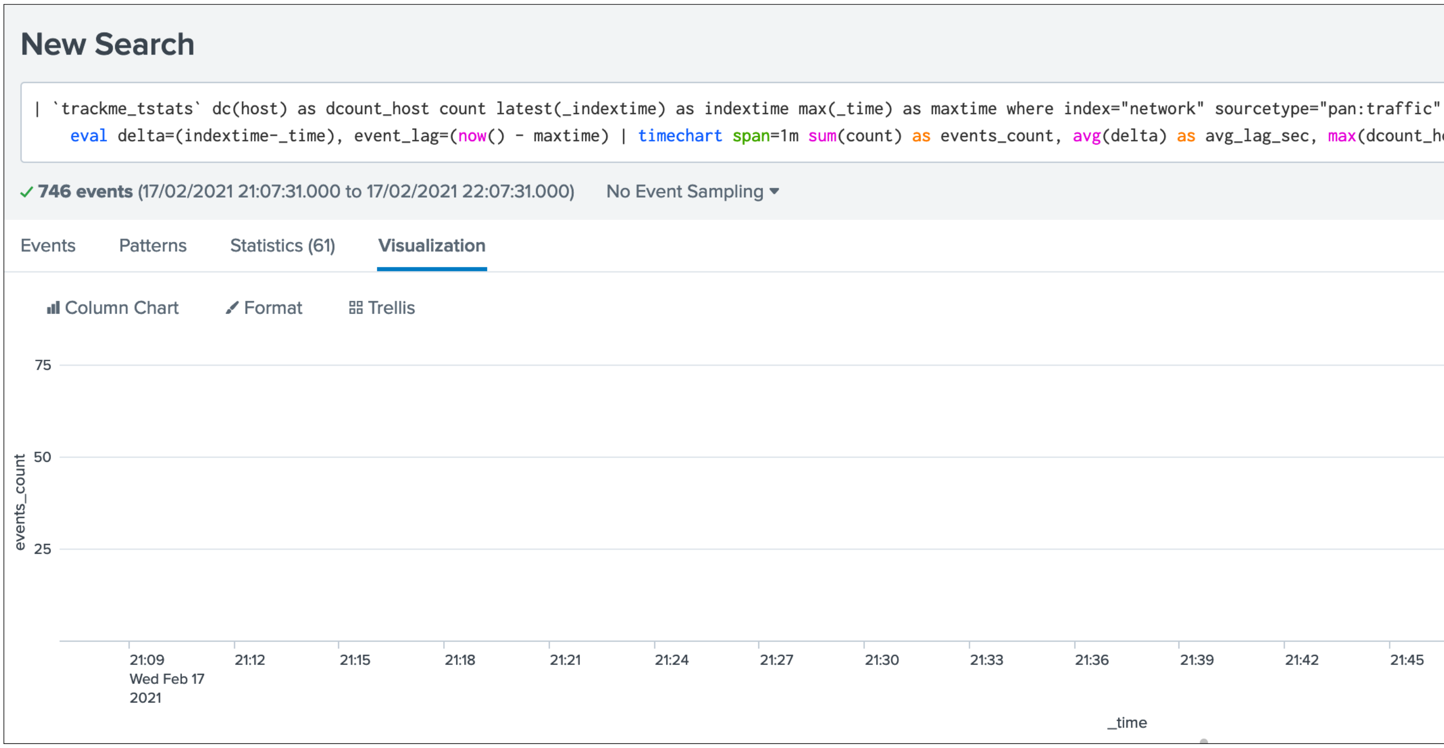
data_monitoring_level: sourcetype

Click here to define a documentation reference / Click here to define tags

2.0 sec
PERC95 INGESTION LAG (sec or [D+]HH:MM:SS)

1.3 sec
AVG INGESTION LAG (sec or [D+]HH:MM:SS)

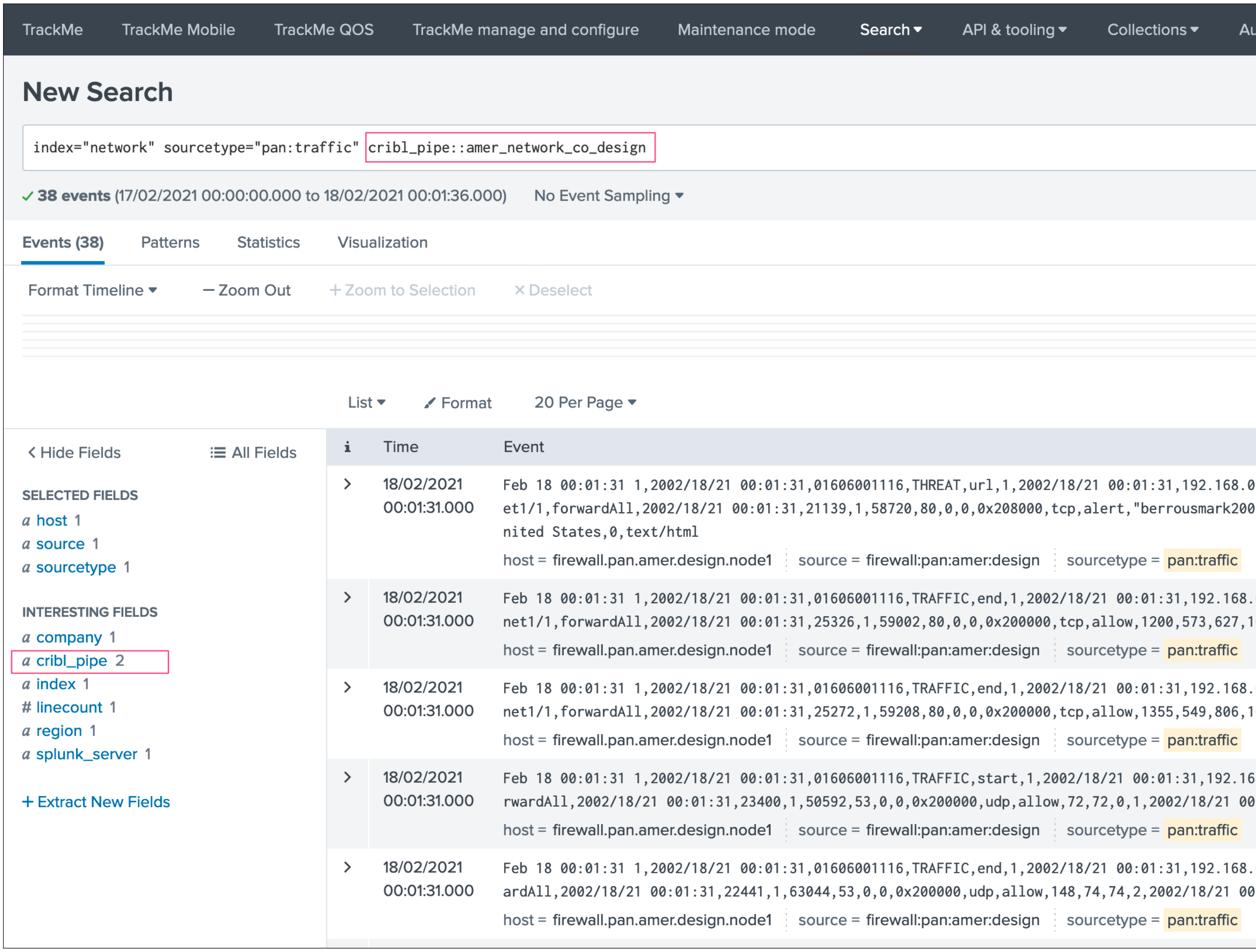
8.(
CURRENT EVEN



New Search

```
| `trackme_tstats` dc(host) as dcount_host count latest(_indextime) as indextime max(_time) as maxtime where index="network" sourcetype="pan:traffic"
   eval delta=(indextime-_time), event_lag=(now() - maxtime) | timechart span=1m sum(count) as events_count, avg(delta) as avg_lag_sec, max(dcount_ho
```

✓ **746 events** (17/02/2021 21:07:31.000 to 17/02/2021 22:07:31.000)



```
index="network" sourcetype="pan:traffic" cribl_pipe::amer_network_co_design
```

✓ 38 events (17/02/2021 00:00:00.000 to 18/02/2021 00:01:36.000)

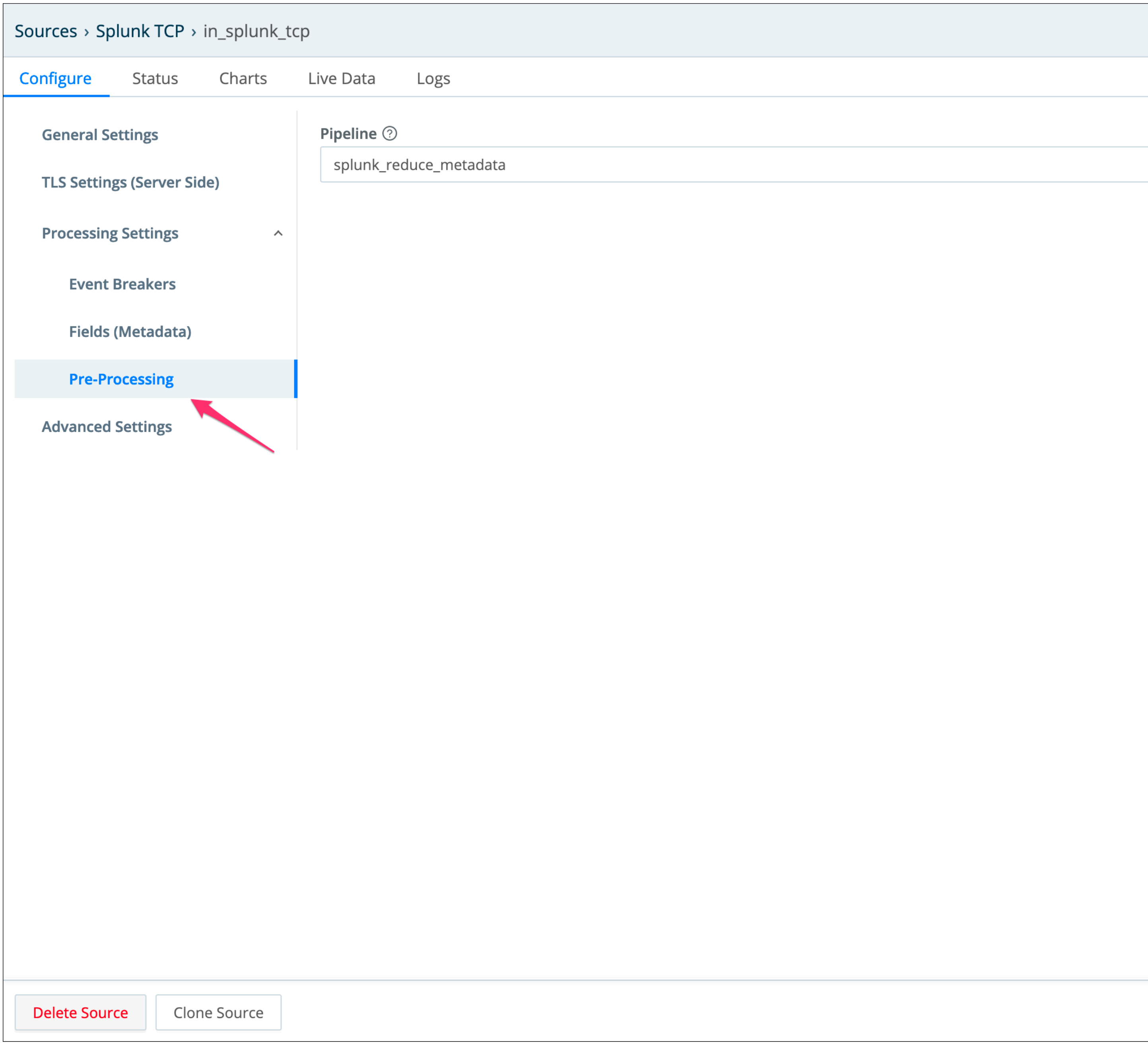
SELECTED FIELDS
- a host 1
- a source 1
- a sourcetype 1

INTERESTING FIELDS
- a company 1
- a cribl_pipe 2
- a index 1
- # linecount 1
- a region 1
- a splunk_server 1

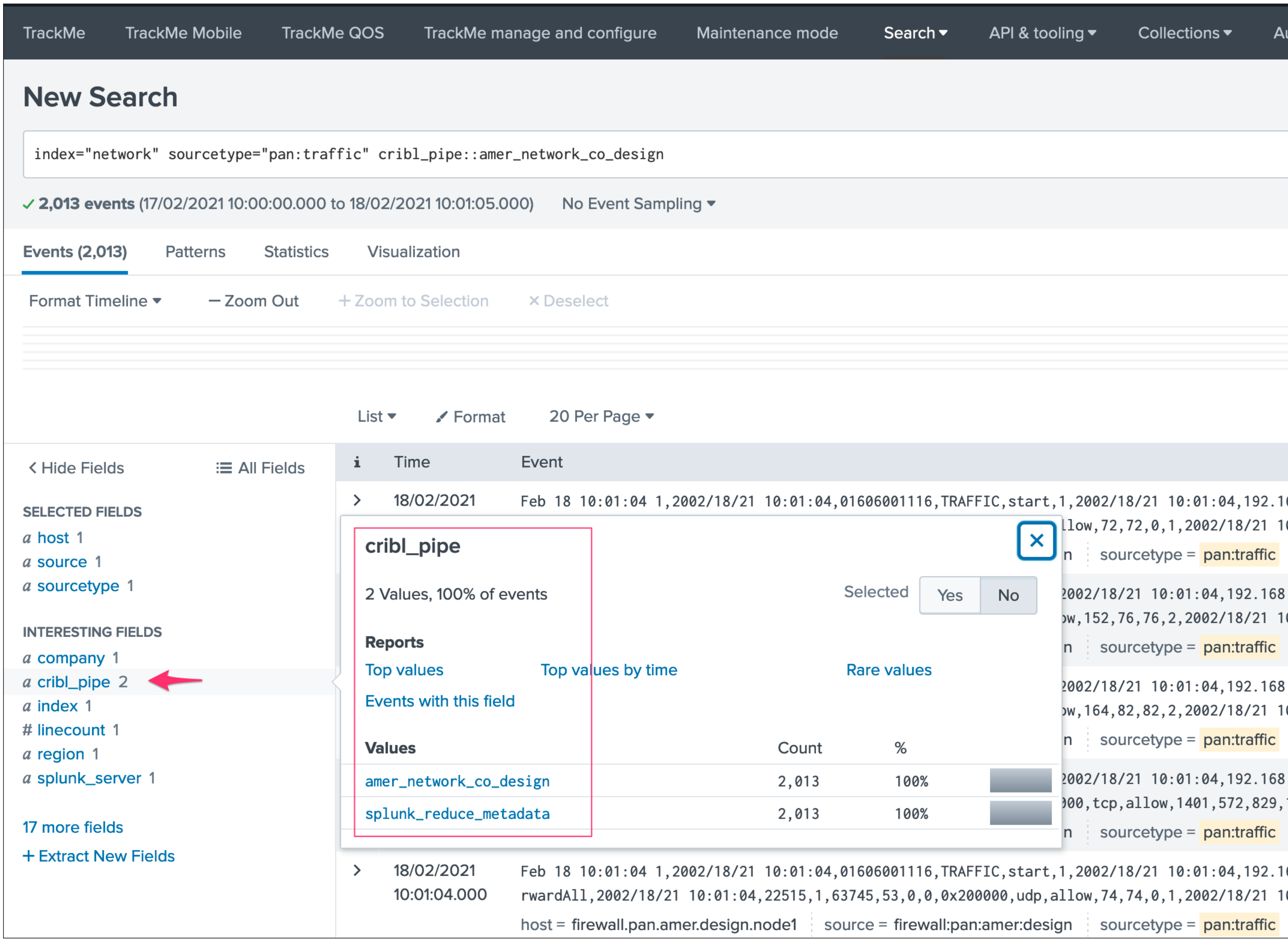
i	Time	Event
>	18/02/2021 00:01:31.000	Feb 18 00:01:31 1,2002/18/21 00:01:31,01606001116,THREAT,url,1,2002/18/21 00:01:31,192.168.0 et1/1,forwardAll,2002/18/21 00:01:31,21139,1,58720,80,0,0,0x208000,tcp,alert,"berrousmark200 nited States,0,text/html host = firewall.pan.amer.design.node1 source = firewall:pan:amer:design sourcetype = pan:traffic
>	18/02/2021 00:01:31.000	Feb 18 00:01:31 1,2002/18/21 00:01:31,01606001116,TRAFFIC,end,1,2002/18/21 00:01:31,192.168. net1/1,forwardAll,2002/18/21 00:01:31,25326,1,59002,80,0,0,0x200000,tcp,allow,1200,573,627,1 host = firewall.pan.amer.design.node1 source = firewall:pan:amer:design sourcetype = pan:traffic
>	18/02/2021 00:01:31.000	Feb 18 00:01:31 1,2002/18/21 00:01:31,01606001116,TRAFFIC,end,1,2002/18/21 00:01:31,192.168. net1/1,forwardAll,2002/18/21 00:01:31,25272,1,59208,80,0,0,0x200000,tcp,allow,1355,549,806,1 host = firewall.pan.amer.design.node1 source = firewall:pan:amer:design sourcetype = pan:traffic
>	18/02/2021 00:01:31.000	Feb 18 00:01:31 1,2002/18/21 00:01:31,01606001116,TRAFFIC,start,1,2002/18/21 00:01:31,192.16 rwardAll,2002/18/21 00:01:31,23400,1,50592,53,0,0,0x200000,udp,allow,72,72,0,1,2002/18/21 00 host = firewall.pan.amer.design.node1 source = firewall:pan:amer:design sourcetype = pan:traffic
>	18/02/2021 00:01:31.000	Feb 18 00:01:31 1,2002/18/21 00:01:31,01606001116,TRAFFIC,end,1,2002/18/21 00:01:31,192.168. ardAll,2002/18/21 00:01:31,22441,1,63044,53,0,0,0x200000,udp,allow,148,74,74,2,2002/18/21 00 host = firewall.pan.amer.design.node1 source = firewall:pan:amer:design sourcetype = pan:traffic

3.3.3 Cribl LogStream pre-processing pipelines and cribl_pipe field

If you have a proprocessing pipelines in your LogStream workflow, the `cribl_pipe` field becomes a multi-value indexed field that contains both the processing pipeline and pre-processing pipeline:



Pipeline: splunk_reduce_metadata



New Search

```
index="network" sourcetype="pan:traffic" cribl_pipe::amer_network_co_design
```

✓ 2,013 events (17/02/2021 10:00:00.000 to 18/02/2021 10:01:05.000)   No Event Sampling ▾

SELECTED FIELDS
- a host 1
- a source 1
- a sourcetype 1

INTERESTING FIELDS
- a company 1
- a cribl_pipe 2
- a index 1
- # linecount 1
- a region 1
- a splunk_server 1

17 more fields
+ Extract New Fields

cribl_pipe

2 Values, 100% of events

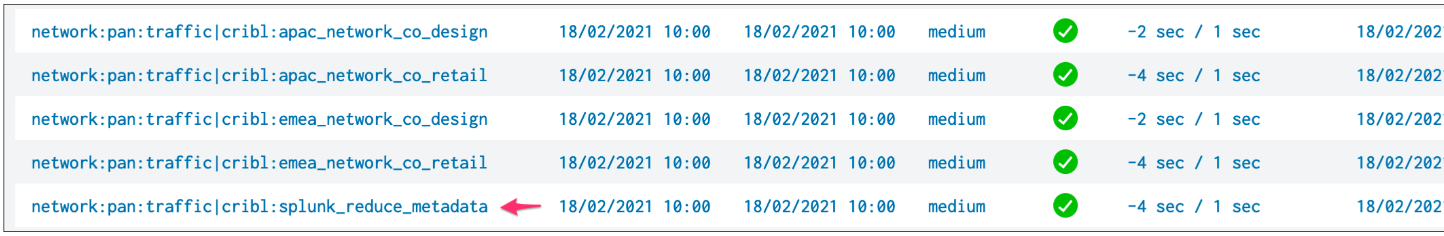
Selected: Yes No

Reports: Top values · Top values by time · Rare values · Events with this field

Values	Count	%
amer_network_co_design	2,013	100%
splunk_reduce_metadata	2,013	100%

i	Time	Event
>	18/02/2021	Feb 18 10:01:04 1,2002/18/21 10:01:04,01606001116,TRAFFIC,start,1,2002/18/21 10:01:04,192.1
>	18/02/2021 10:01:04.000	Feb 18 10:01:04 1,2002/18/21 10:01:04,01606001116,TRAFFIC,start,1,2002/18/21 10:01:04,192.1 rwardAll,2002/18/21 10:01:04,22515,1,63745,53,0,0,0x200000,udp,allow,74,74,0,1,2002/18/21 1 host = firewall.pan.amer.design.node1 source = firewall:pan:amer:design sourcetype = pan:traffic

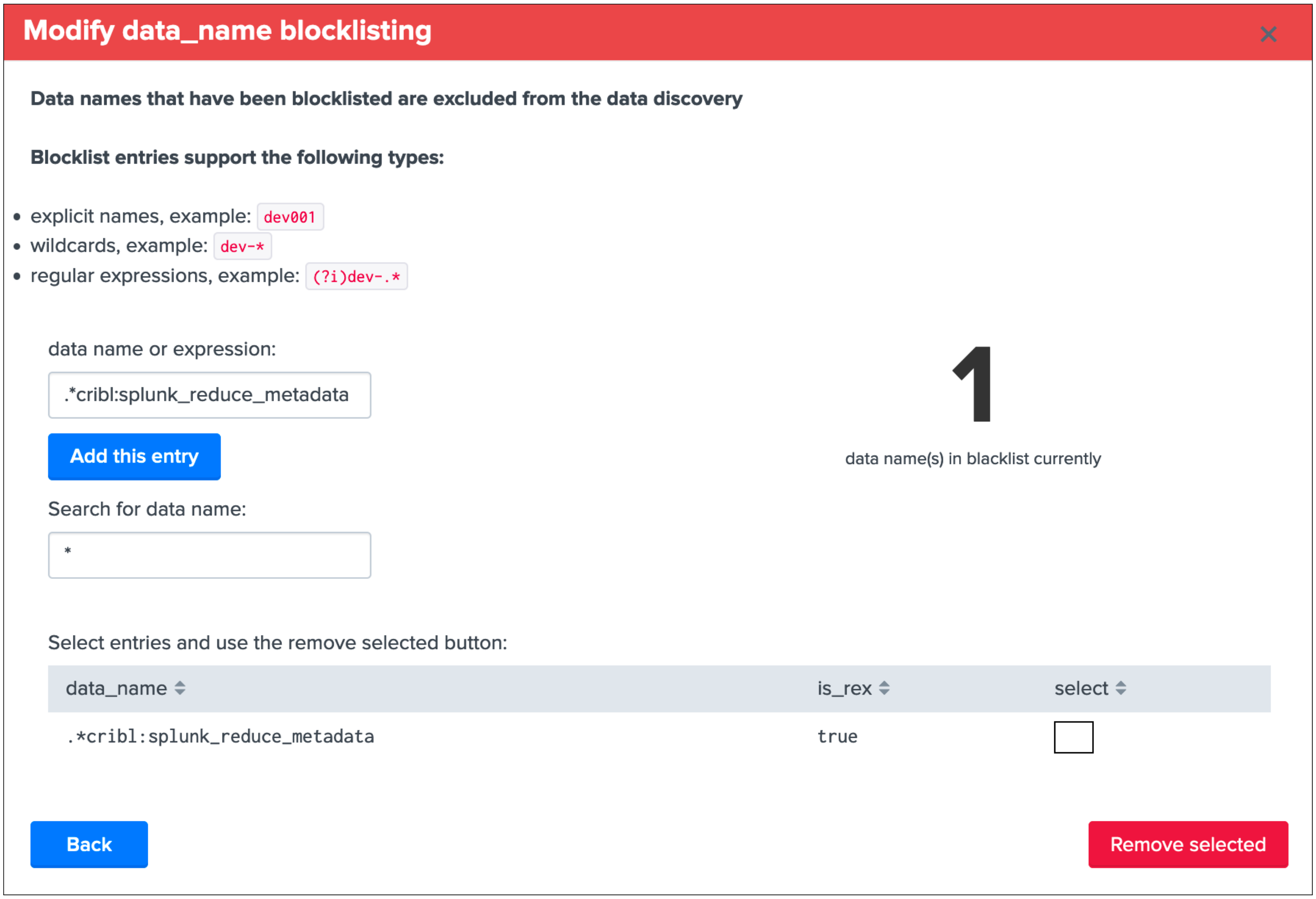
In the TrackMe context, this means that for the same data source, we get at least two entities, one per pipeline and one for the pre-processing pipeline:



network:pan:traffic|cribl:apac_network_co_design	18/02/2021 10:00	18/02/2021 10:00	medium	✓	-2 sec / 1 sec	18/02/202
network:pan:traffic|cribl:apac_network_co_retail	18/02/2021 10:00	18/02/2021 10:00	medium	✓	-4 sec / 1 sec	18/02/202
network:pan:traffic|cribl:emea_network_co_design	18/02/2021 10:00	18/02/2021 10:00	medium	✓	-2 sec / 1 sec	18/02/202
network:pan:traffic|cribl:emea_network_co_retail	18/02/2021 10:00	18/02/2021 10:00	medium	✓	-4 sec / 1 sec	18/02/202
network:pan:traffic|cribl:splunk_reduce_metadata	18/02/2021 10:00	18/02/2021 10:00	medium	✓	-4 sec / 1 sec	18/02/202

From the TrackMe point of view, the pre-processing pipeline view has no value and all that we care about is the data flow itself, to get rid of these entities automatically, we can add a data_name blocklist based in a very simple regular expression:

- from the main TrackMe screen, go to "Manage: allowlists & blocklists"
- add a new data_name blocklist according to the name of your pre-processing pipeline, in our case we will use `.*cribl:splunk_reduce_metadata`
- once it has been added, existing entities are not taken into account anymore, and if new data sources are discovered, these will exclude the pre-processing pipeline automatically



Modify data_name blocklisting

Data names that have been blocklisted are excluded from the data discovery

Blocklist entries support the following types:

- explicit names, example: `dev001`
- wildcards, example: `dev-*`
- regular expressions, example: `(?i)dev-.*`

data name or expression: .*cribl:splunk_reduce_metadata

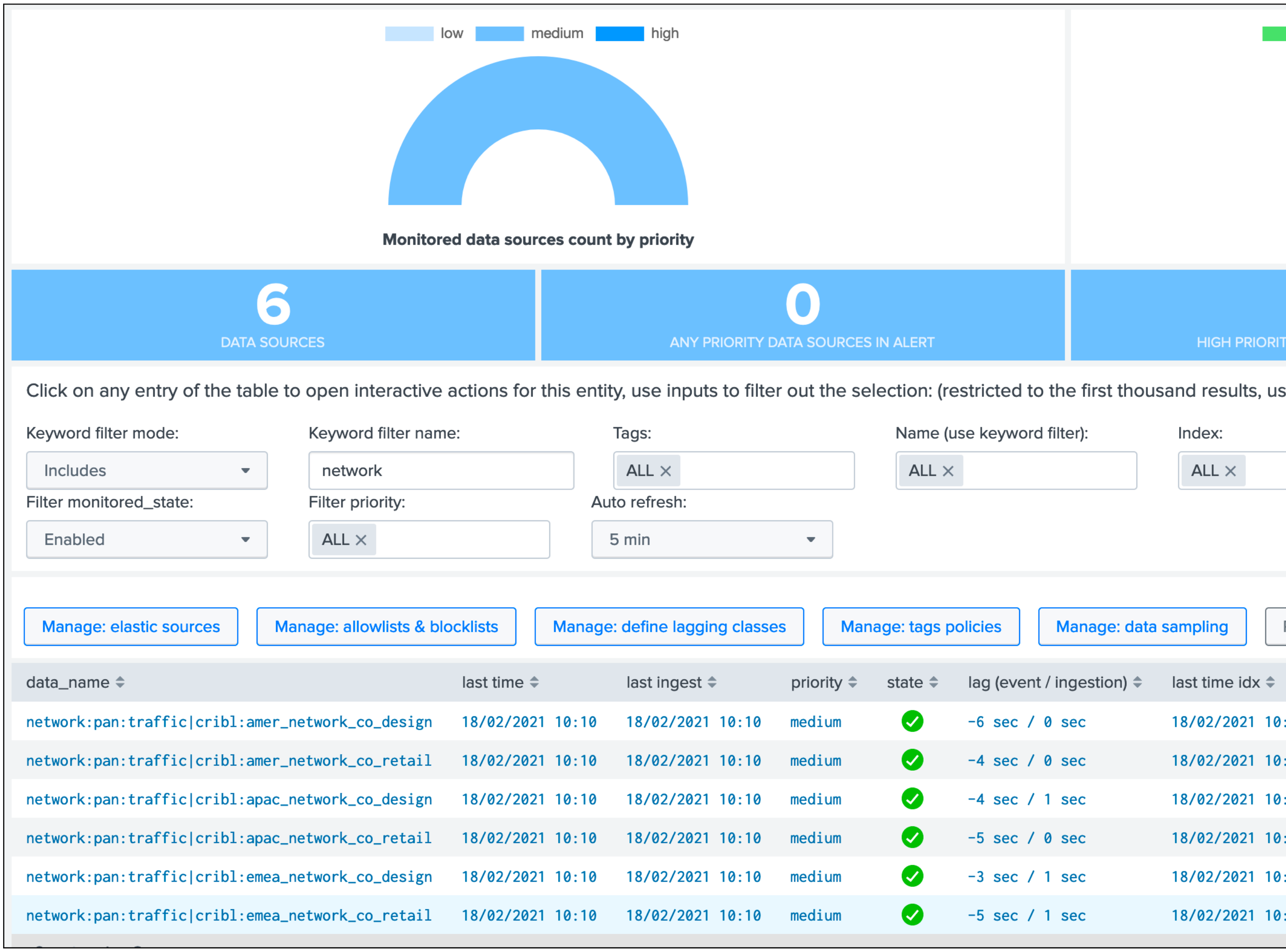
1

data name(s) in blacklist currently

Search for data name: *

Select entries and use the remove selected button:

data_name	is_rex	select
.*cribl:splunk_reduce_metadata	true	□


low
medium
high
Monitored data sources count by priority
6
DATA SOURCES
0
ANY PRIORITY DATA SOURCES IN ALERT
HIGH PRIORIT
Manage: elastic sources
Manage: allowlists & blocklists
Manage: define lagging classes
Manage: tags policies
Manage: data sampling

Click on any entry of the table to open interactive actions for this entity, use inputs to filter out the selection: (restricted to the first thousand results, us

Keyword filter mode: Includes

Keyword filter name: network

Tags: ALL ×

Name (use keyword filter): ALL ×

Index: ALL ×

Filter monitored_state: Enabled

Filter priority: ALL ×

Auto refresh: 5 min

data_name	last time	last ingest	priority	state	lag (event / ingestion)	last time idx
network:pan:traffic|cribl:amer_network_co_design	18/02/2021 10:10	18/02/2021 10:10	medium	✓	-6 sec / 0 sec	18/02/2021 10:
network:pan:traffic|cribl:amer_network_co_retail	18/02/2021 10:10	18/02/2021 10:10	medium	✓	-4 sec / 0 sec	18/02/2021 10:
network:pan:traffic|cribl:apac_network_co_design	18/02/2021 10:10	18/02/2021 10:10	medium	✓	-4 sec / 1 sec	18/02/2021 10:
network:pan:traffic|cribl:apac_network_co_retail	18/02/2021 10:10	18/02/2021 10:10	medium	✓	-5 sec / 0 sec	18/02/2021 10:
network:pan:traffic|cribl:emea_network_co_design	18/02/2021 10:10	18/02/2021 10:10	medium	✓	-3 sec / 1 sec	18/02/2021 10:
network:pan:traffic|cribl:emea_network_co_retail	18/02/2021 10:10	18/02/2021 10:10	medium	✓	-5 sec / 1 sec	18/02/2021 10:

Congratulations!

You have a now a comprehensive integration between the wonderful and amazing Cribl LogStream and TrackMe allowing you to track your Splunk data the easy way!

3.3.4 Handling both Cribl mode and regular mode

In some deployments, you may have both Cribl Logstream feeding Splunk, and regular other types of data coming from Universal or Heavy Forwarders.

When TrackMe is configured in the Cribl mode, only data coming from Cribl Logstream will be taken into account, which happens because we expect a `cribl_pipe` indexed field for every data source to be discovered and maintained.

However, with some minor manual steps, you can easily work in hybrid mode and have data sources handled automatically for both Cribl Logstream originating data, and regular data indexed directly.

This process is currently manual, which might be improved in a future release of TrackMe.

Clone Data sources trackers

First, clone the abstract report named TrackMe - Data sources abstract root tracker, example: TrackMe - Data sources abstract root tracker - Not Cribl

This report is used by both the short term and long term trackers to reduce the amount of duplicated SPL lines of codes, you can achieve the clone via Splunk Web (Settings, Searches, reports and alerts), or manually via configuration files if you prefer.

Next, clone the scheduled report named TrackMe - Data sources availability short term tracker, example: TrackMe - Data sources availability short term tracker - Not Cribl

Finally, clone the scheduled report named TrackMe - Data sources availability long term tracker, example: TrackMe - Data sources availability short term tracker - Not Cribl

Update Data sources abstract report

Edit the newly created abstract report TrackMe - Data sources abstract root tracker - Not Cribl:

1 - Update the split by condition

Locate in the first line:

```
by `trackme_data_source_tstats_root_splitby`
```

And replace with:

```
by `trackme_data_source_tstats_root_splitby_regular`
```

2 - Update the where statement

Still in the first line, locate:

```
where index=* sourcetype=*
```

And replace with:

```
where index=* sourcetype=* cribl_pipe!=*
```

Note: this will ensure these trackers only care about data not originating from Cribl Logstream.

3 - Update the intermediate calculation

locate in the report the line:

```
| `trackme_default_data_source_mode`
```

And replace with:

```
| `trackme_data_source_split_mode`
```

Update Data sources short and long term trackers

For both the short term and long term trackers newly created, edit the report and:

Locate the first line:

```
| savedsearch "TrackMe - Data sources abstract root tracker"
```

And replace with:

```
| savedsearch "TrackMe - Data sources abstract root tracker - Not Cribl"
```

And that's it, after the first executions of the newly created tracker reports, any data source that is not coming from Cribl Logstream will be discovered and maintained as usual.

You can immediately run the short term tracker to get regular data sources added to TrackMe.

Note that the "run trackers" buttons in the TrackMe UI will only handle the main default trackers, which is a minor loss of features as you do not normally need to actively execute the trackers.

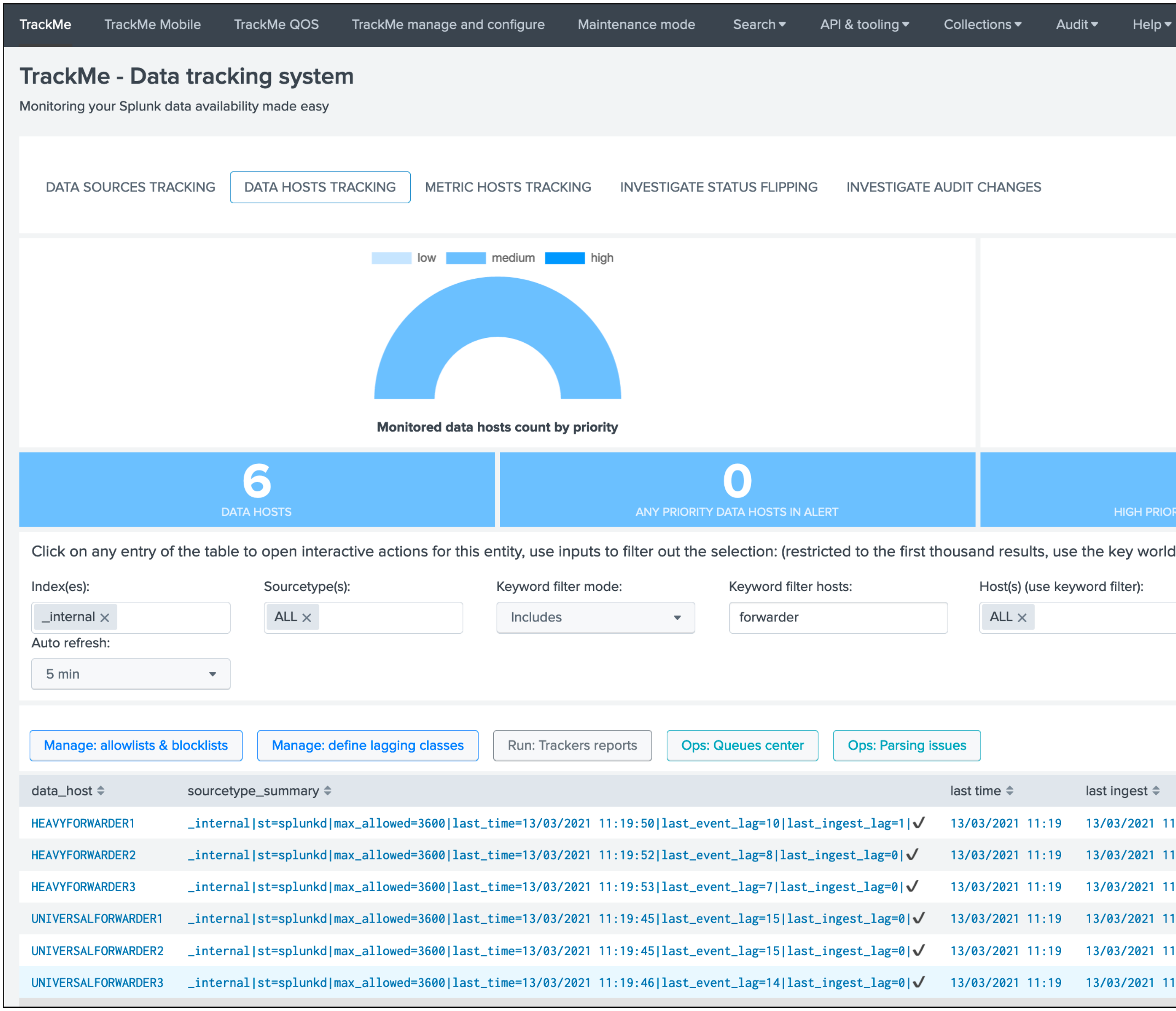
3.4 Monitor Splunk instances forwarding

TrackMe monitors by default any Splunk instance forwarding to the Splunk indexing layer, this includes:

- Universal Forwarder instances
- Heavy Forwarder instances
- All other types of instances from your Splunk infrastructure

Forwarding is monitored via:

- `data hosts` by tracking the `index=_internal sourcetype=splunkd`
- `metric hosts` by trackking the `spl` metrics stored in the `_metrics` index



TrackMe - Data tracking system

Monitoring your Splunk data availability made easy

Click on any entry of the table to open interactive actions for this entity, use inputs to filter out the selection: (restricted to the first thousand results, use the key world

Index(es): _internal ×

Sourcetype(s): ALL ×

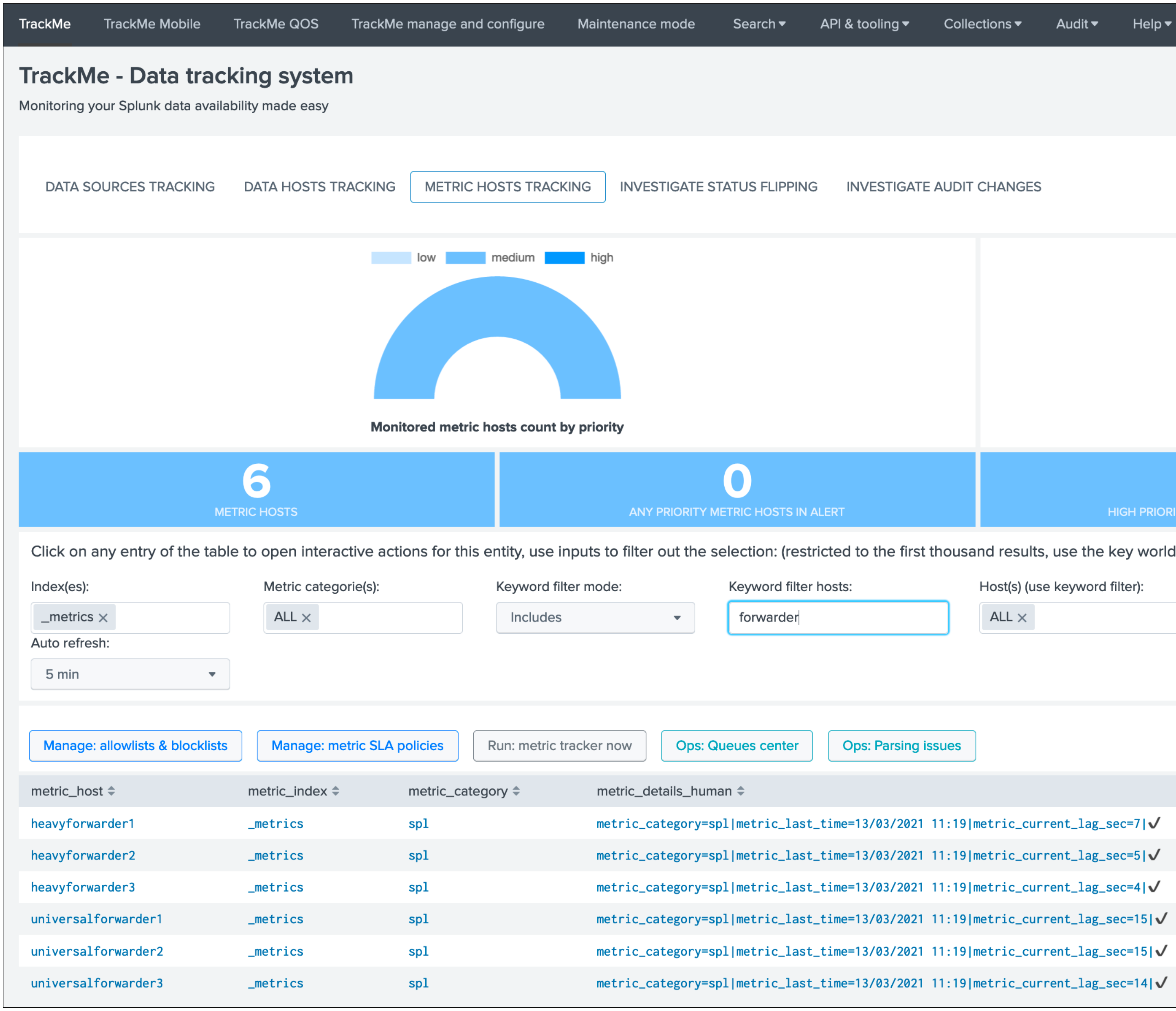
Keyword filter mode: Includes

Keyword filter hosts: forwarder

Host(s) (use keyword filter): ALL ×

Auto refresh: 5 min

data_host	sourcetype_summary	last time	last ingest
HEAVYFORWARDER1	_internal|st=splunkd|max_allowed=3600|last_time=13/03/2021 11:19:50|last_event_lag=10|last_ingest_lag=1|✓	13/03/2021 11:19	13/03/2021 11
HEAVYFORWARDER2	_internal|st=splunkd|max_allowed=3600|last_time=13/03/2021 11:19:52|last_event_lag=8|last_ingest_lag=0|✓	13/03/2021 11:19	13/03/2021 11
HEAVYFORWARDER3	_internal|st=splunkd|max_allowed=3600|last_time=13/03/2021 11:19:53|last_event_lag=7|last_ingest_lag=0|✓	13/03/2021 11:19	13/03/2021 11
UNIVERSALFORWARDER1	_internal|st=splunkd|max_allowed=3600|last_time=13/03/2021 11:19:45|last_event_lag=15|last_ingest_lag=0|✓	13/03/2021 11:19	13/03/2021 11
UNIVERSALFORWARDER2	_internal|st=splunkd|max_allowed=3600|last_time=13/03/2021 11:19:45|last_event_lag=15|last_ingest_lag=0|✓	13/03/2021 11:19	13/03/2021 11
UNIVERSALFORWARDER3	_internal|st=splunkd|max_allowed=3600|last_time=13/03/2021 11:19:46|last_event_lag=14|last_ingest_lag=0|✓	13/03/2021 11:19	13/03/2021 11



TrackMe - Data tracking system

Monitoring your Splunk data availability made easy

Click on any entry of the table to open interactive actions for this entity, use inputs to filter out the selection: (restricted to the first thousand results, use the key world

Index(es): _metrics ×

Metric categorie(s): ALL ×

Keyword filter mode: Includes

Keyword filter hosts: forwarder

Host(s) (use keyword filter): ALL ×

Auto refresh: 5 min

metric_host ⬍	metric_index ⬍	metric_category ⬍	metric_details_human ⬍
heavyforwarder1	_metrics	spl	metric_category=spl|metric_last_time=13/03/2021 11:19|metric_current_lag_sec=7|✓
heavyforwarder2	_metrics	spl	metric_category=spl|metric_last_time=13/03/2021 11:19|metric_current_lag_sec=5|✓
heavyforwarder3	_metrics	spl	metric_category=spl|metric_last_time=13/03/2021 11:19|metric_current_lag_sec=4|✓
universalforwarder1	_metrics	spl	metric_category=spl|metric_last_time=13/03/2021 11:19|metric_current_lag_sec=15|✓
universalforwarder2	_metrics	spl	metric_category=spl|metric_last_time=13/03/2021 11:19|metric_current_lag_sec=15|✓
universalforwarder3	_metrics	spl	metric_category=spl|metric_last_time=13/03/2021 11:19|metric_current_lag_sec=14|✓

3.4.1 Requirements

Splunk forwarding good practices configuration

Splunk good configuration practices implies that you are systematically forwarding the Splunk internals (and metrics) to the indexing layer in your outputs.conf configuration, see the Splunk PS base config apps:

- Configurations Base Apps
- Configurations Cluster Apps

See: org_all_forwarder_outputs / org_cluster_forwarder_outputs

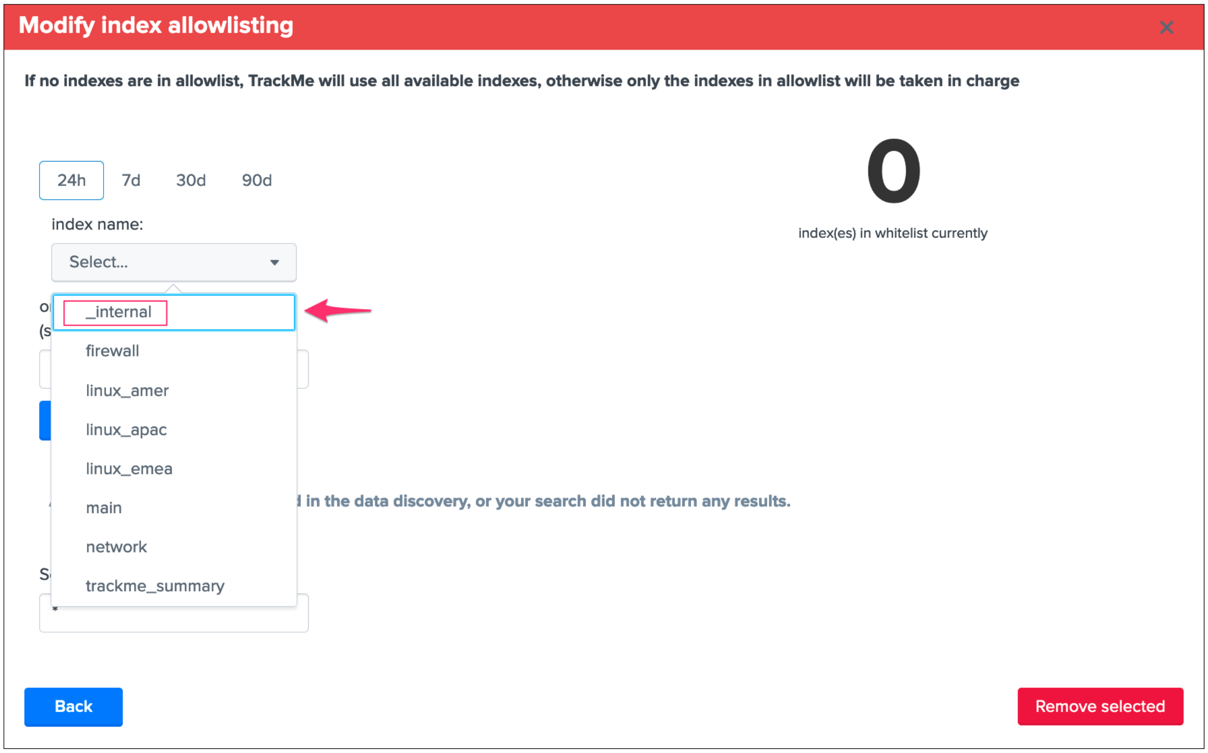
Concretely, this implies that you configure the Splunk instances (all but indexers) to have an outputs.conf sanza similar to:

```
[tcpout]
defaultGroup = primary_indexers

forwardedindex.2.whitelist = (_audit|_introspection|_internal|_metrics)
```

TrackMe allow lists and block lists

The default configuration of TrackMe implies monitoring every single index including the _internal (limited to sourcetype=splunkd) and the _metrics, if you use allowlisting & blocklisting, you need to make sure to include these items accordingly:

For data hosts, allow the _internal



Modify index allowlisting

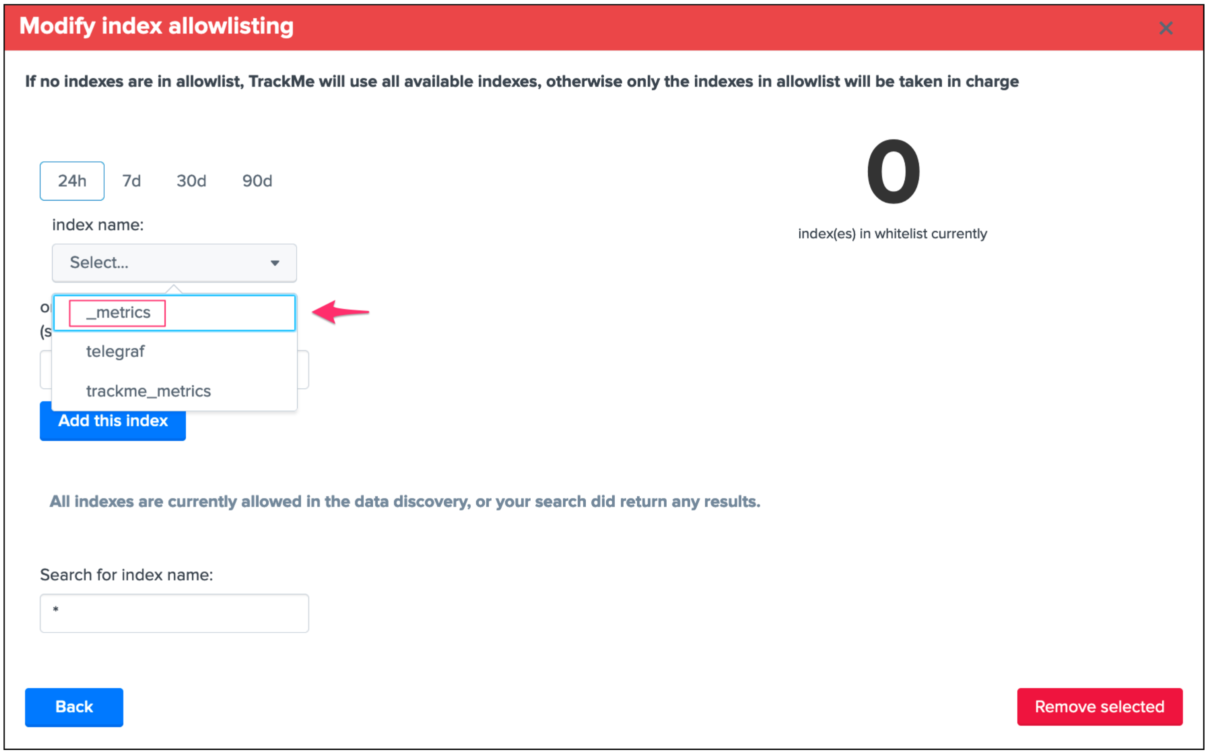
If no indexes are in allowlist, TrackMe will use all available indexes, otherwise only the indexes in allowlist will be taken in charge

0

index(es) in whitelist currently

in the data discovery, or your search did not return any results.

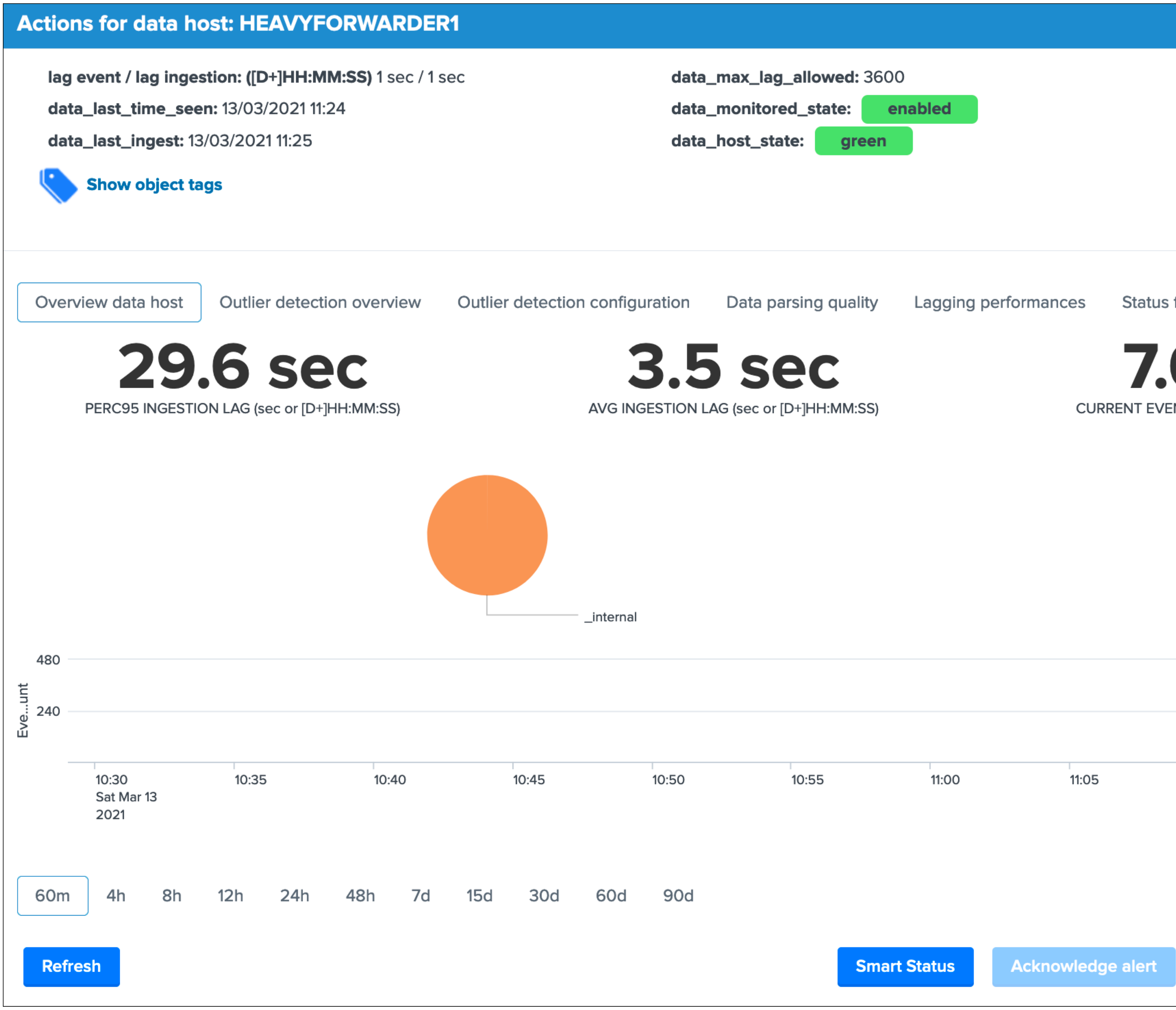
For metric hosts, allow the _metrics



Modify index allowlisting
×
If no indexes are in allowlist, TrackMe will use all available indexes, otherwise only the indexes in allowlist will be taken in charge
24h
7d
30d
90d
0
index(es) in whitelist currently
index name:
Select...
o
(s
_metrics
telegraf
trackme_metrics
Add this index
All indexes are currently allowed in the data discovery, or your search did return any results.
Search for index name:
*
Back
Remove selected

3.4.2 Usage

A Splunk instance that does not generate any data out of the internal will appear with the single _internal / splunkd combination in data hosts, and the spl metrics in metrics hosts:



Actions for data host: HEAVYFORWARDER1

lag event / lag ingestion: ([D+]HH:MM:SS) 1 sec / 1 sec

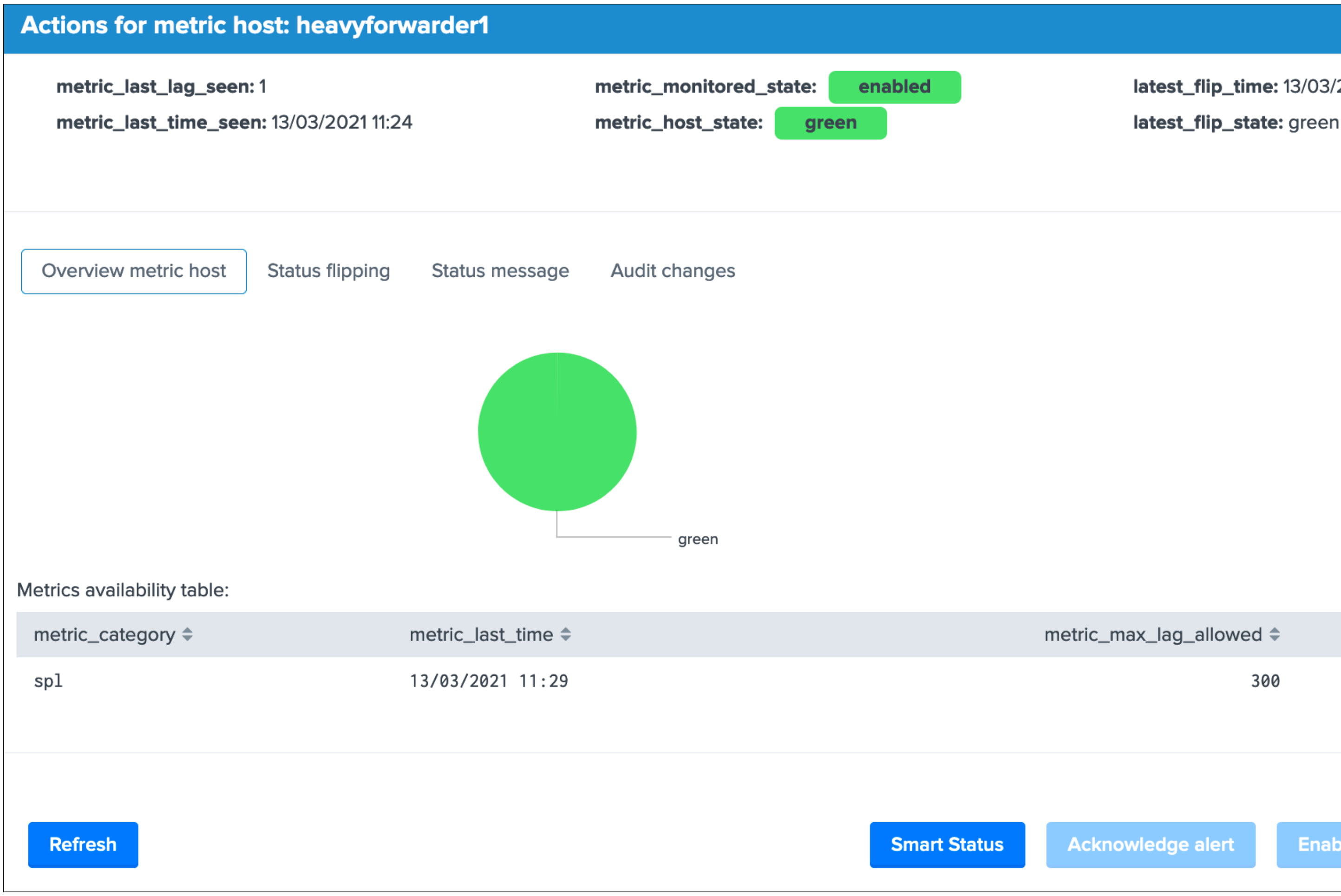
data_last_time_seen: 13/03/2021 11:24

data_last_ingest: 13/03/2021 11:25

data_max_lag_allowed: 3600

data_monitored_state: enabled

data_host_state: green



Actions for metric host: heavyforwarder1

metric_last_lag_seen: 1
metric_last_time_seen: 13/03/2021 11:24

metric_monitored_state: enabled
metric_host_state: green

latest_flip_time: 13/03/2
latest_flip_state: green

Metrics availability table:

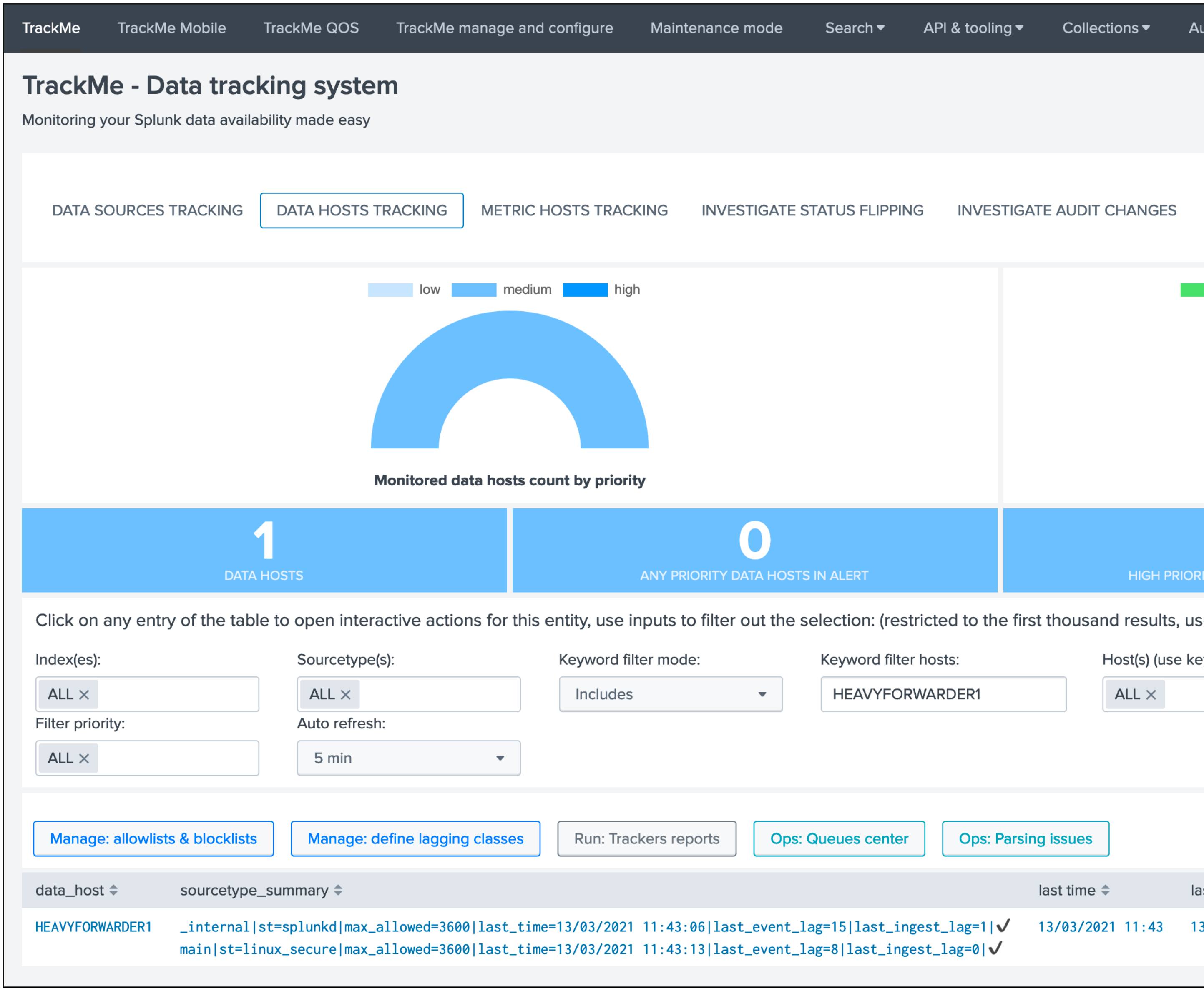
metric_category	metric_last_time	metric_max_lag_allowed
spl	13/03/2021 11:29	300

A green status basically indicates that:

- Splunk service is up and running
- The instance is able to reach the indexing layer and properly ingest data as it is forwarding effectively its own data and metrics (which validates configuration and network layers theoritically)
- The instance is expected to be acting in a normal and sane state

Data hosts tracking

When a Splunk instance does more than just indexing its own data and the host Metadata is refering to itself, the Splunk internal data and metrics appear as part of the indexing flow:



TrackMe - Data tracking system

Monitoring your Splunk data availability made easy

Click on any entry of the table to open interactive actions for this entity, use inputs to filter out the selection: (restricted to the first thousand results, us

Index(es): ALL ×

Sourcetype(s): ALL ×

Keyword filter mode: Includes

Keyword filter hosts: HEAVYFORWARDER1

Host(s) (use ke ALL ×

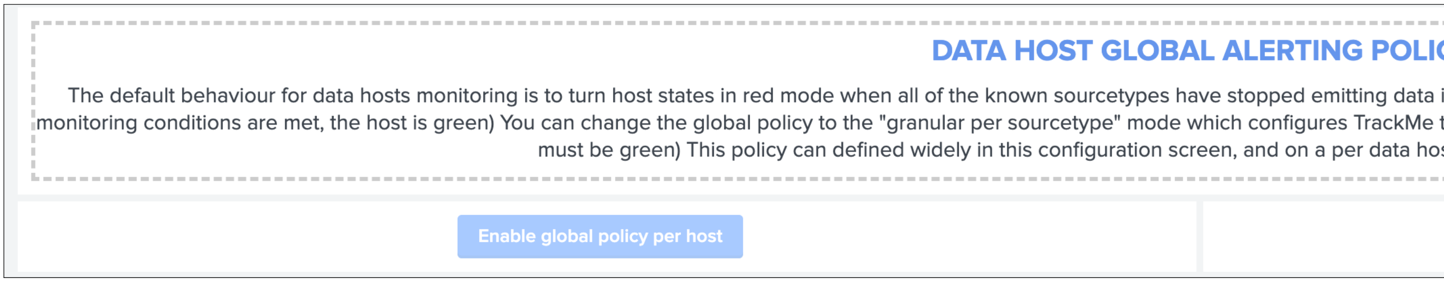
Filter priority: ALL ×

Auto refresh: 5 min

data_host	sourcetype_summary	last time	la
HEAVYFORWARDER1	_internal|st=splunkd|max_allowed=3600|last_time=13/03/2021 11:43:06|last_event_lag=15|last_ingest_lag=1|✓ main|st=linux_secure|max_allowed=3600|last_time=13/03/2021 11:43:13|last_event_lag=8|last_ingest_lag=0|✓	13/03/2021 11:43	13

The default behaviour driven by the global host policy implies that as long as the Splunk instance is forwarding data, the host will remain green even if the sources monitored by and as this host runs into troubles, you can on a global basis change the *Data Hosts alerting policy* or selectively on a per host basis:

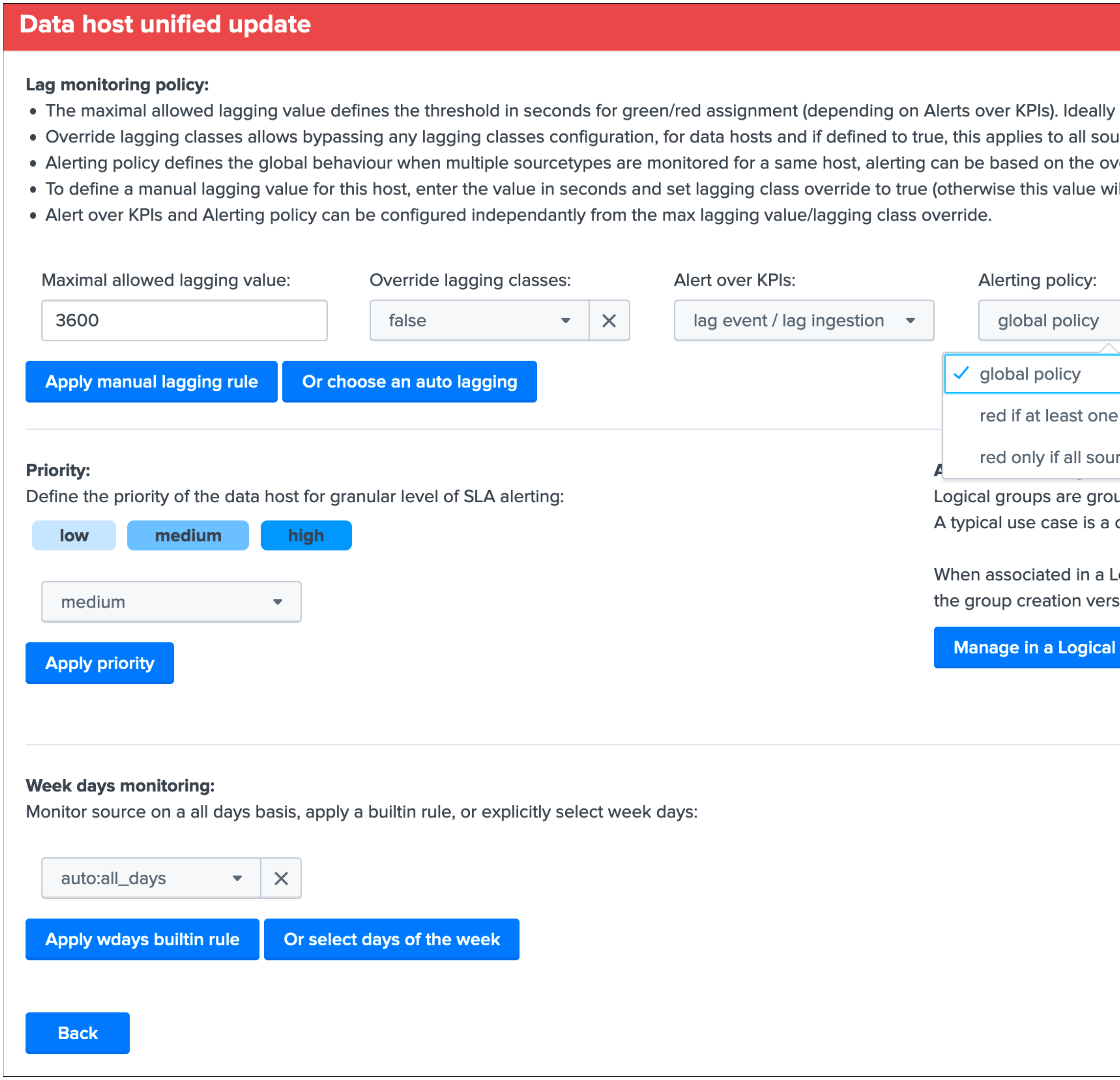
Global policy in TrackMe manage and configure:



DATA HOST GLOBAL ALERTING POLIC

The default behaviour for data hosts monitoring is to turn host states in red mode when all of the known sourcetypes have stopped emitting data i monitoring conditions are met, the host is green) You can change the global policy to the "granular per sourcetype" mode which configures TrackMe t must be green) This policy can defined widely in this configuration screen, and on a per data hos

Per data host policy:



Data host unified update

Lag monitoring policy:

- The maximal allowed lagging value defines the threshold in seconds for green/red assignment (depending on Alerts over KPIs). Ideally
- Override lagging classes allows bypassing any lagging classes configuration, for data hosts and if defined to true, this applies to all sou
- Alerting policy defines the global behaviour when multiple sourcetypes are monitored for a same host, alerting can be based on the ov
- To define a manual lagging value for this host, enter the value in seconds and set lagging class override to true (otherwise this value wi
- Alert over KPIs and Alerting policy can be configured independantly from the max lagging value/lagging class override.

Maximal allowed lagging value: 3600

Override lagging classes: false

Alert over KPIs: lag event / lag ingestion

Alerting policy: global policy

Priority:

Define the priority of the data host for granular level of SLA alerting:

medium

A

Logical groups are grou

A typical use case is a c

When associated in a L

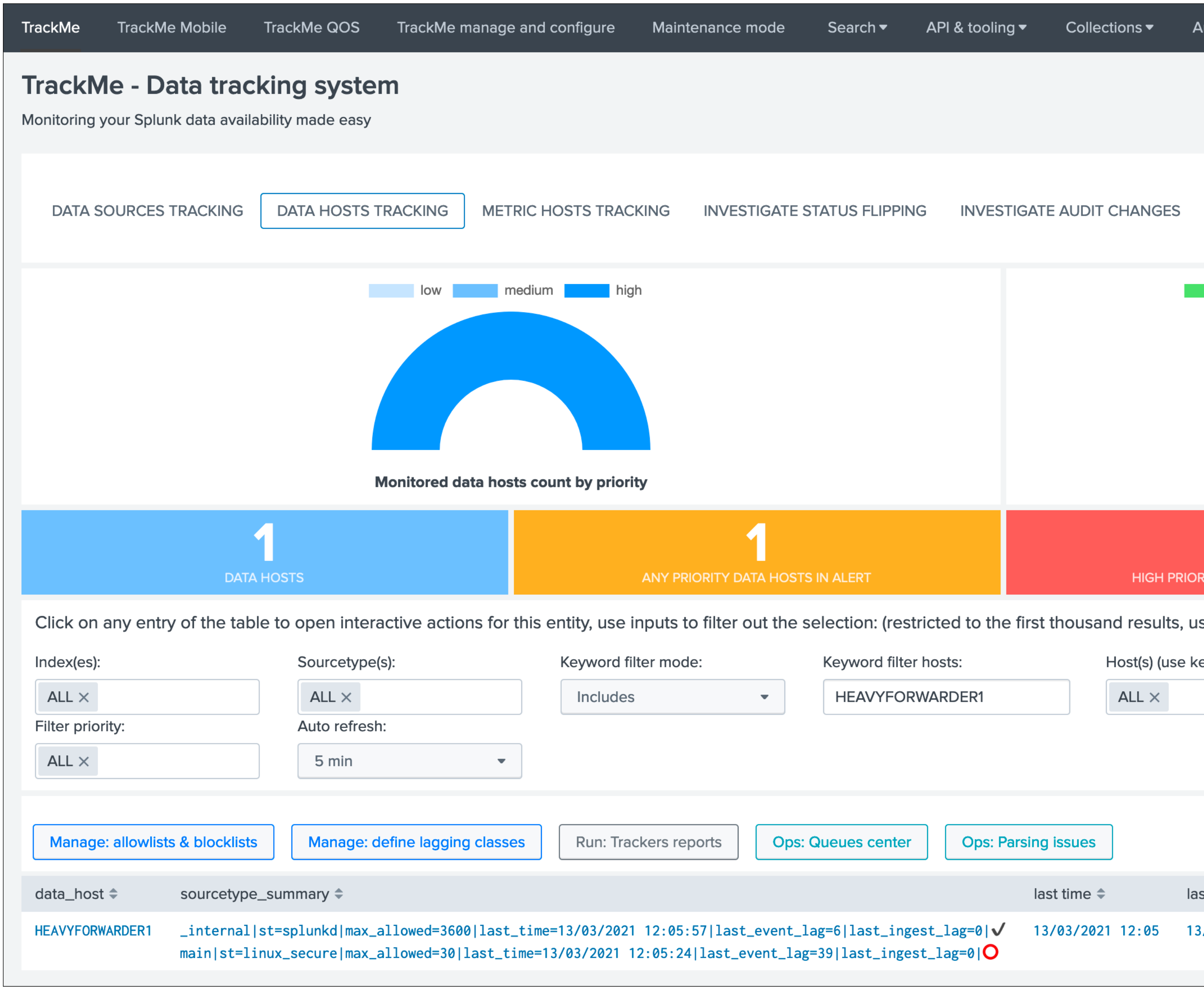
the group creation vers

Week days monitoring:

Monitor source on a all days basis, apply a builtin rule, or explicitly select week days:

auto:all_days

When the global policy, or the per host policy, is set to track per sourcetype, the data host will appear in a non green status if at least one sourcetype is red (for example even if Splunk internal is still going through):



TrackMe - Data tracking system

Monitoring your Splunk data availability made easy

Click on any entry of the table to open interactive actions for this entity, use inputs to filter out the selection: (restricted to the first thousand results, us

Index(es): ALL ×

Sourcetype(s): ALL ×

Keyword filter mode: Includes

Keyword filter hosts: HEAVYFORWARDER1

Host(s) (use ke ALL ×

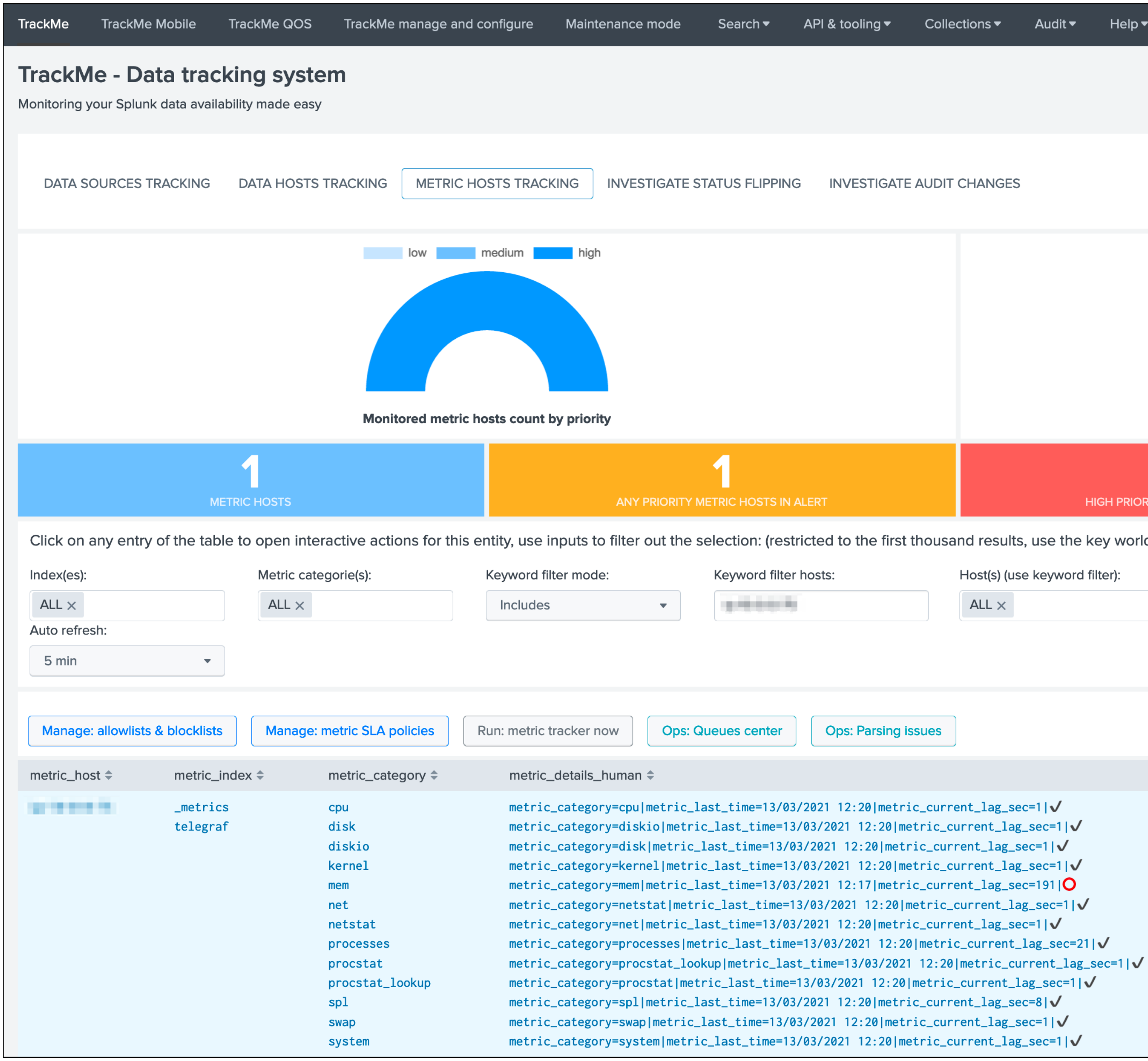
Filter priority: ALL ×

Auto refresh: 5 min

data_host	sourcetype_summary	last time	las
HEAVYFORWARDER1	_internal|st=splunkd|max_allowed=3600|last_time=13/03/2021 12:05:57|last_event_lag=6|last_ingest_lag=0|✓ main|st=linux_secure|max_allowed=30|last_time=13/03/2021 12:05:24|last_event_lag=39|last_ingest_lag=0|○	13/03/2021 12:05	13.

Metric hosts tracking

Metrics tracking acts differently, if any of the metric categories does not comply with monitoring rules (including the spl metrics), the host will turn into a red state:



TrackMe - Data tracking system

Monitoring your Splunk data availability made easy

Click on any entry of the table to open interactive actions for this entity, use inputs to filter out the selection: (restricted to the first thousand results, use the key world

Index(es): ALL ×

Metric categorie(s): ALL ×

Keyword filter mode: Includes

Keyword filter hosts: [redacted]

Host(s) (use keyword filter): ALL ×

Auto refresh: 5 min

metric_host	metric_index	metric_category	metric_details_human
[redacted]	_metrics	cpu	metric_category=cpu|metric_last_time=13/03/2021 12:20|metric_current_lag_sec=1|✓
	telegraf	disk	metric_category=diskio|metric_last_time=13/03/2021 12:20|metric_current_lag_sec=1|✓
		diskio	metric_category=disk|metric_last_time=13/03/2021 12:20|metric_current_lag_sec=1|✓
		kernel	metric_category=kernel|metric_last_time=13/03/2021 12:20|metric_current_lag_sec=1|✓
		mem	metric_category=mem|metric_last_time=13/03/2021 12:17|metric_current_lag_sec=191|○
		net	metric_category=netstat|metric_last_time=13/03/2021 12:20|metric_current_lag_sec=1|✓
		netstat	metric_category=net|metric_last_time=13/03/2021 12:20|metric_current_lag_sec=1|✓
		processes	metric_category=processes|metric_last_time=13/03/2021 12:20|metric_current_lag_sec=21|✓
		procstat	metric_category=procstat_lookup|metric_last_time=13/03/2021 12:20|metric_current_lag_sec=1|✓
		procstat_lookup	metric_category=procstat|metric_last_time=13/03/2021 12:20|metric_current_lag_sec=1|✓
		spl	metric_category=spl|metric_last_time=13/03/2021 12:20|metric_current_lag_sec=8|✓
		swap	metric_category=swap|metric_last_time=13/03/2021 12:20|metric_current_lag_sec=1|✓
		system	metric_category=system|metric_last_time=13/03/2021 12:20|metric_current_lag_sec=1|✓

Congratulations, you have now a builtin, easy and efficient monitoring of your Splunk instances availability, enable and configure alerts up to your preferences the *Alerts tracking* and the job is done!

3.5 REST API Reference Manual

3.5.1 Introduction

TrackMe provides a builtin Python based API, serviced by the Splunk API, and categorized by resource groups.

These resource groups are accessible by specific endpoint paths as following:

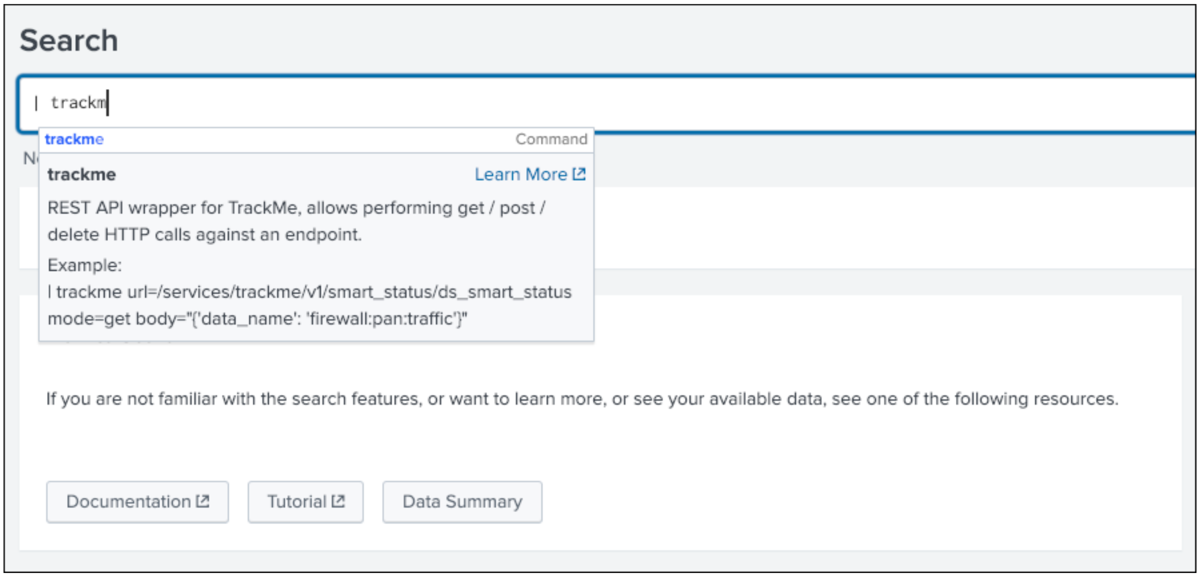
Resource group	API Path
Acknowledgment endpoints	/services/trackme/v1/ack
Data Sources endpoints	/services/trackme/v1/data_sources
Data Hosts endpoints	/services/trackme/v1/data_hosts
Metric Hosts endpoints	/services/trackme/v1/metric_hosts
Elastic Sources endpoints	/services/trackme/v1/elastic_sources
Maintenance mode endpoints	/services/trackme/v1/maintenance
Allow list endpoints	/services/trackme/v1/allowlist
Block list endpoints	/services/trackme/v1/blocklist
Data Sampling endpoints	/services/trackme/v1/data_sampling
Data Sampling models endpoints	/services/trackme/v1/data_sampling_models
Logical Groups endpoints	/services/trackme/v1/logical_groups
Tag policies endpoints	/services/trackme/v1/tag_policies
Lagging classes endpoints	/services/trackme/v1/lagging_classes
Lagging classes metrics endpoints	/services/trackme/v1/lagging_classes_metrics
Smart Status endpoints	/services/trackme/v1/smart_status
Backup and Restore endpoints	/services/trackme/v1/backup_and_restore
Identity Cards endpoints	/services/trackme/v1/identity_cards

These endpoints can be used to interract with TrackMe in a programmatic fashion, for instance to perform integration tasks with automation systems.

3.5.2 REST API trackme SPL command

Interacting with the REST API in SPL queries

TrackMe provides a Python based custom command `trackme` that acts as a REST API wrapper to interract with the API endpoints.



Search

```
| trackm
```

REST API wrapper for TrackMe, allows performing get / post / delete HTTP calls against an endpoint.

Example:

```
| trackme url=/services/trackme/v1/smart_status/ds_smart_status mode=get body="{'data_name': 'firewall:pan:traffic'}"
```

If you are not familiar with the search features, or want to learn more, or see your available data, see one of the following resources.

Syntax

```
| trackme url=<API endpoint> mode=<HTTP method: get/post/delete> body=<Optional:␣
↪provides the HTTP body in a json format>
```

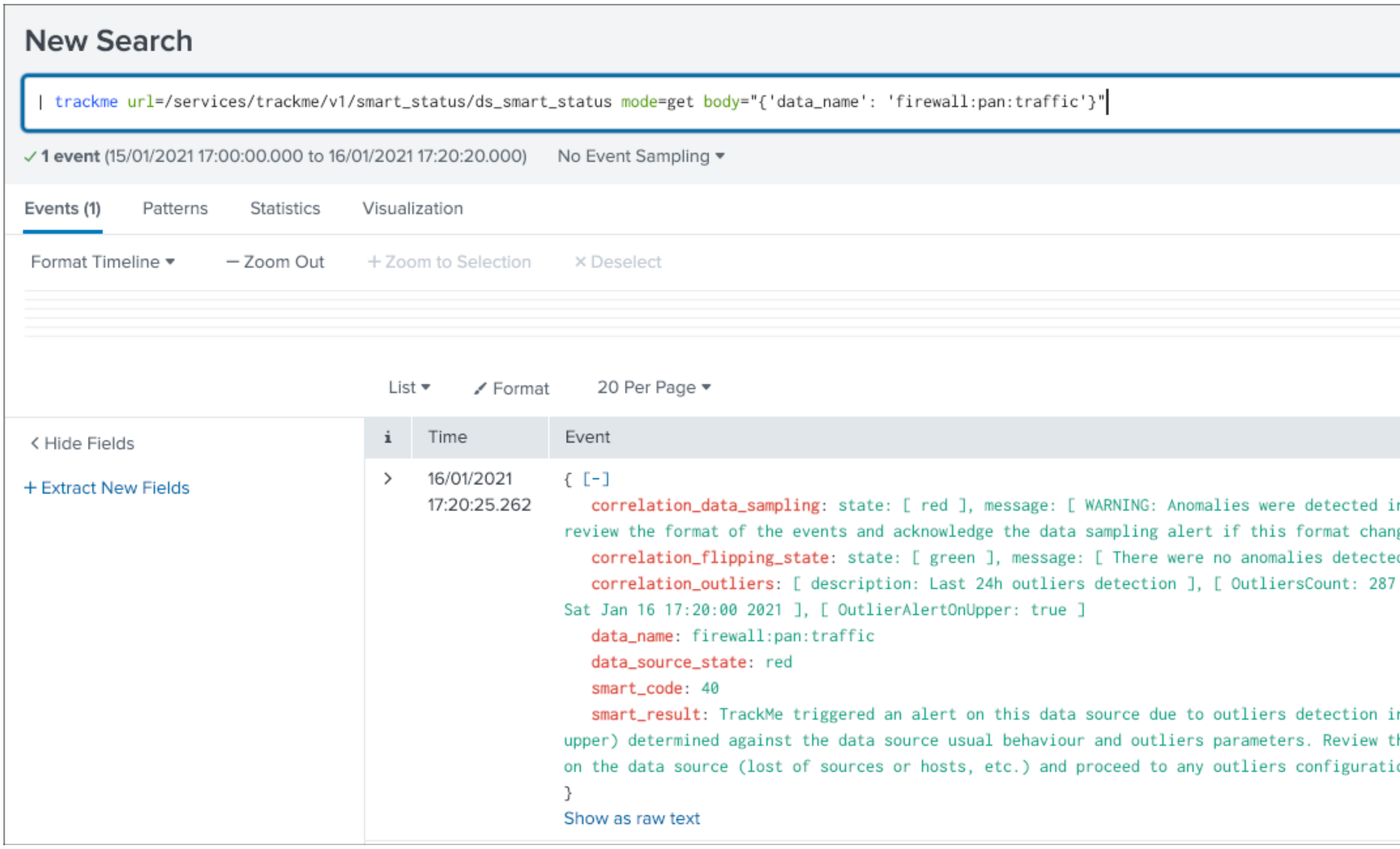
Arguments:

- `url`: (required) describes the API endpoint url, such as `/services/trackme/v1/smart_status/ds_smart_status`
- `mode`: (required) the HTTP mode, valid options are `get`, `post`, `delete`
- `body`: the http body, optional for a get query depending on the endpoint, required for post and delete calls

Example

This example calls the smart_status endpoint for a target data_source:

```
| trackme url=/services/trackme/v1/smart_status/ds_smart_status mode=get body="{'data_
↪name': 'firewall:pan:traffic'}"
```



New Search

```
| trackme url=/services/trackme/v1/smart_status/ds_smart_status mode=get body="{'data_name': 'firewall:pan:traffic'}"
```

✓ 1 event (15/01/2021 17:00:00.000 to 16/01/2021 17:20:20.000)

16/01/2021 17:20:25.262

```
{ [-]
   correlation_data_sampling: state: [ red ], message: [ WARNING: Anomalies were detected i
review the format of the events and acknowledge the data sampling alert if this format chang
   correlation_flipping_state: state: [ green ], message: [ There were no anomalies detecte
   correlation_outliers: [ description: Last 24h outliers detection ], [ OutliersCount: 287
Sat Jan 16 17:20:00 2021 ], [ OutlierAlertOnUpper: true ]
   data_name: firewall:pan:traffic
   data_source_state: red
   smart_code: 40
   smart_result: TrackMe triggered an alert on this data source due to outliers detection i
upper) determined against the data source usual behaviour and outliers parameters. Review th
on the data source (lost of sources or hosts, etc.) and proceed to any outliers configurati
}
```

Every endpoint described in the present REST API reference documentation can be actioned via the trackme custom command, authentication and capabilities are transparently inherited from the user environment running the SPL query.

3.5.3 Authentication

User and roles

You can use any combination of user and roles depending on your preferences, technically, using the TrackMe API endpoint requires read and write permissions to various objects hosted in the TrackMe namespace.

TrackMe contains a builtin role `trackme_admin` which defines required accesses to these objects, you can use this role and make sure the user that will be achieving the rest calls is member of this role, or inherits from it.

Prior Splunk 7.3.0

Prior to Splunk Splunk 7.3.0, the easiest is to used a standard login / password approach to authenticate against Splunk API, similary to:

```
curl -k -u admin:'ch@ngeM3'
```

Alternatively, it is possible to perform first the authentication and retrieve a temporary token to be used for the REST calls:

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"
```

For Splunk 7.3.0 and later

Splunk 7.3.0 introduced the usage of proper authentication tokens, which is the recommended way to authenticate against splunkd API:

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:

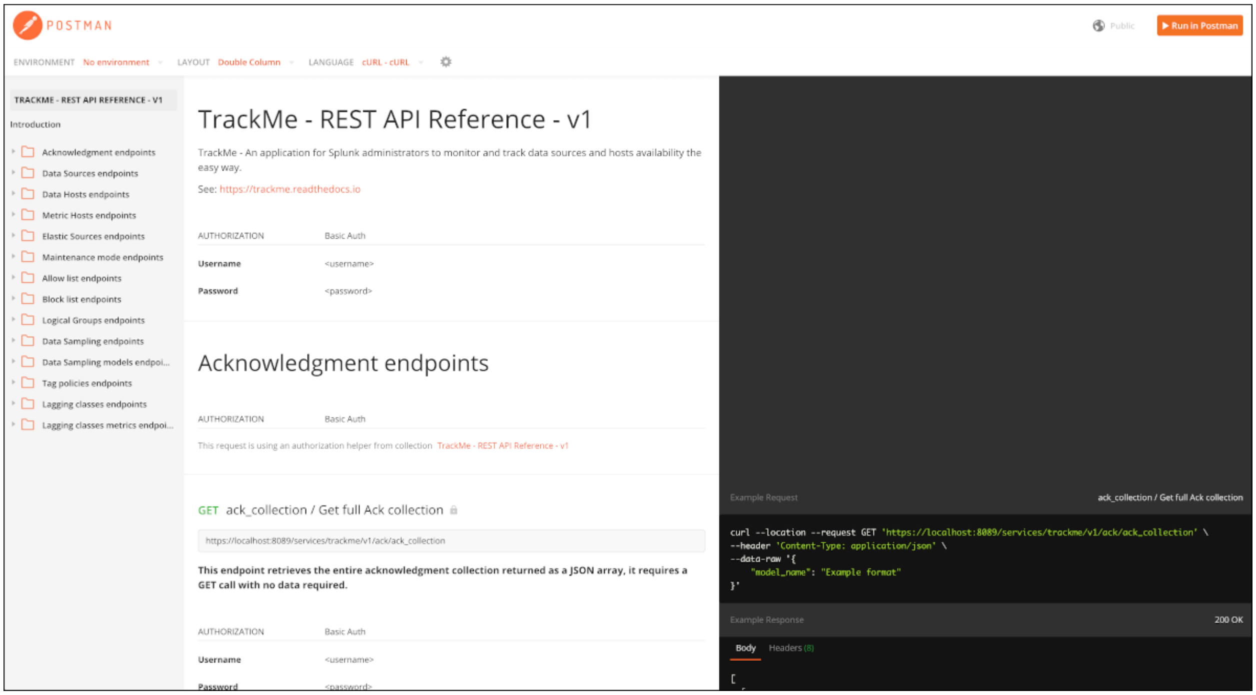
```
curl -k -H "Authorization: Bearer <token>"
```

3.5.4 Postman - API referential and development

TrackMe API endpoints are described in a Postman format at the following public URL:

- https://documenter.getpostman.com/view/7845664/TVt2c3a9#105ff830-5834-4c95-b928-75ab553f5020

If you use Postman, you can consult the API documentation link above and easily import the entire API reference for your testing and development purposes.



TrackMe - REST API Reference - v1

TrackMe - An application for Splunk administrators to monitor and track data sources and hosts availability the easy way.

See: https://trackme.readthedocs.io

AUTHORIZATION	Basic Auth
Username	<username>
Password	<password>

Acknowledgment endpoints

AUTHORIZATION Basic Auth

This request is using an authorization helper from collection TrackMe - REST API Reference - v1

GET ack_collection / Get full Ack collection

https://localhost:8089/services/trackme/v1/ack/ack_collection

This endpoint retrieves the entire acknowledgment collection returned as a JSON array, it requires a GET call with no data required.

AUTHORIZATION	Basic Auth
Username	<username>
Password	<password>

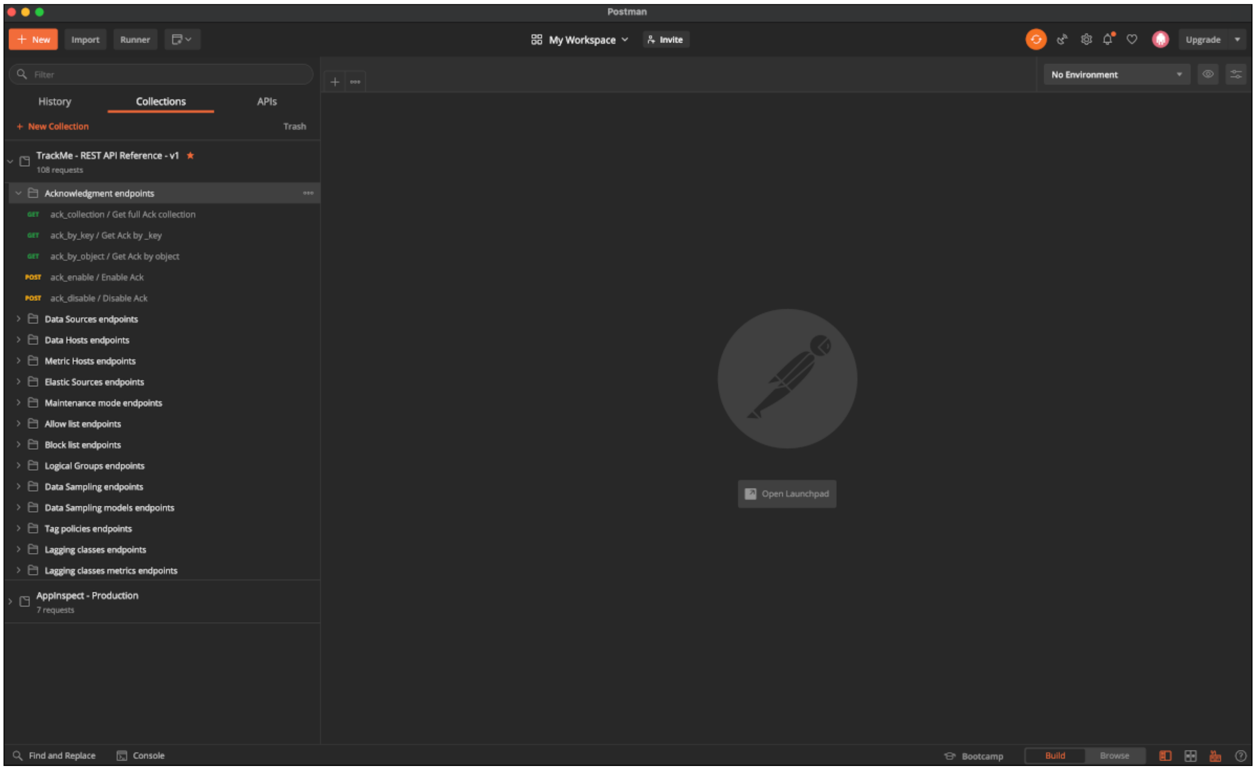
Example Request — ack_collection / Get full Ack collection

```
curl --location --request GET 'https://localhost:8089/services/trackme/v1/ack/ack_collection' \
--header 'Content-Type: application/json' \
--data-raw '{
    "model_name": "Example format"
}'
```

Example Response — 200 OK

Body Headers (8)

```
[
```



3.5.5 Acknowledgment endpoints

Resources summary:

Resource	API Path
ack_collection / Get full Ack collection	/services/trackme/v1/ack/ack_collection
ack_by_key / Get Ack by _key	/services/trackme/v1/ack/ack_by_key
ack_by_object / Get Ack by object	/services/trackme/v1/ack/ack_by_object
ack_enable / Enable Ack	/services/trackme/v1/ack/ack_enable
ack_disable / Disable Ack	/services/trackme/v1/ack/ack_disable

ack_collection / Get full Ack collection

This endpoint retrieves the entire acknowledgment collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/ack/ack_collection
```

SPL query:

```
| trackme url="/services/trackme/v1/ack/ack_collection" mode="get"
```

JSON response: (full collection)

```
[
 {
  "ack_expiration": "1607796255.2581134",
  "ack_mtime": "1607709855.2581134",
  "ack_state": "active",
  "keyid": "5fd3b49f5cfa0d7b797c6181",
  "limit_expiration": "1607795955.2581134",
  "object": "pan:traffic",
  "object_category": "data_source",
  "_user": "nobody",
  "_key": "5fd3b49f5cfa0d7b797c6181"
 },
 {
  "ack_expiration": "1607848092.4875946",
  "ack_mtime": "1607761692.4875946",
  "ack_state": "active",
  "keyid": "5fd47f165cfa0d7b797c8e8f",
  "limit_expiration": "1607847792.4875946",
  "object": "docker_logs:httpevent",
  "object_category": "data_source",
  "object_current_state": "red",
  "_user": "nobody",
  "_key": "5fd47f165cfa0d7b797c8e8f"
 }
]
...
```

ack_by_key / Get Ack by _key

This endpoint retrieves an existing acknowledgment record by the Kvstore key, it requires a GET call with the following information:

- `"_key": KVstore unique identifier for this record`

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/ack/ack_
↪by_key -d '{"_key": "5fd3fe737b1bef735d3f3532"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/ack/ack_by_key" mode="get" body="{\"_key\": \
↪"5fd3fe737b1bef735d3f3532\"}"
```

JSON response:

```
{
 "ack_expiration": "1608333555",
 "ack_mtime": "1607728755",
 "ack_state": "active",
 "keyid": "5fd3fe737b1bef735d3f3532",
 "limit_expiration": "1608333255",
 "object": "network:pan:traffic",
 "object_category": "data_source",
 "object_current_state": "red",
 "_user": "nobody",
 "_key": "5fd3fe737b1bef735d3f3532"
}
```

ack_by_object / Get Ack by object

This endpoint retrieves an existing acknowledgment record by the object name, it requires a GET call with the following information:

External:

- `"object_category": type of object (data_source / data_host / metric_host)"`
- `"object": name of the entity`

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/ack/ack_
↪by_object -d '{"object_category": "data_source", "object": "network:pan:traffic"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/ack/ack_by_object" mode="get" body="{\"object_
↪category\": \"data_source\", \"object\": \"network:pan:traffic\"}"
```

JSON response:

```
[
 {
  "ack_expiration": "1608333555",
  "ack_mtime": "1607728755",
```

(continues on next page)

(continued from previous page)

```
  "ack_state": "active",
  "keyid": "5fd3fe737b1bef735d3f3532",
  "limit_expiration": "1608333255",
  "object": "network:pan:traffic",
  "object_category": "data_source",
  "object_current_state": "red",
  "_user": "nobody",
  "_key": "5fd3fe737b1bef735d3f3532"
 }
]
```

ack_enable / Enable Ack

This endpoint will enable an acknowledgment by the object name, it requires a POST call with the following information:

- `"object_category": type of object (data_source / data_host / metric_host)`
- `"object": name of the entity`
- `"ack_period": period for the acknowledgment in seconds`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/ack/
↪ack_enable -d '{"object_category": "data_source", "object": "network:pan:traffic",
↪"ack_period": "86400", "update_comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/ack/ack_enable" mode="post" body="{\"object_
↪category\": \"data_source\", \"object\": \"network:pan:traffic\", \"ack_period\": \
↪"86400\", \"update_comment\": \"Updated by automation.\"}"
```

JSON response:

```
{
 "object": "network:pan:traffic",
 "object_category": "data_source",
 "ack_expiration": "1607815805.7918282",
 "ack_state": "active",
 "ack_mtime": "1607729405.7918282",
 "_user": "nobody",
 "_key": "5fd3fe737b1bef735d3f3532"
}
```

ack_disable / Disable Ack

This endpoint will disable an acknowledgment by the object name, it requires a POST call with the following information:

- `"object_category": type of object (data_source / data_host / metric_host)"`

- `"object": name of the entity`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/ack/
↪ack_disable -d '{"object_category": "data_source", "object": "network:pan:traffic",
↪"update_comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/ack/ack_disable" mode="post" body="{\"object_
↪category\": \"data_source\", \"object\": \"network:pan:traffic\", \"update_comment\
↪": \"Updated by automation.\"}"
```

JSON response:

```
{
 "object": "network:pan:traffic",
 "object_category": "data_source",
 "ack_expiration": "N/A",
 "ack_state": "inactive",
 "ack_mtime": "1607729326.6667607",
 "_user": "nobody",
 "_key": "5fd3fe737b1bef735d3f3532"
}
```

3.5.6 Data Sources endpoints

Resources summary:

Resource	API Path
ds_collection / Get full Data Sources collection	/services/trackme/v1/data_sources/ds_collection
ds_by_key / Get Data Source by _key	/services/trackme/v1/data_sources/ds_by_key
ds_by_name / Get Data Source by name	/services/trackme/v1/data_sources/ds_by_name
ds_enable_monitoring / Enable monitoring	/services/trackme/v1/data_sources/ds_enable_monitoring
ds_disable_monitoring / Disable monitoring	/services/trackme/v1/data_sources/ds_disable_monitoring
ds_update_priority / Update priority	/services/trackme/v1/data_sources/ds_update_priority
ds_update_lag_policy / Update lagging policy	/services/trackme/v1/data_sources/ds_update_lag_policy
ds_update_min_dcount_host / Update minimal host dcount	/services/trackme/v1/data_sources/ds_update_min_dcount_host
ds_update_wdays_by_name / Update week days monitoring	/services/trackme/v1/data_sources/ds_update_wdays
ds_update_outliers / Update outliers detection configuration	/services/trackme/v1/data_sources/ds_update_outliers
ds_update_monitoring_level / Update monitoring level	/services/trackme/v1/data_sources/ds_update_monitoring_level
ds_delete_temporary / Delete temporary	/services/trackme/v1/data_sources/ds_delete_temporary
ds_delete_permanent / Delete permanently	/services/trackme/v1/data_sources/ds_delete_permanent
ds_enable_data_sampling / Enable data sampling	/services/trackme/v1/data_sources/ds_enable_data_sampling
ds_disable_data_sampling / Disable data sampling	/services/trackme/v1/data_sources/ds_disable_data_sampling
ds_update_data_sampling_records_nr / Update sampling no of records	/services/trackme/v1/data_sources/ds_update_data_sampling_records_nr

ds_collection / Get full Data Sources collection

This endpoint retrieves the entire data sources collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/data_
↪sources/ds_collection
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sources/ds_collection" mode="get"
```

JSON response: (full collection)

```
[
 {
  "OutlierAlertOnUpper": "false",
  "OutlierLowerThresholdMultiplier": "4",
  "OutlierMinEventCount": "0",
  "OutlierSpan": "5m",
  "OutlierTimePeriod": "-7d",
  "OutlierUpperThresholdMultiplier": "4",
  "_time": "1607779500",
  ...
```

ds_by_key / Get Data Source by _key

This endpoint retrieves an existing data source record by the Kvstore key, it requires a GET call with the following information:

- `"_key": KVstore unique identifier for this record`

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/data_
↪sources/ds_by_key -d '{"_key": "7e8670878a9ad91844f18655f1819c06"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sources/ds_by_key" mode="get" body="{\"_key\
↪": \"7e8670878a9ad91844f18655f1819c06\"}"
```

JSON response: (full record)

```
{
"OutlierAlertOnUpper": "false",
"OutlierLowerThresholdMultiplier": "4",
"OutlierMinEventCount": "0",
"OutlierSpan": "5m",
"OutlierTimePeriod": "-7d",
"OutlierUpperThresholdMultiplier": "4",
"_time": "1607770500",
"current_state": "green",
...
```

ds_by_name / Get Data Source by name

This endpoint retrieves an existing data source record by the data source name (data_name), it requires a GET call with the following information:

- `"data_name": name of the data source`

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/data_
↪sources/ds_by_name -d '{"data_name": "network:pan:traffic"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sources/ds_by_name" mode="get" body="{\"data_
↪name\": \"network:pan:traffic\"}"
```

JSON response: (full record)

```
{
"OutlierAlertOnUpper": "false",
"OutlierLowerThresholdMultiplier": "4",
"OutlierMinEventCount": "0",
"OutlierSpan": "5m",
"OutlierTimePeriod": "-7d",
"OutlierUpperThresholdMultiplier": "4",
"_time": "1607770500",
"current_state": "green",
...
```

ds_enable_monitoring / Enable monitoring

This endpoint enables data monitoring for an existing data source by the data source name (data_name), it requires a POST call with the following information:

- `"data_name": name of the data source`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪sources/ds_enable_monitoring -d '{"data_name": "network:pan:traffic", "update_
↪comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sources/ds_enable_monitoring" mode="post"␣
↪body="{\"data_name\": \"network:pan:traffic\", \"update_comment\": \"Updated by␣
↪automation.\"}"
```

JSON response: (full record)

```
{
"OutlierAlertOnUpper": "false",
"OutlierLowerThresholdMultiplier": "4",
"OutlierMinEventCount": "0",
```

(continues on next page)

(continued from previous page)

```
"OutlierSpan": "5m",
"OutlierTimePeriod": "-7d",
"OutlierUpperThresholdMultiplier": "4",
"_time": "1607770500",
"current_state": "green",
...
```

ds_disable_monitoring / Disable monitoring

This endpoint disables data monitoring for an existing data source by the data source name (data_name), it requires a POST call with the following information:

- `"data_name": name of the data source`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪sources/ds_disable_monitoring -d '{"data_name": "network:pan:traffic", "update_
↪comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sources/ds_disable_monitoring" mode="post"␣
↪body="{\"data_name\": \"network:pan:traffic\", \"update_comment\": \"Updated by␣
↪automation.\"}"
```

JSON response: (full record)

```
{
"OutlierAlertOnUpper": "false",
"OutlierLowerThresholdMultiplier": "4",
"OutlierMinEventCount": "0",
"OutlierSpan": "5m",
"OutlierTimePeriod": "-7d",
"OutlierUpperThresholdMultiplier": "4",
"_time": "1607770500",
"current_state": "green",
...
```

ds_update_priority / Update priority

This endpoint updates the priority definition for an existing data source by the data source name (data_name), it requires a POST call with the following information:

- `"data_name": name of the data source`
- `"priority": priority value, valid options are low / medium / high`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪sources/ds_update_priority -d '{"data_name": "network:pan:traffic", "priority":
↪"high", "update_comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sources/ds_update_priority" mode="post" body=
↪"{\"data_name\": \"network:pan:traffic\", \"priority\": \"high\", \"update_comment\
↪": \"Updated by automation.\"}"
```

JSON response: (full record)

```
{
"OutlierAlertOnUpper": "false",
"OutlierLowerThresholdMultiplier": "4",
"OutlierMinEventCount": "0",
"OutlierSpan": "5m",
"OutlierTimePeriod": "-7d",
"OutlierUpperThresholdMultiplier": "4",
"_time": "1607770500",
"current_state": "green",
...
```

ds_update_lag_policy / Update lagging policy

This endpoint configures the lagging policy for an existing data source, it requires a POST call with the following information:

- "data_name": name of the data source
- "data_lag_alert_kpis": KPIs policy to be applied, valid options are all_kpis / lag_ingestion_kpi / lag_event_kpi
- "data_max_lag_allowed": maximal accepted lagging value in seconds, must be an integer
- "data_override_lagging_class": overrides lagging classes, valid options are true / false
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪sources/ds_update_lag_policy -d '{"data_name": "network:pan:traffic", "update_
↪comment": "Updated by automation.", "data_lag_alert_kpis": "lag_ingestion_kpi",
↪"data_max_lag_allowed": "300", "data_override_lagging_class": "true"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sources/ds_update_lag_policy" mode="post"␣
↪body="{\"data_name\": \"network:pan:traffic\", \"update_comment\": \"Updated by␣
↪automation.\", \"data_lag_alert_kpis\": \"lag_ingestion_kpi\", \"data_max_lag_
↪allowed\": \"300\", \"data_override_lagging_class\": \"true\"}"
```

JSON response: (full record)

```
{
"OutlierAlertOnUpper": "false",
"OutlierLowerThresholdMultiplier": "4",
"OutlierMinEventCount": "0",
"OutlierSpan": "5m",
"OutlierTimePeriod": "-7d",
"OutlierUpperThresholdMultiplier": "4",
"_time": "1607770500",
"current_state": "green",
...
```

ds_update_min_dcount_host / Update minimal host dcount

This endpoint configures the minimal number of distinct hosts count for an existing data source, it requires a POST call with the following information:

- "data_name": name of the data source
- "min_dcount_host": minimal accepted number of distinct count hosts, must be an integer or any to disable the feature
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪sources/ds_update_min_dcount_host -d '{"data_name": "network:pan:traffic", "update_
↪comment": "Updated by automation.", "min_dcount_host": "100"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sources/ds_update_min_dcount_host" mode="post
↪" body="{\"data_name\": \"network:pan:traffic\", \"update_comment\": \"Updated by␣
↪automation.\", \"min_dcount_host\": \"100\"}"
```

JSON response: (full record)

```
{
"OutlierAlertOnUpper": "false",
"OutlierLowerThresholdMultiplier": "4",
"OutlierMinEventCount": "0",
"OutlierSpan": "5m",
"OutlierTimePeriod": "-7d",
"OutlierUpperThresholdMultiplier": "4",
"_time": "1607770500",
"current_state": "green",
...
```

ds_update_wdays_by_name / Update week days monitoring

This endpoint configures the week days monitoring rule for an existing data source, it requires a POST call with the following information:

- "data_name": name of the data source

- "data_monitoring_wdays": the week days rule, valid options are manual:all_days / manual:monday-to-friday / manual:monday-to-saturday / [0, 1, 2, 3, 4, 5, 6] where Sunday is 0
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪sources/ds_update_wdays -d '{"data_name": "network:pan:traffic", "update_comment":
↪"Updated by automation.", "data_monitoring_wdays": "manual:monday-to-friday"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sources/ds_update_wdays" mode="post" body="{\
↪"data_name\": \"network:pan:traffic\", \"update_comment\": \"Updated by automation.\
↪", \"data_monitoring_wdays\": \"manual:monday-to-friday\"}"
```

JSON response: (full record)

```
{
"OutlierAlertOnUpper": "false",
"OutlierLowerThresholdMultiplier": "4",
"OutlierMinEventCount": "0",
"OutlierSpan": "5m",
"OutlierTimePeriod": "-7d",
"OutlierUpperThresholdMultiplier": "4",
"_time": "1607770500",
"current_state": "green",
...
```

ds_update_outliers / Update outliers detection configuration

This endpoint configures the week days monitoring rule for an existing data source, it requires a POST call with the following information:

- "data_name": name of the data source
- "OutlierMinEventCount": the minimal number of events, if set to anything bigger than 0, the lower bound becomes a static value, needs to be an integer, default to 0 (disabled)
- "OutlierLowerThresholdMultiplier": The lower bound threshold multiplier, must be an integer, defaults to 4
- "OutlierUpperThresholdMultiplier": The upper bound threshold multiplier, must be integer, defaults to 4
- "OutlierAlertOnUpper": "Enables / Disables alerting on upper outliers detection, valid options are true / false, defaults to false
- "OutlierTimePeriod": relative time period for outliers calculation, default to -7d
- "OutlierSpan": span period Splunk notation for outliers UI rendering, defaults to 5m
- "enable_behaviour_analytic": "Enables / Disables outliers detection for that object, valid options are true / false, defaults to true

- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪sources/ds_update_outliers -d '{"data_name": "network:pan:traffic", "update_comment
↪": "Updated by automation.", "OutlierMinEventCount": "0",
↪"OutlierLowerThresholdMultiplier": "6", "OutlierUpperThresholdMultiplier": "6",
↪"OutlierAlertOnUpper": "false", "OutlierTimePeriod": "7d", "OutlierSpan": "5m",
↪"enable_behaviour_analytic": "true"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sources/ds_update_outliers" mode="post" body=
↪"{\"data_name\": \"network:pan:traffic\", \"update_comment\": \"Updated by␣
↪automation.\", \"OutlierMinEventCount\": \"0\", \"OutlierLowerThresholdMultiplier\
↪": \"6\", \"OutlierUpperThresholdMultiplier\": \"6\", \"OutlierAlertOnUpper\": \
↪"false\", \"OutlierTimePeriod\": \"7d\", \"OutlierSpan\": \"5m\", \"enable_
↪behaviour_analytic\": \"true\"}"
```

JSON response: (full record)

```
{
"OutlierAlertOnUpper": "false",
"OutlierLowerThresholdMultiplier": "4",
"OutlierMinEventCount": "0",
"OutlierSpan": "5m",
"OutlierTimePeriod": "-7d",
"OutlierUpperThresholdMultiplier": "4",
"_time": "1607770500",
"current_state": "green",
...
```

ds_update_monitoring_level / Update monitoring level

This endpoint updates the monitoring level for an existing data source, it requires a POST call with the following information:

- "data_name": name of the data source
- "data_monitoring_level": the monitoring level definition, valid options are index / sourcetype
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪sources/ds_update_monitoring_level -d '{"data_name": "network:pan:traffic", "update_
↪comment": "Updated by automation.", "data_monitoring_level": "sourcetype"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sources/ds_update_monitoring_level" mode=
↪"post" body="{\"data_name\": \"network:pan:traffic\", \"update_comment\": \"Updated␣
↪by automation.\", \"data_monitoring_level\": \"sourcetype\"}"
```

JSON response: (full record)

```
{
"OutlierAlertOnUpper": "false",
"OutlierLowerThresholdMultiplier": "4",
"OutlierMinEventCount": "0",
"OutlierSpan": "5m",
"OutlierTimePeriod": "-7d",
"OutlierUpperThresholdMultiplier": "4",
"_time": "1607770500",
"current_state": "green",
...
```

ds_delete_temporary / Delete temporary

This endpoint performs a temporary deletion of an existing data source, it requires a DELETE call with the following information:

- `"data_name": name of the data source`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

Note: A temporary deletion removes the entity and its configuration, if search conditions such as data avaibility allow it, the same entitiy will be re-created automatically by the Trackers.

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/data_
↪sources/ds_delete_temporary -d '{"data_name": "network:pan:traffic"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sources/ds_delete_temporary" mode="delete"␣
↪body="{\"data_name\": \"network:pan:traffic\"}"
```

JSON response: (full record)

```
Record with _key 7e8670878a9ad91844f18655f1819c06 was temporarily deleted from the␣
↪collection.%
```

ds_delete_permanent / Delete permanently

This endpoint performs a permanent deletion of an existing data source, it requires a DELETE call with the following information:

- `"data_name": name of the data source`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

Note: A permanent deletion removes the entity and its configuration, in addition its a specific audit record to prevent the entity from being created as long as the audit record is not purged. if the audit record is purged and the search conditions such as data avaibility allow it, the same entitiy will be re-created automatically by the Trackers.

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/data_
↪sources/ds_delete_permanent -d '{"data_name": "network:pan:traffic"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sources/ds_delete_permanent" mode="delete"␣
↪body="{\"data_name\": \"network:pan:traffic\"}"
```

JSON response: (full record)

```
Record with _key 7e8670878a9ad91844f18655f1819c06 was permanently deleted from the␣
↪collection.%
```

ds_enable_data_sampling / Enable data sampling

This endpoint enables the data sampling feature for an existing data source by the data source name (data_name), it requires a POST call with the following information:

- "data_name": name of the data source
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪sources/ds_enable_data_sampling -d '{"data_name": "network:pan:traffic", "update_
↪comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sources/ds_enable_data_sampling" mode="post"␣
↪body="{\"data_name\": \"network:pan:traffic\", \"update_comment\": \"Updated by␣
↪automation.\"}"
```

JSON response: (full record)

```
{
 "data_name": "network:pan:traffic",
 "data_sample_feature": "enabled",
 "_user": "nobody",
 "_key": "7e8670878a9ad91844f18655f1819c06"
}
```

ds_disable_data_sampling / Disable data sampling

This endpoint disables the data sampling feature for an existing data source by the data source name (data_name), it requires a POST call with the following information:

- "data_name": name of the data source
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪sources/ds_disable_data_sampling -d '{"data_name": "network:pan:traffic", "update_
↪comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sources/ds_disable_data_sampling" mode="post
↪" body="{\"data_name\": \"network:pan:traffic\", \"update_comment\": \"Updated by␣
↪automation.\"}"
```

JSON response: (full record)

```
{
 "data_name": "network:pan:traffic",
 "data_sample_feature": "disabled",
 "_user": "nobody",
 "_key": "7e8670878a9ad91844f18655f1819c06"
}
```

ds_update_data_sampling_records_nr / Update sampling no of records

This endpoint enables the data sampling feature for an existing data source by the data source name (data_name), it requires a POST call with the following information:

- `"data_name": name of the data source`
- `"data_sampling_nr": number of records to be sampled per data source and data sampling execution (defaults to 100 at first sampling, then 50)`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪sources/ds_update_data_sampling_records_nr -d '{"data_name": "network:pan:traffic",
↪"data_sampling_nr": "200", "update_comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sources/ds_update_data_sampling_records_nr"␣
↪mode="post" body="{\"data_name\": \"network:pan:traffic\", \"data_sampling_nr\": \
↪"200\", \"update_comment\": \"Updated by automation.\"}"
```

JSON response:

```
{
 "data_name": "network:pan:traffic",
 "data_sampling_nr": "200",
 "raw_sample": [
 ...
```

3.5.7 Data Hosts endpoints

Resources summary:

Resource	API Path
dh_collection / Get full Data Hosts collection	/services/trackme/v1/data_hosts/dh_collection
dh_by_key / Get data host by _key	/services/trackme/v1/data_hosts/dh_by_key
dh_by_name / Get data host by name	/services/trackme/v1/data_hosts/dh_by_name
dh_enable_monitoring / Enable monitoring	/services/trackme/v1/data_hosts/dh_enable_monitoring
dh_disable_monitoring / Disable monitoring	/services/trackme/v1/data_hosts/dh_disable_monitoring
dh_update_priority / Update priority	/services/trackme/v1/data_hosts/dh_update_priority
dh_reset / Reset data host	/services/trackme/v1/data_hosts/dh_reset
dh_update_lag_policy / Update lagging policy	/services/trackme/v1/data_hosts/dh_update_lag_policy
dh_update_wdays / Update week days monitoring	/services/trackme/v1/data_hosts/dh_update_wdays
dh_update_outliers / Update outliers detection configuration	/services/trackme/v1/data_hosts/dh_update_outliers
dh_delete_temporary / Delete temporary	/services/trackme/v1/data_hosts/dh_delete_temporary
dh_delete_permanent / Delete permanently	/services/trackme/v1/data_hosts/dh_delete_permanent

dh_collection / Get full Data Hosts collection

This endpoint retrieves the entire data hosts collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/data_
↪hosts/dh_collection
```

SPL query:

```
| trackme url="/services/trackme/v1/data_hosts/dh_collection" mode="get"
```

JSON response: (full collection)

```
[
 {
 "OutlierAlertOnUpper": "false",
 "OutlierLowerThresholdMultiplier": "4",
 "OutlierMinEventCount": "0",
 "OutlierSpan": "5m",
 "OutlierTimePeriod": "-7d",
 "OutlierUpperThresholdMultiplier": "4",
 "_time": "1607781900",
 "current_state": "green",
 "data_custom_max_lag_allowed": "0",
 "data_eventcount": "60",
 "data_first_time_seen": "1607781871",
 "data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1",
 "data_host_alerting_policy": "global_policy",
 ...
```

dh_by_key / Get data host by _key

This endpoint retrieves an existing data host record by the Kvstore key, it requires a GET call with the following information:

- `"_key": KVstore unique identifier for this record`

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/data_
↪hosts/dh_by_key -d '{"_key": "14781cf495c76f1373382197f071c5d6"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_hosts/dh_by_key" mode="get" body="{\"_key\":␣
↪\"14781cf495c76f1373382197f071c5d6\"}"
```

JSON response: (full record)

```
{
 "OutlierAlertOnUpper": "false",
 "OutlierLowerThresholdMultiplier": "4",
 "OutlierMinEventCount": "0",
 "OutlierSpan": "5m",
 "OutlierTimePeriod": "-7d",
 "OutlierUpperThresholdMultiplier": "4",
 "_time": "1607781900",
 "current_state": "green",
 "data_custom_max_lag_allowed": "0",
 "data_eventcount": "60",
 "data_first_time_seen": "1607781871",
 "data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1",
 ...
```

dh_by_name / Get data host by name

This endpoint retrieves an existing data host record by the data host name (data_host), it requires a GET call with the following information:

- `"data_host": name of the data host`

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/data_
↪hosts/dh_by_name -d '{"data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_hosts/dh_by_name" mode="get" body="{\"data_
↪host\": \"FIREWALL.PAN.AMER.DESIGN.NODE1\"}"
```

JSON response: (full record)

```
[
 {
  "OutlierAlertOnUpper": "false",
  "OutlierLowerThresholdMultiplier": "4",
  "OutlierMinEventCount": "0",
  "OutlierSpan": "5m",
  "OutlierTimePeriod": "-7d",
  "OutlierUpperThresholdMultiplier": "4",
  "_time": "1607782200",
  "current_state": "green",
  "data_custom_max_lag_allowed": "0",
```

(continues on next page)

(continued from previous page)

```
  "data_eventcount": "338",
  "data_first_time_seen": "1607781871",
  "data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1",
  ...
```

dh_enable_monitoring / Enable monitoring

This endpoint enables data monitoring for an existing data host by the data host name (data_host), it requires a POST call with the following information:

- "data_host": name of the data host
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪hosts/dh_enable_monitoring -d '{"data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1",
↪"update_comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_hosts/dh_enable_monitoring" mode="post" body=
↪"{\"data_host\": \"FIREWALL.PAN.AMER.DESIGN.NODE1\", \"update_comment\": \"Updated␣
↪by automation.\"}"
```

JSON response: (full record)

```
{
 "object_category": "data_host",
 "data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1",
 "data_index": "firewall",
 "data_sourcetype": "pan:traffic",
 "data_last_lag_seen": "-5",
 "data_last_ingestion_lag_seen": "0",
 "data_eventcount": "338",
 "data_first_time_seen": "1607781871",
 ...
```

dh_disable_monitoring / Disable monitoring

This endpoint disables data monitoring for an existing data host by the data host name (data_host), it requires a POST call with the following information:

- "data_host": name of the data host
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪hosts/dh_disable_monitoring -d '{"data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1",
↪"update_comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_hosts/dh_disable_monitoring" mode="post"␣
↪body="{\"data_host\": \"FIREWALL.PAN.AMER.DESIGN.NODE1\", \"update_comment\": \
↪"Updated by automation.\"}"
```

JSON response: (full record)

```
{
 "object_category": "data_host",
 "data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1",
 "data_index": "firewall",
 "data_sourcetype": "pan:traffic",
 "data_last_lag_seen": "-5",
 "data_last_ingestion_lag_seen": "0",
 "data_eventcount": "338",
 "data_first_time_seen": "1607781871",
 ...
```

dh_update_priority / Update priority

This endpoint updates the priority definition for an existing data host by the data host name (data_host), it requires a POST call with the following information:

- `"data_host": name of the data host`
- `"priority": priority value, valid options are low / medium / high`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪hosts/dh_update_priority -d '{"data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1",
↪"priority": "high", "update_comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_hosts/dh_update_priority" mode="post" body="
↪{\"data_host\": \"FIREWALL.PAN.AMER.DESIGN.NODE1\", \"priority\": \"high\", \
↪"update_comment\": \"Updated by automation.\"}"
```

JSON response: (full record)

```
{
 "object_category": "data_host",
 "data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1",
 "data_last_lag_seen": "-2",
 "data_last_ingestion_lag_seen": "0",
 "data_eventcount": "2585",
 ...
```

dh_reset / Reset data host

This endpoint resets (removal of index and sourcetype knowledge) an existing data host by the data host name (data_host), it requires a POST call with the following information:

- "data_host": name of the data host
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪hosts/dh_reset -d '{"data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1", "update_comment
↪": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_hosts/dh_reset" mode="post" body="{\"data_
↪host\": \"FIREWALL.PAN.AMER.DESIGN.NODE1\", \"update_comment\": \"Updated by␣
↪automation.\"}"
```

JSON response: (full record)

```
{
 "object_category": "data_host",
 "data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1",
 "data_last_lag_seen": "-2",
 "data_last_ingestion_lag_seen": "0",
 "data_eventcount": "2585",
 ...
```

dh_update_lag_policy / Update lagging policy

This endpoint configures the lagging policy for an existing data host, it requires a POST call with the following information:

- "data_host": name of the data host
- "data_lag_alert_kpis": KPIs policy to be applied, valid options are all_kpis / lag_ingestion_kpi / lag_event_kpi
- "data_max_lag_allowed": maximal accepted lagging value in seconds, must be an integer
- "data_override_lagging_class": overrides lagging classes, valid options are true / false
- "data_host_alerting_policy": policy alerting, valid options are global_policy / track_per_sourcetype / track_per_host
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪hosts/dh_update_lag_policy -d '{"data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1",
↪"update_comment": "Updated by automation.", "data_lag_alert_kpis": "lag_ingestion_
↪kpi", "data_max_lag_allowed": "300", "data_override_lagging_class": "true", "data_
↪host_alerting_policy": "track_per_sourcetype"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_hosts/dh_update_lag_policy" mode="post" body=
↪"{\"data_host\": \"FIREWALL.PAN.AMER.DESIGN.NODE1\", \"update_comment\": \"Updated␣
↪by automation.\", \"data_lag_alert_kpis\": \"lag_ingestion_kpi\", \"data_max_lag_
↪allowed\": \"300\", \"data_override_lagging_class\": \"true\", \"data_host_alerting_
↪policy\": \"track_per_sourcetype\"}"
```

JSON response: (full record)

```
{
 "object_category": "data_host",
 "data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1",
 "data_index": "firewall",
 "data_sourcetype": "pan:traffic",
 "data_last_lag_seen": "-4",
 "data_last_ingestion_lag_seen": "0",
 "data_eventcount": "5756",
 "data_first_time_seen": "1607205117",
 ...
```

dh_update_wdays / Update week days monitoring

This endpoint configures the week days monitoring rule for an existing data host, it requires a POST call with the following information:

- `"data_host": name of the data host`
- `"data_monitoring_wdays": the week days rule, valid options are manual:all_days / manual:monday-to-friday / manual:monday-to-saturday / [ 0, 1, 2, 3, 4, 5, 6 ] where Sunday is 0`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪hosts/dh_update_wdays -d '{"data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1", "update_
↪comment": "Updated by automation.", "data_monitoring_wdays": "manual:monday-to-
↪friday"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_hosts/dh_update_wdays" mode="post" body="{\
↪"data_host\": \"FIREWALL.PAN.AMER.DESIGN.NODE1\", \"update_comment\": \"Updated by␣
↪automation.\", \"data_monitoring_wdays\": \"manual:monday-to-friday\"}"
```

JSON response: (full record)

```
{
 "object_category": "data_host",
 "data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1",
 "data_index": "firewall",
 "data_sourcetype": "pan:traffic",
 "data_last_lag_seen": "-7",
 "data_last_ingestion_lag_seen": "0",
 "data_eventcount": "938",
```

(continues on next page)

(continued from previous page)

```
 "data_first_time_seen": "1607781871",
 ...
```

dh_update_outliers / Update outliers detection configuration

This endpoint configures the week days monitoring rule for an existing data host, it requires a POST call with the following information:

- `"data_host": name of the data host`
- `"OutlierMinEventCount": the minimal number of events, if set to anything bigger than 0, the lower bound becomes a static value, needs to be an integer, default to 0 (disabled)`
- `"OutlierLowerThresholdMultiplier": The lower bound threshold multiplier, must be an integer, defaults to 4`
- `"OutlierUpperThresholdMultiplier": The upper bound threshold multiplier, must be integer, defaults to 4`
- `"OutlierAlertOnUpper": "Enables / Disables alerting on upper outliers detection, valid options are true / false, defaults to false`
- `"OutlierTimePeriod": relative time period for outliers calculation, default to -7d`
- `"OutlierSpan": span period Splunk notation for outliers UI rendering, defaults to 5m`
- `"enable_behaviour_analytic": "Enables / Disables outliers detection for that object, valid options are true / false, defaults to true`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪hosts/dh_update_outliers -d '{"data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1",
↪"update_comment": "Updated by automation.", "OutlierMinEventCount": "0",
↪"OutlierLowerThresholdMultiplier": "6", "OutlierUpperThresholdMultiplier": "6",
↪"OutlierAlertOnUpper": "false", "OutlierTimePeriod": "7d", "OutlierSpan": "5m",
↪"enable_behaviour_analytic": "true"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_hosts/dh_update_outliers" mode="post" body="
↪{\"data_host\": \"FIREWALL.PAN.AMER.DESIGN.NODE1\", \"update_comment\": \"Updated␣
↪by automation.\", \"OutlierMinEventCount\": \"0\", \
↪"OutlierLowerThresholdMultiplier\": \"6\", \"OutlierUpperThresholdMultiplier\": \"6\
↪", \"OutlierAlertOnUpper\": \"false\", \"OutlierTimePeriod\": \"7d\", \"OutlierSpan\
↪": \"5m\", \"enable_behaviour_analytic\": \"true\"}"
```

JSON response: (full record)

```
{
 "object_category": "data_host",
 "data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1",
```

(continues on next page)

(continued from previous page)

```
"data_index": "firewall",
"data_sourcetype": "pan:traffic",
"data_last_lag_seen": "-7",
"data_last_ingestion_lag_seen": "0",
"data_eventcount": "938",
...
```

dh_delete_temporary / Delete temporary

This endpoint performs a temporary deletion of an existing data host, it requires a DELETE call with the following information:

- `"data_host": name of the data host`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

Note: A temporary deletion removes the entity and its configuration, if search conditions such as data avaibility allow it, the same entitiy will be re-created automatically by the Trackers.

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/data_
↪hosts/dh_delete_temporary -d '{"data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_hosts/dh_delete_temporary" mode="delete"␣
↪body="{\"data_host\": \"FIREWALL.PAN.AMER.DESIGN.NODE1\"}"
```

JSON response: (full record)

```
Record with _key 7e8670878a9ad91844f18655f1819c06 was temporarily deleted from the␣
↪collection.%
```

dh_delete_permanent / Delete permanently

This endpoint performs a permanent deletion of an existing data host, it requires a DELETE call with the following information:

- `"data_host": name of the data host`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

Note: A permanent deletion removes the entity and its configuration, in addition its a specific audit record to prevent the entity from being created as long as the audit record is not purged. if the audit record is purged and the search conditions such as data avaibility allow it, the same entitiy will be re-created automatically by the Trackers.

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/data_
↪hosts/dh_delete_permanent -d '{"data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_hosts/dh_delete_permanent" mode="delete"␣
↪body="{\"data_host\": \"FIREWALL.PAN.AMER.DESIGN.NODE1\"}"
```

JSON response: (full record)

```
Record with _key 7e8670878a9ad91844f18655f1819c06 was permanently deleted from the␣
↪collection.%
```

3.5.8 Metric Hosts endpoints

Resources summary:

Resource	API Path
mh_collection / Get full Metric Hosts collection	/services/trackme/v1/metric_hosts/mh_collection
mh_by_key / Get metric host by _key	/services/trackme/v1/metric_hosts/mh_by_key
mh_by_name / Get metric host by name	/services/trackme/v1/metric_hosts/mh_by_name
mh_enable_monitoring / Enable monitoring	/services/trackme/v1/metric_hosts/mh_enable_monitoring
mh_disable_monitoring / Disable monitoring	/services/trackme/v1/metric_hosts/mh_disable_monitoring
mh_update_priority / Update priority	/services/trackme/v1/metric_hosts/mh_update_priority
mh_reset / Reset metrics	/services/trackme/v1/metric_hosts/mh_reset
mh_delete_temporary / Delete temporary	/services/trackme/v1/metric_hosts/mh_delete_temporary
mh_delete_permanent / Delete permanently	/services/trackme/v1/metric_hosts/mh_delete_permanent

mh_collection / Get full Metric Hosts collection

This endpoint retrieves the entire metric hosts collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/metric_
↪hosts/mh_collection
```

SPL query:

```
| trackme url="/services/trackme/v1/metric_hosts/mh_collection" mode="get"
```

JSON response: (full collection)

```
[
 {
  "_time": "1607815039",
  "current_state": "green",
  "info_max_time": "1607815039.000",
  "info_min_time": "1607814739.000",
  "info_search_time": "1607815039.524",
  "info_sid": "1607815039.126",
  "latest_flip_state": "green",
  "latest_flip_time": "1607815039",
  ...
```

mh_by_key / Get metric host by _key

This endpoint retrieves an existing metric host record by the Kvstore key, it requires a GET call with the following information:

- `"_key": KVstore unique identifier for this record`

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/metric_
↪hosts/mh_by_key -d '{"_key": "afb0c5fc92f20c8011ecac371b04f77e"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/metric_hosts/mh_by_key" mode="get" body="{\"_key\
↪": \"afb0c5fc92f20c8011ecac371b04f77e\"}"
```

JSON response: (full record)

```
{
 "_time": "1607815039",
 "current_state": "green",
 "info_max_time": "1607815039.000",
 "info_min_time": "1607814739.000",
 "info_search_time": "1607815039.524",
 "info_sid": "1607815039.126",
 "latest_flip_state": "green",
 "latest_flip_time": "1607815039",
 ...
```

mh_by_name / Get metric host by name

This endpoint retrieves an existing metric host record by the metric host name (metric_host), it requires a GET call with the following information:

- `"metric_host": name of the metric host`

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/metric_
↪hosts/mh_by_name -d '{"metric_host": "telegraf-node1"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/metric_hosts/mh_by_name" mode="get" body="{\
↪"metric_host\": \"telegraf-node1\"}"
```

JSON response: (full record)

```
[
 {
  "_time": "1607815200",
  "current_state": "green",
  "info_max_time": "1607815200.000",
  "info_min_time": "1607814900.000",
  "info_search_time": "1607815201.133",
  "info_sid": "scheduler__admin__trackme__RMD56299d9dc7b583db4_at_1607815200_6",
```

(continues on next page)

(continued from previous page)

```
  "latest_flip_state": "green",
  "latest_flip_time": "1607815039",
  ...
```

mh_enable_monitoring / Enable monitoring

This endpoint enables data monitoring for an existing metric host by the metric host name (metric_host), it requires a POST call with the following information:

- "metric_host": name of the metric host
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/metric_
↪hosts/mh_enable_monitoring -d '{"metric_host": "telegraf-node1", "update_comment":
↪"Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/metric_hosts/mh_enable_monitoring" mode="post"␣
↪body="{\"metric_host\": \"telegraf-node1\", \"update_comment\": \"Updated by␣
↪automation.\"}"
```

JSON response: (full record)

```
{
 "object_category": "metric_host",
 "metric_host": "telegraf-node1",
 "metric_index": "telegraf",
 "metric_category": "docker,docker_container_blkio,docker_container_cpu,docker_
↪container_health,docker_container_mem,docker_container_net,docker_container_status",
 ...
```

mh_disable_monitoring / Disable monitoring

This endpoint disables data monitoring for an existing metric host by the metric host name (metric_host), it requires a POST call with the following information:

- "metric_host": name of the metric host
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/metric_
↪hosts/mh_disable_monitoring -d '{"metric_host": "telegraf-node1", "update_comment":
↪"Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/metric_hosts/mh_disable_monitoring" mode="post"␣
↪body="{\"metric_host\": \"telegraf-node1\", \"update_comment\": \"Updated by␣
↪automation.\"}"
```

(continues on next page)

(continued from previous page)

JSON response: (full record)

```
{
 "object_category": "metric_host",
 "metric_host": "telegraf-node1",
 "metric_index": "telegraf",
 "metric_category": "docker,docker_container_blkio,docker_container_cpu,docker_
↪container_health,docker_container_mem,docker_container_net,docker_container_status",
 ...
```

mh_update_priority / Update priority

This endpoint updates the priority definition for an existing metric host, it requires a POST call with the following information:

- `"metric_host": name of the metric host`
- `"priority": priority value, valid options are low / medium / high`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/metric_
↪hosts/mh_update_priority -d '{"metric_host": "telegraf-node1", "priority": "high",
↪"update_comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/metric_hosts/mh_update_priority" mode="post" body=
↪"{\"metric_host\": \"telegraf-node1\", \"priority\": \"high\", \"update_comment\": \
↪"Updated by automation.\"}"
```

JSON response: (full record)

```
{
 "object_category": "metric_host",
 "metric_host": "telegraf-node1",
 "metric_index": "telegraf",
 "metric_last_lag_seen": "8",
 ...
```

mh_reset / Reset metrics

This endpoint resets (removal of indexes and metrics knowledge) an existing metric host by the metric host name (metric_host), it requires a POST call with the following information:

- `"metric_host": name of the metric host`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/metric_
↪hosts/mh_reset -d '{"metric_host": "telegraf-node1", "update_comment": "Updated by␣
↪automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/metric_hosts/mh_reset" mode="post" body="{\
↪"metric_host\": \"telegraf-node1\", \"update_comment\": \"Updated by automation.\"}"
```

JSON response: (full record)

```
{
 "object_category": "metric_host",
 "metric_host": "telegraf-node1",
 "metric_index": "telegraf",
 "metric_last_lag_seen": "8",
 ...
```

mh_delete_temporary / Delete temporary

This endpoint performs a temporary deletion of an existing metric host, it requires a DELETE call with the following information:

- `"metric_host": name of the metric host`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

Note: A temporary deletion removes the entity and its configuration, if search conditions such as data avaibility allow it, the same entitiy will be re-created automatically by the Trackers.

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪metric_hosts/mh_delete_temporary -d '{"metric_host": "telegraf-node1"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/metric_hosts/mh_delete_temporary" mode="delete"␣
↪body="{\"metric_host\": \"telegraf-node1\"}"
```

JSON response: (full record)

```
Record with _key afb0c5fc92f20c8011ecac371b04f77e was temporarily deleted from the␣
↪collection.%
```

mh_delete_permanent / Delete permanently

This endpoint performs a permanent deletion of an existing metric host, it requires a DELETE call with the following information:

- `"metric_host": name of the metric host`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

Note: A permanent deletion removes the entity and its configuration, in addition its a specific audit record to prevent the entity from being created as long as the audit record is not purged. if the audit record is purged and the search conditions such as data avaibility allow it, the same entitiy will be re-created automatically by the Trackers.

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪metric_hosts/mh_delete_permanent -d '{"metric_host": "telegraf-node1"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/metric_hosts/mh_delete_permanent" mode="delete"␣
↪body="{\"metric_host\": \"telegraf-node1\"}"
```

JSON response: (full record)

```
Record with _key afb0c5fc92f20c8011ecac371b04f77e was permanently deleted from the␣
↪collection.%
```

3.5.9 Elastic Sources endpoints

Resources summary:

Resource	API Path
elastic_shared / Get shared Elastic Sources collection	/services/trackme/v1/elastic_sources/elastic_shared
elastic_dedicated / Get dedicated Elastic Sources collection	/services/trackme/v1/elastic_sources/elastic_dedicated
elastic_shared_by_name / Get shared Elastic Source by name	/services/trackme/v1/elastic_sources/elastic_shared_by_name
elastic_dedicated_by_name / Get shared Elastic Source by name	/services/trackme/v1/elastic_sources/elastic_dedicated_by_name
elastic_shared_add / Add or update a new shared Elastic Source	/services/trackme/v1/elastic_sources/elastic_shared_add
elastic_dedicated_add / Add or update a new dedicated Elastic Source	/services/trackme/v1/elastic_sources/elastic_dedicated_add
elastic_shared_del / Delete a new shared Elastic Source	/services/trackme/v1/elastic_sources/elastic_shared_del
elastic_dedicated_del / Delete a new shared Elastic Source	/services/trackme/v1/elastic_sources/elastic_dedicated_del

elastic_shared / Get shared Elastic Sources collection

This endpoint retrieves the entired shared Elastic Sources collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/elastic_
↪sources/elastic_shared
```

SPL query:

```
| trackme url="/services/trackme/v1/elastic_sources/elastic_shared" mode="get"
```

JSON response:

```
[
 {
  "data_name": "elastic:shared:example:tstats",
  "search_constraint": "index=\"network\" sourcetype=\"pan:traffic\" source=\
↪"network:pan:amer\"",
  "search_mode": "tstats",
  "elastic_data_index": "network",
  "elastic_data_sourcetype": "pan:traffic",
  "_user": "nobody",
  "_key": "5fdbc1a4a507cc26ee02af61"
 }
]
```

elastic_dedicated / Get dedicated Elastic Sources collection

This endpoint retrieves the entired dedicated Elastic Sources collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/elastic_
↪sources/elastic_dedicated
```

SPL query:

```
| trackme url="/services/trackme/v1/elastic_sources/elastic_dedicated" mode="get"
```

JSON response:

```
[
 {
  "data_name": "elastic:shared:example:tstats",
  "search_constraint": "index=\"network\" sourcetype=\"pan:traffic\" source=\
↪"network:pan:amer\"",
  "search_mode": "tstats",
  "elastic_data_index": "network",
  "elastic_data_sourcetype": "pan:traffic",
  "_user": "nobody",
  "_key": "5fdbc1a4a507cc26ee02af61"
 }
]
```

elastic_shared_by_name / Get shared Elastic Source by name

This endpoint retrieves a shared Elastic Source configuration stored in the collection returned as a JSON array, it requires a GET call with the following information:

- `"data_name": name of the Elastic Source`

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/elastic_
↪sources/elastic_shared_by_name -d '{"data_name": "elastic:shared:example:tstats"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/elastic_sources/elastic_shared_by_name" mode="get
↪" body="{\"data_name\": \"elastic:shared:example:tstats\"}"
```

JSON response:

```
[
 {
  "data_name": "elastic:shared:example:tstats",
  "search_constraint": "index=\"network\" sourcetype=\"pan:traffic\" source=\
↪"network:pan:amer\"",
  "search_mode": "tstats",
  "elastic_data_index": "network",
  "elastic_data_sourcetype": "pan:traffic",
  "_user": "nobody",
  "_key": "5fdbc1a4a507cc26ee02af61"
 }
]
```

elastic_dedicated_by_name / Get shared Elastic Source by name

This endpoint retrieves a dedicated Elastic Source configuration stored in the collection returned as a JSON array, it requires a GET call with the following information:

- `"data_name": name of the Elastic Source`

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/elastic_
↪sources/elastic_dedicated_by_name -d '{"data_name":
↪"elastic:dedicated:example:tstats"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/elastic_sources/elastic_dedicated_by_name" mode=
↪"get" body="{\"data_name\": \"elastic:dedicated:example:tstats\"}"
```

JSON response:

```
[
 {
  "data_name": "elastic:shared:example:tstats",
  "search_constraint": "index=\"network\" sourcetype=\"pan:traffic\" source=\
↪"network:pan:amer\"",
  "search_mode": "tstats",
  "elastic_data_index": "network",
  "elastic_data_sourcetype": "pan:traffic",
  "_user": "nobody",
  "_key": "5fdbc1a4a507cc26ee02af61"
 }
]
```

elastic_shared_add / Add or update a new shared Elastic Source

This endpoint create a new shared Elastic Source, if the entity already exists it will be updated using the data provided, it requires a POST call with the following information:

- `"data_name": name of the Elastic Source`
- `"search_constraint": the SPL code for this entity, double quotes need to be escaped`
- `"search_mode": the search mode, valid options are tstats / raw / from / mstats / rest_tstats / rest_raw / rest_from / rest_mstats`
- `"elastic_index": pseudo index value, this value will be used in the UI but has no impacts on the search`
- `"elastic_sourcetype": pseudo sourcetype value name, this value will be used in the UI but has no impacts on the search`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

Defining the search constraint:

- **tstats**: this represents the where part of a tstats search, as: `index=my_index source=my_source`
- **raw**: Any filter that is before stats calculation, as: `index=my_index tag=authentication app=my_application`
- **from (datamodel)**: a search using from is in 2 parts with a pipe separation, where the 1st segment is the object and the 2nd a search constraint, as: `datamodel:"Authentication" | search user="*" action="success" app="my_application"`
- **from (lookup)**: A lookup can be monitored with the from command, it requires the lookup to have a time field concept, and a field _time in epoch time format needs to be created using an eval function with strftime/strptime, such as: `lookup:"my_lookup" | eval _time=strptime(lastUpdated, "%d/%m/%Y %H:%M:%S")`
- **mstats**: Allows monitoring metric indexes according to your constraints including dimensions, as: `index="k8s_metrics" metric_name="k8s.*" cluster_name="production"`
- **rest**: these are special remote searches performed against the Splunk API using the SPL rest command. This allows tracking data that is not available to the search head(s) hosting TrackMe.

Syntax examples for rest searches, the first part before the pipe needs to contain the rest target:

`splunk_server="my_search_head" | index=my_index source=my_source`

`splunk_server_group="dmc_searchheadclustergroup_shc1" | lookup:asset_cmdb_lookup | eval _time=strptime(lastUpdated, "%d/%m/%Y %H:%M:%S")`

Filters can include a time range which will override the default 4 hours time range of the wrapper tracker, as: earliest="-15m" latest="+15m"

tstats based example:

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪elastic_sources/elastic_shared_add -d '{"data_name": "elastic:shared:example:tstats
↪", "search_constraint": "index=\"network\" sourcetype=\"pan:traffic\" source=\
↪"network:pan:amer\"", "search_mode": "tstats", "elastic_data_index": "network",
↪"elastic_data_sourcetype": "pan:traffic"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/elastic_sources/elastic_shared_add" mode="post"␣
↪body="{\"data_name\": \"elastic:shared:example:tstats\", \"search_constraint\": \
↪"index=\\\"network\\\" sourcetype=\\\"pan:traffic\\\" source=\\\"network:pan:amer\\\
↪"\", \"search_mode\": \"tstats\", \"elastic_data_index\": \"network\", \"elastic_
↪data_sourcetype\": \"pan:traffic\"}"
```

(continues on next page)

(continued from previous page)

JSON response:

```
[
 {
  "data_name": "elastic:shared:example:tstats",
  "search_constraint": "index=\"network\" sourcetype=\"pan:traffic\" source=\
↪"network:pan:amer\"",
  "search_mode": "tstats",
  "elastic_data_index": "network",
  "elastic_data_sourcetype": "pan:traffic",
  "_user": "nobody",
  "_key": "5fdbc1a4a507cc26ee02af61"
 }
]
```

raw based example:

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪elastic_sources/elastic_shared_add -d '{"data_name": "elastic:shared:example:raw",
↪"search_constraint": "index=\"network\" sourcetype=\"pan:traffic\" source=\
↪"network:pan:amer\" earliest=\"-30m\"", "search_mode": "raw", "elastic_data_index":
↪"network", "elastic_data_sourcetype": "pan:traffic"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/elastic_sources/elastic_shared_add" mode="post"␣
↪body="{\"data_name\": \"elastic:shared:example:raw\", \"search_constraint\": \
↪"index=\\\"network\\\" sourcetype=\\\"pan:traffic\\\" source=\\\"network:pan:amer\\\
↪" earliest=\\\"-30m\\\"\", \"search_mode\": \"raw\", \"elastic_data_index\": \
↪"network\", \"elastic_data_sourcetype\": \"pan:traffic\"}"
```

from datamodel based example:

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪elastic_sources/elastic_shared_add -d '{"data_name":
↪"elastic:shared:from:datamodel:example", "search_constraint": "datamodel:\
↪"Authentication\" | search user=* action=*", "search_mode": "from", "elastic_data_
↪index": "datamodel", "elastic_data_sourcetype": "auth:example"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/elastic_sources/elastic_shared_add" mode="post"␣
↪body="{\"data_name\": \"elastic:shared:from:datamodel:example\", \"search_
↪constraint\": \"datamodel:\\\"Authentication\\\" | search user=* action=*\", \
↪"search_mode\": \"from\", \"elastic_data_index\": \"datamodel\", \"elastic_data_
↪sourcetype\": \"auth:example\"}"
```

from lookup based example:

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪elastic_sources/elastic_shared_add -d '{"data_name":
↪"elastic:shared:from:lookup:example", "search_constraint": "lookup:\"acme_cmdb_
↪lookup\"", "search_mode": "from", "elastic_data_index": "lookup", "elastic_data_
↪sourcetype": "cmdb:example"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/elastic_sources/elastic_shared_add" mode="post"␣
↪body="{\"data_name\": \"elastic:shared:from:lookup:example\", \"search_constraint\
↪": \"lookup:\\\"acme_cmdb_lookup\\\"\", \"search_mode\": \"from\", \"elastic_data_
↪index\": \"lookup\", \"elastic_data_sourcetype\": \"cmdb:example\"}"
```

External:

mstats based example:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪elastic_sources/elastic_shared_add -d '{"data_name": "elastic:shared:mstats:example
↪", "search_constraint": "index=* metric_name=\"docker_container_cpu*\" earliest=\"-
↪5m\" latest=now", "search_mode": "mstats", "elastic_data_index": "metrics",
↪"elastic_data_sourcetype": "metrics:docker"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/elastic_sources/elastic_shared_add" mode="post"␣
↪body="{\"data_name\": \"elastic:shared:mstats:example\", \"search_constraint\": \
↪"index=* metric_name=\\\"docker_container_cpu*\\\" earliest=\\\"-5m\\\" latest=now\
↪", \"search_mode\": \"mstats\", \"elastic_data_index\": \"metrics\", \"elastic_data_
↪sourcetype\": \"metrics:docker\"}"
```

elastic_dedicated_add / Add or update a new dedicated Elastic Source

This endpoint create a new shared Elastic Source, if the entity already exists it will be updated using the data provided, it requires a POST call with the following information:

Note: if the entity exists already, both the collection and the scheduled report (including the search constraint) will be updated

- `"data_name": name of the Elastic Source`
- `"search_constraint": the SPL code for this entity, double quotes need to be escaped`
- `"search_mode": the search mode, valid options are tstats / raw / from / mstats / rest_tstats / rest_raw / rest_from / rest_mstats`
- `"elastic_index": pseudo index value, this value will be used in the UI but has no impacts on the search`
- `"elastic_sourcetype": pseudo sourcetype value name, this value will be used in the UI but has no impacts on the search`
- `"earliest_time": OPTIONAL: earliest time for the scheduled report definition, if unset will be defined to -4h`
- `"latest_time": OPTIONAL: latest time for the scheduled report definition, if unset will be defined to -4h`

- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

Defining the search constraint:

- **tstats**: this represents the where part of a tstats search, as: `index=my_index source=my_source`
- **raw**: Any filter that is before stats calculation, as: `index=my_index tag=authentication app=my_application`
- **from (datamodel)**: a search using from is in 2 parts with a pipe separation, where the 1st segment is the object and the 2nd a search constraint, as: `datamodel:"Authentication" | search user="*" action="success" app="my_application"`
- **from (lookup)**: A lookup can be monitored with the from command, it requires the lookup to have a time field concept, and a field _time in epoch time format needs to be created using an eval function with strftime/strptime, such as: `lookup:"my_lookup" | eval _time=strptime(lastUpdated, "%d/%m/%Y %H:%M:%S")`
- **mstats**: Allows monitoring metric indexes according to your constraints including dimensions, as: `index="k8s_metrics" metric_name="k8s.*" cluster_name="production"`
- **rest**: these are special remote searches performed against the Splunk API using the SPL rest command. This allows tracking data that is not available to the search head(s) hosting TrackMe.

Syntax examples for rest searches, the first part before the pipe needs to contain the rest target:

`splunk_server="my_search_head" | index=my_index source=my_source`

`splunk_server_group="dmc_searchheadclustergroup_shc1" | lookup:asset_cmdb_lookup | eval _time=strptime(lastUpdated, "%d/%m/%Y %H:%M:%S")`

Filters can include a time range which will override the default 4 hours time range of the wrapper tracker, as: earliest="-15m" latest="+15m"

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪elastic_sources/elastic_dedicated_add -d '{"data_name":
↪"elastic:dedicated:example:tstats", "search_constraint": "index=\"network\"␣
↪sourcetype=\"pan:traffic\" source=\"network:pan:amer\"", "search_mode": "tstats",
↪"elastic_data_index": "network", "elastic_data_sourcetype": "pan:traffic",
↪"earliest_time": "-4h", "latest_time": "+4h"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/elastic_sources/elastic_dedicated_add" mode="post
↪" body="{\"data_name\": \"elastic:dedicated:example:tstats\", \"search_constraint\
↪": \"index=\\\"network\\\" sourcetype=\\\"pan:traffic\\\" source=\\\
↪"network:pan:amer\\\"\", \"search_mode\": \"tstats\", \"elastic_data_index\": \
↪"network\", \"elastic_data_sourcetype\": \"pan:traffic\", \"earliest_time\": \"-4h\
↪", \"latest_time\": \"+4h\"}"
```

JSON response:

```
[
 {
  "data_name": "elastic:dedicated:example:tstats",
  "search_constraint": "index=\"network\" sourcetype=\"pan:traffic\" source=\
↪"network:pan:amer\"",
  "search_mode": "tstats",
```

(continues on next page)

(continued from previous page)

```
  "elastic_data_index": "network",
  "elastic_data_sourcetype": "pan:traffic",
  "elastic_report": "TrackMe - elastic:dedicated:example tracker 0e9ec926-b179-4e30-
↪8295-3b2283efbbc6",
  "_user": "nobody",
  "_key": "5fdbc3b5a507cc26ee02af63"
 }
]
```

raw based example:

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪elastic_sources/elastic_dedicated_add -d '{"data_name":
↪"elastic:dedicated:example:raw", "search_constraint": "index=\"network\"␣
↪sourcetype=\"pan:traffic\" source=\"network:pan:amer\"", "search_mode": "raw",
↪"elastic_data_index": "network", "elastic_data_sourcetype": "pan:traffic", "earliest
↪": "-30m", "latest": "now"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/elastic_sources/elastic_dedicated_add" mode="post
↪" body="{\"data_name\": \"elastic:dedicated:example:raw\", \"search_constraint\": \
↪"index=\\\"network\\\" sourcetype=\\\"pan:traffic\\\" source=\\\"network:pan:amer\\\
↪"\", \"search_mode\": \"raw\", \"elastic_data_index\": \"network\", \"elastic_data_
↪sourcetype\": \"pan:traffic\", \"earliest\": \"-30m\", \"latest\": \"now\"}"
```

from datamodel based example:

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪elastic_sources/elastic_dedicated_add -d '{"data_name":
↪"elastic:dedicated:from:datamodel:example", "search_constraint": "datamodel:\
↪"Authentication\" | search user=* action=*", "search_mode": "from", "elastic_data_
↪index": "datamodel", "elastic_data_sourcetype": "auth:example", "earliest": "-30m",
↪"latest": "now"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/elastic_sources/elastic_dedicated_add" mode="post
↪" body="{\"data_name\": \"elastic:dedicated:from:datamodel:example\", \"search_
↪constraint\": \"datamodel:\\\"Authentication\\\" | search user=* action=*\", \
↪"search_mode\": \"from\", \"elastic_data_index\": \"datamodel\", \"elastic_data_
↪sourcetype\": \"auth:example\", \"earliest\": \"-30m\", \"latest\": \"now\"}"
```

from lookup based example:

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪elastic_sources/elastic_dedicated_add -d '{"data_name":
↪"elastic:dedicated:from:lookup:example", "search_constraint": "lookup:\"acme_cmdb_
↪lookup\"", "search_mode": "from", "elastic_data_index": "lookup", "elastic_data_
↪sourcetype": "cmdb:example"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/elastic_sources/elastic_dedicated_add" mode="post
↪" body="{\"data_name\": \"elastic:dedicated:from:lookup:example\", \"search_
↪constraint\": \"lookup:\\\"acme_cmdb_lookup\\\"\", \"search_mode\": \"from\", \
↪"elastic_data_index\": \"lookup\", \"elastic_data_sourcetype\": \"cmdb:example\"}"
```

mstats based example:

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪elastic_sources/elastic_dedicated_add -d '{"data_name":
↪"elastic:dedicated:mstats:example", "search_constraint": "index=* metric_name=\
↪"docker_container_cpu*\"", "search_mode": "mstats", "elastic_data_index": "metrics",
↪ "elastic_data_sourcetype": "metrics:docker", "earliest": "-5m", "latest": "now"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/elastic_sources/elastic_dedicated_add" mode="post
↪" body="{\"data_name\": \"elastic:dedicated:mstats:example\", \"search_constraint\
↪": \"index=* metric_name=\\\"docker_container_cpu*\\\"\", \"search_mode\": \"mstats\
↪", \"elastic_data_index\": \"metrics\", \"elastic_data_sourcetype\": \
↪"metrics:docker\", \"earliest\": \"-5m\", \"latest\": \"now\"}"
```

elastic_shared_del / Delete a new shared Elastic Source

This endpoint deletes a shared Elastic Source, it requires a DELETE call with the following information:

- `"data_name": name of the Elastic Source`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

Notes:

- The elastic source record is deleted from the shared Elastic Sources collection
- The associated record in the data sources collection is deleted
- All settings related to these objects will be removed permanently after being audited

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪elastic_sources/elastic_shared_del -d '{"data_name": "elastic:shared:example:tstats
↪"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/elastic_sources/elastic_shared_del" mode="delete"␣
↪body="{\"data_name\": \"elastic:shared:example:tstats\"}"
```

response:

```
Record with _key 5fdd373e19456602e57e3a38 was deleted from the Elastic source␣
↪collection, record with _key 221edfe4bec95befadc820fd36cbbfba was deleted from the␣
↪data sources collection.
```

elastic_dedicated_del / Delete a new shared Elastic Source

This endpoint deletes a dedicated Elastic Source, it requires a DELETE call with the following information:

- `"data_name": name of the Elastic Source`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

Notes:

- The elastic source record is deleted from the shared Elastic Sources collection
- The associated scheduled report is deleted
- The associated record in the data sources collection is deleted
- All settings related to these objects will be removed permanently after being audited

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪elastic_sources/elastic_dedicated_del -d '{"data_name":
↪"elastic:dedicated:example:tstats"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/elastic_sources/elastic_dedicated_del" mode=
↪"delete" body="{\"data_name\": \"elastic:dedicated:example:tstats\"}"
```

response:

```
Record with _key 5fdd366719456602e57e3a34 was deleted from the Elastic source␣
↪collection, report with name TrackMe - elastic:test:dedicated tracker 64b23aa6-5445-
↪4512-94e8-1130361c3cea was deleted, record with _key␣
↪e903269a757dbdf1a8e4d26feee96d2a was deleted from the data sources collection.
```

3.5.10 Maintenance mode endpoints

Resources summary:

Resource	API Path
maintenance_status / Get maintenance mode	/services/trackme/v1/maintenance/maintenance_status
maintenance_enable / Enable maintenance mode	/services/trackme/v1/maintenance/maintenance_enable
maintenance_disable / Disable maintenance mode	/services/trackme/v1/maintenance/maintenance_disable

maintenance_status / Get maintenance mode

This endpoint retrieves the current maintenance mode collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/
↪maintenance/maintenance_status
```

SPL query:

```
| trackme url="/services/trackme/v1/maintenance/maintenance_status" mode="get"
```

JSON response: (full collection)

```
[
 {
  "maintenance_mode": "disabled",
  "time_updated": "1607859191",
  "_user": "nobody",
  "_key": "5fd5fd92b21b3338341e63c1"
 }
]
```

maintenance_enable / Enable maintenance mode

This endpoint enables the maintenance mode, it requires a POST call with the following information:

- `"maintenance_duration": integer`

OPTIONAL: the duration of the maintenance window in seconds, if unspecified and maintenance_mode_end is not specified either, defaults to now plus 24 hours

- `"maintenance_mode_end": integer`

OPTIONAL: the date time in epochtime format for the end of the maintenance window, it is overriden by maintenance_duration if specified, defaults to now plus 24 hours if not specified and maintenance_duration is not specified

- `"maintenance_mode_start": integer`

OPTIONAL: the date time in epochtime format for the start of the maintennce window, defaults to now if not specified

- `"update_comment": string`

OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

Immediately start a maintenance window for 24 hours:

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪maintenance/maintenance_enable -d '{"updated_comment": "Beginning a 24 hours␣
↪maintenance window."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/maintenance/maintenance_enable" mode="post" body="
↪{\"updated_comment\": \"Beginning a 24 hours maintenance window.\"}"
```

Immediately start a maintenance window for 1 hour:

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪maintenance/maintenance_enable -d '{"updated_comment": "Beginning maintenance␣
↪window for 1 hour.", "maintenance_duration": "3600"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/maintenance/maintenance_enable" mode="post" body="
↪{\"updated_comment\": \"Beginning maintenance window for 1 hour.\", \"maintenance_
↪duration\": \"3600\"}"
```

Create a scheduled maintenance window with an explicit start and end in epochtime:

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪maintenance/maintenance_enable -d '{"updated_comment": "Beginning maintenance␣
↪window for 1 hour.", "maintenance_mode_start": "1607878800", "maintenance_mode_end
↪": "1607904000"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/maintenance/maintenance_enable" mode="post" body="
↪{\"updated_comment\": \"Beginning maintenance window for 1 hour.\", \"maintenance_
↪mode_start\": \"1607878800\", \"maintenance_mode_end\": \"1607904000\"}"
```

JSON response:

```
[
 {
  "maintenance_mode": "enabled",
  "time_updated": "1607859834",
  "maintenance_mode_start": "1607859834",
  "maintenance_mode_end": "1607946234",
  "_user": "nobody",
  "_key": "5fd5fd92b21b3338341e63c1"
 }
]
```

maintenance_disable / Disable maintenance mode

This endpoint disables the maintenance mode, it requires a POST call with the following information:

- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

Immediately stops the maintenance window:

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪maintenance/maintenance_disable -d '{"updated_comment": "Terminating the␣
↪maintenance window."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/maintenance/maintenance_disable" mode="post" body=
↪"{\"updated_comment\": \"Terminating the maintenance window.\"}"
```

JSON response:

```
[
 {
  "maintenance_mode": "disabled",
```

(continues on next page)

(continued from previous page)

```
    "time_updated": "1607860485",
    "maintenance_mode_start": "N/A",
    "maintenance_mode_end": "N/A",
    "_user": "nobody",
    "_key": "5fd600aec14381564521b181"
  }
]
```

3.5.11 Allow list endpoints

Resources summary:

Resource	API Path
allowlist_ds / Get current allow list for data sources	/services/trackme/v1/allowlist/allowlist_ds
allowlist_dh / Get current allow list for data hosts	/services/trackme/v1/allowlist/allowlist_dh
allowlist_mh / Get current allow list for metric hosts	/services/trackme/v1/allowlist/allowlist_mh
allowlist_ds_add / Add index allow list for data sources	/services/trackme/v1/allowlist/allowlist_ds_add
allowlist_ds_del / Remove index allow list for data sources	/services/trackme/v1/allowlist/allowlist_ds_del
allowlist_dh_add / Add index allow list for data hosts	/services/trackme/v1/allowlist/allowlist_dh_add
allowlist_dh_del / Remove index allow list for data hosts	/services/trackme/v1/allowlist/allowlist_dh_del
allowlist_mh_add / Add index allow list for metric hosts	/services/trackme/v1/allowlist/allowlist_mh_add
allowlist_mh_del / Remove index allow list for metric hosts	/services/trackme/v1/allowlist/allowlist_mh_del

allowlist_ds / Get current allow list for data sources

This endpoint retrieves the current allow list collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/
↪allowlist/allowlist_ds
```

SPL query:

```
| trackme url="/services/trackme/v1/allowlist/allowlist_ds" mode="get"
```

JSON response: (full collection)

```
[
  {
    "_user": "nobody",
    "_key": "5fd6378bba5afb0b0c37b5a1",
    "_time": "1607874443",
    "data_index": "network"
  }
]
```

allowlist_dh / Get current allow list for data hosts

This endpoint retrieves the current allow list collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/
↪allowlist/allowlist_dh
```

SPL query:

```
| trackme url="/services/trackme/v1/allowlist/allowlist_dh" mode="get"
```

JSON response: (full collection)

```
[
 {
  "_user": "nobody",
  "_key": "5fd637a6ba5afb0bbe206350",
  "_time": "1607874470",
  "data_index": "network"
 }
]
```

allowlist_mh / Get current allow list for metric hosts

This endpoint retrieves the current allow list collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/
↪allowlist/allowlist_mh
```

SPL query:

```
| trackme url="/services/trackme/v1/allowlist/allowlist_mh" mode="get"
```

JSON response: (full collection)

```
[
 {
  "_user": "nobody",
  "_key": "5fd637b1ba5afb12790c7261",
  "_time": "1607874481",
  "metric_index": "telegraf"
 }
]
```

allowlist_ds_add / Add index allow list for data sources

This endpoint adds a new allow list record for data sources, it requires a POST call with the following information:

- `"data_index": name of the index to be allowed, wildcards are accepted`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪allowlist/allowlist_ds_add -d '{"data_index": "network*"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/allowlist/allowlist_ds_add" mode="post" body="{\
↪"data_index\": \"network*\"}"
```

JSON response:

```
[
 {
  "_key": "5fd638beba5afb01ff1cfd97",
  "data_index": "network*",
  "_user": "nobody"
 }
]
```

allowlist_ds_del / Remove index allow list for data sources

This endpoint deletes an allow list record for data sources, it requires a DELETE call with the following information:

- "data_index": name of the index to be allowed, wildcards are accepted
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪allowlist/allowlist_ds_del -d '{"data_index": "network*"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/allowlist/allowlist_ds_del" mode="delete" body="{\
↪"data_index\": \"network*\"}"
```

response:

```
Record with _key 5fd66c07ba5afb01ff00d595 was deleted from the collection.
```

allowlist_dh_add / Add index allow list for data hosts

This endpoint adds a new allow list record for data hosts, it requires a POST call with the following information:

- "data_index": name of the index to be allowed, wildcards are accepted
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪allowlist/allowlist_dh_add -d '{"data_index": "network*"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/allowlist/allowlist_dh_add" mode="post" body="{\
↪"data_index\": \"network*\"}"
```

response:

```
[
 {
  "_key": "5fd6685eba5afb01ff1cfd99",
  "_user": "nobody",
  "data_index": "network*"
 }
]
```

allowlist_dh_del / Remove index allow list for data hosts

This endpoint deletes an allow list record for data hosts, it requires a DELETE call with the following information:

- `"data_index": name of the index to be allowed, wildcards are accepted`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪allowlist/allowlist_dh_del -d '{"data_index": "network*"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/allowlist/allowlist_dh_del" mode="delete" body="{\
↪"data_index\": \"network*\"}"
```

response:

```
Record with _key 5fd66c3cba5afb01ff00d598 was deleted from the collection.
```

allowlist_mh_add / Add index allow list for metric hosts

This endpoint adds a new allow list record for metric hosts, it requires a POST call with the following information:

- `"metric_index": name of the index to be allowed, wildcards are accepted`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪allowlist/allowlist_mh_add -d '{"metric_index": "telegraf"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/allowlist/allowlist_mh_add" mode="post" body="{\
↪"metric_index\": \"telegraf\"}"
```

JSON response:

```
[
 {
  "_user": "nobody",
  "metric_index": "telegraf",
  "_key": "5fd66877ba5afb01ff1cfd9b"
 }
]
```

allowlist_mh_del / Remove index allow list for metric hosts

This endpoint deletes an allow list record for metric hosts, it requires a DELETE call with the following information:

- `"metric_index": name of the index to be allowed, wildcards are accepted`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪allowlist/allowlist_mh_del -d '{"metric_index": "telegraf"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/allowlist/allowlist_mh_del" mode="delete" body="{\
↪"metric_index\": \"telegraf\"}"
```

response:

```
Record with _key 5fd66c55ba5afb01ff00d59b was deleted from the collection.
```

3.5.12 Block list endpoints

Resources summary:

Resource	API Path
blocklist_ds_host / Get current block list for data sources (hosts)	/services/trackme/v1/blocklist/blocklist_ds
blocklist_ds_index / Get current block list for data sources (index)	/services/trackme/v1/blocklist/blocklist_ds
blocklist_ds_sourcetype / Get current block list for data sources (sourcetype)	/services/trackme/v1/blocklist/blocklist_ds
blocklist_ds_data_name / Get current block list for data names (data_name)	/services/trackme/v1/blocklist/blocklist_ds
blocklist_dh_host / Get current block list for data hosts (hosts)	/services/trackme/v1/blocklist/blocklist_dh
blocklist_dh_index / Get current block list for data hosts (index)	/services/trackme/v1/blocklist/blocklist_dh
blocklist_dh_sourcetype / Get current block list for data hosts (sourcetype)	/services/trackme/v1/blocklist/blocklist_dh
blocklist_mh_host / Get current block list for metric hosts (host)	/services/trackme/v1/blocklist/blocklist_m
blocklist_mh_index / Get current block list for metric hosts (index)	/services/trackme/v1/blocklist/blocklist_m
blocklist_mh_metric_category / Get current block list for metric hosts (metric_category)	/services/trackme/v1/blocklist/blocklist_m
blocklist_ds_host_add / Add host in block list for data sources	/services/trackme/v1/blocklist/blocklist_ds
blocklist_ds_index_add / Add index in block list for data sources	/services/trackme/v1/blocklist/blocklist_ds
blocklist_ds_sourcetype_add / Add sourcetype in block list for data sources	/services/trackme/v1/blocklist/blocklist_ds
blocklist_ds_data_name_add / Add data name in block list for data sources	/services/trackme/v1/blocklist/blocklist_ds

Co

Table 1 – continued from previous page

Resource	API Path
blocklist_dh_host_add / Add host in block list for data hosts	/services/trackme/v1/blocklist/blocklist_dh
blocklist_dh_index_add / Add index in block list for data hosts	/services/trackme/v1/blocklist/blocklist_dh
blocklist_dh_sourcetype_add / Add sourcetype in block list for data hosts	/services/trackme/v1/blocklist/blocklist_dh
blocklist_mh_host_add / Add host in block list for metric hosts	/services/trackme/v1/blocklist/blocklist_m
blocklist_mh_index_add / Add index in block list for metric hosts	/services/trackme/v1/blocklist/blocklist_m
blocklist_mh_metric_category_add / Add metric_category in block list for metric hosts	/services/trackme/v1/blocklist/blocklist_m
blocklist_ds_host_del / Delete host in block list for data sources	/services/trackme/v1/blocklist/blocklist_ds
blocklist_ds_index_del / Delete index in block list for data sources	/services/trackme/v1/blocklist/blocklist_ds
blocklist_ds_sourcetype_del / Delete sourcetype in block list for data sources	/services/trackme/v1/blocklist/blocklist_ds
blocklist_ds_data_name_del / Delete data name in block list for data sources	/services/trackme/v1/blocklist/blocklist_ds
blocklist_dh_host_del / Delete host in block list for data hosts	/services/trackme/v1/blocklist/blocklist_dh
blocklist_dh_index_del / Delete index in block list for data hosts	/services/trackme/v1/blocklist/blocklist_dh
blocklist_dh_sourcetype_del / Delete sourcetype in block list for data hosts	/services/trackme/v1/blocklist/blocklist_dh
blocklist_mh_host_del / Delete host in block list for metric hosts	/services/trackme/v1/blocklist/blocklist_m
blocklist_mh_index_del / Delete index in block list for metric hosts	/services/trackme/v1/blocklist/blocklist_m
blocklist_mh_metric_category_del / Delete metric_category in block list for metric hosts	/services/trackme/v1/blocklist/blocklist_m

blocklist_ds_host / Get current block list for data sources (hosts)

This endpoint retrieves the current block list collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_ds_host
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_ds_host" mode="get"
```

JSON response: (full collection)

```
[
 {
  "data_host": "bad_host",
  "_user": "nobody",
  "data_blacklist_state": "true",
  "_time": "1607890641",
  "_key": "5fd676d1ba5afb1f305fe551"
 }
]
```

blocklist_ds_index / Get current block list for data sources (index)

This endpoint retrieves the current block list collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_ds_index
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_ds_index" mode="get"
```

JSON response: (full collection)

```
[
 {
  "_time": "1607898808",
  "data_blacklist_state": "true",
  "data_index": "bad_index",
  "_user": "nobody",
  "_key": "5fd696b8d2b7c008be57cb71"
 }
]
```

blocklist_ds_sourcetype / Get current block list for data sources (sourcetype)

This endpoint retrieves the current block list collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_ds_sourcetype
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_ds_sourcetype" mode="get"
```

JSON response: (full collection)

```
[
 {
  "data_sourcetype": "bad_sourcetype",
  "_user": "nobody",
  "data_blacklist_state": "true",
  "_time": "1607890661",
  "_key": "5fd676e5ba5afb1f305fe552"
 }
]
```

blocklist_ds_data_name / Get current block list for data names (data_name)

This endpoint retrieves the current block list collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_ds_data_name
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_ds_data_name" mode="get"
```

JSON response: (full collection)

```
[
  {
    "data_name": ".*cribl:splunk_reduce_metadata",
    "data_blacklist_state": "true",
    "_user": "nobody",
    "_key": "602e37bac436b3754709064b"
  }
]
```

blocklist_dh_host / Get current block list for data hosts (hosts)

This endpoint retrieves the current block list collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_dh_host
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_dh_host" mode="get"
```

JSON response:

```
[
 {
  "data_blacklist_state": "true",
  "_user": "nobody",
  "_key": "5fd67e17ba5afb743339de21",
  "_time": "1607892503",
  "data_host": "bad_host"
 }
]
```

blocklist_dh_index / Get current block list for data hosts (index)

This endpoint retrieves the current block list collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_dh_index
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_dh_index" mode="get"
```

JSON response:

```
[
 {
  "data_blacklist_state": "true",
  "data_index": "bad_index",
  "_key": "5fd67e1fba5afb77831e3d51",
  "_time": "1607892511",
  "_user": "nobody"
 }
]
```

blocklist_dh_sourcetype / Get current block list for data hosts (sourcetype)

This endpoint retrieves the current block list collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_dh_sourcetype
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_dh_sourcetype" mode="get"
```

JSON response:

```
[
 {
  "data_blacklist_state": "true",
  "_user": "nobody",
  "_key": "5fd67e2dba5afb743339de22",
  "_time": "1607892525",
  "data_sourcetype": "bad_sourcetype"
 }
]
```

blocklist_mh_host / Get current block list for metric hosts (host)

This endpoint retrieves the current block list collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_mh_host
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_mh_host" mode="get"
```

JSON response:

```
[
 {
  "metric_blacklist_state": "true",
```

(continues on next page)

(continued from previous page)

```
    "_key": "5fd67f35ba5afb8386035111",
    "_time": "1607892789",
    "metric_host": "bad_host",
    "_user": "nobody"
  }
]
```

blocklist_mh_index / Get current block list for metric hosts (index)

This endpoint retrieves the current block list collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_mh_index
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_mh_index" mode="get"
```

JSON response:

```
[
  {
    "metric_blacklist_state": "true",
    "_user": "nobody",
    "_key": "5fd67f3dba5afb8b17532b11",
    "_time": "1607892797",
    "metric_index": "bad_index"
  }
]
```

blocklist_mh_metric_category / Get current block list for metric hosts (metric_category)

This endpoint retrieves the current block list collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_mh_metric_category
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_mh_metric_category" mode="get"
```

JSON response:

```
[
  {
    "metric_blacklist_state": "true",
    "_key": "5fd67f48ba5afb8386035112",
    "metric_category": "docker_container_status",
```

(continues on next page)

(continued from previous page)

```
  "_time": "1607892808",
  "_user": "nobody"
 }
]
```

blocklist_ds_host_add / Add host in block list for data sources

This endpoint adds a new record returned as a JSON array, it requires a POST call with no data required:

- "data_host": value to be added to the blocklist, accepts wildcards and regular expressions
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_ds_host_add -d '{"data_host": "bad_host2", "update_comment":
↪"Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_ds_host_add" mode="post" body=
↪"{\"data_host\": \"bad_host2\", \"update_comment\": \"Updated by automation.\"}"
```

JSON response:

```
[
 {
  "data_host": "bad_host2",
  "data_blacklist_state": "true",
  "_user": "nobody",
  "_key": "5fd6997291a48072a339d0bb"
 }
]
```

blocklist_ds_index_add / Add index in block list for data sources

This endpoint adds a new record returned as a JSON array, it requires a POST call with no data required:

- "data_index": value to be added to the blocklist, accepts wildcards and regular expressions
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_ds_index_add -d '{"data_index": "bad_index2", "update_comment":
↪"Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_ds_index_add" mode="post"␣
↪body="{\"data_index\": \"bad_index2\", \"update_comment\": \"Updated by automation.\
↪"}"
```

JSON response:

```
[
 {
  "data_index": "bad_index2",
  "data_blacklist_state": "true",
  "_user": "nobody",
  "_key": "5fd699b991a48072a339d0bd"
 }
]
```

blocklist_ds_sourcetype_add / Add sourcetype in block list for data sources

This endpoint adds a new record returned as a JSON array, it requires a POST call with no data required:

- `"data_sourcetype": value to be added to the blocklist, accepts wildcards and regular expressions`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_ds_sourcetype_add -d '{"data_sourcetype": "bad_sourcetype2",
↪"update_comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_ds_sourcetype_add" mode="post
↪" body="{\"data_sourcetype\": \"bad_sourcetype2\", \"update_comment\": \"Updated by␣
↪automation.\"}"
```

JSON response:

```
[
 {
  "data_sourcetype": "bad_sourcetype2",
  "data_blacklist_state": "true",
  "_user": "nobody",
  "_key": "5fd69d8b91a48072a339d0bf"
 }
]
```

blocklist_ds_data_name_add / Add data name in block list for data sources

This endpoint adds a new record returned as a JSON array, it requires a POST call with no data required:

- `"data_name": value to be added to the blocklist, accepts wildcards and regular expressions`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_ds_data_name_add -d '{"data_name": ".*cribl:splunk_reduce_
↪metadata", "update_comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_ds_data_name_add" mode="post"␣
↪body="{\"data_name\": \".*cribl:splunk_reduce_metadata\", \"update_comment\": \
↪"Updated by automation.\"}"
```

JSON response:

```
{
"_time": "1613617055",
"data_blacklist_state": "true",
"data_name": ".*cribl:splunk_reduce_metadata",
"_user": "nobody",
"_key": "602dd79f6305d730c367c461"
}
```

blocklist_dh_host_add / Add host in block list for data hosts

This endpoint adds a new record returned as a JSON array, it requires a POST call with no data required:

- `"data_host": value to be added to the blocklist, accepts wildcards and regular expressions`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_dh_host_add -d '{"data_host": "bad_host2", "update_comment":
↪"Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_dh_host_add" mode="post" body=
↪"{\"data_host\": \"bad_host2\", \"update_comment\": \"Updated by automation.\"}"
```

JSON response:

```
[
 {
  "data_host": "bad_host2",
  "data_blacklist_state": "true",
  "_user": "nobody",
  "_key": "5fd69def91a48072a339d0c1"
 }
]
```

blocklist_dh_index_add / Add index in block list for data hosts

This endpoint adds a new record returned as a JSON array, it requires a POST call with no data required:

- "data_index": value to be added to the blocklist, accepts wildcards and regular expressions
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_dh_index_add -d '{"data_index": "bad_index2", "update_comment":
↪"Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_dh_index_add" mode="post"␣
↪body="{\"data_index\": \"bad_index2\", \"update_comment\": \"Updated by automation.\
↪"}"
```

JSON response:

```
[
 {
  "data_index": "bad_index2",
  "data_blacklist_state": "true",
  "_user": "nobody",
  "_key": "5fd69e2d91a48072a339d0c3"
 }
]
```

blocklist_dh_sourcetype_add / Add sourcetype in block list for data hosts

This endpoint adds a new record returned as a JSON array, it requires a POST call with no data required:

- "data_sourcetype": value to be added to the blocklist, accepts wildcards and regular expressions
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_dh_sourcetype_add -d '{"data_sourcetype": "bad_sourcetype2",
↪"update_comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_dh_sourcetype_add" mode="post
↪" body="{\"data_sourcetype\": \"bad_sourcetype2\", \"update_comment\": \"Updated by␣
↪automation.\"}"
```

JSON response:

```
[
 {
  "data_sourcetype": "bad_sourcetype2",
  "data_blacklist_state": "true",
  "_user": "nobody",
```

(continues on next page)

(continued from previous page)

```
  "_key": "5fd69ee291a48072a339d0c5"
 }
]
```

blocklist_mh_host_add / Add host in block list for metric hosts

This endpoint adds a new record returned as a JSON array, it requires a POST call with no data required:

- `"metric_host": value to be added to the blocklist, accepts wildcards and regular expressions`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_mh_host_add -d '{"metric_host": "bad_host2", "update_comment":
↪"Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_mh_host_add" mode="post" body=
↪"{\"metric_host\": \"bad_host2\", \"update_comment\": \"Updated by automation.\"}"
```

JSON response:

```
[
 {
  "metric_host": "bad_host2",
  "data_blacklist_state": "true",
  "_user": "nobody",
  "_key": "5fd69f4a91a48072a339d0c7"
 }
]
```

blocklist_mh_index_add / Add index in block list for metric hosts

This endpoint adds a new record returned as a JSON array, it requires a POST call with no data required:

- `"metric_index": value to be added to the blocklist, accepts wildcards and regular expressions`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_mh_index_add -d '{"metric_index": "bad_index2", "update_comment
↪": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_mh_index_add" mode="post"␣
↪body="{\"metric_index\": \"bad_index2\", \"update_comment\": \"Updated by␣
↪automation.\"}"
```

(continues on next page)

(continued from previous page)

JSON response:

```
[
 {
  "metric_index": "bad_index2",
  "data_blacklist_state": "true",
  "_user": "nobody",
  "_key": "5fd6a34c91a48072a339d0c9"
 }
]
```

blocklist_mh_metric_category_add / Add metric_category in block list for metric hosts

This endpoint adds a new record returned as a JSON array, it requires a POST call with no data required:

- "metric_category": value to be added to the blocklist, accepts wildcards and regular expressions
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_mh_metric_category_add -d '{"metric_category": "bad_metric",
↪"update_comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_mh_metric_category_add" mode=
↪"post" body="{\"metric_category\": \"bad_metric\", \"update_comment\": \"Updated by␣
↪automation.\"}"
```

JSON response:

```
[
 {
  "metric_category": "bad_metric",
  "data_blacklist_state": "true",
  "_user": "nobody",
  "_key": "5fd6a3e091a48072a339d0cc"
 }
]
```

blocklist_ds_host_del / Delete host in block list for data sources

This endpoint deletes an existing record returned as a JSON array, it requires a DELETE call with the following arguments:

- "data_host": value to be removed from the collection
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_ds_host_del -d '{"data_host": "bad_host2", "update_comment":
↪"Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_ds_host_del" mode="delete"␣
↪body="{\"data_host\": \"bad_host2\", \"update_comment\": \"Updated by automation.\"}
↪"
```

response:

```
Record with _key 5fd6997291a48072a339d0bb was deleted from the collection.
```

blocklist_ds_index_del / Delete index in block list for data sources

This endpoint deletes an existing record returned as a JSON array, it requires a DELETE call with the following arguments:

- `"data_index": value to be removed from the collection`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_ds_index_del -d '{"data_index": "bad_index2", "update_comment":
↪"Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_ds_index_del" mode="delete"␣
↪body="{\"data_index\": \"bad_index2\", \"update_comment\": \"Updated by automation.\
↪"}"
```

response:

```
Record with _key 5fd699b991a48072a339d0bd was deleted from the collection.
```

blocklist_ds_sourcetype_del / Delete sourcetype in block list for data sources

This endpoint deletes an existing record returned as a JSON array, it requires a DELETE call with the following arguments:

- `"data_sourcetype": value to be removed from the collection`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_ds_sourcetype_del -d '{"data_sourcetype": "bad_sourcetype2",
↪"update_comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_ds_sourcetype_del" mode=
↪"delete" body="{\"data_sourcetype\": \"bad_sourcetype2\", \"update_comment\": \
↪"Updated by automation.\"}"
```

response:

```
Record with _key 5fd69d8b91a48072a339d0bf was deleted from the collection.
```

blocklist_ds_data_name_del / Delete data name in block list for data sources

This endpoint deletes an existing record returned as a JSON array, it requires a DELETE call with the following arguments:

- `"data_name": value to be removed from the collection`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_ds_data_name_del -d '{"data_name": ".*cribl:splunk_reduce_
↪metadata", "update_comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_ds_data_name_del" mode="delete
↪" body="{\"data_name\": \".*cribl:splunk_reduce_metadata\", \"update_comment\": \
↪"Updated by automation.\"}"
```

response:

```
Record with _key 602e37bac436b3754709064b was deleted from the collection.
```

blocklist_dh_host_del / Delete host in block list for data hosts

This endpoint deletes an existing record returned as a JSON array, it requires a DELETE call with the following arguments:

- `"data_host": value to be removed from the collection`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_dh_host_del -d '{"data_host": "bad_host2", "update_comment":
↪"Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_dh_host_del" mode="delete"␣
↪body="{\"data_host\": \"bad_host2\", \"update_comment\": \"Updated by automation.\"}
↪"
```

response:

```
Record with _key 5fd69def91a48072a339d0c1 was deleted from the collection.
```

blocklist_dh_index_del / Delete index in block list for data hosts

This endpoint deletes an existing record returned as a JSON array, it requires a DELETE call with the following arguments:

- `"data_index": value to be removed from the collection`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_dh_index_del -d '{"data_index": "bad_index2", "update_comment":
↪"Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_dh_index_del" mode="delete"␣
↪body="{\"data_index\": \"bad_index2\", \"update_comment\": \"Updated by automation.\
↪"}"
```

response:

```
Record with _key 5fd69e2d91a48072a339d0c3 was deleted from the collection.
```

blocklist_dh_sourcetype_del / Delete sourcetype in block list for data hosts

This endpoint deletes an existing record returned as a JSON array, it requires a DELETE call with the following arguments:

- `"data_sourcetype": value to be removed from the collection`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_dh_sourcetype_del -d '{"data_sourcetype": "bad_sourcetype2",
↪"update_comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_dh_sourcetype_del" mode=
↪"delete" body="{\"data_sourcetype\": \"bad_sourcetype2\", \"update_comment\": \
↪"Updated by automation.\"}"
```

response:

```
Record with _key 5fd69ee291a48072a339d0c5 was deleted from the collection.
```

blocklist_mh_host_del / Delete host in block list for metric hosts

This endpoint deletes an existing record returned as a JSON array, it requires a DELETE call with the following arguments:

- `"metric_host": value to be removed from the collection`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_mh_host_del -d '{"metric_host": "bad_host2", "update_comment":
↪"Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_mh_host_del" mode="delete"␣
↪body="{\"metric_host\": \"bad_host2\", \"update_comment\": \"Updated by automation.\
↪"}"
```

response:

```
Record with _key 5fd69f4a91a48072a339d0c7 was deleted from the collection.
```

blocklist_mh_index_del / Delete index in block list for metric hosts

This endpoint deletes an existing record returned as a JSON array, it requires a DELETE call with the following arguments:

- `"metric_index": value to be removed from the collection`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_mh_index_del -d '{"metric_index": "bad_index2", "update_comment
↪": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_mh_index_del" mode="delete"␣
↪body="{\"metric_index\": \"bad_index2\", \"update_comment\": \"Updated by␣
↪automation.\"}"
```

response:

```
Record with _key 5fd6ae4c1814da1e704d47a3 was deleted from the collection.
```

blocklist_mh_metric_category_del / Delete metric_category in block list for metric hosts

This endpoint deletes an existing record returned as a JSON array, it requires a DELETE call with the following arguments:

- `"metric_category": value to be removed from the collection`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪blocklist/blocklist_mh_metric_category_del -d '{"metric_category": "bad_metric",
↪"update_comment": "Updated by automation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/blocklist/blocklist_mh_metric_category_del" mode=
↪"delete" body="{\"metric_category\": \"bad_metric\", \"update_comment\": \"Updated␣
↪by automation.\"}"
```

response:

```
Record with _key 5fd6afee8c70e663460209c5 was deleted from the collection.
```

3.5.13 Logical Groups endpoints

Resources summary:

Resource	API Path
logical_groups_collection / Get entire logical groups collection	/services/trackme/v1/logical_groups/logical_groups_collection
logical_groups_get_grp / Get a logical group	/services/trackme/v1/logical_groups/logical_groups_get_grp
logical_groups_add_grp / Add a new or update a logical group	/services/trackme/v1/logical_groups/logical_groups_add_grp
logical_groups_del_grp / Delete a logical group	/services/trackme/v1/logical_groups/logical_groups_del_grp
logical_groups_associate_group / Associate an object with an existing logical group	/services/trackme/v1/logical_groups/logical_groups_associate_grou
logical_groups_unassociate / Unassociate an object from any logical group it is member of	/services/trackme/v1/logical_groups/logical_groups_unassociate

logical_groups_collection / Get entire logical groups collection

This endpoint retrieves the entire Logical Groups collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/logical_
↪groups/logical_groups_collection
```

SPL query:

```
| trackme url="/services/trackme/v1/logical_groups/logical_groups_collection" mode=
↪"get"
```

JSON response:

```
[
 {
  "object_group_name": "logical group example",
  "object_group_members": [
  [
      "telegraf-node1",
      "telegraf-node2"
  ]
 ],
 "object_group_min_green_percent": "50",
 "object_group_mtime": "1608481445.3048441",
 "_user": "nobody",
 "_key": "5fdf7aa55af72855ab693b47"
 }
]
```

logical_groups_get_grp / Get a logical group

This endpoint retrieve a specific logical group record, it requires a GET call with the following information:

- `"object_group_name": name of the logical group`

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/logical_
↪groups/logical_groups_get_grp -d '{"object_group_name": "logical group example"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/logical_groups/logical_groups_get_grp" mode="get"␣
↪body="{\"object_group_name\": \"logical group example\"}"
```

JSON response:

```
[
 {
  "object_group_name": "logical group example",
  "object_group_members": [
  [
      "telegraf-node1",
      "telegraf-node2"
  ]
 ],
 "object_group_min_green_percent": "50",
 "object_group_mtime": "1608481445.3048441",
 "_user": "nobody",
 "_key": "5fdf7aa55af72855ab693b47"
 }
]
```

logical_groups_add_grp / Add a new or update a logical group

This endpoint creates a new logical group, it requires a POST call with the following data required:

- `"object_group_name": name of the logical group to be created`

- `"object_group_members": comma separated list of the group members`
- `"object_group_min_green_percent":`

OPTIONAL: minimal percentage of hosts that need to be green for the logical group to be green, if unset defaults to 50. Recommended options for this value: 12.5 / 33.33 / 50

- `"update_comment":`

OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

If the logical group exists already, it will be updated with the information provided.

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪logical_groups/logical_groups_add_grp -d '{"object_group_name": "logical group␣
↪example", "object_group_members": "telegraf-node1, telegraf-node2", "object_group_
↪min_green_percent": "50", "comment_update": "Automated API driven logical group␣
↪creation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/logical_groups/logical_groups_add_grp" mode="post
↪" body="{\"object_group_name\": \"logical group example\", \"object_group_members\
↪": \"telegraf-node1, telegraf-node2\", \"object_group_min_green_percent\": \"50\", \
↪"comment_update\": \"Automated API driven logical group creation.\"}"
```

JSON response:

```
[
 {
  "object_group_name": "logical group example",
  "object_group_members": [
 [
     "telegraf-node1",
     "telegraf-node2"
 ]
 ],
 "object_group_min_green_percent": "50",
 "object_group_mtime": "1608481445.3048441",
 "_user": "nobody",
 "_key": "5fdf7aa55af72855ab693b47"
 }
]
```

logical_groups_del_grp / Delete a logical group

This endpoint deletes a logical group, it requires a DELETE call with the following data required:

- `"object_group_name": name of the logical group to be removed`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪logical_groups/logical_groups_del_grp -d '{"object_group_name": "logical group␣
↪example", "comment_update": "Automated API driven logical group deletion."}'
```

(continues on next page)

(continued from previous page)

SPL query:

```
| trackme url="/services/trackme/v1/logical_groups/logical_groups_del_grp" mode=
↪"delete" body="{\"object_group_name\": \"logical group example\", \"comment_update\
↪": \"Automated API driven logical group deletion.\"}"
```

response:

```
Record with _key 5fdf7aa55af72855ab693b47 was deleted from the logical groups␣
↪collection.
```

logical_groups_associate_group / Associate an object with an existing logical group

This endpoint associates an object (data host or metric host) with an existing logical group (existing members of the logical groups are preserved and this object membership will be removed), it requires a POST call with the following data required:

- `"object": the name of the data host or the metric host`
- `"key": the KVstore unique key of the logical group`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪logical_groups/logical_groups_associate_group -d '{"object": "telegraf-node3", "key
↪": "604356885ea0f10084356707", "comment_update": "Automated API driven logical␣
↪group creation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/logical_groups/logical_groups_associate_group"␣
↪mode="post" body="{\"object\": \"telegraf-node3\", \"key\": \
↪"604356885ea0f10084356707\", \"comment_update\": \"Automated API driven logical␣
↪group creation.\"}"
```

response:

```
{
  "object_group_name": "logical group example",
  "object_group_members": [
  "telegraf-node1",
  "telegraf-node2",
  "telegraf-node3"
],
  "object_group_min_green_percent": "50",
  "object_group_mtime": "1615025866.585574",
  "_user": "nobody",
  "_key": "604356885ea0f10084356707"
}
```

logical_groups_unassociate / Unassociate an object from any logical group it is member of

This endpoint unassociates an object (data host or metric host) from a logical group it is member of (existing associations of the logical groups are preserved), it requires a POST call with the following data required:

- `"object": the object name (data host or metric host) to remove association for`
- `"key": the KVstore unique key of the logical group`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪logical_groups/logical_groups_unassociate -d '{"object": "telegraf-node3", "key":
↪"6043a23b33d53e70d86fc091", "comment_update": "Automated API driven logical group␣
↪update."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/logical_groups/logical_groups_unassociate" mode=
↪"post" body="{\"object\": \"telegraf-node3\", \"key\": \"6043a23b33d53e70d86fc091\",
↪ \"comment_update\": \"Automated API driven logical group update.\"}"
```

response:

```
{
    "response": "object telegraf-node3 has been unassociated from logical group␣
↪record key: 604356885ea0f10084356707"
}
```

3.5.14 Data Sampling endpoints

Resources summary:

Resource	API Path
data_sampling_collection / Get Data sampling collection	/services/trackme/v1/data_sampling/data_sampling_collection
data_sampling_by_name / Get Data sampling record by data source	/services/trackme/v1/data_sampling/data_sampling_by_name
data_sampling_del / Delete a data sampling record for a given data source	/services/trackme/v1/data_sampling/data_sampling_del
data_sampling_reset / Reset and run data sampling for a given data source	/services/trackme/v1/data_sampling/data_sampling_reset

data_sampling_collection / Get Data sampling collection

This endpoint retrieves the data sampling collection, it requires a GET call with no options required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/data_
↪sampling/data_sampling_collection
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sampling/data_sampling_collection" mode="get"
```

JSON response:

```
[
 {
  "current_detected_format": "syslog_no_timestamp",
  "current_detected_format_dcount": "1",
  "current_detected_format_id": "d01bcd8d79beb285c118872c7c039bd6",
  "data_name": "linux_emea:linux_secure",
  "data_sample_anomaly_ack_mtime": "N/A",
  ...
```

data_sampling_by_name / Get Data sampling record by data source

This endpoint retrieves a data sampling record, it requires a GET call with the following data:

- `"data_name": name of the data source`

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/data_
↪sampling/data_sampling_by_name -d '{"data_name": "main:retail_transaction"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sampling/data_sampling_collection" mode="get
↪" -body "{\"data_name\": \"main:retail_transaction\"}"
```

JSON response:

```
{
 "current_detected_format": [
  "PII",
  "raw_start_by_timestamp %a %d %b %Y %H:%M:%S"
  ],
 "current_detected_format_dcount": "2",
 "current_detected_format_id": [
 "7b5eb471694ac78273e516b7e3fb78c9",
 "84fb236745d5ed942ed495037b8187e8"
 ],
 ...
```

data_sampling_del / Delete a data sampling record for a given data source

This endpoint deletes a data sampling record for a given data source, it requires a DELETE call with the following data:

- `"data_name": name of the data source`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/data_
↪sampling/data_sampling_del -d '{"data_name": "main:retail_transaction", "comment_
↪update": "Automated API driven deletion."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sampling/data_sampling_del" mode="delete"␣
↪body="{\"data_name\": \"main:retail_transaction\", \"comment_update\": \"Automated␣
↪API driven deletion.\"}"
```

response:

```
Record with _key ab994e3b00751d45591c7abc2b7a1061 was deleted from the collection.
```

data_sampling_reset / Reset and run data sampling for a given data source

This endpoint clears the data sampling record state and runs the sampling operation for a given data source, it requires a POST call with the following data:

- `"data_name": name of the data source`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪sampling/data_sampling_reset -d '{"data_name": "main:retail_transaction", "comment_
↪update": "Automated API driven deletion."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sampling/data_sampling_reset" mode="post"␣
↪body="{\"data_name\": \"main:retail_transaction\", \"comment_update\": \"Automated␣
↪API driven deletion.\"}"
```

response:

```
Data sampling state for: main:sample9-customformat was cleared and sampling operation␣
↪ran, data sampling state is: green
```

3.5.15 Data Sampling models endpoints

Resources summary:

Resource	API Path
data_sampling_models / Get data sampling custom models	/services/trackme/v1/data_sampling/data_sampling_models
data_sampling_models_by_name / Get data sampling custom model by name	/services/trackme/v1/data_sampling/data_sampling_models_by_name
data_sampling_models_add / Add a new custom model or update	/services/trackme/v1/data_sampling/data_sampling_models_add
data_sampling_models_del / Delete a custom model	/services/trackme/v1/data_sampling/data_sampling_models_del

data_sampling_models / Get data sampling custom models

This endpoint retrieves the data sampling custom models collection, it requires a GET call with no options required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/data_
↪sampling/data_sampling_models
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sampling/data_sampling_models" mode="get"
```

JSON response:

```
[
 {
  "model_name": "Example format",
  "model_regex": "^\\{\"extraData\":",
  "model_type": "inclusive",
  "model_id": "4c46a2fe5f07006e456bf9b659c7ce7d",
  "sourcetype_scope": "sample9-customformat",
  "mtime": 1609073607143,
  "_user": "nobody",
  "_key": "5fe883c7fdf8f9160636c132"
 }
]
```

data_sampling_models_by_name / Get data sampling custom model by name

This endpoint retrieves a data sampling custom model collection, it requires a GET call with the following data:

- `"model_name": name of the custom model`

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/data_
↪sampling/data_sampling_models_by_name -d '{"model_name": "Example format"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sampling/data_sampling_models_by_name" mode=
↪"get" body="{\"model_name\": \"Example format\"}"
```

JSON response:

```
{
 "model_name": "Example format",
 "model_regex": "^\\{\"extraData\":",
 "model_type": "inclusive",
 "model_id": "4c46a2fe5f07006e456bf9b659c7ce7d",
 "sourcetype_scope": "sample9-customformat",
 "mtime": 1609073607143,
 "_user": "nobody",
 "_key": "5fe883c7fdf8f9160636c132"
}
```

data_sampling_models_add / Add a new custom model or update

This endpoint creates a new data sampling custom model, it requires a POST call with the following data:

- `"model_name": name of the custom model`
- `"model_regex":`

The regular expression to be used by the custom model, special characters should be escaped.

- `"model_type":`

The type of match for this model, valid options are “inclusive” (rule must match) and “exclusive” (rule must not match)

- `"sourcetype_scope":`

OPTIONAL: value of the sourcetype to match, if unset defaults to “*”. You can enter a list of sourcetypes as a comma separated list of values, wilcards and spaces should not be used.

- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

Note: if a custom model referenced under the same name exists already, it will be updated using the information provided.

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/data_
↪sampling/data_sampling_models_add -d '{"model_name": "Example format", "model_type
↪": "inclusive", "model_regex": "^\\{\"extraData\":", "sourcetype_scope": "sample9-
↪customformat", "comment_update": "Automated API driven creation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sampling/data_sampling_models_add" mode="post
↪" body="{\"model_name\": \"Example format\", \"model_type\": \"inclusive\", \"model_
↪regex\": \"^\\\\{\\\"extraData\\\":\", \"sourcetype_scope\": \"sample9-customformat\
↪", \"comment_update\": \"Automated API driven creation.\"}"
```

JSON response:

```
[
 {
  "model_name": "Example format",
  "model_regex": "^\\{\"extraData\":",
  "model_type": "inclusive",
  "model_id": "4c46a2fe5f07006e456bf9b659c7ce7d",
  "sourcetype_scope": "sample9-customformat",
  "mtime": 1609073607143,
  "_user": "nobody",
  "_key": "5fe883c7fdf8f9160636c132"
 }
]
```

data_sampling_models_del / Delete a custom model

This endpoint deletes a custom data sampling model, it requires a DELETE call with the following data:

- `"model_name": name of the custom model`

- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/data_
↪sampling/data_sampling_models_del -d '{"model_name": "Example format", "comment_
↪update": "Automated API driven deletion."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/data_sampling/data_sampling_models_del" mode=
↪"delete" body="{\"model_name\": \"Example format\", \"comment_update\": \"Automated␣
↪API driven deletion.\"}"
```

response:

```
Record with _key 5fe883c7fdf8f9160636c132 was deleted from the collection.
```

3.5.16 Tag policies endpoints

Resources summary:

Resource	API Path
tag_policies / Get tag policies	/services/trackme/v1/tag_policies/tag_policies
tag_policies_by_id / Get tag policy by id	/services/trackme/v1/tag_policies/tag_policies_by_id
tag_policies_add / Add a new tag policy or update	/services/trackme/v1/tag_policies/tag_policies_add
tag_policies_del / Delete a tag policy	/services/trackme/v1/tag_policies/tag_policies_del

tag_policies / Get tag policies

This endpoint retrieves the tag policies collection, it requires a GET call with no options required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/tag_
↪policies/tag_policies
```

SPL query:

```
| trackme url="/services/trackme/v1/tag_policies/tag_policies" mode="get"
```

JSON response:

```
[
 {
  "_time": "1608597719",
  "mtime": "1608597718",
  "tags_policy_id": "Example policy",
  "tags_policy_regex": "linux_*",
  "tags_policy_value": "OS,Linux",
  "_user": "nobody",
  "_key": "5fe140d77f1e835045091651"
 }
]
```

tag_policies_by_id / Get tag policy by id

This endpoint retrieves a tag policy by its id, it requires a GET call with the following data:

- "tags_policy_id": ID of the tags policy

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/tag_
↪policies/tag_policies_by_id -d '{"tags_policy_id": "Example policy"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/tag_policies/tag_policies_by_id" mode="get" body="
↪{\"tags_policy_id\": \"Example policy\"}"
```

JSON response:

```
{
 "_time": "1608597719",
 "mtime": "1608597718",
 "tags_policy_id": "Example policy",
 "tags_policy_regex": "linux_*",
 "tags_policy_value": "OS,Linux",
 "_user": "nobody",
 "_key": "5fe140d77f1e835045091651"
}
```

tag_policies_add / Add a new tag policy or update

This endpoint creates a new tag policy, it requires a POST call with the following data:

- "tags_policy_id": ID of the tag policy
- "tags_policy_regex": The regular expression to be used by the tags policy, special characters should be escaped.
- "tags_policy_value": List of tags to be applied as a comma separated list of values
- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

Note: if a tag policy referenced with the same ID exists already, it will be updated using the information provided.

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/tag_
↪policies/tag_policies_add -d '{"tags_policy_id": "Example policy", "tags_policy_
↪regex": "linux_*", "tags_policy_value": "OS,Linux", "comment_update": "Automated␣
↪API driven creation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/tag_policies/tag_policies_add" mode="post" body="
↪{\"tags_policy_id\": \"Example policy\", \"tags_policy_regex\": \"linux_*\", \"tags_
↪policy_value\": \"OS,Linux\", \"comment_update\": \"Automated API driven creation.\
↪"}"
```

JSON response:

```
{
 "tags_policy_id": "Example policy",
 "tags_policy_value": "OS,Linux",
 "tags_policy_regex": "linux_*",
 "mtime": 1608598325220,
 "_user": "nobody",
 "_key": "5fe140d77f1e835045091651"
}
```

tag_policies_del / Delete a tag policy

This endpoint deletes a tag policy, it requires a DELETE call with the following data:

- `"tags_policy_id": ID of the tag policy`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/tag_
↪policies/tag_policies_del -d '{"tags_policy_id": "Example policy", "comment_update
↪": "Automated API driven deletion."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/tag_policies/tag_policies_del" mode="delete" body=
↪"{\"tags_policy_id\": \"Example policy\", \"comment_update\": \"Automated API␣
↪driven deletion.\"}"
```

response:

```
Record with _key 5fe140d77f1e835045091651 was deleted from the collection.
```

3.5.17 Lagging classes endpoints

Resources summary:

Resource	API Path
lagging_classes / Get lagging classes	/services/trackme/v1/lagging_classes/lagging_classes
lagging_classes_by_name / Get lagging class by name	/services/trackme/v1/lagging_classes/lagging_classes_by_name
lagging_classes_add / Add a new lagging class or update	/services/trackme/v1/lagging_classes/lagging_classes_add
lagging_classes_del / Delete a lagging class	/services/trackme/v1/lagging_classes/lagging_classes_del

lagging_classes / Get lagging classes

This endpoint retrieves the lagging classes collection, it requires a GET call with no options required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/lagging_
↪classes/lagging_classes
```

SPL query:

```
| trackme url="/services/trackme/v1/lagging_classes/lagging_classes" mode="get"
```

JSON response:

```
[
 {
  "name": "pan:traffic",
  "level": "sourcetype",
  "object": "data_source",
  "value": "900",
  "_user": "nobody",
  "_key": "5fe2936d1a568f12a114995a"
 }
]
```

lagging_classes_by_name / Get lagging class by name

This endpoint retrieves a lagging class by its name, it requires a GET call with the following data:

- `"name": name of the lagging class`

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/lagging_
↪classes/lagging_classes_by_name -d '{"name": "pan:traffic"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/lagging_classes/lagging_classes_by_name" mode="get
↪" body="{\"name\": \"pan:traffic\"}"
```

JSON response:

```
{
 "name": "pan:traffic",
 "level": "sourcetype",
 "object": "data_source",
 "value": "900",
 "_user": "nobody",
 "_key": "5fe2936d1a568f12a114995a"
}
```

lagging_classes_add / Add a new lagging class or update

This endpoint creates a new tag policy, it requires a POST call with the following data:

- `"name": name of the lagging class`
- `"level": which level the lagging class is based on, valid otions are: sourcetype / index / priority`

- `"object": which type of objects the lagging class is applied to, valid options are: data_source / data_host / all`
- `"value": the lagging value in seconds, an integer is expected`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

Note: if a lagging class referenced under the same name exists already, it will be updated using the information provided.

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪lagging_classes/lagging_classes_add -d '{"name": "pan:traffic", "level": "sourcetype
↪", "object": "data_source", "value": "900", "comment_update": "Automated API driven␣
↪creation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/lagging_classes/lagging_classes_add" mode="post"␣
↪body="{\"name\": \"pan:traffic\", \"level\": \"sourcetype\", \"object\": \"data_
↪source\", \"value\": \"900\", \"comment_update\": \"Automated API driven creation.\
↪"}"
```

JSON response:

```
[
 {
  "name": "pan:traffic",
  "level": "sourcetype",
  "object": "data_source",
  "value": "900",
  "_user": "nobody",
  "_key": "5fe2936d1a568f12a114995a"
 }
]
```

lagging_classes_del / Delete a lagging class

This endpoint deletes a tag policy, it requires a DELETE call with the following data:

- `"name": name of the lagging class`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪lagging_classes/lagging_classes_del -d '{"name": "pan:traffic", "comment_update":
↪"Automated API driven deletion."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/lagging_classes/lagging_classes_del" mode="delete
↪" body="{\"name\": \"pan:traffic\", \"comment_update\": \"Automated API driven␣
↪deletion.\"}"
```

response:

```
Record with _key 5fe28130efc3a55870259041 was deleted from the collection.
```

3.5.18 Lagging classes metrics endpoints

Resources summary:

Resource	API Path
lagging_classes_metrics / Get lagging classes	/services/trackme/v1/lagging_classes_metrics/lagging_classes_metrics
lagging_classes_metrics_by_name / Get lagging class by name	/services/trackme/v1/lagging_classes_metrics/lagging_classes_metrics_by_name
lagging_classes_metrics_add / Add a new lagging class or update	/services/trackme/v1/lagging_classes_metrics/lagging_classes_metrics_add
lagging_classes_metrics_del / Delete a lagging class	/services/trackme/v1/lagging_classes_metrics/lagging_classes_metrics_del

lagging_classes_metrics / Get lagging classes

This endpoint retrieves the lagging classes collection, it requires a GET call with no options required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/lagging_
↪classes_metrics/lagging_classes_metrics
```

SPL query:

```
| trackme url="/services/trackme/v1/lagging_classes_metrics/lagging_classes_metrics"␣
↪mode="get"
```

JSON response:

```
[
 {
  "metric_category": "docker",
  "metric_max_lag_allowed": "900",
  "_user": "nobody",
  "_key": "5fe2928b1a568f12a1149957"
 }
]
```

lagging_classes_metrics_by_name / Get lagging class by name

This endpoint retrieves a lagging class by its name, it requires a GET call with the following data:

- `"metric_category": name of the metric category`

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/lagging_
↪classes_metrics/lagging_classes_metrics_by_name -d '{"metric_category": "docker"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/lagging_classes_metrics/lagging_classes_metrics_
↪by_name" mode="get" body="{\"metric_category\": \"docker\"}"
```

JSON response:

```
{
 "metric_category": "docker",
 "metric_max_lag_allowed": "900",
 "_user": "nobody",
 "_key": "5fe2928b1a568f12a1149957"
}
```

lagging_classes_metrics_add / Add a new lagging class or update

This endpoint creates a new tag policy, it requires a POST call with the following data:

- `"metric_category": name of the metric category`
- `"metric_max_lag_allowed": the lagging value in seconds, an integer is expected`
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

Note: if a lagging class referenced under the same name exists already, it will be updated using the information provided.

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪lagging_classes_metrics/lagging_classes_metrics_add -d '{"metric_category": "docker
↪", "metric_max_lag_allowed": "900", "comment_update": "Automated API driven␣
↪creation."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/lagging_classes_metrics/lagging_classes_metrics_
↪add" mode="post" body="{\"metric_category\": \"docker\", \"metric_max_lag_allowed\
↪": \"900\", \"comment_update\": \"Automated API driven creation.\"}"
```

JSON response:

```
[
 {
  "metric_category": "docker",
  "metric_max_lag_allowed": "900",
  "_user": "nobody",
  "_key": "5fe2928b1a568f12a1149957"
 }
]
```

lagging_classes_metrics_del / Delete a lagging class

This endpoint deletes a tag policy, it requires a DELETE call with the following data:

- `"metric_category": name of the metric category`

- "update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪lagging_classes_metrics/lagging_classes_metrics_del -d '{"metric_category": "docker
↪", "comment_update": "Automated API driven deletion."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/lagging_classes_metrics/lagging_classes_metrics_
↪del" mode="delete" body="{\"metric_category\": \"docker\", \"comment_update\": \
↪"Automated API driven deletion.\"}"
```

response:

```
Record with _key 5fe2928b1a568f12a1149957 was deleted from the collection.
```

3.5.19 Smart Status endpoints

Resources summary:

Resource	API Path
ds_smart_status / Run Smart Status for a data source	/services/trackme/v1/smart_status/ds_smart_status
dh_smart_status / Run Smart Status for a data host	/services/trackme/v1/smart_status/dh_smart_status
mh_smart_status / Run Smart Status for a metric host	/services/trackme/v1/smart_status/mh_smart_status

ds_smart_status / Run Smart Status for a data source

This endpoints runs the smart status for a given data source, it requires a GET call with the following options:

- "data_name": name of the data source

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/smart_
↪status/ds_smart_status -d '{"data_name": "network:pan:traffic"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/smart_status/ds_smart_status" mode="get" body="{\
↪"data_name\": \"network:pan:traffic\"}"
```

JSON response:

```
{
  "data_name": "network:pan:traffic",
  "data_source_state": "red",
  "smart_result": "TrackMe triggered an alert on this data source due to outliers␣
↪detection in the event count, outliers are based on the calculation of a lower and␣
↪upper bound (if alerting on upper) determined against the data source usual␣
↪behaviour and outliers parameters. Review the correlation results to determine if␣
↪the behaviour is expected or symptomatic of an issue happening on the data source␣
↪(lost of sources or hosts, etc.) and proceed to any outliers configuration fine␣
↪tuning if necessary.",
```

(continues on next page)

(continued from previous page)

```
  "smart_code": "40",
  "correlation_outliers": "[ description: Last 24h outliers detection ], [␣
↪OutliersCount: 288 ], [ latest4hcount: 34560.00 ], [ lowerBound: 120000.00 ], [␣
↪upperBound: 92858.16 ], [ lastOutlier: Sat Jan 16 20:40:00 2021 ], [␣
↪OutlierAlertOnUpper: true ]",
  "correlation_flipping_state": "state: [ green ], message: [ There were no anomalies␣
↪detected in the flipping state activity threshold. ]",
  "correlation_data_sampling": "state: [ red ], message: [ WARNING: Anomalies were␣
↪detected in data sampling, a change with multiple event formats was detected on Fri␣
↪Jan 15 08:30:00 2021, review the format of the events and acknowledge the data␣
↪sampling alert if this format change was expected. Click on the button Manage data␣
↪sampling for more details. ]"
}
```

The API response depends on the smart status results.

dh_smart_status / Run Smart Status for a data host

This endpoints runs the smart status for a given data host, it requires a GET call with the following options:

- `"data_host": name of the data host`

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/smart_
↪status/dh_smart_status -d '{"data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/smart_status/dh_smart_status" mode="get" body="{\
↪"data_host\": \"FIREWALL.PAN.AMER.DESIGN.NODE1\"}"
```

JSON response:

```
{
"data_host": "FIREWALL.PAN.AMER.DESIGN.NODE1",
"data_host_state": "green",
"smart_result": "The data host is currently in a normal state, therefore further␣
↪investigations are not required at this stage.",
"smart_code": "0",
"correlation_flipping_state": "state: [ green ], message: [ There were no anomalies␣
↪detected in the flipping state activity threshold. ]"
}
```

The API response depends on the smart status results.

mh_smart_status / Run Smart Status for a metric host

This endpoints runs the smart status for a given data source, it requires a GET call with the following options:

- `"metric_host": name of the metric host`

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/smart_
↪status/mh_smart_status -d '{"metric_host": "telegraf-node1"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/smart_status/mh_smart_status" mode="get" body="{\
↪"metric_host\": \"telegraf-node1\"}"
```

JSON response:

```
{
"metric_host": "telegraf-node1",
"metric_host_state": "green",
"smart_result": "The metric host is currently in a normal state, therefore further␣
↪investigations are not required at this stage.",
"smart_code": "0",
"correlation_flipping_state": "state: [ green ], message: [ There were no anomalies␣
↪detected in the flipping state activity threshold. ]"
}
```

The API response depends on the smart status results.

3.5.20 Backup and Restore endpoints

Resources summary:

Resource	API Path
backup / Get backup archive files available	/services/trackme/v1/backup_and_restore/backup
backup / Run backup KVstore collections	/services/trackme/v1/backup_and_restore/backup
backup / Purge older backup archive files	/services/trackme/v1/backup_and_restore/backup
restore / Perform a restore of KVstore collections	/services/trackme/v1/backup_and_restore/restore

backup / Get backup archive files available

This endpoint lists all the backup files available on the search head, files are stored in the backup directory of the application, it requires a GET call with no arguments:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/backup_
↪and_restore/backup
```

SPL query:

```
| trackme url="/services/trackme/v1/backup_and_restore/backup" mode="get"
```

JSON response:

```
{"backup_files": "['/opt/splunk/etc/apps/trackme/backup/trackme-backup-20210205-
↪142635.tgz', '/opt/splunk/etc/apps/trackme/backup/trackme-backup-20210205-142607.tgz
↪']"}
```

backup / Run backup KVstore collections

This endpoint performs a backup of all TrackMe collections in a compressed tarball file stored in the backup directory of the application, it requires a POST call with no arguments:

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/backup_
↪and_restore/backup
```

SPL query:

```
| trackme url="/services/trackme/v1/backup_and_restore/backup" mode="post"
```

JSON response:

```
{ "backup_archive": "/opt/splunk/etc/apps/trackme/backup/trackme-backup-20210205-
↪142505.tgz", "report": "23 collections backed up / 5 collections empty"}
```

backup / Purge older backup archive files

This endpoint performs a purge of backup archive files older than x days, it requires a DELETE call with the following arguments:

- `retention_days: (integer) OPTIONAL: the maximal retention for backup archive files in days, if not specified defaults to 7 days`

External:

```
curl -k -u admin:'ch@ngeM3' -X DELETE https://localhost:8089/services/trackme/v1/
↪backup_and_restore/backup -d '{"metric_host": "telegraf-node1"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/backup_and_restore/backup" mode="delete" body="{\
↪"retention_days\": \"7\"}"
```

JSON response:

```
{"status": "There were no backup archive files older than 7 days to be purged"}
```

restore / Perform a restore of KVstore collections

This endpoint performs a backup of all TrackMe collections in compressed tarball file stored in the backup directory of the application, it requires a POST call with thre following arguments:

- `backup_archive:`

The archive file to be restoring from, 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)

- `target:`

(all / name of the KVstore json file) OPTIONAL: restore all available KVstore collection files (all) or choose a specific KVstore json file target to restore a unique collection. (default to all)

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/backup_
↪and_restore/backup -d '{"backup_archive": "trackme-backup-20210205-142635.tgz",
↪"dry_run": "false", "target": "all"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/backup_and_restore/restore" mode="post" body="{\
↪"backup_archive\": \"trackme-backup-20210205-142635.tgz\", \"dry_run\": \"false\", \
↪"target\": \"all\"}"
```

JSON response in dry_run: true:

```
{"response": "Success: the archive /opt/splunk/etc/apps/trackme/backup/trackme-backup-
↪20210205-142635.tgz could be successfully extracted, the following KVstore␣
↪collections can be restored (empty collections are not backed up)", "collections":
↪"['kv_trackme_data_source_monitoring_blacklist_sourcetype.json', 'kv_trackme_
↪maintenance_mode.json', 'kv_trackme_data_host_monitoring_blacklist_host.json', 'kv_
↪trackme_tags_policies.json', 'kv_trackme_metric_lagging_definition.json', 'kv_
↪trackme_data_sampling.json', 'kv_trackme_data_source_monitoring_blacklist_index.json
↪', 'kv_trackme_custom_lagging_definition.json', 'kv_trackme_summary_investigator_
↪volume_outliers.json', 'kv_trackme_host_monitoring.json', 'kv_trackme_data_sampling_
↪custom_models.json', 'kv_trackme_logical_group.json', 'kv_trackme_elastic_sources.
↪json', 'kv_trackme_data_source_monitoring.json', 'kv_trackme_metric_host_monitoring.
↪json', 'kv_trackme_data_source_monitoring_blacklist_host.json', 'kv_trackme_metric_
↪host_monitoring_blacklist_host.json', 'kv_trackme_metric_host_monitoring_blacklist_
↪metric_category.json', 'kv_trackme_data_host_monitoring_blacklist_sourcetype.json',
↪'kv_trackme_audit_changes.json', 'kv_trackme_metric_host_monitoring_blacklist_index.
↪json', 'kv_trackme_data_host_monitoring_blacklist_index.json', 'kv_trackme_elastic_
↪sources_dedicated.json']"}
```

JSON response in dry_run: false:

```
{ "backup_archive": "/opt/splunk/etc/apps/trackme/backup/trackme-backup-20210205-
↪142635.tgz", "status": "restore is now complete, please reload TrackMe",
↪"collections_files_restored": "['kv_trackme_data_source_monitoring_blacklist_
↪sourcetype.json', 'kv_trackme_maintenance_mode.json', 'kv_trackme_data_host_
↪monitoring_blacklist_host.json', 'kv_trackme_tags_policies.json', 'kv_trackme_
↪metric_lagging_definition.json', 'kv_trackme_data_sampling.json', 'kv_trackme_data_
↪source_monitoring_blacklist_index.json', 'kv_trackme_custom_lagging_definition.json
↪', 'kv_trackme_summary_investigator_volume_outliers.json', 'kv_trackme_host_
↪monitoring.json', 'kv_trackme_data_sampling_custom_models.json', 'kv_trackme_
↪logical_group.json', 'kv_trackme_elastic_sources.json', 'kv_trackme_data_source_
↪monitoring.json', 'kv_trackme_metric_host_monitoring.json', 'kv_trackme_data_source_
↪monitoring_blacklist_host.json', 'kv_trackme_metric_host_monitoring_blacklist_host.
↪json', 'kv_trackme_metric_host_monitoring_blacklist_metric_category.json', 'kv_
↪trackme_data_host_monitoring_blacklist_sourcetype.json', 'kv_trackme_audit_changes.
↪json', 'kv_trackme_metric_host_monitoring_blacklist_index.json', 'kv_trackme_data_
↪host_monitoring_blacklist_index.json', 'kv_trackme_elastic_sources_dedicated.json']
↪"}
```

3.5.21 Identity Cards endpoints

Resources summary:

Resource	API Path
identity_cards_collection / Get entire identity cards collection	/services/trackme/v1/identity_cards/identity_cards_collection
identity_cards_get_card / Get an identity card	/services/trackme/v1/identity_cards/identity_cards_get_card
identity_cards_get_card_by_doc_link / Get an identity card for a doc_link	/services/trackme/v1/identity_cards/identity_cards_get_card_by_doc_link
identity_cards_add_card / Add an identity card	/services/trackme/v1/identity_cards/identity_cards_add_card
identity_cards_associate_card / Associate an existing card with an object	/services/trackme/v1/identity_cards/identity_cards_associate_card
identity_cards_unassociate / Unassociate identity card from an object	/services/trackme/v1/identity_cards/identity_cards_unassociate
identity_cards_delete_card / Remove an identity card	/services/trackme/v1/identity_cards/identity_cards_delete_card

identity_cards_collection / Get entire identity cards collection

This endpoint retrieves the entire Identity Cards collection returned as a JSON array, it requires a GET call with no data required:

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/
↪identity_cards_collection/identity_cards_collection
```

SPL query:

```
| trackme url="/services/trackme/v1/identity_cards/identity_cards_collection" mode=
↪"get"
```

JSON response:

```
[
  {
    "doc_link": "https://www.acme.com/splunkadmin",
    "doc_note": "Read the docs.",
    "object": [
      "linux_amer:linux_secure",
      "linux_apac:linux_secure"
    ],
    "_user": "nobody",
    "_key": "60322369c93844004074efa1"
  }
]
```

identity_cards_get_card / Get an identity card

This endpoint retrieves the identity card linked to a specific data source, it requires a GET call with the following information:

- `object`: name of the data source

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/
↪identity_cards_collection/identity_cards_get_card -d '{"object": "linux_amer:linux_
↪secure"}'
```

(continues on next page)

(continued from previous page)

SPL query:

```
| trackme url="/services/trackme/v1/identity_cards/identity_cards_collection" mode=
↪"get" body="{\"object\": \"linux_amer:linux_secure\"}"
```

JSON response:

```
{
  "doc_link": "https://www.acme.com/splunkadmin",
  "doc_note": "Read the docs.",
  "object": [
    "linux_amer:linux_secure",
    "linux_apac:linux_secure"
  ],
  "_user": "nobody",
  "_key": "60322369c93844004074efa1"
}
```

identity_cards_get_card_by_doc_link / Get an identity card for a doc_link

This endpoint retrieves the identity card by the doc_link value, it requires a GET call with the following information:

- `doc_link`: name of the data source

External:

```
curl -k -u admin:'ch@ngeM3' -X GET https://localhost:8089/services/trackme/v1/
↪identity_cards_collection/identity_cards_get_card_by_doc_link -d '{"doc_link":
↪"https://www.acme.com/splunkadmin"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/identity_cards/identity_cards_get_card_by_doc_link
↪" mode="get" body="{\"doc_link\": \"https://www.acme.com/splunkadmin\"}"
```

JSON response:

```
{
  "doc_link": "https://www.acme.com/splunkadmin",
  "doc_note": "Read the docs.",
  "object": [
    "linux_amer:linux_secure",
    "linux_apac:linux_secure"
  ],
  "_user": "nobody",
  "_key": "60322369c93844004074efa1"
}
```

identity_cards_add_card / Add an identity card

This endpoint creates a new identity card that can later on be associated with one or more data sources (if the card based on the doc_link does not exist it is created, if the card exists already, the doc_link and doc_note are

updated and the definition of object is preserved), it requires a POST call with the following data required:

- `doc_link`: “documentation link, this will be made available in the source identity card”
- `doc_note`: “OPTIONAL: documentation note”
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪identity_cards_collection/identity_cards_add_card -d '{"doc_link": "https://www.
↪acme.com/splunkadmin", "doc_note": "Read the docs."}'
```

SPL query:

```
| trackme url="/services/trackme/v1/identity_cards/identity_cards_add_card" mode="post
↪" body="{\"doc_link\": \"https://www.acme.com/splunkadmin\", \"doc_note\": \"Read␣
↪the docs.\"}"
```

JSON response if the card is new and there are no object linked to it:

```
{
  "object": "",
  "doc_link": "https://www.acme.com/splunkadmin",
  "doc_note": "Read the docs.",
  "_user": "nobody",
  "_key": "60323c1d4f7ac770050f4a78"
}
```

JSON response if the card exists already and there are one or more objects (data sources) linked to it:

```
{
  "object": [
  "linux_amer:linux_secure",
  "linux_apac:linux_secure"
  ],
  "doc_link": "https://www.acme.com/splunkadmin",
  "doc_note": "Read the docs.",
  "_user": "nobody",
  "_key": "60323e6c4f7ac770050f4aec"
}
```

identity_cards_associate_card / Associate an existing card with an object

This endpoint associates an existing identity card with a data source (if there are data sources associated with this card already, the list of data sources is preserved and the data source to be associated is added to the list), it requires a POST call with the following data required:

- `object`: the data source name to be associated with this card
- `key`: the KVstore unique key for this identity card
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪identity_cards/identity_cards_associate_card -d '{"key": "60327fd8af39041f28403191",
↪ "object": "linux_apac:linux_secure"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/identity_cards/identity_cards_associate_card"␣
↪mode="post" body="{\"key\": \"60327fd8af39041f28403191\", \"object\": \"linux_
↪apac:linux_secure\"}"
```

JSON response :

```
{
  "object": [
    "linux_amer:linux_secure",
    "linux_apac:linux_secure"
    ],
  "doc_link": "https://www.acme.com/splunkadmin",
  "doc_note": "Read the docs.",
  "_user": "nobody",
  "_key": "60327fd8af39041f28403191"
}
```

Wildcard matching:

Wildcard matching can be performed via the REST API endpoint (but not when managed via the UI), the following example will associate any entities starting by `linux_*`:

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪identity_cards/identity_cards_associate_card -d '{"key": "60327fd8af39041f28403191",
↪ "object": "linux_*"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/identity_cards/identity_cards_associate_card"␣
↪mode="post" body="{\"key\": \"60327fd8af39041f28403191\", \"object\": \"linux_*\"}"
```

JSON response :

identity_cards_unassociate / Unassociate identity card from an object

This endpoint unassociates the identity card for an object (data source), it requires a POST call with the following data required:

- `object`: the object name (data source) to remove association for
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪identity_cards/identity_cards_unassociate -d '{"object": "linux_apac:linux_secure"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/identity_cards/identity_cards_unassociate" mode=
↪"post" body="{\"object\": \"linux_apac:linux_secure\"}"
```

JSON response if association removal is performed:

```
{
    "response": "object linux_apac:linux_secure has been unassociated from identity␣
↪card record key: 60327fd8af39041f28403191"
}
```

JSON response if there is no association to be removed:

```
{
    "response": "object linux_apac:linux_secure already has no identity card␣
↪association."

}
```

identity_cards_delete_card / Remove an identity card

This endpoint deletes an idenfity card by the Kvstore key, it requires a DELETE call with the following informatio:

- `key`: KVstore unique identifier for this record
- `"update_comment": OPTIONAL: a comment for the update, comments are added to the audit record, if unset will be defined to: API update`

External:

```
curl -k -u admin:'ch@ngeM3' -X POST https://localhost:8089/services/trackme/v1/
↪identity_cards/identity_cards_delete_card -d '{"key": "60327fd8af39041f28403191"}'
```

SPL query:

```
| trackme url="/services/trackme/v1/identity_cards/identity_cards_delete_card" mode=
↪"delete" body="{\"key\": \"60327fd8af39041f28403191\"}"
```

response:

```
Record with _key 6032a59c7e8f2844dd3b553e was deleted from the collection.
```

CHAPTER 4

Troubleshoot:

4.1 FAQ

You will find in this page different smart and understandable questions, which are made available for everyone's value, thank you so much for asking!

4.1.1 What is the "data name" useful for?

See *Data Sources tracking and features*

In the context of data source, the field **"data_name"** represents the unique identifier of the data source.

- for regular data sources created by TrackMe, this is equal to combination of <index>:<sourcetype>.
- for Elastic Sources, the definition defined when the entity is created

The data_name unique identifier is used in different parts of the application, such as the search trackers which rely on it to identify the data source.

What are the numbers in the "lag summary" column?

See *Data Sources tracking and features*

The field **"lag summary (lag event / lag ingestion)"** is exposed within the UI to summarise the two key metrics handled by TrackMe to monitor the Splunk data.

The field is composed by:

- lag_ingestion_sec: delta between index time and latest event timestamp, which can be represented as: `(_indextime - _time)`
- lag_event_sec: delta between now when the measure is taken (when the tracker was run), and the latest event timestamp, which can be represented as `(now() - _time)`

In this context:

- event time stamp is the _time field, which is the time stamp that Splunk affected for a given event when it was indexed on disk
- index time is the _indextime field, which is the epoch time corresponding to the moment when Splunk indexed the event on disk

Depending on the use cases, both key performance metrics might be very important, or potentially one will matter more than the other.

For continuous data flow, you need to know that the data is being indexed with the expected performance, and that you are not running late due to some underneath performance issues, this is where the lag_ingestion_sec matters.

On the other side, you need as well to be able to detect if the data flow is somehow broken, and the ingestion stopped unexpectly, this is where the lag_event_sec matters.

4.1.2 Is the "priority" a configurable value? If yes how do you configure it, if not how does the app derives it?

See *Priority management*

The **"priority"** field is configurable within the UI, when you click on any entity and enter the modification space (`Modify` bytton).

Its value is defined automatically when the entity is discovered, stored in the KVstores, its default value is defined by the following macro:

`[trackme_default_priority]`

Each iteration of the trackers preserve the priority value.

The priority is used in the main charts and single forms in the UI to expose the current situation. As well, the OOTB alerts are filtering by default on a given list of priorities, driven by the following macro:

`[trackme_alerts_priority]`

By default, all entities are added as "medium", one option can be to update the macro to be looking at high only, such that you qualify for each entity is you want to be alerted on it and define it to high priority. Another option would be to define everything to low besides what you qualify and want to be monitoring and get alerted.

The purpose of the priority field is to provide a granularity in which entities should be generating alerts, while all information remains easily visible and summarised in the UI.

4.1.3 Can the priority be externally managed?

See *Priority management*

In some use cases you may want to retrieve and/or define the value of the priority field from an external source such as a CMDB lookup stored in Splunk. (specially for data hosts)

As the value of a priority is preserved over iterations, a simple search that does a mapping and finally a KVstore update could be scheduled and run on a regular basis.

Some logic similar to:

```
| inputlookup trackme_host_monitoring | eval key=_key
| lookup <my CMDB lookup> data_host as host OUTPUT priority as cmdb_priority
| eval priority=if(isnotnull(cmdb_priority), cmdb_priority, priority)
| outputlookup trackme_host_monitoring append=t key_field=key
```

This search would take input the content of the lookup, perform a mapping to retrieve the priority value from the CMDB, run a simplistic evaluation and finally updates the KVstore entries.

4.1.4 Why do we need both short and long term trackers?

This is required to cover most of the use cases, in the most performing manner, at the lowest cost for the environment.

For reference:

- Short term trackers run every 5 minutes, earliest=-4h latest=+4h
- Long term trackers run once per hour, earliest=-7d latest=+4h

There are different scenarios where the short term tracker would not be able to catch information about a data flow, for example if you are recovering from an outage and the data is still running late (you are catching up), or if you are indexing data in the past which would be out of the time frame scope of the short term trackers.

For these reasons and for performance considerations, the search workload is split into two main trackers which each cover a specific time frame.

4.1.5 How the app determines what's a good status and what's a bad status?

This depends on different factors, and depends on the configuration of the entity too, but in short:

- Up to the version 1.2.18, if either the lag ingestion or the lag event exceeds the max lag allowed value, the entity status will be `red`
- Starting version 1.2.19, it is possible to define if the status should be defined depending of both KPIs, the lag ingestion only or the lag event only, depending on the configuration the status will `red` if the monitoring conditions are not met
- If Outliers detection is enabled, and if the Outliers status does not meet the policy, the status will be `red`
- If TrackMe detects data ingested in the future, that exceeds the tolerance defined in the macro "trackme_future_indexing_tolerance", the status will be `orange`
- If the status is red, and if the week days monitoring policy implies not triggering, the status will be `orange`
- In addition for data and metric hosts, if the entity is red and is part of a logical group which status complies with its policy (example 50% available and only of the 2 members is red), the status will be `blue`

The OOTB Alerts by default alert on the `red` status only.

For each status condition, a clear description is provided as part of a message which is visible in the UI, visible as focus over the icon, and as part of the alert output.

Example:

```
Alert: data source status is red, monitoring conditions are not met due to lagging or␣
↪interruption in the data flow, latest data available is 24/07/2020 19:30 (7149␣
↪seconds from now) and ingestion latency is approximately 30 seconds, max lag␣
↪configured is 125 seconds.
```

4.1.6 How can you see a list of deleted entries? Can you undelete an entry?

A user can delete an entity stored in the KVstore, assuming the user has write permissions over the KVstores and other objects. (admin, part of trackme_admin role or custom allowed)

The deletion feature is provided natively via the UI, when an entity is deleted the following workflow happens:

- The UI retrieves the key id of the record in the KVstore and performs a DELETE rest call over the KVstore endpoint
- In addition, the full entity record is logged to the audit KVstore, and exposed via the UI within the audit changes tab
- When the user deletes an entity, it can be delete temporary or permanently
- If the deletion is temporary, the entity will be recreated automatically if it is still actively sending data to Splunk, and the conditions (allow lists, block lists. . .) permit it
- If the deletion is permanent, an additional flag is added to the record in the audit, this flag allow the trackers to exclude creating an entitythat was permanently deleted

While it is not supported at the moment to undo the deletion, the audit record contains all the information related to the entitypreviously deleted.

Finally, the audit changes tab provides the relevant filters to allow accessing to all deletion events, including answers to when / who / how and why if an update note was added filled during the operation.

4.1.7 What are Elastic Sources and what are they useful for?

The Elastic source concept is covered in deep in the *Elastic sources* documentation, wich includes comprehensive examples.

4.1.8 How to deal with sourcetypes that are emitting data occasionally or sporadically? Does TrackMe automatically detects this?

There are no easy answers to this question, however:

- The default concept of data sources tracking relies on entities broken per index and sourcetype, this can be extended easily using the Elastic sources feature to fullfil any kind of requirements and make sure that a data source represents the data pipeline
- The data hosts tracking feature provides the vision broken on a per host basis (using the Splunk host Metadata)
- TrackMe does not replace the knowledge you have regarding the way you are ingesting data into Splunk, instead it provides various features and options you can use to configure what should raise an alert or not, and how
- The basic configuration for data tracking are related to the latency and the delta in seconds between the latest time data was indexed in Splunk and now
- In addition, the volume Outliers feature allows detecting automatically behaviour changes in the volume of data indexed in Splunk for a given sourcetype
- In most cases, you should focus on the most valuable and important sourcetypes, TrackMe provides different levels of features (allowlists / blocklists) to exclude automatically data of low interest, and the priority feature allows granular definition of the importance of an entity
- A sourcetype that comes very occasionally in Splunk might be something that you need to track carefully, however if it does you need to define the tresholds accordlingy and TrackMe provides different options to do so on a per data source basis for instance

4.1.9 What is the purpose of the enable / disable button?

The purpose of the enable / disable button is to provide a way to disable the monitoring of an entity, without removing it from the collections entirely.

There are different aspects to consider:

- Sometimes you have some sourcetypes you do not care about really, you can use allowlisting / blocklisting, or disable it
- When an entity is disabled, the value of the field "data_monitored_state" is set to false (default is true when it is discovered initially)
- The UI by default filters on entities which are being monitored effectively, you can show disabled entities by using the "Filter monitored_state:" filter form, or looking at the lookup content manually
- Out of the box alerts do not take in consideration disabled entities
- Various other parts of the application will as well stop considering these disabled entities, for instance there will not be metrics generated anymore, etc.
- When an entity is disabled, all information are preserved, if you re-enable a disabled entity, TrackMe will simply start to consider it again and refresh its state and other actions automatically
- You should consider disabling entities rather than deleting entities if these are actively generating data to Splunk and cannot be excluded easily by allow listing / block listing
- The reason is that if you delete an active entity, in temporary deletion mode it will be re-added very quickly (when the trackers will capture activity for it), and permanent mode it would re-added after a certain period of time

4.1.10 What's the difference between disabled and (permanently) deleted?

The deletion of entities is explained in details in *Deletion of entities*.

In short, the purpose of the permanent deletion is to prevent an entity from being disovered again after it is deleted.

To achieve this, when an entity is permanently deleted the value of the field "change_type" is defined to "delete permanent", when the entity is temporarily deleted, the value is set to "delete tempoary".

Then, Trackers reports wich perform discovery of the data use a filter to exclude entities that have been permanently deleted, such that even if the entity is still actively sending data to Splunk, TrackMe will ignore it automatically as long as the audit record is available. (by default audit records are purged after 90 days)

The UI does not provide a function to undo a permanent deletion, however updating or purging the audit record manually would allow to re-create an entity after it was permanently deleted.

CHAPTER 5

Versioning and build history:

5.1 Release notes

5.1.1 Version 1.2.59

> **Warning: Splunk 8.x and Python3 support only**
>
> - Starting from this release, only Splunk 8.x and Python3 are supported
> - Some functions such as builtin alert actions are not compatible any longer with Python2 and Splunk 7.x
> - For the latest version available for Splunk 7.x, see the release 1.2.51

- Fix Issue #391 - HA group alert mapping is not working as expected #391
- Fix Issue #392 - Data hosts long term tracker should collect after filtering on hosts out of the scope of the short term, and this scope should include data_last_ingest
- Fix Issue #389 - Using double-backslash for regular expression in blocklist fails trackme

5.1.2 Version 1.2.58

> **Warning: Splunk 8.x and Python3 support only**
>
> - Starting from this release, only Splunk 8.x and Python3 are supported
> - Some functions such as builtin alert actions are not compatible any longer with Python2 and Splunk 7.x
> - For the latest version available for Splunk 7.x, see the release 1.2.51

- Fix Issue #379 - urllib3 insecure error messages from custom endpoint when interracting with splunkd
- Fix Issue #378 - backup and restore generate POST related warning messages

- Fix Issue #377 - Typos in sample instructions in backup and restore user interface
- Fix Issue #364 - The long term tracker can impact some data sources unexpectly in some specific conditions

5.1.3 Version 1.2.57

Warning: Splunk 8.x and Python3 support only

- Starting from this release, only Splunk 8.x and Python3 are supported
- Some functions such as builtin alert actions are not compatible any longer with Python2 and Splunk 7.x
- For the latest version available for Splunk 7.x, see the release 1.2.51

- Fix Issue #375 - restore fails due to max document per batch API limit reached
- Fix Issue #371 - typo in Data sampling error messages
- Fix Issue #374 - missing shortcut to trackme_data_source_monitoring_blacklist_data_name in the nav menu

5.1.4 Version 1.2.56

Warning: Splunk 8.x and Python3 support only

- Starting from this release, only Splunk 8.x and Python3 are supported
- Some functions such as builtin alert actions are not compatible any longer with Python2 and Splunk 7.x
- For the latest version available for Splunk 7.x, see the release 1.2.51

- Fix Issue #372 - Non existing id section in app.conf was reported to cause trouble to Splunk Cloud internal automation

5.1.5 Version 1.2.55

Warning: Splunk 8.x and Python3 support only

- Starting from this release, only Splunk 8.x and Python3 are supported
- Some functions such as builtin alert actions are not compatible any longer with Python2 and Splunk 7.x
- For the latest version available for Splunk 7.x, see the release 1.2.51

- Feature: Provides a new split by custom mode to allow defining a custom indexed field in the data source discovery and maintenance workflow
- Feature: Notification bar and various UI improvements in the configuration UI

5.1.6 Version 1.2.54

> **Warning: Splunk 8.x and Python3 support only**
>
> - Starting from this release, only Splunk 8.x and Python3 are supported
> - Some functions such as builtin alert actions are not compatible any longer with Python2 and Splunk 7.x
> - For the latest version available for Splunk 7.x, see the release 1.2.51

- Fix - Issue #368 - Disable the KVstore to indexers replication for the kv_trackme_objects_summary collection

5.1.7 Version 1.2.53

> **Warning: Splunk 8.x and Python3 support only**
>
> - Starting from this release, only Splunk 8.x and Python3 are supported
> - Some functions such as builtin alert actions are not compatible any longer with Python2 and Splunk 7.x
> - For the latest version available for Splunk 7.x, see the release 1.2.51

- Fix - Issue #362 - Windows based deployment reports ERROR JSON reply had no "payload" value in rest calls

5.1.8 Version 1.2.52

> **Warning: Splunk 8.x and Python3 support only**
>
> - Starting from this release, only Splunk 8.x and Python3 are supported
> - Some functions such as builtin alert actions are not compatible any longer with Python2 and Splunk 7.x
> - For the latest version available for Splunk 7.x, see the release 1.2.51

- Enhancement - Issue #360 - JQuery upgrade for Simple XML dashboards
- Enhancement - migration to ucc-gen for the librairies management and the app generation
- Change - Python2 and Splunk 7.x support is dropped starting from this release, TrackMe now only supports Splunk 8.x and Python3

5.1.9 Version 1.2.51

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120
- Splunk SA CIM - Splunk Common Information Model, Splunk Base: https://splunkbase.splunk.com/app/1621

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Fix - Issue #356 - trackme.py endpoint check can be circumvented to perform REST calls to endpoints external to TrackMe
- Fix - Issue #357 - In Splunk Cloud the UI manage and configure will not provide the right URL for quick access to the macro definition

5.1.10 Version 1.2.50

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120
- Splunk SA CIM - Splunk Common Information Model, Splunk Base: https://splunkbase.splunk.com/app/1621

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Fix - Issue #352 - Splunk 8.2 regression in data sampling engine causes only 1 event to be stored in the sampling KVstore post execution due stats first(*) change in behaviour

5.1.11 Version 1.2.49

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120
- Splunk SA CIM - Splunk Common Information Model, Splunk Base: https://splunkbase.splunk.com/app/1621

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Enhancement - Fix Issue #343 - REST CALL - use nobody context to optimize rest calls performance in large scale environments

5.1.12 Version 1.2.48

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120
- Splunk SA CIM - Splunk Common Information Model, Splunk Base: https://splunkbase.splunk.com/app/1621

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Enhancement - Issue #335 - addresses memory overhead of the metric trackers using span=1s by default
- Fix - Issue #336 - Fix - SmartStatus - future tolerance macro is not taken into account by the endpoint
- Fix - Issue #333 - Nav - Wrong search for metric hosts allow list collection
- Fix - Issue #337 - Data sources - Short term tracker run via the UI should use latest=+4h, long term tracker should match savedsearch earliest=-24h latest=-4h
- Fix - Issue #338 - Splunk 8.2 regression in rootUri for UI TrackMe manage drilldowns to macro due to a root URL change in manager
- Fix - Issue #339 - Data sources - Data source overview chart tab should honor the trackme_tstats_main_filter macro
- Change - Nav - remaining whitelist and blocklists terms

5.1.13 Version 1.2.47

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120
- Splunk SA CIM - Splunk Common Information Model, Splunk Base: https://splunkbase.splunk.com/app/1621

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Fix - Issue #328 - Data host - Regex based block lists are not honored as documented
- Fix - Issue #329 - Data host - Splunk 8.2 regression with multivalue aggregation caused by a change in behaviour
- Change: Update splunktaucclib to 4.2.0
- Change: Update splunktalib to 1.2.1

5.1.14 Version 1.2.46

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120
- Splunk SA CIM - Splunk Common Information Model, Splunk Base: https://splunkbase.splunk.com/app/1621

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Enhancement - Issue #327 - Smart Status - Add search history quick access button in Smart Status screens

- Fix - Issue #324 - Lagging classes - lagging classes applying at the same level (all/data_source/data_host) for different types of objects and the same name are not honoured properly due to a logic default in the lookup mapping

5.1.15 Version 1.2.45

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120
- Splunk SA CIM - Splunk Common Information Model, Splunk Base: https://splunkbase.splunk.com/app/1621

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Feature - Issue #312 - Migration from Addon Builder based libs to Splunk Addon factory UCC based libs
- Feature - Issue #316 - Provides day time filtering options when creating custom alerts

5.1.16 Version 1.2.44

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120
- Splunk SA CIM - Splunk Common Information Model, Splunk Base: https://splunkbase.splunk.com/app/1621

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Fix Issue #310 - Alert actions - Dropdown object in Smart Status tab rendering errors

5.1.17 Version 1.2.43

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120
- Splunk SA CIM - Splunk Common Information Model, Splunk Base: https://splunkbase.splunk.com/app/1621

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Fix Issue #308 - Alert actions - extraction failure for Smart Status in the UI for rendering purposes

5.1.18 Version 1.2.42

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120
- Splunk SA CIM - Splunk Common Information Model, Splunk Base: https://splunkbase.splunk.com/app/1621

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Feature - Issue #306 - Alert actions - UI enhancements
- Fix - Issue #305 - Custom alerts - created alerts should set alert.digest_mode

5.1.19 Version 1.2.41

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120
- Splunk SA CIM - Splunk Common Information Model, Splunk Base: https://splunkbase.splunk.com/app/1621

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Feature - Issue #300 - TrackMe now comes builtin with alert actions enabled by default on out of the box alerts, these actions perform auto acknowledgement, call and index the Smart Status result, the third action is a free style action that call any of the TrackMe REST API endpoints
- Change: Normalize the suppress fields for all alerts to use the object/object_category TrackMe naming convention
- Fix - Issue #293 - Splunk telemetry causes DateParserVerbose Warnings logged
- Fix - Issue #299 - Data Sampling - In some circumstances, the custom rule editor might fail to render events
- Fix - Issue #301 - Smart Status - the REST handler should call the eval state status macro in case it is called before the KVstore is updated
- Fix - Issue #302 - REST endpoints - Ack - wrong audit event logged
- Fix - Issue #303 - REST endpoints - Backup and Restore - the purge operation purges the archive but not the KVstore record

5.1.20 Version 1.2.40

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378

- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Enhancement - Issue #297 - Performances - Long term tracker improvements

5.1.21 Version 1.2.39

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Feature - Issue #292 - Alerts - New screen for alerts management in TrackMe, review, edit and add alerts within the UI
- Enhancement - Issue #295 - Long term trackers performance - Major reduction of the long term trackers runtime by better taking into account the existing short term entities knowledge
- Enhancement - Issue #290 - Alerts - OOTB alert TrackMe - Alert on data source availability should suppress on data_name only
- Fix - Issue #291 - REST endpoint - the endpoint mh_update_priority does not preserve the monitored_state
- Fix - Issue #294 - Data hosts - Long term tracker filter error causes the long term to miss entities indexing lately

5.1.22 Version 1.2.38

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Fix - Issue #287 - Since version 1.2.37 most of interractions in the UI are made via TrackMe rest endpoints, however the capability list_settings is required for non privileged users and should be added to the trackme_admin role

5.1.23 Version 1.2.37

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Enhancement - Issue #279 - Decomission of the getlistdef custom command in favor of a simpler and cleaner pure SPL approach
- Enhancement - Issue #280 - Add new REST endpoint to manage logical group associations
- Enhancement - Issue #285 - Flipping statuses workflow improvements
- Change - Issue #275 - permissions - provides a builtin trackme_user role to handle the minimal non admin access for TrackMe
- Change - Issue #276 - User Interface - Migration of Ajax javascript REST calls made within the UI from splunkd to TrackMe based API endpoints
- Change - Issue #278 - Upgrade of splunklib Python SDK to latest release 1.6.15
- Fix - Issue #273 - User Interfaces - Several searches should not kick off start at TrackMe main UI loading time
- Fix - Issue #274 - Data Sources - tags dropdown can render unwanted results when no tags are defined
- Fix - Issue #277 - REST endpoint - the endpoint ds_update_min_dcount_host should allow any as the input

5.1.24 Version 1.2.36

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Feature - Issue #266 - ID cards - Wildcard matching for ID cards allowing matching any number of entities for the same card using wildcards and your naming conventions
- Enhancement - Issue #268 - Backup and Restore - Perform an additional get call in the Backup operation to automically discover any missing backup files
- Fix - Issue #267 - Backup and Restore - Python2 compatibility issues with Splunk 7.x
- Fix - Issue #261 - SLA - SLA reporting should honour allow/block list and not monitored entities #261
- Fix - Issue #266 - ID cards - Updating an existing card within the UI removes other associations with the card that is updated
- Fix - Issue #270 - REST endpoint resources groups wrong exposure for Splunk Web proxied behaviors

5.1.25 Version 1.2.35

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Feature - Issue #249 - CRIBL native integration - TrackMe can now be configured to be transparently reliying on the Cribl pipeline concept to discover and track data sources based on the cirbl_pipe to provide an easy and performing integration
- Feature - Issue #250 - new blocklisting capabilities based on the data_name for data sources
- Feature - Issue #254 - Data Sampling - The new Data Sampling obfuscation mode allows preventing unwanted data accesses to the collection by obfuscating samples at the processing step instead of storing samples within the KVstore collection
- Feature - Issue #253 - Splunk Infrastructure Monitoring, Splunk instances are now monitored automatically in the data hosts tracking via the splunkd sourcetype, this behaviour can be enabled/disabled on demand via the configuration UI
- Feature - Issue #260 - REST API endpoints - new endpoints for identity cards management
- Enhancement - Issue #251 - Reset collections should better run short term trackers rather than long term trackers for data sources and hosts when resetting
- Enhancement - Issue #257 - Allow listing - provides explicit expression addition capabilities with wildcard support
- Enhancement - Issue #258 - Metric hosts - adds the _metrics in hosts and metrics discovery
- Enhancement - Issue #259 - Lagging performances tab - Improve rendering and analytics
- Enhancement - Issue #263 - Default priority taken into account by OOTB alerts should rather filter for high priority by default (macro: trackme_alerts_priority)
- Fix - Issue #245 - SLA & QOS - Inconsistency in the calculations using stats range function, replaced with a streamstats based approach for accurate calculations
- Fix - Issue #246 - Data sources - misleading status message for data sources ingesting data in the future due to bad TZ
- Fix - Issue #256 - host blocking list based on regex does not work properly
- Fix - Issue #261 - SLA reporting should honour allow and block lists for each category

5.1.26 Version 1.2.34

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Enhancement - Issue #241 - KVstore backup and restore - Improved workflow with Metadata recording of backup archives, new dashboard providing insights on the workflow and its features
- Fix - Issues #242 - UI - interfaces like lagging classes, allow and block listing should not remove the search input form if there are no results found

5.1.27 Version 1.2.33

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Fix - Appinspect failures due to CSV lookup files not referenced as lookups (non Cloud failure)

5.1.28 Version 1.2.32

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Enhancement - Issue #230 - data host over time and single search performance improvements
- Enhancement - Issue #222 - Automatically Backup Main KV Store collections, provide endpoints for backup and restore operations
- Enhancement - Issue #232 - REST API and tooling - Provide a new app nav menu and a new dashboard to demonstrate the REST API endpoints and the usage of the trackme API in SPL commands
- Fix - Issue #231 - UI - reduce the max number of entries in the tag policies screen (goes beyond the modal limitation)
- Fix - Issue #233 - Smart Status - orange state due to week days monitoring is not properly handled
- Fix - Issue #235 - Data sources - Week days monitoring rules are not honoured if triggering due to dcount host
- Fix - Issue #236 - Data sources - status message is inaccurate if data source is in data sampling alert but week days monitoring rules are not met

5.1.29 Version 1.2.31

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Feature: Introducing the trackme REST API wrapper SPL command, allows interracting with the TrackMe REST API endpoints within SPL queries!
- Feature: Introducing the smart status REST API endpoints, performs advanced status correlations and investigations easily and automatically, within the UI, as part of an alert action or within your third party automation!
- Feature: REST API endpoint for Data Sampling - allow reset and run sampling
- Feature: UI - Issue #223 - multiselect form enhancement with auto disablement of the ALL choice when selecting at least one entry in the multiselect
- Feature: Identity cards - Issue #226 - allow defining a global default identity card associated with all data sources (per data source identity cards can still be created and take precedence over the global card)
- Feature: Elastic Sources - Issue #227 - allow deletion of both dedicated and shared sources in the UI via the new REST framework, deletion automatically performs the deletion of related objects (KVstore contents, report, etc)
- Fix - Issue #217 - Activity alerts view results link would result to 404 page not found for out of the box alerts
- Fix - Issue #218 - Data sampling - creating custom rule from the main screen, then clicking on back button leads to wrong window
- Fix - Issue #219 - Outliers detection - dropdown for alert on upper is not pre-filled with the actual setting of the entity
- Fix - Issue #220 - Audit scheduling - in some environments, status=”success” is replaced at search time by status=”completed” (internal scheduler) which is not expected by the searches
- Fix - Issue #221 - Data sources - Tags are not preserved following actions in the UI
- Fix - Issue #224 - Outliers - Switching an entity between different periods may lead the outliers generation to be failing
- Fix - Issue #225 - Outliers - Data hosts outliers configuration update within the UI causes an entity refresh which does not generate flipping statuses events as expected
- Fix - Issue #228 - REST API - Endpoints should honour the user context while logging the action in the audit log collection
- Change: Icons change

5.1.30 Version 1.2.30

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378

- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Feature - Issue #210 - new REST API endpoints for Elastic Sources / Logical Groups / Data Sampling / Tags Policies / Lagging Classes / Lagging Classes Metrics
- Feature - Issue #212 - Data sampling - Allows defining exclusive rules for data sampling custom models, this can be used when a regex must not be matched, such as detecting PII data automatically
- Feature - Issue #214 - Data sampling - Allows defining a custom number of records to be sampled on a per data source basis
- Feature - Issue #215 - Data Hosts - Support for priority based lagging classes
- Fix - Data sampling - Clear state and run sampling action would fail if actioned on a data source which data sampling has not run yet at least once, fixes and UI improvements for Data sampling
- Change - Issue #213 - knowledge objects default permissions - Review of the app related KVstores default permissions, fixing missing collections and transforms

5.1.31 Version 1.2.29

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Feature Issue #205 - Introducing TrackMe REST API endpoints for automation integration and future UI evolutions (https://trackme.readthedocs.io/en/latest/rest_api_reference.html)
- Feature Issue #209 - Feature - Provides a new mode for data sources to allow by index level analysis
- Fix Issue #208 - Fix - creating a rest based search causes regression in the data sampling and event recognition engine

5.1.32 Version 1.2.28

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Feature Issue #201 - Elastic Sources - Support for lookup tracking with from commands

- Feature Issue #202 - Elastic Sources - Support for remote searches using rest
- Fix Issue #203 - Provides a macro based definition for first level span of Metrics trackers
- Change: Upgrade of splunklib Python SDK to latest release 1.6.14

5.1.33 Version 1.2.27

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

Major improvements in data host monitoring capabilities:

- Feature: Data hosts - monitoring workflow improvement with alerting policy, monitor hosts with sourcetype level granularity at scale
- Feature: Lagging classes - policies can now be defined against the priority (data sources only), in addition policies can be set for all objects, data sources or hosts only
- Feature: Better management of allow lists / blocking lists for data hosts monitoring
- Feature: Data hosts and metric hosts rendering improvements in multi-value field structure with state rendered as emoji icons for better readability
- Change: Data hosts monitoring uses same default lagging than data sources (3600 sec)

Data sources changes:

- Feature: Issue #196 Data sources - Provides distinct count threshold capabilities to turn a data source red if the number of hosts goes below a static threshold, provides chart visibility in Overview screen of the data source

Others:

- Fix: Issue #193 - data hosts - the refresh button does not refresh the host screen header (priority, etc) #193
- Fix: Issue #198 - Elastic Sources - When creating a from based source, if there are no additional search constraints after the data model name, no results will be returned
- Fix: Issue #199 - Data sampling - some builtin rules are too restrictive regarding multiple spaces
- Change: Increase max height for timeline chart in Status message tab (current max height might be too low when multiple statuses)

5.1.34 Version 1.2.26

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Feature: Issue #186 - Data sampling - during the creation of a custom rule, its scope can now be restricted to a list of specific sourcetypes to dedicate custom rules and avoid rules overlapping issues
- Feature: Issue #188 - SLA calculation migration from flipping statuses events to current statuses events for reliable results / SLA dashboard improvements / Drilldown from SLA single percentage in TrackMe main UI to SLA dashboard
- Feature: Issue #190 - UI improvements - provide quick access to data sampling custom rules in the main data sources tab, unify trackers manual run for data sources and hosts in a single button and window
- Feature: Issue #191 - UI improvements - Load spinner at TrackMe loading stage, Spinner design refresh globally in TrackMe

5.1.35 Version 1.2.25

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Feature: Issue #181 - Disable data sampling on demande via the UI #181
- Fix: Issue #180 - Outliers detection impacts offline data such as low frequency batched data sources #180
- Fix: Issue #182 - Data sampling - Manual run, Clear state and run sampling UI period constraint is too short for cold data sources #182
- FIx: Issue #183 - Data Sampling - number of entities to process calculation can lead to no entities being processes #183

5.1.36 Version 1.2.24

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890
- Splunk Timeline - Custom Visualization, Splunk Base: https://splunkbase.splunk.com/app/3120

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Feature: Issue #153 - For ITSI and timeline integration purposes, generate and store last states information as summary events #153
- Feature: Issue #141 - Enhancement - ability to search for hosts in Data Hosts Tracking by Logical Group Name #141
- Feature: Issue #148 - Enhancement: Allow ‘NOT’ filter for Keyword filter name: #148

- Feature: Issue #166 - Enhancement - Provides a UI feature to allow reseting the list of metrics known for a given metric host
- Feature: Issue #174 - Enhancement - Adding the timeline viz view in the status tabs #174
- Fix: Issue #147 / Issue #161 Outliers management and configuration - fixes and improvements
- Fix: Issue #167 - Issue - Pressing “Manage: manual tags” displays dialog with ALL tags in “List of current tags for this data source” field #167
- Fix: Issue #170 - install_source_checksum should not be in app.conf (appinspect warning) #170

5.1.37 Version 1.2.23

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Fix: Exclusion of metrics generated by TrackMe itself would exclude other metrics generated on the same search head
- Fix: Issue #151 - error handling does not catch a failure during the creation of a new elastic source #151
- Fix: Issue #154 - Splunk Cloud vetting - capability in role will not be be granted #154
- Fix: Issue #155 - Splunk Cloud - In some specific contexts, Elastic source dedicated tracker creation fails #155

5.1.38 Version 1.2.22

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Feature: Extending the Tags features with tags policies, this feature provides a workflow to automatically define tags using regular expressions rules matching the data_name value and its naming convention
- Feature: Improved views for Ops queues (renamed to Ops: Queues center) and Ops parsing, multi hosts selector, improved analytics
- Fix: Issue #131 - The enable data source action does not preserve the current value of data_lag_alert_kpis in the collection, which ends as a null value
- Fix: Issue #138 - Typo in the metrics screen, Metrics categories was mispelled
- Fix: Issue #139 - TrackMe metrics should be excluded out of the box from the metrics tracking
- Fix: Issue #142 - Disabing Acknowledment is broken due to the add comment feature introduction
- Fix: Issue #144 - Ack disable should use the comment for update if any #144

- Change: Include the priority value when generating the flipping status summary events
- Change: Do not load the raw_sample field when during the execution of data sources tracker execution for optimization purposes

5.1.39 Version 1.2.21

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Feature: Introducing a new very hot feature! Data sampling and event format recognition is a new workflow that allows monitoring the event formats behaviour by processing automated sampling of the data sources and monitoring their behaviour over time, builtin rules are provided and can be extended with custom rules to handle any custom data format
- Feature: Introducing the new tags capability, you can now add tags to data sources, tags are keywords which can be set per data source to provide new filtering capabilities
- Fix: When using a custom Splunk URI path (root_endpoint in web.conf), internal calls to splunkd made the UI can fail if splunkd does not accept the root context and only accepts the custom root context
- Fix: When creating new dedicated elastic sources, if the search result name exceeds 100 characters, this results in a silent failure to create the new source
- Fix: Shorten default naming convention used for new Elastic Sources tracker names
- Fix: Limitation of the list function used in stats limits the number for Elastic shared data sources to 99 sources maximum, fixed by alternative improved syntax
- Fix: For Elastic shared sources, if the first source is a raw search, the addition of the “search” key word in the first pipeline fails under some conditions
- Change: Automatically join the acknowledgement comment in the acknowledgement screen
- Change: Time to live definition for scheduled reports (dispatch.ttl) to reduce overhead in the dispatch directory
- Change: Automatically affect a 1 minute time window when creating Elastic dedicated trackers

5.1.40 Version 1.2.20

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Fix: getlistdef.py custom command fails with a Python decode error if running in a Python3 only instance
- Fix: Allowlist / Blacklist and similar deletion checkbox may fail to refresh the window content properly upon record(s) deletion

- Change: UI rendering improvements
- Fix: For metric hosts, logical group mapping generates false positive status flipping events, blue hosts should not appear in single count of hosts in alert, refresh button should respect the current blue status
- Fix: For data hosts, logical group mapping (blue hosts) should not appear in single count of hosts in alert, refresh button should respect the current blue status

5.1.41 Version 1.2.19

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Feature: Improved rendering of the ingestion lag and event lag metrics for data sources and hosts modal windows (new single for event lag, automatically converted to a duration format)
- Feature: over KPI alerting option, this new feature allows for data sources and data hosts entities to choose which KPI to be alerting against, between all KPIS, lag ingestion KPI only or lag event KPI only.
- Feature: Improved look and feel of modal windows with a header color scheme based on the action performed
- Fix: In table checkbox CSS code fix to get square shape instead of a rectangle
- Fix: In auto lagging definition modal windows, the current modal window should be hidden automatically when the action is executed
- Fix: Minor fix of input forms spacing in the main UI related to the keyword search text input box
- Fix: Alignment of header separator issues with Firefox for the main modal Windows
- Change: Remove data_index and data_sourcetype in the table output for data sources as the data_name field itself summarises these information

5.1.42 Version 1.2.18

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

- Fix: Builtin alerts should do not honour index allowlisting (for entities created before allowlists were configured)
- Change: In support with the elimination of long term used negative words in IT, whitelist and blacklist concepts are replaced with allowlist and blocklist concepts
- Fix/Feature: UI improvement with a checkbox in table approach to provide deletion capabilities on the different screens such as allowlist and blocklist, in some circumstances the drilldown approach was causing trouble with unexpected closure of the modal window

- Fix: Outliers generation with mstats and the append=true mode fails in some distributed architecture, the single schedule report is replaced with a scheduled per potential period configured for entities, in a high performing fashion and capable of dealing with any number of entities
- Fix: Active links such as opening in a search a data source might get broken in some environments when using a custom Splunk URI path (root_endpoint in web.conf)

5.1.43 Version 1.2.17

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

release notes:

- Fix: Outliers detection framework issues (some parameters are not properly honoured due to regressions in prior versions)
- Fix: When modifying outliers configuration on Elastic sources, entities could be temporary stated in red state due to entity refresh started as a background action, while for Elastic searches the combo index/sourcetype might not refer to real values depending on their configuration
- Fix: Outliers simulation under some circumstances can show discrepancy in results regarding results which would be calculated once applied
- Feature: Improved refresh capabilities for data sources and automatically define the best suitable search depending on the type of the data source (standard, shared or dedicated Elastic source)

5.1.44 Version 1.2.16

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

release notes:

- Feature: New tab for data sources and hosts exposing recorded metrics in the metric index for ingestion lag and event lag performances
- Feature: Provides metric host search capabilities with msearch button when clicking on a host metric (Splunk 8.x required), which is used as well for Elastic mstats sources
- Feature: Improved readability of high lagging seconds records by using duration formatting rendering automatically
- Fix: Flipping state detection failure for Elastic shared and dedicated sources due to regression introduced in trackMe 1.2.13

- Fix: Outliers table view might seem truncated with high volume sources, improve rendering by using thousands and millions units for high volume sources
- Fix: Outliers detection framework issues rendering current outliers accordingly to the outliers configuration for that entity
- Fix: Outliers detection framework issues generating metrics for some periods and failing to render the expected results
- Fix: Under some specific conditions, blacklist sub-searches at the tstats / mstats first pipeline levels end in error and generated high skipped scheduled rate, the root searches now use the same technique than whitelisting
- Fix: For metric host monitoring, off line hosts are constantly generating flipping status detection while this should happen once and be properly preserved over iterations
- Fix: UI does not honour search parameters and constraints for Elastic sources when clicking on the search button in modal windows

5.1.45 Version 1.2.15

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

release notes:

- Fix: error in metric hosts rendering results which are not exposing the full list of metrics per entity in the UI

5.1.46 Version 1.2.14

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

release notes:

- Fix: error in metric hosts rendering results which are duplicated in the UI after their expansion

5.1.47 Version 1.2.13

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

release notes:

- Fix: Flipping status detection should exclude the short term trackers time range for data sources and hosts
- Fix: Avoids conflicts between data handled in the short term and long term data trackers, by restricting the long term scope out of the short term scope in a improved fashion
- Fix: Long term data trackers calls via the UI should respect the same earliest and latest definition than the scheduler does
- Feature: Enhanced modal window for auto lagging definition for data sources and hosts

5.1.48 Version 1.2.12

- unpublished

5.1.49 Version 1.2.11

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

release notes:

- Feature: New data parsing quality tab, flipping status tab and audit changes tab per entity when applicable to provide quick and fast visibility on a per entity level
- Feature: Design improvements for the status message tab in modal windows which appears now with a new color scheme
- Feature: Provides Outliers span definition capability, the span value to be used for outliers rendering purposes can now be customised per entity
- Feature: Automatically handle metrics re-generation when an entity outliers period calculation is changed
- Feature: Acknowledge icon scheme when Ack is active, improve Ack workflow
- Fix Issue #96: “click save”, but there is no “save”
- Fix: SLA single calculation can show 0% if there are no previous records in audit flipping status and status has changed to non green
- Fix: Remove useless stats call in metric report savedsearch which impacts its performance
- Change: Provides and call a macro per builtin alert to allow customisation of the fields order in the alert results
- Change: Add app.manifest from packaging toolkit to ease dependencies and target workloads deployment

5.1.50 Version 1.2.10

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

release notes:

- Feature: Improved flipping statuses detection workflow, with immediate detection and deprecation of the dedicated flipping statuses tracker and associated collections
- Feature: UI improvements with change to multiselect form input for most of the selectors
- Fix: Flipping statuses table in main UI is not ordered by latest events
- Fix: Error in Elastic source simulation UI, in some conditions, a wrong data_name appears in the table which incorrectly claims that the data source already exists
- Fix: Elastic sources do not honour data_index and data_sourcetype definition, this does not impact the reliability of the results but this impacts sources visibility in the UI when using whitelists / blacklists
- Fix: For data hosts, several information are not properly preserved over tracker iterations, such a custom outliers configuration
- Fix: For data hosts, outlier event count record is not properly aggregated and is summed continuously over time rather a 4 hours event count recording
- Fix: Per entity refresh when outlier modification is saved should run over 4 hours period, and should filter results on the selected entity only
- Fix: UI input selectors for metric hosts should not show content for non whitelisted indexes if whitelists are being used
- Fix: Clean up of various objects which were deprecated in V1.2.x

5.1.51 Version 1.2.9

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

release notes:

- Fix: mcollect syntax compatibility issues with Splunk 7.2.x/7.3.x
- Fix: status_message fields shows N/A for translated last lagging value for data objects, and does not show up for metric objects
- Fix: switch from latest to max for outliers over time calculation, graphical rendering side effects introduced in 1.2.8 with mcollect switch

5.1.52 Version 1.2.8

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

release notes:

- Feature: Design and performances major improvements in the outlier detection workflow with metric based index and mcollect approach, to proper handle any high scale environments
- Feature: Major improvements in UI performance and stability, specially designed and qualified for very high scale environments and a high numbers of entities
- Feature: flipping status collection switches from KVstore based to summary index based for better performances and design at high scale
- Feature: improved workflow for SLA management and calculation based on the summary data
- Fix: Version 1.2.x introduced failures in the management of metric hosts, where detection of entirely inactive entities was not behaving as required
- Fix: hard coded metric index name in the trackme_summary_investigator_mstats macro

5.1.53 Version 1.2.7

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

TrackMe requires a summary index (defaults to trackme_summary) and a metric index (defaults to trackme_metrics): https://trackme.readthedocs.io/en/latest/configuration.html

release notes:

- Feature: Design and performances major improvements in the outlier detection workflow with metric based index and mcollect approach, to proper handle any high scale environments
- Feature: Major improvements in UI performance and stability, specially designed and qualified for very high scale environments and a high numbers of entities
- Feature: flipping status collection switches from KVstore based to summary index based for better performances and design at high scale
- Feature: improved workflow for SLA management and calculation based on the summary data
- Fix: Version 1.2.x introduced failures in the management of metric hosts, where detection of entirely inactive entities was not behaving as required

5.1.54 Version 1.2.5

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

release notes:

- Fix: conflict with Enterprise Security due to the tstats macro defined in TrackMe when co-located, macro renamed to avoid this issue
- Fix: cancel button in Elastic sources main modal, review help content

5.1.55 Version 1.2.4

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

release notes:

- Fix: Remove useless lookup call in the data hosts view which impacts search time performance in large scale environments

5.1.56 Version 1.2.3

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

release notes:

- Feature: Allows running the tracker directly after the Elastic source creation within the modal creation window (feature introduced in version 1.2.0)

5.1.57 Version 1.2.2

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

release notes:

- Feature: TrackMe goes with a touch of Machine Learning! Automatically detect outliers in the event distribution based on the summary investigator, a new internal workflow that records and detects a suspicious decrease of events over time based in the outliers detection method.

- Feature: Improved UI, Donuts charts completing the exposing of statuses, multi tabs in modal windows to navigate through the views, new outliers detection view, new message status view.
- Feature: Elastic tracker concept introduction, create and manage any kind of virtual data sources depending on your needs and requirements using any of the main Splunk search commands available (raw, tstats, from, mstats).
- Fix: collections monitoring are limited to 50K entries #80
- Fix: Modification of objects via the UI do not preserve some fields during KVstore update #81
- Change: fix app.py to avoid Readiness App warning, update Splunk Python SDK splunklib to very last version
- Fix: red donut chart error in metric hosts, rounding not effective of ingestion lag, donut red other priority serie appears in orange (v1.2.0 introduced)

5.1.58 Version 1.2.1

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

release notes:

- Feature: TrackMe goes with a touch of Machine Learning! Automatically detect outliers in the event distribution based on the summary investigator, a new internal workflow that records and detects a suspicious decrease of events over time based in the outliers detection method.
- Feature: Improved UI, Donuts charts completing the exposing of statuses, multi tabs in modal windows to navigate through the views, new outliers detection view, new message status view.
- Feature: Elastic tracker concept introduction, create and manage any kind of virtual data sources depending on your needs and requirements using any of the main Splunk search commands available (raw, tstats, from, mstats).
- Fix: collections monitoring are limited to 50K entries #80
- Fix: Modification of objects via the UI do not preserve some fields during KVstore update #81
- Change: fix app.py to avoid Readiness App warning, update Splunk Python SDK splunklib to very last version
- Fix: red donut chart error in metric hosts, rounding not effective of ingestion lag (v1.2.0 introduced)

5.1.59 Version 1.2.0

CAUTION:

This is a new main release branch, TrackMe 1.2.x requires the deployment of the following dependencies:

- Semicircle Donut Chart Viz, Splunk Base: https://splunkbase.splunk.com/app/4378
- Splunk Machine Learning Toolkit, Splunk Base: https://splunkbase.splunk.com/app/2890

release notes:

- Feature: TrackMe goes with a touch of Machine Learning! Automatically detect outliers in the event distribution based on the summary investigator, a new internal workflow that records and detects a suspicious decrease of events over time based in the outliers detection method.

- Feature: Improved UI, Donuts charts completing the exposing of statuses, multi tabs in modal windows to navigate through the views, new outliers detection view, new message status view.
- Feature: Elastic tracker concept introduction, create and manage any kind of virtual data sources depending on your needs and requirements using any of the main Splunk search commands available (raw, tstats, from, mstats).
- Fix: collections monitoring are limited to 50K entries #80
- Fix: Modification of objects via the UI do not preserve some fields during KVstore update #81
- Change: fix app.py to avoid Readiness App warning, update Splunk Python SDK splunklib to very last version

5.1.60 Version 1.1.16

- change: Decommission of the two auto mode tracker reports, these searches were designed to automatically define a potential value for the week days monitoring, therefore the searches can have a potential cost in term of resources without providing a key value justifying it.

5.1.61 Version 1.1.15

- feature: Introducing the maintenance mode feature, which allows to enable / schedule / disable the maintenance mode to silence all alerts during a scheduled maintenance window
- fix: Refresh buttons should refresh header main information for the entities too

5.1.62 Version 1.1.14

- unpublished

5.1.63 Version 1.1.13

- feature: Introducing inter-updates automatic refresh, operations that involve updates (modification of the max lag, etc) now dynamically refresh the entity drilldown view and related tokens, which prevents and automatically fixes conflicts during entity updates within the UI.
- feature: Introducing on demand auto determination of entity max lagging for data sources and hosts, based on either the percentile 95 or average lagging observed for that entity
- fix: minor fixes and code improvements

5.1.64 Version 1.1.12

- fix: SLA calculation is incorrect, this release fixes and improves the SLA calculation logic
- feature: Monitoring state auto disablement, provides a customizable macro logic that automatically disables the monitoring state of a data source, data host or metric host that has not actively sent data to Splunk since a given amount of days, by default 45 days

5.1.65 Version 1.1.11

- feature: Enrichment tags is a new feature available for data and metric hosts which allows you to provide automatic access to your assets context within TrackMe (Enterprise Security assets, custom CMDB data, etc)
- feature: Provides update comment capabilities for acknowledgments

5.1.66 Version 1.1.10

- fix: TrackMe admin members that are not admin cannot access to the audit collection content due to missing role statement in Metadata
- change: Change priority color code scheme to avoid confusion with object statuses

5.1.67 Version 1.1.9

- fix: Prevents data source identity card note failing if note contains double quotes (leads the underneath Splunk search adding to the collection to fail)
- fix: Reduce the maximal number of tables count in logical group show group table modal window, to avoid large number of groups hidden next pages
- fix: Refresh in modal window does not refresh SLA single forms

5.1.68 Version 1.1.8

- fix: SLA incorrect calculation, improvements and corrections in calculating the percentage of time spent in green/blue mode
- feature: Implement time based approach for SLA calculation restriction, provides time range picker in QOS dashboard

5.1.69 Version 1.1.7

- fix: Blacklist modal windows might under some resolution not be entirely visible, reduce height and max count table
- fix: Acknowledgment expiration is not honoured properly

5.1.70 Version 1.1.6

- feature: Introducing the SLA compliance reporting dashboard and features, providing analytic over the level of compliance based on the time objects have spent in red state (flipping mode detection)
- feature: Allows entering an update note for logging and notification purposes when a modification a KVstore entry is made via the UI
- feature: Regular expression support for data sources and host blacklisting entries
- feature: Pretty parse and print json objects in audit changes
- feature: Provides index and host blacklisting features for metric hosts monitoring
- feature: New tracker "TrackMe - Audit change notification tracker" which is due to be used for dedicated team work for updates notification (Slack…)

- change: Increase default retention for flipping states KVstore collection from 3 months to 6 months
- fix: Provides KVstore entry modification window for entity deletion to allow note update
- fix: Blue state icon will not show up in flipping status tab
- fix: Improvements in initial discovery detection for flipping status and SLA calculation purposes

5.1.71 Version 1.1.5

- fix: Previously added data sources or hosts can under some conditions appear with no state icon if status remained red and added in the collections before collecting last data ingestion statistics

5.1.72 Version 1.1.4 (unpublished)

- fix: Previously added data sources or hosts can under some conditions appear with no state icon if status remained red and added in the collections before collecting last data ingestion statistics

5.1.73 Version 1.1.3

- feature: Creation of an additional blue status, used for data hosts and metric hosts managed in a logical group when logical group monitoring conditions are met but entity is not green
- fix: Improved dynamic icon messages (reference the current latency when state is red)
- change: Increased default tolerance for data indexed in the future detection macro
- change: limit embedded charts searches overhead for data sources (do not split by host which limits accuracy but slightly improves searches performance in large environments)

5.1.74 Version 1.1.2

- fix: Under some circumstances, the last flipping status and date fail to be properly updated in the collections due to a weakness in the merging process
- fix: data_source modal window embedded chart should split by host in a first pipeline level for better lagging calculation accuracy
- fix: modal window embedded chart searches should refer to the tstats macro for consistency
- feature: Add audit view for KVstore collections

5.1.75 Version 1.1.1

- fix: Long term trackers should use latest time in the future too
- fix: New trackme_data_host_rule_filter macro does not show up properly in manage UI

5.1.76 Version 1.1.0

- feature: Better lagging management handling by storing and comparing both event based lagging and ingestion based lagging for multi-factor status definition

- feature: Detection of data indexed in the future, data sources or hosts indexing in the future appear as orange state with a dynamic icon message
- feature: Alert acknowledgment improvements, allows selecting an extended period for acknowledgment
- feature: Provides by default a collection based search rather than a Meta search based approach (dropdown selector in UI) for better performances on large deployments
- feature: Store first time seen and eventcount base for further use
- feature: Provides a rex based filter and length condition to avoid taking incorrect hosts in consideration
- change: Refresh default is now defined to 5 min instead of 1 min by default
- fix: Ensure results coherence with various lookup command calls used for enrichment purposes where never more than one match should be achieved
- fix: Various fixes

5.1.77 Version 1.0.39

- fix: minor audit changes logging improvements for metric SLA policies edition

5.1.78 Version 1.0.38

- fix: Error in TrackMe Mobile dashboard for summary not green statuses for metric hosts (count not green counts green metrics)

5.1.79 Version 1.0.37

- feature: Introducing the logical group concept which allows grouping data hosts and metric hosts in groups of clusters to manage use cases such as active / passive appliances which passive members do not actively generate data
- feature: Icon message are now dynamics and provide inline information describing the reason of the status
- feature: Collection navigation menu to expose quick access to raw KVstore collections content
- fix: Bad modal cancel action for week days (detailed per day selection) for data host monitoring

5.1.80 Version 1.0.36

- feature: Introducing the active alert acknowledgement feature, provides a framework to acknowledge an active alert which will inhibits generating new alerts while continuing to monitor and investigate in the UI.
- feature: Identity card improvements, allow existing identity card records to be associated with sources within the UI

5.1.81 Version 1.0.35

- fix: Ops indexers queues issue, first queue should be splunk tcpin queue

5.1.82 Version 1.0.34

- feature: introduction of the concept of source identity card, allows defining and store a documentation link and note for data sources, which identity cards are made available automatically via the UI and via the OOTB alert. Identity cards records can be created, maintained and delete via the UI.
- feature: increase default size of modal windows
- feature: fixed charts color for data sources and data hosts events vs lag embedded charts
- feature: add last 48 hours in link time selectors

5.1.83 Version 1.0.33

- fix: Avoids post processed searches in the Mobile dashboard, better single form placement for Apple TV rendering

5.1.84 Version 1.0.32

- fix: Performance issues with TrackMe mobile dashboard on mobile devices
- fix: TrackMe does not honour indexes whitelisting for metric hosts
- fix: Add metric host lookup in initial configuration load check operation
- fix: Wrong message for flush of metric KVstore collection
- feature: Remove management features from main UI to be transferred to a second management UI available from the nav menu

5.1.85 Version 1.0.31

- fix: Regression in flipping state introduced by metric implementation, does not trigger anymore for events indexes
- feature: Add auditing view to report on application scheduling search workload
- feature: Nav menus re-organized

5.1.86 Version 1.0.30

- fix: Splunk Mobile Dashboard does not honour whitelist and blacklists for data sources

5.1.87 Version 1.0.29

- fix: errors in Splunk Mobile dashboard (Any priority SLA alerts singles do not filter on red state)
- fix: better table rendering in Splunk Mobile dashboard for metric hosts

5.1.88 Version 1.0.28

- fix: collection key id retrieval fails if a metric category has been blacklisted for an existing object

5.1.89 Version 1.0.27

- fix: appinspect failure with metric_host variable replacement in “trackMe - metric per host table report”

5.1.90 Version 1.0.26

- fix: appinspect failure with metric_host variable replacement in “trackMe - metric host live report” report

5.1.91 Version 1.0.25

- feature: Introducing support for metric store availability monitoring with metric hosts and granular detection of metric availability failure and latency
- feature: Refresh button in all modal windows, improved placements for buttons, improved navigation coherence between modal windows
- fix: data host modal embedded charts and table should honour tstats main filter, whitelists and blacklists
- fix: Improved Mobile dashboard

5.1.92 Version 1.0.24

- fix: appinspect failure to local=true in commands.conf which is not required when chunked = true

5.1.93 Version 1.0.23

- fix: error in lib path call to the new custom command for whitelisting

5.1.94 Version 1.0.22

- feature: Whitelisting major improvement with UI supported and driven whitelisting of indexes at data discovery and search time (Issue #27)
- feature: Improve builtin choices for time input link selection within modal windows
- feature: Abstract tracker saved searches to remove useless code redundancy
- fix: Remove auto-refresh search link for searches which shouldn’t be refreshed automatically (audit changes & flip, various collection management)
- fix: Drilldown on any priority entities in alert should define monitored_state to enabled
- fix: Monitor split share percentage error (Single forms shall share 25% each)
- fix: Lagging class auditing can register an incorrect type of operation
- fix: All time time range picker will not work for audit changes & status flipping
- fix: Auto refresh set to none has random side effects on embedded chart loading, fixed by none set to long period
- fix: Switched from default last 7 days to last 24 hours in audit and status flipping UIs
- fix: TrackMe Mobile view does not honour blacklists

5.1.95 Version 1.0.21

- feature: Introducing a priority (low/medium/high) concept to ease granular alerting of data sources and hosts
- feature: Home landing page reviewed to expose data sources and host and any alert, and with high priority in alerts
- feature: Colored vignette design in modal window to ease investigating statuses
- feature: Default OOTB alerts now filter on red, and medium (default priority) or high priority entities
- feature: Improvement of OOTB alerts (outputs by default human readable time stamps for key fields)
- feature: TrackMe Mobile dashboard for dark theme summary view compatible with Splunk Mobile Experience (Apple TV, Mobile)
- feature: Improved navigation for unified modification modal windows
- feature: Drilldown on single forms, defines filtering based on the single form purpose
- feature: Manage and configure tab in main UI, access to reset collections functions or key macros definition and short cuts
- fix: data sources that came of scope might loose time context upon time and returned as green state
- fix: over time, trackers can re-add old entries due to flipping state cross-searches
- fix: data_host_state icon shown as empty if state=orange due to mismatch in macro eval state icon for data_host
- fix: trackers should refer to the tstats macro

5.1.96 Version 1.0.20

- fix: Issue #34: Lagging class override for data_source is not registered properly

5.1.97 Version 1.0.19

- Fix: Issue #32, if the data is offline for a long period that is out of the scope of the long term trackers, the last lag seen in seconds is not properly updated at each run time of the trackers.

5.1.98 Version 1.0.18

- Fix: data index dropdown shouldn't itself be filtering on selected index

5.1.99 Version 1.0.17

- Feature: Unified update modal Windows for data source and host modification
- Feature: Suspension effect when modification of entity is registered
- Fix: Prevent bootstrap button to remain focused once clicked

5.1.100 Version 1.0.16

- Fix: Dropdown populating issues caused by 1.0.15 update

5.1.101 Version 1.0.15

- Feature: Provide a time range picker for audit flipping and audit changes investigations

5.1.102 Version 1.0.14

- Fix: Flipping chart over time should be stacked

5.1.103 Version 1.0.13

- Fix: Flipping object dropdown populating issue

5.1.104 Version 1.0.12

- Fix: Flipping audit tracker is not filtering on monitored entities

5.1.105 Version 1.0.11

- Feature: Introducing status flipping audit and investigation to record and report on historical changes of data sources and hosts status

5.1.106 Version 1.0.10

- Feature: Provides a trackme_admin role with relevant default meta configuration to allow granular access control for non admin users

5.1.107 Version 1.0.9

- Fix: bad reference to a group in default Meta

5.1.108 Version 1.0.8

- Feature: Add dropdown filters for data host monitoring (data_index, data_sourcetype)
- Feature: Improve filtering logics

5.1.109 Version 1.0.7

- Fix: Missing lagging class button in data sources view

5.1.110 Version 1.0.6

- Fix: Minor UI fixes
- Fix: Remove include_reduced_buckets for Splunk pre 7.3.x compatibility

5.1.111 Version 1.0.5

- Feature: Implementation of audit changes
- Feature: Unify blacklist buttons in main modal
- Feature: Provides entities deletion permanent or temporary options to avoid re-creation of unwanted entities
- Feature: Add last ingest column in data sources and hosts

5.1.112 Version 1.0.4

- Fix: case issue when hosts are seen in both lower and upper case, or a mix or them

5.1.113 Version 1.0.3

- Fix: better bootstrap buttons alignment

5.1.114 Version 1.0.2

- Feature: custom lagging classes feature introduction
- Fix: provides detailed explanation about the reset collection button
- Feature: UI experience improvements

5.1.115 Version 1.0.1

- Fix: bad lookup referenced in host trackers

5.1.116 Version 1.0.0

- initial and first public release