

CCP Management 2.0.1

Managing

Date of publish: 2017-11-06

CLOUDERA

<https://docs.cloudera.com/>

Legal Notice

© Cloudera Inc. 2019. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 ("ASLv2"), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER'S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

- Managing Overview..... 4**
 - Update Properties.....4
 - Understanding ZooKeeper Configurations.....4
 - Managing Sensors.....5
 - Start a Sensor..... 5
 - Stop a Sensor.....6
 - Modify a Sensor..... 7
 - Delete a Sensor.....8
 - Start and Stop Parsers..... 9
 - Start and Stop Enrichments..... 11
 - Start and Stop Indexing..... 13
 - Prune Data from Elasticsearch..... 15
 - Tune Apache Solr..... 15
 - Back Up the Metron Dashboard..... 15
 - Restore Your Metron Dashboard Backup..... 16

Managing Overview

Cloudera Cybersecurity Platform (CCP) powered by Apache Metron provides you with several options for managing your system. Before you perform any of these tasks, you should become familiar with CCP data throughput.

Update Properties

Cloudera Cybersecurity Platform (CCP) configuration information is stored in Apache ZooKeeper as a series of JSON files.

You can populate your ZooKeeper configurations from multiple locations:

- \$METRON_HOME/config/zookeeper
- Management UI
- Ambari
- Stellar REPL

Because Ambari explicitly manages some of these configuration properties, if you change a property explicitly managed by Ambari from a mechanism outside of Ambari, such as the Management UI, Ambari is aware of this change and overwrites it whenever the Metron topology is restarted. Therefore, you should modify Ambari-managed properties only in Ambari.

For example, the `es.ip` property is managed explicitly by Ambari. If you modify `es.ip` and change the `global.json` file outside Ambari, you will not see this change in Ambari. Meanwhile, the indexing topology would be using the new value stored in ZooKeeper. You will not receive any errors notifying you of the discrepancy between ZooKeeper and Ambari. However, when you restart the Metron topology component via Ambari, the `es.ip` property would be set back to the value stored in Ambari.

Following are the Ambari-managed properties:

Table 1: Ambari-Managed Properties

Global Configuration Property Name	Ambari Name
<code>es.clustername</code>	<code>es_cluster_name</code>
<code>es.ip</code>	<code>es_hosts</code>
<code>es.port</code>	<code>es_port</code>
<code>es.date.format</code>	<code>es_date_format</code>
<code>profiler.period.duration</code>	<code>profiler_period_duration</code>
<code>profiler.period.duration.units</code>	<code>profiler_period_units</code>
<code>update.hbase.table</code>	<code>update_hbase_table</code>
<code>update.hbase.cf</code>	<code>update_hbase_cf</code>
<code>geo.hdfs.file</code>	<code>geo_hdfs_file</code>

Understanding ZooKeeper Configurations

ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services.

ZooKeeper configurations should be stored on disk in the following structure starting at `$METRON_HOME/bin/zk_load_configs.sh`:

global.json	The global config
sensors	The subdirectory containing the sensor enrichment configuration JSON (for example, snort.json or bro.json)

By default, the sensors directory as deployed by the Ansible infrastructure is located at `$METRON_HOME/config/zookeeper`.

Although the configurations are stored on disk, they must be loaded into ZooKeeper to be used. You can use the utility program `$METRON_HOME/bin/zk_load_config.sh` to load configurations into ZooKeeper.

-f,--force	Force operation
-h,--help	Generate Help screen
-i,--input_dir <DIR>	The input directory containing the configuration files named, for example <code>\$source.json</code>
-m,--mode <MODE>	The mode of operation: DUMP, PULL, PUSH
-o,--output_dir <DIR>	The output directory that stores the JSON configuration from ZooKeeper
-z,--zk_quorum <host:port,[host:port]*>	The ZooKeeper Quorum URL (zk1:port,zk2:port,...)

See the following list for examples of usage: Usage examples:

- To dump the existing configs from ZooKeeper on the single-ode vagrant machine:
`$METRON_HOME/bin/zk_load_configs.sh -z node1:2181 -m DUMP`
- To push the configs into ZooKeeper on the single-ode vagrant machine:
`$METRON_HOME/bin/zk_load_configs.sh -z node1:2181 -m PUSH -i $METRON_HOME/config/zookeeper`
- To pull the configs from ZooKeeper to the single node vagrant machine disk:
`$METRON_HOME/bin/zk_load_configs.sh -z node1:2181 -m PULL -o $METRON_HOME/config/zookeeper -f`

Managing Sensors

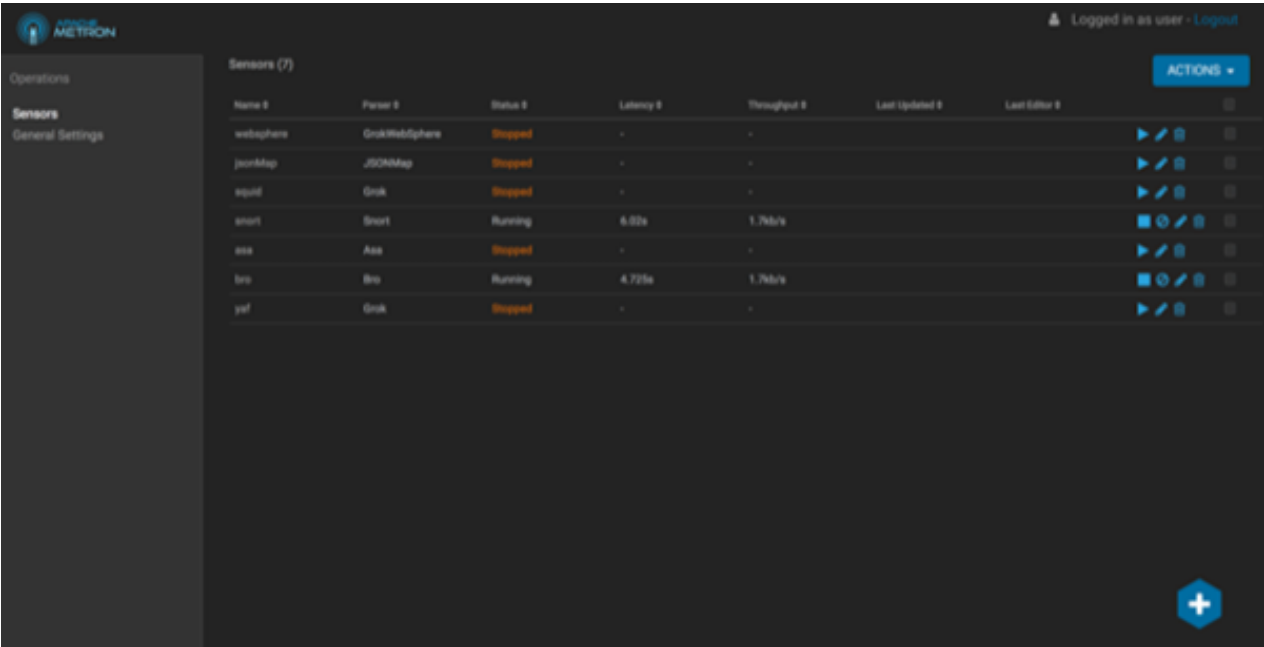
You can manage your sensors and associated topologies using either the Hortonworks Cybersecurity Platform (HCP) Management user interface or the Apache Storm UI. The following procedures use the HCP Management UI to manage sensors. For information about using Storm to manage sensors, see the Storm documentation.

Start a Sensor

After you install a sensor, you can start it using Management user interface.

Procedure

From the main window, click  (start) in the  (tool bar) on the right side of the window.



Starting the sensor might take a few minutes. When the operation completes successfully, the Status value for the sensor changes to Running.

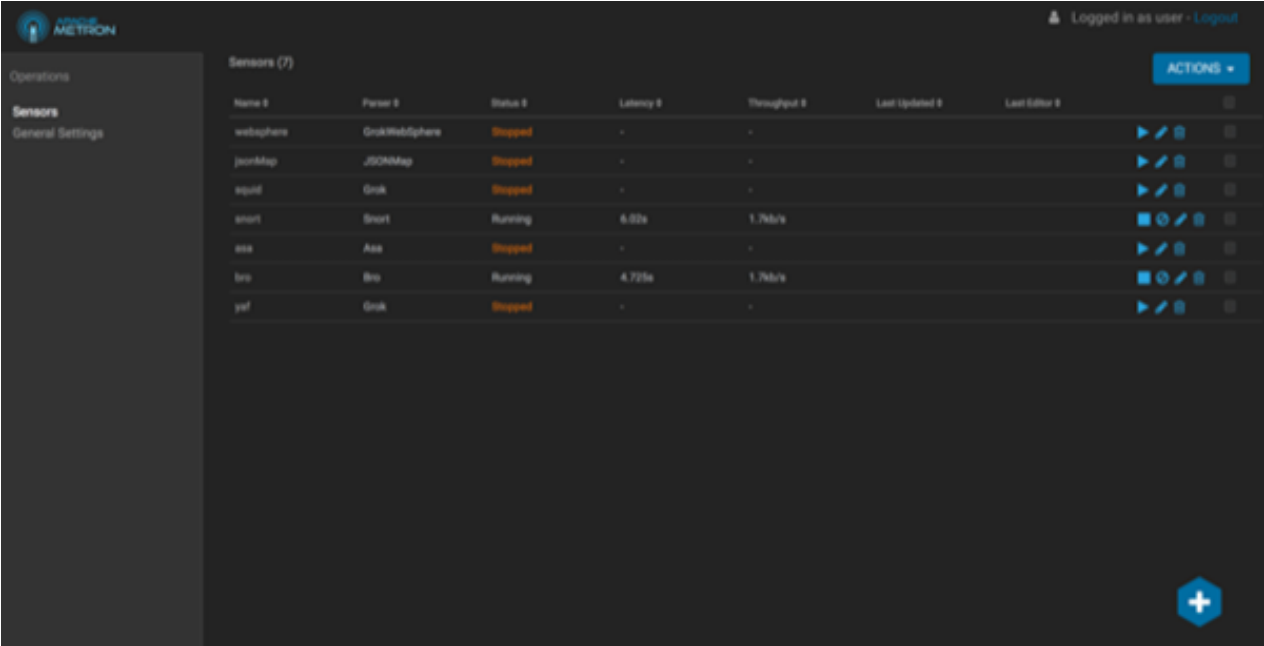
Stop a Sensor

After you install a sensor, you can stop it using the Management user interface.

Procedure



From the main window, click  (stop) in  (tool bar) on the right side of the window.



Stopping the sensor might take a few minutes. When the operation completes successfully, the Status value for the sensor changes to Stopped.

Modify a Sensor

You can modify any sensor listed in Cloudera Cybersecurity Platform (CCP) Management user interface.

Procedure

1.



From the **Operations** panel of the main window, select **Sensors**. click (edit) for the sensor you want to modify.

The Management UI displays a panel populated with the sensor configuration information:

bro

NAME *

bro

Kafka Topic Exists. Enabling

PARSER TYPE *

bro

SCHEMA

TRANSFORMATIONS 0

ENRICHMENTS 3

THREAT INTEL 2

THREAT TRAGE

RULES 0

SAVE

CANCEL

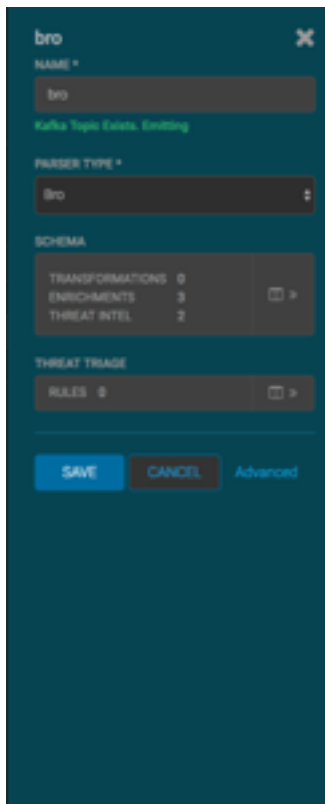
Advanced

2.



Click (edit) for the sensor you want to modify.

The Management UI displays a panel populated with the sensor configuration information:



The screenshot shows a configuration window for a sensor named 'bro'. The window has a dark teal background. At the top, there's a title bar with 'bro' and a close button. Below the title bar, there's a 'NAME' field with 'bro' entered. A green message 'Kafka Topic Exists. Enabling' is displayed. Below that, there's a 'PARSER TYPE' dropdown menu with 'bro' selected. Under the 'SCHEMA' section, there's a table with three rows: 'TRANSFORMATIONS' with value '0', 'ENRICHMENTS' with value '3', and 'THREAT INTEL' with value '2'. To the right of the table is a button with a plus icon and a right arrow. Below the schema section, there's a 'THREAT TRIAGE' section with a 'RULES' field showing '0' and a similar button with a plus icon and a right arrow. At the bottom, there are three buttons: 'SAVE' (blue), 'CANCEL' (dark grey), and 'Advanced' (light blue).

3. Modify the following information for the sensor, as necessary:

- Sensor name
- Parser type
- Schema information
- Threat triage information

4. Click **Save**.

Delete a Sensor

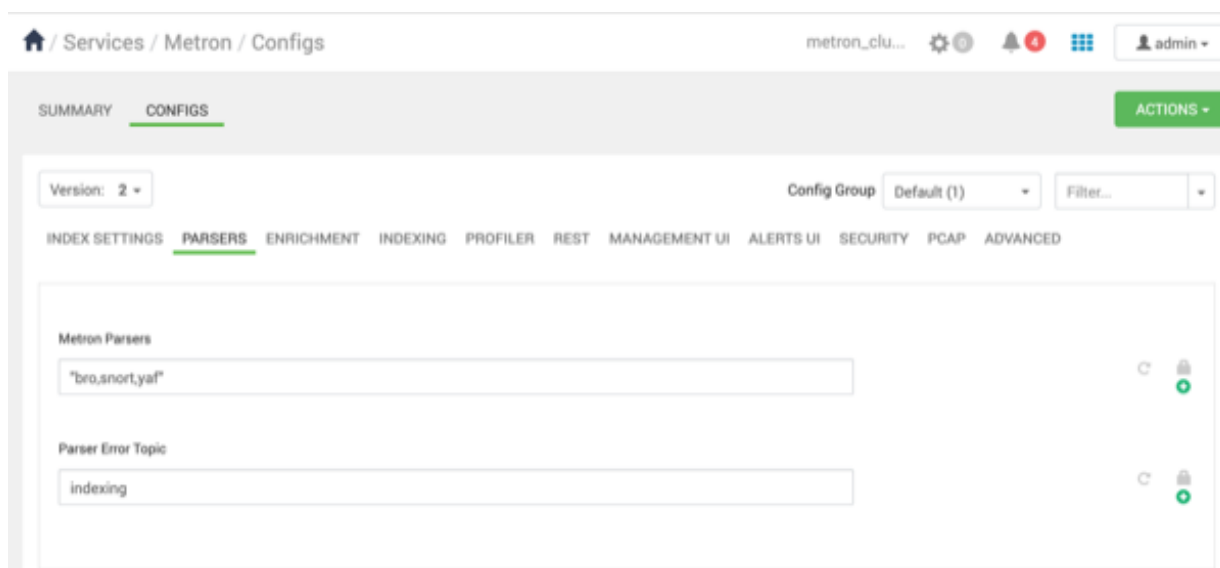
You can delete a sensor if you don't need it.

Before you begin

You must take the sensor offline before deleting it.

Procedure

1. In the Ambari user interface, click the **Services** tab.
2. Click **Metron** from the list of services.
3. Click **CONFIGS** and then click **PARSERS**.



4. Delete the name of the parser you want to delete from the **Metron Parsers** field.
5. Display the Management module.
6. Select the check box next to the appropriate sensor in the Sensors table.
You can delete more than one sensor by clicking multiple check boxes.
7. From the **ACTIONS** menu, select **Delete**.
The Management module deletes the sensor from ZooKeeper.
8. Finally, delete the json file for the sensor on the Ambari master node:

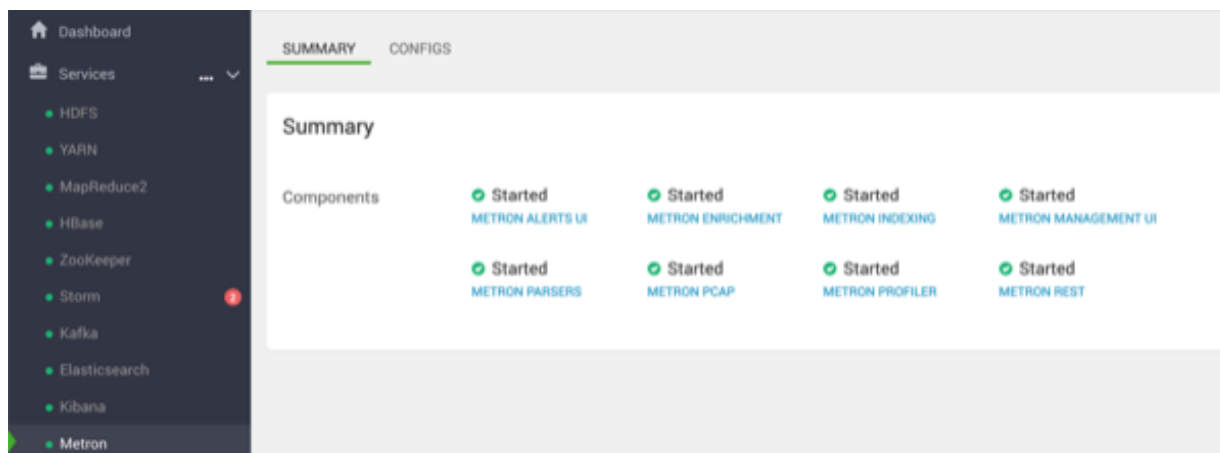
```
ssh $AMBARI_MASTER_NODE
cd $METRON_HOME/config/zookeeper/parser
rm $DATASOURCE.json
```

Start and Stop Parsers

You might want to stop or restart parsers as you refine your cybersecurity monitoring. You can easily stop and start parsers by using Ambari.

Procedure

1. Display the Ambari UI and navigate to **Services > Metron > SUMMARY**:



- Click **METRON PARSERS** to display the **Components** window.

The Components window displays a list of Metron hosts and which components reside on each host.

The screenshot displays the Ambari Components window. On the left, a list of services is shown with their status (green checkmark) and role (Master or Slave). The services include Kibana Server, Metron Alerts UI, Metron Enrichment, Metron Indexing, Metron Management UI, Metron Parsers, Metron PCAP, Metron Profiler, Metron REST, NameNode, Nimbus, ResourceManager, SNameNode, Storm UI Server, Timeline Service V2.0 Reader, ZooKeeper Server, DataNode, RegionServer, NodeManager, Supervisor, HBase Client, HDFS Client, MapReduce2 Client, YARN Client, and ZooKeeper Client. The right side of the window shows a summary of the selected host (node1) with various metrics and status information.

Service	Role	Status
Kibana Server / Kibana	Master	Running
Metron Alerts UI / Metron	Master	Running
Metron Enrichment / Metron	Master	Running
Metron Indexing / Metron	Master	Running
Metron Management UI / Metron	Master	Running
Metron Parsers / Metron	Master	Running
Metron PCAP / Metron	Master	Running
Metron Profiler / Metron	Master	Running
Metron REST / Metron	Master	Running
NameNode / HDFS	Master	Running
Nimbus / Storm	Master	Running
ResourceManager / YARN	Master	Running
SNameNode / HDFS	Master	Running
Storm UI Server / Storm	Master	Running
Timeline Service V2.0 Reader / YARN	Master	Running
ZooKeeper Server / ZooKeeper	Master	Running
DataNode / HDFS	Slave	Running
RegionServer / HBase	Slave	Running
NodeManager / YARN	Slave	Running
Supervisor / Storm	Slave	Running
HBase Client / HBase	Client	Running
HDFS Client / HDFS	Client	Running
MapReduce2 Client / MapReduce2	Client	Running
YARN Client / YARN	Client	Running
ZooKeeper Client / ZooKeeper	Client	Running

Summary

Hostname: node1
 IP Address: 127.0.0.1
 Rack: /default-rack
 OS: centos7 (x86_64)
 Cores (CPU): 4 (4)
 Disk: Data Unavailable
 Memory: 7.64GB
 Load Avg:
 Heartbeat: less than a minute ago
 Current Version: 3.1.0.0-78
 JCE Unlimited: true

- In the **Action** column, click ... next to **Metron Parsers/Metron**, choose **Restart/Stop** to change the status of the parser, then click **OK** in the **Confirmation** dialog box.

Ambari displays the **Background Operations** dialog box which provides the status of the operation.

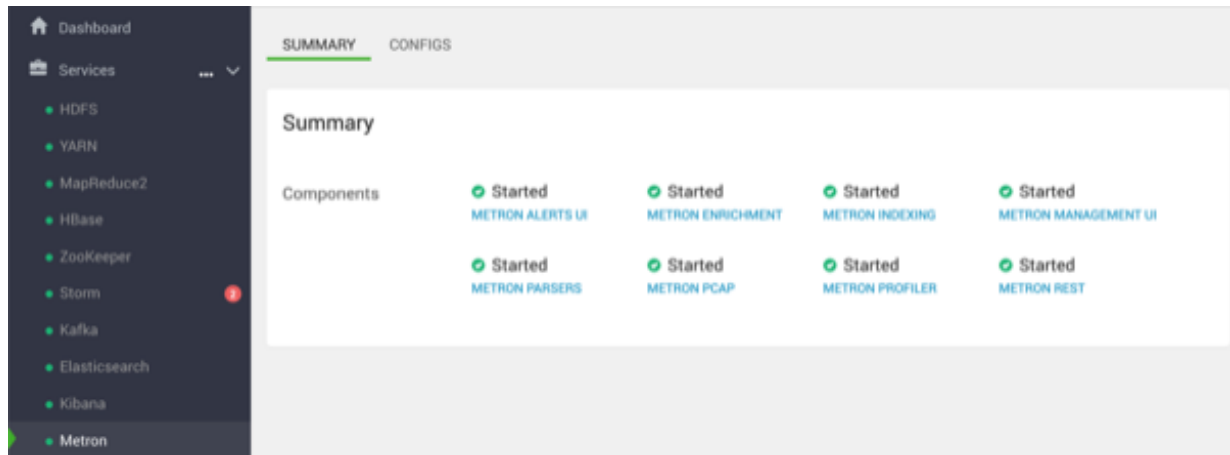
- Click **OK** to exit the **Background Operations** dialog box.

Start and Stop Enrichments

You might want to stop or start enrichments as you refine or focus your cybersecurity monitoring. You can easily stop and start enrichments by using Ambari.

Procedure

1. Display the Ambari tool and navigate to **Services > Metron > SUMMARY**.



2. Click **METRON ENRICHMENT** to display the **Components** window.
This window displays a list of CCP hosts and which components reside on each host.

The screenshot displays the Ambari Overview page. On the left, a table lists various services and their status. The 'Action' column for 'Metron Enrichment/Metron' shows a three-dot menu icon. Below the table is a 'Summary' section for the host 'node1'. On the right, a dashboard shows 'Network Usage' (3%) and 'Processes' (n/a). Below these are 'NameNode RPC' (0.61 ms) and 'NameNode Uptime' (1d 21h 24m).

Service	Component	Role	Status
Kibana Server	Kibana	Master	Running
Metron Alerts UI	Metron	Master	Running
Metron Enrichment	Metron	Master	Running
Metron Indexing	Metron	Master	Running
Metron Management UI	Metron	Master	Running
Metron Parsers	Metron	Master	Running
Metron PCAP	Metron	Master	Running
Metron Profiler	Metron	Master	Running
Metron REST	Metron	Master	Running
NameNode	HDFS	Master	Running
Nimbus	Storm	Master	Running
ResourceManager	YARN	Master	Running
SNameNode	HDFS	Master	Running
Storm UI Server	Storm	Master	Running
Timeline Service V2.0 Reader	YARN	Master	Running
ZooKeeper Server	ZooKeeper	Master	Running
DataNode	HDFS	Slave	Running
RegionServer	HBase	Slave	Running
NodeManager	YARN	Slave	Running
Supervisor	Storm	Slave	Running
HBase Client	HBase	Client	Running
HDFS Client	HDFS	Client	Running
MapReduce2 Client	MapReduce2	Client	Running
YARN Client	YARN	Client	Running
ZooKeeper Client	ZooKeeper	Client	Running

Summary

Hostname: node1
 IP Address: 127.0.0.1
 Rack: /default-rack
 OS: centos7 (x86_64)
 Cores (CPU): 4 (4)
 Disk: Data Unavailable
 Memory: 7.64GB
 Load Avg:
 Heartbeat: less than a minute ago
 Current Version: 3.1.0.0-78
 JCE Unlimited: true

Network Usage
 NameNode Heap: 3%

Processes
 NameNode CPU WIO: n/a
 NameNode RPC: 0.61 ms
 NameNode Uptime: 1d 21h 24m

3. In the **Action** column, click ... by **Metron Enrichment/Metron**, choose **Restart/Stop** to change the status of the Enrichments, then click **OK** in the **Confirmation** dialog box.

Ambari displays the **Background Operations** dialog box which provides the status of the operation.

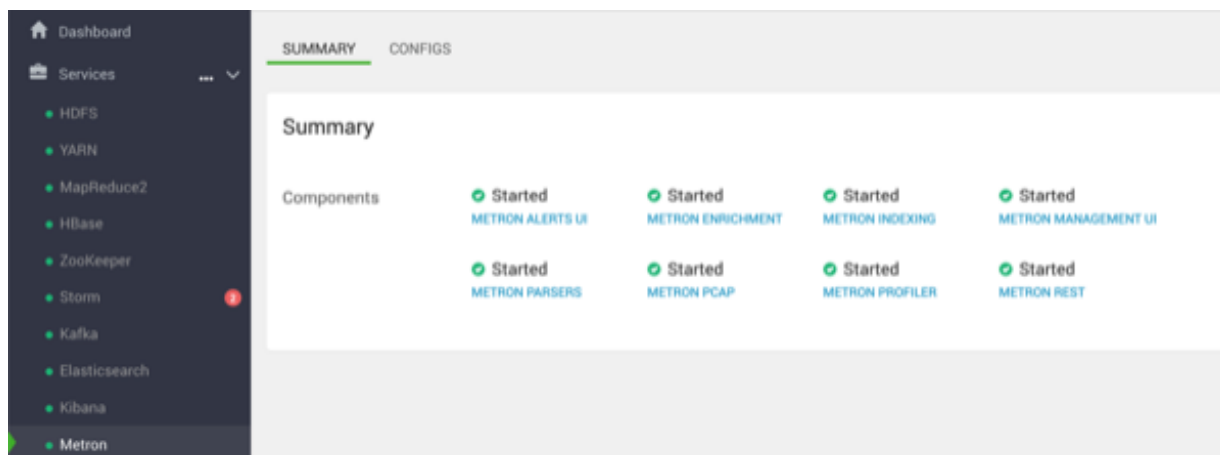
4. Click **OK** to exit the **Background Operations** dialog box.

Start and Stop Indexing

You might want to stop or start indexing as you refine or focus your cybersecurity monitoring. You can easily stop and start indexing by using Ambari.

Procedure

1. Display the Ambari tool and navigate to **Services > Metron > SUMMARY**.



2. Click **METRON INDEXING**.

This window displays a list of CCP hosts and which components reside on each host.

The screenshot displays the Ambari Overview page. On the left, a list of services is shown with their status (green checkmark) and role (Master or Slave). The services include Kibana Server, Metron Alerts UI, Metron Enrichment, Metron Indexing, Metron Management UI, Metron Parsers, Metron PCAP, Metron Profiler, Metron REST, NameNode, Nimbus, ResourceManager, SNameNode, Storm UI Server, Timeline Service V2.0 Reader, ZooKeeper Server, DataNode, RegionServer, NodeManager, Supervisor, HBase Client, HDFS Client, MapReduce2 Client, YARN Client, and ZooKeeper Client. Below the list is a 'Summary' section for the node 'node1' with details such as IP Address (127.0.0.1), Rack (/default-rack), OS (centos7 (x86_64)), Cores (CPU): 4 (4), Disk (Data Unavailable), Memory (7.64GB), Load Avg, Heartbeat (less than a minute ago), Current Version (3.1.0.0-78), and JCE Unlimited (true). On the right, a dashboard shows 'Network Usage' (NameNode Heap at 3%) and 'Processes' (NameNode CPU WIO at n/a). Other metrics include NameNode RPC (0.61 ms) and NameNode Uptime (1d 21h 24m).

Service	Role	Status
Kibana Server / Kibana	Master	Running
Metron Alerts UI / Metron	Master	Running
Metron Enrichment / Metron	Master	Running
Metron Indexing / Metron	Master	Running
Metron Management UI / Metron	Master	Running
Metron Parsers / Metron	Master	Running
Metron PCAP / Metron	Master	Running
Metron Profiler / Metron	Master	Running
Metron REST / Metron	Master	Running
NameNode / HDFS	Master	Running
Nimbus / Storm	Master	Running
ResourceManager / YARN	Master	Running
SNameNode / HDFS	Master	Running
Storm UI Server / Storm	Master	Running
Timeline Service V2.0 Reader / YARN	Master	Running
ZooKeeper Server / ZooKeeper	Master	Running
DataNode / HDFS	Slave	Running
RegionServer / HBase	Slave	Running
NodeManager / YARN	Slave	Running
Supervisor / Storm	Slave	Running
HBase Client / HBase	Client	Running
HDFS Client / HDFS	Client	Running
MapReduce2 Client / MapReduce2	Client	Running
YARN Client / YARN	Client	Running
ZooKeeper Client / ZooKeeper	Client	Running

Summary

Hostname: node1
 IP Address: 127.0.0.1
 Rack: /default-rack
 OS: centos7 (x86_64)
 Cores (CPU): 4 (4)
 Disk: Data Unavailable
 Memory: 7.64GB
 Load Avg:
 Heartbeat: less than a minute ago
 Current Version: 3.1.0.0-78
 JCE Unlimited: true

Network Usage

NameNode Heap: 3%

Processes

NameNode CPU WIO: n/a

NameNode RPC: 0.61 ms

NameNode Uptime: 1d 21h 24m

3. In the **Action** column, click ... next to **Metron Indexing**, then choose **Started/Stopped** to change the status of the Indexing, then click **OK** in the **Confirmation** dialog box.

Ambari displays the **Background Operations** dialog box.

4. Click **OK** to exit the **Background Operations** dialog box.

Prune Data from Elasticsearch

Elasticsearch provides tooling to prune index data through its Curator utility.

Procedure

1. Use the following command to prune the Elasticsearch data:

The following is a sample invocation that you can configure through Cron to prune indexes based on the timestamp in the index name.

```
/opt/elasticsearch-curator/curator_cli --host localhost delete_indices --
filter_list '
{
  "filtertype": "age",
  "source": "name",
  "timestring": "%Y.%m.%d",
  "unit": "days",
  "unit_count": 10,
  "direction": "older"
}'
```

Using name as the source value causes Curator to look for a timestring value within the index or snapshot name, and to convert that into an epoch timestamp (epoch implies UTC).

2. For finer-grained control over indexes pruning, provide multiple filters as an array of JSON objects to filter_list. Chaining multiple filters implies logical AND.

```
--filter_list
' [{"filtertype": "age", "source": "creation_date", "direction": "older", "unit": "days", "un
{"filtertype": "pattern", "kind": "prefix", "value": "logstash"} ] '
```

For finer-grained control over the indexes pruning that will be pruned, you can also provide multiple filters as an array of JSON objects to filter_list. Chaining multiple filters implies there is an implicit logical AND when chaining multiple filters.

```
--filter_list
' [{"filtertype": "age", "source": "creation_date", "direction": "older", "unit": "days", "un
{"filtertype": "pattern", "kind": "prefix", "value": "logstash"} ] '
```

Tune Apache Solr

To tune and customize Apache Solr, refer to the *Apache Solr Reference Guide*.

Back Up the Metron Dashboard

You can back up your Metron dashboard to avoid losing your customizations:

Procedure

To back up your Metron dashboard use the following command:

```
python packaging/ambari/metron-mpack/src/main/resources/common-services/
KIBANA/5.6.x/package/scripts/dashboard/dashboardindex.py \
  $ES_HOST 9200 \
  $SOME_PATH/dashboard.p -s
```

Restore Your Metron Dashboard Backup

You can restore a back up of your Metron dashboard by writing the Kibana dashboard to Solr or Elasticsearch.

Procedure

To restore a back up of your Metron dashboard, you can write the Kibana dashboard to Solr or Elasticsearch.

For example:

```
python packaging/ambari/metron-mpack/src/main/resources/common-services/  
KIBANA/5.6.x/package/scripts/dashboard/dashboardindex.py \  
    $ES_HOST 9200 \  
    $SOME_PATH/dashboard.p
```

Note that this overwrites the .kibana index.