

CCP Ambari Installation 2.0.1

Installing HCP with Ambari

Date of publish: 2017-11-06

CLOUDBERA

<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

Installing CCP Using Ambari.....	4
Prerequisites for an Existing Cluster.....	4
Specifications for Hadoop Cluster.....	4
Specifications for Metron Nodes.....	5
Set up the REST Application Database.....	5
Install CCP on an Ambari Cluster.....	6
Install HCP Ambari Management Pack.....	7
Install Solr.....	8
Install, Configure, and Deploy a HDP Cluster with CCP.....	11
Switch to LDAP Access Privileges.....	19
Import Apache Zeppelin Notebook Using Ambari.....	21
Streaming Data into CCP.....	21
Verify That CCP Deployed Successfully for Ambari Install.....	21
Open the Metron Dashboard.....	24
Opening the Management User Interface.....	24
Opening the Alerts User Interface.....	24
Troubleshooting Your Installation.....	25
Optimization Guidelines.....	25

Installing CCP Using Ambari

Installing Cloudera Cybersecurity Platform (CCP) using Apache Ambari uses both the graphic user interface of Ambari and the Metron user interface. Both of these tools promote a faster installation that preinstalls much of the configuration you need.

Prerequisites for an Existing Cluster

You can install Cloudera Cybersecurity Platform (CCP) on an Ambari-managed cluster running HDP 3.1.4 and Ambari 2.7.3 (or later). However, the cluster must meet requirements for both the Hadoop cluster and the Metron nodes.

Specifications for Hadoop Cluster

All Hadoop-related nodes running Cloudera Cybersecurity Platform (CCP) must meet operating system, HDP, and cluster requirements.

All Hadoop-related nodes must meet the following specifications:

- All cluster nodes must be running CentOS 7.x
- The cluster must be running HDP 3.1.4 managed by Ambari 2.7.3 (or later)
- The cluster must have a minimum of the following nodes:
 - Two Hadoop master nodes
 - Four Hadoop slaves nodes
 - One node for Ambari
- Each of the Hadoop Slave and Master nodes must meet the minimum specifications.
- The following services must be installed across the Hadoop Master and Slave nodes:
 - HDFS
 - HBase
 - Hive
 - ZooKeeper
 - Kafka
 - Storm
 - YARN
 - Spark 2.3.0 or later

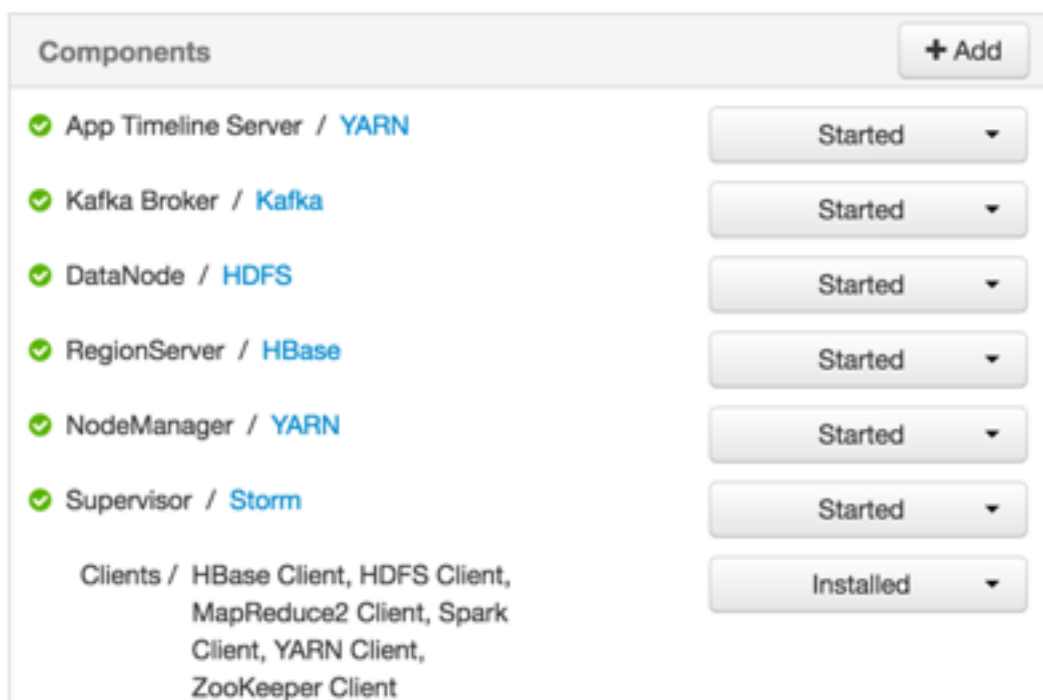
To determine the supported version for each service, refer to Ambari, and choose Admin > Stacks and Versions.

- Each of the following components must be installed on at least one node. The YARN ATS must be installed on the master node. All other services in the list should be installed on multiple nodes.



Note:

For security reasons, no other workloads should be running on the cluster.



Specifications for Metron Nodes

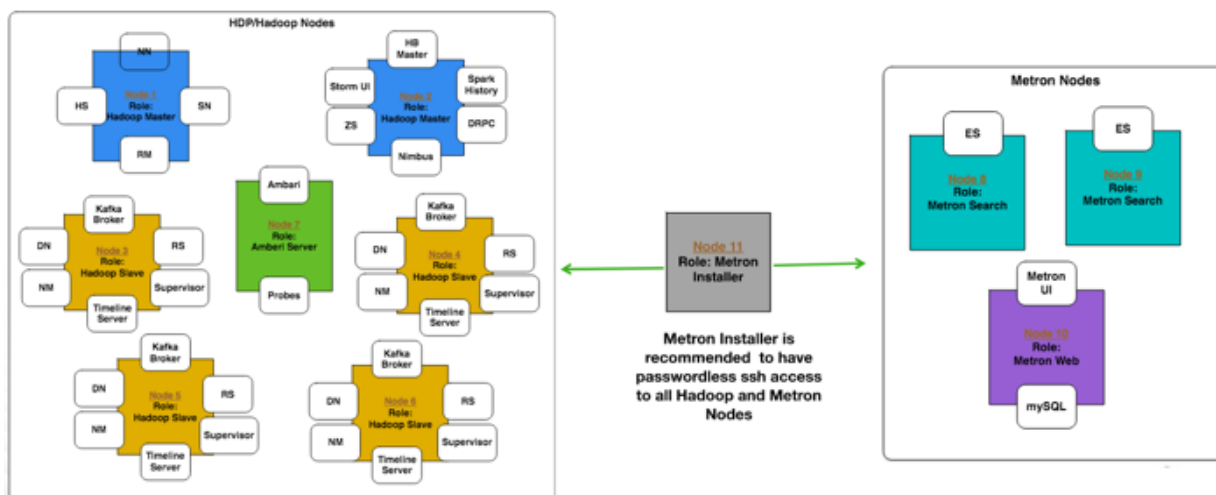
All Metron nodes must meet specifications for the number of nodes dedicated for Metron-specific components and the ability to access the nodes.

The Metron nodes must meet the following specifications:

- At least three nodes must be dedicated for Metron-specific components.
- You must have root access on all Metron nodes.

The following figure illustrates a sample deployment architecture based on the previous specifications:

Sample Deployment Architecture



Set up the REST Application Database

Prior to installing Cloudera Cybersecurity Platform (CCP), you must set up the REST application database.

Procedure

1. Connect to MySQL and create a Metron REST database:

```
mysql -uroot -p -e "CREATE DATABASE IF NOT EXISTS metronrest;"
```

2. Create a Metron user in MySQL with a password, then apply database access permission to the Metron user:

```
CREATE USER 'metron'@'$REST_HOST' IDENTIFIED BY 'Myp@ssw0rd';  
GRANT ALL PRIVILEGES ON metronrest.* TO 'metron'@'$REST_HOST';
```

3. Create user and authorities tables:

```
use metronrest;  
create table if not exists users(  
    username varchar(50) not null primary key,  
    password varchar(50) not null,  
    enabled boolean not null  
);  
create table authorities (  
    username varchar(50) not null,  
    authority varchar(50) not null,  
    constraint fk_authorities_users foreign key(username) references  
    users(username)  
);  
create unique index ix_auth_username on authorities (username,authority);
```

4. Add one or more users to the REST application:

```
use metronrest;  
insert into users (username, password, enabled) values ('your_username',  
    'your_password',1);  
insert into authorities (username, authority) values ('your_username',  
    'ROLE_USER');
```

5. Exit MySQL:

```
quit
```

6. Install the appropriate MySQL client library for your version of MySQL. For example:

```
cd $METRON_HOME/lib  
wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-  
java-5.1.41.tar.gz  
tar xf mysql-connector-java-5.1.41.tar.gz
```

7. To add additional users:

```
use metronrest;  
insert into users (username, password, enabled) values ('your_username',  
    'your_password',1);  
insert into authorities (username, authority) values ('your_username',  
    'ROLE_USER');  
commit;
```

Install CCP on an Ambari Cluster

Prior to installing the Cloudera Cybersecurity Platform (CCP), you must meet CCP's requirements for the cluster, Metron node, and Ambari server.

Before you begin

Prior to installing the HCP Ambari management pack, you must complete the following:

- Meet all of the cluster specifications listed in Specifications for Hadoop Cluster.
- Meet all of the metron node specifications listed in Specifications for Metron Nodes.

For ease of use, consider adding your host entries to your `/etc/hosts` file.

Procedure

1. Download and install Ambari.

For example:

```
wget -nv http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.7.3.0/ambari.repo -O /etc/yum.repos.d/ambari.repo
```

2. Set up the Ambari server.

```
ambari-server setup -s
```

Use the `-s` parameter to install Oracle JDK as part of the Ambari set up.

Install HCP Ambari Management Pack

A Cloudera Cybersecurity Platform (CCP) Ambari management pack bundles service definitions, stack definitions, and stack add-on service definitions so they do not need to be included with the Ambari core functionality and can be updated in between major releases. You can use the HCP management pack to install Metron, plus the parser topologies, indexing topologies, and enrichment topologies.

About this task

You can find the management pack repositories for each of the operating systems supported by CCP in the CCP Release Notes. The following is an example of installing the HCP Ambari management pack on CentOS 7.



Note: Although HCP has been rebranded to CCP, the product management pack is still named HCP management pack and it is still located in the hortonworks repository in the HCP directory.

Procedure

1. Download the HCP management pack tar file from the HCP repo location:

```
wget -nv http://public-repo-1.hortonworks.com/HCP/centos7/2.x/updates/2.0.1.0/tars/metron/hcp-ambari-mpack-2.0.1.0-6.tar.gz
```

You can find the management pack repositories for each of the operating systems supported by CCP at [HCP Repositories](#).

2. If you are using Elasticsearch, download the Elasticsearch management pack tar file from the HCP repo location:

```
wget -nv http://public-repo-1.hortonworks.com/HCP/centos7/2.x/updates/2.0.1.0/tars/metron/hcp-ambari-mpack-2.0.1.0-6.tar.gz
```

3. Install the HCP management packs onto the Ambari server:

Install the `elasticsearch_mpack` only if you are using Elasticsearch.

```
ambari-server install-mpack --mpack=${MPACK_DOWNLOAD_DIRECTORY}/hcp-ambari-mpack-2.0.1.0-6.tar.gz --verbose ambari-server install-mpack --
```

```
mpack=/${MPACK_DOWNLOAD_DIRECTORY}/elasticsearch_mpack-2.0.1.0-6.tar.gz --
verbose
```

You should see a message saying that the install mpack completed successfully.

4. Install the MySQL-connector to enable the installation of the Hive component:

```
sudo yum install mysql-connector-java*
```

5. Make the jar file available for the Hive installation:

```
ls -al /usr/share/java/mysql-connector-java.jar
cd /var/lib/ambari-server/resources/
ln -s /usr/share/java/mysql-connector-java.jar mysql-connector-java.jar
```

6. Start or restart the Ambari Server, depending on whether you are installing CCP on a new or existing cluster:

```
ambari-server start
```

or

```
ambari-server restart
```

Install Solr

If you are using Apache Solr, install it using the Ambari HDP Search management pack.

Procedure

1. From Ambari, stop the following:

- Metron
- Kibana
- Elasticsearch

2. Install the Ambari HDP Search Management pack.

For instructions on downloading and using the Ambari HDP Search management pack, see https://docs.cloudera.com/HDPDocuments/HDP5/HDP5-4.0.0/bk_solr-search-installation/content/hdp-search40-install-mpack.html.



Important: Ensure the Java thread stack size parameter is set to greater than 320kb. The default setting for SOLR_JAVA_STACK_SIZE is not sufficient to start the Solr service.

Ambari automatically creates collections for the following:

- bro
- snort
- yaf
- metaalert
- error

3. If you want to create a collection for a schema not supplied by CCP, perform the following steps:

- a) Set Solr environmental variables in ZooKeeper.

```
# Path to the zookeeper node used by Solr
export ZOOKEEPER=node1:2181/solr
# Define SOLR_HOME
export SOLR_HOME=/opt/lucidworks-hdpsearch/solr/
# Set to true if Kerberos is enabled
export SECURITY_ENABLED=true
```

- b) Create a collection.

For example:

```
su $SOLR_USER -c "$SOLR_HOME/bin/solr create -c bro -d $METRON_HOME/
config/schema/bro/"
```

c) Pull all configurations from ZooKeeper to the Metron config directory:

```
$METRON_HOME/bin/zk_load_configs.sh -m PULL -z $ZOOKEEPER -o
$METRON_HOME/config/zookeeper -f
```

4. From Ambari, select **Metron** in the components panel.
5. Click the **Configs** tab, then click the **Rest** tab.
6. Populate the following fields with the appropriate information:

Source Type Field Name	The source type field name used in the real-time store. Defaults to source:type.
Threat Triage Score Field Name	The threat triage score field name used in the real-time store. Defaults to threat.triage.score.

7. Restart Metron.
8. Start Solr.
9. From Ambari, select **Metron** in the components panel.
10. Click the **Configs** tab, then click the **Indexing** tab.
11. Choose Solr in the **Index Writer - Random Access** pull down menu.

Index Updates

Indexing Update Table

Indexing Update Column Family

Index Writer - Random Access

Random Access Search Engine



Elasticsearch

Solr

Random Access

Enrichment Ackers for Random Access

12. Click **Save**.
13. From Ambari, stop and restart the Metron Alerts user interface.
14. From Ambari, stop and restart Metron REST.

What to do next

You can access Solr by choosing **Solr UI** from the **Quick Links** pull down menu in Ambari.



Install, Configure, and Deploy a HDP Cluster with CCP

You can use the Ambari Install wizard running in your browser to install, configure, and deploy your cluster.

About this task

To keep your changes to the indices writer, you must stop or restart the indexing topology only through Ambari. If you start or stop the indices writer through REST, the writer resets its settings to the Elasticsearch default settings.

Procedure

1. Open Ambari Web using a web browser.
 - a) Point your browser to `http://<your.ambari.server>:8080`, where `<your.ambari.server>` is the name of your ambari server host.
For example, a default Ambari server host is located at `http://c6401.ambari.apache.org:8080`.
 - b) Log in to the Ambari Server using the default user name/password: `admin/admin`.
You can change these credentials later.
For a new cluster, the Ambari install wizard displays a Welcome page from which you launch the Ambari Install wizard.
2. For an existing cluster, select **Choose Services** from the **Actions/Add Service Wizard** menu and skip to Step 7.
3. From the Ambari Welcome page, choose **Launch Install Wizard**.
4. In **Name your cluster**, type a name for the cluster you want to create, and then click **NEXT**.
Avoid white spaces or special characters in the name.
5. Select the HDP stack you want to run.
6. Remove the base URL operating system information for any OSs you will not be running, and then click **NEXT**.
7. Use the **Target Hosts** text box to enter your list of host names, one per line.
You can use ranges inside brackets to indicate larger sets of hosts. For example, for `host-01.domain` through `host-10.domain` use `host-[01-10].domain`
 **Note:** If you are deploying on EC2, use the internal Private DNS host names.
8. If you want to let Ambari automatically install the Ambari Agent on all your hosts using SSH, in the **Host Registration Information** section, select **Provide your SSH Private Key to automatically register hosts** and either use the **Choose File** button to find the private key file that matches the public key you installed earlier on all your hosts or cut and paste the key into the text box manually.
 **Note:** If you are using Internet Explorer 9, the Choose File button might not appear. Use the text box to cut and paste your private key manually. Fill in the user name for the SSH key you have selected. If you do not want to use root, you must provide the user name for an account that can execute sudo without entering a password.
9. Click **Register and Confirm** to continue then click **OK** to confirm the host name pattern expressions.

10. Click **NEXT** to move to the **Choose Services** dialog box that lists the services that Ambari can install onto the cluster.
11. Choose the services to install onto the cluster, and then click **Next**.
You can ignore any limited functionality warnings and click **PROCEED ANYWAY**.

Choose File System

Choose which file system you want to install on your cluster.

Service	Version	Description
<input checked="" type="checkbox"/> HDFS	3.1.1	Apache Hadoop Distributed File System

Choose Services

Choose which services you want to install on your cluster.

<input type="checkbox"/> Service	Version	Description
<input checked="" type="checkbox"/> YARN + MapReduce2	3.1.1	Apache Hadoop NextGen MapReduce (YARN)
<input checked="" type="checkbox"/> Tez	0.9.1	Tez is the next generation Hadoop Query Processing framework written on top of YARN.
<input checked="" type="checkbox"/> Hive	3.1.0	Data warehouse system for ad-hoc queries & analysis of large datasets and table & storage management service
<input checked="" type="checkbox"/> HBase	2.0.2	Non-relational distributed database and centralized service for configuration management & synchronization
<input type="checkbox"/> Pig	0.16.0	Scripting platform for analyzing large datasets
<input type="checkbox"/> Sqoop	1.4.7	Tool for transferring bulk data between Apache Hadoop and structured data stores such as relational databases
<input type="checkbox"/> Oozie	4.3.1	System for workflow coordination and execution of Apache Hadoop jobs. This also includes the installation of the optional Oozie Web Console which relies on and will install the ExtJS Library.
<input checked="" type="checkbox"/> ZooKeeper	3.4.6	Centralized service which provides highly reliable distributed coordination
<input checked="" type="checkbox"/> Storm	1.2.1	Apache Hadoop Stream processing framework
<input type="checkbox"/> Accumulo	1.7.0	Robust, scalable, high performance distributed key/value store.
<input type="checkbox"/> Infra Solr	0.1.0	Core shared service used by Ambari managed components.
<input checked="" type="checkbox"/> Ambari Metrics	0.1.0	A system for metrics collection that provides storage and retrieval capability for metrics collected from the cluster
<input type="checkbox"/> Atlas	1.1.0	Atlas Metadata and Governance platform
<input checked="" type="checkbox"/> Kafka	2.0.0	A high-throughput distributed messaging system
<input type="checkbox"/> Knox	1.0.0	Provides a single point of authentication and access for Apache Hadoop services in a cluster
<input type="checkbox"/> Log Search	0.5.0	Log aggregation, analysis, and visualization for Ambari managed services. This service is Technical Preview .
<input type="checkbox"/> Ranger	1.2.0	Comprehensive security for Hadoop
<input type="checkbox"/> Ranger KMS	1.2.0	Key Management Server
<input checked="" type="checkbox"/> SmartSense	1.5.1.2.7.3.0-139	SmartSense - Hortonworks SmartSense Tool (HST) helps quickly gather configuration, metrics, logs from common HDP services that aids to quickly troubleshoot support cases and receive cluster-specific recommendations.
<input checked="" type="checkbox"/> Spark2	2.3.2	Apache Spark 2.3 is a fast and general engine for large-scale data processing.
<input checked="" type="checkbox"/> Zeppelin Notebook	0.8.0	A web-based notebook that enables interactive data analytics. It enables you to make beautiful data-driven, interactive and collaborative documents with SQL, Scala and more.
<input type="checkbox"/> Druid	0.12.1	A fast column-oriented distributed data store.
<input checked="" type="checkbox"/> Elasticsearch	5.6.14	Indexing and Search
<input checked="" type="checkbox"/> Kibana	5.6.14	Kibana Dashboard
<input checked="" type="checkbox"/> Metron	0.7.1.1.9.2.0	A scalable advanced security analytics framework built on Hadoop
<input type="checkbox"/> Superset	0.23.0	Superset is a data exploration platform designed to be visual, intuitive and interactive. This service is Technical Preview .

NEXT →

CCP requires the following services:

- HDFS
- Yarn + MapReduce2
- Tez
- Hive
- HBase
- ZooKeeper
- Storm
- Ambari Metrics
- Kafka
- Spark2
- Zeppelin Notebook
- Elasticsearch or Solr

Elasticsearch can be installed either manually or by Ambari. We recommend installing Elasticsearch by Ambari.

- Kibana (Can be installed either manually or by Ambari. Hortonworks recommends installing Kibana by Ambari.)
- Metron
- HDFS

Ambari displays the **Assign Masters** window.

12. Use the **Assign Masters** window to assign the Master components to the appropriate hosts in your cluster.

- All Metron components must reside on the same node.
- The node containing the Metron components must have a Kafka broker.
- The node containing the Metron components must have a ZooKeeper server.

See Apache Javadoc, Version 1.8.
jaks.hawaii.edu/Java/links.html and their respective authors.

- If Ambari detects any errors in your master component assignments, it will indicate the error in red.
- To change the host assignment for a service, select a host name from the drop-down menu for that service.
 - To remove a ZooKeeper instance, click the green minus icon next to the host address you want to remove.

- c) When you are satisfied with the assignments, click **Next**.

Ambari displays the **Assign Slaves and Clients** window.

13. Use the **Assign Slaves and Clients** window to assign cluster nodes (DataNodes, NodeManagers, and RegionServers) to run with worker processes such as Elasticsearch.

The node containing the Metron host must have a Supervisor.

- a) Use all or none to select all of the hosts in the column or none of the hosts, respectively.

If a host has an asterisk next to it, that host is also running one or more master components. Hover your mouse over the asterisk to see which master components are on that host.

- b) Select a minimum of one Elasticsearch data node.

The data node cannot be on same host as the master.

- c) Select **Supervisor** for all data nodes.

Supervisor is used by Storm to run the topologies, so selecting Supervisor on all of the nodes enables you to distribute the workload among the nodes.

- d) Fine-tune your selections by using the check boxes next to specific hosts.

Assign Slaves and Clients

Assign slave and client components to hosts you want to run them on.
 Hosts that are assigned master components are shown with *.
 Client will install HDFS Client, YARN Client, MapReduce2 Client, Tez Client, Hive Client, HBase Client, Zookeeper Client, Spark2 Client and Metron Client.

Host	<input type="checkbox"/> all	<input type="checkbox"/> none	<input type="checkbox"/> all	<input type="checkbox"/> none	<input type="checkbox"/> all	<input type="checkbox"/> none	<input type="checkbox"/> all	<input type="checkbox"/> none	<input type="checkbox"/> all	<input type="checkbox"/> none	<input type="checkbox"/> all	<input type="checkbox"/> none	<input type="checkbox"/> all	<input type="checkbox"/> none
metron19-1.openstacklocal*	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
metron19-2.openstacklocal*	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
metron19-3.openstacklocal*	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
metron19-4.openstacklocal*	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
metron19-5.openstacklocal*	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

- e) When you are satisfied with your assignments, click **NEXT**.

14. Assign a username and password for any services that require credentials.

CREDENTIALS — DATABASES — DIRECTORIES — ACCOUNTS — ALL CONFIGURATIONS

Please provide credentials for these services

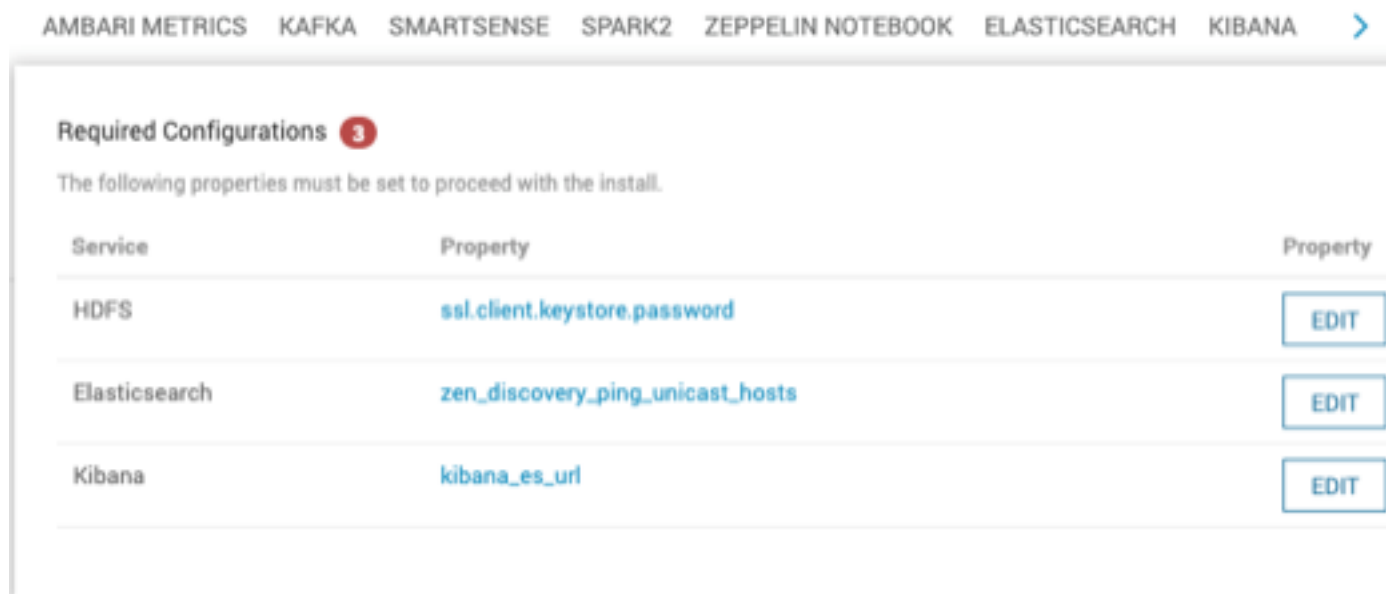
	Username*	Password*	Confirm Password*
Grafana Admin	admin	****	
Hive Database	hive	Type password	
Activity Explorer's Admin	N/A	Type password	

15. No changes are necessary for the **DATABASES**, **DIRECTORIES**, and **ACCOUNTS** windows, so click **NEXT** to proceed to the next window.

Ambari displays the **SETTINGS** window.

16.

Click  at the top right of the **SETTINGS** window to list the configuration properties you must set to proceed with the installation:



Service	Property	Property
HDFS	ssl.client.keystore.password	EDIT
Elasticsearch	zen_discovery_ping_unicast_hosts	EDIT
Kibana	kibana_es_url	EDIT

17. Click **EDIT** to display the appropriate window for each item in the list and provide the required information.

18. After you have completed configuring the required properties, click the **BACK** button to display the **ALL CONFIGURATIONS** window, then click **METRON** to display the Metron configurations.

You might need to click the arrow button to see METRON:



19. If you are using Elasticsearch, you must provide the Elasticsearch Host name under the **INDEX SETTINGS** tab in the **Elasticsearch Hosts** field.

20. You must either set up your JDBC information or enable LDAP prior to deploying METRON:

a) To set up your JDBC information, provide the required information under the **REST** tab:

INDEX SETTINGS PARSERS ENRICHMENT INDEXING PROFILER REST MANAGEMENT UI ALERTS UI SECURITY PCAP ADVANCED

Metron REST port

8082

Metron JDBC URL

Metron JDBC Driver

Metron JDBC username

Metron JDBC password

Type password

Metron JDBC platform

Metron JDBC client path

Metron temp grok path

{{(metron_apps_hdfs_dir)}/patterns/tmp

Active Spring profiles

Metron JVM flags

Metron Spring options

Metron REST port	Use 8082.
Metron JDBC URL	jdbc:mysql://mysql_host:3306/metronrest
Metron JDBC Driver	com.mysql.jdbc.Driver

You can choose between the following databases for the REST configuration.

- PostgreSQL
- MySQL
- H2
- Oracle

Metron JDBC Username

Metron REST user name

Metron JDBC Password

Metron REST password

Metron JDBC client path

<MYSQL_JAVA_CONNECTOR_PATH>/mysql-connector-java-5.1.41-bin.jar

- b) To switch to using LDAP to define access privileges see *Switch to LDAP Access Privileges*. Now, when you go to Swagger or the UIs, you should be able to view your assigned roles and permissions.

21. Browse through each service tab to determine if you want to provide any additional configuration information.

Most of the fields in the **Advanced** tab are auto populated and should not be modified.

22. OPTIONAL: Configure the PCAP topology by setting your PCAP properties in the **PCAP** tab.

The screenshot displays the Ambari configuration interface for the PCAP topology. The left sidebar contains the following fields:

- Workers for PCAP Topology: 1
- PCAP Topology childopts: (empty)
- PCAP Input Topic: pcap
- HDFS Sync Every: 1
- HDFS Replication Factor: -1
- PCAP Topology Offset: UNCOMMITTED

The right sidebar contains the following fields:

- Number of Packets to keep in one file: 1000
- Number of packets to keep in terms of duration: 300000
- Kafka PCAP Timestamp Scheme: FROM_KEY
- HDFS Directory to store PCAPs: /apps/metron/pcap/input
- Granularity of Timing in Timestamps: MICROSECOND
- PCAP Topology Spout Parallelism: 1

A tooltip for the 'Number of Packets to keep in one file' field shows the property 'kafka_pcap_numpackets' and the label 'Number of Packets'.

23. Check the assignments displayed by Ambari to ensure that everything is correct, and then click **Deploy**.

If you need to make changes, use the **BACK** button to return to the appropriate screen.



The progress of the install displays on the screen. Ambari installs, starts, and runs a simple test on each component. Overall status of the process displays in a progress bar at the top of the screen and host-by-host status displays in the main section. Do not refresh your browser during this process. Refreshing the browser might interrupt the progress indicators.

24. OPTIONAL: To see specific information on what tasks have been completed per host, click the link in the **Message** column for the appropriate host. In the **Tasks** pop-up, click the individual task to see the related log files. You can select filter conditions by using the **Show** drop-down list. To see a larger version of the log contents, click the **Open** icon or, to copy the contents to the clipboard, use the **Copy** icon.
25. When Successfully installed and started the services appears, click **NEXT**.

Switch to LDAP Access Privileges

CCP defaults to Java Database Connectivity (JDBC) to define access privileges. You can easily switch to using LDAP.

Procedure

1. In Ambari, select **Add Service** from the **Actions** menu.
2. Select **Knox** from the list of available components and click **Next**.
Now, when you go to Swagger or the UIs, you should be able to view your assigned roles and permissions.
3. Accept all of the defaults and click **Next** then **Deploy** to install Knox.
4. In Ambari, click **Metron** in the **Actions** menu and then click the **Config** tab.
5. Click the **Security** tab.

6. Populate the **User Role Name** field with the name of the LDAP group that provides user access to CCP.
7. Populate the **Admin Role Name** field with the name of the LDAP group that provides administration access to CCP.
8. Set **LDAP Enabled** to **On**.
9. Modify each of the fields to match your LDAP configuration.

Table 1:

Field	Definition	Example
LDAP URL	The url for the LDAP server. This must be in the following format: ldap://[host]:[port] or ldaps://[host]:[port]	ldap://localhost:33389 Active Directory: ldaps://23.96.22.127:636
Bind User	The fully distinguished name (DN) of an LDAP user account that has privileges to search for users.	uid=admin,ou=people,dc=hadoop,dc=apache,dc=org Active Directory: bind-admin@mycompany.local
Bind User Password	The password for the bind user account. Set this password to match the admin user's password from the ldif file.	adminPassword
User dn pattern	The pattern used to create a distinguished name (DN) from a username. This pattern is used to create a DN string for direct user authentication. The pattern argument {0} will be replaced with the username at runtime.	uid={0},ou=people,dc=hadoop,dc=apache,dc=org Active Directory: uid={0},cn=users,dc=mycompany,dc=local
User password attribute	The name of an attribute containing the SHA-encoded user password. This attribute is used to perform a remote compare operation to authenticate the user. To use bind authentication, leave this field blank.	userPassword Active Directory: leave blank
User Search Base	The location from which the search starts for user entries.	ou=people,dc=hadoop,dc=apache,dc=org Active Directory: cn=users,dc=mycompany,dc=local
User Search Filter	The search filter used to locate a user. The pattern argument {0} is replaced with the username at runtime.	Active Directory: (&(objectClass=user)(sAMAccountName={0}))
Group Search Base	The location from which the search starts for group entries.	ou=groups,dc=hadoop,dc=apache,dc=org Active Directory: cn=users,dc=mycompany, dc=local
Group Search Filter	The search filter used to locate a group. The pattern argument {0} is replaced with the username at runtime.	member={0} Active Directory: (&(objectClass=group)(member={0}))
LDAP group role attribute	The attribute of a group that defines the group name.	cn
LDAP Truststore	The path of the truststore containing SSL certs for LDAP.	/usr/metron/0.7.1/keystore
LDAP Truststore Password	The password for the truststore containing SSL certs for LDAP.	



Note: If you want to use bind authentication rather than performing a remote password comparison (using an SHA-encoded password) to authenticate users, leave the **User password attribute** field blank.

10. Click the **Summary** tab to display all of the Metron components.
11. Restart **Metron REST**.

Now, when you go to Swagger or the UIs, you should be able to view your assigned roles and permissions.

Import Apache Zeppelin Notebook Using Ambari

If you would like to install Apache Zeppelin, complete the following steps after you have successfully installed CCP. You can use the Apache Zeppelin dashboard to view and analyze telemetry data provided by CCP.

Procedure

1. Login to Ambari at `http://$AMBARI_HOST:8080`.
2. In Ambari, click **Metron>Service Actions>Zeppelin Notebook Import**.
Ambari imports the Zeppelin Notebook.
3. Login to Zeppelin at `http://$ZEPPELIN_HOST:9995`.
4. Search for the notebook named **Metron - YAF Telemetry**.

Streaming Data into CCP

To prepare for Cloudera Cybersecurity Platform (CCP) to ingest data source data into CCP, you must stream each raw event stream from the telemetry data source into its own individual Kafka topic. This applies to the telemetry data sources for which CCP includes parsers (for example, Bro, Snort, and YAF). Even though CCP includes parsers for these data sources, CCP does not install these data sources or ingest the raw data. This is something that you must do.

Depending on the type of data you are streaming into CCP, you can use one of the following methods:

NiFi

This type of streaming method works for most types of data sources.



Note:

Ensure that the NiFi web application is using port 8089.

Performant network ingestion probes

This type of streaming method is ideal for streaming high volume packet data.

Real-time and batch threat intelligence feed loaders

This type of streaming method is used for real-time and batch threat intelligence feed loadNiFiers.

Verify That CCP Deployed Successfully for Ambari Install

After you install Cloudera Cybersecurity Platform (CCP), you need to verify that your services are displayed in Ambari and that you can access the Metron Dashboard.

Procedure

1. Verify that the topologies bundled with CCP are deployed.
From Ambari, navigate to **Storm > Quick Links > Storm UI**.
You should see the following topologies listed:
 - Snort
 - pcap

- YAF (Yet Another Flowmeter)
 - Bro Network Security Monitor
 - Indexing topology
- 2.** Check that the enrichment topology has emitted some data.
- This could take a few minutes to show up in the Storm UI. The Storm enrichment topology UI should look something like the following:
- Storm UI with Enrichment Details

Storm UI

Cluster Summary

Version	Supervisors	Used slots	Free slots	Total slots	Executors	Tasks
1.2.1.3.1.0.0-78	1	6	0	6	35	35

Nimbus Summary

Search:

Host	Port	Status	Version	UpTime
node1	6627	Leader	1.2.1.3.1.0.0-78	6h 17m 12s

Showing 1 to 1 of 1 entries

Topology Summary

Search:

Name	Owner	Status	Uptime	Num workers	Num executors	Num tasks	Replication count	Assigned Mem (MB)	Scheduler Info
batch_indexing	storm	ACTIVE	5h 58m 0s	1	5	5	1	832	
bro_snort_yaf	storm	ACTIVE	6h 0m 33s	1	7	7	1	832	
enrichment	storm	ACTIVE	6h 9m 45s	1	8	8	1	832	
pcap	storm	ACTIVE	6h 7m 56s	1	3	3	1	832	
profiler	storm	ACTIVE	6h 4m 59s	1	7	7	1	832	
random_access_indexing	storm	ACTIVE	5h 56m 28s	1	5	5	1	832	

Showing 1 to 6 of 6 entries

Supervisor Summary

Search:

Host	Id	Uptime	Slots	Used slots	Avail slots	Used Mem (MB)	Version
node1 [log]	0afa1bb9-1938-4258-b134-214d882132df	6h 15m 3s	6	6	0	4992	1.2.1.3.1.0.0-78

Showing 1 to 1 of 1 entries

Nimbus Configuration

Show: 20 entries Search:

Key	Value
backpressure.disruptor.high.watermark	6.9
backpressure.disruptor.low.watermark	6.4
backpressure.znode.timeout.secs	30
backpressure.znode.update.freq.secs	15
client.blobstore.class	"org.apache.storm.blobstore.NimbusBlobStore"
client.jartransformer.class	"org.apache.storm.hack.StoreShadeTransformer"
dev.zookeeper.path	"/tmp/dev-storm-zookeeper"
drpc.authorizer.acl.filename	"drpc-auth-acl.yaml"
drpc.authorizer.acl.strict	false
drpc.childopts	"-Xmx768m"
drpc.http.creds.plugin	"org.apache.storm.security.auth.DefaultHttpCredentialsPlugin"
drpc.http.port	3774
drpc.https.keystore.password	""
drpc.https.keystore.type	"JKS"
drpc.https.port	-1
drpc.invocations.port	3773
drpc.invocations.threads	64
drpc.max_buffer_size	1048576
drpc.port	3772
drpc.queue.size	128

Showing 1 to 20 of 220 entries

Previous 1 2 3 4 5 ... 11 Next

- Ensure that the Metron dashboard is available and receiving data by displaying the dashboard at \$METRON_UI_HOST:5000.

Check to ensure that the indexing is done correctly and the data is visualized.

4. Check to ensure that some data is written into HDFS at /apps/metron for at least one of the data sources.

What to do next

Customize CCP to meet your own needs.

Open the Metron Dashboard

After you install and configure CCP, you can load and launch the Metron dashboard. The Metron dashboard enables you to identify, investigate, and analyze cybersecurity data.

Procedure

1. Ensure that you have selected **Metron** in the left navigation panel in Ambari.
2. From the **Service Action** menu, select **Kibana Dashboard Install**.
3. After the dashboard installs, click **Kibana** in the left navigation panel.
4. From the **Quick Links** pull-down menu, select **Metron UI**.

The Metron dashboard should display in a separate browser tab.

What to do next

If you have already installed the Metron dashboard, reloading the dashboard will not overwrite your customizations to the dashboard. If you want to overwrite your customizations to the dashboard, you must delete the .kibana index from Elasticsearch and reload the Metron dashboard again from Ambari.

Opening the Management User Interface

You can use the CCP Management user interface to add and configure telemetry parsers to Cloudera Cybersecurity Platform (CCP). You can launch the UI either from Ambari or from a browser.

Procedure

1. From the Ambari Dashboard navigation panel, click **Metron**.
2. Verify that the **Summary** tab is selected.
3. From the **Quick Links** menu, select **Management UI**.

The Metron Management UI tool displays in a separate browser tab.

Alternatively, you can launch the module from \$METRON_MANAGEMENT_UI_HOST:4200 in a browser.

Opening the Alerts User Interface

You can use the Alerts user interface to display, filter, and sort events and their associated fields. You can also use the UI to escalate, add comments to, and group events.

Procedure

1. From the Ambari Dashboard navigation panel, click **Metron**.
2. Verify that the **Summary** tab is selected.
3. From the **Quick Links** menu, select **Alerts UI**.

The Alerts UI tool displays in a separate browser tab.

Troubleshooting Your Installation

If you encounter errors or issues during your Cloudera Cybersecurity Platform (CCP) installation, you can uninstall your installation.

Uninstalling your Metron installation

You can follow the uninstallation steps in *Upgrade Metron* to uninstall your Metron installation.

If Ambari does not allow you to uninstall after you have stopped all of the Metron services and the Storm topologies, perform the following from any node that has access to Ambari and has curl installed:

```
curl -u $USERNAME:$PASSWORD -H "X-
Requested-By: ambari" -X DELETE
http://$AMBARI_HOST:
$AMBARI_PORT/api/v1/clusters/
$METRON_CLUSTER_NAME/services/METRON
```

If you don't know the name of your Ambari host, choose **Manage Ambari** from the **admin** menu on the Ambari **Dashboard** tab. The name of your cluster is displayed in the **Clusters** section. If you cannot see the entire cluster name, click the icon to rename your cluster and the entire cluster name will display. Click **x** to dismiss the rename field without renaming your cluster.

Optimization Guidelines

In any Storm-based platform, there are many parameters that control the system's performance. The values of these parameters vary greatly with differences in cluster size and data velocity. You will need to ensure that you have a properly tuned index is key to overall system performance. See the Storm user guide for detailed discussion.

- num.workers
- num.ackers
- max.spout.pending
- topology.worker.childopts – increase heap size (-XmxNNNNm -XmsNNNNm)
- topology.workers