HCP Installation Guide 1

Hortonworks Cybersecurity Platform

Date of Publish: 2018-08-23

http://docs.hortonworks.com

Contents

Introduction to Hortonworks Cybersecurity Platform

Hortonworks Cybersecurity Platform (HCP) is a cybersecurity application framework that provides the ability to parse diverse security data feeds, enrich, triage, and store the data at scale, and detect cybersecurity anomalies.

HCP integrates a variety of open source big data technologies in order to offer a centralized tool for security monitoring and analysis. HCP provides capabilities for log aggregation, full packet capture indexing, storage, advanced behavioral analytics and data enrichment, while applying the most current threat intelligence information to security telemetry within a single platform.

Preparing to Install

Prior to installing HCP for the first time, you must ensure that you meet the minimum system requirements.

Operating System Requirements

Prior to installing HCP, ensure that your meet the operating system requirements for HCP.

HCP currently supports CentOS v6.x, CentOS v7.x, and Ubuntu 14.0.

Important:

If you are using CentOS 6.x or CentOS 7.x, you must install the EPEL repo. Also make sure you are using python-requests version 2.6.1 or later.

Related Information

How to Install EPEL Repo on a CentOS and RHEL

Browser Requirements

The Ambari Install Wizard runs as a browser-based Web application. You must have a machine capable of running a graphical browser to use this tool.

The minimum required browser versions are:

- Windows (7, 8)
  - Firefox 18
  - Google Chrome 26
- Mac OS x (10.6 or later)
  - Firefox 18
  - Safari 5
  - Google Chrome 26
- Linux (CentOS)
  - Firefox 18
  - Google Chrome 26

On any platform, we recommend updating your browser to the latest, stable version.

Infrastructure Requirements

Prior to installing HCP, ensure that your physical nodes adhere to the specifications required by HCP.

HCP requires the following indicative specifications for your physical nodes:

Table 1: Physical Nodes

Role	Indicative Specifications
PCAP Collector Card	Ethernet—Adapter—X520—DA2 or DPDK compatible card 20 GB/Sec
PCAP Collector Server	• CPUs: 2 x 8 Core Processors • Memory: 128 GB RAM • Disk Storage: 10 x 2 TB SATA Drives • Network: 2 x 10 GB NIC
NiFi Server	• CPUs: 2 x 8 Core Processors • Memory: 128 GB RAM • Disk Storage: 10 x 2 TB SATA Drives • Network: 2 x 10 GB NIC
Apache Kafka / Storm Server	• CPUs: 2 x 8 Core Processors • Memory: 128 GB RAM • Disk Storage: 10 x 2 TB SATA Drives • Network: 2 through 10 GB NIC
Metron Master Nodes	• CPUs: 2 x 8 Core Processors • Memory: 128 GB RAM • Disk Storage: 10 x 2 TB SATA Drives • Network: 2 x 10 GB NIC
HCP Worker Nodes— Balanced	• CPUs: 2 x 8 Core Processors • Memory: 128 GB RAM • Disk Storage: 10—2 TB SATA Drives • Network: 2—10 GB NIC

Software Requirements

Prior to installing HCP, ensure that you meet the software specifications required by HCP.

The host that you choose to use to deploy Apache Metron must have the following software tools installed:

- Hadoop (HDP 2.5 or HDP 2.6 recommended)

  The following are the required components for HDP 2.5.x and HDP 2.6.x:

  - Apache Hadoop
  - Apache Storm
  - Apache Kafka
  - Apache HBase
  - Apache ZooKeeper

    Note:

    Supervisor, Kafka Broker, and the HBase client must be installed on the Metron Install Host.
- To use the PCAP query user interface, you must perform the following:
  - Install Wireshark.

    For example, for CentOS, use the following command:

    ```
    yum -y install wireshark
    ```
  - Add a Metron user to the Wireshark group.

For example, for CentOS, use the following command:

```
-usermod -a -G wireshark metron
```

- MySQL
- Node.js repository installed on the Management UI host

  You can add the Node.js repository with the instructions from the Node.js Package Manager documentation.
- Installable during the Ambari installation of HCP

  The following software is required for HCP, but this software can be installed manually or during the HCP Ambari installation. Hortonworks recommends that you wait to install this software until the Ambari installation of HCP.

  - Elasticsearch 2.3.3
  - Kibana 4.5.1

Related Information

HDP supported component versions

Node.js Package Manager documentation

Memory Requirements

Prior to installing HCP, ensure that your meet the memory requirements for HCP.

For memory requirements, see the Memory Requirements provided in the *Apache Ambari Installation* guide.

Maximum Open File Descriptors

Prior to installing HCP, ensure that you meet the maximum number of open file descriptors required by HCP.

The recommended maximum number of open file descriptors is 50,000, or more. To check the current value set for the maximum number of open file descriptors, execute the following shell commands on each host:

```
ulimit -Sn

ulimit -Hn
```

If the output is not greater than 50,000, run the following command to set it to a suitable default:

```
ulimit -n 50000
```

Installing HCP on an Ambari-Managed Cluster Using Ambari

Installing HCP using Ambari utilizes both the graphic user interface of Ambari and the Metron management pack. Both of these tools promote a faster installation that pre-installs much of the configuration you will need.

Prerequisites for an Existing Cluster

You can install HCP on an Ambari-managed cluster running HDP 2.5.x or 2.6.x and Ambari 2.4.2 (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 HCP must meet operating system, HDP, and cluster requirements.

All Hadoop-related nodes must meet the following specifications:

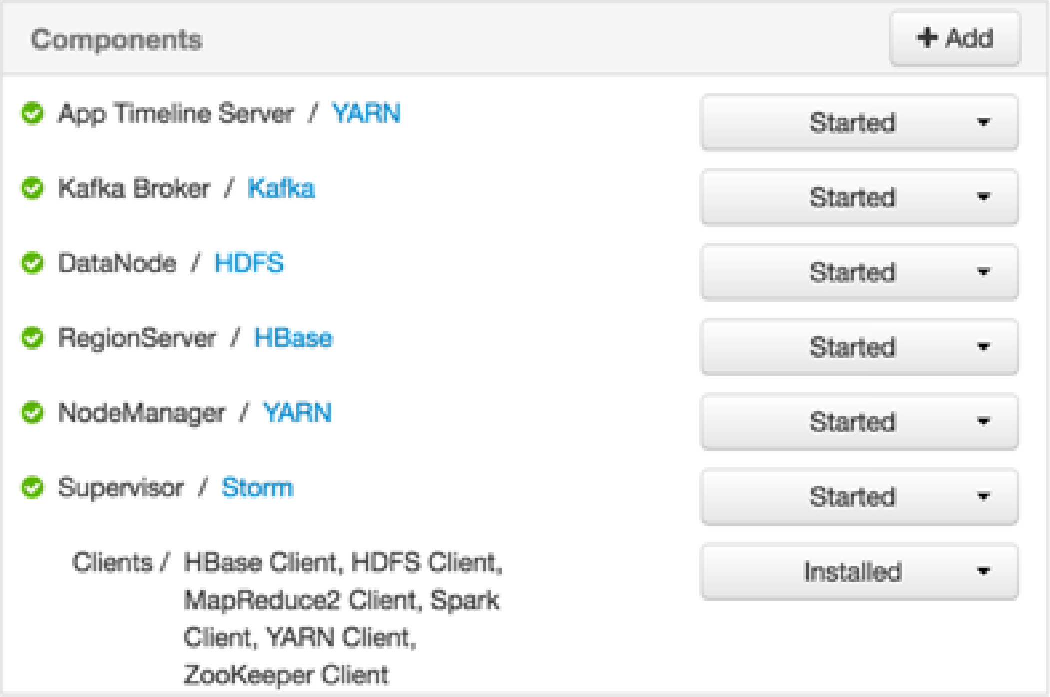
- All cluster nodes must be running CentOS 6.x, CentOS 7.x, or Ubuntu 14.04
- The cluster must be running HDP 2.5.x or HDP 2.6.x managed by Ambari 2.4.2 (or later)
- The cluster must have a minimum of the following nodes:
  - Two Hadoop master nodes
  - Four Hadoop slaves nodes
  - One node for Ambari
- 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
  - ZooKeeper
  - Kafka
  - Storm
  - YARN

  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 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.

  Ambari Component



Components

App Timeline Server / YARN — Started

Kafka Broker / Kafka — Started

DataNode / HDFS — Started

RegionServer / HBase — Started

NodeManager / YARN — Started

Supervisor / Storm — Started

Clients / HBase Client, HDFS Client, MapReduce2 Client, Spark Client, YARN Client, ZooKeeper Client — Installed

Related Information

Preparing to Install

Specifications for Metron Nodes

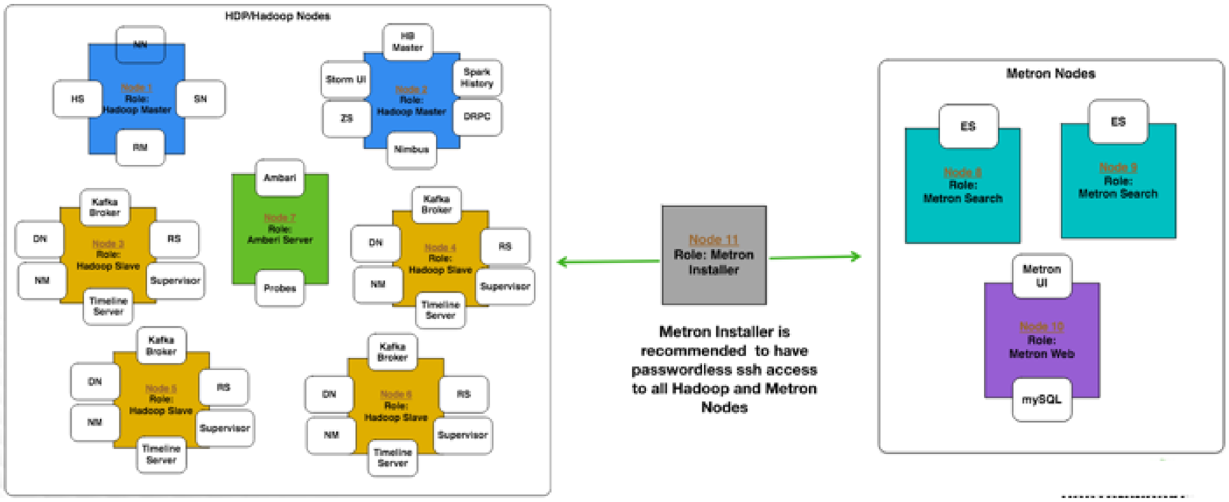
All Metron nodes must meet specifications for the number of nodes dedicated for Metron-specific components and the ability to access the nodes.

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 HCP, 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 HCP on an Ambari Cluster

Prior to installing the HCP Ambari management pack, you must meet HCP'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:

Procedure

- 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.
- Download and install Ambari.
- Set up the Ambari server.

Install HCP Ambari Management Pack

An HCP 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 HCP in the HCP Release Notes. The following is an example of installing the HCP Ambari management pack on CentosOS 7.

Procedure

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

   ```
   wget -nv http://public-repo-1.hortonworks.com/HCP/centos7/1.x/
   updates/1.6.1.0/tars/metron/hcp-ambari-mpack-1.6.1.0-23.tar.gz
   ```

   You can find the management pack repositories for each of the operating systems supported by HCP at HCP Repositories.

   > **Note:** When installing Elasticsearch with the HCP management pack on Ubuntu, you must manually install the Elasticsearch repositories. You also do not need to download and install the elasticsearch_mpack.

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/centos7/1.x/
   updates/1.6.1.0/tars/metron/elasticsearch_mpack-1.6.1.0-23.tar.gz
   ```

3. Install the HCP management packs:

   Install the elasticsearch_mpack only if you are using Elasticsearch.

   ```
   ambari-server install-mpack --mpack=/${MPACK_DOWNLOAD_DIRECTORY}/hcp-
   ambari-mpack-1.6.1.0-23.tar.gz --verbose ambari-server install-mpack --
   mpack=/${MPACK_DOWNLOAD_DIRECTORY}/elasticsearch_mpack-1.6.1.0-23.tar.gz
    --verbose
   ```

   You should see a message saying that the management pack completed successfully.

Related Information

Apache Solr Search Installation

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 Apache Solr Search Installation.

   The Meta Alerts UI feature with Solr is technical preview in this release. We do not yet recommend this for production use, but please let us know about any bugs you might find. We appreciate your feedback.

   > **Important:** Ensure the Java threat 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 HCP, 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. Add "source.type.field" : "source.type" and threat.triage.score.field" : "threat.triage.score" to the global.json file located at$METRON_HOME/config/zookeeper/global.json:

```
$METRON_HOME/bin/zk_load_configs.sh -m PUSH -z $ZOOKEEPER -i $METRON_HOME/
config/zookeeper
```

The global.json file should look similar to:

```
{
  "es.clustername" : "metron",
  "es.ip" : "blah:9300",
  "es.date.format" : "yyyy.MM.dd.HH",
  "parser.error.topic" : "indexing",
  "update.hbase.table" : "metron_update",
  "update.hbase.cf" : "t",
  "es.client.settings" : {
    "client.transport.ping_timeout" : "500s"
  },
  "profiler.client.period.duration" : "15",
  "profiler.client.period.duration.units" : "MINUTES",
  "source.type.field" : "source.type",
  "threat.triage.score.field" : "threat:triage:score",
  "user.settings.hbase.table" : "user_settings",
  "user.settings.hbase.cf" : "cf",
  "geo.hdfs.file" : "/apps/metron/geo/default/GeoLite2-City.mmdb.gz"
}
```

5. Push the configuration to ZooKeeper:

```
$METRON_HOME/bin/zk_load_configs.sh -m PUSH -z $ZOOKEEPER -i $METRON_HOME/
config/zookeeper
```

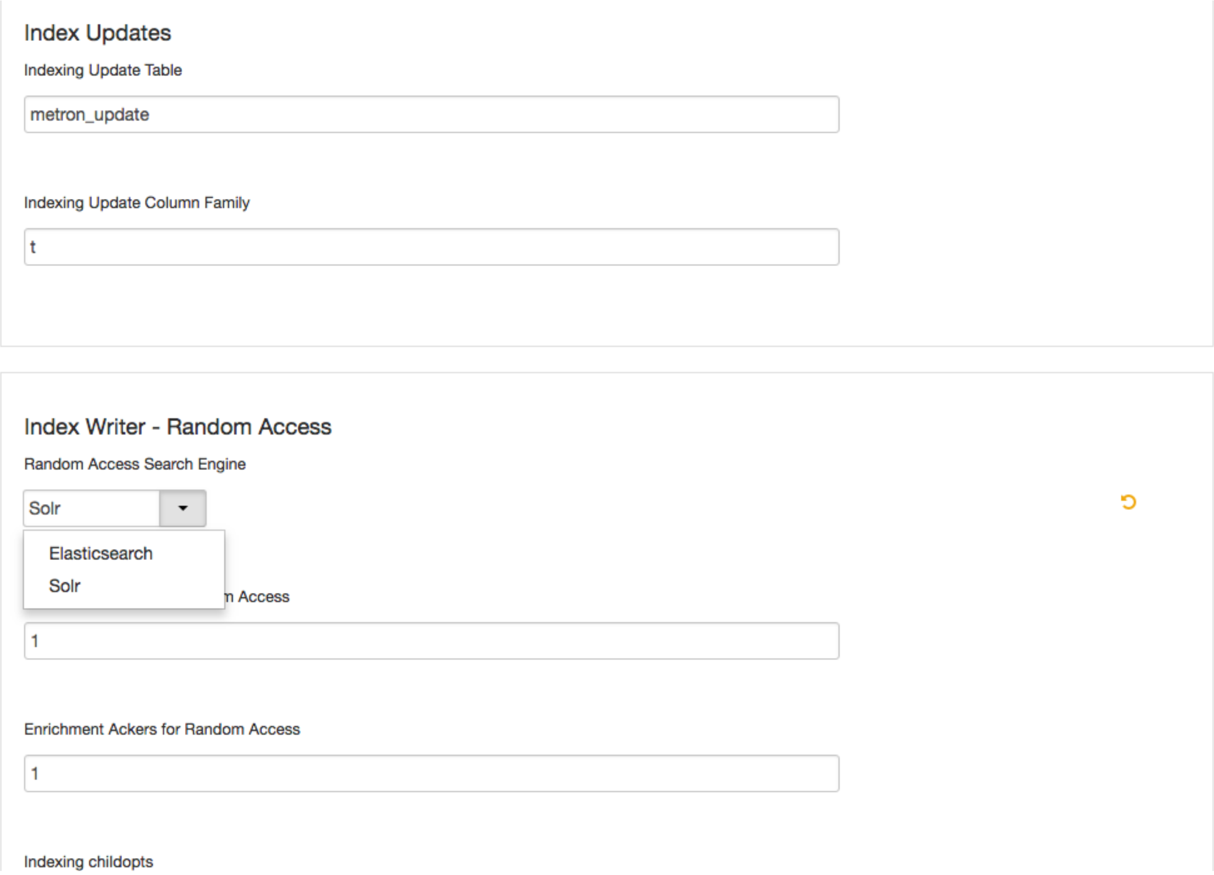
6. Restart Metron.

7. Start Solr.

8. From Ambari, select **Metron** in the components panel.

9. Click the **Configs** tab, then click the **Indexing** tab.

10. Choose Solr in the **Index Writer - Random Access** pull down menu.



Index Updates

Indexing Update Table

metron_update

Indexing Update Column Family

t

Index Writer - Random Access

Random Access Search Engine

Solr

n Access

1

Enrichment Ackers for Random Access

1

Indexing childopts

11. Click **Save**.

12. From Ambari, stop and restart the Metron Alerts user interface.

13. 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.

Start the Ambari Server

After you install the HCP Ambari management pack, you need to start or restart the Ambari server, depending on whether your are installing HCP on a new or existing cluster.

Procedure

1. To start the Ambari server, enter the following:

ambari-server start

2. To restart the Ambari server, enter the following:

ambari-server restart

Install, Configure, and Deploy a HDP Cluster with HCP

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, choose **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 choose **Next**.

   o white spaces or special characters in the name.
5. Select the HDP stack you want to run.
6. Enter the set up information for which the install wizard prompts you.

   You need to supply the FQDN of each of your hosts. The wizard also needs to access the private key file you created in Set Up Password-less SSH. Using the host names and key file information, the wizard can locate, access, and interact securely with all hosts in the cluster.
   a) Use the **Target Hosts** text box to enter your list of host names, one per line.

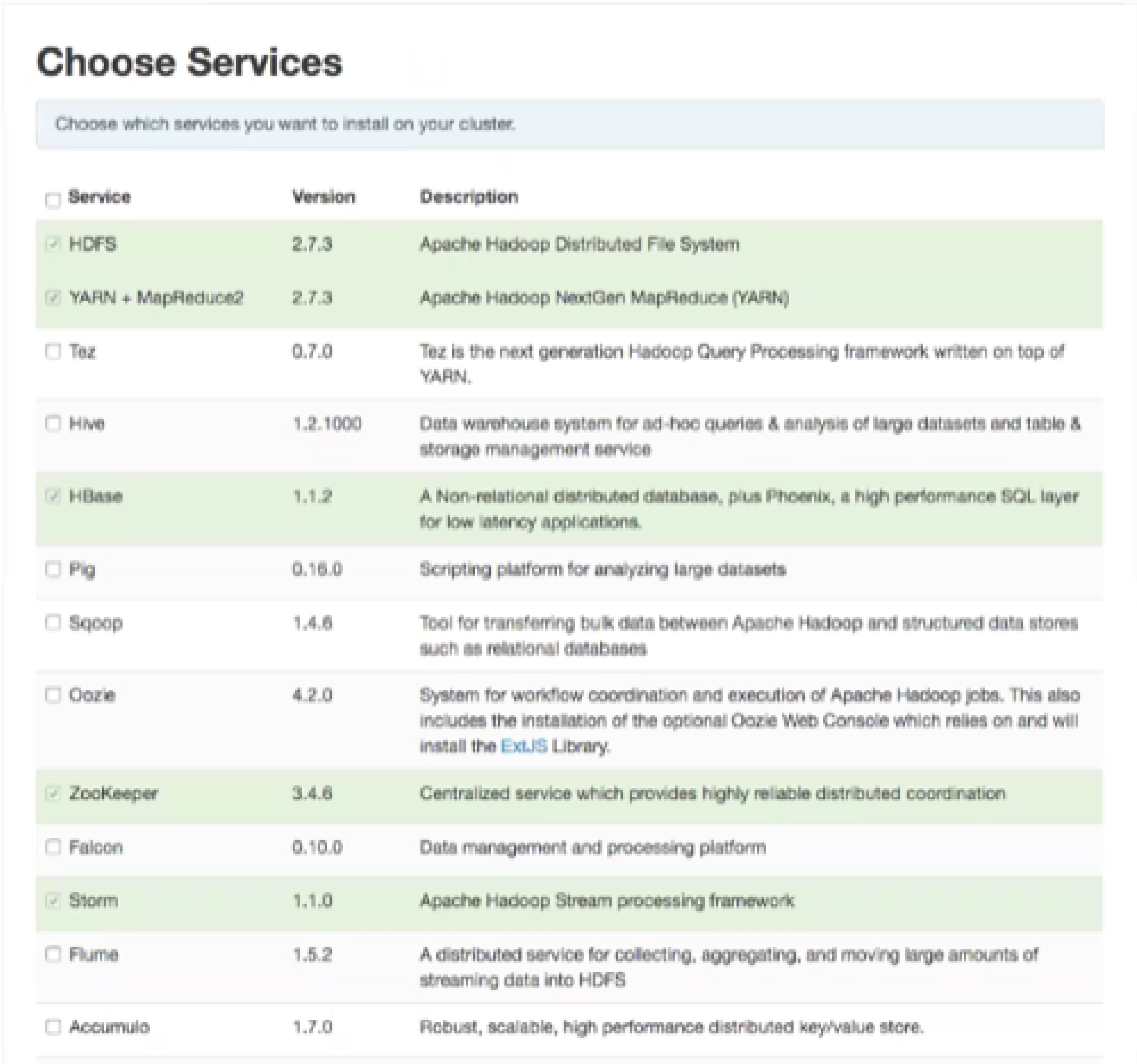
      You can use ranges inside brackets to indicate larger sets of hosts. For example, for host01.domain through host10.domain use host[01-10].domain

      > **Note:** If you are deploying on EC2, use the internal Private DNS host names.
   b) If you want to let Ambari automatically install the Ambari Agent on all your hosts using SSH, select **Provide your SSH Private Key** and either use the **Choose File** button in the **Host Registration Information** section 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 IE 9, the Choose File button may 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.
   c) Choose **Register** and **Confirm** to continue.

      Ambari displays the **Choose Services** dialog box that lists the services that Ambari can install into the cluster.
7. Choose the services to install into the cluster, and then click **Next**.

   Ambari Choose Services Window



Choose Services

Choose which services you want to install on your cluster.

□ Service	Version	Description
☑ HDFS	2.7.3	Apache Hadoop Distributed File System
☑ YARN + MapReduce2	2.7.3	Apache Hadoop NextGen MapReduce (YARN)
□ Tez	0.7.0	Tez is the next generation Hadoop Query Processing framework written on top of YARN.
□ Hive	1.2.1000	Data warehouse system for ad-hoc queries & analysis of large datasets and table & storage management service
☑ HBase	1.1.2	A Non-relational distributed database, plus Phoenix, a high performance SQL layer for low latency applications.
□ Pig	0.16.0	Scripting platform for analyzing large datasets
□ Sqoop	1.4.6	Tool for transferring bulk data between Apache Hadoop and structured data stores such as relational databases
□ Oozie	4.2.0	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.
☑ ZooKeeper	3.4.6	Centralized service which provides highly reliable distributed coordination
□ Falcon	0.10.0	Data management and processing platform
☑ Storm	1.1.0	Apache Hadoop Stream processing framework
□ Flume	1.5.2	A distributed service for collecting, aggregating, and moving large amounts of streaming data into HDFS
□ Accumulo	1.7.0	Robust, scalable, high performance distributed key/value store.

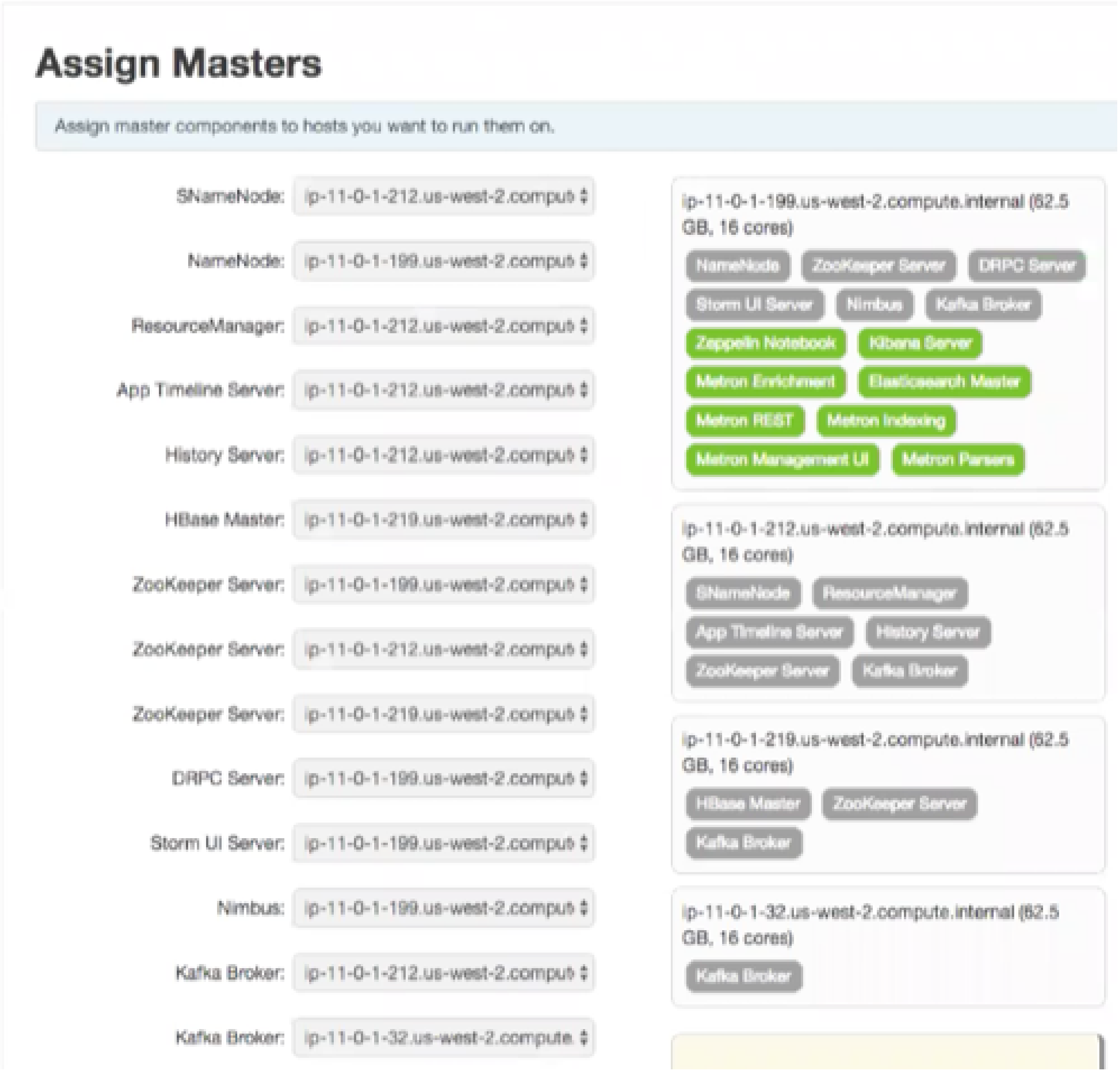
HCP requires the following services:

- HDFS
- HBase
- ZooKeeper
- Storm
- Kafka
- Ambari Metric Service
- Metron
- Elasticsearch (Can be installed either manually or by Ambari. Hortonworks recommends installing Elasticsearch by Ambari.)
- Kibana (Can be installed either manually or by Ambari. Hortonworks recommends installing Kibana by Ambari.)
- Zeppelin Notebook
- Spark
- Hive
- Tez
- Yarn

Ambari displays the **Assign Masters** window.

8. Verify that the Ambari install wizard has assigned the master components for selected services to appropriate hosts in your cluster.

   Ambari Assign Masters Window



Assign Masters

Assign master components to hosts you want to run them on.

SNameNode: ip-11-0-1-212.us-west-2.comput

NameNode: ip-11-0-1-199.us-west-2.comput

ResourceManager: ip-11-0-1-212.us-west-2.comput

App Timeline Server: ip-11-0-1-212.us-west-2.comput

History Server: ip-11-0-1-212.us-west-2.comput

HBase Master: ip-11-0-1-219.us-west-2.comput

ZooKeeper Server: ip-11-0-1-199.us-west-2.comput

ZooKeeper Server: ip-11-0-1-212.us-west-2.comput

ZooKeeper Server: ip-11-0-1-219.us-west-2.comput

DRPC Server: ip-11-0-1-199.us-west-2.comput

Storm UI Server: ip-11-0-1-199.us-west-2.comput

Nimbus: ip-11-0-1-199.us-west-2.comput

Kafka Broker: ip-11-0-1-212.us-west-2.comput

Kafka Broker: ip-11-0-1-32.us-west-2.compute

ip-11-0-1-199.us-west-2.compute.internal (62.5 GB, 16 cores)

NameNode, ZooKeeper Server, DRPC Server, Storm UI Server, Nimbus, Kafka Broker, Zeppelin Notebook, Kibana Server, Metron Enrichment, Elasticsearch Master, Metron REST, Metron Indexing, Metron Management UI, Metron Parsers

ip-11-0-1-212.us-west-2.compute.internal (62.5 GB, 16 cores)

SNameNode, ResourceManager, App Timeline Server, History Server, ZooKeeper Server, Kafka Broker

ip-11-0-1-219.us-west-2.compute.internal (62.5 GB, 16 cores)

HBase Master, ZooKeeper Server, Kafka Broker

ip-11-0-1-32.us-west-2.compute.internal (62.5 GB, 16 cores)

Kafka Broker

If Ambari detects any errors in your master component assignments, it will indicate the error in red.

a) To change the host assignment for a service, select a host name from the drop-down menu for that service.
b) 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**.

9. Verify that the Ambari install wizard has assigned the slave components (DataNodes, NodeManagers, and RegionServers) to appropriate hosts in your cluster.

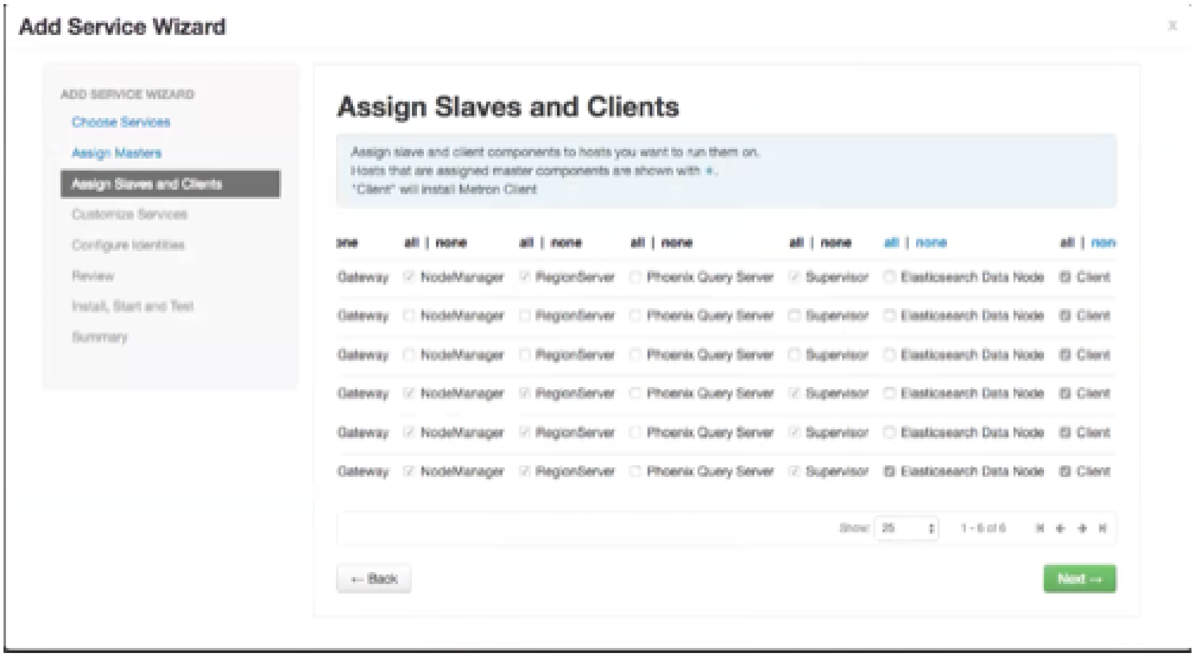
a) Use all or none to select all of the hosts in the column or none of the hosts, repsectively.

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

d) Check the **Client** checkbox for any components that have the **Supervisor** checkbox checked.

Ambari Assign Slaves and Clients Window



Assign Slaves and Clients

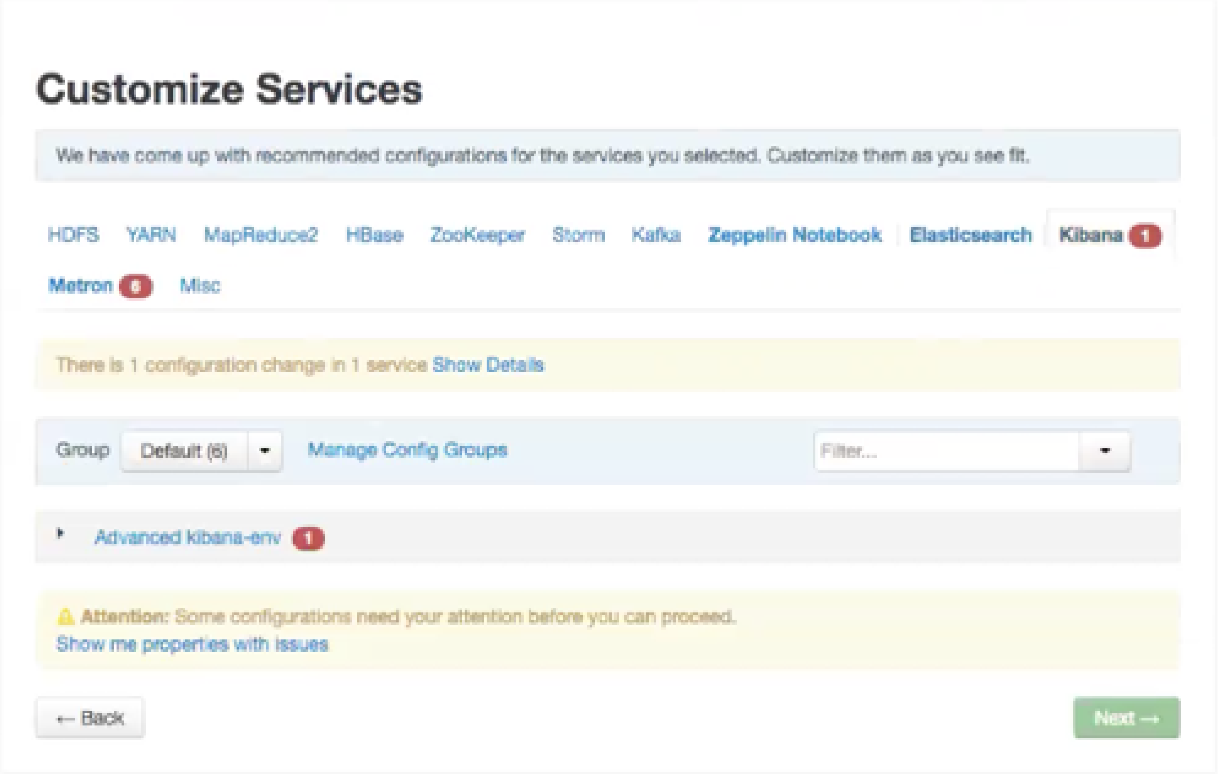
Assign slave and client components to hosts you want to run them on.
Hosts that are assigned master components are shown with ✶.
"Client" will install Metron Client

one	all | none	all | none	all | none	all | none	all | none	all | non
Gateway	☑ NodeManager	☑ RegionServer	☐ Phoenix Query Server	☑ Supervisor	☐ Elasticsearch Data Node	☒ Client
Gateway	☐ NodeManager	☐ RegionServer	☐ Phoenix Query Server	☐ Supervisor	☐ Elasticsearch Data Node	☒ Client
Gateway	☐ NodeManager	☐ RegionServer	☐ Phoenix Query Server	☐ Supervisor	☐ Elasticsearch Data Node	☒ Client
Gateway	☑ NodeManager	☑ RegionServer	☐ Phoenix Query Server	☑ Supervisor	☐ Elasticsearch Data Node	☒ Client
Gateway	☑ NodeManager	☑ RegionServer	☐ Phoenix Query Server	☑ Supervisor	☐ Elasticsearch Data Node	☒ Client
Gateway	☑ NodeManager	☑ RegionServer	☐ Phoenix Query Server	☑ Supervisor	☒ Elasticsearch Data Node	☒ Client

Show: 25 1 - 6 of 6

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

10. Review each service tab in the **Customize Services** window and modify your HDP cluster setup if appropriate.



Customize Services

We have come up with recommended configurations for the services you selected. Customize them as you see fit.

There is 1 configuration change in 1 service Show Details

⚠ Attention: Some configurations need your attention before you can proceed.
Show me properties with issues

a) Browse through each service tab. By hovering your cursor over each of the properties, you can see a brief description of what the property does.

The number of service tabs shown depends on the services you decided to install in your cluster. Any tab that requires input displays a red badge with the number of properties that need attention. Select each service tab that displays a red badge number and enter the appropriate information.

The following is a list of service tabs for which you'll need to provide information:

Kibana

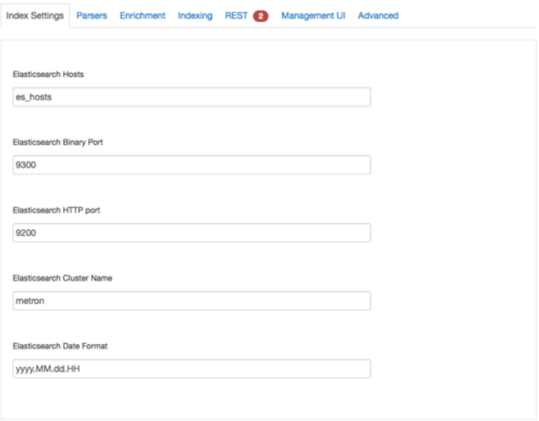
kibana_es_url

Set to the fully-qualified url for the Elasticsearch master: http://es-master-host:9200.

Metron

The Metron tab contains a few tabs that contain information that is critical to HCP set up.

- Index Settings



Elasticsearch Hosts
es_hosts

Elasticsearch Binary Port
9300

Elasticsearch HTTP port
9200

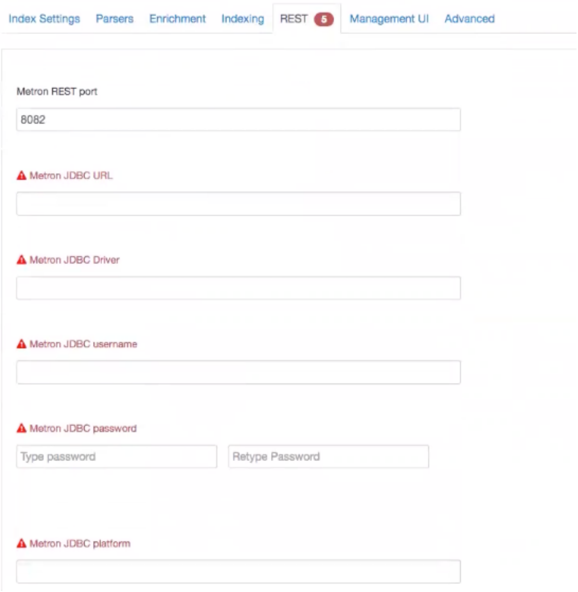
Elasticsearch Cluster Name
metron

Elasticsearch Date Format
yyyy.MM.dd.HH

tab

- Elasticsearch Hosts
- A comma separated list of Elasticsearch data nodes that you identified in Step 10.

REST tab



Metron REST port
8082

Metron JDBC URL

Metron JDBC Driver

Metron JDBC username

Metron JDBC password
Type password
Retype Password

Metron JDBC platform

Metron REST port

Use 8082.

JDBC URL

jdbc:mysql://mysql_host:3306/metronrest

JDBC Driver

com.mysql.jdbc.Driver

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

- PostgreSQL
- MySQL
- H2
- Oracle

JDBC Username

Metron REST user name

JDBC Password

Metron REST password

Metron JDBC client path

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

Advanced Tab (Metron)

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

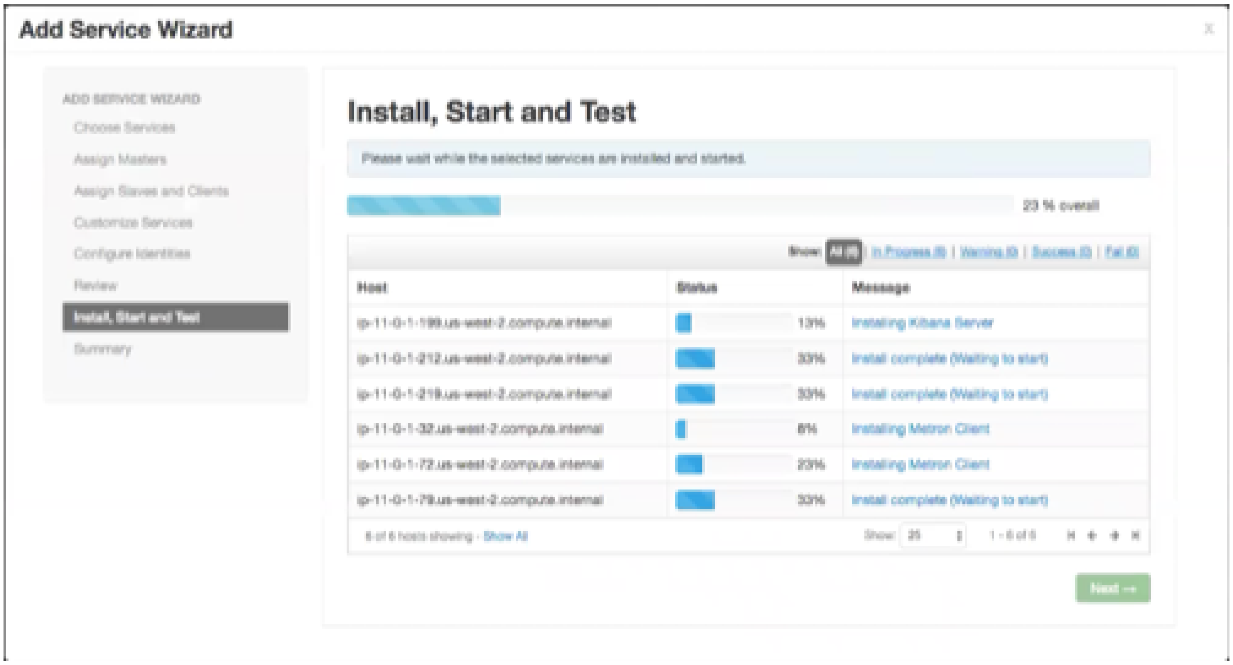
Misc tab

The service account users and groups are available under the **Misc** tab. These are the operating system accounts the service components will run as. If these users do not exist on your hosts, Ambari will automatically create the users and groups locally on the hosts. If these users already exist, Ambari will use those accounts.

Depending on how your environment is configured, you might not allow groupmod or usermod operations. If this is the case, you must be sure all users and groups are already created and be sure to select the **Skip group modifications** option on the **Misc** tab. This tells Ambari to not modify group membership for the service users.

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

Install, Start and Test Window



Install, Start and Test

Please wait while the selected services are installed and started.

23 % overall

Host	Status	Message
ip-11-0-1-199.us-west-2.compute.internal	13%	Installing Kibana Server
ip-11-0-1-212.us-west-2.compute.internal	33%	Install complete (Waiting to start)
ip-11-0-1-219.us-west-2.compute.internal	33%	Install complete (Waiting to start)
ip-11-0-1-32.us-west-2.compute.internal	8%	Installing Metron Client
ip-11-0-1-72.us-west-2.compute.internal	23%	Installing Metron Client
ip-11-0-1-79.us-west-2.compute.internal	33%	Install complete (Waiting to start)

6 of 6 hosts showing - Show All

12. If you need to make changes, use the left navigation bar 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.
13. (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.
14. When Successfully installed and started the services appears, click **Next**.

Launch the Metron Dashboard

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

Procedure

1. Select the Ambari **Service Action** menu and click **Kibana>Load Template** to load the Metron dashboards.
2. From the Quick Links pull-down menu, select Metron UI **Kibana**.

   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.

Switch Your Indexing Tool

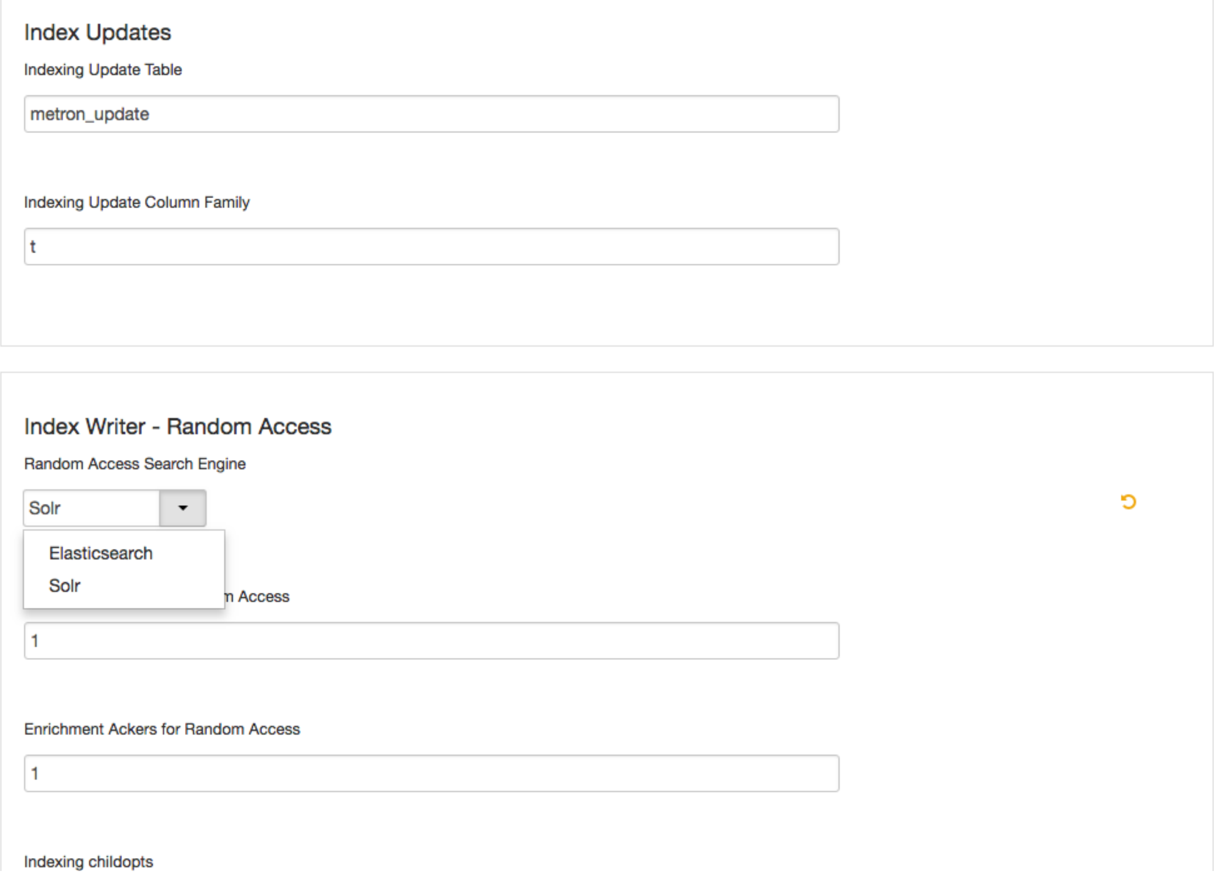
You can easily switch the indexing tool you use by using Ambari.

Before you begin

You must install and deploy Solr before you switch to using Solr for indexing.

Procedure

1. In Ambari, select **Metron** from the components panel.
2. If it is not already selected, click the **Configs** tab.
3. Click the **Indexing** tab.
4. Choose Solr in the **Index Writer - Random Access** pull down menu.



Index Updates

Indexing Update Table: metron_update

Indexing Update Column Family: t

Index Writer - Random Access

Random Access Search Engine: Solr

1

Enrichment Ackers for Random Access: 1

Indexing childopts

5. Click **Save**.

Import Apache Zeppelin Notebook Using Ambari

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

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**.

Related Tasks

Analyzing Enriched Data Using Apache Zeppelin

Streaming Data into HCP

To prepare for HCP to ingest data source data into HCP, 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 HCP includes parsers (for example, Bro, Snort, and YAF). Even though HCP includes parsers for these data sources, HCP 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 HCP, 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.

Related Tasks

Setting Up pcap to View Your Raw Data

Creating a Streaming Threat Intel Feed Source

Create a NiFi Flow to Stream Events to HCP

You can use NiFi to create a flow to capture events from the new data source and push them into HCP.

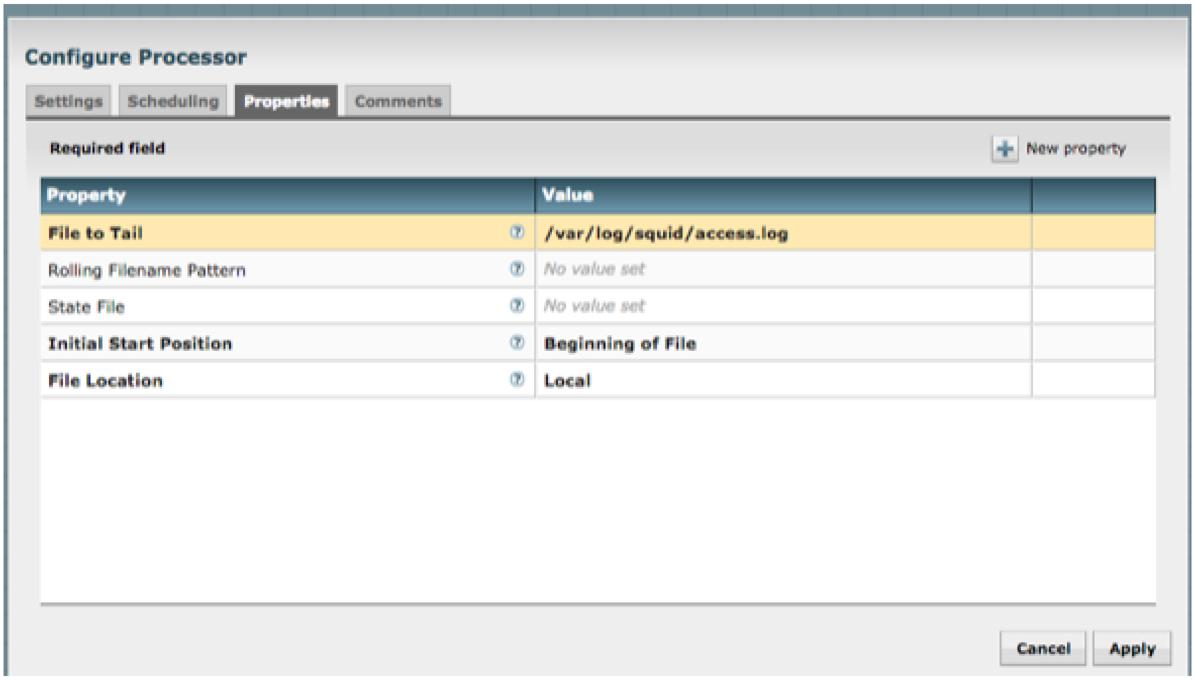
About this task

The following task is an example using the Squid data source. Prior to creating a NiFi flow to stream Squid events to HCP, you would need to install Squid and create parsers for the data source.

Procedure

1. Drag the first icon on the toolbar (the processor icon) to your workspace.
2. Select the TailFile type of processor and click **Add**.
3. Right-click the processor icon and select **Configure** to display the **Configure Processor** dialog box.
   a) In the **Settings** tab, change the name to Ingest $DATASOURCE Events.
   b) In the **Properties** tab, configure the following:

      NiFi Configure Processor Dialog Box EC2 Dashboard



Configure Processor

Required field

Property	Value
File to Tail	/var/log/squid/access.log
Rolling Filename Pattern	No value set
State File	No value set
Initial Start Position	Beginning of File
File Location	Local

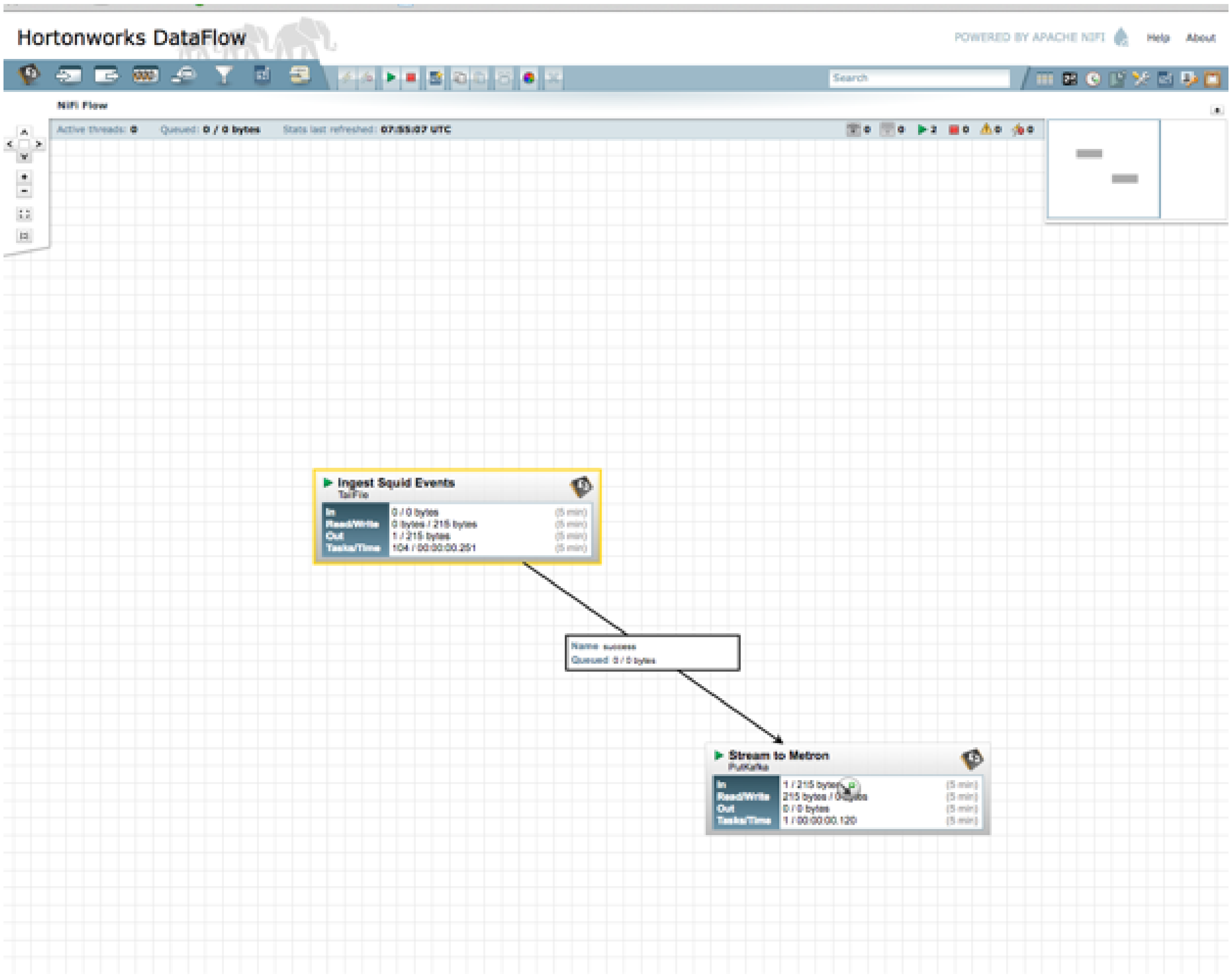
4. Repeat Step 1.
5. Select the PutKafka type of processor and click **Add**.
6. Right-click the processor and select **Configure**.
7. In the **Settings** tab, change the name to Stream to Metron and then click the relationship check boxes for failure and success.
8. In the Properties tab, set the following three properties:

   - Known Brokers: $KAFKA_HOST:6667
   - Topic Name: $DATAPROCESSOR
   - Client Name: nifi-$DATAPROCESSOR

9. Create a connection by dragging the arrow from the Ingest $DATAPROCESSOR Events processor to the Stream to Metron processor.
10. Press the Shift key and draw a box around both parsers to select the entire flow; then click the play button (green arrow).

    You should see all of the processor icons turn into green arrows:

    NiFi Configure Processor Dialog Box EC2 Dashboard



Active threads: 0 Queued: 0 / 0 bytes Stats last refreshed: 07:55:07 UTC

Ingest Squid Events
TailFile
In 0 / 0 bytes (5 min)
Read/Write 0 bytes / 215 bytes (5 min)
Out 1 / 215 bytes (5 min)
Tasks/Time 104 / 00:00:00.251 (5 min)

Name success
Queued 0 / 0 bytes

Stream to Metron
PutKafka
In 1 / 215 bytes (5 min)
Read/Write 215 bytes / 0 bytes (5 min)
Out 0 / 0 bytes (5 min)
Tasks/Time 1 / 00:00:00.120 (5 min)

11. Generate some data using the new data processor client.

You should see metrics on the processor of data being pushed into Metron.

12. Look at the Storm UI for the parser topology and you should see tuples coming in.

13. After about five minutes, you should see a new Elastic Search index called $DATAPROCESSOR_index* in the Elastic Admin UI.

What to do next

For more information about creating a NiFi data flow, see the NiFi documentation.

Verify That HCP Deployed Successfully for Ambari Install

After you install HCP, 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 HCP are deployed.

From Ambari, navigate to **Storm** > **Quick Links** > **Storm UI**.

You should see the following topologies listed:

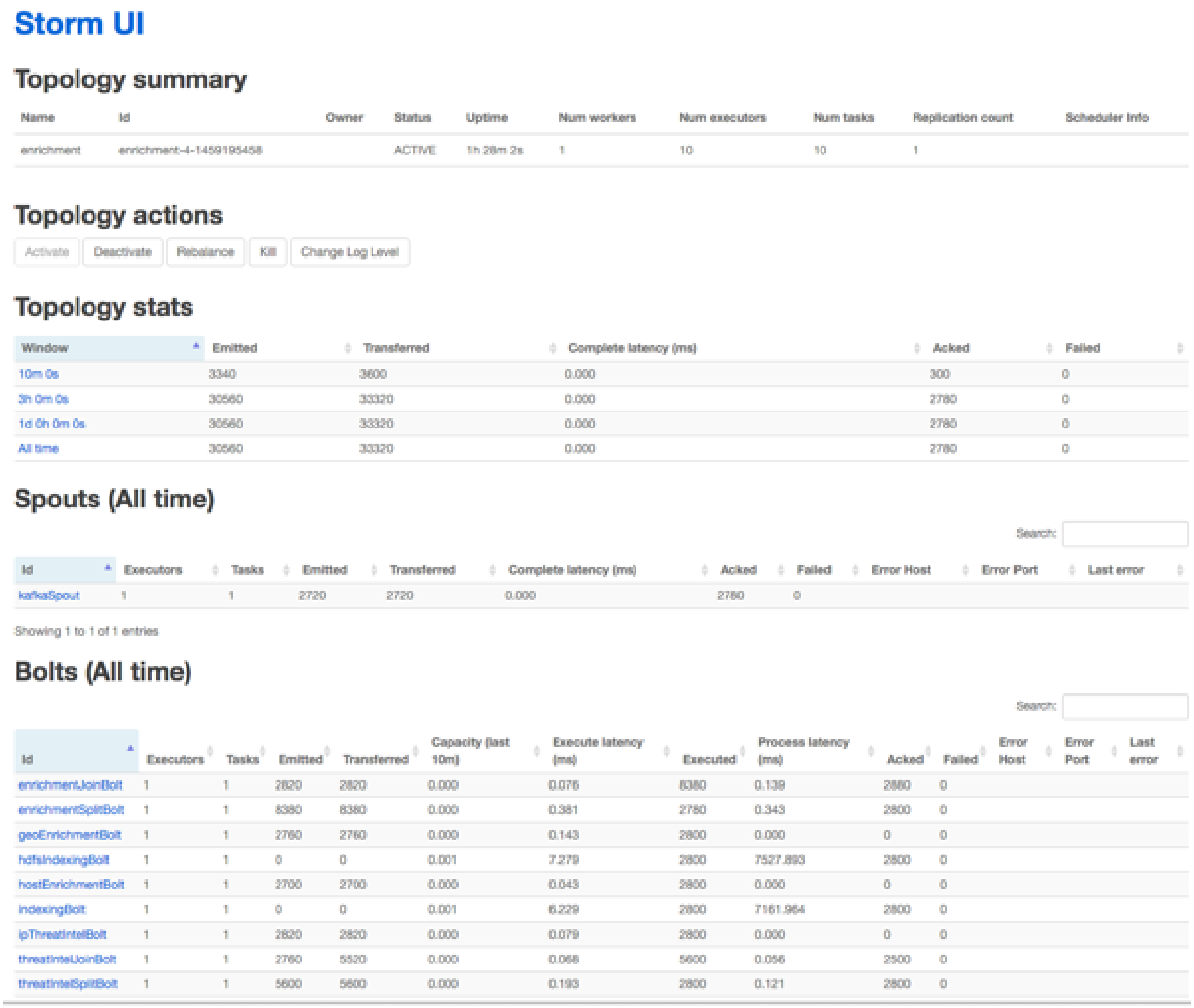
- Snort
- pcap
- YAF (Yet Another Flowmeter)

- 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

Topology summary

Name	Id	Owner	Status	Uptime	Num workers	Num executors	Num tasks	Replication count	Scheduler Info
enrichment	enrichment-4-1459195458		ACTIVE	1h 28m 2s	1	10	10	1	

Topology actions

Topology stats

Window	Emitted	Transferred	Complete latency (ms)	Acked	Failed
10m 0s	3340	3600	0.000	300	0
3h 0m 0s	30560	33320	0.000	2780	0
1d 0h 0m 0s	30560	33320	0.000	2780	0
All time	30560	33320	0.000	2780	0

Spouts (All time)

Id	Executors	Tasks	Emitted	Transferred	Complete latency (ms)	Acked	Failed	Error Host	Error Port	Last error
kafkaSpout	1	1	2720	2720	0.000	2780	0			

Showing 1 to 1 of 1 entries

Bolts (All time)

Id	Executors	Tasks	Emitted	Transferred	Capacity (last 10m)	Execute latency (ms)	Executed	Process latency (ms)	Acked	Failed	Error Host	Error Port	Last error
enrichmentJoinBolt	1	1	2820	2820	0.000	0.076	8380	0.139	2880	0			
enrichmentSplitBolt	1	1	8380	8380	0.000	0.381	2780	0.343	2800	0			
geoEnrichmentBolt	1	1	2760	2760	0.000	0.143	2800	0.000	0	0			
hdfsIndexingBolt	1	1	0	0	0.001	7.279	2800	7527.893	2800	0			
hostEnrichmentBolt	1	1	2700	2700	0.000	0.043	2800	0.000	0	0			
indexingBolt	1	1	0	0	0.001	6.229	2800	7161.964	2800	0			
ipThreatIntelBolt	1	1	2820	2820	0.000	0.079	2800	0.000	0	0			
threatIntelJoinBolt	1	1	2760	5520	0.000	0.068	5600	0.056	2500	0			
threatIntelSplitBolt	1	1	5600	5600	0.000	0.193	2800	0.121	2800	0			

3. 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 HCP to meet your own needs.

Launch HCP Management Module User Interface

The HCP Management Module user interface enables you to add and configure telemetry parsers to HCP. The Management Module UI is bundled with the HCP bits and you can launch the UI when you've completed your installation.

Procedure

1. From the Ambari Dashboard panel, click **Metron**.
2. Make sure the **Summary** tab is selected.
3. Double-click the Metron Management UI in the **Summary** list.

   The Metron Management UI tool should display in a separate browser tab.

What to do next

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

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"

Enable Kerberos

You can use Ambari to enable Kerberos for your Hortonworks Cybersecurity Platform (HCP) environment.

Checklist: Installing and Configuring the KDC

Ambari is able to configure Kerberos in the cluster to work with an existing MIT KDC, or existing Active Directory installation. This section describes the steps necessary to prepare for this integration.

You can choose to have Ambari connect to the KDC and automatically create the necessary Service and Ambari principals, generate and distribute the keytabs (“Automated Kerberos Setup”). Ambari also provides an advanced option to manually configure Kerberos. If you choose this option, you must create the principals, generate and distribute the keytabs. Ambari will not do this automatically (“Manual Kerberos Setup”).

Supported Key Distribution Center (KDC) Versions

- Microsoft Active Directory 2008 and above
- MIT Kerberos v5
- FreeIPA 4.x and above

There are four ways to install/configure the KDC:

- Using an existing MIT KDC
- Install a new MIT KDC (See "Optional: Install a new MIT KDC")
- Using an existing IPA
- Using an existing AD
- Using manual Kerberos setup

Option	Checklist
Using an existing MIT KDC	• Ambari Server and cluster hosts have network access to both the KDC and KDC admin hosts. • KDC administrative credentials are on-hand.
Install a new MIT KDC	See “Optional: Install a new MIT KDC”
Using an existing IPA	See “Optional: Use an Existing IPA”
Using an existing AD	• Ambari Server and cluster hosts have network access to, and be able to resolve the DNS names of, the Domain Controllers. • Active Directory secure LDAP (LDAPS) connectivity has been configured. • Active Directory User container for service principals has been created and is on-hand. For example, "OU=Hadoop,OU=People,dc=apache,dc=org" • Active Directory administrative credentials with delegated control of “Create, delete, and manage user accounts” on the previously mentioned User container are on-hand.
Using manual Kerberos setup	• Cluster hosts have network access to the KDC. • Kerberos client utilities (such as kinit) have been installed on every cluster host. • The Java Cryptography Extensions (JCE) have been setup on the Ambari Server host and all hosts in the cluster. • The Service and Ambari Principals will be manually created in the KDC before completing this wizard. • The keytabs for the Service and Ambari Principals will be manually created and distributed to cluster hosts before completing this wizard.

Optional: Install a new MIT KDC

The following gives a very high level description of the KDC installation process.

About this task

To get more information see specific Operating Systems documentation, such as RHEL documentation, CentOS documentation, or SLES documentation (links below).

Procedure

1. Install the KDC Server:

a) Install a new version of the KDC server:

OS Flavor	Enter
RHEL/CentOS/Oracle Linux	yum install krb5-server krb5-libs krb5-workstation
SLES	zypper install krb5 krb5-server krb5-client
Ubuntu/Debian	apt-get install krb5-kdc krb5-admin-server

b) Using a text editor, open the KDC server configuration file, located by default here: vi /etc/krb5.conf.

c) Change the [realms] section of this file by replacing the default “kerberos.example.com” setting for the kdc and admin_server properties with the Fully Qualified Domain Name of the KDC server host. In the following example, “kerberos.example.com” has been replaced with “my.kdc.server”.

```
realms]
 EXAMPLE.COM = {
   kdc = my.kdc.server
   admin_server = my.kdc.server
}
```

2. Use the utility kdb5_util to create the Kerberos database:

OS Flavor	**Enter**
RHEL/CentOS/Oracle Linux	kdb5_util create -s
SLES	kdb5_util create -s
Ubuntu/Debian	krb5_newrealm

3. Start the KDC server and the KDC admin server:

OS Flavor	**Enter**
RHEL/CentOS/Oracle Linux 6	/etc/rc.d/init.d/krb5kdc start /etc/rc.d/init.d/kadmin start
RHEL/CentOS/Oracle Linux 7	systemctl start krb5kdc systemctl start kadmin
SLES	rckrb5kdc start rckadmind start
Ubuntu/Debian	service krb5-kdc restart service krb5-admin-server restart

4. Set up the KDC server to auto-start on boot:

OS Flavor	**Enter**
RHEL/CentOS/Oracle Linux 6	chkconfig krb5kdc on chkconfig kadmin on
RHEL/CentOS/Oracle Linux 7	systemctl enable krb5kdc systemctl enable kadmin
SLES	chkconfig rckrb5kdc on chkconfig rckadmind on
Ubuntu/Debian	update-rc.d krb5-kdc defaults update-rc.d krb5-admin-server defaults

5. Create a Kerberos Admin:

Kerberos principals can be created either on the KDC machine itself or through the network, using an “admin” principal. The following instructions assume you are using the KDC machine and using the kadmin.local command line administration utility. Using kadmin.local on the KDC machine allows you to create principals without needing to create a separate "admin" principal before you start.

a) Create a KDC admin by creating an admin principal: kadmin.local -q "addprinc admin/admin".

b) Confirm that this admin principal has permissions in the KDC ACL. Using a text editor, open the KDC ACL file:

OS Flavor	**Enter**
RHEL/CentOS/Oracle Linux	vi /var/kerberos/krb5kdc/kadm5.acl
SLES	vi /var/lib/kerberos/krb5kdc/kadm5.acl
Ubuntu/Debian	vi /etc/krb5kdc/kadm5.acl

c) Ensure that the KDC ACL file includes an entry so to allow the admin principal to administer the KDC for your specific realm. When using a realm that is different than EXAMPLE.COM, be sure there is an entry for the realm you are using. If not present, principal creation will fail. For example, for an admin/admin@HADOOP.COM principal, you should have an entry: */admin@HADOOP.COM *.

d) After editing and saving the kadm5.acl file, you must restart the kadmin process:

OS Flavor	Enter
RHEL/CentOS/Oracle Linux 6	/etc/rc.d/init.d/kadmin restart
RHEL/CentOS/Oracle Linux 7	systemctl restart kadmin
SLES	rckadmind restart
Ubuntu/Debian	service krb5-admin-server restart

Optional: Use an Existing IPA

You can use an existing FreeIPA setup with Kerberos.

To use an existing IPA KDC with Automated Kerberos Setup, you must prepare the following:

- All cluster hosts should be joined to the IPA domain and registered in DNS- If IPA is not configured to authoritatively manage DNS, explicitly configuring the private IP and corresponding fully qualified domain names of all hosts, in the /etc/hosts file on all the hosts is recommended.
- If you do not plan on using Ambari to manage the krb5.conf file, ensure the following is set in each krb5.conf file in your cluster: default_ccache_name = /tmp/krb5cc_%{uid} - Redhat/Centos 7.x changed the default ticket cache to keyring, which is problematic for the hadoop components.
- The Java Cryptography Extensions (JCE) have been setup on the Ambari Server host and all hosts in the cluster- If during installation you chose to use the Ambari provided JDK, this has already been done for you. If you configured a custom JDK, ensure the unlimited strength JCE policies are in place on all nodes. For more information, refer to “Install the JCE for Kerberos”.

Please also note:

- If you plan on leveraging this IPA to create trusts with other KDCs, please follow the FreeIPA “Considerations for Active Directory integration” to ensure your hosts use a non-overlapping DNS domain, with matching uppercase REALM.
- Kerberos authentication allows maximum 3 seconds time discrepancy. Use of IPA’s NTP server or an external time management service is highly recommended for all cluster hosts, including the FreeIPA host.
- To avoid exposing the IPA admin account, consider creating a dedicated hadoopadmin account that is a member of the admins group, or has been added to a role with User & Service Administratration privileges. Remember to reset the initial temporary password for the account before use in Ambari. For more details on this process see the section below.

Creating an IPA account for use with Ambari

Example creating hadoopadmin account with explicit privileges

```
# obtain valid ticket as IPA administrator
kinit admin

# create a new principal to be used for ambari kerberos administration
ipa user-add hadoopadmin --first=Hadoop --last=Admin --password

# create a role and and give it privilege to manage users and services
ipa role-add hadoopadminrole
ipa role-add-privilege hadoopadminrole --privileges="User Administrators"
ipa role-add-privilege hadoopadminrole --privileges="Service Administrators"

# add the hadoopadmin user to the role
ipa role-add-member hadoopadminrole --users=hadoopadmin

# login once, or kinit, to reset the initial temporary password for the
 hadoopadmin account
kinit hadoopadmin
```

Important: Do not install an Ambari Agent on the IPA host.

- IPA leverages the SPNEGO principal (HTTP/ipa.your.domain.com) for secure access to its Web UI component. Installing the Ambari Agent on the IPA host causes the kvno of SPNEGO principal to increase, which causes problems for IPA HTTP server. If you have already accidentally done this and IPA is not able to start, symlink IPA’s http keytab path (/var/lib/ipa/gssproxy/http.keytab) to /etc/security/keytabs/spnego.service.keytab and contact your IPA provider’s support.
- The /etc/krb5.conf file on the IPA host has some additional properties not captured in Ambari’s krb5.conf template. Since letting Ambari manage krb5.conf on the cluster hosts is recommended, making the IPA host a part of the cluster is problematic for the IPA services. If you had this option checked when the ambari agent was installed, and do not have a backup of the original krb5.conf, reference the “krb5.conf template” to restore immediate functionality.

Install the JCE for Kerberos

Before enabling Kerberos in the cluster, you must deploy the Java Cryptography Extension (JCE) security policy files on the Ambari Server and on all hosts in the cluster, including the Ambari Server. If you are using OpenJDK, some distributions of the OpenJDK (such as RHEL/CentOS and Ubuntu) come with unlimited strength JCE automatically and therefore, installation of JCE is not required.

Procedure

1. On the Ambari Server, obtain the JCE policy file appropriate for the JDK version in your cluster:

Option	
Oracle JDK 1.8	JCE Unlimited Strength Jurisdiction Policy Files 8 Download
Oracle JDK 1.7	JCE Unlimited Strength Jurisdiction Policy Files 7 Download

   ```
   wget --no-check-certificate --no-cookies --header "Cookie: oraclelicense=accept-securebackup-cookie" "http://
   download.oracle.com/otn-pub/java/jce/8/jce_policy-8.zip"
   ```
2. Save the policy file archive in a temporary location.
3. On Ambari Server and on each host in the cluster, add the unlimited security policy JCE jars to $JAVA_HOME/jre/lib/security/.

   For example, run the following to extract the policy jars into the JDK installed on your host:

   ```
   unzip -o -j -q jce_policy-8.zip -d /usr/jdk64/jdk1.8.0_40/jre/lib/
   security/
   ```
4. Restart Ambari Server: sudo ambari-server restart.

What to do next

Proceed to “Running the Kerberos Security Wizard”.

Launch the Kerberos Wizard (Automated Setup)

Choose the Kerberos Wizard Automated Setup if you will use an existing MIT KDC or Active Directory, as opposed to managing Kerberos principals and keytabs manually.

Procedure

1. Be sure you have installed and configured your KDC and have prepared the JCE on each host in the cluster.
2. Log in to Ambari Web and Browse to Admin > Kerberos.

3. Click “Enable Kerberos” to launch the wizard.
4. Select the type of KDC you are using and confirm you have met the prerequisites.
5. Provide information about the KDC and admin account.

   a) In the KDC section, enter the following information:

   - In the KDC Host field, the IP address or FQDN for the KDC host. Optionally a port number may be included.
   - In the Realm name field, the default realm to use when creating service principals.
   - (Optional) In the Domains field, provide a list of patterns to use to map hosts in the cluster to the appropriate realm. For example, if your hosts have a common domain in their FQDN such as host1.hortonworks.local and host2.hortonworks.local, you would set this to: .hortonworks.local,hortonworks.local

   b) In the Kadmin section, enter the following information:

   - In the Kadmin Host field, the IP address or FQDN for the KDC administrative host. Optionally a port number may be included.
   - The Admin principal and password that will be used to create principals and keytabs.
   - (Optional) If you have configured Ambari for encrypted passwords, the Save Admin Credentials option will be enabled. With this option, you can have Ambari store the KDC Admin credentials to use when making cluster changes. Refer to “Managing Admin Credentials” for more information on this option.

6. Modify any advanced Kerberos settings based on your environment.

   a) (Optional) To manage your Kerberos client krb5.conf manually (and not have Ambari manage the krb5.conf), expand the Advanced krb5-conf section and uncheck the "Manage" option. You must have the krb5.conf configured on each host.

   When manually managing the krb5.conf it is recommended to ensure that DNS is not used for looking up KDC, and REALM entries. Relying on DNS can cause negative performance, and functional impact. To ensure that DNS is not used, ensure the following entries are set in the libdefaults section of your configuration.

   ```
   [libdefaults]
   dns_lookup_kdc = false
   dns_lookup_realm = false
   ```

   b) (Optional) to configure any additional KDC's to be used for this environment, add an entry for each additional KDC to the realms section of the Advanced krb5-conf's krb5-conf template.

   ```
   kdc = {{kdc_host}}
   kdc = otherkdc.example.com
   ```

   c) (Optional) To not have Ambari install the Kerberos client libraries on all hosts, expand the Advanced kerberos-env section and uncheck the “Install OS-specific Kerberos client package(s)” option. You must have the Kerberos client utilities installed on each host.

   d) (Optional) If your Kerberos client libraries are in non-standard path locations, expand the Advanced kerberos-env section and adjust the “Executable Search Paths” option.

   e) (Optional) If your KDC has a password policy, expand the Advanced kerberos-env section and adjust the Password options.

   f) (Optional) Ambari will test your Kerberos settings by generating a test principal and authenticating with that principal. To customize the test principal name that Ambari will use, expand the Advanced kerberos-env section and adjust the Test Kerberos Principal value. By default, the test principal name is a combination of cluster name and date (${cluster_name}-${short_date}). This test principal will be deleted after the test is complete.

   g) (Optional) If you need to customize the attributes for the principals Ambari will create, when using Active Directory, see “Customizing the Attribute Template” for more information. When using MIT KDC, you can pass Principal Attributes options in the Advanced kerberos-env section. For example, you can set options related to pre-auth or max. renew life by passing:
   -requires_preauth -maxrenewlife "7 days"

7. Proceed with the install.
8. Ambari will install Kerberos clients on the hosts and test access to the KDC by testing that Ambari can create a principal, generate a keytab and distribute that keytab.
9. Customize the Kerberos identities used by Hadoop and proceed to kerberize the cluster.

   On the Configure Identities step, be sure to review the principal names, particularly the Ambari Principals on the General tab. These principal names, by default, append the name of the cluster to each of the Ambari principals. You can leave this as default or adjust these by removing the "-${cluster-name}" from principal name string. For example, if your cluster is named HDP and your realm is EXAMPLE.COM, the hdfs principal will be created as hdfs-HDP@EXAMPLE.COM.
10. Confirm your configuration. You can optionally download a CSV file of the principals and keytabs that Ambari will automatically create.
11. Click Next to start the process.
12. After principals have been created and keytabs have been generated and distributed, Ambari updates the cluster configurations, then starts and tests the Services in the cluster.
13. Exit the wizard when complete.
14. Ambari Server communicates with components in the cluster, and now with Kerberos setup, you need to make sure Ambari Server is setup for Kerberos. As part of the automated Kerberos setup process, Ambari Server has been given a keytab and setup is performed. All you need to do is restart Ambari Server for that to take effect. Therefore, restart Ambari Server at this time: ambari-server restart.