Cloudera Flow Management 2.1.2

In-place Upgrade of HDF to CFM on CDP Guide

Date published: 2019-06-26
Date modified: 2021-12-21

CLOUDERA

https://docs.cloudera.com/

Legal Notice

Contents

In-place Upgrade of HDF to CFM on CDP

Review the steps to upgrade from HDF to CFM on CDP Private Cloud Base. Learn the general steps required to move your NiFi dataflow and NiFi Registry versioned flows from an HDF cluster to a CFM cluster on CDP Private Cloud Base.

Cloudera encourages you to read through this entire document before starting the upgrade process, so that you understand the interdependencies and order of the steps. Cloudera also recommends that you validate these steps in a test environment to adjust and account for any special configurations for your cluster.

Important: Authorization through Apache Ranger is just one element of a secure production cluster: Cloudera supports Ranger only when it runs on a cluster where Kerberos is enabled to authenticate users.

The high-level steps for the in-place upgrade are as shown in the following image:

Before you upgrade

Before you start the HDF to CFM in-place upgrade process, ensure you understand the upgrade path available to you and your cluster meets the required prerequisites.

Upgrade path

The upgrade path is to use AM2CM tool and to upgrade from HDF 3.5.2.0 to CDP 7.1.7 with CFM 2.1.2. If you have an earlier version of HDF, then first upgrade to HDF 3.5.2.0 and then upgrade to CDP 7.1.7 with CFM 2.1.2.

Note: HDF and CFM are packaged with a different component list. For details, see CFM component versions and HDF Component Support. Components that are part of HDF but not part of CFM will become part of the CDP package. For information on migrating workloads of HDF components that are not included in CFM into CDP, see Migrating workloads in the *CDP Private Cloud Upgrade* guide.

Procedure

1. SSH to host where Ambari is installed.

   The Ambari machine contains the Cloudera Manager server.
2. Assign values to the following variables:

```
export clustername=HDFTOCDF
export pfork=6
export ambariuser=admin
export ambaripwd=admin
export ambariport=8080
export backupdir=/data/backups
export ambariprotocol=http
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.262.b10-0.el7_8.x8
6_64
export ambariserver=`hostname -f`
export metricscollectorhost=ccycloud-3.am2cmhdf.root.hwx.site
export grafanahost=ccycloud-3.am2cmhdf.root.hwx.site
export infrahost=ccycloud-1.am2cmhdf.root.hwx.site
export kdchost=`hostname -f`
export kdcrealm=EXAMPLE.COM
export kdcpasswd=Cloudera123
[ "${ambariprotocol}" = "https" ] && export securecurl="-k" || export sec
urecurl=""
```

   You need to assign values to these variables because the scripts used in this migration uses these predefined variables. The following list explains the environment variables:

   - export clustername=HDFTOCDF

     This is the cluster name in the Ambari UI. An Ambari server could manage more than one cluster, and when that happens the cluster name identifies the cluster. This is important, because this value is used when calling Ambari REST API to export blueprint.
   - export ambariuser=admin

     Ambari user name which has right to perform Ambari REST calls and admin operations on the cluster.
   - export ambaripwd=admin

     Ambari password for the above user (the user name and password variables are used by curl bash scripts later in the documentation).
   - export ambariport=8080

     HTTP or HTTPs port number which is needed for Ambari REST API calls.
   - export backupdir=/data/backups

     The directory where you save the cluster blueprint.
   - export ambariprotocol=http

     Ambari server supported procotol (HTTP or HTTPS).
   - export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.262.b10-0.el7_8.x86_64

     JAVA_HOME directory which is needed for some bash script.
   - export ambariserver=`hostname -f`

     Hostname of the machine where the Ambari server is running.

- export metricscollectorhost=ccycloud-3.am2cmhdf.root.hwx.site

  Hostname of the metrics collector. Usually it is the machine where the Ambari server is running.
- export grafanahost=ccycloud-3.am2cmhdf.root.hwx.site

  Hostname of the Grafana host. Usually it is the machine where the Ambari server is running.
- export infrahost=ccycloud-1.am2cmhdf.root.hwx.site

  Hostname of the Infra host. Usually it is the machine where the Ambari server is running.
- export kdchost=`hostname -f`

  If Kerberos is used, this should point to the machine where Kerberos server is running.
- export kdcrealm=EXAMPLE.COM

  If Kerberos is used, this should point to the Kerberos server realm.
- export kdcpasswd=Cloudera123

  If Kerberos is used, this should point to Kerberos server admin password.
- ["${ambariprotocol}" = "https"] && export securecurl="-k" || export securecurl=""

  Initializes securecurl variable, based on previous values, used in some of the bash script later in the documentation.

Installing Solr service

Ranger uses Apache Solr service to store data. So, you must install Solr service on the cluster before the upgrade, if it is not installed already. You can do it through the Ambari UI.

Procedure

1. Go to Services Add Service on the Ambari UI.

   The Choose Services window appears.
2. Select Infra Solr service in the wizard and click Next.
3. Follow the instructions to configure settings and advanced properties and install the service.
4. Click Deploy.

Checking cluster services

You must ensure that all services, you installed, are running in your cluster.

Perform one of the following tasks for checking if the cluster services are running:

- Manually from the Ambari UI

  Run the service check for each service under Service Actions Run Service Check .
- Through Ambari API

```
curl ${securecurl} -u "${ambariuser}":"${ambaripwd}" -i -H 'X-Requested-
By: ambari' -X PUT -d '{"RequestInfo": {"context" :"Start All via REST"}
, "Body": {"ServiceInfo": {"state": "STARTED"}}}' ${ambariprotocol}://${
ambariserver}:${ambariport}/api/v1/clusters/${clustername}/services/
```

  Restart all services. Check if time service stops and starts:

```
###STOP
# curl ${securecurl} -u "${ambariuser}":"${ambaripwd}" -i -H 'X-Requeste
d-By: ambari' -X PUT -d '{"RequestInfo": {"context" :"Stop All via REST"},
```

```
 "Body": {"ServiceInfo": {"state": "INSTALLED"}}}' ${ambariprotocol}://${a
mbariserver}:${ambariport}/api/v1/clusters/${clustername}/services/
###START (takes a long time)
# curl ${securecurl} -u "${ambariuser}":"${ambaripwd}" -i -H 'X-Requested-
By: ambari' -X PUT -d '{"RequestInfo": {"context" :"Start All via REST"},
 "Body": {"ServiceInfo": {"state": "STARTED"}}}' ${ambariprotocol}://${a
mbariserver}:${ambariport}/api/v1/clusters/${clustername}/services/
```

If any cluster service is not running, fix the service settings before you proceed.

Checking service accounts

You must ensure that all service accounts required for the upgrade are present. These service accounts include Ambari metrics user, smoke user, Hadoop group, infra solr user, NiFi user, NiFi Registry user, Ranger group, Ranger user, and ZooKeeper user.

Procedure

1. Run the following command:
   - For service account: cat /etc/passwd
   - For groups: cat /etc/group
2. Check if all the service accounts, mentioned in the following table, are present.

Name	Value
Ambari Metrics User	ams
Smoke User	ambari-qa
Hadoop Group	hadoop
Infra Solr User	infra-solr
Nifi User	nifi
Nifi Registry User	nifiregistry
Ranger Group	ranger (Optional. Needed when you use ranger service in Ambari)
Ranger User	ranger
ZooKeeper User	zookeeper

3. If any service account is missing, create that account in your cluster.

Collect data for upgrade

Collect passwords and Kafka broker IDs before you start the upgrade to make sure that required data is available during upgrade and post configuration.

Collect Ranger passwords

If you use Ranger service, you need to collect the passwords for Ranger.

Collect the following passwords for Ranger:

- Ranger database password

  The password used for ranger database
- Ranger admin password

  The admin web interface password
- Ranger usersync password

- Ranger tagsync password
- Ranger keyadmin password
- Ranger usersync ldap password

  If LDAP user sync setting is configured.

Collect Nifi Registry database password

You need to migrate the NiFi Registry database password manually during the upgrade.

If you forget your database password, but you know your password which used to encrypt the NiFi Registry password, you can decrypt the nifi.registry.properties file properties with the following command:

```
/usr/hdf/current/nifi-toolkit/bin/encrypt-config.sh -r /usr/hdf/current/nifi
-registry/conf/nifi-registry.properties --decrypt --nifiRegistry -v -p <<pas
sword_used_for_encryption>>
```

The output of the command contains the database password.

Extracting Kafka broker IDs

You must extract the Kafka broker IDs before you upgrade the HDF cluster. The Kafka broker IDs are used in upgrading HDF to CFM on CDP cluster, where you need to override the kafka-broker-ids.ini files with the created one.

You must extract the Kafka broker IDs manually from the HDF 3.5.2.0 cluster. When you upgrade to Cloudera Manager, you must manually enter the broker IDs to the real values in Kafka.

1. Create the kafka-broker-ids.ini file.
2. Navigate to each Kafka broker host > $log.dirs/meta.properties and collect the broker.id value.

   Here, $log.dirs refers to an Ambari configuration value.
3. Copy hostname broker.id to the kafka-broker-ids.ini file.

   An example of the file format is as follows:

   ```
   ctr-e153-xxxxx-xxxx71.cloudera.site 1001
   ctr-e153-xxxxx-xxxx72.cloudera.site 1002
   ctr-e153-xxxxx-xxxx73.cloudera.site 1003
   ```

Related Information

Kafka documentation

Downloading Ambari blueprint

Ambari blueprint is a JSON file that describes the entire cluster including number of clusters present, names of the clusters, services installed on those clusters, configuration details of the installed services, and so on. The JSON file helps to upgrade the cluster. The Ambari blueprint helps you to import or export your clusters if you want to replicate your clusters with the same configuration to any other cluster.

About this task

You need the Ambari configuration to install and configure Ranger. You need to get the Ambari blueprint.

If your cluster has Kerberos and SSL enabled, you need to disable them.

Procedure

1. Go to Dashboard Kerberos in the Ambari UI.

2. Run the following command to download the Ambari blueprint:

```
curl ${securecurl} -H "X-Requested-By: ambari" -X GET -u ${ambariuser}:$
{ambaripwd} ${ambariprotocol}://${ambariserver}:${ambariport}/api/v1/clu
sters/${clustername}?format=blueprint > "${backupdir}"/${clustername}_bl
ueprint_"$(date +"%Y%m%d%H%M%S")".json
```

Extending the JSON file

After you download your Ambari blueprint, you have to extend the JSON file manually with the host details. You need to extend the JSON file in order to let Cloudera Manager know which component to install at which host.

Procedure

1. Check how many hostgroups exist in the blueprint.
2. Collect all machine hostnames and IP addresses.
3. Identify which machine belongs to which hostgroup.
4. Extend all hostgroups with the data which describe the belonging host with the following JSON format:

```
"hosts" : [
            {
              "hostname" : "machine host name",
              "hostipaddress" : "machine ip address"
            }
          ]
```

Note: The section should be properly inserted with comma to preserve the valid JSON format.

Cloudera Manager installation and setup

After you successfully complete the prerequisites, you can install and configure Cloudera Manager in your cluster.

Checking pre-installation setup

You must ensure that the Ambari blueprint, that you download, contains extended hostgroups with hosts, and hosts and all hostgroups contain host entries.

Procedure

1. Download the latest Ambari blueprint and extend hostgroups with hosts.
2. Ensure that the blueprint contains hosts included and all host groups contain host entries:

```
[root@ccycloud-1 backups]# egrep '"hostname" :' /data/backups/HDFTOCDF_b
lueprint_with_hosts_20201103144235.json | sort
          "hostname" : "ccycloud-1.am2cmhdf.root.hwx.site",
          "hostname" : "ccycloud-2.am2cmhdf.root.hwx.site",
          "hostname" : "ccycloud-3.am2cmhdf.root.hwx.site",
```

```
          "hostname" : "ccycloud-4.am2cmhdf.root.hwx.site",
```

For more details,

```
egrep -C3 '"hostname" :' /data/backups/HDFTOCDF_blueprint_with_hosts_202
01103144235.json
      ],
        "hosts" : [
        {
          "hostname" : "ccycloud-2.am2cmhdf.root.hwx.site",
          "hostipaddress" : "172.27.133.196"
        }
      ],
--
      ],
      "hosts" : [
        {
          "hostname" : "ccycloud-1.am2cmhdf.root.hwx.site",
          "hostipaddress" : "172.27.60.128"
        }
      ],
--
      ],
       "hosts" : [
        {
          "hostname" : "ccycloud-4.am2cmhdf.root.hwx.site",
          "hostipaddress" : "172.27.92.133"
        }
      ],
--
      ],
       "hosts" : [
        {
          "hostname" : "ccycloud-3.am2cmhdf.root.hwx.site",
          "hostipaddress" : "172.27.18.194"
        }
      ],
```

Configuring Cloudera Manager repository

You need to download the Cloudera Manager repository, configure the repository file, and copy the file on all hosts in the cluster. You configure the repository to deploy Cloudera Manager and daemons on the machines.

Procedure

1. Download the repository on the host currently running the Ambari server:

```
wget https://<username>:<password>@archive.cloudera.com/p/cm7/7.4.4/red
hat7/yum/cloudera-manager.repo -P /etc/yum.repos.d/
```

2. Configure the repository file to include username and password for the Cloudera paywall:

```
# vi /etc/yum.repos.d/cloudera-manager.repo

[cloudera-manager]
name=Cloudera Manager 7.4.4
baseurl=https://archive.cloudera.com/p/cm7/7.4.4/redhat7/yum/
gpgkey=https://archive.cloudera.com/p/cm7/7.4.4/redhat7/yum/RPM-GPG-KEY-
cloudera
username=changeme
```

```
password=changeme
gpgcheck=1
enabled=1
autorefresh=0
type=rpm-md
```

3. Copy the repository file on all hosts in the cluster:

```
for host in $(echo \
ccycloud-2.am2cmhdf.root.hwx.site \
ccycloud-3.am2cmhdf.root.hwx.site \
ccycloud-4.am2cmhdf.root.hwx.site \
); do scp /etc/yum.repos.d/cloudera-manager.repo $host:/etc/yum.repos.d/
cloudera-manager.repo ;done
```

Installing Cloudera Manager server and agents

You need to install Cloudera Manager server on the Ambari server host, and Cloudera Manager agents and daemons on all hosts. Otherwise, you will not be able to upgrade your cluster.

Procedure

1. Install Cloudera Manager server on the Ambari server host:

```
yum install cloudera-manager-server
```

2. Install Cloudera Manager agents and daemons on all hosts:

```
for host in $(echo \
ccycloud-1.am2cmhdf.root.hwx.site  \
ccycloud-2.am2cmhdf.root.hwx.site \
ccycloud-3.am2cmhdf.root.hwx.site \
ccycloud-4.am2cmhdf.root.hwx.site \
); do ssh $host "yum install -y cloudera-manager-daemons cloudera-manager-
agent";done
```

Configuring database for Cloudera Manager

You need to configure the database for Cloudera Manager to use.

About this task

The examples you view in this section are for PostgreSQL. Cloudera Manager supports other databases as well. For more details about the syntax details of other databases, see *Syntax for scm_prepare_database.sh*.

Procedure

1. Configure the Reports Manager database:

```
[root@ccycloud-1 yum.repos.d]# sudo -u postgres psql
psql (9.2.24, server 10.12)
WARNING: psql version 9.2, server version 10.0.
         Some psql features might not work.
Type "help" for help.

postgres=# CREATE ROLE scm  LOGIN PASSWORD 'scm';
CREATE ROLE
postgres=# CREATE ROLE rman  LOGIN PASSWORD 'rman';
```

```
CREATE ROLE
postgres=# CREATE DATABASE rman OWNER rman ENCODING 'UTF8';
CREATE DATABASE
postgres=# CREATE DATABASE scm OWNER scm ENCODING 'UTF8';
CREATE DATABASE
postgres=#
```

2. Configure the Cloudera Manager database on the Cloudera Manager server host:

```
[root@ccycloud-1 yum.repos.d]# /opt/cloudera/cm/schema/scm_prepare_datab
ase.sh postgresql scm scm scm
JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.262.b10-0.el7_8.x86_64
Verifying that we can write to /etc/cloudera-scm-server
Creating SCM configuration file in /etc/cloudera-scm-server
Executing:  /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.262.b10-0.el7_8.x86_64/
bin/java -cp /usr/share/java/mysql-connector-java.jar:/usr/share/java/orac
le-connector-java.jar:/usr/share/java/postgresql-connector-java.jar:/opt
/cloudera/cm/schema/../lib/* com.cloudera.enterprise.dbutil.DbCommandExe
cutor /etc/cloudera-scm-server/db.properties com.cloudera.cmf.db.
log4j:ERROR Could not find value for key log4j.appender.A
log4j:ERROR Could not instantiate appender named "A".
[2020-11-05 11:17:18,802] INFO     0[main] - com.cloudera.enterprise.dbuti
l.DbCommandExecutor.testDbConnection(DbCommandExecutor.java) - Successfu
lly connected to database.
All done, your SCM database is configured correctly!
[root@ccycloud-1 yum.repos.d]#
```

3. Check the db.properties file and ensure that the settings match with the settings mentioned in the following script:

```
[root@ccycloud-1 yum.repos.d]# cat /etc/cloudera-scm-server/db.properties
# Auto-generated by scm_prepare_database.sh on Thu Nov  5 11:17:18 PST 202
0
#
# For information describing how to configure the Cloudera Manager Server
# to connect to databases, see the "Cloudera Manager Installation Guide."
#
com.cloudera.cmf.db.type=postgresql
com.cloudera.cmf.db.host=localhost
com.cloudera.cmf.db.name=scm
com.cloudera.cmf.db.user=scm
com.cloudera.cmf.db.setupType=EXTERNAL
com.cloudera.cmf.db.password=scm
[root@ccycloud-1 yum.repos.d]#
```

Related Information

Syntax for scm_prepare_database.sh

Starting Cloudera Manager server and adding license

To use Cloudera Manager after the installation, you must start the Cloudera Manager server, open the Cloudera Manager web UI, and add Cloudera Manager license.

Procedure

1. Start Cloudera Manager server:

```
[root@ccycloud-1 yum.repos.d]# systemctl start cloudera-scm-server
[root@ccycloud-1 yum.repos.d]# systemctl enable cloudera-scm-server
[root@ccycloud-1 yum.repos.d]# tail -f /var/log/cloudera-scm-server/cloud
era-scm-server.log
```

```
####Look for  'Started Jetty Server'
```

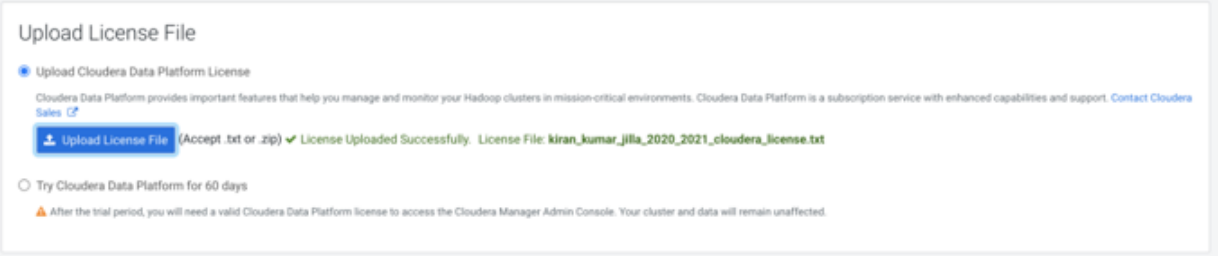
2. Open the Cloudera Manager web UI on browser:

   http://ccycloud-1.am2cmhdf.root.hwx.site:7180/cmf/login

   Username: admin

   Password: admin

3. Add the Cloudera Manager license.



4. Click Cloudera Manager on the top-left corner of your screen to exit the adding cluster step.

   Do not click Add Cluster using Wizard. If you click, it starts a wizard where you can create a cluster through the UI and not through the blueprint which is created by the AM2CM tool.

Configuring Cloudera agents and hosts

You need to add Cloudera agents and hosts before you perform the HDF upgrade.

About this task

You can either automate the process of configuring Cloudera agents and hosts, or manually perform the configuration.

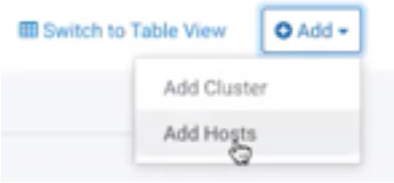
- To automate, change the agents configuration file before adding hosts to Cloudera Manager:

```
for host in $(echo \
ccycloud-1.am2cmhdf.root.hwx.site \
ccycloud-2.am2cmhdf.root.hwx.site \
ccycloud-3.am2cmhdf.root.hwx.site \
ccycloud-4.am2cmhdf.root.hwx.site \
); do ssh $host "sed -i "s/server_host=localhost/server_host=ccycloud-1
.am2cmhdf.root.hwx.site/" /etc/cloudera-scm-agent/config.ini
";done

for host in $(echo \
ccycloud-1.am2cmhdf.root.hwx.site  \
ccycloud-2.am2cmhdf.root.hwx.site \
ccycloud-3.am2cmhdf.root.hwx.site \
ccycloud-4.am2cmhdf.root.hwx.site \
); do ssh $host "systemctl restart cloudera-scm-agent";done

for host in $(echo \
ccycloud-1.am2cmhdf.root.hwx.site  \
ccycloud-2.am2cmhdf.root.hwx.site \
ccycloud-3.am2cmhdf.root.hwx.site \
ccycloud-4.am2cmhdf.root.hwx.site \
); do ssh $host "systemctl status cloudera-scm-agent";done
```

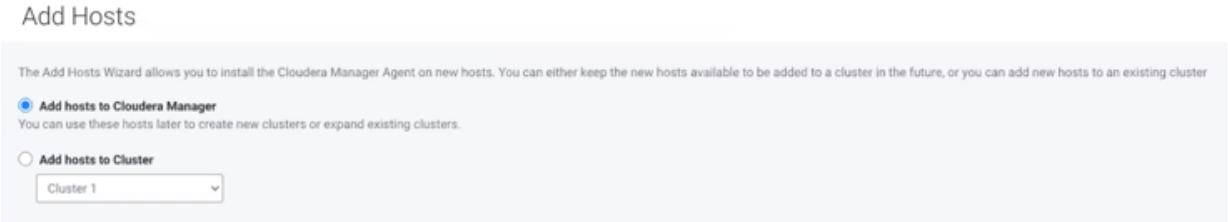
- To manually configure agents and hosts, perform the following steps:

Procedure

1. Go to the Cloudera Manager UI.
2. Click Add Add Hosts at the top-right corner of your screen.



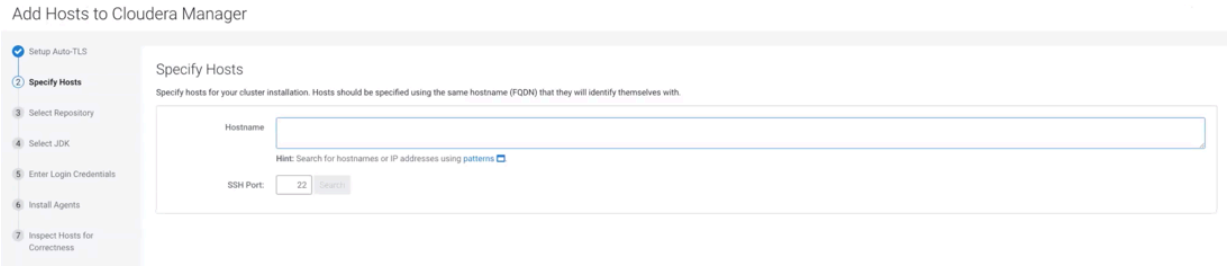
The Add Hosts window appears.



3. Select Add hosts to Cloudera Manager and click Continue.
   Do not add hosts to the cluster yet.

   The Setup Auto-TLS screen appears.
4. Click Continue.

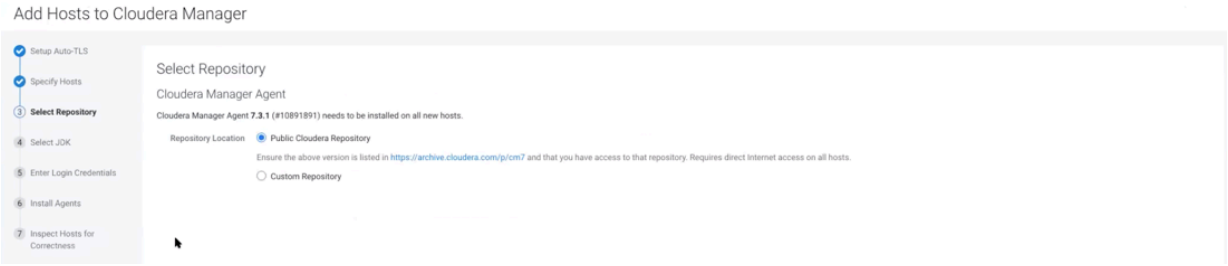
   The Specify Hosts screen appears.



5. Specify the hostname and click Search.

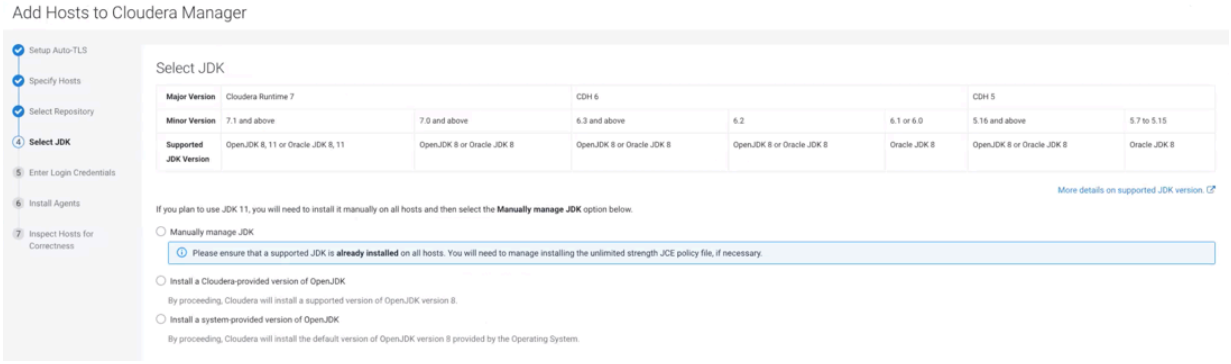
   The hostname appears.
6. Select the host and click Continue.

   The Select Repository screen appears.



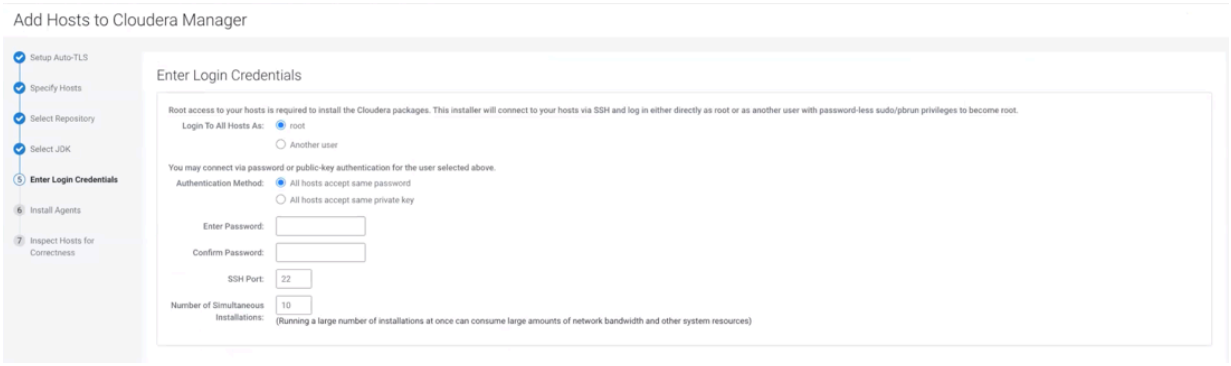
7. Select Public Cloudera Repository and click Continue.

   The Select JDK screen appears.



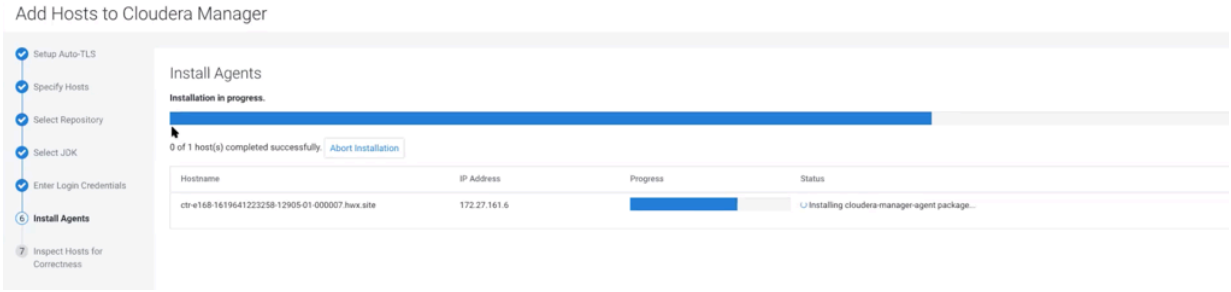
8. Select Manually manage JDK and click Continue.

   The Enter Login Credentials screen appears.



9. Enter your login credentials and click Continue.

   The Install Agents screen appears where the progress of the installation process is displayed.



   After the installation process is complete, the Inspect Hosts for Correctness screen appears. In this step, Cloudera Manager inspects the settings of your host and displays some suggestions.

10. Validate your settings and click Finish.

Adding Cloudera Management services

You need to add Cloudera Management services before you perform the HDF upgrade.

Procedure

1. Log in to the Cloudera Manager UI.

2. Click Add Add Cloudera Management Service .

Note: If you run into any issues at this step while you are upgrading to Cloudera Manager, you need to add Cloudera Management services manually, using a REST API.

a. Use a REST API client like Postman or curl to call the following URL with PUT method:

http://CM_SERVER_IP:7180/api/v49/cm/service

With raw/JSON: {}

(empty object)

Headers:

Content-Type: application/json

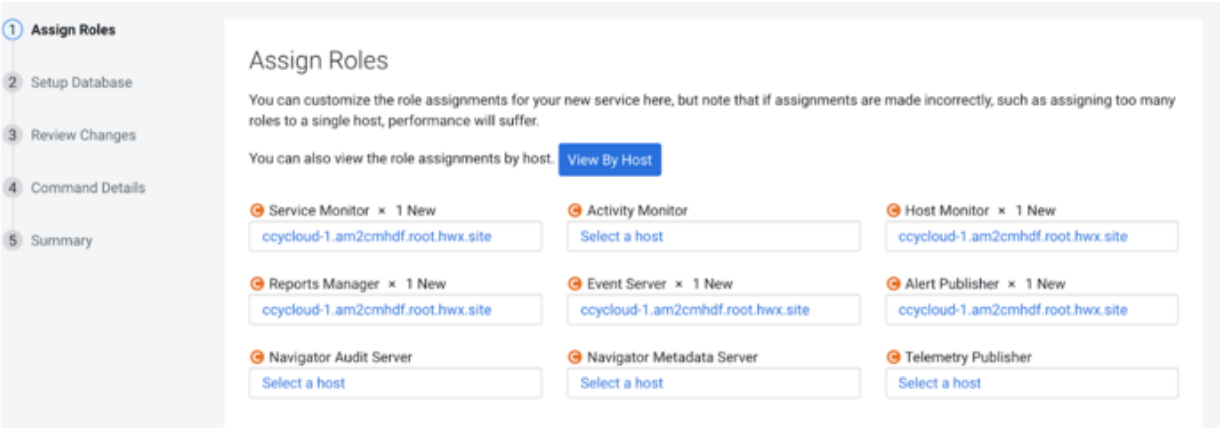
Curl example:

```
curl -X PUT -d '{}' -H "Content-Type: application/json" http://C
M_SERVER_URL_OR_IP:7180/api/v49/cm/service --user CM_USER:CM_PAS
SWORD
```

The Cloudera Management Service appears on the Cloudera Manager UI.

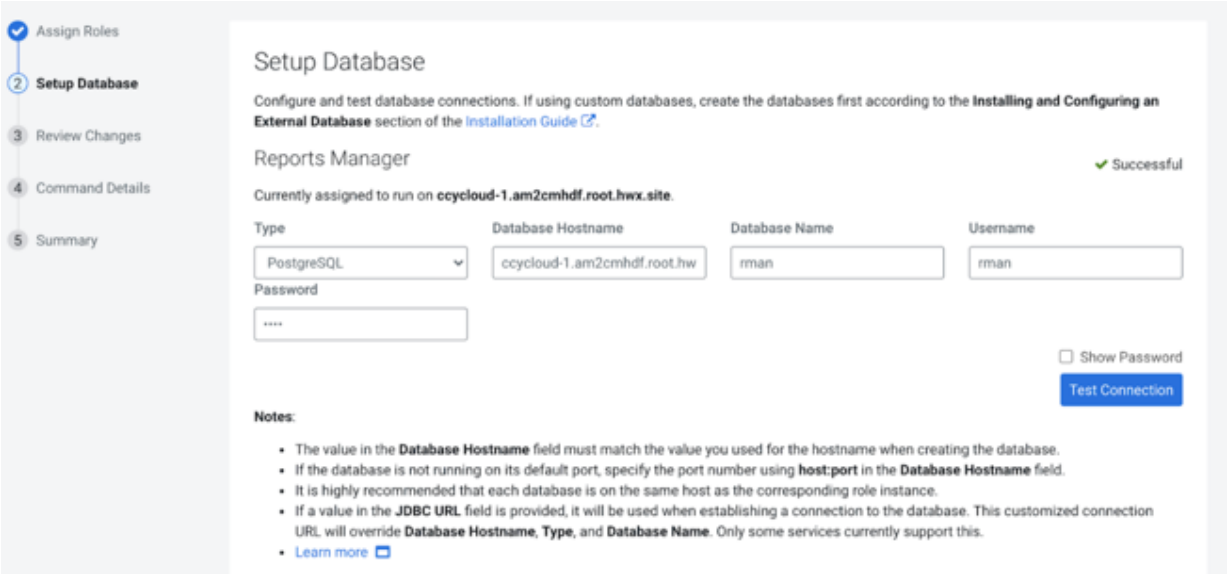
b. Select Cloudera Management Service and click Actions Add Role Instances .

The Assign Roles screen appears.



3. Customize the role assignments and click Continue.

   The Setup Database screen appears.



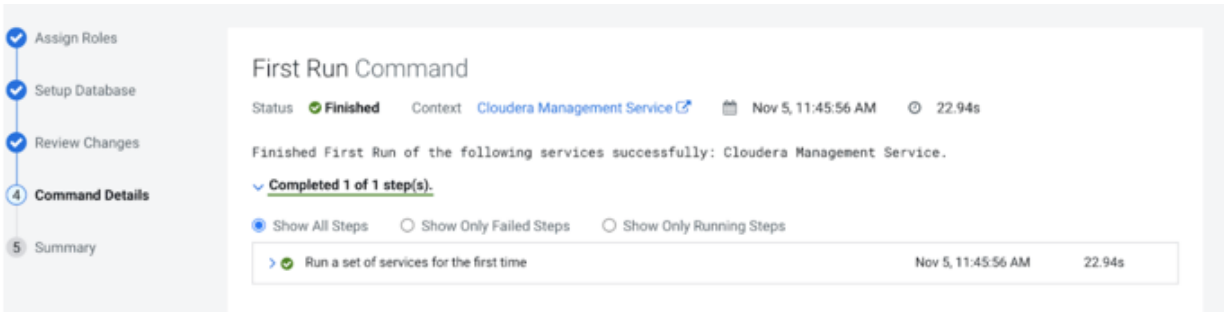
4. Configure and test the database connections, and click Continue.

   The Review Changes screen appears.

   Note: The database connection configuration is needed only for Cloudera Manager versions older than 7.4.4. Starting from Cloudera Manager version 7.4.4, you do not have to configure database connection.

5. Review the changes and click Continue.

   The Command Details screen appears.



6. Click Continue after the commands run successfully.

   The Summary screen appears.

7. Check the summary and click Finish.

Modifying host monitor port number

If you use NiFi with security enabled, then the Cloudera host monitor port number and the NiFi port number might be the same. That causes problems. So before continuing, you must modify it.

Procedure

1. Go to the Cloudera Manager main dashboard.
2. Select Cloudera Management Service.
3. Click Configuration.
4. Search for the Host Monitor Web UI HTTPS Port configuration value.
5. If the value is 9091, modify it to 9092 or other free port numbers, and save the configuration.

6. Restart Cloudera Management Service.

Upgrading HDF to CFM on CDP Private Cloud Base

Cloudera provides an AM2CM tool to enable you to upgrade HDF cluster to CFM cluster on CDP Private Cloud Base easily. You need to download the AM2CM tool, configure settings for the tool, and generate Cloudera Manager template. After you generate the Cloudera Manager template, you need to configure the parcel, deploy Cloudera Manager, add CFM parcel, and activate the parcel through the Cloudera Manager UI.

Procedure

1. Ensure that you have the latest Ambari blueprint (JSON file).
2. Download the AM2CM tool:

```
wget https://archive.cloudera.com/am2cm/2.3.0.0/redhat7/yum/tars/am
2cm/am2cm-tool-2.3.0.0-60.tar.gz
```

3. Extract the TAR file:

```
tar -xvf am2cm-tool-2.3.0.0-60.tar.gz
```

4. Check the project files:

```
# cd am2cm-2.3.0.0-60
                         # [root@ccycloud-1 am2cm-2.3.0.0-60]# ls
am2cm-2.2.0.2.3.0.0-60.jar  am2cm.sh  conf  lib  ranger_policy_migration.p
y


# [root@ccycloud-1 am2cm-2.3.0.0-60]# ls conf
ambari_blueprint_hostgroups.josn  blueprint.json  cm-config-mapping-custo
m.ini  cm-config-mapping.ini  log4j.properties  ranger_policy_migration.py
  service-config.ini  service-default-config.ini  user-settings.ini


# mv conf/blueprint.json conf/blueprint_initial.json
# [root@ccycloud-1 am2cm-2.3.0.0-60]# ls conf
ambari_blueprint_hostgroups.josn  blueprint_initial.json  cm-config-mappi
ng-custom.ini  cm-config-mapping.ini  log4j.properties  ranger_policy_mi
gration.py  service-config.ini  service-default-config.ini  user-setting
s.ini
```

5. Set JAVA_HOME if not already done:

```
[root@ccycloud-1 am2cm-2.3.0.0-60]# echo $JAVA_HOME
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.262.b10-0.el7_8.x86_64
```

6. Copy your Ambari blueprint to the right path, that is the conf directory of the AM2CM tool, from the Home directory:

```
cp /data/backups/HDFTOCDF_blueprint_with_hosts_20201103144235.json /
root/am2cm-2.3.0.0-60/conf
```

7. Configure the user-settings.ini file:

   It is really important to specify the cluster.name and cluster.displayname to the same original Ambari cluster name. Also, it is worth to fill all password field.

```
# Cluster details
```

```
cluster.name=zt-HDFTOCDF
cluster.displayname=HDFTOCDF
# Update this with correct CM/CDP version before running the tool
cluster.fullversion=7.1.7

cluster.originating_source=AM2CM version 2.0

# This is needed to update Ranger/Atlas policy service names.
ambari.cluster.name=HDFTOCDF

#This flag enables to translate Config Groups to Role config groups.
cm.rolegroups.enable = false

# FROM JDBC URL - tool reads only Hostname and port number - rest of the
details like Database type, Database name, database user reads from from
 Blueprint.
# Do not enclose double quotes for passwords, provide exact password.

# Hive JDBC settings
SERVICE_HIVE_hive_metastore_database_password=password
SERVICE_HIVE_hive_jdbc_url_override=JDBC_URL

# Oozie JDBC settings
SERVICE_OOZIE_oozie_database_password=password
SERVICE_OOZIE_oozie_service_JPAService_jdbc_url=JDBC_URL
# Ranger JDBC settings
SERVICE_RANGER_ranger_database_password=password
SERVICE_RANGER_rangeradmin_user_password=password
SERVICE_RANGER_rangerusersync_user_password=password
SERVICE_RANGER_rangertagsync_user_password=password
SERVICE_RANGER_rangerkeyadmin_user_password=password
SERVICE_RANGER_ranger_usersync_ldap_ldapbindpassword=password

SERVICE_RANGER_KMS_ranger_kms_master_key_password=password
SERVICE_RANGER_KMS_ranger_kms_database_password=password
#Knox Settings
SERVICE_KNOX_gateway_master_secret=admin
```

8. If your cluster contains Kafka service, please override the kafka-broker-ids.ini files with the one you created.
   For more details, see Extracting Kafka broker IDs.
9. Run the command to generate the Cloudera Manager template:

```
# cd am2cm-2.3.0.0-60
# chmod +x ./am2cm.sh
[root@ccycloud-1 am2cm-2.3.0.0-60]# ./am2cm.sh -bp conf/HDFTOCDF_blueprin
t_with_hosts_20201103144235.json -dt conf/cm_deployment_template.json --
source-version=HDF352

INPUT Ambari Blueprint : conf/HDFTOCDF_blueprint_with_hosts_202011031442
35.json
OUTPUT CM Template     : conf/cm_deployment_template.json

Starting blueprint to CM Template migration
Total number of hosts in blueprint: 4

Your cluster has services (listed below) that are not handled by this mi
gration tool.
AMBARI_METRICS
The tool will skip the above identified service related configs.

** This migration tool is a technical preview only **

Do you want to proceed with migration (Y OR N)? (N):
```

```
Y
Processing: LIVY
Processing: SOLR
Processing: TEZ
Processing: HDFS
Processing: OOZIE
Processing: SQOOP_CLIENT
Processing: NIFIREGISTRY
Processing: ZOOKEEPER
Processing: HBASE
Processing: YARN
Processing: RANGER_KMS
Processing: KNOX
Processing: ATLAS
Processing: HIVE_ON_TEZ
Processing: RANGER
Processing: HIVE
Processing: KAFKA
Processing: NIFI
Processing: SPARK_ON_YARN
Adding: QUEUEMANAGER
CM Template is generated at : /root/am2cm-2.3.0.0-60/conf/cm_deployment_
template.json
Successfully completed
```

10. Check for errors in the tool log for Cloudera Manager migration:

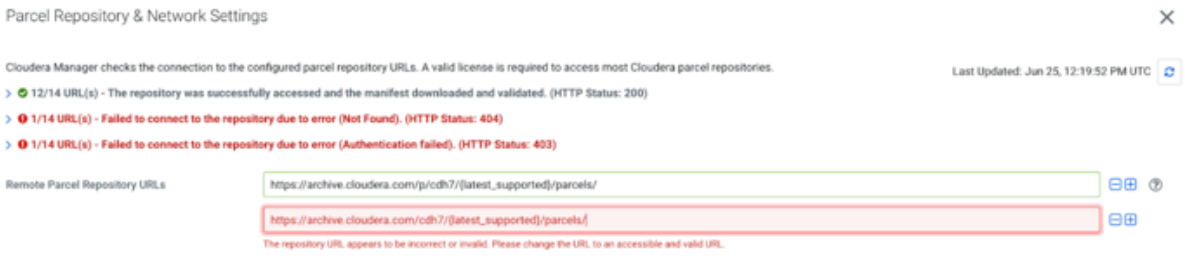
```
less am2cm-2.3.0.0-60/cm_migration.log
```

Configuring parcel

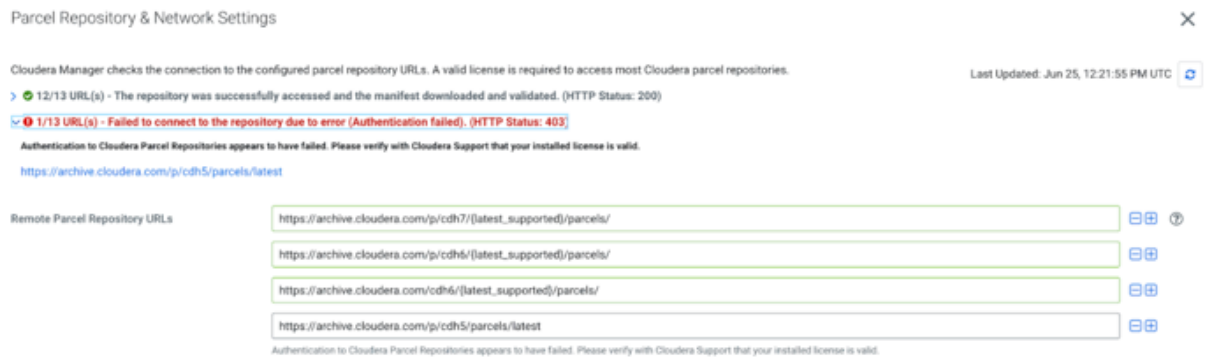
You need to check parcel repositories and network settings, and remove parcels that you do not need.

Procedure

1. Click Parcels in the left navigation pane of the Cloudera Manager UI.
2. Click Parcel Repositories and Network Settings and check for parcels with errors.



3. Remove parcels that you do not need.



4. Click Save and Verify Configuration.

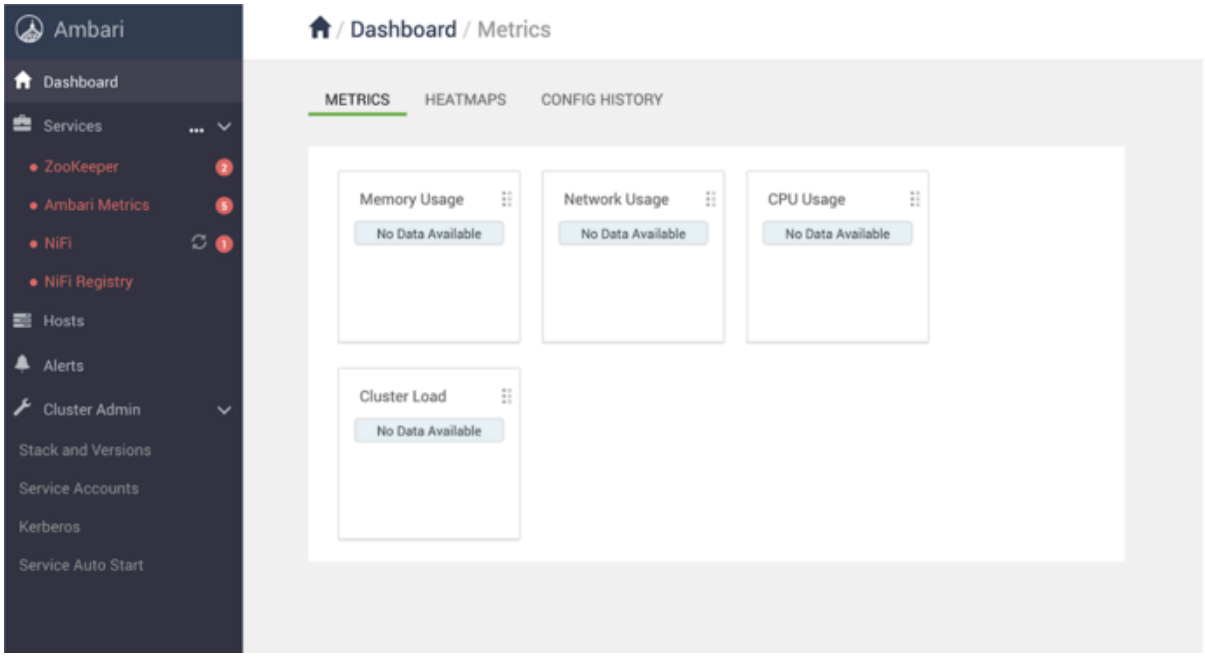
Deploying Cloudera Manager

You need to deploy Cloudera Manager for using it. To deploy Cloudera Manager, you need to stop all HDP services, download the CFM custom service descriptor files, deploy the existing cluster on CDP, and refresh Cloudera Manager.

Procedure

1. Stop all HDP services from Ambari.

   The following image shows that all HDF services are stopped.



2. Download the CFM Custom Service Descriptor files:

```
cd /opt/cloudera/csd
wget https://<username>:<password>@archive.cloudera.com/p/cfm2/2.1.2/redhat7/yum/tars/parcel/NIFI-<version>-<build>.jar
wget https://<username>:<password>@archive.cloudera.com/p/cfm2/2.1.2/redhat7/yum/tars/parcel/NIFIREGISTRY-<version>-<build>.jar
chown cloudera-scm:cloudera-scm ./*
chmod 644 ./*
```

```
systemctl restart cloudera-scm-server
```

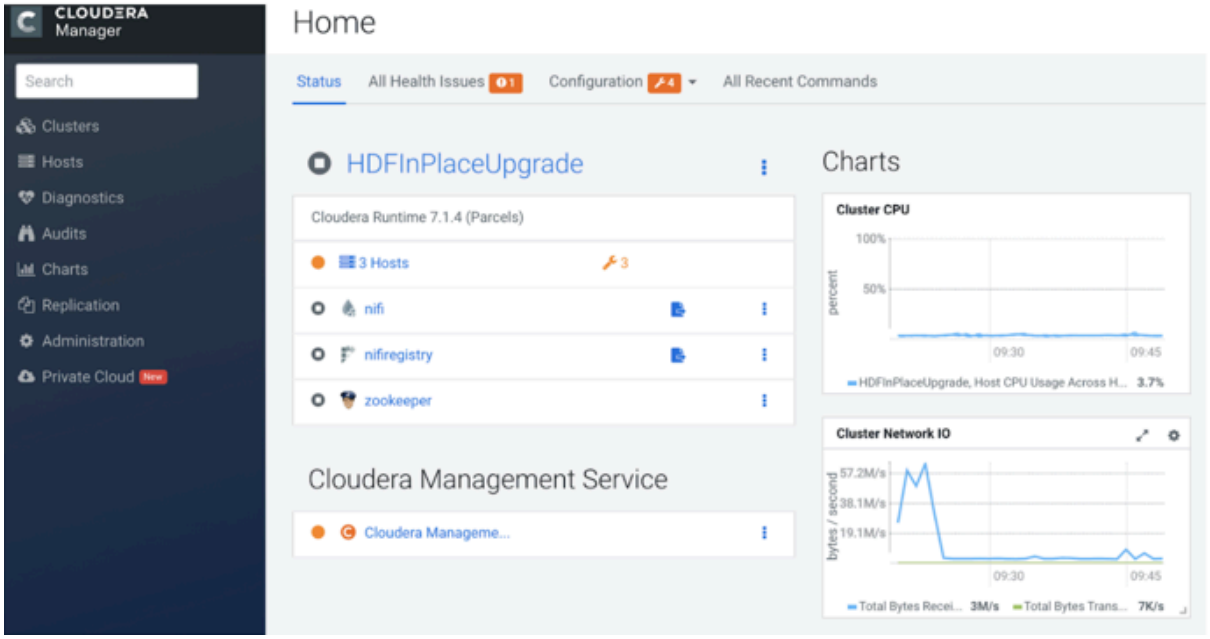
3. Deploy the existing cluster on CDP, using the Cloudera Manager deployment template:

```
# cd am2cm-2.3.0.0-60/conf
[root@ccycloud-1 conf]# curl --user admin:admin -k -X PUT -H "Content-Type
: application/json" -d @cm_deployment_template.json 'http://ccycloud-1.a
m2cmhdf.root.hwx.site:7180/api/v41/cm/deployment?deleteCurrentDeployment
=false'
{
  "message" : "\"Role name 'nifiregistry-NIFI_REGISTRY_SERVER-108d5ac54728
29ddbaa38dfb3fd8ad' is not compliant. Use 'nifi0cb063fc-NIFI_REGISTRY_SE
RVER-108d5ac5472829ddbaa38dfb3fd8ad', or do not use a name of the format
 <service name>-<roletype>-<arbitrary value>.\""
}[root@ccycloud-1 conf]#vi cm_deployment_template.json

Update nifiregistry-NIFI_REGISTRY_SERVER-108d5ac5472829ddbaa38dfb3fd8ad to
 nifiregistry-NIFI_REGISTRY_SERVER

# curl --user admin:admin -k -X PUT -H "Content-Type: application/json" -
d @cm_deployment_template.json 'http://ccycloud-1.am2cmhdf.root.hwx.site:
7180/api/v41/cm/deployment?deleteCurrentDeployment=false'
```

Note: You must not forget to edit the cm_deployment_template.json file and rename nifiregistry-NIFI_REGISTRY_SERVER-* to nifiregistry-NIFI_REGISTRY_SERVER.

4. Refresh the Cloudera Manager browser.

   The following image shows that the cluster is running with components including NiFi, NiFi Registry, and Zookeeper.



Adding CFM parcel in Cloudera Manager

You need to add the CFM parcel after you deploy Cloudera Manager. To add the CFM parcel, you need to download the CSD files for the CFM parcel and add an additional parcel URL for CFM in Cloudera Manager.

Procedure

1. Go to the Cloudera Manager UI and click Parcels in the left navigation pane.
2. Click Parcel Repositories and Network Settings.
3. Extend the Remote Parcel Repository URLs part, with the following additional parcel URL:

```
https://archive.cloudera.com/p/cfm2/2.1.2/redhat7/yum/tars/parcel/
```

4. Click Save and Verify Configuration.

   The parcel URL depends on the operating system. To choose the appropriate URL, see *Download from the CFM Repository*.

Related Information

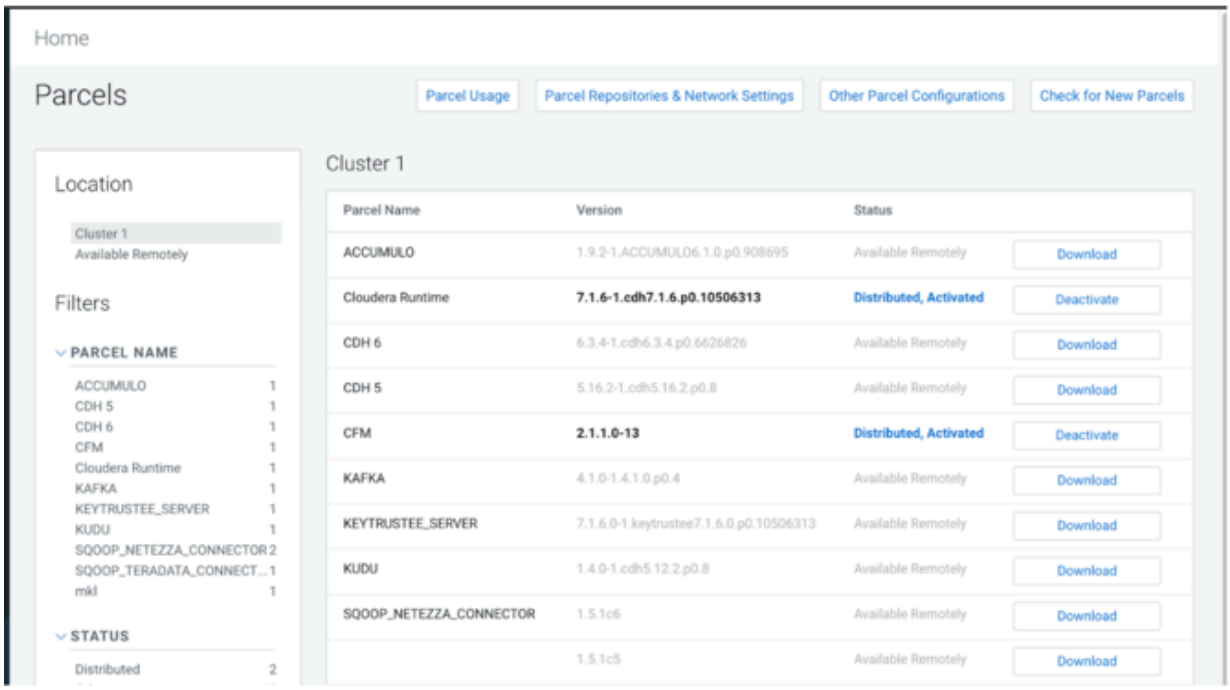
Download from the CFM Repository

Activating parcel

After you add the CFM parcel, you need to activate both the CFM and the Cloudera Runtime parcels.

Procedure

1. Click Parcels in the left navigation pane of the Cloudera Manager UI.
2. Find out the parcel corresponding to the required version of Cloudera Runtime and CFM.



3. Click Download, Distribute, and Activate.

   Only the valid and available button appears for a parcel and the order is download, distribute, and then activate.

Troubleshooting HDF upgrade

Learn about issues that might occur during the upgrade process, and how to resolve them.

Issue: Host health error

The host health shows the following error:

```
Clock Offset Suppress...The host's NTP service could not be loca
ted or did not respond to a request for the clock offset.
```

Solution:

Enable and start the NTP service for all hosts:

```
yum install -y ntp
systemctl start ntpd.service
systemctl enable ntpd.service
systemctl status ntpd.service
```

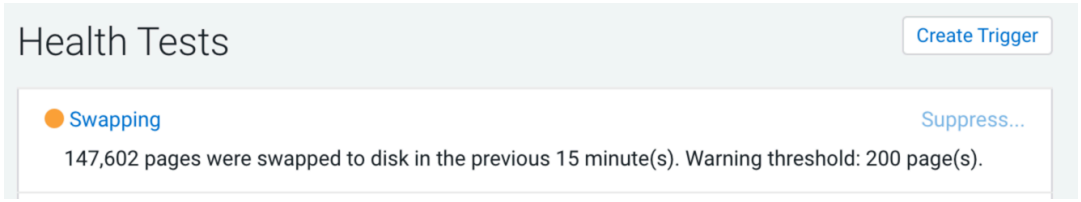
Otherwise perform the following steps:

1. Click on the hosts health error, and navigate to the Host Clock Offset Threshold configuration.
2. Configure the values for thresholds to Never.

3. Click Save Changes.

Issue: Swappiness

The host check process checks how many pages were swapped in a given time, and an error might appear if that number is not reached, as shown in the following image:



Solution:

Set swappiness to one for all hosts.

```
cat /proc/sys/vm/swappiness
echo 1 > /proc/sys/vm/swappiness
cat /proc/sys/vm/swappiness
```

It suppresses the swappiness alerts for the cluster. Additionally, you can perform the following checks:

- Check configuration warnings for each service.
- Review JVM parameters and configuration for all services (some services are not transitioned).
- Review the Log4J configuration such as logs dir, size, and backup index.

Issue: Cloudera agent security

You might experience an issue with Cloudera agent security and Cloudera agents might not connect to the server.

Solution:

You can perform the following checks:

- Fix ownership of /var/lib/cloudera-scm-agent/agent-cert

- chown cloudera-scm:cloudera-scm /var/lib/cloudera-scm-agent/agent-cert
- chmod 755 /var/lib/cloudera-scm-agent/agent-cert

Post-upgrade steps on CDP

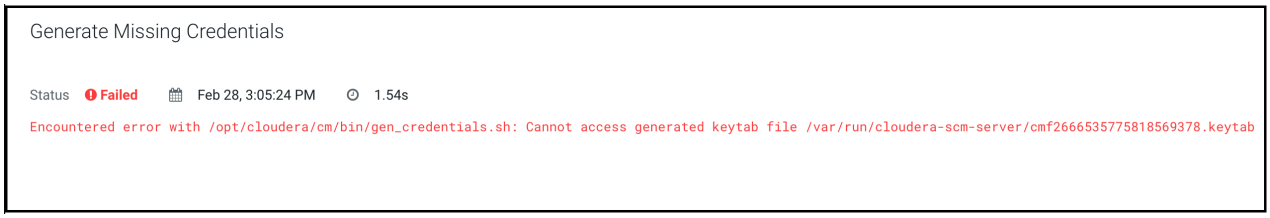
After you complete the upgrade of HDF, you need to verify or modify properties for NiFi Registry, NiFi, Ranger, Solr, and Zookeeper, and enable security if you want to use a secure cluster. You also need to set core configuration service.

Enable security

If your previous cluster was secure or if you want to use a secure cluster, you have to enable Kerberos and auto-TLS.

For information about how to enable Kerberos and auto-TLS, see *Encrypting Data in Transit* and *Security Kerberos Authentication Overview*.

After setting up Kerberos, go to Administration Security and click Generate missing credentials on the Kerberos Credentials tab. You might get the following error depending on your Kerberos server settings:



If you get this error, it means that the principals generated by Ambari have a different maximum renewable ticket time what Cloudera Manager wants to use, which causes this error. To fix this you have to modify the principals created by Ambari to have the same maximum renewable ticket time what Cloudera Manager wants to use (5 days):

```
# Get a keytab where the user have right to modify principals
# kadmin -q "ktadd -k /tmp/admin.keytab -norandkey admin/admin@HDF.COM" -p
 admin/admin@HDF.COM
# Get principals generated by ambari via ambari rest api call
principals=($(curl -H "Content-Type: text/csv" "${ambariprotocol}://${amb
ariuser}:${ambaripwd}@${ambariserver}:${ambariport}/api/v1/clusters/${cluste
rname}/kerberos_identities?format=csv" | tail -n +2 | awk -F , '{ print $3}'
))
# Modify principal maxrenewlife to 5 day
for princ in "${principals[@]}"
do
   kadmin -k -t /tmp/admin.keytab -p admin/admin@HDF.COM -q "modprinc -ma
xrenewlife 432000 $princ"
done
#Delete keytab for security reasons
#rm -f /tmp/admin.keytab
```

Related Information

Encrypting Data in Transit

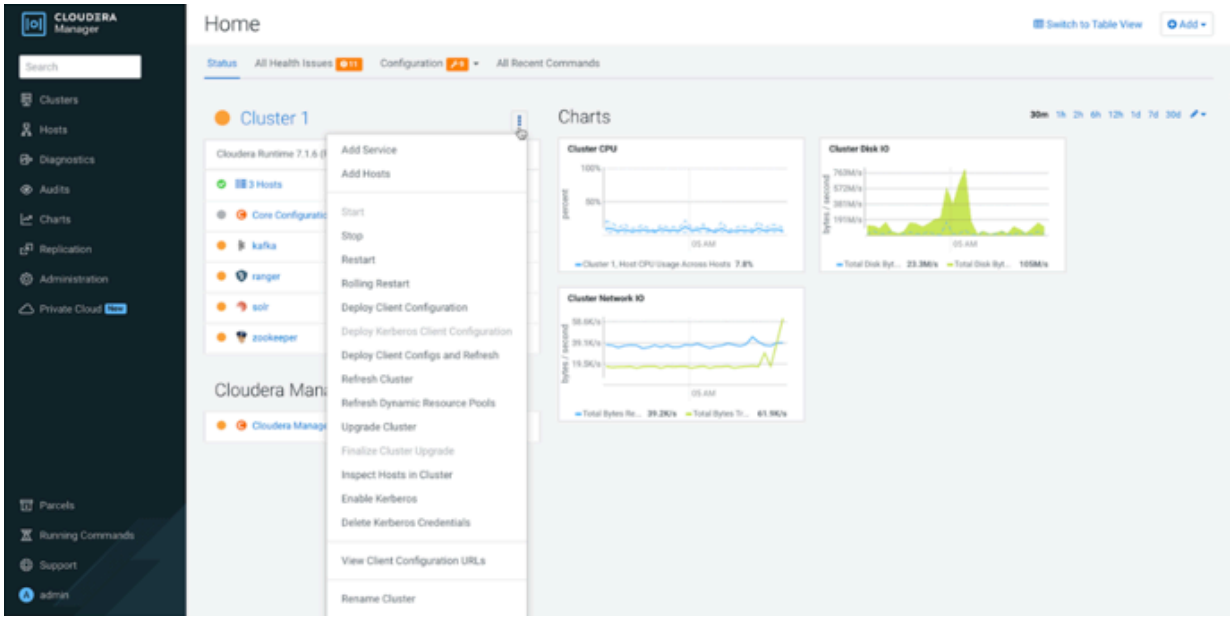
Security Kerberos Authentication Overview

Setting core configuration service

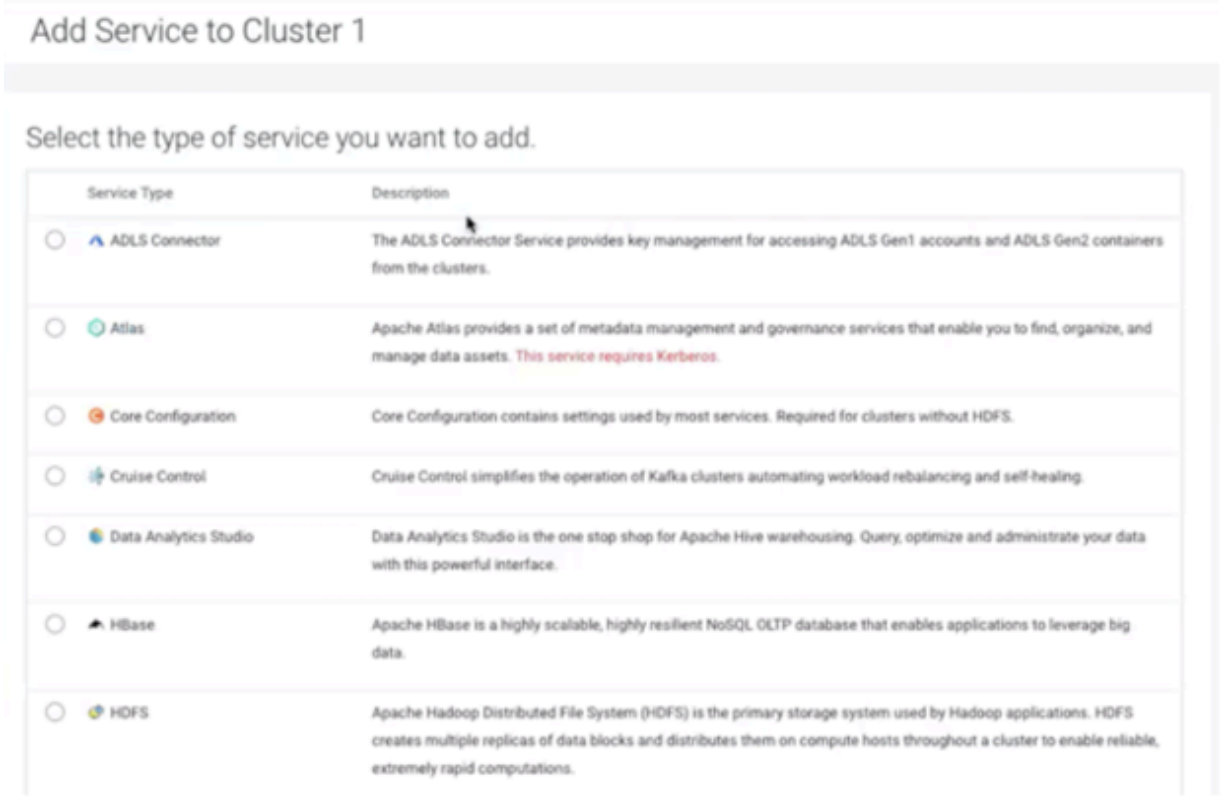
You need to add core configuration service to your cluster manually. The core configuration service allows you to create clusters without the HDFS service.

Procedure

1. Click Add Service in the Cloudera Manager UI.



The Add Service to Cluster window appears.



2. Select Core Configuration and click Continue.
3. Verify the assigned roles and click Continue.
4. Review the changes and click Continue.

   The commands run to add core configuration settings.
5. Click Continue after all commands execute successfully.

6. Check the summary and click Finish.

Starting Zookeeper service

You must configure the Zookeeper server hosts to start the Zookeeper service.

Procedure

1. Add zookeeper.snapshot.trust.empty=true to your server configuration file.

   This can be set in the zoo.cfg advanced configuration snippet in the Cloudera Manager UI.
2. Start the server.

   Cloudera recommends removing the zookeeper.snapshot.trust.empty property after you have a working server.
3. Remove or move the myid file from the Zookeeper server hosts.

   The path is ${dataDir}/myid. For example, # mv /hadoop/zookeeper/myid hadoop/zookeeper/myid.bak.

   It is possible that the myid file is in a different path. To determine the exact path, check Zookeeper service data directory configuration value in the Cloudera Manager UI.
4. Start the Zookeeper service from Cloudera Manager.

Configuring NiFi Registry settings

Before you first start NiFi Registry, you need to set the database password for NiFi Registry in Cloudera Manager and you need to migrate the NiFi Registry directories. If you have Kerberos enabled and Ranger installed, you also need to configure those for NiFi Registry before the first start.

Setting database password for NiFi Registry

Before the first starting of NiFi Registry, you need to set the NiFi Registry database password in Cloudera Manager. This password was collected from the Ambari-managed cluster earlier in the in-place upgrade process.

Before you begin

You have collected the NiFi Registry database password.

Procedure

1. Go to Cloudera Manager Clusters .
2. Select NiFi Registry.
3. Go to the Configuration tab.
4. Search for the NiFi Registry Database Password configuration and specify the password that you have earlier collected.

Related Information

Collect Nifi Registry database password

Configuring Kerberos for NiFi Registry

After you enable Kerberos, as described in *Enabling security*, you have to enable Kerberos for NiFi Registry and configure the initial admin identity setting, if the initial admin identity setting was configured before migration.

Before you begin

You have enabled Kerberos as described in *Enabling security*.

Procedure

1. Go to Cloudera Manager Clusters .
2. Select NiFiRegistry.
3. Go to the Configuration tab.
4. Search for the Enable Kerberos Authentication configuration value and enable it.
5. Optional. Search for the Initial Admin Identity configuration and specify the correct principal name.

Related Information

Enable security

Configuring Ranger for NiFi Registry

If your cluster contains Ranger, then you need to configure Ranger service before starting NiFi Registry.

Procedure

1. Go to Cloudera Manager Clusters .
2. Select NiFiRegistry.
3. Go to the Configuration tab.
4. Search for the RANGER Service configuration and enable it.
5. Modify the ranger.plugin.nifi-registry.service.name property to match with the new ranger service name.

Modifying the service name in Ranger

If your NiFi Registry service name in Ranger contains the - character, then you must change it to the _ character.

Procedure

1. Go to Ranger Admin Web UI.
2. Go to resource based policies.
3. Click the edit button and modify the necessary characters in the service name and display name.
4. Click Save.

Migrating NiFi Registry directories

You must create a working directory to start the NiFi Registry service. The directory is the place where NiFi Registry stores existing buckets, configuration files, database files, and so on.

Before you begin

- You have set the NiFi Registry database password in Cloudera Manager.
- You have configured Ranger for NiFi Registry if your cluster contains Ranger.

Procedure

1. Create a working directory for NiFi Registry and copy the content from the old directory:

```
mkdir /var/lib/nifiregistry
cp  -R /var/lib/nifi-registry/* /var/lib/nifiregistry
chmod 755 /var/lib/nifiregistry
chown -R nifiregistry:nifiregistry /var/lib/nifiregistry
```

2. Start the NiFi Registry service.

Related Information

Setting database password for NiFi Registry

Configuring Ranger for NiFi Registry

Verifying Ranger configurations

You need to verify Ranger configurations before you start using the Ranger service.

Procedure

1. Go to Cloudera Manager Clusters .
2. Select Ranger.
3. Click Configuration.
4. Ensure that the following configurations have actual database host, user and password used by Ranger:
   - Ranger Database Host (ranger_database_host)
   - Ranger Database User (ranger_database_user)
   - Ranger Database User Password (ranger_database_password)
5. Ensure that the load_balancer_url configuration point to the proper hostname and contains the correct schema and port number.

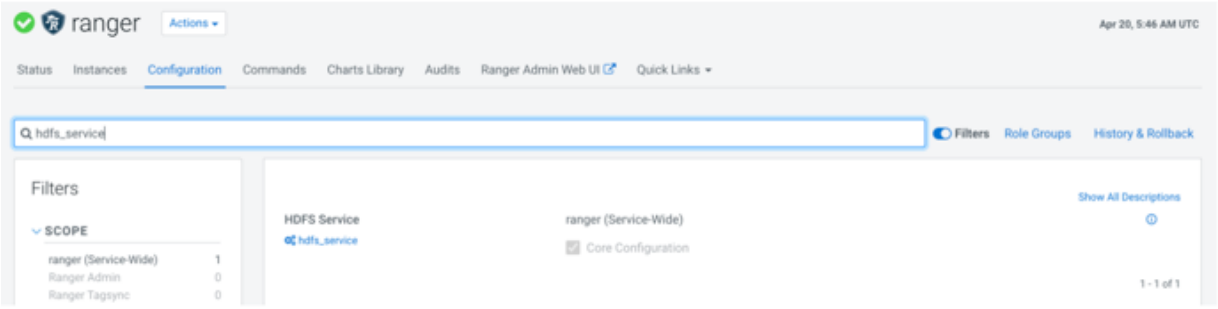
   Otherwise, services will not be able to communicate with Ranger.

Configuring Ranger settings

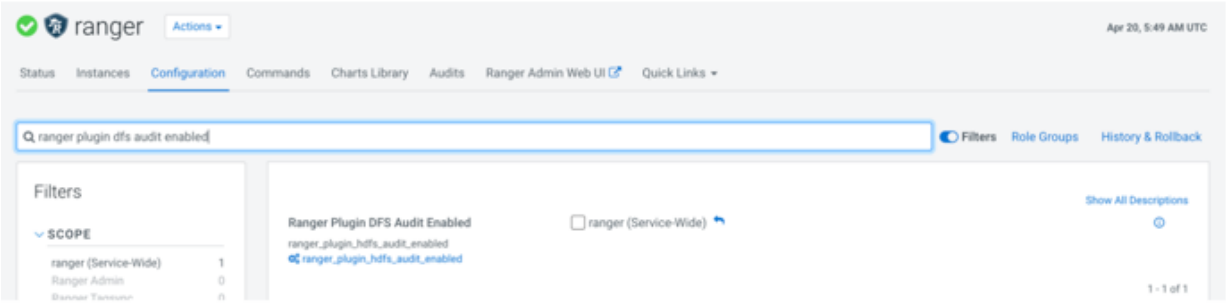
You need to configure Ranger settings and execute Ranger actions.

Procedure

1. Click ranger in the Cloudera Manager UI.
2. Go to the Configuration tab.
3. Locate the HDFS Service (hdfs_service) property and select the Core Configuration option.



4. Click Save Changes.
5. Locate the Ranger Plugin HDFS Audit Enabled (ranger_plugin_hdfs_audit_enabled) property and set to false.



6. Locate and configure the following properties:
   - Ranger Admin User Initial Password

     rangeradmin_user_password=<yourpassword>
   - Ranger Usersync User Initial Password

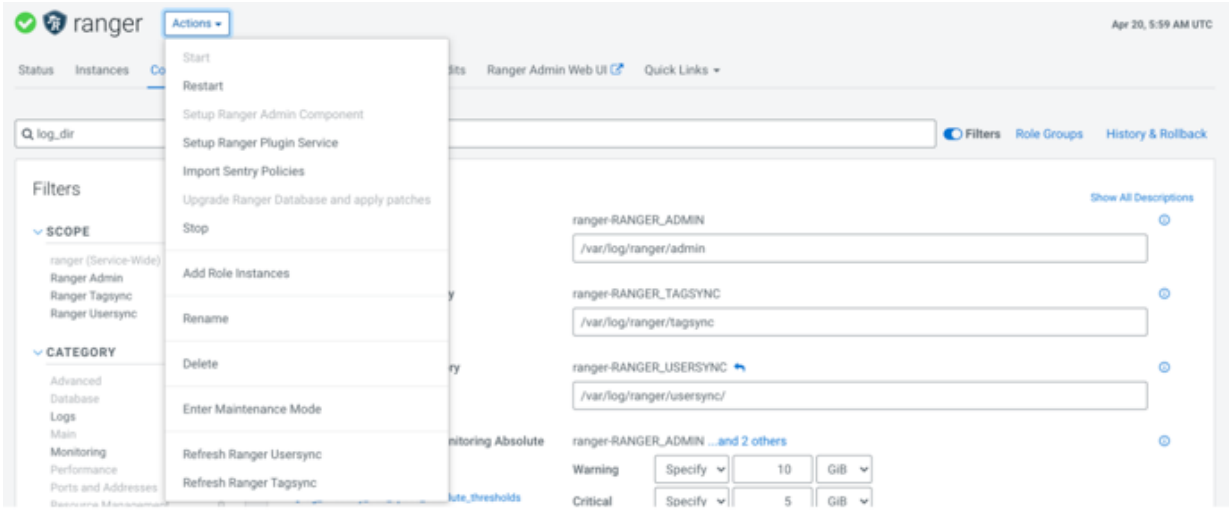
     rangerusersync_user_password=<yourpassword>
   - Ranger Tagsync User Initial Password

     rangertagsync_user_password=<yourpassword>
   - Ranger KMS Keyadmin User Initial Password

     keyadmin_user_password=<yourpassword>
   - Ranger Database User Password

     ranger_database_password=<yourpassword>
   - Ranger Usersync Log Directory

     log_dir=/var/log/ranger/usersync/
7. Click Actions Upgrade Ranger Database and apply patches .
8. Click Actions Setup Ranger Admin Component .



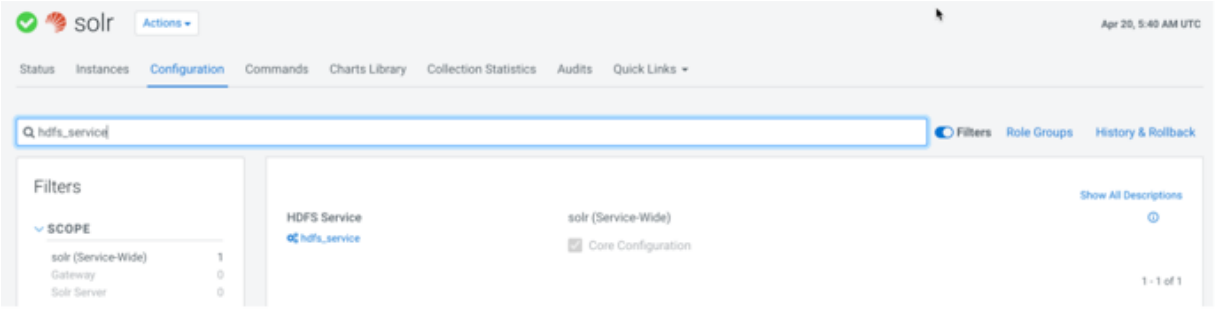
Configuring Solr settings

You need to configure Solr settings before you start using the Solr service. You also need to start Zookeeper service.

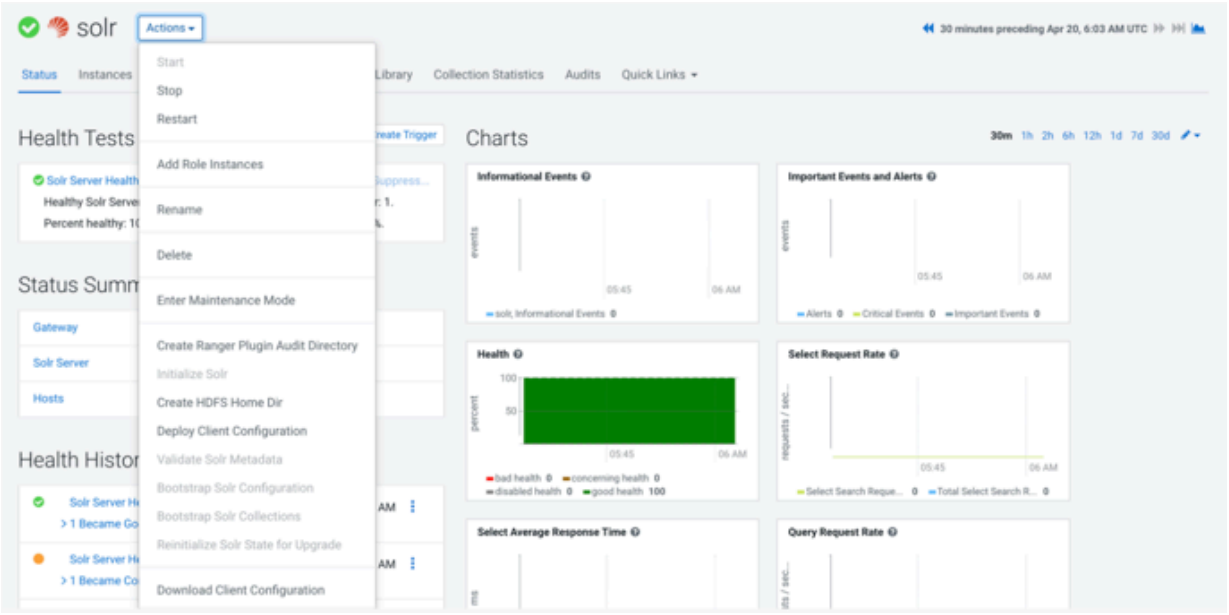
Procedure

1. Click solr in the Cloudera Manager UI.
2. Go to the Configuration tab.

3. Locate the HDFS Service (hdfs_service) property and select the Core Configuration option.



4. Click Save Changes.
5. Click Actions Initialize Solr .



6. Go to the initial Cloudera Manager UI by clicking the Cloudera Manager icon at the top-left corner of the screen.
7. Click zookeeper.
8. Click Actions Start in the Zookeeper cluster window.

   The Start dialog box appears.
9. Confirm start operation by clicking Start again.

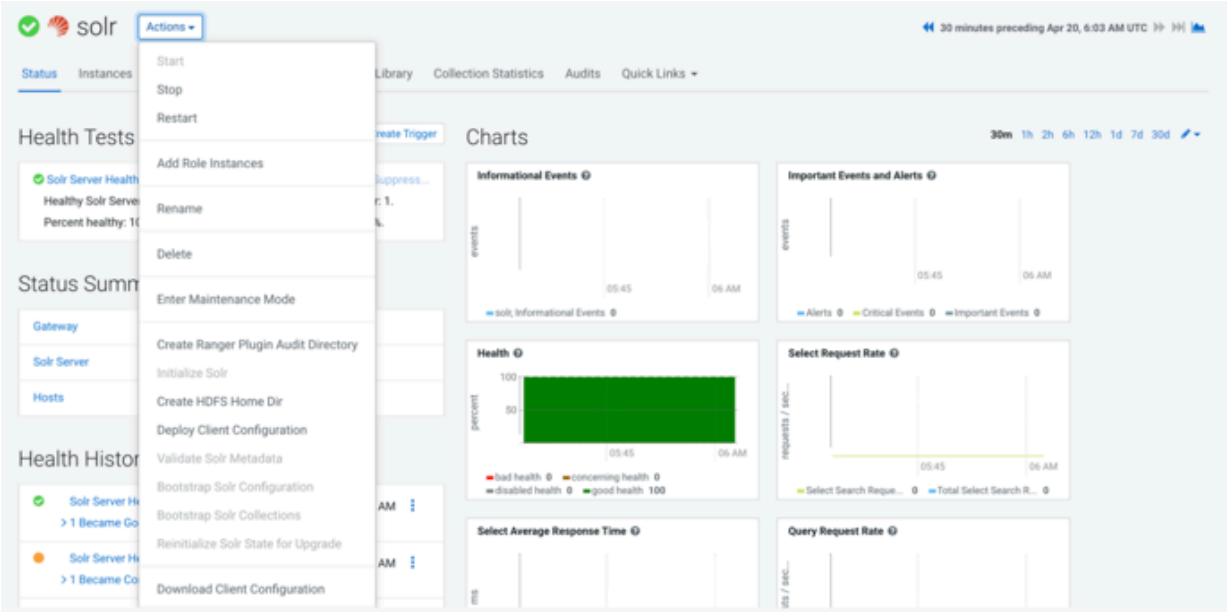
Initializing Solr

You need to execute Solr actions before you start using the Solr service to ensure that Solr is initialized correctly.

Procedure

1. Go to Cloudera Manager Clusters .
2. Select Solr, and click Actions.
3. Execute the Initialize Solr action.

4. Execute the Create HDFS Home Dir action.



Configuring NiFi settings

You need to configure NiFi settings after you start NiFi successfully.

Procedure

1. Go to Cloudera Manager Clusters .
2. Select NiFi.
3. Click NiFi Web UI.
4. Log in to NiFi.
5. On the NiFi UI, click Controller Settings.

   The NiFi Settings screen appears.
6. Go to the Registry Clients tab.
7. Recheck or configure NiFi Registry URL on the NiFi UI to point to the correct hostname and port number.
8. Go to the Reporting Tasks tab.
9. Remove the AmbariReportingTask setting.

Configuring Kerberos for NiFi

After you enable Kerberos, as described in *Enable security*, you have to enable Kerberos for NiFi and configure the initial admin identity setting, if the initial admin identity setting was configured before migration.

Procedure

1. Go to Cloudera Manager Clusters .
2. Select NiFi.
3. Go to the Configuration tab.
4. Search for the Enable Kerberos Authentication configuration value and enable it.
5. Optional. Search for the Initial Admin Identity configuration and specify the correct principal name.

Configuring Ranger for NiFi

If your cluster contains Ranger, then you need to configure Ranger service, before starting NiFi.

Procedure

1. Go to Cloudera Manager Clusters .
2. Select NiFi.
3. Go to the Configuration tab.
4. Search for the RANGER Service configuration and enable it.
5. Modify the ranger.plugin.nifi.service.name property to match with the new ranger service name.

Modifying the service name in Ranger

If your NiFi service name in Ranger contains the - character, then you must change it to the _ character.

Procedure

1. Go to Ranger Admin Web UI.
2. Go to resource based policies.
3. Click the edit button and modify the necessary characters in the service name and display name.
4. Click Save.

Migrating LDAP authentication configuration

If your NiFi used LDAP authentication in HDF cluster, you need to migrate the settings manually.

Procedure

1. Collect all necessary configuration for LDAP login provider.

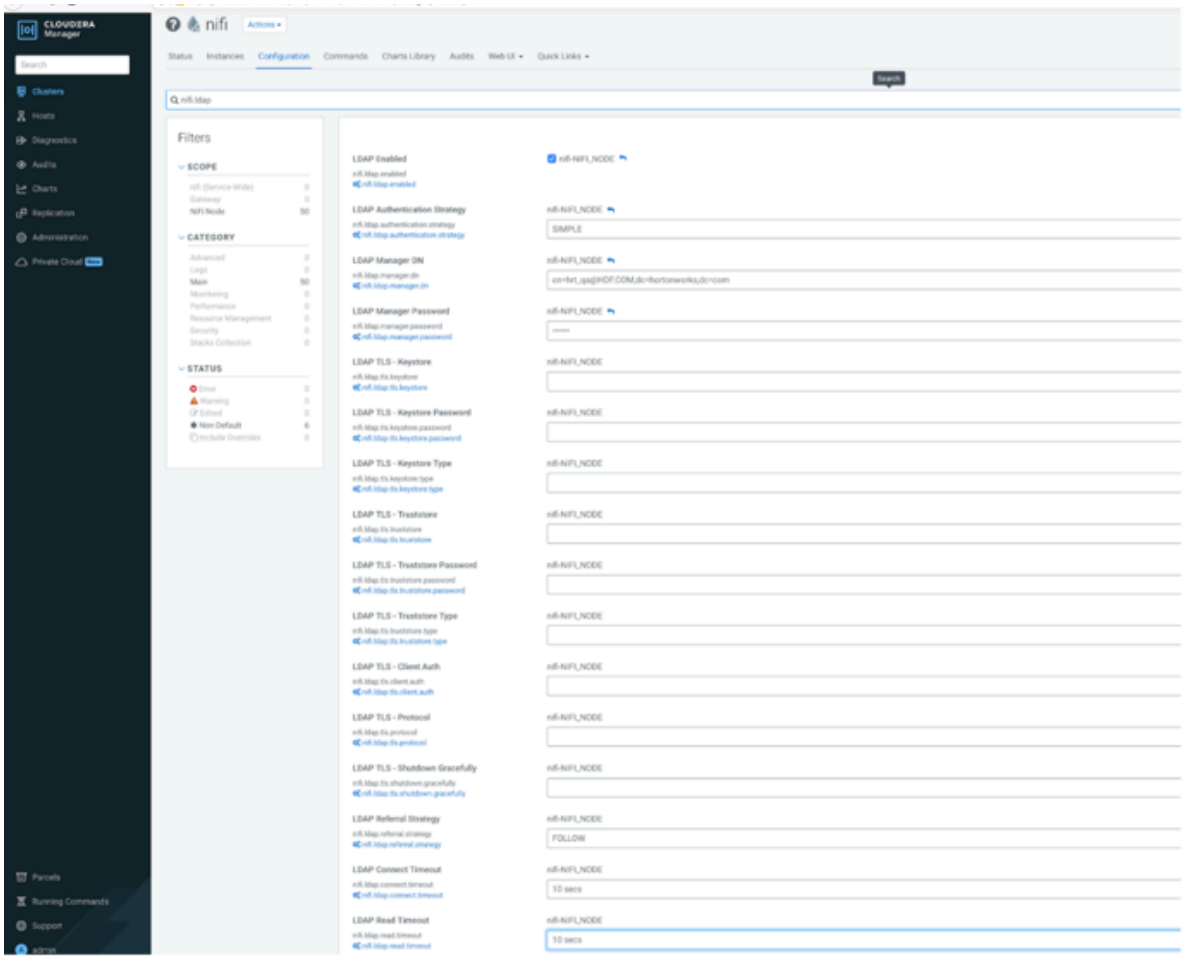
   For that, you can check the old cluster configuration file or check the configuration in the Ambari UI:

   ```
   cat /usr/hdf/current/nifi/conf/login-identity-providers.xml
   ```

   Note: The passwords are encrypted in the XML file and cannot be fetched.

2. Configure the NiFi-LDAP properties in the Cloudera Manager UI:



3. Set the nifi.security.user.login.identity.provider configuration value to ldap-provider.
4. Set the nifi.ldap.enabled configuration value to true.
5. Configure the value of the nifi.initial.admin.identity property.
6. Remove the new cluster NiFi users.xml and authorizations.xml files for NiFi to generate these XML files with proper values.

   The default path for these files is /var/lib/nifi/users.xml and /var/lib/nifi/authorizations.xml.

Migrating file-based user handling and policies

If you use NiFi built-in file-based policy and user handling, then you have to migrate the content of the users.xml and authorizations.xml files. This should be done after LDAP migration, if NiFi used LDAP.

Procedure

1. Copy the user or group entries from /var/lib/nifi/conf/users.xml and add the entries into /var/lib/users.xml for every NiFi instance.
2. Copy the policy entries from /var/lib/nifi/conf/authorizations.xml and add the entries into /var/lib/authorizations.xml for every NiFi instance.

   If you have only a few user entries and policy configurations, it is quicker to re-apply them through the NiFi UI, instead of synchronizing the old and the new users and authorization XML files.

Post-upgrade steps on CDP for HDF on HDP

After you complete the upgrade of HDF on HDP, you need to verify or modify YARN properties.

Configuring YARN settings

You need to configure YARN settings before you start using the YARN service.

Procedure

1. Go to Cloudera Manager Clusters .
2. Select Yarn.
3. Click Configuration.
4. Set value for the ApplicationMaster Maximum Attempts configuration property.

   The value should be the same as the Maximum Number of Attempts for MapReduce Jobs configuration property value.
5. Remove the spark2_shuffle property from yarn.nodemanager.aux-services.

Kafka in-place migration with Ranger

After the upgrade of HDF to CFM on CDP, you must configure Kafka centric clusters that use Ranger. You need to set Kafka Ranger policies.

Migrating Kafka Ranger policies

You need to ensure that the Ranger service name property has identical value for both Kafka and Ranger.

Before you begin

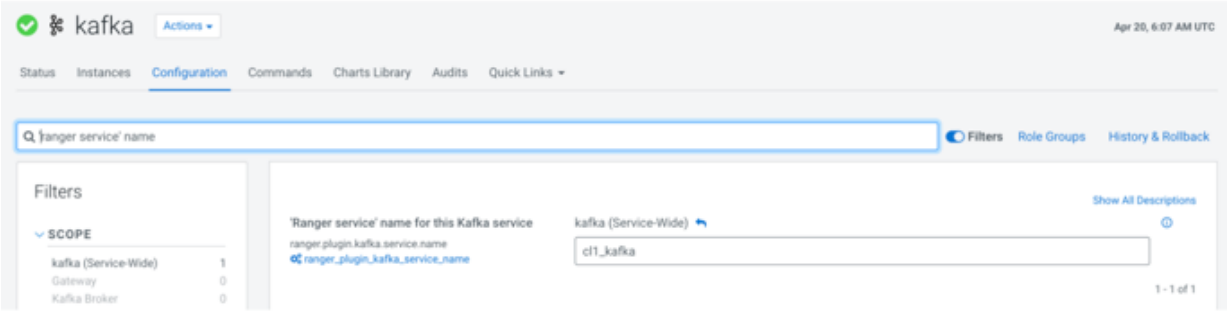
You have set core configuration service, and configured Ranger and SOLR settings.

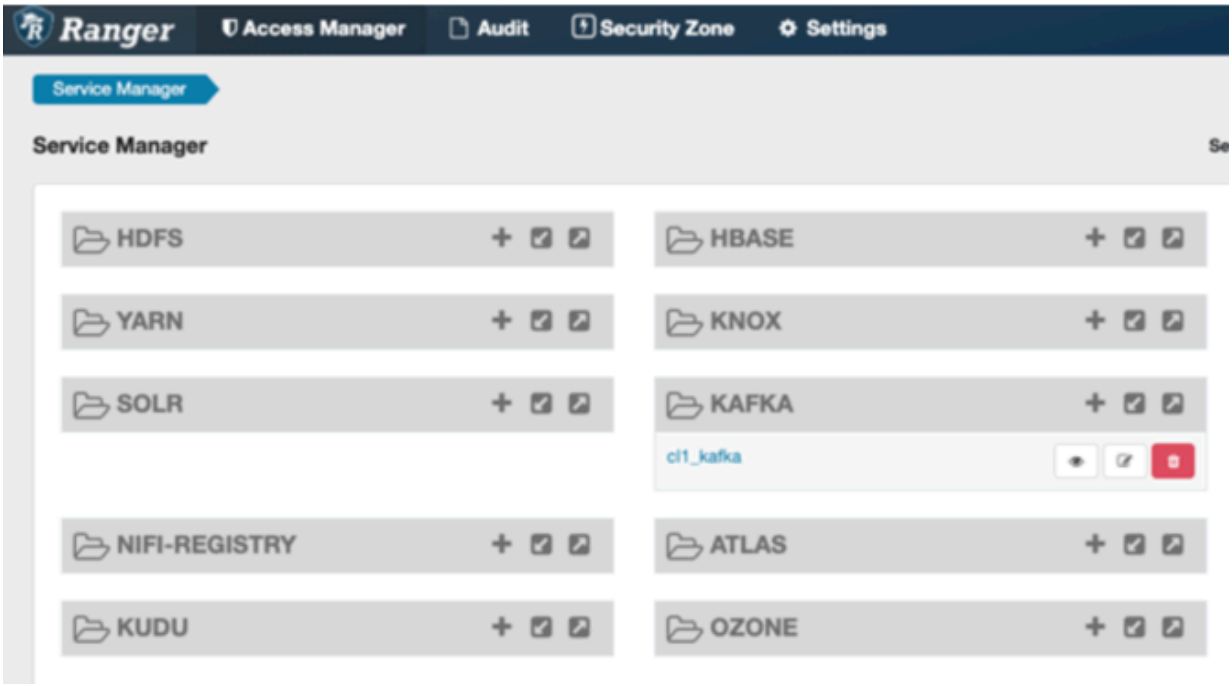
Procedure

1. Click kafka in the Cloudera Manager UI.
2. Go to the Configuration tab.
3. Locate the ‘Ranger service’ name for this Kafka service property.

4. Ensure that the ‘Ranger service’ name for this Kafka service (ranger.plugins.kafka.service.name) kafka configuration matches the Ranger service name in the Ranger web UI.

   For example, in the following images, the name of the Ranger service in both Kafka and Ranger is cl1_kafka.

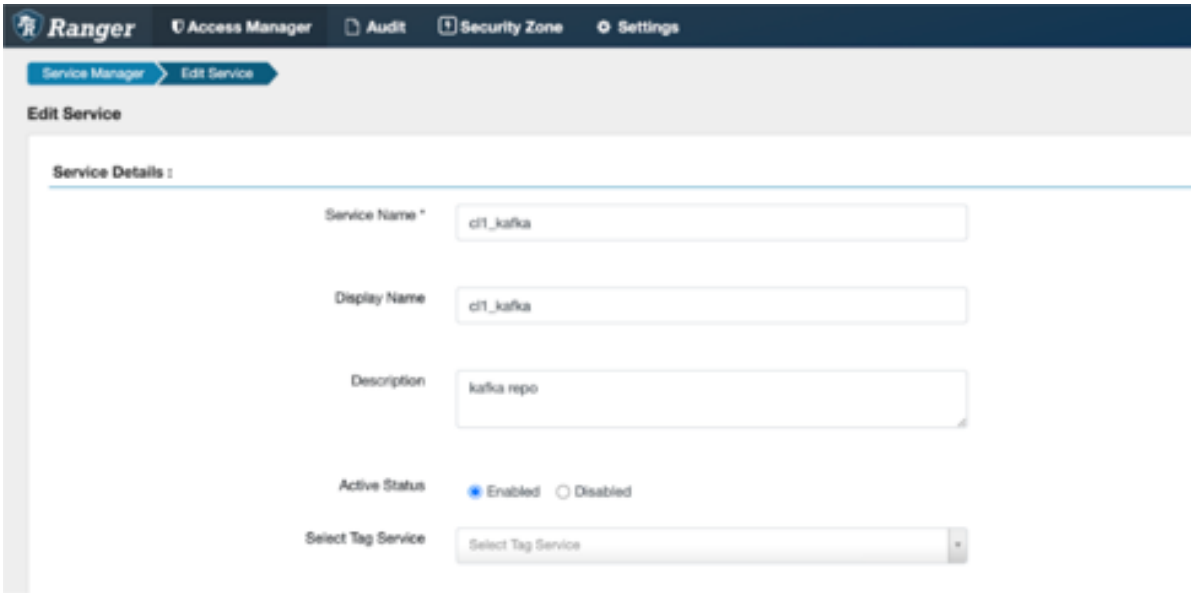




   To check the property name in Ranger:

   a. Click ranger in the Cloudera Manager UI.
   b. Click Ranger Admin Web UI.

      The Ranger UI opens in another window.
   c. Click KAFKA service and check the Service Name property.
   d. If the service name is different, then set it to the same value as configured in the Kafka cluster.


Ranger
Access Manager
Audit
Security Zone
Settings
Service Manager
Edit Service
Edit Service
Service Details :
Service Name *
cl1_kafka
Display Name
cl1_kafka
Description
kafka repo
Active Status
Enabled
Disabled
Select Tag Service
Select Tag Service