Cloudera Flow Management 2.1.1

HDF to CFM Migration Guide

Date published: 2019-06-26
Date modified: 2021-04-28

CLOUDERA

https://docs.cloudera.com/

Legal Notice

Contents

Before you begin

This document describes the general steps required to move your NiFi dataflow and NiFi Registry versioned flows from an HDF 3.5.x or CFM 1.1.0 Standalone cluster to a CFM 2.1.1 cluster in CDP Private Cloud Base.

Migration and upgrade scenarios

You have two options for moving to CFM 2.1.1.

- Upgrade refers to a full in-place upgrade of CFM to CDP Private Cloud Base. For upgrade information see the *Upgrade Guide*.
- Data migration refers to moving existing HDF cluster workloads to a new installation of CDP Private Cloud Base.

Migration paths

You must migrate from one of the following environments:

- HDF 3.5.x
  - Standalone (Not managed by Ambari)
  - Ambari (with or without Ranger)
  - +HDP 3.1.5 and Ambari (with or without Ranger)
- CFM 1.1.0
  - Standalone (Not managed by Cloudera Manager)

Additional requirements

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.

- You must migrate HDF 3.5.x or CFM 1.1.0 Standalone to an equivalent target environment. For example:
  - From HDF 3.5.x without Ranger to CFM 2.1.1 without Ranger
  - From HDF 3.5.x with Ranger to CFM 2.1.1 with Ranger
  - From CFM 1.1.x without Ranger to CFM 2.1.1 without Ranger
- You have deployed a CDP Private Cloud Base cluster with the necessary services running (ZooKeeper, Ranger, and similar) and out-of-the-box configurations.
- You have sufficient networking connectivity and capacity between the source cluster and destination cluster.
- You have the permissions required to access and modify files on cluster nodes.
- You have consulted the *Cloudera Flow Management Release Notes* and are aware of known issues.
- You have reviewed *Apache NiFi Migration Guidance* and *Apache NiFi Registry Migration Guidance* and are aware of any differences between source and destination components.

Related Information

Cloudera Flow Management Release Notes
Upgrade Guide
Apache NiFi Registry Migration Guidance
Apache NiFi Migration Guidance
CFM component versions
HDF Component Support
Migrating workloads

Preserve source cluster files and directories

Learn how to locate and backup the NiFi and NiFi Registry files on the source cluster. You will need these files later in the migration process. It is also important to preserve configuration files so that you can retain any cluster customizations that you implemented on your source cluster.

NiFi files to preserve

Lists the NiFi files and directories to preserve along with their default locations.

Items to preserve	Default location
The following configuration files: • authorizers.xml • bootstrap.conf • bootstrap-notification-services.xml • login-identity-providers.xml • nifi.properties • state-management.xml	• HDF – /usr/hdf/current/nifi/conf • HDF or CFM standalone installations – <nifi-installation-directory>/conf
If NiFi is secured: • authorizations.xml • users.xml	• HDF – /var/lib/nifi/conf • HDF or CFM standalone installations – <nifi-installation-directory>/conf
The following directories: • database_repository • provenance_repository • local state directory	• HDF – /var/lib/nifi • HDF or CFM standalone installations – <nifi-installation-directory>/
flow.xml.gz	• HDF – /var/lib/nifi/conf • HDF or CFM standalone installations – <nifi-installation-directory>/conf

Note:

It is a best practice to create NiFi content, database, flowfile and provenance repositories on separate physical disks and reference them using the directory path properties in nifi.properties. This best practice facilitates the migration process.

Preserve custom processors/NARs

Describes the steps to preserve custom processors/NARs.

If you have created any custom NARs, a best practice is to store and reference them in a centralized location.

On each source node:

1. Create a second library directory called custom_lib. For example: /opt/configuration_resources/custom_lib.
2. Move your custom NARs to this new directory.
3. Add a new line to the nifi.properties file to specify this new lib directory. For example:

```
nifi.nar.library.directory.custom=/opt/configuration_resources/custom_lib
```

You will perform a similar procedure on the destination cluster nodes later in the migration process.

Note:

For HDF NARs, you may need to rebuild them to correctly depend on the CFM NARs, instead of HDF.

NiFi Registry files to preserve

Lists the NiFi Registry files and directories to preserve and their default locations on the source cluster.

Items to preserve	Default location
The following configuration files: • authorizers.xml • bootstrap.conf • identity-providers.xml • logback.xml • nifi-registry.properties • providers.xml	• HDF – /usr/hdf/current/nifi-registry/conf • HDF or CFM standalone installations – <registry-installation-directory>/conf
If NiFi Registry is secured: • authorizations.xml • users.xml	• HDF – /var/lib/nifi-registry/conf • HDF or CFM standalone installations – <registry-installation-directory>/conf
The following directories: • database The nifi.registry.db.url property in nifi-registry.properties points to the location of the NiFi Registry metadata database. The default database is H2. • flow_storage	• HDF – /var/lib/nifi-registry • HDF or CFM standalone installations – <registry-installation-directory>/

Install CFM 2.1.1

Provides information about installing CFM 2.1.1.

- You have installed a CDP Private Cloud Base cluster and prepared it for the CFM deployment. For more information, see the *Deployment Guide.*
- You have equivalence between source and target clusters. For example, if your source NiFi cluster has 3 nodes, the CFM 2.1.1 NiFi cluster must have at least 3 nodes as well.

Related Information

Deployment Guide

Add and configure the NiFi service

Provides the steps for how to add and configure your NiFi service.

Before you begin

- You have installed a CDP Private Cloud Base cluster and prepared it for the CFM deployment. For more information, see the *Deployment Guide.*
- You have equivalence between source and target clusters. For example, if your source NiFi cluster has 3 nodes, the CFM 2.1.1 NiFi cluster must have at least 3 nodes as well.
- You have reviewed the information about preserving your source cluster files and directories and made the necessary backups.

Before you begin

Procedure

1. From Cloudera Manager, add the CFM 2.1.1 NiFi service.
2. Set some initial configurations.

   Generally, you can accept default values during the initial installation. However, there are some settings that you should configure before proceeding:

Property	Description
Master Key Password `nifi.master.key.password`	This password is used when you generate the master key for sensitive properties encryption in the NiFi properties file when it is written to disk. It must contain at least 12 characters.
Sensitive Properties Key `nifi.sensitive.props.key`	This is the password used when you encrypt any sensitive property values that are configured in NiFi components. It must contain at least 12 characters. If you change the Sensitive Properties Key from what was used in your source cluster, you must also update the encrypted sensitive property values in the flow.xml.gz. Refer to the section “Migrating a Flow with Sensitive Properties” below.

3. Stop the NiFi service.
4. Update the NiFi configuration.

   In your CFM 2.1.1 NiFi, use Cloudera Manager to walk through all the configuration values and match the values from your source cluster that are not cluster specific. Examples of cluster specific values include keystore, truststore, ZooKeeper hostnames, and similar.

   Reference the source NiFi configuration files collected earlier as needed. Double check all entries for typos.

   Note:

   Do not copy over or migrate any data repositories, local state, ZooKeeper state, or flow.xml.gz from your source NiFi at this time.

Example

Sample configuration changes

Update the Login Identity Provider properties.

The Template for login-identity-providers.xml from Ambari is now composed of individual properties in Cloudera Manager.

As an example, if using LDAP for authentication, the following login-identity-providers.xml:

```
<loginIdentityProviders>
    <provider>
            <identifier>ldap-provider</identifier>
            <class>org.apache.nifi.ldap.LdapProvider</class>
            <property name="Authentication Strategy">SIMPLE</property>
            <property name="Manager DN">uid=admin,ou=people,dc=hadoop,dc=
apache,dc=org</property>
            <property name="Manager Password">admin-password</property>
            <property name="Referral Strategy">FOLLOW</property>
            <property name="Connect Timeout">10 secs</property>
            <property name="Read Timeout">10 secs</property>
            <property name="Url">ldap://ctr-e144-1587379642025-3931-01-00
0003.hwx.site:33389</property>
            <property name="User Search Base">ou=people,dc=hadoop,dc=apac
he,dc=org</property>
            <property name="User Search Filter">uid={0}</property>
            <property name="Identity Strategy">USE_USERNAME</property>
            <property name="Authentication Expiration">12 hours</property>
    </provider>
```

```
</loginIdentityProviders>
```

You would use Cloudera Manager to set the following NiFi service properties instead.

- LDAP Enabled is checked
- Login Identity Provider: Default LDAP Provider Class set to org.apache.nifi.ldap.LdapProvider
- LDAP Authentication Strategy set to SIMPLE
- LDAP Manager DN set to uid=admin,ou=people,dc=hadoop,dc=apache,dc=org
- LDAP Manager Password set to admin-password
- LDAP Referral Strategy set to FOLLOW
- LDAP Connect Timeout set to 10 secs
- LDAP Read Timeout set to 10 secs
- LDAP Url set to ldap://ctr-e144-1587379642025-3931-01-000003.hwx.site:33389
- LDAP User Search Base set to ou=people,dc=hadoop,dc=apache,dc=org
- Login Identity Provider: Default LDAP User Search Filter set to uid={0}
- Login Identity Provider: Default LDAP Identity Strategy set to USE_USERNAME
- Login Identity Provider: Default LDAP Authentication Expiration set to 12 hours

There are several additional LDAP configuration requirements:

- Enable TLS/SSL for NiFi Node is checked
- Initial Admin Identity set to admin
- Login Identity Provider ID set to ldap-provider
- Authorizers: LDAP User Search Filter set to (uid=*)
- Authorizers: LDAP User Identity Attribute set to uid

What to do next

When you have completed the steps for adding and configuring the NiFi Service, you may proceed with adding and configuring the NiFi Registry service.

Related Information

Deployment Guide

Add and configure the NiFi Registry service

Provides the steps for how to add and configure your NiFi Registry service.

Before you begin

- You have installed a CDP Private Cloud Base cluster and prepared it for the CFM deployment. For more information, see the *Deployment Guide*.
- You have equivalence between source and target clusters. For example, if your source NiFi cluster has 3 nodes, the CFM 2.1.1 NiFi cluster must have at least 3 nodes as well.
- You have added the NiFi service.

Procedure

1. Add CFM 2.1.1 NiFi Registry service.

2. Set some initial configurations.

   Generally, you can accept default values during the initial installation. However, there are some settings that you should configure before proceeding:

Property	Description
Master Key Password `nifi.registry.master.key.password`	This password is used to generate the master key for encrypting NiFi Registry properties on the filesystem.

3. Stop the NiFi Registry service.
4. Update the NiFi Registry configuration.

   In your CFM 2.1.1 NiFi Registry, use Cloudera Manager to walk through all the configuration values and match the values from your source cluster that are not cluster specific. Examples of cluster specific values include keystore, truststore, and similar.

   Reference the source NiFi Registry configuration files collected earlier as needed. Double check all entries for typos.

   Note:

   Do not copy over or migrate any data repositories, local state, ZooKeeper state, or flow.xml.gz from your source NiFi Registry at this time.

Example

Sample configuration changes

Update the Login Identity Provider properties.

The Template for identity-providers.xml from Ambari is now composed of individual properties in Cloudera Manager.

As an example, if using LDAP for authentication, the following identity-providers.xml:

```
<identityProviders>
    <provider>
        <identifier>ldap-provider</identifier>
     <class>org.apache.nifi.registry.security.ldap.LdapIdentityProvider</c
lass>
        <property name="Authentication Strategy">SIMPLE</property>
        <property name="Manager DN">uid=admin,ou=people,dc=hadoop,dc=apache
,dc=org</property>
        <property name="Manager Password">admin-password</property>
        <property name="Referral Strategy">FOLLOW</property>
        <property name="Connect Timeout">10 secs</property>
        <property name="Read Timeout">10 secs</property>
        <property name="Url">ldap://ctr-e144-1587379642025-3931-01-000003.h
wx.site:33389</property>
        <property name="User Search Base">ou=people,dc=hadoop,dc=apache,dc=
org</property>
        <property name="User Search Filter">uid={0}</property>
        <property name="Identity Strategy">USE_USERNAME</property>
        <property name="Authentication Expiration">12 hours</property>
    </provider>
</identityProviders>
```

You would use Cloudera Manager to set the following NiFi Registry service properties instead.

- LDAP Enabled is checked
- Identity Provider: Default LDAP Provider Class set to org.apache.nifi.registry.security.ldap.LdapIdentityProvider
- LDAP Authentication Strategy set to SIMPLE
- LDAP Manager DN set to uid=admin,ou=people,dc=hadoop,dc=apache,dc=org
- LDAP Manager Password set to admin-password

- LDAP Referral Strategy set to FOLLOW
- LDAP Connect Timeout set to 10 secs
- LDAP Read Timeout set to 10 secs
- LDAP Url set to ldap://ctr-e144-1587379642025-3931-01-000003.hwx.site:33389
- LDAP User Search Base set to ou=people,dc=hadoop,dc=apache,dc=org
- Identity Provider: Default LDAP User Search Filter set to uid={0}
- Identity Provider: Default LDAP Identity Strategy set to USE_USERNAME
- Identity Provider: Default LDAP Authentication Expiration set to 12 hours

There are several additional LDAP configuration requirements:

- Enable TLS/SSL for NiFi Registry is checked
- Initial Admin Identity set to admin
- Identity Provider Identifier set to ldap-provider
- Authorizers: LDAP User Search Filter set to (uid=*)
- Authorizers: LDAP User Identity Attribute set to uid
- Client Authentication Required is unchecked

What to do next

When you have finished adding and configuring both the NiFi and NiFi Registry services, verify your CFM 2.1.1 installation.

Related Information

Deployment Guide

Verify CFM 2.1.1

Provides steps to verify your CFM 2.1.1 installation.

Before you begin

If you updated the Initial Admin Identity (nifi.initial.admin.identity) after the initial NiFi service installation, select Actions | Reset File-based Authorizer Users and Policies before you start the NiFi service. This archives the existing users.xml and authorizations.xml files and new ones will be generated based on configuration changes. Similarly, if you updated the Initial Admin Identity (nifi.registry.initial.admin.identity) after the initial NiFi Registry service installation, select Actions | Reset File-based Authorizer Users and Policies before you start the NiFi Registry service.

Procedure

1. Start the NiFi service.
2. Start the NiFi Registry service.
3. Verify NiFi and NiFi Registry started successfully securely over HTTPS.
4. Login to NiFi using your authorized admin user.

   You should see a blank canvas.
5. Login to NiFi Registry using your authorized admin user.

   You will see no resources.

   Tip:

   Note the Registry URL so that it can be referenced later.
6. Once you are satisfied that all services are working correctly, shutdown both NiFi and NiFi Registry services and continue with the migration.

What to do next

Important:

The following sections may incur downtime. Read all sections before proceeding. A migration without downtime is possible depending on the use cases involved. Contact your Cloudera representative to engage Professional Services for guidance.

When you are ready to proceed, your next step is to shut down the source services.

Clear activity and shut down source services

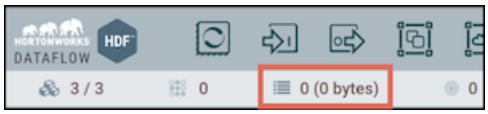
Provides the steps to clear the activity and shut down your source cluster NiFi and NiFi Registry services.

Before you begin

- You have preserved your source cluster files and directories.
- You have installed and verified CFM 2.1.1.

Procedure

1. In the source NiFi cluster, stop all the source processors to prevent the ingestion of new data.
2. Confirm that there is no active data in any of the queues by monitoring the left side of the NiFi status bar. When all active data has stopped, the status bar shows 0 FlowFiles and 0 bytes of data:



3. Shutdown the source NiFi service.
4. Shutdown the source NiFi Registry service.

What to do next

When you have finished shutting down your source services, follow the steps to migrate the NiFi data directories.

Migrate the NiFi data directories

Provides the steps for migrating the NiFi data directories.

Before you begin

When you are migrating any of the following from the source cluster to the destination cluster, ensure that the file and directory ownerships and permissions are consistent.

About this task

The database_repository consists of two H2 databases and their corresponding lock files:

- nifi-flow-audit.h2.db
- nifi-flow-audit.lock.db
- nifi-user-keys.h2.db
- nifi-user-keys.lock.db

The nifi-user-keys.h2.db contains information about who has logged into NiFi, if NiFi has been secured. The nifi-flow-audit.h2.db contains flow configuration history. These databases only need to be migrated if this historical

information needs to be retained, but note that any NiFi nodes referenced in them will refer to the source nodes. If you do not wish to retain the database information, you may skip step 2.

Procedure

1. Copy the entire provenance_repository directory from a source cluster node to a node on the destination cluster. Provenance repositories cannot be merged. For example: source Node 1 # destination Node 1, source Node 2 # destination Node 2, … source Node N # destination Node N.

   Note:

   The path to the destination provenance repository can be the same as the source provenance repository (if on a separate physical disk, for example), but it is not a requirement. You must place the provenan ce_repository directory in the path assigned to the nifi.provenance.repository.directory property in the destination cluster. The default path is /var/lib/nifi/provenance_repository.
2. Copy the entire database_repository directory from a source cluster node to a node on the destination cluster. For example: source Node 1 # destination Node 1, source Node 2 # destination Node 2, … source Node N # destination Node N.

   Note:

   The path to the destination database repository can be the same as the source database repository (if on a separate physical disk, for example), but it is not a requirement. You must place the database_repository directory in the path assigned to the nifi.database.directory property in the destination cluster. The default path is /var/lib/nifi/database_repository.

What to do next

When you have completed the provenance_repository and database_repository migration, proceed by migrating the NiFi flow.xml.gz file.

Migrate the NiFi flow.xml.gz file

Provides steps for migrating the NiFi flow.xml.gz file.

Before you begin

When you migrate the flow.xml.gz from the source cluster to the destination cluster, ensure that the file ownership and permission are consistent.

Copy the flow.xml.gz file from any node in the source cluster to all nodes on the destination cluster. The default CFM 2.1.1 location is the var/lib/nifi directory.

Before you copy the flow.xml.gz file to the destination cluster, you must edit it to remove any references to the source cluster. The content of the flow.xml.gz file is different, based on source cluster activities. Here are some examples of common edits you must make.

Remove unnecessary reporting tasks

Provides steps for removing unnecessary reporting tasks in the flow.xml.gz file.

About this task

You can remove the `AmbariReportingTask` used by your HDF NiFi cluster, as it is not used by Cloudera Manager.

Procedure

1. Unzip flow.xml.gz.
2. Edit the resulting flow.xml file by removing the reporting task:

```
<reportingTask>
      <id>3b80ba0f-a6c0-48db-b721-4dbc04cef28e</id>
      <name>AmbariReportingTask</name>
      <comment/>
   <class>org.apache.nifi.reporting.ambari.AmbariReportingTask</class>
      <bundle>
        <group>org.apache.nifi</group>
        <artifact>nifi-ambari-nar</artifact>
        <version>1.11.4.3.5.1.0-17</version>
      </bundle>
      <schedulingPeriod>1 mins</schedulingPeriod>
      <scheduledState>RUNNING</scheduledState>
      <schedulingStrategy>TIMER_DRIVEN</schedulingStrategy>
      <property>
        <name>Metrics Collector URL</name>
        <value>${ambari.metrics.collector.url}</value>
      </property>
      <property>
        <name>Application ID</name>
        <value>${ambari.application.id}</value>
      </property>
      <property>
        <name>Hostname</name>
        <value>${hostname(true)}</value>
      </property>
      <property>
        <name>Process Group ID</name>
      </property>
    </reportingTask>
```

Note:

Reporting tasks are bookended by the tags <reportingTasks> and <reportingTasks/>. If you no longer have any reporting tasks in your flow, you should still retain the closing tag <reportingTasks/> in the flow.xml file.

3. Save your changes.
4. Zip the flow.xml file:

```
gzip flow.xml
```

What to do next

When you have completed the steps to remove unnecessary reporting tasks, proceed by updating the Registry Client.

Update the Registry Client

You must update the Registry Client information in the flow.xml.gz file.

Before you begin

You have removed any unnecessary reporting tasks.

Procedure

1. Unzip flow.xml.gz.
2. Edit the resulting flow.xml file by replacing the source Registry URL with the destination Registry URL and change the Registry Name if desired:

```
<registries>
    <flowRegistry>
      <id>95503839-0172-1000-ffff-ffffe2b7b914</id>
      <name>CFM Registry</name>
      <url>https://cfm_registry_node:61443</url>
      <description/>
    </flowRegistry>
  </registries>
```

3. Save your changes.
4. Zip the flow.xml file:

```
gzip flow.xml
```

What to do next

When you have updated the Registry Client, proceed by updating references to source cluster nodes with destination cluster nodes.

Update references to cluster nodes

Provides steps for updating references to source cluster nodes with destination cluster nodes in the flow.xml.gz file.

Before you begin

You have removed unnecessary reporting tasks and updated the Registry Client information.

Procedure

1. Unzip flow.xml.gz.
2. Search and replace URLs that refer to the source NiFi nodes with the equivalent destination NiFi nodes. For example, in a Remote Process Group:

```
<remoteProcessGroup>
      <id>c8647bee-0172-1000-0000-00001ff1d10f</id>
      <name>NiFi Flow</name>
      <position x="968.0" y="520.0"/>
      <comment/>
      <url>https://remote_instance_host:8443/nifi</url>
      <urls>https://remote_instance_host:8443/nifi</urls>
      <timeout>30 sec</timeout>
      <yieldPeriod>10 sec</yieldPeriod>
      <transmitting>false</transmitting>
      <transportProtocol>HTTP</transportProtocol>
      <proxyHost/>
      <proxyUser/>
      <inputPort>
        <id>6ade8167-dc3c-3afc-b22e-f38465fdf601</id>
        <name>File Listing</name>
        <position x="0.0" y="0.0"/>
        <comments/>
```

```
        <scheduledState>STOPPED</scheduledState>
        <targetId>c85b69ba-0172-1000-0000-000077bde971</targetId>
        <maxConcurrentTasks>1</maxConcurrentTasks>
        <useCompression>false</useCompression>
        <batchCount>1</batchCount>
      </inputPort>
    </remoteProcessGroup>
```

3. Save your changes.
4. Zip the flow.xml file:

```
gzip flow.xml
```

What to do next

When you have finished updating the cluster node references, proceed by updating a flow with sensitive properties.

Update a flow with sensitive properties

If the Sensitive Properties Key (nifi.sensitive.props.key) is changing from the source cluster to the destination cluster, you must update the flow.xml.gz file prior to copying it to each node.

When a value is set for nifi.sensitive.props.key, the specified key is used to encrypt sensitive properties in the flow (password fields in components for example). You can use the Encrypt-Config tool in the NiFi Toolkit to migrate the key and update the flow.xml.gz. Encrypt-Config performs the following actions:

- Reads the existing flow.xml.gz and decrypts the sensitive values using the current key.
- Encrypts all the sensitive values with a specified new key.
- Updates the existing nifi.properties and flow.xml.gz files or creates new versions of them.

See *Using the Apache NiFi Toolkit* for complete information on Encrypt-Config.

Note:

In an HDF cluster, the NiFi Toolkit scripts are located in /usr/hdf/current/nifi-toolkit/bin.

Here is an example Encrypt-Config tool command:

```
$ ./nifi-toolkit-<version>/bin/encrypt-config.sh
-f /path/to/nifi_source/flow.xml.gz
-g /path/to/create/updated/flow.xml.gz
-s <new-password>
-n /path/to/nifi_source/nifi.properties
-o /path/to/create/updated/nifi.properties
-x
```

Where:

- -f specifies the source flow.xml.gz
- -g specifies the destination flow.xml.gz
- -s specifies the new sensitive properties key
- -n specifies the source nifi.properties
- -o specifies the destination nifi.properties
- -x tells the Encrypt-Config tool to only process the sensitive properties

If values in nifi.properties have been encrypted using the Encrypt Configuration Master Key Password property in Ambari (equivalent to the nifi.master.key.password property in CFM), add the -b option:

```

```

```
$ ./nifi-toolkit-<version>/bin/encrypt-config.sh
-b /path/to/nifi_source/bootstrap.conf
-f /path/to/nifi_source/flow.xml.gz
-g /path/to/create/updated/flow.xml.gz
-s <new-password>
-n /path/to/nifi_source/nifi.properties
-o /path/to/create/updated/nifi.properties
-x
```

Where:

- -b specifies the source NiFi bootstrap.conf

Related Information

Using the Apache NiFi Toolkit

Migrate authorization policies

Describes how to migrate both Ranger and file-based policies for NiFi and NiFi Registry.

Migrate NiFi Ranger-based policies

Provides the steps for migrating NiFi Ranger-based policies

Before you begin

- You have installed the Ranger service on your destination cluster.
- You have selected Ranger as a NiFi dependency.

About this task

If the source cluster uses Ranger policies for NiFi authorizations and you require the same Ranger policies on the destination cluster, migrate the existing Ranger policies using the Ranger Import/Export feature.

Procedure

1. In your source Ranger UI, select Access Manager | Resource Based Policies. On the Service Manager page, select Export. Remove all services listed except NiFi and select Export. A JSON file is exported.
2. In your destination Ranger UI, select Access Manager | Resource Based Policies. On the Service Manager page, select the NiFi service. Delete all of the existing policies on the service, being careful not to delete the NiFi service.
3. Return to the Service Manager page in your destination Ranger. Select Import. Select the source JSON file you exported in Step 1. Map the source NiFi Ranger service to the destination NiFi Ranger service. Select Import.
4. For NiFi service policies where source NiFi nodes are referenced (for example, Proxy policy), add the group nifi to those conditions, then delete the source nodes from those policies.
5. Edit the users.xml from the source cluster by removing source node users. Replace the users.xml on each destination cluster NiFi node with the modified users.xml. The default CFM 2.1.1 location is /var/lib/nifi.

What to do next

When you have finished migrating NiFi Ranger-based policies, proceed with the steps for migrating NiFi Registry Ranger-based policies.

Migrate NiFi Registry Ranger-based policies

Provides the steps for migrating NiFi Registry Ranger-based policies.

About this task

If the source cluster uses Ranger policies for NiFi Registry authorizations and you require the same Ranger policies on the destination cluster, migrate the existing Ranger policies using the Ranger Import/Export feature.

Before you begin

- You have installed the Ranger service on your destination cluster.
- You have selected Ranger as a NiFi Registry dependency.

Procedure

1. In your source Ranger UI, select Access Manager | Resource Based Policies. On the Service Manager page, select Export. Remove all services listed except NiFi Registry and select Export. A JSON file is exported.
2. In your destination Ranger UI, select Access Manager | Resource Based Policies. On the Service Manager page, select the NiFi Registry Ranger service. Delete all of the existing policies on the service, being careful not to delete the NiFi Registry service.
3. Return to the Service Manager page in your destination Ranger. Select Import. Select the source JSON file you exported in Step 1. Map the source NiFi Registry Ranger service to the destination NiFi Registry Ranger service. Select Import.
4. For NiFi Registry service policies where source NiFi nodes are referenced (for example, Proxy and Bucket policies), add the group nifiregistry to those conditions, then delete the source nodes from those policies.
5. Edit the users.xml from the source cluster by removing source node users. Replace the users.xml on the destination cluster NiFi Registry node with the modified users.xml. The default CFM 2.1.1 location is /var/lib/nifiregistry.

What to do next

When you have completed migrating NiFi Registry policies, you may proceed to migrating NiFi state and custom components.

Migrate NiFi file-based policies

Provides information about and examples of migrating NiFi file-based policies.

To migrate NiFi file-based authorization policies, you will perform the following edits.

Edit the users.xml from the source cluster by removing references to the source node users. Replace the users.xml on each destination NiFi node with the modified users.xml. The default CFM 2.1.1 location is /var/lib/nifi.

CFM NiFi uses the CMUserGroupProvider, configured in the authorizers.xml file, and places all the NiFi node hostnames in the nifi group. Edit the authorizations.xml from the source cluster by removing source node users from each policy they were assigned and replacing them with the nifi group identifier.

For example, if HDF NiFi had three NiFi node users on the “proxy user requests” policy:

```
<policy identifier="287edf48-da72-359b-8f61-da5d4c45a270" resource="/proxy"
 action="W">
            <user identifier="fd4f71d3-7b7c-3286-bec0-d42a752f1e0c"/>
            <user identifier="80ba71da-7789-3c7b-924f-041d598c8137"/>
            <user identifier="e32ed0da-e652-39e6-86db-48cfa3750a9f"/>
       </policy>
```

You should edit the policy with the following information for CFM 2.1.1:

```
<policy identifier="287edf48-da72-359b-8f61-da5d4c45a270" resource="/proxy"
 action="W">
            <group identifier="nifi"/>
</policy>
```

You must make similar changes if your flow uses Site-to-Site. You must update the “retrieve site-to-site details” global access policy:

```
<policy identifier="c86712ce-0172-1000-0000-00007a4ea450" resource="/site-to
-site" action="R">
            <group identifier="nifi"/>
</policy>
```

You must also update the “retrieve data via site-to-site” policy on related input ports:

```
<policy identifier="c8a608dd-0172-1000-ffff-ffffec1193ac" resource="/data-t
ransfer/input-ports/c85b69ba-0172-1000-0000-000077bde971" action="W">
            <group identifier="nifi"/>
       </policy>
```

Note:

For any policies that require both group and user identifiers, the group identifier must be placed before the user identifier on each policy.

Replace the authorizations.xml on each destination cluster NiFi node with the modified authorizations.xml. The default CFM 2.1.1 location is /var/lib/nifi.

After you finish

When you have finished migrating NiFi file-based policies, proceed with the steps for migrating NiFi Registry file-based policies.

Migrate NiFi Registry file-based policies

Provides information about and examples of migrating NiFi Registry file-based policies.

Edit the users.xml from the source cluster by removing source node users. Replace the users.xml on the NiFi Registry node with the modified users.xml. The default CFM 2.1.1 location is /var/lib/nifiregistry.

CFM NiFi Registry uses the CMUserGroupProvider, configured in the authorizers.xml file, and places all the NiFi node hostnames in the nifiregistry group. Edit the authorizations.xml from the source cluster by removing source node users from each policy they were assigned and replacing them with the nifiregistry group identifier.

For example, if HDF NiFi Registry had three NiFi node users on each of the Read/Write/Delete proxy policies:

```
<policy identifier="d59a54f7-6dd6-34ad-a279-a26ffdb9eef8" resource="/proxy"
 action="R">
            <user identifier="21232f29-7a57-35a7-8389-4a0e4a801fc3"/>
            <user identifier="fd4f71d3-7b7c-3286-bec0-d42a752f1e0c"/>
            <user identifier="80ba71da-7789-3c7b-924f-041d598c8137"/>
            <user identifier="e32ed0da-e652-39e6-86db-48cfa3750a9f"/>
        </policy>
<policy identifier="287edf48-da72-359b-8f61-da5d4c45a270" resource="/proxy"
 action="W">
            <user identifier="21232f29-7a57-35a7-8389-4a0e4a801fc3"/>
            <user identifier="fd4f71d3-7b7c-3286-bec0-d42a752f1e0c"/>
```

```
            <user identifier="80ba71da-7789-3c7b-924f-041d598c8137"/>
            <user identifier="e32ed0da-e652-39e6-86db-48cfa3750a9f"/>
        </policy>
<policy identifier="6dbdbffd-8a7d-32e1-ba3e-f600e6c69791" resource="/proxy"
 action="D">
            <user identifier="21232f29-7a57-35a7-8389-4a0e4a801fc3"/>
            <user identifier="fd4f71d3-7b7c-3286-bec0-d42a752f1e0c"/>
            <user identifier="80ba71da-7789-3c7b-924f-041d598c8137"/>
            <user identifier="e32ed0da-e652-39e6-86db-48cfa3750a9f"/>
        </policy>
```

You should edit these policies with the following information for CFM 2.1.1:

```
<policy identifier="d59a54f7-6dd6-34ad-a279-a26ffdb9eef8" resource="/proxy"
 action="R">
            <group identifier="nifiregistry"/>
            <user identifier="21232f29-7a57-35a7-8389-4a0e4a801fc3"/>
        </policy>
<policy identifier="287edf48-da72-359b-8f61-da5d4c45a270" resource="/proxy"
action="W">
            <group identifier="nifiregistry"/>
            <user identifier="21232f29-7a57-35a7-8389-4a0e4a801fc3"/>
        </policy>
<policy identifier="6dbdbffd-8a7d-32e1-ba3e-f600e6c69791" resource="/proxy"
 action="D">
            <group identifier="nifiregistry"/>
            <user identifier="21232f29-7a57-35a7-8389-4a0e4a801fc3"/>
        </policy>
```

Note:

For any policies that require both group and user identifiers, the group identifier must be placed before the user identifier on each policy.

Replace the authorizations.xml on the NiFi Registry node with the modified authorizations.xml . The default CFM 2.1.1 location is /var/lib/nifiregistry.

After you finish

When you have completed migrating NiFi Registry file-based policies, you may proceed to migrating NiFi state and custom components.

Migrate NiFi state and custom components

Provides steps for migrating NiFi state and custom components.

Copy local state

The state-management.xml file contains a local-provider value with a directory path. Copy the contents of local state directory from each source NiFi node to a node in the destination cluster. For example: source Node 1 # destination Node 1, source Node 2 # destination Node 2, …, source Node N # destination Node N.

The default state directory path in CFM 2.1.1 is /var/lib/nifi/state.

Copy cluster state

The state-management.xml file contains a cluster-provider value with ZooKeeper configuration. Use the ZooKeeper Migrator in the NiFi Toolkit to migrate NiFi ZooKeeper content from the source cluster to the destination.

Note:

In an HDF cluster, the NiFi Toolkit scripts are located in /usr/hdf/current/nifi-toolkit/bin.

1. Export the NiFi component data from the source ZooKeeper:

```
./zk-migrator.sh
-r
-z sourceHostname:sourceClientPort/sourceRootPath/components
-f /path/to/export/zk-source-data.json
```

Note:

Ensure that the export directory exists as it is not created by the command.

2. Migrate the source ZooKeeper data to the destination ZooKeeper:

```
./zk-migrator.sh
-s
-z destinationHostname:destinationClientPort/destinationRootPath/compon
ents
-f /path/to/export/zk-source-data.json
```

Note:

In CFM 2.1.1, the NiFi Toolkit scripts are located in /opt/cloudera/parcels/CFM-<version>/TOOLKIT/bin.

For more information on the ZooKeeper Migrator, see *Using the Apache NiFi Toolkit*.

Manage custom components

1. In Cloudera Manager:
   a. Find the NiFi configuration property “NiFi Node Advanced Configuration Snippet (Safety Valve) for staging/nifi.properties.xml”
   b. Click “+” to add a snippet.
   c. For the Name, enter: nifi.nar.library.directory.custom
   d. For the Value, enter: /opt/configuration_resources/custom_lib
   e. Enter a description.

   For example:

Tip:

To specify that the property values cannot be overridden, select the Final checkbox.

2. On every destination NiFi cluster node, copy any custom NiFi NARs to the directory path specified by the Value field.

After you finish

When you have finished migrating NiFi state and any custom components, proceed by migrating the NiFi Registry data storage.

Related Information

Using the Apache NiFi Toolkit

Migrate NiFi Registry data storage

Provides information for migrating your NiFi Registry metadata database, your flow storage, and your bundle storage configurations.

Migrate the metadata database

Provides steps for migrating the H2, PostgreSQL, or MySQL metadata database.

H2 database (default)

1. Confirm the location of the source Registry metadata database specified by the nifi.registry.db.url property in nifi-registry.properties.
2. Copy the database from the source NiFi Registry to the location specified by the NiFi Registry JDBC Url (nifi.registry.db.url) in the destination NiFi Registry configuration. The default directory path in CFM 2.1.1 is /var/lib/nifiregistry/database.

Note:

The NiFi Registry Database Password (nifi.registry.db.password) in CFM must match the password used with the source H2 database.

PostgreSQL

1. In the destination Registry, match the configuration in the source nifi-registry.properties. Sample properties:
   - NiFi Registry JDBC Url (nifi.registry.db.url) – jdbc:postgresql://<POSTGRES-HOSTNAME>/nifireg
   - NiFi Registry JDBC Driver (nifi.registry.db.driver.class) – org.postgresql.Driver
   - NiFi Registry H2 directory storage location (nifi.registry.db.driver.directory) – /path/to/drivers

     Note:

     The NiFi Registry H2 directory storage location specifies the NiFi Registry database driver directory. The H2 database is used by default. Update this field when you are configuring it for an external database.
   - Username for NiFi Registry metadata database (nifi.registry.db.username) – nifireg
   - Password for NiFi Registry metadata database (nifi.registry.db.password) – changeme
2. Save the changes.
3. Download the Postgres JDBC driver and place it in the expected driver directory:

```
/path/to/drivers/postgresql-driver.jar
```

Note:

These steps assume the destination registry is pointing to the same external database referenced by the source registry. If the source registry has new databases, you must migrate the data from the source databases by following the database specific steps for export and import.

MySQL

1. In the destination Registry, match the configuration in the source nifi-registry.properties. Sample properties:
   - NiFi Registry JDBC Url (nifi.registry.db.url) – jdbc:mysql://<MYSQL-HOSTNAME>/nifi_registry
   - NiFi Registry JDBC Driver (nifi.registry.db.driver.class) – com.mysql.cj.jdbc.Driver
   - NiFi Registry H2 directory storage location (nifi.registry.db.driver.directory) – /path/to/drivers

     Note:

     The NiFi Registry H2 directory storage location specifies the NiFi Registry database driver directory. The H2 database is used by default. Update this field when you are configuring it for an external database.
   - Username for NiFi Registry metadata database (nifi.registry.db.username) – nifireg
   - Password for NiFi Registry metadata database (nifi.registry.db.password) – changeme
2. Save the changes.
3. Download the MySQL JDBC driver and place it in the expected driver directory:

```
/path/to/drivers/mysql-connector-java-driver.jar
```

Note:

These steps assume the destination registry is pointing to the same external database referenced by the source registry. If the source registry has new databases, you must migrate the data from the source databases by following the database specific steps for export and import.

After you finish

When you have completed your metadata database migration, you may proceed by migrating your flow storage.

Migrate flow storage

Provides steps for migrating your local file system, git, or database table flow storage.

Local file system flow storage (`FileSystemFlowPersistenceProvider`)

This provider is the default Flow Persistence Provider.

1. Confirm the configuration of the flow storage directory specified in the source registry providers.xml:

```
<flowPersistenceProvider>
<class>org.apache.nifi.registry.provider.flow.FileSystemFlowPersistencePr
ovider</class>
<property name="Flow Storage Directory">./flow_storage</property>
</flowPersistenceProvider>
```

2. Copy the flow storage directory from the source registry installation to the location specified by the Providers: Default Flow Persistence File Provider Property - Flow Storage Directory (xml.providers.flowPersistenceProvider.file-provider.property.Flow Storage Directory) configuration property in the destination registry. The default directory path in CFM 2.1.1 is /var/lib/nifiregistry/flow_storage.

Git flow storage (`GitFlowPersistenceProvider`)

This provider stores flow contents under a Git directory.

1. Confirm the configuration of the flow storage directory specified in the source registry providers.xml:

```
<flowPersistenceProvider>
```

```
<class>org.apache.nifi.registry.provider.flow.git.GitFlowPersistenceProvi
der</class>
        <property name="Flow Storage Directory">./your_repo</property>
        <property name="Remote To Push">origin</property>
        <property name="Remote Access User">git_user</property>
        <property name="Remote Access Password">git_password</property>
</flowPersistenceProvider>
```

2. In the git repository directory, run git remote -v to show the URL of the remote.
3. Clone the git repository in the destination environment.
4. Configure the destination registry to point at the git repo. In Cloudera Manager:
   - Uncheck Providers: Enable File Flow Persistence Provider (xml.providers.flowPersistenceProvider.file-provider.enabled) so that FileSystemFlowPersistenceProvider is no longer enabled
   - Find the NiFi Registry configuration property NiFi Registry Advanced Configuration Snippet (Safety Valve) for staging/providers.xml. Add the following snippets:
     - Name:

       ```
       xml.providers.flowPersistenceProvider.git-provider.enabled
       ```

       Value: true
     - Name:

       ```
       xml.providers.flowPersistenceProvider.git-provider.class
       ```

       Value: org.apache.nifi.registry.provider.flow.git.GitFlowPersistenceProvider
     - Name:

       ```
       xml.providers.flowPersistenceProvider.git-provider.property.Flow Sto
       rage
                       Directory
       ```

       Value: ./your_repo
     - Name:

       ```
       xml.providers.flowPersistenceProvider.git-provider.property.Remote To
                       Push
       ```

       Value: origin
     - Name:

       ```
       xml.providers.flowPersistenceProvider.git-provider.property.Remote A
       ccess
                       User
       ```

       Value: git_user
     - Name:

       ```
       xml.providers.flowPersistenceProvider.git-provider.property.Remote A
       ccess
               Password
       ```

       Value: git_password
5. Save the changes.

Note:

If “Remote To Push” is not defined and flows are stored in a local git repo, copy the flow storage directory from the source NiFi Registry installation to the referenced location in the destination NiFi Registry.

Database table storage (DatabaseFlowPersistenceProvider)

This provider leverages the same database used for the metadata database.

1. Confirm the configuration of the flow storage directory specified in the source registry providers.xml:

```
<flowPersistenceProvider>
<class>org.apache.nifi.registry.provider.flow.DatabaseFlowPersistenceProv
ider</class>
</flowPersistenceProvider>
```

2. Configure the destination registry. In Cloudera Manager:
   - Uncheck Providers: Enable File Flow Persistence Provider (xml.providers.flowPersistenceProvider.file-provider.enabled) so that FileSystemFlowPersistenceProvider is no longer enabled
   - Find the NiFi Registry configuration property NiFi Registry Advanced Configuration Snippet (Safety Valve) for staging/providers.xml. Add the following snippets:
     - Name:

       ```
       xml.providers.flowPersistenceProvider.database-provider.enabled
       ```

       Value: true
     - Name:

       ```
       xml.providers.flowPersistenceProvider.database-provider.class
       ```

       Value: org.apache.nifi.registry.provider.flow.DatabaseFlowPersistenceProvider
3. Save the changes.

Note:

If the versioned flows have nested version controlled process groups, you must update the registry URL in the stored flows, as they reference the source NiFi Registry. For example, a flow snapshot file would have this sample content:

```
"versionedFlowCoordinates" : {
            "bucketId" : "4d00e96b-dd33-447a-81bf-a240832e3272",
            "flowId" : "873ef981-48c1-41c2-9c58-89c6ae93d891",
            "registryUrl" : "https://hdf_registry_hostname:61080",
            "version" : 1
          }
```

Replace the HDF Registry URL with the CFM Registry URL in each of the flow snapshot files.

After you finish

When you having completed the flow storage migration, you may proceed by migrating your bundle storage configurations

Migrate bundle storage configurations

Provides steps on how to migrate your local file system or AWS S3 bucket bundle storage configurations.

Local file system bundle storage (FileSystemBundlePersistenceProvider)

This is the default Bundle Persistence Provider.

1. Confirm the location of the bundle storage directory specified in the source registry providers.xml:

```
<extensionBundlePersistenceProvider>
<class>org.apache.nifi.registry.provider.extension.FileSystemBundlePersist
enceProvider</class>
<property name="Extension Bundle Storage Directory">/var/lib/nifi-regis
try/extension_bundles</property>
</extensionBundlePersistenceProvider>
```

2. Copy the bundle storage directory from the source Registry installation to the location specified by the Providers: Default Extension Bundle Persistence File Provider Property - Extension Bundle Storage Directory (xml.prov iders.extensionBundlePersistenceProvider.file-bundle-provider.property.Extension Bundle Storage Dire ctory) configuration property in the destination registry. The default CFM 2.1.1 location is /var/lib/nifiregistry/ extension_bundles.

AWS S3 bucket bundle storage (`S3BundlePersistenceProvider`)

This provider stores the content of extension bundles in a AWS S3 bucket.

1. Confirm the location of the bundle storage directory specified in the source registry providers.xml:

```
<extensionBundlePersistenceProvider>
<class>org.apache.nifi.registry.provider.extension.S3BundlePersistenceProv
ider</class>
<property name="Region">us-east-1</property>
<property name="Bucket Name">my-bundles</property>
<property name="Key Prefix"></property>
<property name="Credentials Provider"DEFAULT_CHAIN</property>
<property name="Access Key">************</property>
<property name="Secret Access Key">************</property>
<property name="Endpoint URL"></property>
</extensionBundlePersistenceProvider>
```

2. Depending on your source credential provider (assuming the "default chain" which checks system properties, environment variables, and credential profiles), set up the credential provider on the destination system similarly. If you are using the "static" provider, specify the Access Key and Secret Access Key. In Cloudera Manager:
   - Uncheck Providers: Enable File Flow Persistence Provider (xml.providers.flowPersistenceProvider.file-provider.enabled) so that FileSystemFlowPersistenceProvider is no longer enabled
   - Find the NiFi Registry configuration property NiFi Registry Advanced Configuration Snippet (Safety Valve) for staging/providers.xml. Add the following snippets:
     - Name:

       ```
       xml.providers.extensionBundlePersistenceProvider.s3-bundle-provider.
       enabled
       ```

       Value: true
     - Name:

       ```
       xml.providers.extensionBundlePersistenceProvider.s3-bundle-provider.
       class
       ```

       Value: org.apache.nifi.registry.provider.extension.S3BundlePersistenceProvider
     - Name:

       ```
       xml.providers.extensionBundlePersistenceProvider.s3-bundle-provider.
       property.Region
       ```

       Value: us-east-1
     - Name:

       ```
       xml.providers.extensionBundlePersistenceProvider.s3-bundle-provider.
       property.Bucket
                       Name
       ```

       Value: my-bundles
     - Name:

       ```
       xml.providers.extensionBundlePersistenceProvider.s3-bundle-provider.
       property.Credentials
                       Provider
       ```

       Value: DEFAULT_CHAIN
     - Name:

       ```
       xml.providers.extensionBundlePersistenceProvider.s3-bundle-provider.
       property.Access
                       Key
       ```

       Value: **********
     - Name:

       ```
       xml.providers.extensionBundlePersistenceProvider.s3-bundle-provider.
       property.Secret Access
               Key
       ```

       Value: **********
3. Save the changes.

Post migration steps

After you have completed the migration, it is a best practice to review the NiFi components (processors, controller services, and reporting tasks) for any cluster specific configurations so that you can update them before you start your data flows.

Procedure

1. Start NiFi service and access UI.

   Tip:

   In NiFi configuration, consider setting Flow Controller Auto Resume State (nifi.flowcontroller.autoResumeState) to false by unchecking it. This sets the state of all components to stopped on startup, allowing you to make any post migration flow adjustments before the components run.
2. Make the following changes as needed within the NiFi UI:
   - If any components (processors, controller services, or reporting tasks) are configured specifically for the source cluster, edit for the destination cluster.
   - Address any components (processors, controller services or reporting tasks) that are marked Invalid. For example:
     - `StandardSSLContextService` or `StandardRestrictedContextService` controller services may need to be updated to use the new certs setup for the destination cluster.
     - Any client controller services that connect to a server controller service will need to be updated. For example, `DistributedMapCacheClientService` and `DistributedMapCacheServer`).
     - You may need to update Kafka and Hive processor versions.
     - You may need to update HBase and HDFS processor configurations.
3. Start NiFi Registry and access UI.

Migrate file-based authorization to Ranger

Both NiFi and NiFi Registry services have the option to convert existing file-based provider policies to Ranger provider policies.

Migrate NiFi File-Based Authorization to Ranger

You can convert existing file-based provider NiFi policies to Ranger provider policies.

Before you begin

The following steps assume that the Ranger service is installed in the CDP-DC cluster.

Procedure

1. Create any users and groups from the NiFi users.xml that do not already exist in Ranger.
2. Select Ranger as a dependency from NiFi configuration.
3. Restart NiFi.
4. Select Migrate File-based Authorizations to Ranger from the Actions drop-down. Confirm the action.
5. After a successful migration, verify that the policies are available in the NiFi Ranger service.

Migrate NiFi Registry File-Based Authorization to Ranger

You can convert existing file-based provider NiFi Registry policies to Ranger provider policies.

Before you begin

The following steps assume that the Ranger service is installed in the CDP-DC cluster.

Procedure

1. Create any users and groups from the NiFi Registry users.xml that do not already exist in Ranger.
2. Select Ranger as a dependency from NiFi Registry configuration.
3. Restart NiFi Registry.
4. Select Migrate File-based Authorizations to Ranger from the Actions drop-down. Confirm the action.
5. After a successful migration, verify that the policies are available in the NiFi Registry Ranger service.