Cloudera Flow Management 4.11.0

# Integrations

**Date published: 2019-06-26**
**Date modified: 2025-10-08**

**https://docs.cloudera.com/**

# Legal Notice

# Contents

# Integrating NiFi with Atlas when Auto-TLS is not enabled

If Auto-TLS enabled is not enabled on your Cloudera Flow Management or on the Cloudera Base on premises cluster, you must manually integrate NiFi with Atlas. To do this, create and configure the `ReportLineageToAtlas` reporting task in NiFi.

## About this task

Perform this task if **all** of the following conditions are true:

- Cloudera Flow Management does not have TLS enabled
- The Cloudera Base on premises cluster does not have auto-TLS enabled
- You do not plan to enable auto-TLS
- You want to use Atlas with Cloudera Flow Management on a Cloudera Base on premises deployment.

## Procedure

1. In NiFI, open the Global Menu in the upper-right corner, and select Controller Services.
2. Click the Reporting Tasks tab.
3. Click the Add (+) to open the Add Reporting Task dialog.
4. Select ReportLineageToAtlas, then click Add.
5. Click the Edit icon to open the Configure Reporting Task dialog.
6. Configure the required properties described below.

**Atlas URLs**

Shows a comma-separated list of Atlas Server URLs.

After the reporting task starts, you cannot modify this property to add new Atlas servers. To add an additional Atlas server, you must create a new reporting task.

**Atlas Configuration Directory**

Specifies the directory where the atlas-applications.properties file is created.

The directory must:

- Be accessible and writable by the user running the NiFi process
- Be available on each NiFi node
- Already exist (the reporting task does not create it)
- Not be located in the /tmp directory.

**Create Atlas Configuration File**

Set this property to True.

When enabled, the atlas-application-properties file is automatically created in the Atlas configuration directory when the reporting task starts.

**Lineage Strategy**

Specifies the level of granularity for your NiFi dataflow reporting to Atlas.

After reporting starts, do not switch between simple and complete lineage strategies.

**Provenance Record Start Position**

Specifies where the reporting task starts reading from the provenance event stream.

**Provenance Record Batch Size**

Specifies the number of provenance records sent to Atlas in a single batch.

**NiFi URL for Atlas**

Specifies the NiFi cluster URL.

**Atlas Authentication Method**

Specifies how the reporting task authenticates with the Atlas server. Basic authentication is the default.

**Kafka Security Protocol**

Specifies the protocol used to communicate with Kafka brokers when sending Atlas hook notification messages. This value must match Kafka's `security.protocol` property value.

# Integrating NiFi with Atlas when Auto-TLS is enabled

If Auto-TLS enabled is enabled on the Cloudera Base on premises cluster, additional manual steps are required to integrate NiFi with Atlas. These steps ensure that NiFi can securely communicate with Atlas in a TLS-enabled environment.

## About this task

Perform this task if both of the following conditions are true:

- You want to integrate Cloudera Flow Management with Atlas
- Auto-TLS is enabled on the Cloudera Base on premises cluster.

## Procedure

1. In Cloudera Manager, enable the Atlas integration option for Cloudera Flow Management.
2. Restart the NiFi service.
3. From the Cloudera Manager Actions menu for the NiFi service, select Create required NiFi object.

# Integrating NiFi and NiFi Registry with Knox

You can integrate NiFi and NiFi Registry with with Apache Knox Gateway (Knox) to securely access NiFi and NiFi Registry through a single, centralized entry point.

Knox provides the following benefits:

- Centralized access to all services in the cluster
- Single sign-on (SSO) authentication
- Service-level authorization to the cluster
- No direct exposure of service endpoints such as URLs, ports, or IP addresses.

When NiFi and NiFi Registry are integrated with Knox, users access the services through the Knox URL instead of connecting directly to individual NiFi nodes. This allows secure access to all NiFi nodes and enables seamless switching between nodes if one becomes unavailable.

For information more information on Knox, see the *Apache Knox Overview*.

For instructions on how to select Knox during the NiFi and NiFi Registry installation, see the *Cloudera Flow Management Deployment* documentation.

## Related Information

Apache Knox Overview
Cloudera Flow Management Deployment

# Customizing properties in Cloudera Manager

You can customize NiFi and NiFi Registry beyond what the customization page in Cloudera Manager allows. To make any changes, use the dot notation to represent the actual schema for a given property file.

## About this task

The following steps show how to enhance or overwrite xml based properties in Cloudera Manager using dot notation.

## Procedure

Use the following structure:

```
xml.<properties-type>.<entity>.<identifier>.class
xml.<properties-type>.<entity>.<identifier>.property.<property-value>
```

Where:

- <properties-type> for NiFi can be authorizers and loginIdentityProviders
- <properties-type> for NiFi Registry can be authorizers and identityProviders.

The following property key/value example creates a user group provider entry into the authorizers file for NiFi:

```
Name: xml.authorizers.userGroupProvider.file-user-group-provider.class
Value: org.apache.nifi.authorization.FileUserGroupProvider

Name: xml.authorizers.userGroupProvider.file-user-group-provider.property
.Initial User Identity 2
Value: CN=localhost, OU=NIFI
```

This translates to the following entry in the generated authorizers.xml file:

```
<authorizers>
…...
    <userGroupProvider>
        <identifier>file-user-group-provider</identifier>
        <class>org.apache.nifi.authorization.FileUserGroupProvider</class>
        <property name="Initial User Identity 2">CN=localhost, OU=NIFI</prop
erty>
    </userGroupProvider>
…
...
</authorizers>
```

Properties names that have spaces are supported and do not need to be escaped.

## Example

For an example, see *Pairing LDAP with a Composite Group Provider*.