

# Working with flows in NiFi Registry using NiFi Toolkit CLI

Date published: 2019-12-16

Date modified: 2023-01-11

The Cloudera logo is displayed in a bold, orange, sans-serif font. The word "CLOUDERA" is written in all caps, with a stylized 'E' that has a horizontal bar extending to the right.

# Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

# Contents

<b>Exporting or importing data flows with NiFi Toolkit CLI.....</b>	<b>4</b>
Connecting to NiFi Registry with NiFi Toolkit CLI.....	4
Exporting a flow from NiFi Registry.....	6
Importing a new flow into NiFi Registry.....	6

## Exporting or importing data flows with NiFi Toolkit CLI

You can use the NiFi CLI to export flows from the NiFi Registry or import flows to the NiFi Registry.

NiFi CLI is a command line utility included in the NiFi Toolkit. You can use it to interact with NiFi and NiFi Registry instances to automate tasks such as deploying versioned flows and managing process groups and cluster nodes.

In this use case, you can see how to connect the NiFi CLI to a NiFi Registry running in a Flow Management Data Hub cluster using basic authentication. You can also find details on how to use the NiFi CLI to export a versioned flow from this Registry instance, and on how to import it into a different one.

## Connecting to NiFi Registry with NiFi Toolkit CLI

In CDP Public Cloud, access from outside the cluster should go through the cluster's Knox gateway. Knox provides basic authentication to its API topologies, allowing you to authenticate with your CDP workload username and password.

### About this task

Knox has two ports that can be accessed, 443 and 8443. This procedure shows you how to connect to the 8443 port of the Knox Gateway Server from outside the Flow Management Data Hub cluster (for example, from a local machine). This port is secured by Cloudera Manager AutoTLS. If you want to use the 443 port, you can use this procedure as a template and modify it for your production certs.

### Before you begin

- You have a Flow Management Data Hub cluster in your CDP Public Cloud environment.
- You have the necessary permissions to create a NiFi Registry CLI properties file.

### Procedure

1. Confirm or set your CDP workload username and password.

To use non-SSO interfaces, each user and machine user must set a workload password. You can manage your workload password from your user profile page under User Management in the CDP Management Console. For instructions, see *Setting the workload password*.

You will need to reference your workload username and password in the basic authentication parameters (basicAuthUsername, basicAuthPassword) in *Step 4*.

2. Identify the Knox Gateway Server in your Flow Management Data Hub cluster.

You can find the hostname of the Knox Gateway Server in Cloudera Manager by checking the hostname in Clusters Knox Instances .

Example hostname: flow-management-azure-management0.dataflow.xcu2-8y8x.dev.cldr.work



**Note:** You can also find the Knox Gateway Server hostname on the Hardware tab of the Flow Management cluster overview page. It is the FQDN of the Management node.

3. Transfer the truststore JKS file (cm-auto-global\_truststore.jks) from one of your CDP Data Hub nodes to your local machine.
  - a) Connect to one of your NiFi nodes.
  - b) Locate the truststore file.

You can find it in the following directory:

```
/var/run/cloudera-scm-agent/process/XYZ-nifi-NIFI_NODE/
```

- c) Get the truststore password with `grep PASSWORD proc.json`.
  - d) Copy the cm-auto-global\_truststore.jks file to your local machine.
4. Create a NiFi Registry CLI properties file referring to:
    - Knox Gateway Server (specifically pointing to port 8443)
    - Truststore JKS file
    - User and workload password using Basic Authentication

You can use this properties file to store all your settings/default values for the Registry related CLI commands, so you do not need to specify the baseUrl and other arguments for every command.

Example properties file datahub-registry-8443.properties:

```
baseUrl=https://flow-management-azure-management0.dataflow.xcu2-8y8x.dev.cldr.work/flow-management-azure/cdp-proxy/nifi-registry-app
truststore=/Users/john.doe/Downloads/nifi-toolkit-1.13.2/bin/cm-auto-global_truststore.jks
truststoreType=jks
truststorePasswd=password
basicAuthUsername=jdoe156
basicAuthPassword=password1
```



**Note:** If you want to use the 443 port, you can use this procedure as a template and modify it for your production certs.

5. Check that the 8443 port is open.

If the port is not open, configure the security policy in your given cloud provider to allow external traffic to reach port 8443.

6. Set your CLI session to refer to the NiFi Registry CLI properties file you created.

```
john.doe@john bin % ./cli.sh session set nifi.reg.props datahub-registry-8443.properties
```

7. List the Registry buckets as a quick test for your connection.

You get the list of available buckets in the NiFi Registry.

In this example, the Registry has two buckets, TestProject and OtherProject:

```
john.doe@john bin % ./cli.sh registry list-buckets
```

#	Name	Id	Description
1	TestProject	2a5566b5-7380-46b8-9598-328e092e8899	(empty)
2	OtherProject	175fb557-43a2-4abb-871f-81a354f47bc2	(empty)

## Related Information

[Setting the workload password](#)

## Exporting a flow from NiFi Registry

This procedure shows you how to use the NiFi Toolkit CLI to export a versioned flow from a Registry instance.

### Before you begin

- You have established a connection to the 8443 port of the Knox Gateway Server from outside the Flow Management Data Hub cluster.
- You have a flow in the Registry that you want to export.

### Procedure

1. List the flows in the bucket that you want to explore.

You get the list of the flows in the chosen bucket.

In this example, the flows of the TestProject bucket are listed:

```
john.doe@john bin % ./cli.sh registry list-flows -bucketIdentifier 2a556
6b5-7380-46b8-9598-328e092e8899
# Name      Id      Description
-----
1 BaseFlow  06acb207-d2f1-447f-85ed-9b8672fe6d30  First flow.
2 TestFlow  ce2fdaf9-bf81-43fa-846f-f38f6891be95  Second flow.
```

2. Export a flow from the Registry to a defined location using the flow ID.

In this example, you export TestFlow as a JSON file called mytestflow.json to the tmp folder:

```
john.doe@john bin % ./cli.sh registry export-flow-version -f
ce2fdaf9-bf81-43fa-846f-f38f6891be95 -o /tmp/mytestflow.json -
ot json
```

where `-f` specifies the flow ID, `-o` specifies the name of the output file, and `-ot` specifies the output type.



**Note:** The output types supported are `json` and `simple`. NiFi and NiFi Registry support flow definitions in JSON format.

### Results

The flow is exported as a JSON file, so you have a NiFi flow definition on your local machine that you can import and deploy, for example in a different NiFi instance.

## Importing a new flow into NiFi Registry

This procedure shows you how to import a brand new flow or a new version of a flow into a Registry instance.

### Procedure

1. Create a new one or update the NiFi Registry CLI properties file that refers to the Registry of interest. Set your CLI session to refer to this NiFi Registry CLI properties file.

For instructions, see [Steps 4-6](#) in *Connecting to NiFi Registry with NiFi Toolkit CLI*.

2. Create a new flow.

```
john.doe@john bin % ./cli.sh registry create-flow -b 699ac53c-7be7-4082-
8711-f9fff823826c -fd "This is the flow description" -fn "This is the flow
name"
```

```
b748827f-565f-4a69-8ce0-7218fcab35f4
```

where `-b` specifies the bucket ID in the Registry where you want to add the new flow, `-fd` specifies the description of the flow, and `-fn` specifies the name of the flow.

This generates a flow ID for an empty flow with an initial version of 0.

### 3. Import a flow definition to the Registry as version 1.

```
john.doe@john bin % ./cli.sh registry import-flow-version -f b748827f-565f-4a69-8ce0-7218fcab35f4 -i dh_export_test.json 1
```

where `-f` specifies the flow ID and `-i` specifies the name of the input file.



**Note:** If a flow already exists in the Registry, and you want to import a new version of the flow into Registry, you just need to reference the existing flow ID in the import command and the Registry will increment it as the next version.

For example:

```
john.doe@john bin % ./cli.sh registry import-flow-version -f 7ce2fdaf9-bf81-43fa-846f-f38f6891be95 -i mytestflow.json 2
```

## Related Information

[Connecting to NiFi Registry with NiFi Toolkit CLI](#)