

Using Python scripts

Date published: 2021-04-06

Date modified: 2024-06-03

CLOUdera

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

Running Python scripts in Cloudera DataFlow flow deployments.....	4
Upload and run Python scripts in flow deployments.....	4
Install custom Python libraries in flow deployments.....	5

Running Python scripts in Cloudera DataFlow flow deployments

Relying on Python scripts to perform data transformations within data flows is a common pattern for NiFi users. Cloudera DataFlow flow deployments come with Python 3 and the following custom pre-installed packages: requests, urllib3. You can design your data flows to use the pre-installed Python runtime as well as install additional custom packages which you might require.

Upload and run Python scripts in flow deployments

If running your data flow requires executing a Python script, you have to upload it when creating your data flow deployment through the Deployment Wizard or the CLI. Follow these steps to configure your NiFi processors correctly and upload your Python script.

Procedure

1. Create your Python script and save it as a file.

For example:

```
#!/usr/bin/python3
print("Hello, World!")
```

2. Open the flow definition which requires a Python script in NiFi.
3. Add and configure an ExecuteStreamCommand processor to run your script.

Make the following property settings:

Command Arguments

provide #{Script}

Command Path

provide python

Leave all other properties with their default values.



Note: If you need to upload additional supporting files for use by your script, add a dynamic property named Additional Resources referencing a parameter #{AdditionalResources}. The primary script may reference these files through the path `/nifi-flow-assets/[***PARAMETER CONTEXT NAME**]/[***PARAMETER NAME**]/[***FILE NAME***]@f0`.

4. If you have edited your data flow in NiFi, download it as a flow definition and import it to Cloudera DataFlow. If you have edited your data flow in the Flow Designer, publish the flow to the Catalog.
5. Initiate a flow deployment from the Catalog. In the Parameters step of the Deployment Wizard, upload your Python script to the Script parameter. Upload additional files to the AdditionalResources parameter if applicable. Complete the Wizard and submit your deployment request.

Results

Your Python script is uploaded to the flow deployment and executed as part of the data flow.

Related Information

[Add a parameter in Flow Designer](#)

[Publish a draft to Catalog as a flow definition](#)

[Download a flow definition from NiFi](#)

[Import a flow definition to Cloudera DataFlow](#)

[Deploy a flow definition in Cloudera DataFlow](#)

Install custom Python libraries in flow deployments

If your data flow requires custom Python packages you can modify your Python script to install these dependencies through the use of NiFi processors.

Procedure

1. Create a Python script, to install the package you want to add:

```
#!/usr/bin/python3
try: import [***PACKAGE NAME***] as [***IMPORT AS***]
except ImportError:
    from pip._internal import main as pip
    pip(['install', '--user', '[***PACKAGE NAME***]'])
    import [***PACKAGE NAME***] as [***IMPORT AS***]
import sys
file = [***IMPORT AS***].read_csv(sys.stdin)
```

Replace `[***PACKAGE NAME***]` with the name of the package you want to import and `[***IMPORT AS***]` with a meaningful name you want the package to be called in your data flow.

```
#!/usr/bin/python3
try: import pandas as pd
except ImportError:
    from pip._internal import main as pip
    pip(['install', '--user', 'pandas'])
    import pandas as pd
import sys
file = pd.read_csv(sys.stdin)
```

2. Open the flow definition which requires custom packages in NiFi.
3. Add and configure an ExecuteStreamCommand processor to run your script.

Make the following property settings:

Command Arguments

provide #{Script}

Command Path

provide python

Leave all other properties with their default values.



Note: If you need to upload additional supporting files for use by your script, add a dynamic property named Additional Resources referencing a parameter `#{AdditionalResources}`. The primary script may reference these files through the path `/nifi-flow-assets/[***PARAMETER CONTEXT NAME***]/[***PARAMETER NAME***]/[***FILE NAME***]@f0`.

4. If you have edited your data flow in NiFi, download it as a flow definition and import it to Cloudera DataFlow. If you have edited your data flow in the Flow Designer, publish the flow to the Catalog.
5. Initiate a flow deployment from the Catalog. In the Parameters step of the Deployment Wizard, upload your Python script to the Script parameter. Upload additional files to the AdditionalResources parameter if applicable. Complete the Wizard and submit your deployment request.

Results

Your Python script is uploaded to the flow deployment and the required custom libraries are installed when the script is executed as part of the data flow.