

Installing Additional Packages

Date published: 2020-02-28

Date modified: 2021-06-24



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

Installing Additional Packages.....	4
(Python Only) Using a Requirements File.....	5
Using Conda with Cloudera Data Science Workbench.....	5
Creating an Extensible Engine with Conda.....	6

Installing Additional Packages

Cloudera Data Science Workbench engines are preloaded with a few common packages and libraries for R, Python, and Scala. However, a key feature of Cloudera Data Science Workbench is the ability of different projects to install and use libraries pinned to specific versions, just as you would on your local computer.

About this task

Generally, Cloudera recommends you install all required packages locally into your project. This will ensure you have the exact versions you want and that these libraries will not be upgraded when Cloudera upgrades the base engine image. You only need to install libraries and packages once per project. From then on, they are available to any new engine you spawn throughout the lifetime of the project.

You can install additional libraries and packages from the workbench, using either the command prompt or the terminal. Alternatively, you might choose to use a package manager such as [Conda](#) to install and maintain packages and their dependencies. For some basic usage guidelines for Conda, see [Using Conda with Cloudera Data Science Workbench](#).



Note:

Cloudera Data Science Workbench does not currently support installation of packages that require root access to the hosts. For such use-cases, you will need to create a new custom engine that extends the base engine image to include the required packages. For instructions, see [Creating Custom Engine Images](#).

Procedure

1. Navigate to your project's Overview page. Click New Session and start a session.
2. At the command prompt in the bottom right, enter the command to install the package. Some examples using Python and R have been provided.

R

```
# Install from CRAN
install.packages("ggplot2")

# Install using devtools
install.packages('devtools')
library(devtools)
install_github("hadley/ggplot2")
```

Python 2

```
# Installing from console using ! shell operator and pip:
!pip install beautifulsoup

# Installing from terminal
pip install beautifulsoup
```

Python 3

```
# Installing from console using ! shell operator and pip3:
!pip3 install beautifulsoup4
# Installing from terminal
pip3 install beautifulsoup4
```

(Python Only) Using a Requirements File

For a Python project, you can specify a list of the packages you want in a requirements.txt file that lives in your project. The packages can be installed all at once using pip/pip3.

Procedure

1. Create a new file called requirements.txt file within your project:

```
beautifulsoup4==4.6.0  
seaborn==0.7.1
```

2. To install the packages in a Python 3 engine, run the following command in the workbench command prompt.

```
!pip3 install -r requirements.txt
```

For Python 2 engines, use pip.

```
!pip install -r requirements.txt
```

Using Conda with Cloudera Data Science Workbench

Cloudera Data Science Workbench recommends using pip for package management along with a requirements.txt file (as described in the previous section).

Cloudera Data Science Workbench recommends using pip for package management along with a requirements.txt file (as described in the previous section). However, for users that prefer Conda, the default engine in Cloudera Data Science Workbench includes two environments called python2.7, and python3.6. These environments are added to sys.path, depending on the version of Python selected when you launch a new session.

In Python 2 and Python 3 sessions and attached terminals, Cloudera Data Science Workbench automatically sets the CONDA_DEFAULT_ENV and CONDA_PREFIX environment variables to point to Conda environments under /home/cdsw/.conda.

However, Cloudera Data Science Workbench does not automatically configure Conda to pin the actual Python version. Therefore if you are using Conda to install a package, you must specify the version of Python. For example, to use Conda to install the feather-format package into the python3.6 environment, run the following command in the Workbench command prompt:

```
!conda install -y -c conda-forge python=3.6.1 feather-format
```

To install a package into the python2.7 environment, run:

```
!conda install -y -c conda-forge python=2.7.11 feather-format
```

Note that on sys.path, pip packages have precedence over conda packages.

**Note:**

- If your project is using an older base engine image (version 3 and lower), you will need to specify both the Python version as well as the Conda environment. For example:

```
!conda install -y -c conda-forge --name python3.6 python=3.6.1 feather-format
```

The Conda environment is also required when you create an extensible engine using Conda (as described in the following section).

- Cloudera Data Science Workbench does not automatically configure a Conda environment for R and Scala sessions and attached terminals. If you want to use Conda to install packages from an R or Scala session or terminal, you must manually configure Conda to install packages into the desired environment.

Creating an Extensible Engine with Conda

Cloudera Data Science Workbench recommends using pip for package management along with a requirements.txt file (as described in the previous section).

About this task

Cloudera Data Science Workbench also allows you to extend its base engine image to include packages of your choice using Conda. To create an extended engine:

Procedure

1. Add the following lines to a Dockerfile to extend the base engine, push the engine image to your Docker registry, and include the new engine in the allowlist for your project. For more details on this step, see [Extensible Engines](#).
Python 2

```
RUN mkdir -p /opt/conda/envs/python2.7
RUN conda install -y nbconvert python=2.7.11 -n python2.7
```

Python 3

```
RUN mkdir -p /opt/conda/envs/python3.6
RUN conda install -y nbconvert python=3.6.1 -n python3.6
```

2. Set the PYTHONPATH environmental variable as shown below. You can set this either globally in the site administrator dashboard, or for a specific project by going to the project's **Settings Engine** page.

Python 2

```
PYTHONPATH=$PYTHONPATH:/opt/conda/envs/python2.7/lib/python2.7/site-packages
```

Python 3

```
PYTHONPATH=$PYTHONPATH:/opt/conda/envs/python3.6/lib/python3.6/site-packages
```