

## Managing Engine Dependencies

Date published: 2020-02-28

Date modified:

**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

<b>Managing Engine Dependencies.....</b>	<b>4</b>
Installing Packages Directly Within Projects.....	4
Creating a Customized Engine with the Required Package(s).....	5
Mounting Additional Dependencies from the Host.....	5
Managing Dependencies for Spark 2 Projects.....	6

## Managing Engine Dependencies

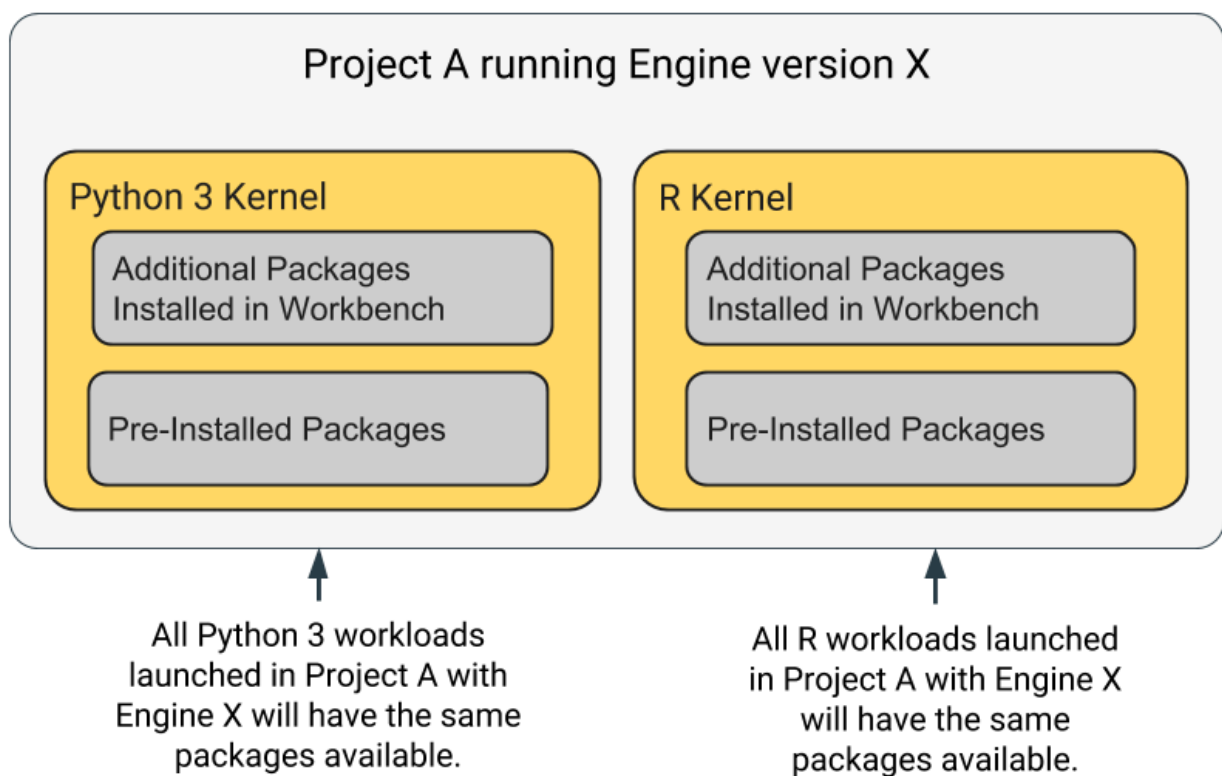
This page describes the options available to you for mounting a project's dependencies into its engine environment:



**Important:** Even though experiments and models are created within the scope of a project, the engines they use are completely isolated from those used by sessions or jobs launched within the same project.

### Installing Packages Directly Within Projects

Cloudera Data Science Workbench engines are preloaded with a few common packages and libraries for R, Python, and Scala. In addition to these, Cloudera Data Science Workbench allows you to install any other packages or libraries required by your projects just as you would on your local computer.



Each project's environment is completely isolated from others, which means you can install different versions of libraries pinned to different projects.

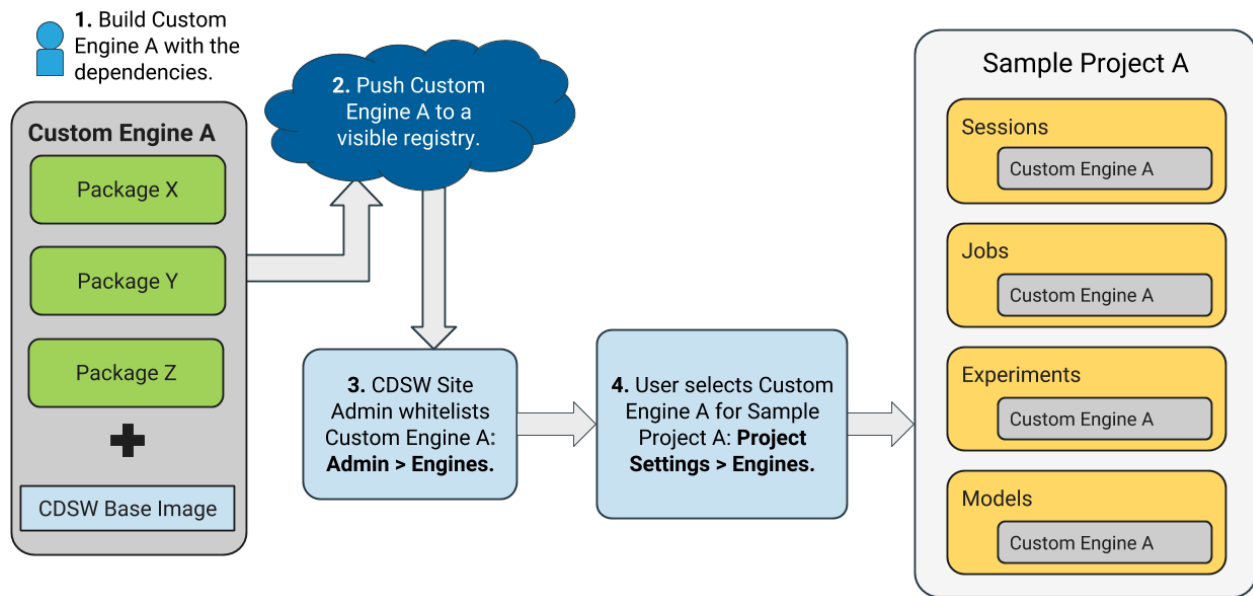
Libraries can be installed from the workbench using the inbuilt interactive command prompt or the terminal. Any dependencies installed this way are mounted to the project environment at `/home/cdsw`. Alternatively, you could choose to use a package manager such as [Conda](#) to install and maintain packages and their dependencies.

Note that overriding pre-installed packages by installing packages directly in the workbench can have unwanted side effects. It is not recommended or supported.

For detailed instructions, see [Installing Additional Packages](#).

## Creating a Customized Engine with the Required Package(s)

Directly installing a package to a project as described above might not always be feasible. For example, packages that require root access to be installed, or that must be installed to a path outside /home/cdsw (outside the project mount), cannot be installed directly from the workbench. For such circumstances, Cloudera recommends you extend the base Cloudera Data Science Workbench engine image to build a customized image with all the required packages installed to it.

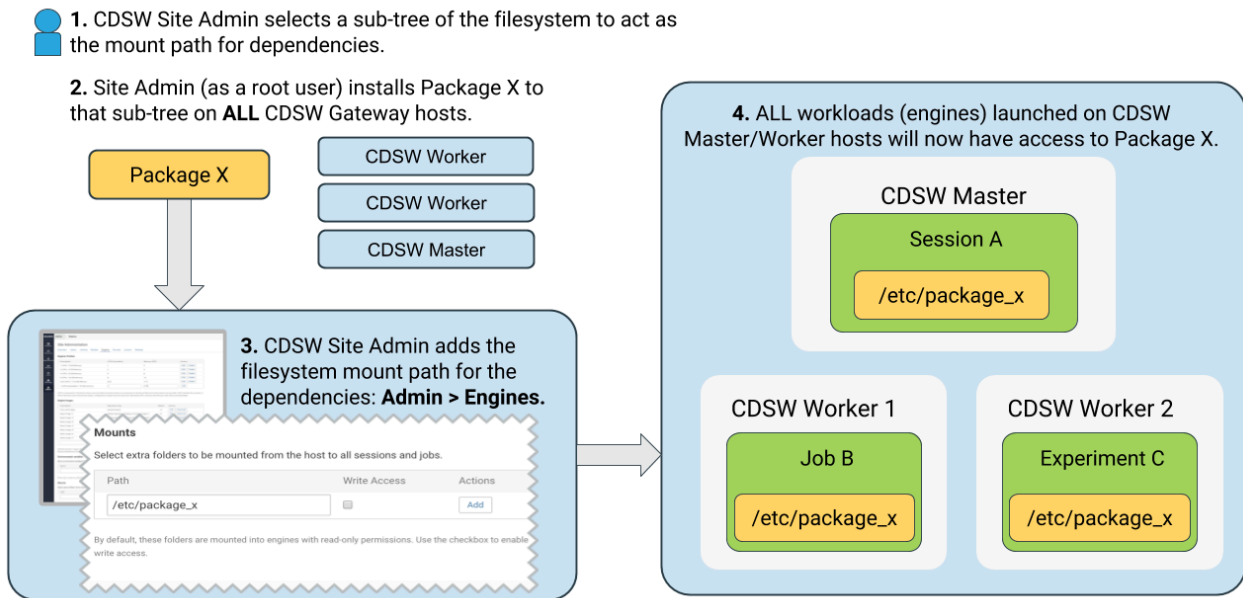


This approach can also be used to accelerate project setup across the deployment. For example, if you want multiple projects on your deployment to have access to some common dependencies out of the box or if a package just has a complicated setup, it might be easier to simply provide users with an engine environment that has already been customized for their project(s).

For detailed instructions with an example, see [Customized Engine Images](#).

## Mounting Additional Dependencies from the Host

As described previously, all Cloudera Data Science Workbench projects run within an engine.



By default, Cloudera Data Science Workbench automatically mounts the CDH parcel directory and client configuration for required services, such as HDFS, Spark, and YARN, into the engine. However, if users want to reference any additional files/folders on the host, site administrators need to explicitly load them into engine containers at runtime.

### Limitation

If you use this option, you must self-manage the dependencies installed on the gateway hosts to ensure consistency across them all. Cloudera Data Science Workbench cannot manage or control the packages with this method. For example, you must manually ensure that version mismatches do not occur. Additionally, this method can lead to issues with experiment repeatability since CDSW does not control or keep track of the packages on the host. Contrast this with [Installing Packages Directory Within Projects](#) and [Creating a Customized Engine](#), where Cloudera Data Science Workbench is aware of the packages and manages them.

For instructions on how to configure host mounts, see [Configuring the Engine Environment](#). Note that the directories specified here will be available to all projects across the deployment.

## Managing Dependencies for Spark 2 Projects

With Spark projects, you can add external packages to Spark executors on startup.

To add external dependencies to Spark jobs, specify the libraries you want added by using the appropriate configuration parameters in a spark-defaults.conf file.

For a list of the relevant properties and examples, see [Managing Dependencies for Spark 2 Jobs](#).