

Engine Environment Variables

Date published: 2020-02-28

Date modified: 2021-02-25



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

Engine Environment Variables.....	4
Environment Variables from Cloudera Manager.....	4
Accessing Environmental Variables from Projects.....	5
Engine Environment Variables.....	5

Engine Environment Variables

Environmental variables allow you to customize engine environments for projects.

For example, if you need to configure a particular timezone for a project, or increase the length of the session/job timeout windows, you can use environmental variables to do so. Environmental variables can also be used to assign variable names to secrets such as passwords or authentication tokens to avoid including these directly in the code.

In general, Cloudera recommends that you do not include passwords, tokens, or any other secrets directly in your code because anyone with read access to your project will be able to view this information. A better place to store secrets is in your project's environment variables, where only project collaborators and admins have view access. They can therefore be used to securely store confidential information such as your AWS keys or database credentials.

Cloudera Data Science Workbench allows you to define environmental variables for the following scopes:

Global

A site administrator for your Cloudera Data Science Workbench deployment can set environmental variables on a global level. These values will apply to every project on the deployment.

To set global environmental variables, go to [Admin Engines](#).

Project

Project administrators can set project-specific environmental variables to customize the engines launched for a project. Variables set here will override the global values set in the site administration panel.

To set environmental variables for a project, go to the project's Overview page and click [Settings Engine](#).

Job

Environments for individual jobs within a project can be customized while creating the job. Variables set per-job will override the project-level and global settings.

To set environmental variables for a job, go to the job's Overview page and click [Settings Set Environmental Variables](#).

Experiments

Engines created for execution of experiments are completely isolated from the project. However, these engines inherit values from environmental variables set at the project-level and/or global level. Variables set at the project-level will override the global values set in the site administration panel.

Models

Model environments are completely isolated from the project. Environmental variables for these engines can be configured during the build stage of the model deployment process. Models will also inherit any environment variables set at the project and global level. However, variables set per-model build will override other settings.



Important: Note that custom mounts or environment variables configured in `cdsw.conf` (such as `NO_PROXY`, `HTTP(S)_PROXY`, etc.) are still not passed to the container builds for experiments and models (even though they are applied to sessions, jobs, and deployed models/experiments).

Environment Variables from Cloudera Manager

In addition to the environment variables that you can specify with different scopes, Cloudera Data Science Workbench inherits a set of environment variables from Cloudera Manager.

Cloudera Data Science Workbench inherits the following set of environment variables from Cloudera Manager:

- HTTP_PROXY
- HTTPS_PROXY
- ALL_PROXY
- NO_PROXY

For information about what these variables are used for, see [Configuring Cloudera Data Science Workbench Deployments Behind a Proxy](#).

Site and project administrators can change these values by manually modifying them at the project or global level. The values set within Cloudera Data Science Workbench take precedence over the ones inherited from Cloudera Manager.

Accessing Environmental Variables from Projects

Environmental variables are injected into every engine launched for a project, contingent on the scope at which the variable was set (global, project, etc.).

The following code samples show how to access a sample environment variable called *DATABASE_PASSWORD* from your project code.

R

```
database.password <- Sys.getenv( "DATABASE_PASSWORD" )
```

Python

```
import os
database_password = os.environ[ "DATABASE_PASSWORD" ]
```

Scala

```
System.getenv( "DATABASE_PASSWORD" )
```

Appending Values to Environment Variables:

You can also set environment variables to append to existing values instead of replacing them. For example, when setting the LD_LIBRARY_PATH variable, you can set the value to LD_LIBRARY_PATH:/path/to/set.

Engine Environment Variables

The following table lists Cloudera Data Science Workbench environment variables that you can use to customize your project environments. These can be set either as a site administrator or within the scope of a project or a job.

Environment Variable	Description
MAX_TEXT_LENGTH	Maximum number of characters that can be displayed in a single text cell. By default, this value is set to 800,000 and any more characters will be truncated. Default: 800,000
SESSION_MAXIMUM_MINUTES	Maximum number of minutes a session can run before it times out. Default: 60*24*7 minutes (7 days) Maximum Value: 35,000 minutes

Environment Variable	Description
JOB_MAXIMUM_MINUTES	Maximum number of minutes a job can run before it times out. Default: 60*24*7 minutes (7 days) Maximum Value: 35,000 minutes
IDLE_MAXIMUM_MINUTES	Maximum number of minutes a session can remain idle before it exits. An idle session is defined as no browser interaction with the Editor. Terminal interactions are not considered as such. Contrast this to SESSION_MAXIMUM_MINUTES which is the total time the session is open, regardless of browser interaction. This variable is effective only when using the Workbench or the Jupyterlab editor. When using Cloudera's Jupyterlab Runtimes, the Editor itself is automatically configured to exit after idling for IDLE_MAXIMUM_MINUTES minutes by setting the MappingKernelManager.cull_idle_timeout and TerminalManager.cull_inactive_timeout Jupyterlab parameters accordingly. Sessions using custom Editors or the PBJ Workbench Editor do not exit due to idling. Default: 60 minutes Maximum Value: 35,000 minutes
CONDA_DEFAULT_ENV	Points to the default Conda environment so you can use Conda to install/manage packages in the Workbench. For more details on when to use this variable, see Using Conda with Cloudera Data Science Workbench .

Per-Engine Environmental Variables: In addition to the previous table, there are some more built-in environmental variables that are set by the Cloudera Data Science Workbench application itself and do not need to be modified by users. These variables are set per-engine launched by Cloudera Data Science Workbench and only apply within the scope of each engine.

Environment Variable	Description
CDSW_PROJECT	The project to which this engine belongs.
CDSW_ENGINE_ID	The ID of this engine. For sessions, this appears in your browser's URL bar.
CDSW_MASTER_ID	If this engine is a worker, this is the CDSW_ENGINE_ID of its master.
CDSW_MASTER_IP	If this engine is a worker, this is the IP address of its master.
CDSW_PUBLIC_PORT	Note: This property is deprecated as of Cloudera Data Science Workbench 1.6.0. See CDSW_APP_PORT and CDSW_READONLY_PORT for alternatives. A port on which you can expose HTTP services in the engine to browsers. HTTP services that bind CDSW_PUBLIC_PORT will be available in browsers at: <code>http(s)://<CDSW_ENGINE_ID>.<CDSW_DOMAIN></code> . By default, CDSW_PUBLIC_PORT is set to 8080. Setting CDSW_PUBLIC_PORT to a non-default port number is not supported. A direct link to these web services will be available from the grid icon in the upper right corner of the Cloudera Data Science Workbench web application, as long as the job or session is still running. For more details, see Accessing Web User Interfaces from Cloudera Data Science Workbench .

Environment Variable	Description
CDSW_APP_PORT	<p>A port on which you can expose HTTP services in the engine to browsers. HTTP services that bind CDSW_APP_PORT will be available in browsers at: <code>http(s)://<\$CDSW_ENGINE_ID>.<\$CDSW_DOMAIN></code>. Use this port for applications that grant some control to the project, such as access to the session or terminal. Use 127.0.0.1 as the IP.</p> <p>A direct link to these web applications will be available from the grid icon in the upper right corner of the Cloudera Data Science Workbench web application as long as the job or session runs. Even if the web application itself does not have authentication enabled, only project Contributors and Admins will be able to access it. For more details, see Accessing Web User Interfaces from Cloudera Data Science Workbench.</p> <p>Note that if the Site Administrator has enabled Allow only session creators to execute commands on active sessions, then the UI is only available to the session creator. Other users will not be able to access it.</p>
CDSW_READONLY_PORT	<p>A port on which you can expose HTTP services in the engine to browsers. HTTP services that bind CDSW_READONLY_PORT will be available in browsers at: <code>http(s)://<\$CDSW_ENGINE_ID>.<\$CDSW_DOMAIN></code>. Use this port for applications that grant read-only access to project results. Use 127.0.0.1 as the IP.</p> <p>A direct link to these web applications will be available to users with from the grid icon in the upper right corner of the Cloudera Data Science Workbench web application as long as the job or session runs. Even if the web application itself does not have authentication enabled, only project collaborators will be able to access the application. For more details, see Accessing Web User Interfaces from Cloudera Data Science Workbench.</p>
CDSW_DOMAIN	The domain on which Cloudera Data Science Workbench is being served. This can be useful for iframing services, as demonstrated in the Shiny example .
CDSW_CPU_MILLICORES	The number of CPU cores allocated to this engine, expressed in thousandths of a core.
CDSW_MEMORY_MB	The number of megabytes of memory allocated to this engine.
CDSW_IP_ADDRESS	Other engines in the Cloudera Data Science Workbench cluster can contact this engine on this IP address.