

Applications

Date published: 2020-07-16

Date modified: 2024-07-31

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

Analytical Applications.....	4
Securing Applications.....	5
Limitations with Analytical Applications.....	6
Monitoring applications.....	6

Analytical Applications

This topic describes how to use a to create long-running web applications.

About this task:

This feature gives data scientists a way to create web applications/dashboards and easily share them with other business stakeholders. Applications can range from single visualizations embedded in reports, to rich dashboard solutions such as Tableau. They can be interactive or non-interactive.

Applications stand alongside other existing forms of workloads in (sessions, jobs, experiments, models). Like all other workloads, applications must be created within the scope of a project. Each application is launched within its own isolated engine. Additionally, like models, engines launched for applications do not time out automatically. They will run as long as the web application needs to be accessible by any users and must be stopped manually when needed.

Before you begin:

Testing applications before you deploy

Before you deploy an application using the steps described here, make sure your application has been thoroughly tested. You can use sessions to develop, test, and debug your applications. You can test web apps by embedding them in sessions as described here: [Web Applications Embedded in Sessions](#).

For CML UI

1. Go to a project's **Overview** page.
2. Click Applications.
3. Click New Application.
4. Fill out the following fields.

- **Name:** Enter a name for the application.
- **Run Application as:** If the application is to run in a service account, select Service Account and choose the account from the dropdown menu.
- **Subdomain:** Enter a subdomain that will be used to construct the URL for the web application. For example, if you use test-app as the subdomain, the application will be accessible at test-app.<ml-workspace-domain-name>.

Subdomains should be valid DNS hostname characters: letters from a to z, digits from 0 to 9, and the hyphen.

- **Description:** Enter a description for the application.
- **Script:** Select a script that hosts a web application on either CDSW_READONLY_PORT or CDSW_APP_PORT. Applications running on either of these ports are available to any users with at least read access to the project. The Python template project includes an entry.py script that you can use to test this out.



Note: does not prevent you from running an application that allows a read-only user (i.e. Viewers) to modify files belonging to the project. It is up to you to make the application truly read-only in terms of files, models, and other resources belonging to the project.

- **Engine Kernel and Resource Profile:** Select the kernel and computing resources needed for this application.
- **Set Environment Variables:** Click Set Environment Variables, enter the name and value for the new application variable, and click Add.

If there is a conflict between the project-level and application-level environment variables, the application-level environment variables override the project-level environment variables.

5. Click Create Application.

For CML APIv2

To create an application using the API, refer to this example:

Here is an example of using the Application API.

```
application_request = cmlapi.CreateApplicationRequest(  
    name = "application_name",  
    description = "application_description",  
    project_id = project_id,  
    subdomain = "application-subdomain",  
    kernel = "python3",  
    script = "entry.py",  
    environment = {"KEY": "VAL"}  
)  
app = client.create_application(  
    project_id = project_id,  
    body = application_request  
)
```

Results:

In a few minutes, you should see the application status change to **Running** on the **Applications** page. Click on the name of the application to access the web application interface.

What to do next:

You can Stop, Restart, or Delete an application from the **Applications** page.

If you want to make changes to an existing application, click Overview under the application name. Then go to the Settings tab to make any changes and update the application.

Securing Applications

You can provide access to Applications via either the CDSW_APP_PORT or the CDSW_READONLY_PORT. Any user with read or higher permissions to the project is able to access an application served through either port.

- Securing project resources

CML applications are accessible by any user with read-only or higher permissions to the project. The creator of the application is responsible for managing the level of permissions the application users have on the project through the application. CML does not actively prevent you from running an application that allows a read-only user (i.e. Viewers) to modify files belonging to the project.

- Public Applications

By default, authentication for applications is enforced on all ports and users cannot create public applications. If desired, the Admin user can allow users to create public applications that can be accessed by unauthenticated users.

To allow users to create public applications on an ML workspace:

1. As an Admin user, turn on the feature flag in Admin Security by selecting Allow applications to be configured with unauthenticated access.
2. When creating a new application, select Enable Unauthenticated Access.
3. For an existing application, in Settings select Enable Unauthenticated Access.

To prevent all users from creating public applications, go to Admin Security and deselect Allow applications to be configured with unauthenticated access. After one minute, all existing public applications stop being publicly accessible.

- Transparent Authentication

CML can pass user authentication to an Application, if the Application expects an authorized request. The REMOTE-USER field is used for this task.

Limitations with Analytical Applications

This topic lists all the limitations associated with the Applications feature.

- Port availability

Cloudera Machine Learning exposes only 2 ports per workload. Therefore, you can run a maximum of 2 web applications simultaneously, on these ports:

- CDSW_APP_PORT
- CDSW_READONLY_PORT

By default, third-party browser-based editors run on CDSW_APP_PORT. Therefore, for projects that are already using browser-based editors, you are left with only one other port to run applications on: CDSW_READONLY_PORT.

Monitoring applications

You can monitor the CPU and memory usage of deployed applications.

To view the Monitoring Applications UI, in Applications, select the Monitoring Applications icon.

The Monitoring Applications UI provides the following information to help you monitor application health. The information is shown both as an instantaneous value and as a time series.

- CPU Usage: Shows the cpu usage as a percentage of total available CPU resources, and as amount of vCPU.
- Memory Usage: Shows the memory usage as a percentage of total available memory, and as amount of memory.

You can also select to view a specific application or all applications, and also select the time period to view.