Web Applications Embedded in Cloudera Data Science Workbench

Date published: 2020-02-28
Date modified:

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

Web Applications Embedded in Cloudera Data Science Workbench.....4
Spark 2 Web UIs (CDSW_SPARK_PORT).....4
TensorBoard, Shiny, and others (CDSW_APP_PORT or CDSW_READONLY_PORT).....4
Limitations with Port Availability.....5
Example: A Shiny Application.....5

Web Applications Embedded in Cloudera Data Science Workbench

This topic describes how Cloudera Data Science Workbench allows you to embed web applications for frameworks such as Spark 2, TensorFlow, Shiny, and so on within sessions.

Many data science libraries and processing frameworks include user interfaces to help track progress of your jobs and break down workflows. These are instrumental in debugging and using the platforms themselves. For example, Spark provides a Spark Web UI to monitor running applications and TensorFlow visualizations can be run on TensorBoard. Other web application frameworks such as Shiny and Flask are popular ways for data scientists to display additional interactive analysis in the languages they already know.

Cloudera Data Science Workbench allows you to access these web UIs directly from sessions and jobs. This feature is particularly helpful when you want to monitor and track progress for batch jobs. Even though jobs don't give you access to the interactive workbench console, you can still track long running jobs through the UI. However, note that the UI is only active so long as the job/session is active. If your session times out after 60 minutes (default timeout value), so will the UI.

Note: If you want to share your web application as a long-running standalone application that other business users can access, Cloudera recommends you now use the Applications feature to support long-running web applications on ML workspaces. If you are only running a server-backed visualization as part of your own analysis, then you can continue to keep embedding web applications in sessions as described in this topic. Note that running web applications in sessions is also the recommended way to develop, test, and debug analytical apps before deployment.

Cloudera Data Science Workbench exposes web applications for Spark and other machine learning frameworks as described here.

Spark 2 Web UIs (CDSW_SPARK_PORT)

Spark 2 exposes one web UI for each Spark application driver running in Cloudera Data Science Workbench. The UI will be running within the container, on the port specified by the environmental variable CDSW_SPARK_PORT.

By default, CDSW_SPARK_PORT is set to 20049. The web UI will exist only as long as a SparkContext is active within a session. The port is freed up when the SparkContext is shutdown.

Spark 2 web UI is available as a tab in the session, or alternatively in browsers at https://spark-< $*CDSW_ENGINE_ID*>".<$*CDSW_DOMAIN*>. For a running job, navigate to the **Job Overview** page and click the History tab. Click on the running job and select Spark UI.

TensorBoard, Shiny, and others (CDSW_APP_PORT or CDSW_READONLY_PORT)

CDSW_APP_PORT and CDSW_READONLY_PORT are environment variables that point to general purpose public ports.

Note: Previously, this section referred to CDSW_PUBLIC_PORT, which is deprecated as of Cloudera Data Science Workbench 1.6.0. Based on your use case, use CDSW_APP_PORT or CDSW_READONLY_PORT instead. However, the example below still uses CDSW_PUBLIC_PORT.

HTTP services running in containers that bind to CDSW_APP_PORT, CDSW_READONLY_PORT, and CDSW_PUBLIC_PORT are available in browsers at the following urls:

- CDSW_APP_PORT is available at

```
https://<$CDSW_ENGINE_ID>.<$CDSW_DOMAIN>
```

- CDSW_READONLY_PORT is available at

```
https://read-only-<$CDSW_ENGINE_ID>.<$CDSW_DOMAIN>
```

- CDSW_PUBLIC_PORT is available at

```
https://public-<$CDSW_ENGINE_ID>.<$CDSW_DOMAIN>
```

Therefore, TensorBoard, Shiny, Flask or any other web framework accompanying a project can be accessed directly from within a session or job, as long as it is run on CDSW_APP_PORT or CDSW_READONLY_PORT.

CDSW_APP_PORT is meant for applications that grant some level of control to the project, such as access to the active session or terminal. CDSW_READONLY_PORT must be used for applications that grant read-only access to project results.

The host address should be localhost.

To access the UI while you are in an active session, click the grid icon in the upper right hand corner of the Cloudera Data Science Workbench web application, and select the UI from the dropdown. For a job, navigate to the job overview page and click the History tab. Click on a job run to open the session output for the job. You can now click the grid icon in the upper right hand corner of the Cloudera Data Science Workbench web application to access the UI for this session.

Limitations with Port Availability

Cloudera Data Science Workbench exposes only one port per-access level.

In version 1.6.0, you can run a maximum of 3 web applications simultaneously.

- One on CDSW_APP_PORT, which can be used for applications that grant some level of control over the project to Contributors and Admins,
- One on CDSW_READONLY_PORT, which can be used for applications that only need to give read-only access to project collaborators,
- One on the now-deprecated CDSW_PUBLIC_PORT, which is accessible by all users.

However, by default the editors feature (introduced in version 1.6) runs third-party browser-based editors on CDSW_APP_PORT. Therefore, for projects that are already using browser-based third-party editors, you are left with only 2 other ports to run applications on: CDSW_READONLY_PORT and CDSW_PUBLIC_PORT. Keep in mind the level of access you want to grant users when you are selecting one of these ports for a web application.

Example: A Shiny Application

This example demonstrates how to create and run a Shiny application and view the associated UI while in an active session.

Create a new, blank project and run an R console. Create the files, ui.R and server.R, in the project, and copy the contents of the following example files provided by Shiny by RStudio:

R

```
# ui.R

library(shiny)

# Define UI for application that draws a histogram
shinyUI(fluidPage(
```

```
  # Application title
  titlePanel("Hello Shiny!"),

  # Sidebar with a slider input for the number of bins
  sidebarLayout(
    sidebarPanel(
      sliderInput("bins",
                  "Number of bins:",
                  min = 1,
                  max = 50,
                  value = 30)
    ),

    # Show a plot of the generated distribution
    mainPanel(
      plotOutput("distPlot")
    )
  )
))
```

R

```
# server.R
library(shiny)
# Define server logic required to draw a histogram
shinyServer(function(input, output) {

  # Expression that generates a histogram. The expression is
  # wrapped in a call to renderPlot to indicate that:
  #
  #  1) It is "reactive" and therefore should re-execute automatically
  #     when inputs change
  #  2) Its output type is a plot

  output$distPlot <- renderPlot({
    x    <- faithful[, 2]  # Old Faithful Geyser data
    bins <- seq(min(x), max(x), length.out = input$bins + 1)
    # draw the histogram with the specified number of bins
    hist(x, breaks = bins, col = 'darkgray', border = 'white')
  })
})
```

Run the following code in the interactive workbench prompt to install the Shiny package, load the library into the engine, and run the Shiny application.

R

```
install.packages('shiny')

library('shiny')

runApp(port=as.numeric(Sys.getenv("CDSW_PUBLIC_PORT")), host="127.0.0.1",
launch.browser="FALSE")
```

Finally, to access the web application, either:

- Click the grid icon in the upper right hand corner of the Cloudera Data Science Workbench web application, and select the Shiny UI, Hello Shiny!, from the dropdown.
- Access the web application directly by visiting the URL: https://public-[session-id].[CML host]/

The UI will be active as long as the session is still running.