

DSCI-272: Predicting with Cloudera Machine Learning

None

Table of contents

DSCI-272: Predicting with Cloudera Machine Learning	4
Exercises	4
Introduction to CML on CDP	5
Login to Cloudera Data Platform	5
Overview of CML Workspace	7
Project and Session Overview	18
Streamlit on CML	33
Go to CML Workspace	34
Create a New Project Using the Streamlit AMP	35
Explore the Streamlit Project	39
Modify the Application	47
Delete the Project	48
Bonus	49
Data - Access, Audit, and Mask	50
Access Data	50
Visualize Duocar Data	60
Create Workload Password (if needed)	66
Create a New Dataset	78
Create a Dashboard	80
Create Filters	101
Share the Dashboard	105
Experiment Tracking	107
Create a New Project from Github	107
View Code and Run Job	111
Creating an Experiment	119
Change the Input and Compare Runs	128
Commit the Changes to Git	133
Using Workbench for Lecture and Exercises	138
Autoscaling, Performance, and GPU Settings	148
View Workspace Details and Allocated GPUs	148
Create a New Project	150
Create a Session without a GPU	152
Create a Session with a GPU	155
Continuous Model Monitoring with Evidently	162
Start the Continuous Model Monitoring AMP	162

Launch the Price Regressor Monitoring dashboard	166
Explore drift and variations in the model performance	170
Identify the file that creates the Evidently dashboard	171

DSCI-272: Predicting with Cloudera Machine Learning

Exercises

- [Introduction to CML on CDP](#)
- [Streamlit on CML](#)
- [Data - Access, Audit, and Mask](#)
- [Experiment Tracking](#)
- [Visualize Duocar Data](#)
- [Workbench Lecture and Exercises](#)
- [Autoscaling, Performance, and GPU Settings](#)
- [Continuous Model Modeling with Evidently AI](#)

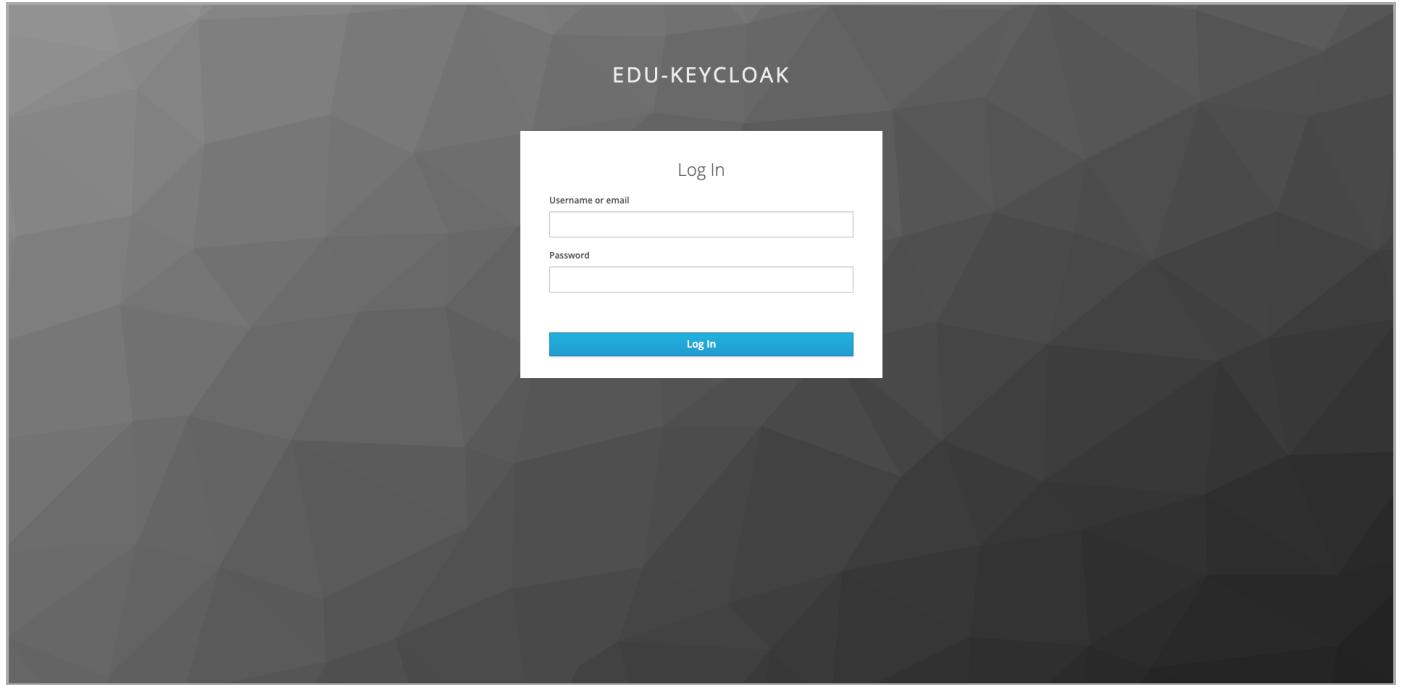
Introduction to CML on CDP

This exercise will introduce the Cloudera Data Platform (CDP) and Cloudera Machine Learning (CML). In this exercise, you will:

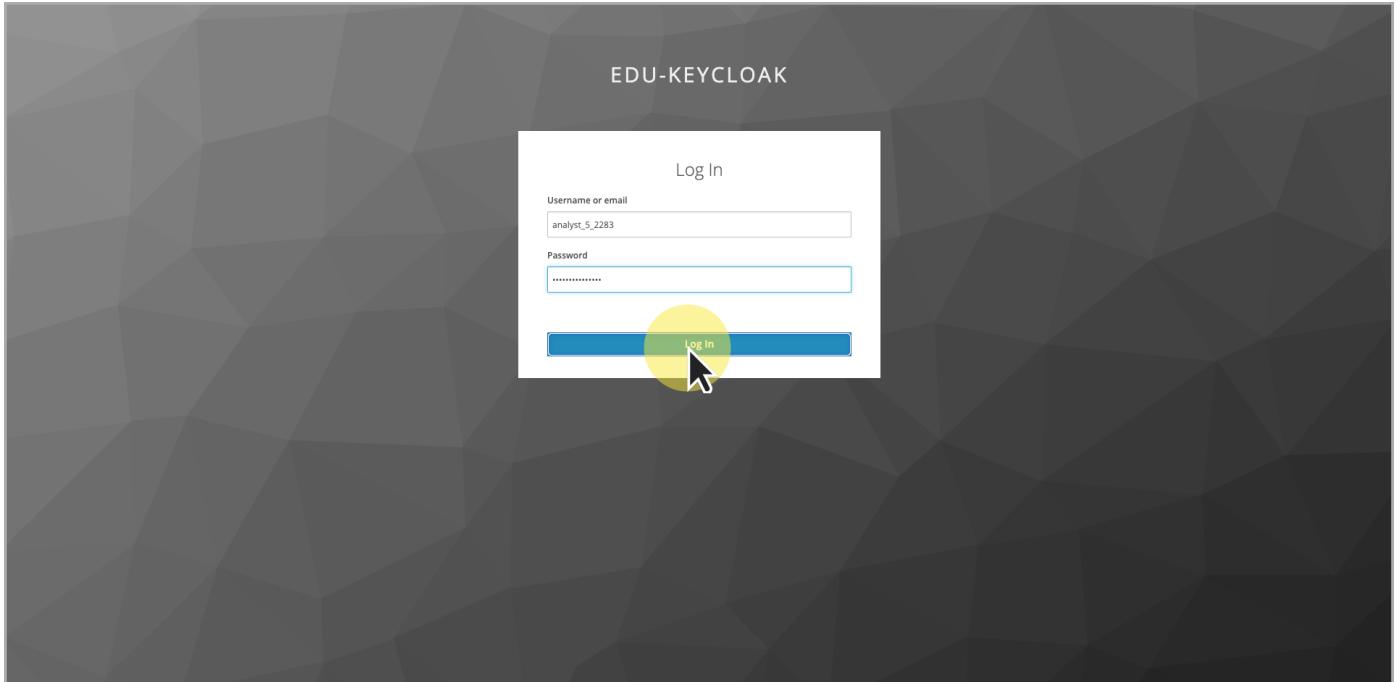
- login to the Cloudera Data Platform exercise environment,
- view the Cloudera Machine Learning workspace,
- create a new CML project,
- create a new session,
- and delete a CML project.

Login to Cloudera Data Platform

1. Open the [Log In page for the CDP Public Cloud environment](#)



2. Enter the username and password provided by your instructor. Click **Log In**.



3. The Cloudera Data Platform page is displayed. Some of the additional features of the platform will be explored in other exercises. For now, click **Machine Learning**.

A screenshot of the Cloudera Data Platform homepage. At the top left is the "CLOUDERA Data Platform" logo. The main title "Your Enterprise Data Cloud" is centered above a grid of icons. The grid includes: "Data Hub Clusters" (camera icon), "DataFlow" (wave icon), "Data Engineering" (target icon), "Data Warehouse" (database icon), "Operational Database" (clock icon), and "Machine Learning" (brain icon, highlighted with a yellow circle and a cursor). Below this is a section titled "Control Plane" with icons for "Data Catalog" (catalog icon), "Replication Manager" (replica icon), "Workload Manager" (graph icon), and "Management Console" (management icon). At the bottom left is a user status bar showing "User 5". At the bottom right is the text "Powered by Cloudera".

Overview of CML Workspace

- The list of Machine Learning workspaces is displayed. In this case, one workspace is provisioned. Click on **cml-on-cdp**.

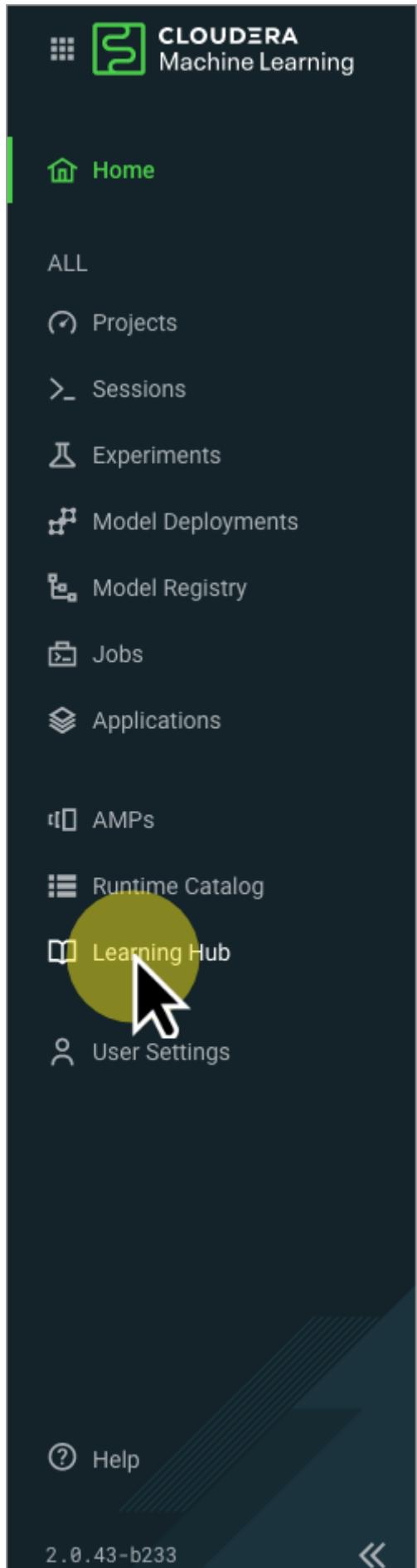
The screenshot shows the 'Machine Learning Workspaces' page. On the left is a sidebar with options like 'Workspaces', 'Workspace Backups', 'Model Registries', 'Help', and 'User 10'. The main area has a table titled 'Machine Learning Workspaces' with columns: Status, Version, Workspace, Environment, Region, Creation Date, Cloud Provider, and Actions. A single row is shown: 'Ready' status, '2.0.43' version, 'cml-on-cdp' workspace name (highlighted with a yellow circle), 'All' environment, 'us-east-2' region, '03/15/2024 12:46 PM CDT' creation date, 'aws' cloud provider, and 'AWS' actions. At the bottom right of the table, it says 'Displaying 1 - 1 of 1 < 1 > 25 / page ▾'.

- The Projects page is displayed. The Projects page lists all of the projects in the workspace. (Your screen will vary from the screen shown below.)

The screenshot shows the 'Home' page of the Cloudy Machine Learning interface. The left sidebar includes 'ALL', 'Projects', 'Sessions', 'Experiments', 'Model Deployments', 'Model Registry', 'Jobs', 'Applications', 'AMPS', 'Runtime Catalog', 'Learning Hub', 'User Settings', 'Help', and a note about the workspace being 'aws (AWS)'. The main content area has sections for 'Recent Projects' (empty), 'Product Tour' (with 'Take a CML product tour', 'Explore Use Cases', and 'Enable exploratory Data Science' cards), 'Featured Announcements' (with cards for 'Model Registry is Generally Available!', 'AMP - Using Amazon Bedrock for Text Summarization and More', and 'AMP - Fine Tuning a Foundation Model for Multiple Tasks'), and 'Helpful Links' (with 'Documentation' and 'Community' links). At the bottom, it says 'Workspace: cml-on-cdp-dfheinz'.

The workspace has several helpful features.

1. Select **Learning Hub** from the workspace menu.



2. View the Learning Hub content. The Learning Hub is a great resource to learn about the new features in CML and read blog posts, research reports, and documentation.

Learning Hub

Featured Announcements

- Model Registry is Generally Available!** NEW
The Model Registry serves as a centralized hub for storing, managing, and deploying machine ...
October 22, 2023
- AMP - Using Amazon Bedrock for Text** NEW
Amazon Bedrock is a new AWS Cloud service which allows convenient api access to a num...
September 28, 2023
- AMP - Fine Tuning a Foundation Model for Fine Tuning LLMs** NEW
Fine Tuning LLMs using techniques like Parameter-Efficient Fine-Tuning (PEFT) and ...
September 14, 2023
- AMP - LLM Chatbot Augmented with Enterprise** NEW
This new AMP demonstrates how enterprises can seamlessly integrate their own ...
May 21, 2023

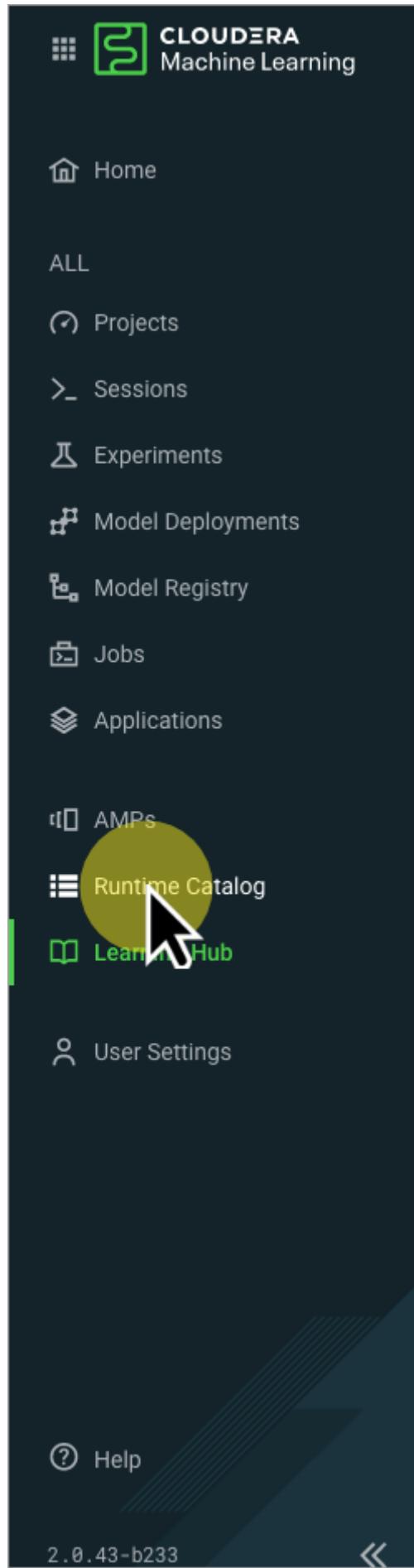
Research and Resources

Blog Posts **Research Reports** **Documentation**

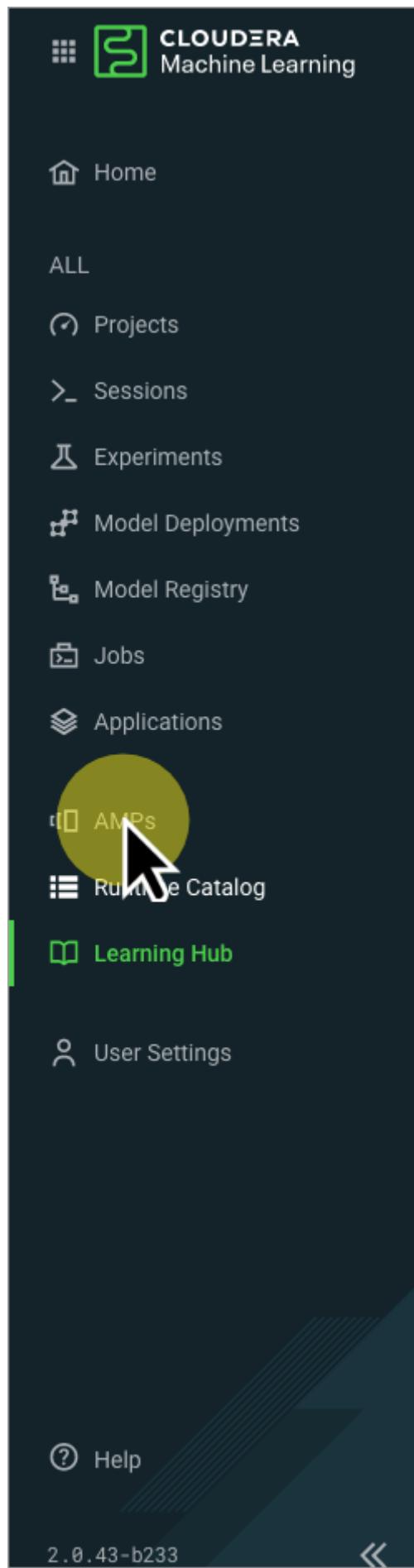
- Implementing CycleGAN**
This post we walk through how to implement CycleGAN to generate synthetic images to help train a deep learning model for detecting manufacturing defects on steel surfaces.
November 13, 2022
- How to Distribute Machine Learning Workloads with Dask**
Learn how to distribute your ML workload in Cloudera Machine Learning when your data is too big or your workload is too complex to run on a single machine.
October 2, 2022
- Ethics Sheet for AI-assisted Comic Book Art Generation**
This article is a simplified take on an ethics sheet for the task of AI-assisted comic book art generation, inspired by "Ethics Sheets for AI Tasks". In it, we take a look at some of the...
September 19, 2022
- Thought experiment: Human-centric machine learning for comic book creation**
Our newest research engineer, Mike Gallaspy, lightheartedly speculates on using machine learning techniques for creating comic book art in his first blog post. Take a peek and enjoy...
September 7, 2022

Workspace: cmi-on-cdp-dfheinz aws (AWS)
<https://ml-3248cf5e-a25.dfheinz.kfp-x0dh.cloudera.site/runtime-catalog>

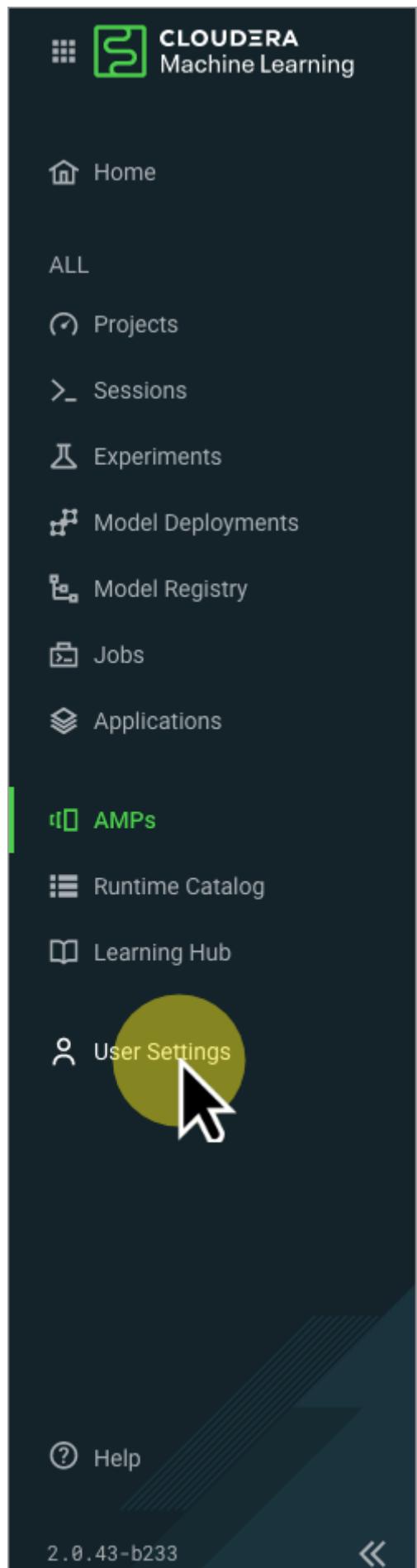
3. Select **Runtime Catalog**. The Runtime Catalog shows a list of available runtimes and allows new runtimes to be added.



4. Select **AMPs**. The AMPs page shows a list of available Applied ML Prototypes (AMPs).



5. Select **User Settings**.



6. Select the **Outbound SSH Key** tab. The Outbound SSH Key is used for connecting to external resources, for example Github.

The screenshot shows the 'User Settings' page with the 'Outbound SSH' tab highlighted. The 'User Public SSH Key' section displays a long string of text representing the public SSH key. A 'Reset SSH key' button is visible below the key text. The left sidebar contains links for Home, Projects, Sessions, Experiments, Model Deployments, Model Registry, Jobs, Applications, AMPS, Runtime Catalog, Learning Hub, and User Settings. The bottom status bar indicates a workspace of 'cmi-on-cdp-dfheinz' on 'aws (AWS)'.

Applications, Jobs, Models, Experiments, and Sessions will be covered later.

Project and Session Overview

Create a New Project

1. Select **Projects** from the workspace menu.

2. Click the **New Project** button.

The screenshot shows the 'Projects' page with the 'New Project' button highlighted. The central message says 'You currently don't have any projects'. Below it, a note explains that projects hold code, configuration, and libraries needed for reproducible runs. The left sidebar includes links for Home, Projects, Sessions, Experiments, Model Deployments, Model Registry, Jobs, Applications, AMPS, Runtime Catalog, Learning Hub, and User Settings. The bottom status bar shows the same workspace information as the previous screenshot.

3. For **Project Name**, enter `Student # - First Project`, where # is your student number.

4. Leave the following items set to their default values:

- **Description:** Empty
- **Project Visibility:** Private
- **Initial Setup:** Template - Python

5. Click Create Project.

New Project

Initial Setup

Templates include example code to help you get started.

Runtimes

Editor	Kernel	Edition	Version	Remove
JupyterLab	Python 3.10	Nvidia GPU	2024.02	<button>Remove</button>
JupyterLab	Python 3.10	Standard	2024.02	<button>Remove</button>
PBJ Workbench	Python 3.10	Nvidia GPU	2024.02	<button>Remove</button>
PBJ Workbench	Python 3.10	Standard	2024.02	<button>Remove</button>
PBJ Workbench	R 4.3	Standard	2024.02	<button>Remove</button>

Advanced Options

Create Project

Create a New Session

After the new project is created, the project overview page is displayed and the left-hand menu displays project items instead of workspace items. Next, create a session to interact with your project.

1. Click New Session.

sci_10_3678619 / Student 10 - First Project

Student 10 - First Project

Models
This project has no models yet. Create a [new model](#).

Jobs
This project has no jobs yet. Create a [new job](#) to document your analytics pipelines.

Files

Name	Size	Last Modified
seaborn-data	-	15 days ago
analysis.ipynb	241.55 kB	15 days ago
analysis.py	1.42 kB	15 days ago
cdsw-build.sh	100 B	15 days ago
config.yml	123 B	15 days ago
entry.py	276 B	15 days ago
fit.py	1.26 kB	15 days ago
lineage.yaml	1.12 kB	15 days ago
pi.py	739 B	15 days ago
predict.py	380 B	15 days ago
predict_with_metrics.py	1.49 kB	15 days ago
README.md	1.51 kB	15 days ago
requirements.txt	15 B	15 days ago
use_model_metrics.py	4.80 kB	15 days ago

Download New Upload

Workspace: cml-on-cdp-dfheinz aws_jackson

https://ml-3248cf5e-a25.dfehnz.kfp-x0dh.cloudera.site/sci_10_3678619/student-10-first-project/sessions/new

2. Enter First Session for the Session Name.

3. Select JupyterLab as the Editor.

sci_10_3678619 / Student 10 - First Project

Start A New Session

Session Name

Runtime

Editor Kernel Edition Version

Please select one
JupyterLab
PBJ Workbench

Runtime Image
Spark 2.4.8 - CDE 1.19.2 - HOTFIX-2

Resource Profile
2 vCPU / 4 GiB Memory | 0 GPUs

Cancel Start Session

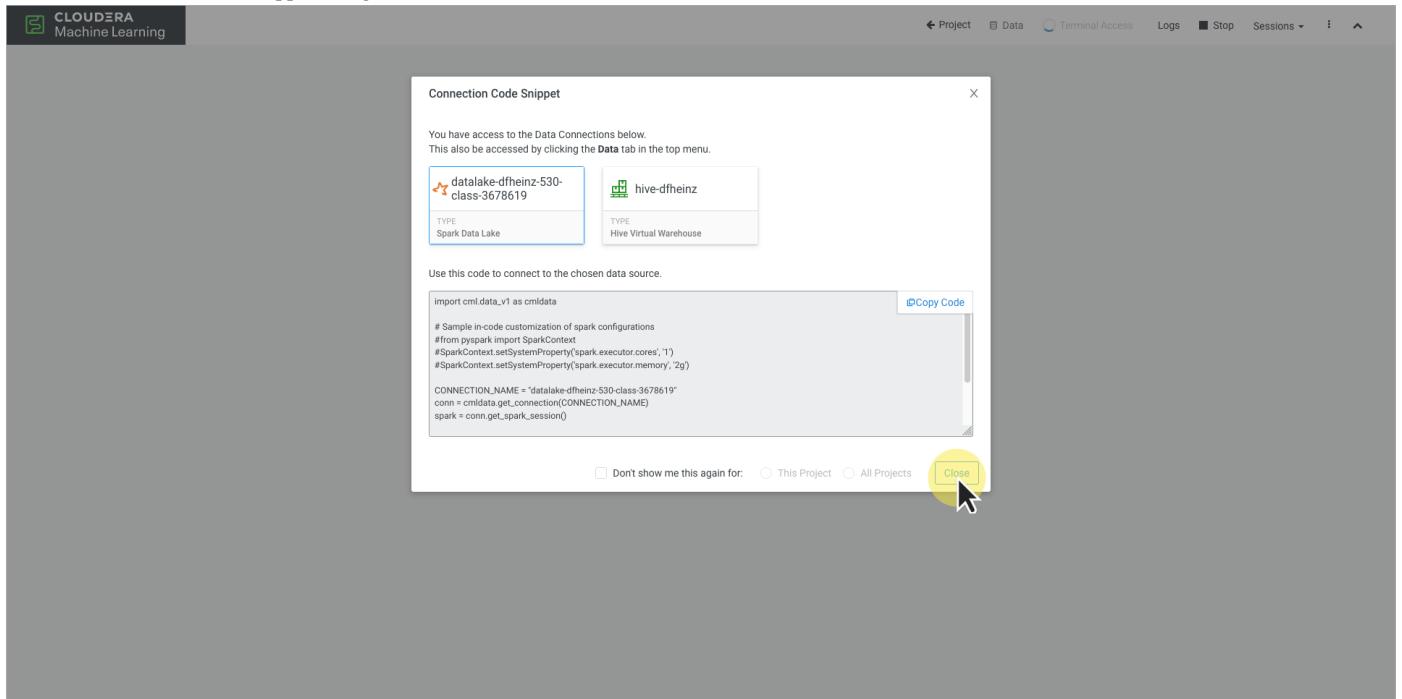
4. Select Standard as the Edition.

The screenshot shows the 'Start A New Session' dialog. In the 'Runtime' section, the 'Edition' dropdown is open, displaying 'Nvidia GPU' and 'Standard'. The 'Standard' option is highlighted with a yellow circle and has a cursor pointing at it. Other visible fields include 'Session Name' (set to 'First Session'), 'Kernel' (set to 'Python 3.10'), 'Version' (set to '2024.02'), and 'Resource Profile' (set to '2 vCPU / 4 GiB Memory').

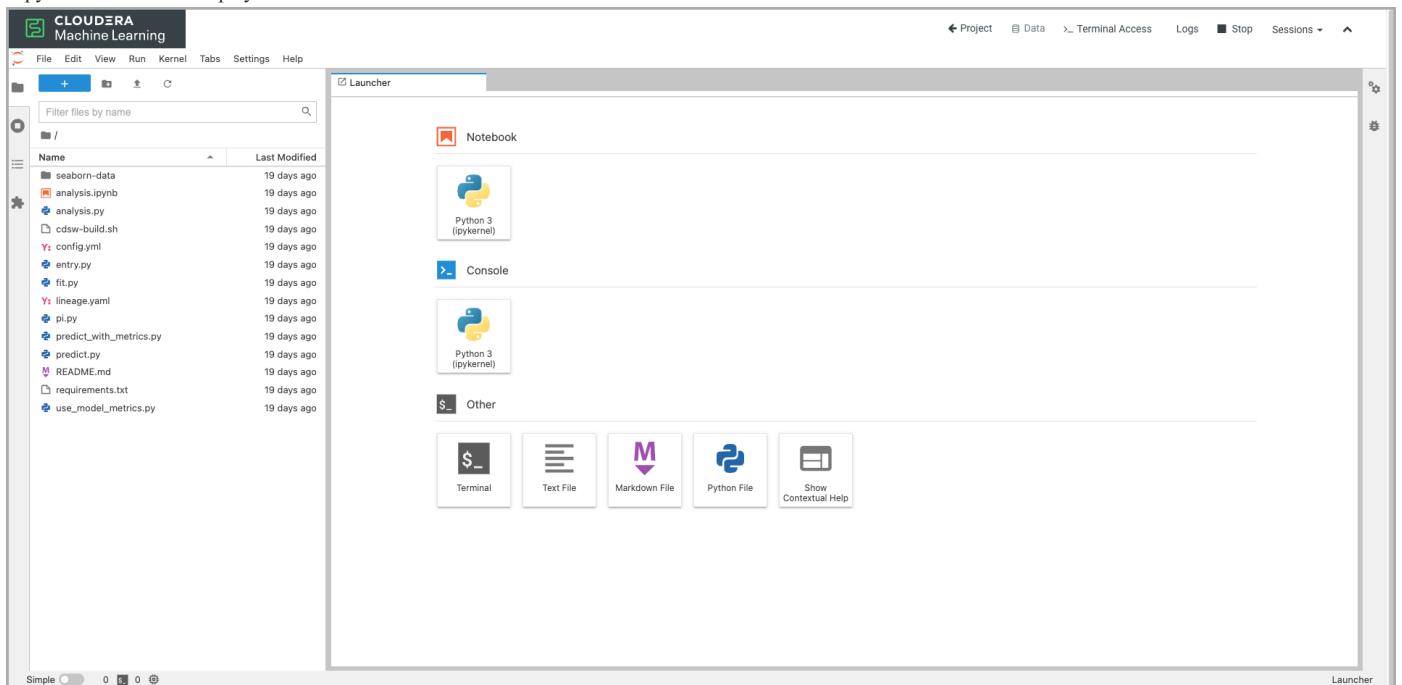
5. Click Start Session.

The screenshot shows the 'Start A New Session' dialog. The 'Edition' dropdown is set to 'Standard'. The 'Start Session' button at the bottom right is highlighted with a yellow circle and has a cursor pointing at it. Other fields are identical to the previous screenshot.

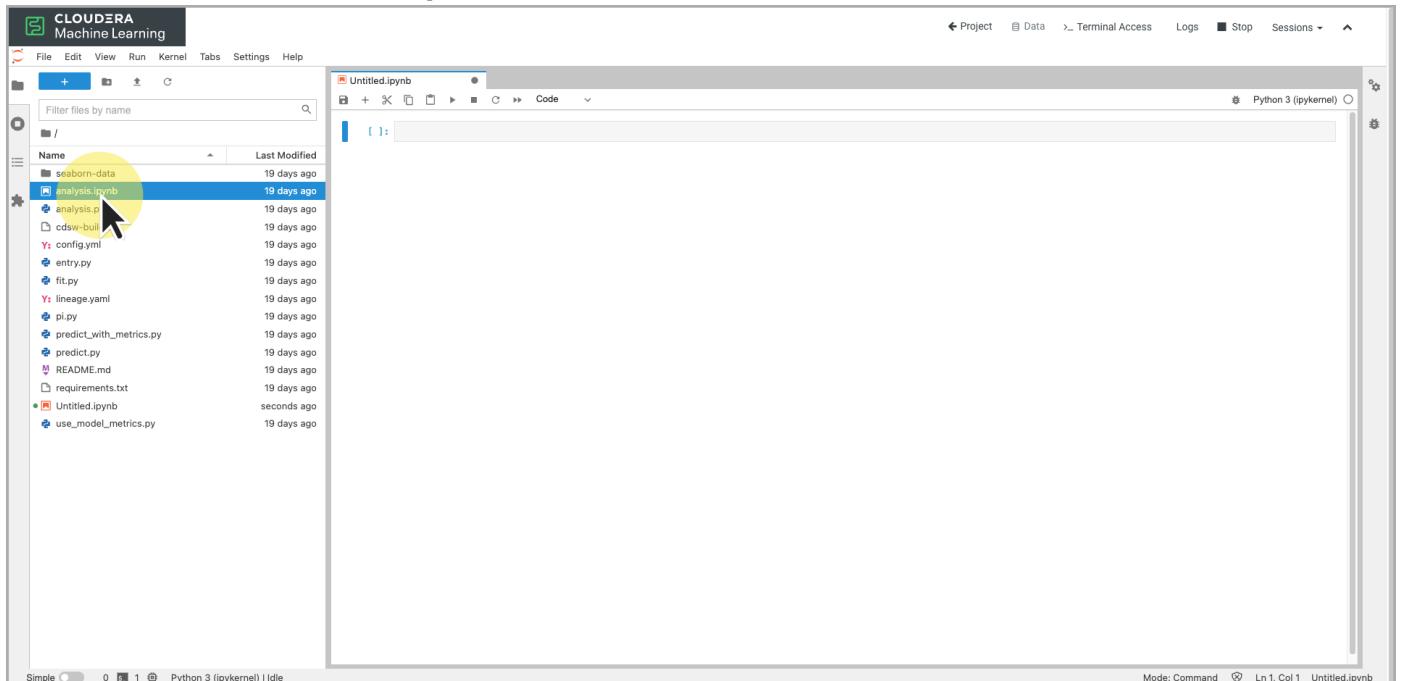
6. Close the Connection Code Snippet dialog.



7. JupyterLab in CML is displayed.



8. Double-click the `analysis.ipynb` notebook to open it.



Note

This exercise is just an introduction to creating a session that uses JupyterLabs. If you are familiar with JupyterLabs and would like to work through the `analysis.ipynb` notebook, be sure to uncomment `%pip install -r requirements.txt` in the first cell.

9. Click ← Project to leave the session and return to the project.

The screenshot shows the JupyterLab interface with the `analysis.ipynb` notebook open. The code editor contains the following code:

```

sns.set(font="DejaVu Sans")
sns.jointplot(data=tips, x="total_bill", y="tip", kind='reg').fig.suptitle("Tips Regression", y=1.01)

Examine the difference between smokers and non smokers

sns.lmplot(data=tips, x="total_bill", y="tip", col="smoker").fig.suptitle("Tips Regression - categorized by smoker", y=1.05)

Explore the dataframe

tips.head()

```

Below the code, there is a section titled "Using IPython's Rich Display System" with the following text and code:

IPython has a rich display system for interactive widgets.

```

from IPython.display import IFrame
from IPython.core.display import display

```

Define a google maps function.

```

def gmaps(query):
    url = "https://maps.google.com/maps?q={}&output=embed".format(query)
    display(IFrame(url, '700px', '450px'))
    gmaps("Golden Gate Bridge")

```

At the bottom, there is a section titled "Worker Engines" with the following text and code:

You can launch worker engines to distribute your work across a cluster. Uncomment the following to launch two workers with 2 cpu cores and 0.5GB memory each.

```

# import cds
# workers = cds.launch_workers(n=2, cpu=0.2, memory=0.5, code="print('Hello from a CDSW Worker')")

```

The status bar at the bottom shows the URL `https://ml-0fb1313c-d50.bshimel.kf.r-x0dh.cloudera.site/analyst_5_2283/student-5-first-project`, "Mode: Edit", "Ln 4, Col 1", and "analysis.ipynb".

10. Select **Sessions**.

The screenshot shows the left sidebar of the Cloudera Machine Learning application. At the top is the Cloudera logo and the text "CLOUDERA Machine Learning". Below this is a horizontal line of icons: Home, All Projects, Overview (which is highlighted in green), Sessions (which has a yellow circular highlight and a cursor arrow pointing to it), Data, Experiments, Model Deployments, Jobs, Applications, Files, Collaborators, Project Settings, AMPs, Runtime Catalog, Learning Hub, User Settings, and Help.

- Home
- All Projects
- PROJECT
- Overview
- Sessions
- Data
- Experiments
- Model Deployments
- Jobs
- Applications
- Files
- Collaborators
- Project Settings
- AMPs
- Runtime Catalog
- Learning Hub
- User Settings
- Help

2.0.43-b233

11. The list of sessions is displayed. Your session should be listed and show a status of **Running**.

The screenshot shows the Cloudera Machine Learning interface. On the left is a dark sidebar with various project and system navigation links. The main area is titled "sci_10_3678619 / Student 10 - First Project / Sessions". A search bar at the top right contains "Project quick find" and a user icon. Below it are buttons for "Stop Selected", "Delete Selected", and "New Session". A table lists sessions with columns: Status, Session, Kernel, Creator, Created At, and Duration. One row is highlighted in blue, showing "Running", "First Session", "(Python 3.10 JupyterLab Standard)", "User 10", "03/17/2024 2:01 PM", and "Running since 49s". Action buttons "Edit", "Stop", and "Delete" are to the right of this row. At the bottom of the table, it says "Displaying 1 - 1 < 1 > 25 / page". The bottom of the screen shows a workspace summary: "Workspace: cmr-on-cdp-dfheinz", "aws", and "AWS Lambda".

Status	Session	Kernel	Creator	Created At	Duration
Running	First Session	(Python 3.10 JupyterLab Standard)	User 10	03/17/2024 2:01 PM	Running since 49s

Delete a Project

1. Select Project Settings.

The screenshot shows the Cloudera Machine Learning interface with a dark theme. On the left is a sidebar containing the following items:

- Home
- All Projects
- PROJECT**
- Overview
- Sessions** (highlighted with a green background)
- Data
- Experiments
- Model Deployments
- Jobs
- Applications
- Files
- Collaborators
- Project Settings** (highlighted with a yellow circle and a cursor icon pointing at it)
- AMPs
- Runtime Catalog
- Learning Hub
- User Settings
- Help

At the bottom of the sidebar, the version is listed as 2.0.43-b233. In the bottom right corner of the sidebar, there is a double-left arrow icon.

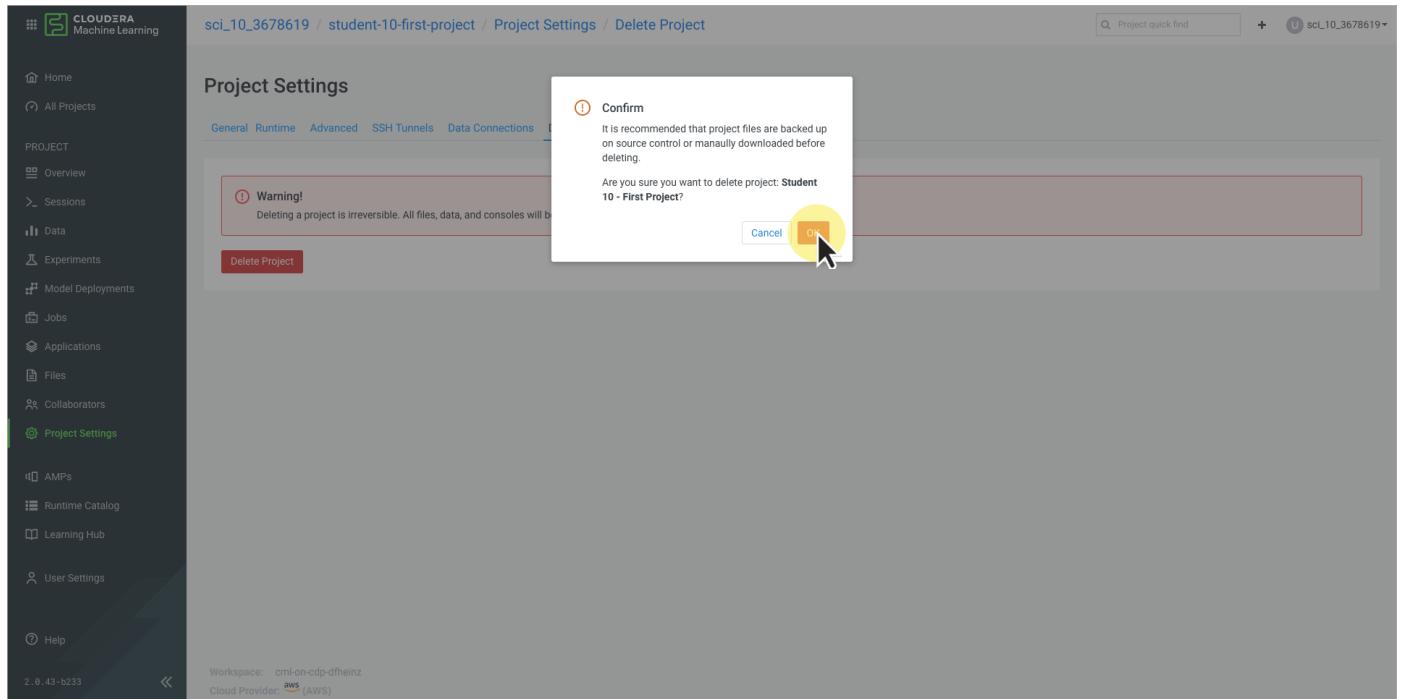
2. Select the Delete Project tab.

The screenshot shows the 'Project Settings' page for a project named 'Student 10 - First Project'. The 'Delete Project' tab is highlighted with a yellow circle and a cursor. The page includes fields for 'Project Name' (Student 10 - First Project), 'Project Description', 'Visibility' (Private selected), 'Project Owner' (User 10), and an 'Update Project' button. The URL in the address bar is https://ml-3248cf5e-a25.dfheinz.kfp-x0dh.cloudera.site/sci_10_3678619/student-10-first-project/settings/delete.

3. Click Delete Project.

The screenshot shows the 'Project Settings' page with a red warning box containing the text: 'Warning! Deleting a project is irreversible. All files, data, and consoles will be lost.' Below the warning, the 'Delete Project' button is highlighted with a yellow circle and a cursor. The page includes fields for 'Project Name' (Student 10 - First Project), 'Project Description', 'Visibility' (Private selected), 'Project Owner' (User 10), and an 'Update Project' button. The URL in the address bar is https://ml-3248cf5e-a25.dfheinz.kfp-x0dh.cloudera.site/sci_10_3678619/student-10-first-project/settings/delete. The workspace is 'cmi-on-cdp-dfheinz' and the cloud provider is 'aws (AWS)'.

4. Click **OK** to confirm.



End of Exercise

Streamlit on CML

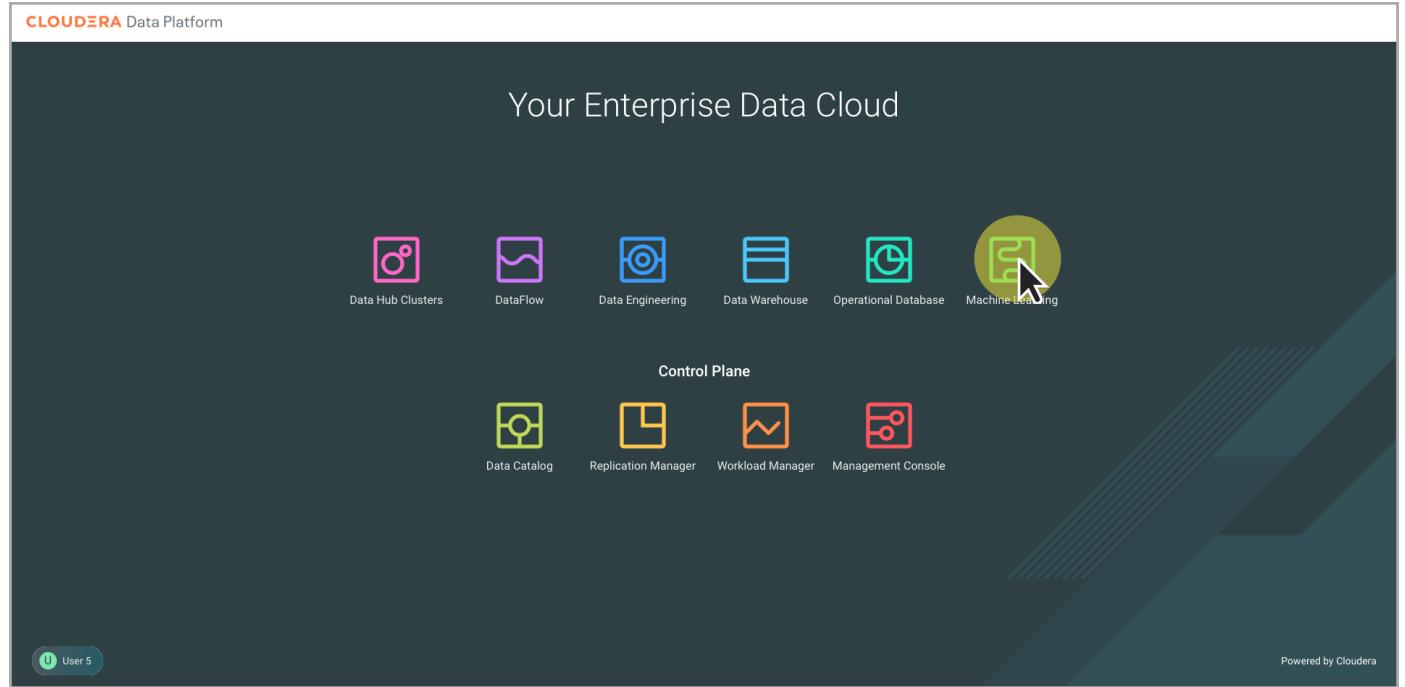
"[Streamlit](#) turns data scripts into shareable web apps in minutes." Streamlit is a web application framework that allows data scientists to quickly build web applications to share data and analytics.

In this exercise, you will use an AMP (Applied ML Prototype) to deploy a simple Streamlit application using CML. You will learn how to deploy an AMP and how applications work in CML.

Go to CML Workspace

Begin the exercise by navigating to your workspace.

1. Click Machine Learning.



2. Select the workspace.

The screenshot shows the "Machine Learning Workspaces" page. On the left is a sidebar with icons for "Workspaces" (selected) and "Workspace Backups". The main area has a header with "Machine Learning Workspaces", a search bar, and a "Provision Workspace" button. A table lists one workspace: "cml-on-cdp" (Status: Ready, Environment: bshimel-456-class-2283, Region: us-east-2, Creation Date: 08/04/2022 9:40 AM CDT, Cloud Provider: AWS). The "cml-on-cdp" row is highlighted with a yellow circle and a cursor arrow pointing at it. At the bottom right of the table are pagination controls: "Displaying 1 - 1 of 1" and "25 / page". The bottom left of the sidebar shows "Help" and "User 5".

Create a New Project Using the Streamlit AMP

Creating a project from an AMP is easy.

1. Click AMPs in the workspace menu.

The screenshot shows the Cloudera Machine Learning workspace interface. On the left, there is a sidebar with various navigation options: Projects, Sessions, Experiments, Models, Jobs, Applications, User Settings, AMPs (which is highlighted with a yellow circle), Runtime Catalog, Site Administration, and Learning Hub. Below the sidebar, it says 'dev' and has a 'Help' link. At the bottom, it shows 'Workspace: cml-on-cdp' and 'Cloud Provider: AWS (AWS)'. The main area is titled 'Projects' and shows 'Active Workloads' with counts for Sessions (0), Experiments (0), Models (0), Jobs (0), and Applications (3). It also displays 'User Resources' and 'Workspace Resources' for CPU, Memory, and GPU. A search bar at the top right says 'Project quick find' and shows 'analyst_5_2283'. A 'New Project' button is visible in the top right corner of the main area.

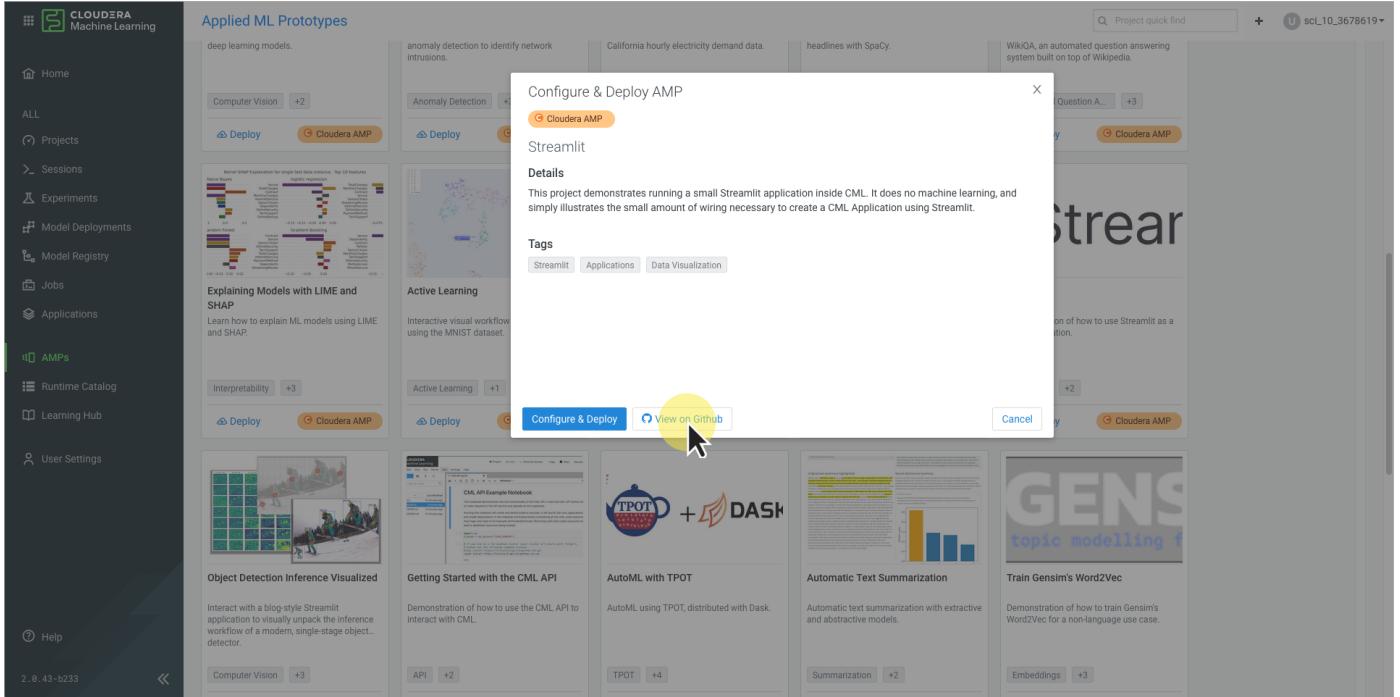
You will see a catalog of AMPs. New AMPs are being added all of the time. In addition to AMPs provided by Cloudera, you can [create your own AMPs](#) and your own [AMP catalog](#).

1. Click the Streamlit tile.

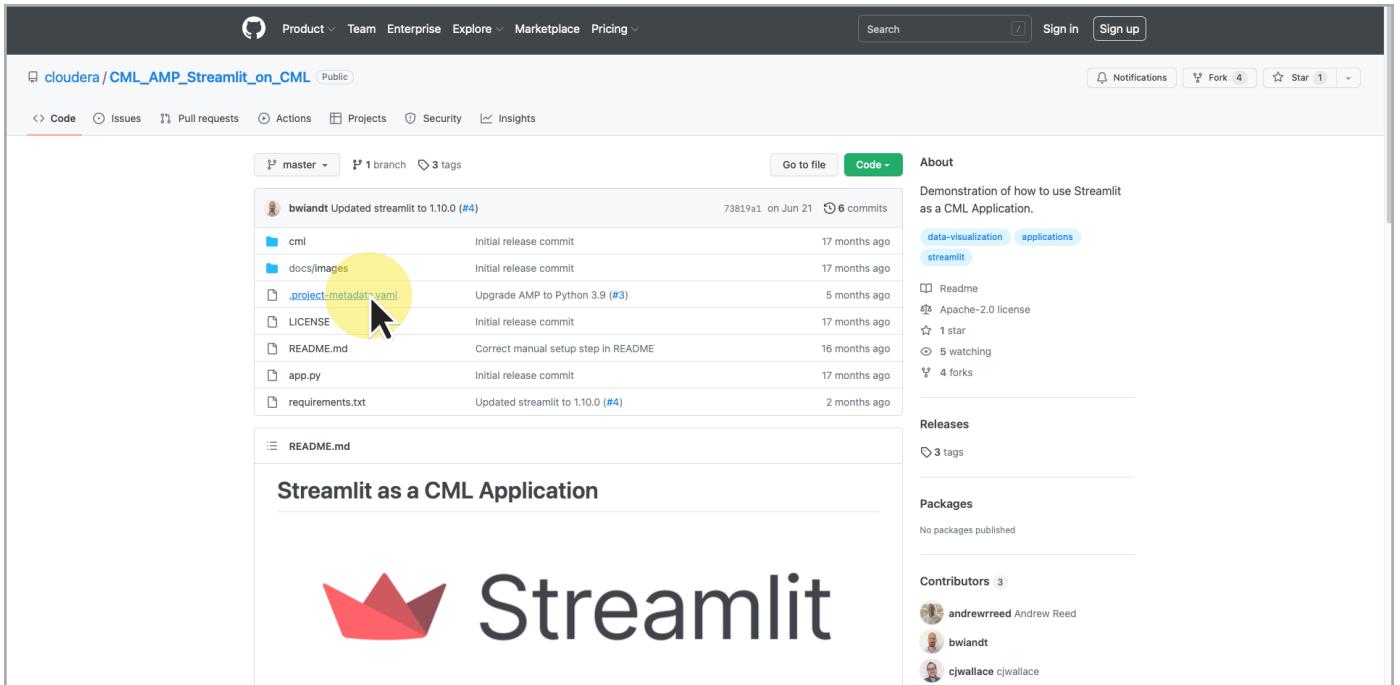
The screenshot shows the Cloudera Machine Learning workspace interface with the AMPs menu option selected. The main area displays a grid of AMP prototypes. One specific AMP, 'Strear', is highlighted with a yellow circle. The AMPs listed include: Applied ML Prototypes (deep learning models, Computer Vision, +2), anomaly detection to identify network intrusions (Anomaly Detection, +3), California hourly electricity demand data (Time Series, +2), headlines with SpaCy (SpaCy, +2), WikQA, an automated question answering system built on top of Wikipedia (Automated Question A..., +3), Canceled Flight Prediction (Binary Classification, +3), Streamlit (Streamlit, +2), Object Detection Inference Visualized (Computer Vision, +3), Getting Started with the CML API (API, +2), AutoML with TPOT (TPOT, +4), Automatic Text Summarization (Summarization, +2), and Train Gensim's Word2Vec (Embeddings, +3). Each AMP card includes a 'Deploy' button and a 'Cloudera AMP' badge.

A dialog will popup that describes the AMP. The dialog also contains a link to the Github repository for the AMP. Viewing an AMP's repository is a good way to learn more about the AMP and how AMPs are created, in general.

1. Click View on Github.



2. Click the `project-metadata.yaml` file.



In the `project-metadata.yaml` file, you can see the steps that are used to deploy the AMP. In this case, you can see that AMP runs a session to install the Python dependencies and then starts a new application called Streamlit App.

```
name: Streamlit
description: Run a Streamlit app inside CML.
author: Cloudera Inc.
specification_version: 1.0
prototype_version: 2.0
date: "2023-03-24"

runtimes:
```

```
- editor: Workbench
  kernel: Python 3.9
  edition: Standard

tasks:
- type: run_session
  name: Install Dependencies
  script: cml/install_dependencies.py
  kernel: python3
  cpu: 1
  memory: 2

- type: start_application
  name: Streamlit App
  subdomain: streamlit
  script: cml/launch_app.py
  short_summary: Start Streamlit application
```

```
environment_variables:
TASK_TYPE: START_APPLICATION
```

1. Close the Github tab and return to the AMP dialog. Click the **Configure & Deploy** button.

The screenshot shows the CloudERA Machine Learning interface with the 'Applied ML Prototypes' section. A modal window titled 'Configure & Deploy AMP' is overlaid. Inside the modal, there's a heading 'Streamlit', a brief description of the project, and a 'Tags' section with 'Streamlit', 'Applications', and 'Data Visualization'. At the bottom of the modal, there are two buttons: 'Configure & Deploy' (highlighted with a yellow circle and a cursor arrow) and 'View on Github'. The background shows various ML prototypes like 'Explaining Models with LIME and SHAP', 'Object Detection Inference Visualized', and 'AutoML with TPOT'.

2. Click the **Launch Project** button.

The screenshot shows the 'Configure Project' dialog for the Streamlit AMP. The title is 'Configure Project: Streamlit - sci_10_3678619'. It includes sections for 'Environment Variables' (which is empty), 'Runtime' (with dropdowns for Editor, Kernel, Edition, Version, and a 'Enable Spark' toggle), 'Runtime Image' (set to 'Spark 3.2.3 - CDE 1.19.2 - HOTFIX-2'), and 'Setup Steps' (with a checked checkbox for 'Execute AMP setup steps'). At the bottom right of the dialog, the 'Launch Project' button is highlighted with a yellow circle and a cursor arrow.

The AMP will begin building. You will see the two steps that were in the `project-metadata.yaml` file execute.

1. Wait for AMP to complete. It will take approximately five minutes for the AMP to complete both steps.

analyst_5_2283 / Streamlit - analyst_5_2283 / AMP Status

AMP Setup Steps

AMP Name: Streamlit (v2)
Run a Streamlit app inside CML

Completed 0 of 2 steps

Step 1 Run session [View details](#) started 8/23/2022 11:00 AM

```
AND WILL NOT DEFEND, INDEMNIFY, NOR HOLD YOU HARMLESS FOR ANY CLAIMS ARISING FROM OR RELATED TO THE CODE; AND (D) WITH RESPECT TO YOUR EXERCISE OF ANY RIGHTS GRANTED TO YOU FOR THE CODE, CLOUDERA IS NOT LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, PUNITIVE OR CONSEQUENTIAL DAMAGES INCLUDING, BUT NOT LIMITED TO, DAMAGES RELATED TO LOST REVENUE, LOST PROFITS, LOSS OF INCOME, LOSS OF BUSINESS ADVANTAGE OR UNAVAILABILITY, OR LOSS OR CORRUPTION OF DATA.
```

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

Step 2 Start Streamlit application not yet started

Explore the Streamlit Project

Now that the AMP has completed creating a new project, you can view the contents of the project to see what was created.

1. Click **Overview** in the project menu.

analyst_5_2283 / Streamlit - analyst_5_2283 / AMP Status

AMP Setup Steps

AMP Name: Streamlit (v2)
Run a Streamlit app inside CML

Completed all steps

Step 1 Run session [View details](#) completed 8/23/2022 11:04 AM

```
AND WILL NOT DEFEND, INDEMNIFY, NOR HOLD YOU HARMLESS FOR ANY CLAIMS ARISING FROM OR RELATED TO THE CODE; AND (D) WITH RESPECT TO YOUR EXERCISE OF ANY RIGHTS GRANTED TO YOU FOR THE CODE, CLOUDERA IS NOT LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, PUNITIVE OR CONSEQUENTIAL DAMAGES INCLUDING, BUT NOT LIMITED TO, DAMAGES RELATED TO LOST REVENUE, LOST PROFITS, LOSS OF INCOME, LOSS OF BUSINESS ADVANTAGE OR UNAVAILABILITY, OR LOSS OR CORRUPTION OF DATA.
```

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

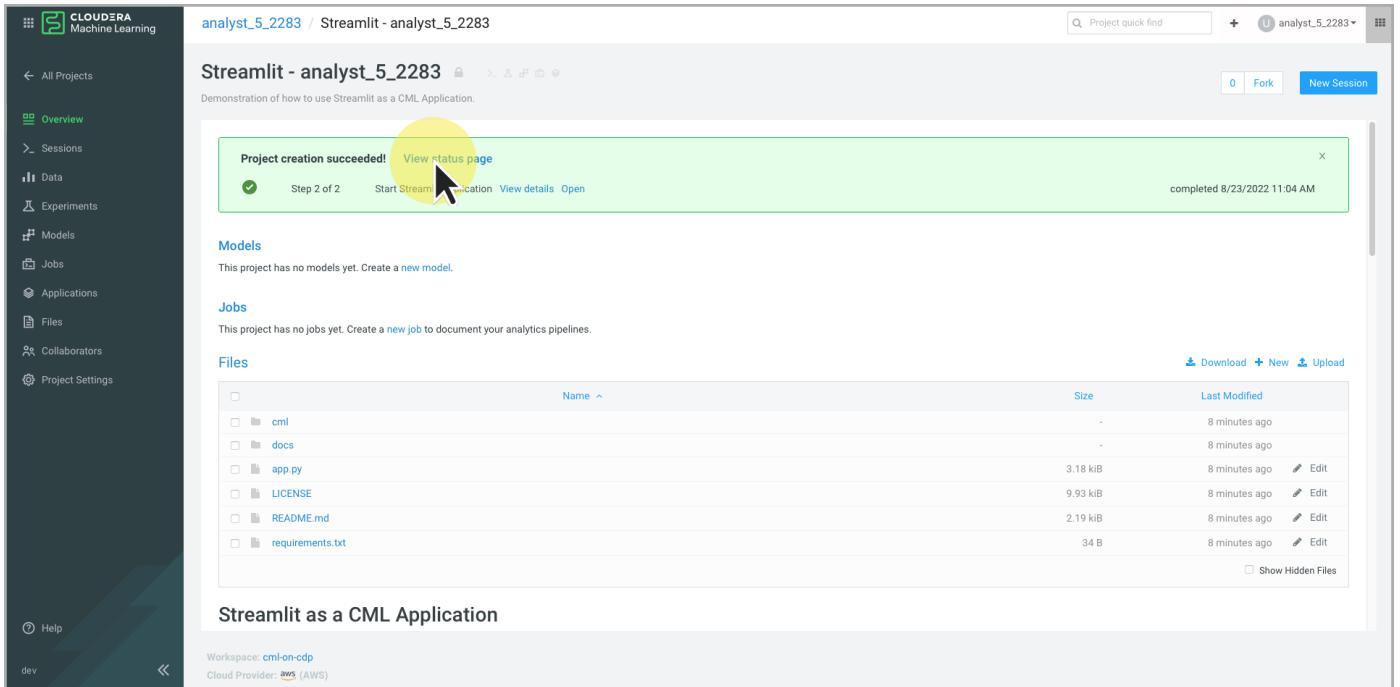
Step 2 Start Streamlit application [View details](#) completed 8/23/2022 11:04 AM

```
AND WILL NOT DEFEND, INDEMNIFY, NOR HOLD YOU HARMLESS FOR ANY CLAIMS ARISING FROM OR RELATED TO THE CODE; AND (D) WITH RESPECT TO YOUR EXERCISE OF ANY RIGHTS GRANTED TO YOU FOR THE CODE, CLOUDERA IS NOT LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, PUNITIVE OR CONSEQUENTIAL DAMAGES INCLUDING, BUT NOT LIMITED TO, DAMAGES RELATED TO LOST REVENUE, LOST PROFITS, LOSS OF INCOME, LOSS OF BUSINESS ADVANTAGE OR UNAVAILABILITY, OR LOSS OR CORRUPTION OF DATA.
```

```
#####
!streamlit run app.py --server.port $CDSW_APP_PORT --server.address 127.0.0.1
```

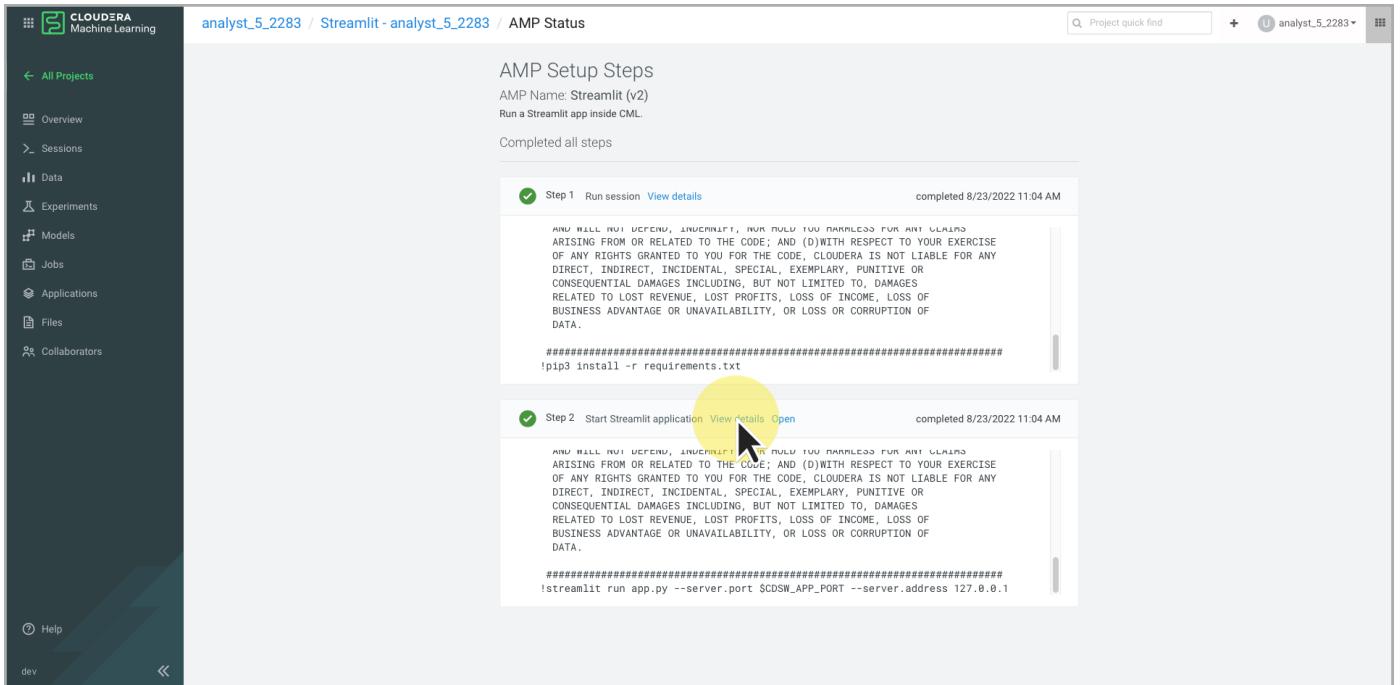
The project overview shows any models or jobs that were created, the files contained in the project, and the contents of the `README.md` markdown file which describe the project. At the top of the overview page is the message regarding the project creation status which contains a useful link to the return to the project creation status page.

1. Click View status page.



The screenshot shows the Streamlit project overview page for 'analyst_5_2283'. A yellow circle highlights the 'View status page' button under the 'Project creation succeeded!' message. The message also includes a link to 'View details' and an 'Open' button. The sidebar on the left lists various project components like Sessions, Data, Experiments, Models, Jobs, Applications, Files, Collaborators, and Project Settings. The main content area shows sections for Models, Jobs, and Files, with a table listing files like app.py, LICENSE, README.md, and requirements.txt. A 'Streamlit as a CML Application' section is also present.

2. Click View details for Step 2 - Start Streamlit Application.



The screenshot shows the Streamlit AMP Status page for 'analyst_5_2283'. A yellow circle highlights the 'View details' button for Step 2: 'Start Streamlit application'. The page displays two steps: Step 1 (Run session) and Step 2 (Start Streamlit application). Both steps show a green checkmark and the text 'completed 8/23/2022 11:04 AM'. The Step 2 section includes a warning text about liability and a command prompt showing the execution of '!streamlit run app.py --server.port \$CDSW_APP_PORT --server.address 127.0.0.1'.

A new tab will open with an overview of the application deployment. The overview provides a link to the application, the script that created the application, and when the application was last started.

1. Click the Application Logs tab.

The screenshot shows the Streamlit App Overview page. The 'Application Logs' tab is highlighted with a yellow circle and a cursor arrow pointing to it. The page displays the following information:

- Application:** Streamlit App
- Script:** cml/launch_app.py
- Description:** No description for the app
- Created by:** scl_10_3678619
- Most Recent Start/Restart by:** scl_10_3678619
- Ran:** 1 time

The left sidebar shows the CloudPilot interface with various project and application management options. The 'Applications' section is currently selected. The URL in the browser is https://ml-3248cf5e-a25.dfeinzel.kfp-x0dn.cloudera.site/scl_10_3678619/streamlit-scl_10_3678619/applications/1/logs.

2. View the log output.

The screenshot shows the Streamlit App Application Logs page. The 'Application Logs' tab is highlighted with a yellow circle and a cursor arrow pointing to it. The page displays the following log output:

```
#####
# CLOUDERA APPLIED MACHINE LEARNING PROTOTYPE A M P C Cloudera, Inc. 2021 All rights reserved.
#
# Applicable Open Source License: Apache 2.0
#
# NOTE: Cloudera open source products are modular software products made up of hundreds of individual components, each of which was
# individually copyrighted. Each Cloudera open source product is a collective work under U.S. Copyright Law. Your license to use the
# collective work is as provided in your written agreement with Cloudera. Used apart from the collective work, this file is licensed for your
# use pursuant to the open source license identified above.
#
# This code is provided to you pursuant to a written agreement with i Cloudera, Inc. or i a third-party authorized to distribute this code. If you
# do not have a written agreement with Cloudera nor with an authorized and properly licensed third party, you do not have any rights to
# access nor to use this code.
#
# Absent a written agreement with Cloudera, Inc. ("Cloudera") to the contrary, A) CLOUDERA PROVIDES THIS CODE TO YOU WITHOUT
# WARRANTIES OF ANY KIND; B) CLOUDERA DISCLAIMS ANY AND ALL EXPRESS AND IMPLIED WARRANTIES WITH RESPECT TO THIS
# CODE, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS FOR
# A PARTICULAR PURPOSE; C) CLOUDERA IS NOT LIABLE TO YOU, AND WILL NOT DEFEND, INDEMNIFY, NOR HOLD YOU HARMLESS FOR
# ANY CLAIMS ARISING FROM OR RELATED TO THE CODE, AND D) WITH RESPECT TO YOUR EXERCISE OF ANY RIGHTS GRANTED TO YOU
# FOR THE CODE, CLOUDERA IS NOT LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, PUNITIVE OR
# CONSEQUENTIAL DAMAGES INCLUDING, BUT NOT LIMITED TO, DAMAGES RELATED TO LOST REVENUE, LOST PROFITS, LOSS OF
# INCOME, LOSS OF BUSINESS ADVANTAGE OR UNAVAILABILITY, OR LOSS OR CORRUPTION OF DATA
#
#####
> !streamlit run app.py --server.port $CDSW_APP_PORT --server.address 127.0.0.1
You can now view your Streamlit app in your browser.

URL: http://127.0.0.1:8100
```

The left sidebar shows the CloudPilot interface with various project and application management options. The 'Applications' section is currently selected. The URL in the browser is https://ml-3248cf5e-a25.dfeinzel.kfp-x0dn.cloudera.site/scl_10_3678619/streamlit-scl_10_3678619/applications/1/logs.

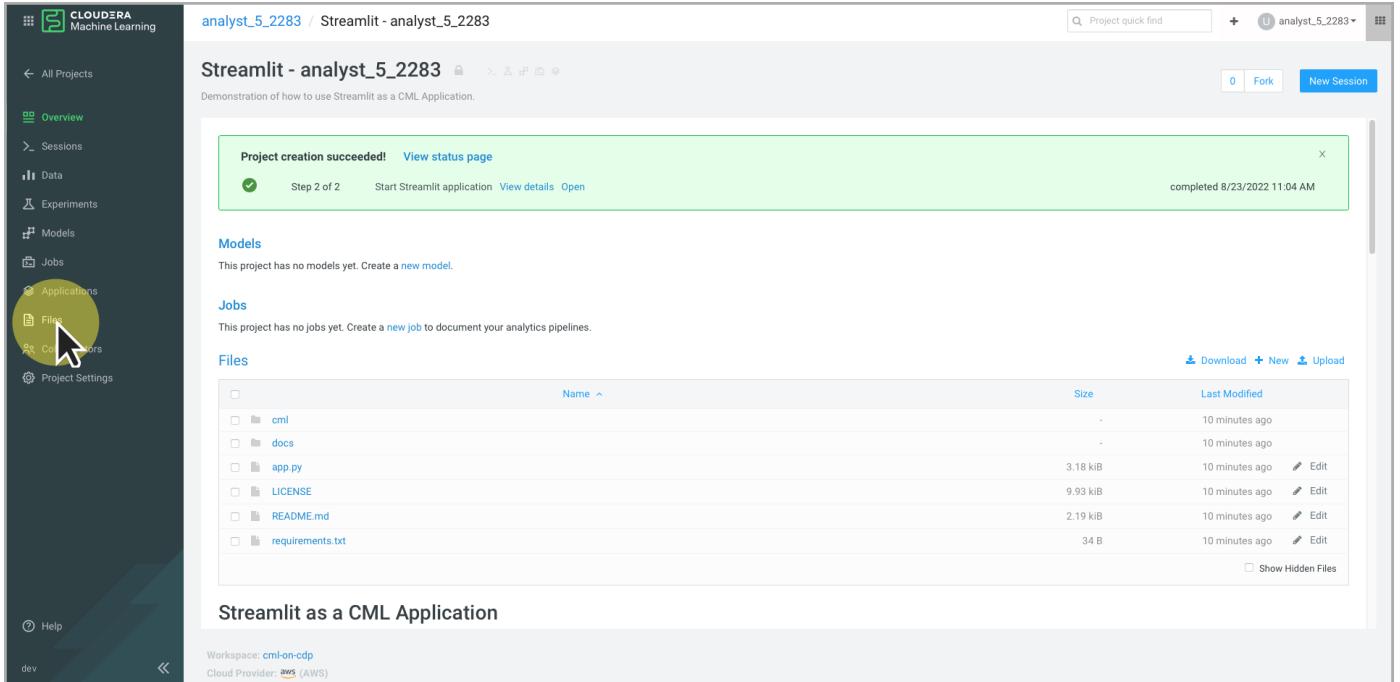
Logs shows the output from the application. In this case, you can see the command that was used to start the application:

```
!streamlit run app.py --server.port $CDSW_APP_PORT --server.address 127.0.0.1
```

 Note

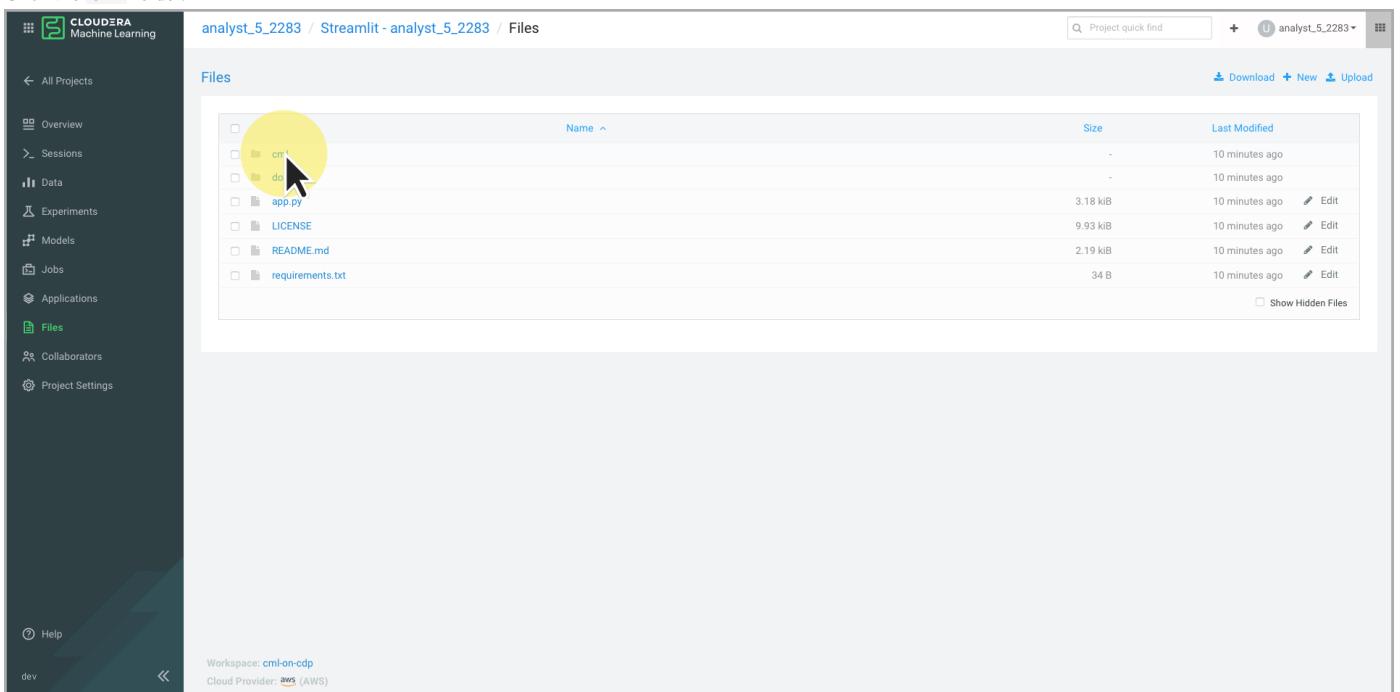
CML applications are long-running web applications. Unlike sessions, applications do not timeout from inactivity. The application must use either `CDSW_APP_PORT` or `CDSW_READONLY_PORT` as its server port.

1. Click **Files** in the project menu.



The screenshot shows the Cloudera Machine Learning interface with the 'Files' section selected. The left sidebar has a 'Files' icon highlighted with a yellow circle. The main content area displays a file tree under the 'Streamlit - analyst_5_2283' tab. A green banner at the top says 'Project creation succeeded!' with a link to 'View status page'. Below it, there are sections for 'Models' (empty), 'Jobs' (empty), and 'Files'. The 'Files' section lists several files: 'cml', 'docs', 'app.py', 'LICENSE', 'README.md', and 'requirements.txt'. Each file entry includes a checkbox, name, size, and last modified timestamp. Buttons for 'Download', 'New', and 'Upload' are at the top right of the file list.

2. Click the **CML** folder.



This screenshot is similar to the previous one but focuses on the 'cml' folder within the 'Files' section. A yellow circle highlights the 'cml' folder icon. The file list below it contains 'do', 'app.py', 'LICENSE', 'README.md', and 'requirements.txt'. The rest of the interface, including the sidebar and top navigation, is identical to the first screenshot.

3. Click `launch_app.py`.

The screenshot shows the Cloudera Machine Learning interface with the project 'analyst_5_2283'. In the left sidebar, the 'Sessions' icon is highlighted with a yellow circle. The main area displays a file list titled 'Files / cml'. Two files are listed: 'install_dependencies.py' and 'launch_app.py'. Both files were modified 10 minutes ago. The 'launch_app.py' file is selected, indicated by a yellow circle around its row.

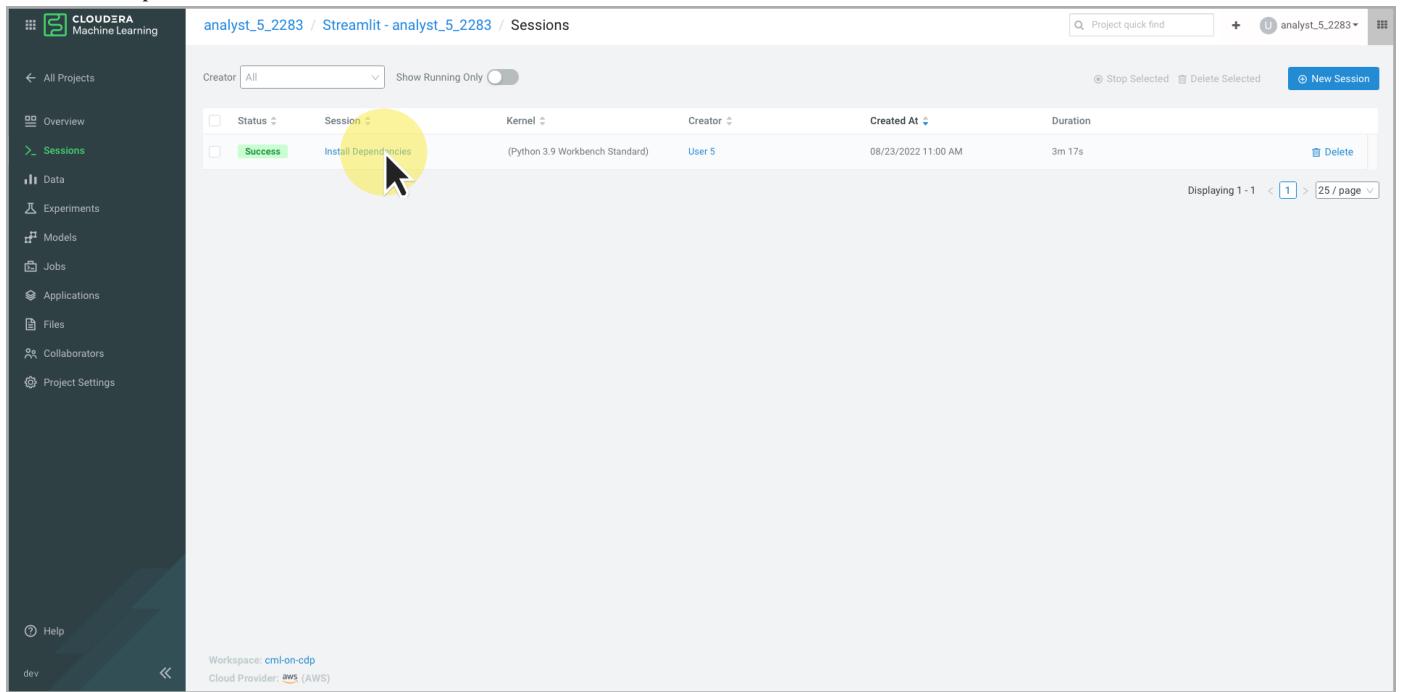
This is the file that was used to start the application. If you look back at the `project-metadata.yaml` file, you will see where it was specified in the `start_application` task.

1. Click Sessions in the project menu.

The screenshot shows the Cloudera Machine Learning interface with the project 'analyst_5_2283'. The 'Sessions' icon in the left sidebar is highlighted with a yellow circle. The main area displays the 'Sessions' view, which lists several sessions. One session, 'Session 1', is currently active and highlighted with a yellow circle.

This is a list of all of the project sessions. Here, you can see the session used by the AMP to install the Python dependencies.

1. Click **Install Dependencies** in the list of sessions.



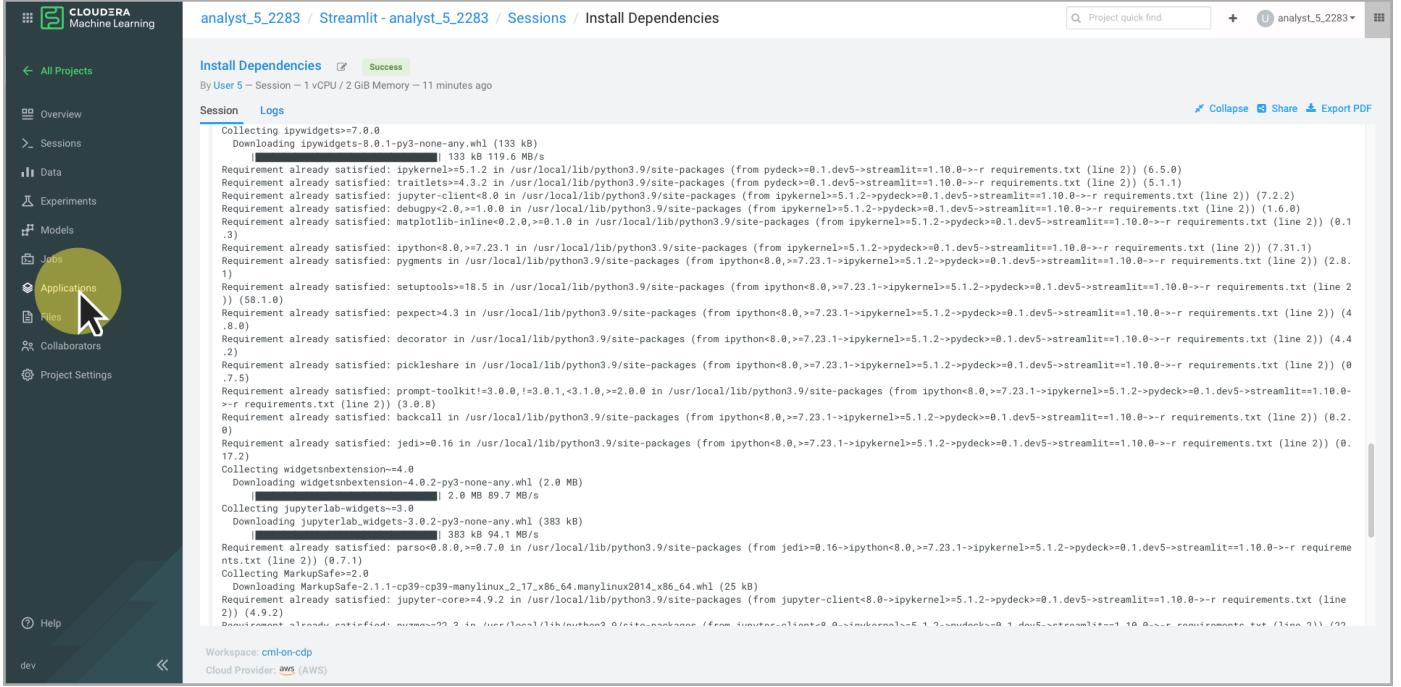
The screenshot shows the Cloudera Machine Learning interface with the 'Sessions' tab selected. A single session is listed:

Status	Session	Kernel	Creator	Created At	Duration
Success	Install Dependencies	(Python 3.9 Workbench Standard)	User 5	08/23/2022 11:00 AM	3m 17s

A yellow circle highlights the 'Install Dependencies' button next to the session name. A cursor arrow points at this highlighted button. The interface includes a sidebar with various project management options like All Projects, Overview, Sessions, Data, Experiments, Models, Jobs, Applications, Files, Collaborators, and Project Settings. The bottom of the screen displays workspace and cloud provider information: 'Workspace: cml-on-cdp' and 'Cloud Provider: AWS (AWS)'.

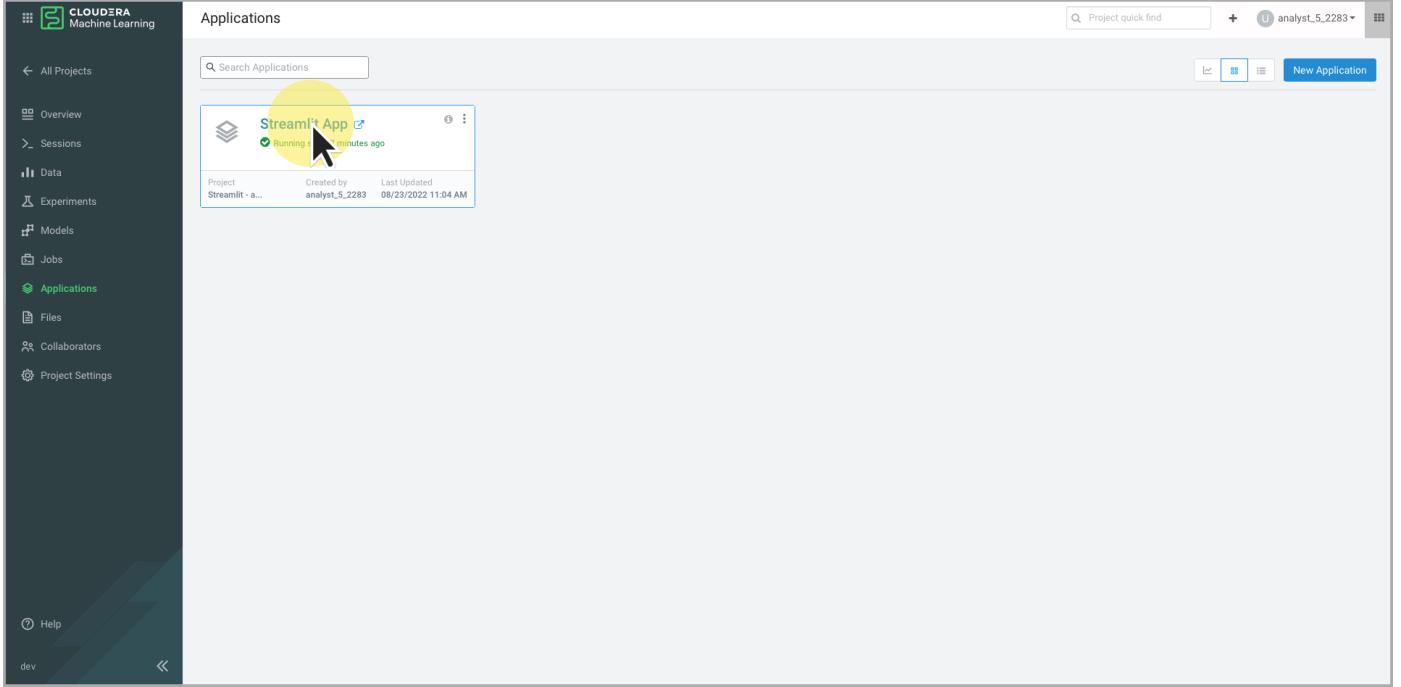
Again, you see the output of the session. Similarly, you can view the application created by the AMP.

1. Click Applications in the project menu.



The screenshot shows the CloudERA Machine Learning interface. On the left, there's a sidebar with various project management options like Overview, Sessions, Data, Experiments, Models, Jobs, Applications (which is highlighted with a yellow circle), and Collaborators. The main area displays a session titled "Install Dependencies" with a status of "Success". It shows the command "Collecting ipywidgets>=7.0.0" followed by a long list of dependency download logs. At the bottom, it says "Workspace: cm-on-cdp" and "Cloud Provider: AWS (AWS)".

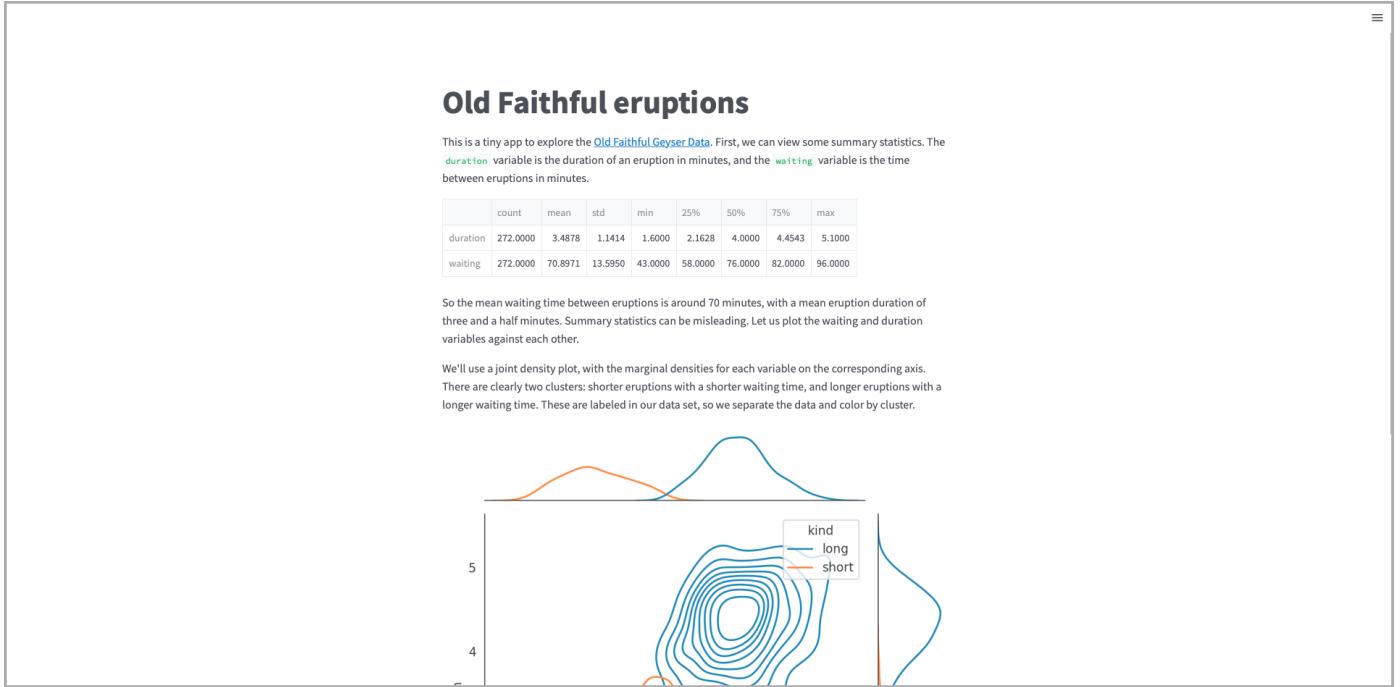
2. Click Streamlit App.



This screenshot shows the same CloudERA interface, but the main focus is now the "Applications" section. It lists a single application named "Streamlit App" which is currently "Running" (indicated by a green status icon). Below the card, it shows the project name "Streamlit - a...", the creator "analyst_5_2283", and the last update time "08/23/2022 11:04 AM". The sidebar remains the same as in the previous screenshot.

This opens a new tab and displays the application. The URL can be shared and viewed by others. 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. Therefore, users will typically have to sign into the CDP environment before opening an application's URL.

1. View Streamlit App.

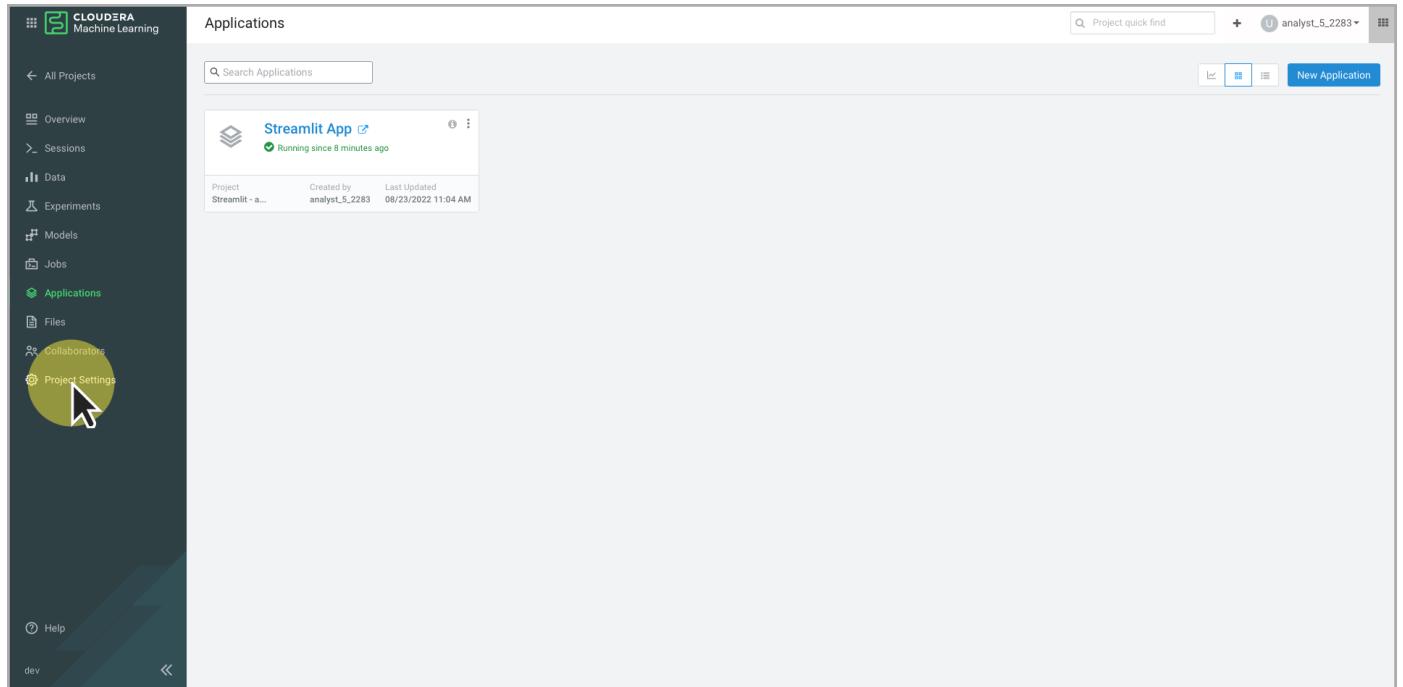


Modify the Application

1. Return to the Streamlit project.
2. Click **Files** in the project menu.
3. Compare the `app.py` code to the output of the app itself.
4. Click the **Open in Session** button.
5. Change some of the markdown content and reload the app.
6. Did you need to rebuild the app?
7. Why is the first markdown block statement in a `st.markdown` call and not the second one?
8. Study the `jointplot` seaborn function: <https://seaborn.pydata.org/generated/seaborn.jointplot.html>
9. Change the type of the `jointplot`.

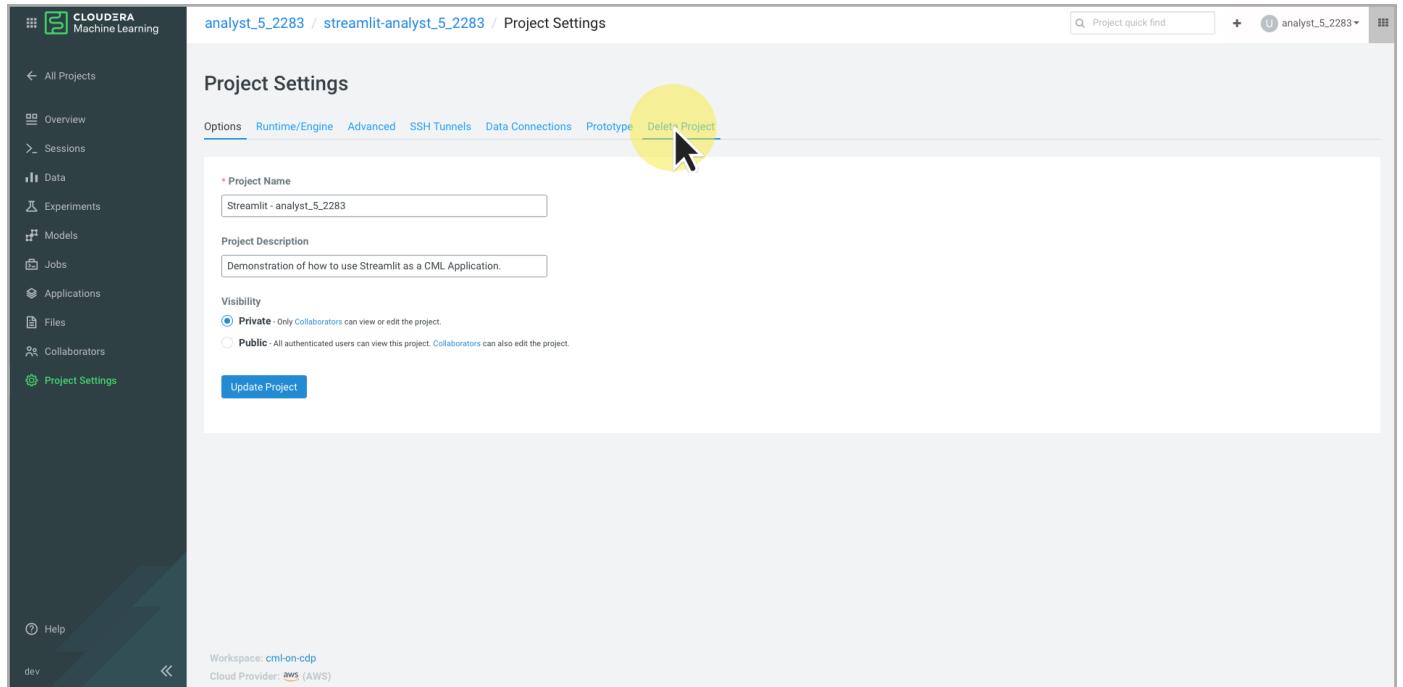
Delete the Project

1. Click Project Settings



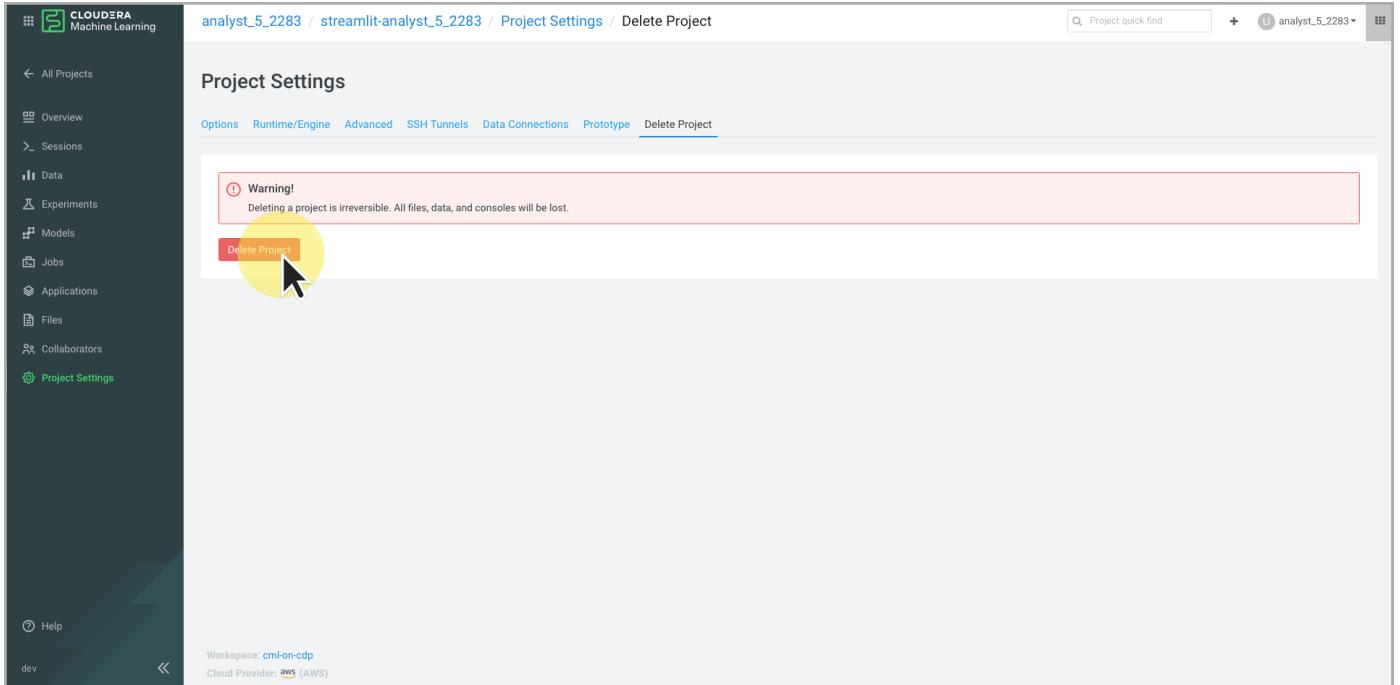
The screenshot shows the CloudERA Machine Learning application interface. On the left, a dark sidebar contains various project management links: All Projects, Overview, Sessions, Data, Experiments, Models, Jobs, Applications (which is selected and highlighted in green), Files, Collaborators, and Project Settings. A yellow circle with a cursor icon is positioned over the 'Project Settings' link. The main content area is titled 'Applications' and displays a single entry: 'Streamlit App' with a status of 'Running since 8 minutes ago'. Below this entry, there is a table with columns: Project, Created by, and Last Updated, showing 'Streamlit - a...' as the project name, 'analyst_5_2283' as the creator, and '08/23/2022 11:04 AM' as the last update time.

2. Click Delete Project tab

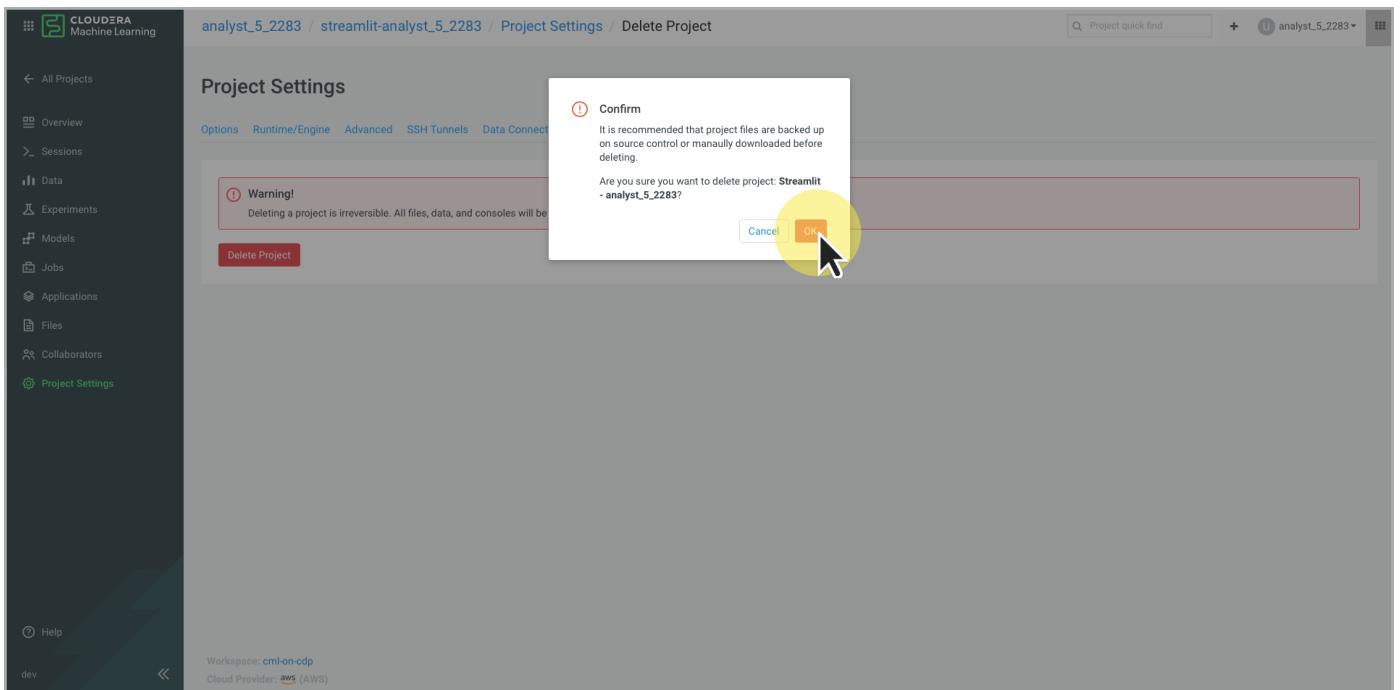


The screenshot shows the 'Project Settings' page for the project 'analyst_5_2283'. The top navigation bar includes tabs for Options, Runtime/Engine, Advanced, SSH Tunnels, Data Connections, Prototype, and Delete Project (which is highlighted with a yellow circle). The main form fields include 'Project Name' (set to 'Streamlit - analyst_5_2283'), 'Project Description' (containing the text 'Demonstration of how to use Streamlit as a CML Application.'), and 'Visibility' options ('Private' is selected, indicated by a yellow circle). At the bottom of the form is a blue 'Update Project' button. The sidebar on the left is identical to the one in the first screenshot, showing the 'Project Settings' link is also highlighted in green. The bottom of the screen shows workspace and cloud provider information: 'Workspace: cml-on-cdp' and 'Cloud Provider: AWS (AWS)'.

3. Click Delete Project button



4. Click OK to confirm



Bonus

If you finish early, explore other AMPs in the AMP catalog.

End of Exercise

Solution

Data - Access, Audit, and Mask

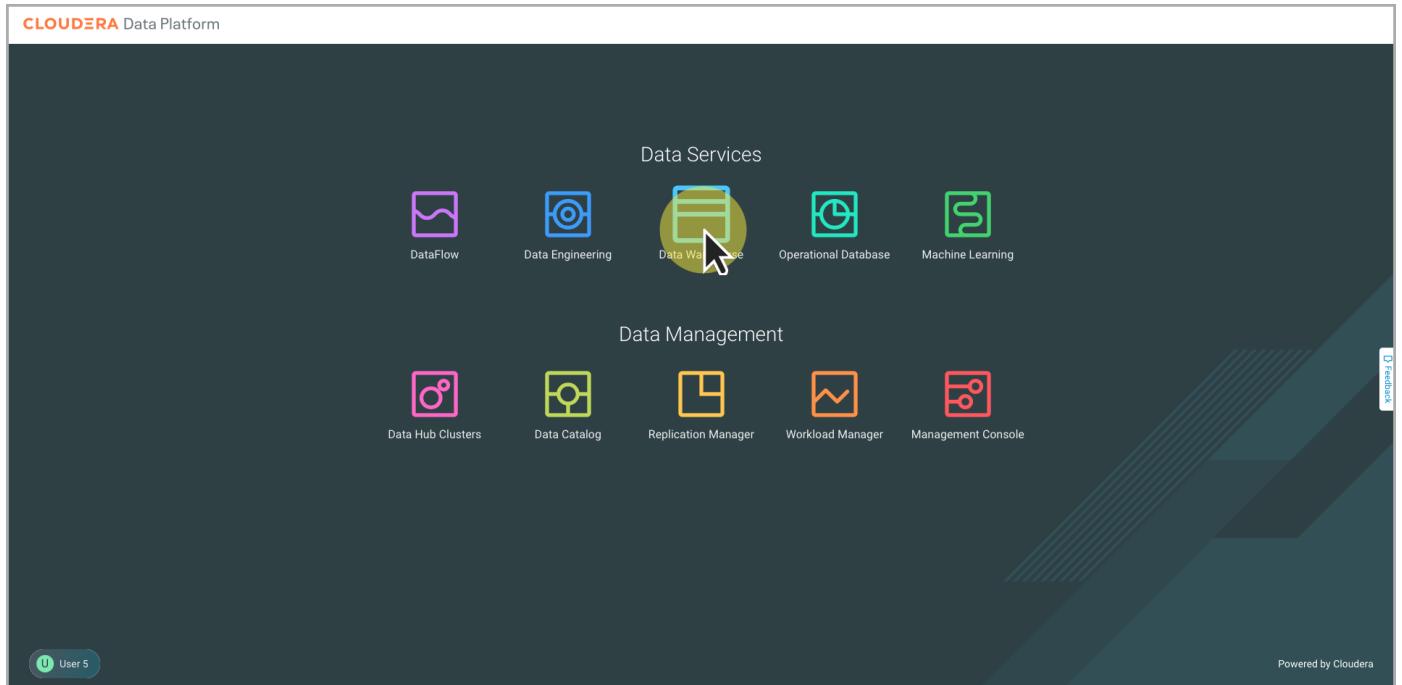
This exercise uses the fictitious Duocar dataset. The data is stored in Amazon S3 and has been added to the company's data warehouse. Some of the data contains Personal Identifiable Information (PII). The company has requested the administrator only allow access to the information to those who need access to it to perform their job. In this instance, the full birth date is considered to be PII. Therefore, the birth date fields have been classified as PII and a PII policy has been created to mask the month and day.

Note

This exercise uses birth dates as PII. While a birth date may be considered PII in some scenarios, this exercise is completely hypothetical and has been designed to demonstrate the software and concepts. What data is PII and how to protect it is a legal issue and is beyond the scope of this course.

Access Data

1. Access the Data Warehouse by clicking **Data Warehouse**.



Warning

There is a known issue opening Hue with Safari on MacOS that results in an "Invalid CORS request." Please use another browser, like Chrome or Firefox, if you experience this issue.



If you see a message near the top of the page that says, "Having trouble connection to server." or notice that the Virtual Warehouse is stopped. These are both normal. The Virtual Warehouse will start automatically when required.

1. Click **Hue** to launch Hue in a new browser tab.

2. Once Hue launches, make sure you are in the Hive editor and `duocar` is the selected database. If you are not in the Hive editor, select the </> from the left side menu.

3. If `duocar` is not selected, select `duocar` from the list of databases. If `default` selected and Tables is displayed, click the < to navigate back to the list of databases and then select `duocar`.

4. Next, execute the following SQL by entering it into the editor and clicking the play/run button to show the drivers' data:

```
select * from drivers limit 10;
```

Column	Type
<code>id</code>	string
<code>birth_date</code>	timestamp
<code>start_date</code>	timestamp
<code>first_name</code>	string
<code>last_name</code>	string
<code>gender</code>	string
<code>ethnicity</code>	string
<code>student</code>	boolean
<code>home_block</code>	string
<code>home_lat</code>	decimal(9,6)
<code>home_lon</code>	decimal(9,6)
<code>vehicle_make</code>	string
<code>vehicle_model</code>	string
<code>vehicle_year</code>	int
<code>vehicle_color</code>	string
<code>vehicle_grand</code>	boolean
<code>vehicle_noir</code>	boolean
<code>vehicle_elite</code>	boolean
<code>rides</code>	int
<code>stars</code>	int

5. Notice that the year for each driver's birth date varies, but the day and month are always 01 . This is a result of the custom tag-based policy.

drivers.id	drivers.birth_date	drivers.start_date	drivers.first_name	drivers.last_name	drivers.gender	drivers.ethnicity	drivers.student	drivers
1	1996-01-01 00:00:00	2017-01-01 00:00:00	Adam	Abrahamsen	male	White	true	270270
2	1993-01-01 00:00:00	2017-01-02 00:00:00	Brandon	Rutherford	male	White	false	380170
3	1985-01-01 00:00:00	2017-01-02 00:00:00	Sean	Woodhouse	male	White	false	380170
4	1964-01-01 00:00:00	2017-01-03 00:00:00	Logan	Smith	male	White	false	380170
5	1995-01-01 00:00:00	2017-01-03 00:00:00	Brooke	Label	female	White	false	380170
6	1961-01-01 00:00:00	2017-01-03 00:00:00	C	Hilton	male	White	false	380170
7	1967-01-01 00:00:00	2017-01-03 00:00:00	Patrick	Skelton	male	White	false	380170
8	1981-01-01 00:00:00	2017-01-03 00:00:00	Zachary	Atchley	male	White	false	380170
9	1993-01-01 00:00:00	2017-01-03 00:00:00	Dale	Duncomb	male	White	true	380170
10	1993-01-01 00:00:00	2017-01-04 00:00:00	James	Williams	male	White	false	380170

Create a new joined table of birth dates.

Now that you have seen the effects of masking on the original tables, what happens when you create a new table from tables that have been masked? In the following steps, you will create a new table from the `drivers` and `riders` tables. You will view the new table and its data. You will also view the table in the data catalog.

1. Use the following SQL to create a new table that contains the driver's and rider's birth date for each ride. Replace the XX in the table name with your student number.

```
create table birth_dates_XX
as
select riders.birth_date as rider_bd, drivers.birth_date as driver_bd from rides
join riders on rides.rider_id = riders.id
join drivers on rides.driver_id = drivers.id;
```

The screenshot shows the Cloudera Manager Hive interface. On the left, there's a sidebar with icons for duocar, Tables, drivers, joined, ride_reviews, and riders. The main area has a search bar at the top. Below it, a table titled 'Hive' shows the query being run. The query text is:

```
create table birth_dates_XX
as
select riders.birth_date as rider_bd, drivers.birth_date as driver_bd from rides
join riders on rides.rider_id = riders.id
join drivers on rides.driver_id = drivers.id;
```

Below the query, the output shows:

```
drop table birth_dates
INFO : Starting task [Stage-0:00L] in serial mode
INFO : Completed executing command(queryId=hive_20220904203528_c16b5f1f-6fad-42f1-a1c9-7afdb8dbc573c); Time taken: 0.463 seconds
INFO : OK
```

A yellow circle highlights the 'Run' button (a green triangle icon) next to the query text. To the right, a sidebar titled 'Tables' lists tables: duocar.drivers, duocar.riders, and duocar.rides. At the bottom, tabs for 'Query History' and 'Saved Queries' are shown, with a note: 'You don't have any saved queries.'

2. You may have to wait for Tez session. If so, it usually takes a few minutes for a session to start.

The screenshot shows the Cloudera Manager Tez interface. On the left, there's a sidebar with icons for ride_reviews, riders, and rides. The main area shows a job status with a progress bar. The log output shows:

```
INFO : Total jobs = 1
INFO : Launching Job 1 out of 1
INFO : Starting task [Stage-1:MAPRED] in serial mode
INFO : Subscribed to counters: [] for queryId: hive_202209042035181_cfecb627-8cd8-4573-951a-5feb4678a176
INFO : Tez session hasn't been created yet. Opening session
```

A red box highlights the last line of the log: 'INFO : Tez session hasn't been created yet. Opening session'. Below the log, tabs for 'Query History' and 'Saved Queries' are shown.

3. Once the table is created, use a SELECT statement to view the new table's contents.

The screenshot shows the Cloudera Manager Hive interface. On the left, there is a sidebar with icons for Overview, Database Catalogs, Virtual Warehouses, Data Visualization, Help, and User 5. The main area has tabs for Tables, Hive, Add a name..., and Add a description... At the top, there is a search bar with the placeholder "Search data and saved documents...". Below the search bar, there is a command line interface (CLI) window showing the execution of a query:

```
select * from birth_dates_5;
```

Output from the CLI:

```
INFO : Completed compiling command(queryId=hive_28220984203642_aebed1b1-9d22-4359-847f-54c0b946a414); Time taken: 0.075 seconds
INFO : Executing command(queryId=hive_28220984203642_aebed1b1-9d22-4359-847f-54c0b946a414); select * from birth_dates_5
INFO : Completed executing command(queryId=hive_28220984203642_aebed1b1-9d22-4359-847f-54c0b946a414); Time taken: 0.085 seconds
INFO : OK
```

Below the CLI, there are two tables displayed:

	birth_dates_5.rider_bd	birth_dates_5.driver_bd
1	NULL	1985-01-01 00:00:00
2	NULL	1968-01-01 00:00:00
3	NULL	1985-01-01 00:00:00
4	NULL	1993-01-01 00:00:00
5	NULL	1993-01-01 00:00:00
6	NULL	1985-01-01 00:00:00
7	NULL	1993-01-01 00:00:00
8	NULL	1985-01-01 00:00:00
9	NULL	1993-01-01 00:00:00
10	NULL	1985-01-01 00:00:00
11	NULL	1993-01-01 00:00:00
12	NULL	1963-01-01 00:00:00
13	NULL	1998-01-01 00:00:00
14	NULL	1963-01-01 00:00:00

4. Return to the Data Warehouse tab in your browser and click the main menu icon.

The screenshot shows the Cloudera Data Warehouse Overview page. On the left, there is a sidebar with icons for Overview, Database Catalogs, Virtual Warehouses, Data Visualization, Help, and User 5. The main area displays two database catalogs and one virtual warehouse:

- Database Catalogs | 1**
 - datalake-bshimel-500-class-...**
warehouse-1661876422-9278
Running bshimel-500-class-22829
- Virtual Warehouses | 1**
 - edu-cml-on-cdp-vwarehouse**
compute-1662233143-pe6w
Running datalake-bshimel-500-class-22829-default

At the bottom, there are navigation links for Help, User 5, and the version 1.4.2-b118.

5. Click Data Catalog.

The screenshot shows the Cloudera Data Platform interface. On the left, a sidebar titled 'CLOUDERA Data Platform' lists various services: Home, DataFlow, Data Engineering, Data Warehouse, Operational Database, Machine Learning, Data Hub Clusters, Data Catalog (which is highlighted with a yellow circle), Replicator, Workload Manager, and Management Console. The main area is titled 'Overview' and contains sections for 'Database Catalogs' and 'Virtual Warehouses'. Under 'Database Catalogs', there is one entry: 'datalake-bshimel-500-class...' with 'TOTAL CORES 9', 'TOTAL MEMORY 25 GB', and 'VIRTUAL WAREHOUSES 1'. Under 'Virtual Warehouses', there is one entry: 'edu-cml-on-cdp-vwarehouse' with 'EXECUTORS 10', 'TOTAL CORES 143', 'TOTAL MEMORY 1.21 TB', and 'TYPE HIVE UNIFIED ANALYTICS COMPACTOR'. There are also tabs for 'DAS' and 'HUE'.

6. Select the new table, `birth_dates_XX` where XX is your student number, from the list.

The screenshot shows the Cloudera Data Catalog interface. The left sidebar includes 'Search', 'Datasets', 'Bookmarks', 'Profilers', 'Atlas Tags', 'Help', and 'User 5'. The main area has a search bar and a message: 'The Profiler is not enabled on the Data Lake. Contact your Administrator.' Below this, under 'Data Lakes', there is one entry: 'aws_datalake-bshimel-500-class... 140'. A 'Filters' section is open, showing 'TYPE' (with 'Hive Table' selected) and 'OWNERS' (with 'adm_bshimel_22829' selected). A 'Actions' button is visible. The main table lists tables and columns from the data lake. One row, 'birth_dates_5', is highlighted with a yellow circle and a cursor pointing at it. The table columns are: Type, Name, Qualified Name, Created On, Owner, and Source. The table rows include: AWS S3 V2 Bucket, `cdf-storage-bshimel-500-class-22829`, `s3a://cdf-storage-bshimel-500-class-22829@cm`, -NA-, -NA-, aws; Hive Table, `birth_dates_5`, `duocar.birth_dates_5@cm`, Sun Sep 04 2022, dev_5_22829, hive; Hive Table, `ride_review`, `duocar.ride_reviews@cm`, Thu Sep 01 2022, adm_bshimel_22829, hive; Hive Table, `rides`, `duocar.rides@cm`, Thu Sep 01 2022, adm_bshimel_22829, hive; Hive Table, `joined`, `duocar.joined@cm`, Thu Sep 01 2022, adm_bshimel_22829, hive; Hive Table, `drivers`, `duocar.drivers@cm`, Thu Sep 01 2022, adm_bshimel_22829, hive; Hive Table, `riders`, `duocar.riders@cm`, Thu Sep 01 2022, adm_bshimel_22829, hive; Hive DB, `duocar`, `duocar@cm`, -NA-, adm_bshimel_22829, hive; Hive Column, `rider_bd`, `duocar.birth_dates_5.rider_bd@cm`, -NA-, dev_5_22829, hive; Hive Column, `driver_bd`, `duocar.birth_dates_5.driver_bd@cm`, -NA-, dev_5_22829, hive; Hive Column, `ethnicity`, `duocar.drivers.ethnicity@cm`, -NA-, adm_bshimel_22829, hive; Hive Column, `id`, `duocar.rides.id@cm`, -NA-, adm_bshimel_22829, hive; Hive Column, `date_time`, `duocar.rides.date_time@cm`, -NA-, adm_bshimel_22829, hive; Hive Column, `origin_lat`, `duocar.rides.origin_lat@cm`, -NA-, adm_bshimel_22829, hive; Hive Column, `duration`, `duocar.rides.duration@cm`, -NA-, adm_bshimel_22829, hive; Hive Column, `first_name`, `duocar.drivers.first_name@cm`, -NA-, adm_bshimel_22829, hive; and Hive Column, `origin_lon`, `duocar.joined.origin_lon@cm`, -NA-, adm_bshimel_22829, hive.

7. View the new table's lineage.

The screenshot shows the 'Asset Details' page for a table named 'birth_dates_5'. The 'Lineage' tab is selected, displaying a diagram of the data flow. Three input tables ('rides', 'drivers', and 'riders') are shown at the top, each with a green arrow pointing down to a single output table ('birth_dates_5') at the bottom. The 'birth_dates_5' table is highlighted with a red circle. A legend at the bottom of the diagram defines the colors: green for Lineage, red for Impact, blue for Replication, and orange for Current Entity.

8. Select the Schema tab. What are the classifications for the new fields?

The screenshot shows the 'Asset Details' page for the same table 'birth_dates_5'. The 'Schema' tab is highlighted with a yellow circle and a mouse cursor. The 'Overview' tab is also visible in the navigation bar. The rest of the interface is identical to the previous screenshot, showing the lineage diagram below the tabs.

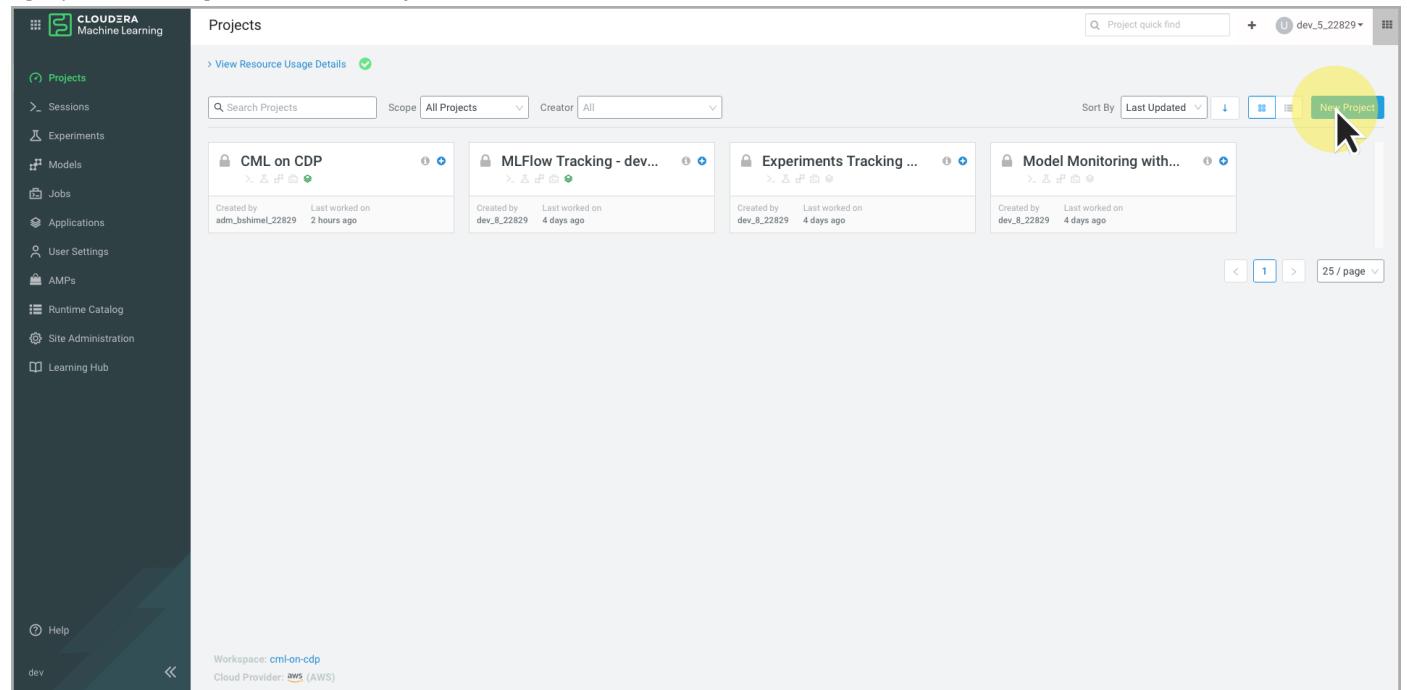
End of Exercise

Visualize Duocar Data

The Data Visualization application is good for exploring your data and sharing with others. In this exercise, you will:

- use the Data Visualization application to make a connection to the data warehouse,
- and create a dashboard to explore the ride data.

1. Open your CML workspace and click New Project.

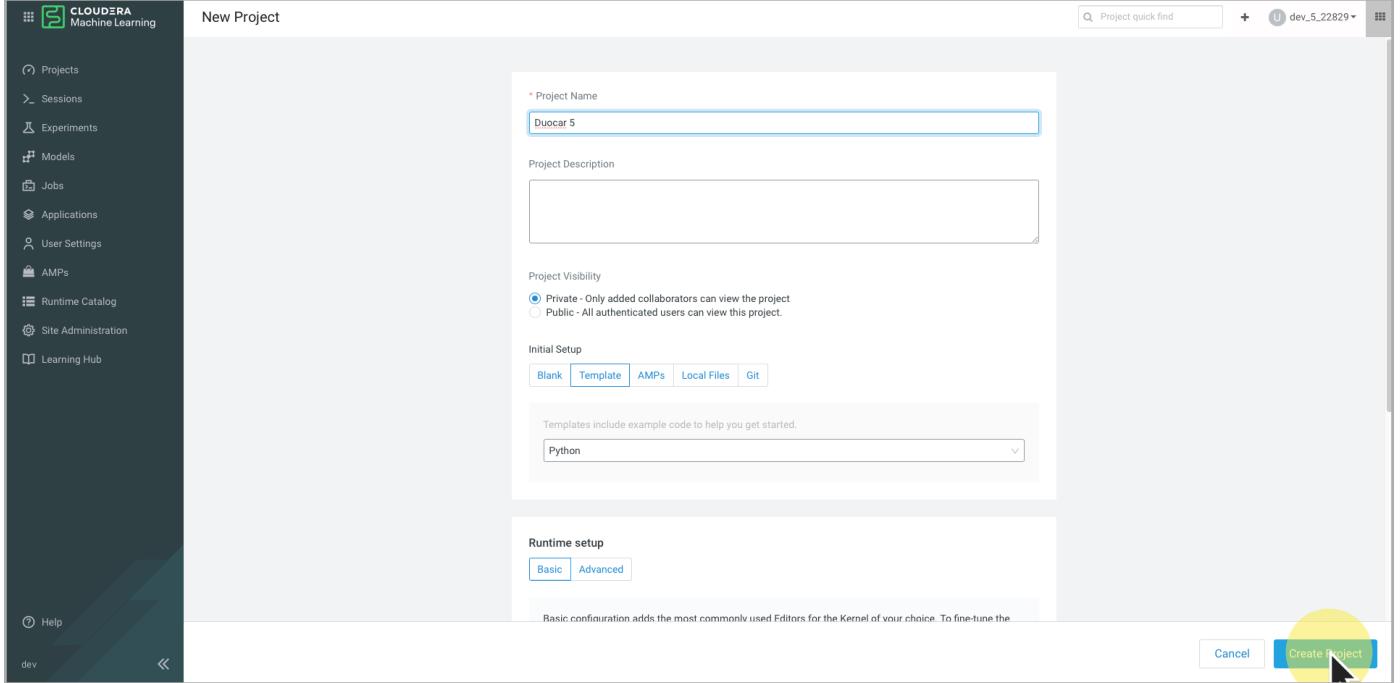


The screenshot shows the Cloudera Machine Learning (CML) interface. On the left, there's a sidebar with various navigation options: Projects, Sessions, Experiments, Models, Jobs, Applications, User Settings, Runtime Catalog, Site Administration, and Learning Hub. The main area is titled 'Projects' and displays four existing projects: 'CML on CDP', 'MLFlow Tracking - dev...', 'Experiments Tracking ...', and 'Model Monitoring with...'. Each project card includes details like 'Created by' and 'Last worked on'. At the top right of the main area, there's a green button labeled 'New Project' with a yellow circle and a cursor pointing at it. The bottom of the screen shows the workspace name 'cml-on-cdp' and the cloud provider 'aws (AWS)'.

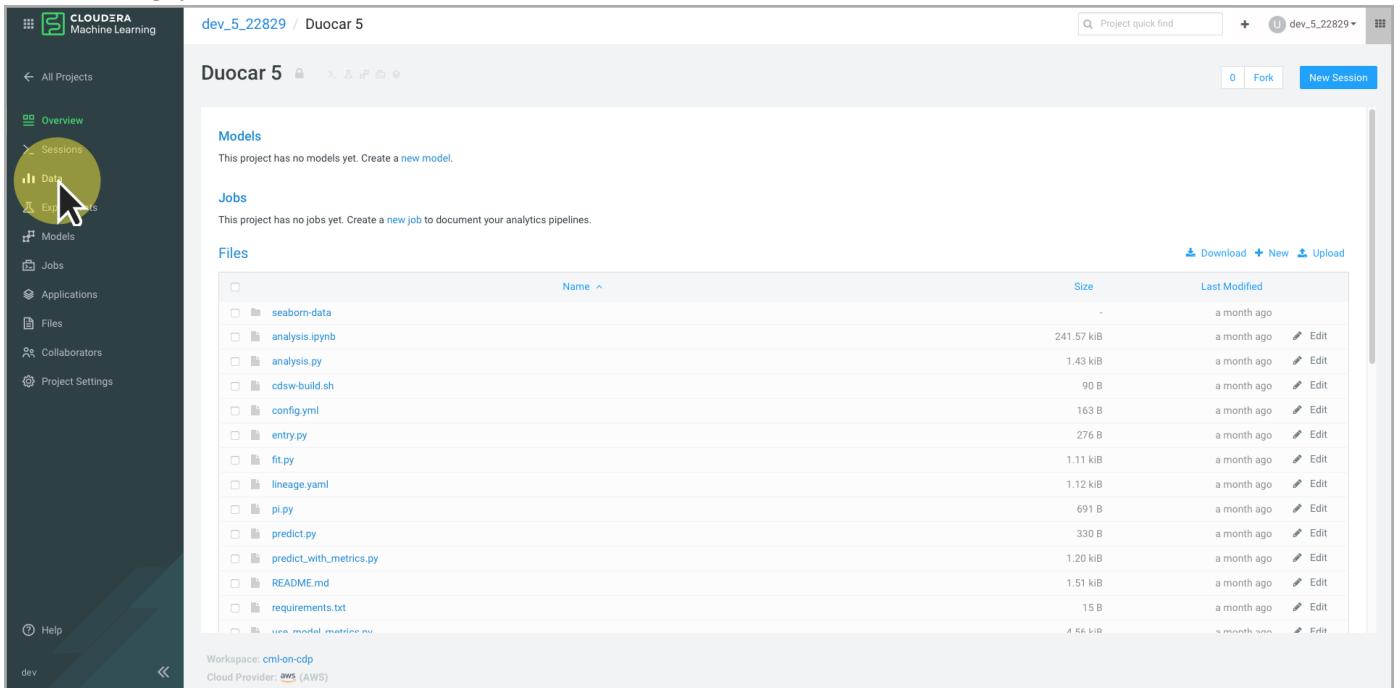
 Note

It is not necessary to create a new project to use Data Visualization. In this case, each student is creating their own project in order to demonstrate some features which are project-wide.

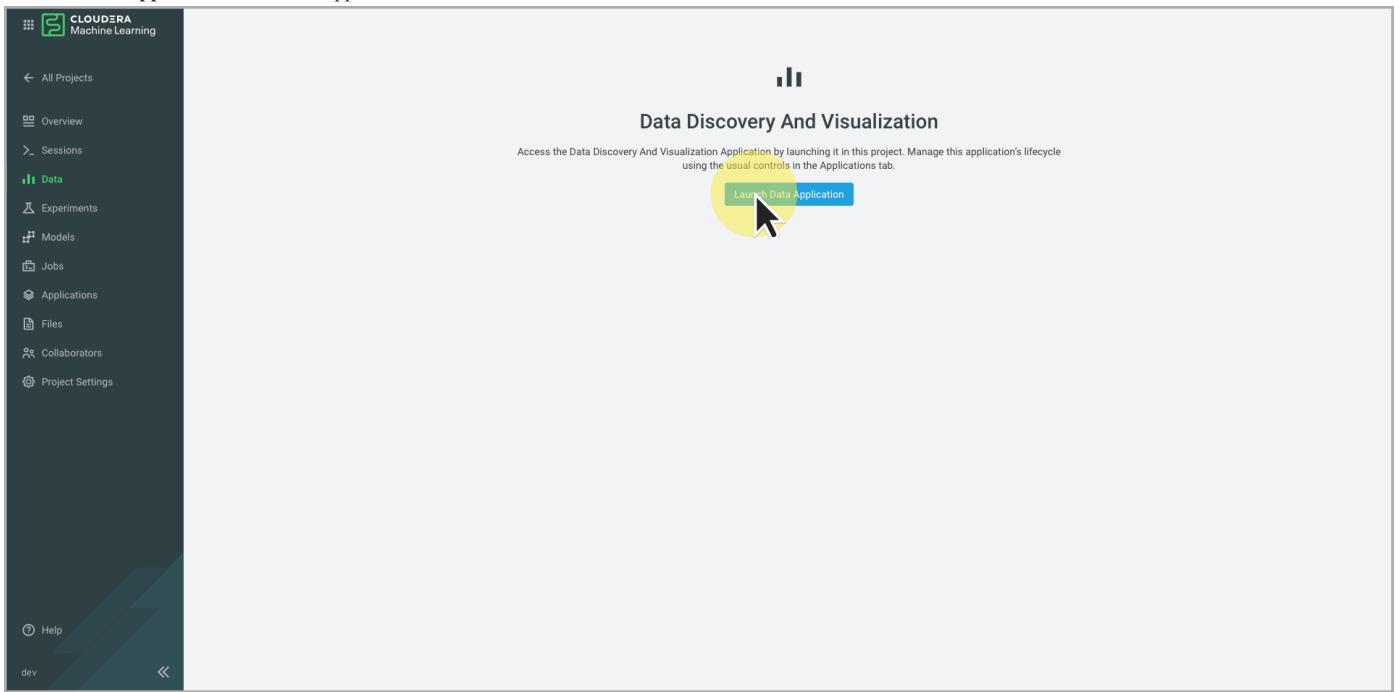
1. Enter `Duocar X` for the project name, where X is your student number. Click **Create Project**.



2. Click **Data** in the project menu.



3. Data Discovery and Visualization is just an application. The Data link in the project menu is just a convenient shortcut to launch and access the application. Click **Launch Data Application** to start the application.



4. The Data application homepage has useful information to get started and shortcut to common and recently used items.

Get Started

- Sync Connections
- Explore with SQL
- Create a Dashboard
- CML Notebook
- What's Next?

User Settings

cloudera_user - User Settings - Environment Variables

Environment Variables

reserved environment variables are required for specific features of CML. The variables can be one or more of the following:
WORKLOAD_PASSWORD WORKLOAD_PASSWORD

Data Connections

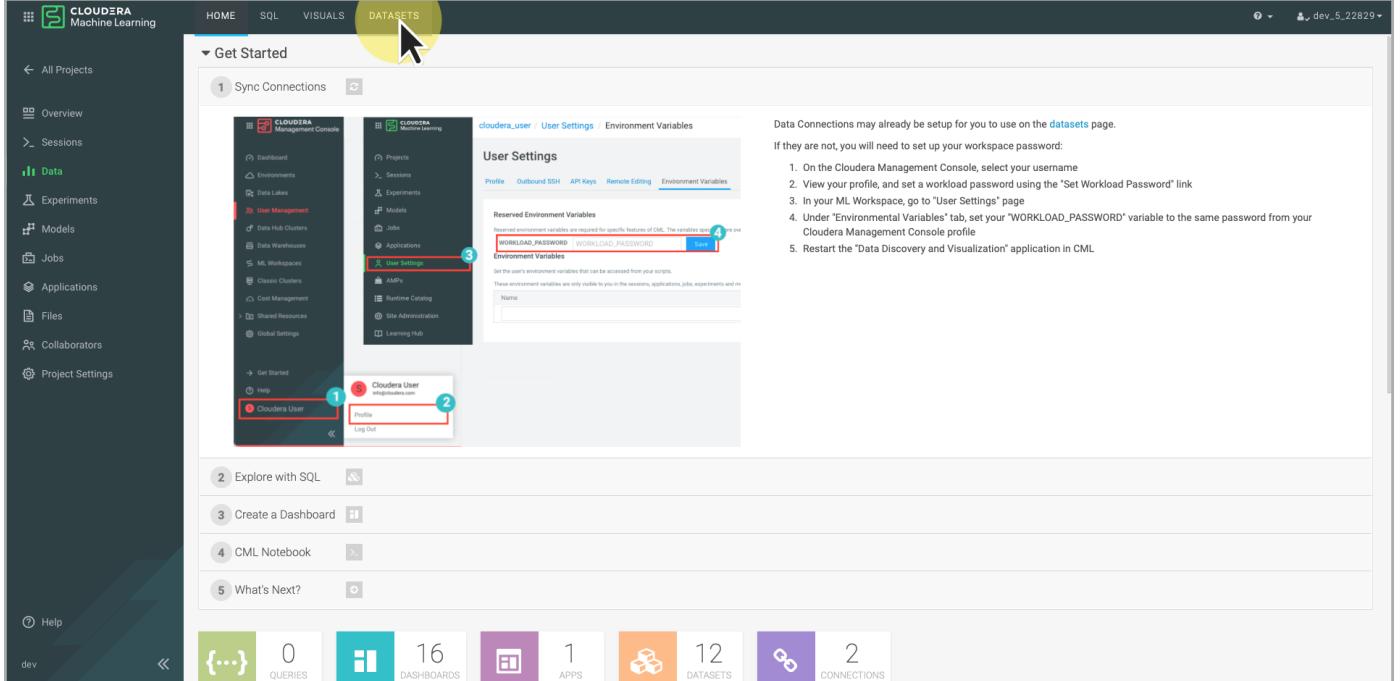
If they are not, you will need to set up your workspace password:

- On the Cloudera Management Console, select your username
- View your profile, and set a workload password using the "Set Workload Password" link
- In your ML Workspace, go to "User Settings" page
- Under "Environmental Variables" tab, set your "WORKLOAD_PASSWORD" variable to the same password from your Cloudera Management Console profile
- Restart the "Data Discovery and Visualization" application in CML

Dashboard Metrics

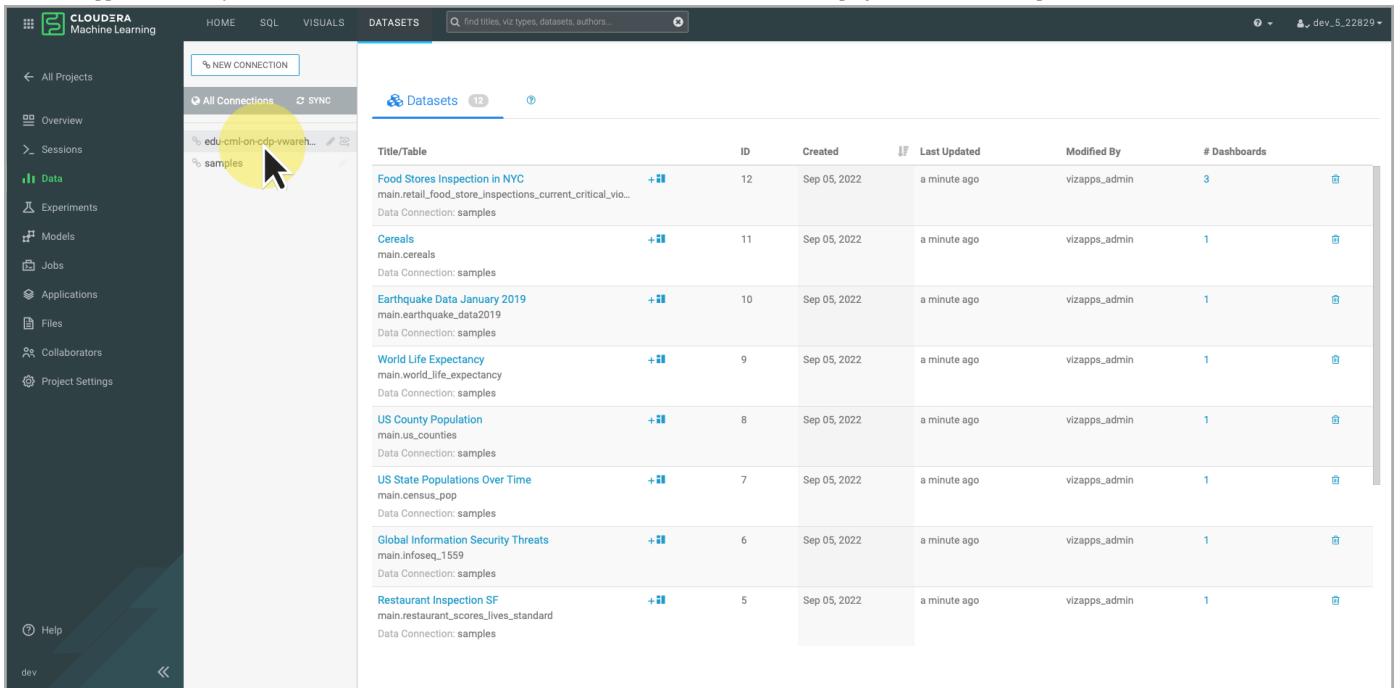
- 0 QUERIES
- 16 DASHBOARDS
- 1 APPS
- 12 DATASETS
- 2 CONNECTIONS

5. Click **Datasets** in the menu at the top of the application.



The screenshot shows the Cloudera Machine Learning interface. The top navigation bar has tabs: HOME, SQL, VISUALS, and DATASETS. The DATASETS tab is highlighted with a yellow circle and a mouse cursor. On the left sidebar, under the Data category, there is a 'Data' link. The main content area shows a 'Get Started' section with five numbered steps: 1. Sync Connections, 2. Explore with SQL, 3. Create a Dashboard, 4. CML Notebook, and 5. What's Next? Below this is a summary bar with icons for Queries (0), Dashboards (16), Apps (1), Datasets (12), and Connections (2). The central part of the screen displays 'User Settings' for 'cloudera_user'. It includes sections for Profile, Outbound SSH, API Keys, Remote Editing, and Environment Variables. The Environment Variables section is expanded, showing 'Reserved environment variables' and a table with rows for WORKLOAD_PASSWORD and WORKLOAD_PASSWORD. A note says: 'Data Connections may already be setup for you to use on the datasets page. If they are not, you will need to set up your workspace password.' Below this are instructions: 1. On the Cloudera Management Console, select your username. 2. View your profile, and set a workload password using the 'Set Workload Password' link. 3. In your ML Workspace, go to 'User Settings' page. 4. Under 'Environmental Variables' tab, set your 'WORKLOAD_PASSWORD' variable to the same password from your Cloudera Management Console profile. 5. Restart the 'Data Discovery and Visualization' application in CML.

6. The Data application always contains a `samples` dataset. Additional datasets are inherited from the project and CML workspace. Click the data warehouse dataset.



The screenshot shows the Cloudera Machine Learning interface with the Datasets tab selected. The left sidebar shows the 'Data' category with a 'samples' link. The main content area displays a list of datasets. The 'Datasets' tab is highlighted with a yellow circle and a mouse cursor. The table lists the following datasets:

Title/Table	ID	Created	Last Updated	Modified By	# Dashboards
Food Stores Inspection in NYC main.retail_food_store_inspections_current_critical_vio...	12	Sep 05, 2022	a minute ago	vizapps_admin	3
Cereals main.cereals	11	Sep 05, 2022	a minute ago	vizapps_admin	1
Earthquake Data January 2019 main.earthquake_data2019	10	Sep 05, 2022	a minute ago	vizapps_admin	1
World Life Expectancy main.world_life_expectancy	9	Sep 05, 2022	a minute ago	vizapps_admin	1
US County Population main.us_counties	8	Sep 05, 2022	a minute ago	vizapps_admin	1
US State Populations Over Time main.census_pop	7	Sep 05, 2022	a minute ago	vizapps_admin	1
Global Information Security Threats main.infoseq_1559	6	Sep 05, 2022	a minute ago	vizapps_admin	1
Restaurant Inspection SF main.restaurant_scores_lives_standard	5	Sep 05, 2022	a minute ago	vizapps_admin	1

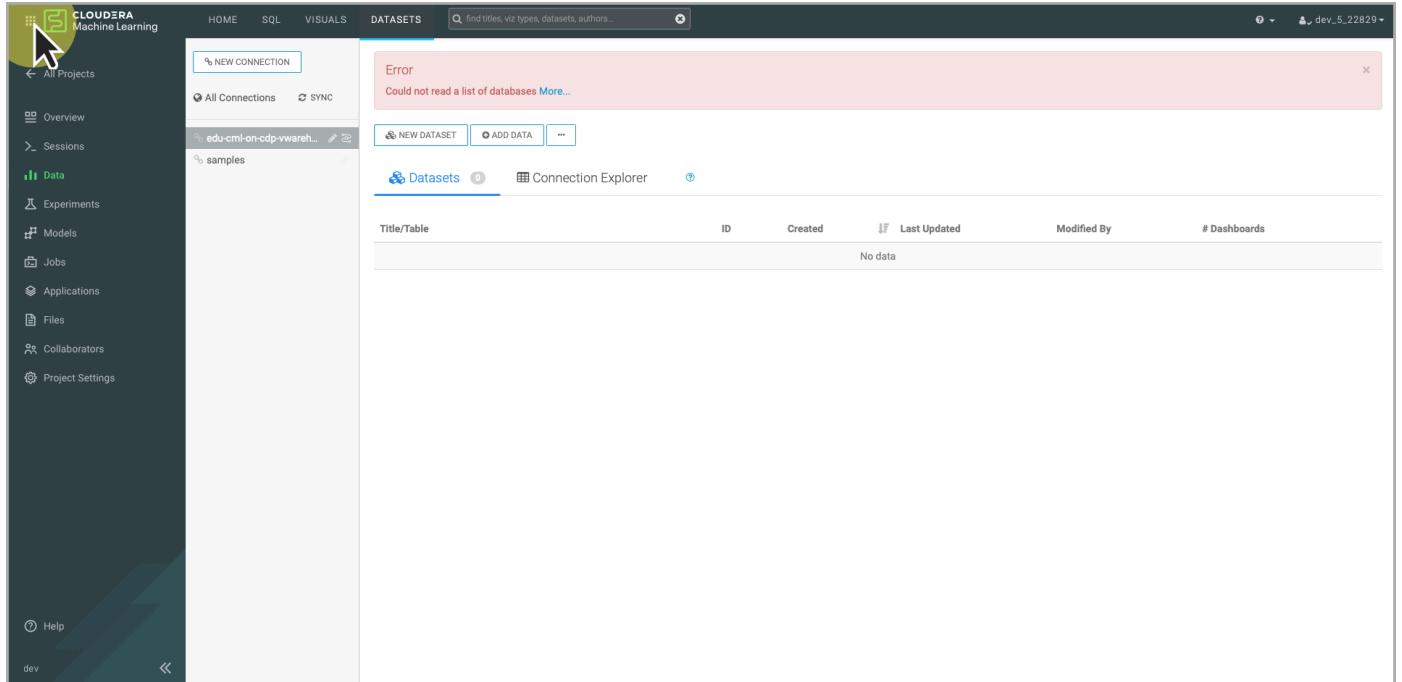
7. If this is your first time accessing a dataset, you will receive an error that Data application cannot read a list of databases. This is because the Data application needs a Workload Password to access the data warehouse.

The screenshot shows the Cloudera Data application interface. The left sidebar contains navigation links such as All Projects, Overview, Sessions, Data, Experiments, Models, Jobs, Applications, Files, Collaborators, and Project Settings. The main area has tabs for HOME, SQL, VISUALS, and DATASETS. The DATASETS tab is selected, showing a search bar and buttons for NEW CONNECTION, NEW DATASET, ADD DATA, and more. A prominent red error message box states: "Error" and "Could not read a list of databases More...". Below the error message is a table header for "Datasets" with columns: Title/Table, ID, Created, Last Updated, Modified By, and # Dashboards. The table body displays the message "No data".

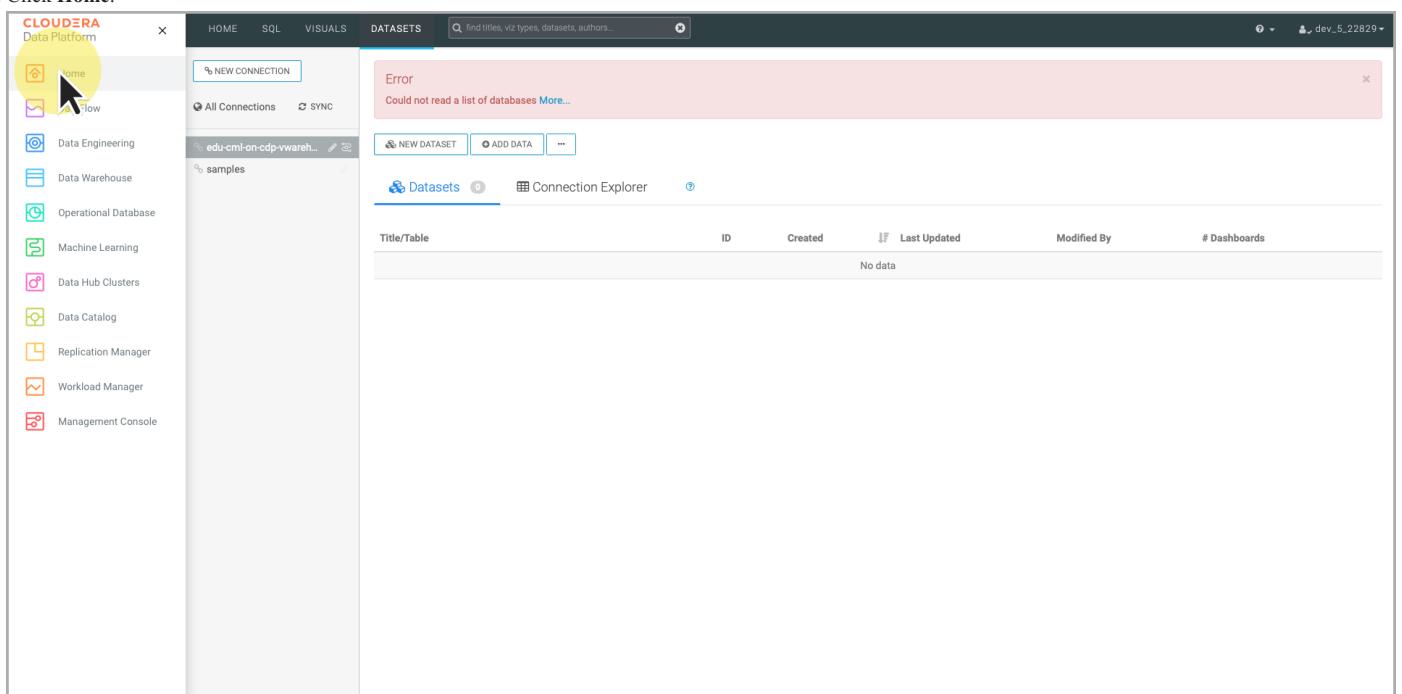
Create Workload Password (if needed)

If you received an error on the prior step, continue to follow the instructions below. If you did not receive an error, skip ahead to [create a new dataset](#).

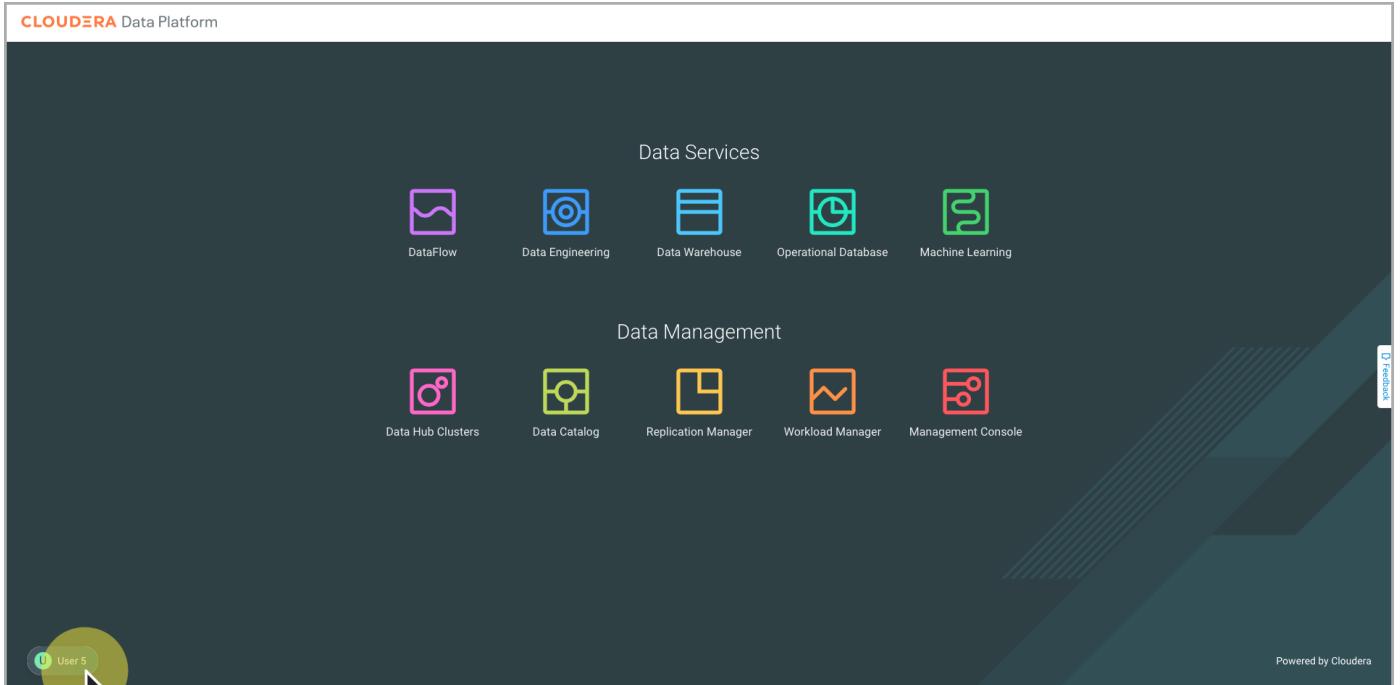
1. Click the **Main Menu** in the upper left corner.



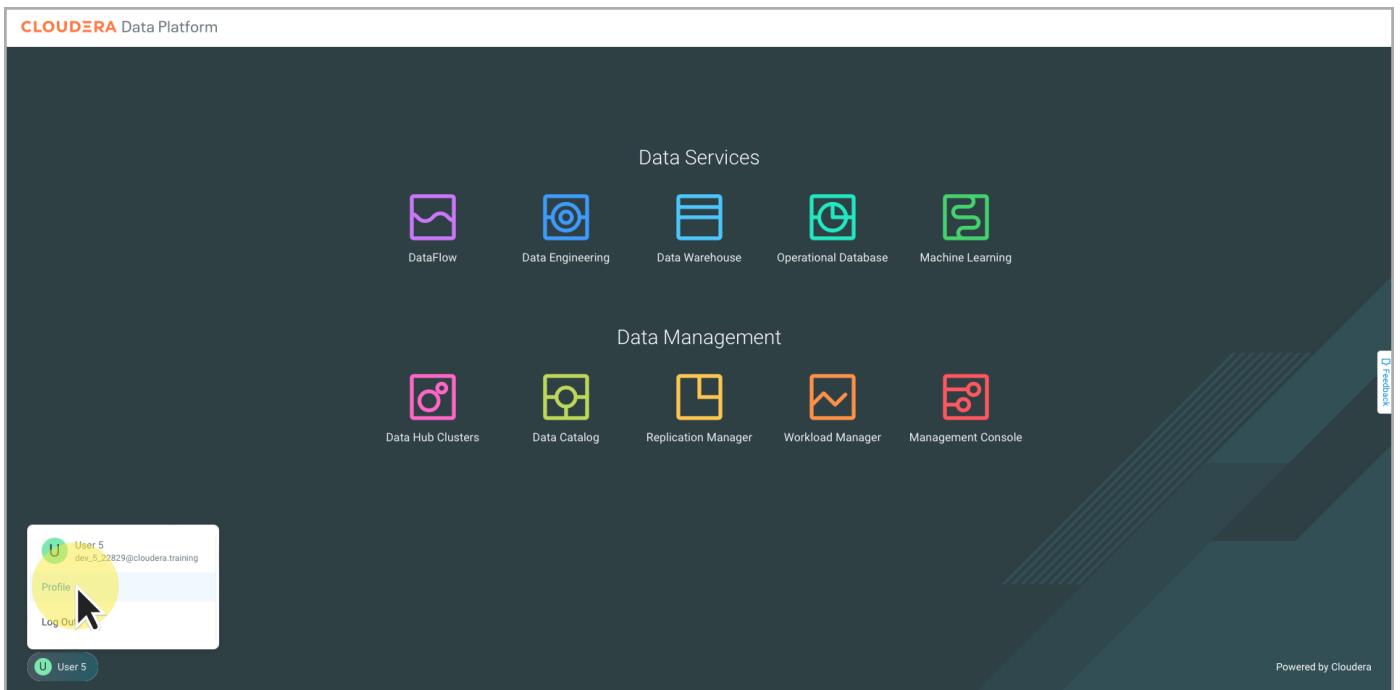
2. Click **Home**.



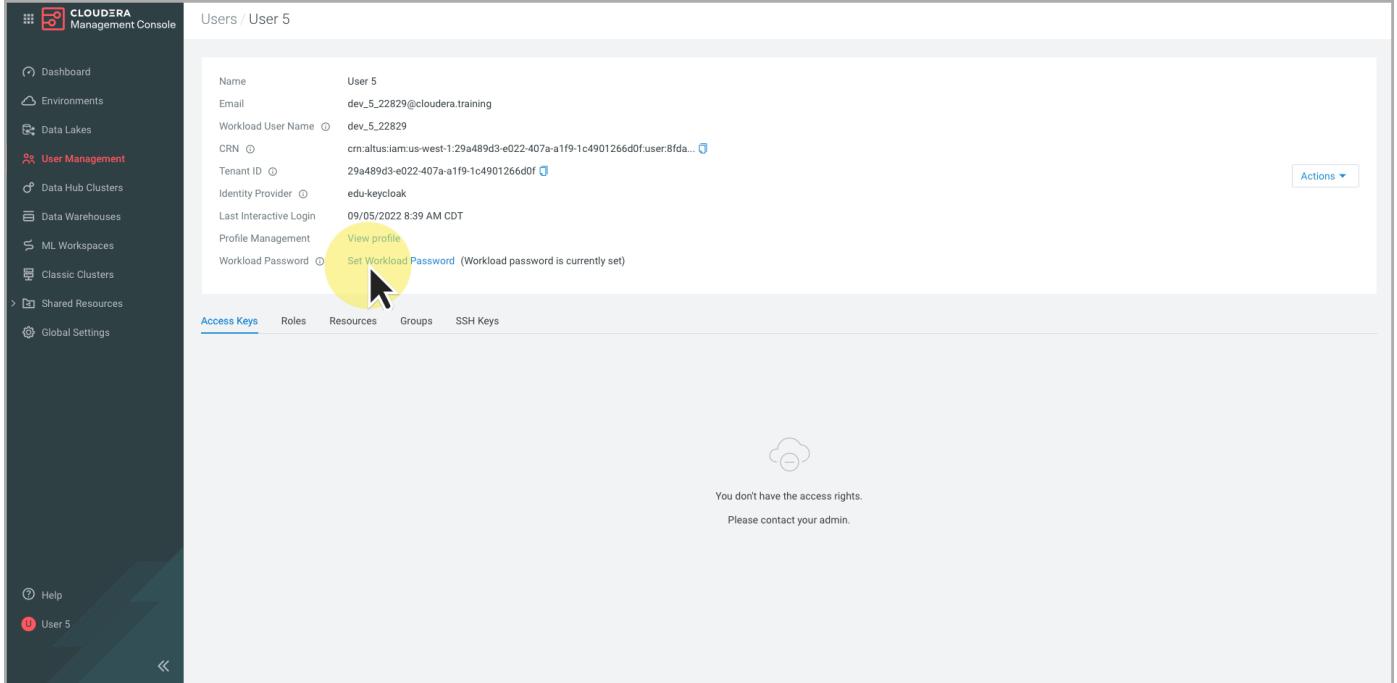
3. Click on your username in the lower right corner.



4. Click Profile.

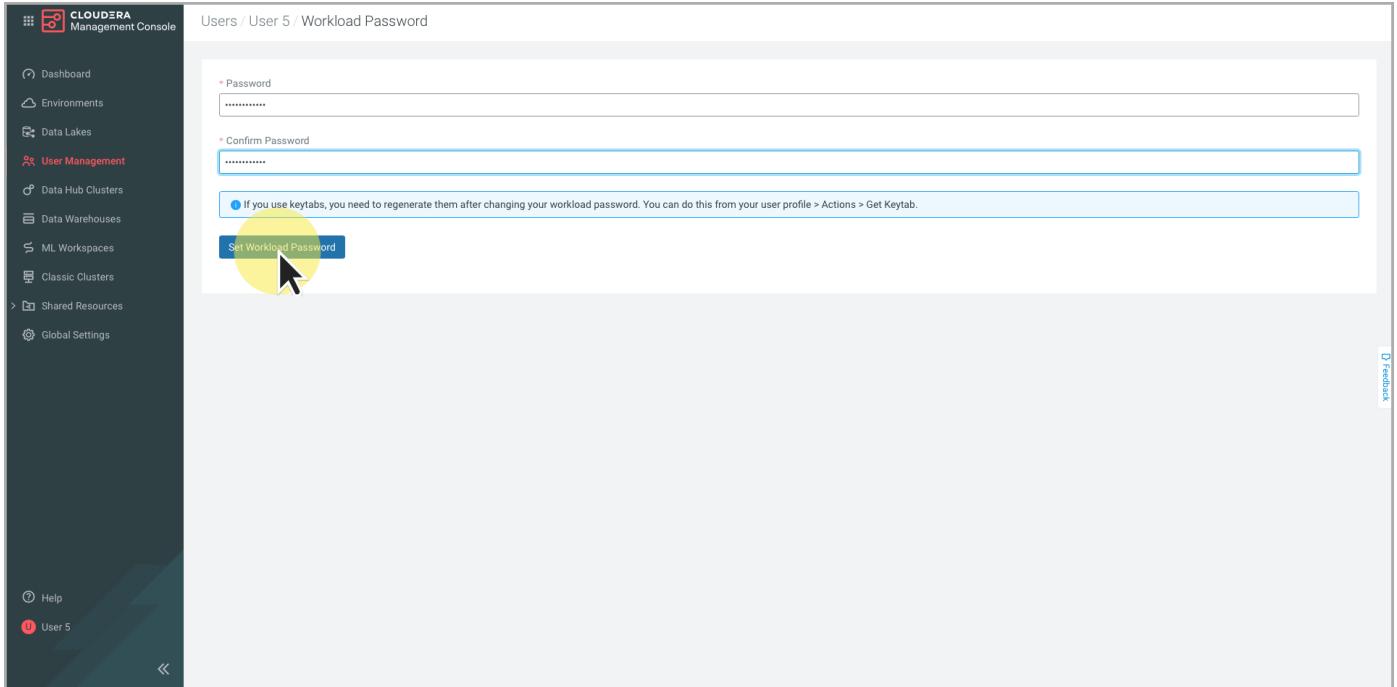


5. Click Set Workload Password.

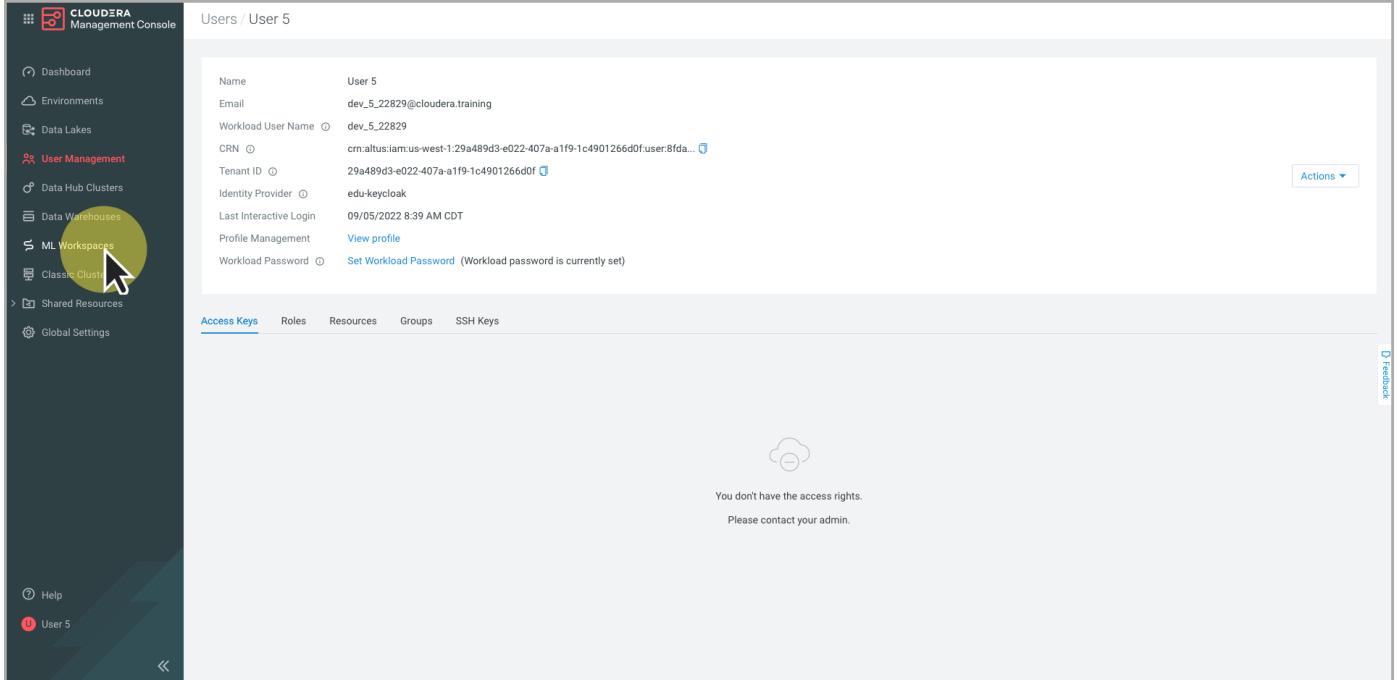


The screenshot shows the Cloudera Management Console interface. On the left is a dark sidebar with various navigation options like Dashboard, Environments, Data Lakes, User Management, etc. The main area is titled 'Users / User 5'. It displays detailed user information: Name (User 5), Email (dev_5_22829@cloudera.training), Workload User Name (dev_5_22829), CRN (cm.altus.iam.us-west-1:29a489d3-e022-407a-a1f9-1c4901266d0f), Tenant ID (29a489d3-e022-407a-a1f9-1c4901266d0f), Identity Provider (edu-keycloak), Last Interactive Login (09/05/2022 8:39 AM CDT), and Profile Management (View profile). Below this, it says 'Workload Password' followed by a blue link 'Password' (Workload password is currently set). A yellow circle with a cursor icon is placed over the 'Set Workload Password' button. At the bottom, there are tabs for Access Keys, Roles, Resources, Groups, and SSH Keys, with 'Access Keys' being the active tab.

6. Enter your workload password and click Set Workload Password.

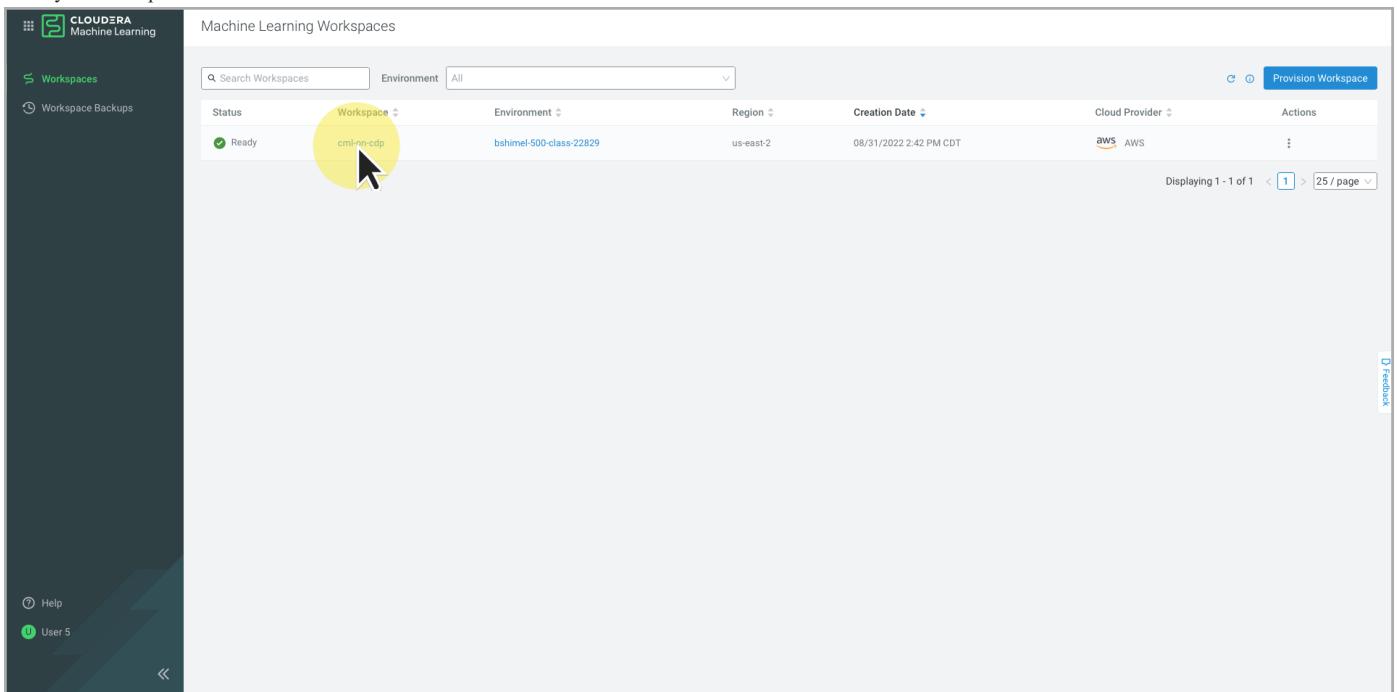


This screenshot shows the 'Workload Password' sub-page for User 5. The sidebar and top navigation are identical to the previous screenshot. The main content area has two input fields: 'Password' and 'Confirm Password', both containing placeholder text '*****'. Below these fields is a note: 'If you use keytabs, you need to regenerate them after changing your workload password. You can do this from your user profile > Actions > Get Keytab.' At the bottom is a blue button labeled 'Set Workload Password' with a yellow circle and cursor icon highlighting it.

7. Click **ML Workspaces** in the left menu.


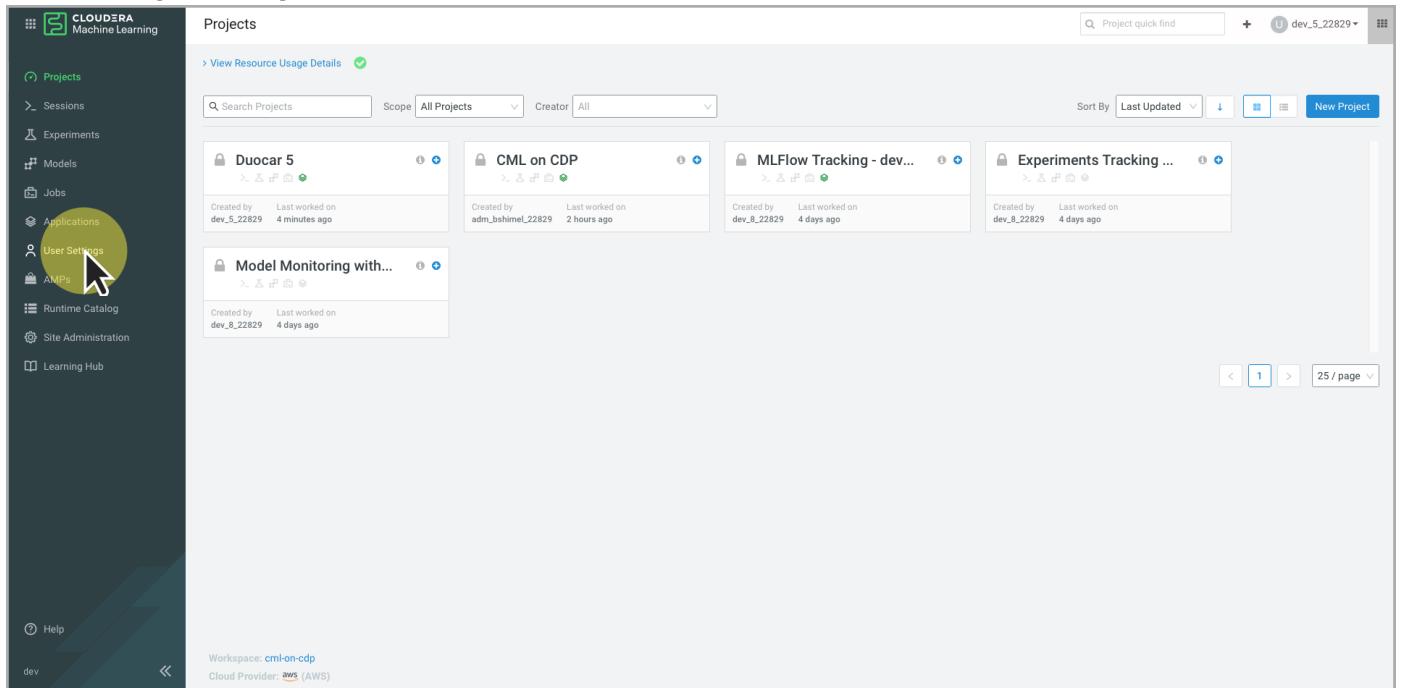
The screenshot shows the Cloudera Management Console interface. On the left, a sidebar lists various management options like Dashboard, Environments, Data Lakes, User Management, Data Hub Clusters, Data Warehouses, ML Workspaces (which is highlighted with a yellow circle), and Classic Clusters. The main content area is titled 'Users / User 5' and displays detailed information for 'User 5': Name (User 5), Email (dev_5_22829@cloudera.training), Workload User Name (dev_5_22829), CRN (cm:altru:iam:us-west-1:29a489d3-e022-407a-a1f9-1c4901266d0f:user:8fd...), Tenant ID (29a489d3-e022-407a-a1f9-1c4901266d0f), Identity Provider (edu-keycloak), Last Interactive Login (09/05/2022 8:39 AM CDT), Profile Management (View profile), and Workload Password (Set Workload Password (Workload password is currently set)). Below this, there are tabs for Access Keys, Roles, Resources, Groups, and SSH Keys, with 'Access Keys' being the active tab. A note at the bottom states: 'You don't have the access rights. Please contact your admin.' There is also a 'Feedback' link on the right.

8. Click your workspace.



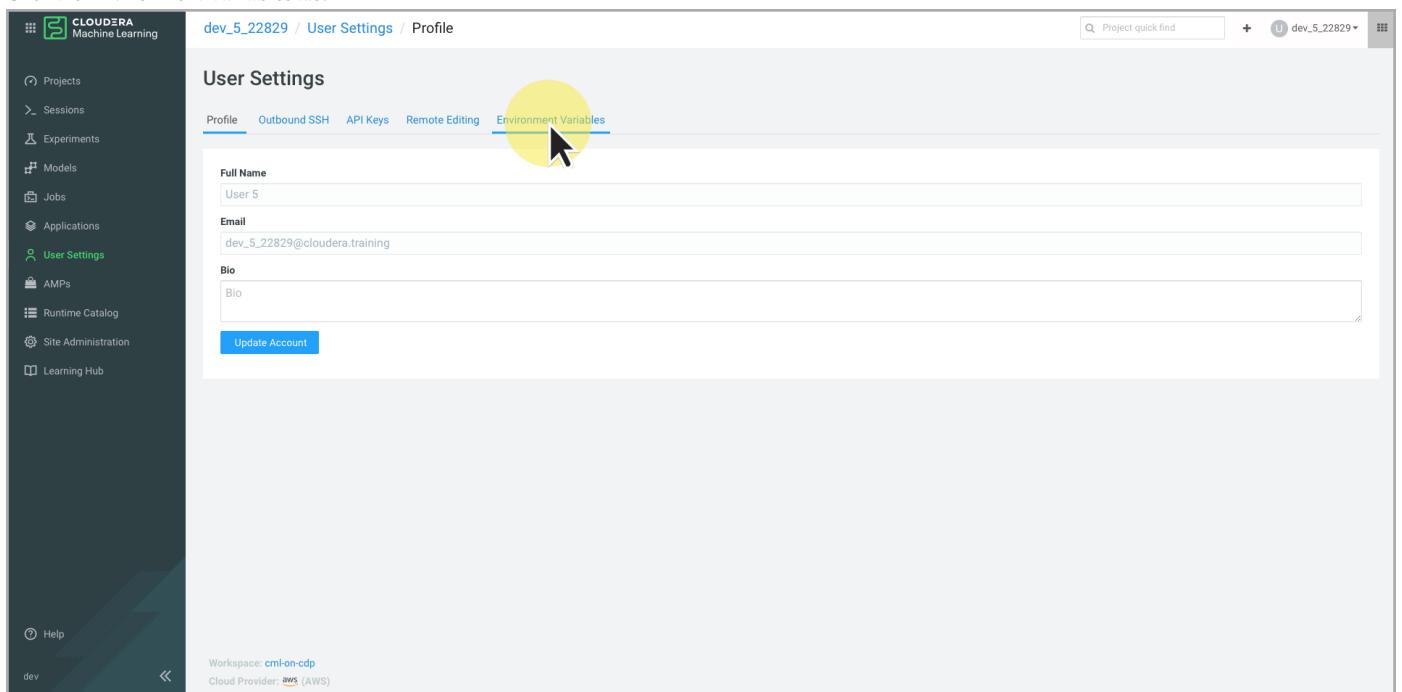
The screenshot shows the Cloudera Machine Learning interface. On the left, a sidebar lists Workspaces and Workspace Backups. The main content area is titled 'Machine Learning Workspaces' and displays a table of workspaces. The columns include Status (Ready), Workspace (cmi-ml-edp), Environment (bshimel-500-class-22829), Region (us-east-2), Creation Date (08/31/2022 2:42 PM CDT), Cloud Provider (aws AWS), and Actions. A note at the bottom says 'Displaying 1 - 1 of 1 < [1] > [25 / page]'. There is also a 'Feedback' link on the right.

9. Click User Settings in the workspace menu on the left.



The screenshot shows the Cloudera Machine Learning interface. On the left, there is a dark sidebar with various navigation options: Projects, Sessions, Experiments, Models, Jobs, Applications, **User Settings** (which is highlighted with a yellow circle), AMPs, Runtime Catalog, Site Administration, and Learning Hub. Below the sidebar, it says 'dev' and has a 'Help' link. At the bottom, it shows 'Workspace: cml-on-cdp' and 'Cloud Provider: AWS (AWS)'. The main area is titled 'Projects' and lists several projects: Duocar 5, CML on CDP, MLFlow Tracking - dev..., Experiments Tracking ..., and Model Monitoring with... (with a yellow circle around it). There are search and filter tools at the top of the project list.

10. Click the Environment Variables tab.



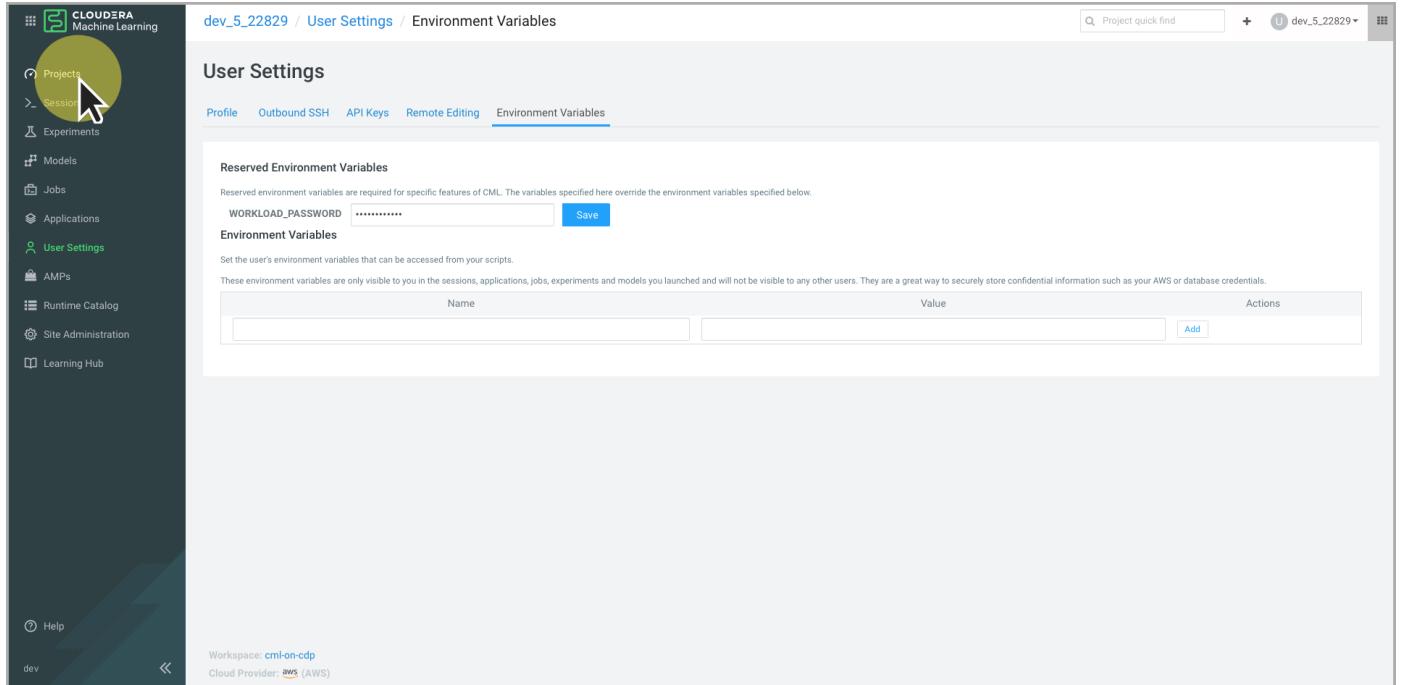
The screenshot shows the 'User Settings' page under 'Profile'. The top navigation bar includes tabs for Profile, Outbound SSH, API Keys, Remote Editing, and **Environment Variables** (which is highlighted with a yellow circle). Below the tabs, there are fields for Full Name (User 5), Email (dev_5_22829@cloudera.training), and Bio (Bio). At the bottom is a blue 'Update Account' button. The left sidebar is identical to the one in the previous screenshot. The bottom of the page shows 'Workspace: cml-on-cdp' and 'Cloud Provider: AWS (AWS)'.

11. Enter the workload password you just created and click **Save**.

The screenshot shows the Cloudera Machine Learning interface. On the left is a dark sidebar with various navigation options: Projects, Sessions, Experiments, Models, Jobs, Applications, User Settings (which is selected and highlighted in green), AMPs, Runtime Catalog, Site Administration, and Learning Hub. At the bottom of the sidebar are Help and Dev links. The main content area has a header 'User Settings' with tabs for Profile, Outbound SSH, API Keys, Remote Editing, and Environment Variables (which is underlined). Below the tabs is a section titled 'Reserved Environment Variables' with a note about overriding environment variables. A text input field contains 'WORKLOAD_PASSWORD' followed by a series of asterisks. To the right of the input field is a yellow circle containing a cursor icon pointing at a green 'Save' button. Below this section is another titled 'Environment Variables' with a note about visibility. A table with columns 'Name', 'Value', and 'Actions' is shown, with one row currently listed. At the bottom of the page, it says 'Workspace: cml-on-cdp' and 'Cloud Provider: aws (AWS)'. The top right corner shows a project quick find bar and a user icon.

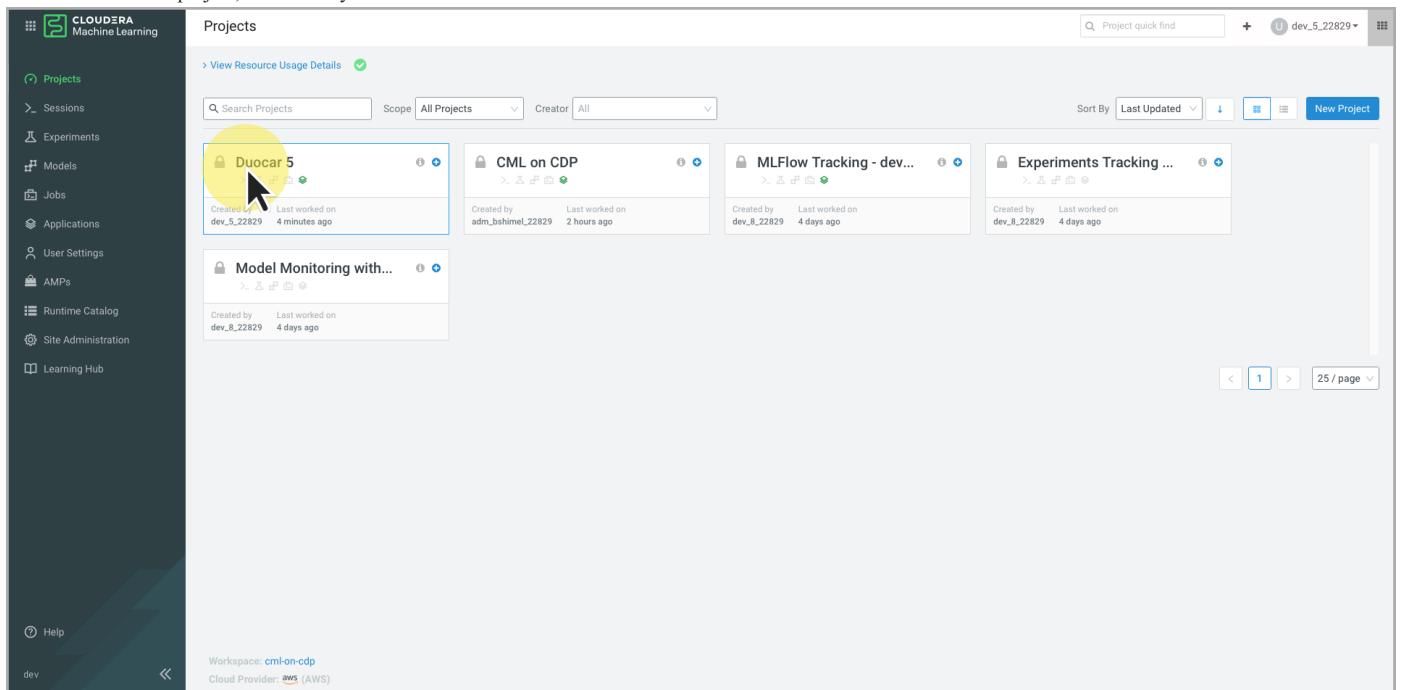
Now that your workload password has been set. The **Data Discovery and Visualization** application needs to be restarted.

1. Click Projects.



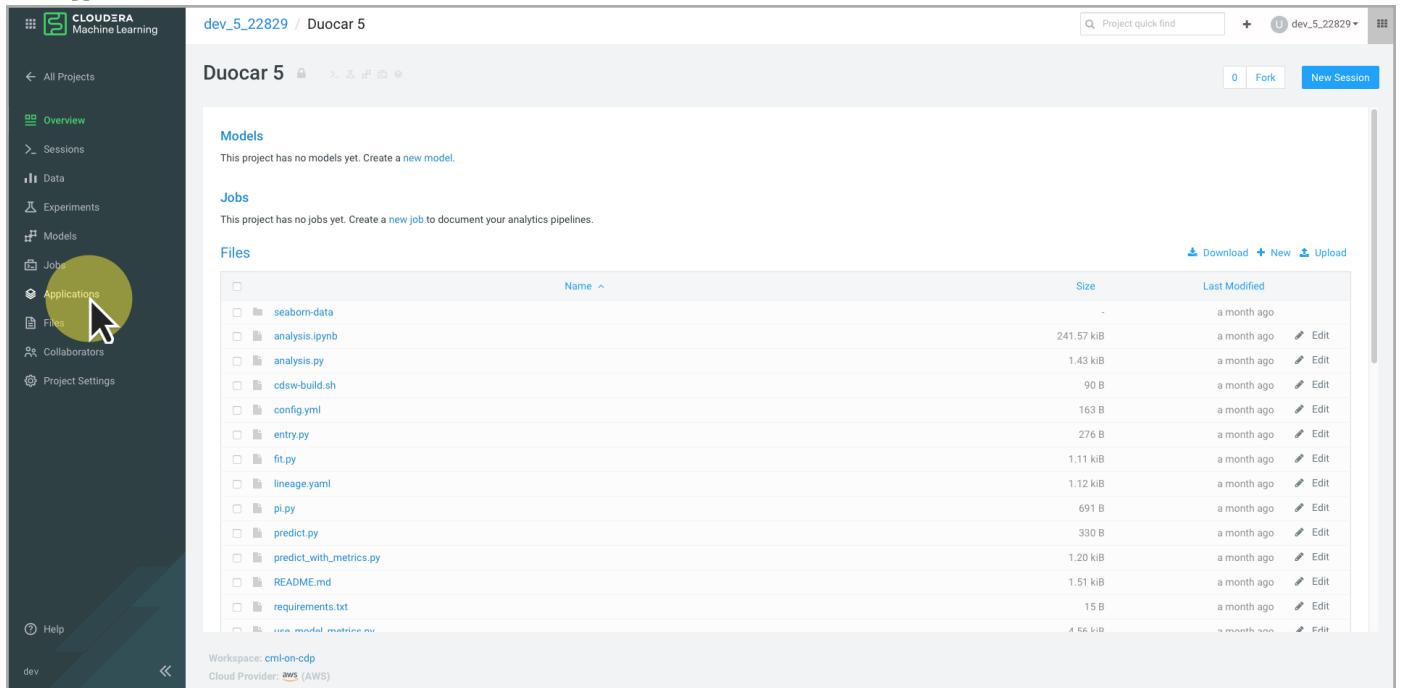
The screenshot shows the Cloudera Machine Learning interface. The left sidebar has a 'Projects' link highlighted with a yellow circle. The main content area is titled 'User Settings' with tabs for Profile, Outbound SSH, API Keys, Remote Editing, and Environment Variables. Under 'Environment Variables', there's a section for 'Reserved Environment Variables' where 'WORKLOAD_PASSWORD' is set to '*****'. Below it is a table for 'Environment Variables' with one row added. The bottom status bar shows 'Workspace: cml-on-cdp' and 'Cloud Provider: AWS (AWS)'.

2. Click the Duocar X project, where X is your student number.



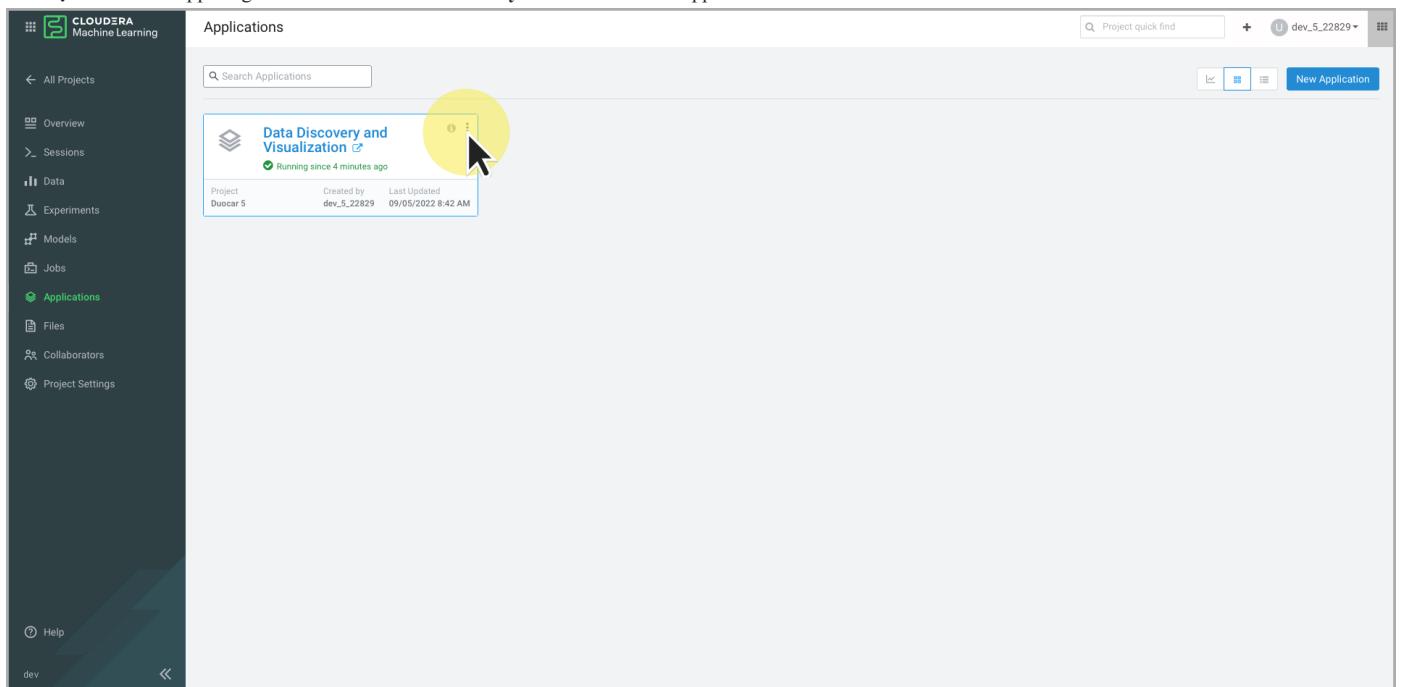
The screenshot shows the 'Projects' page. A yellow circle highlights the 'Duocar 5' project card, which is the first item in the list. Other projects shown are 'CML on CDP', 'MLFlow Tracking - dev...', and 'Experiments Tracking ...'. The bottom status bar shows 'Workspace: cml-on-cdp' and 'Cloud Provider: AWS (AWS)'.

3. Click Applications.



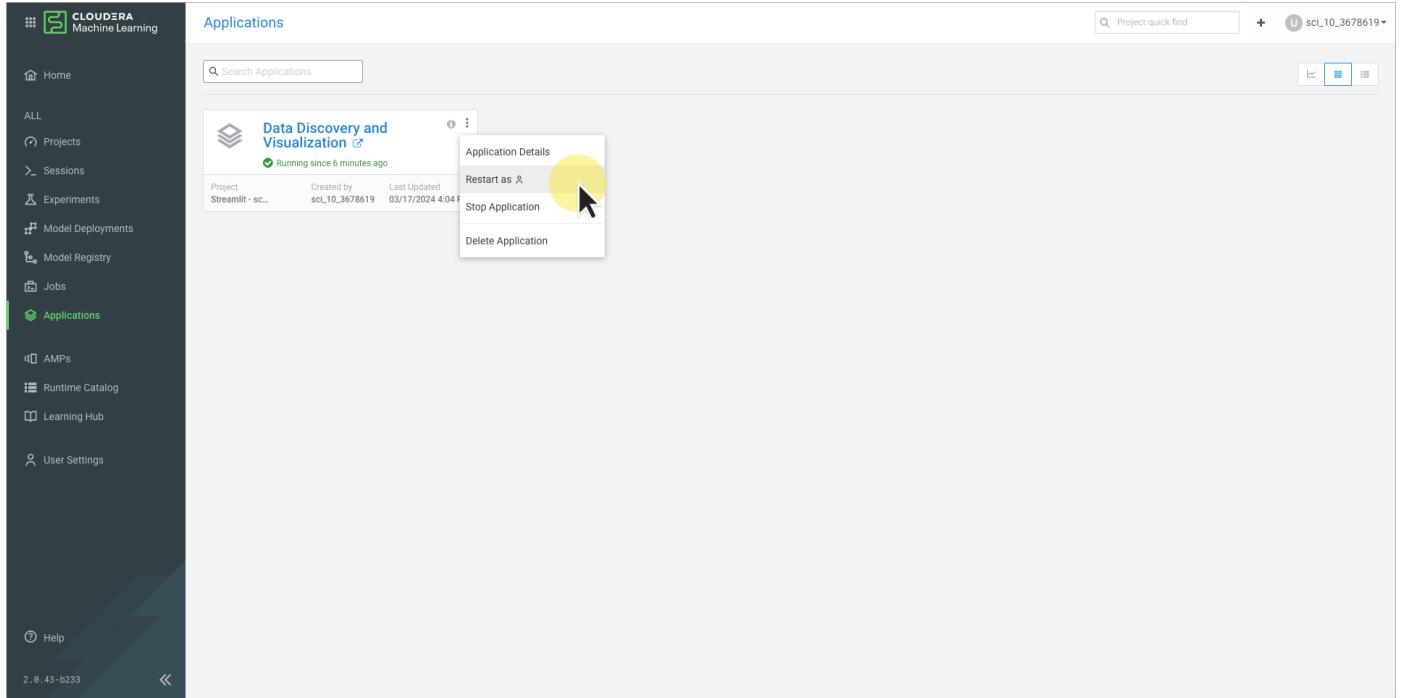
The screenshot shows the Cloudera Machine Learning interface. On the left sidebar, under the 'Applications' heading, there is a circular icon with a cursor arrow pointing to it. The main workspace is titled 'Duocar 5' and contains sections for 'Models' (with a note about no models yet), 'Jobs' (with a note about no jobs yet), and 'Files'. The 'Files' section displays a list of files in a table format. The table has columns for 'Name', 'Size', and 'Last Modified'. The files listed include 'seaborn-data', 'analysis.ipynb', 'analysis.py', 'cdsw-build.sh', 'config.yml', 'entry.py', 'fit.py', 'lineage.yaml', 'pi.py', 'predict.py', 'predict_with_metrics.py', 'README.md', and 'requirements.txt'. The last file listed is 'use model machine.py'.

4. Click ; menu in the upper right corner of the Data Discovery and Visualization application.

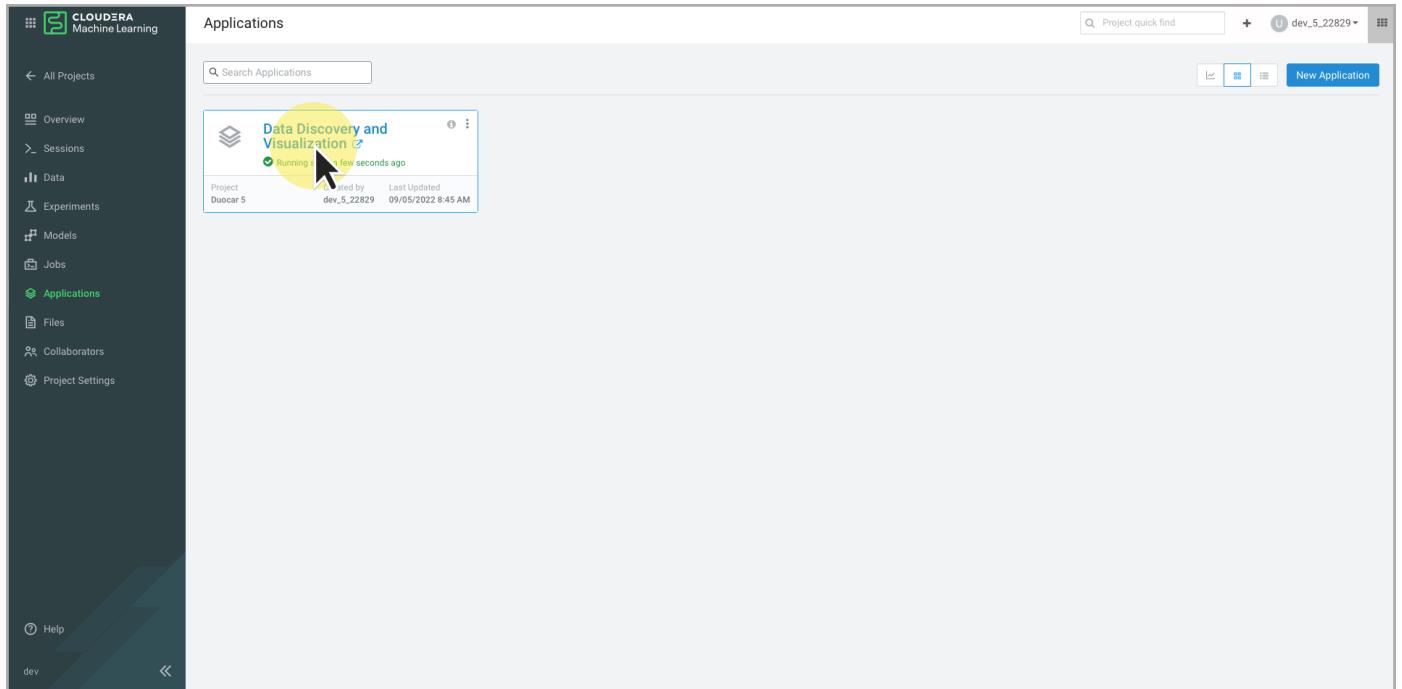


The screenshot shows the Cloudera Machine Learning interface. On the left sidebar, under the 'Applications' heading, there is a circular icon with a cursor arrow pointing to it. The main workspace is titled 'Applications' and shows a card for the 'Data Discovery and Visualization' application. The card includes a logo, the application name, a status message 'Running since 4 minutes ago', and details like 'Project Duocar 5', 'Created by dev_5_22829', and 'Last Updated 09/05/2022 8:42 AM'. The 'New Application' button is located in the top right corner of the application card.

5. Click Restart as [user icon].



6. Click the Data Discovery and Visualization application.

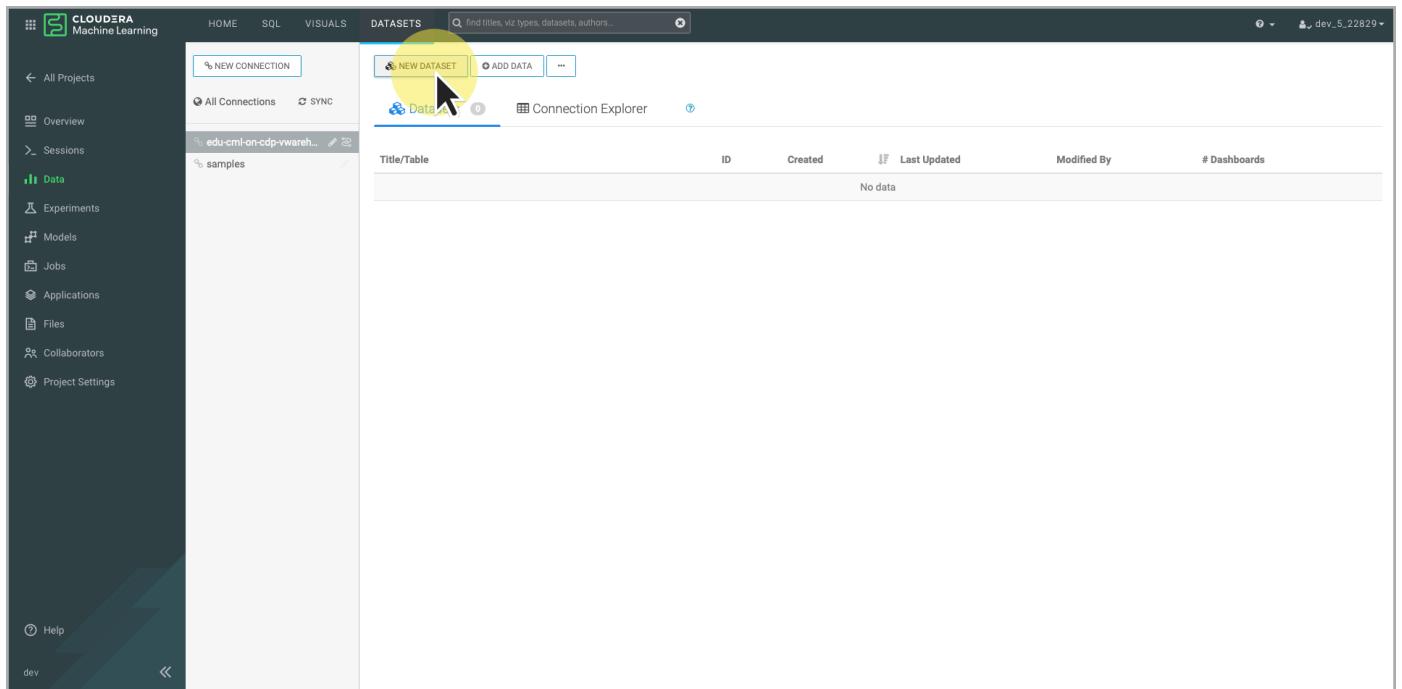


7. Click Datasets.

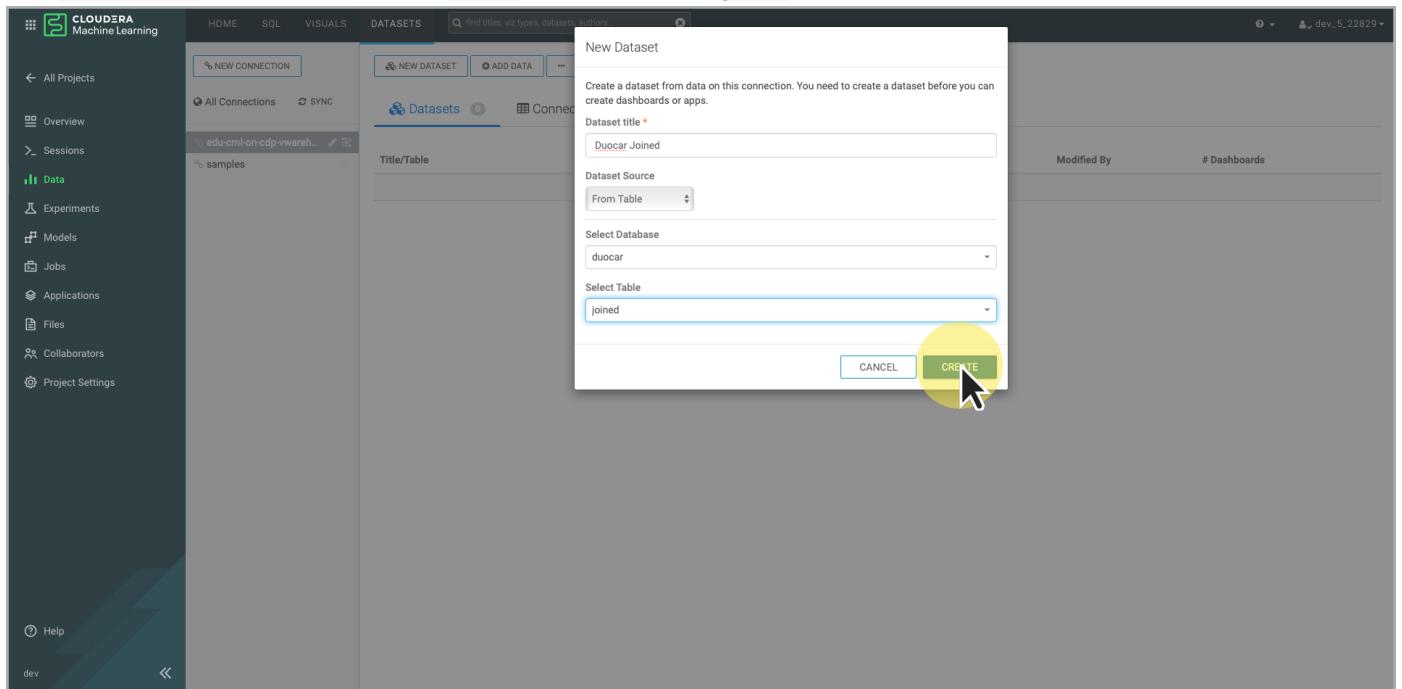
8. Click on your data warehouse connection.

Create a New Dataset

1. Click New Dataset



2. Enter Duocar Joined for the Dataset Title. Select duocar as the database. Select joined as the table. Click Create.



3. Click the newly created dataset, **Duocar Joined**.

The screenshot shows the Cloudera Machine Learning interface with the 'DATASETS' tab selected. On the left, there's a sidebar with various project management and data-related links. The main area displays a table of datasets. One dataset, 'Duocar Joined', is highlighted with a yellow circle and a cursor pointing at it. The table columns include 'Title/Table', 'ID', 'Created', 'Last Updated', 'Modified By', and '# Dashboards'. The 'Duocar Joined' entry has ID 13, was created on Sep 05, 2022, and last updated a few seconds ago by 'dev_5_22829'.

4. Click Data Model in the Dataset Detail menu.

The screenshot shows the 'Dataset Detail' page for 'Duocar Joined'. The left sidebar lists several sections: 'Dataset Detail', 'Related Dashboards', 'Fields', 'Data Model' (which is highlighted with a yellow circle and has a cursor over it), 'Time Slicing', 'Segments', 'Filter Associations', and 'Permissions'. The main content area displays dataset details like 'Dataset: Duocar Joined', 'Table: duocar.joined', and 'Connection Type: Hive'. It also shows the data source ('Data Connection: edu-cml-on-cdp-vwarehouse') and various configuration settings. At the bottom, it provides metadata such as 'ID: 13', 'Created on: Sep 05, 2022 08:46 AM', and 'Last updated: Sep 05, 2022 08:46 AM'.

5. Click **Show Data**. This is a quick test to verify the dataset is working and has data.

The screenshot shows the 'Data Model' section of the Cloudera Machine Learning application. On the left, there's a sidebar with various project management and data-related tabs like 'All Projects', 'Overview', 'Sessions', 'Data', 'Experiments', etc. The 'Data Model' tab is currently selected. In the main area, there's a 'joined' dataset detail card. Below it, a large green button labeled 'SHOW DATA' is highlighted with a yellow circle and a cursor is clicking on it. There's also a smaller button labeled 'Apply Display Format' below it.

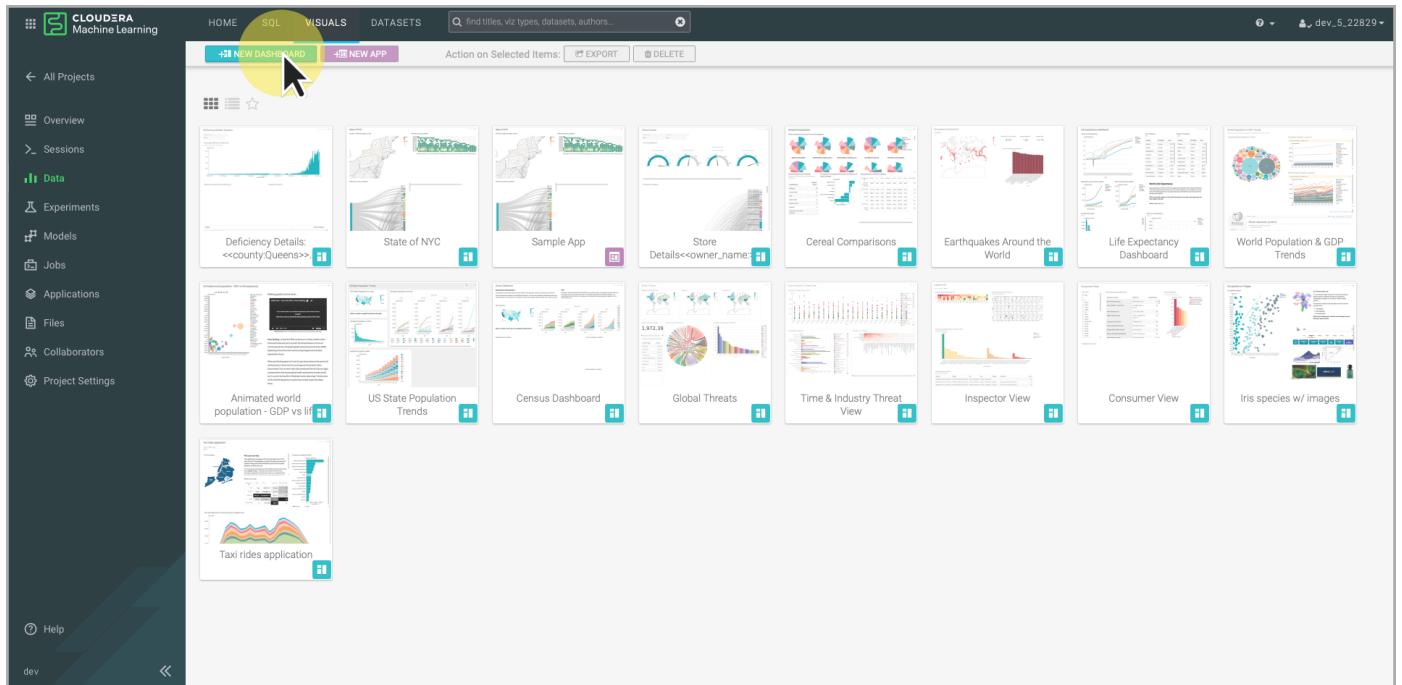
Create a Dashboard

1. Click **Visuals** in the menu at the top of the application.

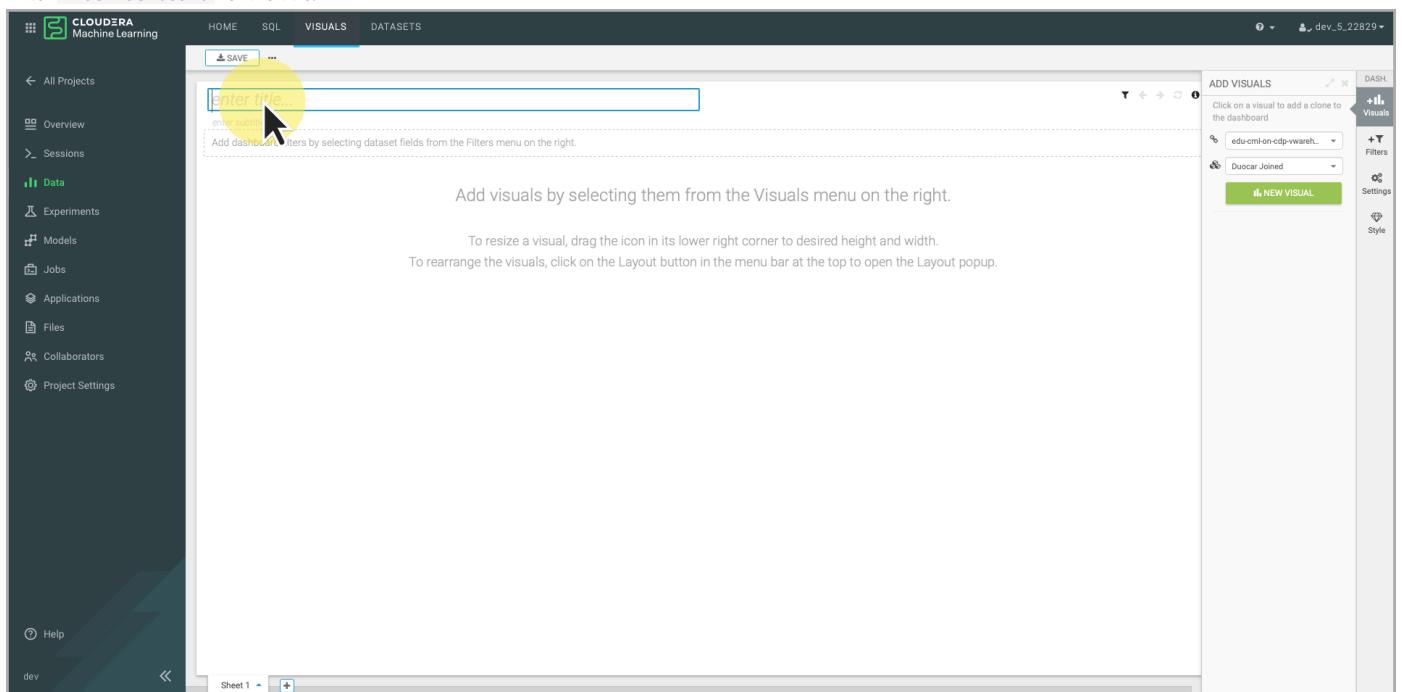
This screenshot shows the 'VISUALS' tab selected at the top of the application. The interface is similar to the previous one, with the 'Data Model' section visible. The 'VISUALS' tab is highlighted with a yellow circle. Below it, the 'SHOW DATA' button is now labeled 'HIDE DATA' with a checked checkbox next to it. The table of data rows is still present.

ride_id	rider_id	driver_id	date_time	utc_offset	service	origin_lat	origin_lon	dest_lat	dest_lon	distance	duration	cancelled	star_rating	driver_birth_date	driver_start_date	driver_
0000000001	220200000084	220200000214	2017-02-01 00:14:00	-6	Car	46.850956	-96.902849	46.860050	-96.825442	10123	729	false	5	1985-08-31 00:00:00	2017-01-12 00:00:00	Ryan
0000000002	220200000462	220200000107	2017-02-01 00:36:00	-6	Car	46.900432	-96.765807	46.840588	-96.868087	16043	1299	false	5	1968-06-03 00:00:00	2017-01-06 00:00:00	Dillon
0000000003	220200000489	220200000214	2017-02-01 02:26:00	-6	Noir	46.868382	-96.902718	46.815272	-96.862056	9362	736	false	5	1985-08-31 00:00:00	2017-01-12 00:00:00	Ryan
0000000004	220200000057	220200000067	2017-02-01 03:00:00	-6	Car	46.908567	-96.905391	46.904380	-96.793999	9060	773	false	5	1993-03-08 00:00:00	2017-01-03 00:00:00	Dale
0000000005	220200000012	220200000067	2017-02-01 03:49:00	-6	Car	46.895864	-96.805807	46.869030	-96.785232	5076	721	false	5	1993-03-08 00:00:00	2017-01-03 00:00:00	Dale
0000000006	220200000157	220200000214	2017-02-01 03:53:00	-6	Car	46.840562	-96.916740	46.850929	-96.883703	4683	427	false	5	1985-08-31 00:00:00	2017-01-12 00:00:00	Ryan
0000000007	220200000499	220200000067	2017-02-01 04:13:00	-6	Car	46.920253	-96.780229	46.874190	-96.788312	6230	699	false	3	1993-03-08 00:00:00	2017-01-03 00:00:00	Dale
0000000008	220200000256	220200000214	2017-02-01 04:13:00	-6	Car	46.846283	-96.801273	46.874445	-96.793427	3824	501	false	2	1985-08-31 00:00:00	2017-01-12 00:00:00	Ryan

2. Click the New Dashboard button.



3. Enter Ride Dashboard for the title.



4. Click **Visuals** in the **DASH.** menu on the right. Click the **NEW VISUAL** button.

5. Click the **Bar Chart** icon. The bar chart properties are displayed.

6. Drag **service** from the list of **Dimensions** and drop it in the **X Axis** field.

The screenshot shows the Cloudera Machine Learning interface for creating a dashboard. On the left, there's a sidebar with various project and data management options like Overview, Sessions, Data, Experiments, Models, Jobs, Applications, Files, Collaborators, and Project Settings. The main area is titled 'Ride Dashboard' and contains a placeholder for a visual with the text 'enter title...' and 'enter subtitle...'. To the right is a large sidebar with several panels: Visuals, Data, Dimensions, Measures, and Filters. The 'Dimensions' panel is open, showing a list of fields under 'joined' such as ride_id, rider_id, driver_id, date_time, service, etc. A yellow circle highlights the 'service' field, which is being dragged into the 'X Axis' field, indicated by a dashed box. The 'Measures' panel also lists various metrics like Record Count, utc_offset, origin_lat, dest_lat, distance, duration, star_rating, and driver_home_lat.

7. Drag **Record Count** from the list of **Measures** and drop it in the **Y Axis** field.

This screenshot shows the Cloudera Data Platform interface for dashboard creation. The left sidebar lists various platform components: Home, DataFlow, Data Engineering, Data Warehouse, Operational Database, Machine Learning, Data Hub Clusters, Data Catalog, Replication Manager, Workload Manager, and Management Console. The main area displays two visual components: a bar chart titled 'Record Count' showing values for categories like 'Col', 'Elite', 'Guard', and 'Nerf', and a donut chart. Below these is another placeholder with 'enter title...' and 'enter subtitle...'. The right sidebar features the same panels as the previous screenshot, but the 'Measures' panel is currently active. It lists joined measures like Record Count, utc_offset, origin_lat, dest_lat, distance, duration, star_rating, and driver_home_lat. A yellow circle highlights the 'Record Count' measure, which is being dragged into the 'Y Axis' field of the bar chart's configuration area.

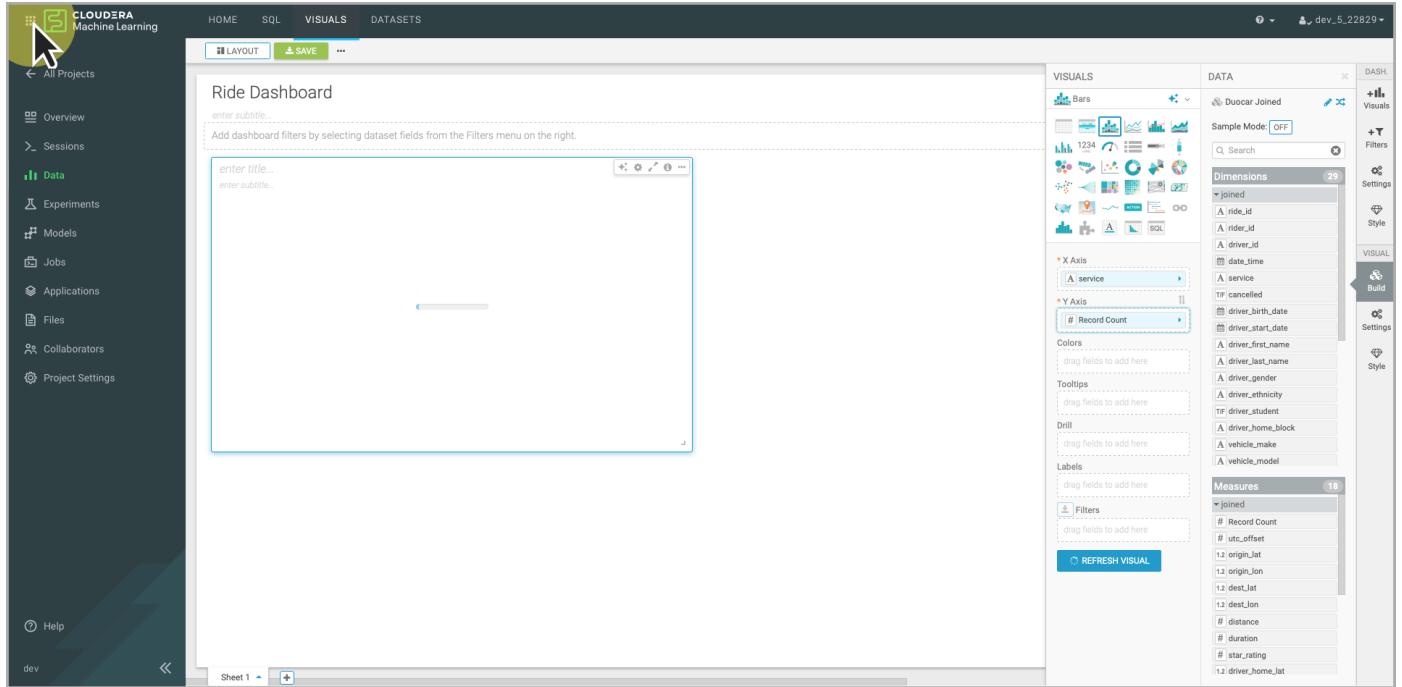
8. Click the Refresh Visual button.

The screenshot shows the Cloudera Machine Learning interface for creating a dashboard. The top navigation bar includes HOME, SQL, VISUALS, and DATASETS. The left sidebar contains links for All Projects, Overview, Sessions, Data, Experiments, Models, Jobs, Applications, Files, Collaborators, Project Settings, Help, and a dev section. The main area is titled 'Ride Dashboard' with placeholder text 'enter title...' and 'enter subtitle...'. The right side features a sidebar with sections for VISUALS, DATA, DASH., and various settings. The VISUALS section lists 'Bars', '1234', '3D Scatter', etc. The DATA section shows a dataset named 'Duocar Joined' with 'Sample Mode OFF'. The sidebar also includes Dimensions, Measures, and a 'Filters' section where the 'REFRESH VISUAL' button is highlighted with a yellow circle and a cursor icon.

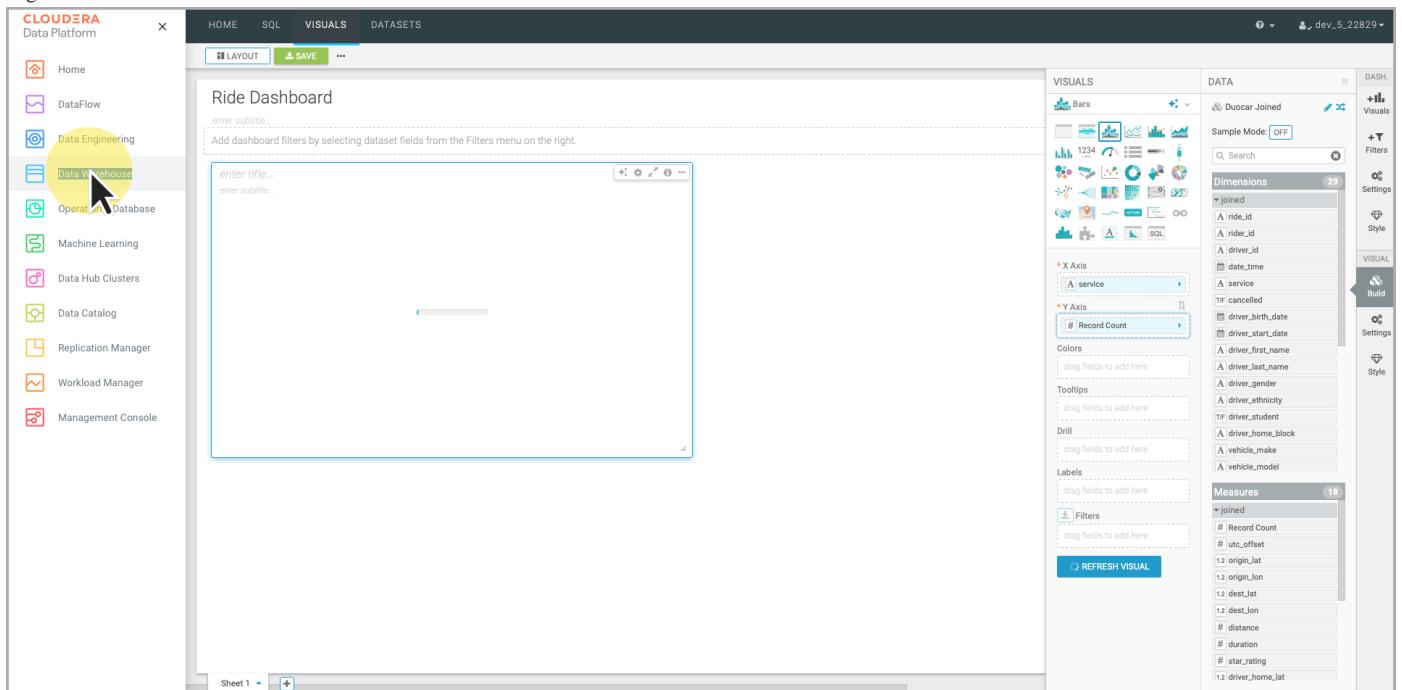
 Note

The first time you make a connection to the data, the data warehouse may need start. This will cause the visual to take a long time to update as it is waiting for the data. The next five steps check the status of the data warehouse and watch as it auto provisions pods and enters a running state. If your visual updates quickly, you can skip these steps.

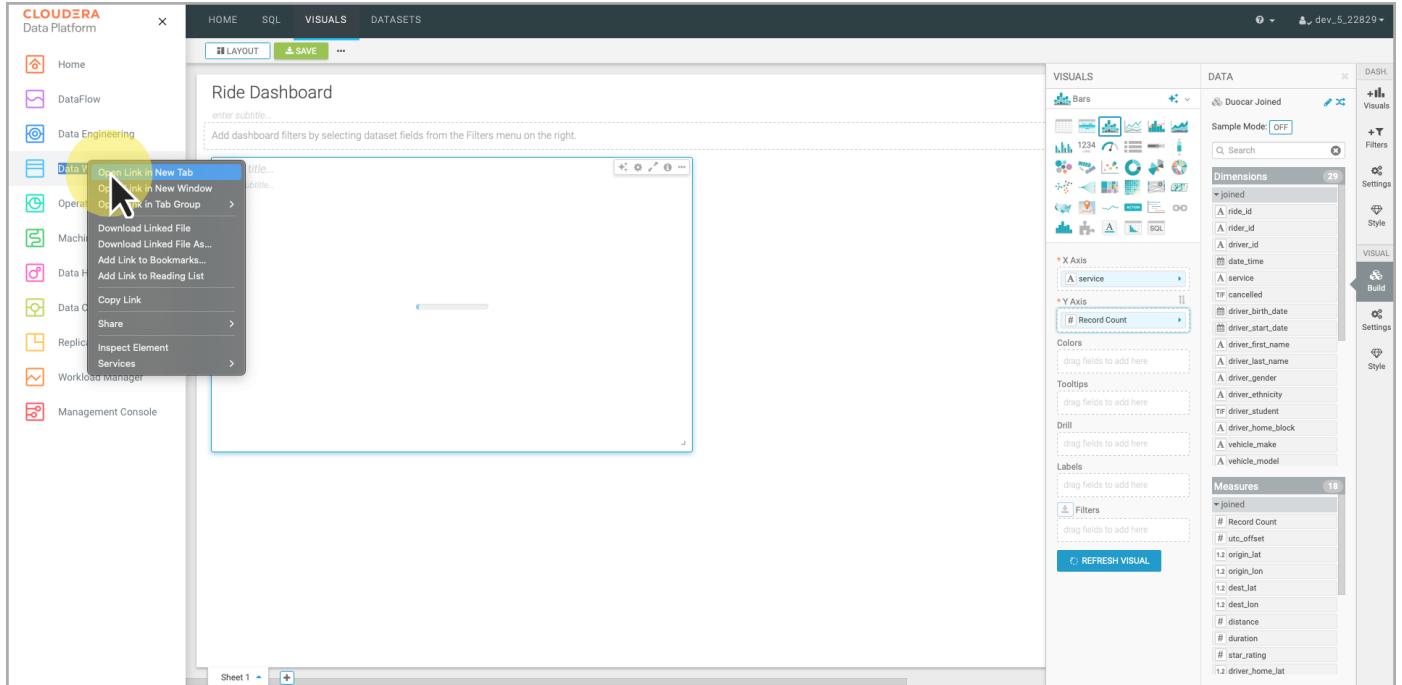
1. Click Main Menu.



2. Right-click on Data Warehouse.



3. Click Open in New Tab to open a new browser tab.



4. If your data warehouse was previously suspended, it shows Starting as it is provisioning resources.

Virtual Warehouses | 1

Virtual Warehouse	Status	Compute Resource
edu-cml-on-cdp-vwarehouse	Starting	compute-1662233143-psw datalake-bshimel-500-class-22829

5. Once the data warehouse has provisioned its resources, the state will change to **Running**. Close the browser tab and return to the tab with the **Data Discovery and Visualization** application.

CLOUDERA Data Warehouse

Overview

Database Catalogs | 1

- datalake-bschmel-500-class-...
Running

TOTAL CORES: 9 TOTAL MEMORY: 25 GB VIRTUAL WAREHOUSES: 1

Virtual Warehouses | 1

- edu-cml-on-cdp-vwarehouse
Running

EXECUTORS: 10 TOTAL CORES: 143 TOTAL MEMORY: 1.21 TB TYPE: HIVE UNIFIED ANALYTICS COMPACTOR

Help User 5 1.4.2-b118

6. Enter **Rides by Service** as the title for the bar chart visual.

CLOUDERA Data Platform

HOME SQL VISUALS DATASETS

Ride Dashboard

enter subtitle... Add dashboard filters by selecting dataset fields from the Filters menu on the right.

enter title... enter subtitle...

VISUALS

Pie Bar Line Area Scatter Heat Map Box Plot Histogram

DATA

Duocar Joined Sample Mode OFF

Dimensions

- # star_rating
- # Record Count
- rider_star_date

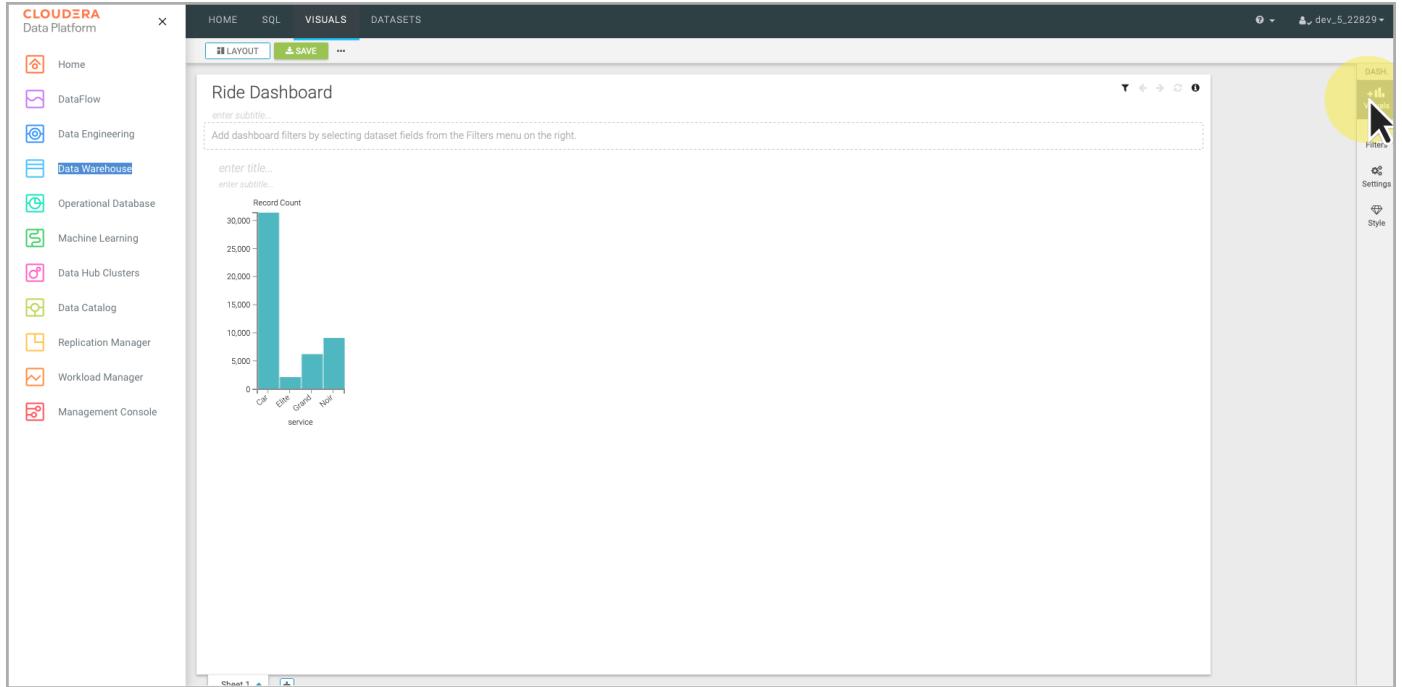
Measures

- joined
- # Record Count
- # utc_offset
- 1.2 origin_lat
- 1.2 origin_lon
- 1.2 dest_lat
- 1.2 dest_lon
- # distance
- # duration
- # star_rating
- 1.2 driver_home_lat

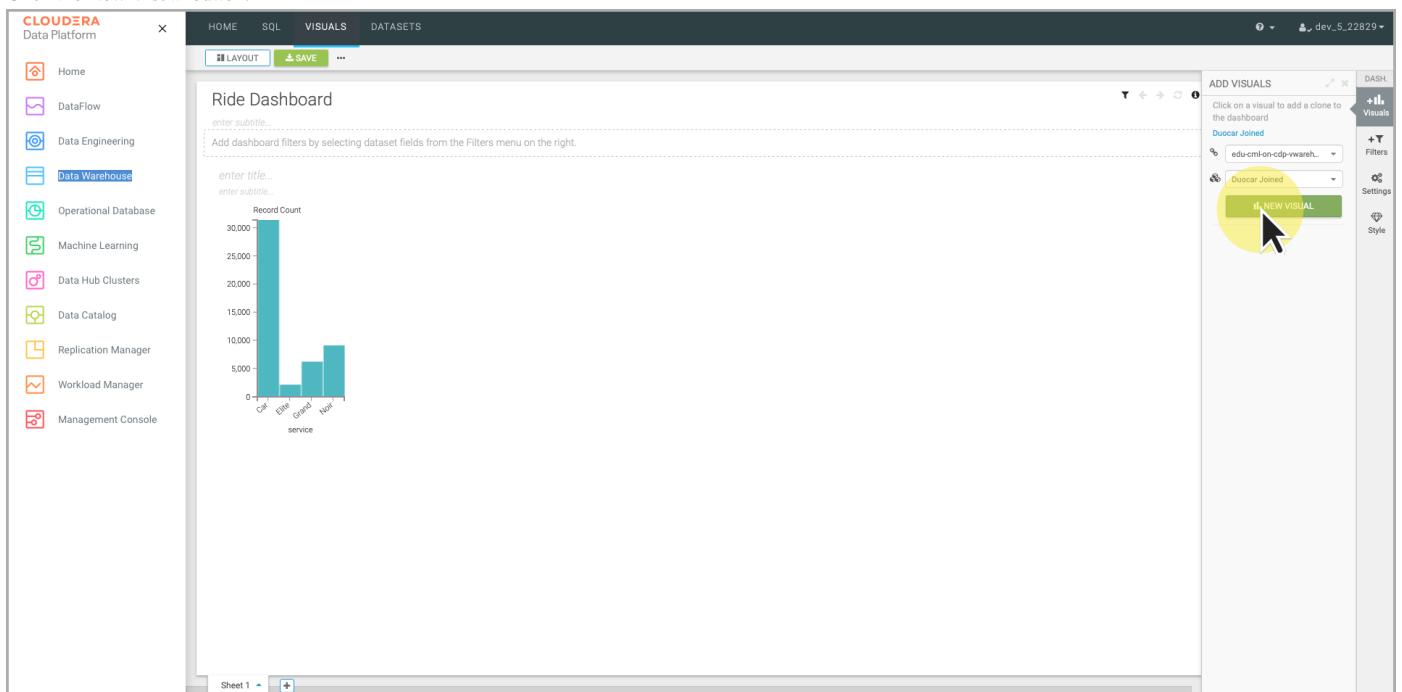
REFRESH VISUAL

Sheet 1

7. Click **Visuals** in the **DASH.** menu on the right.



8. Click the **New Visual** button.



9. Click the **Pie** icon. The pie chart properties are displayed.

10. Drag `star_rating` from the list of **Measures** and drop it in the **Dimension** field.

11. Drag Record Count from the list of Measures and drop it in the Measure field.

The screenshot shows the Cloudera Data Platform interface. On the left, the sidebar lists various services: Home, DataFlow, Data Engineering, Data Warehouse, Operational Database, Machine Learning, Data Hub Clusters, Data Catalog, Replication Manager, Workload Manager, and Management Console. The main area displays a bar chart titled "Record Count" with the following data:

Service	Record Count
Car	30,000
Elec	~1,000
Grand	~5,000
Nox	~8,000

To the right, the "Visuals" and "DATA" panels are visible. The "Measures" panel is open, showing a list of measures including "Record Count". A yellow circle highlights the "Record Count" item in the list, and a mouse cursor is positioned over it.

12. Enter Star Rating as the title for the pie chart visual.

This screenshot is similar to the previous one but shows a different state. The sidebar and main dashboard area remain the same. However, the "Measures" panel has been closed or moved, and the "Visuals" panel is now active. The "Measures" panel is still visible at the bottom right. A yellow circle highlights the "Record Count" item in the "Measures" list, and a mouse cursor is positioned over it.

13. Click **Visuals** in the **DASH.** menu on the right.

The screenshot shows the Cloudera Data Platform interface. On the left, there's a sidebar with icons for Home, DataFlow, Data Engineering, Data Warehouse, Operational Database, Machine Learning, Data Hub Clusters, Data Catalog, Replication Manager, Workload Manager, and Management Console. The main area displays the 'Ride Dashboard' with two visualizations: a bar chart titled 'Riders By Service' and a donut chart titled 'Star Rating'. The 'DASH.' menu on the right is highlighted with a yellow circle around the 'VISUALS' button. The menu also includes 'PIE', 'DATA', and other options.

14. Click the **New Visual** button.

This screenshot shows the same interface as the previous one, but the 'DASH.' menu is now open. The 'NEW VISUAL' button is highlighted with a yellow circle. The menu also includes 'PIE', 'VISUALS', and 'DATA' options.

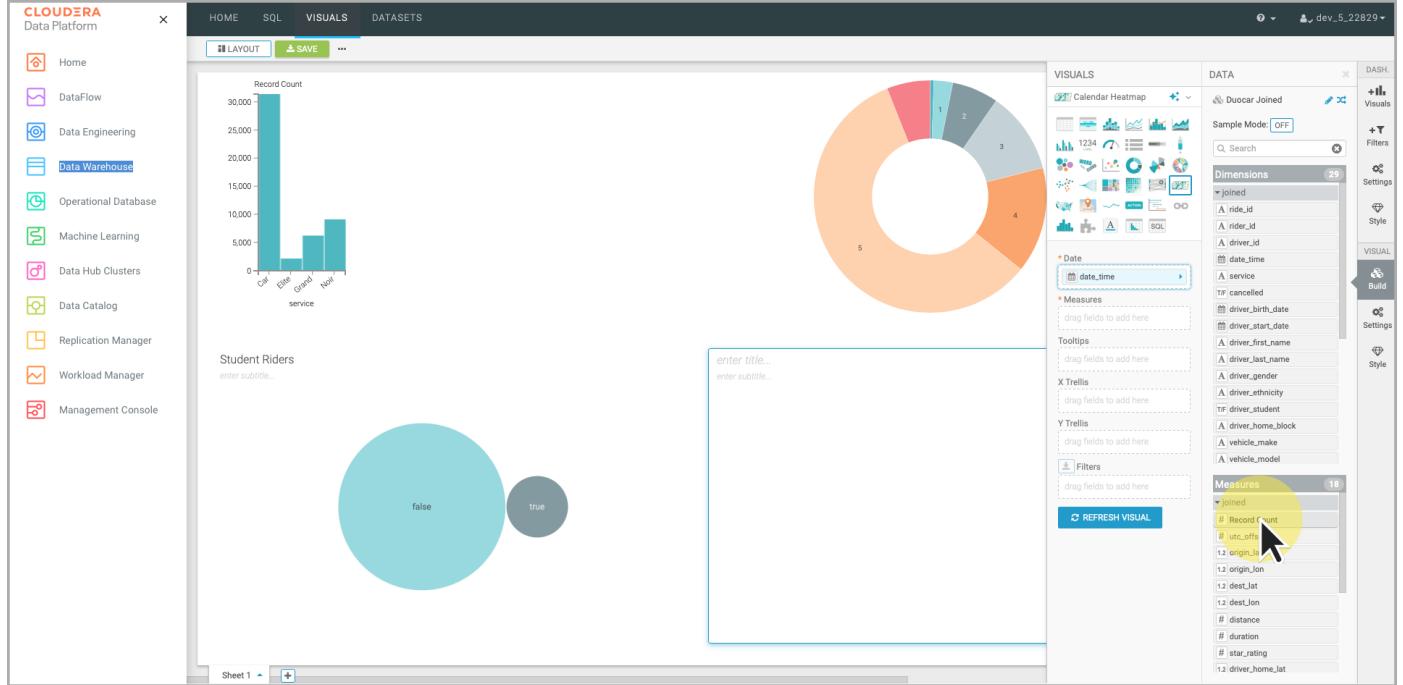
15. Click the **Packed Bubbles** icon. The packed bubbles properties are displayed.

The screenshot shows the Cloudera Data Platform interface with the 'Ride Dashboard' open. The left sidebar lists various data management tools. The dashboard itself contains three visualizations: a bar chart titled 'Riders By Service' showing record counts for 'Car', 'Elite', 'Grand', and 'Noir'; a donut chart titled 'Star Rating' showing proportions for categories 1 through 5; and a table titled 'enter title...' showing five rows of ride data with columns for ride_id, rider_id, driver_id, date_time, utc_offset, and service. The right side of the screen displays the 'Visuals' palette with the 'Packed Bubbles' icon highlighted by a yellow circle. The 'Dimensions' and 'Measures' sections are also visible.

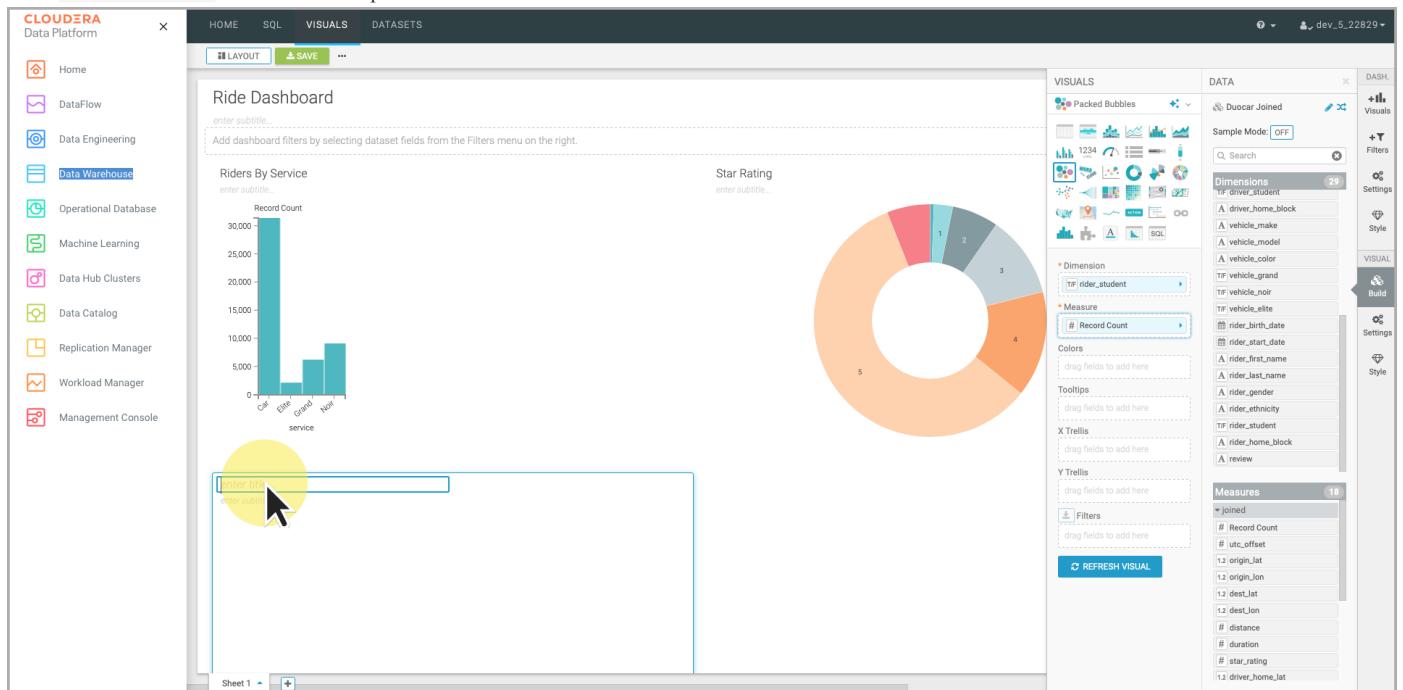
16. Drag `rider_student` from the list of **Dimensions** and drop it in the **Dimension** field.

This screenshot shows the same Ride Dashboard setup as the previous one, but with a change in the 'Dimensions' section of the Visuals palette. The dimension 'rider_student' has been selected and is highlighted with a yellow circle. It is now listed under the 'Dimension' section, indicating it has been dropped into the Dimension field of the packed bubbles visualization.

17. Drag Record Count from the list of Measures and drop it in the Measure field.



18. Enter Student Riders as the title for the packed bubbles visual.



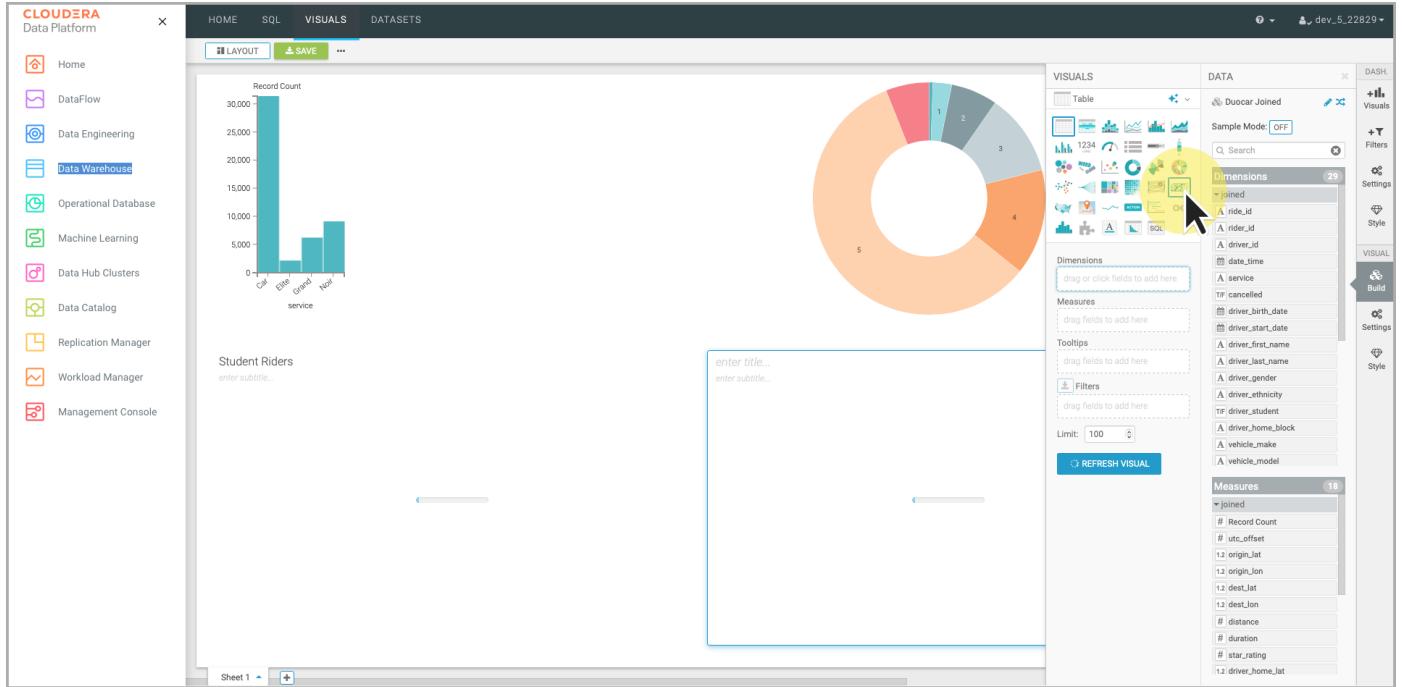
19. Click **Visuals** in the **DASH.** menu on the right.

The screenshot shows the Cloudera Data Platform interface with the 'Ride Dashboard' selected. The left sidebar lists various services: Home, DataFlow, Data Engineering, Data Warehouse, Operational Database, Machine Learning, Data Hub Clusters, Data Catalog, Replication Manager, Workload Manager, and Management Console. The top navigation bar has tabs for HOME, SQL, VISUALS (which is highlighted in blue), and DATASETS. Below the dashboard title, there are two visual components: a bar chart titled 'Riders By Service' showing record counts for different services, and a donut chart titled 'Star Rating' showing the distribution of star ratings from 0 to 5. The DASH. menu on the right is highlighted with a yellow circle around the 'Visuals' option.

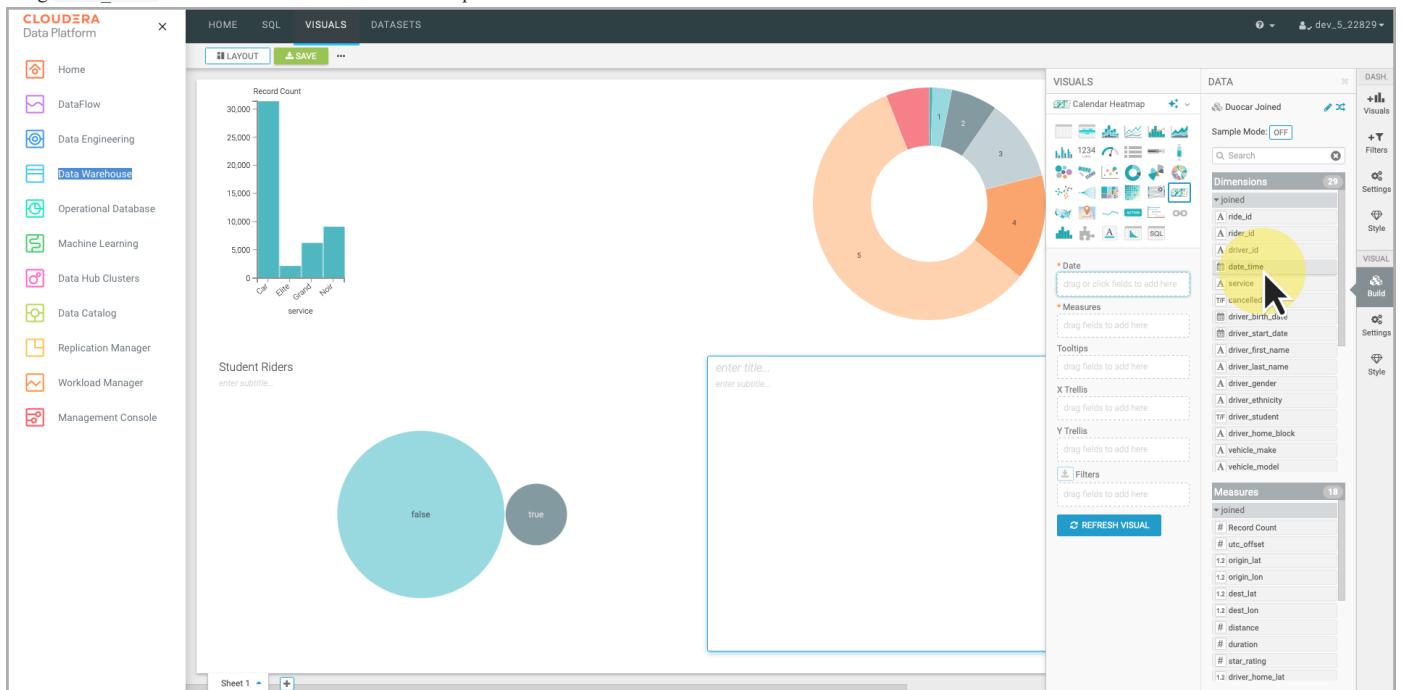
20. Click the **New Visual** button.

The screenshot shows the same Cloudera Data Platform interface as the previous one, but the ADD VISUALS panel on the right is now open. This panel contains a list of available datasets: 'edu-cml-on-cdp-warehouse' and 'Duocar_Joined'. A large green button labeled 'NEW VISUAL' is prominently displayed at the bottom of the panel. The cursor is hovering over this button, which is also highlighted with a yellow circle.

21. Click the **Calendar Heatmap** icon. The calendar heatmap properties are displayed.



22. Drag `date_time` from the list of **Dimensions** and drop it in the **Date** field.



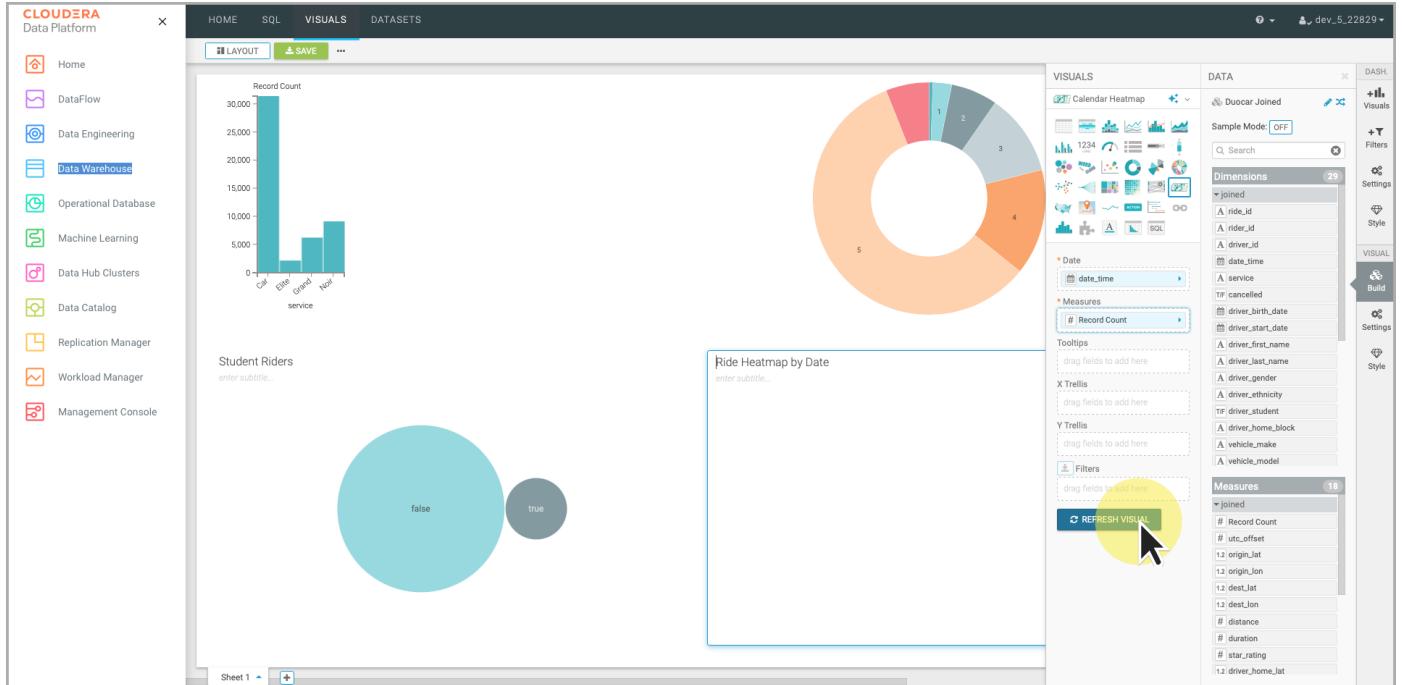
23. Drag Record Count from the list of Measures and drop it in the Measure field.

The screenshot shows the Cloudera Data Platform interface with a dashboard titled "dev_5_22829". The dashboard contains several visualizations: a bar chart titled "Record Count" showing values for categories like "Cabs", "Elite", "Grand", and "Navi" with counts ranging from 0 to 30,000; a donut chart divided into five segments labeled 1 through 5; a bubble chart titled "Student Riders" with two bubbles labeled "false" and "true"; and a large empty box labeled "enter title..." and "enter subtitle...". On the right side, the "Measures" section of the "Visuals" panel is highlighted with a yellow circle, showing the "Record Count" measure selected. Other measures listed include # utc_offset, # origin_lat, # origin_lon, # dest_lat, # dest_lon, # duration, # star_rating, and # driver_home_lat.

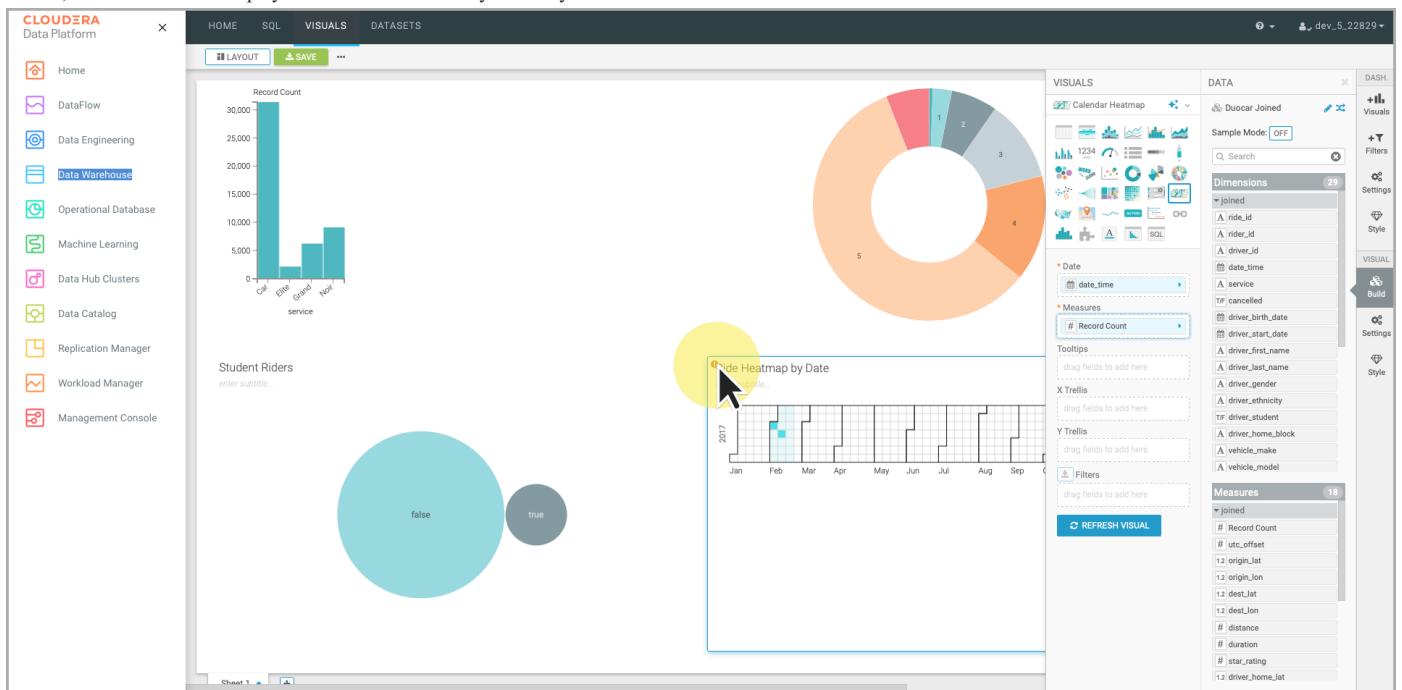
24. Enter Ride Heatmap by Date as the title for the calendar heatmap visual

This screenshot is identical to the previous one, showing the same dashboard and measures. However, the input field for the title of the calendar heatmap visual, located in the "enter title..." box, is now highlighted with a yellow circle and has the text "Ride Heatmap by Date" entered into it.

25. Click Refresh Visual.



26. Notice, there is an ⓘ icon displayed on the visual and only February has rides. Click the ⓘ icon.



27. The default limit for the number of rows fetched is 5,000. This is limiting the rides displayed to February. Click **Settings** in the right menu.

CLOUDERA Data Platform

HOME SQL VISUALS DATASETS

LAYOUT SAVE ...

Record Count

30,000
25,000
20,000
15,000
10,000
5,000
0

Cat Elite Grand Nov service

Student Riders enter subtitle...

false true

Error - Visual 98

Visual 98 - Rows fetched for this visual have been limited to 5000 rows automatically

To stop this warning from showing, set visual limits explicitly in the setting labeled "Maximum number of rows to fetch" (global limit might still be applied.)

CLOSE COPY ERROR TEXT CLEAR ERRORS

2017 Jan Feb Mar Apr May Jun Jul Aug Sep

VISUALS

DATA

DASH.

Dimensions 29

- joined
 - A ride_id
 - A rider_id
 - A driver_id
 - date_time
 - date
 - time
 - tips
 - ells
 - drag fields to add here
 - Filters

Measures 18

- joined
 - # Record Count
 - # utc_offset
 - 1:2 origin_lat
 - 1:2 origin_lon
 - 1:2 dest_lat
 - 1:2 dest_lon
 - # distance
 - # duration
 - # star_rating
 - 1:2 driver_home_list

28. Expand the **Data** category in the **Visual Settings**.

CLOUDERA Data Platform

HOME SQL VISUALS DATASETS

LAYOUT SAVE ...

Record Count

30,000
25,000
20,000
15,000
10,000
5,000
0

Cat Elite Grand Nov service

Student Riders enter subtitle...

false true

Ride Heatmap by Date

enter subtitle...

2017 Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Mon Wed Fri Sun

VISUAL SETTINGS

Apply future changes to all visuals

General

Downloads

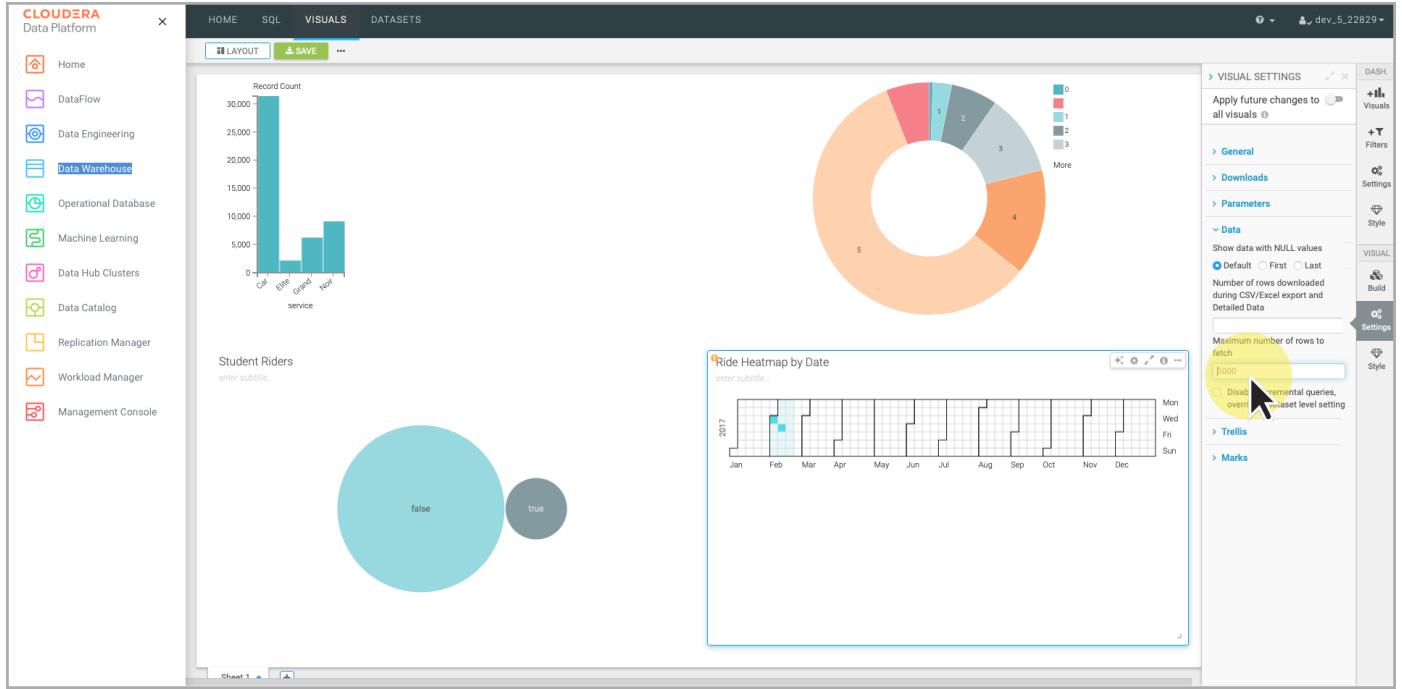
Parameters

Data

Marks

DASH.

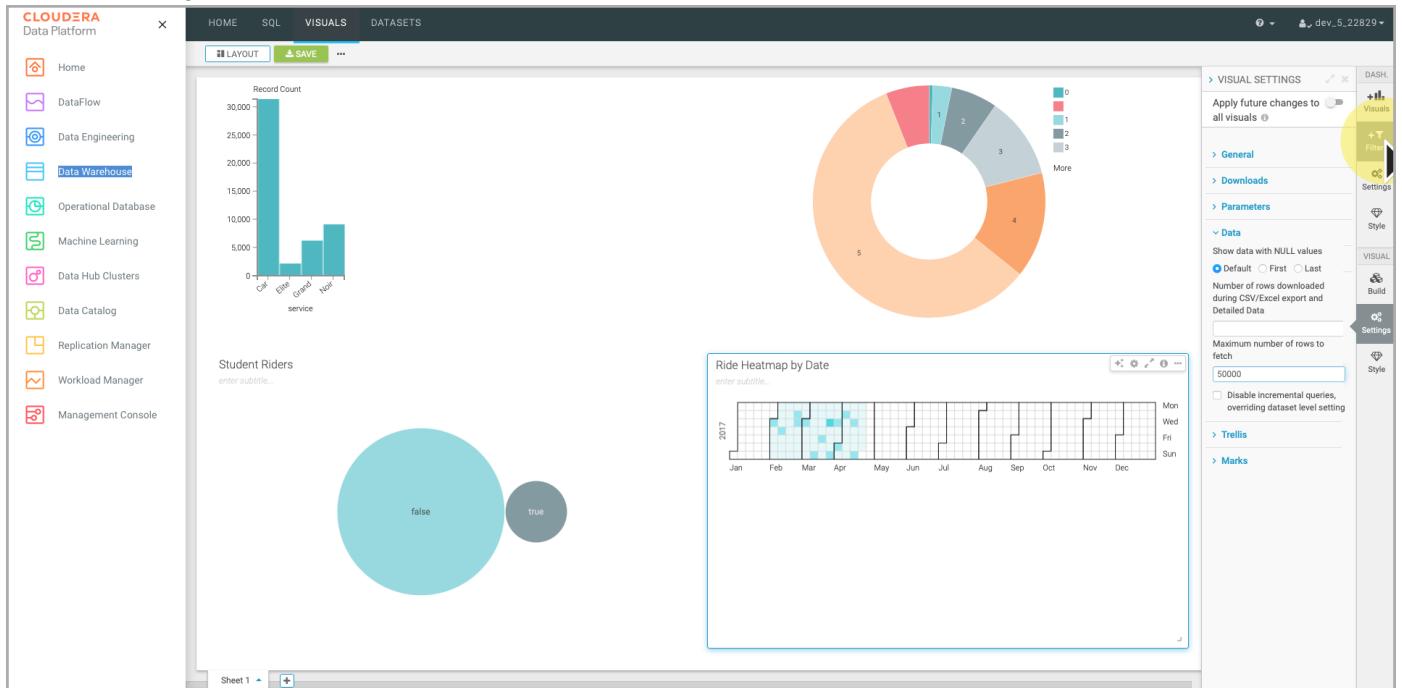
29. Enter 50000 for the **Maximum number of rows to fetch**. The heatmap will show data for February through April.



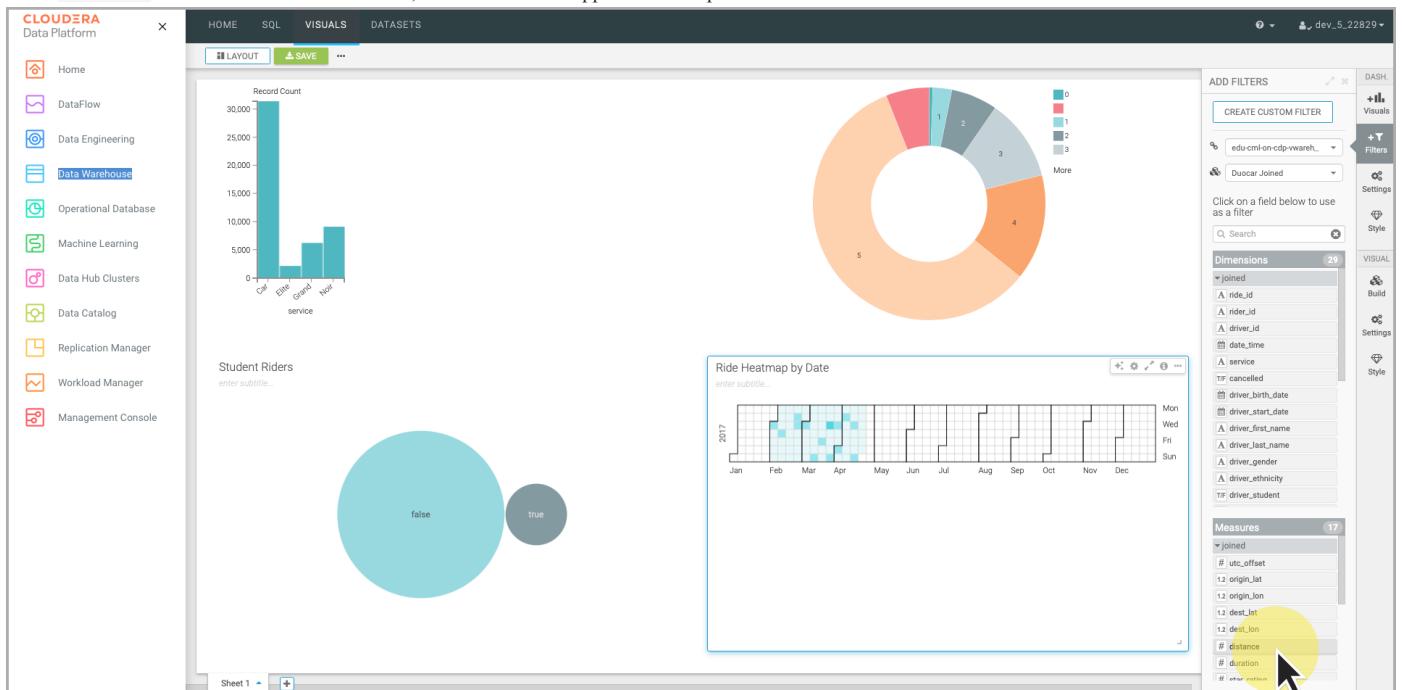
Create Filters

Displaying the data is interesting but it is static. In order to explore the data and find the answers to questions like, "What service do students prefer?", filters can be added to the dashboard to make it interactive.

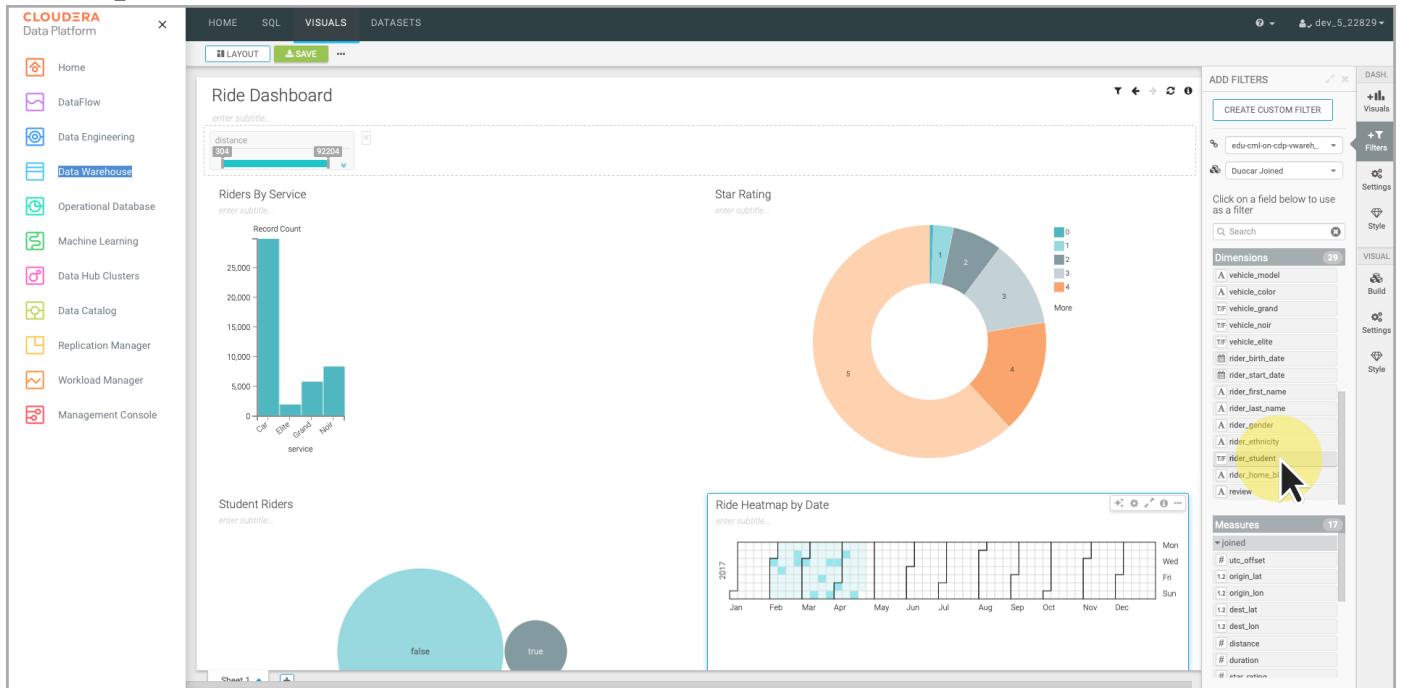
1. Click **Filters** in the right menu.



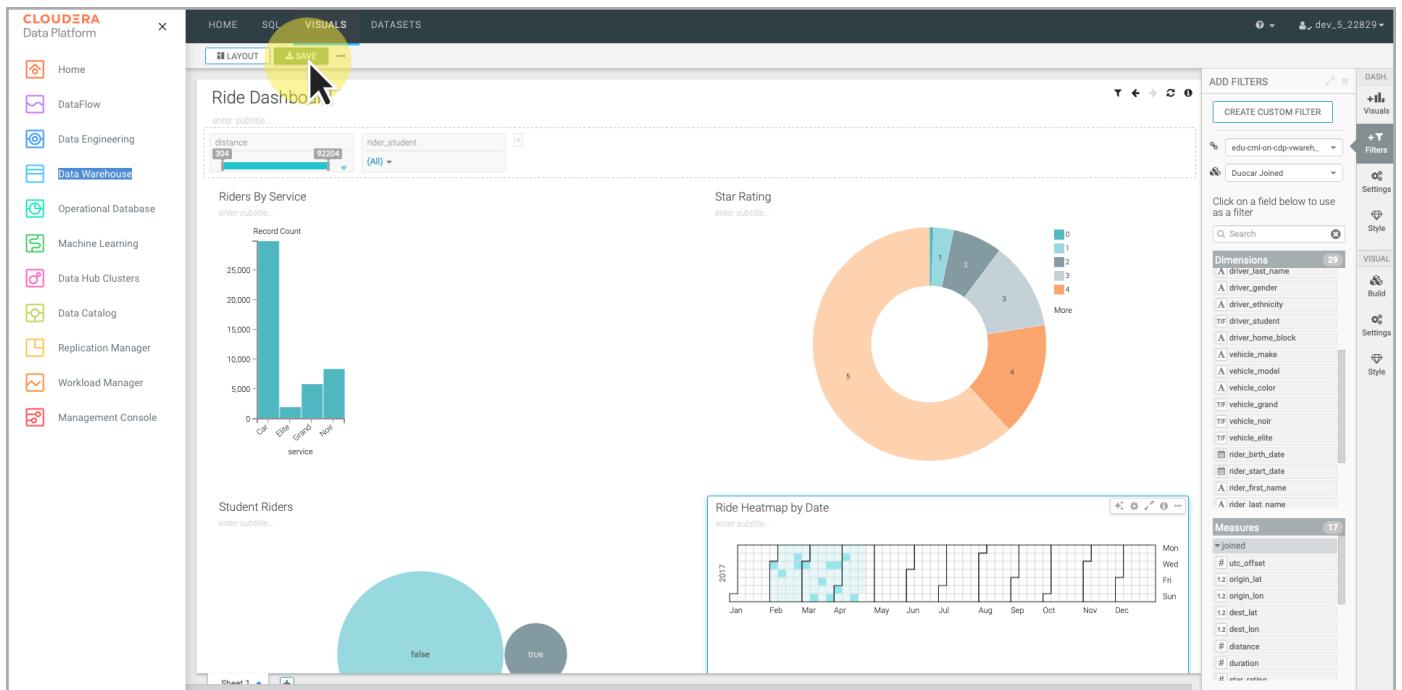
2. Click **distance** in the list of **Measures**. Notice, a distance filter is applied to the top of the dashboard.



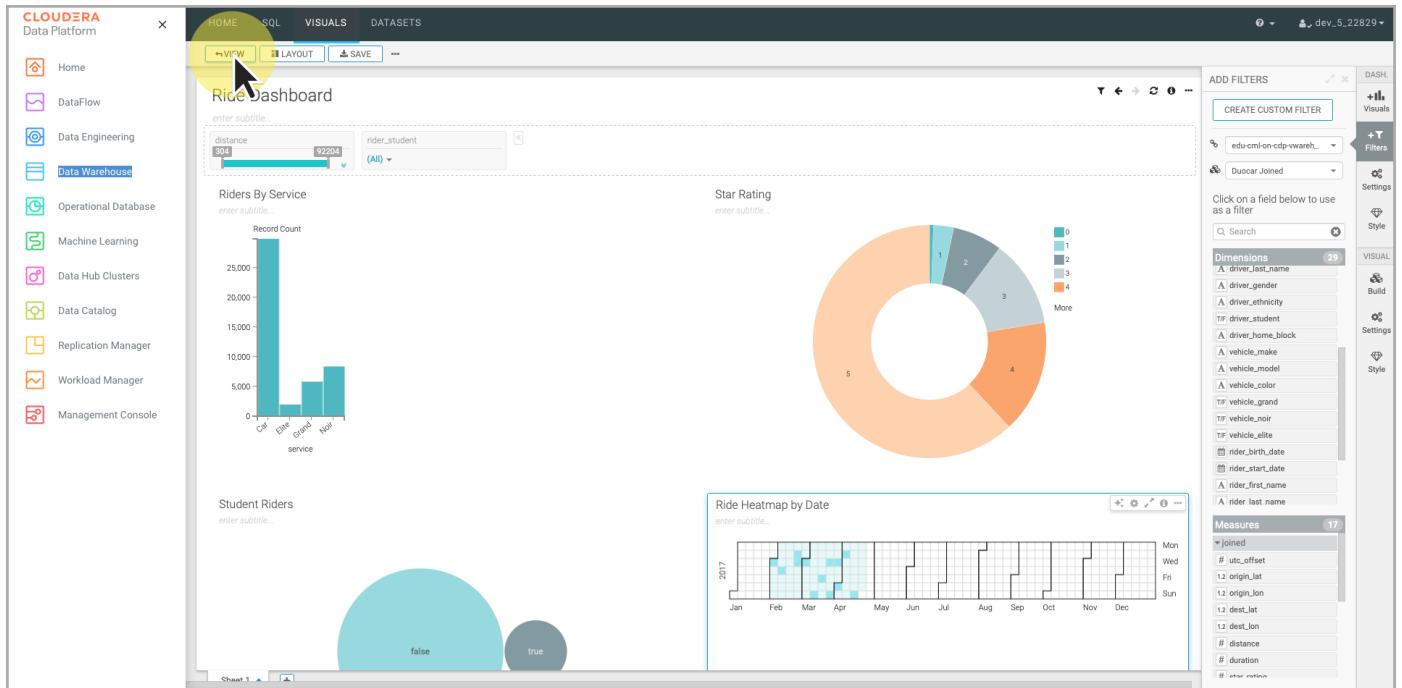
3. Click `rider_student` in the list of Dimensions.



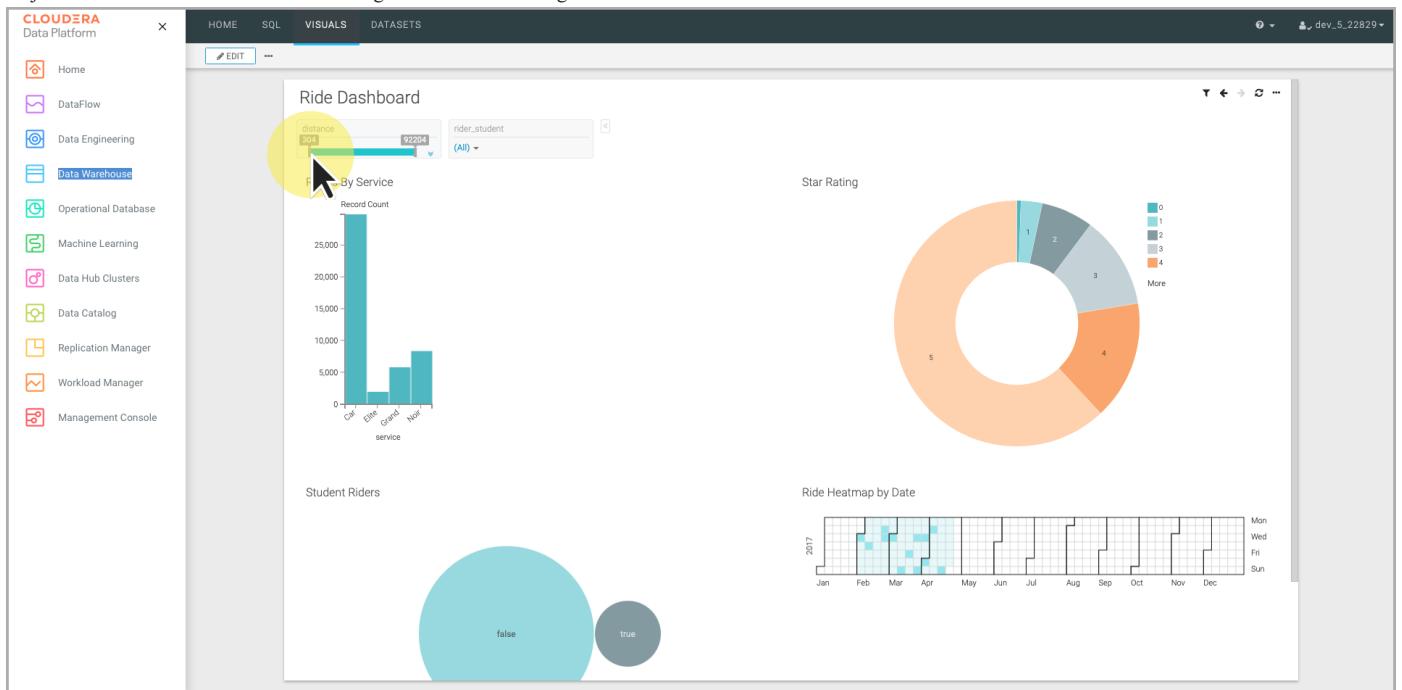
4. Click Save to save the dashboard.



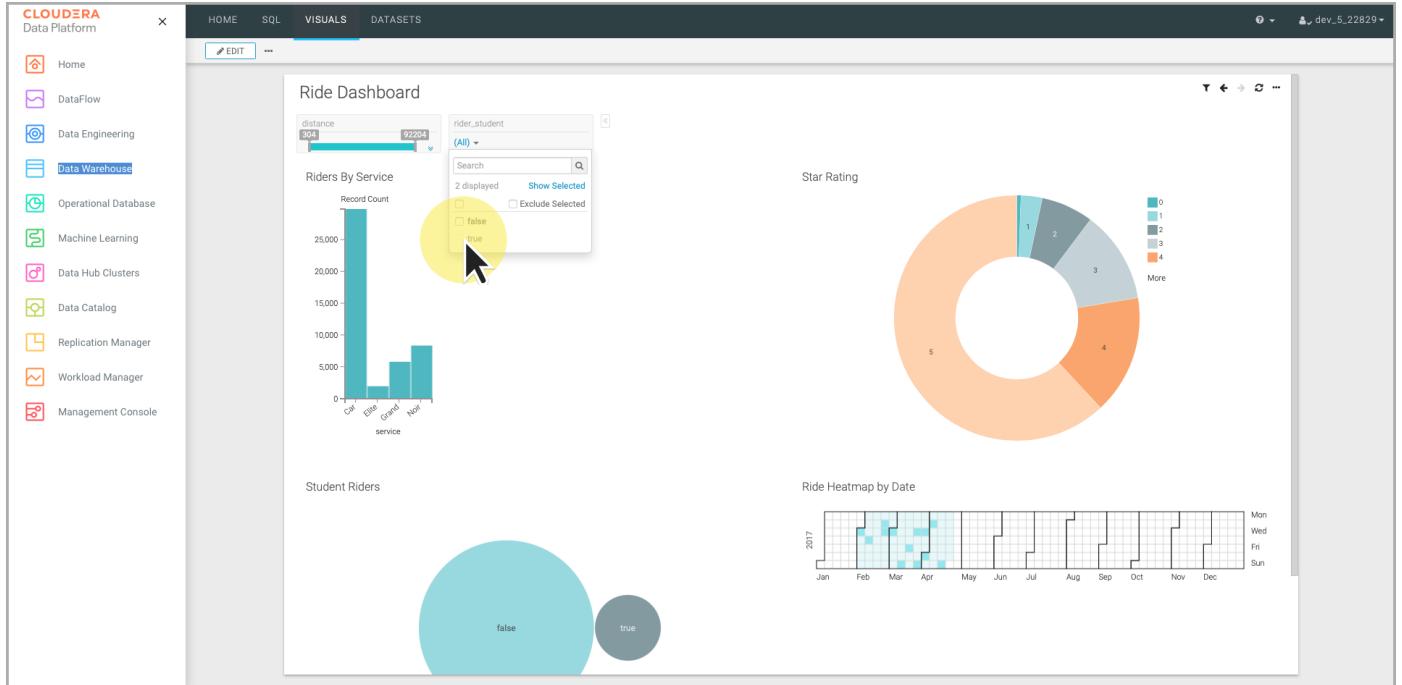
5. Click View to view the dashboard.



6. Adjust distance sliders. Do students take longer or shorter rides in general?

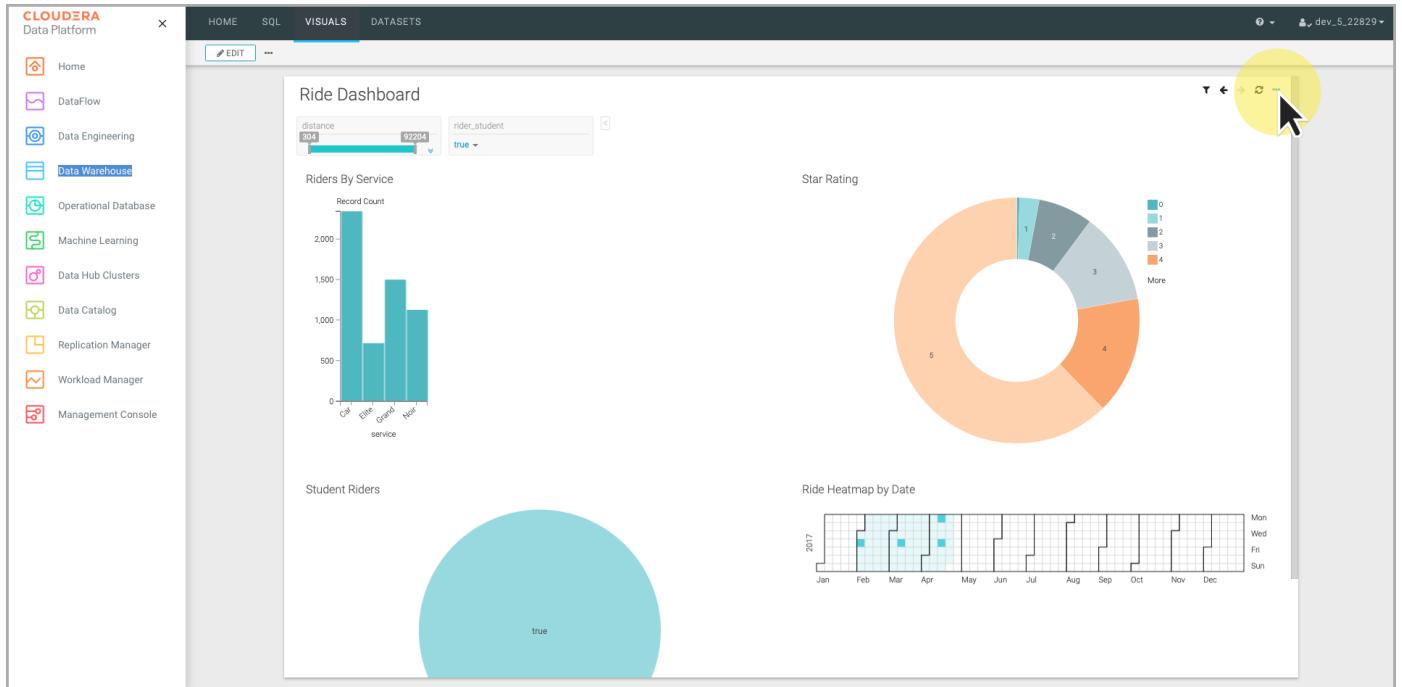


7. Reset the distance sliders. Use the filter to select only student riders. What day of the week has the most student riders in general?

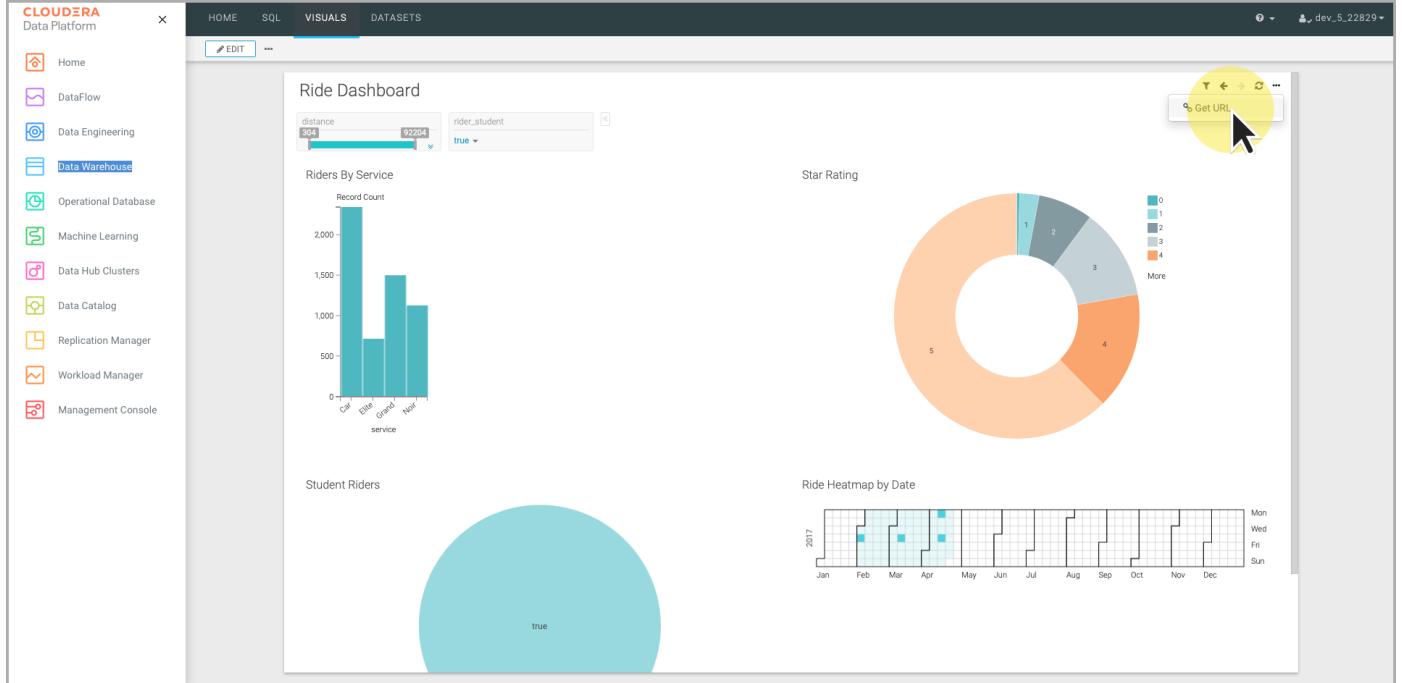


Share the Dashboard

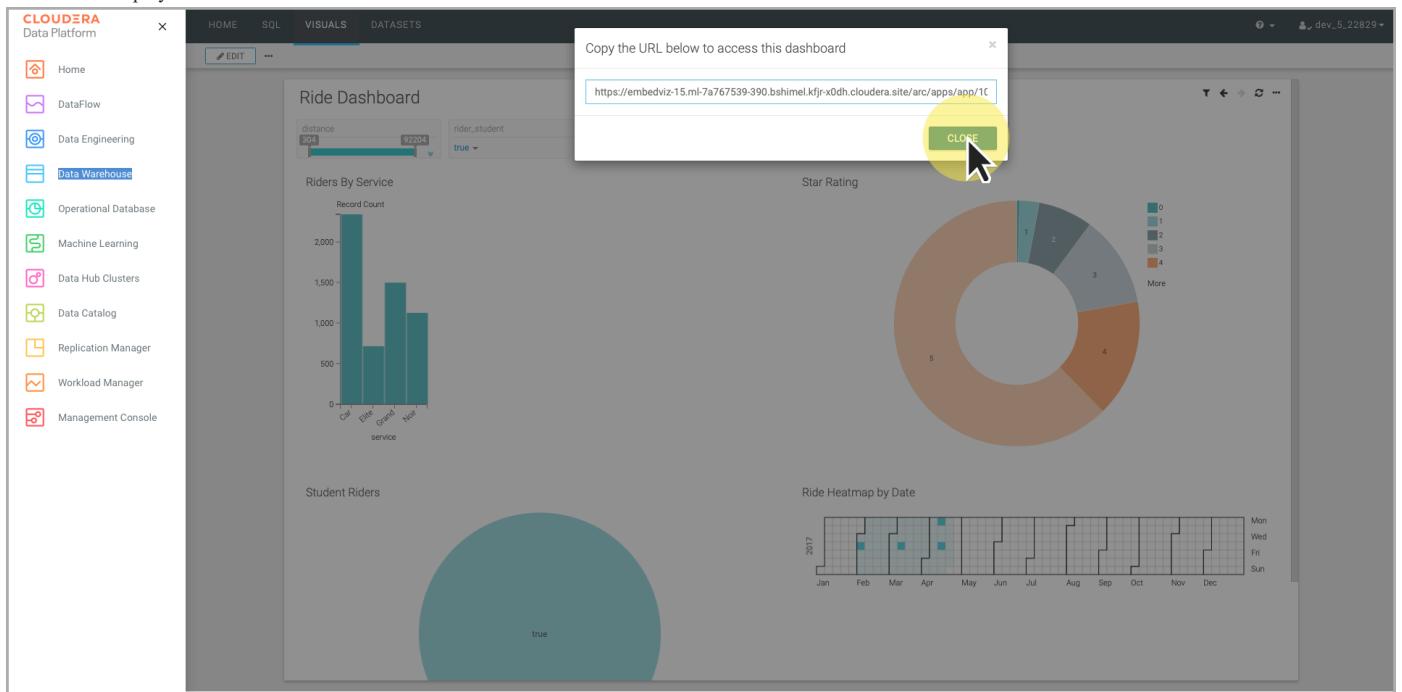
1. Click the **...** menu in the upper right corner of the dashboard.



2. Click Get URL.



3. The URL is displayed. The URL can be used to share the dashboard with other users.



End of Exercise

Experiment Tracking

Tracking experiments is critical for knowing what parameters were used to generate a model and how the model is performing. CML projects provide experiment tracking based on **MLflow** to make this task easier.

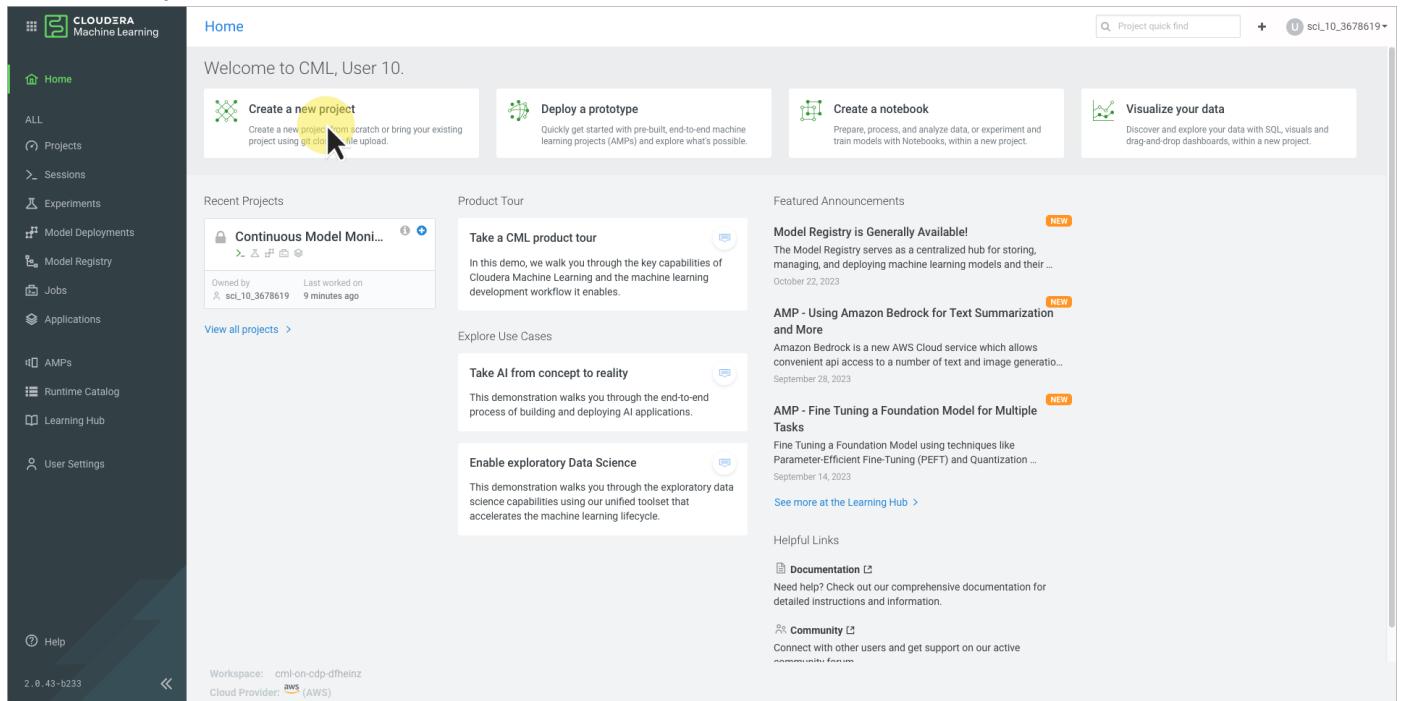
In this exercise, you will:

- use a Git repository to create your project,
- use the CML experiments feature to track a experiment, and
- commit the changes back to git.

Create a New Project from Github

1. Navigate to your workspace.

2. Click the **New Project** button.



The screenshot shows the Cloudera Machine Learning (CML) web interface. On the left, there is a sidebar with various navigation links: Home, ALL, Projects, Sessions, Experiments, Model Deployments, Model Registry, Jobs, Applications, AMPS, Runtime Catalog, Learning Hub, and User Settings. At the bottom of the sidebar, it says "2.0.43-b233". The main content area has a header "Welcome to CML, User 10." and several sections: "Recent Projects" (listing "Continuous Model Moni..." owned by "A. sc1_10_3678619" last worked on 9 minutes ago), "Product Tour" (with cards for "Take a CML product tour", "Enable exploratory Data Science", and "Explore Use Cases"), and "Featured Announcements" (listing "Model Registry is Generally Available!", "AMP - Using Amazon Bedrock for Text Summarization and More", and "AMP - Fine Tuning a Foundation Model for Multiple Tasks"). At the top right, there is a search bar "Project quick find" and a user icon "sc1_10_3678619". The "Create a new project" button in the top center is highlighted with a yellow arrow.

3. Enter Experiments - Student # for the project name.

New Project

* Project Name
Experiments - Student 10

Project Description

Project Visibility
 Private - Only added collaborators can view the project
 Public - All authenticated users can view this project.

Initial Setup
Blank Template AMPS Local Files Git

Templates include example code to help you get started.
Python

Runtimes
Projects are configured with the latest Python and R ML Runtimes. You can change this configuration under the Advanced Options.

Cancel Create Project

4. Click Git under Initial Setup.

New Project

* Project Name
Experiments - Student 10

Project Description

Project Visibility
 Private - Only added collaborators can view the project
 Public - All authenticated users can view this project.

Initial Setup
Blank Template AMPS Local Files Git

Templates include example code to help you get started.
Python

Runtimes
Projects are configured with the latest Python and R ML Runtimes. You can change this configuration under the Advanced Options.

Cancel Create Project

5. Enter the following for the **Git URL**:

```
https://github.com/bshimel-cloudera/edu-cml-on-cdp-experiments
```

New Project

Project Name: Experiments - Student 10

Project Description:

Project Visibility: Private - Only added collaborators can view the project

Initial Setup: Git

Provide the Git URL of the project to clone. Select the option that applies to your URL access.

HTTPS (selected) SSH

Git URL of Project: https://github.com/bshimel-cloudera/edu-cml-on-cdp-experiments

You are able to provide username/password.
e.g. https://username:password@mygitghost.com/my/repository

Create Project

6. Select Advanced Options

New Project

Blank Template AMPs Local Files Git

Provide the Git URL of the project to clone. Select the option that applies to your URL access.

HTTPS (selected) SSH

Git URL of Project: https://github.com/bshimel-cloudera/edu-cml-on-cdp-experiments

You are able to provide username/password.
e.g. https://username:password@mygitghost.com/my/repository

Runtimes

Projects are configured with the latest Python and R ML Runtimes. You can change this configuration under the Advanced Options.

Editor	Kernel	Edition	Version	Remove
JupyterLab	Python 3.10	Nvidia GPU	2024.02	Remove
JupyterLab	Python 3.10	Standard	2024.02	Remove
PBJ Workbench	Python 3.10	Nvidia GPU	2024.02	Remove
PBJ Workbench	Python 3.10	Standard	2024.02	Remove
PBJ Workbench	R 4.3	Standard	2024.02	Remove

Advanced Options

Create Project

7. Ensure Editor is set to **Workbench**, Kernel is set to **Python 3.9**, Edition is set to **Standard**, and select **2023.12** for Version.

The screenshot shows the 'New Project' dialog in the Cloudera Machine Learning interface. The 'Version' dropdown menu is open, displaying options: '2023.12' (highlighted with a yellow circle), '2023.12', and '2024.02'. The '2023.12' option is selected. Other fields in the dialog include 'Editor: Workbench', 'Kernel: Python 3.9', and 'Edition: Standard'.

8. Click **Add Runtime**.

The screenshot shows the 'New Project' dialog in the Cloudera Machine Learning interface. The 'Add Runtime' button is highlighted with a yellow circle. Other fields in the dialog include 'Editor: Workbench', 'Kernel: Python 3.9', and 'Edition: Standard'.

9. Click Create Project.

The screenshot shows the Cloudera Machine Learning interface with the 'New Project' screen. The left sidebar has a dark theme with various navigation options like Home, Projects (which is selected), Sessions, Experiments, Model Deployments, Model Registry, Jobs, Applications, AMPs, Runtime Catalog, Learning Hub, User Settings, Help, and a version number (2.0.43-2023). The main area has a light gray background. At the top, there's a URL input field with 'https://github.com/osnime/ci-on-cap-experiments' and a note about providing credentials. Below that is a 'Runtimes' section with a table:

Editor	Kernel	Edition	Version	Remove
JupyterLab	Python 3.10	Nvidia GPU	2024.02	<button>Remove</button>
JupyterLab	Python 3.10	Standard	2024.02	<button>Remove</button>
PBJ Workbench	Python 3.10	Nvidia GPU	2024.02	<button>Remove</button>
PBJ Workbench	Python 3.10	Standard	2024.02	<button>Remove</button>
PBJ Workbench	R 4.3	Standard	2024.02	<button>Remove</button>
Workbench	Python 3.9	Standard	2023.12	<button>Remove</button>

Below the table is a 'Advanced Options' toggle switch, followed by dropdown menus for Editor (set to Workbench), Kernel (set to Python 3.9), Edition (set to Standard), and Version (set to 2023.12). A blue 'Add Runtime' button is below these. At the bottom right are 'Cancel' and 'Create Project' buttons, with the 'Create Project' button being highlighted with a yellow circle and a mouse cursor pointing at it.

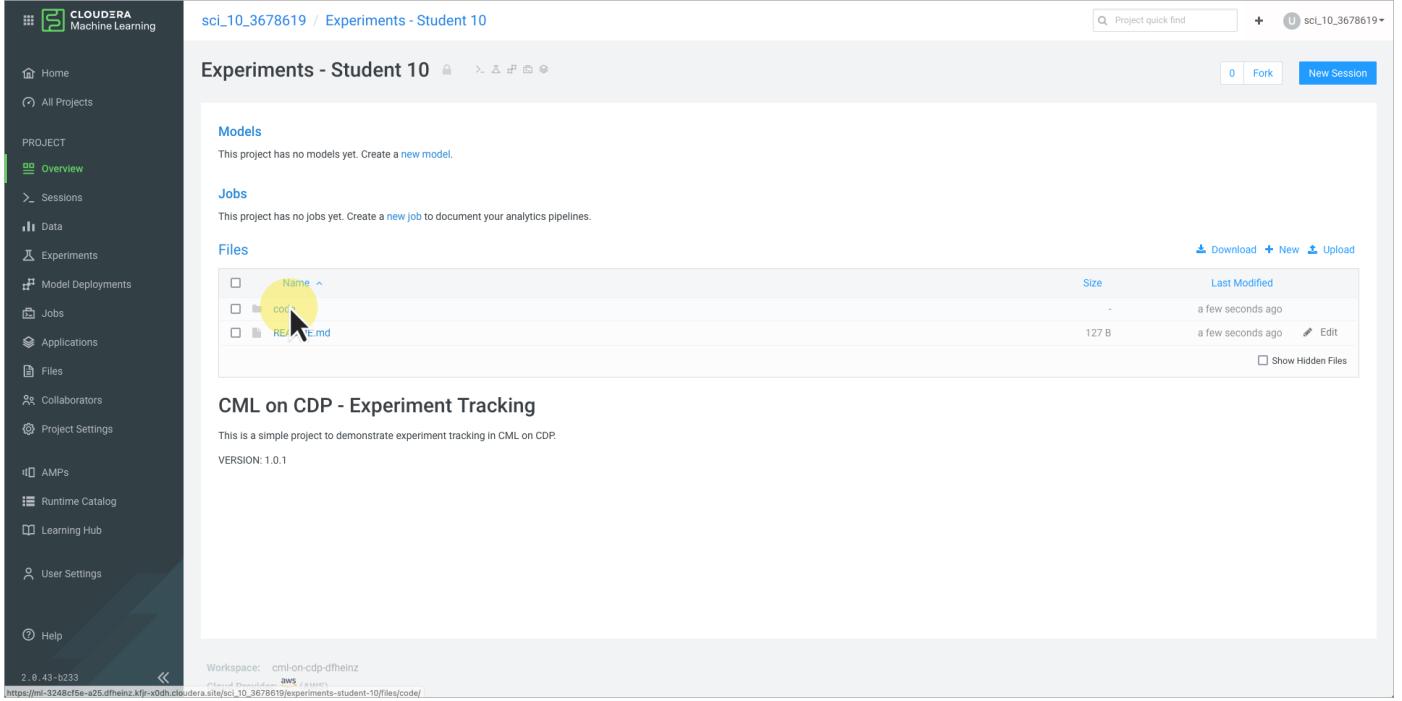
CML will now checkout the content from Github and add it to the project.

View Code and Run Job

The next step is to look at the code and then run it. Since this exercise is focused on the use of Experiments feature, the code is very simple. There is a single file, `add.py`. It takes a series of numbers and calculates the sum. In this example, the numbers passed as arguments to the `add.py` program are acting as our

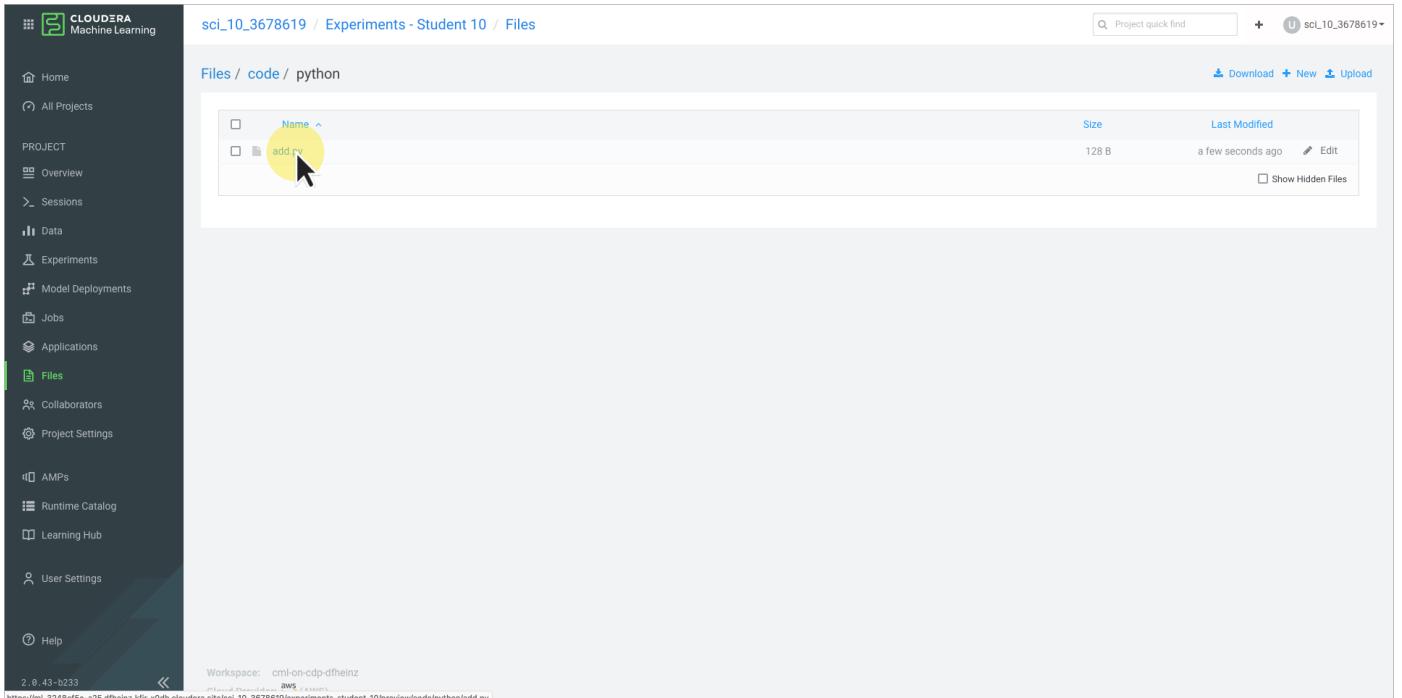
parameters, the `sum()` function is acting as our machine learning model, and the `total` is serving as the prediction or model output. Obviously, the "model" in this case will be 100% accurate, but it will allow the exploration of the Experiment tracking features of CML.

1. View the file by selecting the `code` folder from the list of files.



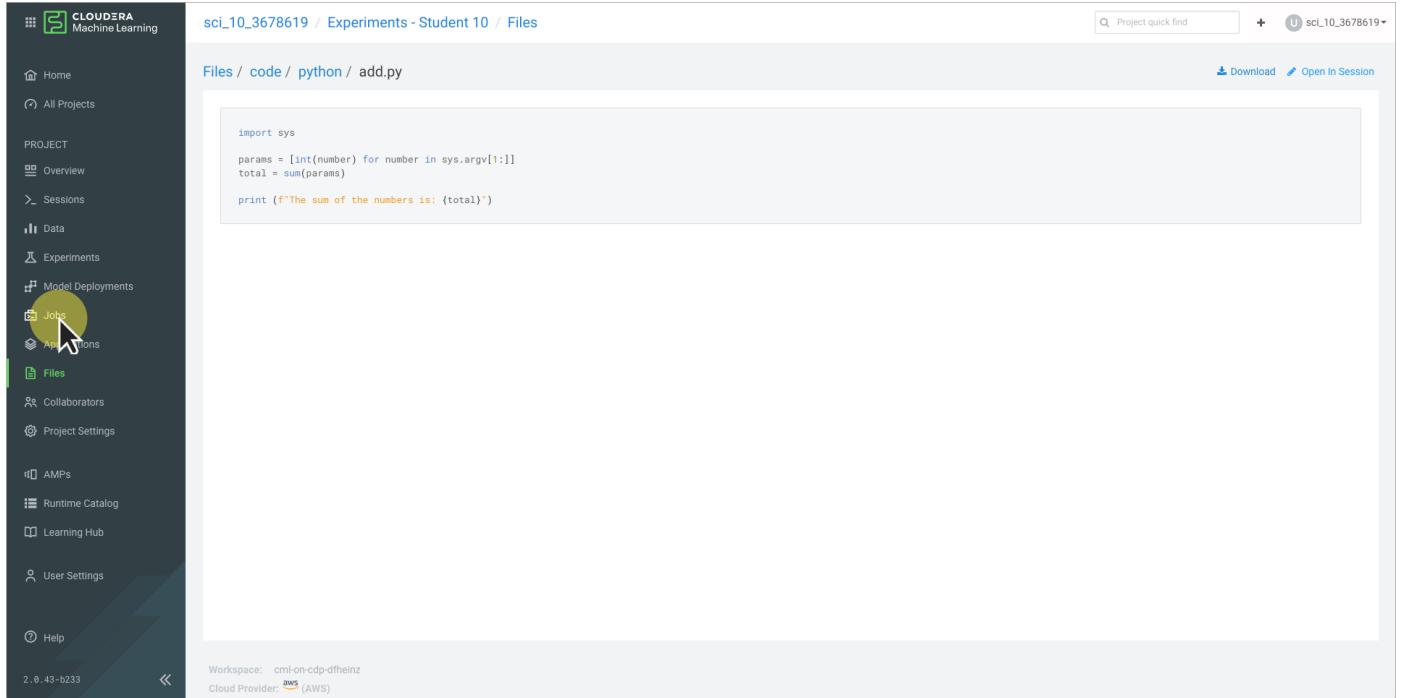
The screenshot shows the Cloudera Machine Learning interface. On the left, a sidebar lists various project components like Home, All Projects, Data, Experiments, and Files. The 'Files' option is currently selected. In the main workspace, a project titled 'Experiments - Student 10' is displayed. Under the 'Files' section, there is a list of files. A yellow circle highlights the 'code' folder, which contains two files: 'conf' and 'README.md'. The 'README.md' file is shown in a preview pane below. The URL at the bottom of the screen is https://ml-3248cf5e-a25.dfeheinz.kfp-x0dh.cloudera.site/sci_10_3678619/experiments-student-10/files/code/.

2. Click on the `python` folder, and then click on the `add.py` file.



This screenshot continues from the previous one, showing the same interface but with a different focus. The 'Files' section now shows a nested folder structure under 'code': 'code / python'. A yellow circle highlights the 'python' folder, which contains a single file named 'add.py'. The URL at the bottom of the screen is https://ml-3248cf5e-a25.dfeheinz.kfp-x0dh.cloudera.site/sci_10_3678619/experiments-student-10/preview/code/python/add.py.

3. You can see that it is a very simple, three-line program. Click **Jobs** in the left-side menu.

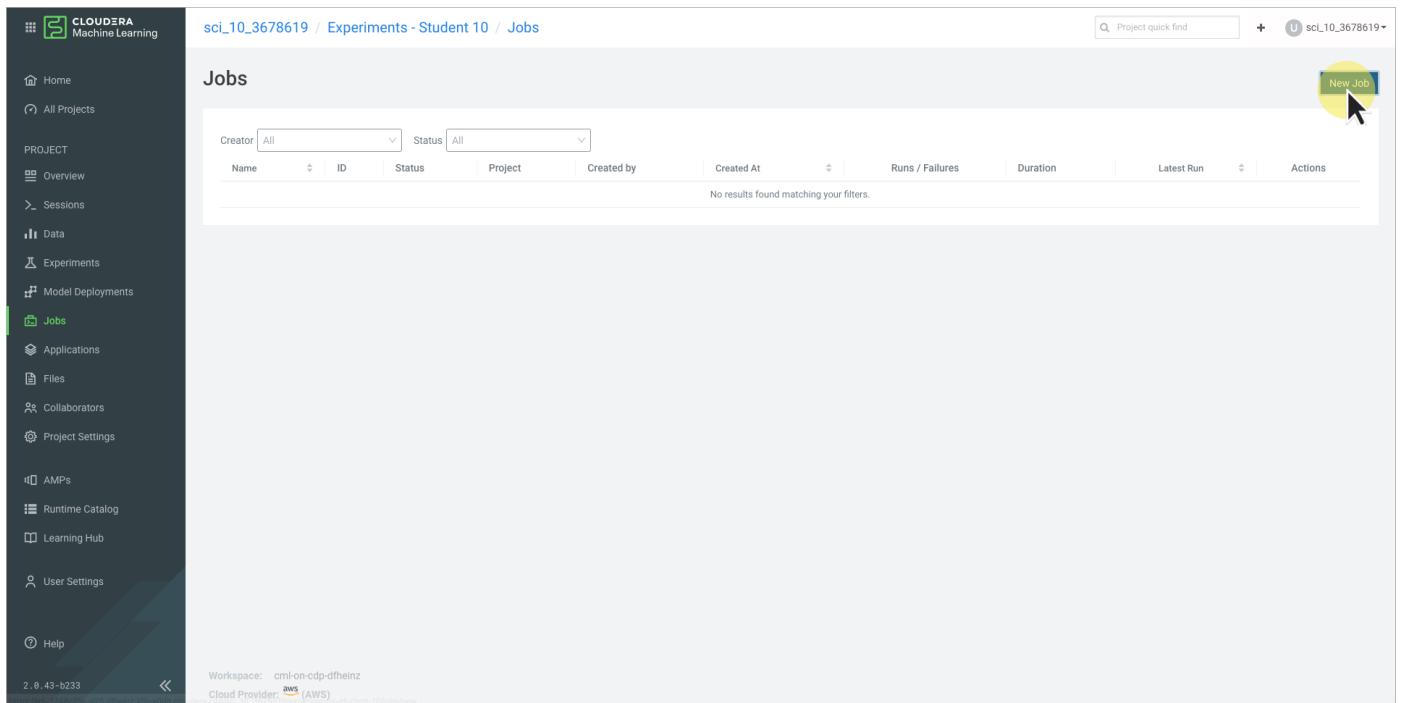


The screenshot shows the Cloudera Machine Learning interface. On the left, the navigation bar has a 'Jobs' icon highlighted with a yellow circle. The main content area shows a file named 'add.py' with the following code:

```
import sys
params = [int(number) for number in sys.argv[1:]]
total = sum(params)
print(f"The sum of the numbers is: {total}")
```

At the top right, there are 'Download' and 'Open In Session' buttons. The bottom status bar indicates the workspace is 'cmi-on-cdp-dfheinz' and the cloud provider is 'aws (AWS)'.

4. Click **New Job**.



The screenshot shows the 'Jobs' page in the Cloudera Machine Learning interface. The 'Jobs' icon in the navigation bar is highlighted with a yellow circle. The main content area displays a table header for 'Jobs' with columns: Creator, Status, Name, ID, Status, Project, Created by, Created At, Runs / Failures, Duration, Latest Run, and Actions. A message at the bottom states 'No results found matching your filters.' On the far right, there is a large green button labeled 'New Job' with a white arrow pointing to it, also highlighted with a yellow circle.

The bottom status bar indicates the workspace is 'cmi-on-cdp-dfheinz' and the cloud provider is 'aws (AWS)'.

5. Name the new job Add It Up . Click the **Browse** folder.

Create a Job

General

Name: Add It Up

Run Job as: me (Service Account: No active service account is available)

Script: Script

Arguments: Arguments

Select... to run

Runtime

Editor: Workbench Kernel: Python 3.9

Edition: Standard Version: 2023.12

Configure additional runtime options in Project Settings.

Enable Spark: Spark 3.2.3 - CDE 1.19.2 - HOTFIX-2

Runtime Image
- docker.repository.cloudera.com/cloudera/cdsw/ml-runtime-workbench-python3.9-standard: 2023.12.1-b8

Workspace: cmi-on-cdp-dfheinz Cloud Provider: AWS (AWS)

6. Select the `add.py` file.

Select Script

python

add.py

Select Cancel

7. The program accepts a list of numbers, separated by spaces. Enter something like 20 30 for the Arguments.

Create a Job

General

Name: Add It Up

Run Job as: me (Service Account: No active service account is available)

Script: code/python/add.py

Arguments: [Arguments] (highlighted with a yellow circle)

Runtime

Editor: Workbench Kernel: Python 3.9

Edition: Standard Version: 2023.12

Configure additional runtime options in Project Settings.

Enable Spark: Spark 3.2.3 - CDE 1.19.2 - HOTFIX-2

Runtime Image
- docker.repository.cloudera.com/cloudera/cdsw/ml-runtime-workbench-python3.9-standard: 2023.12.1-b8

Workspace: cmi-on-cdp-dfheinz Cloud Provider: AWS (AWS)

8. Click Create Job.

scI_10_3678619 / Experiments - Student 10 / Jobs / New Job

Enable Spark: Spark 3.2.3 - CDE 1.19.2 - HOTFIX-2

Runtime Image
- docker.repository.cloudera.com/cloudera/cdsw/ml-runtime-workbench-python3.9-standard: 2023.12.1-b8

Schedule: Manual

Resource Profile: 2 vCPU / 4 GiB Memory

GPUs: 0 GPUs

Timeout In Minutes (optional): 30 Kill on Timeout

Environment Variables:

Name	Value	Actions
		Add

Environment variables will override the project environment.

Job Notifications

Email Notifications Unavailable
Outbound email configuration is missing or incorrect, preventing email notifications from being set up. Please contact your administrator to resolve this issue.

Create Job (highlighted with a yellow circle)

Workspace: cmi-on-cdp-dfheinz Cloud Provider: AWS (AWS)

9. The new job is created, but not executed. Click **Run as [user icon]** to execute the job.

scI_10_3678619 / Experiments - Student 10 / Jobs

Jobs

Job Dependencies for Add It Up

Name	ID	Status	Project	Created by	Created At	Runs / Failures	Duration	Latest Run	Actions
Add It Up	2	Not Yet Run	Experiments - Student 10	User 10	03/17/2024 5:50 PM	0 / 0	Not yet run	not run	Run as

Displaying 1 - 1 of 1 < 1 > 25 / page

10. Once the **Status** has changed to **Success**, click on the job name.

scI_10_3678619 / Experiments - Student 10 / Jobs

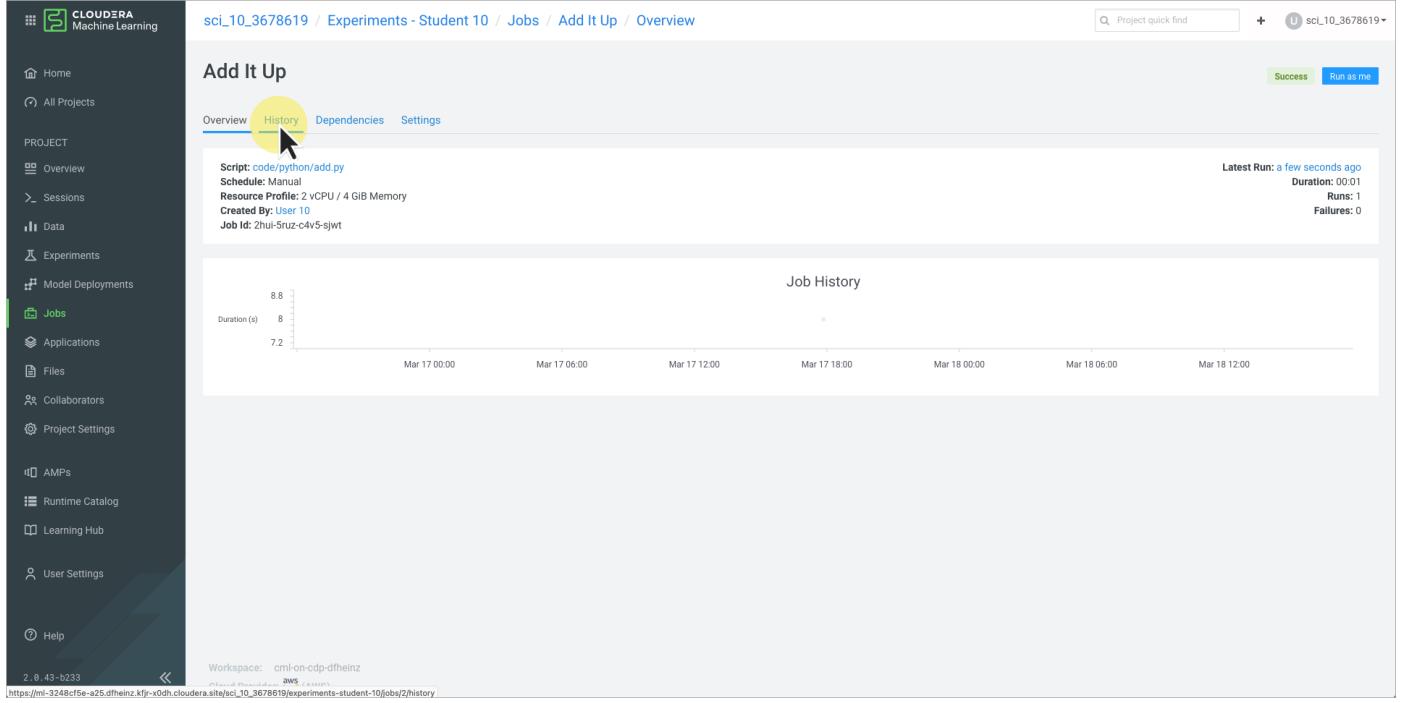
Jobs

Job Dependencies for Add It Up

Name	ID	Status	Project	Created by	Created At	Runs / Failures	Duration	Latest Run	Actions
Add It Up	2	Success	Experiments - Student 10	User 10	03/17/2024 5:50 PM	1 / 0	1s	a few seconds ago	Run as

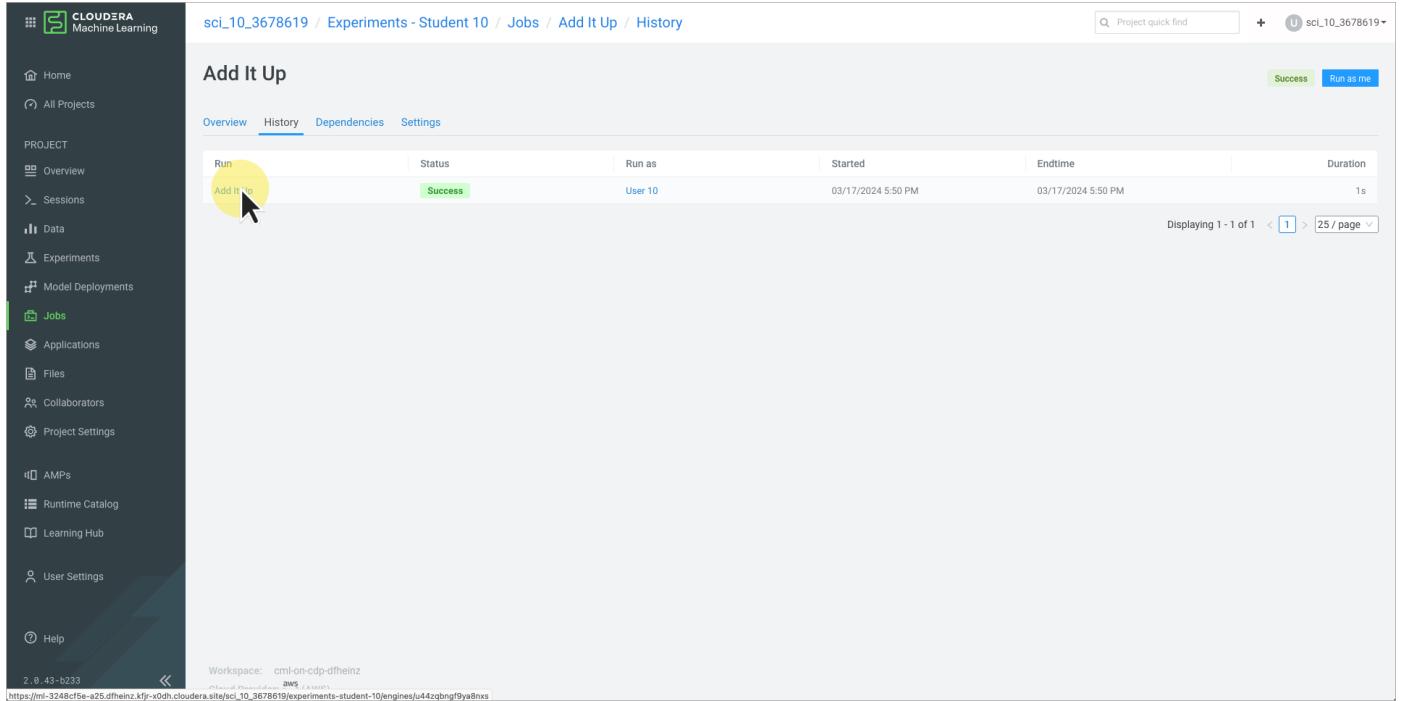
Displaying 1 - 1 of 1 < 1 > 25 / page

11. The job overview is displayed. Select the **History** tab.



The screenshot shows the 'Add It Up' job details page. The 'History' tab is highlighted with a yellow circle and a cursor. The 'Overview' tab is also visible. The job configuration includes a script of 'code/python/add.py', a manual schedule, and a resource profile of 2 vCPU / 4 GiB Memory. The job was created by 'User 10' and has a job ID of '2huu-5uz-c4v5-sjw1'. The latest run was successful, taking 0.01 seconds. The 'Job History' section displays a chart of duration over time, with a single data point at 0.01 seconds. The x-axis shows dates from March 17 to March 18, and the y-axis shows duration in seconds from 7.2 to 8.8.

12. A list of job runs is displayed. There should only be one job run at this point. Click on the run.



The screenshot shows the 'Add It Up' job history page. The 'History' tab is selected, indicated by a yellow circle and a cursor. The table lists one run: 'Add It Up' with status 'Success', run as 'User 10', started at 03/17/2024 5:50 PM, ended at 03/17/2024 5:50 PM, and duration 1s. The table has columns for Run, Status, Run as, Started, Endtime, and Duration. The bottom right corner shows pagination information: 'Displaying 1 - 1 of 1 < 1 > 25 / page'.

13. View the session output. The sum of the arguments is displayed.

The screenshot shows the Cloudera Machine Learning interface. On the left, there's a sidebar with various project management and data analysis tools like Home, All Projects, Overview, Sessions, Data, Experiments, Model Deployments, Jobs, Applications, Files (which is highlighted with a yellow circle), Collaborators, Project Settings, AMPs, Runtime Catalog, and Learning Hub. The main area is titled 'Add It Up' and shows a 'Success' status. It displays the command-line session logs:

```
> import sys
> params = [int(number) for number in sys.argv[1:]]
> total = sum(params)
> print(f"The sum of the numbers is: {total}")
The sum of the numbers is: 58
```

At the bottom, it says 'Workspace: cml-on-cdp-dfheinz' and 'Cloud Provider: AWS (AWS)'.

Creating an Experiment

1. Click **Files** in the left-side menu.

The screenshot shows the Cloudera Machine Learning interface with the 'Files' section highlighted in the sidebar. The main area shows a list of files with one item named 'add my' selected. The file details are shown on the right: Size 128 B, Last Modified 2 minutes ago, and Edit and Show Hidden Files options. At the top, there are 'Download', 'New', and 'Upload' buttons. The workspace is 'cml-on-cdp-dfheinz' and the cloud provider is 'AWS (AWS)'.

2. Navigate to `add.py` and click **Open in Session**.

```

import sys
params = [int(number) for number in sys.argv[1:]]
total = sum(params)
print(f"The sum of the numbers is: {total}")

```

3. The file is opened in the Workspace file editor. An actual session is not needed to edit the file.

Edit the file to look as follows:

```

import sys
import mlflow

mlflow.set_experiment("Add It Up")
mlflow.start_run()

params = [int(number) for number in sys.argv[1:]]
total = sum(params)

mlflow.log_param("Input", params)
mlflow.log_metric("Sum", total)

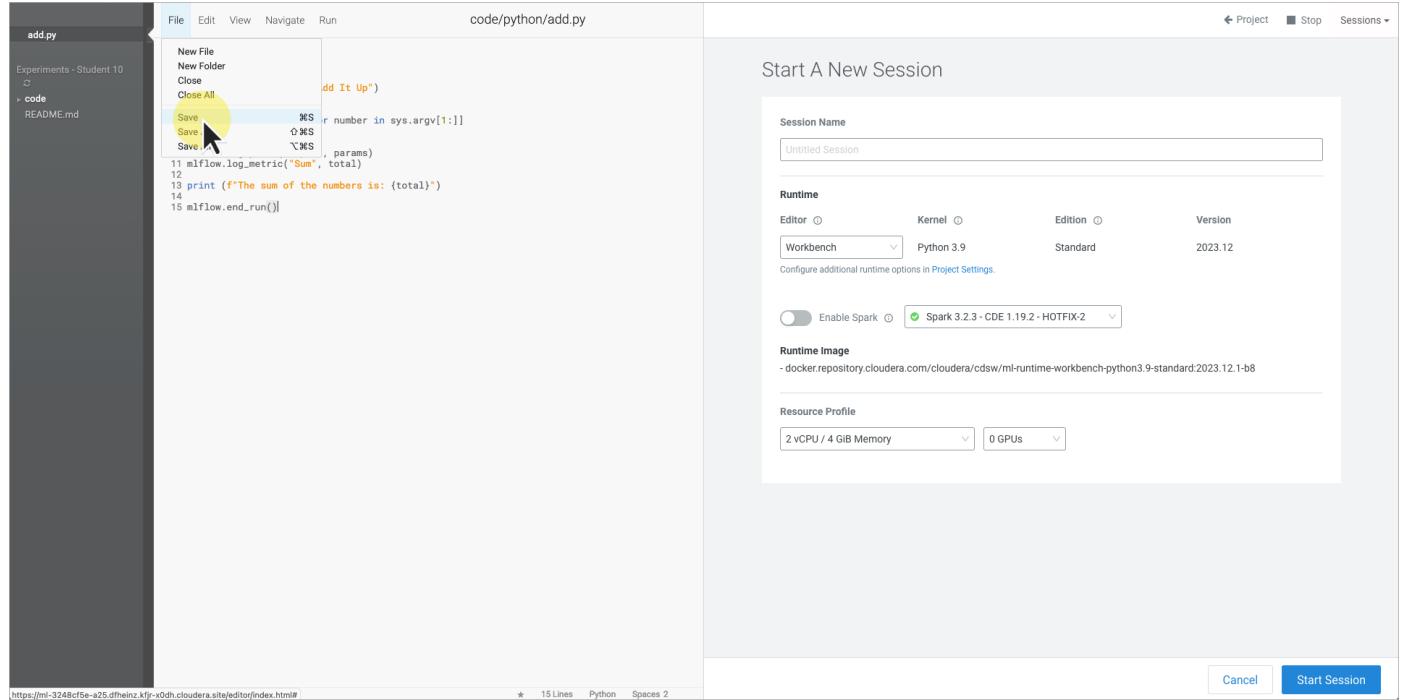
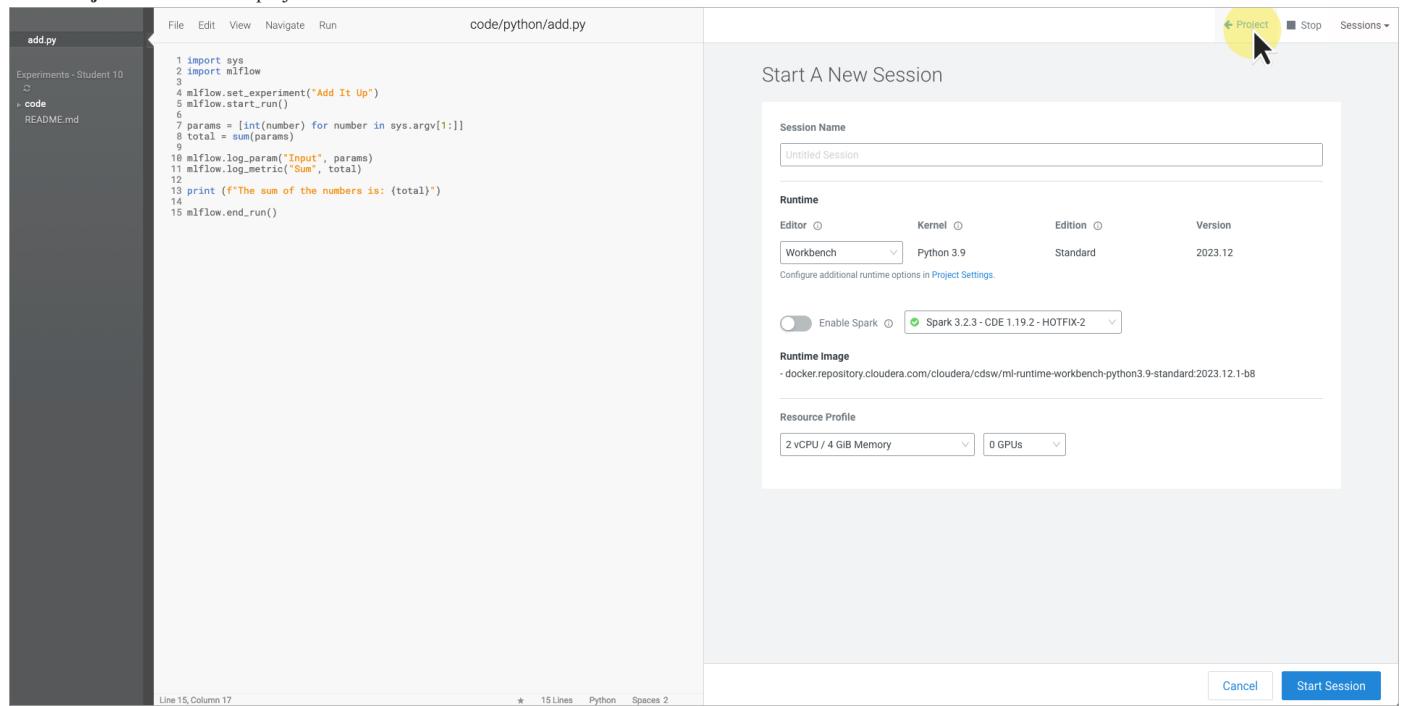
print(f"The sum of the numbers is: {total}")

mlflow.end_run()

```

Here is a brief description of the added lines:

- `import mlflow` - This line imports the mlflow module.
- `mlflow.start_run()` - MLflow can track multiple runs. In this case, only one run is tracked per execution. This line marks the start of the run.
- `mlflow.set_experiment("Add It Up")` - This line specifies which experiment to use. If the experiment does not exist, it will be created.
- `mlflow.log_param("Input", params)` - This line tracks the input parameters.
- `mlflow.log_metric("Sum", total)` - This line tracks the output metrics.
- `mlflow.end_run()` - This line marks the end of the run.

Click File and then Save.**4. Click Project to return to the project.**

5. Click **Jobs** in the left-side menu.

sci_10_3678619 / Experiments - Student 10

Experiments - Student 10

Models
This project has no models yet. Create a [new model](#).

Jobs

Name	Runs / Failures	Duration	Status	Latest Run	Actions
Add It Up	1 / 0	00:01	Success	2 minutes ago	Run

Files

Name	Size	Last Modified
code	-	3 minutes ago
README.md	127 B	3 minutes ago

CML on CDP - Experiment Tracking
This is a simple project to demonstrate experiment tracking in CML on CDP.
VERSION: 1.0.1

Workspace: cml-on-cdp-dfheinz
Cloud Provider: AWS (AWS)

6. Click **Run as [user icon]** to execute the job again.

sci_10_3678619 / Experiments - Student 10 / Jobs

Jobs

Job Dependencies for Add It Up

Name	ID	Status	Project	Created by	Created At	Runs / Failures	Duration	Latest Run	Actions
Add It Up	2	Success	Experiments - Student 10	User 10	03/17/2024 5:50 PM	1 / 0	1s	2 minutes ago	Run as

Displaying 1 - 1 of 1 < [1] > [25 / page]

Workspace: cml-on-cdp-dfheinz
Cloud Provider: AWS (AWS)

7. Once the **Status** changes to **Success**, click on the job name.

scI_10_3678619 / Experiments - Student 10 / Jobs

Jobs

Job Dependencies for Add It Up

Add It Up

+ Add Job Dependency

Name	ID	Status	Project	Created by	Created At	Runs / Failures	Duration	Latest Run	Actions
Add It Up	2	Success	Experiments - Student 10	User 10	03/17/2024 5:50 PM	2 / 0	3s	a few seconds ago	Run as ↗

Displaying 1 - 1 of 1 < 1 > 25 / page ↴

8. Click on the **History** tab.

scI_10_3678619 / Experiments - Student 10 / Jobs / Add It Up / Overview

Add It Up

Overview History Dependencies Settings

Script: code/python/add.py
Schedule: Manual
Resource Profile: 2 vCPU / 4 GiB Memory
Created By: User 10
Job Id: 2hui-5ruz-c4v5-sjwjt

Latest Run: a few seconds ago
Duration: 00:03
Runs: 2
Failures: 0

Job History

Duration (s)

8.5

8

Mar 17 17:50 Mar 17 17:51 Mar 17 17:51 Mar 17 17:51 Mar 17 17:51 Mar 17 17:52 Mar 17 17:52 Mar 17 17:53 Mar 17 17:53 Mar 17 17:53

Mar 17 17:50 Mar 17 17:51 Mar 17 17:51 Mar 17 17:51 Mar 17 17:51 Mar 17 17:52 Mar 17 17:52 Mar 17 17:53 Mar 17 17:53 Mar 17 17:53

Workspace: cml-on-cdp-dfheinz aws s3a

https://ml-3248cf5e-a25.dfeheinz.kfp-x0dh.cloudera.site/sci_10_3678619/experiments-student-10/jobs/2/history

9. Click on the latest run.

The screenshot shows the 'History' tab for the 'Add It Up' experiment. There are two runs listed:

Run	Status	Run as	Started	Endtime	Duration
Add It Up	Success	User 10	03/17/2024 5:53 PM	03/17/2024 5:53 PM	3s
Add it	Success	User 10	03/17/2024 5:50 PM	03/17/2024 5:50 PM	1s

10. In the output, a message is displayed to notify you that a new experiment was created. Click **Experiments** in the left-side menu.

The screenshot shows the 'Logs' tab for the 'Add It Up' session. The log output includes the following message:

```

2024/03/17 22:53:26 INFO mlflow.tracking.fluent: Experiment with name 'Add It Up' does not exist. Creating a new experiment.
<Experiment: artifact_location='/home/cdsw/.experiments/0pmq-9vy2-le6w-x1pq', creation_time=None, experiment_id='0pmq-9vy2-le6w-x1pq', last_update_time=None, lifecycle_stage='active', name='Add It Up', tags={}>

```

11. Click on the newly created experiment.

The screenshot shows the 'Experiments' page in the Cloudera Machine Learning interface. The URL is https://ml-3248cf5e-a25.dheinz.kfp-x0dh.cloudera.site/sci_10_3678619/experiments-student-10/cmflow/0pmq-9vy2-le6w-x1pq. The page displays a single experiment named 'Add It Up' created by 'User 10' on '03/17/2024 5:53 PM'. A yellow circle highlights the experiment row in the table.

12. Information about the experiment and a list of runs is displayed. The parameters and metrics for the run are displayed. Click on the run.

The screenshot shows the 'Experiment' details page for 'Add It Up'. The URL is https://ml-3248cf5e-a25.dheinz.kfp-x0dh.cloudera.site/sci_10_3678619/experiments-student-10/cmflow/0pmq-9vy2-le6w-x1pq/run/10zy-hp2q-gh0-0u5o. The page shows the experiment configuration and a table of runs. A yellow circle highlights the first run row in the 'Runs (1)' table.

	Status	Start Time	Run Name	Duration	User	Source	Version	Models	Parameters	Metrics	Tags
<input type="checkbox"/>	✓	2024-03-17 05:53:35	10zy-hp2q-gh0-0u5o	48ms	sci_10_3678619	python3	6ab41f	-	[20, 30]	50	s5y gabihpf gmsj18

13. A more detailed view of the run is displayed. Notice that in addition to the parameters and metrics, a run can have more information like notes, tags, and artifacts. Click **Files** in the left-side menu.

14. Navigate to `add.py` and click **Open in Session**. Add the following code to create a new metric called `count` and a new artifact, the `result.txt` file:

```
import sys
import mlflow

mlflow.set_experiment("Add It Up")
mlflow.start_run()

params = [int(number) for number in sys.argv[1:]]
count = len(params)
total = sum(params)

with open("result.txt", "w") as output_file:
    output_file.write(f"Input: {params},")
    output_file.write(f"Count: {count},")
    output_file.write(f"Sum: {total}\n")
    output_file.close()

mlflow.log_param("Input", params)
mlflow.log_metric("Count", count)
mlflow.log_metric("Sum", total)

mlflow.log_artifact("result.txt")

print(f"The sum of the numbers is: {total}")

mlflow.end_run()
```

Here is a brief description of the added lines: * `count = len(params)` - This is a new metric for tracking the number of numbers to sum.

- `with open("result.txt", "w") ...` - This line and the following four lines create a file (artifact) with the results of the run.
- `mlflow.log_metric("Count", count)` - This line tracks the new count metric.
- `mlflow.log_artifact("result.txt")` - This line tracks the new artifact.

Save the file and click **Project** to return to the project.

15. Click **Jobs** in the left-side menu and click **Run as me** on the job.
 16. Click **Experiments** in the left-side menu and click the experiment in the list.

17. Notice, the list of runs includes the new **Count** metric. Click on the latest run.

Parameters	Metrics	Tags
Input	Count 2 Sum 50	engineID s5ygabihpfgm... gr6hm09d92v...

18. The run details includes the new metric too.

Name	Value
Count	2
Sum	50

Name	Value	Actions
engineID	gr6hm09d92vcarn	

19. Scroll down, if needed, and view the Artifacts. If the job recently completed, the Artifacts may not have updated. Wait a minute and refresh the page, if you do not see the `result.txt` file in the list of Artifacts.

The screenshot shows the 'Artifacts' section of the Cloudera Machine Learning interface. The 'result.txt' file is listed with the following details:

- Full Path: /experiments/0pmq-9vy2-le6w-x1pq/1ftw-4efw-q8va-j9yt/artifacts/result.txt
- Size: 338
- Input: [20, 30], Count: 2, Sum: 50

Change the Input and Compare Runs

1. Click **Jobs** in the left-side menu and click the `Add It Up` job in the list.

The screenshot shows the 'Jobs' list in the Cloudera Machine Learning interface. The 'Add It Up' job is listed with the following details:

Name	ID	Status	Project	Created by	Created At	Runs / Failures	Duration	Latest Run	Actions
Add It Up	2	Success	Experiments - Student 10	User 10	03/17/2024 5:50 PM	3 / 0	4s	2 minutes ago	Run as ↗

Job Dependencies for Add It Up

2. Click the **Settings** tab.

sci_10_3678619 / Experiments - Student 10 / Jobs / Add It Up / Overview

Add It Up

Overview History Dependencies Settings

Script: code/python/add.py
Schedule: Manual
Resource Profile: 2 vCPU / 4 GiB Memory
Created By: User 10
Job Id: 2hu:5ruz:c4v5-sjw1

Latest Run: 2 minutes ago
Duration: 00:04
Runs: 3
Failures: 0

Job History

Duration (s) 11
9.5
8

Mar 17 17:51 Mar 17 17:52 Mar 17 17:53 Mar 17 17:54 Mar 17 17:55 Mar 17 17:56

2.0, 43-b233 << https://ml-3248cf5e-a2b.dfehrinz.kfj-x0dh.cloudera.site/sci_10_3678619/experiments-student-10/jobs/2/settings

3. Edit the Arguments to be something different, like 20 30 40.

sci_10_3678619 / Experiments - Student 10 / Jobs / Add It Up / Settings

Add It Up

Overview History Dependencies Settings

General

Name: Add It Up

Run Job as: me

Script: code/python/add.py

Arguments: 20 30

Runtime

Editor: Workbench

Kernel: Python 3.9

Edition: Standard

Configure additional runtime options in Project Settings.

Enable Spark: Spark 3.2.3 - CDE 1.19.2 - HOTFIX-2

Workspace: cmi-on-cdp-dfheinz
Cloud Provider: aws (AWS)

4. Scroll down and click **Update Job**

The screenshot shows the 'Add It Up' job configuration page. Key details include:

- Runtime Image:** docker.repository.cloudera.com/cloudera/cdsw/ml-runtime-workbench:python3.9-standard:2023.12.1-b8
- Schedule:** Manual
- Resource Profile:** 2 vCPU / 4 GiB Memory
- GPUs:** 0 GPUs
- Timeout In Minutes (optional):** 30
- Email Notifications Unavailable:** Outbound email configuration is missing or incorrect, preventing email notifications from being set up. Please contact your administrator to resolve this issue.
- Job Notifications:** Success, Run as me, Run
- Buttons:** Update Job (highlighted), Delete Job

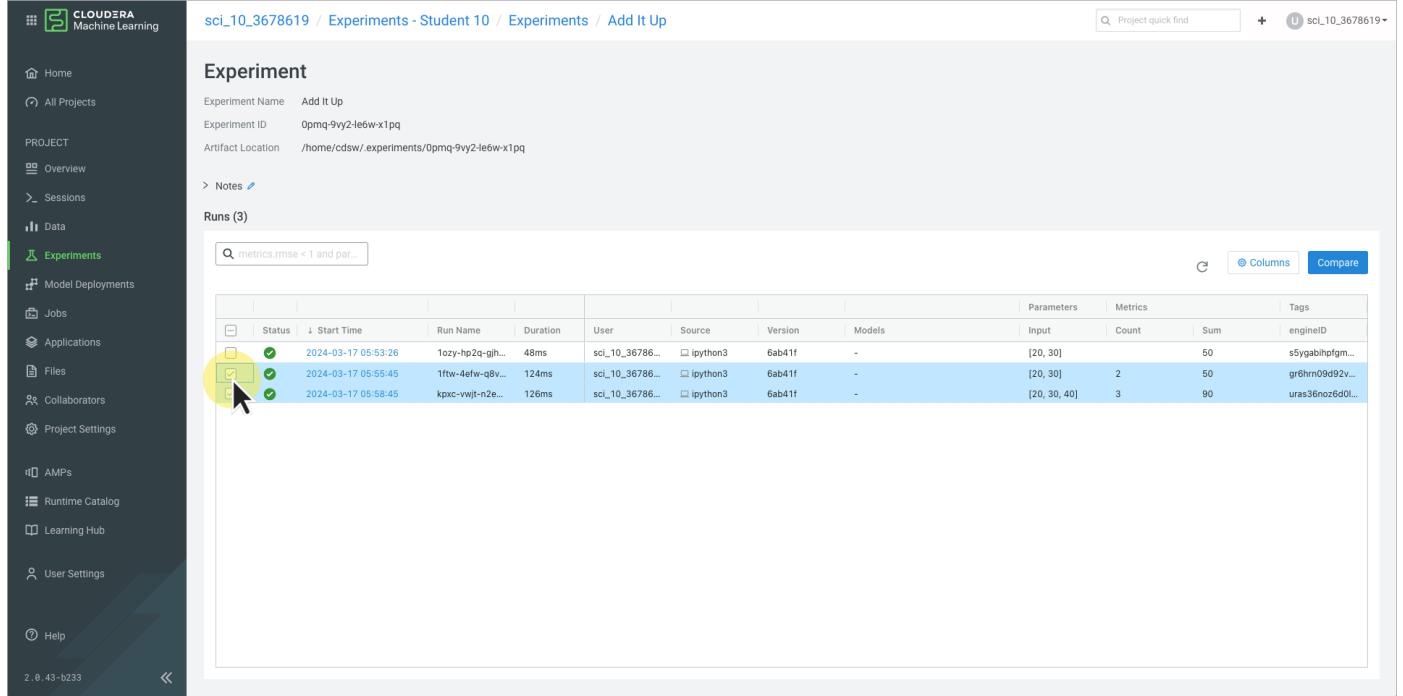
5. Click **Run as me**.

The screenshot shows the 'Add It Up' job overview page. Key details include:

- Script:** code/python/add.py
- Schedule:** Manual
- Resource Profile:** 2 vCPU / 4 GiB Memory
- Created By:** User 10
- Job ID:** 2hui-5ruz-c4v5-sjw!
- Latest Run:** 3 minutes ago, Duration: 00:04, Runs: 3, Failures: 0
- Job History:** A line chart showing Duration (s) over time, starting around 8 seconds and increasing slightly to about 9.5 seconds by the end of the period shown.
- Buttons:** Success, Run as me (highlighted), Run

6. Click **Experiments** in the left-side menu and click on the **Add It Up** experiment.

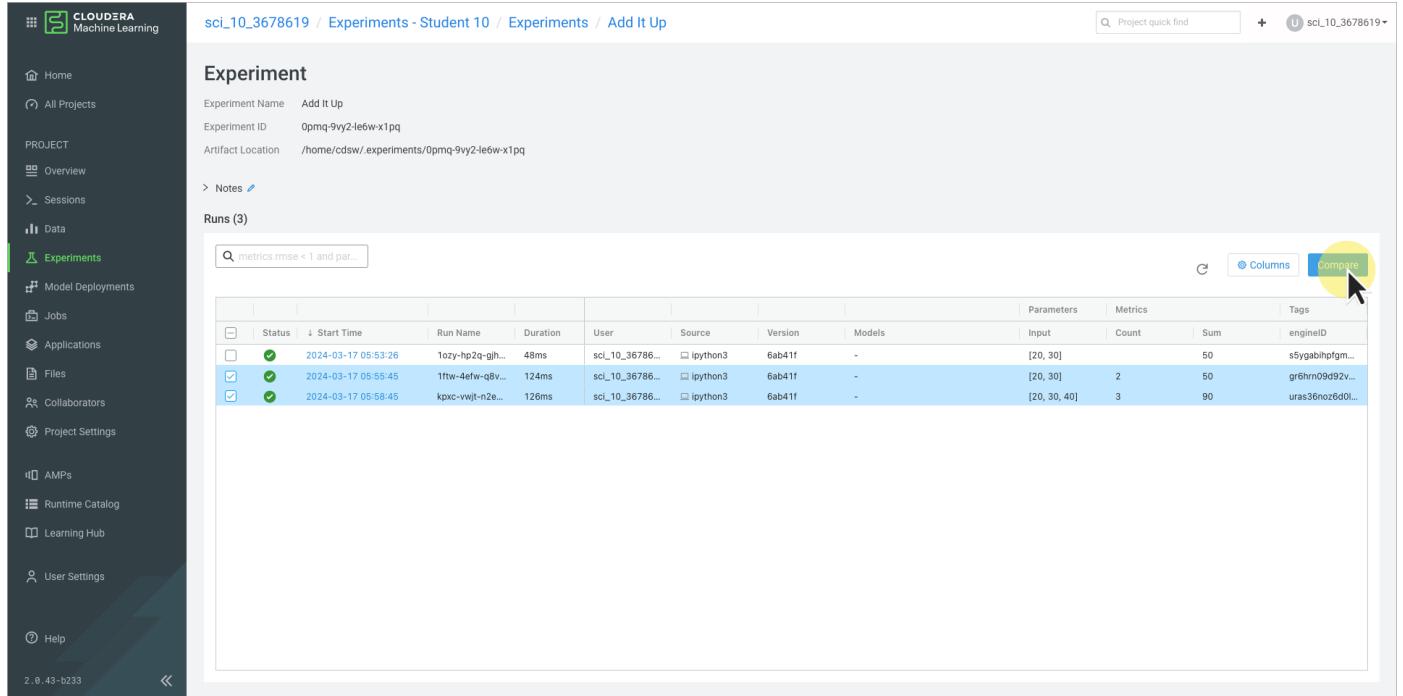
7. Select last two runs by clicking the respective checkboxes.



The screenshot shows the 'Experiment' page for the experiment 'Add It Up'. The left sidebar includes links for Home, All Projects, Overview, Sessions, Data, Experiments (which is selected), Model Deployments, Jobs, Applications, Files, Collaborators, Project Settings, AMPs, Runtime Catalog, Learning Hub, User Settings, and Help. The main content area displays the experiment details: Experiment Name 'Add It Up', Experiment ID '0pmq-9vy2-le6w-x1pq', and Artifact Location '/home/cdsw/experiments/0pmq-9vy2-le6w-x1pq'. Below this is a 'Notes' section and a 'Runs (3)' table. The table has columns for Status, Start Time, Run Name, Duration, User, Source, Version, Models, Parameters, Metrics, and Tags. The first run (2024-03-17 05:53:26) has its checkbox highlighted with a yellow circle. The other two runs (2024-03-17 05:55:45 and 2024-03-17 05:58:45) also have their checkboxes checked.

	Status	Start Time	Run Name	Duration	User	Source	Version	Models	Parameters	Metrics	Tags
<input type="checkbox"/>	✓	2024-03-17 05:53:26	1ozy-hp2q-gh...	48ms	sci_10_36786...	ipython3	6ab41f	-	[20, 30]	50	s5ygabihpfgrm...
<input checked="" type="checkbox"/>	✓	2024-03-17 05:55:45	1ftw-4efw-q8v...	124ms	sci_10_36786...	ipython3	6ab41f	-	[20, 30]	2	gr6hrn09d2v...
<input checked="" type="checkbox"/>	✓	2024-03-17 05:58:45	kpxc-vwjt-n2e...	126ms	sci_10_36786...	ipython3	6ab41f	-	[20, 30, 40]	3	uras36noz6d0l...

8. Click Compare.



The screenshot shows the same 'Experiment' page for 'Add It Up'. The left sidebar and experiment details are identical to the previous screenshot. In the 'Runs (3)' table, the first run's checkbox is unselected, while the other two runs remain checked. A yellow circle highlights the 'Compare' button in the top right corner of the table header. The 'Compare' button is blue with white text.

9. Compare the results of the two runs using the table at the top and the various plots.

Run Comparison (2)

Run ID	1ftw-4efw-q8va-j9yt	kpxc-vvjt-n2ea-3uu2
Run Name	1ftw-4efw-q8va-j9yt	kpxc-vvjt-n2ea-3uu2
Start Time	2024-03-17 05:55:45	2024-03-17 05:58:45

Parameters

Input	[20, 30]	[20, 30, 40]
-------	----------	--------------

Metrics

Count	2	3
Sum	50	90

Scatter Plot Contour Plot Parallel Coordinates Plot

X Axis: Input
Y Axis: Count

The scatter plot shows one data point at (Input: 30, Count: 3).

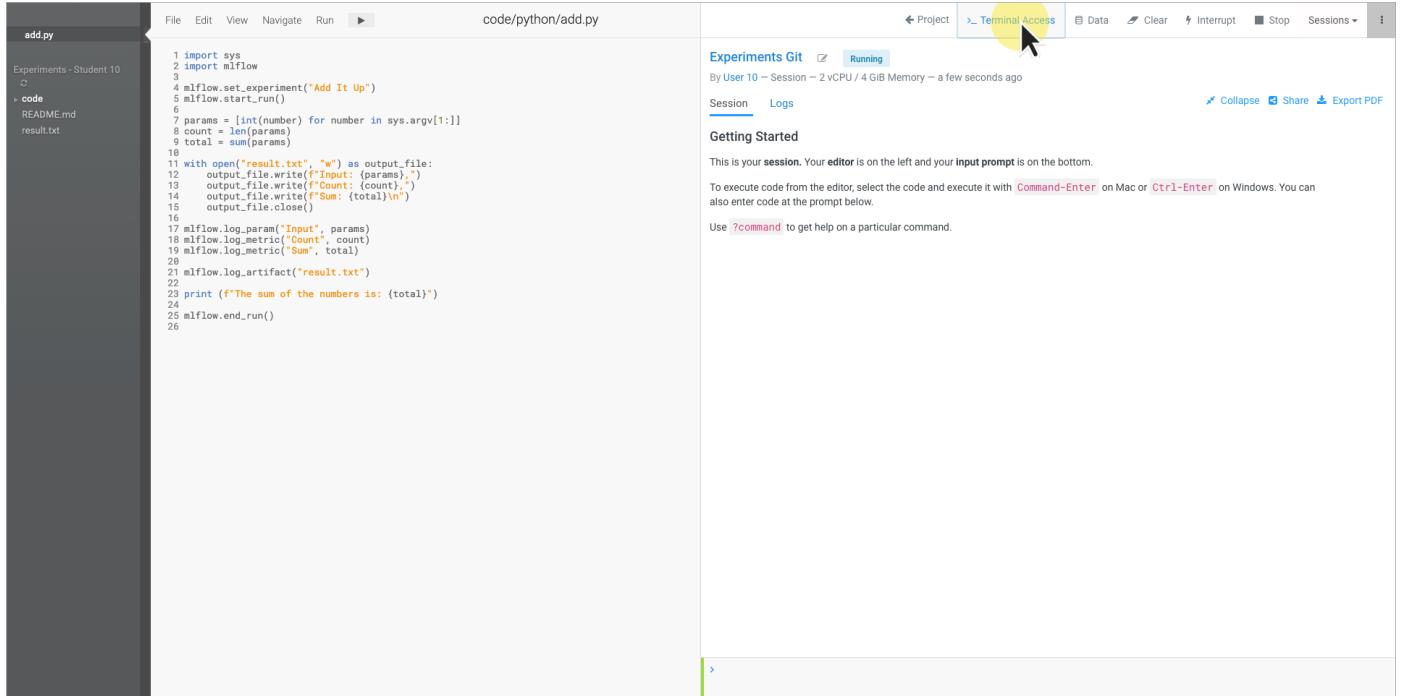
Commit the Changes to Git

Now that you have modified the `add.py` code to track metrics, you will commit the changes to git so they won't be lost.

1. Click **Sessions** in the left-side menu and click **New Session**.

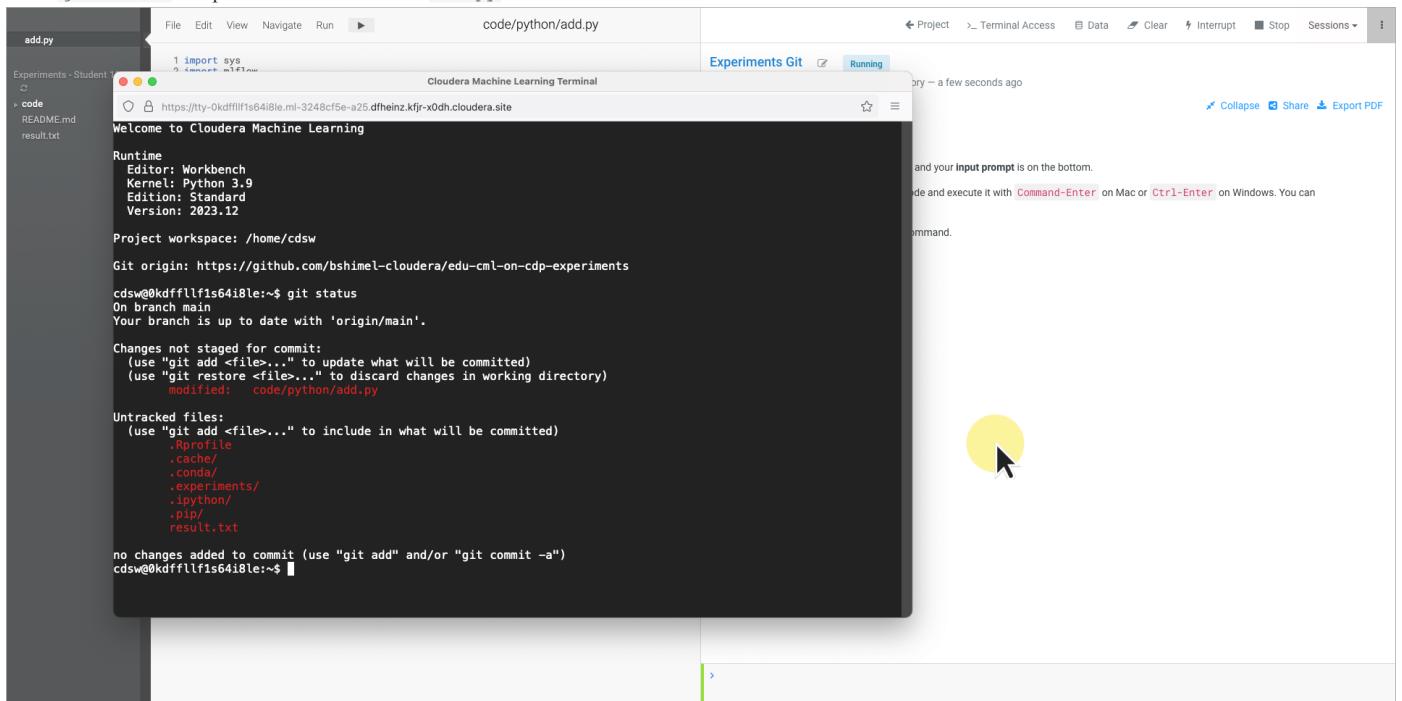
2. Enter a session name and click **Start Session**.

3. Once the session has started, click the **Terminal Access** button.



The screenshot shows the Cloudera Machine Learning interface. On the left, there's a file explorer with files like 'add.py', 'README.md', and 'result.txt'. In the center, there's a code editor window titled 'code/python/add.py' containing Python code for summing numbers from command-line arguments. On the right, there's a terminal window titled 'Experiments Git' with the status 'Running'. A yellow circle highlights the 'Terminal Access' button in the top navigation bar.

4. Enter `git status` and press **Enter**. You will see `add.py` has been modified.



The screenshot shows the terminal window from the previous step. It displays the output of the `git status` command. The output shows that the file `add.py` has been modified. A yellow circle highlights the terminal window area.

```

File Edit View Navigate Run > code/python/add.py
Experiments - Student 10
code README.md result.txt
1 import sys
2 import miflow
3
4 miflow.set_experiment("Add It Up")
5 miflow.start_run()
6
7 params = [int(number) for number in sys.argv[1:]]
8 count = len(params)
9 total = sum(params)
10
11 with open('result.txt', 'w') as output_file:
12     output_file.write(f"Input: {params}\n")
13     output_file.write(f"Count: {count}\n")
14     output_file.write(f"Sum: {total}\n")
15     output_file.close()
16
17 miflow.log_params("Input", params)
18 miflow.log_metric("Count", count)
19 miflow.log_metric("Sum", total)
20
21 miflow.log_artifact("result.txt")
22
23 print(f"The sum of the numbers is: {total}")
24
25 miflow.end_run()

File Edit View Navigate Run > code/python/add.py
Experiments Git Running
By User 10 — Session — 2 vCPU / 4 GiB Memory — a few seconds ago
Session Logs
Getting Started
This is your session. Your editor is on the left and your input prompt is on the bottom.
To execute code from the editor, select the code and execute it with Command-Enter on Mac or Ctrl-Enter on Windows. You can also enter code at the prompt below.
Use ?command to get help on a particular command.

>
1 import sys
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

Welcome to Cloudera Machine Learning
Runtime
Editor: Workbench
Kernel: Python 3.9
Edition: Standard
Version: 2023.12

Project workspace: /home/cdsw

Git origin: https://github.com/bshimel-cloudera/edu-cml-on-cdp-experiments
cdsw@0kdfllf1s64i8le:~$ git status
On branch main
Your branch is up to date with 'origin/main'.

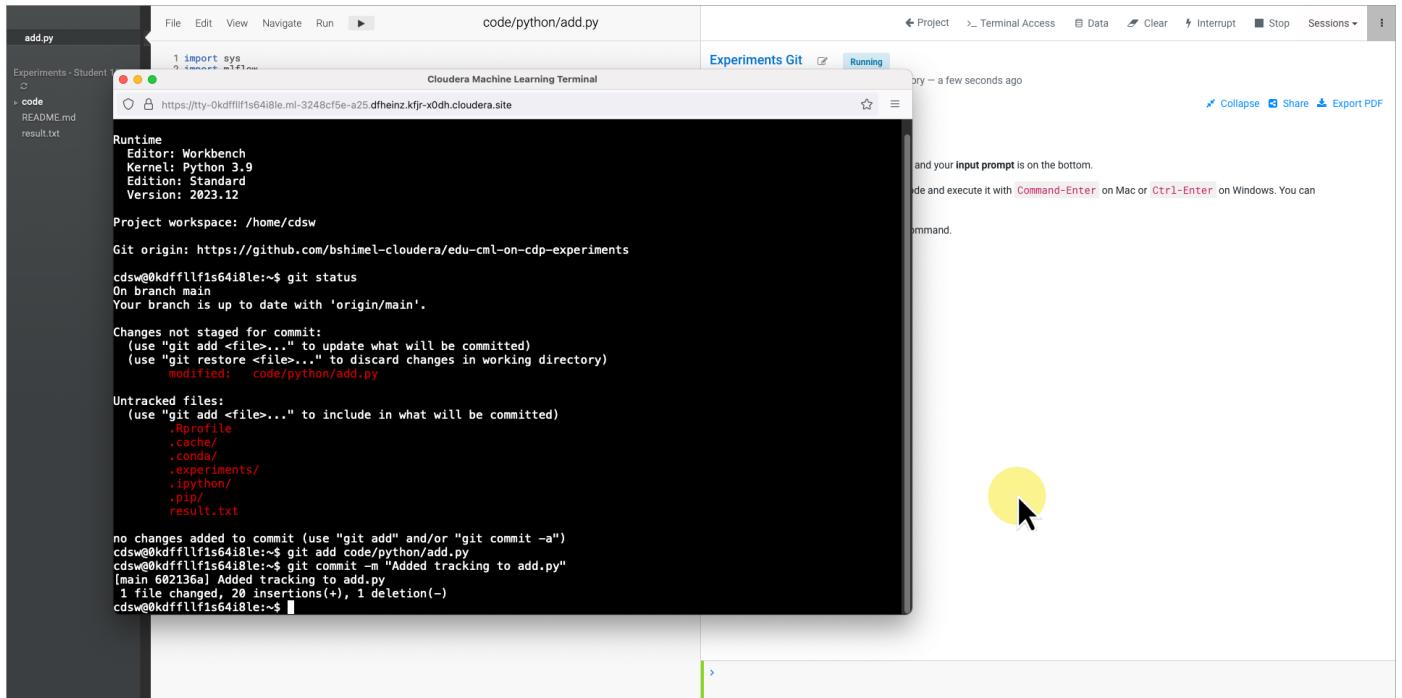
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:  code/python/add.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .Rprofile
    .cache/
    .conda/
    .experiments/
    .ipython/
    .pip/
    result.txt

no changes added to commit (use "git add" and/or "git commit -a")
cdsw@0kdfllf1s64i8le:~$ 
```

5. Add the file to git and commit the changes.

```
git add code/python/add.py
git commit -m "Added tracking to add.py"
```



The screenshot shows a terminal window titled "Experiments - Student View" with the tab "code/python/add.py" selected. The terminal output is as follows:

```

add.py
File Edit View Navigate Run Project Terminal Access Data Clear Interrupt Stop Sessions

1 import sys
2 import os
3
4 print("Hello, World!")

Cloudera Machine Learning Terminal
Experiments Git Running
try -- a few seconds ago
https://ity-kdflllf1s64i8le.ml-3248cf5e-a25 dfheinz.kfr-x0dh.cloudera.site
Collapse Share Export PDF

Runtime
Editor: Workbench
Kernel: Python 3.9
Edition: Standard
Version: 2023.12

Project workspace: /home/cdsw

Git origin: https://github.com/bshimmel-cloudera/edu-cml-on-cdp-experiments

cdsw@0kdflllf1s64i8le:~$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
      modified:   code/python/add.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .Rprofile
    .cache/
    .conda/
    .experiments/
    .ipython/
    .pip/
    result.txt

no changes added to commit (use "git add" and/or "git commit -a")
cdsw@0kdflllf1s64i8le:~$ git add code/python/add.py
cdsw@0kdflllf1s64i8le:~$ git commit -m "Added tracking to add.py"
[main 602136a] Added tracking to add.py
1 file changed, 20 insertions(+), 1 deletion(-)
cdsw@0kdflllf1s64i8le:~$ 

```

Your file changes have been committed. Normally, you would push the changes back to a remote repository. However, in this case, you do not have permission to commit to the remote repository.

End of Exercise

Using Workbench for Lecture and Exercises

The next several lectures and exercises will be completed using the CML Workbench. The files for the lectures and exercises are provided as an AMP. In this brief exercise, you will create a new project based on the AMP.

1. Navigate to your Cloudera Machine Learning workspace.

2. Click the New Project button.

Welcome to CML, User 10.

Create a new project

Deploy a prototype

Create a notebook

Visualize your data

Recent Projects

You currently don't have any projects.

Select an option above to get started.

Product Tour

Take a CML product tour

In this demo, we walk you through the key capabilities of Cloudera Machine Learning and the machine learning development workflow it enables.

Explore Use Cases

Take AI from concept to reality

This demonstration walks you through the end-to-end process of building and deploying AI applications.

Enable exploratory Data Science

This demonstration walks you through the exploratory data science capabilities using our unified toolset that accelerates the machine learning lifecycle.

Featured Announcements

Model Registry is Generally Available! NEW

The Model Registry serves as a centralized hub for storing, managing, and deploying machine learning models and their...

October 22, 2023

AMP - Using Amazon Bedrock for Text Summarization and More NEW

Amazon Bedrock is a new AWS Cloud service which allows convenient api access to a number of text and image generat...

September 28, 2023

AMP - Fine Tuning a Foundation Model for Multiple Tasks NEW

Fine Tuning a Foundation Model using techniques like Parameter-Efficient Fine-Tuning (PEFT) and Quantization ...

September 14, 2023

See more at the Learning Hub >

Helpful Links

Documentation

Need help? Check out our comprehensive documentation for detailed instructions and information.

Community

Connect with other users and get support on our active community forum.

Workspace: cmi-on-cdp-dfheinz AWS (AWS)

https://ml-3248cf5e-a25.dfheinz.ktj-x0dh.cloudera.site/projects/new

3. Enter Student # for Project Name

New Project

Project Name

Project Description

Project Visibility

Private - Only added collaborators can view the project

Public - All authenticated users can view this project.

Initial Setup

Blank Template AMPs Local Files Git

Templates include example code to help you get started.

Runtimes

Projects are configured with the latest Python and R ML Runtimes. You can change this configuration under the Advanced Options.

Cancel Create Project

4. Select AMPs for the Initial Setup.

New Project

Project Name: Student 10

Project Description:

Project Visibility: Private - Only added collaborators can view the project

Initial Setup: AMPs (highlighted)

Runtimes: Projects are configured with the latest Python and R ML Runtimes. You can change this configuration under the Advanced Options.

Cancel Create Project

5. Enter <https://github.com/bshimel-cloudera/edu-cml-on-cdp> into the Provide Git URL of your AMPs field.

New Project

Project Name: Student 10

Project Description:

Project Visibility: Private - Only added collaborators can view the project

Initial Setup: AMPs (highlighted)

Applied ML Prototypes provide components to create a complete project. They may include jobs, models and experiments.

Provide the Git URL of the project to clone. Select the option that applies to your URL access.

HTTPS (highlighted)

GIT URL: https://username:password@mygithost.com/my/repository

You are able to provide username/password.
e.g. https://username:password@mygithost.com/my/repository

Create Project

6. Click the **Create Project** button.

7. Select **Python 3.7** as the **Kernel**, if it is not already selected.

8. Click the **Launch Project** button.

Configure Project: Student 10

AMP Name: Predicting with Cloudera Machine Learning (v2)

Exercise Guide for Predicting with Cloudera Machine Learning

Environment Variables
This prototype does not define any environment variables.

Runtime

Editor: Workbench Kernel: Python 3.7 Edition: Standard Version: 2024.02

Selected Python kernel is deprecated!
Please consider using kernel with higher Python version.

Enable Spark: Spark 3.2.3 - CDE 1.19.2 - HOTFIX-2

Runtime Image: docker.repository.cloudera/cdsw/ml-runtime-workbench-python3.7-standard:2024.02.1-b4

No Runtime Addon is required for this AMP.

Setup Steps

Execute AMP setup steps

Launch Project

9. Wait for AMP Setup to Complete. The AMP is installing the Python modules and exercise files. The process should take about 5 minutes to complete.

10. Select **Overview** from the project menu on the left.

sci_10_3678619 / Student 10 / AMP Status

AMP Name: Predicting with Cloudera Machine Learning (v2)

Exercise Guide for Predicting with Cloudera Machine Learning

Completed all steps

Step	Action	Status
Step 1	Run session View details	completed 3/17/2024 5:18 PM
<code>!pip3 install -r requirements.txt --progress-bar off</code>		
Step 2	Run session View details	completed 3/17/2024 5:18 PM
<code>!chmod 755 cml/setup-env.sh !cml/setup-env.sh</code>		

Overview

AMP Status

AMP Name: Predicting with Cloudera Machine Learning (v2)

Exercise Guide for Predicting with Cloudera Machine Learning

Completed all steps

Step 1 Run session View details completed 3/17/2024 5:18 PM

`!pip3 install -r requirements.txt --progress-bar off`

Step 2 Run session View details completed 3/17/2024 5:18 PM

`!chmod 755 cml/setup-env.sh
!cml/setup-env.sh`

11. Click the New Session button.

The screenshot shows the Cloudera Machine Learning Workbench interface. On the left is a dark sidebar with various navigation options like Home, All Projects, PROJECT Overview, Sessions, Data, Experiments, Model Deployments, Jobs, Applications, Files, Collaborators, Project Settings, AMPs, Runtime Catalog, Learning Hub, User Settings, and Help. The main workspace is titled "Student 10". It displays a message "Project creation succeeded!" with a link to "View status page". Below this, there are sections for Models (with a note about no models yet), Jobs (with a note about no jobs yet), and Files (listing files like cml, exercises, env.py, README.md, and requirements.txt). At the top right of the workspace, there is a "New Session" button, which is highlighted with a yellow circle. Other buttons visible include "Fork" and "Completed 3/17/2024 5:18 PM". The bottom of the screen shows the URL "https://ml-3248cf5e-a25.dheinz.ktj-x0dh.cloudera.site/sci_10_3678619/student-10/sessions/new" and the status "2.0.43-b233".

12. Enter Student # for Session Name.

The screenshot shows the "Start A New Session" dialog box. It has a "Session Name" input field where "Student 10" is typed, which is highlighted with a yellow circle. Below it are sections for "Runtime" (with options for Editor, Kernel, Edition, and Version), "Resource Profile" (set to 2 vCPU / 4 GiB Memory and 0 GPUs), and "Runtime Image" (set to docker.repository.cloudera.com/cloudera/cdsw/ml-runtime-jupyterlab-python3.10-standard:2024.02.1-b4). At the bottom of the dialog are "Cancel" and "Start Session" buttons. The background shows the same Cloudera Machine Learning Workbench interface as the previous screenshot.



The workbench exercises require the Workbench Editor, Python 3.7 Kernel, and Spark 2.4.8 enabled. Also, you will receive a depreciation warning for Python 3.7 and Spark 2. This is normal.

1. Click the **Start Session** button.

Session Name: Student 10

Runtime:

- Editor: Workbench
- Kernel: Python 3.7
- Edition: Standard
- Version: 2024.02

Configure additional runtime options in [Project Settings](#).

Selected Python kernel is deprecated!
Please consider using kernel with higher Python version.

Spark 2 is deprecated!
Please consider using Spark 3.

Enable Spark: Spark 2.4.8 - CDE 1.19.2 - HOTFIX.2

Runtime Image: docker repository cloudera/cdsu/ml-runtime-workbench-python3.7-standard:2024.02.1-b4

Resource Profile:

- 2 vCPU / 4 GiB Memory
- 0 GPUs

Buttons: Cancel, Start Session

2. Close the **Connection Code Snippet** dialog.

File Edit View Navigate Run

Student 10 Project Terminal Access Data Clear Interrupt Stop Sessions

Connection Code Snippet

You have access to the Data Connections below.
This also be accessed by clicking the **Data** tab in the top menu.

datalake-dfheinz-530-class-3678619	hive-dfheinz
TYPE: Spark Data Lake	TYPE: Hive Virtual Warehouse

Use this code to connect to the chosen data source.

```
import cmldata_v1 as cmldata
# Sample in-code customization of spark configurations
#from pyspark import SparkContext
#SparkContext.setSystemProperty('spark.executor.cores', '1')
#SparkContext.setSystemProperty('spark.executor.memory', '2g')

CONNECTION_NAME = 'datalake-dfheinz-530-class-3678619'
conn = cmldata.get_connection(CONNECTION_NAME)
spark = conn.get_spark_session()
```

Copy Code

Don't show me this again for: This Project All Projects **Close**

3. Navigate to `exercises/code/python` in the files on the left side of the editor.

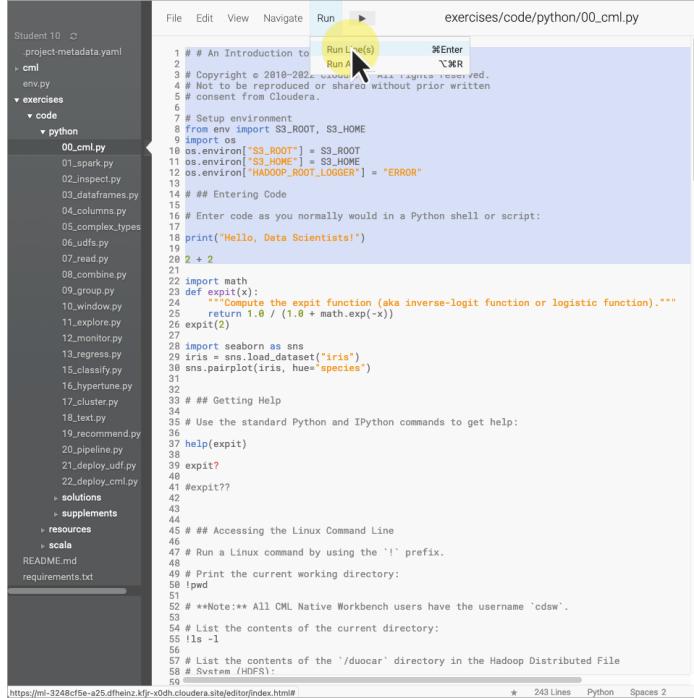
The screenshot shows the Cloudera Workbench interface. On the left, there is a file tree for 'Student 5'. The 'code' directory under 'exercises' contains several Python files, with '00_cml.py' highlighted. The main area is a code editor titled 'exercises/code/python/00_cml.py'. At the top of the editor, there are tabs for 'File', 'Edit', 'View', 'Navigate', and 'Run'. Below the tabs, the code for '00_cml.py' is displayed. The code includes imports like 'os', 'math', and 'seaborn', and defines functions for computing the expit function and plotting iris data. The status bar at the bottom indicates the URL <https://mi-3248cf6e-a26.dheinz.kylo-edn.cloudera.svc/editor/index.html#>, 243 lines of Python code, and 2 spaces.

4. Click on `cml.py` to open the file in the editor.

5. Highlight rows 1 - 20.

The screenshot shows the Cloudera Workbench interface. On the left, there is a file tree for 'Student 10'. The 'code' directory under 'exercises' contains several Python files, with '00_cml.py' highlighted. The main area is a code editor titled 'exercises/code/python/00_cml.py'. The code is identical to the one in the previous screenshot. The status bar at the bottom indicates the URL <https://mi-3248cf6e-a26.dheinz.kylo-edn.cloudera.svc/editor/index.html#>, 243 lines of Python code, and 2 spaces.

6. Select Run / Run Lines from them menu.



The screenshot shows the Cloudera Workbench interface. On the left, there's a file tree for a project named 'Student 10'. The 'code/python' directory contains several Python files like '00_cml.py', '01_spark.py', etc. In the center, a code editor window is open with the file 'exercises/code/python/00_cml.py'. At the top of this window, the 'Run' menu is open, and the option 'Run Line(s)' is highlighted with a yellow circle. To the right of the code editor, there's a session panel titled 'Student 10' which says 'Running'. It has tabs for 'Session' and 'Logs'. Below the tabs, there's some descriptive text about running code and help for commands.

The output from the first twenty lines should be displayed on the right.

End of Exercise

Autoscaling, Performance, and GPU Settings

The time-saving potential of using GPUs for complex and large tasks is massive, setting up these environments and tasks such as wrangling NVIDIA drivers, managing CUDA versions and deploying custom engines for your specific project needs can be time consuming and challenging. To make these processes simpler — and to get data scientists working on ML use cases faster — CML made it simple to configure and leverage NVIDIA GPUs natively.

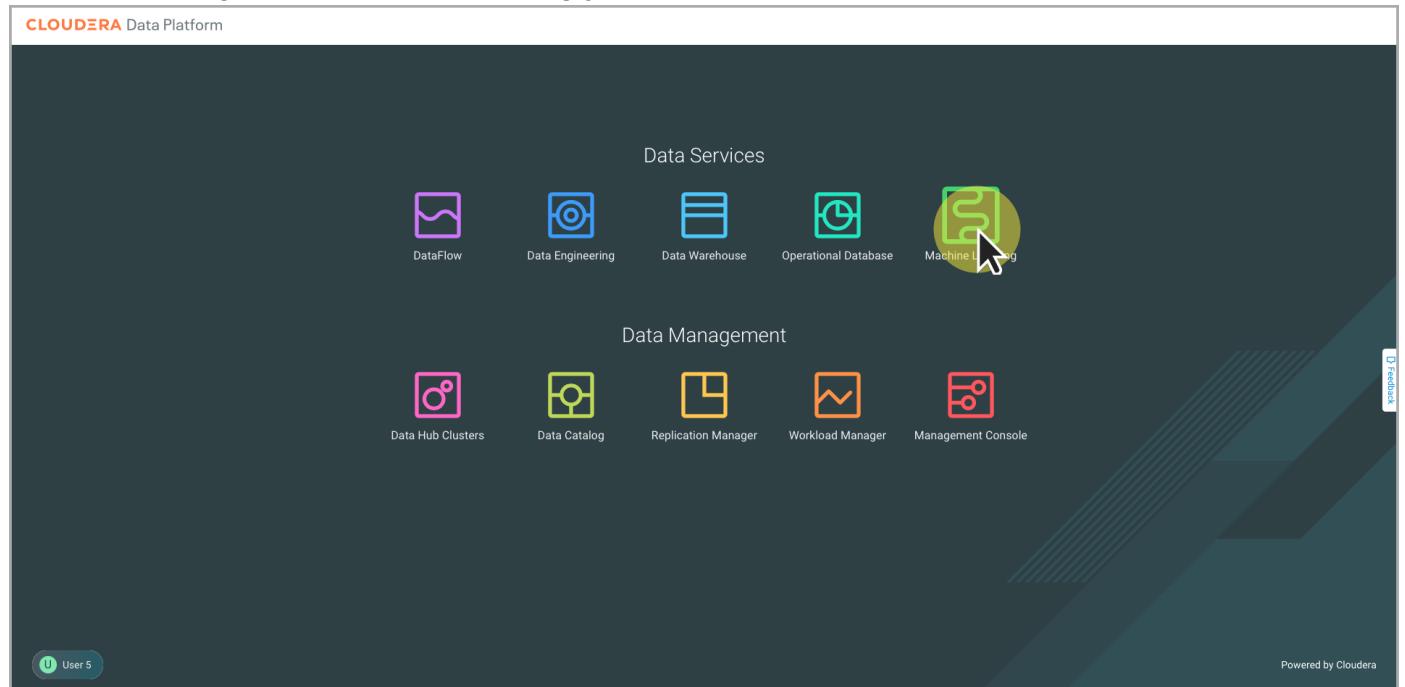
In this exercise, you will use a Computer Vision Image Classification example and train a deep learning model to classify fashion items leveraging the Fashion MNIST Dataset. The main focus of the exercise is to demonstrate how to use the GPU and auto scaling features of CML. If you would like to know more about the Computer Vision Image Classification example, you can [view the project on Github](#).

In this exercise, you will:

- Run an image classification without a GPU,
- Run an image classification example with a GPU, and
- Examine the difference in performance.

View Workspace Details and Allocated GPUs

1. Select **Machine Learning** from the Cloudera Data Platform home page.



2. Select the three dot icon in the **Actions** column for your workspace.

Status	Workspace	Environment	Region	Creation Date	Cloud Provider
Ready	cml-on-cdp	bshimel-500-class-22829	us-east-2	08/31/2022 2:42 PM CDT	aws AWS
Ready	bshimel-test-workspace	bshimel-500-class-22829	us-east-2	08/30/2022 12:44 PM CDT	aws AWS

3. Click **View Workspace Details**.

- [View Workspace Details](#)
- [View Event](#)
- [Manage Access](#)
- [Manage Remote Access](#)
- [Download Kubeconfig](#)
- [Open Grafana](#)
- [Upgrade Workspace](#)
- [Backup Workspace](#)
- [Remove Workspace](#)

4. Scroll down to view **Workspace Instances**. In the example below, there are no **CML GPU Workers** as indicated by the zero in the Count program. The **Autoscale Range** indicates that there could be zero to ten instances. Your workspace may show different values. For example, if someone else already requested a session with a GPU, you may have a one or greater value in the **Count** column. In the example below, the **Instance Type** for the CML GPU Worker is an AWS **p2.8xlarge**. This is a fairly beefy instance with 8 GPU cores. Therefore, if one instance is running, it can support up to eight sessions with one GPU before another instance will be

automatically allocated by the auto scaling feature.

The screenshot shows the 'Machine Learning Workspaces / cml-on-cdp' page. It includes sections for 'Machine Learning Workspaces' (listing environment details like Cloudera-Environment-Resource-Name, Cloudera-Resource-Name, ExperienceType, Owner, TenantID, WorkspaceCn, and WorkspaceName), 'Workspace Instances' (listing CPU Workers, GPU Workers, Infra, and Platform Infra with their respective instance types, CPU/GPU counts, and memory), and 'Subnets for Worker Nodes' (listing subnets with their availability zones and CIDRs). A red box highlights the 'CML GPU Workers' row.

5. Leave the Workspace Details tab open in your browser.

Create a New Project

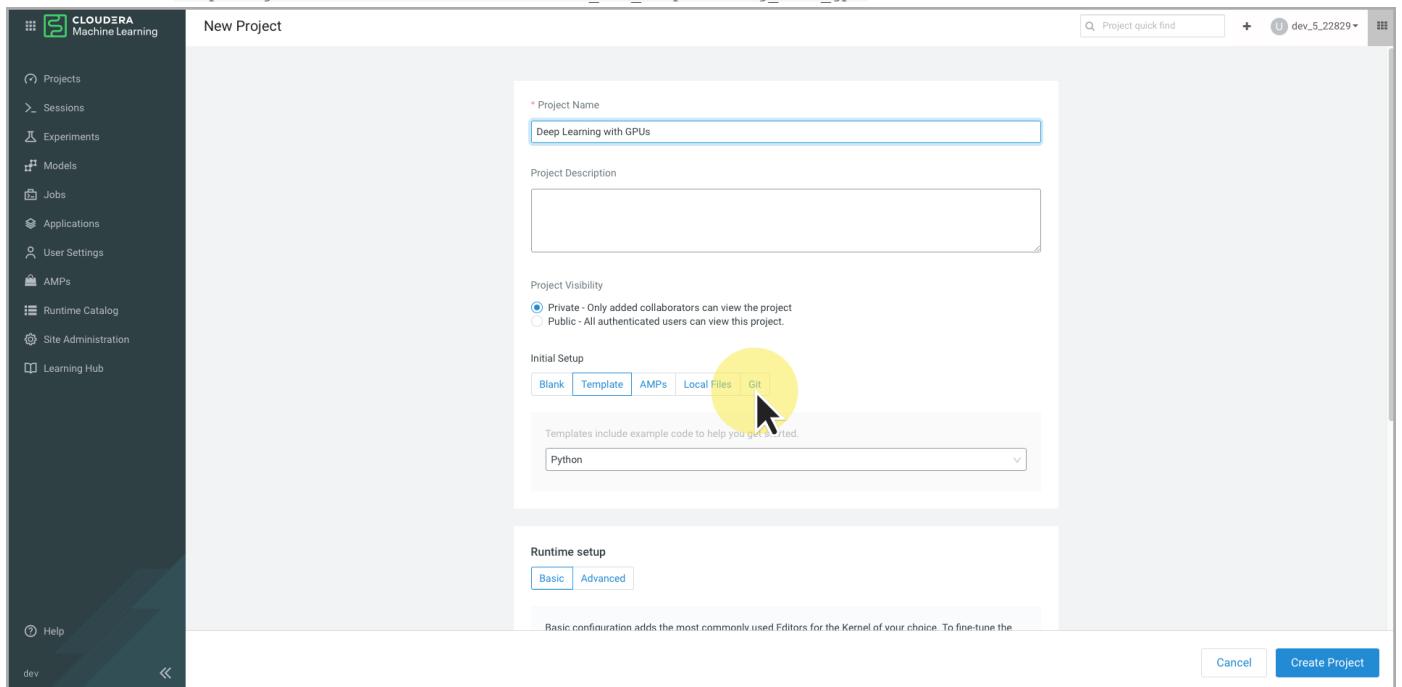
1. Open a new browser tab and navigate to your CML workspace.

2. Click the **New Project** button.

The screenshot shows the 'Projects' page within the CML interface. It lists existing projects like 'CML on CDP'. A large yellow circle highlights the green 'New Project' button in the top right corner of the toolbar. The left sidebar contains navigation links for Sessions, Experiments, Models, Jobs, Applications, User Settings, Runtime Catalog, Site Administration, and Learning Hub. The bottom of the screen shows the workspace name 'cml-on-cdp' and cloud provider information.

3. Enter **Deep Learning with GPUs** for the Name.

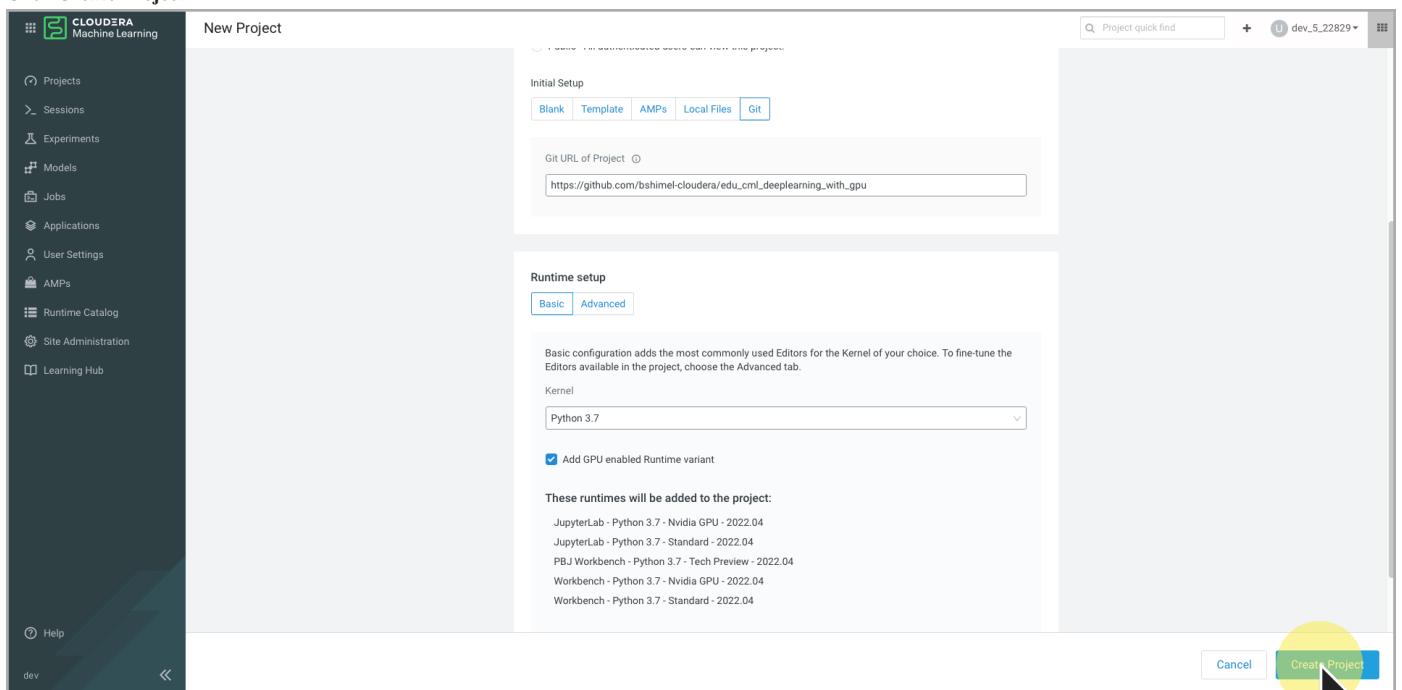
4. Click **Git** and enter https://github.com/bshimel-cloudera/edu_cml_deeplearning_with_gpu for the URL.



5. Select **Python 3.7** as the **Kernel**.

6. Check **Add GPU enabled Runtime variant**. This is extremely important. It tells CML to use the runtime with the GPU drivers.

7. Click **Create Project**



Create a Session without a GPU

1. Click the New Session button.

The screenshot shows the Cloudera Machine Learning interface. On the left, there's a sidebar with various project management options like All Projects, Overview, Sessions, Data, Experiments, Models, Jobs, Applications, Files, Collaborators, and Project Settings. The main workspace is titled 'dev_5_22829 / Deep Learning with GPUs'. It contains sections for Models, Jobs, and Files. Under 'Files', there's a list of files including 'images', 'mxnet', 'pytorch', 'tensorflow', 'LICENSE', and 'README.md'. The 'New Session' button is highlighted with a yellow circle in the top right corner of the workspace area.

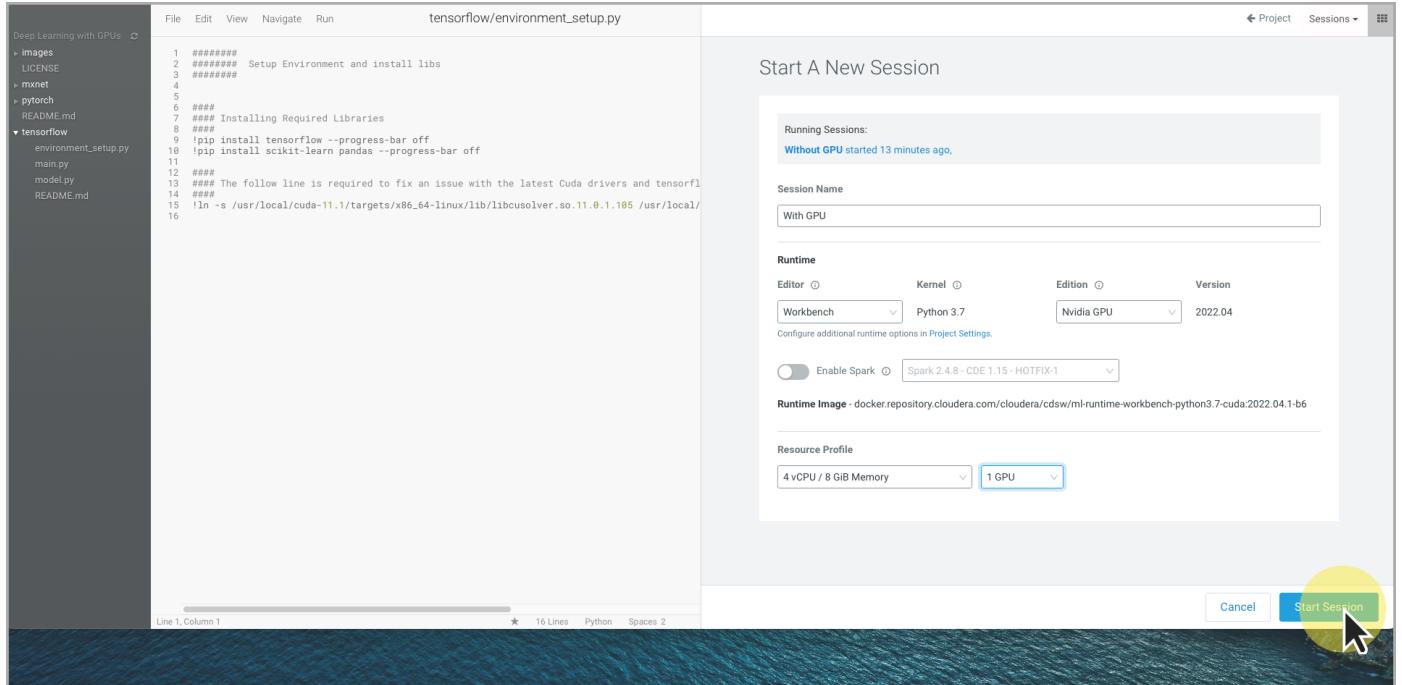
2. Enter **Without GPU** for the Session Name.

3. Under **Runtime**, select **Workbench**, **Nvidia GPU** and verify **Version** is **2023.05**. Even though you will not be using a GPU in this example, you want the runtime with the Nvidia libraries for TensorFlow to work properly.

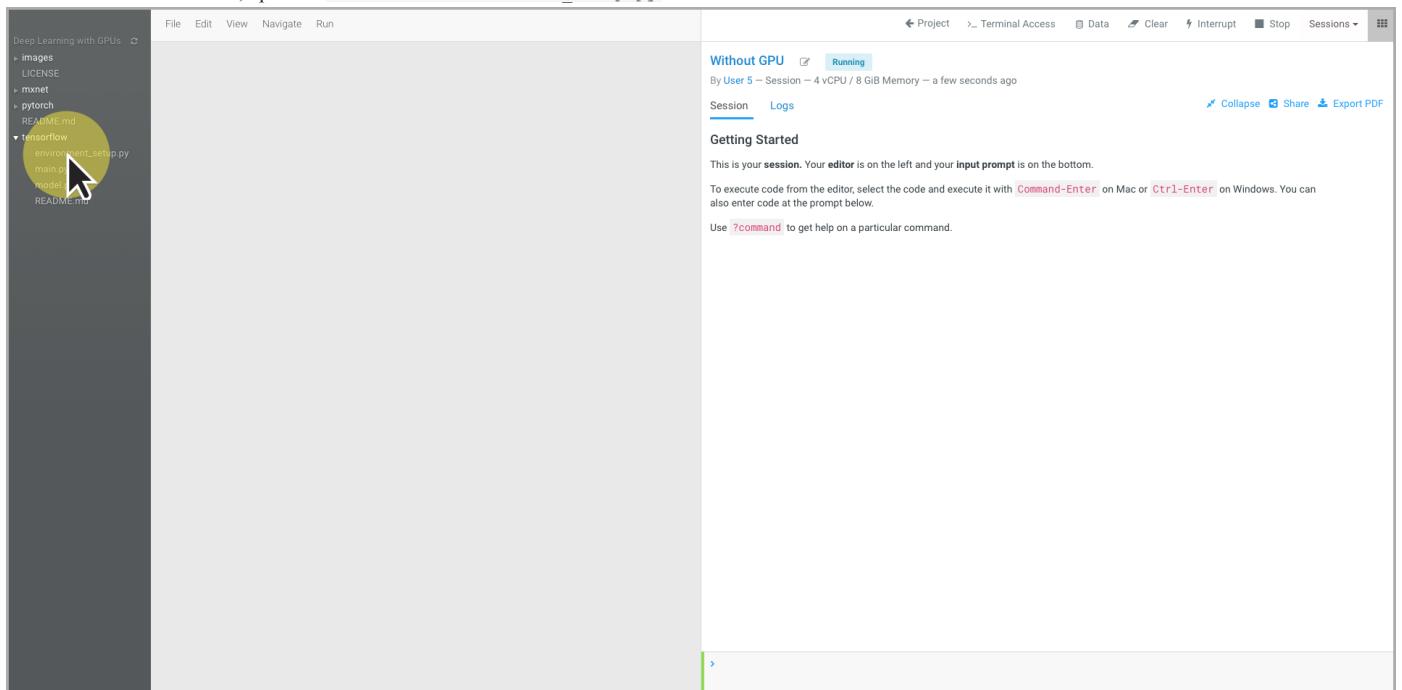
4. Under **Resource Profile**, select **4 vCPU/8 GiB Memory** and **0 GPUs**.

The screenshot shows the 'Start A New Session' dialog box. It has fields for 'Session Name' (set to 'Without GPU'), 'Runtime' (Editor: Workbench, Kernel: Python 3.7, Edition: Nvidia GPU, Version: 2022.04), and 'Resource Profile' (4 vCPU / 8 GiB Memory). A dropdown menu for 'GPUs' is open, showing options from 0 to 6. The '0 GPUs' option is highlighted with a yellow circle. At the bottom right of the dialog are 'Cancel' and 'Start Session' buttons.

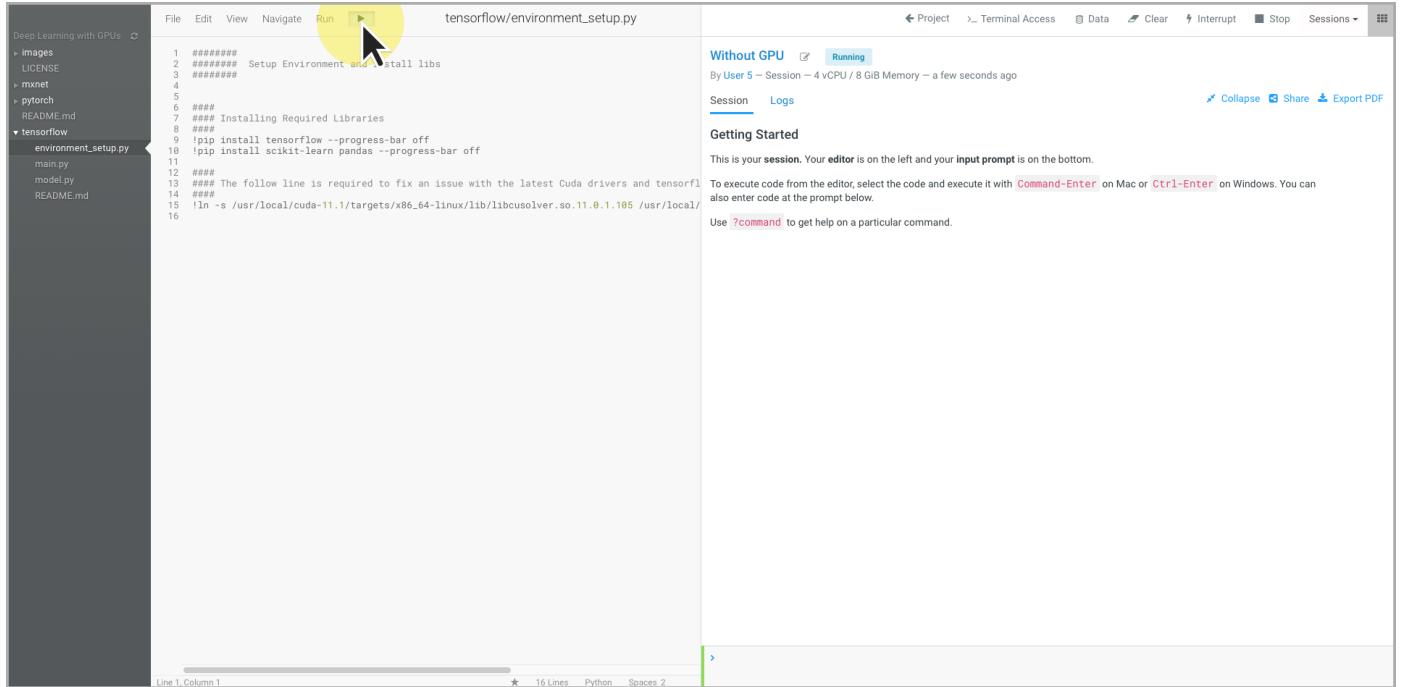
5. Click the **Start Session** button.



6. Once the session has started, open the tensorflow/environment_setup.py file.



7. Click the **Run** button to setup the environment. This step will take about seven minutes.



```

Deep Learning with GPUs
  - images
  - LICENSE
  - mxnet
  - pytorch
  - README.md
  - tensorflow
    - environment_setup.py
      - main.py
      - model.py
      - README.md

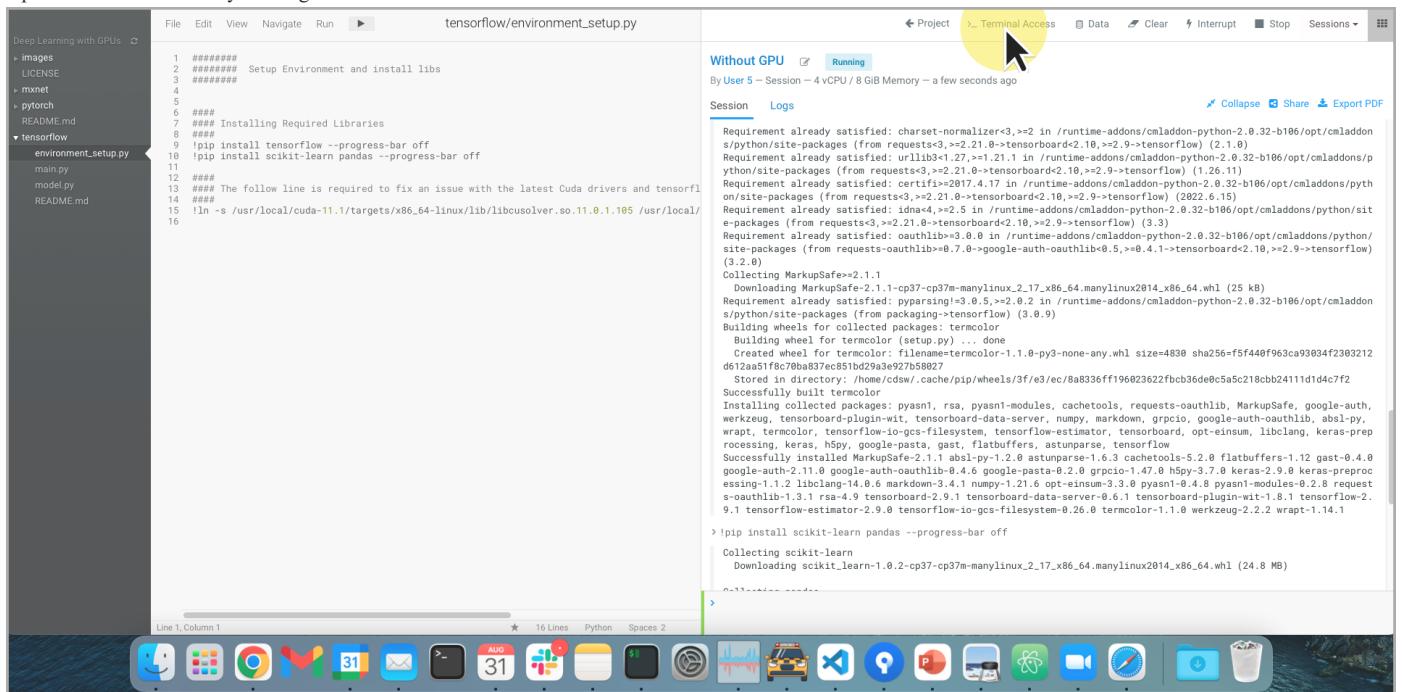
File Edit View Navigate Run tensorflow/environment_setup.py
Without GPU [Running]
By User 5 -- Session -> 4 vCPU / 8 GiB Memory -- a few seconds ago
Session Logs
Getting Started
This is your session. Your editor is on the left and your input prompt is on the bottom.
To execute code from the editor, select the code and execute it with Command-Enter on Mac or Ctrl-Enter on Windows. You can also enter code at the prompt below.
Use ?command to get help on a particular command.

1 ##### Setup Environment and install libs
2 ##### Installing Required Libraries
3 #####
4 #####
5 !pip install tensorflow --progress-bar off
6 #####
7 #####
8 #####
9 !pip install scikit-learn pandas --progress-bar off
10 #####
11 #####
12 #####
13 #####
14 #####
15 !ln -s /usr/local/cuda-11.1/targets/x86_64-linux/lib/libcusolver.so.11.0.1.105 /usr/local/
16 #####

```

Line 1, Column 1 ★ 16 Lines Python Spaces 2

8. Open a terminal window by clicking **Terminal Access**.



```

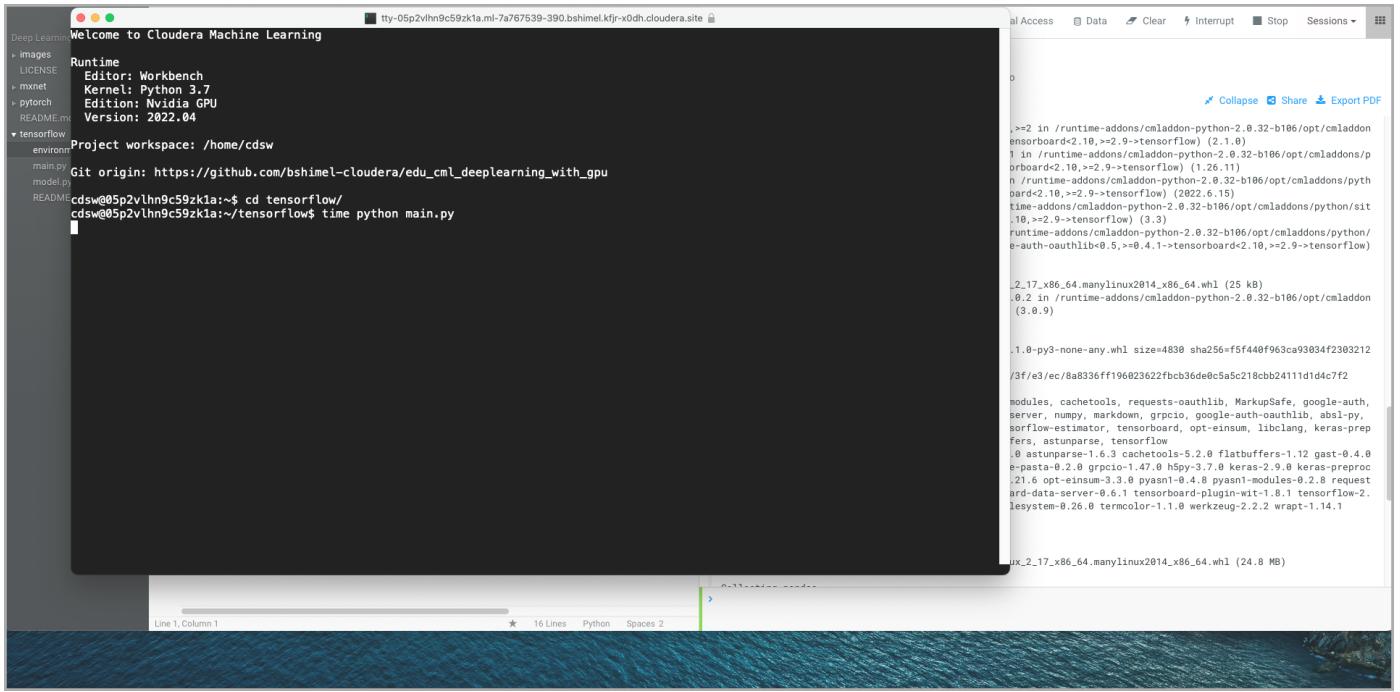
Deep Learning with GPUs
  - images
  - LICENSE
  - mxnet
  - pytorch
  - README.md
  - tensorflow
    - environment_setup.py
      - main.py
      - model.py
      - README.md

File Edit View Navigate Run Without GPU [Running]
By User 5 -- Session -> 4 vCPU / 8 GiB Memory -- a few seconds ago
Session Logs
Requirement already satisfied: charset-normalizer<3,>=2 in /runtime-addons/cmladd-on-python-2.0.32-b106/opt/cmladd-on/python/site-packages (from requests<3,>=2.21.0->tensorflow>2.10,>=2.9->tensorflow) (2.1.0)
Requirement already satisfied: urllib3<2.27,>=1.21.1 in /runtime-addons/cmladd-on-python-2.0.32-b106/opt/cmladd-on/python/site-packages (from requests<3,>=2.21.0->tensorflow>2.10,>=2.9->tensorflow) (2022.16.15)
Requirement already satisfied: certifi>=2017.4.17 in /runtime-addons/cmladd-on-python-2.0.32-b106/opt/cmladd-on/python/site-packages (from requests<3,>=2.21.0->tensorflow>2.10,>=2.9->tensorflow) (1.0.2)
Requirement already satisfied: idna<4,>=2.5 in /runtime-addons/cmladd-on-python-2.0.32-b106/opt/cmladd-on/python/site-packages (from requests<3,>=2.21.0->tensorflow>2.10,>=2.9->tensorflow) (3.3)
Requirement already satisfied: oauthlib<=3.0.0 in /runtime-addons/cmladd-on-python-2.0.32-b106/opt/cmladd-on/python/site-packages (from requests<3,>=2.21.0->tensorflow>2.10,>=2.9->tensorflow) (3.2.0)
Collecting MarkupSafe==2.1.1
  Downloading MarkupSafe-2.1.1-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Requirement already satisfied: parsimonious<3.0.5,>=2.0.2 in /runtime-addons/cmladd-on-python-2.0.32-b106/opt/cmladd-on/python/site-packages (from packaging->tensorflow) (3.0.9)
Building wheels for collected packages: termcolor
  Building wheel for termcolor: fullname=termcolor-1.1.0-py3-none-any.whl size=4830 sha256=f5f440f963ca93034f2380212d512aa51f8c709a837ec851bd293e9e927058627
  Stored in directory: /home/cdwaw/.cache/pip/wheels/3f/e3/ec/8a8336ff196023622fbc36de0c5a5c218ccb24111d1d4c7f2
Successfully built termcolor
Installing collected packages: pyasn1, rsa, pyasn1-modules, cachetools, requests-oauthlib, MarkupSafe, google-auth, werkzeug, tensorflow-plugin-wit, tensorflow-data-server, numpy, markdown, grpcio, google-auth-oauthlib, abslib-py, wrapt, termcolor, tensorflow-io-gcs-filesystem, tensorflow-estimator, tensorflow, opt-einsum, libclang, keras-preprocessing, keras, h5py, google-pasta, gast, flatbuffers, astunparse, tensorflow
Successfully installed MarkupSafe-2.1.1-absl-py-1.2.0.astunparse-1.6.3.cachetools-5.2.0.flatbuffers-1.12.gast-0.4.8.google-auth-2.11.0.google-auth-oauthlib-0.4.6.google-pasta-0.2.0.grpcio-1.47.0.h5py-3.7.0.keras-2.9.0.keras-preprocessing-1.1.2.libclang-14.0.6.markdown-3.4.1.numpy-1.21.1.opt-einsum-3.3.0.pyasn1-0.4.8.pyasn1-modules-0.2.8.request-s-0.10.1.rsa-4.9.tensorboard-2.9.1.tensorboard-data-server-0.6.1.tensorboard-plugin-wit-1.8.1.tensorflow-2.9.1.tensorflow-estimator-2.9.0.tensorflow-io-gcs-filesystem-0.26.0.termcolor-1.1.0.werkzeug-2.2.2.wrapt-1.14.1.
> !pip install scikit-learn pandas --progress-bar off
Collecting scikit-learn
  Downloading scikit_learn-1.0.2-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (24.8 MB)

```

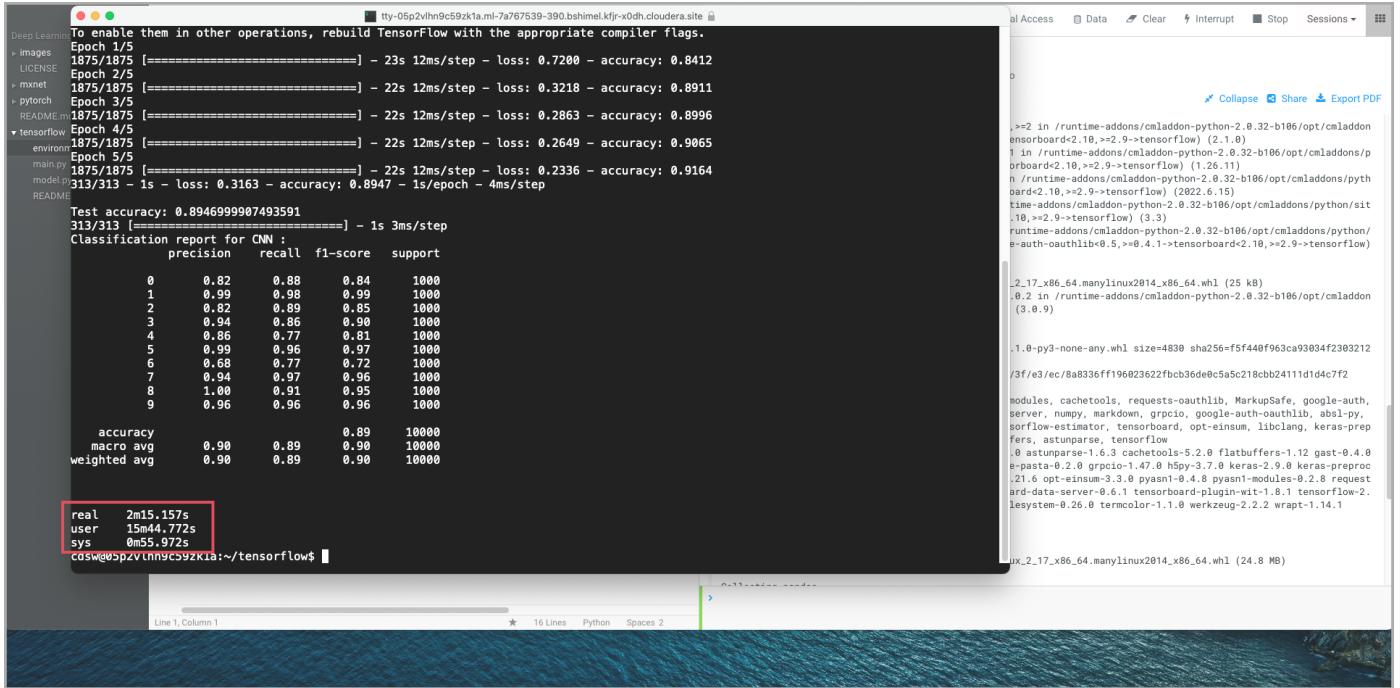
Line 1, Column 1 ★ 16 Lines Python Spaces 2

9. Enter `cd tensorflow` to change the tensorflow directory. Then, enter `time python main.py`. This will run the image classification program and record the time to run the program.



```
tty-05p2vlhn9c59zk1a.m1-7a767539-390.bshimel.kfr-x0dh.cloudera.site ~
Welcome to Cloudera Machine Learning
Deep Learning
  - images
  - LICENSE
  - mxnet
  - pytorch
  - README.md
  - tensorflow
    - environment
      - main.py
      - model.py
  - README
Project workspace: /home/cds
Git origin: https://github.com/bshimel-cloudera/edu_cml_deeplearning_with_gpu
cdsw@05p2vlhn9c59zk1a:~$ cd tensorflow/
cdsw@05p2vlhn9c59zk1a:~/tensorflow$ time python main.py
[...]
2.9s user 0m55.972s
2.9s sys 0m0.000s
2.9s total 0m55.972s
cdsw@05p2vlhn9c59zk1a:~/tensorflow$
```

10. When the program finishes, note the real, user, and sys time required to run the program.



```
tty-05p2vlhn9c59zk1a.m1-7a767539-390.bshimel.kfr-x0dh.cloudera.site ~
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Deep Learning
  - images
  - LICENSE
  - mxnet
  - pytorch
  - README.md
  - tensorflow
    - environment
      - main.py
      - model.py
  - README
Epoch 1/5
1875/1875 [=====] - 23s 12ms/step - loss: 0.7200 - accuracy: 0.8412
Epoch 2/5
1875/1875 [=====] - 22s 12ms/step - loss: 0.3218 - accuracy: 0.8911
Epoch 3/5
1875/1875 [=====] - 22s 12ms/step - loss: 0.2863 - accuracy: 0.8996
Epoch 4/5
1875/1875 [=====] - 22s 12ms/step - loss: 0.2649 - accuracy: 0.9065
Epoch 5/5
1875/1875 [=====] - 22s 12ms/step - loss: 0.2336 - accuracy: 0.9164
313/313 - 1s - loss: 0.3163 - accuracy: 0.8947 - 1s/epoch - 4ms/step
Test accuracy: 0.8946999907493591
313/313 [=====] - 1s 3ms/step
Classification report for CNN :
precision    recall    f1-score   support
          0       0.82      0.88      0.84     1000
          1       0.99      0.98      0.99     1000
          2       0.82      0.89      0.85     1000
          3       0.94      0.86      0.90     1000
          4       0.86      0.77      0.81     1000
          5       0.99      0.96      0.97     1000
          6       0.68      0.77      0.72     1000
          7       0.94      0.97      0.96     1000
          8       1.00      0.91      0.95     1000
          9       0.96      0.96      0.96     1000

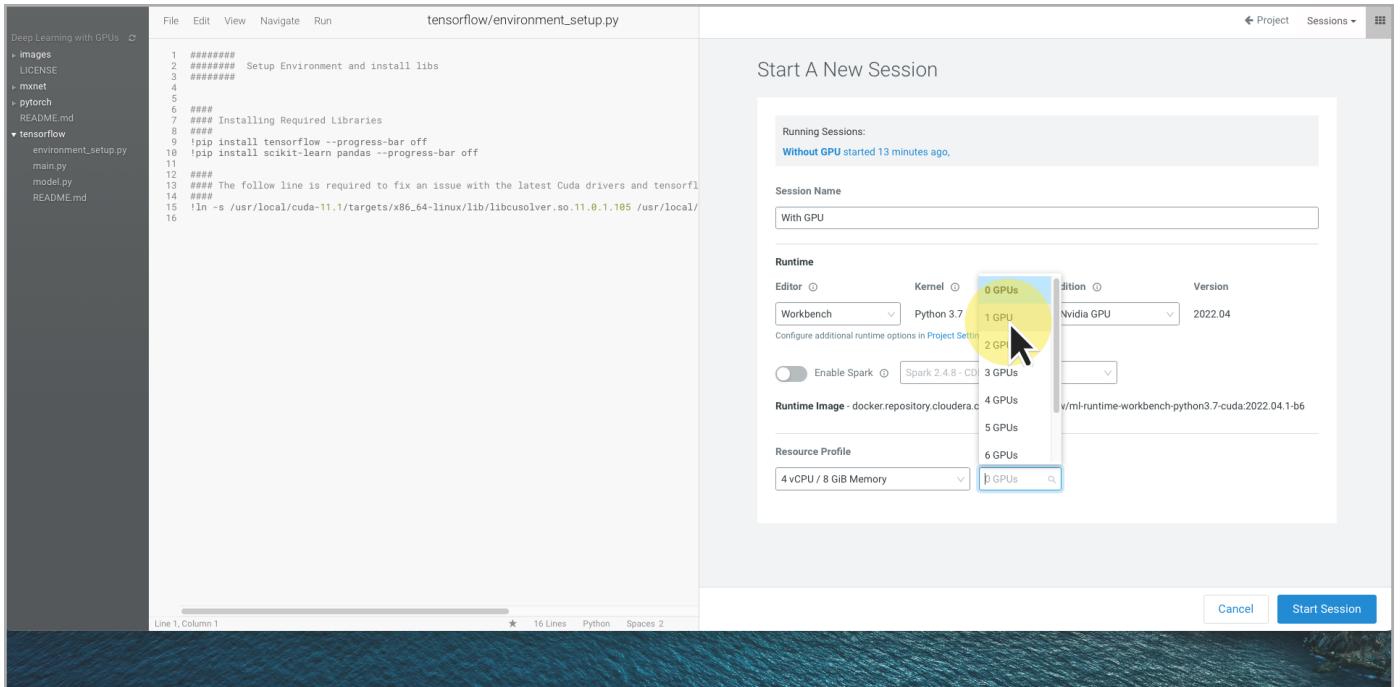
accuracy         0.89      10000
macro avg       0.90      0.89      0.90     10000
weighted avg    0.90      0.89      0.90     10000

real    2m15.157s
user    1m44.772s
sys     0m55.972s
cdsw@05p2vlhn9c59zk1a:~/tensorflow$
```

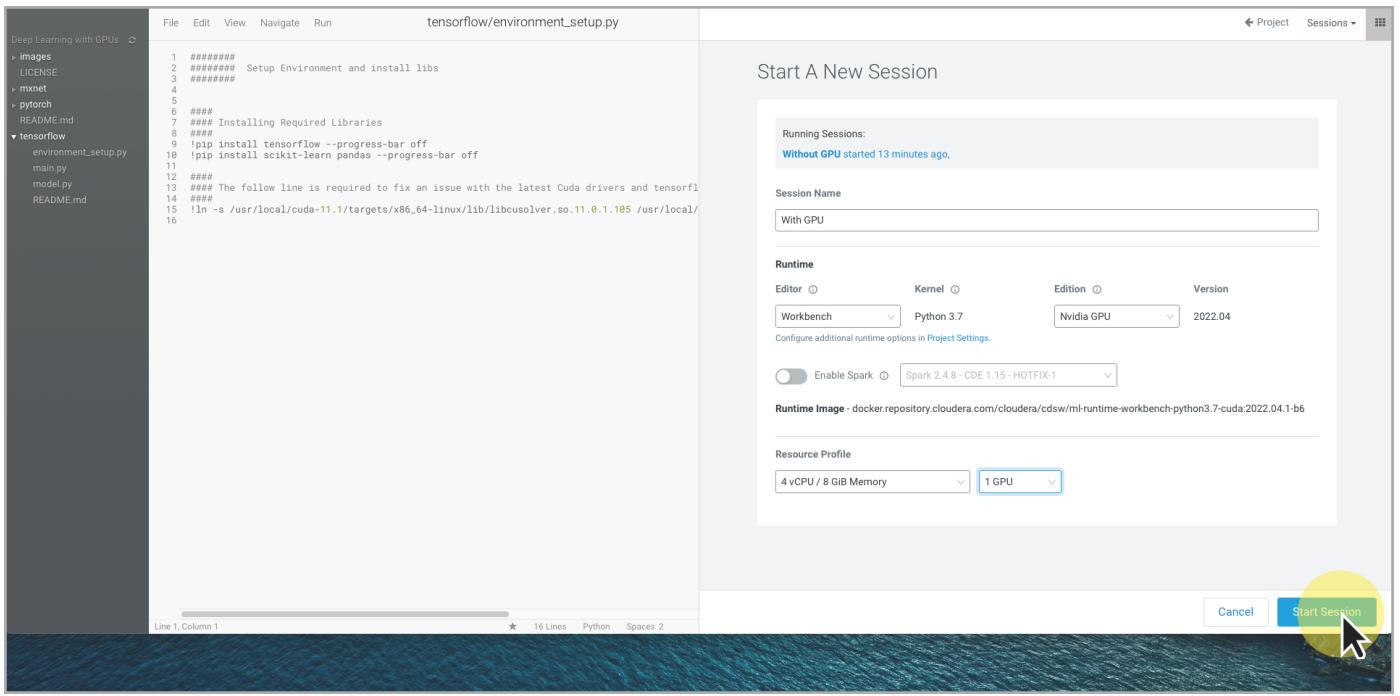
Create a Session with a GPU

1. Close the terminal and go back to the project.
2. Create a new session titled `With GPU`.

3. Select **4 vCPU/8 GiB Memory** and **1 GPU**. It might be tempting to select more than one GPU, but do not. The example is not configured to take advantage of more than one GPU.



4. Click the **Start Session** button.



It is hard to predict what will be seen in a shared environment. If a CML GPU Worker node is not available, you might see a message about *Insufficient nvidia resources or pod taint*.

```

Deep Learning with GPUs
File Edit View Navigate Run tensorflow/environment_setup.py
With GPU Scheduling
By User 5 – Session – 4 vCPU / 8 GiB Memory – 1 GPU – a few seconds ago
Session Logs
Logs
Getting Started
This is your session. Your editor is on the left and your input prompt is on the bottom.
To execute code from the editor, select the code and execute it with Command-Enter on Mac or Ctrl-Enter on Windows. You can also enter code at the prompt below.
Use ?command to get help on a particular command.
Unschedulable: 0/5 nodes are available: 1 Insufficient nvidia.com/gpu, 2 node(s) had taint (role.node.kubernetes.io/infra: true), that the pod didn't tolerate, 2 node(s) had taint (role.node.kubernetes.io/little-infra: true), that the pod didn't tolerate.

```

Line 1, Column 1 ★ 16 Lines Python Spaces 2

If you navigate back to Workspace Details in your other browser tab, you might see the CML GPU Worker node count still at zero.

Name	Instance Type	CPU	GPU	Memory	Count	Autoscale Range Min - Max
CML CPU Workers	m5.4xlarge	16	-	64 GiB	1	1 - 10
CML GPU Workers	p2.8xlarge	32	8	488 GiB	0	0 - 10
CML Infra	m5.2xlarge	8	-	32 GiB	2	2 - 3
Platform Infra	m5.large	2	-	8 GiB	2	2 - 4

[Delete GPU](#)

Subnets for Worker Nodes

Subnet Id	Subnet Name	Availability Zone	CIDR
-----------	-------------	-------------------	------

If this the case, the auto scaling will take over and start up a CML GPU Worker node. You will see a message like the following.

The screenshot shows a Jupyter Notebook interface. On the left, a sidebar lists project files: images, LICENSE, mxnet, pytorch, README.md, and tensorflow. The main area displays a code cell with the following Python script:

```

1 ##### Setup Environment and install libs
2 #####
3 #####
4 #####
5 #####
6 #### Installing Required Libraries
7 #####
8 #####
9 !pip install tensorflow --progress-bar off
10 !pip install scikit-learn pandas --progress-bar off
11 #####
12 #####
13 #### The follow line is required to fix an issue with the latest Cuda drivers and tensorflow
14 #####
15 !ln -s /usr/local/cuda-11.1/targets/x86_64-linux/lib/libcud solver.so.11.0.1.105 /usr/local/
16 #####

```

The right panel shows the session status: "With GPU" and "Scheduling". It indicates "By User 5 – Session – 4 vCPU / 8 GiB Memory – 1 GPU – a few seconds ago". The "Logs" tab is selected, showing the output of the command `!ln -s /usr/local/cuda-11.1/targets/x86_64-linux/lib/libcud solver.so.11.0.1.105 /usr/local/`. A message at the bottom states: "Auto scaling in progress: pod triggered scale-up: [[lifte-zk2qdr2-mlgpu1-NodeGroup 0->1 (max: 10)]]". The bottom status bar shows "Line 1, Column 1", "16 Lines", "Python", and "Spaces 2".

Once the auto scaling does its magic, the Workspace Details will show that there is a new CML GPU Worker node.

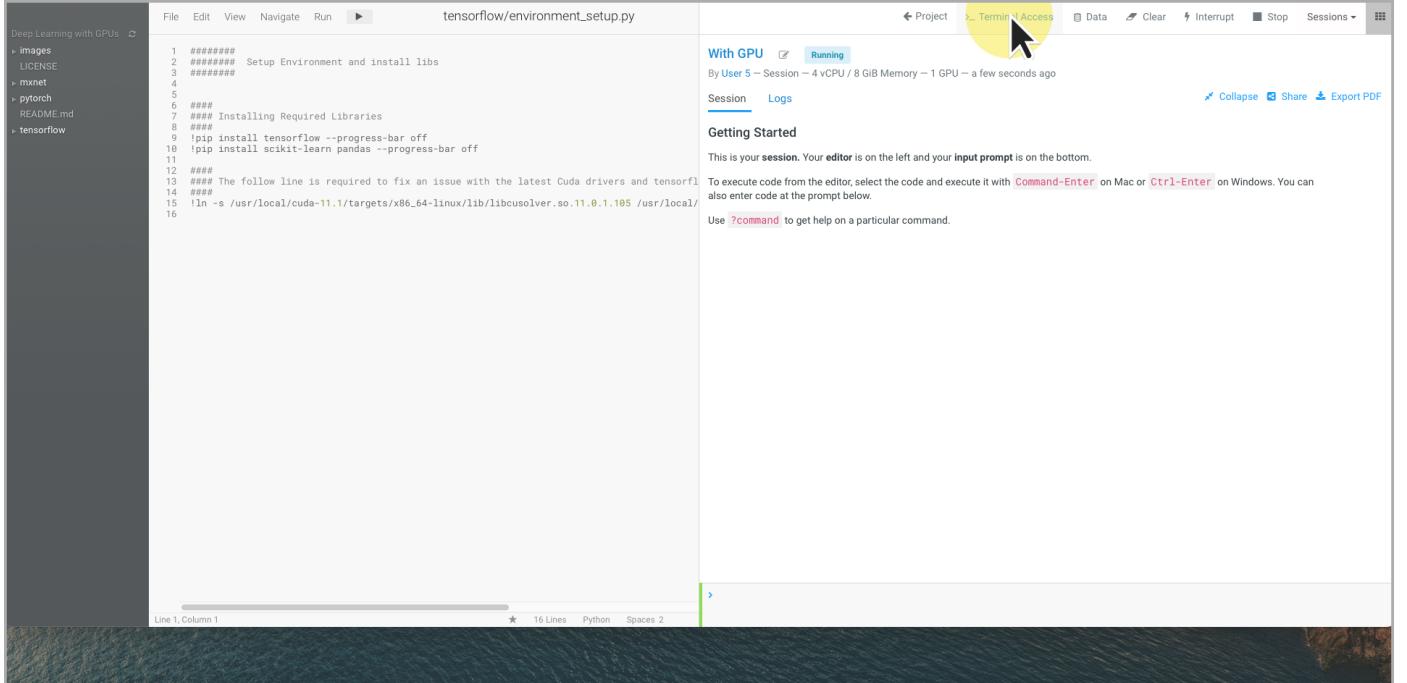
The screenshot shows the "Machine Learning Workspaces / cml-on-cdp" page. The left sidebar includes links for Help, User 5, Workspaces, and Workspace Backups. The main content area is divided into sections:

- Machine Learning Workspaces / cml-on-cdp**: Displays workspace metadata such as Cloudera-Environment-Resource-Name, Cloudera-Resource-Name, ExperienceType (cml), Owner (adm_bshimel@cloudera.com), TenantID (29a489d3-e022-407a-a1f9-1c4901266d0f), WorkspaceCrn (crn:cdrml:us-west-1:29a489d3-e022-407a-a1f9-1c4901266d0f:workspace:fbc44900-890b-4aaa-805e-1b3b71982ec4), and WorkspaceName (cml-on-cdp).
- Workspace Instances**: A table listing four instances:

Name	Instance Type	CPU	GPU	Memory	Count	Autoscale Range Min - Max
CML CPU Workers	m5.4xlarge	16	-	64 GiB	1	1 - 10
CML GPU Workers	p2.8xlarge	32	8	488 GiB	1	0 - 10
CML Infra	m5.2xlarge	8	-	32 GiB	2	2 - 3
Platform Infra	m5.large	2	-	8 GiB	2	2 - 4
- Delete GPU** button (disabled).
- Subnets for Worker Nodes**: A table with columns: Subnet Id, Subnet Name, Availability Zone, and CIDR.

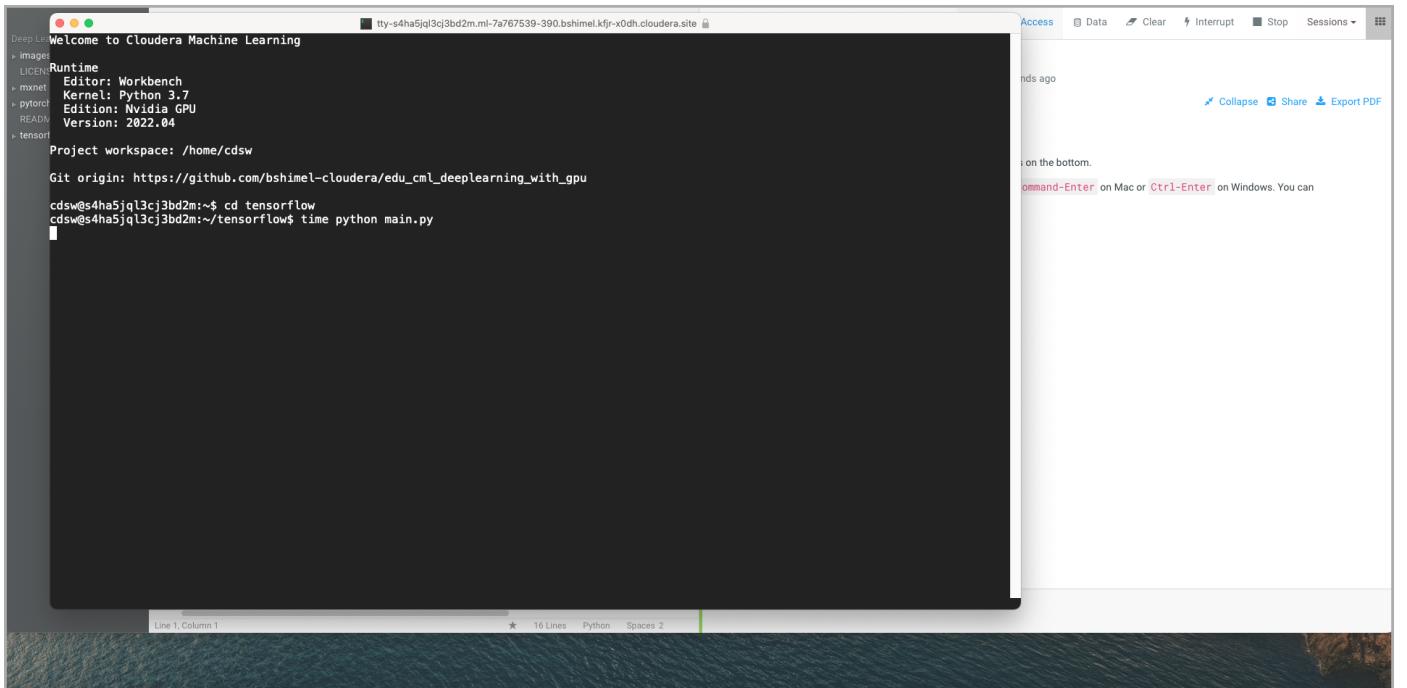
The whole process of auto scaling and adding a new worker node can take awhile (like ten minutes). Be patient, it is worth the wait.

1. Once the new session has started, click **Terminal Access** to open a new terminal.



The screenshot shows the Cloudera Machine Learning interface. On the left, there's a sidebar with project files like 'Deep Learning with GPUs', 'LICENSE', 'mxnet', 'pytorch', 'README.md', and 'tensorflow'. The main area has a code editor with a Python script named 'tensorflow/environment_setup.py'. To the right of the code editor is a terminal window titled 'With GPU' which is 'Running'. The terminal shows some setup code and a note about fixing CUDA drivers. Below the terminal, there's a 'Getting Started' section with instructions for executing code. The 'Terminal Access' button is highlighted with a yellow circle.

2. Enter `cd tensorflow`, then enter `time python main.py`.



This screenshot shows the terminal window from the previous step. The terminal has a black background and displays the command 'cd tensorflow' followed by 'time python main.py'. The rest of the screen shows the Cloudera Machine Learning environment, including the sidebar with project files and the code editor window.

3. Compare the new runtime using the GPU with your previous runtime without the GPU. The real runtime with a GPU should be about half of the real runtime without a GPU. Discuss the results with your instructor and classmates. Why is the user runtime so much greater for the non-GPU than the real runtime?

The terminal window displays the following output:

```

Deep Learning          [ 2022-08-31 23:03:16.604746: W tensorflow/stream_executor/gpu/redzone_allocator.cc:314] INTERNAL: Failed to launch ptxas
Relying on driver to perform ptx compilation.
LIVENESS: Modify $PATH to customize ptxas location.
mininet: This message will be only logged once.
pytorch: Epoch 2/5
READER: 1875/1875 [=====] - 9s 5ms/step - loss: 0.3274 - accuracy: 0.8883
tensorboard: Epoch 3/5
1875/1875 [=====] - 9s 5ms/step - loss: 0.2843 - accuracy: 0.9003
Epoch 4/5
1875/1875 [=====] - 9s 5ms/step - loss: 0.2603 - accuracy: 0.9073
Epoch 5/5
1875/1875 [=====] - 9s 5ms/step - loss: 0.2347 - accuracy: 0.9155
313/313 - 1s - loss: 0.2842 - accuracy: 0.8968 - 1s/epoch - 4ms/step

Test accuracy: 0.8967999815940857
313/313 [=====] - 1s 2ms/step
Classification report for CNN :
precision    recall   f1-score   support
      0       0.87      0.84      0.86     1000
      1       0.99      0.98      0.99     1000
      2       0.87      0.85      0.86     1000
      3       0.92      0.98      0.91     1000
      4       0.89      0.71      0.79     1000
      5       0.96      0.98      0.97     1000
      6       0.64      0.88      0.71     1000
      7       0.57      0.94      0.95     1000
      8       0.97      0.98      0.98     1000
      9       0.97      0.96      0.96     1000

accuracy           0.90
macro avg       0.90      0.90      0.90     10000
weighted avg    0.90      0.90      0.90     10000

real    1m41.875s
user    1m3.044s
sys     0m13.667s
cpu@cpu:~$
```

The system metrics at the bottom of the terminal are highlighted in a red box:

real	1m41.875s
user	1m3.044s
sys	0m13.667s

End of Exercise

Continuous Model Monitoring with Evidently

Monitoring a deployed machine learning model is critical to ensuring model performance over time. CML's **model metrics** combined with **Evidently**, an open source tool for evaluating, testing, and monitoring machine learning models, is a powerful tool to combat model drift.

In this exercise, you will:

- Start the Continuous Model Monitoring AMP
- Launch the Price Regressor Monitoring dashboard
- Explore drift and variations in the model performance
- Identify the file that creates the Evidently dashboard
- Experiment with Evidently reports



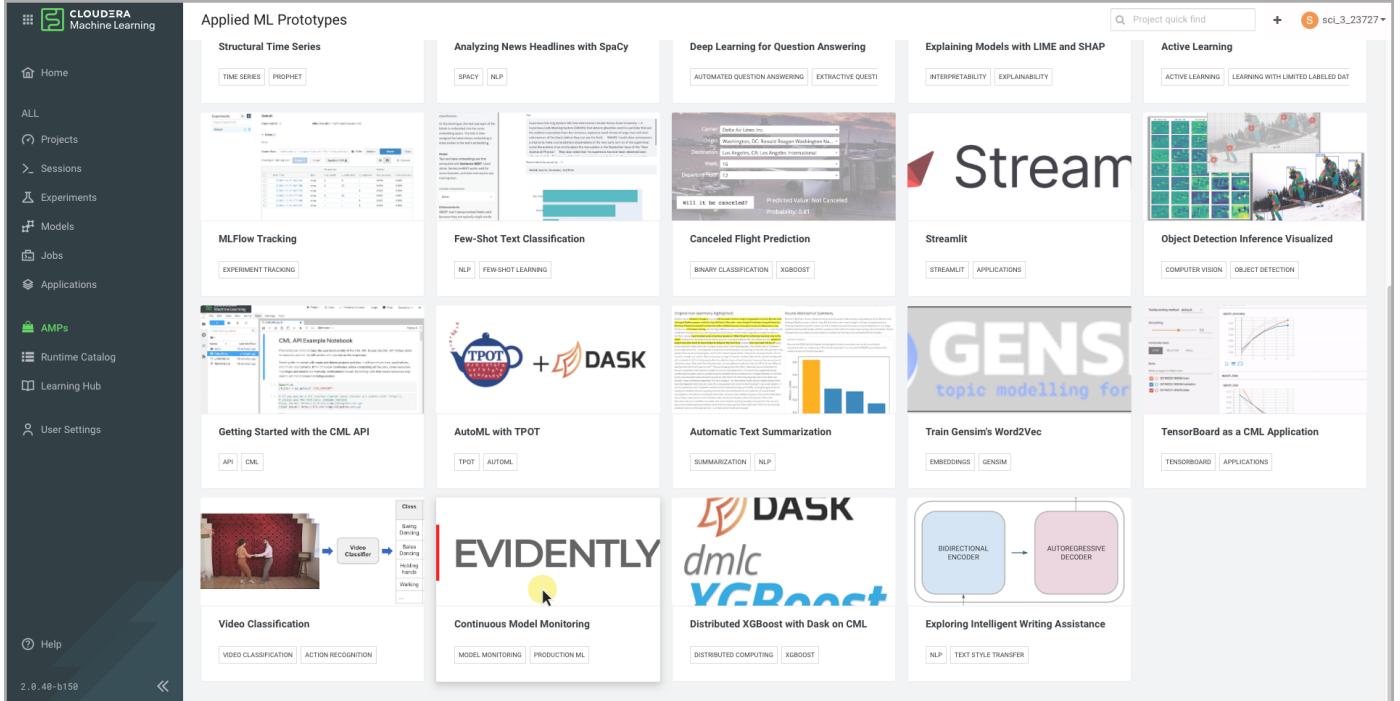
The initial deployment of the AMP takes approximately 30 minutes.

Start the Continuous Model Monitoring AMP

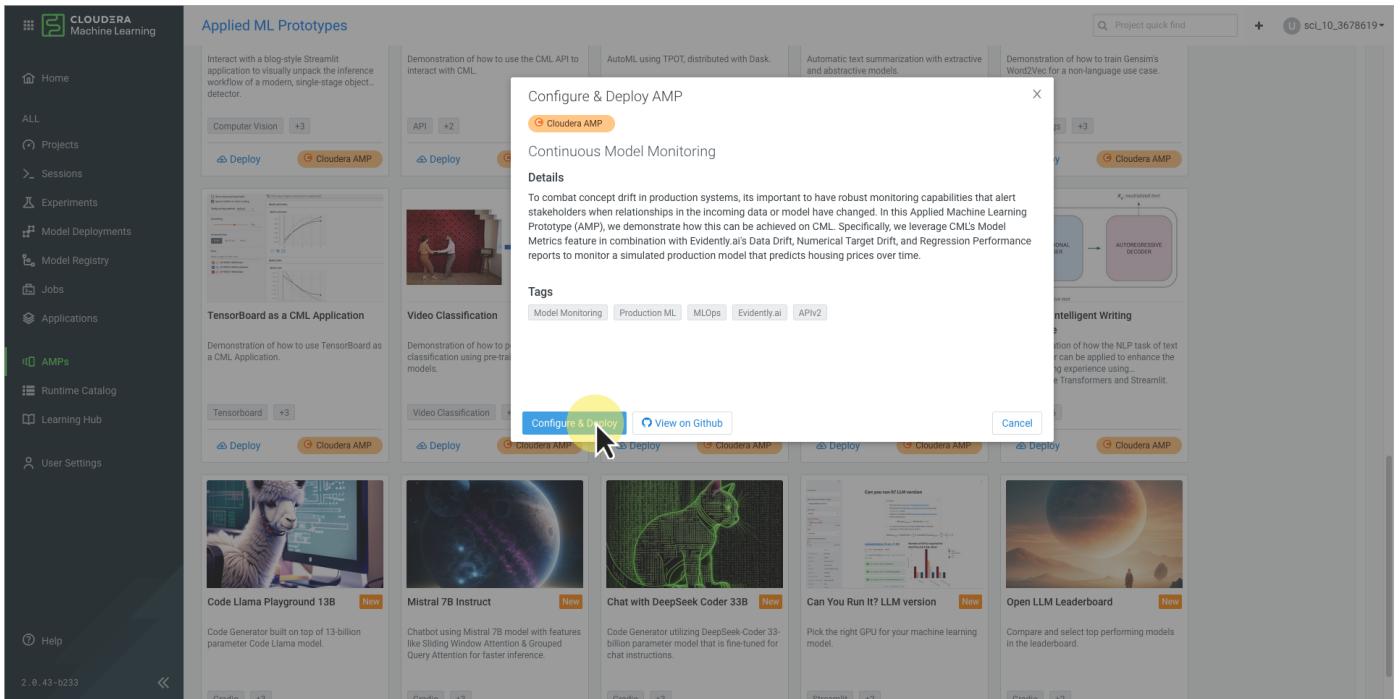
1. Click AMPs

The screenshot shows the Cloudera Machine Learning (CML) interface. The sidebar on the left is dark blue with white icons and text. The 'AMPs' icon is highlighted with a yellow circle and a cursor arrow pointing to it. The main content area is light gray. At the top, it says 'Welcome to CML, sci_3_23727.' Below that are four cards: 'Create a new project', 'Deploy a prototype', 'Create a notebook', and 'Visualize your data'. Under 'Recent Projects', there is a message: 'You currently don't have any projects. Select an option above to get started.' Below this are three cards: 'Take a CML product tour', 'Explore Use Cases', and 'Enable exploratory Data Science'. To the right, under 'Featured Announcements', are three news items: 'New AMP - LLM Chatbot Augmented with Enterprise Data', 'New "Add Data" feature to simplifies data ingestion', and 'Simplify Data Access with Custom Connection Support in CML'. At the bottom of the main area, it says 'Workspace: cml-on-cdp-heinz' and 'Cloud Provider: aws (AWS)'.

2. Click Continuous Model Monitoring AMP



3. Click Configure & Deploy



4. Set Dev Mode to True

Configure Project: Continuous Model Monitoring - sci_3_23727

AMP Name: Continuous Model Monitoring (v2)

Demonstration of how to perform continuous model monitoring on CML using Model Metrics and Evidently.ai dashboards

Environment Variables

The settings below were defined by the AMP:

Name	Value	Description
* DEV_MODE	false	Flag to indicate if the AMP should run on a 5% sample of the dataset (True) to facilitate efficient project development or the full dataset (False).

Runtime

Editor: Workbench Kernel: Python 3.9 Edition: Standard Version: 2023.05

Enable Spark: Spark 3.2.3 - CDE 1.19.2 - HOTFIX-2

Runtime Image
- docker.repository.cloudera.com/cloudera/cdsw/ml-runtime-workbench-python3.9-standard:2023.05.2-b7

No Runtime Addon is required for this AMP.

Setup Steps

Execute AMP setup steps

Buttons

Cancel Launch Project



You must select Python 3.7 or the AMP will fail.

1. Select Python 3.7

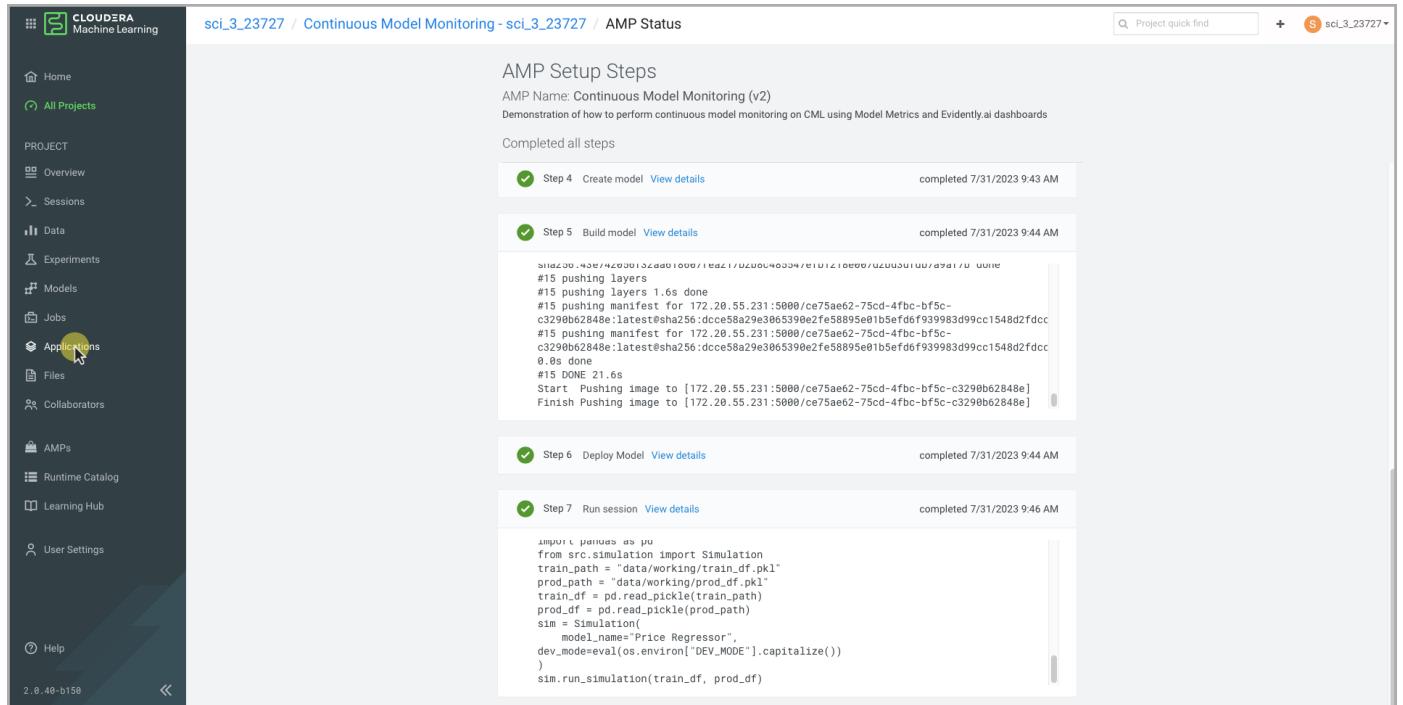
The screenshot shows the "Configure Project" dialog for "Continuous Model Monitoring - sci_3_23727". In the "Runtime" section, the "Kernel" dropdown is set to "Python 3.9". A dropdown menu is open over the "Python 3.9" option, showing "Python 3.7" highlighted with a yellow circle. Other options in the menu include "Python 3.8", "Python 3.9" (which is currently selected), "R 3.6", "R 4.0", and "R 4.1". The "Version" dropdown is set to "2023.05". The "Setup Steps" section has a checked checkbox for "Execute AMP setup steps". At the bottom right are "Cancel" and "Launch Project" buttons.

2. Click Launch Project

The screenshot shows the same "Configure Project" dialog after the user has selected "Python 3.7" from the dropdown. The "Kernel" dropdown now shows "Python 3.7". The "Setup Steps" section still has the "Execute AMP setup steps" checkbox checked. The "Launch Project" button at the bottom right is highlighted with a yellow circle. The rest of the interface remains the same as the previous screenshot.

Launch the Price Regressor Monitoring dashboard

1. Click Applications in the left-side menu



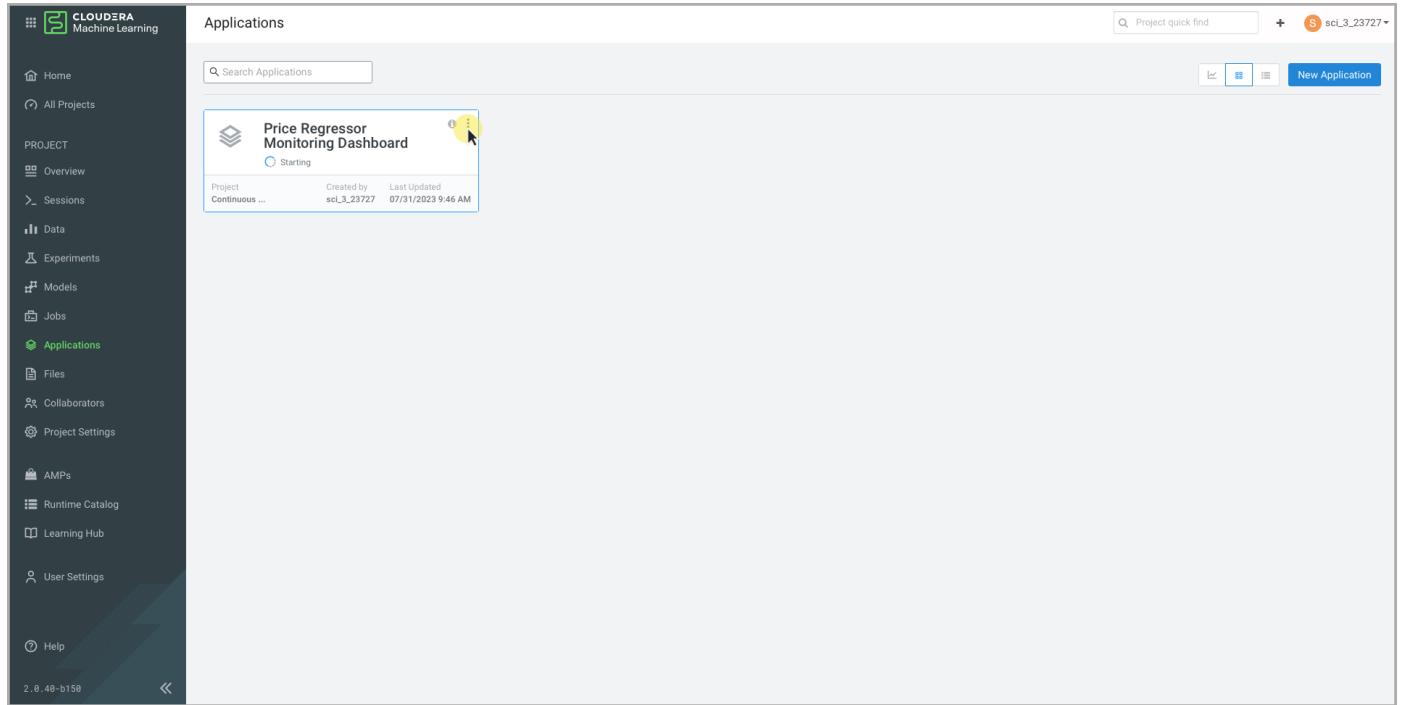
The screenshot shows the Cloudera Machine Learning interface. On the left, the navigation sidebar is visible with various options like Home, All Projects, PROJECT, Data, Experiments, Models, Jobs, Applications (which is highlighted with a yellow circle), Files, Collaborators, AMPs, Runtime Catalog, Learning Hub, User Settings, and Help. The version is listed as 2.0.40-b150. The main content area is titled "sci_3_23727 / Continuous Model Monitoring - sci_3_23727 / AMP Status". It displays the "AMP Setup Steps" with seven completed steps:

- Step 4: Create model (View details) completed 7/31/2023 9:43 AM
- Step 5: Build model (View details) completed 7/31/2023 9:44 AM
- Step 6: Deploy Model (View details) completed 7/31/2023 9:44 AM
- Step 7: Run session (View details) completed 7/31/2023 9:46 AM

The logs for Step 5 show the command to push the image to Docker Hub:

```
#15 pushing layers
#15 pushing layers 1.6s done
#15 pushing manifest for 172.20.55.231:5000/ce75ae62-75cd-4fbc-bf5c-c3290b62848e:latest@sha256:dce58a29e3065390e2fe58895e01b5efdf939983d99cc1548d2fdcc
#15 pushing manifest for 172.20.55.231:5000/ce75ae62-75cd-4fbc-bf5c-c3290b62848e:latest@sha256:dcce58a29e3065390e2fe58895e01b5efdf939983d99cc1548d2fdcc
0.0s done
#15 DONE 21.6s
Start Pushing image to [172.20.55.231:5000/ce75ae62-75cd-4fbc-bf5c-c3290b62848e]
Finish Pushing image to [172.20.55.231:5000/ce75ae62-75cd-4fbc-bf5c-c3290b62848e]
```

2. Click the application : menu



The screenshot shows the Cloudera Machine Learning interface. The left sidebar is identical to the previous screenshot. The main content area is titled "Applications". A single application card is visible for the "Price Regressor Monitoring Dashboard". The card includes the application name, a status indicator (Starting), and details: Project Continuous ..., Created by sci_3_23727, and Last Updated 07/31/2023 9:46 AM. A yellow circle highlights the three-dot menu icon next to the application name.

3. Click Application Details

The screenshot shows the Cloudera Machine Learning interface. On the left is a dark sidebar with various navigation options like Home, All Projects, Data, Experiments, Models, Jobs, Applications, Files, Collaborators, Project Settings, AMPs, Runtime Catalog, Learning Hub, User Settings, and Help. The Applications section is currently selected. In the main area, there's a search bar and a table of applications. One application, 'Price Regressor Monitoring Dashboard', is highlighted. A context menu is open over this application, with the 'Application Details' option highlighted by a yellow circle.

4. Click Settings

The screenshot shows the 'Price Regressor Monitoring Dashboard' settings page. The navigation bar at the top includes 'Overview', 'Logs', and 'Settings', with 'Settings' being the active tab. The main content area displays details about the application, including its script (app.py), description (An Evidently.ai dashboard for monitoring data drift, target drift, and regression performance.), and creation information (Created by sci_3_23727, Most Recent Start/Restart by sci_3_23727, Ran: 6 times). At the bottom, it shows the workspace (cmi-on-cdp-heinz) and cloud provider (aws (AWS)).

5. Change Kernel to Python 3.7

The screenshot shows the 'Price Regressor Monitoring Dashboard' settings page. In the 'Runtime' section, the 'Kernel' dropdown is set to 'Python 3.9'. A modal window is open, listing several kernel options: Python 3.9 (highlighted in blue), Python 3.8, Python 3.7 (highlighted with a yellow circle), R 3.6, R 4.0, and R 4.1.

6. Click Update Application button

The screenshot shows the 'Price Regressor Monitoring Dashboard' settings page. At the bottom left, the 'Update Application' button is highlighted with a yellow circle.

7. Click Applications in the left-side menu

The screenshot shows the 'Price Regressor Monitoring Dashboard' configuration page. The sidebar on the left has 'Applications' selected. The main area contains fields for 'Name' (Price Regressor Monitoring Dashboard), 'Subdomain' (tenkpn), 'Description' (An Evidently.ai dashboard for monitoring data drift, target drift, and regression performance.), 'Script' (apps/app.py), and runtime options like 'Editor' (Workbench) and 'Kernel' (Python 3.7). A tooltip indicates 'Configure additional runtime options in Project Settings.' At the top right, there are buttons for 'Starting' (highlighted), 'Stop', and 'Restart'.

8. Click the application : menu, and select Restart Application

The screenshot shows the 'Applications' list page. The 'Price Regressor Monitoring Dashboard' is listed and selected. A context menu is open over the application card, with 'Restart Application' highlighted. Other options in the menu include 'Application Details', 'Stop Application', and 'Delete Application'.

9. Click Price Regressor Tile

The screenshot shows the Cloudera Machine Learning interface. On the left, there's a sidebar with various project management and data-related tabs like Home, All Projects, Data, Experiments, Models, Jobs, Applications (which is currently selected), Collaborators, Project Settings, AMPs, Runtime Catalog, Learning Hub, and User Settings. The main area is titled 'Applications' and contains a search bar and a list of running applications. One application, 'Price Regressor Monitoring Dashboard', is highlighted with a yellow circle over its title, indicating it has been selected.

Explore drift and variations in the model performance

1. Explore the Dashboard

The screenshot shows the 'Price Regressor Monitoring' dashboard. At the top, there's a message: 'Drift is detected for 55.6% of features (5 out of 9). Dataset Drift is detected.' Below this, there's a table comparing 'Reference Distribution' and 'Current Distribution' for two categorical features: 'zipcode' and 'view'. The table includes columns for Feature, Type, Reference Distribution, Current Distribution, Data drift, and P-Value for Similarity Test. Both features show 'Detected' drift with a p-value of 0. At the bottom, there's a chart titled 'Numerical Target Drift' showing a line graph with several red markers, indicating significant variations in model performance over time.

Feature	Type	Reference Distribution	Current Distribution	Data drift	P-Value for Similarity Test
> zipcode	cat			Detected	0
> view	cat			Detected	0

2. Explore the three tabs at different dates

- Is data monotonously drifting?
- Is there Numerical Target Drift at any point?
- Are there any significant variations in the model performance?

Identify the file that creates the Evidently dashboard

Project Structure

```

18.
├── LICENSE
├── README.md
├── .project-metadata.yaml          # declarative specification for AMP logic
├── apps
│   ├── reports                   # folder to collect monitoring reports
│   └── app.py                     # Flask app to serve monitoring reports
├── cdswe-build.sh
├── data
├── requirements.txt
└── scripts
    ├── install_dependencies.py    # commands to install python package dependencies
    ├── predict.py                 # inference script that utilizes cdswe.model_metrics
    ├── prepare_data.py            # splits raw data into training and production sets
    ├── simulate.py                # script that runs simulated production logic
    └── train.py                  # build and train an sklearn pipeline for regression
└── setup.py
src
├── __init__.py
├── api.py                      # utility class for working with CML APIv2
├── inference.py                 # utility class for concurrent model requests
├── simulation.py                # utility class for simulation logic
└── utils.py                     # various utility functions

```

The Continuous Model Monitoring AMP project structure is above.

1. Select **Models** from the project menu, you see the Price Regressor model is deployed.
2. Click on the **Price Regressor** model.
3. Explore the model's properties. What file is the model based on? Which function is called?
4. Click the **Test** button to test the model. View the results.
5. Click **Files** in the project menu. Navigate to the `scripts/predict.py` file. Click the **Open in Session** button.
6. Go to line 87 and examine the `cdsw.track_metric` call that is used to store the prediction metrics in the CML PostgreSQL metrics database.



Your CML workspace must have the **Enable Model Metrics** option selected when it is provisioned in order to track metrics.

The `src\simulation.py` file is the main simulation routine and mimics a production monitoring use case.

This simulation assumes the Price Regressor model has already been deployed, and accepts that model name as input. The `.run_simulation()` method operates the main logic of this class. Namely, it:

- Scores all training data against the deployed model so we can query metrics from the Model Metrics database for evaluation
- Initializes a simulation clock, which is just a list of date ranges from the `prod_df` to iterate over. These batches mimic the cadence upon which new data "arrives" in a production setting.
- For each simulation clock date_range, we:
 - Query the `prod_df` for newly *listed* records and score them using deployed model
 - Query the `prod_df` for newly *sold* records and add ground truths to metric store
 - Query the metric store for those newly *sold* records and generate new Evidently report
- Redeploy the hosted Application to surface the new monitoring report

1. Navigate to `src/simulation.py` and open the file in the Workbench Editor.
2. Examine the `query_model_metrics` method on line 340. This method reads the metrics stored in the CML model metric database.
3. Examine the `build_evidently_reports` method on line 404. This method builds the reports and saves them as HTML to be served by the Price Regressor Monitoring Dashboard application.
4. Go to line 186 and examine how the simulation deploys the updated Price Regressor Monitoring Dashboard application upon simulation completion. This is an example of using the CML API to deploy an application. More details can be seen in the `src/api.py` file.

End of Exercise