AWS Immersion - Hands-On-Labs Student Guide

Text Summarization with CML and Amazon Bedrock - Hands On Lab

Objective:

In this exercise we will implement an LLM application using Cloudera Machine Learning and Amazon Bedrock, including:

Lab 1: CML Walkthrough - Tour (CML Tour) **2**
Lab 2: Text Summarization AMP Walkthrough **7**
Lab 3: Manually Create Project via Git **9**
Lab 4: Test Amazon Bedrock API using Jupyter Notebook **12**
Lab 5: Deploy LLM application **17**
Lab 6: Deploy AMP **23**

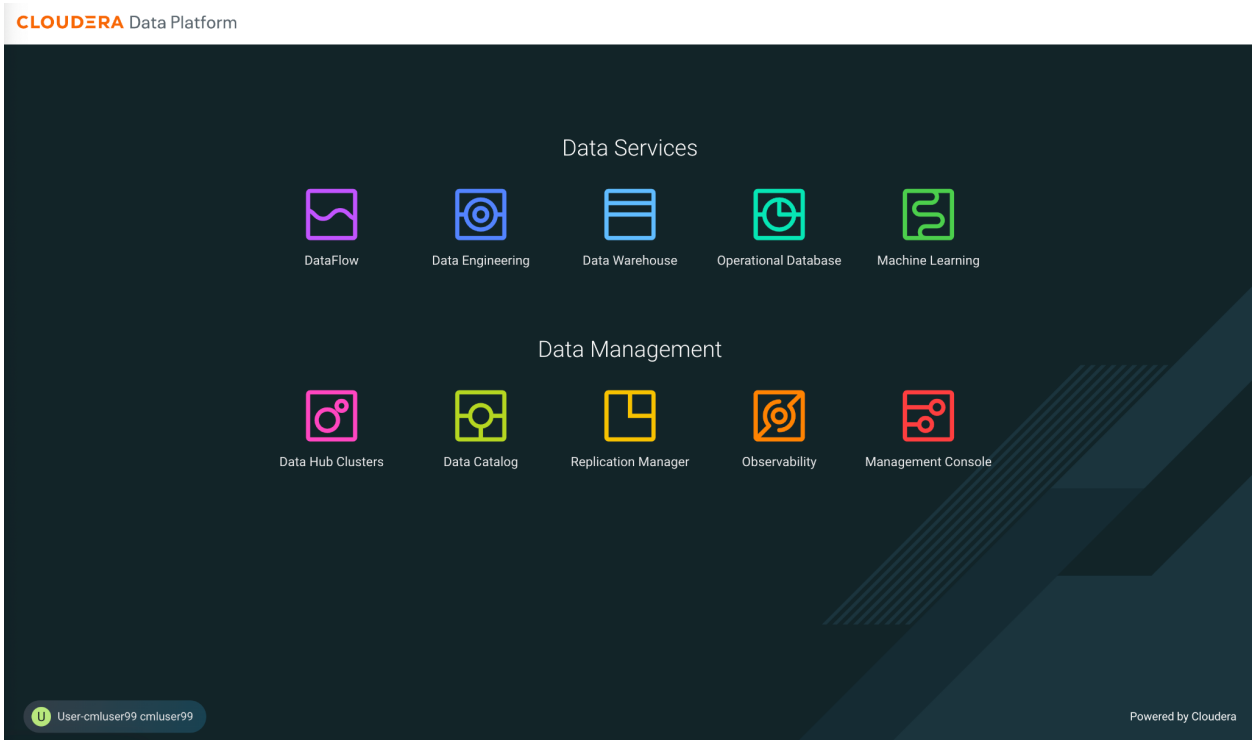
Lab 1: CML Walkthrough - Tour (CML Tour)

Cloudera Machine Learning (CML) makes it easy to bring Data Scientists to the data platform and makes it easier to overcome the challenges they face on a day to day basis.

With CML you can focus on writing the machine learning code and productionizing models instead of worrying about how to access the data.

Workshop user login Keycloak SSO URL:
http://3.16.19.115/auth/realms/master/protocol/saml/clients/cdp-sso

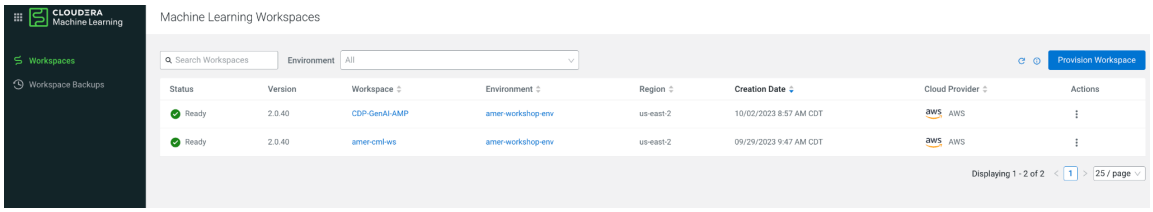
1. Click on the Machine Learning icon to continue.



2. Open this workspace by clicking on its name -**amer-cml-ws**

   Workspaces are different areas where teams of data scientists can be separated based on factors such as the cloud provider they're working on and the environment they have access to.

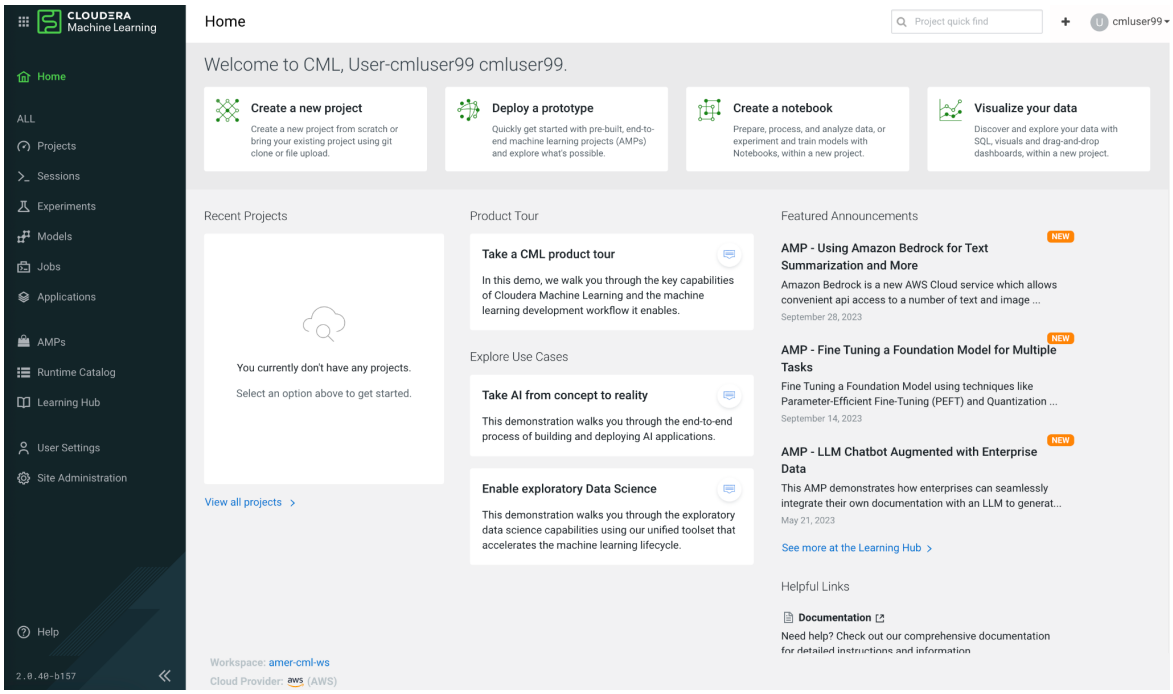
   Here we have the **amer-cml-ws** that is created on an environment called **amer-workshop-env** hosted on **AWS**. CML allows you to choose the particular compute instances you would like for a new Workspace, including using GPUs.



   On the main page you can see a list of projects that you are working on with your data science team members and action cards that take you through the process of creating a project, deploying prototypes, creating a notebook, and visualizing your data.

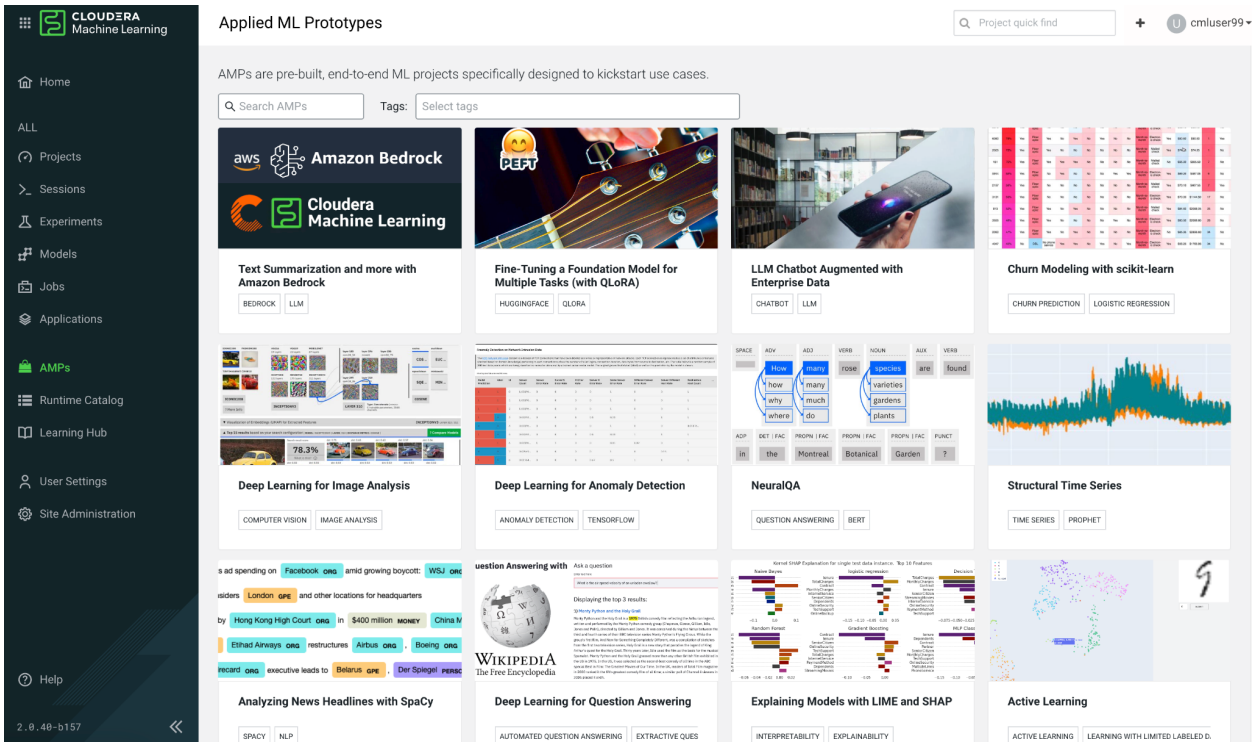
   A project is a collaborative space for a team of Data Scientists to work on a single project together. Here you can view the code and files that belong to a project.

   Also, all of the other assets that stem from this project (i.e. Models, Jobs) are visible here as well.



3. Click on **AMPs** on the left menu bar

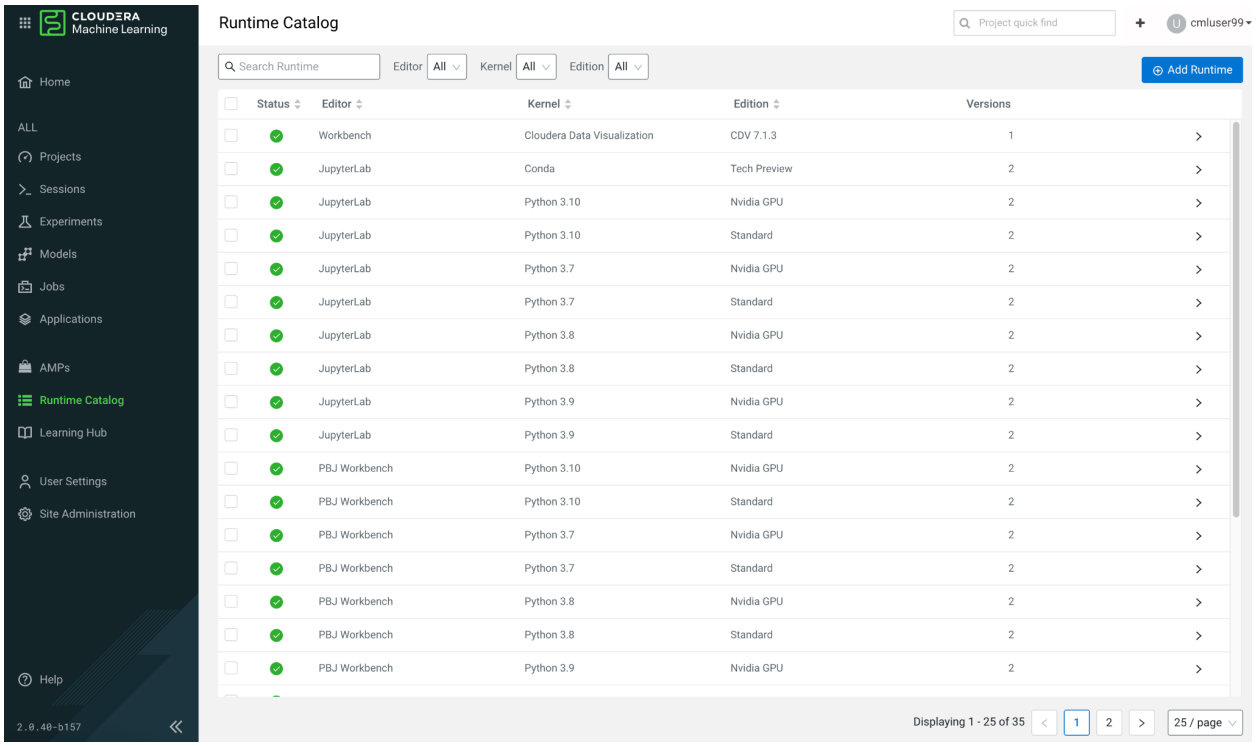
   Applied ML Prototypes (AMPs) provide reference example machine learning projects in Cloudera Machine Learning. More than simplified quickstarts or tutorials, AMPs are fully-developed expert solutions created by Cloudera's research arm, Fast Forward Labs.



4. Click on **Runtime Catalog** on the left menu bar

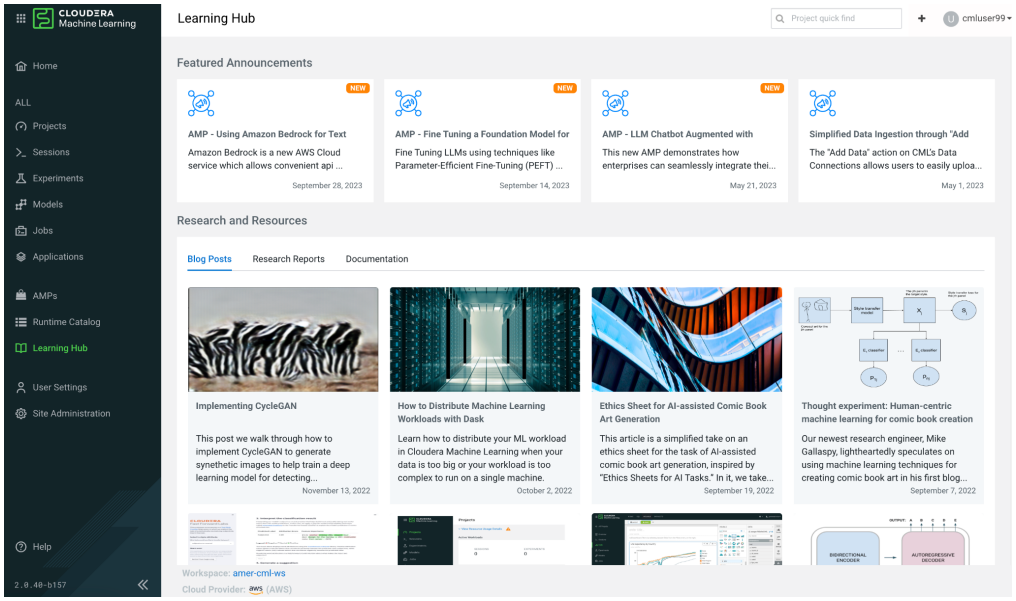
   Managing ML Runtimes provides overview, installation, set up, configuration, and customization information for Machine Learning Runtimes. ML Runtimes are responsible for running the code written by users and intermediating access to the Data Hub.

   You can think of an ML Runtime as a virtual machine, customized to have all the necessary dependencies to access the computing cluster while keeping each project's environment entirely isolated.



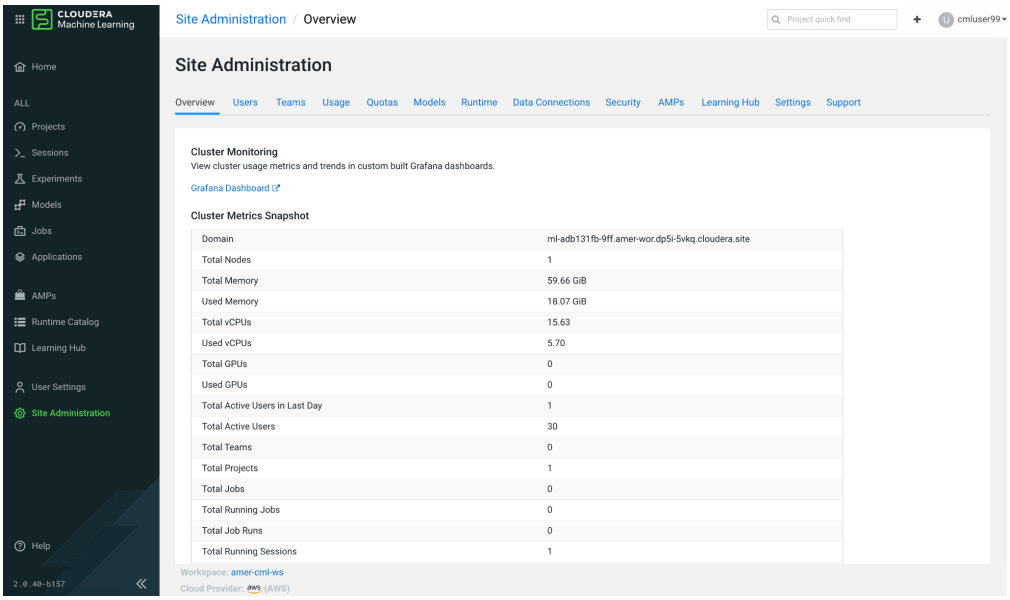
5. Click on **Learning Hub** on the left menu bar

   Learning Hub highlights Featured Announcements, Research and Resources, including Blog Posts, Research Reports and Documentation. It also contains a Feed associated with Cloudera Machine Learning that contains features, blog posts, documentation updates, etc...



6. Click on **Site Administration** on the left menu bar

   Only site administrators have access to a Site Administration dashboard that can be used to manage the workspace. When you're in Site Administration, you will see an array of tabs for all the tasks you can perform as a site administrator.



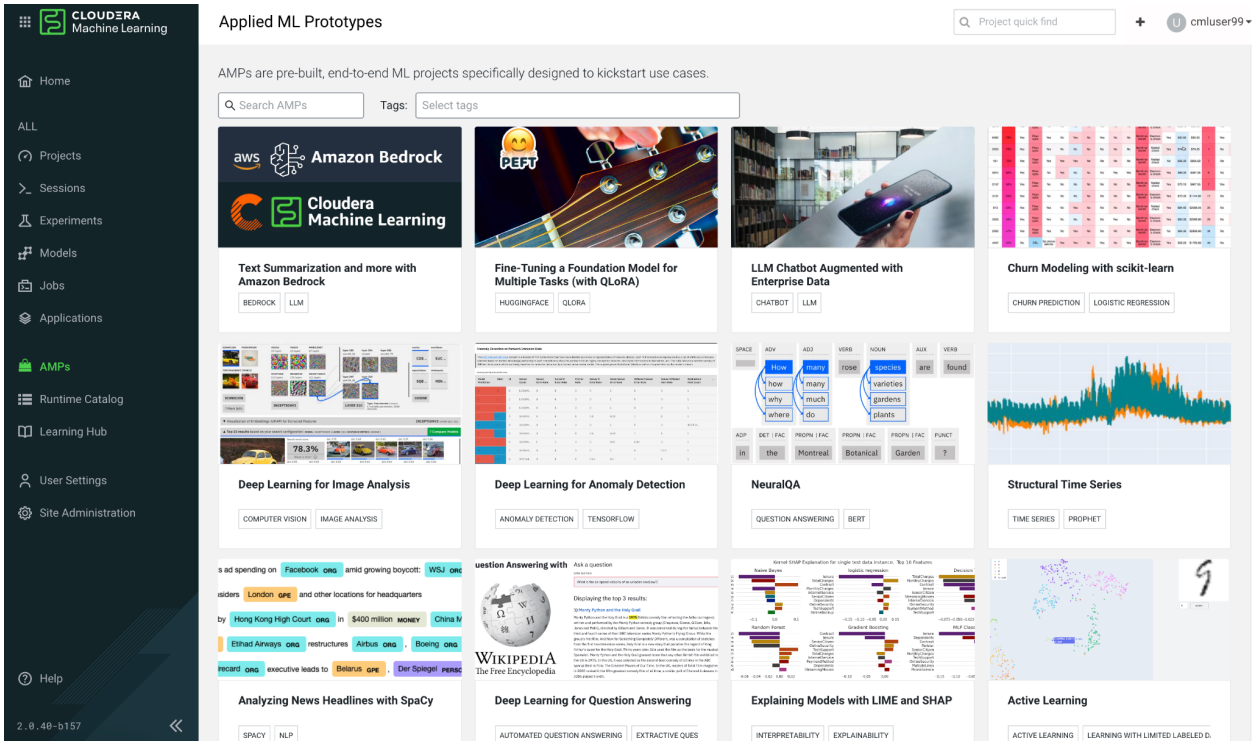
Lab 2: Text Summarization AMP Walkthrough

Text Summarization and more with Amazon Bedrock
Amazon Bedrock is a fully managed service that makes foundation models from leading AI startups and Amazon available via an API, so you can choose from a wide range of foundation models to find the one that is best suited for your use case.

This repository demonstrates using an instruction to summarize a text using the Amazon Bedrock API.

AMP Overview
In this AMP we will show you how to set up your environment and use the Amazon Bedrock API and its models. There are two models that you can choose from.

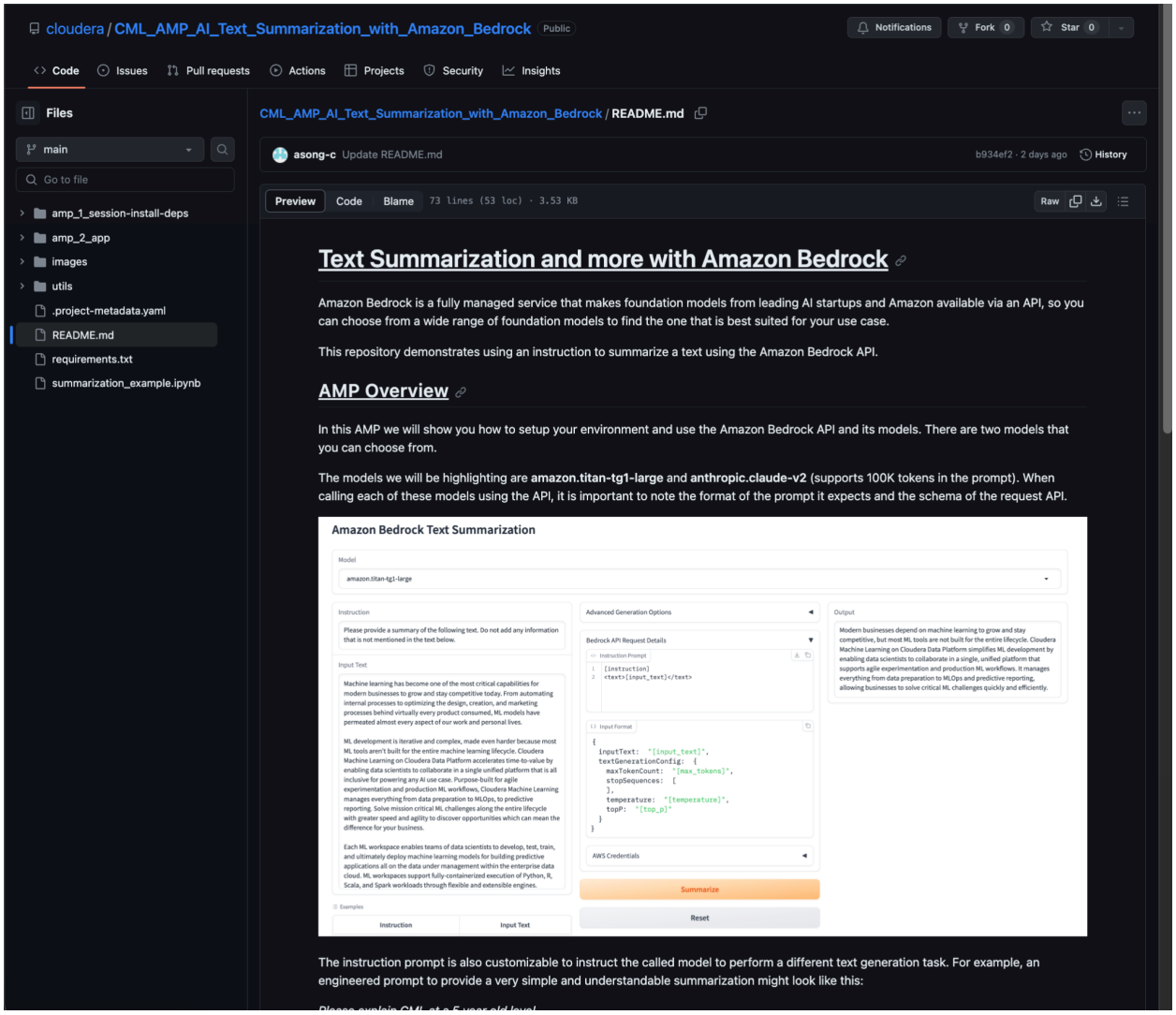
The models we will be highlighting are amazon.titan-tg1-large and anthropic.claude-v2 (supports 100K tokens in the prompt). When calling each of these models using the API, it is important to note the format of the prompt it expects and the schema of the request API.



The instruction prompt is also customizable to instruct the called model to perform a different text generation task. For example, an engineered prompt to provide a very simple and understandable summarization might look like this:

> Please explain CML at a 5 year old level.

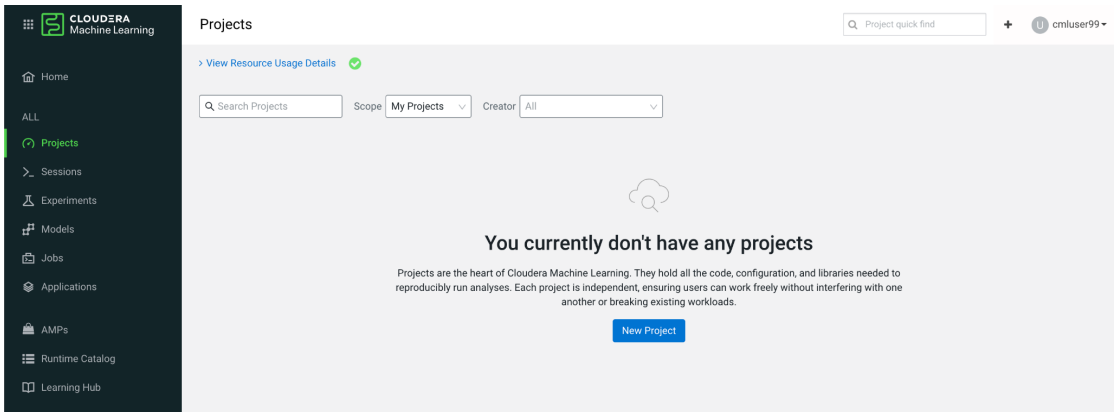
Machine learning is a big helper for companies to do their work and make their products better. It helps automate things and make decisions faster. Cloudera Machine Learning makes it easier for scientists to work together and create machine learning models. They can use different languages and computers to do this in one place. It also helps keep track of everything and make sure it works well.



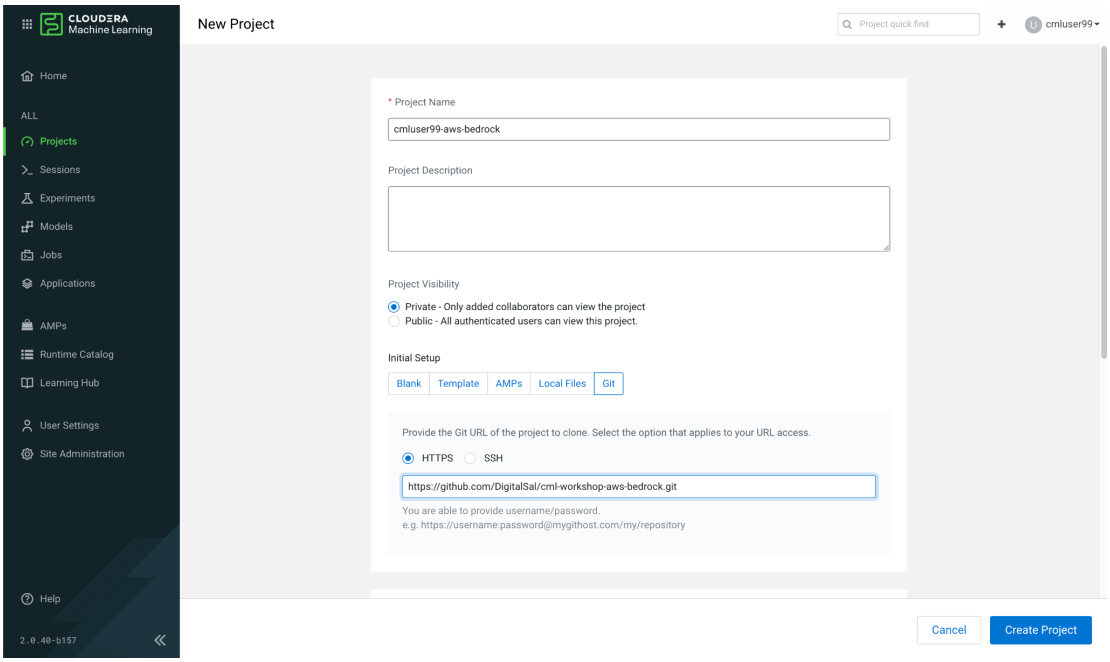
Lab 3: Manually Create Project via Git

In Lab 3 we're going to manually create a project from github and configure it to run a Text Summarization app that uses Amazon Bedrock APIs.

1. Click on **Projects** on the left menu bar
2. Click on the **New Project** button on the main screen

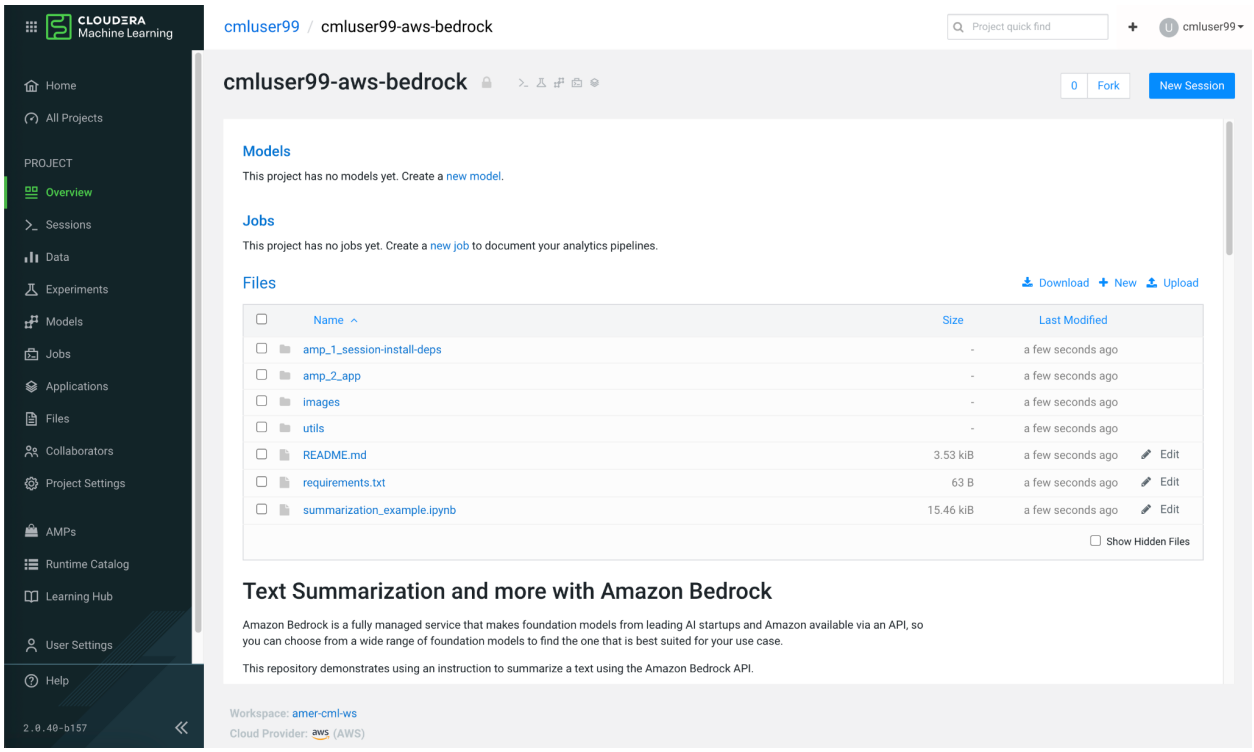


3. Complete the **New Project** form.
    - Provide a **Project Name**.
    - Project Visibility: **Private**
    - Initial Setup: **Git**
        - Git URL: **https://github.com/DigitalSal/cml-workshop-aws-bedrock.git**
    - Click on the **Create Project** button

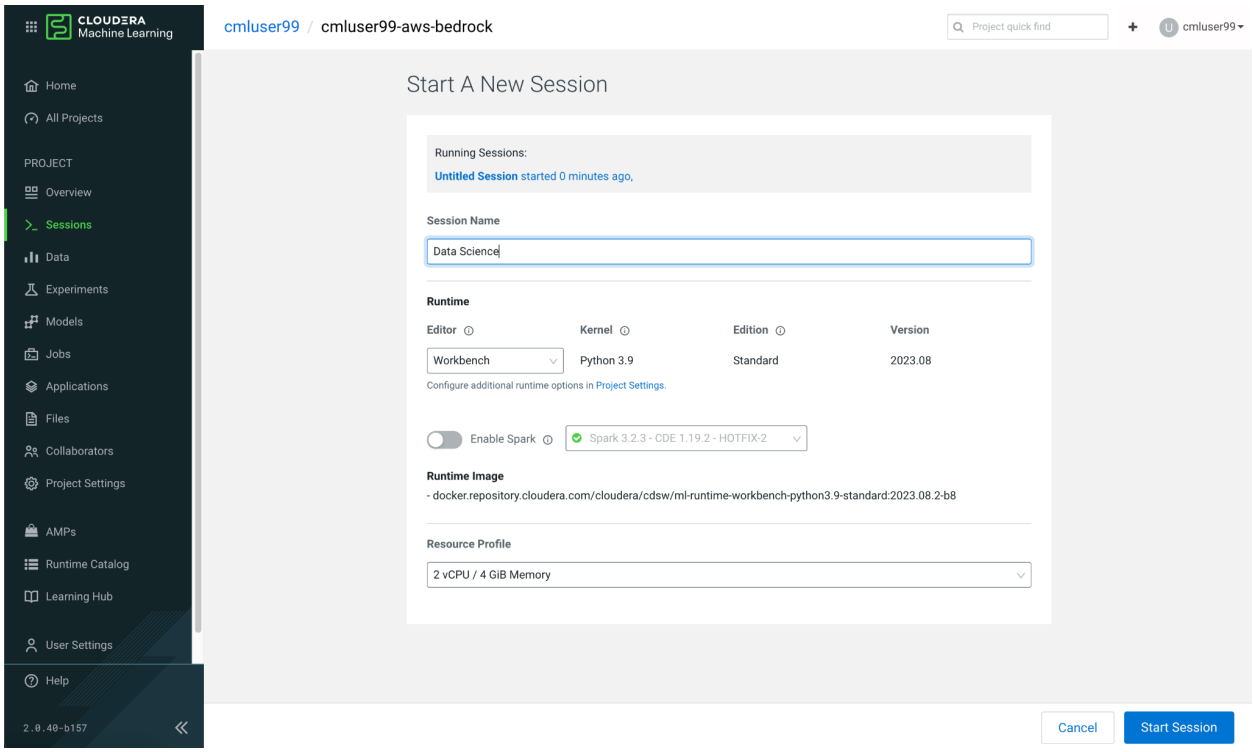


Your new project should look something like this.

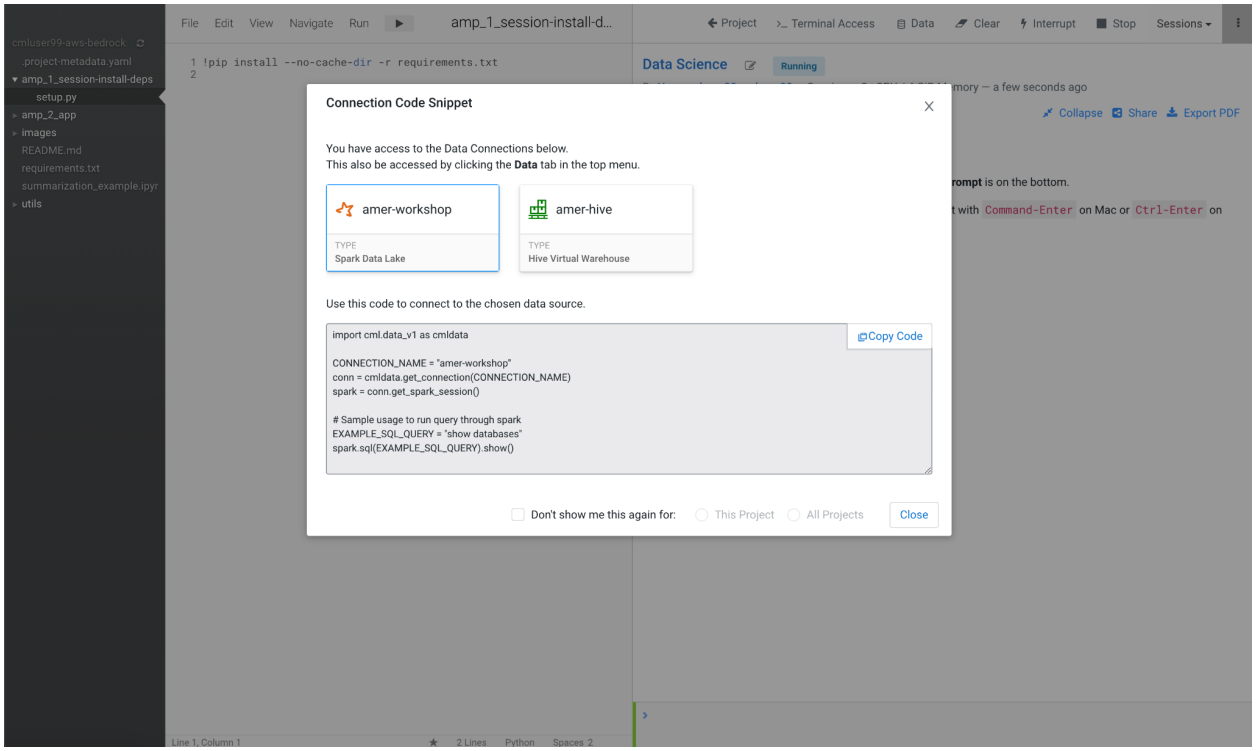
4. Click on the **New Session** button.



5. Provide a session name and click the **Start Session** button to start a Workbench session.

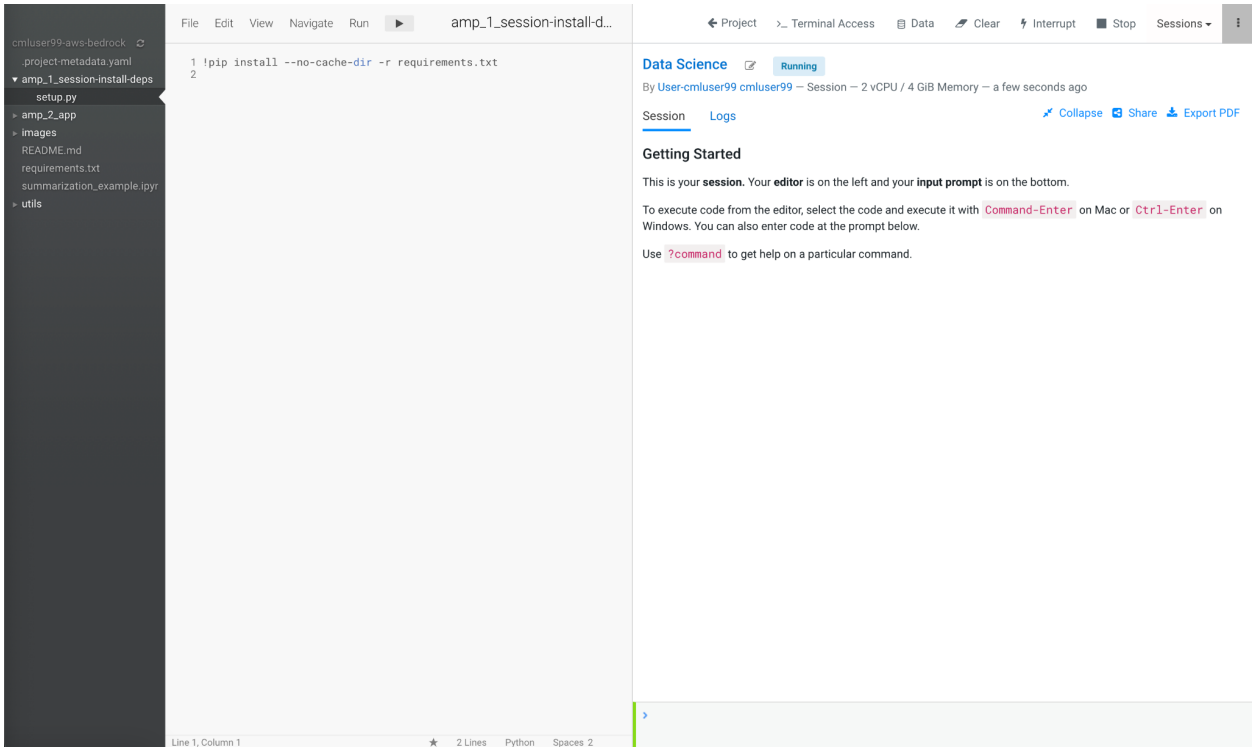


6. Click the **Close** button to close the Connection code snippet pop up.



Lab 4: Test Amazon Bedrock API using Jupyter Notebook

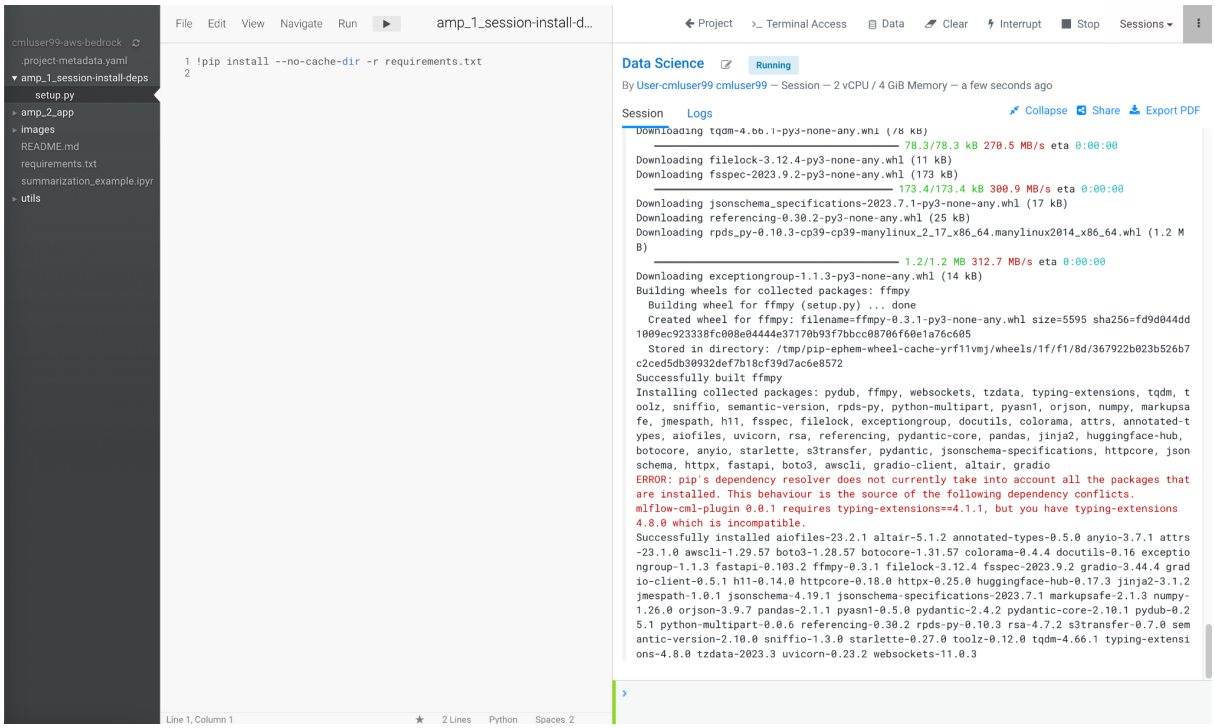
1. Select the **setup.py** file in the amp_1_session-install-deps folder. This file installs requirements listed in the requirements.txt file.



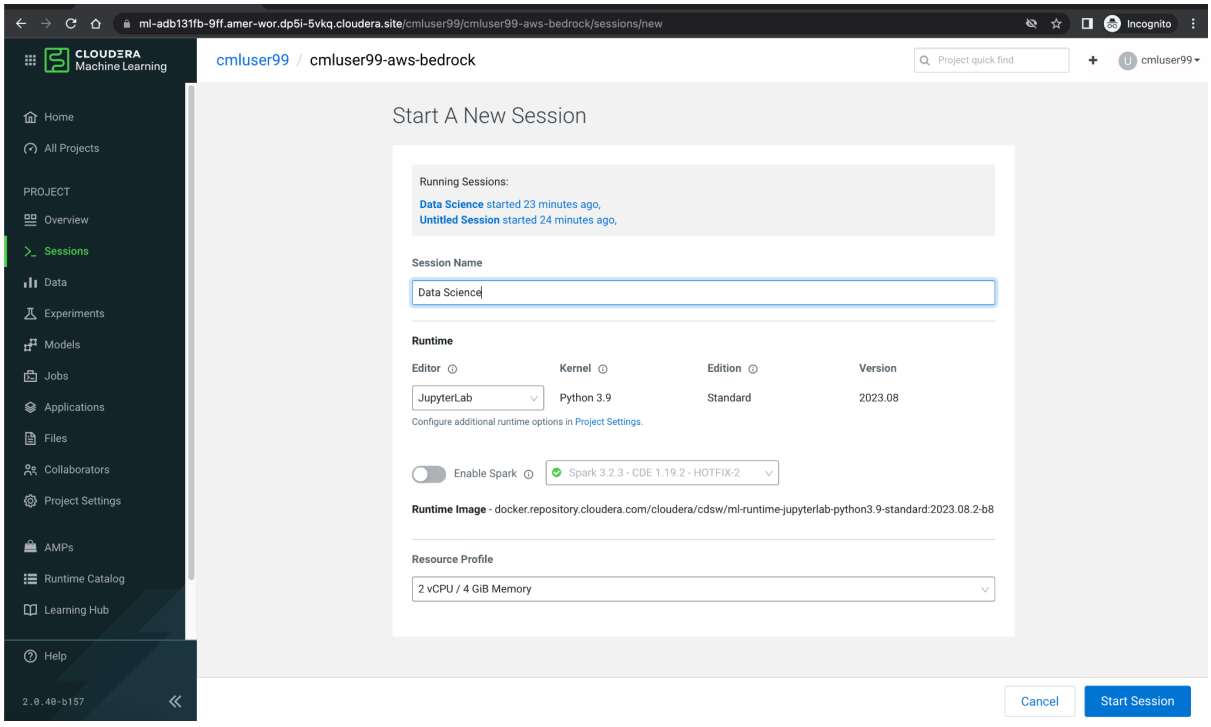
This code will take a few minutes to run. When code has finished you will see the green indicator that shows the cursor is ready.

Now we're going to run a jupyter notebook to test our code.

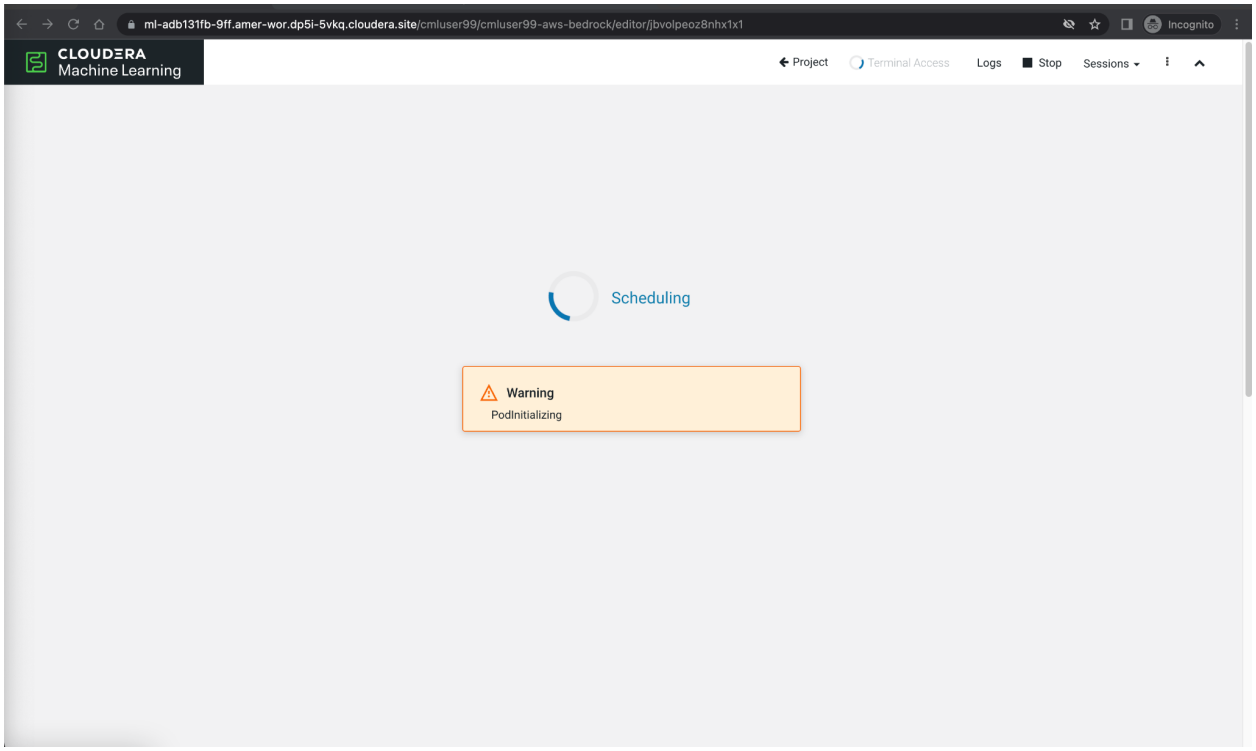
2. Click on **Sessions** menu and then click on **+New Session**



3. Make sure to select the **JupyterLab** editor and click **Start Session** button

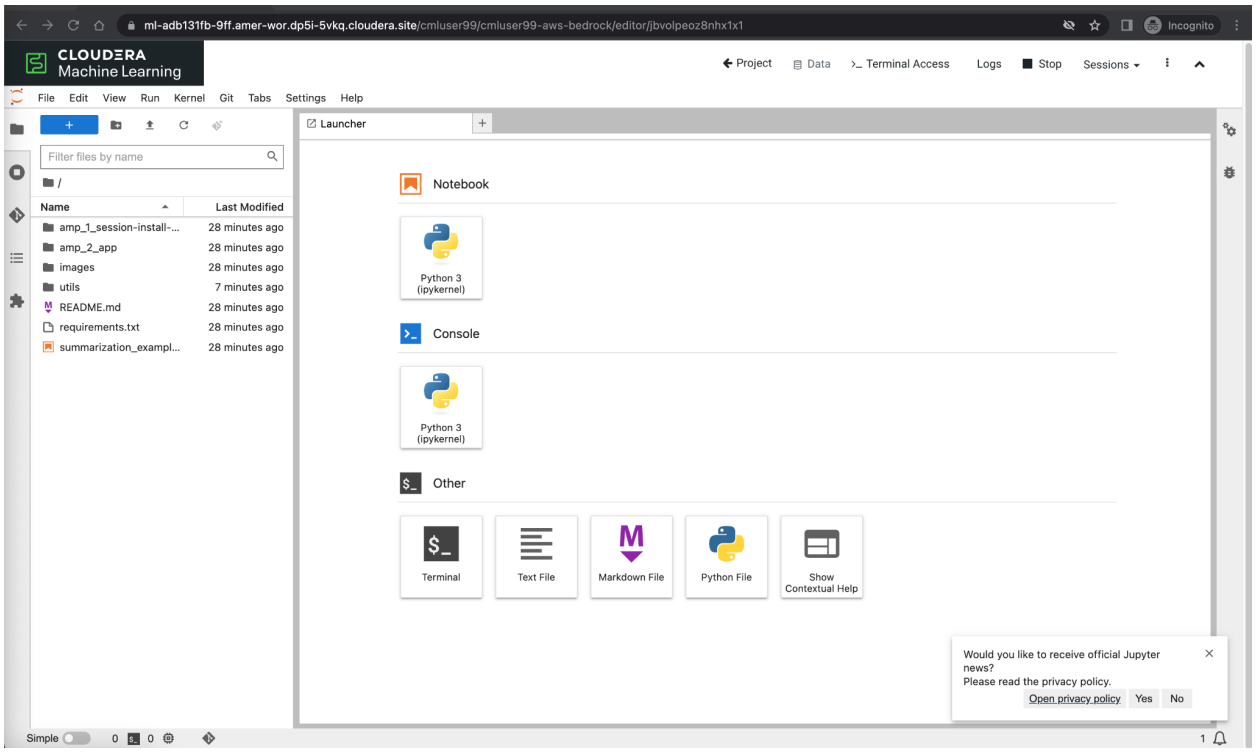


This will take a few seconds to load.



Once the JupyterLab editor is open your screen should look like this.

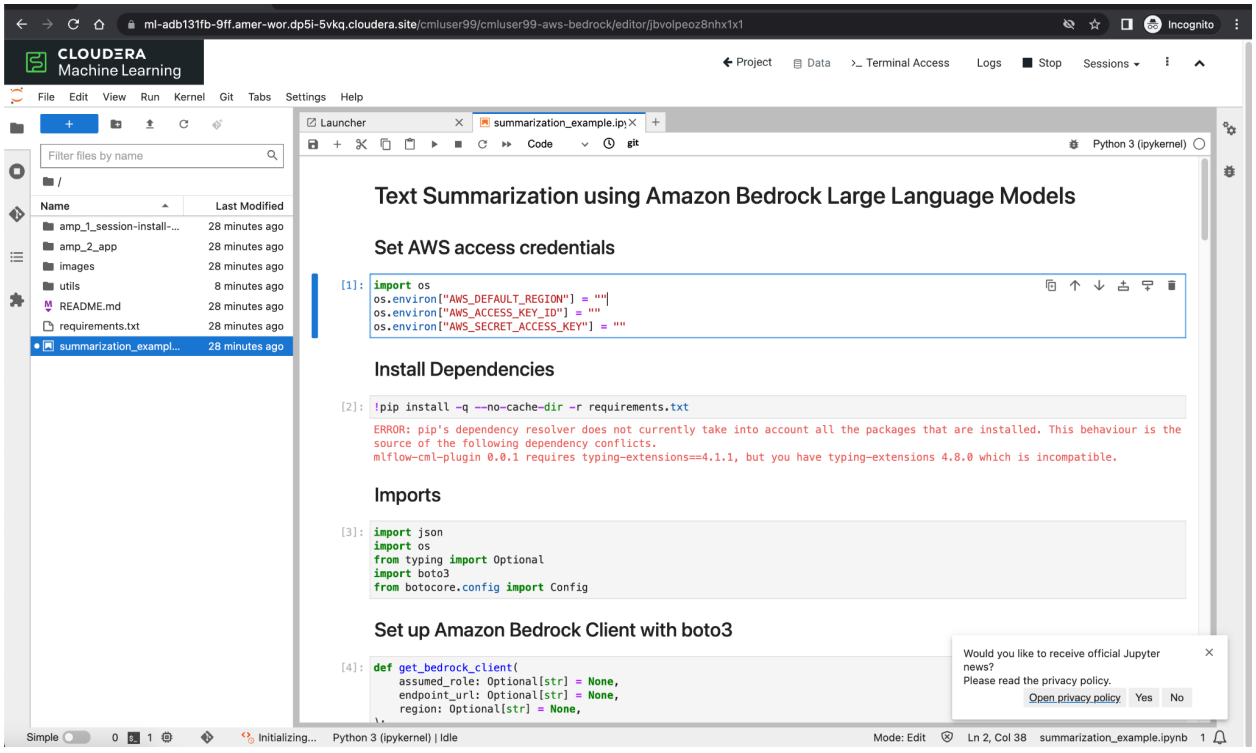
4. Click on the **summarization_example.ipynb** file



Feel free to look over the code.

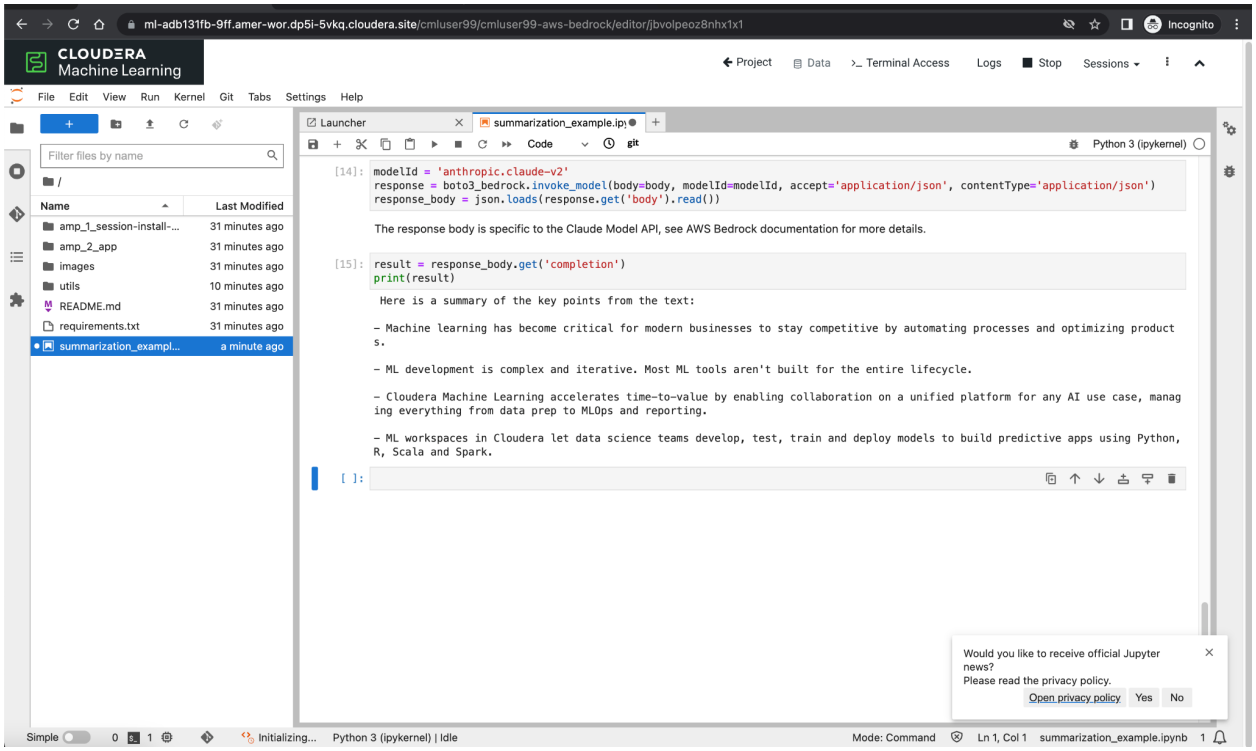
5. In the first block, update the code to the following:
    - os.environ["AWS_DEFAULT_REGION"] = "us-east-1"
    - os.environ["AWS_ACCESS_KEY_ID"] = "AKIA6I6SNFMLB7UQIFWL"
    - os.environ["AWS_SECRET_ACCESS_KEY"] = "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"

6. Click on **Run** -> **Run All Cells** or run them line by line



Note: Titan model will error out.

The screen should look like this when completed:



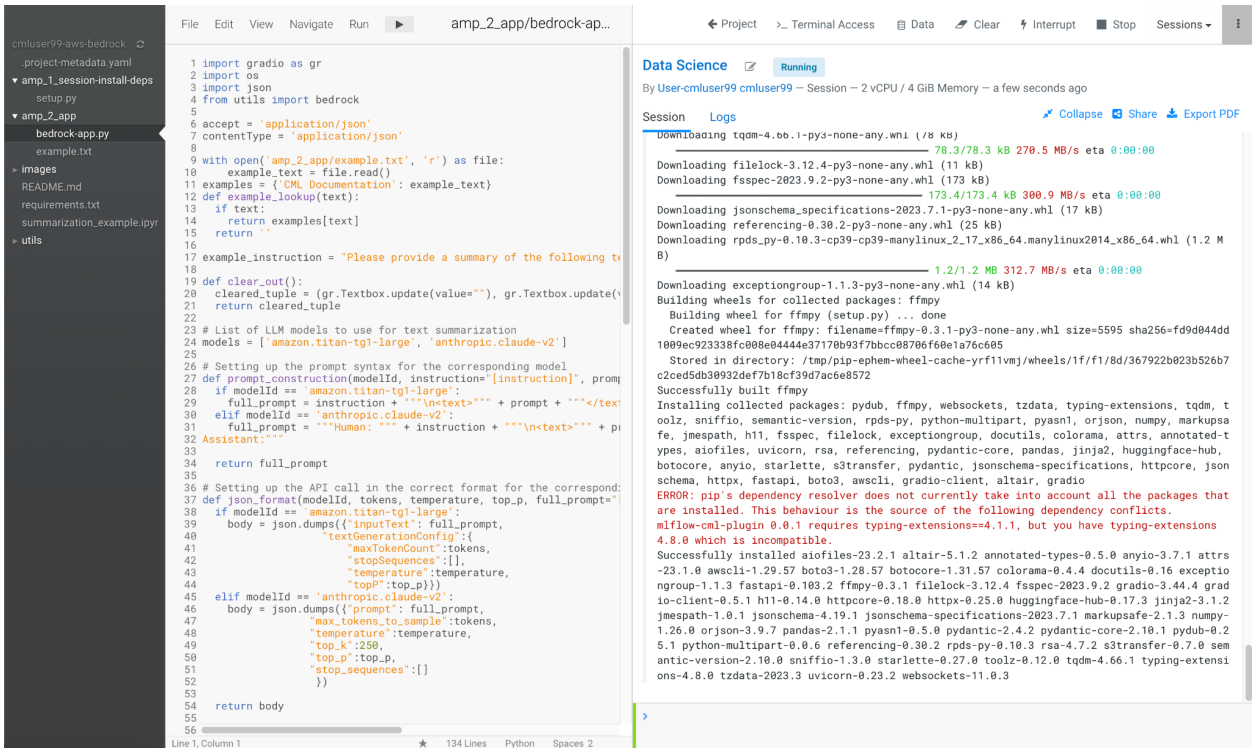
7. Stop the JupyterLab session, by clicking on **Stop** in the top menu.

Lab 5: Deploy LLM application

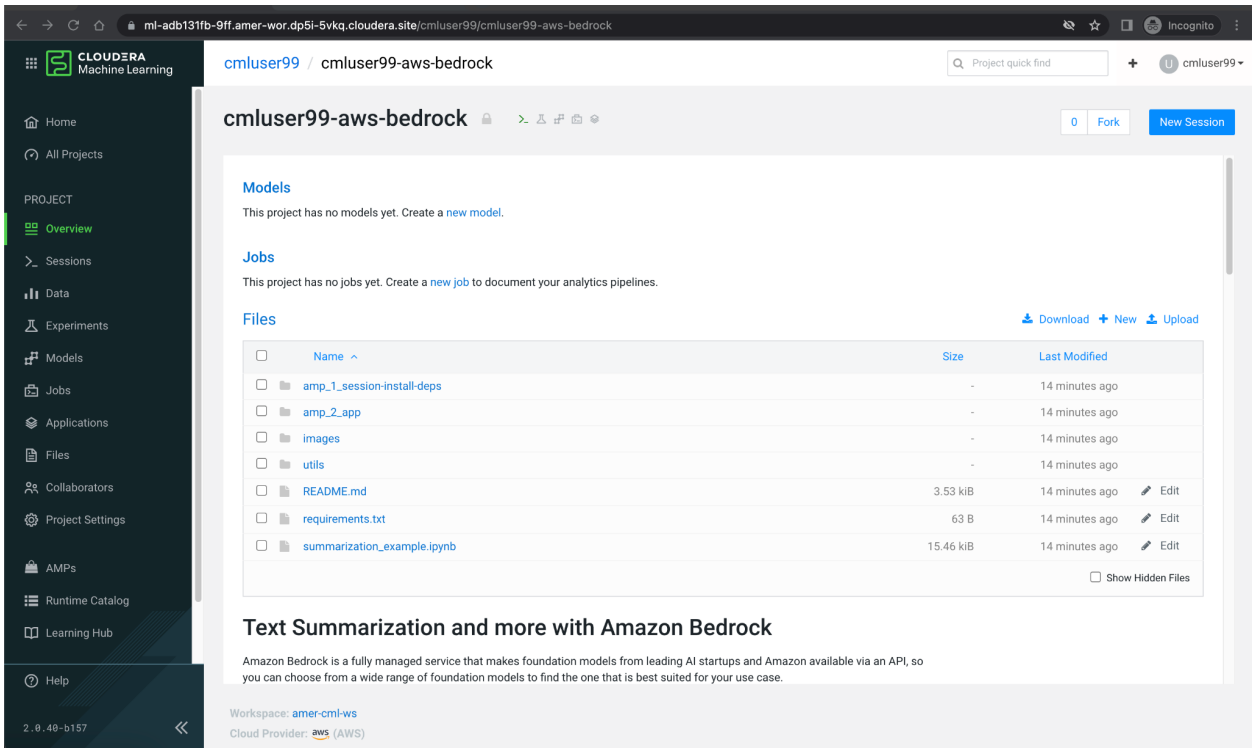
With CML can deploy web applications. In this lab we will manually deploy a UI that interacts with the LLM models.

The application file is located in the **amp_2_app** folder called **bedrock-app.py**.

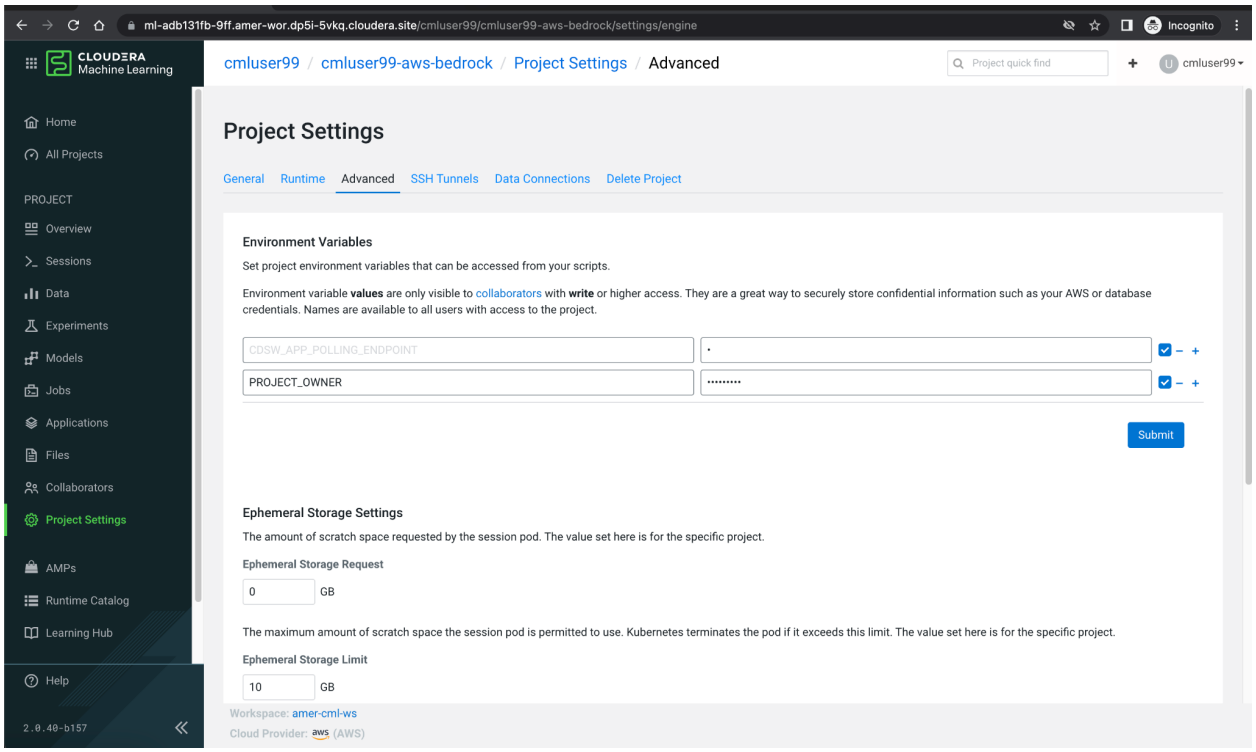
1. Stop session and get back to the Project.



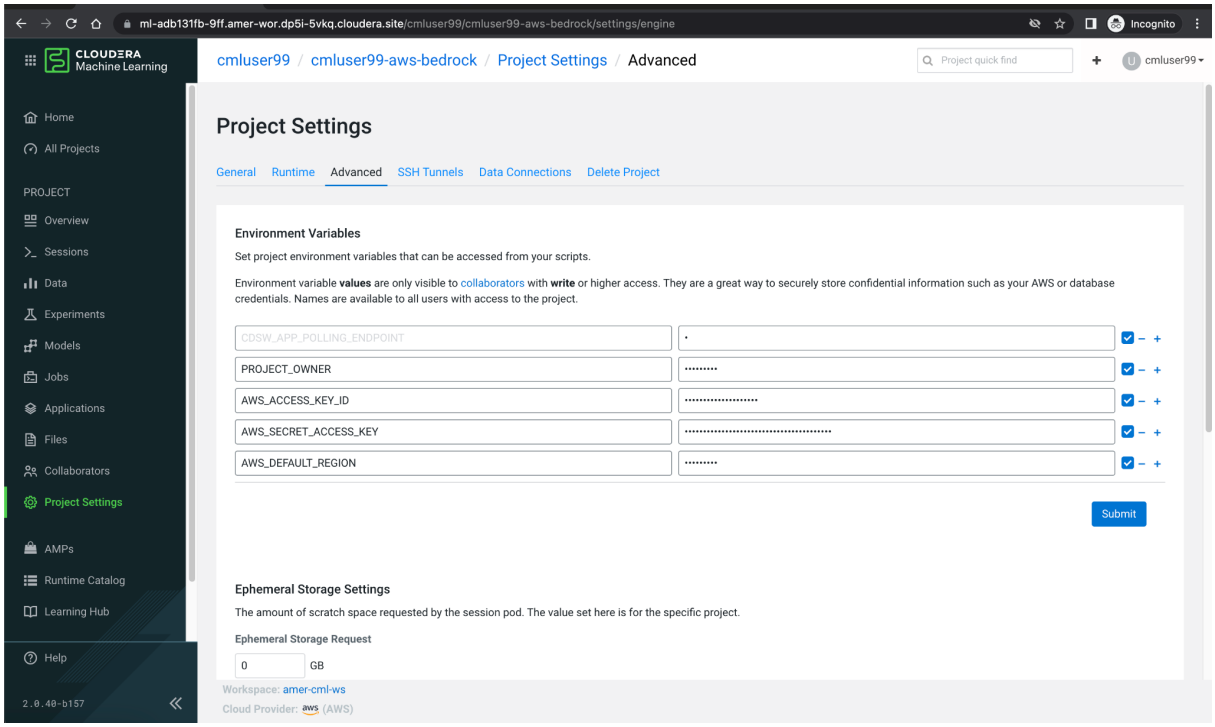
2. Click on the **Project Settings** menu item on the left.



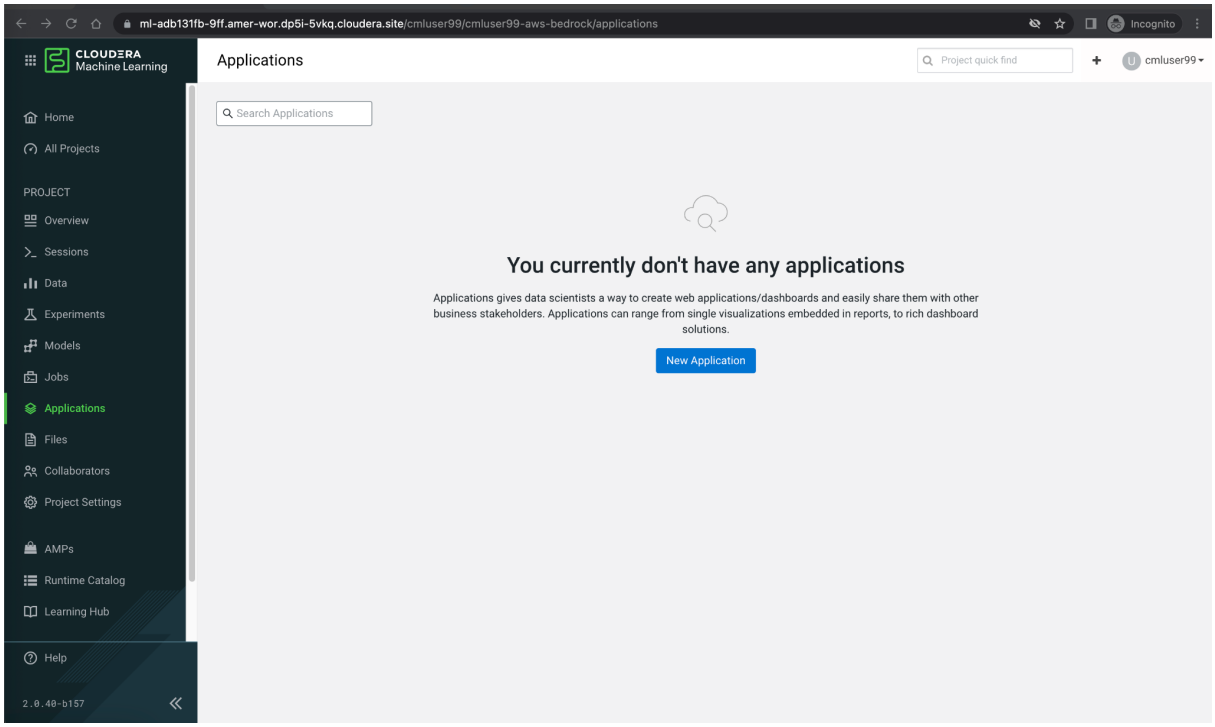
3. Click on the **Advanced** tab.



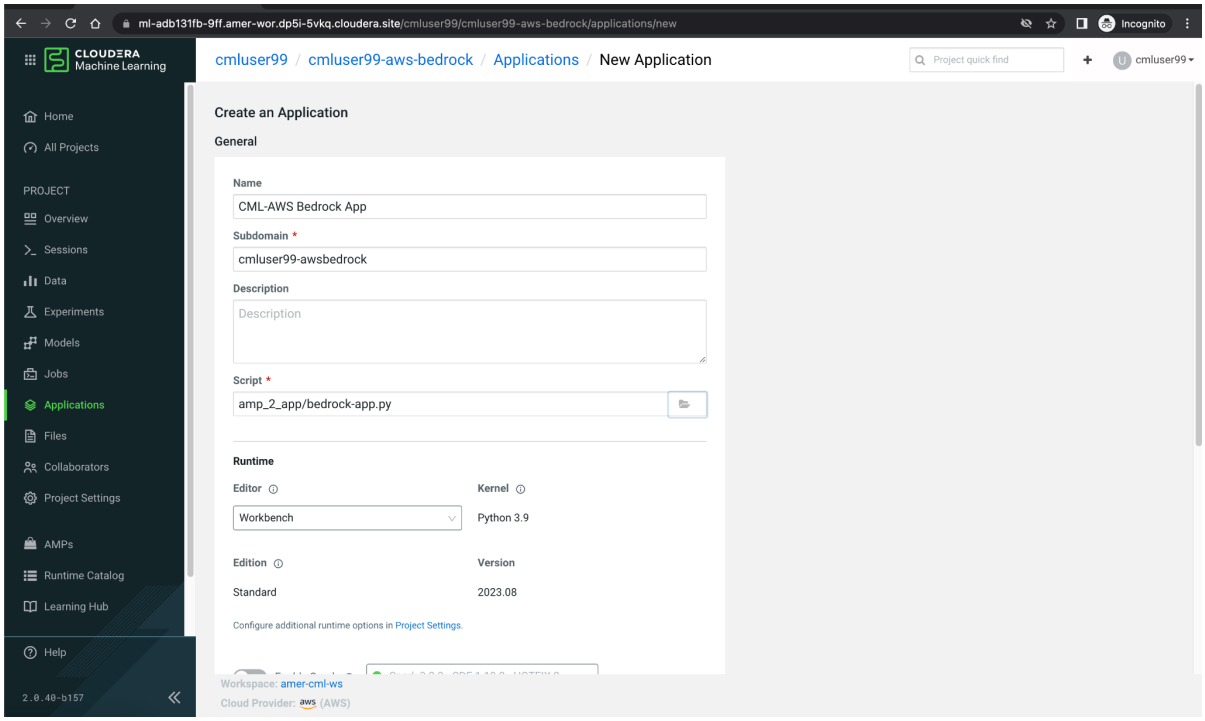
4. Enter 3 Environment Variables and their values that the application will reference:
   - **AWS_DEFAULT_REGION** = us-east-1
   - **AWS_ACCESS_KEY_ID** = AKIA6I6SNFMLB7UQIFWL
   - **AWS_SECRET_ACCESS_KEY** = 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
5. Click the **Submit** button when complete.

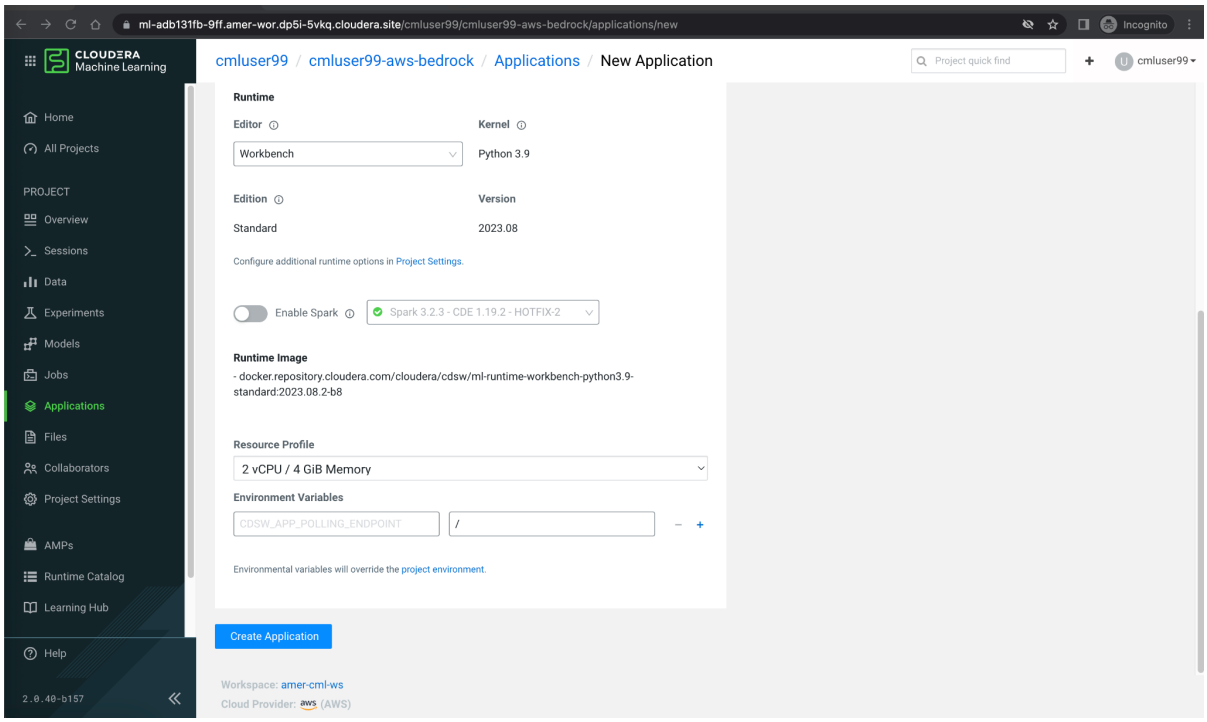


6. Click on the **Application** menu item on the left.
7. Click on the **New Application** button

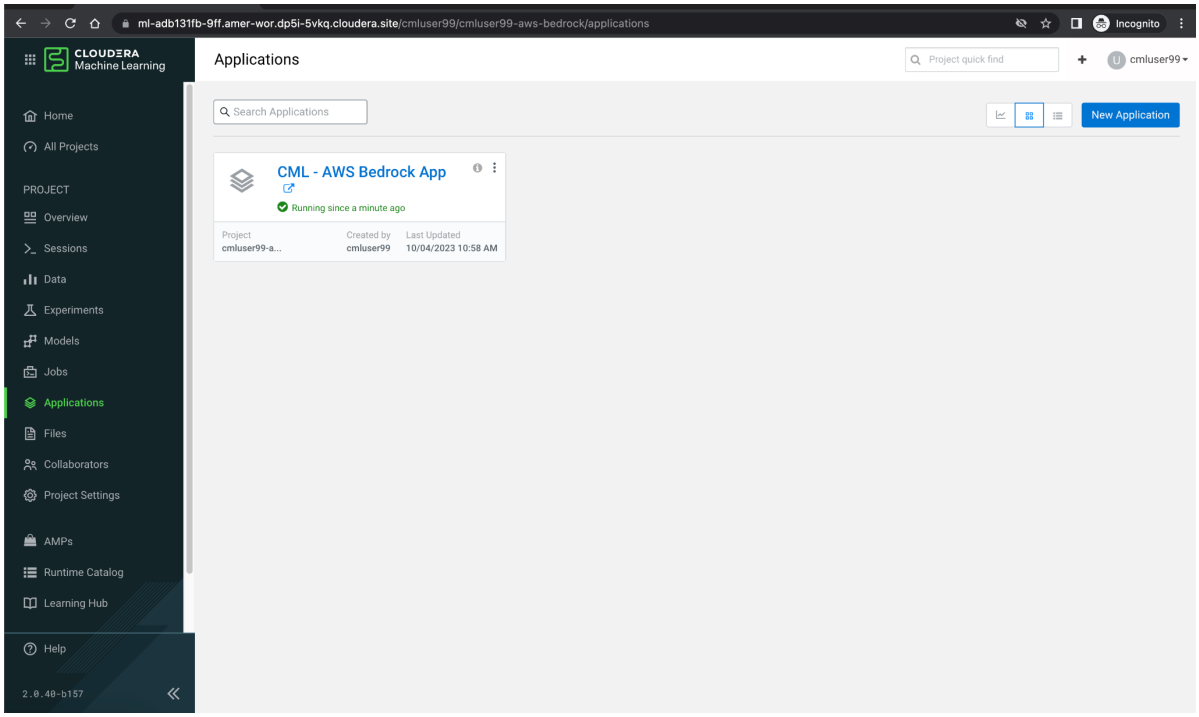


8. Create an Application - enter the following details
    - Name: Provide your app a name
    - Subdomain: make it unique by adding your username
    - Script: Select the bedrock-app.py file in the amp_2_app folder
    - Runtime Editor: Workbench
    - Click the Create Application button at the bottom of the page.

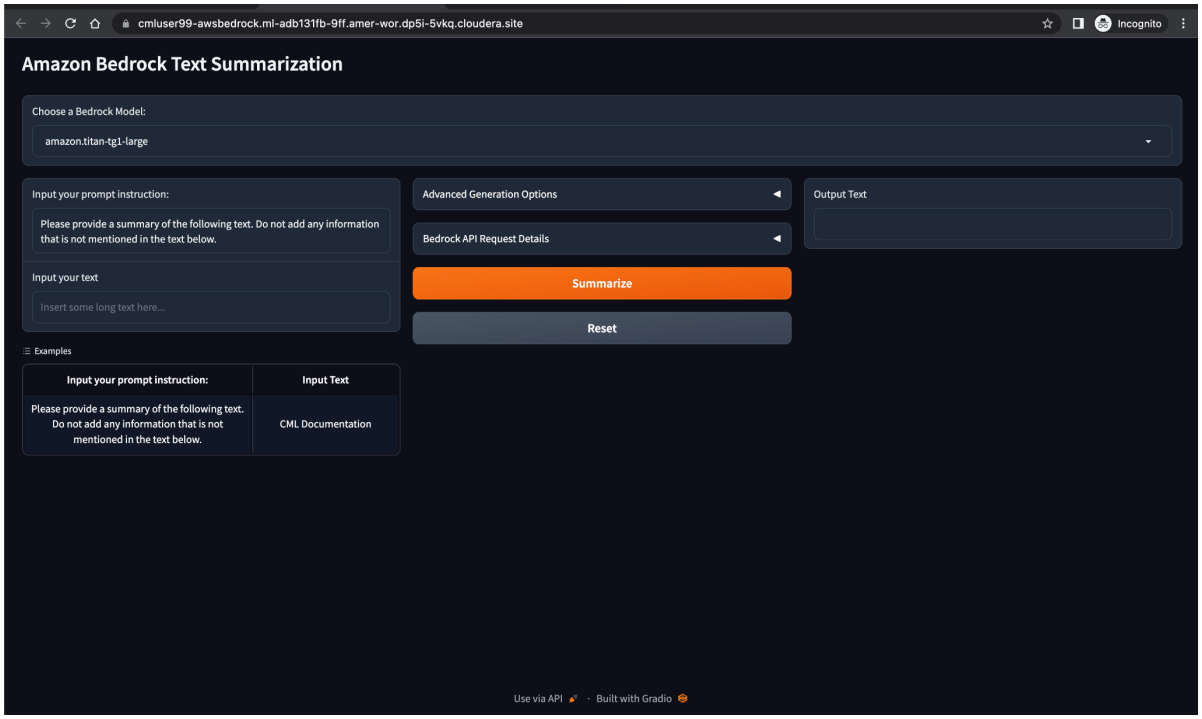




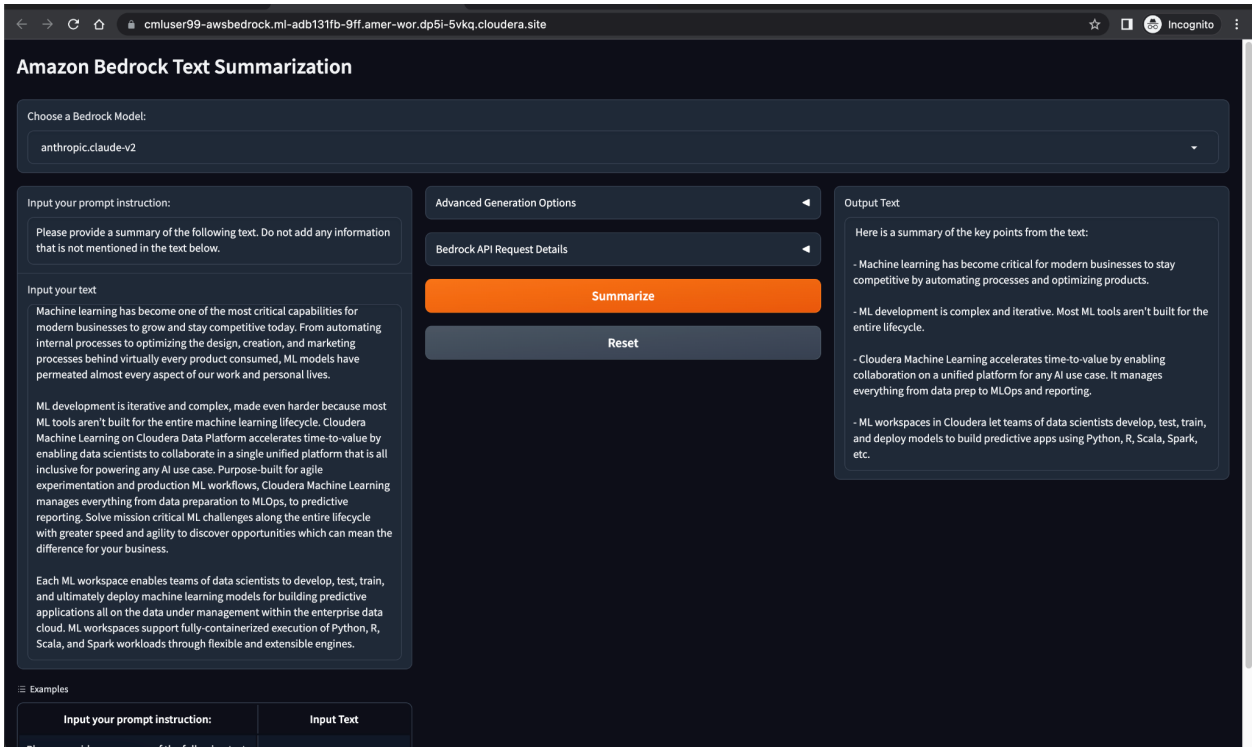
9. Now that you have an application, click on it to open the app.



10. For the Titan Model, click on the **Examples** table. This will populate the "**Input your text"** box
11. Click on the **Summarize** button. This should generate an error message, like when we were testing.



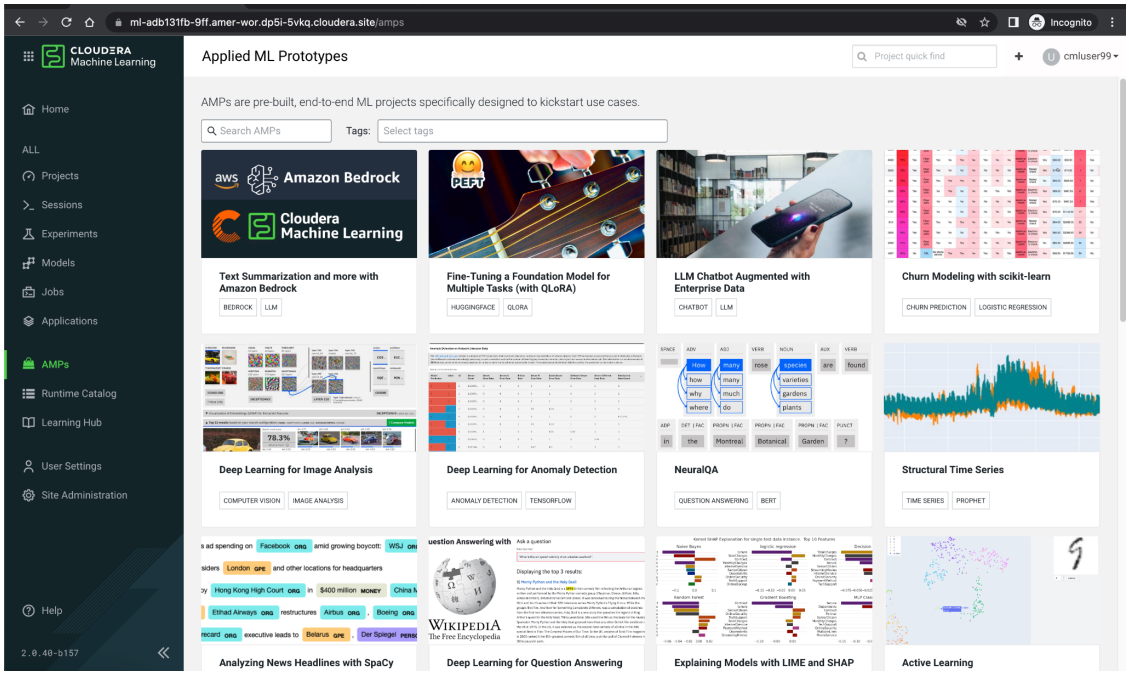
12. **Choose a Bedrock Model**: Select the Claude Model
13. Click the **Summarize** button. You should see the **Output Text**
    Feel free to test with other text



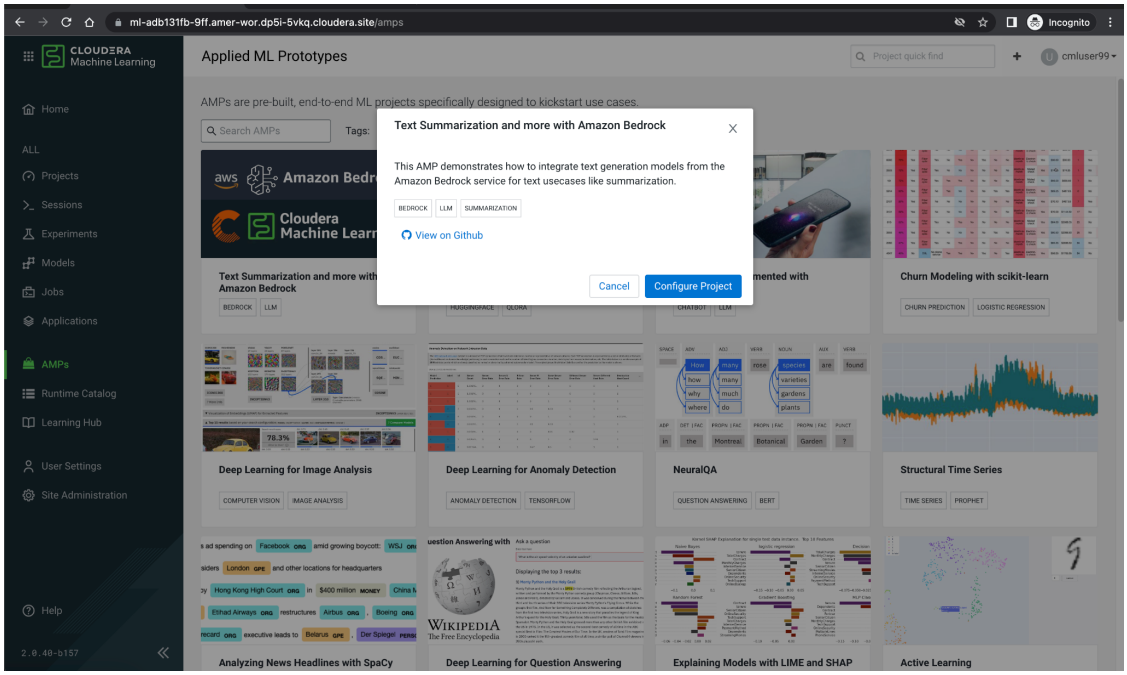
Lab 6: Deploy AMP

Now that we've manually created a project, tested code and manually deployed an app, let's deploy a CML AMP from the AMP catalog.

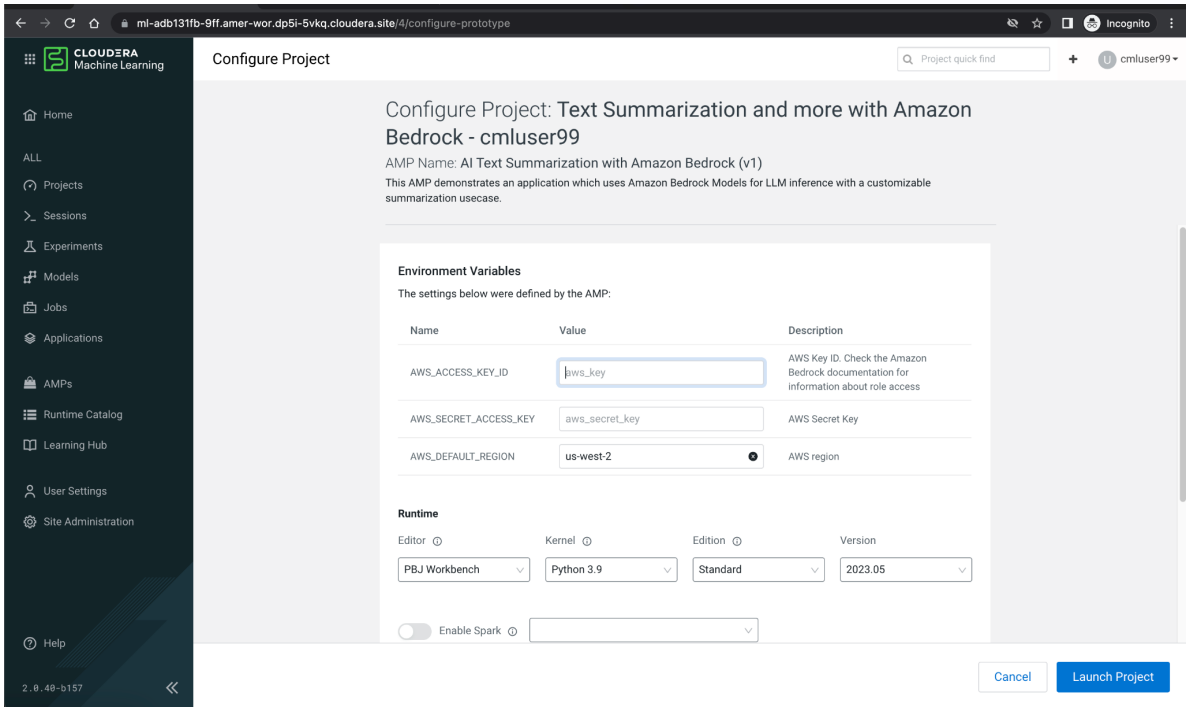
1. Click on **AMPs** on the menu item on the left
2. Click on the **Text Summarization and more with Amazon Bedrock** AMP



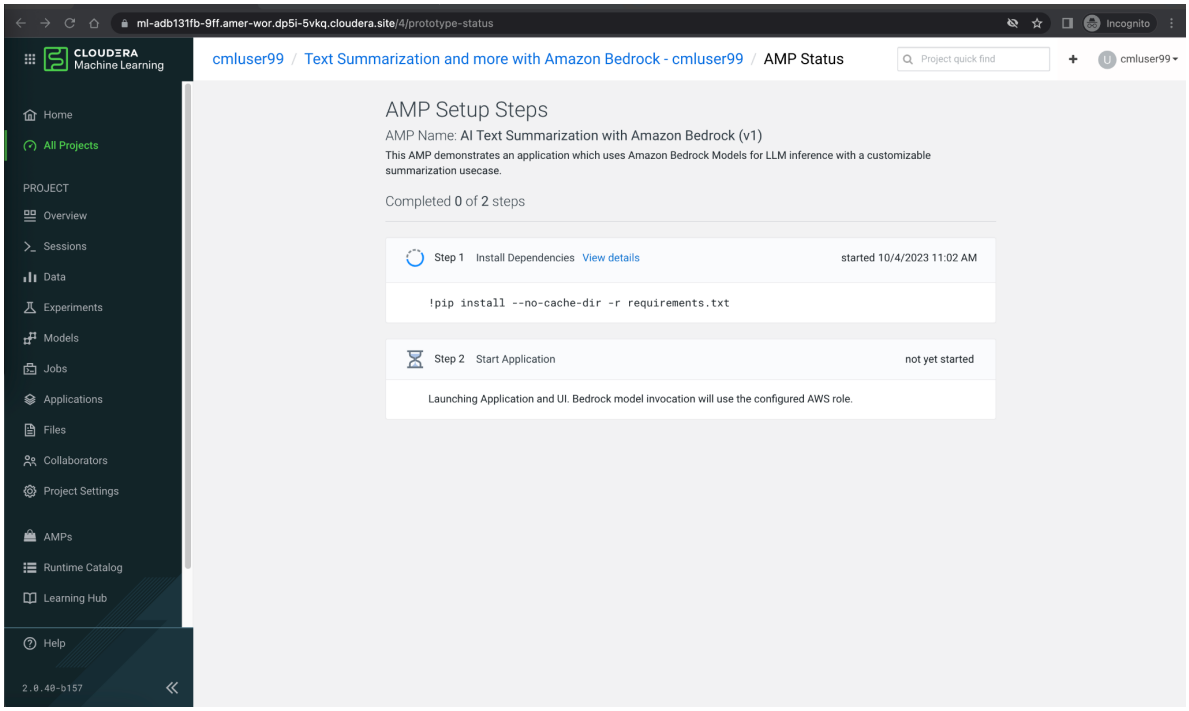
3. Click on the **Configure Project** button



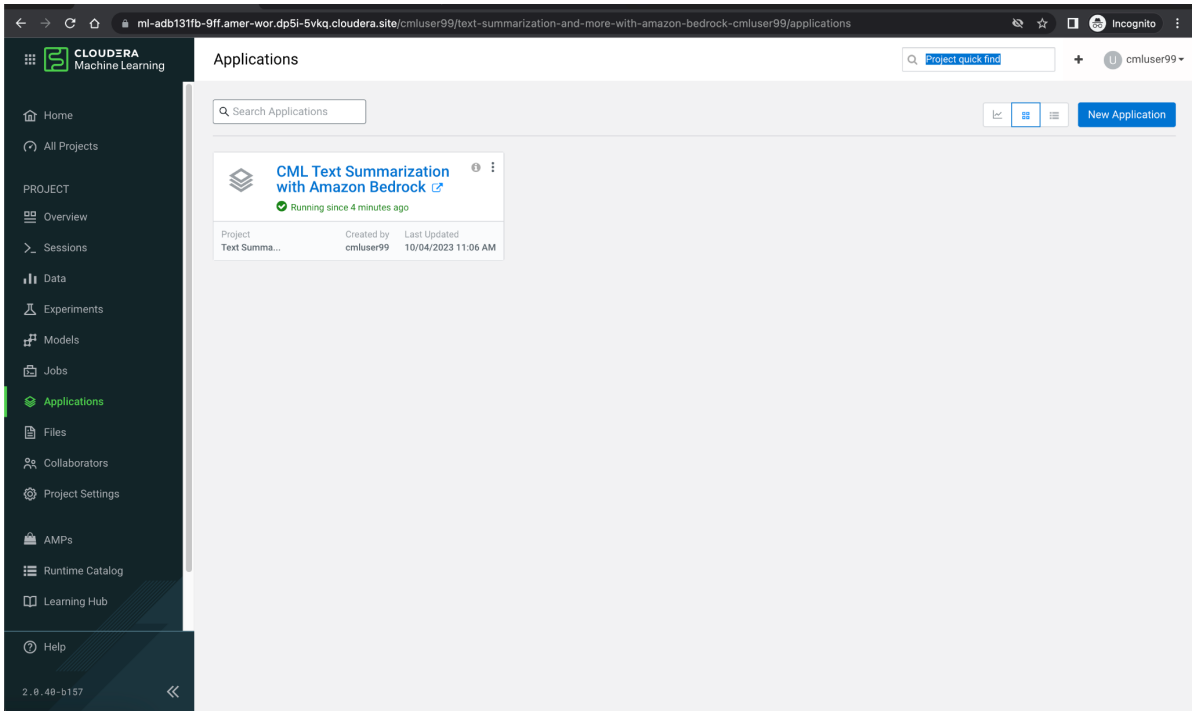
4. Populate the Environment Variables:
    - **AWS_DEFAULT_REGION** = us-east-1
    - **AWS_ACCESS_KEY_ID** = AKIA6I6SNFMLB7UQIFWL
    - **AWS_SECRET_ACCESS_KEY** = 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
5. Click on the **Launch Project** button



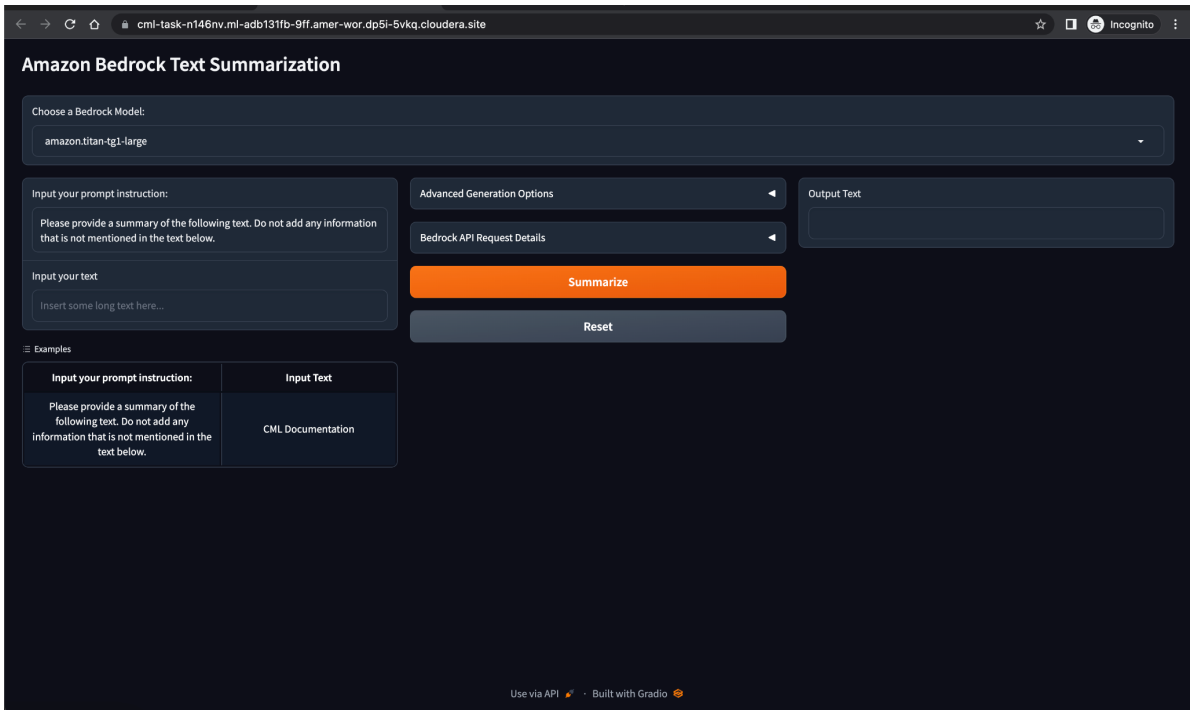
This will go through a couple of steps to complete. This may take a few minutes.

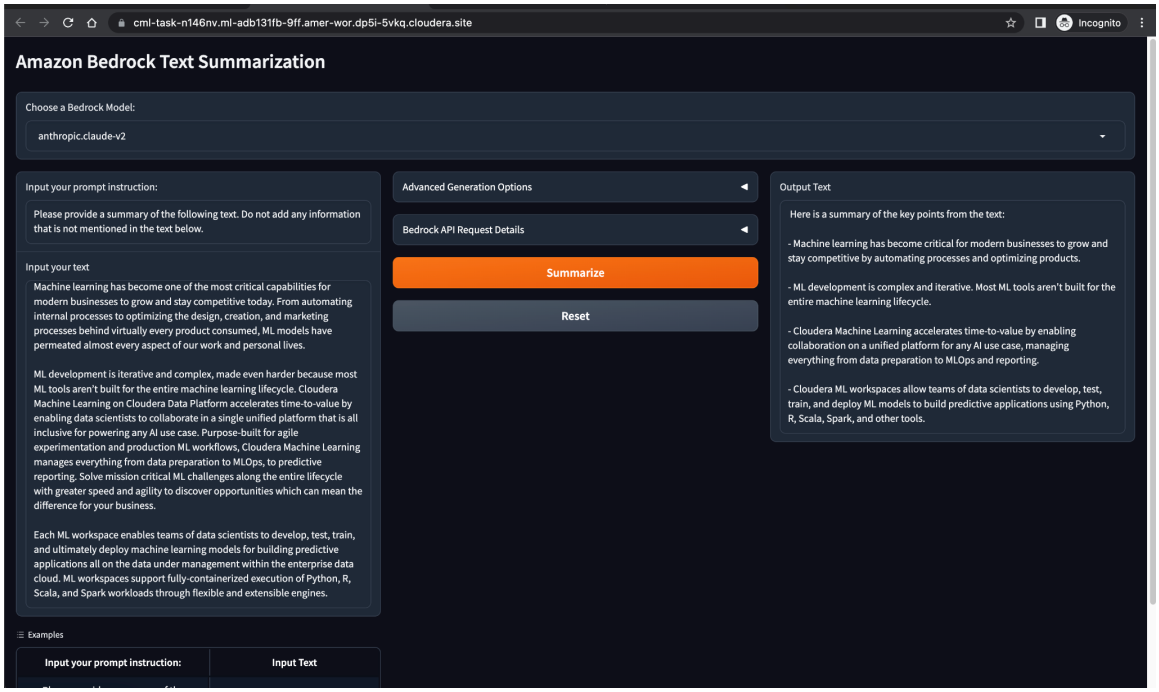


6. When complete, go to the Applications section by clicking on the menu item on the left.



7. Feel free to skip, or test as in the previous lab.





All Done!

THANK YOU!

APPENDIX

AMP Setup
You must have an AWS account with access to Bedrock and the following environment variables defined with your corresponding account details.

AWS_DEFAULT_REGION
AWS_ACCESS_KEY_ID
AWS_SECRET_ACCESS_KEY
If you would like to use a federated role, you can set up BEDROCK_ASSUME_ROLE in the Project Environment variables after AMP launch. (Remember to restart the configured CML Application as well.)

Enable AWS IAM permissions for Bedrock
The AWS identity you set here must have sufficient AWS IAM permissions to call the Amazon Bedrock service.

For example, To grant full bedrock access to your identity, you can:

Open the AWS IAM Console
Find your Role
Select Add Permissions > Create Inline Policy to attach new inline permissions, open the JSON editor and paste in the below example policy:

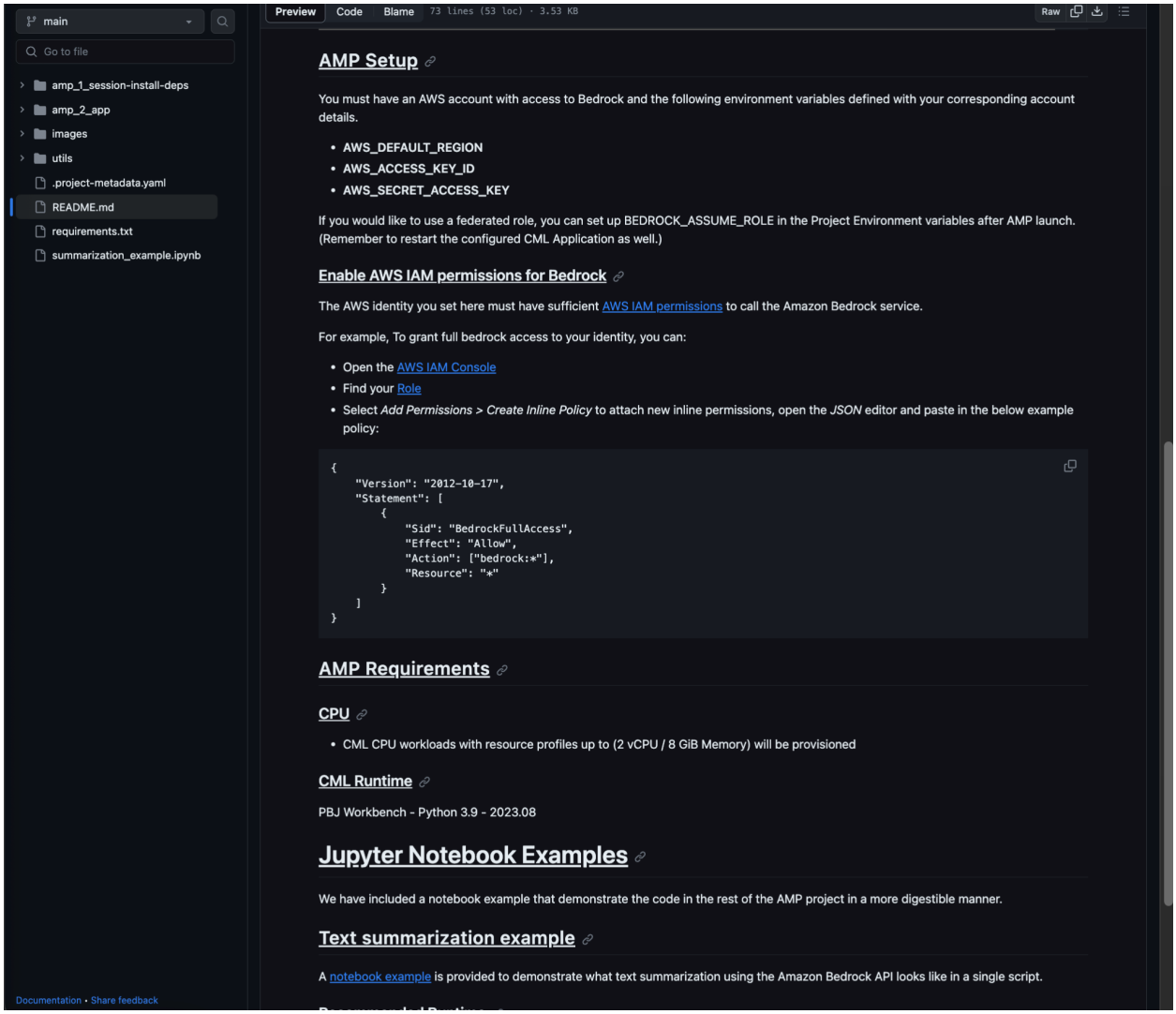
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "BedrockFullAccess",
            "Effect": "Allow",
            "Action": ["bedrock:*"],
            "Resource": "*"
        }
    ]
}
```

AMP Requirements
CPU

- CML CPU workloads with resource profiles up to (2 vCPU / 8 GiB Memory) will be provisioned

CML Runtime

- PBJ Workbench - Python 3.9 - 2023.08


main
Go to file
amp_1_session-install-deps
amp_2_app
images
utils
.project-metadata.yaml
README.md
requirements.txt
summarization_example.ipynb
Documentation • Share feedback
Preview
Code
Blame
73 lines (53 loc) · 3.53 KB
Raw
AMP Setup
You must have an AWS account with access to Bedrock and the following environment variables defined with your corresponding account details.
AWS_DEFAULT_REGION
AWS_ACCESS_KEY_ID
AWS_SECRET_ACCESS_KEY
If you would like to use a federated role, you can set up BEDROCK_ASSUME_ROLE in the Project Environment variables after AMP launch. (Remember to restart the configured CML Application as well.)
Enable AWS IAM permissions for Bedrock
The AWS identity you set here must have sufficient AWS IAM permissions to call the Amazon Bedrock service.
For example, To grant full bedrock access to your identity, you can:
Open the AWS IAM Console
Find your Role
Select Add Permissions > Create Inline Policy to attach new inline permissions, open the JSON editor and paste in the below example policy:
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "BedrockFullAccess",
"Effect": "Allow",
"Action": ["bedrock:*"],
"Resource": "*"
}
]
}
AMP Requirements
CPU
CML CPU workloads with resource profiles up to (2 vCPU / 8 GiB Memory) will be provisioned
CML Runtime
PBJ Workbench - Python 3.9 - 2023.08
Jupyter Notebook Examples
We have included a notebook example that demonstrate the code in the rest of the AMP project in a more digestible manner.
Text summarization example
A notebook example is provided to demonstrate what text summarization using the Amazon Bedrock API looks like in a single script.