Create and Run ML Pipelines with Vertex Pipelines

Activate Cloud Shell

Cloud Shell is a virtual machine that contains development tools. It offers a persistent 5-GB home directory and runs on Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources. `gcloud` is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab completion.

1. Click the **Activate Cloud Shell** button (>_) at the top right of the console.

2. Click **Continue**.
   It takes a few moments to provision and connect to the environment. When you are connected, you are also authenticated, and the project is set to your *PROJECT_ID*.

Sample commands

- List the active account name:

  ```
  gcloud auth list
  ```

  Copied!

  content_copy

  (Output)

  ```
  Credentialed accounts:
   - <myaccount>@<mydomain>.com (active)
  ```

  (Example output)

  ```
  Credentialed accounts:
   - google1623327_student@qwiklabs.net
  ```

- List the project ID:

  ```
  gcloud config list project
  ```

  Copied!

  content_copy

  (Output)

```
[core]
project = <project_ID>
```

(Example output)

```
[core]
project = qwiklabs-gcp-44776a13dea667a6
```

Note: Full documentation of **gcloud** is available in the gcloud CLI overview guide.

Task 1. Cloud environment setup

Cloud Shell has a few environment variables, including GOOGLE_CLOUD_PROJECT which contains the name of our current Cloud project. We use this in various places throughout this lab. You can see it by running:

```
echo $GOOGLE_CLOUD_PROJECT
```

Copied!

content_copy

Enable APIs

- In later steps you see where these services are needed (and why), but to begin, run this command to give your project access to the Compute Engine, Container Registry, and Vertex AI services:

```
gcloud services enable compute.googleapis.com         \
                       containerregistry.googleapis.com  \
                       aiplatform.googleapis.com  \
                       cloudbuild.googleapis.com \
                       cloudfunctions.googleapis.com
```

Copied!

content_copy

This should produce a successful message similar to this one:

```
Operation "operations/acf.cc11852d-40af-47ad-9d59-477a12847c9e"
finished successfully.
```

Task 2. Create a Cloud Storage bucket

To run a training job on Vertex AI, you need a storage bucket in which to store your saved model assets. The bucket must be regional. These instructions specify US-central, but you can use another region (just replace it throughout this lab).

1. To create a bucket, in the Cloud Shell terminal, run the following command:

```
BUCKET_NAME=gs://$GOOGLE_CLOUD_PROJECT-bucket
gsutil mb -l us-central1 $BUCKET_NAME
```

Copied!

content_copy

2. Grant access to this bucket to your compute service account:

```
gcloud projects describe $GOOGLE_CLOUD_PROJECT > project-info.txt
PROJECT_NUM=$(cat project-info.txt | sed -nre 's:.*projectNumber\:
(.*):\1:p')
SVC_ACCOUNT="${PROJECT_NUM//\'/}-compute@developer.gserviceaccount.com"
gcloud projects add-iam-policy-binding $GOOGLE_CLOUD_PROJECT --member
serviceAccount:$SVC_ACCOUNT --role roles/storage.objectAdmin
```

Copied!

content_copy

This ensures that Vertex Pipelines has the necessary permissions to write files to this bucket.

Task 3. Enable the Recommended APIs

1. In the Google Cloud console, in the **Navigation menu** (≡), click **Vertex AI > Dashboard**.
2. Click **Enable All Recommended API**.

Task 4. Launch a Vertex AI Notebooks instance

1. In the Google Cloud Console, on the **Navigation Menu**, click **Vertex AI > Workbench**.

2. On the User-Managed Notebooks page, click **CREATE NEW**, select **TensorFlow Enterprise 2.11 (Intel® MKL-DNN/MKL)**.

3. In the **New notebook** instance dialog, confirm the name of the deep learning VM, if you don’t want to change the region and zone, leave all settings as they are and then click **Create**. The new VM will take 2-3 minutes to start.

4. Click **Open JupyterLab**.
   A JupyterLab window will open in a new tab.

5. If you see “Build recommended” pop up, click **Build**. If you see the build failed, ignore it.

Task 5. Clone a course repo within your Vertex AI Notebooks instance

To clone the training-data-analyst notebook in your JupyterLab instance:

1. In JupyterLab, to open a new terminal, click the **Terminal** icon.

2. At the command-line prompt, run the following command:

   ```
   git clone https://github.com/GoogleCloudPlatform/training-data-analyst
   ```

   Copied!

   content_copy

3. To confirm that you have cloned the repository, double-click on the training-data-analyst directory and ensure that you can see its contents.
   The files for all the Jupyter notebook-based labs throughout this course are available in this directory.

Task 6. Create and Run ML Pipelines with Vertex Pipelines

1. In the notebook interface, navigate to **training-data-analyst > courses > machine_learning > deepdive2 > machine_learning_in_the_enterprise > labs**, and open **pipelines_intro_kfp.ipynb**.

2. In the notebook interface, click **Edit > Clear All Outputs**.

3. Carefully read through the notebook instructions and fill in lines marked with #TODO where you need to complete the code.

Tip: To run the current cell, click the cell and press SHIFT+ENTER. Other cell commands are listed in the notebook UI under **Run**.

- Hints may also be provided for the tasks to guide you along. Highlight the text to read the hints (they are in white text).
- If you need more help, look at the complete solution at **training-data-analyst > courses > machine_learning > deepdive2 > machine_learning_in_the_enterprise > solutions**, and open **pipelines_intro_kfp.ipynb**