ML Pipelines on Google Cloud

Course ·
59% complete
Professional Machine Learning Engineer Certification Learning Path navigate_next ML Pipelines on Google Cloud navigate_next Continuous Training with multiple SDKs, KubeFlow & AI Platform Pipelines

Continuous Training with TensorFlow, PyTorch, XGBoost, and Scikit Learn Models with Kubeflow and AI Platform Pipelines

2 hours Free

Overview

In this lab, we will create containerized training applications for ML models in TensorFlow, PyTorch, XGBoost, and Scikit-learn. Will will then use these images as ops in a KubeFlow pipeline and train multiple models in parallel. We will then set up recurring runs of our KubeFlow pipeline in the UI.

Objectives

- Create the training script
- Package training script into a Docker Image
- Build and push training image to Google Cloud Container Registry
- Build a Kubeflow pipeline that queries BigQuery to create training/validation splits and export results as sharded CSV files in GCS
- Launch AI Platform training jobs with the four containerized training applications, using the exported CSV data as input

Setup and requirements

For each lab, you get a new Google Cloud project and set of resources for a fixed time at no cost.

1. Sign in to Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, `1:15:00`), and make sure you can finish within that time.
   There is no pause feature. You can restart if needed, but you have to start at the beginning.
3. When ready, click **Start lab**.
4. Note your lab credentials (**Username** and **Password**). You will use them to sign in to the Google Cloud Console.

5. Click **Open Google Console**.
6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.
   If you use other credentials, you'll receive errors or **incur charges**.
7. Accept the terms and skip the recovery resource page.

Note: Do not click **End Lab** unless you have finished the lab or want to restart it. This clears your work and removes the project.

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

(Output)

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

(Example output)

Credentialed accounts: - google1623327_student@qwiklabs.net

- List the project ID:

gcloud config list project

(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. Create a Cloud Storage bucket

- In Cloud Shell, run the following command to create a Cloud Storage bucket. For the sake of simplicity, you will make this bucket public.

PROJECT_ID=`gcloud config list --format 'value(core.project)'` gsutil mb -p $PROJECT_ID gs://$PROJECT_ID gsutil acl ch -u AllUsers:R gs://$PROJECT_ID

Task 2. Enable Cloud services

1. Next, execute the following commands to enable the required Cloud services:

gcloud services enable \ cloudbuild.googleapis.com \ container.googleapis.com \ cloudresourcemanager.googleapis.com \ iam.googleapis.com \ containerregistry.googleapis.com \ containeranalysis.googleapis.com \ ml.googleapis.com \ dataflow.googleapis.com

2. Add the Editor permission for your Cloud Build service account:

PROJECT_NUMBER=$(gcloud projects describe $PROJECT_ID --format="value(projectNumber)") CLOUD_BUILD_SERVICE_ACCOUNT="${PROJECT_NUMBER}@cloudbuild.gserviceaccount.com" gcloud projects add-iam-policy-binding $PROJECT_ID \ --member serviceAccount:$CLOUD_BUILD_SERVICE_ACCOUNT \ --role roles/editor

Click **Check my progress** to verify the objective. Add the Editor permission for Cloud Build service account

Task 3. Create an instance of AI Platform Pipelines

1. In the Google Cloud Console, on the Navigation menu (☰), scroll down to **AI Platform** and pin the section for easier access later in the lab.
2. Navigate to **AI Platform** > **Pipelines**.
3. Then click **New Instance**.
4. Click **Configure**.
5. To create cluster select **Zone** as then check **Allow access to the following Cloud APIs**, leave the name as is, and then click **Create New Cluster**.

Note: The cluster creation will take 3 - 5 minutes. You need to wait until this step completes before you proceed to the next step.

6. Scroll to the bottom of the page, accept the marketplace terms, and click **Deploy**. You will see the individual services of KFP deployed to your GKE cluster. Wait for the deployment to finish before proceeding to the next task.
7. In **Cloud Shell**, run the following to configure kubectl command line access

gcloud container clusters get-credentials cluster-1 --zone {{{project_0.default_zone|place_holder_text}}} --project {{{project_0.project_id|place_holder_text}}}

8. In **Cloud Shell**, run the following to get the ENDPOINT of your KFP deployment

kubectl describe configmap inverse-proxy-config | grep googleusercontent.com Important: In task 6, you will need to set the endpoint for your KFP in one of the cells in your notebook. Remember to use the above output as your **ENDPOINT**.

https:// 5f3aa6770af7e77f-dot-us-central1.pipelines.googleusercontent.com

Click **Check my progress** to verify the objective. Creating an instance of AI Platform Pipelines

Task 4. Create an instance of Vertex AI Platform Notebooks

An instance of Vertex AI Platform Notebooks is used as a primary experimentation/development workbench. The instance is configured using a custom container image that includes all Python packages required for this lab.

1. In the Cloud Console, on the **Navigation menu**, click **Vertex AI > Workbench**.
2. Click **ENABLE NOTEBOOKS API** if it is not enabled yet.

Note: In this lab, the Vertex AI Notebook instance is created for you at the start time of the lab. **Note:** It may take 5 minutes for the notebook instance to appear.

3. Please refresh the page, once you see **Include legacy instances** checkbox, enable it to see your instance.

Note: Wait for the instance to become available before proceeding to the next step.

3. When the instance appears, click on **REGISTER ALL** to register the notebook with Notebooks API.

Note: If the registration fails, wait for a minute and try again.

4. Click the **Open Jupyterlab** link.

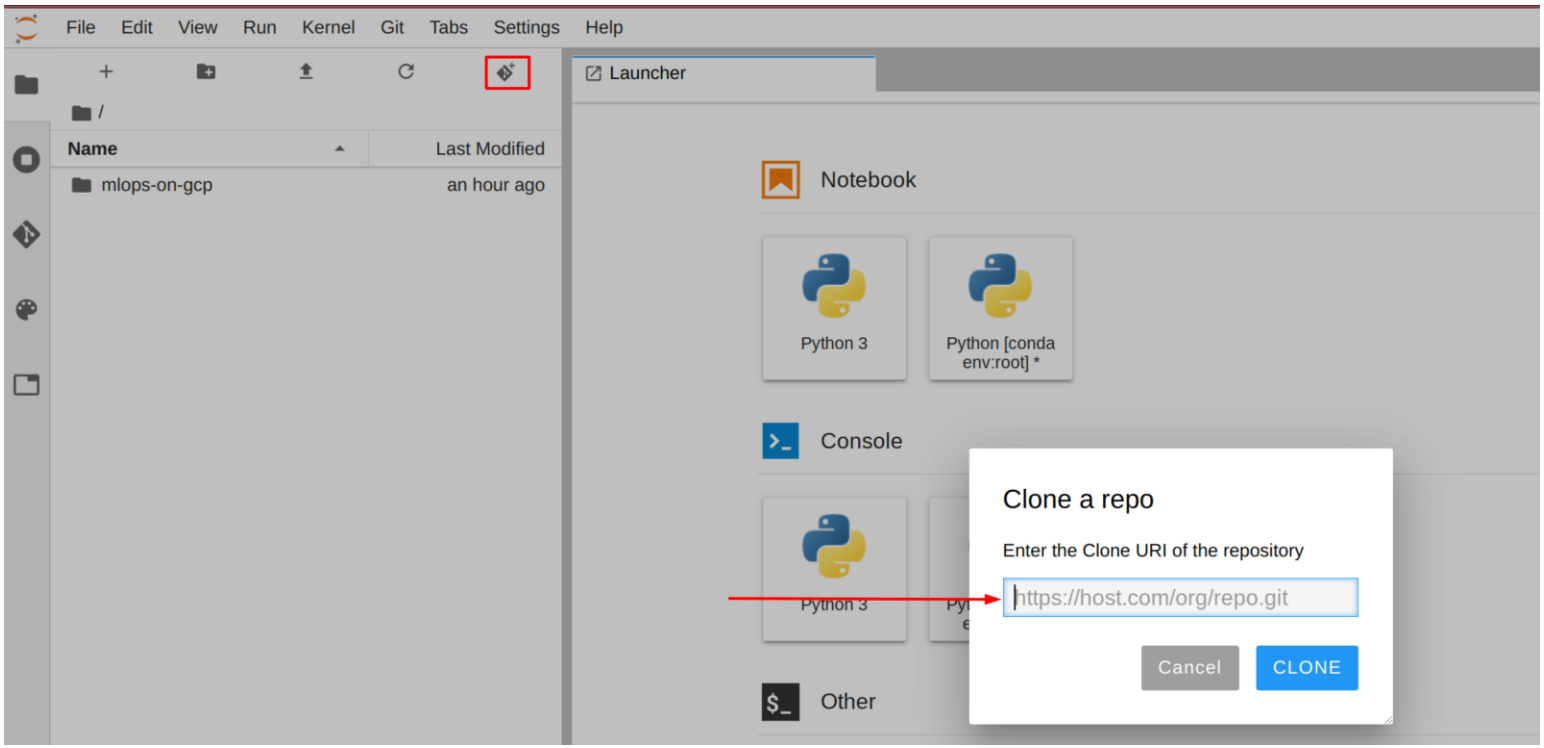
Click **Check my progress** to verify the objective. Create an instance of AI Platform Notebooks

Task 5. Clone the example repo within your AI Platform Notebooks instance

To clone the `mlops-on-gcp` notebook in your JupyterLab instance:

1. In JupyterLab, click the **Terminal** icon to open a new terminal.
2. At the command-line prompt, type in the following command and press Enter:

   git clone https://github.com/GoogleCloudPlatform/mlops-on-gcp **Note:** If the cloned repo does not appear in the JupyterLab UI, you can use the top line menu and under **Git** > **Clone a repository**, clone the repo (https://github.com/GoogleCloudPlatform/mlops-on-gcp) using the UI.



3. Confirm that you have cloned the repository by double clicking on the `mlops-on-gcp` directory and ensuring 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. Navigate to the lab notebook

1. In JupyterLab UI, navigate to `mlops-on-gcp/continuous_training/kubeflow/labs` and open `multiple_frameworks_lab.ipynb`.
2. Clear all the cells in the notebook (look for the Clear button on the notebook toolbar) and then Run the cells one by one. Note the some cells have a #TODO for you to write the code before you run the cell.
3. When prompted, come back to these instructions to check your progress.

If you need more help, you may take a look at the complete solution by navigating to `mlops-on-gcp/continuous_training/kubeflow/solutions` open `multiple_frameworks_kubeflow.ipynb`.

Task 7. Run your training job in the cloud

Test completed tasks - Create a Cloud Storage Bucket, Bigquery dataset and table

Click **Check my progress** to verify the objective. Create a Cloud Storage Bucket, Bigquery dataset and table

Test completed tasks - Build and push training images to Google Cloud Container Registry

Click **Check my progress** to verify the objective. Build and push training images to Google Cloud Container Registry

Test completed tasks - Deploy your KubeFlow Pipeline

Click **Check my progress** to verify the objective. Deploy your KubeFlow Pipeline

Test completed tasks - Create Pipeline Runs

Click **Check my progress** to verify the objective. Create Pipeline Runs

Congratulations!

In this lab you've learned how to develop, package as a docker image, and run on AI Platform Training to training application.

End your lab

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you've used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.

NOTES:

BQ Dataset is NOT available, see how to do that -- 75% passed