ML Pipelines on Google Cloud

Course ·
38% complete
Professional Machine Learning Engineer Certification Learning Path navigate_next ML Pipelines on Google Cloud navigate_next Custom components and CI/CD for TFX pipelines

CI/CD for a TFX pipeline

2 hours Free

Overview

In this lab, you will walk through authoring a Cloud Build CI/CD workflow that automatically builds and deploys the same TFX pipeline from lab-02.ipynb. You will also integrate your workflow with GitHub by setting up a trigger that starts the workflow when a new tag is applied to the GitHub repo hosting the pipeline's code.

Objectives

- Develop a CI/CD workflow with Cloud Build to build and deploy a machine learning pipeline.
- Integrate with Github to trigger workflows with pipeline source repository changes.

Setup

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.

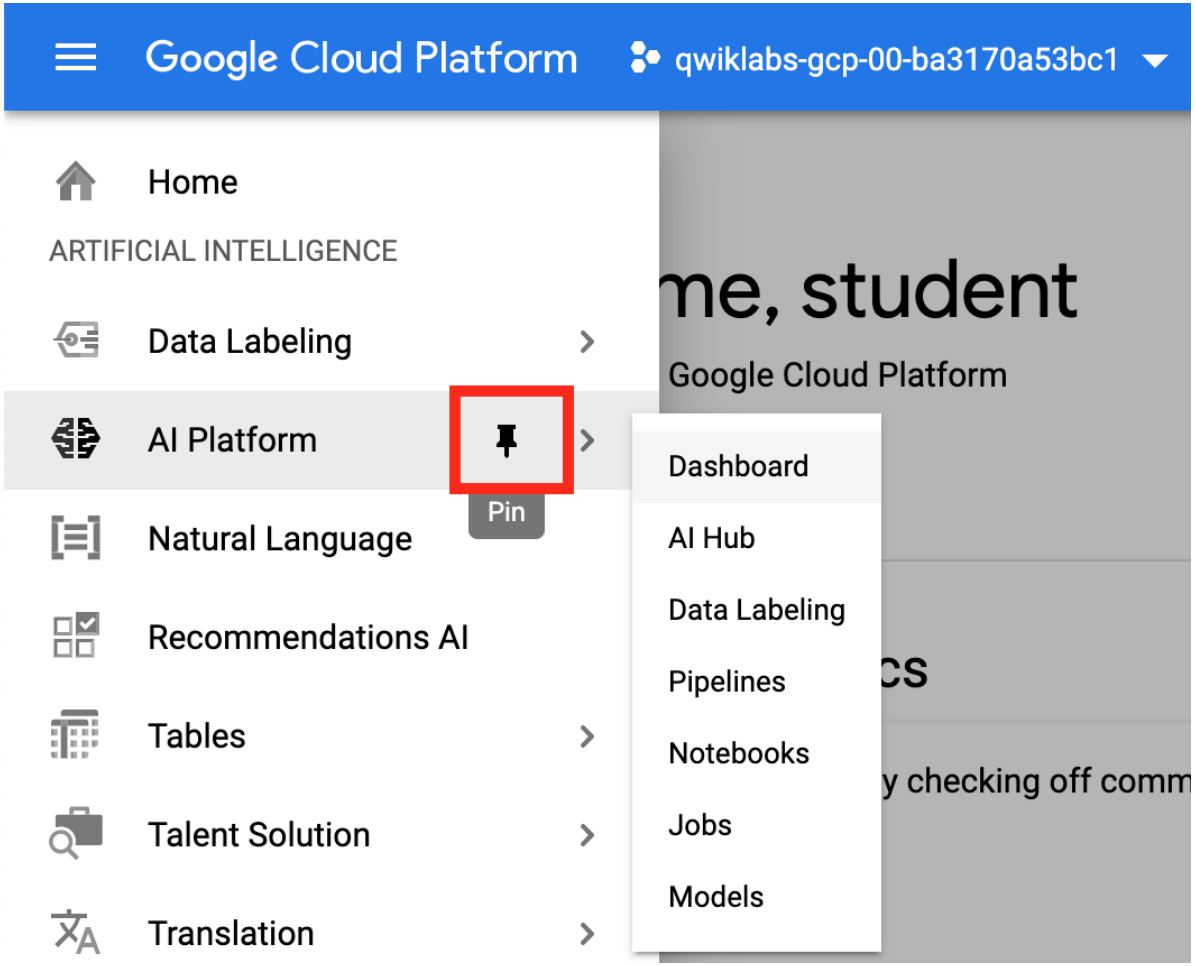
Task 1. Create an instance of AI Platform Pipelines

In this task, you deploy Kubeflow Pipelines as a Kuberenetes App, which are solutions with simple click to deploy to Google Kubernetes Engine and that have the flexibility to deploy to Kubernetes clusters on-premises or in third-party clouds. You will see Kubeflow Pipelines integrated into your Google Cloud environment as **AI Platform Pipelines**. If interested, learn more about Kubeflow Pipelines in the Introduction documentation during installation steps.

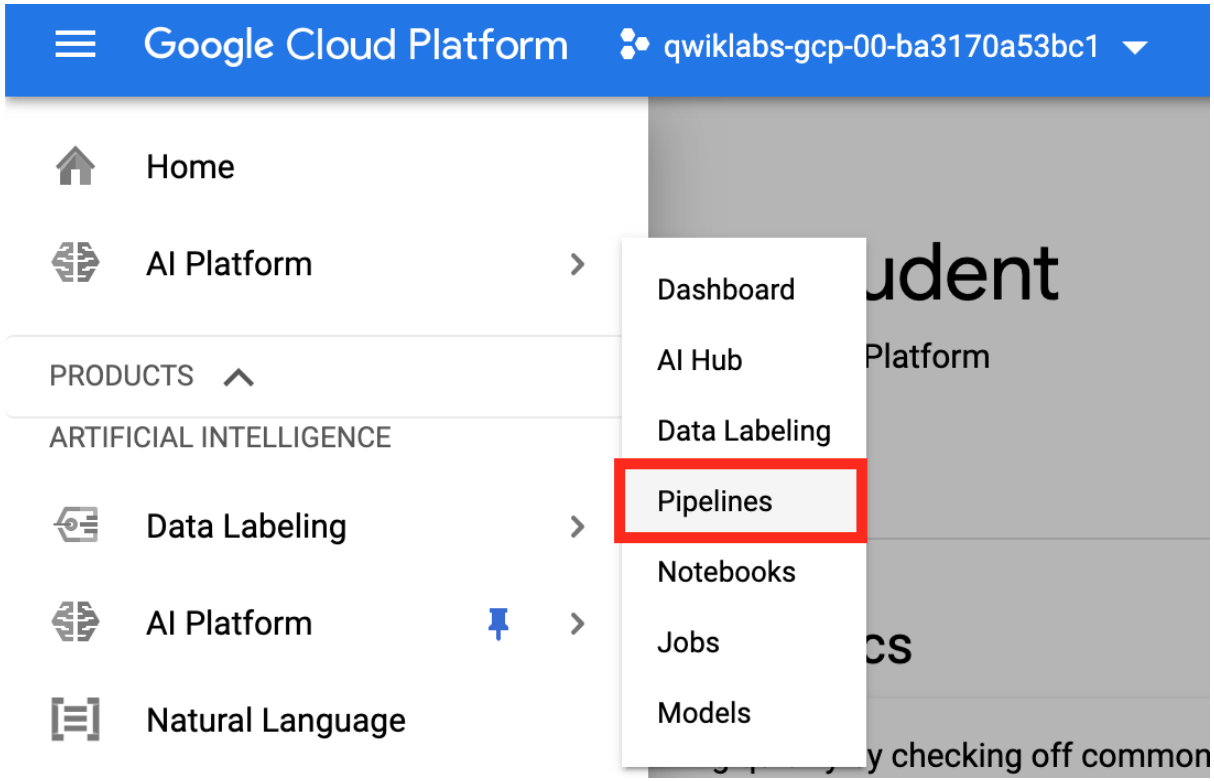
1. Run the below commands to initialize the lab resources:

gsutil cp gs://cloud-training/TFX-Script/tfx-init.sh . sh tfx-init.sh

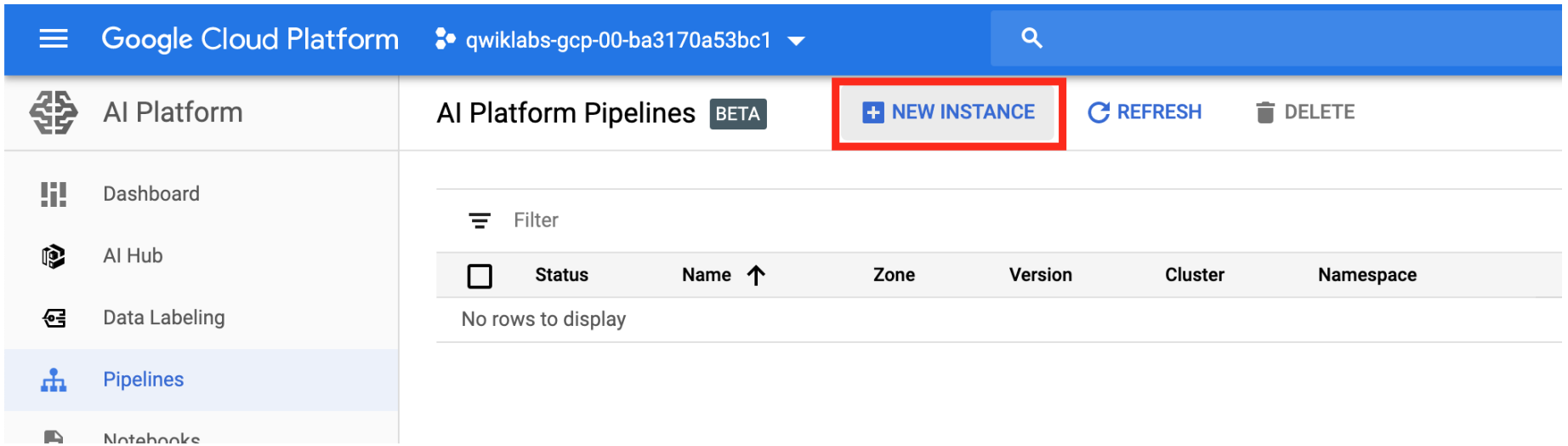
2. In the Google Cloud Console, on the Navigation menu, scroll down to **AI Platform** and pin the section for easier access later in the lab.



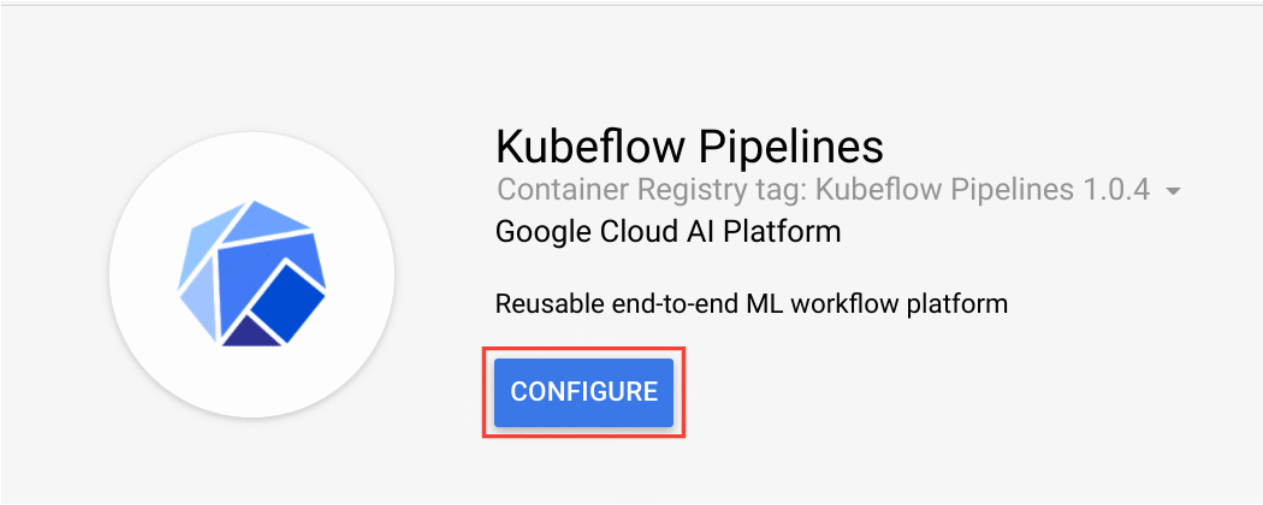
3. Navigate to **AI Platform** > **Pipelines**.



4. Then click **New Instance**.



5. Click **Configure**.



6. 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.

7. 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.
8. 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}}}

9. 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 a later task, 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**.

NOTES:

client = kfp.Client(host='https:// 614a94c744270b6-dot-us-central1.pipelines.googleusercontent.com')

Endpoint: https://614a94c744270b6-dot-us-central1.pipelines.googleusercontent.com

Storage: gs://qwiklabs-gcp-01-eef1db9f1e62-kubeflowpipelines-default/

Service acct: tfx-tuner-caip-service-account@qwiklabs-gcp-01-eef1db9f1e62.iam.gserviceaccount.com

/***** Actual real value***/ use this in Edit Trigger

PIPELINE_NAME='tfx_covertype_continuous_training'

MODEL_NAME='tfx_covertype_classifier'

DATA_ROOT_URI='gs://cloud-training/OCBL203/workshop-datasets'

TAG_NAME='test'

TFX_IMAGE_NAME='lab-03-tfx-image'

PIPELINE_FOLDER='pipeline'

PIPELINE_DSL='runner.py'

RUNTIME_VERSION='2.3'

PYTHON_VERSION='3.7'

USE_KFP_SA='False'

ENABLE_TUNING='True'

// END OF NOTES

Enable Secret Manager API

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

Task 2. Access AI Platform Notebooks

To launch AI Platform Notebooks:

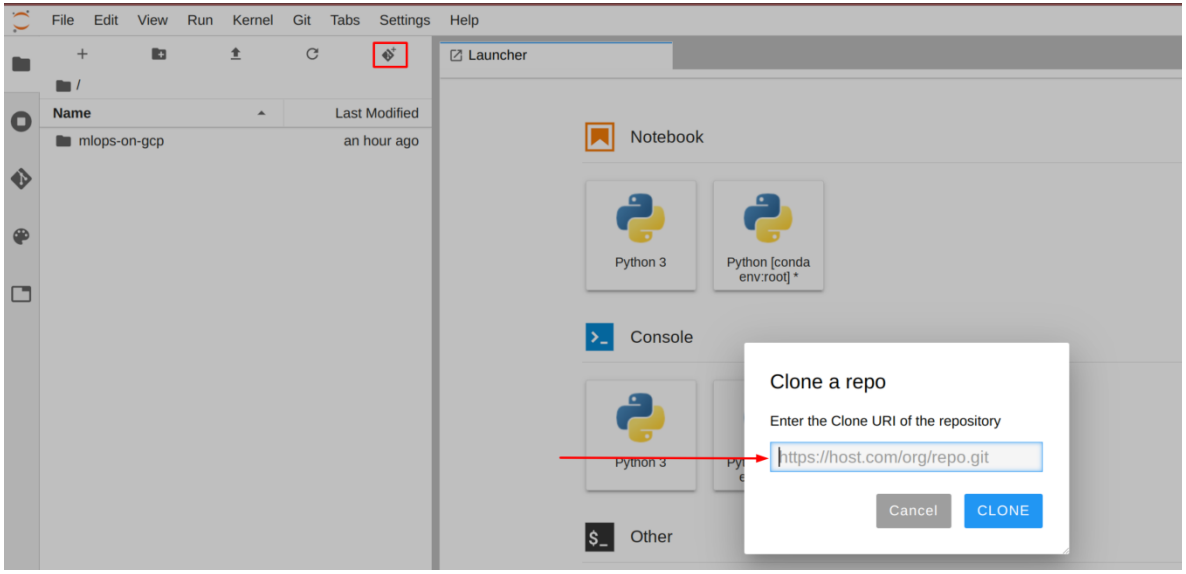
1. Click on the **Navigation Menu** and navigate to **AI Platform**, then to **Workbench**.
2. You should see `tfx-on-googlecloud` notebook preprovisioned for you. If not, wait a few minutes and refresh the page.
3. Click **Open JupyterLab**. A JupyterLab window will open in a new tab.

Task 3. 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.

Navigate to the example notebook

1. In the JupyterLab, open a terminal and execute the following commands:

cd mlops-on-gcp/workshops/tfx-caip-tf23 ./install.sh

2. Now, in AI Platform Notebooks, navigate to `mlops-on-gcp/workshops/tfx-caip-tf23/lab-03-tfx-cicd/labs` and open `lab-03.ipynb`.
3. Clear all the cells in the notebook (look for the Clear button on the notebook toolbar) and then Run the cells one by one.
4. 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/workshops/tfx-caip-tf23/lab-03-tfx-cicd/solutions` open `lab-03.ipynb`.

Task 4. Run your training job in the cloud

Test completed tasks

Click *Check my progress* to verify the objective. Build the image and push it to your project's Container Registry

Test completed task

Click *Check my progress* to verify the objective. Manually trigger CI/CD runs

Test completed tasks

Click *Check my progress* to verify the objective. Create a cloud build trigger

Test completed tasks

Click *Check my progress* to verify the objective. Trigger the build

NOTES:

https://github.com/GoogleCloudPlatform/mlops-on-gcp

Trigger vars

```
|Field|Value|
```

```
|-----|-----|
|Name|[YOUR TRIGGER NAME]|
|Description|[YOUR TRIGGER DESCRIPTION]|
|Event| Tag|
|Source| [YOUR FORK]|
|Tag (regex)|.\*|
|Build Configuration|Cloud Build configuration file (yaml or json)|
|Cloud Build configuration file location|/ workshops/tfx-caip-tf23/lab-03-tfx-cicd/labs/cloudbuild.yaml|
```

Substitution vars

```
|Variable|Value|
|--------|-----|
|_GCP_REGION|[YOUR GCP_REGION]|
|_CUSTOM_SERVICE_ACCOUNT|[YOUR CUSTOM_SERVICE_ACCOUNT]|
|_ENDPOINT|[Your inverting proxy host pipeline ENDPOINT]|
|_TFX_IMAGE_NAME|lab-03-tfx-image|
|_PIPELINE_NAME|tfx_covertype_continuous_training|
|_MODEL_NAME|tfx_covertype_classifier|
|_DATA_ROOT_URI|gs://cloud-training/OCBL203/workshop-datasets|
|_PIPELINE_FOLDER|workshops/tfx-caip-tf23/lab-03-tfx-cicd/labs/pipeline|
|_PIPELINE_DSL|runner.py|
|_PYTHON_VERSION|3.7|
|_RUNTIME_VERSION|2.3|
|_USE_KFP_SA|False|
|_ENABLE_TUNING|True|
```

Congratulations!

In this lab, you walked through authoring a Cloud Build CI/CD workflow that automatically builds and deploys a TFX pipeline. You also integrated your TFX workflow with GitHub by setting up a Cloud Build trigger. In the next lab, you will walk through inspection of TFX metadata and pipeline artifacts created during TFX pipeline runs.

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:

https://cloud.google.com/build/docs/automating-builds/create-manage-triggers - substitution vars for build failed error

https://notebook.community/GoogleCloudPlatform/mlops-on-gcp/workshops/tfx-caip-tf21/lab-03-tfx-cicd/lab-03