

# ML Pipelines on Google Cloud

Course ·

44% complete

[Professional Machine Learning Engineer Certification Learning Path](#) navigate\_next [ML Pipelines on Google Cloud](#) navigate\_next ML Metadata with TFX

## TFX Metadata

2 hours 20 minutes Free

### Overview

In this lab, you will explore TFX pipeline metadata including pipeline and run artifacts. An AI Platform Pipelines instance includes the ML Metadata service. In AI Platform Pipelines, ML Metadata uses MySQL as a database backend and can be accessed using a GRPC server.

### Objectives

Use a GRPC server to access and analyze pipeline artifacts stored in the ML Metadata service of your AI Platform Pipelines instance.

### 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.

## 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>
```

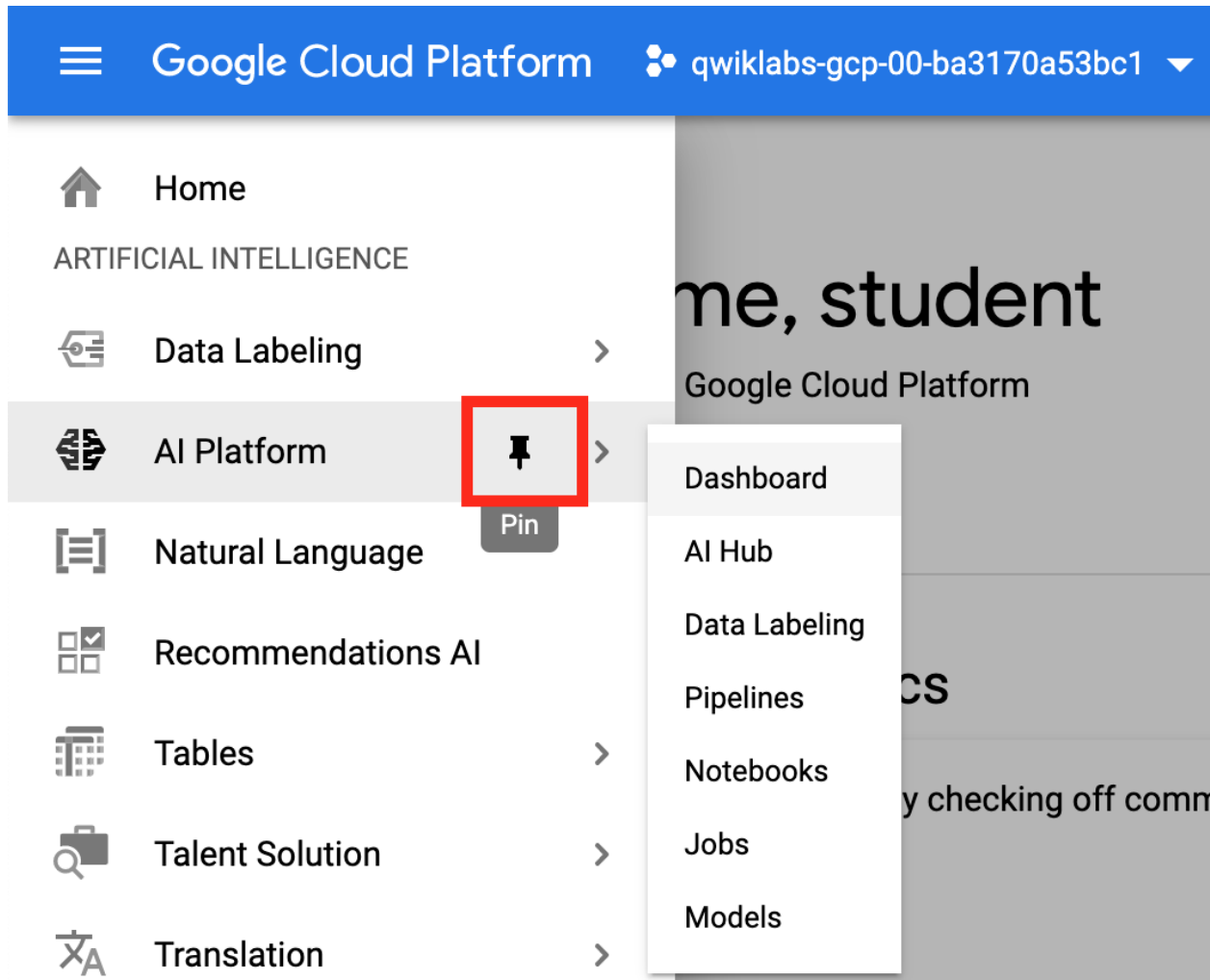
(Example output)

[core] project = qwiklabs-gcp-44776a13dea667a6 **Note:** Full documentation of **gcloud** is available in the [gcloud CLI overview guide](#).

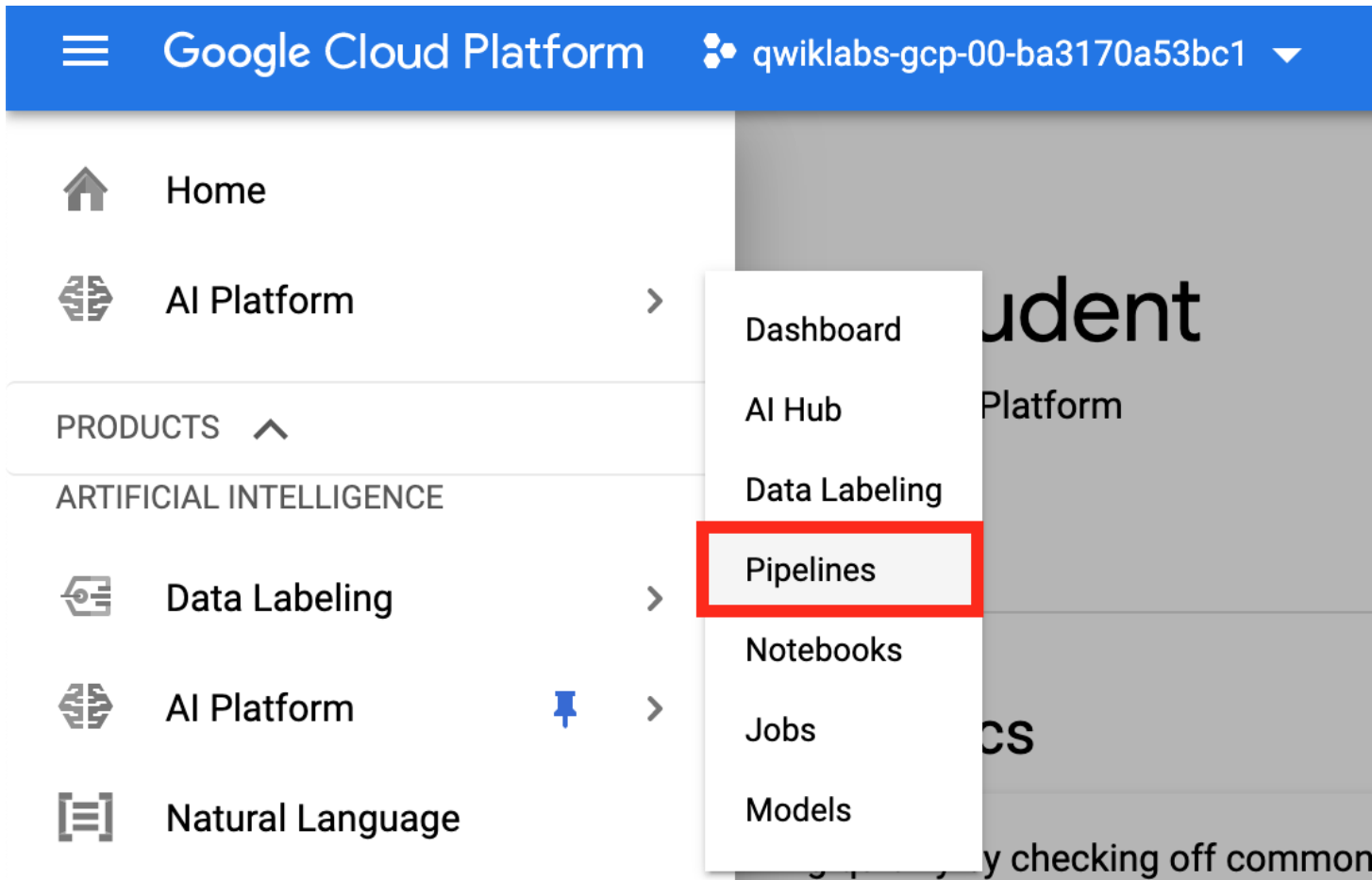
## Task 1. Create an instance of AI Platform Pipelines

In this task, you deploy Kubeflow Pipelines as a Kubernetes 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 to Kubeflow documentation](#) during installation steps.

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**.

Google Cloud Platform qwiklabs-gcp-00-ba3170a53bc1

AI Platform

AI Platform Pipelines BETA

+ NEW INSTANCE REFRESH DELETE

Filter

Status	Name ↑	Zone	Version	Cluster	Namespace
No rows to display					

Dashboard


AI Hub

Data Labeling

Pipelines

Notebooks

4. Click **Configure**.



## Kubeflow Pipelines

Container Registry tag: Kubeflow Pipelines 1.0.4

Google Cloud AI Platform

Reusable end-to-end ML workflow platform

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. **Note:** If the cluster creation fails because of insufficient resources in a certain region/zone, try again with a different zone.

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 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:

<https://3a162736a1835bf8-dot-us-central1.pipelines.googleusercontent.com>

Storage: gs://qwiklabs-gcp-01-f4877dc88a1b-kubeflowpipelines-default/

Service Acct [tfx-tuner-caip-service-account@qwiklabs-gcp-01-f4877dc88a1b.iam.gserviceaccount.com](mailto:tfx-tuner-caip-service-account@qwiklabs-gcp-01-f4877dc88a1b.iam.gserviceaccount.com)

```
env: PIPELINE_NAME=tfx_covertypes_lab_04
env: MODEL_NAME=tfx_covertypes_classifier
env: DATA_ROOT_URI=gs://cloud-training/OCBL203/workshop-datasets
env: KUBEFLOW_TFX_IMAGE=gcr.io/qwiklabs-gcp-01-f4877dc88a1b/tfx_covertypes_lab_04
env: RUNTIME_VERSION=2.3
env: PYTHON_VERSIONS=3.7
env: USE_KFP_SA=False
env: ENABLE_TUNING=True
```

#### //-END of NOTES

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

## Task 2. Access Vertex AI Workbench

To launch AI Platform Workbench:

1. Click on the **Navigation Menu** and navigate to **Vertex AI**, then to **Workbench**.
2. Click on **USER-MANAGED NOTEBOOKS**.

You should see `tfx-on-googlecloud` notebook preprovisioned for you. If not, wait a few minutes and refresh the page.

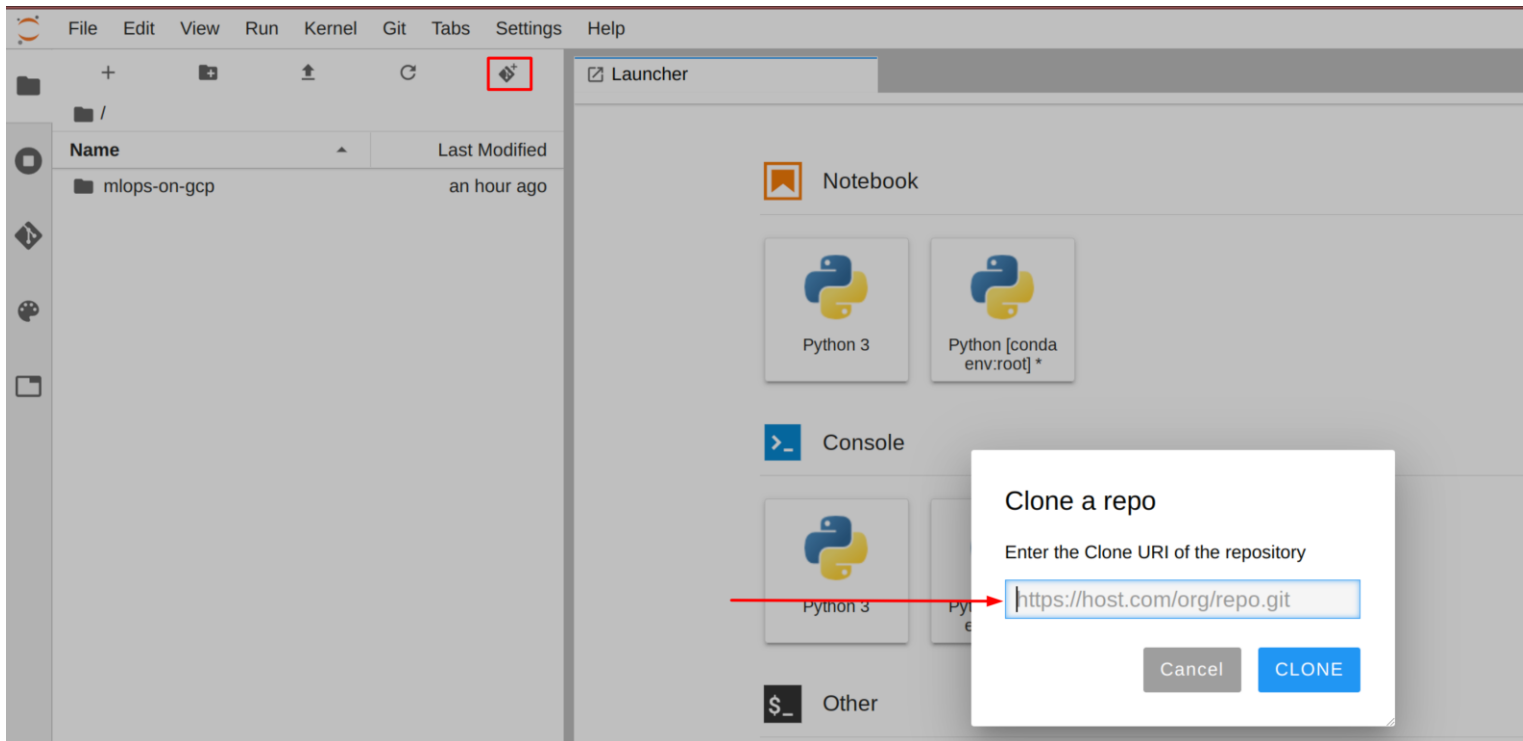
3. Click **Open JupyterLab**. A JupyterLab window will open in a new tab.

## 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 lab notebook

1. From the `mlops-labs/workshops/tfx-caip-tf23` folder execute the `install.sh` script to install TFX and KFP SDKs:

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

2. Now, in AI Platform Notebook, navigate to `mlops-labs/workshops/tfx-caip-tf23/lab-04-tfx-metadata/labs` and open `lab-04.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-04-tfx-metadata/solutions` and open `lab-04.ipynb`.

## Task 4. Run your training job in the cloud

### Test completed tasks - Compile the kubeflow pipeline

Click *Check my progress* to verify the objective. Compile the kubeflow pipeline

### Test completed tasks - Deploy the pipeline package to AI Platform Pipelines

Click *Check my progress* to verify the objective. Deploy the pipeline package to AI Platform Pipelines

## Congratulations!

In this lab, you explored ML metadata and ML artifacts created by TFX pipeline runs using TFX pipeline ResolverNodes.

## 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.

Copyright 2022 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.

- [Overview](#)
- [Setup](#)
- [Task 1. Create an instance of AI Platform Pipelines](#)
- [Task 2. Access Vertex AI Workbench](#)
- [Task 3. Clone the example repo within your AI Platform Notebooks instance](#)
- [Task 4. Run your training job in the cloud](#)
- [Congratulations!](#)
- [End your lab](#)

#### NOTES:

```
023-10-23 13:33:42.309395: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic
library 'libcudart.so.10.1'; dLError: libcudart.so.10.1: cannot open shared object file: No such file or directory;
LD_LIBRARY_PATH: /usr/lo
```