Running ML Pipelines on Kubeflow 2.5

## Overview

In this lab, you learn how to configure and create an AI Platform Pipelines instance. After you configure the AI Platform Pipelines instance, you run an example pipeline and visualize the pipeline graph to see the different metrics, logs, and parameters that are available.

## Objectives

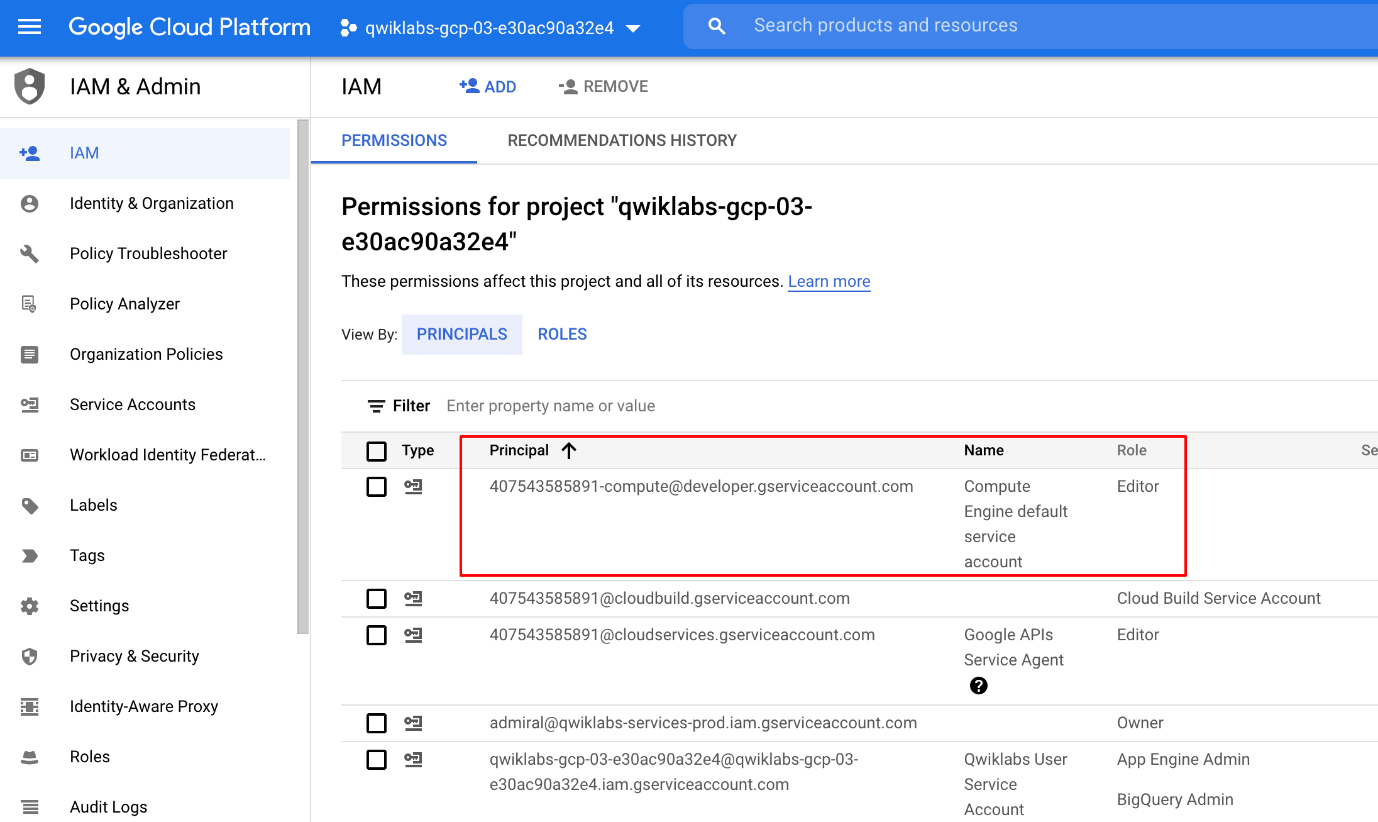
In this lab, you perform the following tasks:

* Create a Kubernetes cluster and configure AI Platform pipelines
* Launch pipelines dashboard
* Create and run an experiment from an example end-to-end ML Pipeline
* Examine and verify the output of each step
* Inspect the pipeline graph, various metrics, logs, charts and parameters

### **Check project permissions**

Before you begin your work on Google Cloud, you need to ensure that your project has the correct permissions within Identity and Access Management (IAM).

1. In the Google Cloud console, on the **Navigation menu** (Navigation menu icon), click **IAM & Admin** > **IAM**.
2. Confirm that the default compute Service Account {project-number}-compute@developer.gserviceaccount.com is present and has the editor role assigned. The account prefix is the project number, which you can find on **Navigation menu** > **Home**.



If the account is not present in IAM or does not have the editor role, follow the steps below to assign the required role.

* In the Google Cloud console, on the **Navigation menu**, click **Home**.
* Copy the project number (e.g. 729328892908).
* On the **Navigation menu**, click **IAM & Admin** > **IAM**.
* At the top of the **IAM** page, click **Add**.
* For **New principals**, type:

{project-number}-compute@developer.gserviceaccount.com

Copied!

content\_copy

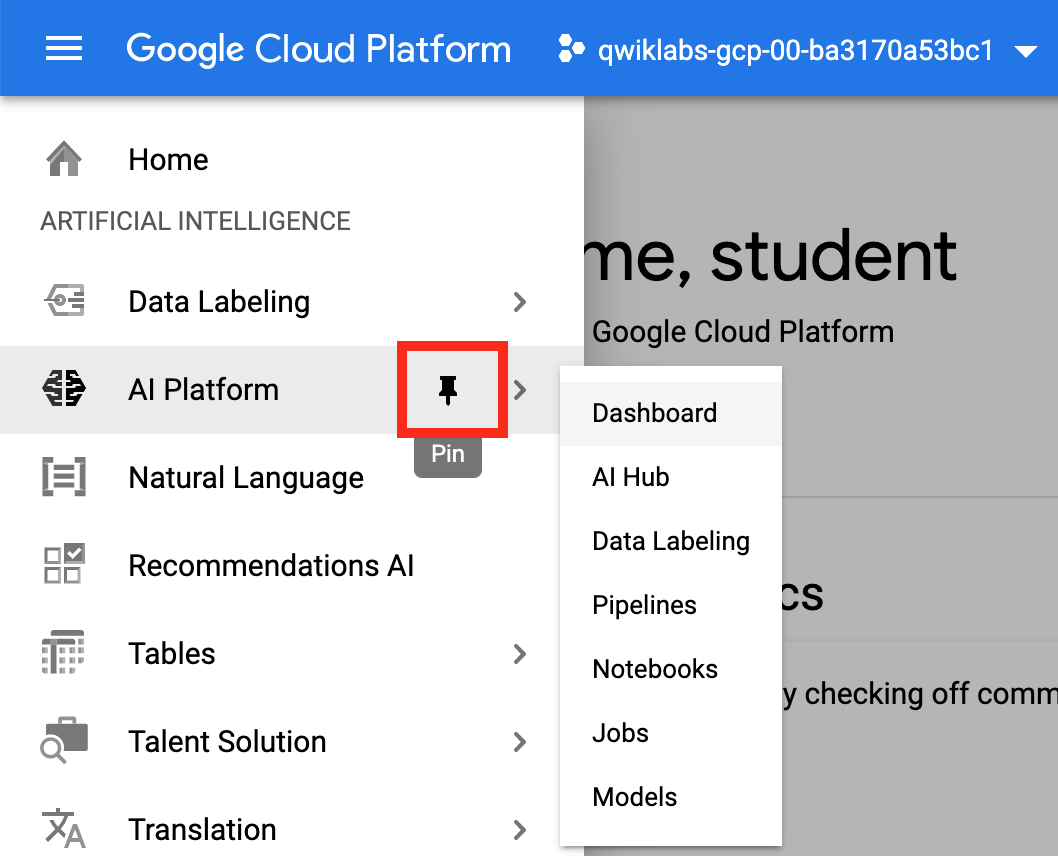
Replace {project-number} with your project number.

* For **Role**, select **Project** (or Basic) > **Editor**. Click **Save**.

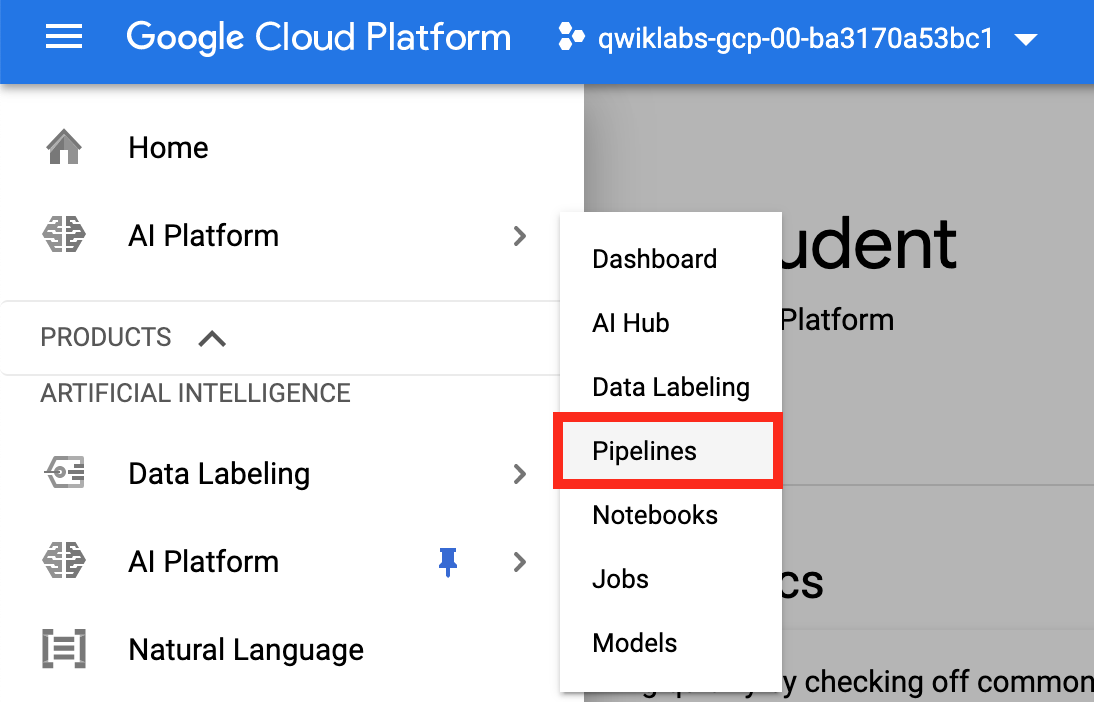
## Task 1. Set up an AI Platform Pipelines instance

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 [documentation](https://www.kubeflow.org/docs/components/pipelines/introduction/) during installation steps.

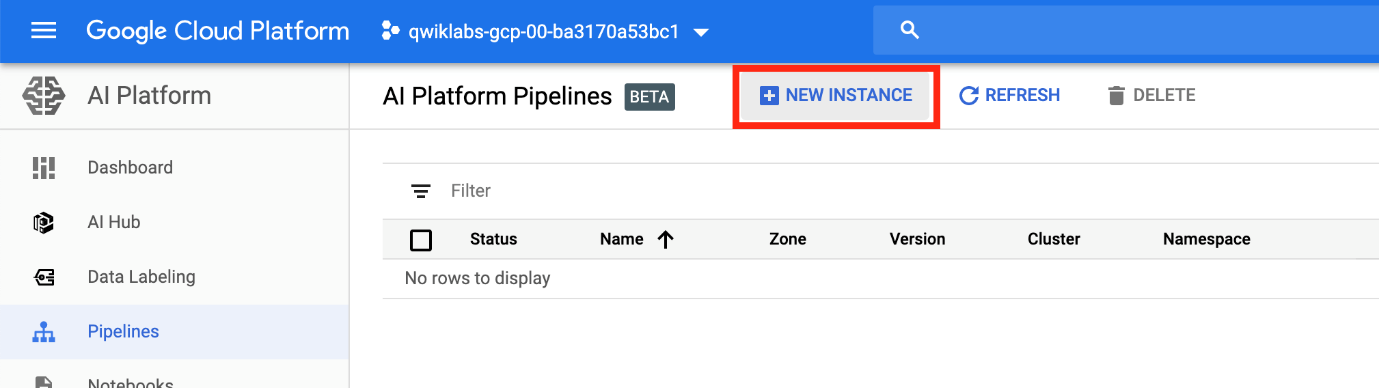
1. From the the Navigation menu, scroll down to **AI Platform** and pin the section for easier access later in the lab.



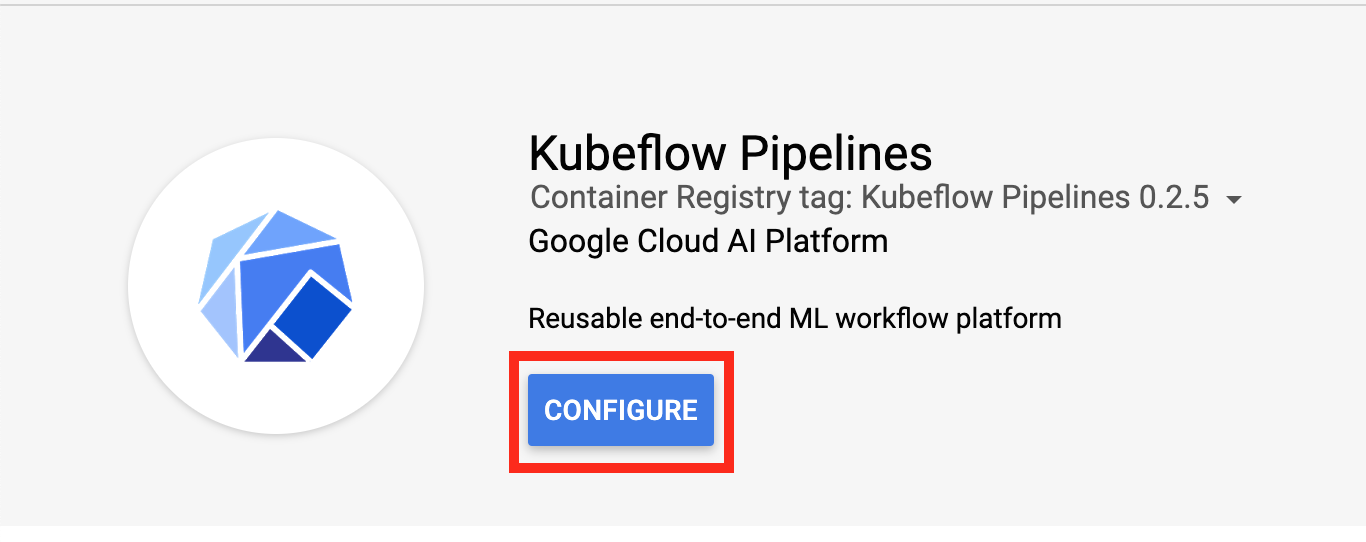
1. Click **Pipelines**.



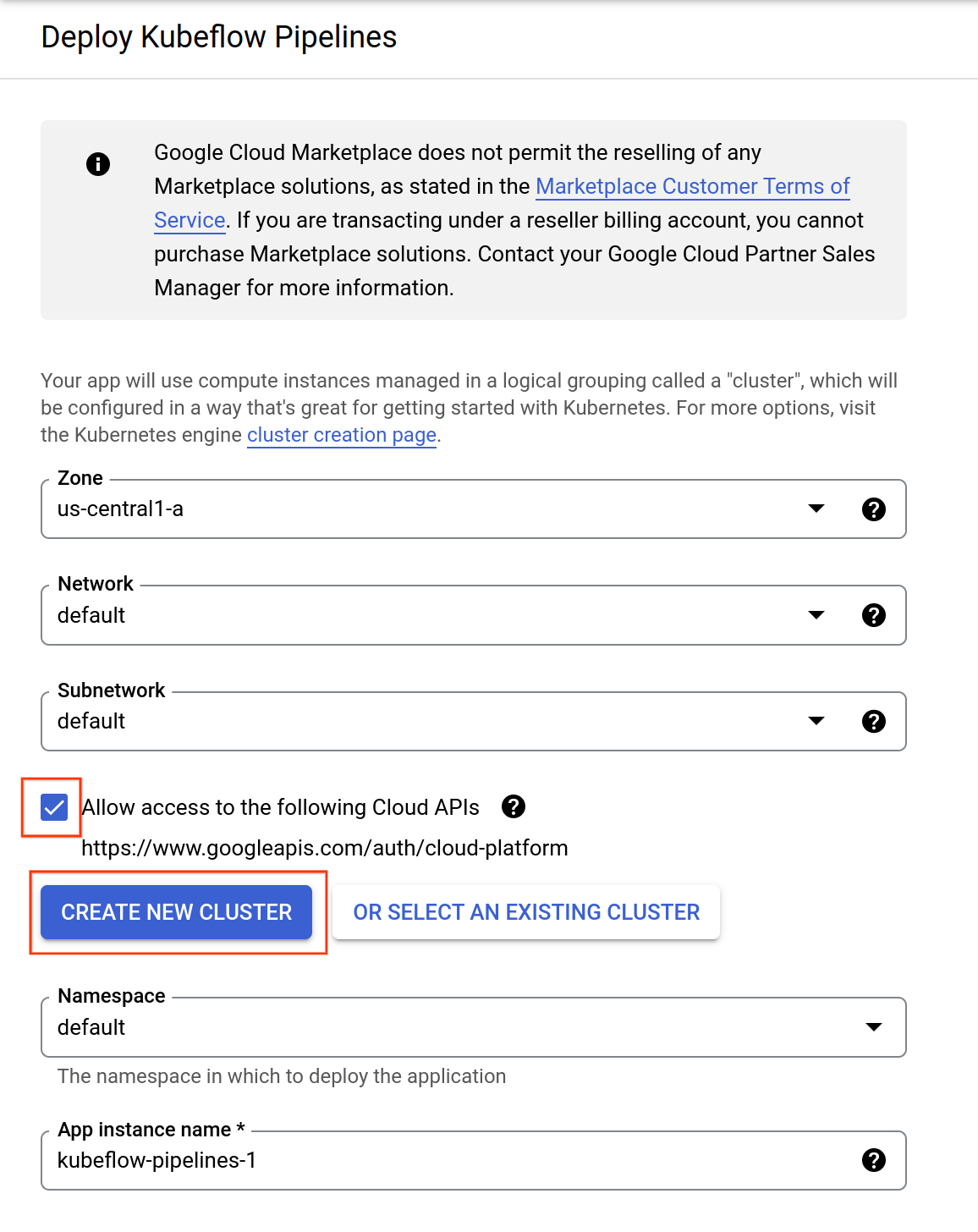
1. Then click **New Instance**.



1. Click **Configure**.

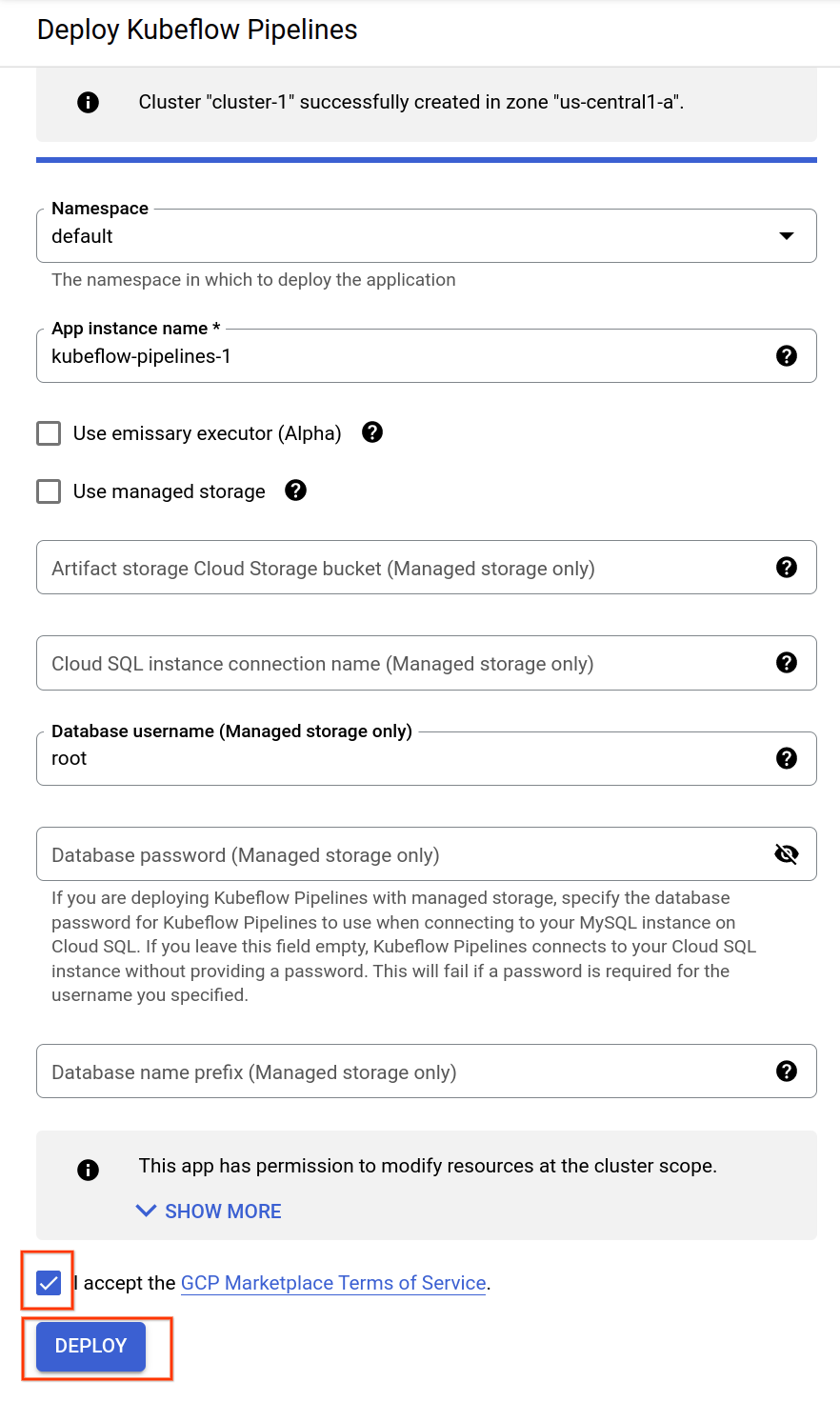


1. Check **Allow access to the following Cloud APIs**, leave the name as is, and then click **Create New Cluster**.



This should take 2-3 minutes to complete. Wait for the cluster to finish before proceeding to the next step. In the first tab opened, you can view the Cluster Creation taking place in the [GKE section of the Cloud Console](https://console.cloud.google.com/kubernetes/), or see the individual VMs spinning up in the [GCE section of the Cloud Console](https://console.cloud.google.com/compute/).

1. When the cluster creation is complete, check the **Terms of Service** box, leave other settings unchanged, and then click **Deploy**. You will see the individual services of KFP deployed to your GKE cluster. Proceed to the next step while installation occurs.



Click Check my progress to verify the objective.

Setup AI Platform Pipelines Instance

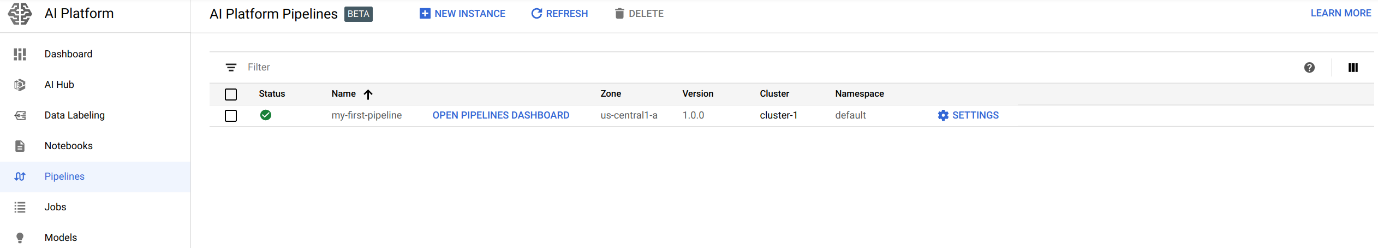
Check my progress

## Task 2. Run an example pipeline

1. In the Google Cloud Console, on the Navigation menu, click **AI Platform** > **Pipelines**.

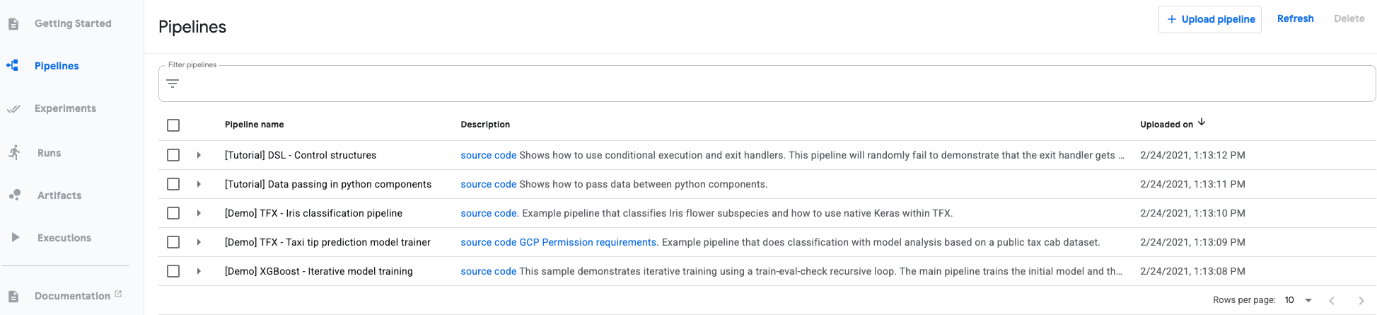
You will see the newly created Pipelines instance. If needed, click **Refresh** to update the page.

1. Click on the **OPEN PIPELINES DASHBOARD** link next to your instance name.



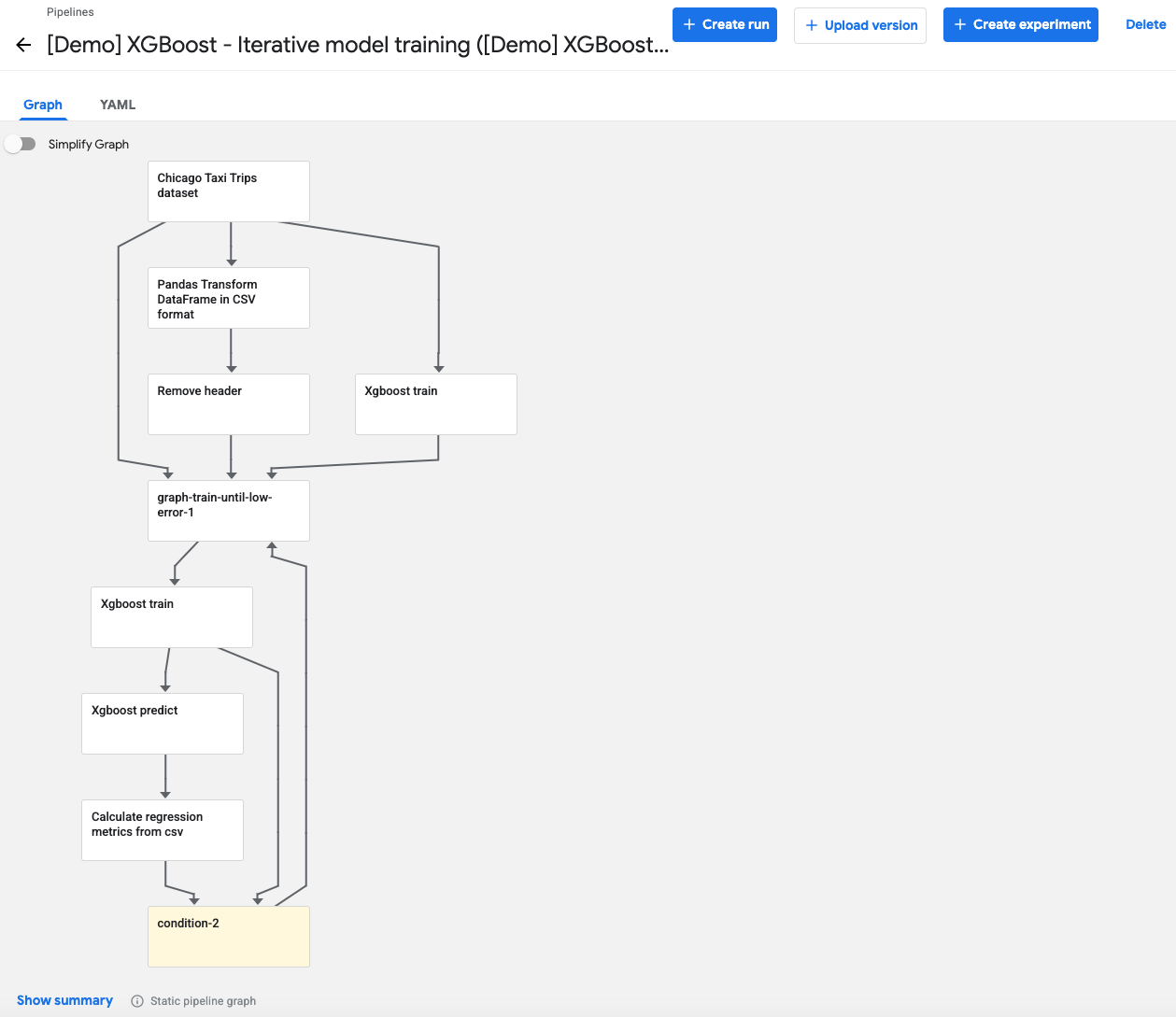
1. On the new page that loads, on the Navigation Menu on the left, click on **Pipelines**.

You will see a list of pipelines that have been provided for demo and tutorial purposes. For this lab, you will use the **[Demo] XGBoost - Iterative model training** sample pipeline. This sample demonstrates continuous training using a train-eval-check recursive loop, in which the model is trained iteratively until the model evaluation metrics are adequate.



1. Click on the **[Demo] XGBoost - Iterative model training** pipeline.

When it loads, you can see what the graph for this pipeline looks like. Next, you will create a run to test this pipeline.

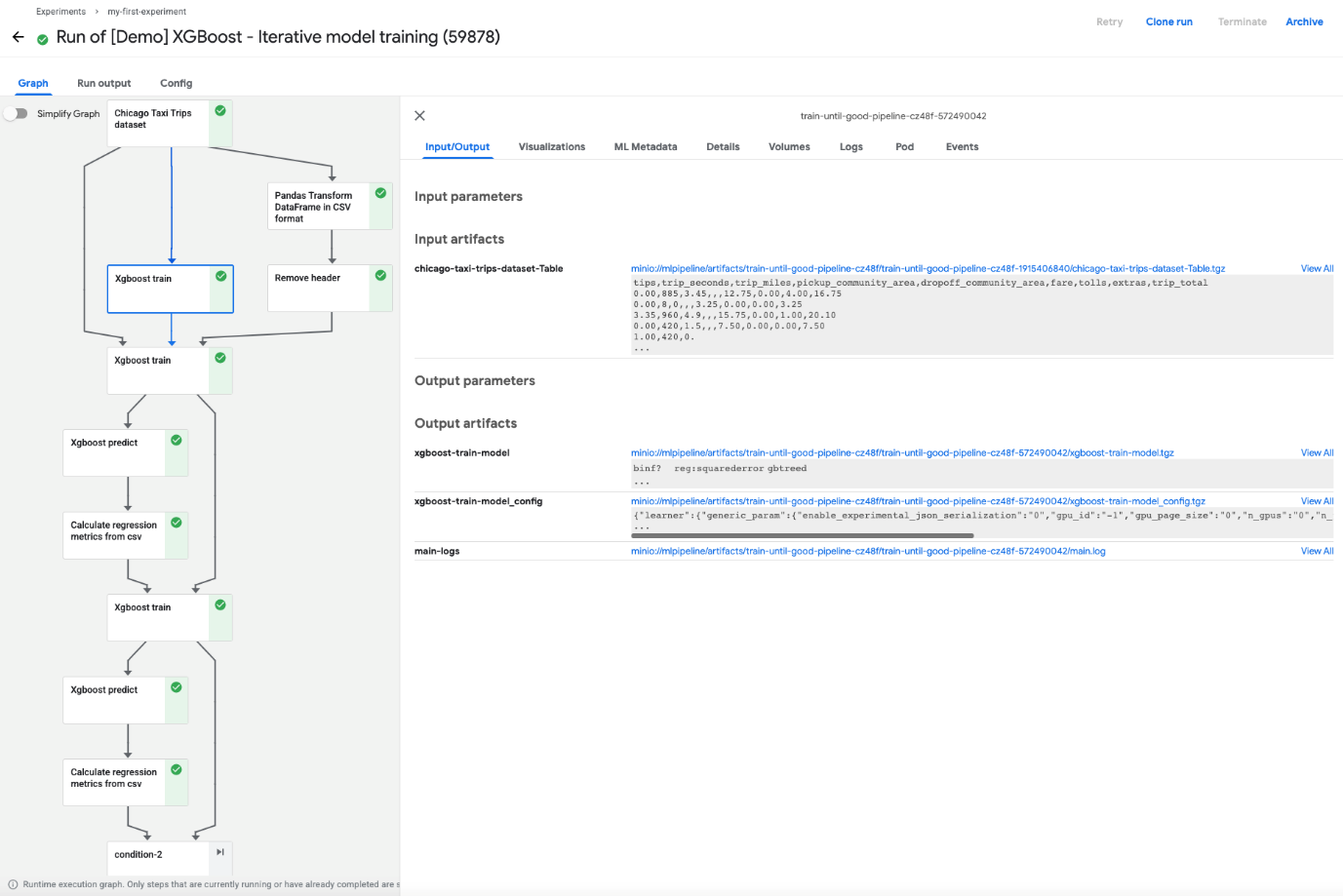


1. Click on **Create experiment** on the top right to associate a new experiment for the run.
2. Enter the name **my-first-experiment** in the form that loads, and then click **Next**.
3. Leave the default options, and click **Start** to run the pipeline. The pipeline run may take a few minutes to complete.

You can click **Refresh** to update the page and see the latest status.

1. Once the pipeline run has finished, you can click on the run name to see the fully generated graph as well as performance metrics and graphs.

The green check marks means every part of the pipeline ran successfully. You can click on any box and see the outputs for that part like input/output, visualizations, logs, events, etc.



Click Check my progress to verify the objective.

Run an Example Pipeline

Check my progress

### **Congratulations!**

In this lab, you used Kubeflow Pipelines to create and orchestrate an ML pipeline on Cloud AI Platform.