

# Running Pipelines on Vertex AI 2.5

1 hourFree

## Overview

In this lab, you learn how to utilize Vertex AI Pipelines to execute a simple Kubeflow Pipeline SDK derived ML Pipeline.

## Objectives

In this lab, you perform the following tasks:

- Set up the Project Environment
- Inspect and Configure Pipeline Code
- Execute the AI Pipeline

# Setup and requirements

Before you click the Start Lab button

**Note:** Read these instructions.

Labs are timed and you cannot pause them. The timer, which starts when you click Start Lab, shows how long Google Cloud resources will be made available to you.

This Qwiklabs hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

What you need

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.


**Note:** If you already have your own personal Google Cloud account or project, do not use it for this lab. **Note:** If you are using a Pixelbook, open an Incognito window to run this lab.


How to start your lab and sign in to the Console


1. Click the Start Lab button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.

[Open Google Console](#)

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. [Learn more.](#)

Username  
google2727032\_student@qwiklabs.n 

Password  
k68CZXsxMZ 

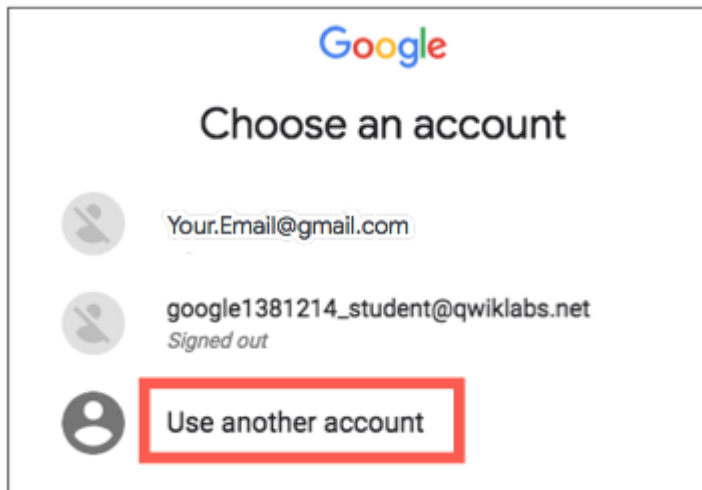
GCP Project ID  
qwiklabs-gcp-4fbfecac8667e457 

[New to labs? View our introductory video!](#)

2. Copy the username, and then click Open Google Console. The lab spins up resources, and then opens another tab that shows the Choose an account page.

**Note:** Open the tabs in separate windows, side-by-side.

3. On the Choose an account page, click Use Another Account. The Sign in page opens.



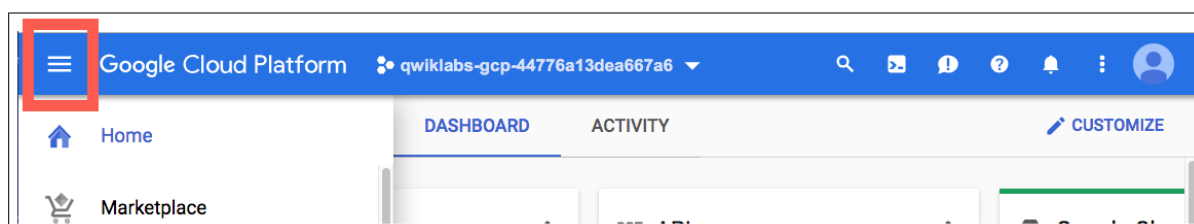
4. Paste the username that you copied from the Connection Details panel. Then copy and paste the password.

**Note:** You must use the credentials from the Connection Details panel. Do not use your Google Cloud Skills Boost credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

5. Click through the subsequent pages:
  - Accept the terms and conditions.
  - Do not add recovery options or two-factor authentication (because this is a temporary account).
  - Do not sign up for free trials.

After a few moments, the Cloud console opens in this tab.

**Note:** You can view the menu with a list of Google Cloud Products and Services by clicking the Navigation menu at the top-left.



# 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 (☰), select 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.

The screenshot shows the Google Cloud Platform interface with the IAM & Admin section selected. The left sidebar lists various IAM-related tools. The main content area displays the 'Permissions for project "qwiklabs-gcp-03-e30ac90a32e4"'. A table lists the principals and their roles. The first entry, '407543585891-compute@developer.gserviceaccount.com', is highlighted with a red box and shows the 'Editor' role.

Type	Principal	Name	Role
<input type="checkbox"/>	407543585891-compute@developer.gserviceaccount.com	Compute Engine default service account	Editor
<input type="checkbox"/>	407543585891@cloudbuild.gserviceaccount.com		Cloud Build Service Account
<input type="checkbox"/>	407543585891@cloudservices.gserviceaccount.com	Google APIs Service Agent	Editor
<input type="checkbox"/>	admiral@qwiklabs-services-prod.iam.gserviceaccount.com		Owner
<input type="checkbox"/>	qwiklabs-gcp-03-e30ac90a32e4@qwiklabs-gcp-03-e30ac90a32e4.iam.gserviceaccount.com	Qwiklabs User Service Account	App Engine Admin BigQuery Admin

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

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

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

**Copied!**


**content\_copy**

6. **Replace** `{project-number}` **with your project number.**
7. **For Role, select Project (or Basic) > Editor.**
8. **Click Save.**

## Task 1. Set up the project environment

Vertex AI Pipelines run in a serverless framework whereby pre-compiled pipelines are deployed on-demand or on a schedule. In order to facilitate smooth execution some environment configuration is required.

For the seamless execution of Pipeline code in a Qwiklabs environment the Compute Service Account needs elevated privileges on Cloud Storage.

1. In the Google Cloud console, on the Navigation menu () , click IAM & Admin > IAM.
2. Click the pencil icon for default compute Service Account `{project-number}-compute@developer.gserviceaccount.com` to assign the Storage Admin role.
3. On the slide-out window, click Add Another Role. Type Storage Admin in the search box. Select Storage Admin with Full control of GCS resources from the results list.
4. Click Save to assign the role to the Compute Service Account.

## Edit permissions

### Principal

449683761124-compute@developer.gserviceaccount.com

### Project

Role

Editor

Condition

[Add condition](#)



View, create, update, and delete most Google Cloud resources. See the list of included permissions.

Role

Storage Admin

Condition

[Add condition](#)



Full control of GCS resources.

[+ ADD ANOTHER ROLE](#)

SAVE

TEST CHANGES



CANCEL

Artifacts will be accessed on ingest and export as the Pipeline executes.

5. Run this code block in the Cloud Shell to create a bucket in your project and two folders each with an empty file:

```
gsutil mb gs://Project ID
touch emptyfile1
touch emptyfile2
gsutil cp emptyfile1 gs://Project ID/pipeline-output/emptyfile1
gsutil cp emptyfile2 gs://Project ID/pipeline-input/emptyfile2
```

**Copied!**

**content\_copy**

The Pipeline has already been created for you and simply requires a few minor adjustments to allow it to run in your Qwiklabs project.

6. Download the AI Pipeline from the lab assets folder:

```
wget https://storage.googleapis.com/cloud-training/dataengineering/lab_assets/ai_pipelines/basic_pipeline.json
```

**Copied!**

**content\_copy**

Click Check my progress to verify the objective.

Configure the environment

Check my progress

## Task 2. Configure and inspect the Pipeline code

The Pipeline code is a compilation of two AI operations written in Python. The example is very simple but demonstrates how easy it is to orchestrate ML procedures written in a variety of languages (TensorFlow, Python, Java, etc.) into an easy-to-deploy AI Pipeline. The lab example performs two operations, concatenation and reverse, on two string values.

1. First you must make an adjustment to update the output folder for the AI Pipeline execution. In the Cloud Shell use the Linux Stream Editor (sed) command to adjust this setting:

```
sed -i 's/PROJECT_ID/Project ID/g' basic_pipeline.json
```

Copied!

content\_copy

2. Inspect basic\_pipeline.json to confirm that the output folder is set to your project:

```
tail -20 basic_pipeline.json
```

Copied!

content\_copy

The key sections of code in basic\_pipeline.json are the deploymentSpec and command blocks. Below is the first command block, the job that concatenates the input strings. This is Kubeflow Pipeline SDK (kfp) code that is designated to be executed by the Python 3.7 engine. You will not change any code, the section is shown here for your reference:

```
"program_path=$(mktemp -d)\nprintf \"%s\\\" \"$0\\\" >\n\\\"$program_path/ephemeral_component.py\\\"\\npython3 -m\nkfp.v2.components.executor_main\ncomponent_module_path\n\\\"$program_path/ephemeral_component.py\\\"\n\\\"$@\\\"\\n",\n\n    "\nimport kfp\nfrom kfp.v2 import dsl\nfrom kfp.v2.dsl\nimport *\nfrom typing import *\n\ndef concat(a: str, b: str) -> str:\n    return a + b\n\n",\n\n    "image": "python:3.7"
```

3. You can explore the entire file by issuing the command below:

```
more basic_pipeline.json
```

**Copied!**

**content\_copy**

**Note:** Press the spacebar to advance through the file until its end. If you wish to close the file early, type `q` to close the `more` command.

4. Next, move the updated `basic_pipeline.json` file to the Cloud Storage bucket created earlier so that it can be accessed to run an AI Pipeline job:

```
gsutil cp basic_pipeline.json gs://Project ID/pipeline-  
input/basic_pipeline.json
```

**Copied!**


**content\_copy**

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

**Deploy Pipeline**

**Check my progress**

## Task 3. Execute the AI Pipeline

1. From the Console, open the navigation menu () , under Artificial Intelligence click Vertex AI.
2. Activate the Vertex AI API. Click the blue **Enable Vertex AI API** button.
3. Once the API is enabled, click **Pipelines** in the left menu.
4. Click **Create Run** on the top menu.
5. Under **Run detail**, for **Pipeline** select **Import from Cloud Storage** from the dropdown and for **Cloud Storage URL** browse to the `pipeline-input` folder you created inside your project's cloud storage bucket. Select the `basic_pipeline.json` file.
6. Leave the remaining default values, click **Continue**.

You may leave the default values for Runtime configuration. Notice that the GCS Output Directory is set to the bucket folder created in an earlier step. The Pipeline



Parameters are pre-filled from the values in the `basic_pipeline.json` file but you have the option of changing those at runtime via this wizard.

7. Click Submit to start the Pipeline execution.
8. You will be returned to the Pipeline dashboard and your run will progress from Pending to Running to Succeeded. The entire run will take between 3 and 6 minutes.
9. Once the status reaches Succeeded, click on the run name to see the execution graph and details.

The screenshot displays the 'Pipeline run analysis' interface. On the left, the 'Runtime Graph' shows two steps: 'concat python:3.7' and 'reverse python:3.7', both marked as completed with green checkmarks. A blue box highlights the 'concat' node. On the right, the 'NODE INFO' tab is active, showing execution details for the 'concat' node.

**Execution Info** Completed

[VIEW JOB](#) [VIEW LOGS](#)

Display name	concat
Name	concat
Type	system.ContainerExecution
Duration	1 min 52 sec
Started	Jul 27, 2022, 4:33:11 PM
Completed	Jul 27, 2022, 4:35:03 PM

**Input Parameters**

Parameter	Type	Value
a	string	stres
b	string	sed

**Output Parameters**

Parameter	Type	Value
Output	string	stressed

10. A graph element exists for each step. Click on the concat object to see the details for the job.
11. Click on the View Job button. A new tab will open with the Vertex AI Custom Job that was submitted to the backend to satisfy the pipeline request.

← caip\_pipelines\_4991375970800762880\_-306294802440257536

**i** Custom job was completed on Jul 27, 2022, 4:34:53 PM.

Status	Finished
Custom job ID	1349617812624113664
Created	Jul 27, 2022, 4:33:11 PM
Start time	Jul 27, 2022, 4:34:22 PM
Elapsed time	31 sec
Region	us-central1
Encryption type	Google-managed key
Machine type (Worker pool 0)	e2-standard-4
Machine count (Worker pool 0)	1
Container Location (Worker pool 0)	python:3.7
Arguments (Worker pool 0)	--executor_input; {"inputs":{"parameterValues":{"a":"stres","b":"sed"},"parameters":{"a":{"stringValue":"stres"},"b":{"stringValue":"sed"}}},"outputs":{"outputFile":"/gcs/qwiklabs-gcp-03-e66bb00a2df2/pipeline-output/938406161599/basic-pipeline-20220727043032/concat_-306294802440257536/executor_output.json","parameters":{"Output":{"outputFile":"/gcs/qwiklabs-gcp-03-e66bb00a2df2/pipeline-output/938406161599/basic-pipeline-20220727043032/concat_-306294802440257536/Output"}}}}; --function_to_execute; concat
Dataset	No managed dataset
Algorithm	Custom training
Objective	Custom
Container (Training)	Custom
Logs	<a href="#">View logs</a>

[VIEW CUSTOM JOB INPUTS IN JSON](#)

Feel free explore more details on the Pipeline execution.

## Congratulations!

You have successfully used Vertex AI Pipelines to execute a simple Kubeflow Pipeline SDK derived ML Pipeline.

Manual Last Updated November 18, 2022

Lab Last Tested August 02, 2022

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.

