

Assignment 7: optional catchup assignment 2 - VERTEX AI - for midterm and quiz - this will catch up midterm.

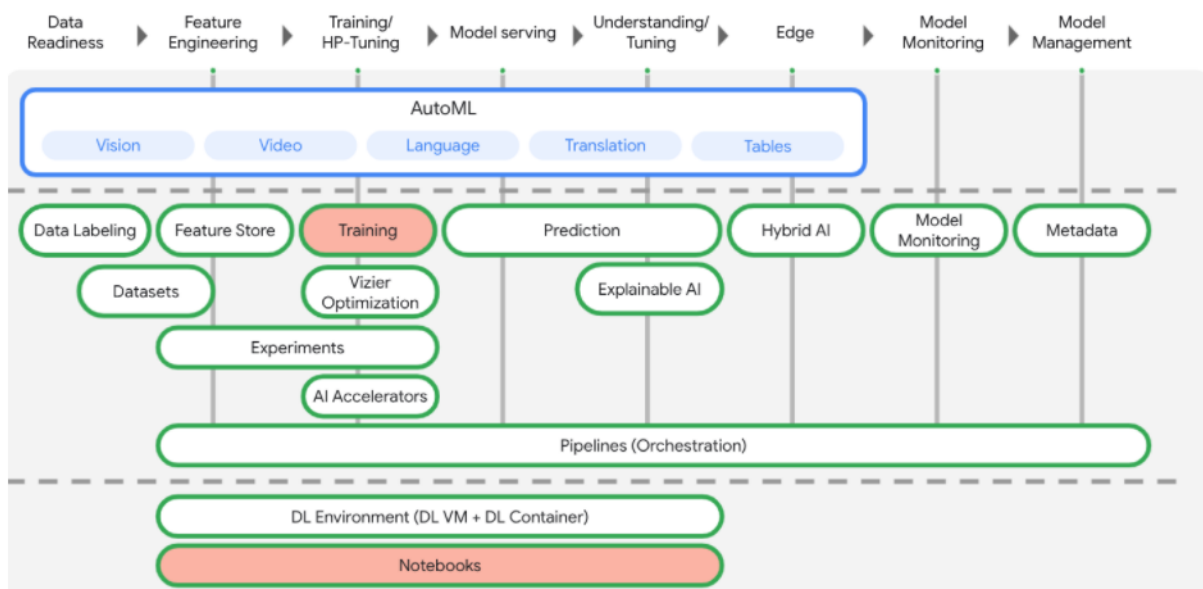
a) Vertex AI to run a hyperparameter tuning job for a TensorFlow model

Reference: https://codelabs.developers.google.com/vertex_hyperparameter_tuning#0

Objectives:

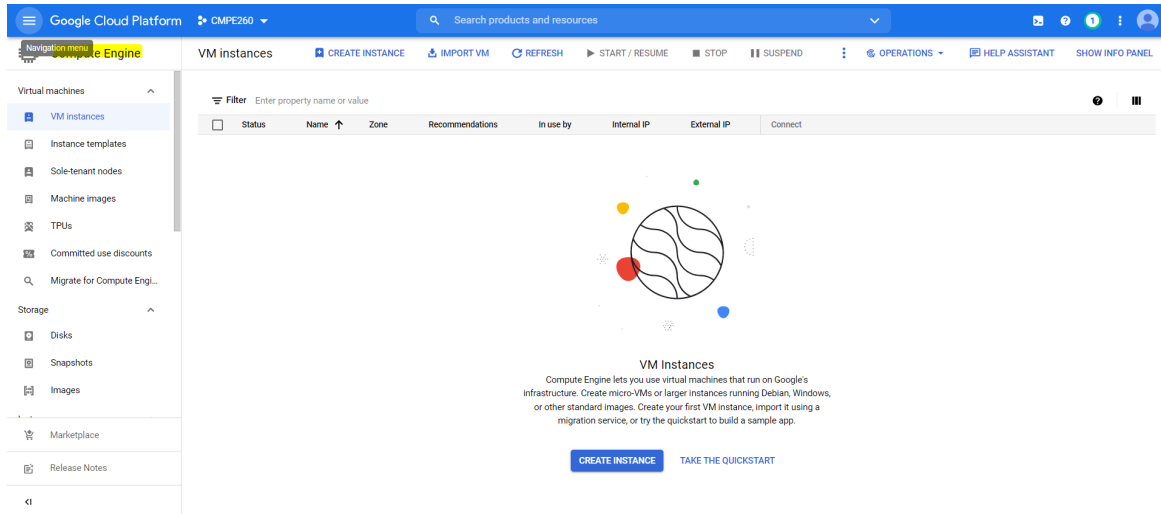
- Modify training application code for hyperparameter tuning
- Configure and launch a hyperparameter tuning job from the Vertex AI UI
- Configure and launch a hyperparameter tuning job with the Vertex SDK

Vertex AI includes many different products to support end-to-end ML workflows. This document will focus on the products highlighted below: Training/HP-Tuning and Notebooks

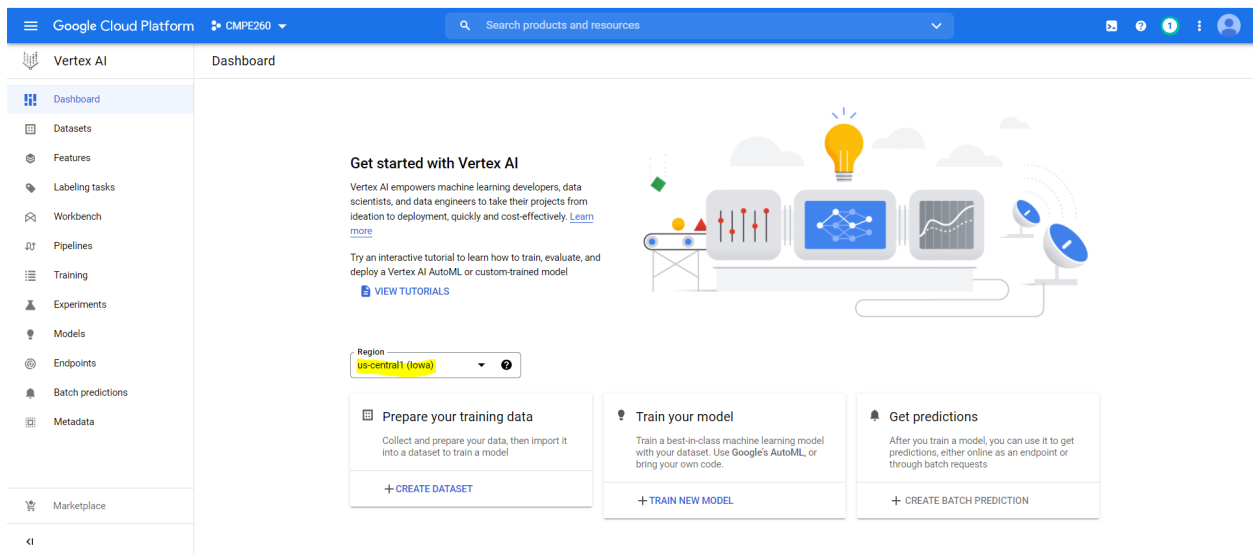


Setup your environment

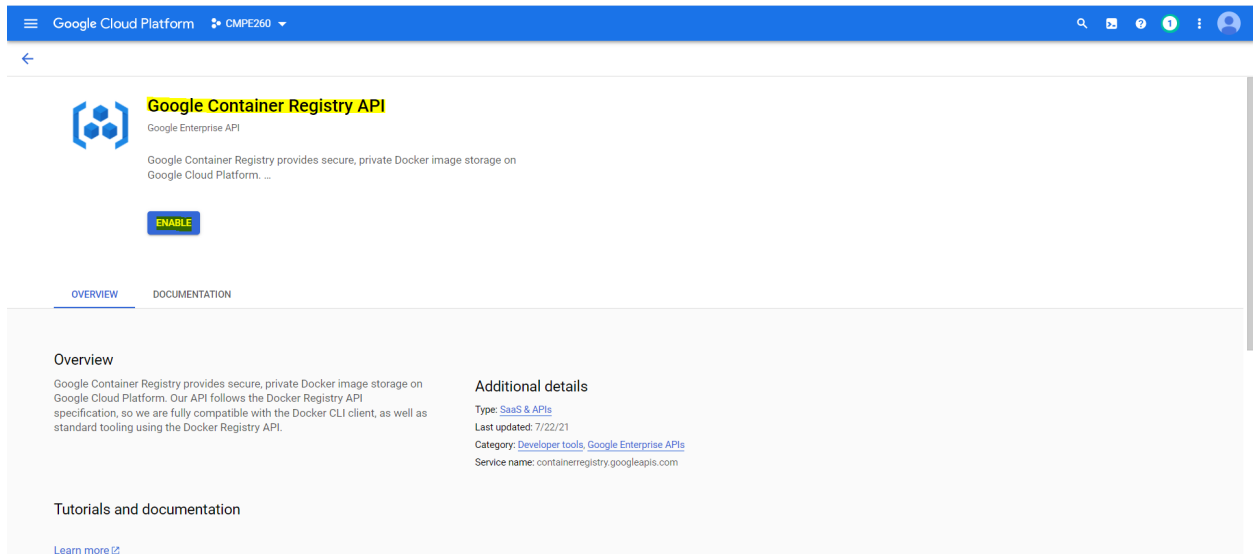
Step 1: Enable the Compute Engine API



Step 2: Enable the Vertex AI API

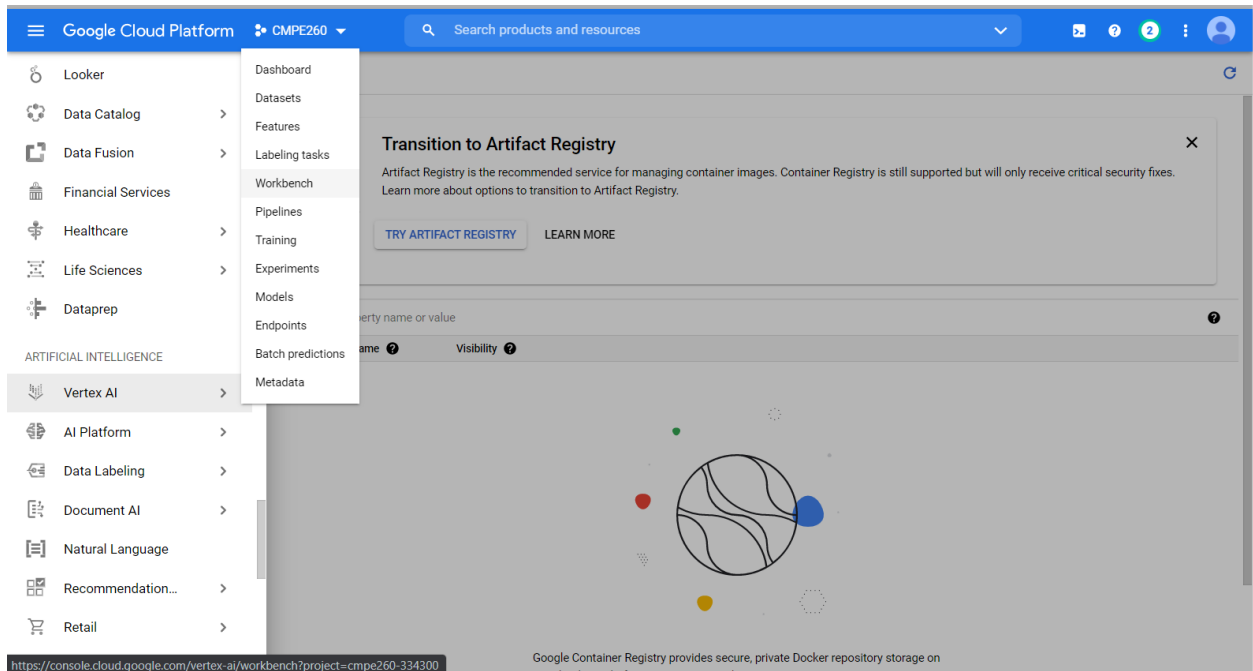


Step 3: Enable the Container Registry API

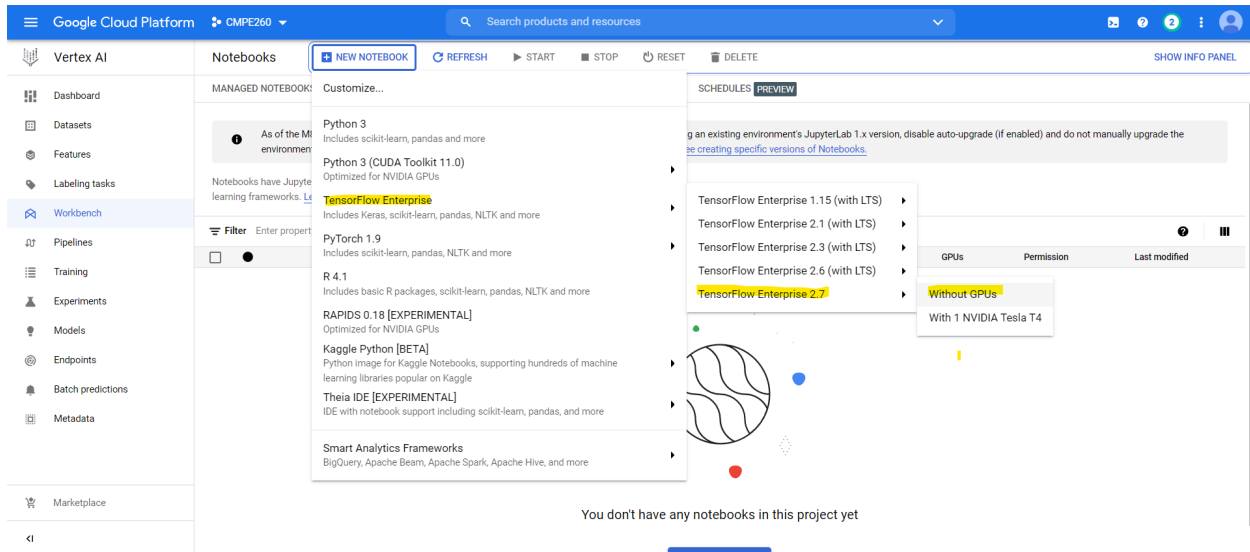


Step 4: Create a Vertex AI Workbench instance

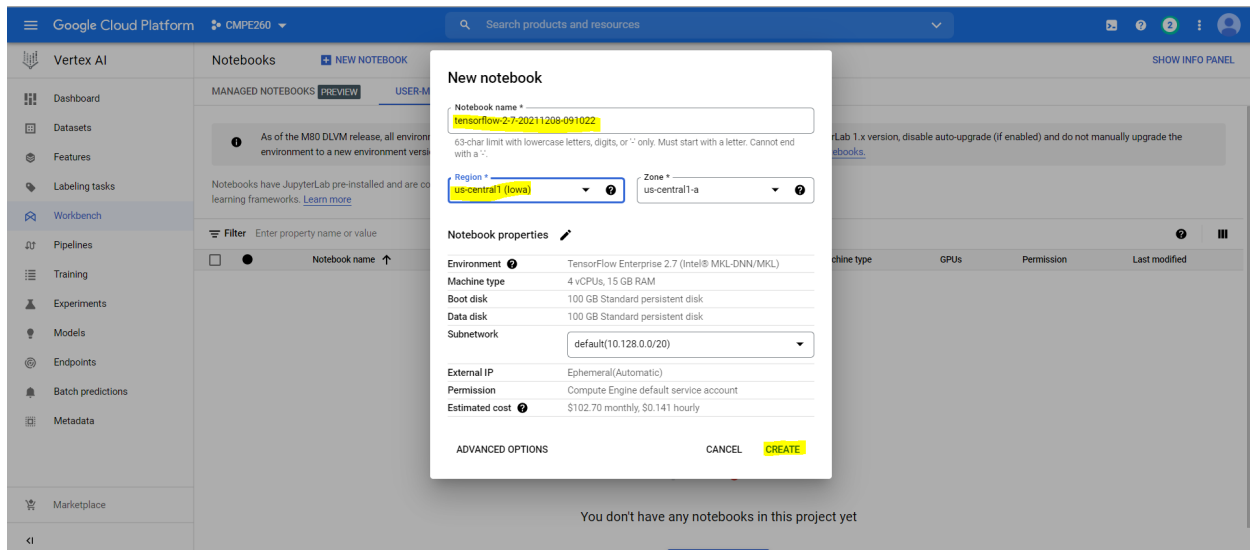
In the Left Nav, select Vertex AI → Workbench



Select **New Notebook**. Then select the **TensorFlow Enterprise 2.7** instance type **without GPUs**:



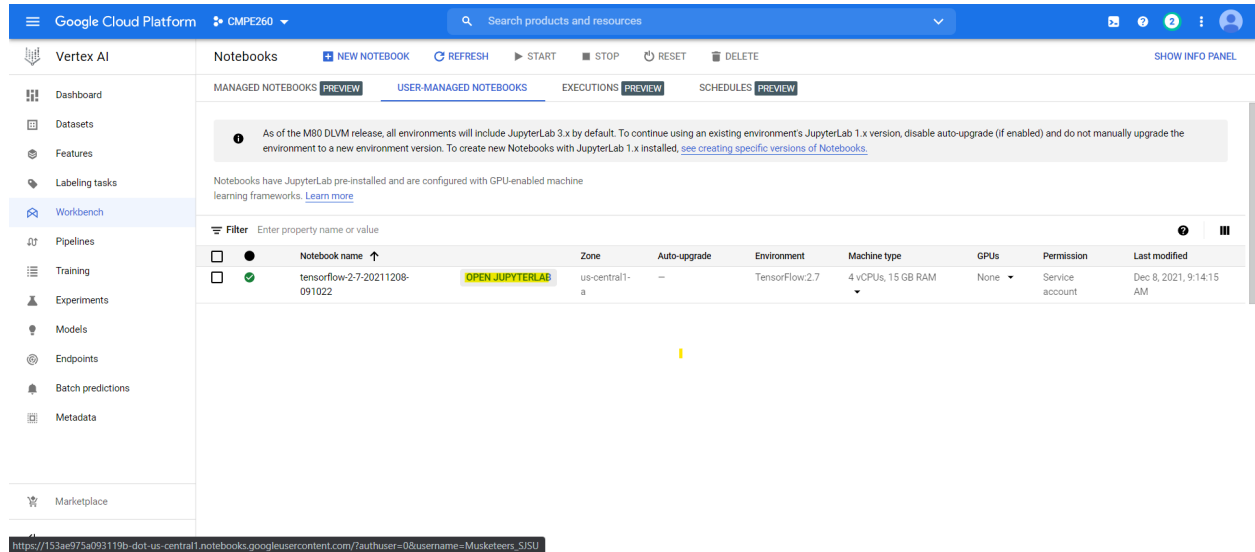
Enter the name, region and click 'Create'



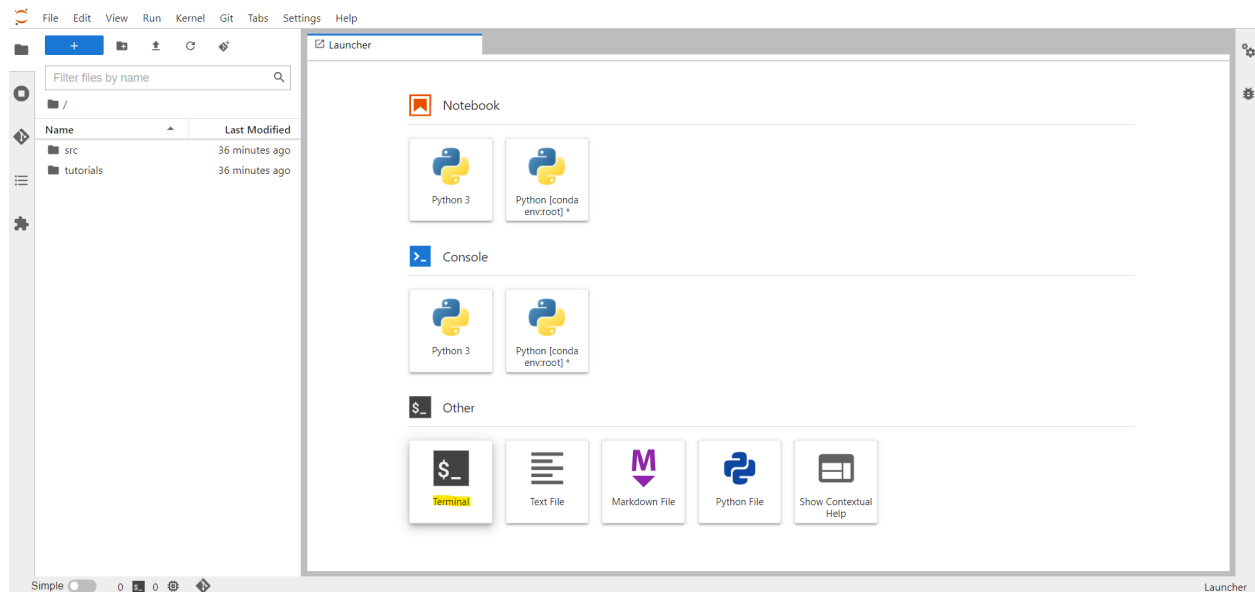
Containerize training application code

In this approach, we will submit this hyperparameter tuning job to Vertex by putting the training application code in a Docker container and pushing this container to Google Container Registry. Using this approach, you can tune hyperparameters for a model built with any framework.

1. Click 'Open JupyterLab'



2. Click 'Terminal'



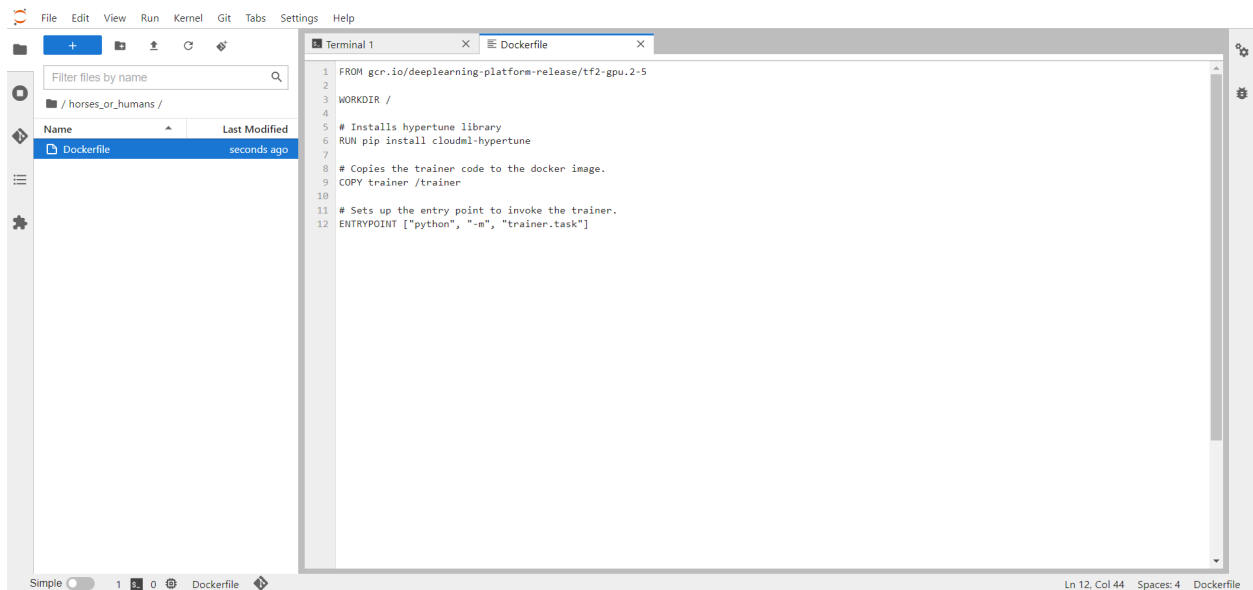
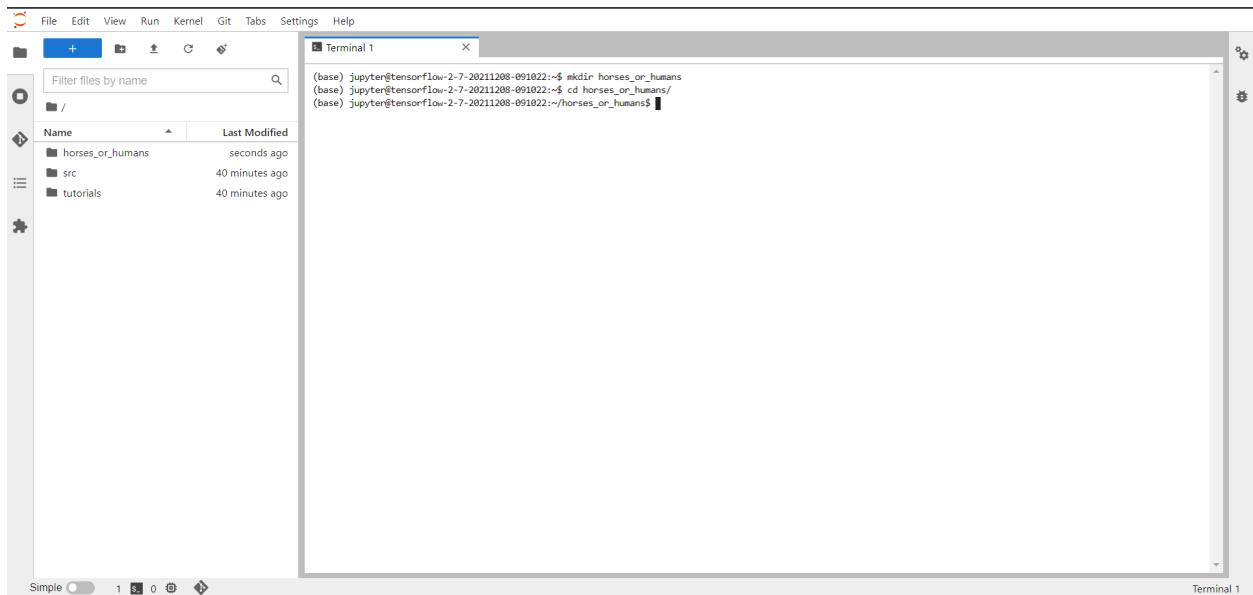
Create Docker file:

Run the following commands to create a folder

```
mkdir horses_or_humans
```

```
cd horses_or_humans
```

```
touch Dockerfile
```

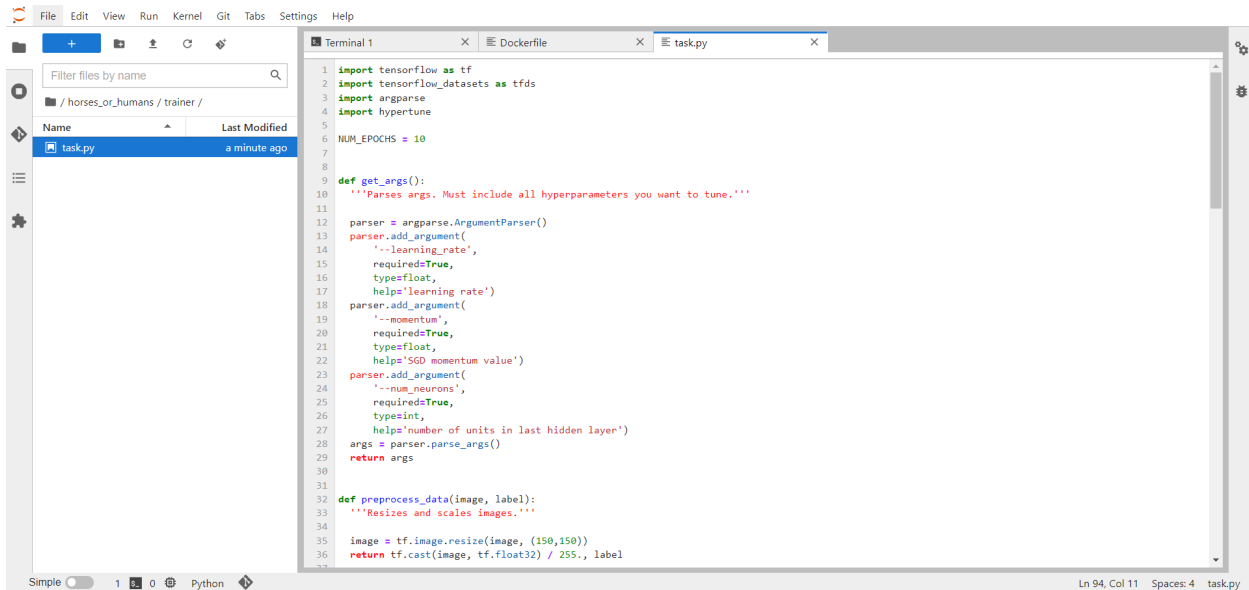


Add model training code

Run the following commands:

```
mkdir trainer
```

```
touch trainer/task.py
```



```
1 import tensorflow as tf
2 import tensorflow_datasets as tfds
3 import argparse
4 import hypertune
5
6 NUM_EPOCHS = 10
7
8
9 def get_args():
10     """Parses args. Must include all hyperparameters you want to tune."""
11
12     parser = argparse.ArgumentParser()
13     parser.add_argument(
14         '--learning_rate',
15         required=True,
16         type=float,
17         help='learning rate')
18     parser.add_argument(
19         '--momentum',
20         required=True,
21         type=float,
22         help='SGD momentum value')
23     parser.add_argument(
24         '--num_neurons',
25         required=True,
26         type=int,
27         help='number of units in last hidden layer')
28     args = parser.parse_args()
29     return args
30
31
32 def preprocess_data(image, label):
33     """Resizes and scales images."""
34
35     image = tf.image.resize(image, (150,150))
36     return tf.cast(image, tf.float32) / 255., label
```

There are a few components that are specific to using the hyperparameter tuning service.

1. The script imports the hypertune library. Note that the Dockerfile from Step 1 included instructions to pip install this library.
2. The function `get_args()` defines a command-line argument for each hyperparameter you want to tune. In this example, the hyperparameters that will be tuned are the learning rate, the momentum value in the optimizer, and the number of neurons in the last hidden layer of the model, but feel free to experiment with others. The value passed in those arguments is then used to set the corresponding hyperparameter in the code.
3. At the end of the `main()` function, the hypertune library is used to define the metric you want to optimize. In TensorFlow, the `keras model.fit` method returns a History object. The `History.history` attribute is a record of training loss values and metrics values at successive epochs. If you pass validation data to `model.fit` the `History.history` attribute will include validation loss and metrics values as well. For example, if you trained a model for three epochs with validation data and provided accuracy as a metric, the `History.history` attribute would look similar to the following dictionary.

Build the container

Run the following commands:

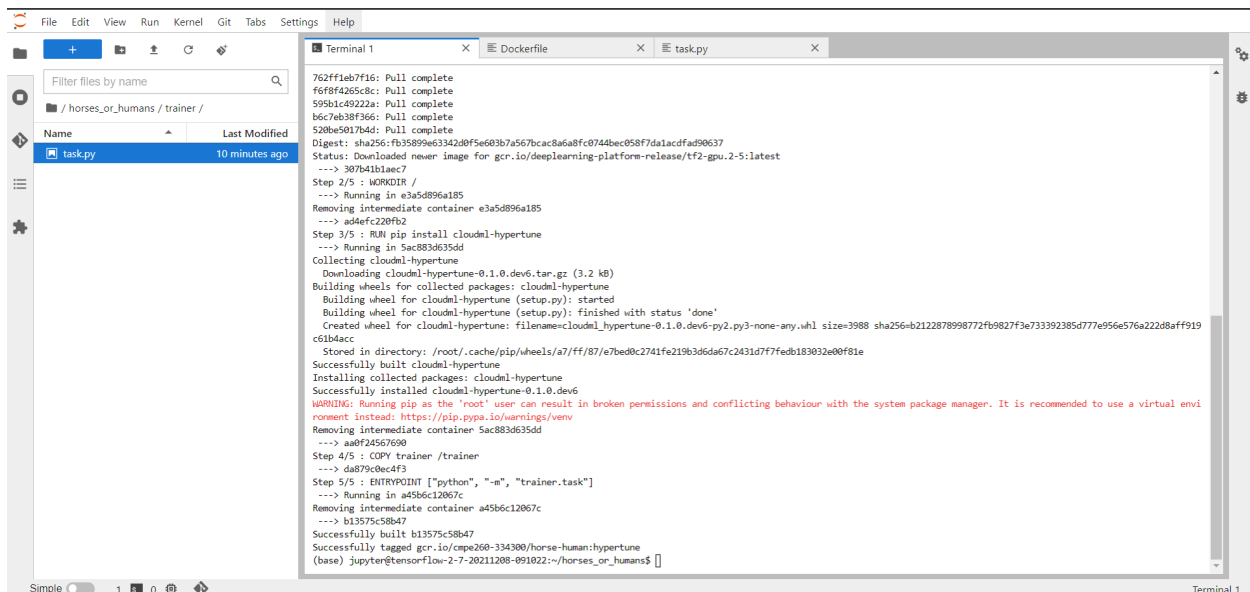
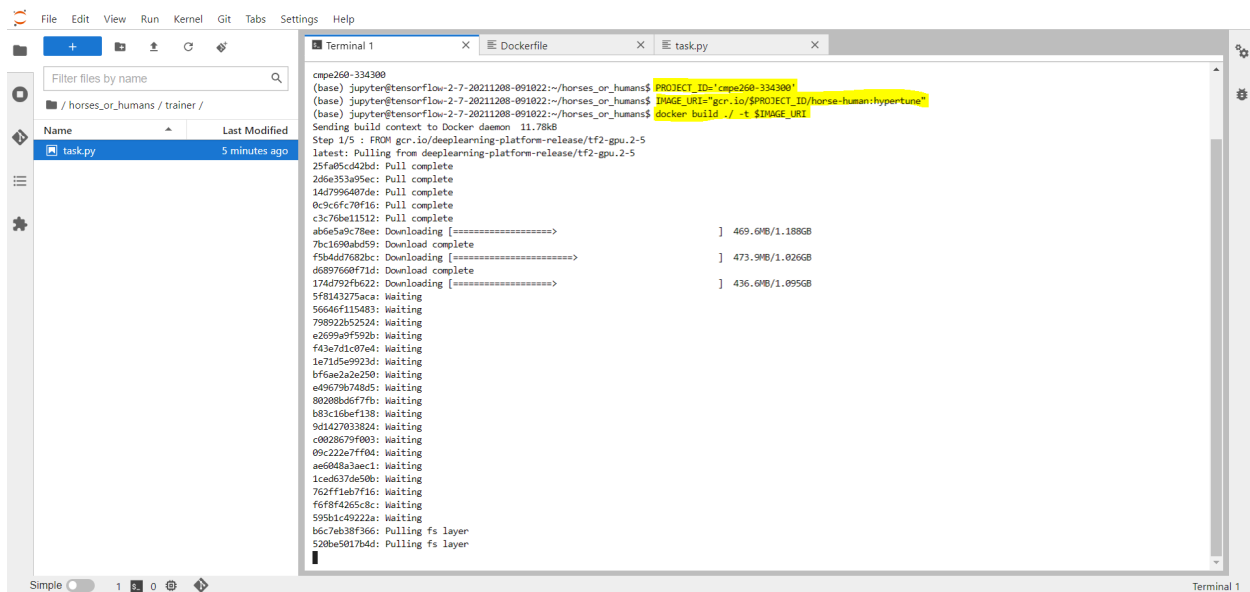
```
gcloud config list --format 'value(core.project)'
```

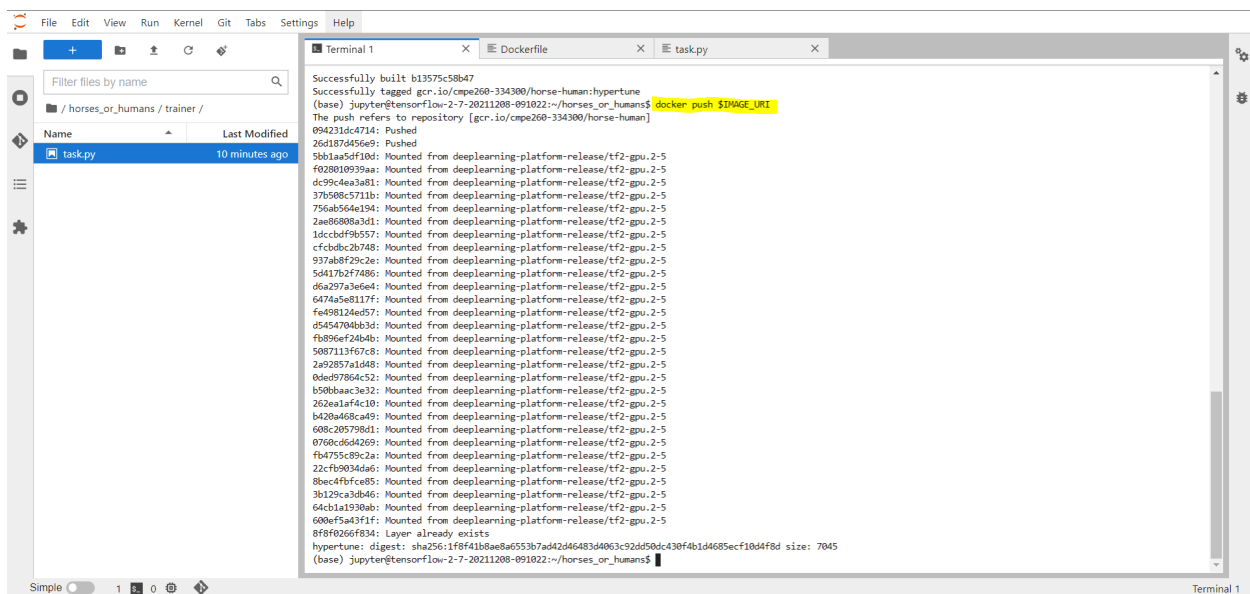
```
PROJECT_ID='cmpe260-334300'
```

```
IMAGE_URI="gcr.io/$PROJECT_ID/horse-human:hypertune"
```

```
docker build ./ -t $IMAGE_URI
```

```
docker push $IMAGE_URI
```





Run a hyperparameter tuning job on Vertex AI

1. Navigate to Vertex AI → Training

Google Cloud Platform CMP260 Search products and resources

Vertex AI Training CREATE REFRESH

TRAINING PIPELINES CUSTOM JOBS HYPERPARAMETER TUNING JOBS

Training pipelines are the primary model training workflow in Vertex AI. You can use training pipelines to create an AutoML-trained model or a custom-trained model. For custom-trained models, training pipelines orchestrate custom training jobs and hyperparameter tuning with additional steps like adding a dataset or uploading the model to Vertex AI for prediction serving. [Learn More](#)

Region us-central1 (Iowa)

Filter Enter a property name

Name	ID	Status	Job type	Model type	Created	Elapsed time	Labels
titanic_202112744834	1504731140634705920	Finished	Training pipeline	Custom	Dec 6, 2021, 9:17:57 PM	12 min 39 sec	—

<https://console.cloud.google.com/vertex-ai/training?project=cmp260-334300>

Step 1: Configure training job

Click **Create** to enter the parameters for your hyperparameter tuning job.

- Under **Dataset**, select **No managed dataset**
- Then select **Custom training (advanced)** as your training method and click **Continue**.
- Click **Continue**

The screenshot shows the 'Train new model' dialog in the Google Cloud Platform console. The left sidebar lists various services, with 'Vertex AI' and 'Training' highlighted. The main panel is titled 'Train new model' and has a progress bar with six steps: 1. Training method (selected), 2. Model details, 3. Training container, 4. Hyperparameters (optional), 5. Compute and pricing, and 6. Prediction container (optional). Below the progress bar are 'START TRAINING' and 'CANCEL' buttons. The right panel shows the configuration for the first step. The 'Dataset' dropdown is set to 'No managed dataset'. The 'Annotation set' dropdown is set to '-'. The 'Objective' dropdown is set to 'Custom'. A note states: 'Please refer to the pricing guide for more details (and available deployment options) for each method.' Below this, a message says: 'AutoML options are only available when you train with a managed dataset.' There are three radio button options: 'AutoML' (disabled), 'AutoML Edge' (disabled), and 'Custom training (advanced)' (selected). The 'Custom training (advanced)' option has a description: 'Run your TensorFlow, scikit-learn, and XGBoost training applications in the cloud. Train with one of Google Cloud's pre-built containers or use your own.' A 'CONTINUE' button is at the bottom right.

- Enter `horses-humans-hyptertune` (or whatever you'd like to call your model) for **Model name**
- Click **Continue**

The screenshot shows the 'Train new model' dialog in the Google Cloud Platform console, now at Step 2: Model details. The progress bar shows 'Model details' as the selected step. The right panel shows the configuration for this step. The 'Model name' dropdown is set to 'horses-humans-hyptertune'. Below this, there are three sections: 'Encryption' with a checkbox 'Use a customer-managed encryption key (CMEK)' which is unchecked; 'Service account' with a text field 'Service account' and a 'BROWSE' button; and 'Network' with a text field 'Peered VPC network'. At the bottom, there is a 'Training Debugging' section with a checkbox 'Enable training debugging' which is unchecked, and a 'SHOW LESS' link. A 'CONTINUE' button is at the bottom right.

- Select 'Custom Container'
- Enter **Container Image: URI to the Image**

gcr.io/cmpe260-334300/horse-human@sha256:1f8f41b8ae8a6553b7ad42d46483d4063c92dd50dc430f4b1d4685ecf10d4f8d

The screenshot shows the 'Train new model' wizard in the Google Cloud Platform console. The 'Training container' step is selected. The 'Custom container' option is chosen under 'Pre-built container'. The 'Container image' field contains the URI: `gcr.io/cmpe260-334300/horse-human@sha256:1f8f41b8ae8a6553b7ad42d46483d4063c92dd50dc430f4b1d4685ecf10d4f8d`. The 'Model output directory' is set to `gs://`. The 'Arguments' field contains: `-flag_1=xxxxx
-flag2
flag3`. The 'CONTINUE' button is visible at the bottom.

Step 2: Configure hyperparameter tuning job

- Select 'Enable Hyperparameter tuning'
- Enter the Parameter details for `learning_rate`

The screenshot shows the 'Hyperparameters (optional)' step in the 'Train new model' wizard. The 'Enable hyperparameter tuning' checkbox is checked. The 'Hyperparameter tuning variables' section shows a 'New Hyperparameter' form with the following details:

- Parameter name: `learning_rate`
- Type: Double
- Min: 0.01
- Max: 1
- Scaling: Log

 The 'ADD NEW PARAMETER' button is visible at the bottom.

Upgrade your account to avoid a break in service

Google Cloud Platform

Vertex AI

Dashboard

Datasets

Features

Labeling tasks

Workbench

Pipelines

Training

Experiments

Models

Endpoints

Batch predictions

Metadata

Marketplace

Train new model

Training method

Model details

Training container

Hyperparameters (optional)

Compute and pricing

START TRAINING CANCEL

Hyperparameter tuning optimizes your model through multiple trials in one training job, but will increase the cost of this job. After training finishes, the best-performing model will be saved to your Model List. [Learn more](#)

☒ Enable hyperparameter tuning

Hyperparameter tuning variables

1 Ensure that your hyperparameter variables are named and typed correctly. [VIEW DOCS](#)

learning_rate (Double), 0.01 - 1

New Hyperparameter

Parameter name *
momentum

Type *
Double

Min *
0

Max *
1

Scaling *
Linear

CANCEL DONE

ADD NEW PARAMETER

Upgrade your account to avoid a break in service

Google Cloud Platform

Vertex AI

Dashboard

Datasets

Features

Labeling tasks

Workbench

Pipelines

Training

Experiments

Models

Endpoints

Batch predictions

Metadata

Marketplace

Train new model

Training method

Model details

Training container

Hyperparameters (optional)

Compute and pricing

START TRAINING CANCEL

☒ Enable hyperparameter tuning

Hyperparameter tuning variables

1 Ensure that your hyperparameter variables are named and typed correctly. [VIEW DOCS](#)

learning_rate (Double), 0.01 - 1

momentum (Double), 0 - 1

New Hyperparameter

Parameter name *
num_neurons

Type *
Discrete

Values *
64,128,512

Enter all the values you want to tune, separated with comma.

Scaling *
No scaling

CANCEL DONE

ADD NEW PARAMETER

Metric to optimize *

Upgrade your account to avoid a break in service

Google Cloud Platform

Vertex AI

Dashboard

Datasets

Features

Labeling tasks

Workbench

Pipelines

Training

Experiments

Models

Endpoints

Batch predictions

Metadata

Marketplace

Train new model

Training method

Model details

Training container

Hyperparameters (optional)

Compute and pricing

START TRAINING CANCEL

Hyperparameter tuning variables

Ensure that your hyperparameter variables are named and typed correctly.

[VIEW DOCS](#)

learning_rate (Double): 0.01 - 1

momentum (Double): 0 - 1

num_neurons (Discrete): 64-128-512

[ADD NEW PARAMETER](#)

Metric to optimize *

accuracy

Goal *

Maximize

Maximum number of trials *

15

How many training trials should be attempted to optimize the specified hyperparameters. Increasing the number of trials generally yields better results but also increases cost. [Learn more](#)

Maximum number of parallel trials *

4

The number of training trials to run concurrently. More parallel trials shortens training time but reduces the effectiveness of the tuning.

Algorithm *

Default

Search algorithms for hyperparameter tuning.

[CONTINUE](#)

Step 3: Configure compute

Upgrade your account to avoid a break in service

Google Cloud Platform

Vertex AI

Dashboard

Datasets

Features

Labeling tasks

Workbench

Pipelines

Training

Experiments

Models

Endpoints

Batch predictions

Metadata

Marketplace

Train new model

Training method

Model details

Training container

Hyperparameters (optional)

Compute and pricing

START TRAINING CANCEL

Model training pricing is based on the length of time spent training, machine types, and any accelerators used. [Learn more](#)

Region

us-central1 (Iowa)

Compute settings

Select the type of virtual machine to use for your worker pool. You can add up to 4 worker pools. To learn about compute costs and how to map your ML framework's roles to specific worker pools, consult the [documentation](#)

Worker pool 0

Machine type *

n1-standard-4 4 vCPUs, 15 GiB memory

Accelerator type

NVIDIA_TESLA_T4

Accelerators can speed up model training that involves intensive compute tasks. [Learn more](#)

Accelerator count

1

Worker count

1

Disk type

SSD

Disk size (GB)

100

[ADD MORE WORKER POOLS \(OPTIONAL\)](#)

Click 'Start Tuning'

Upgrade your account to avoid a break in service

Google Cloud Platform

Vertex AI

Dashboard

Datasets

Features

Labeling tasks

Workbench

Pipelines

Training

Experiments

Models

Endpoints

Batch predictions

Metadata

Marketplace

Train new model

Training method

Model details

Training container

Hyperparameters (optional)

Compute and pricing

START TRAINING

CANCEL

Model training pricing is based on the length of time spent training, machine types, and any accelerators used. [Learn more](#)

Region
us-central1 (Iowa)

Compute settings

Select the type of virtual machine to use for your worker pool. You can add up to 4 worker pools. To learn about compute costs and how to map your ML framework's roles to specific worker pools, consult the [documentation](#)

Worker pool 0

Machine type *
n1-standard-4, 4 vCPUs, 15 GiB memory

Accelerator type
NVIDIA_TESLA_T4
Accelerators can speed up model training that involves intensive compute tasks. [Learn more](#)

Accelerator count
1

Worker count
1

Disk type
SSD

Disk size (GB)
100

ADD MORE WORKER POOLS (OPTIONAL)

HyperParameter Tuning job is in progress:

Upgrade your account to avoid a break in service (\$159.65 credit and 6 days left in your trial)

LEARN MORE

UPGRADE

Google Cloud Platform

CMPE260

Search products and resources

2

Vertex AI

Training

CREATE

REFRESH

TRAINING PIPELINES

CUSTOM JOBS

HYPERPARAMETER TUNING JOBS

Hyperparameter tuning searches for the best combination of hyperparameter values by optimizing metric values across a series of trials. Hyperparameter tuning is only used by custom-trained models and not AutoML models. [Learn More](#)

Region
us-central1 (Iowa)

Filter Enter a property name

Name	ID	Status	Job type	Model type	Created	Elapsed time	Labels
horses-humans-hypertune-hyperparameter-tuning-job	2213537908550270976	Running	Hyperparameter tuning job	—	Dec 8, 2021, 10:24:04 AM	19 sec	—

Cleanup

Stop the notebook instance

Google Cloud Platform

CMPE260

Search products and resources

Vertex AI

Dashboard

Datasets

Features

Labeling tasks

Workbench

Pipelines

Training

Experiments

Models

Endpoints

Batch predictions

Metadata

Marketplace

Notebooks

NEW NOTEBOOK

REFRESH

START

STOP

RESET

DELETE

SHOW INFO PANEL

MANAGED NOTEBOOKS

USER-MANAGED NOTEBOOKS

EXECUTIONS

SCHEDULES

As of the M80 DLVM release, all environments will include JupyterLab 3.x by default. To continue using an existing environment's JupyterLab 1.x version, disable auto-upgrade (if enabled) and do not manually upgrade the environment to a new environment version. To create new Notebooks with JupyterLab 1.x installed, see creating specific versions of Notebooks.

Notebooks have JupyterLab pre-installed and are configured with GPU-enabled machine learning frameworks. Learn more

Filter

Enter property name or value

Filter icon

<input checked="" type="checkbox"/>	<input type="checkbox"/>	Notebook name <input type="text"/>	Zone	Auto-upgrade	Environment	Machine type	GPUs	Permission	Last modified	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	tensorflow-2.7-20211208-09102	OPEN JUPYTERLAB	us-central1-a	—	TensorFlow:2.7	4 vCPUs, 15 GB RAM	None	Service account	Dec 8, 2021, 9:14:15 AM

Delete the Storage Bucket

Google Cloud Platform

CMPE260

Search products and resources

Cloud Storage

Browser

CREATE BUCKET

DELETE

REFRESH

SHOW INFO PANEL

Browser

Monitoring

Settings

Filter

Filter buckets

Filter icon

<input checked="" type="checkbox"/>	Name <input type="text"/>	Created	Location type	Location	Default storage class	Last modified	Public access	Access control
<input checked="" type="checkbox"/>	artifacts.cmpe260-334300.appspot.c...	Dec 8, 2021, 10:07:17 AM	Multi-region	us (multiple re...	Standard	Dec 8, 2021, 10:07:17 AM	Subject to object ACLs	Fine-grained
<input type="checkbox"/>	cloud-ai-platform-65473780-0280-46f...	Dec 6, 2021, 4:17:06 PM	Region	us-central1 (lo...	Regional	Dec 6, 2021, 4:17:06 PM	Subject to object ACLs	Fine-grained