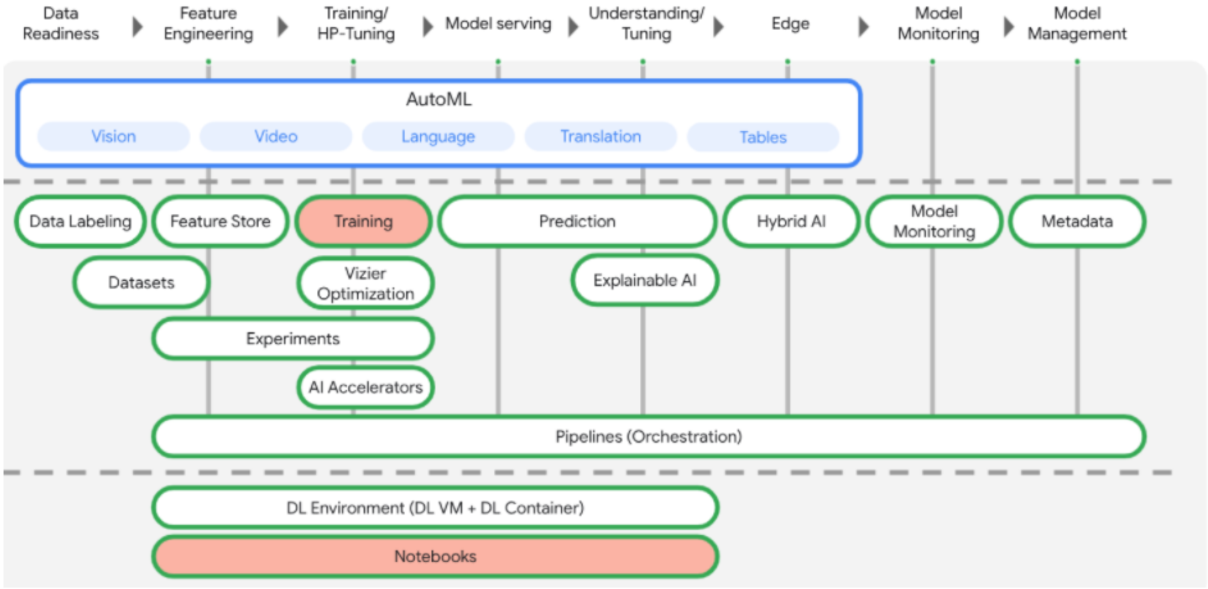
a) Vertex AI to run a hyperparameter tuning job for a TensorFlow model

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

Objectives:

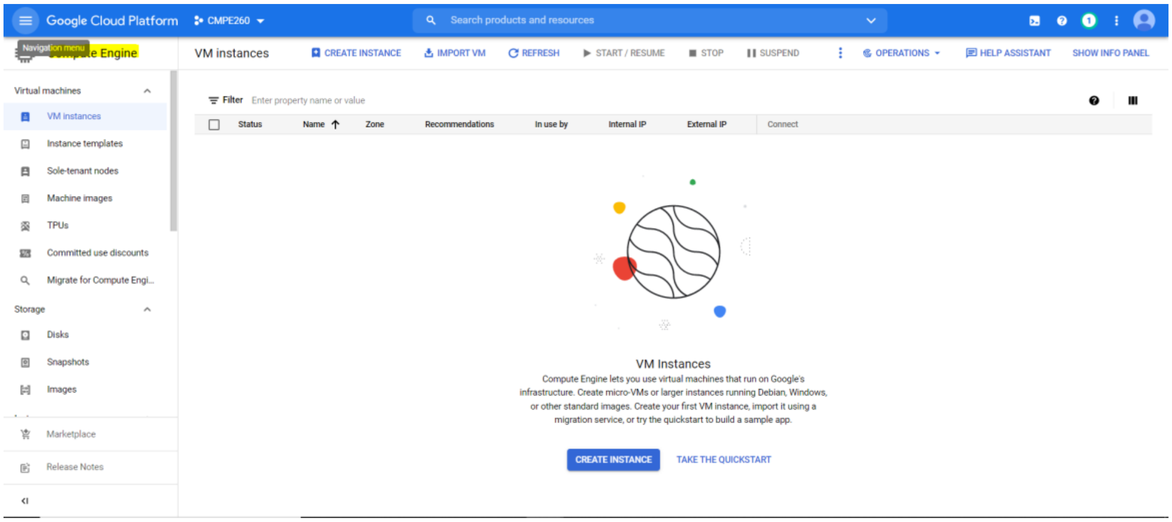
- Modify training application code for hyperparameter tuning
- Configure and launch a hyperparameter tuning job from the Vertex AI UI
- Configure and launch a hyperparameter tuning job with the Vertex SDK

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

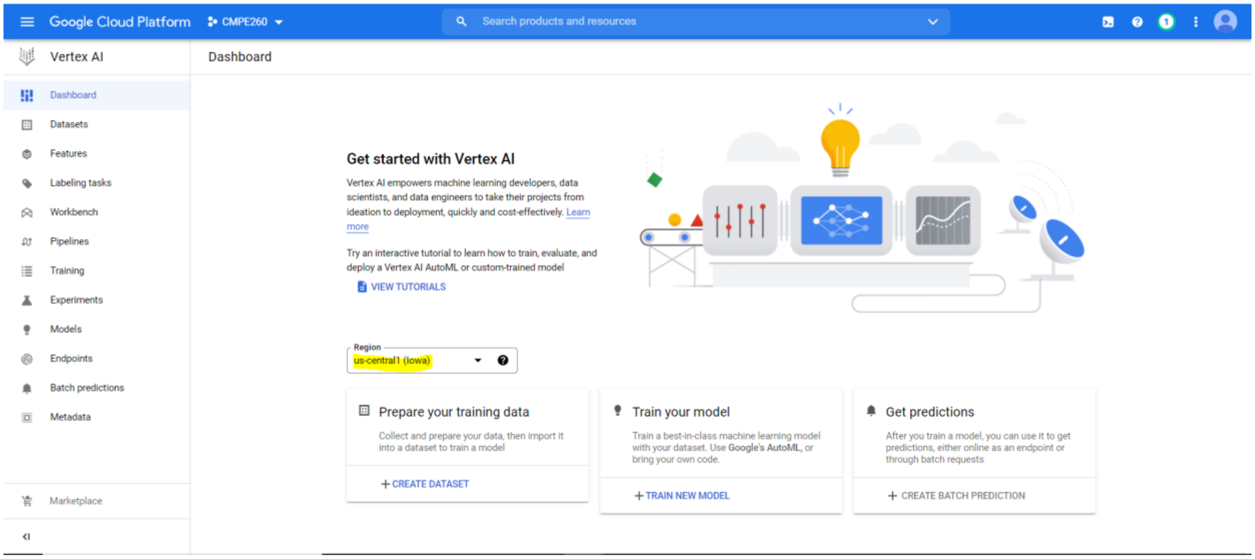
Step 1: Enable the Compute Engine API



VM Instances

Compute Engine lets you use virtual machines that run on Google's infrastructure. Create micro-VMs or larger instances running Debian, Windows, or other standard images. Create your first VM instance, import it using a migration service, or try the quickstart to build a sample app.

Step 2: Enable the Vertex AI API



Get started with Vertex AI

Vertex AI empowers machine learning developers, data scientists, and data engineers to take their projects from ideation to deployment, quickly and cost-effectively. Learn more

Try an interactive tutorial to learn how to train, evaluate, and deploy a Vertex AI AutoML or custom-trained model

Region: us-central1 (Iowa)

Prepare your training data

Collect and prepare your data, then import it into a dataset to train a model

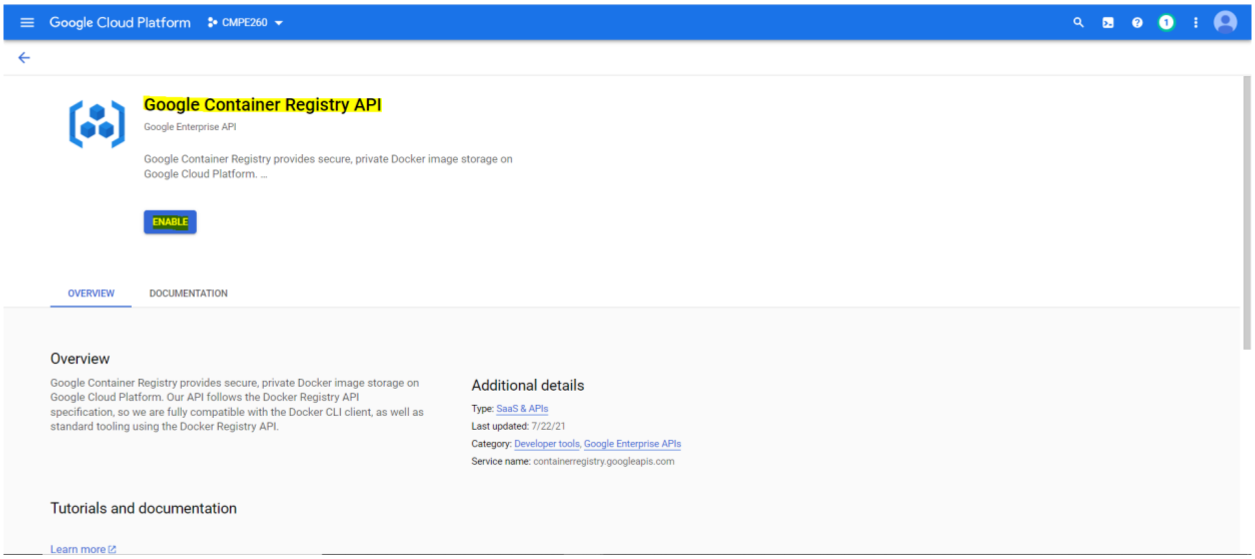
Train your model

Train a best-in-class machine learning model with your dataset. Use Google's AutoML, or bring your own code.

Get predictions

After you train a model, you can use it to get predictions, either online as an endpoint or through batch requests

Step 3: Enable the Container Registry API



Google Container Registry API

Google Enterprise API

Google Container Registry provides secure, private Docker image storage on Google Cloud Platform. ...

ENABLE

OVERVIEW DOCUMENTATION

Overview

Google Container Registry provides secure, private Docker image storage on Google Cloud Platform. Our API follows the Docker Registry API specification, so we are fully compatible with the Docker CLI client, as well as standard tooling using the Docker Registry API.

Additional details

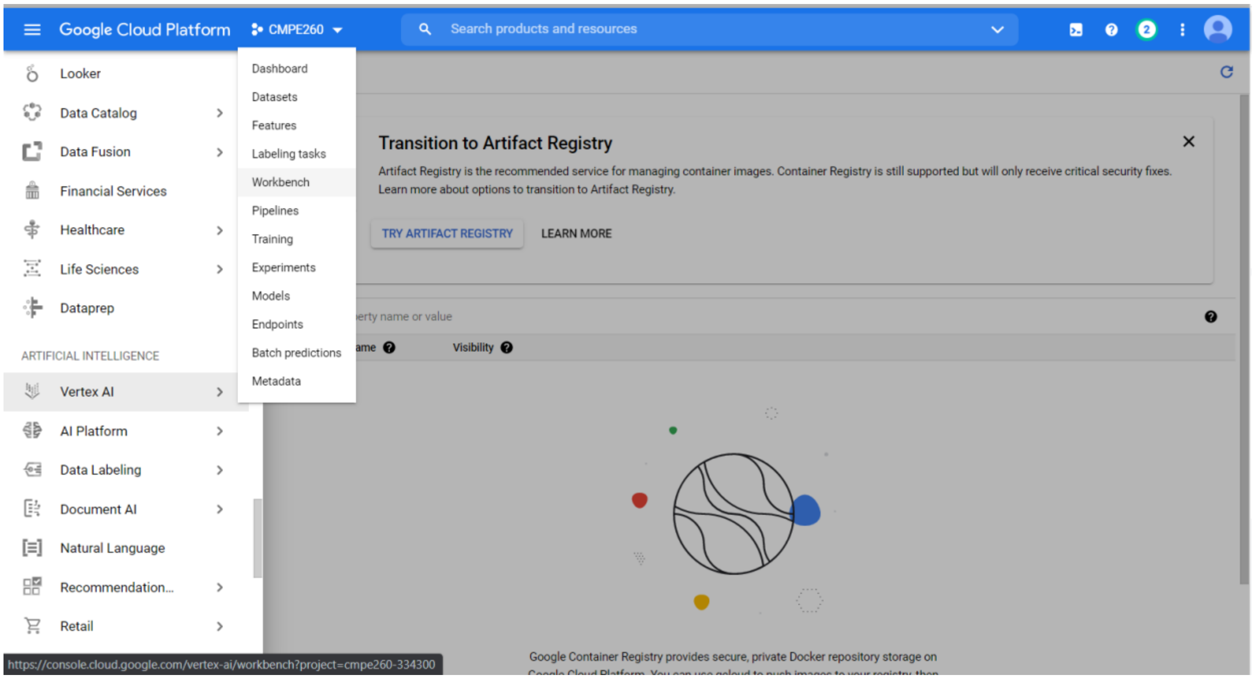
Type: SaaS & APIs
Last updated: 7/22/21
Category: Developer tools, Google Enterprise APIs
Service name: containerregistry.googleapis.com

Tutorials and documentation

Learn more

Step 4: Create a Vertex AI Workbench instance

In the Left Nav, select Vertex AI → Workbench



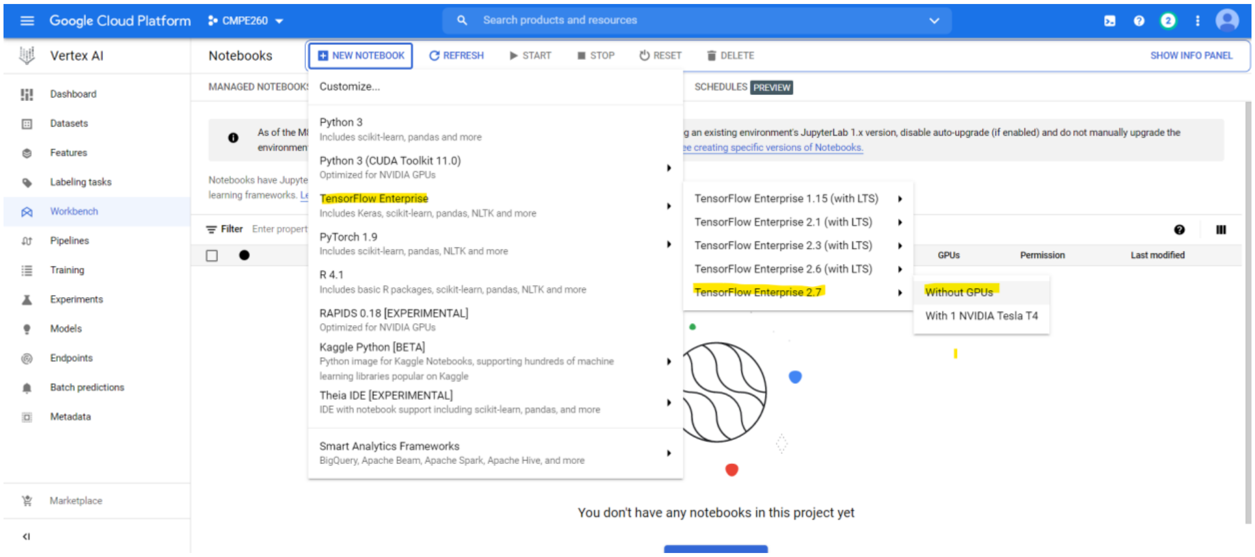
Transition to Artifact Registry

Artifact Registry is the recommended service for managing container images. Container Registry is still supported but will only receive critical security fixes. Learn more about options to transition to Artifact Registry.

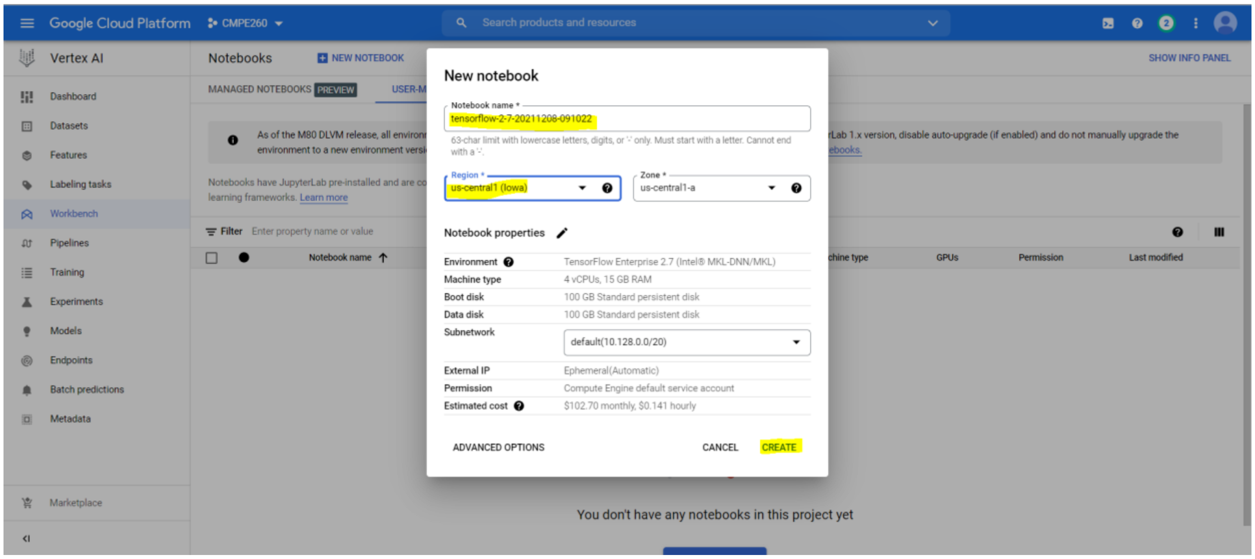
Google Container Registry provides secure, private Docker repository storage on

https://console.cloud.google.com/vertex-ai/workbench?project=cmpe260-334300

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


Google Cloud Platform
CMPE260
Search products and resources
Vertex AI
Notebooks
NEW NOTEBOOK
REFRESH
START
STOP
RESET
DELETE
SHOW INFO PANEL
Dashboard
Datasets
Features
Labeling tasks
Workbench
Pipelines
Training
Experiments
Models
Endpoints
Batch predictions
Metadata
Marketplace
MANAGED NOTEBOOKS
SCHEDULES PREVIEW
Customize...
Python 3
Includes scikit-learn, pandas and more
Python 3 (CUDA Toolkit 11.0)
Optimized for NVIDIA GPUs
TensorFlow Enterprise
Includes Keras, scikit-learn, pandas, NLTK and more
PyTorch 1.9
Includes scikit-learn, pandas, NLTK and more
R 4.1
Includes basic R packages, scikit-learn, pandas, NLTK and more
RAPIDS 0.18 [EXPERIMENTAL]
Optimized for NVIDIA GPUs
Kaggle Python [BETA]
Python image for Kaggle Notebooks, supporting hundreds of machine learning libraries popular on Kaggle
Theia IDE [EXPERIMENTAL]
IDE with notebook support including scikit-learn, pandas, and more
Smart Analytics Frameworks
BigQuery, Apache Beam, Apache Spark, Apache Hive, and more
TensorFlow Enterprise 1.15 (with LTS)
TensorFlow Enterprise 2.1 (with LTS)
TensorFlow Enterprise 2.3 (with LTS)
TensorFlow Enterprise 2.6 (with LTS)
TensorFlow Enterprise 2.7
Without GPUs
With 1 NVIDIA Tesla T4
Filter Enter property
GPUs
Permission
Last modified
You don't have any notebooks in this project yet

Enter the name, region and click ‘Create’



New notebook

Notebook name *
tensorflow-2-7-20211208-091022

63-char limit with lowercase letters, digits, or '-' only. Must start with a letter. Cannot end with a '-'.

Region *
us-central1 (Iowa)

Zone *
us-central1-a

Notebook properties

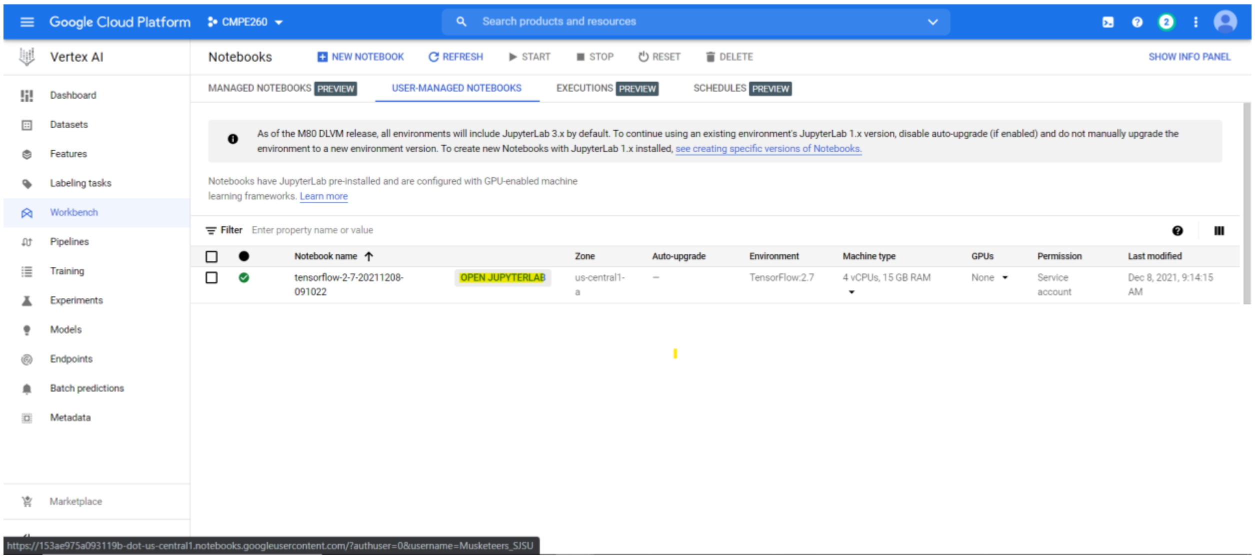
Environment	TensorFlow Enterprise 2.7 (Intel® MKL-DNN/MKL)
Machine type	4 vCPUs, 15 GB RAM
Boot disk	100 GB Standard persistent disk
Data disk	100 GB Standard persistent disk
Subnetwork	default(10.128.0.0/20)
External IP	Ephemeral(Automatic)
Permission	Compute Engine default service account
Estimated cost	$102.70 monthly, $0.141 hourly

ADVANCED OPTIONS CANCEL 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’



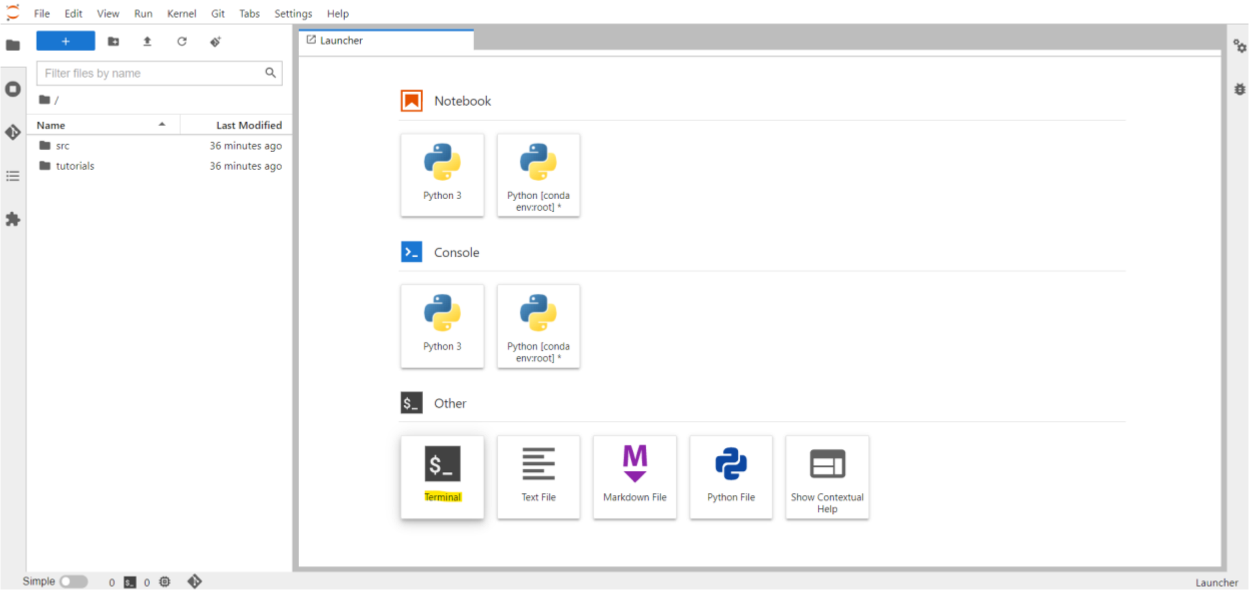
As of the M80 DLVM release, all environments will include JupyterLab 3.x by default. To continue using an existing environment's JupyterLab 1.x version, disable auto-upgrade (if enabled) and do not manually upgrade the environment to a new environment version. To create new Notebooks with JupyterLab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

□	●	Notebook name ↑		Zone	Auto-upgrade	Environment	Machine type	GPUs	Permission	Last modified
□	✓	tensorflow-2-7-20211208-091022	OPEN JUPYTERLAB	us-central1-a	–	TensorFlow:2.7	4 vCPUs, 15 GB RAM	None	Service account	Dec 8, 2021, 9:14:15 AM

https://153ae975a093119b-dot-us-central1.notebooks.googleusercontent.com/?authuser=0&username=Musketeers_SJSU

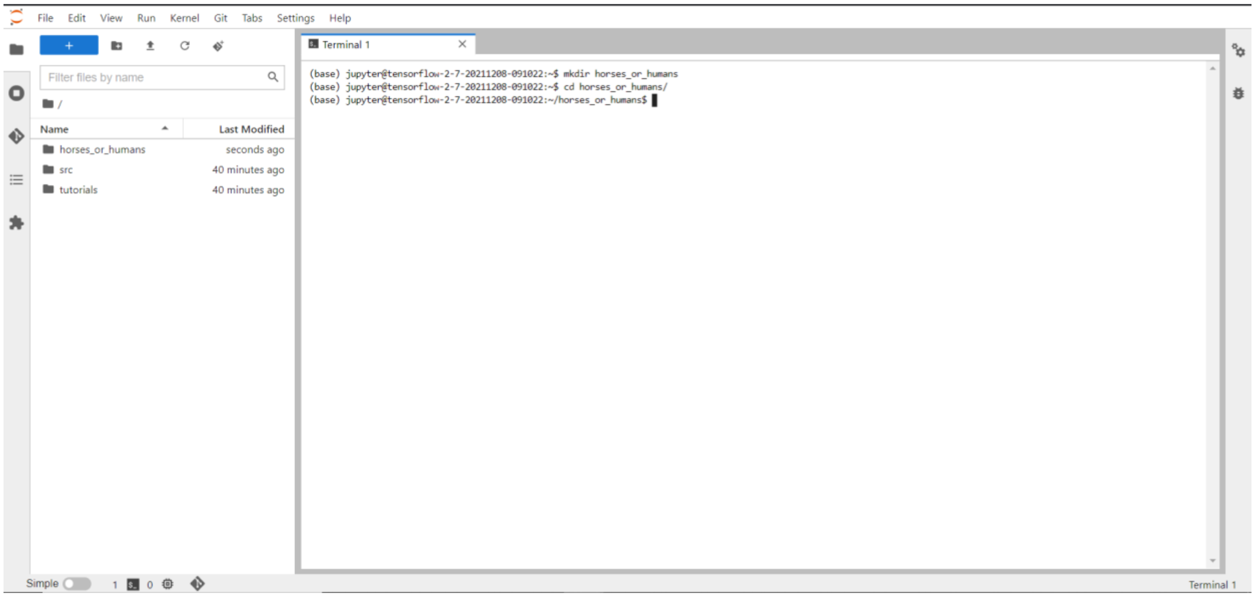
2. Click ‘Terminal’



Create Docker file:

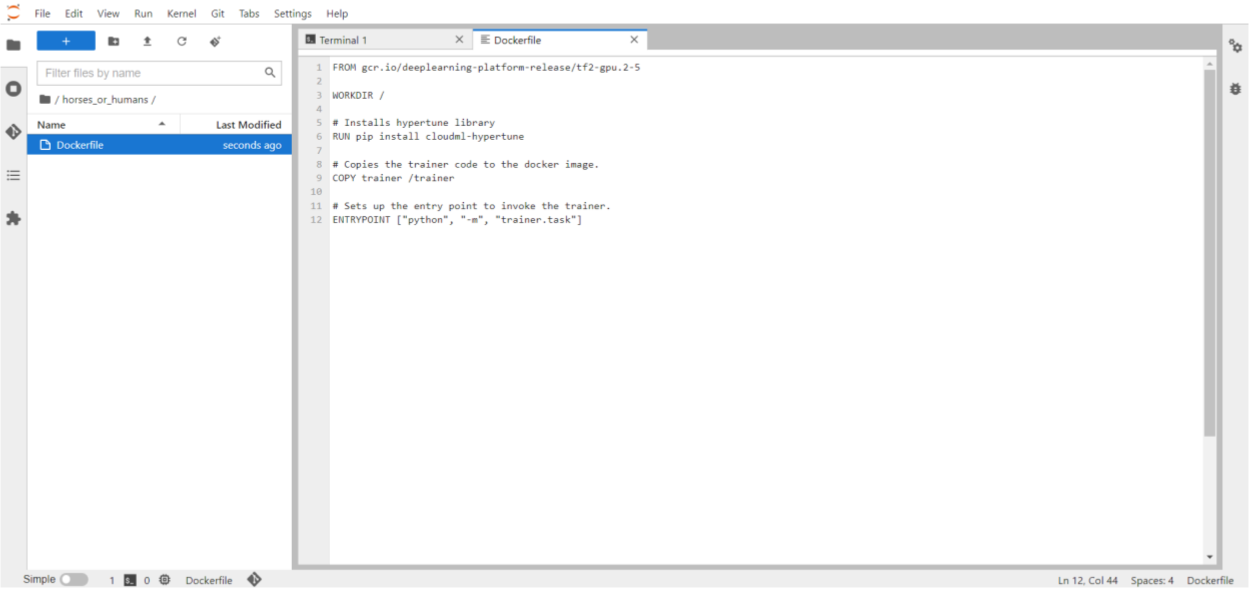
Run the following commands to create a folder

mkdir horses_or_humans
cd horses_or_humans
touch Dockerfile



Name	Last Modified
horses_or_humans	seconds ago
src	40 minutes ago
tutorials	40 minutes ago

```
(base) jupyter@tensorflow-2-7-20211208-091022:~$ mkdir horses_or_humans
(base) jupyter@tensorflow-2-7-20211208-091022:~$ cd horses_or_humans/
(base) jupyter@tensorflow-2-7-20211208-091022:~/horses_or_humans$
```



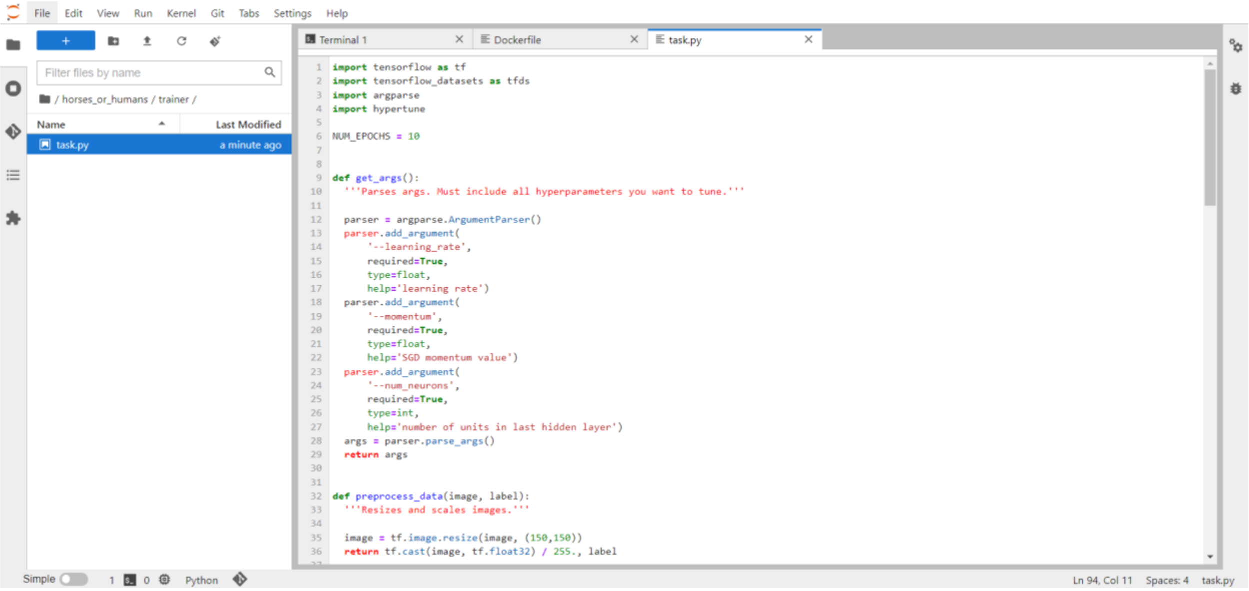
Name	Last Modified
Dockerfile	seconds ago

```
FROM gcr.io/deeplearning-platform-release/tf2-gpu.2-5

WORKDIR /

# Installs hypertune library
RUN pip install cloudml-hypertune

# Copies the trainer code to the docker image.
COPY trainer /trainer

# Sets up the entry point to invoke the trainer.
ENTRYPOINT ["python", "-m", "trainer.task"]
```

Add model training code

Run the following commands:

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



```python
import tensorflow as tf
import tensorflow_datasets as tfds
import argparse
import hypertune

NUM_EPOCHS = 10


def get_args():
  '''Parses args. Must include all hyperparameters you want to tune.'''

  parser = argparse.ArgumentParser()
  parser.add_argument(
      '--learning_rate',
      required=True,
      type=float,
      help='learning rate')
  parser.add_argument(
      '--momentum',
      required=True,
      type=float,
      help='SGD momentum value')
  parser.add_argument(
      '--num_neurons',
      required=True,
      type=int,
      help='number of units in last hidden layer')
  args = parser.parse_args()
  return args


def preprocess_data(image, label):
  '''Resizes and scales images.'''

  image = tf.image.resize(image, (150,150))
  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.

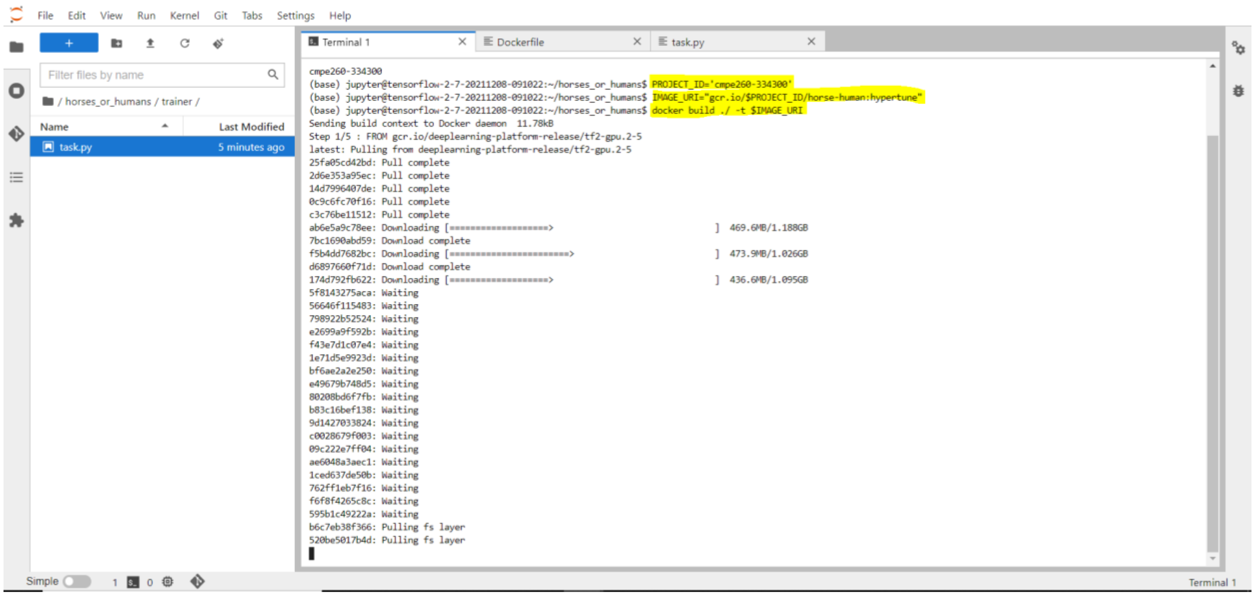
Build the container

Run the following commands:

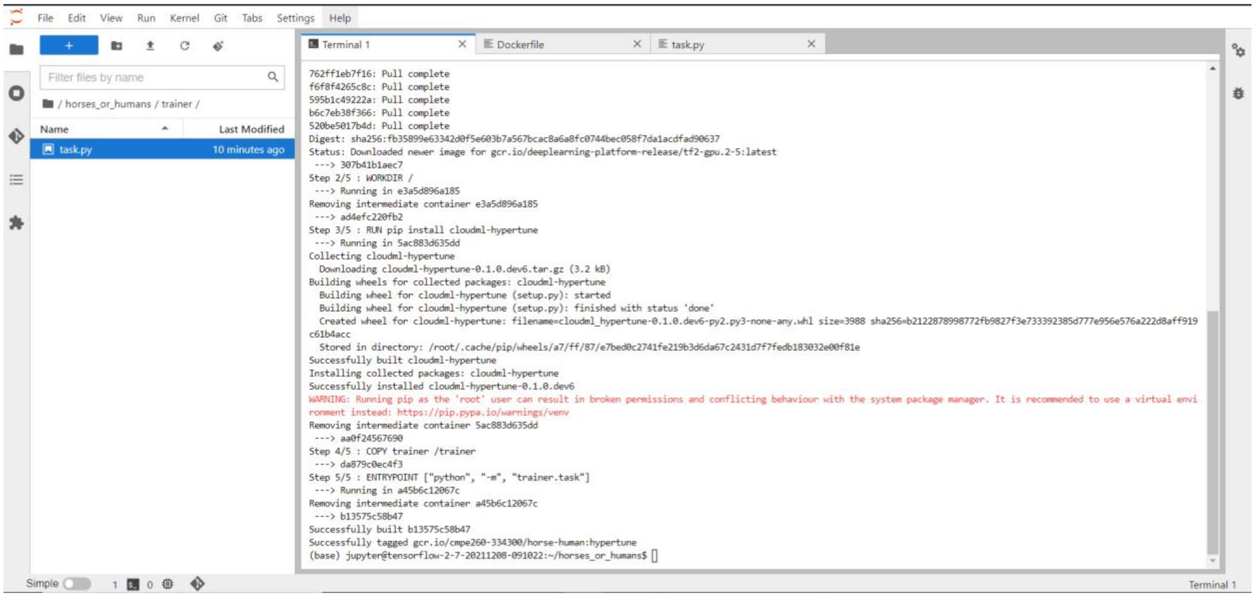
```
gcloud config list --format 'value(core.project)'
PROJECT_ID='cmpe260-334300'
```

IMAGE_URI="gcr.io/$PROJECT_ID/horse-human:hypertune"
docker build ./ -t $IMAGE_URI

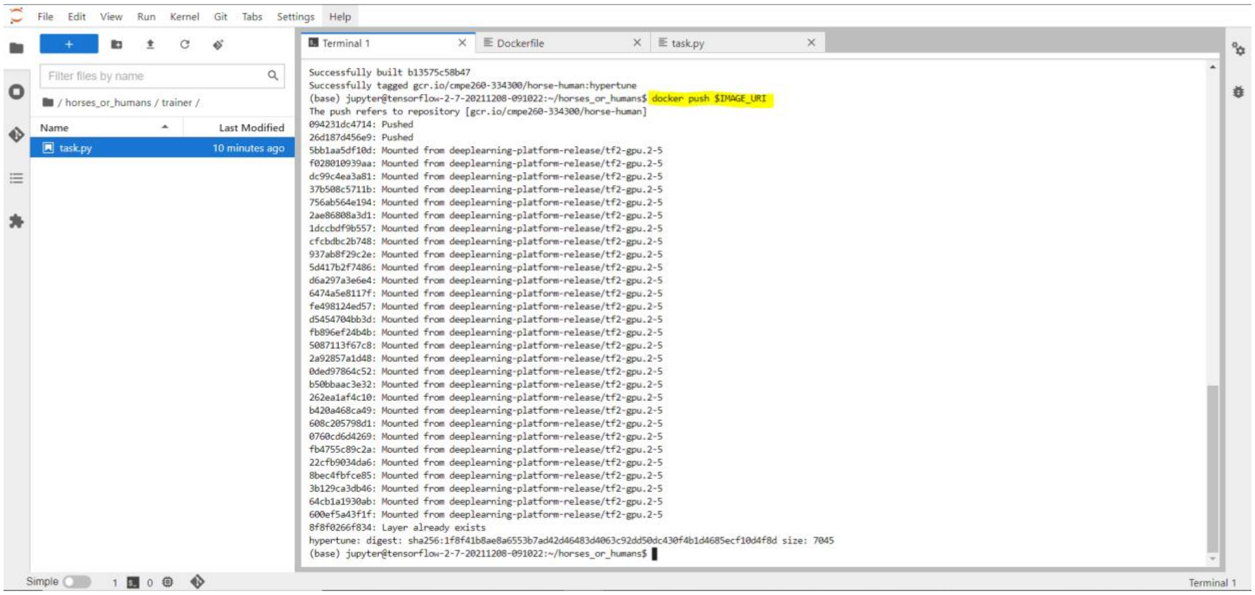
docker push $IMAGE_URI



```
cmpe260-334300
(base) jupyter@tensorflow-2-7-20211208-091022:~/horses_or_humans$ PROJECT_ID='cmpe260-334300'
(base) jupyter@tensorflow-2-7-20211208-091022:~/horses_or_humans$ IMAGE_URI="gcr.io/$PROJECT_ID/horse-human:hypertune"
(base) jupyter@tensorflow-2-7-20211208-091022:~/horses_or_humans$ docker build ./ -t $IMAGE_URI
Sending build context to Docker daemon  11.78kB
Step 1/5 : FROM gcr.io/deeplearning-platform-release/tf2-gpu.2-5
latest: Pulling from deeplearning-platform-release/tf2-gpu.2-5
25fa05cd42bd: Pull complete
2d6e353a95ec: Pull complete
14d7996407de: Pull complete
0c9c6fc70f16: Pull complete
c3c76be11512: Pull complete
ab6e5a9c78ee: Downloading [===================>                               ]  469.6MB/1.188GB
7bc1690abd59: Download complete
f5b4dd7682bc: Downloading [=======================>                           ]  473.9MB/1.026GB
d6897660f71d: Download complete
174d792fb622: Downloading [===================>                               ]  436.6MB/1.095GB
5f8143275aca: Waiting
56646f115483: Waiting
798922b52524: Waiting
e2699a9f592b: Waiting
f43e7d1c07e4: Waiting
1e71d5e9923d: Waiting
bf6ae2a2e250: Waiting
e49679b748d5: Waiting
80208bd6f7fb: Waiting
b83c16bef138: Waiting
9d1427033824: Waiting
c0028679f003: Waiting
09c222e7ff04: Waiting
ae6048a3aec1: Waiting
1ced637de50b: Waiting
762ff1eb7f16: Waiting
f6f8f4265c8c: Waiting
595b1c49222a: Waiting
b6c7eb38f366: Pulling fs layer
520be5017b4d: Pulling fs layer
█
```



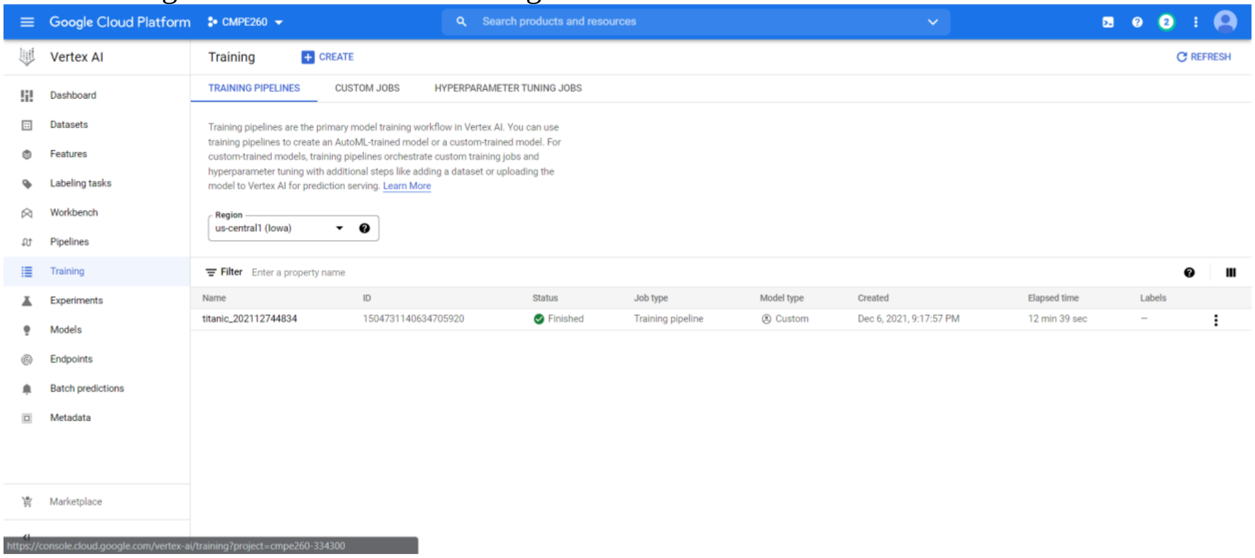
```
762ff1eb7f16: Pull complete
f6f8f4265c8c: Pull complete
595b1c49222a: Pull complete
b6c7eb38f366: Pull complete
520be5017b4d: Pull complete
Digest: sha256:fb35899e63342d0f5e603b7a567bcac8a6a8fc0744bec058f7da1acdfad90637
Status: Downloaded newer image for gcr.io/deeplearning-platform-release/tf2-gpu.2-5:latest
 ---> 307b41b1aec7
Step 2/5 : WORKDIR /
 ---> Running in e3a5d896a185
Removing intermediate container e3a5d896a185
 ---> ad4efc220fb2
Step 3/5 : RUN pip install cloudml-hypertune
 ---> Running in 5ac883d635dd
Collecting cloudml-hypertune
  Downloading cloudml-hypertune-0.1.0.dev6.tar.gz (3.2 kB)
Building wheels for collected packages: cloudml-hypertune
  Building wheel for cloudml-hypertune (setup.py): started
  Building wheel for cloudml-hypertune (setup.py): finished with status 'done'
  Created wheel for cloudml-hypertune: filename=cloudml_hypertune-0.1.0.dev6-py2.py3-none-any.whl size=3988 sha256=b2122878998772fb9827f3e733392385d777e956e576a222d8aff919
c61b4acc
  Stored in directory: /root/.cache/pip/wheels/a7/ff/87/e7bed0c2741fe219b3d6da67c2431d7f7fedb183032e00f81e
Successfully built cloudml-hypertune
Installing collected packages: cloudml-hypertune
Successfully installed cloudml-hypertune-0.1.0.dev6
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual envi
ronment instead: https://pip.pypa.io/warnings/venv
Removing intermediate container 5ac883d635dd
 ---> aa0f24567690
Step 4/5 : COPY trainer /trainer
 ---> da879c0ec4f3
Step 5/5 : ENTRYPOINT ["python", "-m", "trainer.task"]
 ---> Running in a45b6c12067c
Removing intermediate container a45b6c12067c
 ---> b13575c58b47
Successfully built b13575c58b47
Successfully tagged gcr.io/cmpe260-334300/horse-human:hypertune
(base) jupyter@tensorflow-2-7-20211208-091022:~/horses_or_humans$ []
```



```
Successfully built b13575c58b47
Successfully tagged gcr.io/cmpe260-334300/horse-human:hypertune
(base) jupyter@tensorflow-2-7-20211208-091022:~/horses_or_humans$ docker push $IMAGE_URI
The push refers to repository [gcr.io/cmpe260-334300/horse-human]
094231dc4714: Pushed
26d187d456e9: Pushed
5bb1aa5df10d: Mounted from deeplearning-platform-release/tf2-gpu.2-5
f028010939aa: Mounted from deeplearning-platform-release/tf2-gpu.2-5
dc99c4ea3a81: Mounted from deeplearning-platform-release/tf2-gpu.2-5
37b508c5711b: Mounted from deeplearning-platform-release/tf2-gpu.2-5
756ab564e194: Mounted from deeplearning-platform-release/tf2-gpu.2-5
2ae86808a3d1: Mounted from deeplearning-platform-release/tf2-gpu.2-5
1dccbdf9b557: Mounted from deeplearning-platform-release/tf2-gpu.2-5
cfcbdbc2b748: Mounted from deeplearning-platform-release/tf2-gpu.2-5
937ab8f29c2e: Mounted from deeplearning-platform-release/tf2-gpu.2-5
5d417b2f7486: Mounted from deeplearning-platform-release/tf2-gpu.2-5
d6a297a3e6e4: Mounted from deeplearning-platform-release/tf2-gpu.2-5
6474a5e8117f: Mounted from deeplearning-platform-release/tf2-gpu.2-5
fe498124ed57: Mounted from deeplearning-platform-release/tf2-gpu.2-5
d5454704bb3d: Mounted from deeplearning-platform-release/tf2-gpu.2-5
fb896ef24b4b: Mounted from deeplearning-platform-release/tf2-gpu.2-5
5087113f67c8: Mounted from deeplearning-platform-release/tf2-gpu.2-5
2a92857a1d48: Mounted from deeplearning-platform-release/tf2-gpu.2-5
0ded97864c52: Mounted from deeplearning-platform-release/tf2-gpu.2-5
b50bbaac3e32: Mounted from deeplearning-platform-release/tf2-gpu.2-5
262ea1af4c10: Mounted from deeplearning-platform-release/tf2-gpu.2-5
b420a468ca49: Mounted from deeplearning-platform-release/tf2-gpu.2-5
608c205798d1: Mounted from deeplearning-platform-release/tf2-gpu.2-5
0760cd6d4269: Mounted from deeplearning-platform-release/tf2-gpu.2-5
fb4755c89c2a: Mounted from deeplearning-platform-release/tf2-gpu.2-5
22cfb9034da6: Mounted from deeplearning-platform-release/tf2-gpu.2-5
8bec4fbfce85: Mounted from deeplearning-platform-release/tf2-gpu.2-5
3b129ca3db46: Mounted from deeplearning-platform-release/tf2-gpu.2-5
64cb1a1930ab: Mounted from deeplearning-platform-release/tf2-gpu.2-5
600ef5a43f1f: Mounted from deeplearning-platform-release/tf2-gpu.2-5
8f8f0266f834: Layer already exists
hypertune: digest: sha256:1f8f41b8ae8a6553b7ad42d46483d4063c92dd50dc430f4b1d4685ecf10d4f8d size: 7045
(base) jupyter@tensorflow-2-7-20211208-091022:~/horses_or_humans$ 
```

Run a hyperparameter tuning job on Vertex AI

1. Navigate to Vertex AI → Training



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

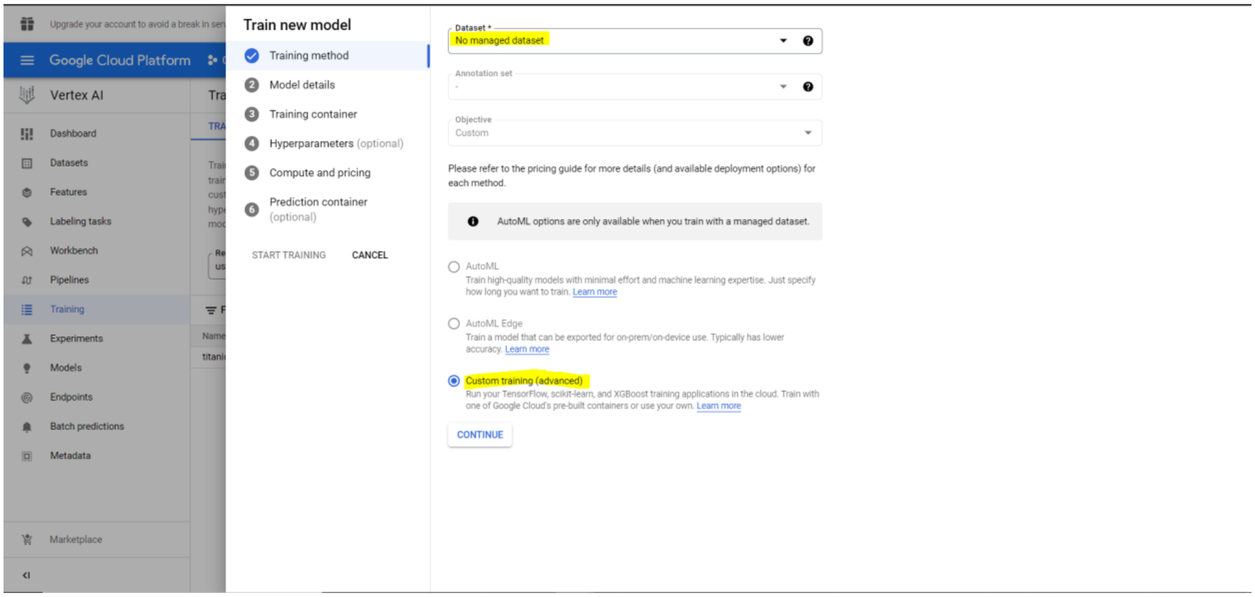
Region: us-central1 (Iowa)

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	–

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**



Dataset *
No managed dataset

Annotation set
-

Objective
Custom

Please refer to the pricing guide for more details (and available deployment options) for each method.

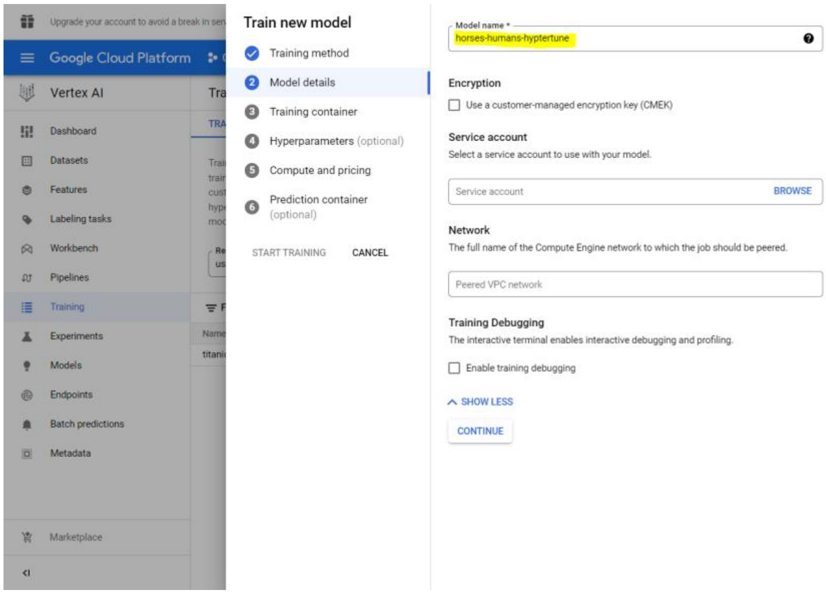
AutoML options are only available when you train with a managed dataset.

○ AutoML
Train high-quality models with minimal effort and machine learning expertise. Just specify how long you want to train. Learn more

○ AutoML Edge
Train a model that can be exported for on-prem/on-device use. Typically has lower accuracy. Learn more

● Custom training (advanced)
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. Learn more

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



Model name *
horses-humans-hyptertune

Encryption
□ Use a customer-managed encryption key (CMEK)

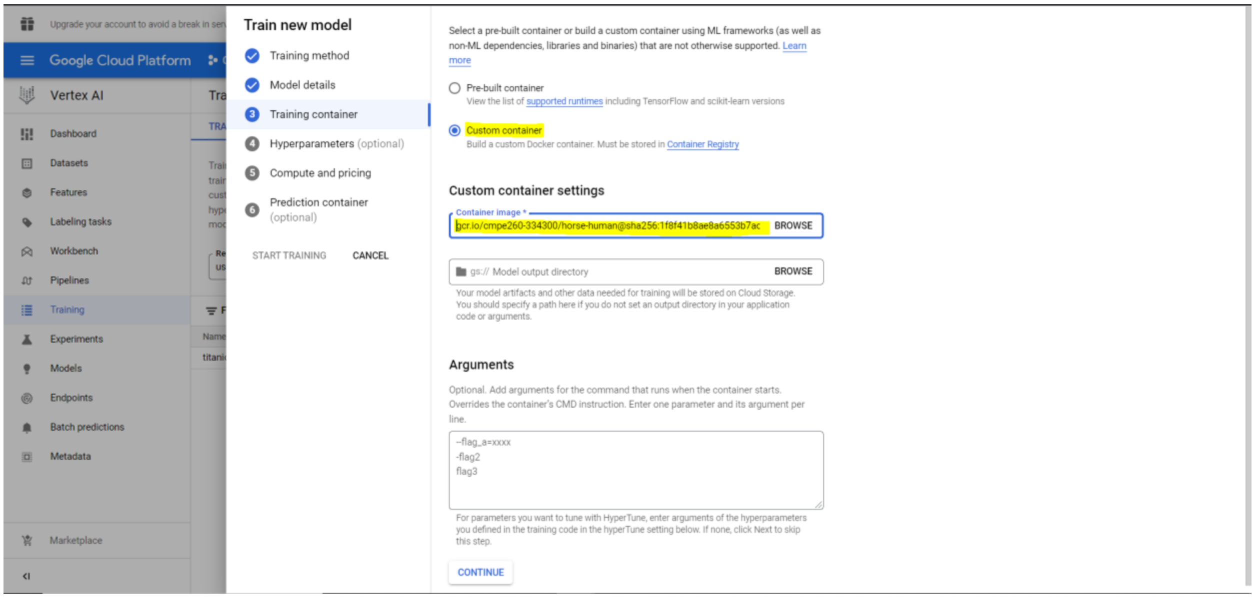
Service account
Select a service account to use with your model.
Service account

Network
The full name of the Compute Engine network to which the job should be peered.
Peered VPC network

Training Debugging
The interactive terminal enables interactive debugging and profiling.
□ Enable training debugging

- Select ‘Custom Container’
- Enter **Container Image: URI to the Image**

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



Select a pre-built container or build a custom container using ML frameworks (as well as non-ML dependencies, libraries and binaries) that are not otherwise supported. Learn more

○ Pre-built container
View the list of supported runtimes including TensorFlow and scikit-learn versions

● Custom container
Build a custom Docker container. Must be stored in Container Registry

Custom container settings

Container image *: gcr.io/cmpe260-334300/horse-human@sha256:1f8f41b8ae8a6553b7ac

gs:// Model output directory

Your model artifacts and other data needed for training will be stored on Cloud Storage. You should specify a path here if you do not set an output directory in your application code or arguments.

Arguments

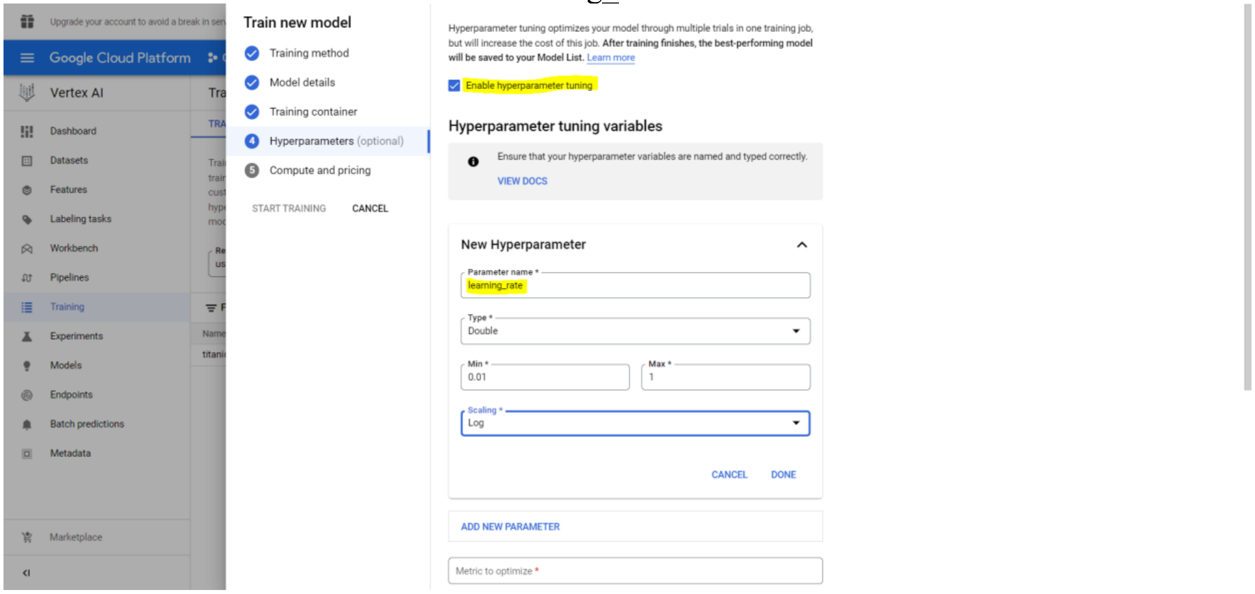
Optional. Add arguments for the command that runs when the container starts. Overrides the container's CMD instruction. Enter one parameter and its argument per line.

```
--flag_a=xxxx
-flag2
flag3
```

For parameters you want to tune with HyperTune, enter arguments of the hyperparameters you defined in the training code in the hyperTune setting below. If none, click Next to skip this step.

Step 2: Configure hyperparameter tuning job

- Select ‘Enable Hyperparameter tuning
- Enter the Parameter details for learning_rate’



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

Ensure that your hyperparameter variables are named and typed correctly.

New Hyperparameter

Parameter name *: learning_rate

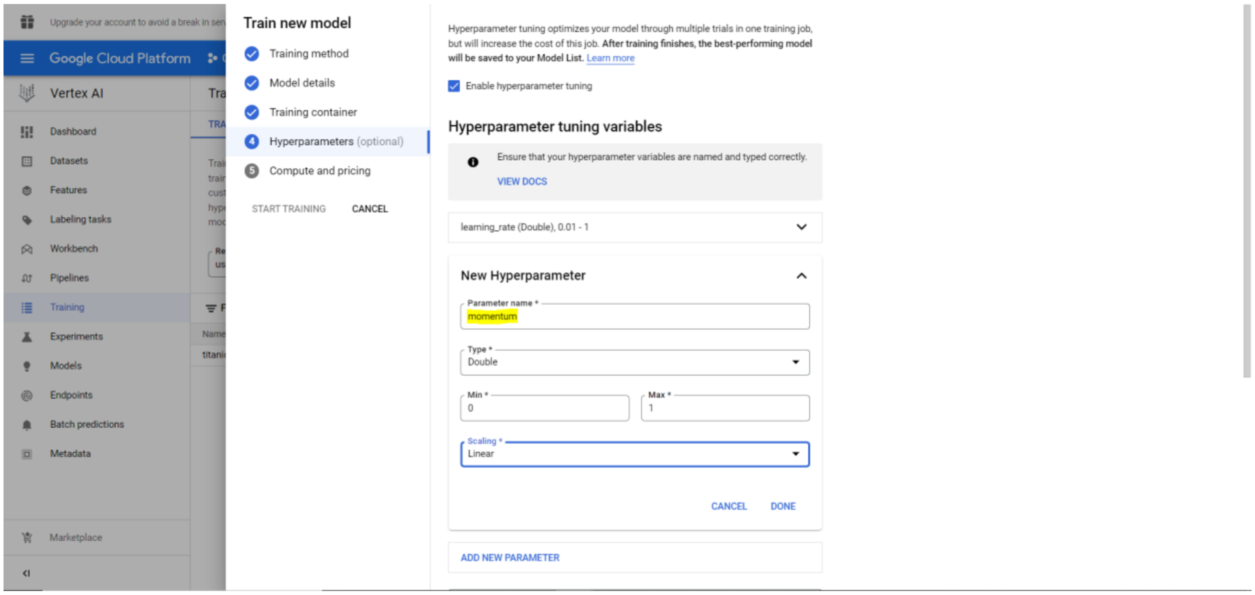
Type *: Double

Min *: 0.01

Max *: 1

Scaling *: Log

Metric to optimize *



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

Ensure that your hyperparameter variables are named and typed correctly.

learning_rate (Double), 0.01 - 1

New Hyperparameter

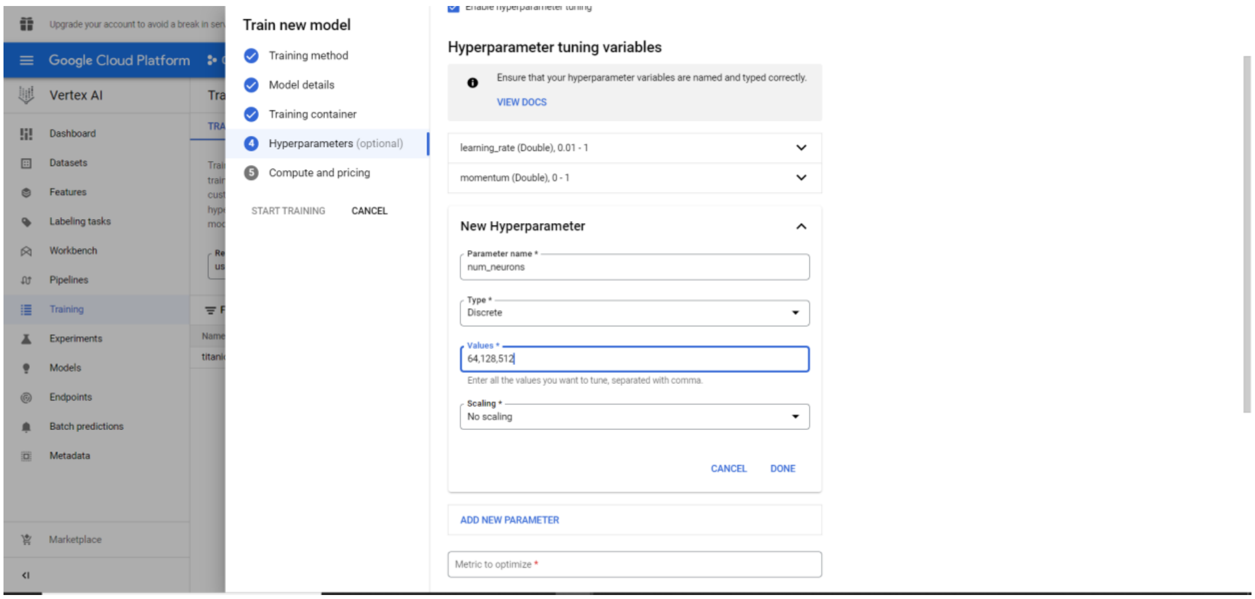
Parameter name *: momentum

Type *: Double

Min *: 0

Max *: 1

Scaling *: Linear



☑ Enable hyperparameter tuning

Hyperparameter tuning variables

Ensure that your hyperparameter variables are named and typed correctly.

learning_rate (Double), 0.01 - 1

momentum (Double), 0 - 1

New Hyperparameter

Parameter name *: num_neurons

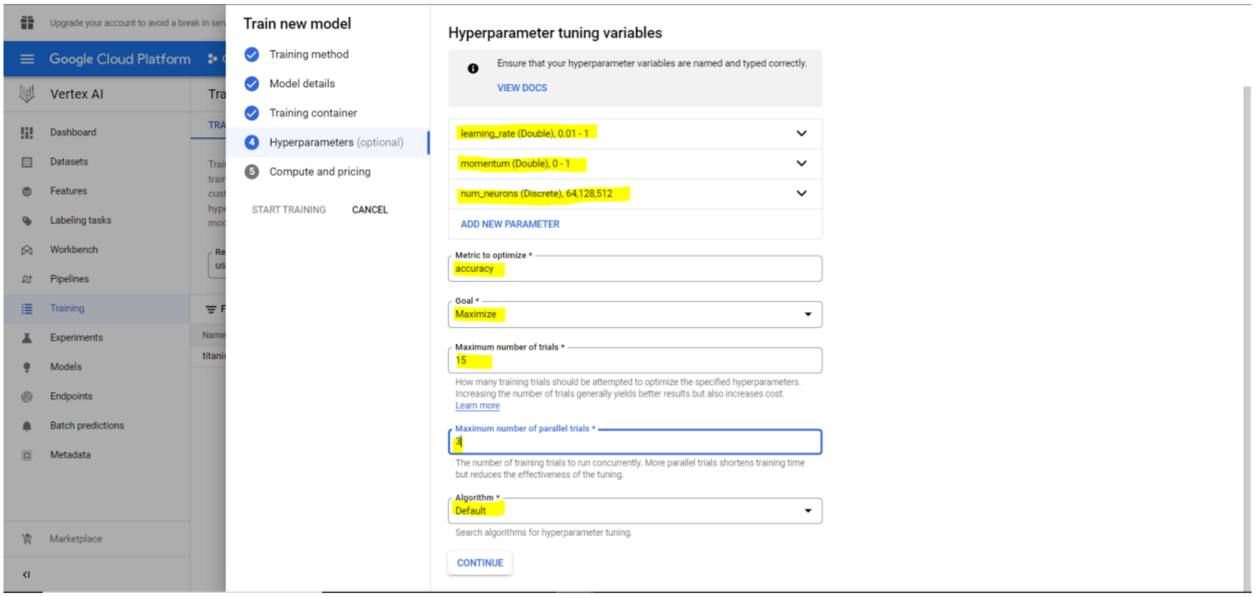
Type *: Discrete

Values *: 64,128,512

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

Scaling *: No scaling

Metric to optimize *



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 *: 3

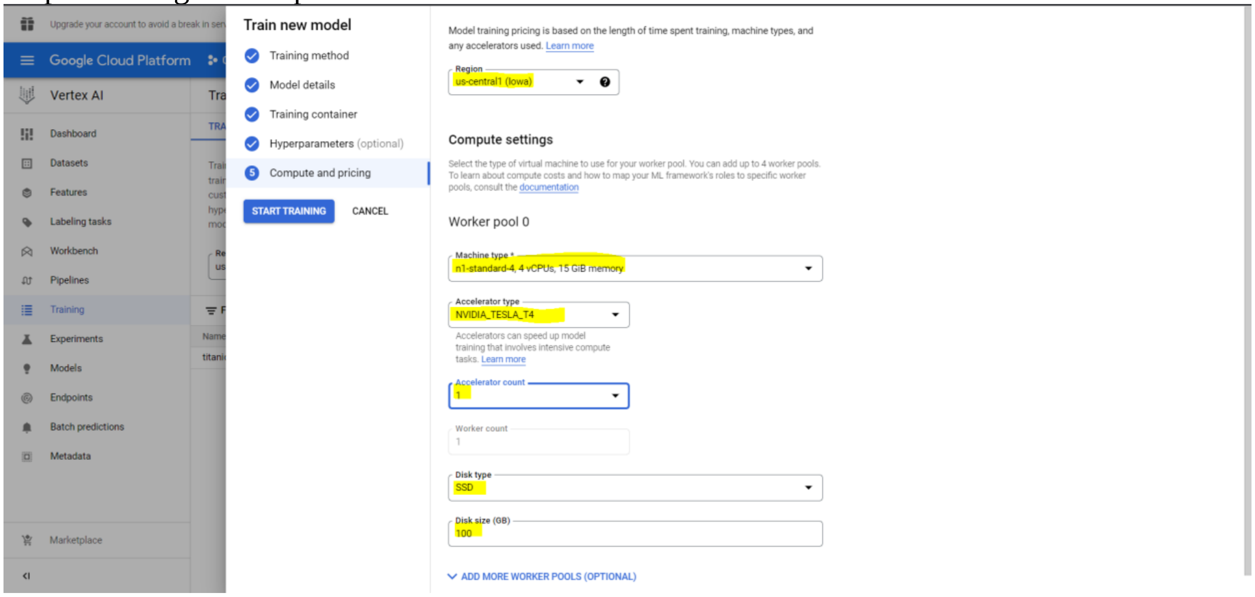
The number of training trials to run concurrently. More parallel trials shortens training time but reduces the effectiveness of the tuning.

Algorithm *: Default

Search algorithms for hyperparameter tuning.

CONTINUE

Step 3: Configure compute



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

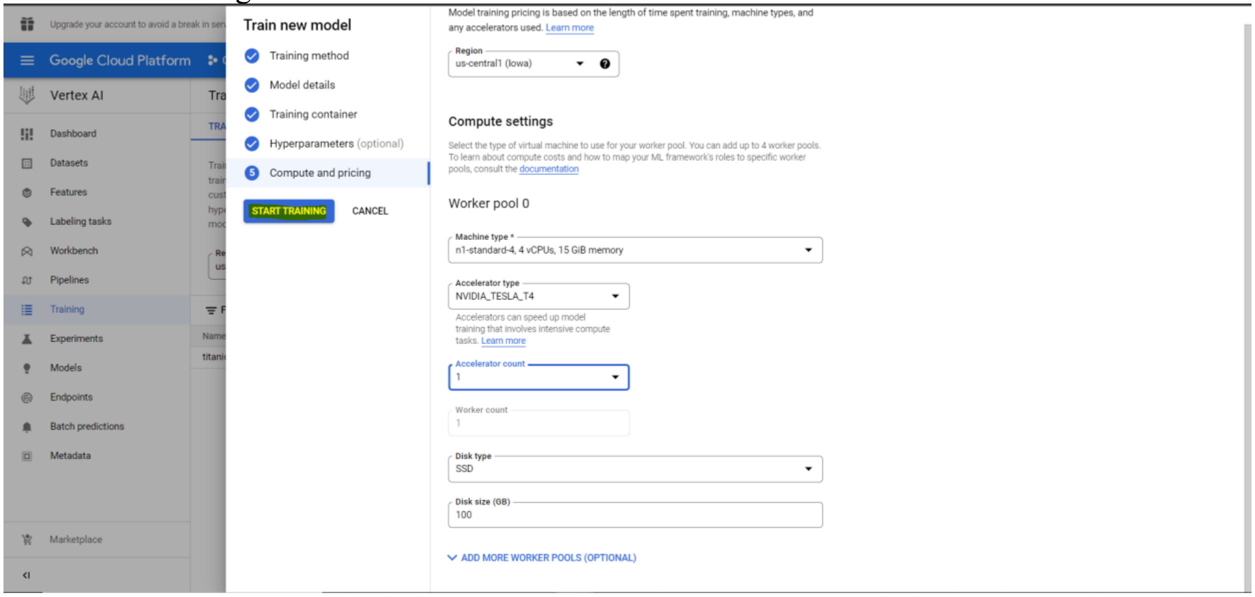
Worker count: 1

Disk type: SSD

Disk size (GB): 100

ADD MORE WORKER POOLS (OPTIONAL)

Click ‘Start Tuning’



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

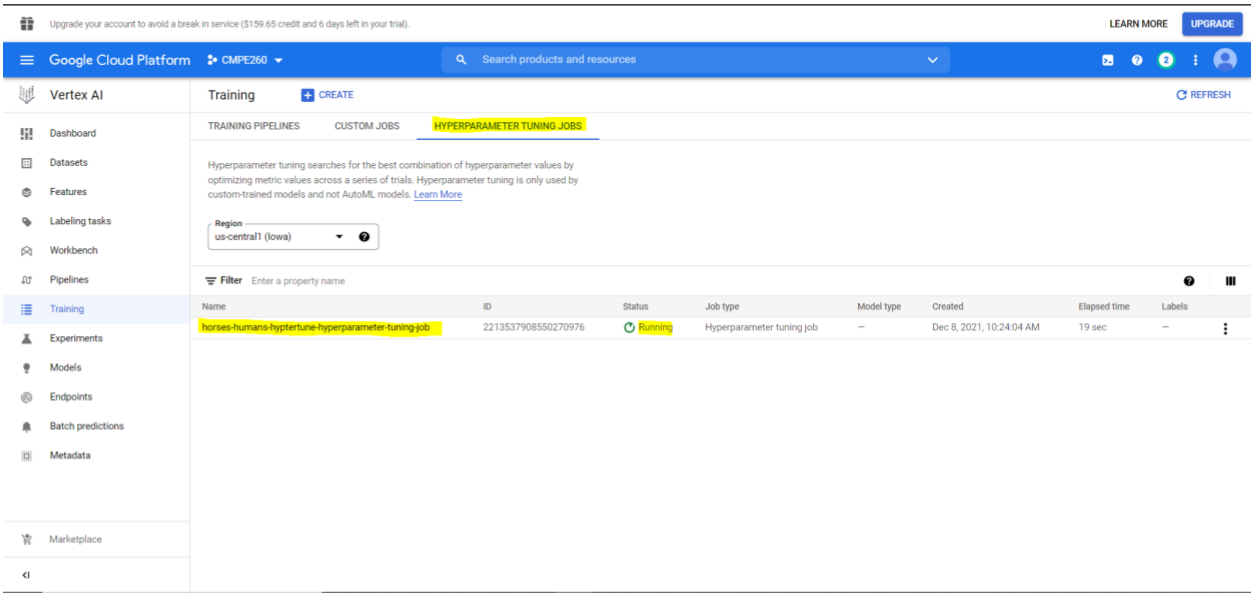
Worker count: 1

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).

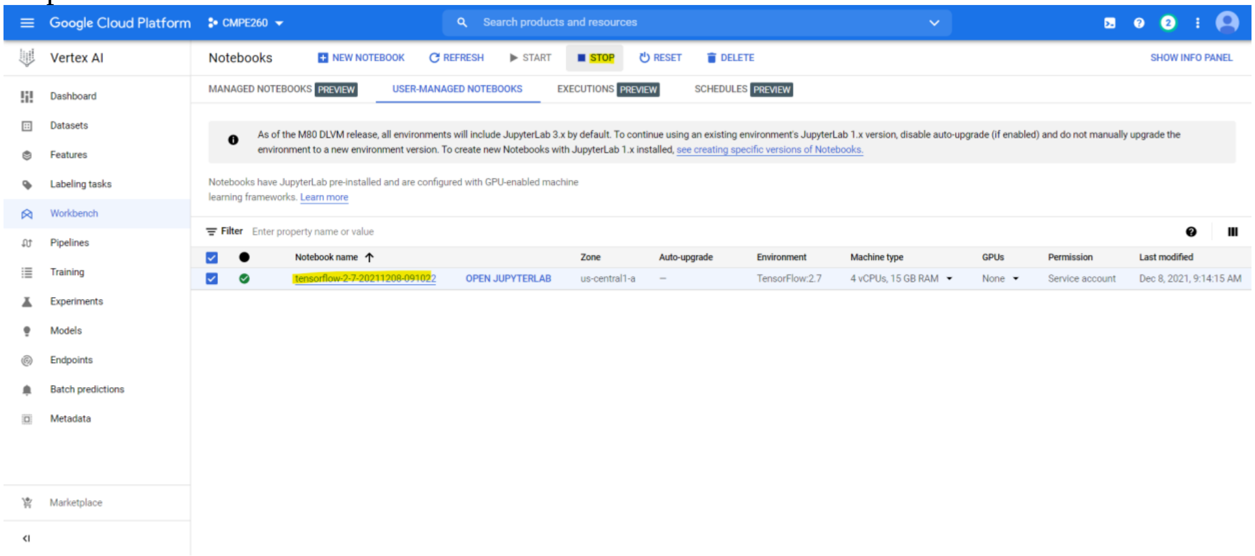
Hyperparameter tuning searches for the best combination of hyperparameter values by optimizing metric values across a series of trials. Hyperparameter tuning is only used by custom-trained models and not AutoML models. Learn More

Region: us-central1 (Iowa)

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

Cleanup

Stop the notebook instance

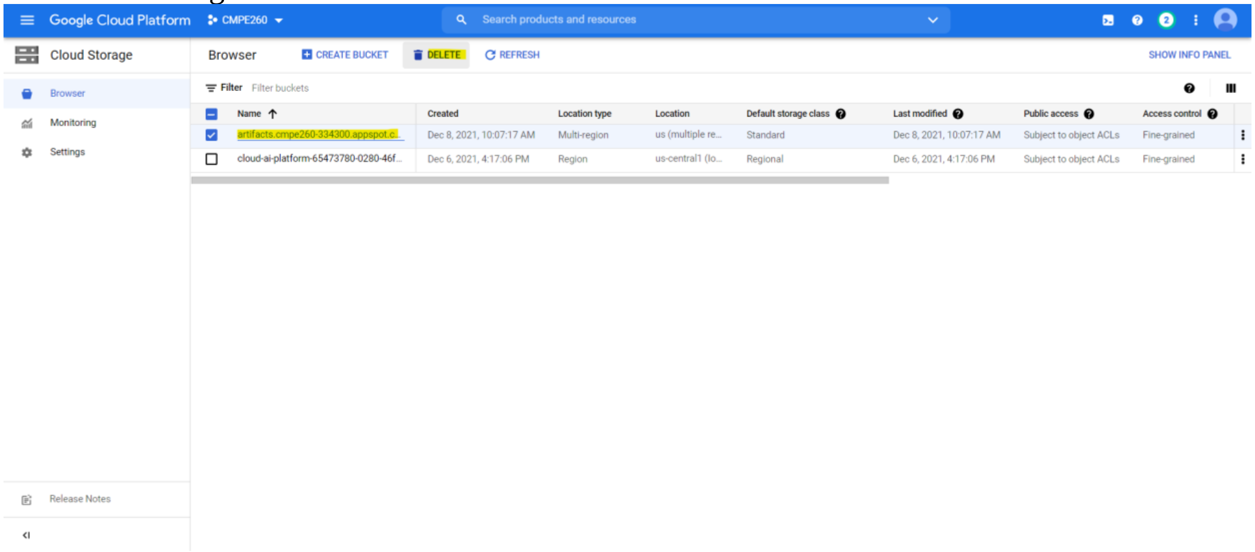


As of the M80 DLVM release, all environments will include JupyterLab 3.x by default. To continue using an existing environment's JupyterLab 1.x version, disable auto-upgrade (if enabled) and do not manually upgrade the environment to a new environment version. To create new Notebooks with JupyterLab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

☑	●	Notebook name ↑		Zone	Auto-upgrade	Environment	Machine type	GPUs	Permission	Last modified
☑	✓	tensorflow-2-7-20211208-091022	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



☐	Name ↑	Created	Location type	Location	Default storage class	Last modified	Public access	Access control
☑	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
☐	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