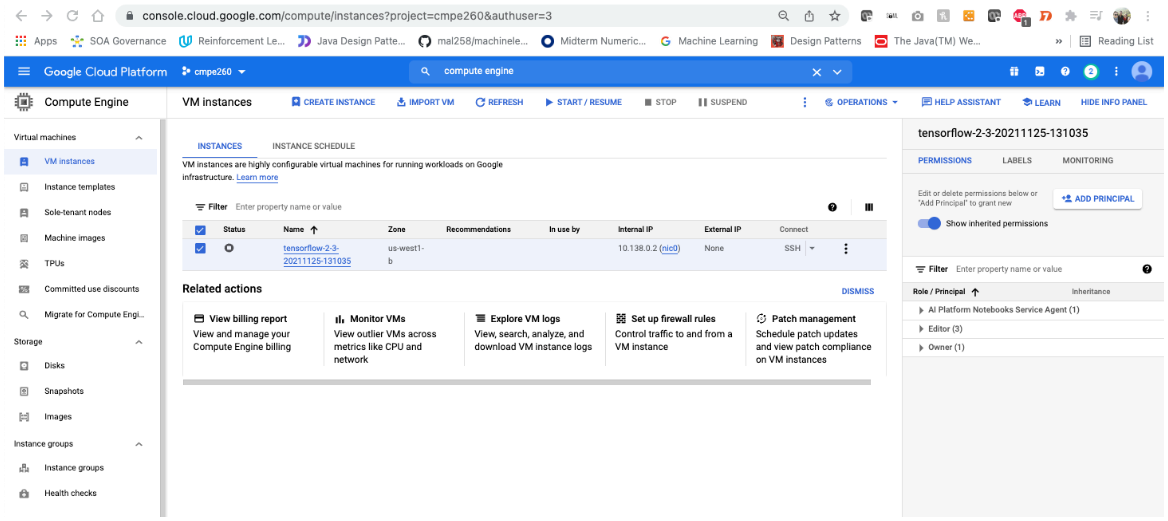
Assignment 7: optional catchup assignment 2 - VERTEX AI - for midterm and quiz - this will catch up midterm.

Vertex AI: Training and serving a custom model

Setting up the environment

1) Enable Compute Engine:



VM instances are highly configurable virtual machines for running workloads on Google infrastructure. Learn more

☑	Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
☑	○	tensorflow-2-3-20211125-131035	us-west1-b			10.138.0.2 (nic0)	None	SSH

Related actions

View billing report
View and manage your Compute Engine billing

Monitor VMs
View outlier VMs across metrics like CPU and network

Explore VM logs
View, search, analyze, and download VM instance logs

Set up firewall rules
Control traffic to and from a VM instance

Patch management
Schedule patch updates and view patch compliance on VM instances

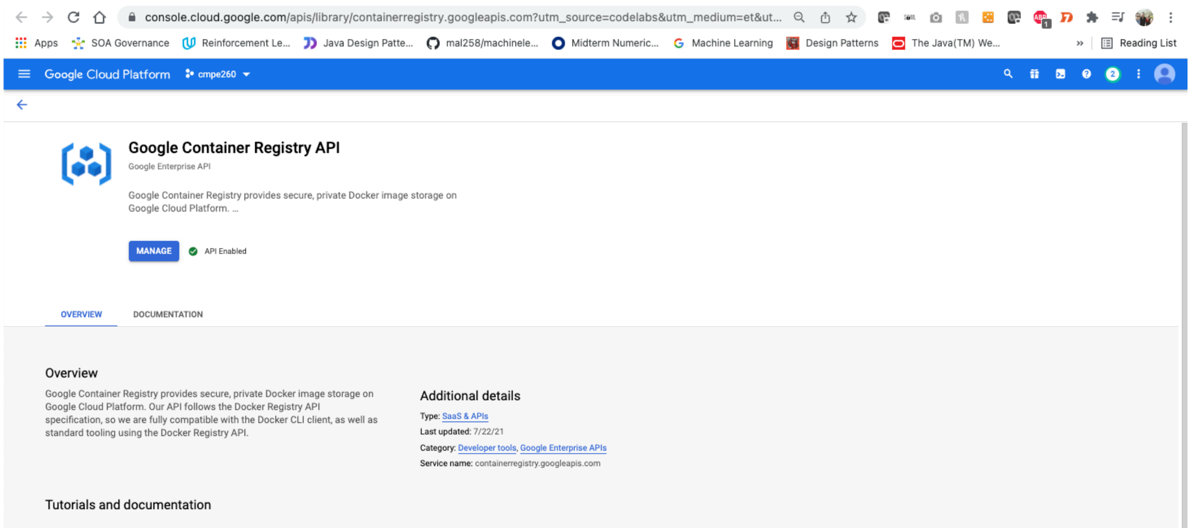
tensorflow-2-3-20211125-131035

Edit or delete permissions below or "Add Principal" to grant new

Show inherited permissions

Role / Principal	Inheritance
AI Platform Notebooks Service Agent (1)	
Editor (3)	
Owner (1)	

2) Enable Google Container Registry API



Google Container Registry API

Google Enterprise API

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

API Enabled

Overview

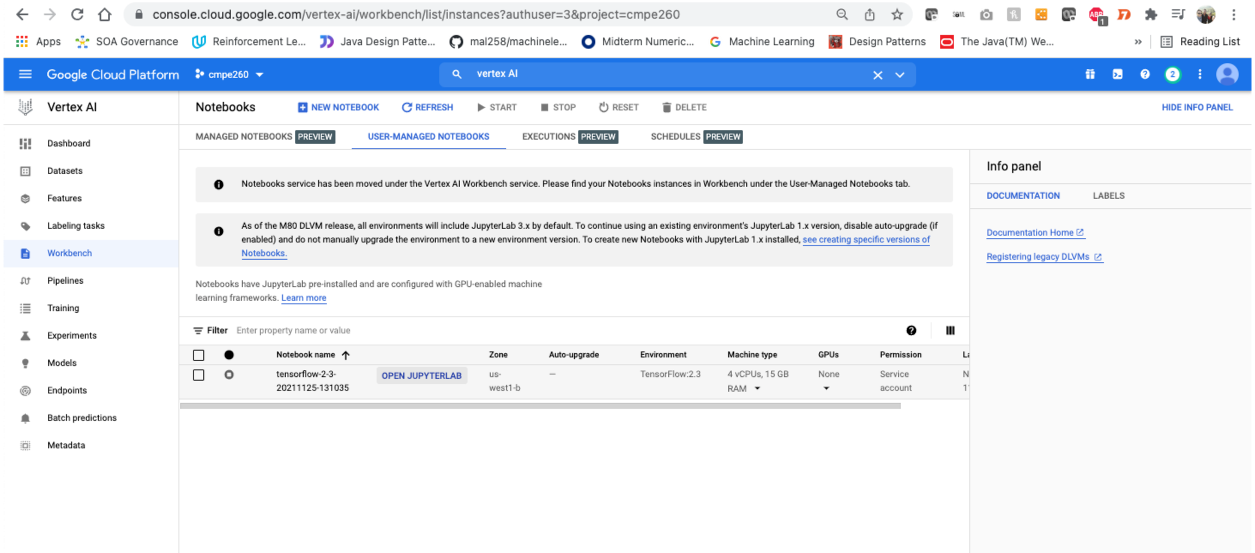
Google Container Registry provides secure, private Docker image storage on Google Cloud Platform. Our API follows the Docker Registry API specification, so we are fully compatible with the Docker CLI client, as well as standard tooling using the Docker Registry API.

Additional details

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

Tutorials and documentation

3) Enable workbench



Notebooks service has been moved under the Vertex AI Workbench service. Please find your Notebooks instances in Workbench under the User-Managed Notebooks tab.

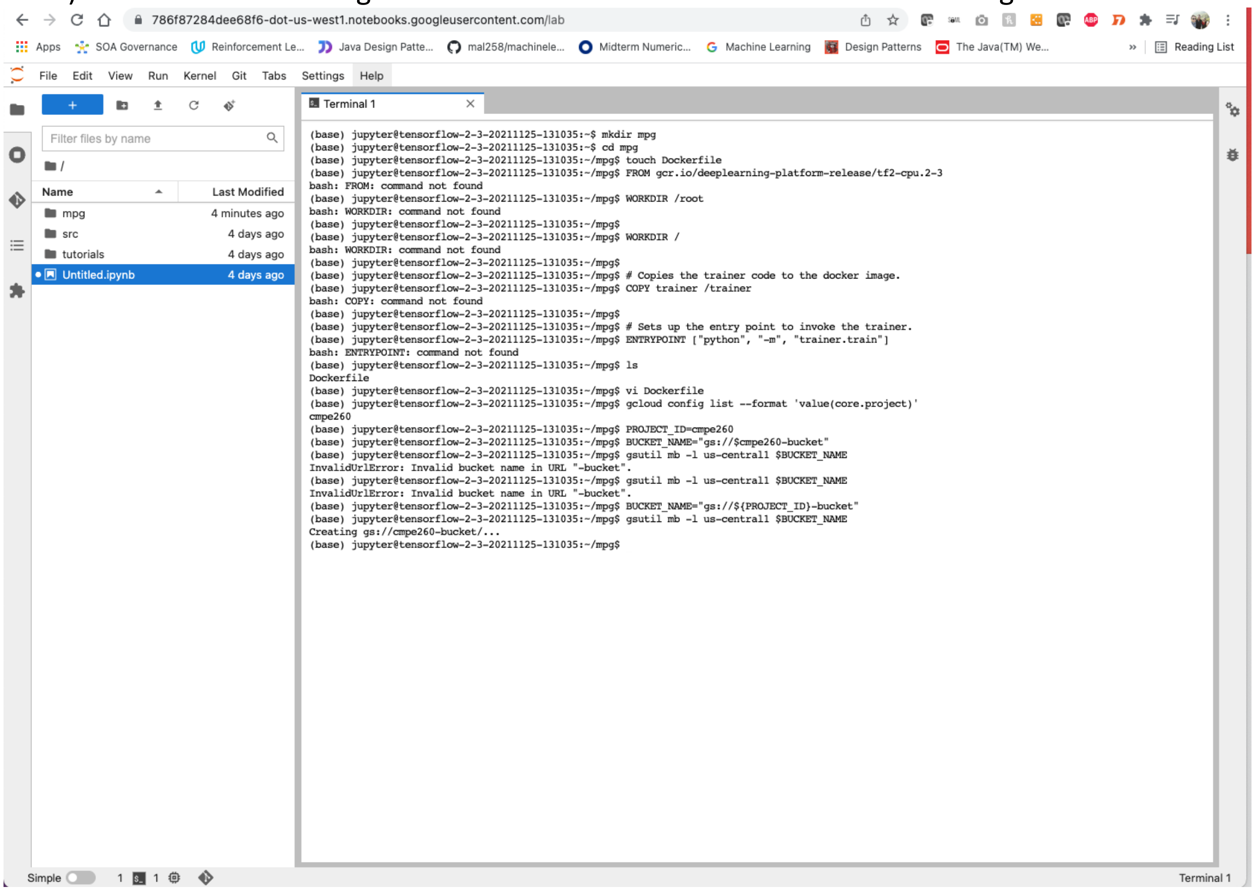
As of the M80 DLVM release, all environments will include JupyterLab 3.x by default. To continue using an existing environment's JupyterLab 1.x version, disable auto-upgrade (if enabled) and do not manually upgrade the environment to a new environment version. To create new Notebooks with JupyterLab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

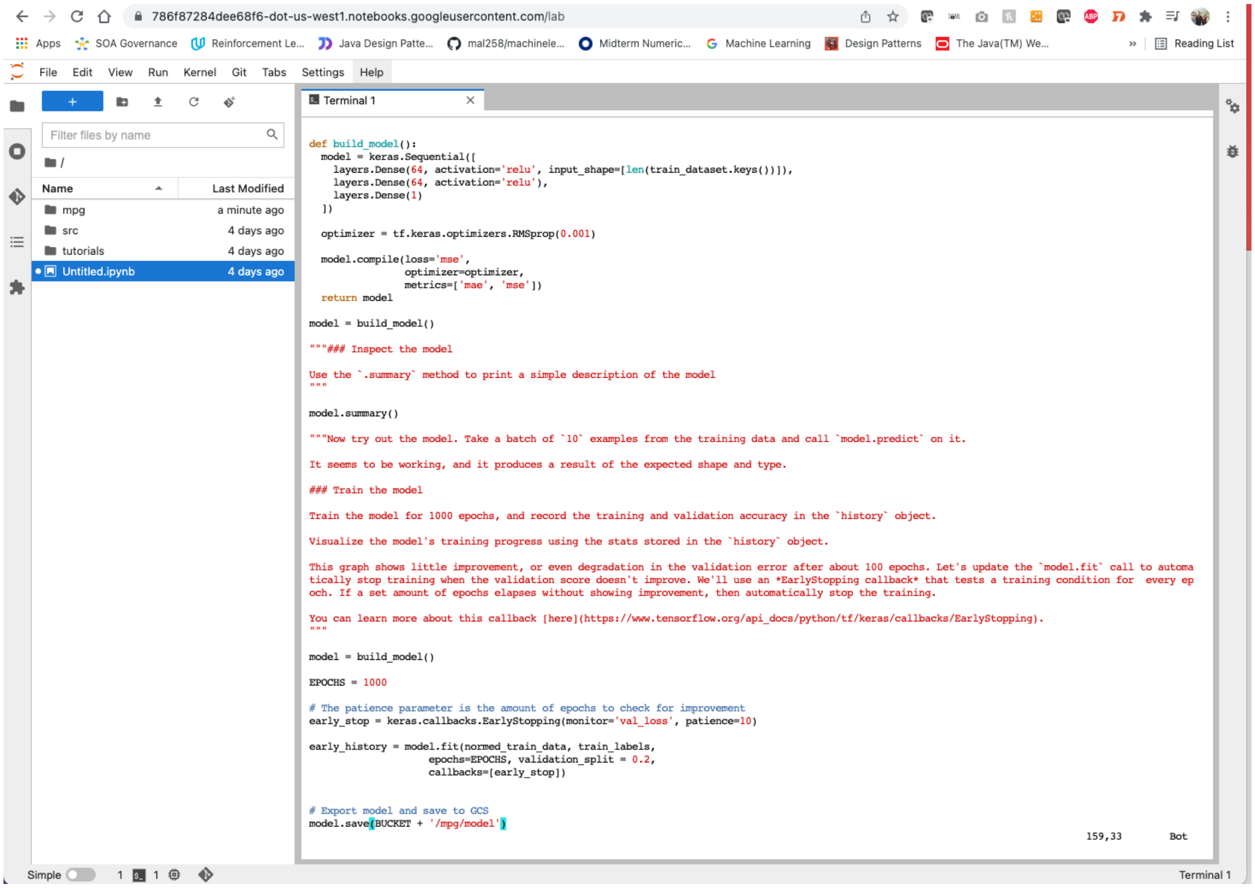
□	●	Notebook name ↑		Zone	Auto-upgrade	Environment	Machine type	GPUs	Permission
□	○	tensorflow-2-3-20211125-131035	OPEN JUPYTERLAB	us-west1-b	–	TensorFlow:2.3	4 vCPUs, 15 GB RAM	None	Service account

4) Containerize training code and create a Docker file and create a storage bucket



```
(base) jupyter@tensorflow-2-3-20211125-131035:~$ mkdir mpg
(base) jupyter@tensorflow-2-3-20211125-131035:~$ cd mpg
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ touch Dockerfile
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ FROM gcr.io/deeplearning-platform-release/tf2-cpu.2-3
bash: FROM: command not found
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ WORKDIR /root
bash: WORKDIR: command not found
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ WORKDIR /
bash: WORKDIR: command not found
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ # Copies the trainer code to the docker image.
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ COPY trainer /trainer
bash: COPY: command not found
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ # Sets up the entry point to invoke the trainer.
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ ENTRYPOINT ["python", "-m", "trainer.train"]
bash: ENTRYPOINT: command not found
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ ls
Dockerfile
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ vi Dockerfile
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ gcloud config list --format 'value(core.project)'
cmpe260
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ PROJECT_ID=cmpe260
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ BUCKET_NAME="gs://$cmpe260-bucket"
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ gsutil mb -l us-central1 $BUCKET_NAME
InvalidUrlError: Invalid bucket name in URL "-bucket".
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ gsutil mb -l us-central1 $BUCKET_NAME
InvalidUrlError: Invalid bucket name in URL "-bucket".
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ BUCKET_NAME="gs://${PROJECT_ID}-bucket"
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ gsutil mb -l us-central1 $BUCKET_NAME
Creating gs://cmpe260-bucket/...
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$
```

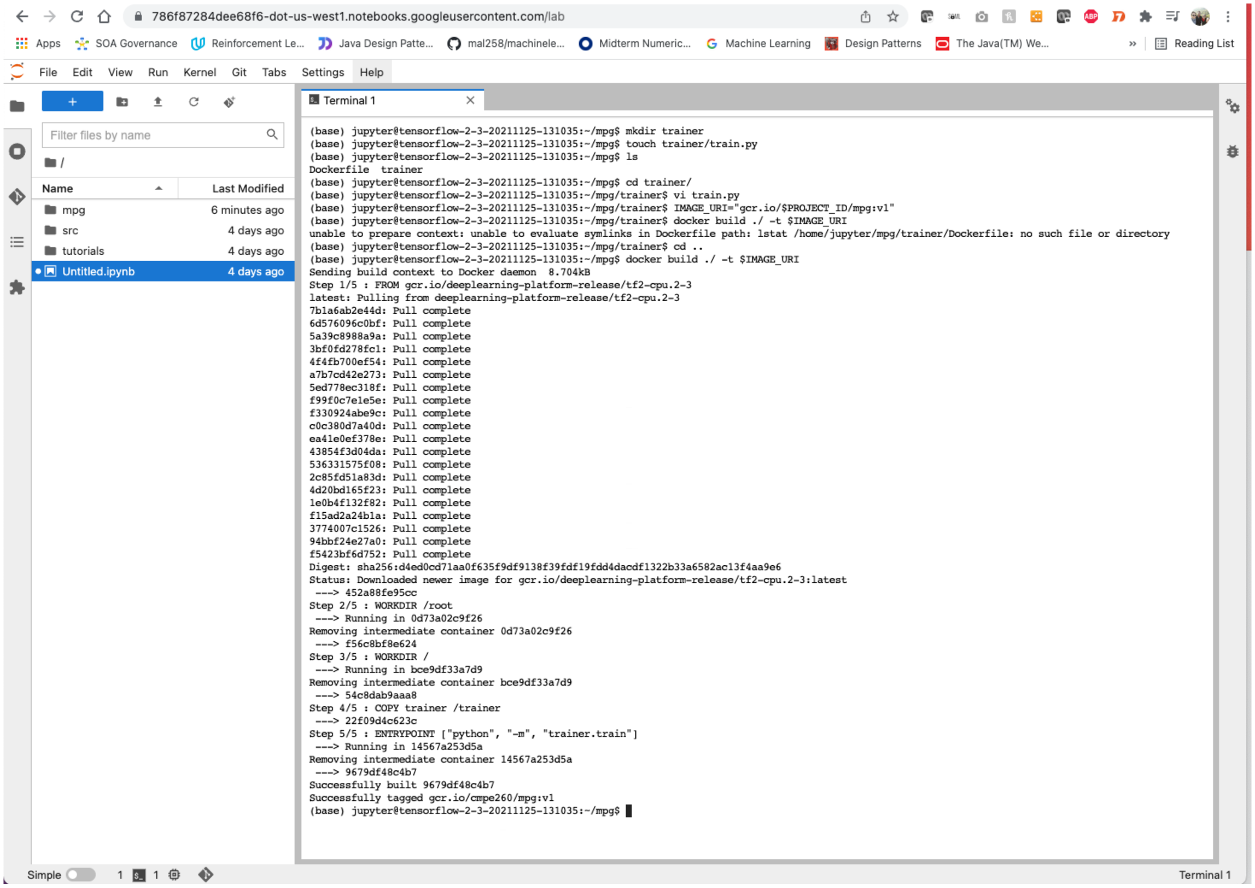
5) Create training code:



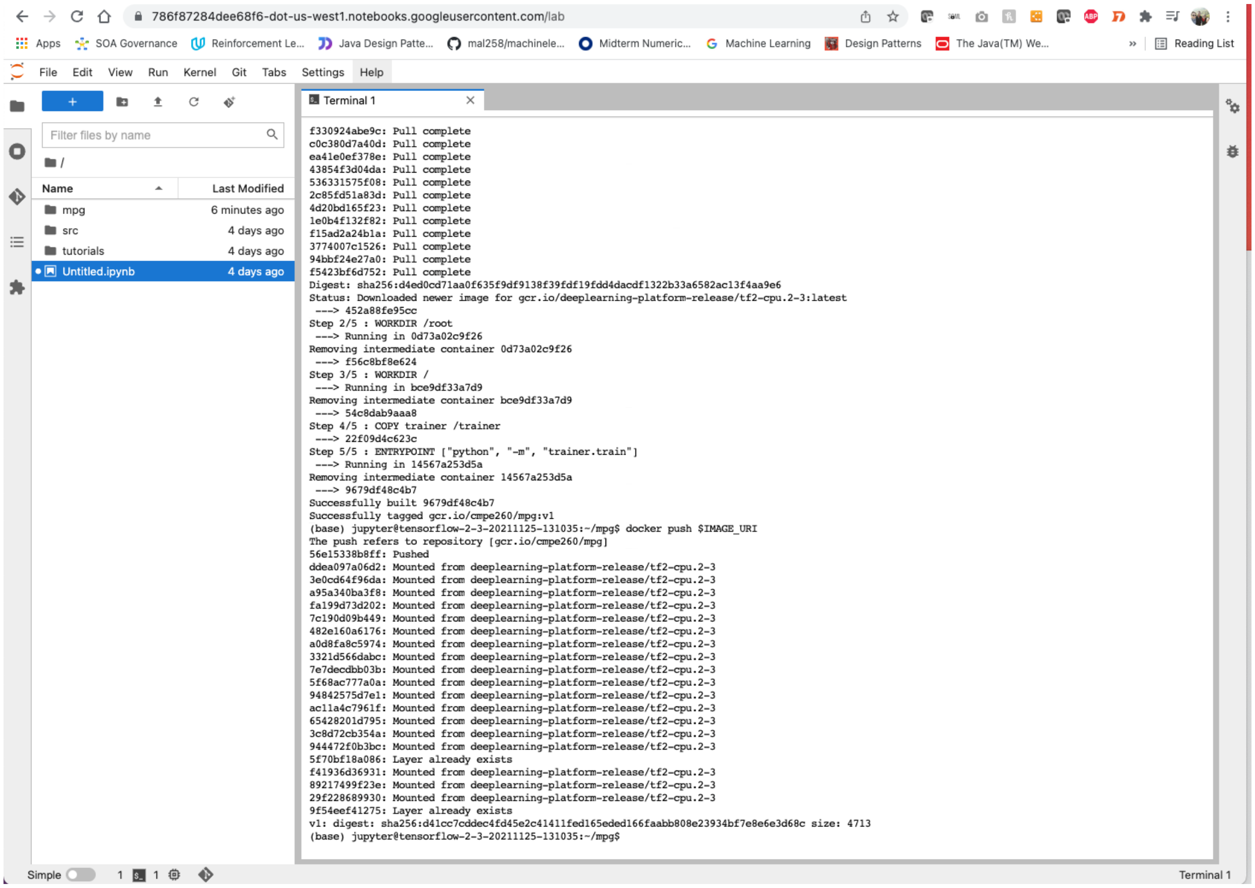
Name	Last Modified
mpg	a minute ago
src	4 days ago
tutorials	4 days ago
Untitled.ipynb	4 days ago

```
def build_model():
  model = keras.Sequential([
    layers.Dense(64, activation='relu', input_shape=[len(train_dataset.keys())]),
    layers.Dense(64, activation='relu'),
    layers.Dense(1)
  ])

  optimizer = tf.keras.optimizers.RMSprop(0.001)

  model.compile(loss='mse',
                optimizer=optimizer,
                metrics=['mae', 'mse'])
  return model

model = build_model()

"""### Inspect the model

Use the `.summary` method to print a simple description of the model
"""

model.summary()

"""Now try out the model. Take a batch of `10` examples from the training data and call `model.predict` on it.

It seems to be working, and it produces a result of the expected shape and type.

### Train the model

Train the model for 1000 epochs, and record the training and validation accuracy in the `history` object.

Visualize the model's training progress using the stats stored in the `history` object.

This graph shows little improvement, or even degradation in the validation error after about 100 epochs. Let's update the `model.fit` call to automa
tically stop training when the validation score doesn't improve. We'll use an *EarlyStopping callback* that tests a training condition for  every ep
och. If a set amount of epochs elapses without showing improvement, then automatically stop the training.

You can learn more about this callback [here](https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping).
"""

model = build_model()

EPOCHS = 1000

# The patience parameter is the amount of epochs to check for improvement
early_stop = keras.callbacks.EarlyStopping(monitor='val_loss', patience=10)

early_history = model.fit(normed_train_data, train_labels,
                    epochs=EPOCHS, validation_split = 0.2,
                    callbacks=[early_stop])


# Export model and save to GCS
model.save(BUCKET + '/mpg/model')
                                                                      159,33        Bot
```

6) Build and test the container locally

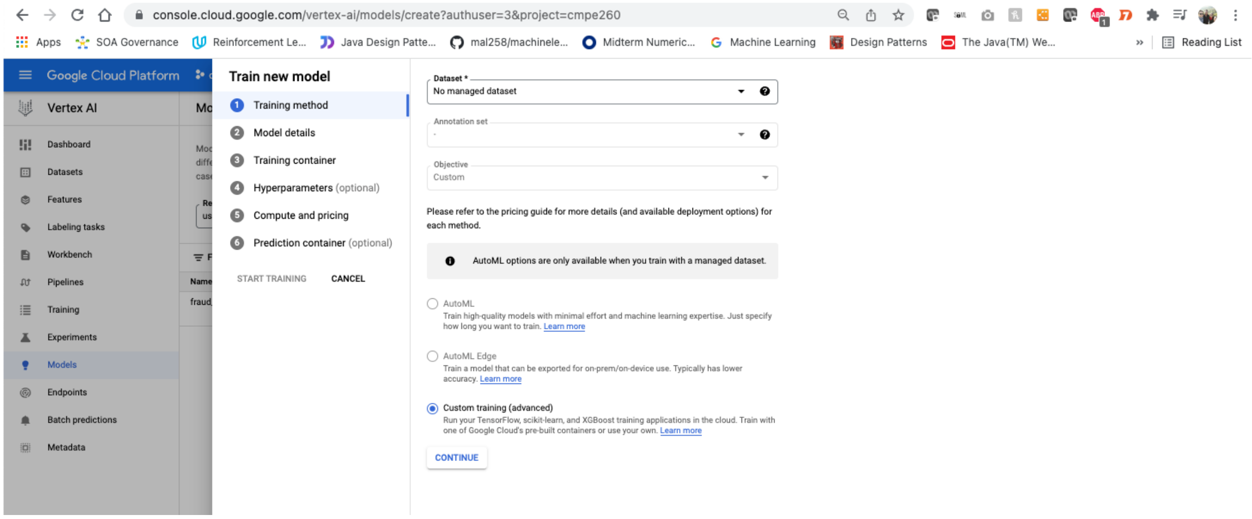


```
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ mkdir trainer
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ touch trainer/train.py
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ ls
Dockerfile  trainer
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ cd trainer/
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg/trainer$ vi train.py
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg/trainer$ IMAGE_URI="gcr.io/$PROJECT_ID/mpg:v1"
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg/trainer$ docker build ./ -t $IMAGE_URI
unable to prepare context: unable to evaluate symlinks in Dockerfile path: lstat /home/jupyter/mpg/trainer/Dockerfile: no such file or directory
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg/trainer$ cd ..
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ docker build ./ -t $IMAGE_URI
Sending build context to Docker daemon  8.704kB
Step 1/5 : FROM gcr.io/deeplearning-platform-release/tf2-cpu.2-3
latest: Pulling from deeplearning-platform-release/tf2-cpu.2-3
7b1a6ab2e44d: Pull complete
6d576096c0bf: Pull complete
5a39c8988a9a: Pull complete
3bf0fd278fc1: Pull complete
4f4fb700ef54: Pull complete
a7b7cd42e273: Pull complete
5ed778ec318f: Pull complete
f99f0c7e1e5e: Pull complete
f330924abe9c: Pull complete
c0c380d7a40d: Pull complete
ea41e0ef378e: Pull complete
43854f3d04da: Pull complete
536331575f08: Pull complete
2c85fd51a83d: Pull complete
4d20bd165f23: Pull complete
1e0b4f132f82: Pull complete
f15ad2a24b1a: Pull complete
3774007c1526: Pull complete
94bbf24e27a0: Pull complete
f5423bf6d752: Pull complete
Digest: sha256:d4ed0cd71aa0f635f9df9138f39fdf19fdd4dacdf1322b33a6582ac13f4aa9e6
Status: Downloaded newer image for gcr.io/deeplearning-platform-release/tf2-cpu.2-3:latest
 ---> 452a88fe95cc
Step 2/5 : WORKDIR /root
 ---> Running in 0d73a02c9f26
Removing intermediate container 0d73a02c9f26
 ---> f56c8bf8e624
Step 3/5 : WORKDIR /
 ---> Running in bce9df33a7d9
Removing intermediate container bce9df33a7d9
 ---> 54c8dab9aaa8
Step 4/5 : COPY trainer /trainer
 ---> 22f09d4c623c
Step 5/5 : ENTRYPOINT ["python", "-m", "trainer.train"]
 ---> Running in 14567a253d5a
Removing intermediate container 14567a253d5a
 ---> 9679df48c4b7
Successfully built 9679df48c4b7
Successfully tagged gcr.io/cmpe260/mpg:v1
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ 
```



```
f330924abe9c: Pull complete
c0c380d7a40d: Pull complete
ea41e0ef378e: Pull complete
43854f3d04da: Pull complete
536331575f08: Pull complete
2c85fd51a83d: Pull complete
4d20bd165f23: Pull complete
1e0b4f132f82: Pull complete
f15ad2a24b1a: Pull complete
3774007c1526: Pull complete
94bbf24e27a0: Pull complete
f5423bf6d752: Pull complete
Digest: sha256:d4ed0cd71aa0f635f9df9138f39fdf19fdd4dacdf1322b33a6582ac13f4aa9e6
Status: Downloaded newer image for gcr.io/deeplearning-platform-release/tf2-cpu.2-3:latest
 ---> 452a88fe95cc
Step 2/5 : WORKDIR /root
 ---> Running in 0d73a02c9f26
Removing intermediate container 0d73a02c9f26
 ---> f56c8bf8e624
Step 3/5 : WORKDIR /
 ---> Running in bce9df33a7d9
Removing intermediate container bce9df33a7d9
 ---> 54c8dab9aaa8
Step 4/5 : COPY trainer /trainer
 ---> 22f09d4c623c
Step 5/5 : ENTRYPOINT ["python", "-m", "trainer.train"]
 ---> Running in 14567a253d5a
Removing intermediate container 14567a253d5a
 ---> 9679df48c4b7
Successfully built 9679df48c4b7
Successfully tagged gcr.io/cmpe260/mpg:v1
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$ docker push $IMAGE_URI
The push refers to repository [gcr.io/cmpe260/mpg]
56e15338b8ff: Pushed
ddea097a06d2: Mounted from deeplearning-platform-release/tf2-cpu.2-3
3e0cd64f96da: Mounted from deeplearning-platform-release/tf2-cpu.2-3
a95a340ba3f8: Mounted from deeplearning-platform-release/tf2-cpu.2-3
fa199d73d202: Mounted from deeplearning-platform-release/tf2-cpu.2-3
7c190d09b449: Mounted from deeplearning-platform-release/tf2-cpu.2-3
482e160a6176: Mounted from deeplearning-platform-release/tf2-cpu.2-3
a0d8fa8c5974: Mounted from deeplearning-platform-release/tf2-cpu.2-3
3321d566dabc: Mounted from deeplearning-platform-release/tf2-cpu.2-3
7e7decdbb03b: Mounted from deeplearning-platform-release/tf2-cpu.2-3
5f68ac777a0a: Mounted from deeplearning-platform-release/tf2-cpu.2-3
94842575d7e1: Mounted from deeplearning-platform-release/tf2-cpu.2-3
ac11a4c7961f: Mounted from deeplearning-platform-release/tf2-cpu.2-3
65428201d795: Mounted from deeplearning-platform-release/tf2-cpu.2-3
3c8d72cb354a: Mounted from deeplearning-platform-release/tf2-cpu.2-3
944472f0b3bc: Mounted from deeplearning-platform-release/tf2-cpu.2-3
5f70bf18a086: Layer already exists
f41936d36931: Mounted from deeplearning-platform-release/tf2-cpu.2-3
89217499f23e: Mounted from deeplearning-platform-release/tf2-cpu.2-3
29f228689930: Mounted from deeplearning-platform-release/tf2-cpu.2-3
9f54eef41275: Layer already exists
v1: digest: sha256:d41cc7cddec4fd45e2c41411fed165eded166faabb808e23934bf7e8e6e3d68c size: 4713
(base) jupyter@tensorflow-2-3-20211125-131035:~/mpg$
```

7) Run a training job in vertex AI



Train new model

1. Training method
2. Model details
3. Training container
4. Hyperparameters (optional)
5. Compute and pricing
6. Prediction container (optional)

START TRAINING CANCEL

Dataset *: No managed dataset

Annotation set: -

Objective: Custom

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

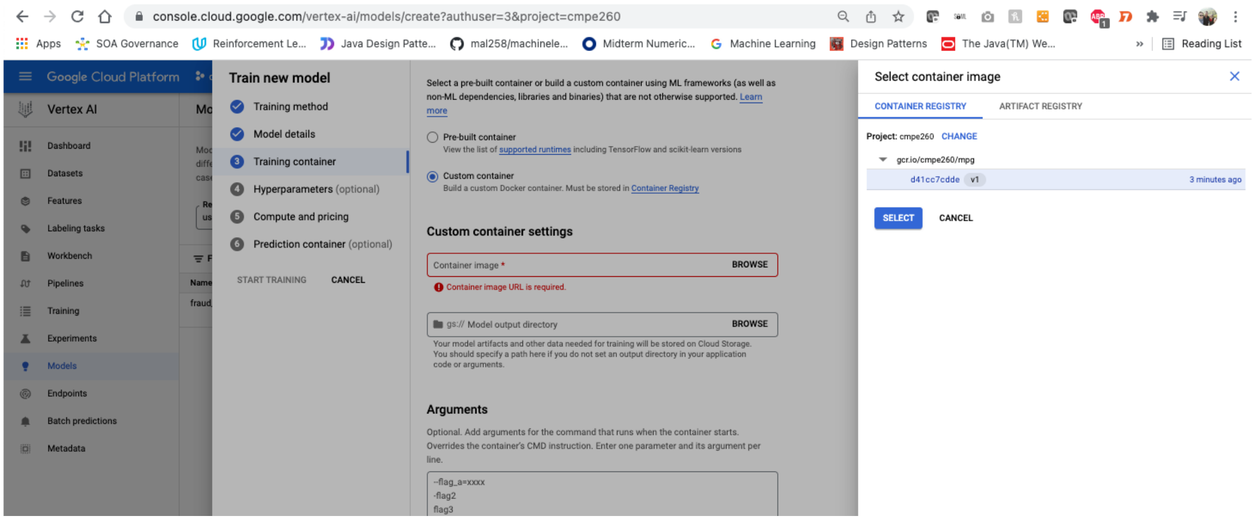
AutoML options are only available when you train with a managed dataset.

○ 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

CONTINUE



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 * BROWSE
Container image URL is required.

gs:// Model output directory BROWSE
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.

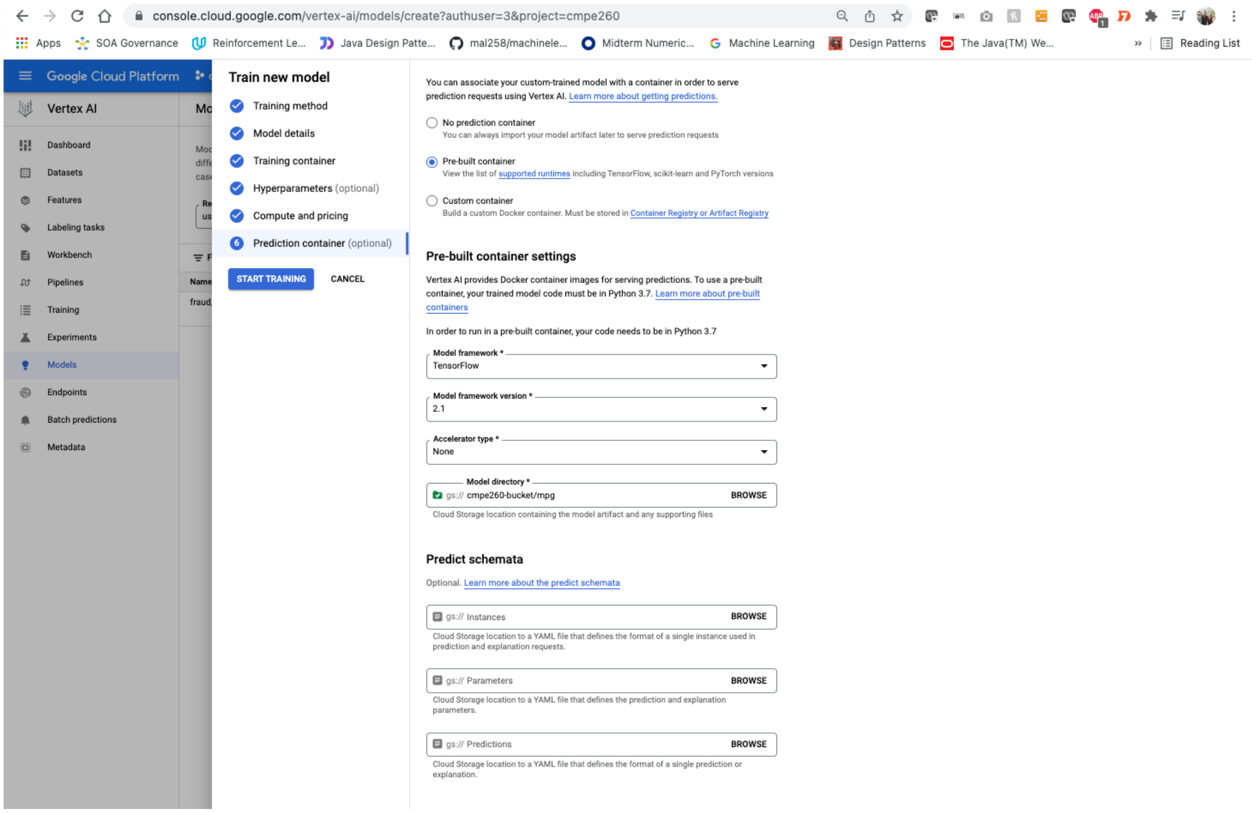
```
--flag_a=xxxx
-flag2
flag3
```

Select container image

Project: cmpe260 CHANGE

gcr.io/cmpe260/mpg

d41cc7cdde v1 3 minutes ago



You can associate your custom-trained model with a container in order to serve prediction requests using Vertex AI. Learn more about getting predictions.

○ No prediction container
You can always import your model artifact later to serve prediction requests

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

○ Custom container
Build a custom Docker container. Must be stored in Container Registry or Artifact Registry

Pre-built container settings

Vertex AI provides Docker container images for serving predictions. To use a pre-built container, your trained model code must be in Python 3.7. Learn more about pre-built containers

In order to run in a pre-built container, your code needs to be in Python 3.7

Model framework *: TensorFlow

Model framework version *: 2.1

Accelerator type *: None

Model directory *: gs:// cmpe260-bucket/mpg BROWSE
Cloud Storage location containing the model artifact and any supporting files

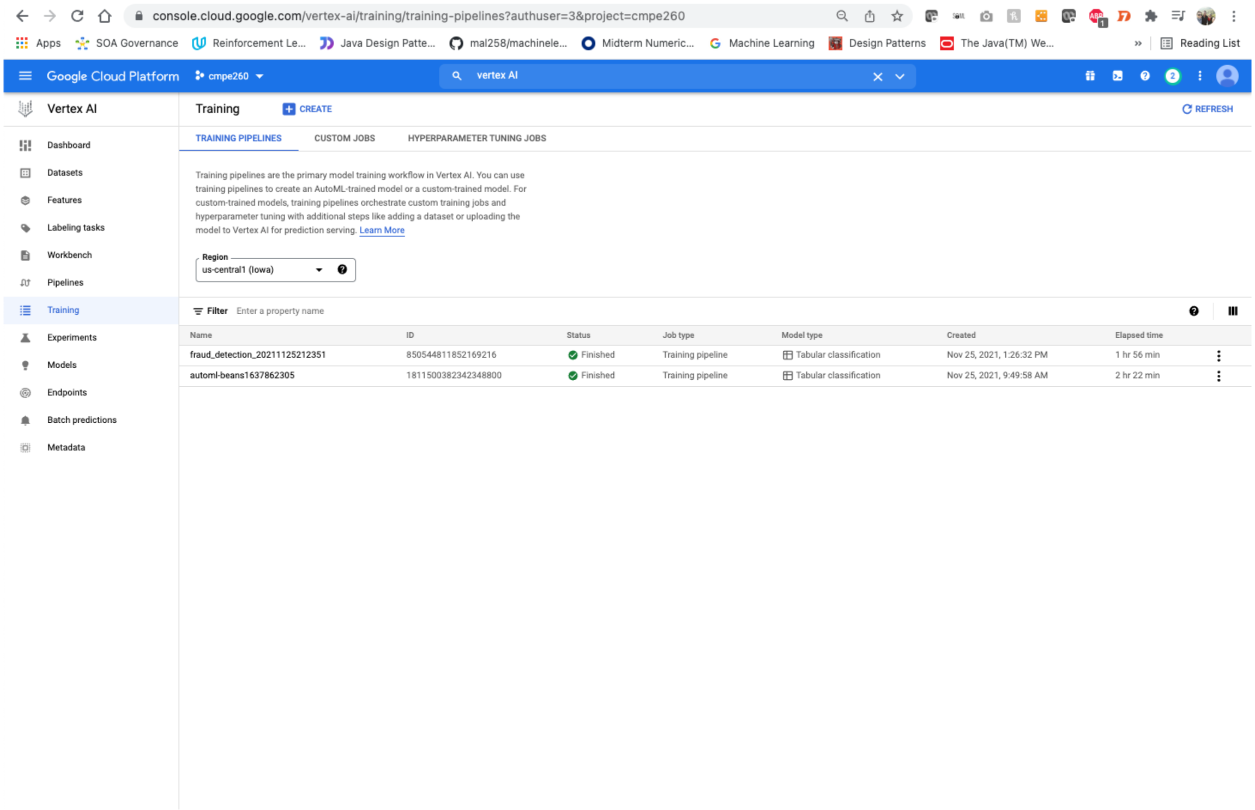
Predict schemata

Optional. Learn more about the predict schemata

gs:// Instances BROWSE
Cloud Storage location to a YAML file that defines the format of a single instance used in prediction and explanation requests.

gs:// Parameters BROWSE
Cloud Storage location to a YAML file that defines the prediction and explanation parameters.

gs:// Predictions BROWSE
Cloud Storage location to a YAML file that defines the format of a single prediction or explanation.

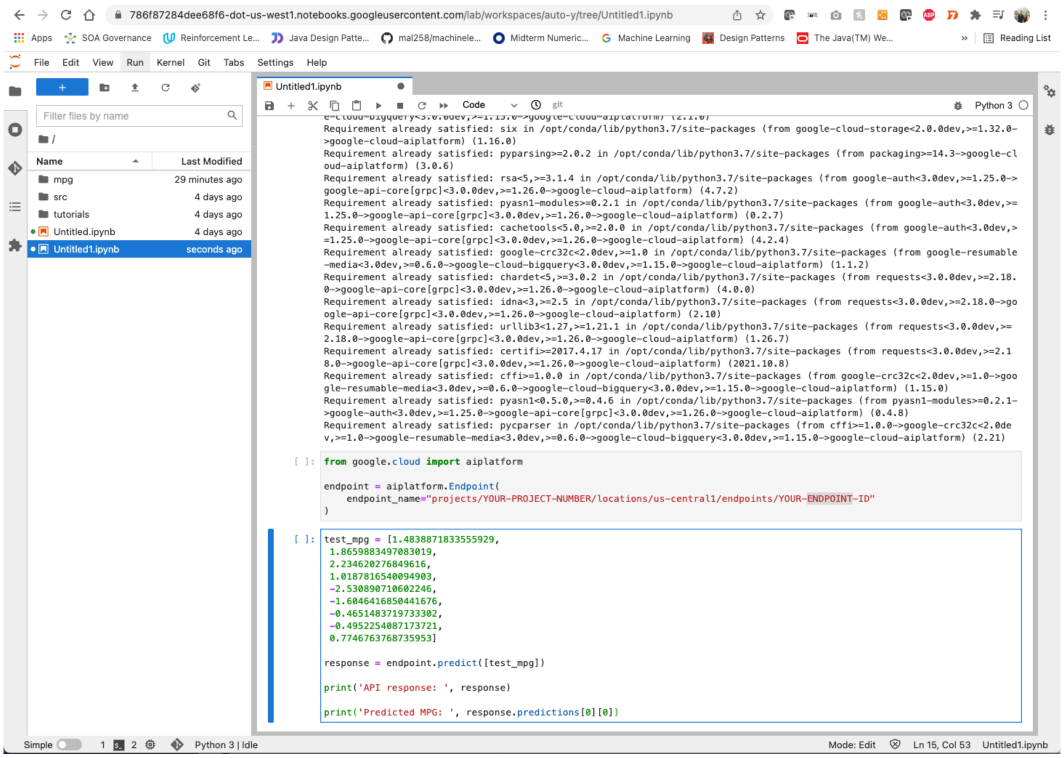


Training pipelines are the primary model training workflow in Vertex AI. You can use training pipelines to create an AutoML-trained model or a custom-trained model. For custom-trained models, training pipelines orchestrate custom training jobs and hyperparameter tuning with additional steps like adding 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
fraud_detection_20211125212351	850544811852169216	✓ Finished	Training pipeline	Tabular classification	Nov 25, 2021, 1:26:32 PM	1 hr 56 min
automl-beans1637862305	1811500382342348800	✓ Finished	Training pipeline	Tabular classification	Nov 25, 2021, 9:49:58 AM	2 hr 22 min

Deploy a model endpoint and get prediction on deployed model



Name	Last Modified
mpg	29 minutes ago
src	4 days ago
tutorials	4 days ago
Untitled.ipynb	4 days ago
Untitled1.ipynb	seconds ago

```
e-cloud-bigquery<3.0.0dev,>=1.15.0->google-cloud-aiplatform) (2.1.0)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages (from google-cloud-storage<2.0.0dev,>=1.32.0-
>google-cloud-aiplatform) (1.16.0)
Requirement already satisfied: pyparsing>=2.0.2 in /opt/conda/lib/python3.7/site-packages (from packaging>=14.3->google-cl
oud-aiplatform) (3.0.6)
Requirement already satisfied: rsa<5,>=3.1.4 in /opt/conda/lib/python3.7/site-packages (from google-auth<3.0dev,>=1.25.0->
google-api-core[grpc]<3.0.0dev,>=1.26.0->google-cloud-aiplatform) (4.7.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /opt/conda/lib/python3.7/site-packages (from google-auth<3.0dev,>=
1.25.0->google-api-core[grpc]<3.0.0dev,>=1.26.0->google-cloud-aiplatform) (0.2.7)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /opt/conda/lib/python3.7/site-packages (from google-auth<3.0dev,>
=1.25.0->google-api-core[grpc]<3.0.0dev,>=1.26.0->google-cloud-aiplatform) (4.2.4)
Requirement already satisfied: google-crc32c<2.0dev,>=1.0 in /opt/conda/lib/python3.7/site-packages (from google-resumable
-media<3.0dev,>=0.6.0->google-cloud-bigquery<3.0.0dev,>=1.15.0->google-cloud-aiplatform) (1.1.2)
Requirement already satisfied: chardet<5,>=3.0.2 in /opt/conda/lib/python3.7/site-packages (from requests<3.0.0dev,>=2.18.
0->google-api-core[grpc]<3.0.0dev,>=1.26.0->google-cloud-aiplatform) (4.0.0)
Requirement already satisfied: idna<3,>=2.5 in /opt/conda/lib/python3.7/site-packages (from requests<3.0.0dev,>=2.18.0->go
ogle-api-core[grpc]<3.0.0dev,>=1.26.0->google-cloud-aiplatform) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/conda/lib/python3.7/site-packages (from requests<3.0.0dev,>=
2.18.0->google-api-core[grpc]<3.0.0dev,>=1.26.0->google-cloud-aiplatform) (1.26.7)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.7/site-packages (from requests<3.0.0dev,>=2.1
8.0->google-api-core[grpc]<3.0.0dev,>=1.26.0->google-cloud-aiplatform) (2021.10.8)
Requirement already satisfied: cffi>=1.0.0 in /opt/conda/lib/python3.7/site-packages (from google-crc32c<2.0dev,>=1.0->goo
gle-resumable-media<3.0dev,>=0.6.0->google-cloud-bigquery<3.0.0dev,>=1.15.0->google-cloud-aiplatform) (1.15.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /opt/conda/lib/python3.7/site-packages (from pyasn1-modules>=0.2.1-
>google-auth<3.0dev,>=1.25.0->google-api-core[grpc]<3.0.0dev,>=1.26.0->google-cloud-aiplatform) (0.4.8)
Requirement already satisfied: pycparser in /opt/conda/lib/python3.7/site-packages (from cffi>=1.0.0->google-crc32c<2.0de
v,>=1.0->google-resumable-media<3.0dev,>=0.6.0->google-cloud-bigquery<3.0.0dev,>=1.15.0->google-cloud-aiplatform) (2.21)
```

```
[ ]: from google.cloud import aiplatform

     endpoint = aiplatform.Endpoint(
         endpoint_name="projects/YOUR-PROJECT-NUMBER/locations/us-central1/endpoints/YOUR-ENDPOINT-ID"
     )
```

```
[ ]: test_mpg = [1.4838871833555929,
      1.8659883497083019,
      2.234620276849616,
      1.0187816540094903,
      -2.530890710602246,
      -1.6046416850441676,
      -0.4651483719733302,
      -0.4952254087173721,
      0.7746763768735953]

     response = endpoint.predict([test_mpg])

     print('API response: ', response)

     print('Predicted MPG: ', response.predictions[0][0])
```