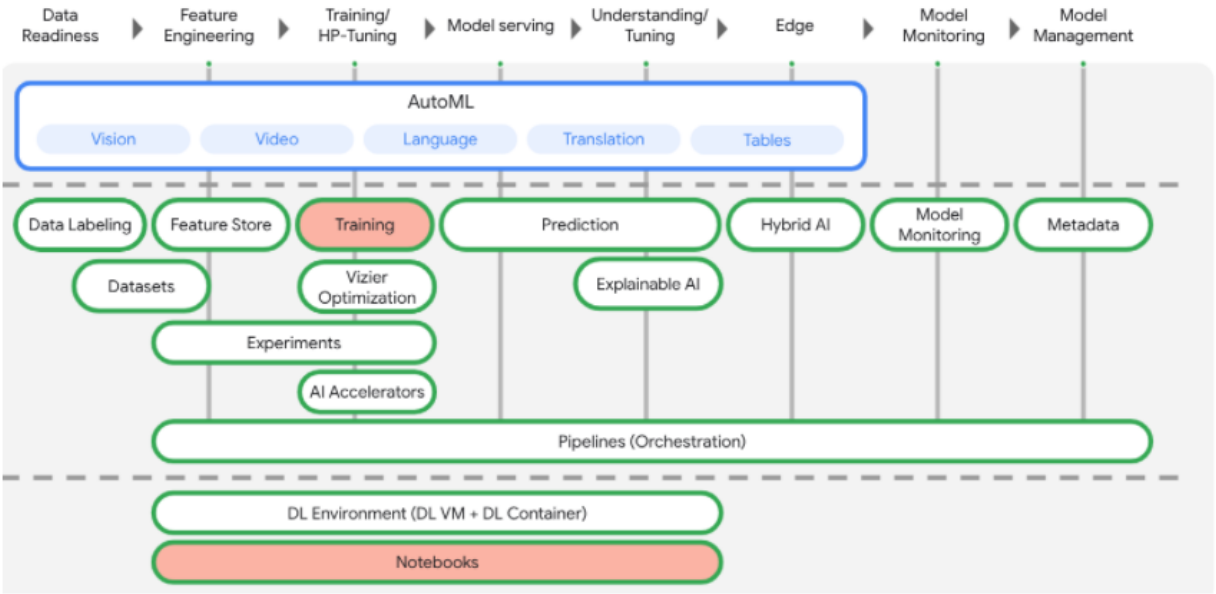
Multi-Worker Training and Transfer Learning with TensorFlow

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

Objectives:

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

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



Use Case Overview

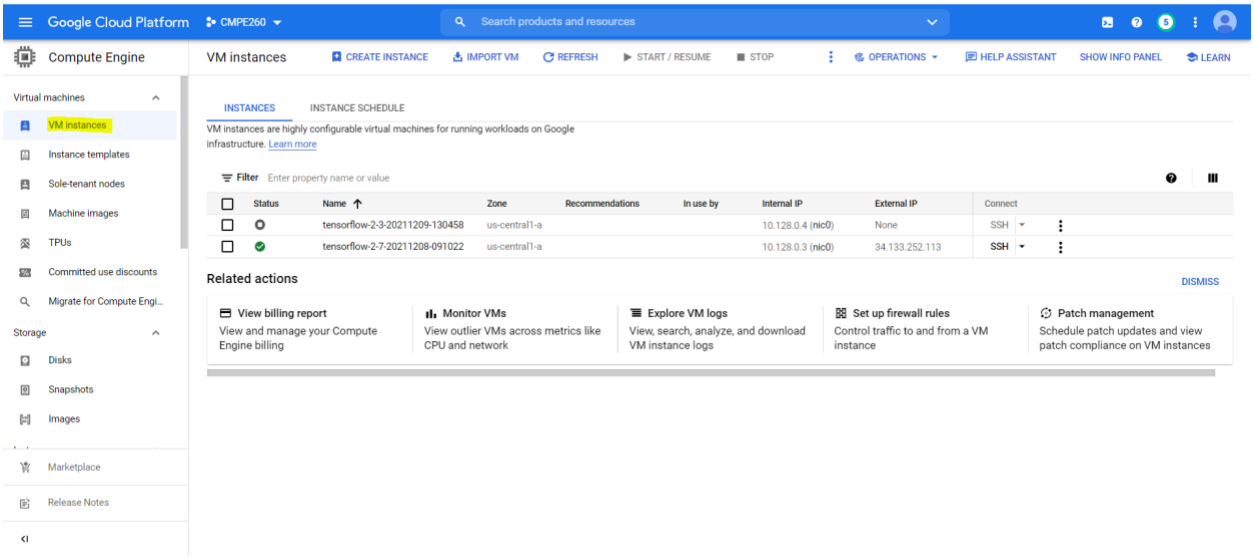
Use transfer learning to train an image classification model on the cassava dataset from TensorFlow Datasets. The architecture you'll use is a ResNet50 model from the tf.keras.applications library pretrained on the Imagenet dataset.

Why Distributed Training?

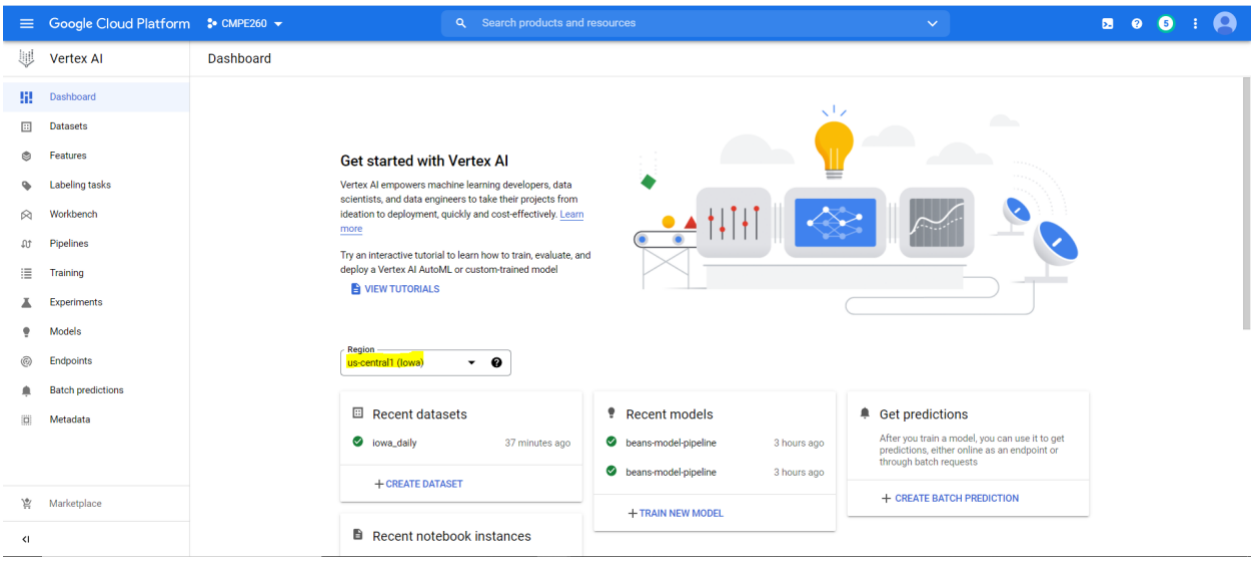
If you have a single GPU, TensorFlow will use this accelerator to speed up model training with no extra work on your part. However, if you want to get an additional boost from using multiple GPUs on a single machine or multiple machines (each with potentially multiple GPUs), then you'll need to use tf.distribute, which is TensorFlow's library for running a computation across multiple devices. A device refers to a CPU or accelerator, such as GPUs or TPUs, on some machine that TensorFlow can run operations on.

Set up your environment

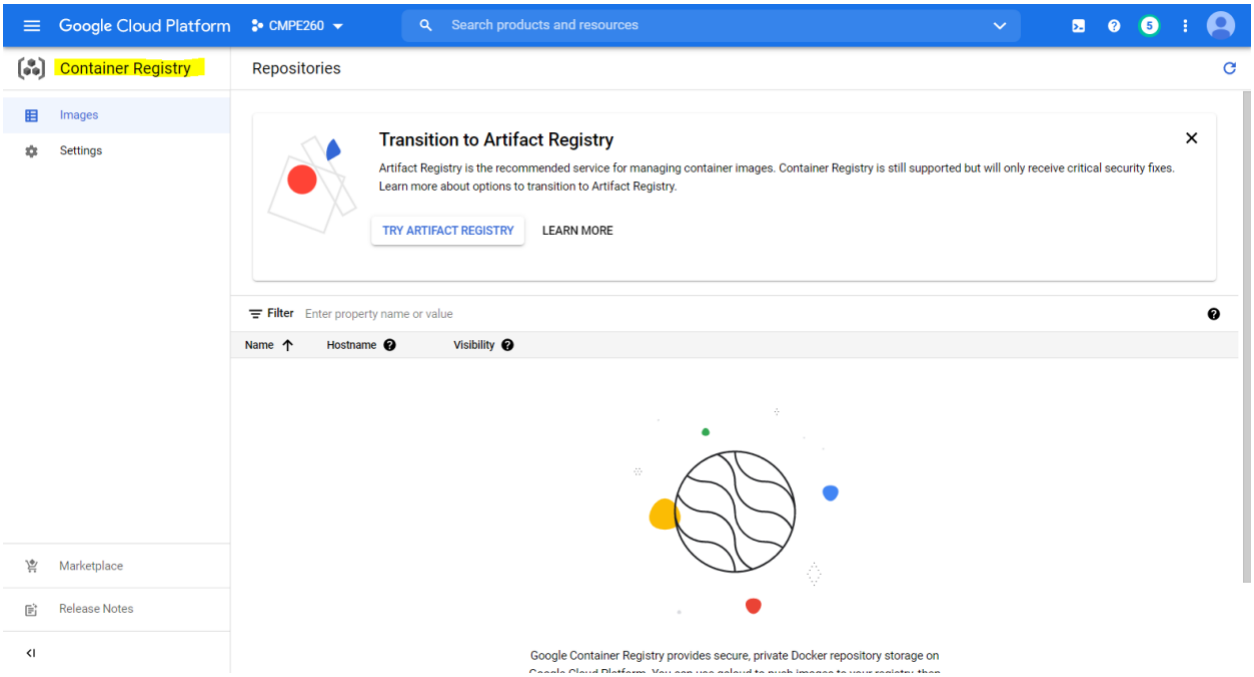
Enable the Compute Engine API



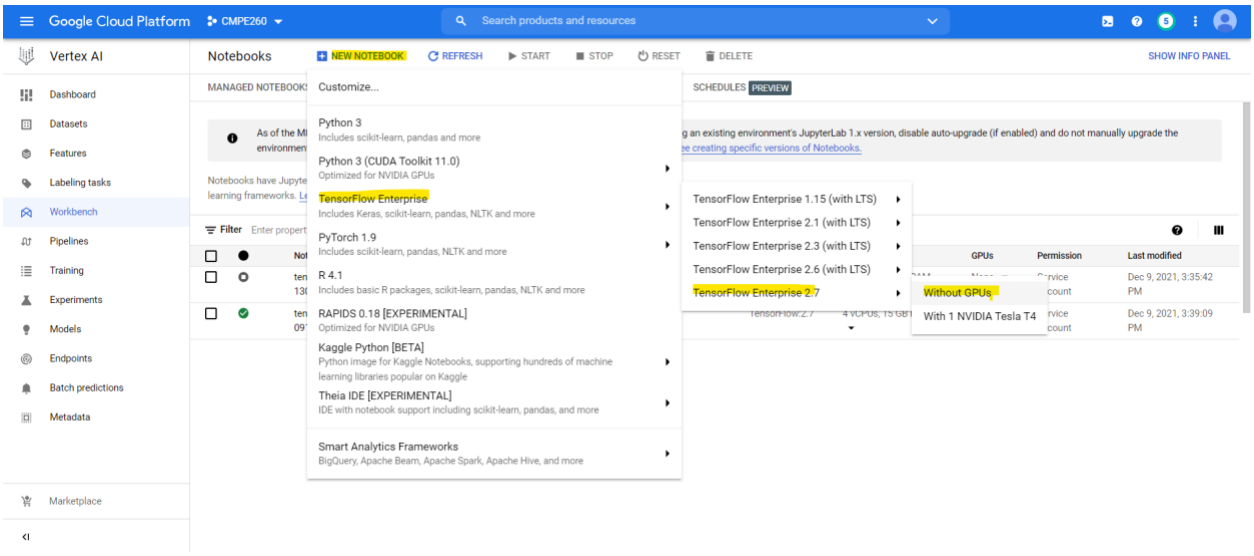
Enable the Vertex AI API



Enable the Container Registry API

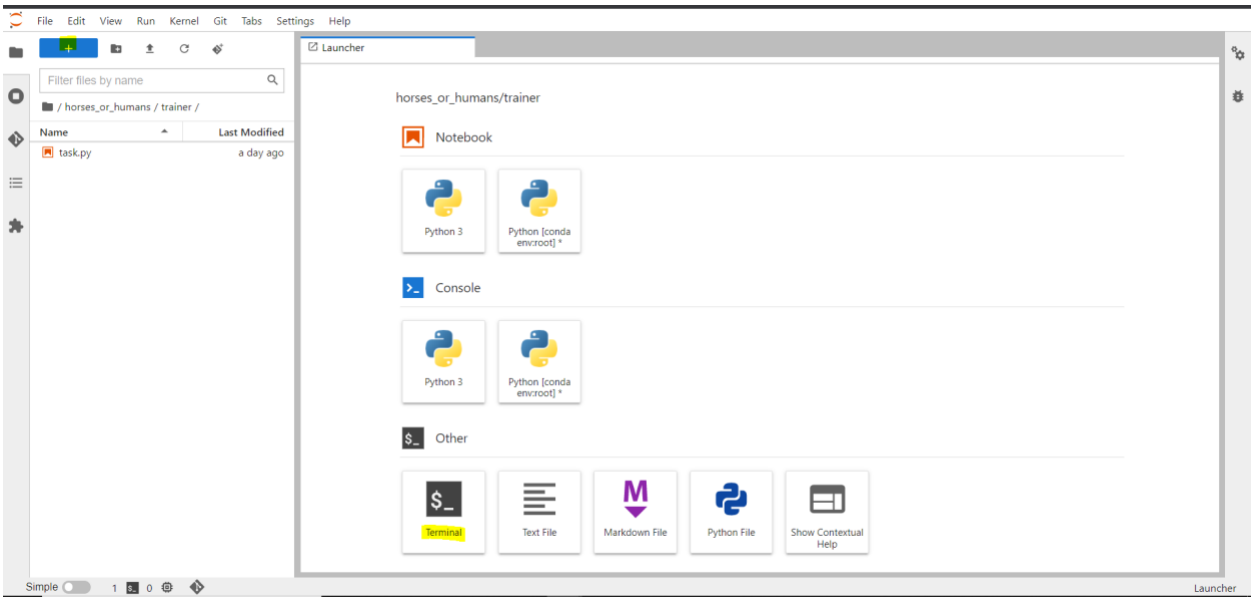


Create a Vertex AI Workbench instance

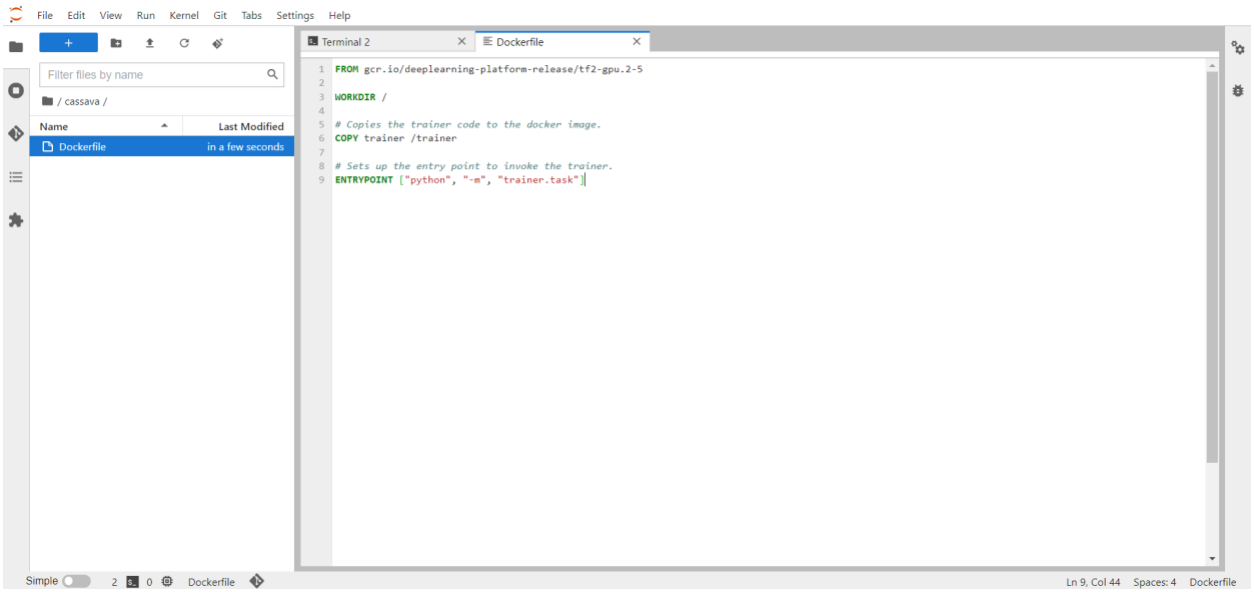


Containerize training application code

To start, from the Launcher menu, open a Terminal window in your notebook instance:



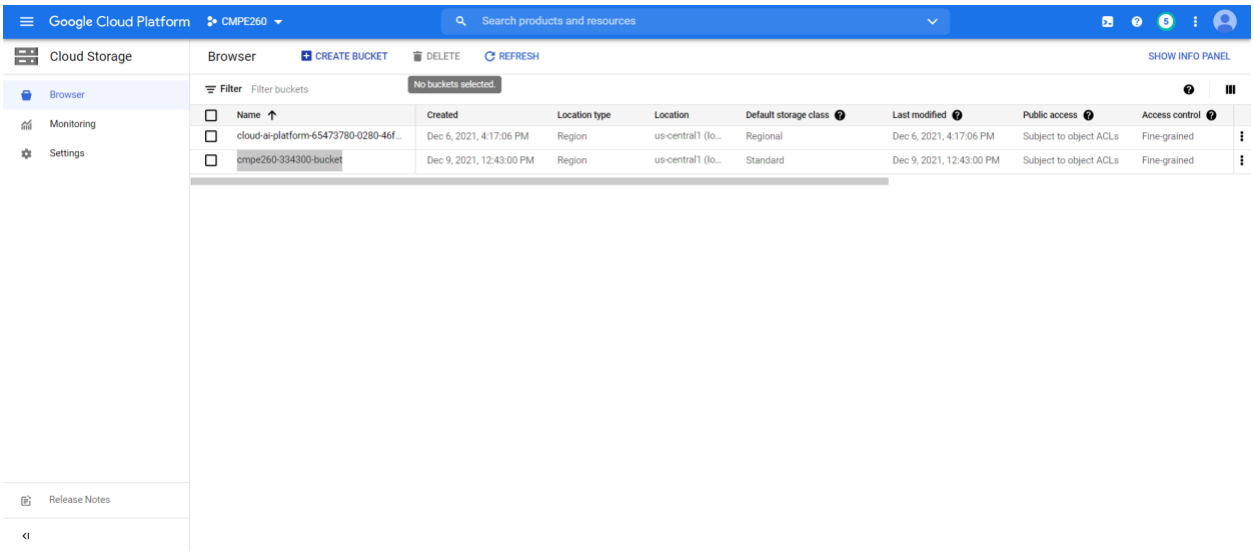
Step 1: Create a Dockerfile



Step 2: Create a Cloud Storage bucket

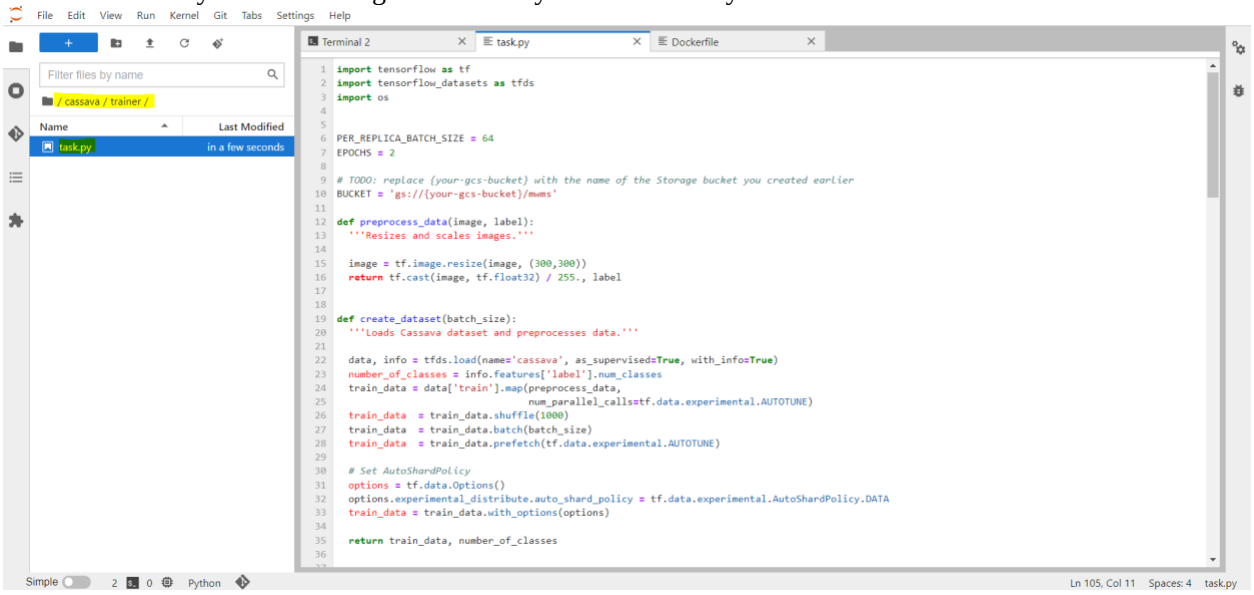
Run the following commands, to created the Storage Bucket

```
gcloud config list --format 'value(core.project)'
PROJECT_ID='your-cloud-project'
BUCKET="gs://${PROJECT_ID}-bucket"
gsutil mb -l us-central1 $BUCKET
```



Step 3: Add model training code

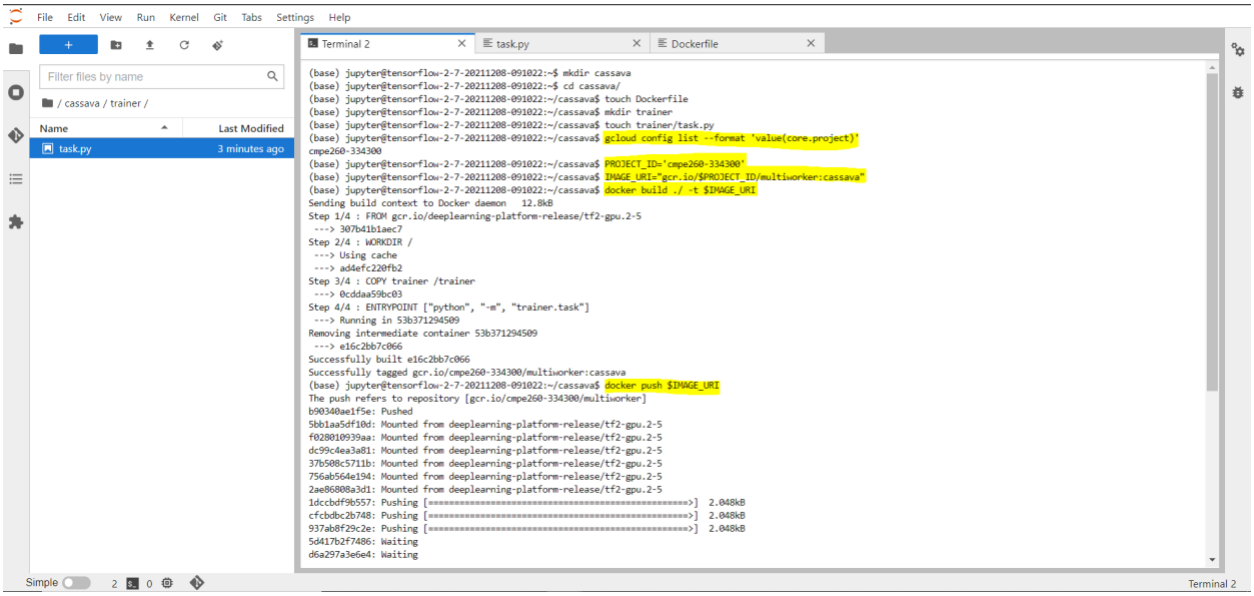
Create a directory for the training code and a Python file where you'll add the code:



Step 4: Build the container

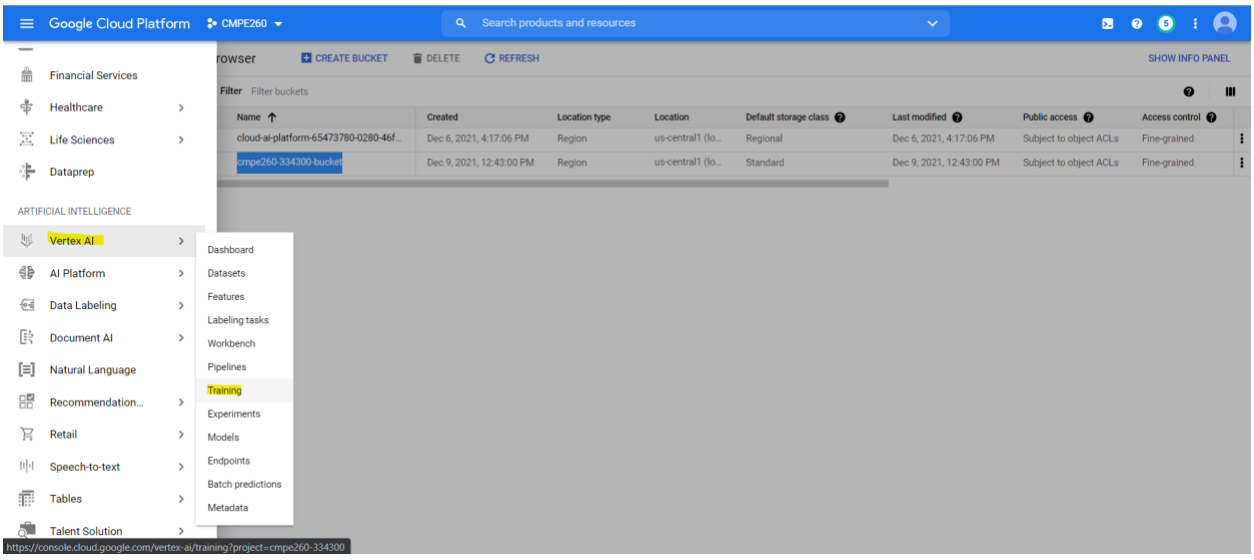
Run the following commands:

```
PROJECT_ID='your-cloud-project'
IMAGE_URI="gcr.io/$PROJECT_ID/multiworker:cassava"
docker build ./ -t $IMAGE_URI
docker push $IMAGE_URI
```



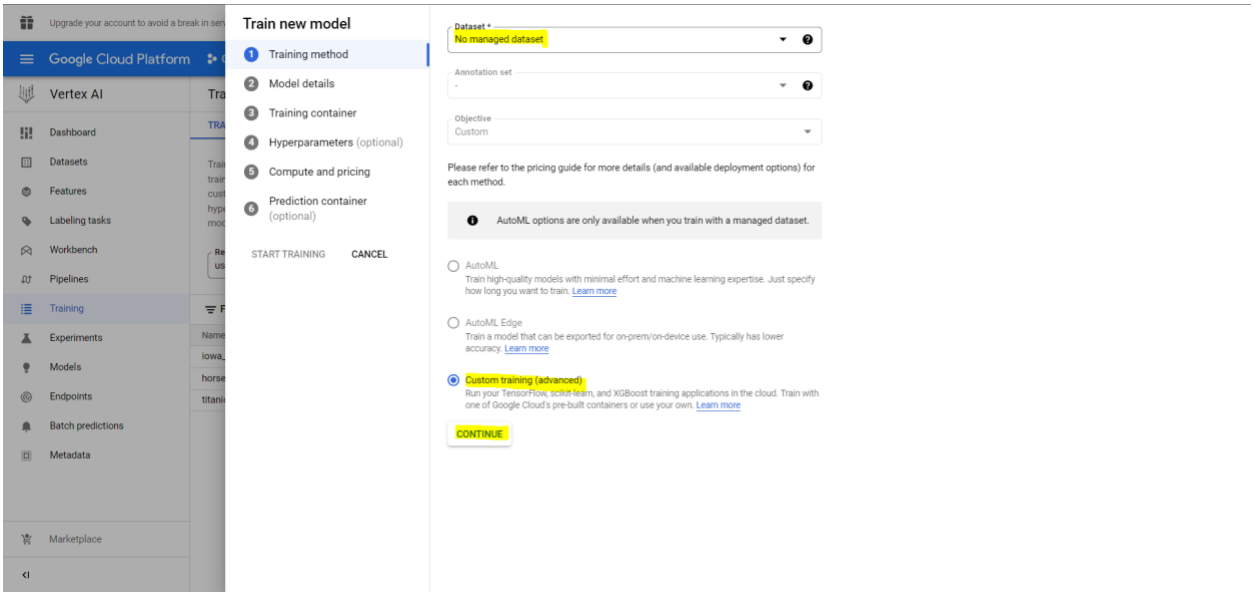
Run a multi-worker training job on Vertex AI

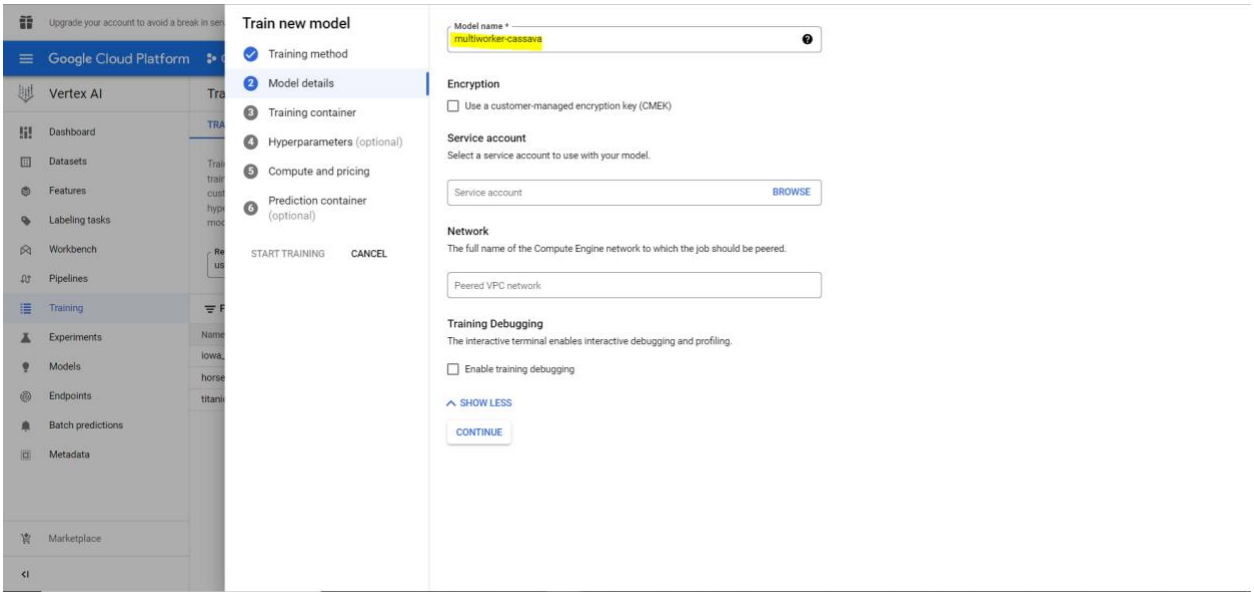
Navigate to Vertex AI → Training

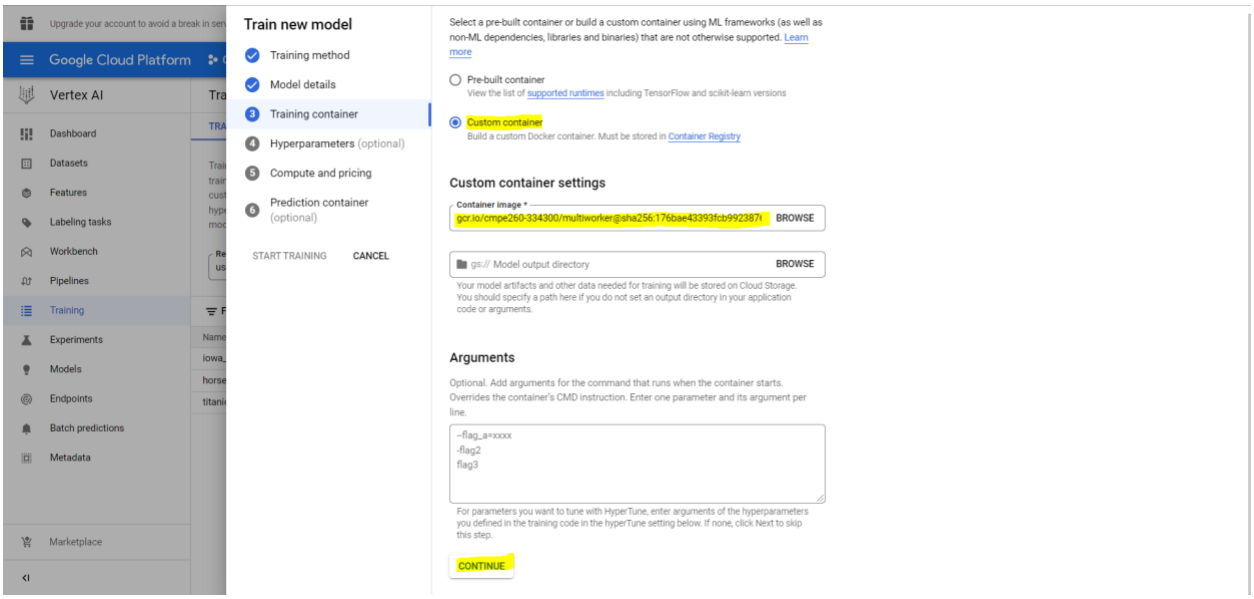


Step 1: Configure training job

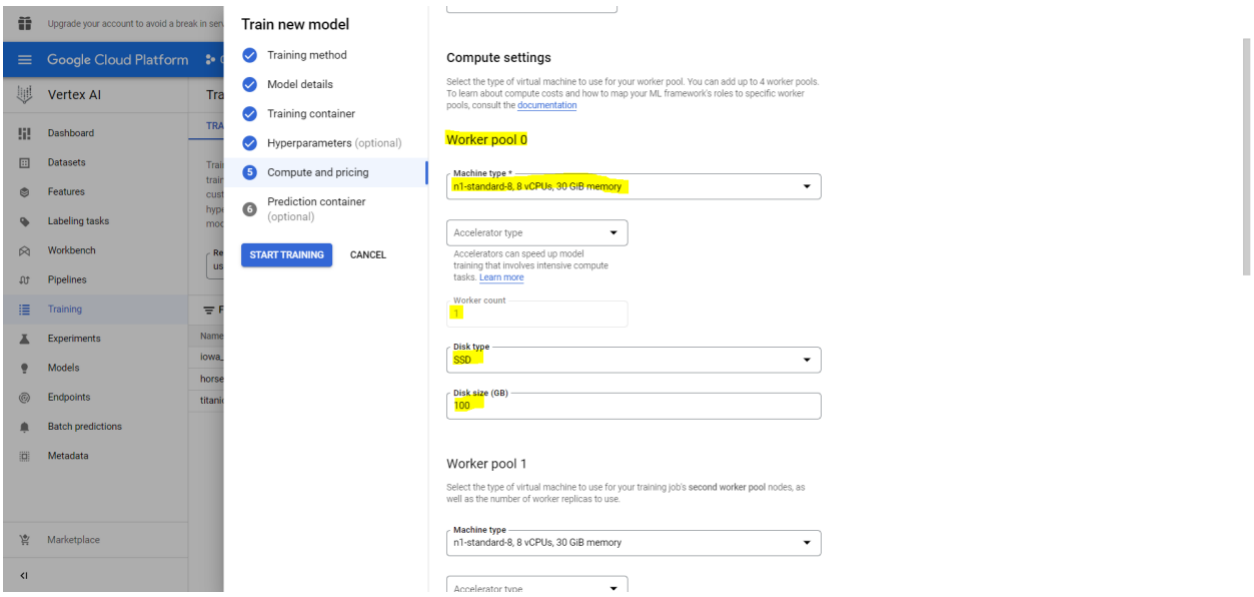
Click **Create** to enter the parameters for your training job.


Upgrade your account to avoid a break in ser
Google Cloud Platform
Vertex AI
Dashboard
Datasets
Features
Labeling tasks
Workbench
Pipelines
Training
Experiments
Models
Endpoints
Batch predictions
Metadata
Marketplace
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

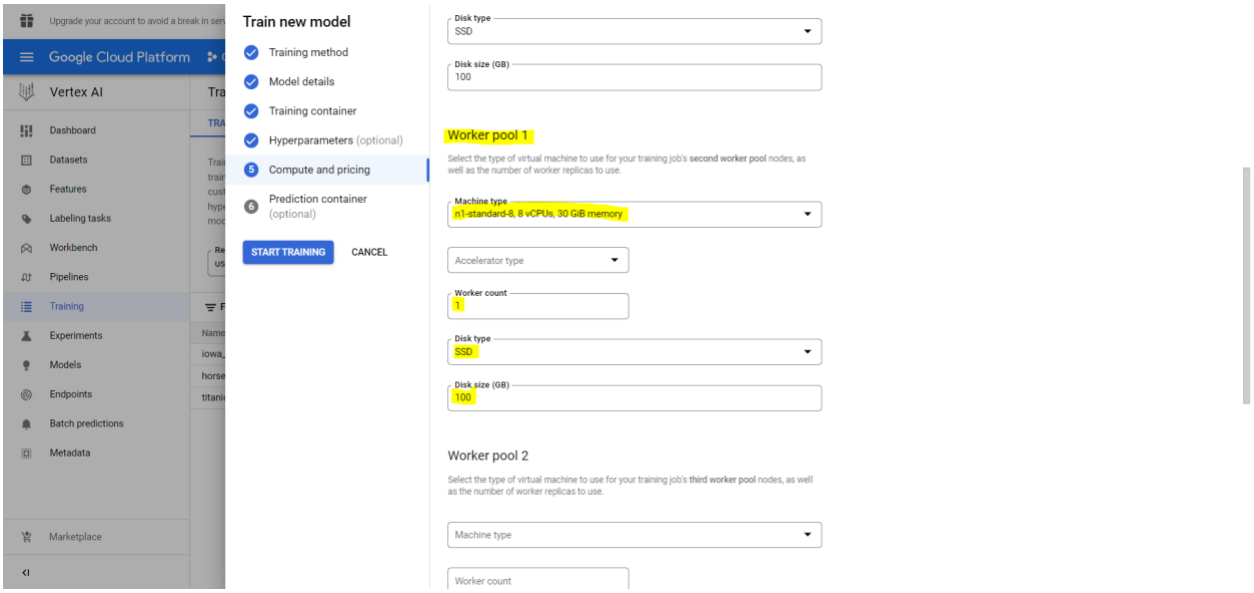

Upgrade your account to avoid a break in ser
Google Cloud Platform
Vertex AI
Dashboard
Datasets
Features
Labeling tasks
Workbench
Pipelines
Training
Experiments
Models
Endpoints
Batch predictions
Metadata
Marketplace
Train new model
Training method
2 Model details
3 Training container
4 Hyperparameters (optional)
5 Compute and pricing
6 Prediction container (optional)
START TRAINING
CANCEL
Model name *
multiworker-cassava
Encryption
Use a customer-managed encryption key (CMEK)
Service account
Select a service account to use with your model.
Service account
BROWSE
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
SHOW LESS
CONTINUE



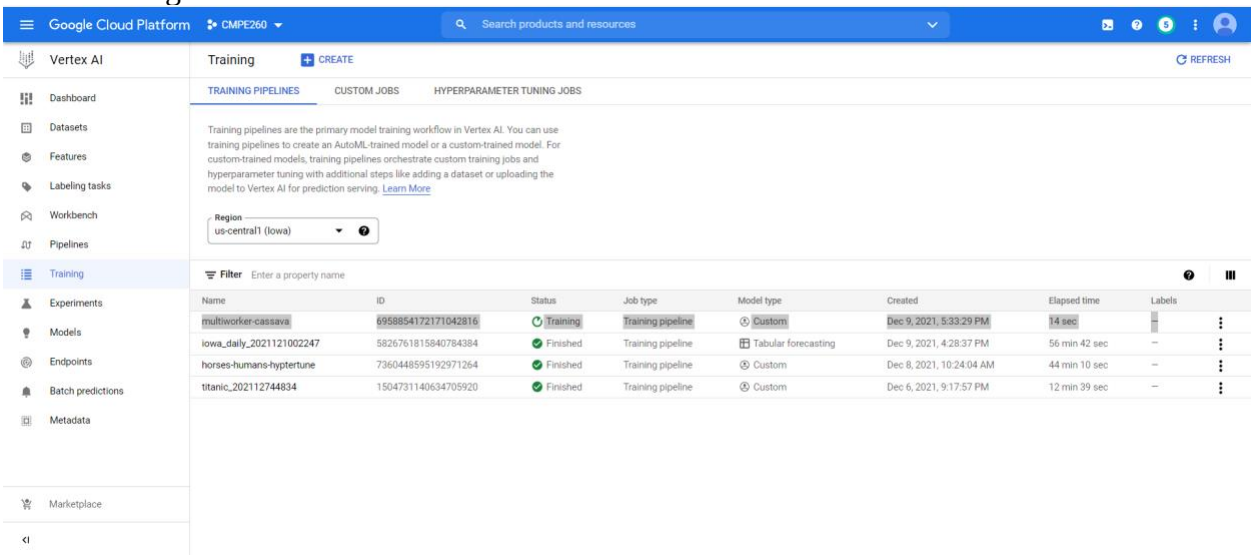
Worker Pool 0 Configuration:



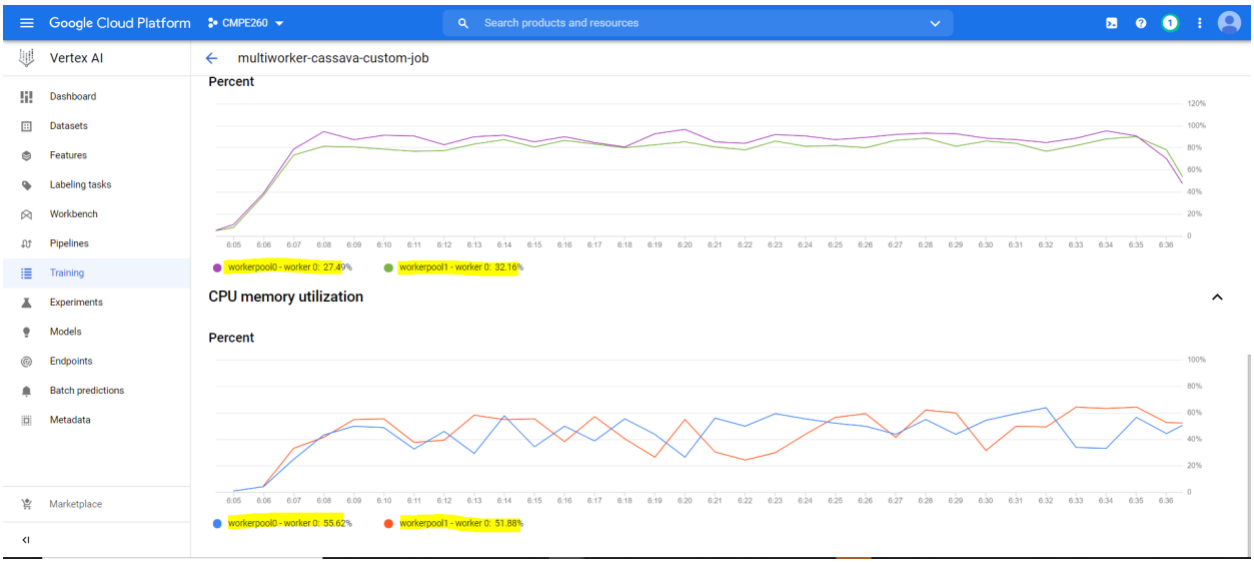
Worker Pool 1 Configuration:



Start Training



Results


Google Cloud Platform
CMPE260
Search products and resources
Vertex AI
multiworker-cassava-custom-job
Dashboard
Datasets
Features
Labeling tasks
Workbench
Pipelines
Training
Experiments
Models
Endpoints
Batch predictions
Metadata
Marketplace
Percent
workerpool0 - worker 0: 27.40%
workerpool1 - worker 0: 32.16%
CPU memory utilization
Percent
workerpool0 - worker 0: 55.62%
workerpool1 - worker 0: 51.88%