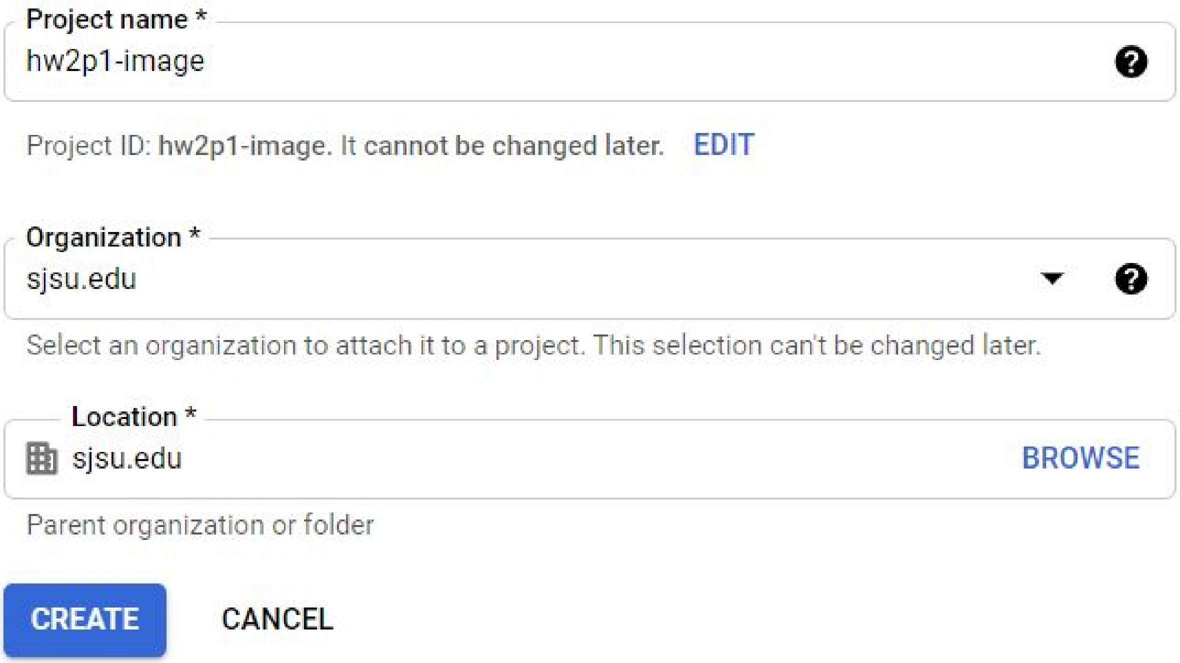
1. Custom Training:

Here I am using the same project that I have created for Image classification AutoML.

1. Project id: hw2p1-image



Project name *
hw2p1-image

Project ID: **hw2p1-image**. It cannot be changed later.

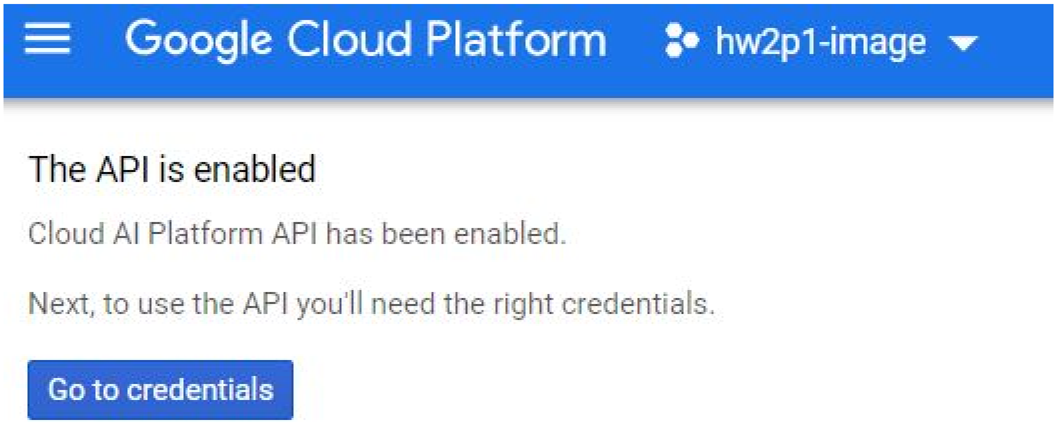
Organization *
sjsu.edu

Select an organization to attach it to a project. This selection can't be changed later.

Location *
sjsu.edu

Parent organization or folder

2. API is enabled

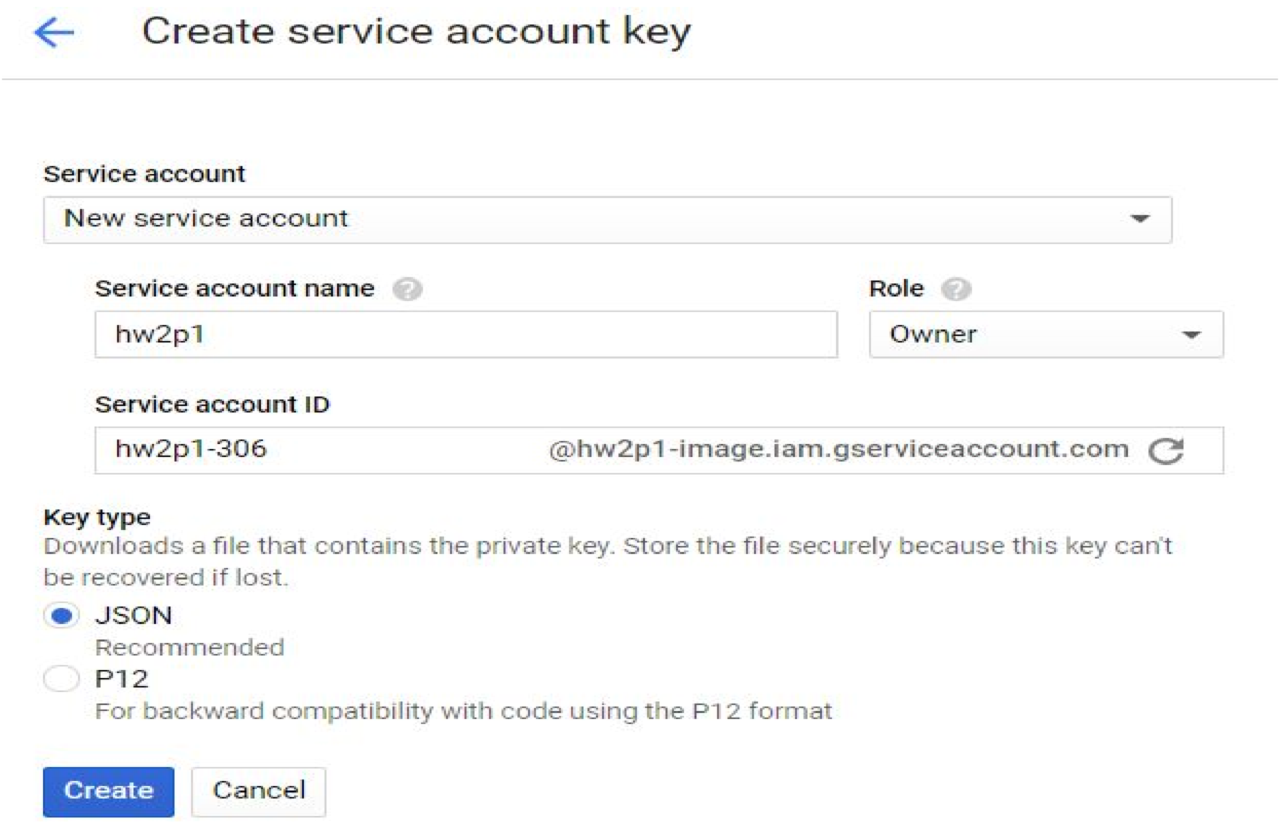


The API is enabled

Cloud AI Platform API has been enabled.

Next, to use the API you'll need the right credentials.

3. Authentication Setup



Create service account key

Service account

New service account

Service account name: hw2p1

Role: Owner

Service account ID: hw2p1-306 @hw2p1-image.iam.gserviceaccount.com

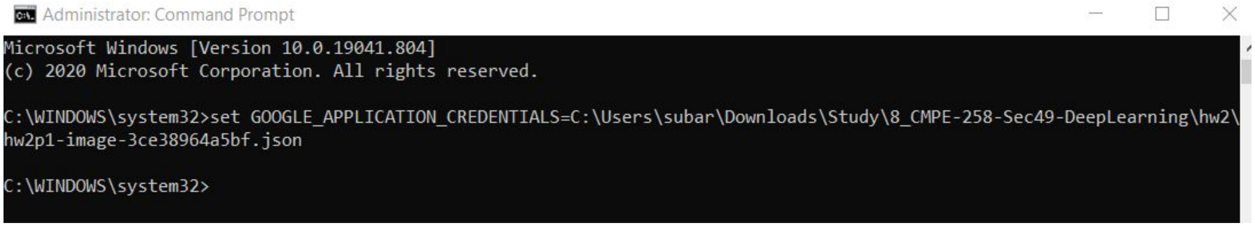
Key type

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

● JSON
Recommended

○ P12
For backward compatibility with code using the P12 format

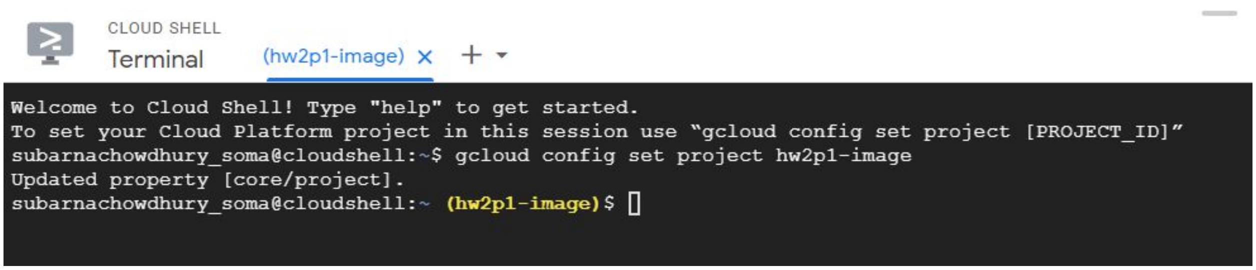
4. Set environment variable



Administrator: Command Prompt

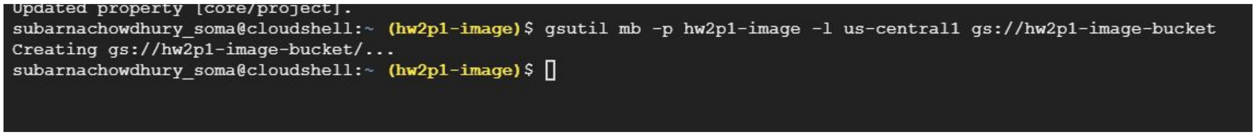
```
Microsoft Windows [Version 10.0.19041.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>set GOOGLE_APPLICATION_CREDENTIALS=C:\Users\subar\Downloads\Study\8_CMPE-258-Sec49-DeepLearning\hw2\
hw2p1-image-3ce38964a5bf.json

C:\WINDOWS\system32>
```

5. Cloud Console:



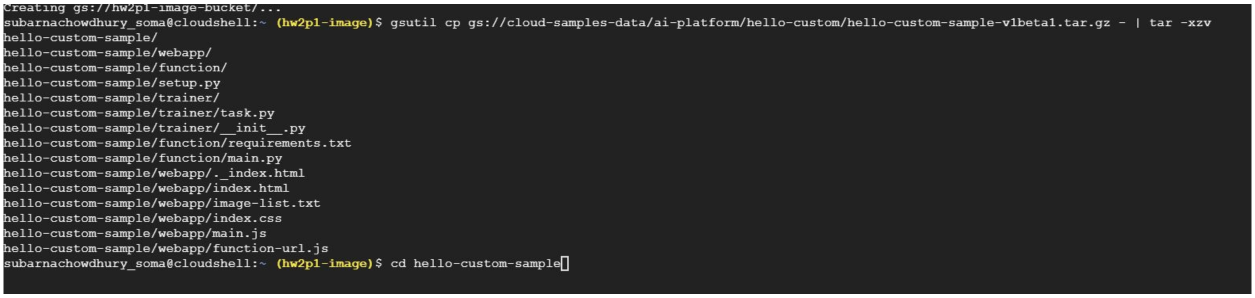
```
Welcome to Cloud Shell! Type "help" to get started.
To set your Cloud Platform project in this session use "gcloud config set project [PROJECT_ID]"
subarnachowdhury_soma@cloudshell:~$ gcloud config set project hw2p1-image
Updated property [core/project].
subarnachowdhury_soma@cloudshell:~ (hw2p1-image)$
```

6. Created Google Cloud Storage Bucket



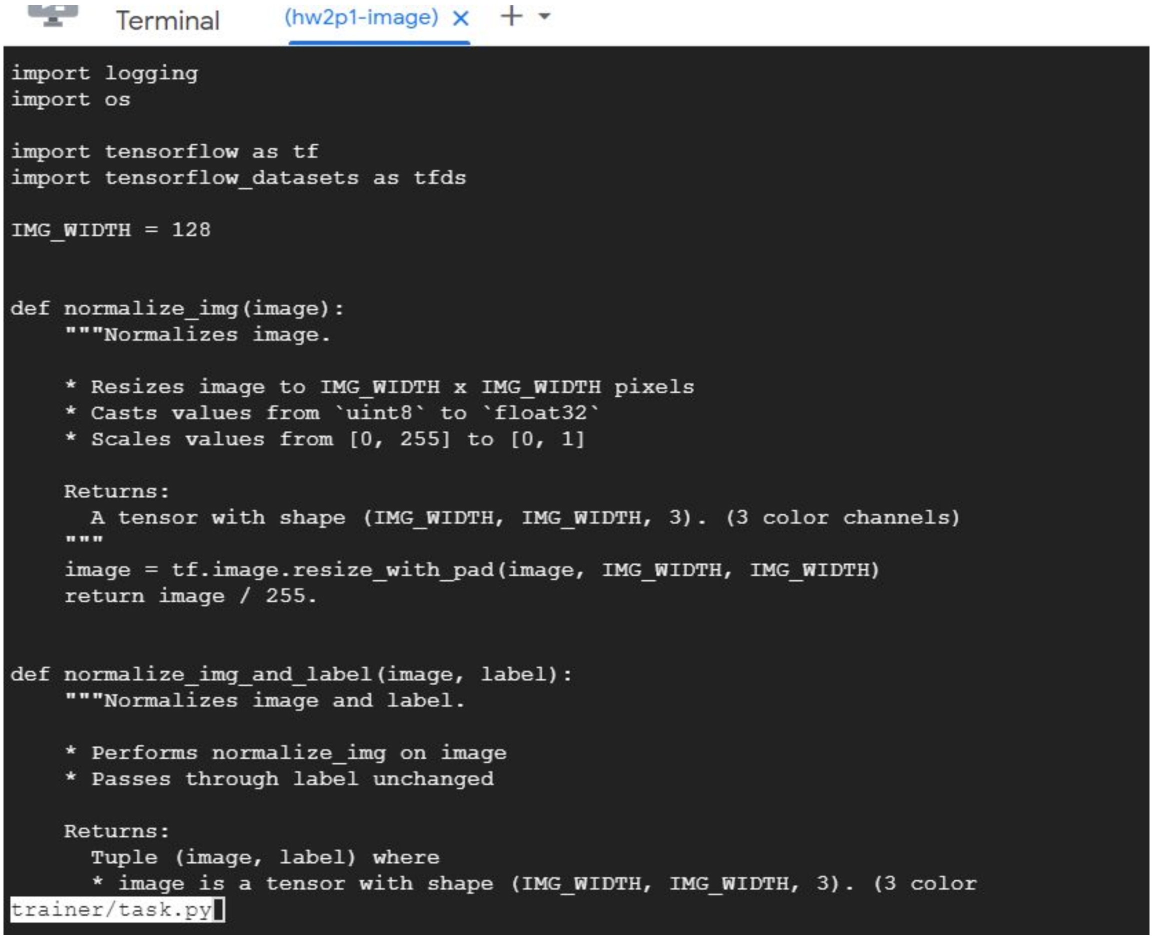
```
Updated property [core/project].
subarnachowdhury_soma@cloudshell:~ (hw2p1-image)$ gsutil mb -p hw2p1-image -l us-central1 gs://hw2p1-image-bucket
Creating gs://hw2p1-image-bucket/...
subarnachowdhury_soma@cloudshell:~ (hw2p1-image)$ 
```

7. Downloaded sample code:



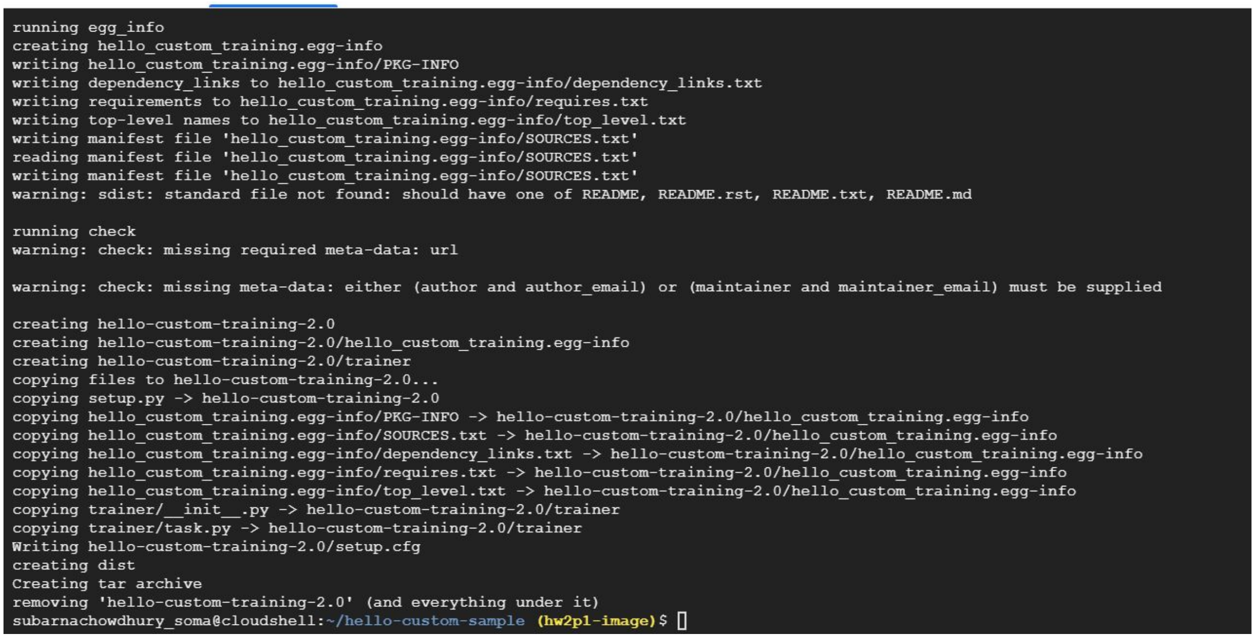
```
Creating gs://hw2p1-image-bucket/...
subarnachowdhury_soma@cloudshell:~ (hw2p1-image)$ gsutil cp gs://cloud-samples-data/ai-platform/hello-custom/hello-custom-sample-v1beta1.tar.gz - | tar -xzv
hello-custom-sample/
hello-custom-sample/webapp/
hello-custom-sample/function/
hello-custom-sample/setup.py
hello-custom-sample/trainer/
hello-custom-sample/trainer/task.py
hello-custom-sample/trainer/__init__.py
hello-custom-sample/function/requirements.txt
hello-custom-sample/function/main.py
hello-custom-sample/webapp/._index.html
hello-custom-sample/webapp/index.html
hello-custom-sample/webapp/image-list.txt
hello-custom-sample/webapp/index.css
hello-custom-sample/webapp/main.js
hello-custom-sample/webapp/function-url.js
subarnachowdhury_soma@cloudshell:~ (hw2p1-image)$ cd hello-custom-sample
```

8. Explore the training code:



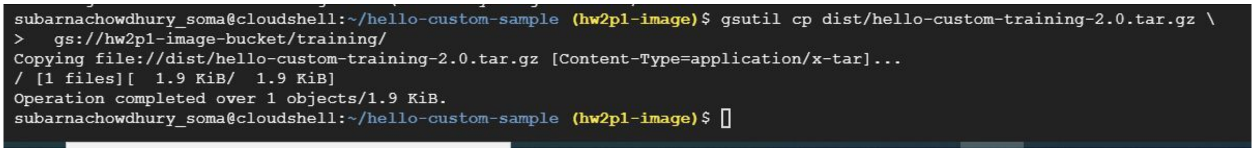
```
import logging
import os

import tensorflow as tf
import tensorflow_datasets as tfds

IMG_WIDTH = 128


def normalize_img(image):
    """Normalizes image.

    * Resizes image to IMG_WIDTH x IMG_WIDTH pixels
    * Casts values from `uint8` to `float32`
    * Scales values from [0, 255] to [0, 1]

    Returns:
      A tensor with shape (IMG_WIDTH, IMG_WIDTH, 3). (3 color channels)
    """
    image = tf.image.resize_with_pad(image, IMG_WIDTH, IMG_WIDTH)
    return image / 255.


def normalize_img_and_label(image, label):
    """Normalizes image and label.

    * Performs normalize_img on image
    * Passes through label unchanged

    Returns:
      Tuple (image, label) where
      * image is a tensor with shape (IMG_WIDTH, IMG_WIDTH, 3). (3 color
trainer/task.py
```

9. Setup.py



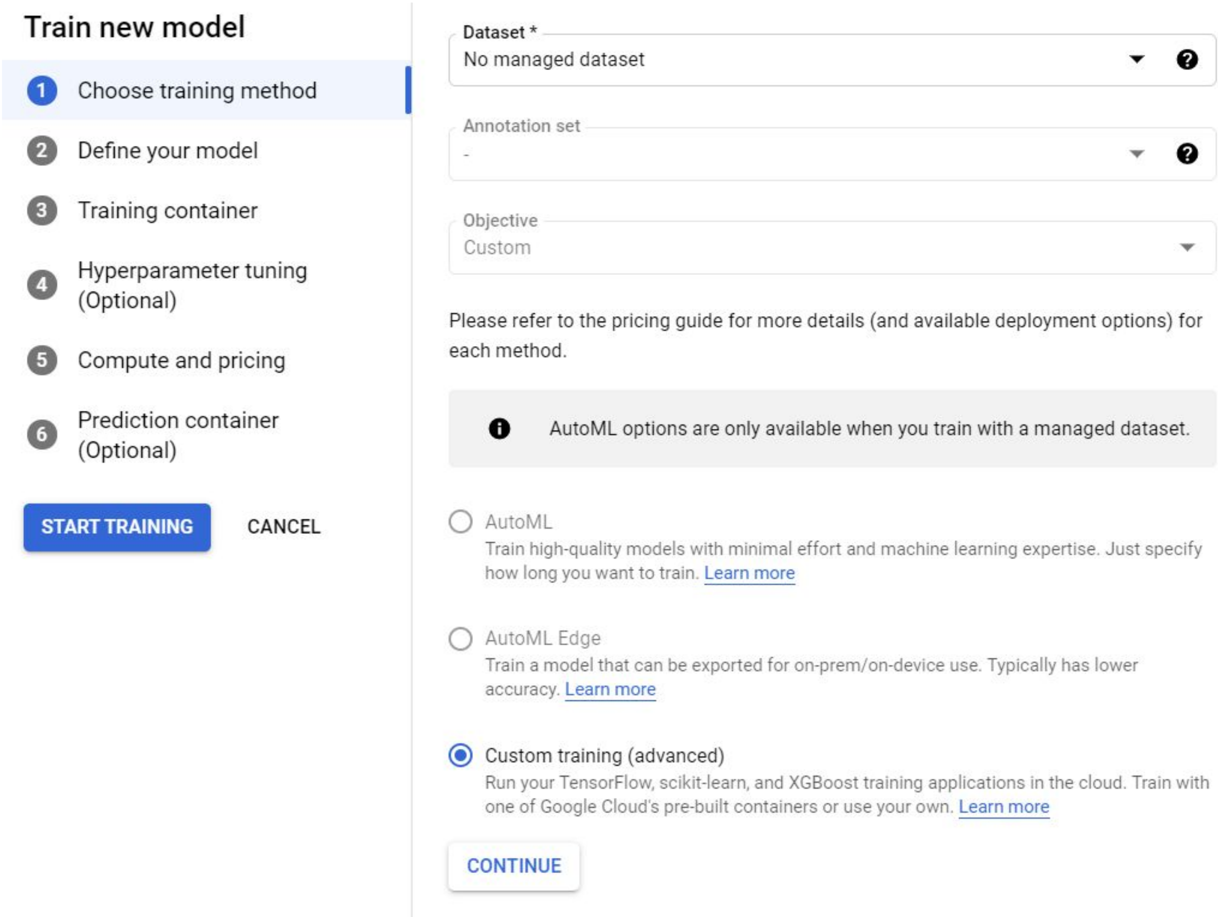
```
running egg_info
creating hello_custom_training.egg-info
writing hello_custom_training.egg-info/PKG-INFO
writing dependency_links to hello_custom_training.egg-info/dependency_links.txt
writing requirements to hello_custom_training.egg-info/requires.txt
writing top-level names to hello_custom_training.egg-info/top_level.txt
writing manifest file 'hello_custom_training.egg-info/SOURCES.txt'
reading manifest file 'hello_custom_training.egg-info/SOURCES.txt'
writing manifest file 'hello_custom_training.egg-info/SOURCES.txt'
warning: sdist: standard file not found: should have one of README, README.rst, README.txt, README.md

running check
warning: check: missing required meta-data: url

warning: check: missing meta-data: either (author and author_email) or (maintainer and maintainer_email) must be supplied

creating hello-custom-training-2.0
creating hello-custom-training-2.0/hello_custom_training.egg-info
creating hello-custom-training-2.0/trainer
copying files to hello-custom-training-2.0...
copying setup.py -> hello-custom-training-2.0
copying hello_custom_training.egg-info/PKG-INFO -> hello-custom-training-2.0/hello_custom_training.egg-info
copying hello_custom_training.egg-info/SOURCES.txt -> hello-custom-training-2.0/hello_custom_training.egg-info
copying hello_custom_training.egg-info/dependency_links.txt -> hello-custom-training-2.0/hello_custom_training.egg-info
copying hello_custom_training.egg-info/requires.txt -> hello-custom-training-2.0/hello_custom_training.egg-info
copying hello_custom_training.egg-info/top_level.txt -> hello-custom-training-2.0/hello_custom_training.egg-info
copying trainer/__init__.py -> hello-custom-training-2.0/trainer
copying trainer/task.py -> hello-custom-training-2.0/trainer
Writing hello-custom-training-2.0/setup.cfg
creating dist
Creating tar archive
removing 'hello-custom-training-2.0' (and everything under it)
subarnachowdhury_soma@cloudshell:~/hello-custom-sample (hw2p1-image)$ 
```

10. Running Hello Custom Training to Cloud Storage



```
subarnachowdhury_soma@cloudshell:~/hello-custom-sample (hw2p1-image)$ gsutil cp dist/hello-custom-training-2.0.tar.gz \
>   gs://hw2p1-image-bucket/training/
Copying file://dist/hello-custom-training-2.0.tar.gz [Content-Type=application/x-tar]...
/ [1 files][  1.9 KiB/  1.9 KiB]
Operation completed over 1 objects/1.9 KiB.
subarnachowdhury_soma@cloudshell:~/hello-custom-sample (hw2p1-image)$ 
```

11. Created Custom Training Pipeline- Training method



Train new model

1 Choose training method
2 Define your model
3 Training container
4 Hyperparameter tuning (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.

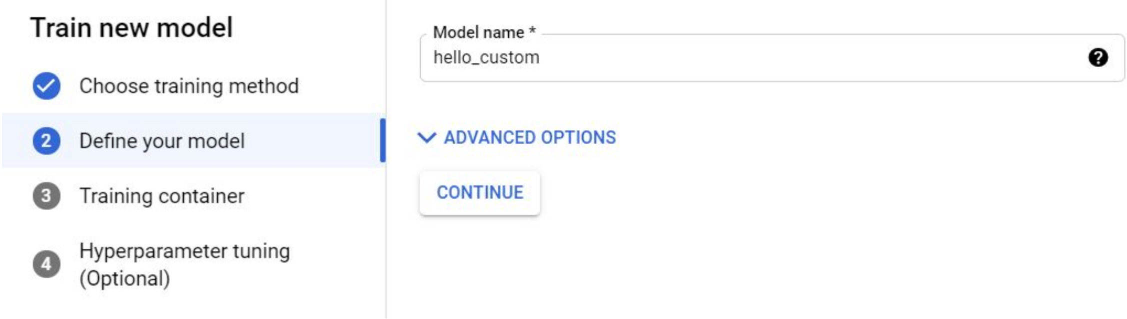
○ AutoML
Train high-quality models with minimal effort and machine learning expertise. Just specify how long you want to train. Learn more

○ 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

12. Defined Model Name:



Train new model

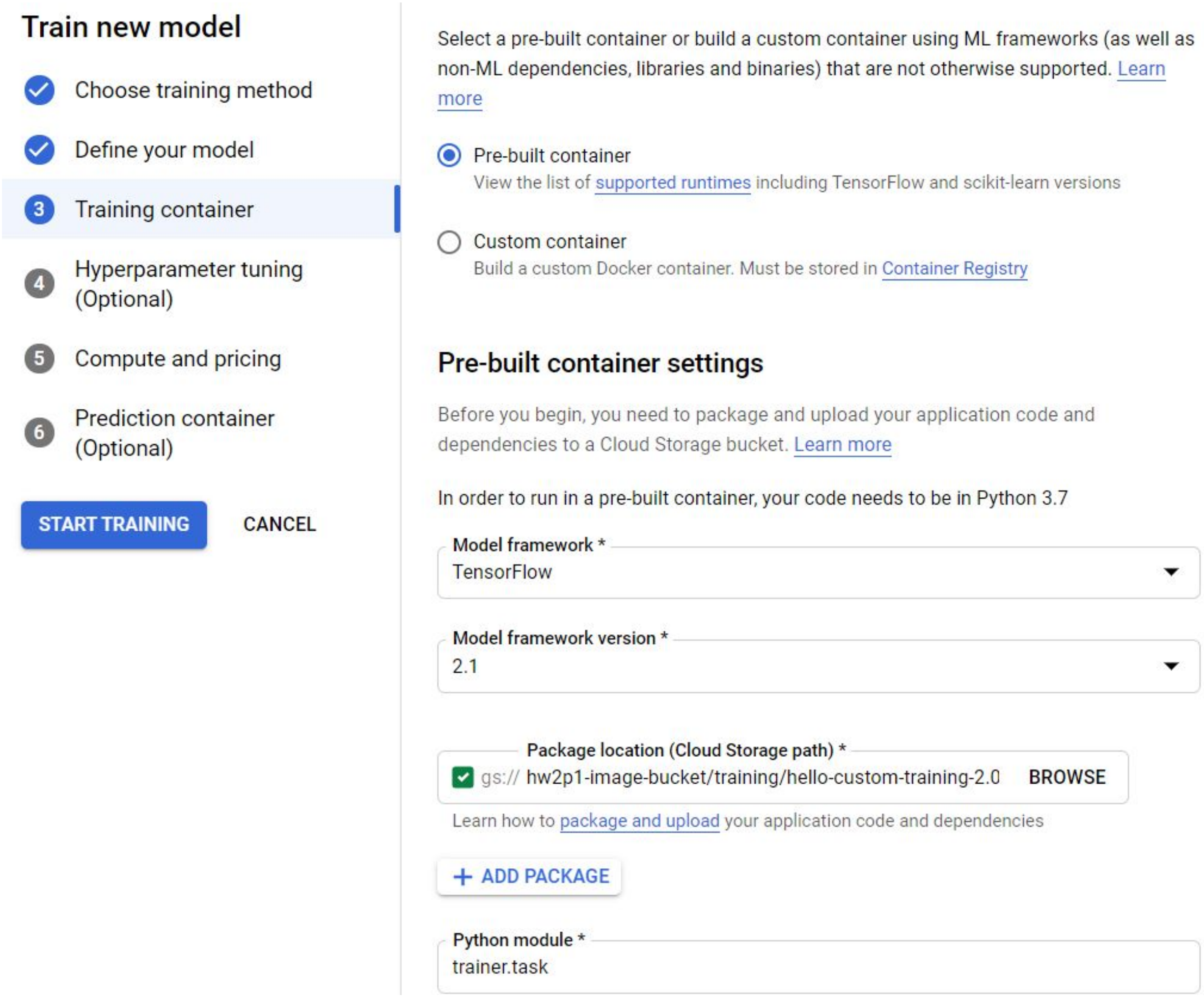
✓ Choose training method
2 Define your model
3 Training container
4 Hyperparameter tuning (Optional)

Model name *
hello_custom

ADVANCED OPTIONS

CONTINUE

13. Training Container



Train new model

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

Pre-built container settings

Before you begin, you need to package and upload your application code and dependencies to a Cloud Storage bucket. Learn more

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

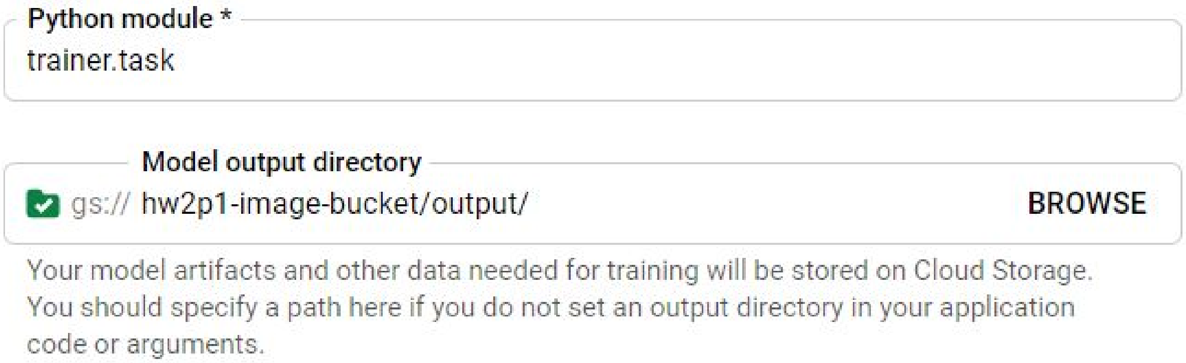
Model framework * : TensorFlow

Model framework version * : 2.1

Package location (Cloud Storage path) * : ✓ gs:// hw2p1-image-bucket/training/hello-custom-training-2.0 BROWSE

Learn how to package and upload your application code and dependencies

Python module * : trainer.task

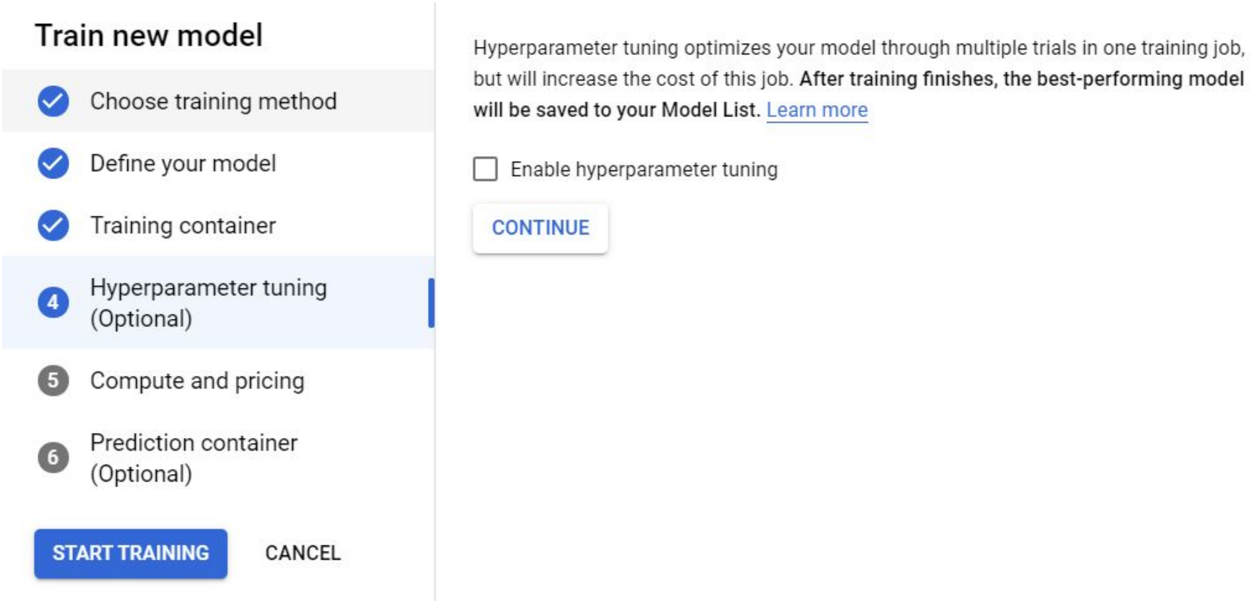


Python module * : trainer.task

Model output directory : ✓ gs:// hw2p1-image-bucket/output/ 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.

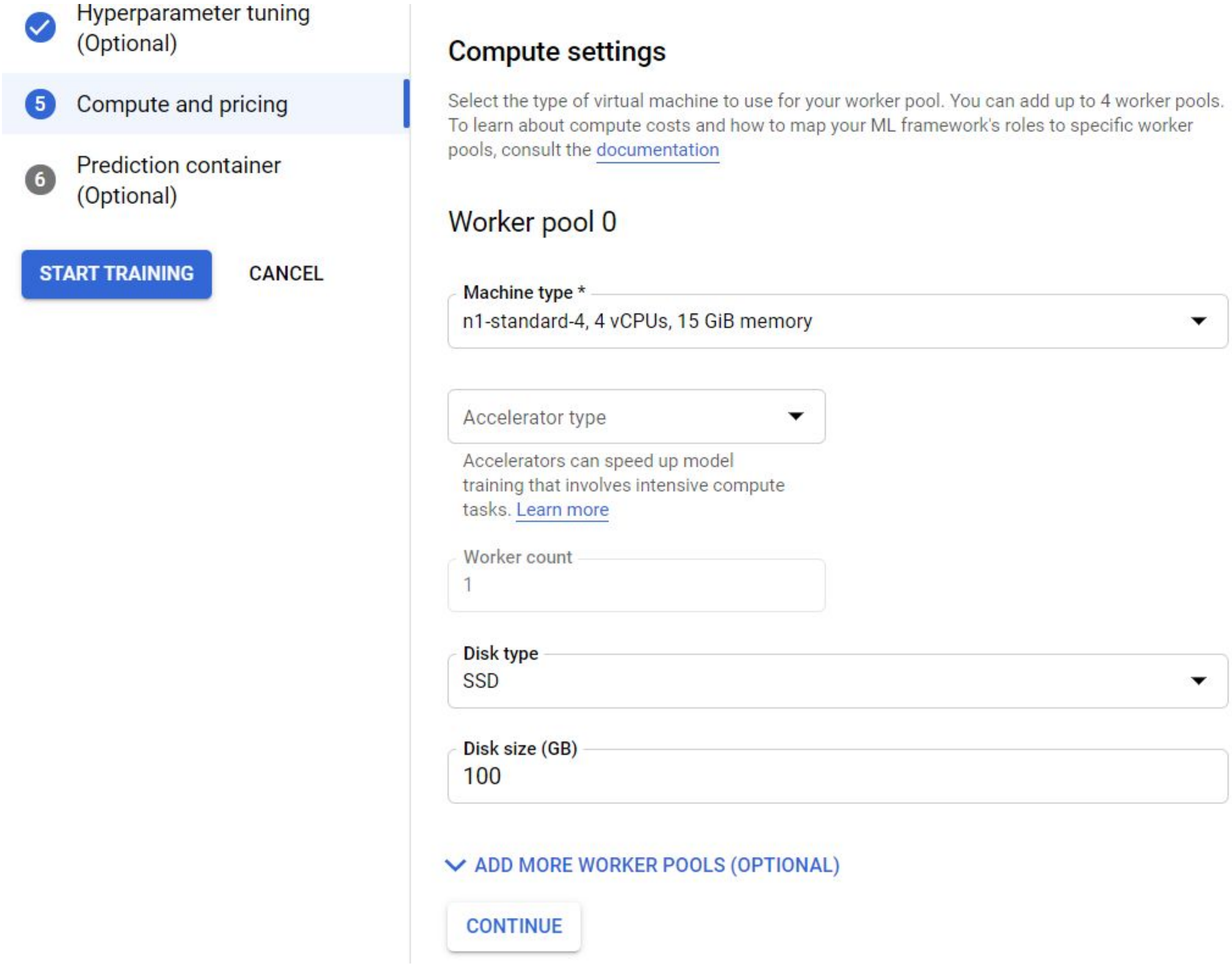
14. Hyperparam Tuning Cleared



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

15. Compute and Pricing


Hyperparameter tuning (Optional)
5 Compute and pricing
6 Prediction container (Optional)
START TRAINING
CANCEL
ADD MORE WORKER POOLS (OPTIONAL)
CONTINUE

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

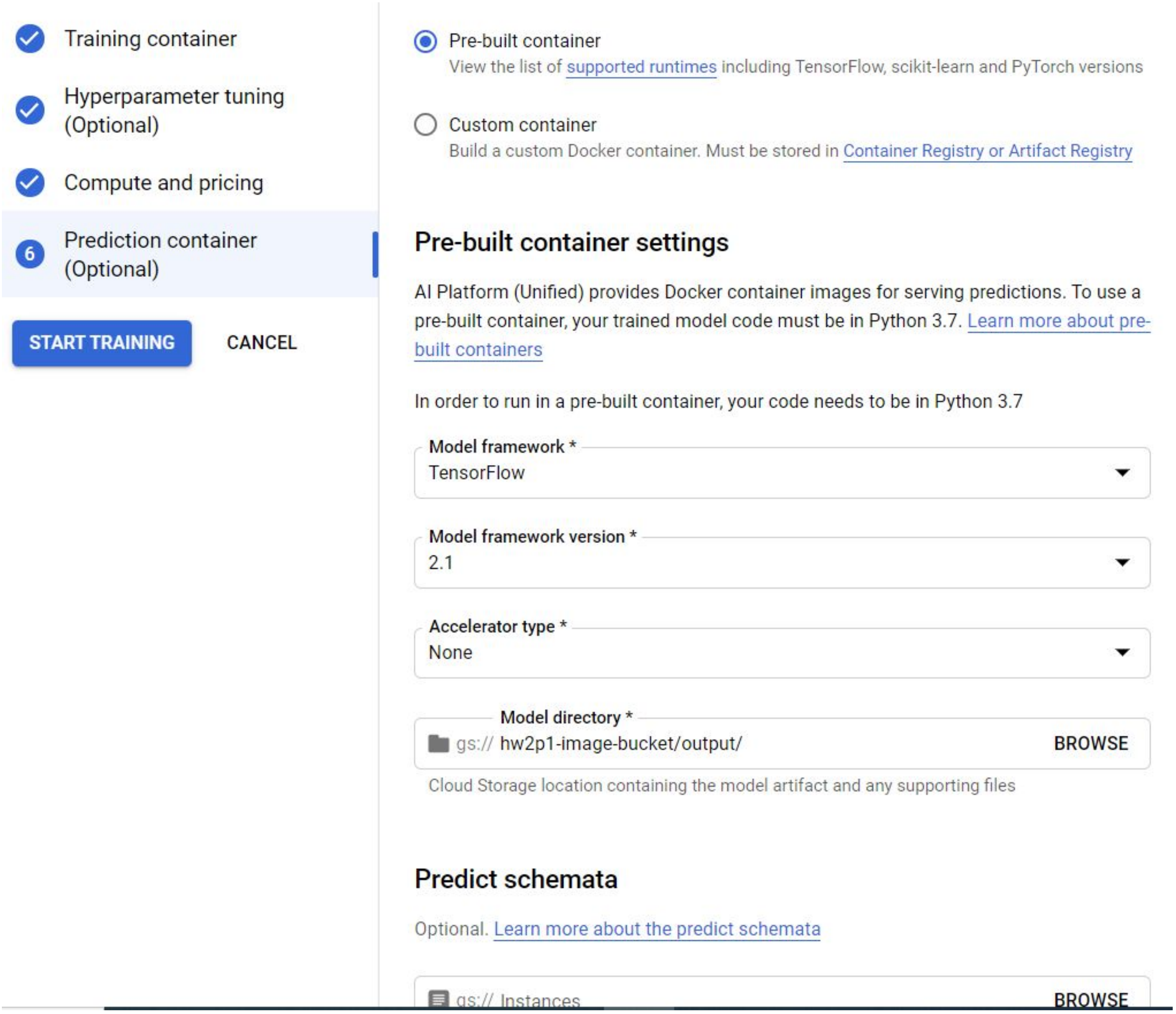
Accelerators can speed up model training that involves intensive compute tasks. Learn more

Worker count: 1

Disk type: SSD

Disk size (GB): 100

16. Prediction Container:



✓ Training container
✓ Hyperparameter tuning (Optional)
✓ Compute and pricing
6 Prediction container (Optional)
START TRAINING
CANCEL

● 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

AI Platform (Unified) 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

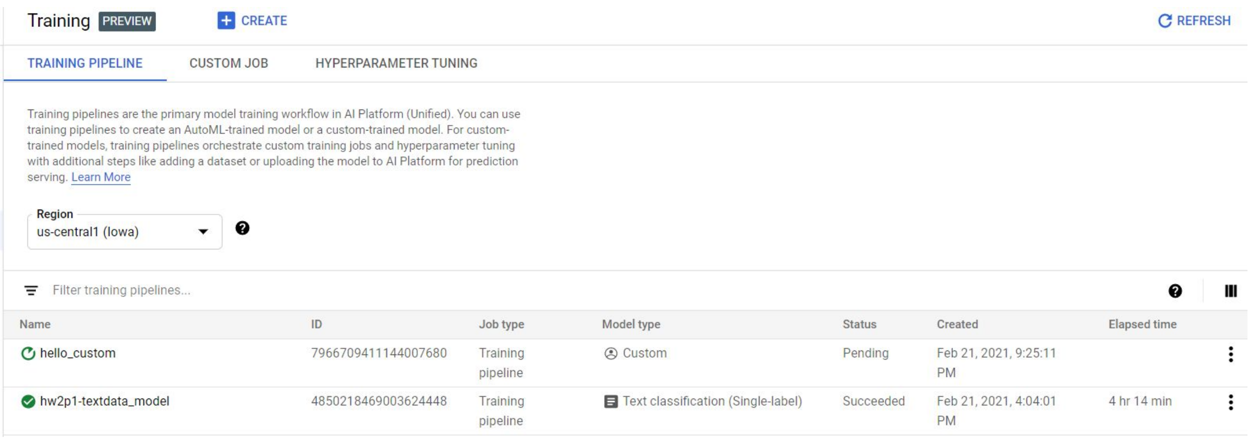
Model directory *: gs:// hw2p1-image-bucket/output/ BROWSE
Cloud Storage location containing the model artifact and any supporting files

Predict schemata

Optional. Learn more about the predict schemata

gs:// Instances BROWSE

17. Training Started:



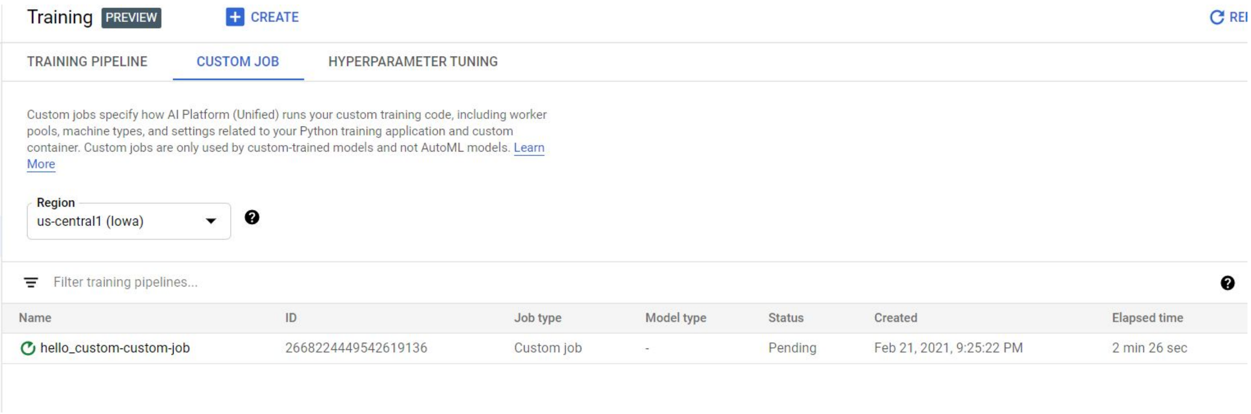
Training pipelines are the primary model training workflow in AI Platform (Unified). You can use training pipelines to create an AutoML-trained model or a custom-trained model. For custom-trained models, training pipelines orchestrate custom training jobs and hyperparameter tuning with additional steps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

Region: us-central1 (Iowa)

Filter training pipelines...

Name	ID	Job type	Model type	Status	Created	Elapsed time
hello_custom	7966709411144007680	Training pipeline	Custom	Pending	Feb 21, 2021, 9:25:11 PM	
hw2p1-textdata_model	4850218469003624448	Training pipeline	Text classification (Single-label)	Succeeded	Feb 21, 2021, 4:04:01 PM	4 hr 14 min

18. Created Custom Job



Training PREVIEW ➕ CREATE ↻ REI

TRAINING PIPELINE | CUSTOM JOB | HYPERPARAMETER TUNING

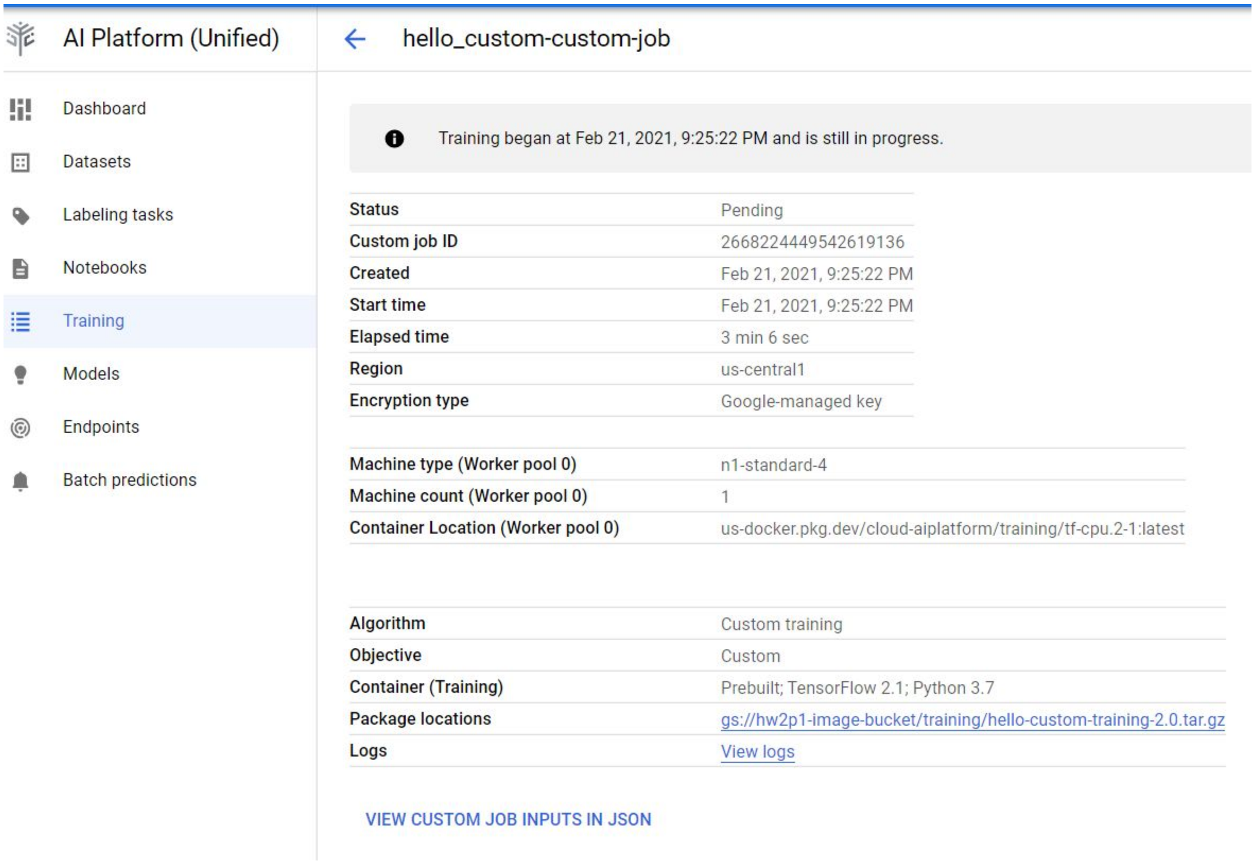
Custom jobs specify how AI Platform (Unified) runs your custom training code, including worker pools, machine types, and settings related to your Python training application and custom container. Custom jobs are only used by custom-trained models and not AutoML models. Learn More

Region: us-central1 (Iowa)

Filter training pipelines...

Name	ID	Job type	Model type	Status	Created	Elapsed time
hello_custom-custom-job	2668224449542619136	Custom job	-	Pending	Feb 21, 2021, 9:25:22 PM	2 min 26 sec

19. Custom Job Log



← hello_custom-custom-job

Training began at Feb 21, 2021, 9:25:22 PM and is still in progress.

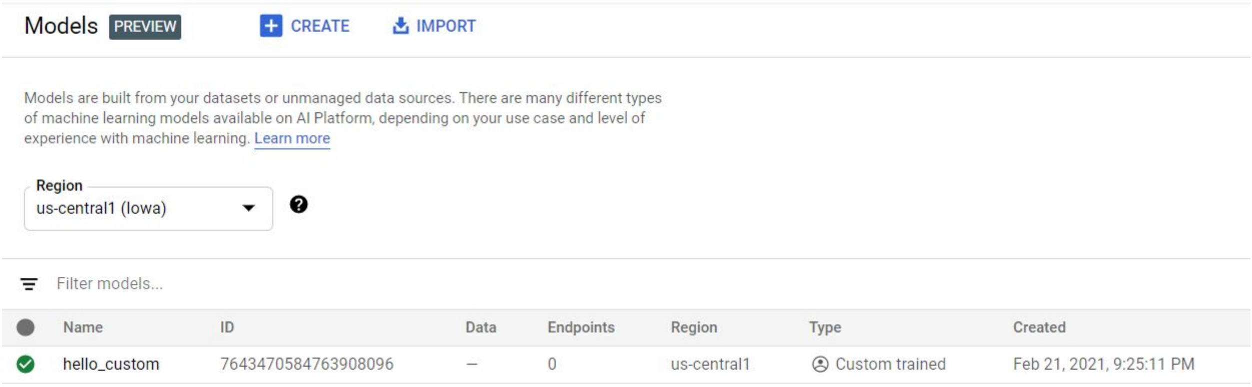
Status	Pending
Custom job ID	2668224449542619136
Created	Feb 21, 2021, 9:25:22 PM
Start time	Feb 21, 2021, 9:25:22 PM
Elapsed time	3 min 6 sec
Region	us-central1
Encryption type	Google-managed key

Machine type (Worker pool 0)	n1-standard-4
Machine count (Worker pool 0)	1
Container Location (Worker pool 0)	us-docker.pkg.dev/cloud-aiplatform/training/tf-cpu.2-1:latest

Algorithm	Custom training
Objective	Custom
Container (Training)	Prebuilt; TensorFlow 2.1; Python 3.7
Package locations	gs://hw2p1-image-bucket/training/hello-custom-training-2.0.tar.gz
Logs	View logs

VIEW CUSTOM JOB INPUTS IN JSON

20. Model Created

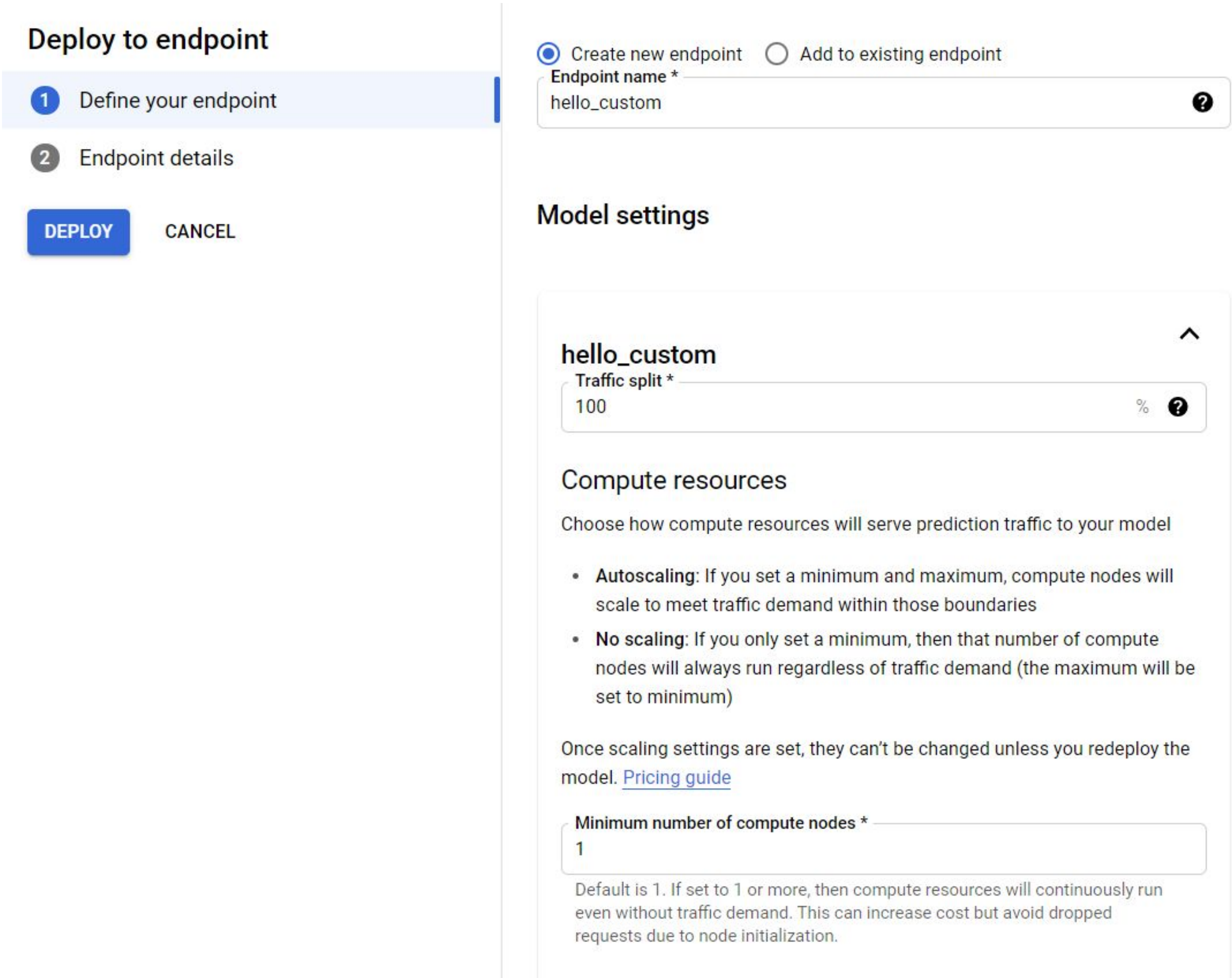


Models are built from your datasets or unmanaged data sources. There are many different types of machine learning models available on AI Platform, depending on your use case and level of experience with machine learning. Learn more

Region: us-central1 (Iowa)

	Name	ID	Data	Endpoints	Region	Type	Created
✓	hello_custom	7643470584763908096	–	0	us-central1	Custom trained	Feb 21, 2021, 9:25:11 PM

21. Created Deployment Endpoint



Deploy to endpoint

1 Define your endpoint

2 Endpoint details

● Create new endpoint ○ Add to existing endpoint

Endpoint name *: hello_custom

Model settings

hello_custom

Traffic split *: 100 %

Compute resources

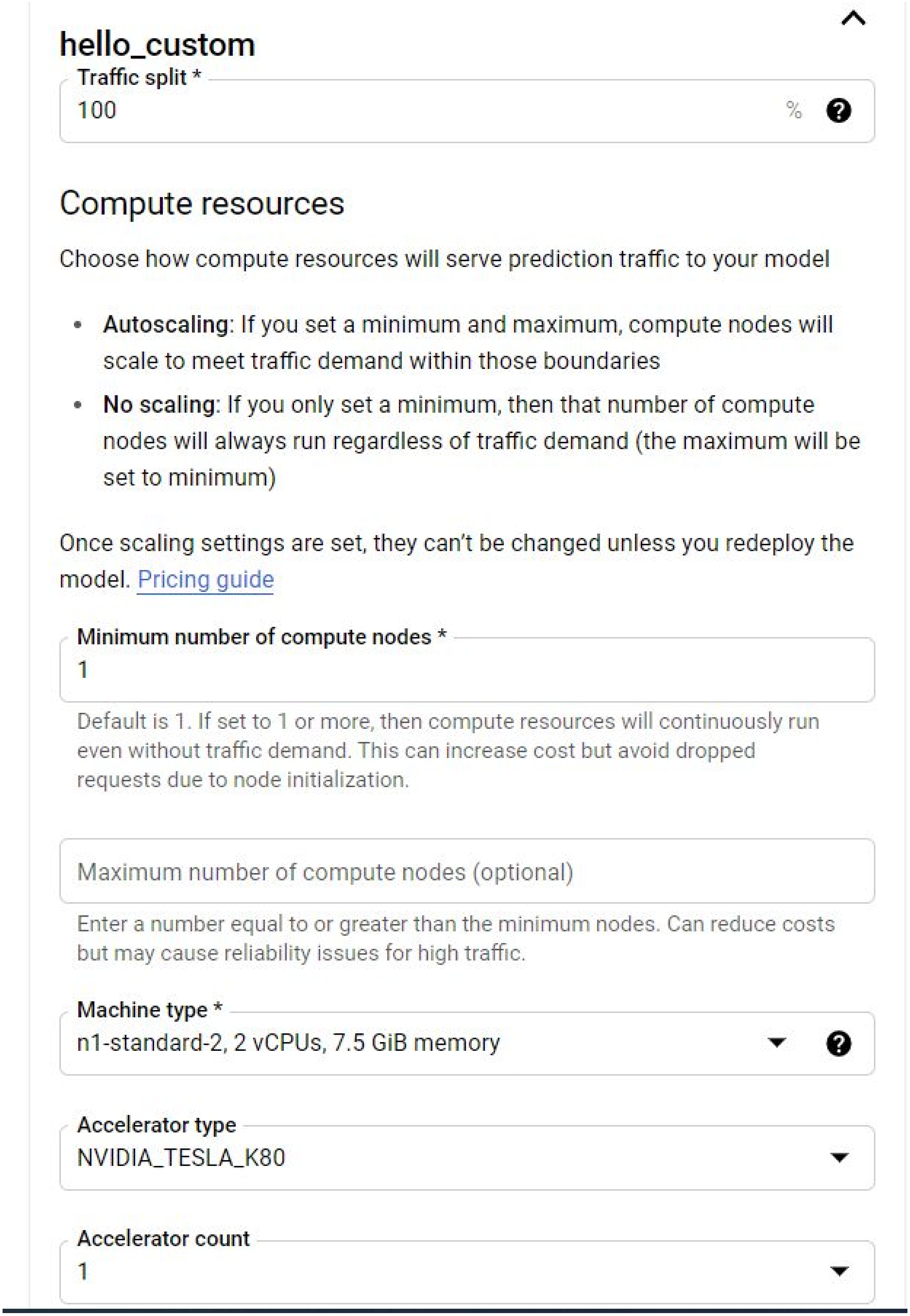
Choose how compute resources will serve prediction traffic to your model

- **Autoscaling**: If you set a minimum and maximum, compute nodes will scale to meet traffic demand within those boundaries
- **No scaling**: If you only set a minimum, then that number of compute nodes will always run regardless of traffic demand (the maximum will be set to minimum)

Once scaling settings are set, they can't be changed unless you redeploy the model. Pricing guide

Minimum number of compute nodes *: 1

Default is 1. If set to 1 or more, then compute resources will continuously run even without traffic demand. This can increase cost but avoid dropped requests due to node initialization.



hello_custom

Traffic split *
100 %

Compute resources

Choose how compute resources will serve prediction traffic to your model

- **Autoscaling**: If you set a minimum and maximum, compute nodes will scale to meet traffic demand within those boundaries
- **No scaling**: If you only set a minimum, then that number of compute nodes will always run regardless of traffic demand (the maximum will be set to minimum)

Once scaling settings are set, they can't be changed unless you redeploy the model. Pricing guide

Minimum number of compute nodes *
1

Default is 1. If set to 1 or more, then compute resources will continuously run even without traffic demand. This can increase cost but avoid dropped requests due to node initialization.

Maximum number of compute nodes (optional)

Enter a number equal to or greater than the minimum nodes. Can reduce costs but may cause reliability issues for high traffic.

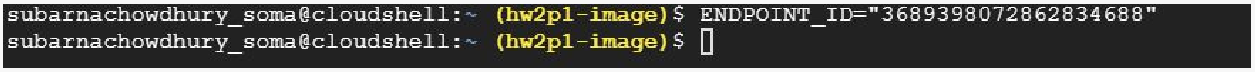
Machine type *
n1-standard-2, 2 vCPUs, 7.5 GiB memory

Accelerator type
NVIDIA_TESLA_K80

Accelerator count
1

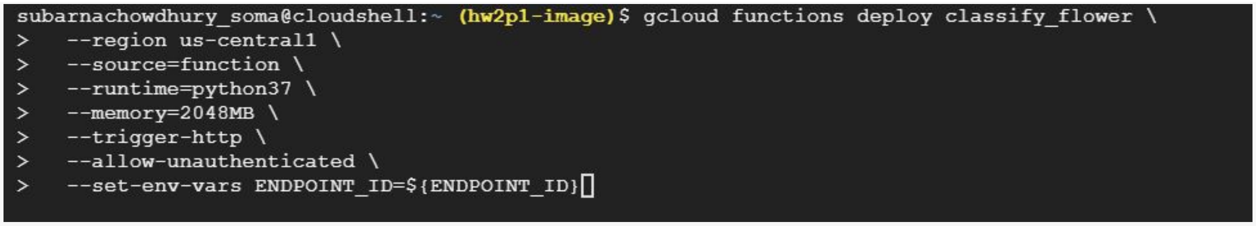
22. From Sample Request Pan:

```
ENDPOINT_ID="3689398072862834688"
PROJECT_ID="hw2p1-image"
INPUT_DATA_FILE="INPUT-JSON"
```



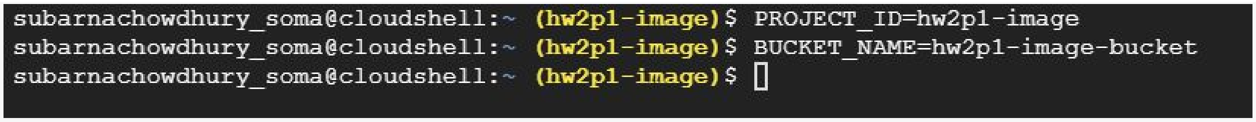
```
subarnachowdhury_soma@cloudshell:~ (hw2p1-image)$ ENDPOINT_ID="3689398072862834688"
subarnachowdhury_soma@cloudshell:~ (hw2p1-image)$
```

23. Deploying Cloud Function:

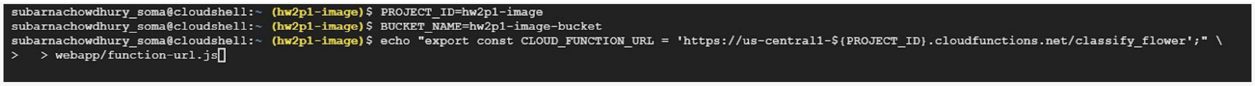


```
subarnachowdhury_soma@cloudshell:~ (hw2p1-image)$ gcloud functions deploy classify_flower \
>   --region us-central1 \
>   --source=function \
>   --runtime=python37 \
>   --memory=2048MB \
>   --trigger-http \
>   --allow-unauthenticated \
>   --set-env-vars ENDPOINT_ID=${ENDPOINT_ID}
```

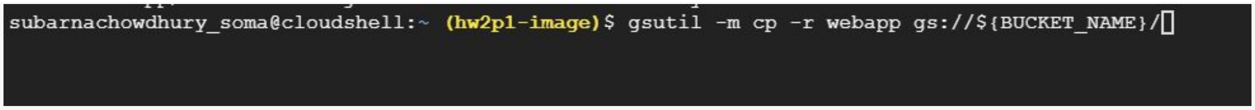
24. Web App deployment:



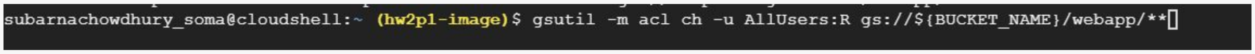
```
subarnachowdhury_soma@cloudshell:~ (hw2p1-image)$ PROJECT_ID=hw2p1-image
subarnachowdhury_soma@cloudshell:~ (hw2p1-image)$ BUCKET_NAME=hw2p1-image-bucket
subarnachowdhury_soma@cloudshell:~ (hw2p1-image)$
```



```
subarnachowdhury_soma@cloudshell:~ (hw2p1-image)$ PROJECT_ID=hw2p1-image
subarnachowdhury_soma@cloudshell:~ (hw2p1-image)$ BUCKET_NAME=hw2p1-image-bucket
subarnachowdhury_soma@cloudshell:~ (hw2p1-image)$ echo "export const CLOUD_FUNCTION_URL = 'https://us-central1-${PROJECT_ID}.cloudfunctions.net/classify_flower';" \
>   > webapp/function-url.js
```

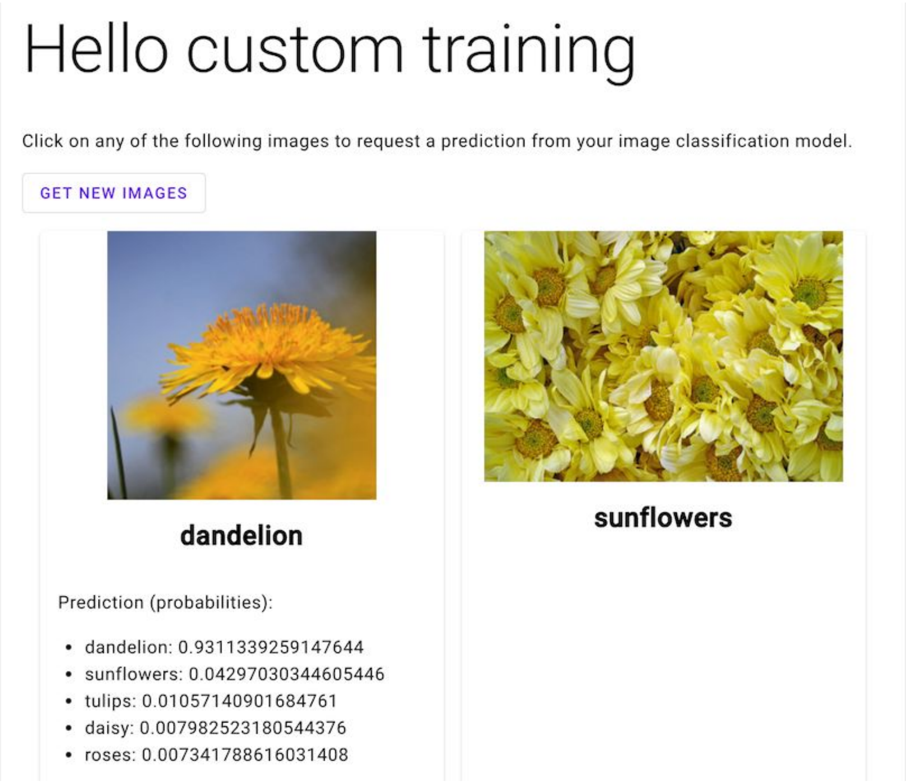


```
subarnachowdhury_soma@cloudshell:~ (hw2p1-image)$ gsutil -m cp -r webapp gs://${BUCKET_NAME}/
```



```
subarnachowdhury_soma@cloudshell:~ (hw2p1-image)$ gsutil -m acl ch -u AllUsers:R gs://${BUCKET_NAME}/webapp/**
```

25. Prediction in web app: Navigate to this url:
https://storage.googleapis.com/hw2p1-image-bucket/webapp/index.html



Hello custom training

Click on any of the following images to request a prediction from your image classification model.

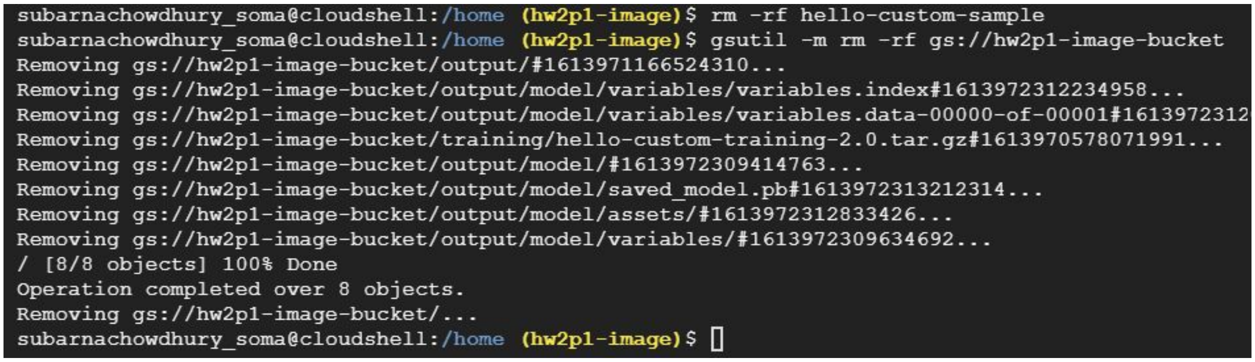
dandelion

Prediction (probabilities):

- dandelion: 0.9311339259147644
- sunflowers: 0.04297030344605446
- tulips: 0.01057140901684761
- daisy: 0.007982523180544376
- roses: 0.007341788616031408

sunflowers

26. Deleted Bucket, function etc



```
subarnachowdhury_soma@cloudshell:/home (hw2p1-image)$ rm -rf hello-custom-sample
subarnachowdhury_soma@cloudshell:/home (hw2p1-image)$ gsutil -m rm -rf gs://hw2p1-image-bucket
Removing gs://hw2p1-image-bucket/output/#1613971166524310...
Removing gs://hw2p1-image-bucket/output/model/variables/variables.index#1613972312234958...
Removing gs://hw2p1-image-bucket/output/model/variables/variables.data-00000-of-00001#1613972312
Removing gs://hw2p1-image-bucket/training/hello-custom-training-2.0.tar.gz#1613970578071991...
Removing gs://hw2p1-image-bucket/output/model/#1613972309414763...
Removing gs://hw2p1-image-bucket/output/model/saved_model.pb#1613972313212314...
Removing gs://hw2p1-image-bucket/output/model/assets/#1613972312833426...
Removing gs://hw2p1-image-bucket/output/model/variables/#1613972309634692...
/ [8/8 objects] 100% Done
Operation completed over 8 objects.
Removing gs://hw2p1-image-bucket/...
subarnachowdhury_soma@cloudshell:/home (hw2p1-image)$ 
```