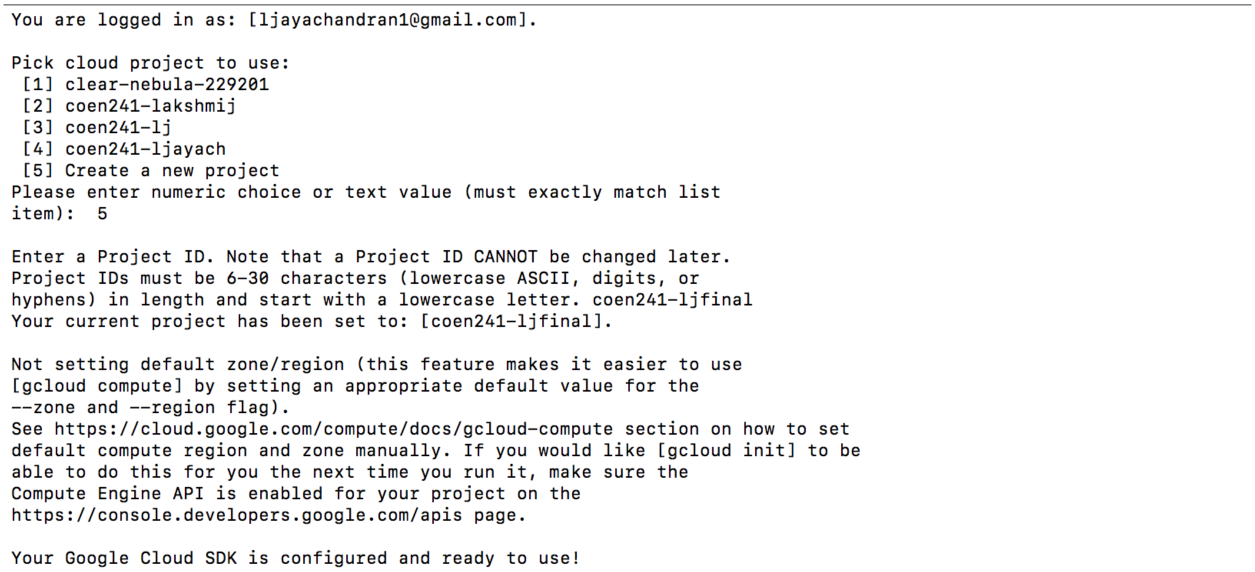
HW 2: Distributed ML training using Kubeflow on GKE

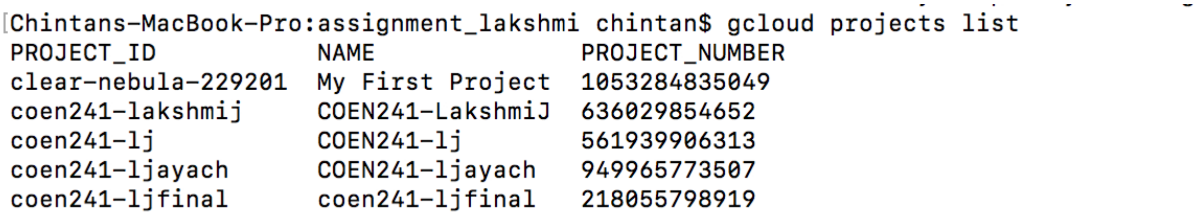
Steps:

1. Create a project on GCP. Install gcloud, kubectl, and docker in local machine.



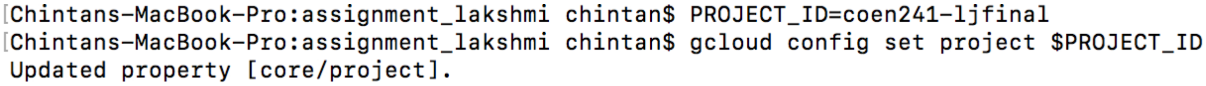
```
You are logged in as: [ljayachandran1@gmail.com].

Pick cloud project to use:
 [1] clear-nebula-229201
 [2] coen241-lakshmij
 [3] coen241-lj
 [4] coen241-ljayach
 [5] Create a new project
Please enter numeric choice or text value (must exactly match list
item):  5

Enter a Project ID. Note that a Project ID CANNOT be changed later.
Project IDs must be 6-30 characters (lowercase ASCII, digits, or
hyphens) in length and start with a lowercase letter. coen241-ljfinal
Your current project has been set to: [coen241-ljfinal].

Not setting default zone/region (this feature makes it easier to use
[gcloud compute] by setting an appropriate default value for the
--zone and --region flag).
See https://cloud.google.com/compute/docs/gcloud-compute section on how to set
default compute region and zone manually. If you would like [gcloud init] to be
able to do this for you the next time you run it, make sure the
Compute Engine API is enabled for your project on the
https://console.developers.google.com/apis page.

Your Google Cloud SDK is configured and ready to use!
```

2. Set up a Kubernetes cluster for our project through GKE.

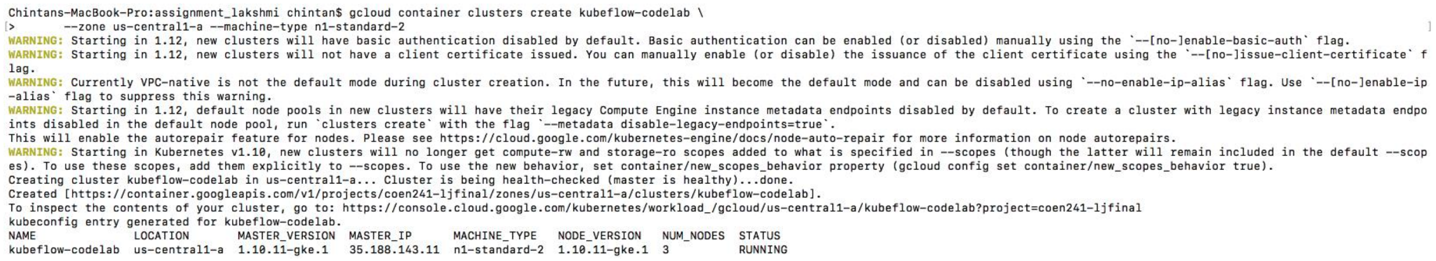


```
[Chintans-MacBook-Pro:assignment_lakshmi chintan$ gcloud projects list
PROJECT_ID           NAME                 PROJECT_NUMBER
clear-nebula-229201  My First Project     1053284835049
coen241-lakshmij     COEN241-LakshmiJ     636029854652
coen241-lj           COEN241-lj           561939906313
coen241-ljayach      COEN241-ljayach      949965773507
coen241-ljfinal      coen241-ljfinal      218055798919
```

3. Create a new cluster named "kubeflow-codelab", located in the zone us-central1-a

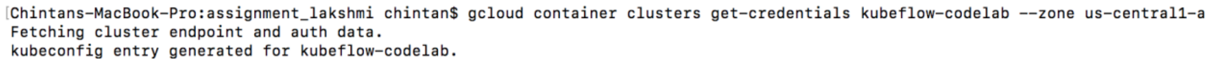


```
[Chintans-MacBook-Pro:assignment_lakshmi chintan$ PROJECT_ID=coen241-ljfinal
[Chintans-MacBook-Pro:assignment_lakshmi chintan$ gcloud config set project $PROJECT_ID
 Updated property [core/project].
```



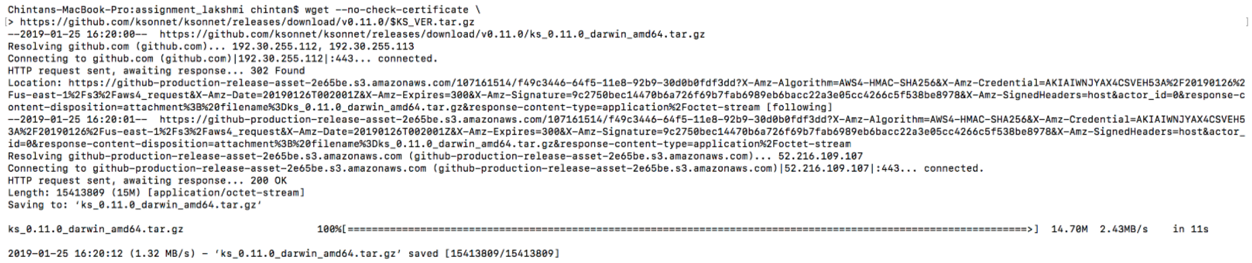
```
Chintans-MacBook-Pro:assignment_lakshmi chintan$ gcloud container clusters create kubeflow-codelab \
[>       --zone us-central1-a --machine-type n1-standard-2
WARNING: Starting in 1.12, new clusters will have basic authentication disabled by default. Basic authentication can be enabled (or disabled) manually using the `--[no-]enable-basic-auth` flag.
WARNING: Starting in 1.12, new clusters will not have a client certificate issued. You can manually enable (or disable) the issuance of the client certificate using the `--[no-]issue-client-certificate` f
lag.
WARNING: Currently VPC-native is not the default mode during cluster creation. In the future, this will become the default mode and can be disabled using `--no-enable-ip-alias` flag. Use `--[no-]enable-ip
-alias` flag to suppress this warning.
WARNING: Starting in 1.12, default node pools in new clusters will have their legacy Compute Engine instance metadata endpoints disabled by default. To create a cluster with legacy instance metadata endpo
ints disabled in the default node pool, run `clusters create` with the flag `--metadata disable-legacy-endpoints=true`.
This will enable the autorepair feature for nodes. Please see https://cloud.google.com/kubernetes-engine/docs/node-auto-repair for more information on node autorepairs.
WARNING: Starting in Kubernetes v1.10, new clusters will no longer get compute-rw and storage-ro scopes added to what is specified in --scopes (though the latter will remain included in the default --scop
es). To use these scopes, add them explicitly to --scopes. To use the new behavior, set container/new_scopes_behavior property (gcloud config set container/new_scopes_behavior true).
Creating cluster kubeflow-codelab in us-central1-a... Cluster is being health-checked (master is healthy)...done.
Created [https://container.googleapis.com/v1/projects/coen241-ljfinal/zones/us-central1-a/clusters/kubeflow-codelab].
To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload_/gcloud/us-central1-a/kubeflow-codelab?project=coen241-ljfinal
kubeconfig entry generated for kubeflow-codelab.
NAME              LOCATION       MASTER_VERSION  MASTER_IP      MACHINE_TYPE   NODE_VERSION   NUM_NODES  STATUS
kubeflow-codelab  us-central1-a  1.10.11-gke.1   35.188.143.11  n1-standard-2  1.10.11-gke.1  3          RUNNING
```

4. Connect our local environment to the cluster so we can interact with it locally using the Kubernetes CLI tool and change the permissions on the cluster to allow kubeflow to run properly.

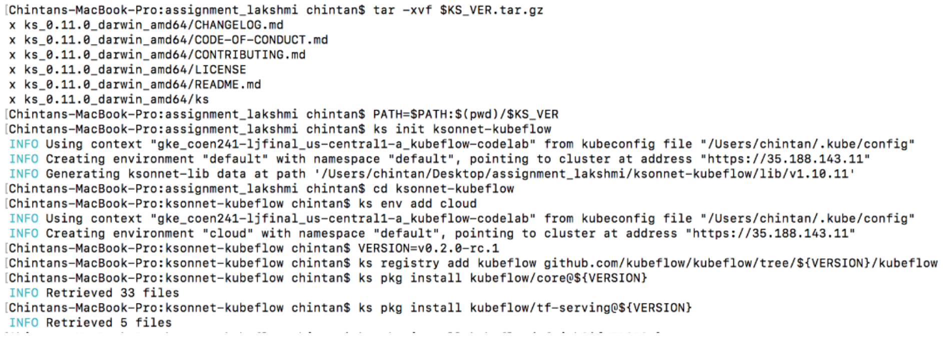


```
[Chintans-MacBook-Pro:assignment_lakshmi chintan$ gcloud container clusters get-credentials kubeflow-codelab --zone us-central1-a
Fetching cluster endpoint and auth data.
kubeconfig entry generated for kubeflow-codelab.
```

5. Download ksonnet's ks command and add it to your path. To create our ksonnet project directory, we will use ks init. Add our cluster as an available ksonnet environment.

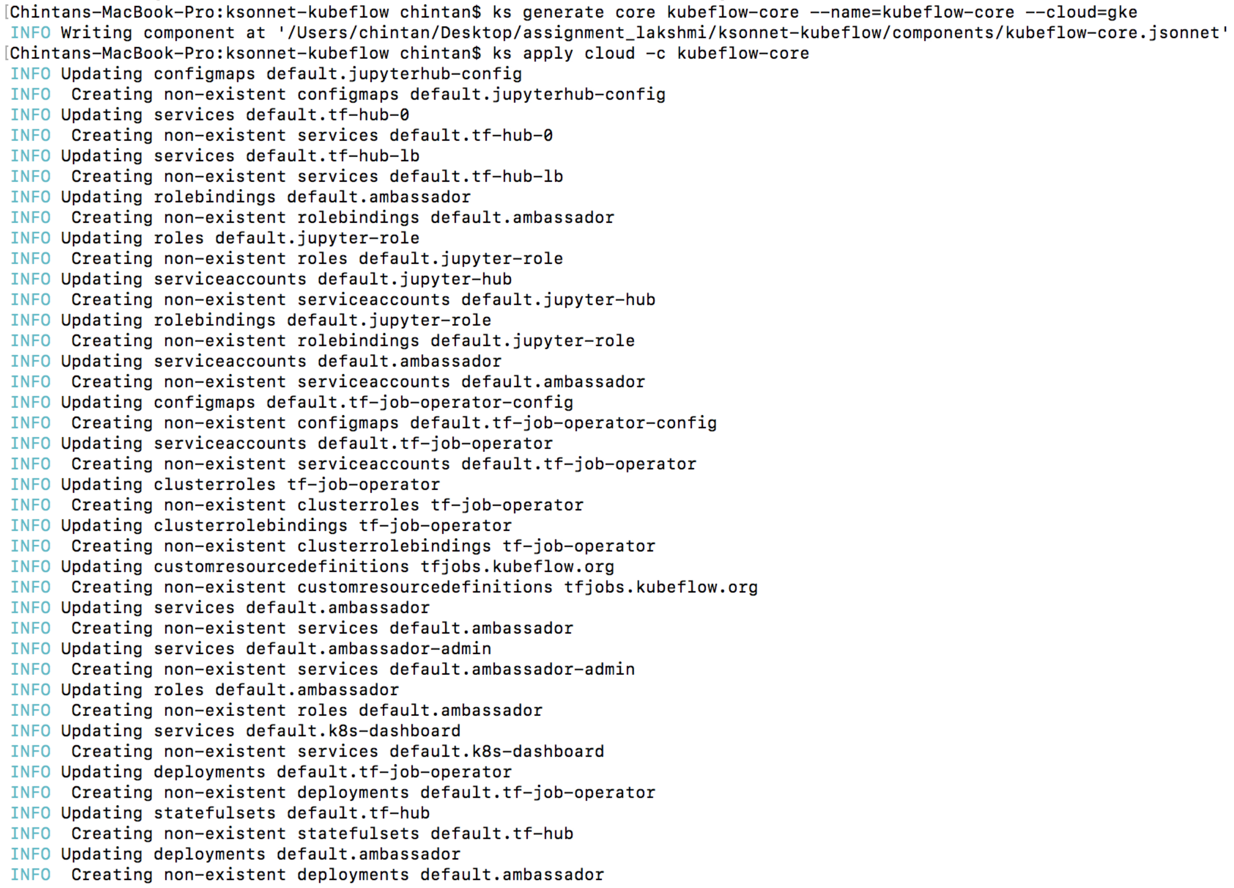


```
Chintans-MacBook-Pro:assignment_lakshmi chintan$ wget --no-check-certificate \
[> https://github.com/ksonnet/ksonnet/releases/download/v0.11.0/$KS_VER.tar.gz
--2019-01-25 16:20:00--  https://github.com/ksonnet/ksonnet/releases/download/v0.11.0/ks_0.11.0_darwin_amd64.tar.gz
Resolving github.com (github.com)... 192.30.255.112, 192.30.255.113
Connecting to github.com (github.com)|192.30.255.112|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-production-release-asset-2e65be.s3.amazonaws.com/107161514/f49c3446-64f5-11e8-92b9-30d0b0fdf3dd?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20190126%2
Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20190126T002001Z&X-Amz-Expires=300&X-Amz-Signature=9c2750bec14470b6a726f69b7fab6989eb6bacc22a3e05cc4266c5f538be8978&X-Amz-SignedHeaders=host&actor_id=0&response-c
ontent-disposition=attachment%3B%20filename%3Dks_0.11.0_darwin_amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2019-01-25 16:20:01--  https://github-production-release-asset-2e65be.s3.amazonaws.com/107161514/f49c3446-64f5-11e8-92b9-30d0b0fdf3dd?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH5
3A%2F20190126%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20190126T002001Z&X-Amz-Expires=300&X-Amz-Signature=9c2750bec14470b6a726f69b7fab6989eb6bacc22a3e05cc4266c5f538be8978&X-Amz-SignedHeaders=host&actor_
id=0&response-content-disposition=attachment%3B%20filename%3Dks_0.11.0_darwin_amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)... 52.216.109.107
Connecting to github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)|52.216.109.107|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 15413809 (15M) [application/octet-stream]
Saving to: ‘ks_0.11.0_darwin_amd64.tar.gz’

ks_0.11.0_darwin_amd64.tar.gz                100%[===================================================================================================>]  14.70M  2.43MB/s    in 11s

2019-01-25 16:20:12 (1.32 MB/s) - ‘ks_0.11.0_darwin_amd64.tar.gz’ saved [15413809/15413809]
```



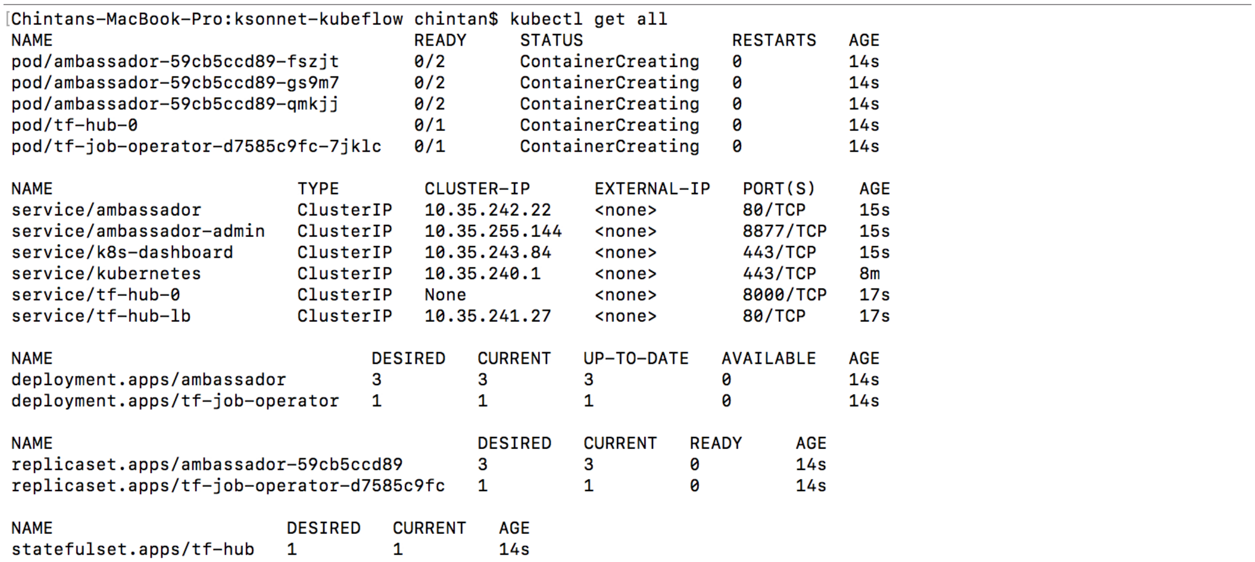
```
[Chintans-MacBook-Pro:assignment_lakshmi chintan$ tar -xvf $KS_VER.tar.gz
x ks_0.11.0_darwin_amd64/CHANGELOG.md
x ks_0.11.0_darwin_amd64/CODE-OF-CONDUCT.md
x ks_0.11.0_darwin_amd64/CONTRIBUTING.md
x ks_0.11.0_darwin_amd64/LICENSE
x ks_0.11.0_darwin_amd64/README.md
x ks_0.11.0_darwin_amd64/ks
[Chintans-MacBook-Pro:assignment_lakshmi chintan$ PATH=$PATH:$(pwd)/$KS_VER
[Chintans-MacBook-Pro:assignment_lakshmi chintan$ ks init ksonnet-kubeflow
 INFO Using context "gke_coen241-ljfinal_us-central1-a_kubeflow-codelab" from kubeconfig file "/Users/chintan/.kube/config"
 INFO Creating environment "default" with namespace "default", pointing to cluster at address "https://35.188.143.11"
 INFO Generating ksonnet-lib data at path '/Users/chintan/Desktop/assignment_lakshmi/ksonnet-kubeflow/lib/v1.10.11'
[Chintans-MacBook-Pro:assignment_lakshmi chintan$ cd ksonnet-kubeflow
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ ks env add cloud
 INFO Using context "gke_coen241-ljfinal_us-central1-a_kubeflow-codelab" from kubeconfig file "/Users/chintan/.kube/config"
 INFO Creating environment "cloud" with namespace "default", pointing to cluster at address "https://35.188.143.11"
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ VERSION=v0.2.0-rc.1
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ ks registry add kubeflow github.com/kubeflow/kubeflow/tree/${VERSION}/kubeflow
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ ks pkg install kubeflow/core@${VERSION}
 INFO Retrieved 33 files
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ ks pkg install kubeflow/tf-serving@${VERSION}
 INFO Retrieved 5 files
```

6. Add the kubeflow repository to our project, and then pull the Kubeflow packages down into our local project and apply the standard Kubeflow components to our cluster.



```
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ ks generate core kubeflow-core --name=kubeflow-core --cloud=gke
 INFO Writing component at '/Users/chintan/Desktop/assignment_lakshmi/ksonnet-kubeflow/components/kubeflow-core.jsonnet'
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ ks apply cloud -c kubeflow-core
 INFO Updating configmaps default.jupyterhub-config
 INFO  Creating non-existent configmaps default.jupyterhub-config
 INFO Updating services default.tf-hub-0
 INFO  Creating non-existent services default.tf-hub-0
 INFO Updating services default.tf-hub-lb
 INFO  Creating non-existent services default.tf-hub-lb
 INFO Updating rolebindings default.ambassador
 INFO  Creating non-existent rolebindings default.ambassador
 INFO Updating roles default.jupyter-role
 INFO  Creating non-existent roles default.jupyter-role
 INFO Updating serviceaccounts default.jupyter-hub
 INFO  Creating non-existent serviceaccounts default.jupyter-hub
 INFO Updating rolebindings default.jupyter-role
 INFO  Creating non-existent rolebindings default.jupyter-role
 INFO Updating serviceaccounts default.ambassador
 INFO  Creating non-existent serviceaccounts default.ambassador
 INFO Updating configmaps default.tf-job-operator-config
 INFO  Creating non-existent configmaps default.tf-job-operator-config
 INFO Updating serviceaccounts default.tf-job-operator
 INFO  Creating non-existent serviceaccounts default.tf-job-operator
 INFO Updating clusterroles tf-job-operator
 INFO  Creating non-existent clusterroles tf-job-operator
 INFO Updating clusterrolebindings tf-job-operator
 INFO  Creating non-existent clusterrolebindings tf-job-operator
 INFO Updating customresourcedefinitions tfjobs.kubeflow.org
 INFO  Creating non-existent customresourcedefinitions tfjobs.kubeflow.org
 INFO Updating services default.ambassador
 INFO  Creating non-existent services default.ambassador
 INFO Updating services default.ambassador-admin
 INFO  Creating non-existent services default.ambassador-admin
 INFO Updating roles default.ambassador
 INFO  Creating non-existent roles default.ambassador
 INFO Updating services default.k8s-dashboard
 INFO  Creating non-existent services default.k8s-dashboard
 INFO Updating deployments default.tf-job-operator
 INFO  Creating non-existent deployments default.tf-job-operator
 INFO Updating statefulsets default.tf-hub
 INFO  Creating non-existent statefulsets default.tf-hub
 INFO Updating deployments default.ambassador
 INFO  Creating non-existent deployments default.ambassador
```

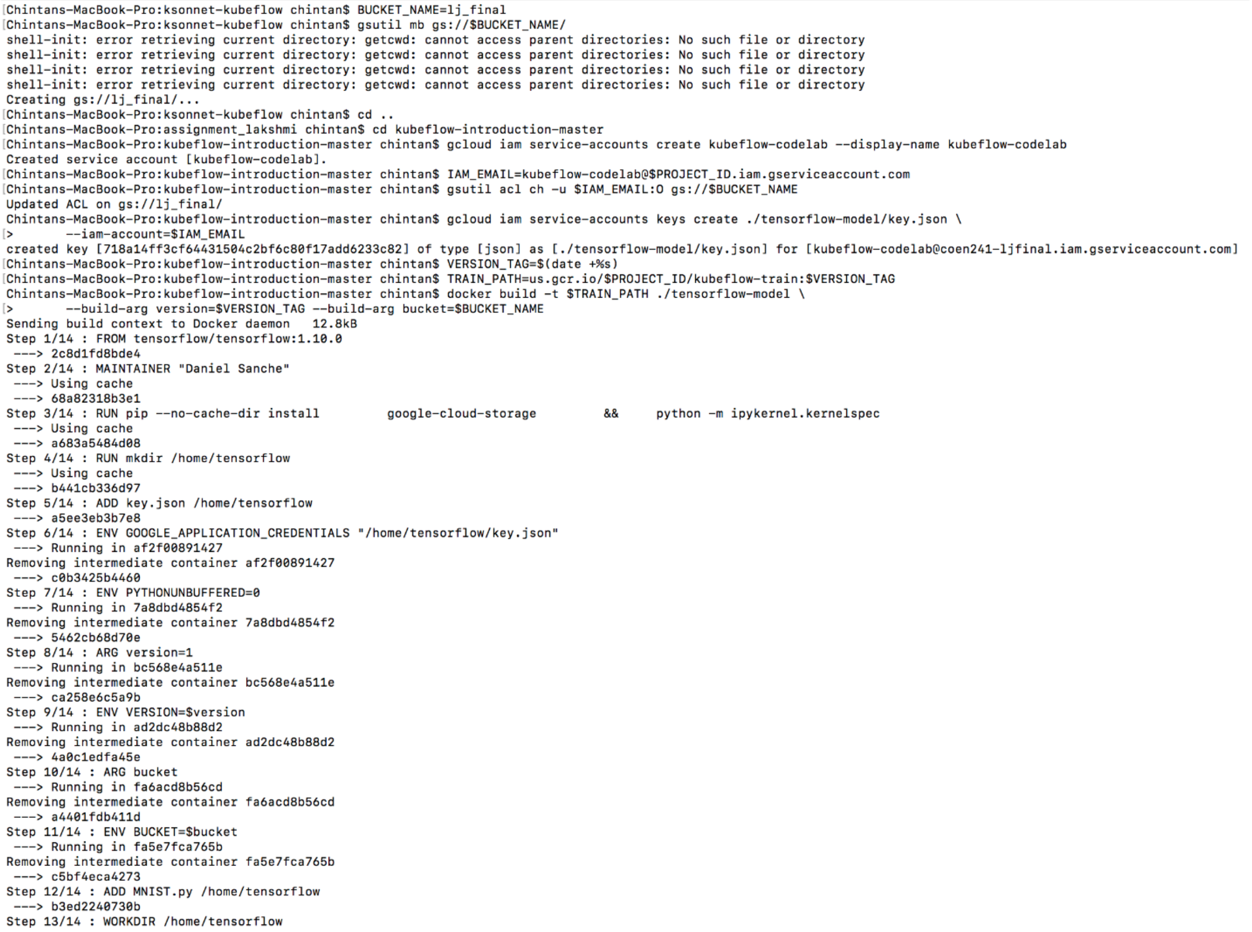
7. Query your cluster with kubectl and see a list of resources provisioned by kubeflow-core.



```
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ kubectl get all
 NAME                                      READY     STATUS              RESTARTS   AGE
 pod/ambassador-59cb5ccd89-fszjt           0/2       ContainerCreating   0          14s
 pod/ambassador-59cb5ccd89-gs9m7           0/2       ContainerCreating   0          14s
 pod/ambassador-59cb5ccd89-qmkjj           0/2       ContainerCreating   0          14s
 pod/tf-hub-0                              0/1       ContainerCreating   0          14s
 pod/tf-job-operator-d7585c9fc-7jklc       0/1       ContainerCreating   0          14s

 NAME                          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
 service/ambassador            ClusterIP   10.35.242.22    <none>        80/TCP     15s
 service/ambassador-admin      ClusterIP   10.35.255.144   <none>        8877/TCP   15s
 service/k8s-dashboard         ClusterIP   10.35.243.84    <none>        443/TCP    15s
 service/kubernetes            ClusterIP   10.35.240.1     <none>        443/TCP    8m
 service/tf-hub-0              ClusterIP   None            <none>        8000/TCP   17s
 service/tf-hub-lb             ClusterIP   10.35.241.27    <none>        80/TCP     17s

 NAME                                  DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
 deployment.apps/ambassador            3         3         3            0           14s
 deployment.apps/tf-job-operator       1         1         1            0           14s

 NAME                                            DESIRED   CURRENT   READY     AGE
 replicaset.apps/ambassador-59cb5ccd89           3         3         0         14s
 replicaset.apps/tf-job-operator-d7585c9fc       1         1         0         14s

 NAME                          DESIRED   CURRENT   AGE
 statefulset.apps/tf-hub       1         1         14s
```

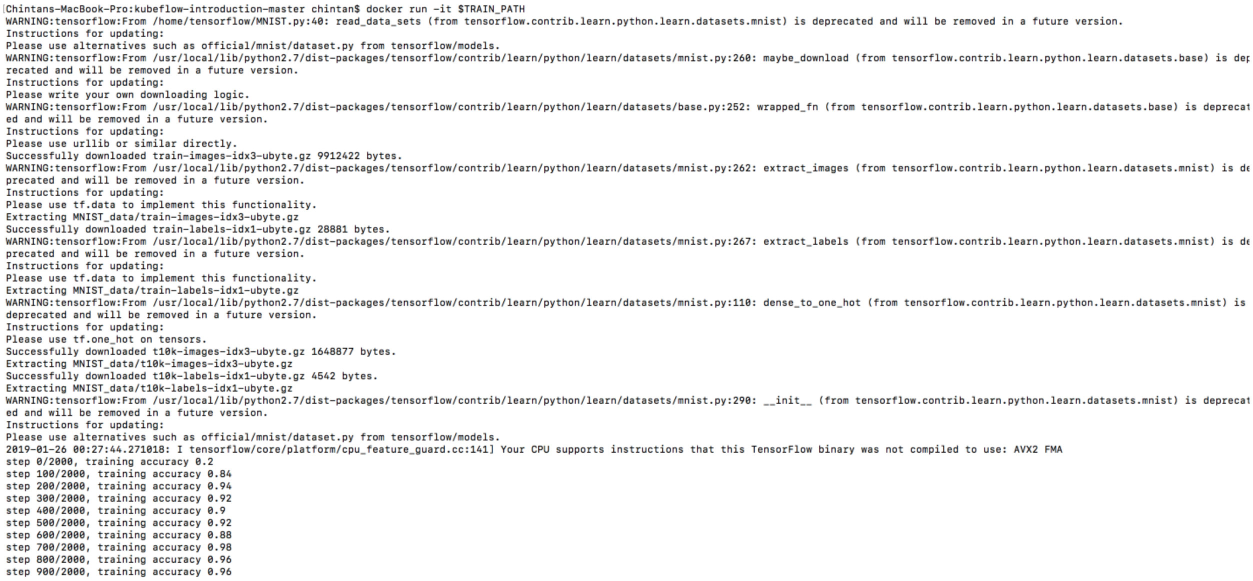
8. Create a storage bucket that will hold our trained model and a service account. Grant the service account permissions to read/write to the bucket.

Download the key file that lets us authenticate as the service account. Build our local project into a container.



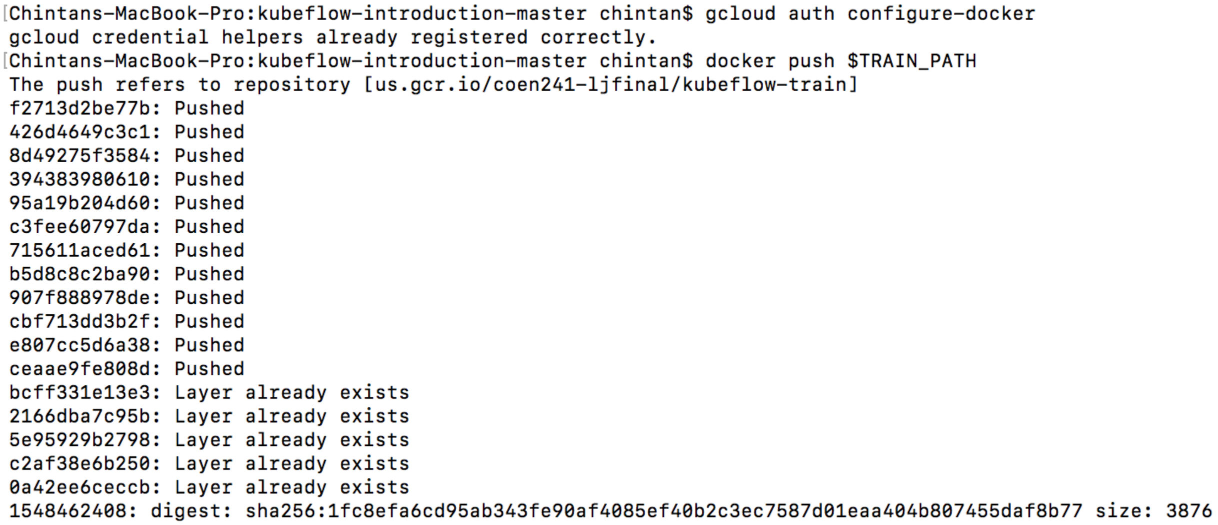
```
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ BUCKET_NAME=lj_final
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ gsutil mb gs://$BUCKET_NAME/
 shell-init: error retrieving current directory: getcwd: cannot access parent directories: No such file or directory
 shell-init: error retrieving current directory: getcwd: cannot access parent directories: No such file or directory
 shell-init: error retrieving current directory: getcwd: cannot access parent directories: No such file or directory
 shell-init: error retrieving current directory: getcwd: cannot access parent directories: No such file or directory
 Creating gs://lj_final/...
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ cd ..
[Chintans-MacBook-Pro:assignment_lakshmi chintan$ cd kubeflow-introduction-master
[Chintans-MacBook-Pro:kubeflow-introduction-master chintan$ gcloud iam service-accounts create kubeflow-codelab --display-name kubeflow-codelab
 Created service account [kubeflow-codelab].
[Chintans-MacBook-Pro:kubeflow-introduction-master chintan$ IAM_EMAIL=kubeflow-codelab@$PROJECT_ID.iam.gserviceaccount.com
[Chintans-MacBook-Pro:kubeflow-introduction-master chintan$ gsutil acl ch -u $IAM_EMAIL:O gs://$BUCKET_NAME
 Updated ACL on gs://lj_final/
 Chintans-MacBook-Pro:kubeflow-introduction-master chintan$ gcloud iam service-accounts keys create ./tensorflow-model/key.json \
[>       --iam-account=$IAM_EMAIL
 created key [718a14ff3cf64431504c2bf6c80f17add6233c82] of type [json] as [./tensorflow-model/key.json] for [kubeflow-codelab@coen241-ljfinal.iam.gserviceaccount.com]
[Chintans-MacBook-Pro:kubeflow-introduction-master chintan$ VERSION_TAG=$(date +%s)
[Chintans-MacBook-Pro:kubeflow-introduction-master chintan$ TRAIN_PATH=us.gcr.io/$PROJECT_ID/kubeflow-train:$VERSION_TAG
 Chintans-MacBook-Pro:kubeflow-introduction-master chintan$ docker build -t $TRAIN_PATH ./tensorflow-model \
[>       --build-arg version=$VERSION_TAG --build-arg bucket=$BUCKET_NAME
 Sending build context to Docker daemon   12.8kB
 Step 1/14 : FROM tensorflow/tensorflow:1.10.0
  ---> 2c8d1fd8bde4
 Step 2/14 : MAINTAINER "Daniel Sanche"
  ---> Using cache
  ---> 68a82318b3e1
 Step 3/14 : RUN pip --no-cache-dir install         google-cloud-storage         &&     python -m ipykernel.kernelspec
  ---> Using cache
  ---> a683a5484d08
 Step 4/14 : RUN mkdir /home/tensorflow
  ---> Using cache
  ---> b441cb336d97
 Step 5/14 : ADD key.json /home/tensorflow
  ---> a5ee3eb3b7e8
 Step 6/14 : ENV GOOGLE_APPLICATION_CREDENTIALS "/home/tensorflow/key.json"
  ---> Running in af2f00891427
 Removing intermediate container af2f00891427
  ---> c0b3425b4460
 Step 7/14 : ENV PYTHONUNBUFFERED=0
  ---> Running in 7a8dbd4854f2
 Removing intermediate container 7a8dbd4854f2
  ---> 5462cb68d70e
 Step 8/14 : ARG version=1
  ---> Running in bc568e4a511e
 Removing intermediate container bc568e4a511e
  ---> ca258e6c5a9b
 Step 9/14 : ENV VERSION=$version
  ---> Running in ad2dc48b88d2
 Removing intermediate container ad2dc48b88d2
  ---> 4a0c1edfa45e
 Step 10/14 : ARG bucket
  ---> Running in fa6acd8b56cd
 Removing intermediate container fa6acd8b56cd
  ---> a4401fdb411d
 Step 11/14 : ENV BUCKET=$bucket
  ---> Running in fa5e7fca765b
 Removing intermediate container fa5e7fca765b
  ---> c5bf4eca4273
 Step 12/14 : ADD MNIST.py /home/tensorflow
  ---> b3ed2240730b
 Step 13/14 : WORKDIR /home/tensorflow
```

9. Test it locally to make sure everything is working.



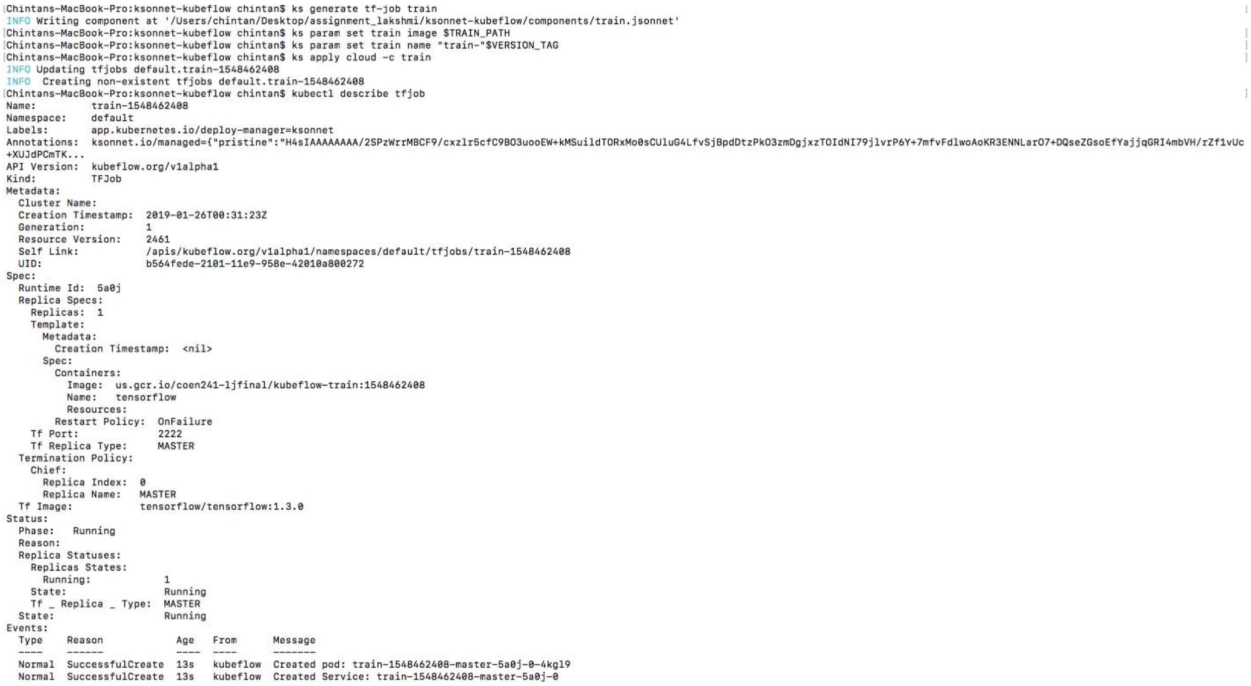
```
[Chintans-MacBook-Pro:kubeflow-introduction-master chintan$ docker run -it $TRAIN_PATH
 WARNING:tensorflow:From /home/tensorflow/MNIST.py:40: read_data_sets (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
 Instructions for updating:
 Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
 WARNING:tensorflow:From /usr/local/lib/python2.7/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:260: maybe_download (from tensorflow.contrib.learn.python.learn.datasets.base) is dep
 recated and will be removed in a future version.
 Instructions for updating:
 Please write your own downloading logic.
 WARNING:tensorflow:From /usr/local/lib/python2.7/dist-packages/tensorflow/contrib/learn/python/learn/datasets/base.py:252: wrapped_fn (from tensorflow.contrib.learn.python.learn.datasets.base) is deprecat
 ed and will be removed in a future version.
 Instructions for updating:
 Please use urllib or similar directly.
 Successfully downloaded train-images-idx3-ubyte.gz 9912422 bytes.
 WARNING:tensorflow:From /usr/local/lib/python2.7/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:262: extract_images (from tensorflow.contrib.learn.python.learn.datasets.mnist) is de
 precated and will be removed in a future version.
 Instructions for updating:
 Please use tf.data to implement this functionality.
 Extracting MNIST_data/train-images-idx3-ubyte.gz
 Successfully downloaded train-labels-idx1-ubyte.gz 28881 bytes.
 WARNING:tensorflow:From /usr/local/lib/python2.7/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:267: extract_labels (from tensorflow.contrib.learn.python.learn.datasets.mnist) is de
 precated and will be removed in a future version.
 Instructions for updating:
 Please use tf.data to implement this functionality.
 Extracting MNIST_data/train-labels-idx1-ubyte.gz
 WARNING:tensorflow:From /usr/local/lib/python2.7/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:110: dense_to_one_hot (from tensorflow.contrib.learn.python.learn.datasets.mnist) is
 deprecated and will be removed in a future version.
 Instructions for updating:
 Please use tf.one_hot on tensors.
 Successfully downloaded t10k-images-idx3-ubyte.gz 1648877 bytes.
 Extracting MNIST_data/t10k-images-idx3-ubyte.gz
 Successfully downloaded t10k-labels-idx1-ubyte.gz 4542 bytes.
 Extracting MNIST_data/t10k-labels-idx1-ubyte.gz
 WARNING:tensorflow:From /usr/local/lib/python2.7/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:290: __init__ (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecat
 ed and will be removed in a future version.
 Instructions for updating:
 Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
 2019-01-26 00:27:44.271018: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
 step 0/2000, training accuracy 0.2
 step 100/2000, training accuracy 0.84
 step 200/2000, training accuracy 0.94
 step 300/2000, training accuracy 0.92
 step 400/2000, training accuracy 0.9
 step 500/2000, training accuracy 0.92
 step 600/2000, training accuracy 0.88
 step 700/2000, training accuracy 0.98
 step 800/2000, training accuracy 0.96
 step 900/2000, training accuracy 0.96
```

10. Allow docker to access our GCR registry and push container to GCR.



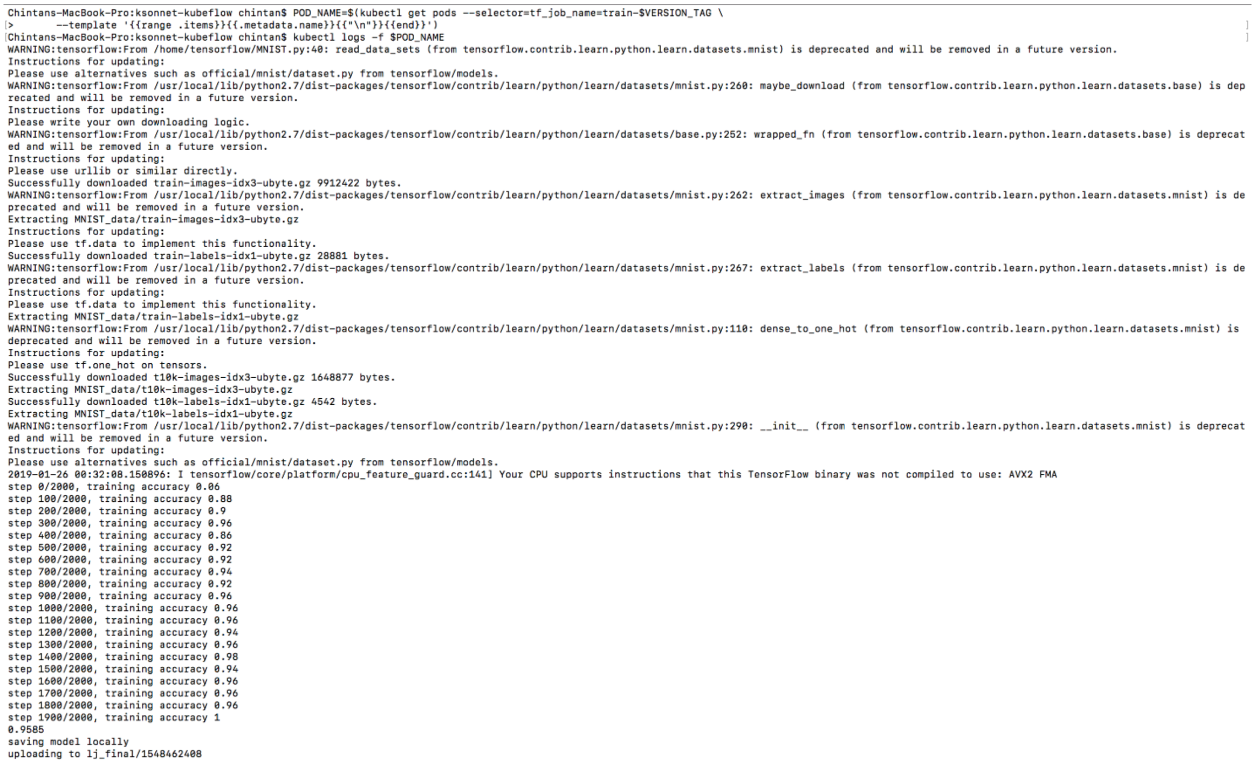
```
[Chintans-MacBook-Pro:kubeflow-introduction-master chintan$ gcloud auth configure-docker
gcloud credential helpers already registered correctly.
[Chintans-MacBook-Pro:kubeflow-introduction-master chintan$ docker push $TRAIN_PATH
The push refers to repository [us.gcr.io/coen241-ljfinal/kubeflow-train]
f2713d2be77b: Pushed
426d4649c3c1: Pushed
8d49275f3584: Pushed
394383980610: Pushed
95a19b204d60: Pushed
c3fee60797da: Pushed
715611aced61: Pushed
b5d8c8c2ba90: Pushed
907f888978de: Pushed
cbf713dd3b2f: Pushed
e807cc5d6a38: Pushed
ceaae9fe808d: Pushed
bcff331e13e3: Layer already exists
2166dba7c95b: Layer already exists
5e95929b2798: Layer already exists
c2af38e6b250: Layer already exists
0a42ee6ceccb: Layer already exists
1548462408: digest: sha256:1fc8efa6cd95ab343fe90af4085ef40b2c3ec7587d01eaa404b807455daf8b77 size: 3876
```

11. Generate component from prototype and set the parameters for this job.Apply the container to the cluster.



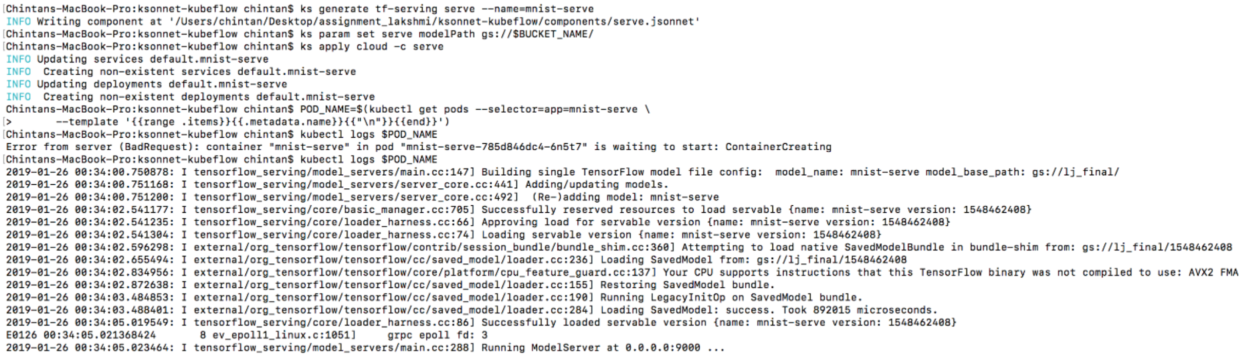
```
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ ks generate tf-job train
 INFO Writing component at '/Users/chintan/Desktop/assignment_lakshmi/ksonnet-kubeflow/components/train.jsonnet'
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ ks param set train image $TRAIN_PATH
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ ks param set train name "train-"$VERSION_TAG
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ ks apply cloud -c train
 INFO Updating tfjobs default.train-1548462408
 INFO  Creating non-existent tfjobs default.train-1548462408
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ kubectl describe tfjob
Name:         train-1548462408
Namespace:    default
Labels:       app.kubernetes.io/deploy-manager=ksonnet
Annotations:  ksonnet.io/managed={"pristine":"H4sIAAAAAAAA/2SPzWrrMBCF9/cxzlr5cfC9BO3uooEW+kMSuildTORxMo0sCUluG4LfvSjBpdDtzPkO3zmDgjxzTOIdNI79jlvrP6Y+7mfvFdlwoAoKR3ENNLarO7+DQseZGsoEfYajjqGRI4mbVH/rZf1vUc
+XUJdPCmTK...
API Version:  kubeflow.org/v1alpha1
Kind:         TFJob
Metadata:
  Cluster Name:
  Creation Timestamp:  2019-01-26T00:31:23Z
  Generation:          1
  Resource Version:    2461
  Self Link:           /apis/kubeflow.org/v1alpha1/namespaces/default/tfjobs/train-1548462408
  UID:                 b564fede-2101-11e9-958e-42010a800272
Spec:
  Runtime Id:  5a0j
  Replica Specs:
    Replicas:  1
    Template:
      Metadata:
        Creation Timestamp:  <nil>
      Spec:
        Containers:
          Image:  us.gcr.io/coen241-ljfinal/kubeflow-train:1548462408
          Name:   tensorflow
          Resources:
        Restart Policy:  OnFailure
    Tf Port:             2222
    Tf Replica Type:     MASTER
  Termination Policy:
    Chief:
      Replica Index:  0
      Replica Name:   MASTER
  Tf Image:           tensorflow/tensorflow:1.3.0
Status:
  Phase:   Running
  Reason:
  Replica Statuses:
    Replicas States:
      Running:            1
    State:                Running
    Tf _ Replica _ Type:  MASTER
  State:                  Running
Events:
  Type    Reason            Age   From      Message
  ----    ------            ----  ----      -------
  Normal  SuccessfulCreate  13s   kubeflow  Created pod: train-1548462408-master-5a0j-0-4kgl9
  Normal  SuccessfulCreate  13s   kubeflow  Created Service: train-1548462408-master-5a0j-0
```

12. Use kubectl to query some information about the job, including its current state and for more information you can retrieve the python logs from the pod that's running the container itself.



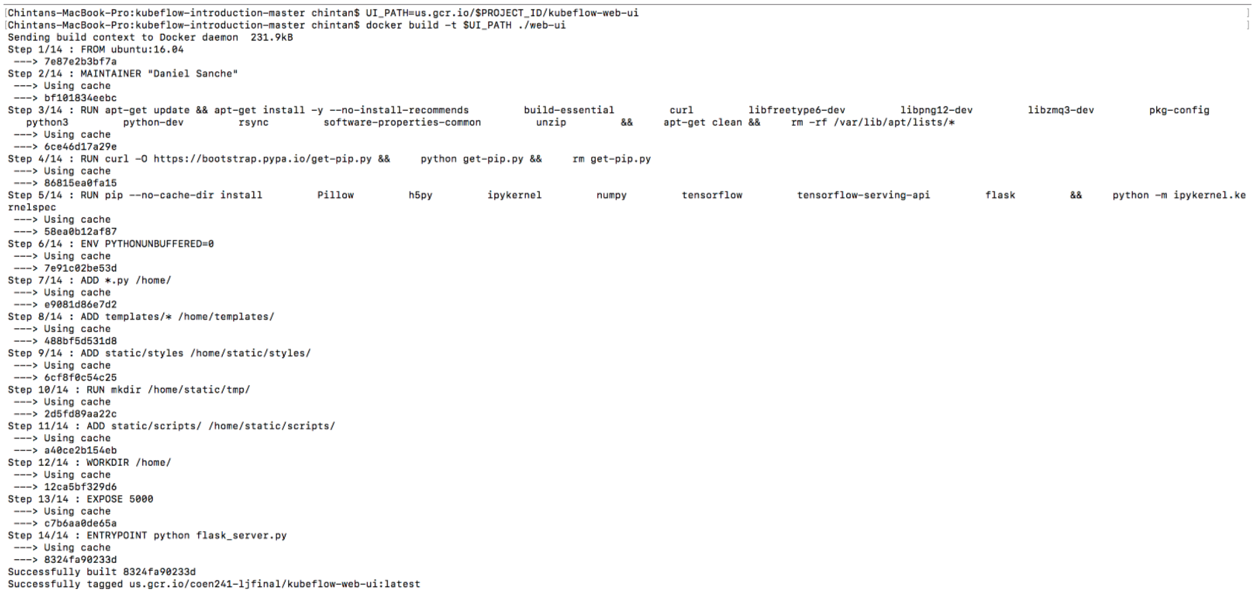
```
Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ POD_NAME=$(kubectl get pods --selector=tf_job_name=train-$VERSION_TAG \
>       --template '{{range .items}}{{.metadata.name}}{{"\n"}}{{end}}')
Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ kubectl logs -f $POD_NAME
WARNING:tensorflow:From /home/tensorflow/MNIST.py:40: read_data_sets (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
WARNING:tensorflow:From /usr/local/lib/python2.7/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:260: maybe_download (from tensorflow.contrib.learn.python.learn.datasets.base) is dep
recated and will be removed in a future version.
Instructions for updating:
Please write your own downloading logic.
WARNING:tensorflow:From /usr/local/lib/python2.7/dist-packages/tensorflow/contrib/learn/python/learn/datasets/base.py:252: wrapped_fn (from tensorflow.contrib.learn.python.learn.datasets.base) is deprecat
ed and will be removed in a future version.
Instructions for updating:
Please use urllib or similar directly.
Successfully downloaded train-images-idx3-ubyte.gz 9912422 bytes.
WARNING:tensorflow:From /usr/local/lib/python2.7/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:262: extract_images (from tensorflow.contrib.learn.python.learn.datasets.mnist) is de
precated and will be removed in a future version.
Extracting MNIST_data/train-images-idx3-ubyte.gz
Instructions for updating:
Please use tf.data to implement this functionality.
Successfully downloaded train-labels-idx1-ubyte.gz 28881 bytes.
WARNING:tensorflow:From /usr/local/lib/python2.7/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:267: extract_labels (from tensorflow.contrib.learn.python.learn.datasets.mnist) is de
precated and will be removed in a future version.
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting MNIST_data/train-labels-idx1-ubyte.gz
WARNING:tensorflow:From /usr/local/lib/python2.7/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:110: dense_to_one_hot (from tensorflow.contrib.learn.python.learn.datasets.mnist) is
deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.one_hot on tensors.
Successfully downloaded t10k-images-idx3-ubyte.gz 1648877 bytes.
Extracting MNIST_data/t10k-images-idx3-ubyte.gz
Successfully downloaded t10k-labels-idx1-ubyte.gz 4542 bytes.
Extracting MNIST_data/t10k-labels-idx1-ubyte.gz
WARNING:tensorflow:From /usr/local/lib/python2.7/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:290: __init__ (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecat
ed and will be removed in a future version.
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
2019-01-26 00:32:08.150896: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
step 0/2000, training accuracy 0.06
step 100/2000, training accuracy 0.88
step 200/2000, training accuracy 0.9
step 300/2000, training accuracy 0.96
step 400/2000, training accuracy 0.86
step 500/2000, training accuracy 0.92
step 600/2000, training accuracy 0.92
step 700/2000, training accuracy 0.94
step 800/2000, training accuracy 0.92
step 900/2000, training accuracy 0.96
step 1000/2000, training accuracy 0.96
step 1100/2000, training accuracy 0.96
step 1200/2000, training accuracy 0.94
step 1300/2000, training accuracy 0.96
step 1400/2000, training accuracy 0.98
step 1500/2000, training accuracy 0.94
step 1600/2000, training accuracy 0.96
step 1700/2000, training accuracy 0.96
step 1800/2000, training accuracy 0.96
step 1900/2000, training accuracy 1
0.9585
saving model locally
uploading to lj_final/1548462408
```

13. Create a ksonnet component from the prototype and set the parameters and apply to the cluster. Check the logs of the running server pod to ensure everything is working as expected.



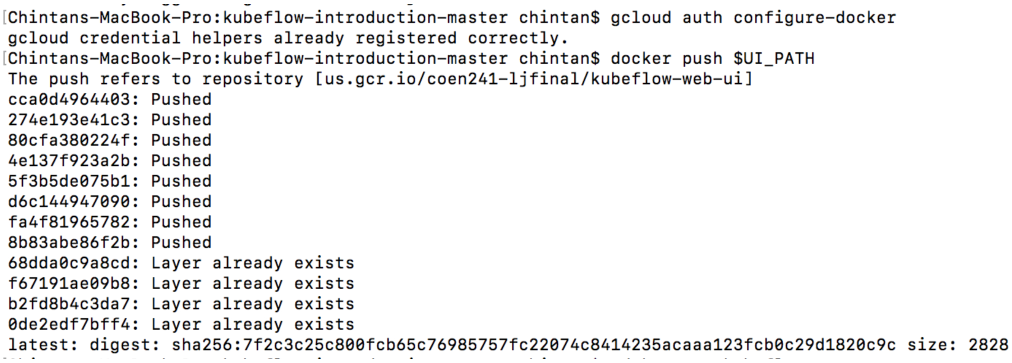
```
Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ ks generate tf-serving serve --name=mnist-serve
INFO Writing component at '/Users/chintan/Desktop/assignment_lakshmi/ksonnet-kubeflow/components/serve.jsonnet'
Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ ks param set serve modelPath gs://$BUCKET_NAME/
Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ ks apply cloud -c serve
INFO Updating services default.mnist-serve
INFO  Creating non-existent services default.mnist-serve
INFO Updating deployments default.mnist-serve
INFO  Creating non-existent deployments default.mnist-serve
Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ POD_NAME=$(kubectl get pods --selector=app=mnist-serve \
>       --template '{{range .items}}{{.metadata.name}}{{"\n"}}{{end}}')
Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ kubectl logs $POD_NAME
Error from server (BadRequest): container "mnist-serve" in pod "mnist-serve-785d846dc4-6n5t7" is waiting to start: ContainerCreating
Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ kubectl logs $POD_NAME
2019-01-26 00:34:00.750878: I tensorflow_serving/model_servers/main.cc:147] Building single TensorFlow model file config:  model_name: mnist-serve model_base_path: gs://lj_final/
2019-01-26 00:34:00.751168: I tensorflow_serving/model_servers/server_core.cc:441] Adding/updating models.
2019-01-26 00:34:00.751200: I tensorflow_serving/model_servers/server_core.cc:492]  (Re-)adding model: mnist-serve
2019-01-26 00:34:02.541177: I tensorflow_serving/core/basic_manager.cc:705] Successfully reserved resources to load servable {name: mnist-serve version: 1548462408}
2019-01-26 00:34:02.541235: I tensorflow_serving/core/loader_harness.cc:66] Approving load for servable version {name: mnist-serve version: 1548462408}
2019-01-26 00:34:02.541304: I tensorflow_serving/core/loader_harness.cc:74] Loading servable version {name: mnist-serve version: 1548462408}
2019-01-26 00:34:02.596298: I external/org_tensorflow/tensorflow/contrib/session_bundle/bundle_shim.cc:360] Attempting to load native SavedModelBundle in bundle-shim from: gs://lj_final/1548462408
2019-01-26 00:34:02.655494: I external/org_tensorflow/tensorflow/cc/saved_model/loader.cc:236] Loading SavedModel from: gs://lj_final/1548462408
2019-01-26 00:34:02.834956: I external/org_tensorflow/tensorflow/core/platform/cpu_feature_guard.cc:137] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
2019-01-26 00:34:02.872638: I external/org_tensorflow/tensorflow/cc/saved_model/loader.cc:155] Restoring SavedModel bundle.
2019-01-26 00:34:03.484853: I external/org_tensorflow/tensorflow/cc/saved_model/loader.cc:190] Running LegacyInitOp on SavedModel bundle.
2019-01-26 00:34:03.488401: I external/org_tensorflow/tensorflow/cc/saved_model/loader.cc:284] Loading SavedModel: success. Took 892015 microseconds.
2019-01-26 00:34:05.019549: I tensorflow_serving/core/loader_harness.cc:86] Successfully loaded servable version {name: mnist-serve version: 1548462408}
E0126 00:34:05.021368424       8 ev_epoll1_linux.c:1051]     grpc epoll fd: 3
2019-01-26 00:34:05.023464: I tensorflow_serving/model_servers/main.cc:288] Running ModelServer at 0.0.0.0:9000 ...
```

14. Set the path on GCR you want to push the image to and build the web-ui directory.



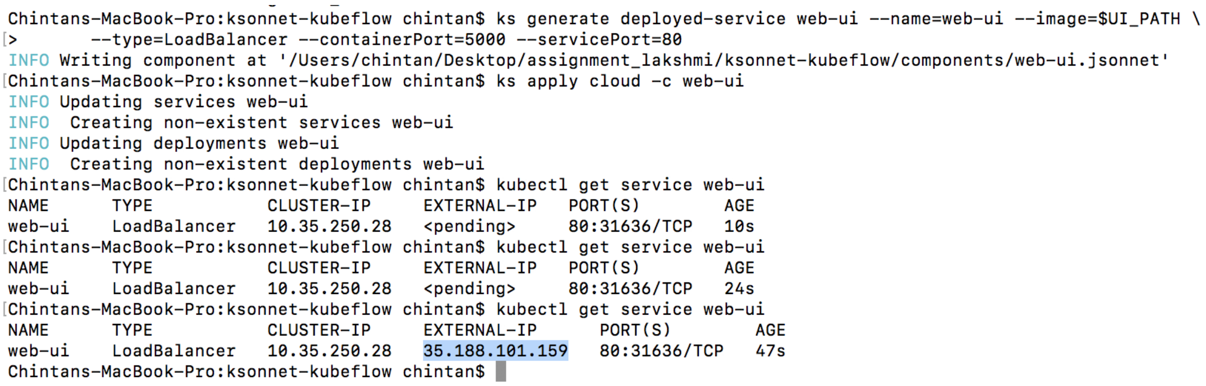
```
[Chintans-MacBook-Pro:kubeflow-introduction-master chintan$ UI_PATH=us.gcr.io/$PROJECT_ID/kubeflow-web-ui
[Chintans-MacBook-Pro:kubeflow-introduction-master chintan$ docker build -t $UI_PATH ./web-ui
Sending build context to Docker daemon  231.9kB
Step 1/14 : FROM ubuntu:16.04
 ---> 7e87e2b3bf7a
Step 2/14 : MAINTAINER "Daniel Sanche"
 ---> Using cache
 ---> bf101834eebc
Step 3/14 : RUN apt-get update && apt-get install -y --no-install-recommends         build-essential         curl         libfreetype6-dev         libpng12-dev         libzmq3-dev         pkg-config
   python3         python-dev         rsync         software-properties-common         unzip         &&     apt-get clean &&     rm -rf /var/lib/apt/lists/*
 ---> Using cache
 ---> 6ce46d17a29e
Step 4/14 : RUN curl -O https://bootstrap.pypa.io/get-pip.py &&     python get-pip.py &&     rm get-pip.py
 ---> Using cache
 ---> 86815ea0fa15
Step 5/14 : RUN pip --no-cache-dir install         Pillow         h5py         ipykernel         numpy         tensorflow         tensorflow-serving-api         flask         &&     python -m ipykernel.ke
rnelspec
 ---> Using cache
 ---> 58ea0b12af87
Step 6/14 : ENV PYTHONUNBUFFERED=0
 ---> Using cache
 ---> 7e91c02be53d
Step 7/14 : ADD *.py /home/
 ---> Using cache
 ---> e9081d86e7d2
Step 8/14 : ADD templates/* /home/templates/
 ---> Using cache
 ---> 488bf5d531d8
Step 9/14 : ADD static/styles /home/static/styles/
 ---> Using cache
 ---> 6cf8f0c54c25
Step 10/14 : RUN mkdir /home/static/tmp/
 ---> Using cache
 ---> 2d5fd89aa22c
Step 11/14 : ADD static/scripts/ /home/static/scripts/
 ---> Using cache
 ---> a40ce2b154eb
Step 12/14 : WORKDIR /home/
 ---> Using cache
 ---> 12ca5bf329d6
Step 13/14 : EXPOSE 5000
 ---> Using cache
 ---> c7b6aa0de65a
Step 14/14 : ENTRYPOINT python flask_server.py
 ---> Using cache
 ---> 8324fa90233d
Successfully built 8324fa90233d
Successfully tagged us.gcr.io/coen241-ljfinal/kubeflow-web-ui:latest
```

15. Allow docker to access our GCR registry and push the container to GCR.



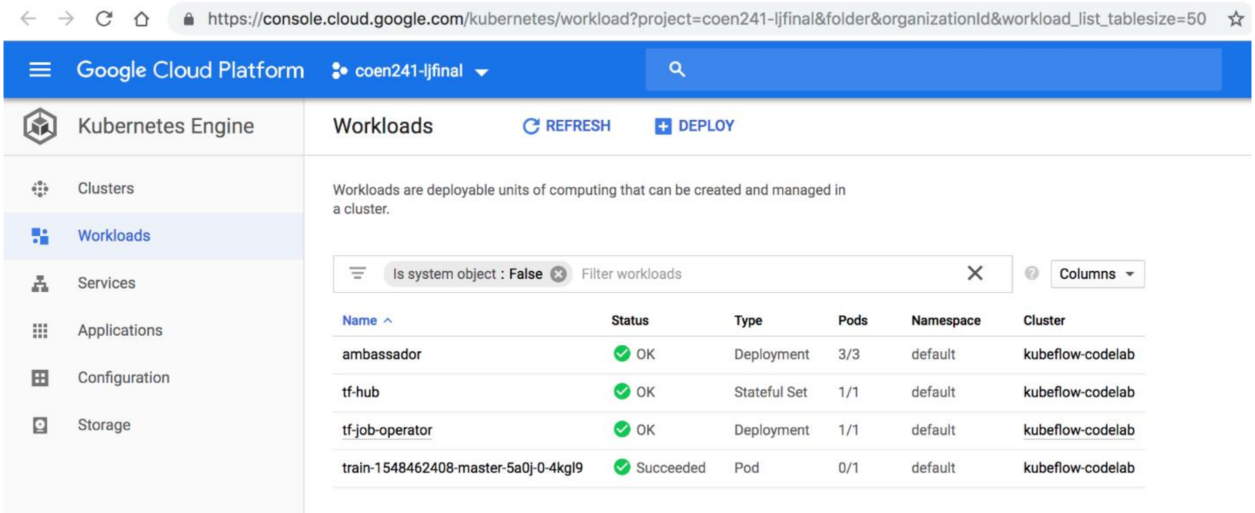
```
[Chintans-MacBook-Pro:kubeflow-introduction-master chintan$ gcloud auth configure-docker
gcloud credential helpers already registered correctly.
[Chintans-MacBook-Pro:kubeflow-introduction-master chintan$ docker push $UI_PATH
The push refers to repository [us.gcr.io/coen241-ljfinal/kubeflow-web-ui]
cca0d4964403: Pushed
274e193e41c3: Pushed
80cfa380224f: Pushed
4e137f923a2b: Pushed
5f3b5de075b1: Pushed
d6c144947090: Pushed
fa4f81965782: Pushed
8b83abe86f2b: Pushed
68dda0c9a8cd: Layer already exists
f67191ae09b8: Layer already exists
b2fd8b4c3da7: Layer already exists
0de2edf7bff4: Layer already exists
latest: digest: sha256:7f2c3c25c800fcb65c76985757fc22074c8414235acaaa123fcb0c29d1820c9c size: 2828
```

16. Create generate the component from its prototype and apply component to our cluster. Now, there should be a new web UI running in the cluster.



```
Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ ks generate deployed-service web-ui --name=web-ui --image=$UI_PATH \
[>       --type=LoadBalancer --containerPort=5000 --servicePort=80
INFO Writing component at '/Users/chintan/Desktop/assignment_lakshmi/ksonnet-kubeflow/components/web-ui.jsonnet'
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ ks apply cloud -c web-ui
INFO Updating services web-ui
INFO  Creating non-existent services web-ui
INFO Updating deployments web-ui
INFO  Creating non-existent deployments web-ui
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ kubectl get service web-ui
NAME      TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
web-ui    LoadBalancer   10.35.250.28   <pending>     80:31636/TCP   10s
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ kubectl get service web-ui
NAME      TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
web-ui    LoadBalancer   10.35.250.28   <pending>     80:31636/TCP   24s
[Chintans-MacBook-Pro:ksonnet-kubeflow chintan$ kubectl get service web-ui
NAME      TYPE           CLUSTER-IP     EXTERNAL-IP      PORT(S)        AGE
web-ui    LoadBalancer   10.35.250.28   35.188.101.159   80:31636/TCP   47s
Chintans-MacBook-Pro:ksonnet-kubeflow chintan$
```

17. When training is complete, you should see the model data pushed into your GCS bucket, tagged with the same version number as the container that generated it.

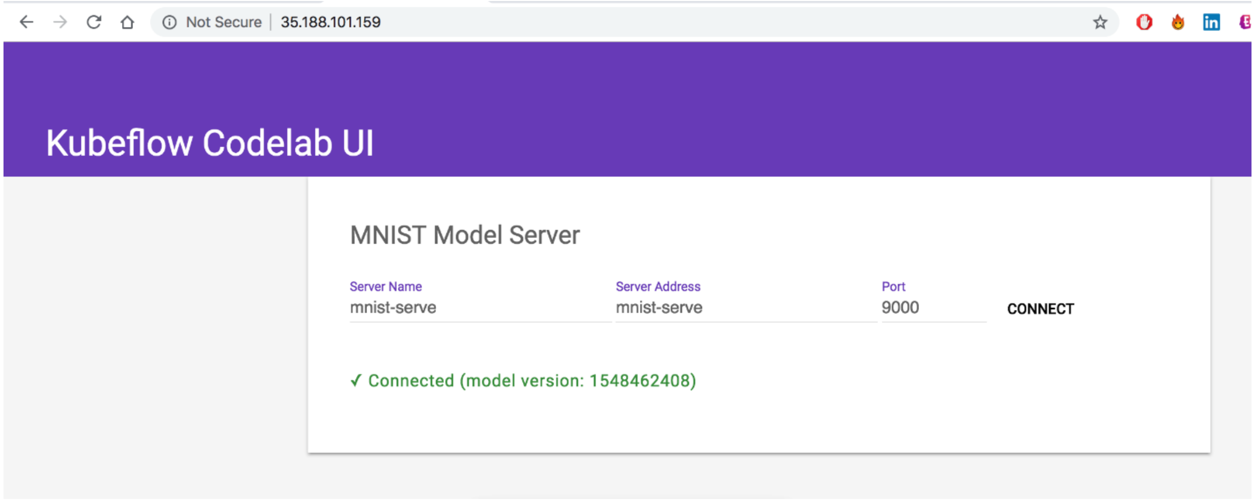


Workloads

Workloads are deployable units of computing that can be created and managed in a cluster.

Name	Status	Type	Pods	Namespace	Cluster
ambassador	✓ OK	Deployment	3/3	default	kubeflow-codelab
tf-hub	✓ OK	Stateful Set	1/1	default	kubeflow-codelab
tf-job-operator	✓ OK	Deployment	1/1	default	kubeflow-codelab
train-1548462408-master-5a0j-0-4kgl9	✓ Succeeded	Pod	0/1	default	kubeflow-codelab

18. New web UI running in the cluster accessed through web browser with external IP address of the service.

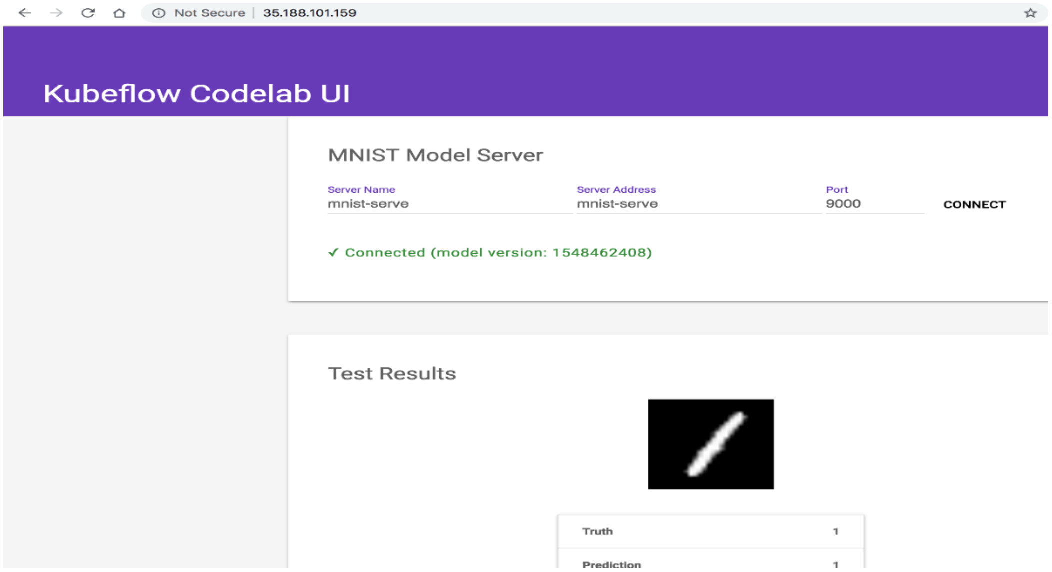


Kubeflow Codelab UI

MNIST Model Server

Server Name: mnist-serve
Server Address: mnist-serve
Port: 9000

✓ Connected (model version: 1548462408)



MNIST Model Server

Server Name: mnist-serve

Server Address: mnist-serve

Port: 9000

✓ Connected (model version: 1548462408)

Test Results

Truth	1
Prediction	1

Answer the following questions:

1. What are the advantages of using Kubeflow for training?

- Easily develop, train, and deploy machine learning models in scalable and flexible way.
- It supports distributive training as well as training on GPUs. With multiple frameworks being supported (Tensorflow, PyTorch and Mxnet), writing a distributed training or serving application (TFServing or Seldon) becomes a lot easier.
- Kubeflow's TFJob makes it easy to run distributed TensorFlow training jobs on Kubernetes.

2. What benefits does Kubernetes provide as opposed to managing VMs using separate scripts?

Kubernetes are a type of virtualization that let you run an application and its dependencies in an isolated process. It doesn’t have its own OS and It shares the kernel with not only the host but also the neighboring containers. So, it let us run a single application in a lightweight fashion and removes the requirement of having a separate VM for each application. Kubernetes are less for developers to have to manage or even know about. Everything in kubernetes is a declarative configuration object that represents the desired state of the system. Kubernetes achieves scalability by favoring decoupled architectures.