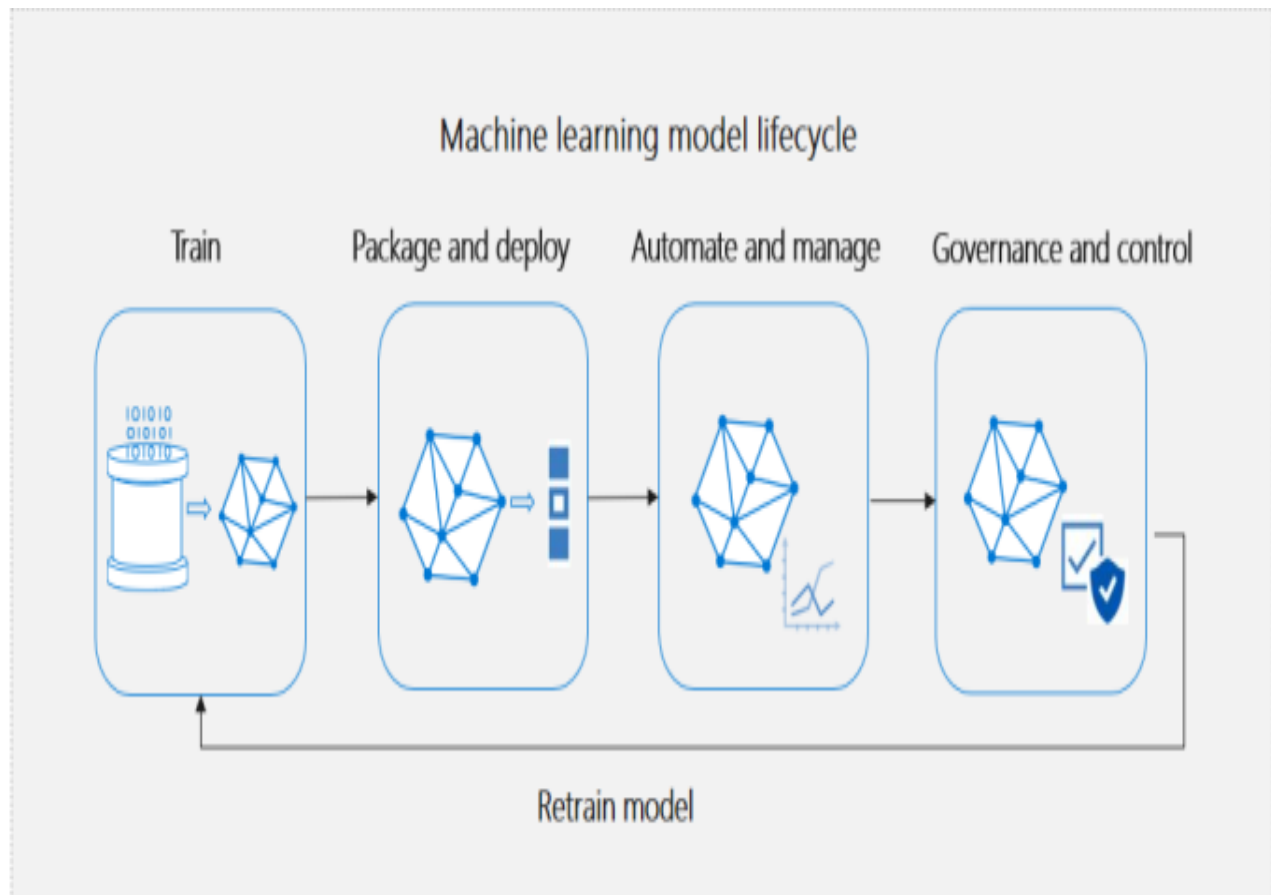



MLOPS



TASK DESCRIPTION

1. Create container image that's has Python3 and Keras or numpy installed using dockerfile
2. When we launch this image, it should automatically starts train the model in the container.
3. Create a job chain of job1, job2, job3, job4 and job5 using build pipeline plugin in Jenkins
4. Job1 : Pull the Github repo automatically when some developers push repo to Github.
5. Job2 : By looking at the code or program file, Jenkins should automatically start the respective machine learning software installed interpreter install image container to deploy code and start training(eg. If code uses CNN, then Jenkins should start the container that has already installed all the softwares required for the cnn processing).
6. Job3 : Train your model and predict accuracy or metrics.
7. Job4 : if metrics accuracy is less than 80% , then tweak the machine learning model architecture.
8. Job5: Retrain the model or notify that the best model is being created
9. Create One extra job job6 for monitor : If container where app is running. fails due to any reason then this job should automatically start the container again from where the last trained model left

STEP ->> By this Dockerfile I created the docker Image , that has all the required modules for running my model

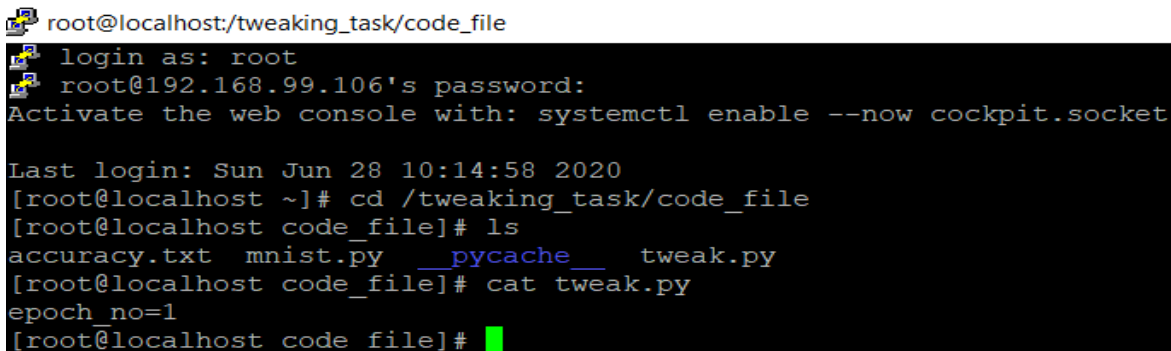
A screenshot of a text editor window titled 'Dockerfile' with a subtitle '/tensor'. The editor shows a Dockerfile with the following content:

```
FROM centos:7
RUN yum install python36 -y
RUN pip3 install --upgrade pip
RUN pip3 install tensorflow
RUN pip install keras
RUN pip3 install pillow

VOLUME /mlops
COPY . /mlops
WORKDIR /mlops
CMD ["python", "./mnist.py"]
```

The editor interface includes a top bar with 'Activities', 'Text Editor', and a clock showing 'Sun 12:54'. Below the top bar are buttons for 'Open', 'Save', and a menu icon. The status bar at the bottom indicates 'Plain Text', 'Tab Width: 8', 'Ln 6, Col 8', and 'INS'.

Here is the directory where I stored my code form github in my RHEL8 VM

A terminal window screenshot showing a user logging in as root on a system with IP 192.168.99.106. The user navigates to the directory /tweaking_task/code_file and lists its contents. The output shows files: accuracy.txt, mnist.py, __pycache__, and tweak.py. The user then cat's the tweak.py file, which contains the line epoch_no=1.

```
root@localhost:/tweaking_task/code_file
login as: root
root@192.168.99.106's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Sun Jun 28 10:14:58 2020
[root@localhost ~]# cd /tweaking_task/code_file
[root@localhost code_file]# ls
accuracy.txt  mnist.py  __pycache__  tweak.py
[root@localhost code_file]# cat tweak.py
epoch_no=1
[root@localhost code_file]#
```

JOB 1 ->> This Job is pulling the model code from my github repository

pull_model_code

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Description: This Job will pull the my model code from github

[Plain text] [Preview](#)

☐ Commit agent's Docker container
☐ Define a Docker template
☐ Discard old builds
☐ GitHub project
☐ This build requires lockable resources
☐ This project is parameterized
☐ Throttle builds
☐ Disable this project
☐ Execute concurrent builds if necessary
☐ Restrict where this project can be run

[Advanced...](#)

Source Code Management

☐ None
☒ Git

Repositories

Repository URL:

Credentials: [Add](#)

[Advanced...](#)
[Add Repository](#)

[Discard old builds](#)

[Save](#) [Apply](#)

Branch Specifier (blank for 'any'):

General Source Code Management **Build Triggers** Build Environment Build Post-build Actions

Schedule

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour. Would last have run at Sunday, June 28, 2020 1:07:43 PM IST; would next run at Sunday, June 28, 2020 1:07:43 PM IST.

☐ Ignore post-commit hooks

Build Environment

☐ Delete workspace before build starts
☐ Use secret text(s) or file(s)
☐ Abort the build if it's stuck
☐ Add timestamps to the Console Output
☐ Inspect build log for published Gradle build scans
☐ With Ant

Build

Execute shell

Command:

[See the list of available environment variables](#)

[Advanced...](#)

[Add build step](#)

Post-build Actions

Build other projects

[Discards to build](#) [Trigger the model](#)

[Save](#) [Apply](#)

Trigger only if build is stable

JOB 2 ->> This Job will train the model and give the accuracy and but I have created this Job only for mnist code , you can use same code for multiple type of Code , just by adding few more lines in build shell

Train_the_model >

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Description This Job will train and run the model

[Plain text] [Preview](#)

☐ Commit agent's Docker container

☐ Define a Docker template

☐ Discard old builds

☐ GitHub project

☐ This build requires lockable resources

☐ This project is parameterized

☐ Throttle builds

☐ Disable this project

☐ Execute concurrent builds if necessary

☐ Restrict where this project can be run

[Advanced...](#)

Source Code Management

☒ None

☐ Git

Build Triggers

☐ Trigger builds remotely (e.g., from scripts)

☒ Build after other projects are built

Projects to watch

☐ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

[Save](#) [Apply](#)

Train_the_model >

General Source Code Management Build Triggers Build Environment **Build** Post-build Actions

☐ Inspect build log for published Gradle build scans

☐ With Ant

Build

Execute shell

Command

```
sudo cat /tweaking_task/code_file/mnist.py | grep keras
if sudo docker ps -a | grep keras_OS
then
sudo docker rm -f keras_OS
else
sudo docker run --name keras_OS -v /tweaking_task/code_file/:/mlops tensorflow:v1
fi
```

[See the list of available environment variables](#)

[Advanced...](#)

[Add build step](#)

Post-build Actions

Build other projects

Projects to build

☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

E-mail Notification

Recipients

Whitespace-separated list of recipient addresses. May reference build parameters like \$PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable.

☒ Send e-mail for every unstable build

☐ Send separate e-mails to individuals who broke the build

[Save](#) [Apply](#)

Here is Output of the JOB 2 and I have got 89% accuracy in one epoch

Train_the_model #2

Console Output

```
Started by upstream project "pull_model_code" build number 2
originally caused by:
  Started by user admin
Started by upstream project "pull_model_code" build number 2
originally caused by:
  Started by user admin
Running as SYSTEM
Building on master in workspace /var/lib/jenkins/workspace/Train_the_model
[Train_the_model] $ /bin/sh -xe /tmp/jenkins2217219505617638202.sh
+ sudo cat /tweaking_task/code_file/mnist.py
+ grep keras
from keras.datasets import mnist
from keras.utils.np_utils import to_categorical
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import RMSprop
+ sudo docker ps -a
+ grep keras_OS
+ sudo docker run --name keras_OS -v /tweaking_task/code_file:/mlops tensorflow:v1
Downloading data from https://s3.amazonaws.com/img-datasets/mnist.npz
 8192/11490434 [.....] - ETA: 8:03
 40960/11490434 [.....] - ETA: 3:12
 73728/11490434 [.....] - ETA: 2:46
106496/11490434 [.....] - ETA: 2:34
163840/11490434 [.....] - ETA: 2:05
180224/11490434 [.....] - ETA: 2:16
196608/11490434 [.....] - ETA: 2:23
212992/11490434 [.....] - ETA: 2:31
229376/11490434 [.....] - ETA: 3:26
245760/11490434 [.....] - ETA: 4:15
262144/11490434 [.....] - ETA: 4:29
278528/11490434 [.....] - ETA: 4:51
294912/11490434 [.....] - ETA: 4:37
303104/11490434 [.....] - ETA: 4:51
319488/11490434 [.....] - ETA: 4:48
335872/11490434 [.....] - ETA: 4:45
352256/11490434 [.....] - ETA: 4:42
368640/11490434 [.....] - ETA: 4:32
385024/11490434 [>.....] - ETA: 4:28
401408/11490434 [>.....] - ETA: 4:20
417792/11490434 [>.....] - ETA: 4:15
442368/11490434 [>.....] - ETA: 4:09
```

Train_the_model #2

```
8585216/11490434 [=====] - ETA: 16s
8740864/11490434 [=====] - ETA: 15s
8792080/11490434 [=====] - ETA: 15s
8863744/11490434 [=====] - ETA: 14s
8953856/11490434 [=====] - ETA: 13s
9019392/11490434 [=====] - ETA: 13s
9437184/11490434 [=====] - ETA: 10s
9510912/11490434 [=====] - ETA: 10s
9576448/11490434 [=====] - ETA: 9s
9609216/11490434 [=====] - ETA: 9s
9707520/11490434 [=====] - ETA: 9s
9822208/11490434 [=====] - ETA: 8s
9912320/11490434 [=====] - ETA: 7s
10010624/11490434 [=====] - ETA: 7s
10100736/11490434 [=====] - ETA: 6s
10223616/11490434 [=====] - ETA: 6s
10346496/11490434 [=====] - ETA: 5s
10436608/11490434 [=====] - ETA: 5s
10567680/11490434 [=====] - ETA: 4s
10674176/11490434 [=====] - ETA: 3s
10747904/11490434 [=====] - ETA: 3s
10780672/11490434 [=====] - ETA: 3s
10960896/11490434 [=====] - ETA: 2s
11091968/11490434 [=====] - ETA: 1s
11264000/11490434 [=====] - ETA: 1s
11403264/11490434 [=====] - ETA: 0s
11493376/11490434 [=====] - 51s 4us/step
2020-06-28 07:31:33.050934: W tensorflow/stream_executor/cuda/cuda_driver.cc:313] failed call to cuInit: UNK
2020-06-28 07:31:33.051377: E tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not
2020-06-28 07:31:33.051665: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructio
2020-06-28 07:31:34.050668: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7fa2f400b20 ini
2020-06-28 07:31:34.052655: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): H
Epoch 1/1
 32/60000 [.....] - ETA: 44:43 - loss: 142.5349 - accuracy: 0.0000e+00
160/60000 [.....] - ETA: 9:18 - loss: 209.3978 - accuracy: 0.1937
288/60000 [.....] - ETA: 5:20 - loss: 130.9213 - accuracy: 0.3750
448/60000 [.....] - ETA: 3:33 - loss: 88.5913 - accuracy: 0.4821
608/60000 [.....] - ETA: 2:41 - loss: 67.8683 - accuracy: 0.5378
768/60000 [.....] - ETA: 2:11 - loss: 55.4421 - accuracy: 0.5872
896/60000 [.....] - ETA: 1:56 - loss: 49.5984 - accuracy: 0.5960
1056/60000 [.....] - ETA: 1:41 - loss: 43.2604 - accuracy: 0.6241
1216/60000 [.....] - ETA: 1:30 - loss: 38.8343 - accuracy: 0.6390
1408/60000 [.....] - ETA: 1:20 - loss: 35.1918 - accuracy: 0.6420
1600/60000 [.....] - ETA: 1:12 - loss: 31.6080 - accuracy: 0.6612
1792/60000 [.....] - ETA: 1:06 - loss: 28.6405 - accuracy: 0.6797
1984/60000 [.....] - ETA: 1:01 - loss: 26.5519 - accuracy: 0.6900
2176/60000 [>.....] - ETA: 57s - loss: 24.6479 - accuracy: 0.7031
2368/60000 [>.....] - ETA: 53s - loss: 23.2681 - accuracy: 0.7090
```

```
57888/60000 [=====>..] - ETA: 0s - loss: 2.0848 - accuracy: 0.1
58080/60000 [=====>.] - ETA: 0s - loss: 2.0810 - accuracy: 0.1
58272/60000 [=====>.] - ETA: 0s - loss: 2.0746 - accuracy: 0.1
58464/60000 [=====>.] - ETA: 0s - loss: 2.0694 - accuracy: 0.1
58656/60000 [=====>.] - ETA: 0s - loss: 2.0641 - accuracy: 0.1
58848/60000 [=====>.] - ETA: 0s - loss: 2.0584 - accuracy: 0.1
59040/60000 [=====>.] - ETA: 0s - loss: 2.0548 - accuracy: 0.1
59232/60000 [=====>.] - ETA: 0s - loss: 2.0494 - accuracy: 0.1
59264/60000 [=====>.] - ETA: 0s - loss: 2.0486 - accuracy: 0.1
59296/60000 [=====>.] - ETA: 0s - loss: 2.0479 - accuracy: 0.1
59328/60000 [=====>.] - ETA: 0s - loss: 2.0470 - accuracy: 0.1
59424/60000 [=====>.] - ETA: 0s - loss: 2.0447 - accuracy: 0.1
59552/60000 [=====>.] - ETA: 0s - loss: 2.0410 - accuracy: 0.1
59680/60000 [=====>.] - ETA: 0s - loss: 2.0386 - accuracy: 0.1
59776/60000 [=====>.] - ETA: 0s - loss: 2.0366 - accuracy: 0.1
59872/60000 [=====>.] - ETA: 0s - loss: 2.0340 - accuracy: 0.1
60000/60000 [=====] - 24s 394us/step - loss: 2.0311 - accuracy: 0.1

    32/10000 [.....] - ETA: 39s
   480/10000 [>.....] - ETA: 3s
   960/10000 [= >.....] - ETA: 2s
  1408/10000 [===>.....] - ETA: 1s
  1984/10000 [====>.....] - ETA: 1s
  2336/10000 [=====>.....] - ETA: 1s
  2656/10000 [=====>.....] - ETA: 1s
  3232/10000 [=====>.....] - ETA: 1s
  3808/10000 [=====>.....] - ETA: 0s
  4320/10000 [=====>.....] - ETA: 0s
  4480/10000 [=====>.....] - ETA: 0s
  4672/10000 [=====>.....] - ETA: 0s
  5280/10000 [=====>.....] - ETA: 0s
  5728/10000 [=====>.....] - ETA: 0s
  5952/10000 [=====>.....] - ETA: 0s
  6176/10000 [=====>.....] - ETA: 0s
  6656/10000 [=====>.....] - ETA: 0s
  7168/10000 [=====>.....] - ETA: 0s
  7680/10000 [=====>.....] - ETA: 0s
  8032/10000 [=====>.....] - ETA: 0s
  8512/10000 [=====>.....] - ETA: 0s
  9088/10000 [=====>.....] - ETA: 0s
  9632/10000 [=====>.....] - ETA: 0s
 10000/10000 [=====] - 2s 154us/step
Using TensorFlow backend.
acc=89.73%
Triggering a new build of accuracy\_checking
Triggering a new build of accuracy\_checking
Finished: SUCCESS
```

JOB 3 ->> This checks the accuracy the accuracy of model if accuracy is greater than 98% than it exxcute exit 1 else exit 0

accuracy_checking >

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Description: This will check the accuracy if accuracy is greater than 98% then will run next Job , if less then fail the build

[Plain text] [Preview](#)

- ☐ Commit agent's Docker container
- ☐ Define a Docker template
- ☐ Discard old builds
- ☐ GitHub project
- ☐ This build requires lockable resources
- ☐ This project is parameterized
- ☐ Throttle builds
- ☐ Disable this project
- ☐ Execute concurrent builds if necessary
- ☐ Restrict where this project can be run

[Advanced...](#)

Source Code Management

☒ None

☐ Git

Build Triggers

☐ Trigger builds remotely (e.g., from scripts)

☒ Build after other projects are built

Projects to watch:

☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

[Save](#) [Apply](#)

accuracy_checking >

General Source Code Management Build Triggers Build Environment **Build** Post-build Actions

Build

Execute shell

Command:

```
if sudo cat /tweaking_task/code_file/accuracy.txt | grep false
then
  exit 0
else
  exit 1
fi
```

See [the list of available environment variables](#)

[Advanced...](#)

[Add build step](#)

Post-build Actions

Build other projects

Projects to build:

☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

E-mail Notification

Recipients:

Whitespace-separated list of recipient addresses. May reference build parameters like \$PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable.

☒ Send e-mail for every unstable build

☐ Send separate e-mails to individuals who broke the build

Editable Email Notification

Disable Extended Email Publisher ☐

Allows the user to disable the publisher, while maintaining the settings

[Save](#) [Apply](#)

JOB 4 ->> This Job wil tweak the no. of epochs of model

tweaking_model

General

Source Code Management

Build Triggers

Build Environment

Build

Post-build Actions

Description

This Job will tweak the model

[Plain text] [Preview](#)

☐ Commit agent's Docker container

☐ Define a Docker template

☐ Discard old builds

☐ GitHub project

☐ This build requires lockable resources

☐ This project is parameterized

☐ Throttle builds

☐ Disable this project

☐ Execute concurrent builds if necessary

☒ Restrict where this project can be run

Label Expression

redhat_node

Label redhat_node is serviced by 1 node. Permissions or other restrictions provided by plugins may prevent this job from running on those nodes.

Advanced...

Source Code Management

☒ None

☐ Git

Build Triggers

☐ Trigger builds remotely (e.g., from scripts)

☒ Build after other projects are built

[Propagate to watch...](#)

accuracy_checking

Save

Apply

Trigger only if build is stable

tweaking_model

General

Source Code Management

Build Triggers

Build Environment

Build

Post-build Actions

Build Environment

☐ Delete workspace before build starts

☐ Use secret text(s) or file(s)

☐ Abort the build if it's stuck

☐ Add timestamps to the Console Output

☐ Inspect build log for published Gradle build scans

☐ With Ant

Build

Execute shell

Command

```
if sudo cat /tweaking_task/code_file/tweak.py | grep epoch_no=1
then
sed -i -e 's/epoch_no=1/epoch_no=10/' /tweaking_task/code_file/tweak.py
elif sudo cat /tweaking_task/code_file/tweak.py | grep epoch_no=10
then
sed -i -e 's/epoch_no=10/epoch_no=30/' /tweaking_task/code_file/tweak.py
elif sudo cat /tweaking_task/code_file/tweak.py | grep epoch_no=30
then
sed -i -e 's/epoch_no=30/epoch_no=50/' /tweaking_task/code_file/tweak.py
else
echo "Your model is trained successfully"
fi
```

See [the list of available environment variables](#)

Advanced...

Add build step

Post-build Actions

Build other projects

Projects to build

retraining_the_model,

Save

Apply

Trigger only if build is stable

Trigger even if the build is unstable

Here is the Output of the JOB 3

tweaking_model > #12

Console Output

```
Started by upstream project "accuracy\_checking" build number 3
originally caused by:
  Started by upstream project "Train\_the\_model" build number 2
  originally caused by:
    Started by upstream project "pull\_model\_code" build number 2
    originally caused by:
      Started by user admin
      Started by upstream project "pull\_model\_code" build number 2
      Started by upstream project "Train\_the\_model" build number 2
      Started by upstream project "accuracy\_checking" build number 3
      originally caused by:
        Started by upstream project "Train\_the\_model" build number 2
        originally caused by:
          Started by upstream project "pull\_model\_code" build number 2
          originally caused by:
            Started by user admin
            Started by upstream project "pull\_model\_code" build number 2
            Started by upstream project "Train\_the\_model" build number 2
Running as SYSTEM
Building remotely on redhat\_node (redhat) in workspace /mlops/workspace/tweaking_model
[tweaking_model] $ /bin/sh -xe /tmp/jenkins7197480978607770608.sh
+ grep epoch_no=1
+ sudo cat /tweaking_task/code_file/tweak.py
epoch_no=1
+ sed -i -e s/epoch_no=1/epoch_no=10/ /tweaking_task/code_file/tweak.py
Triggering a new build of retraining\_the\_model
Finished: SUCCESS
```

JOB 5 ->> This Job will retrain the model , after increasing the no. of epochs

retraining_the_model

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Description: This Job will retrain the model

[Plain text] [Preview](#)

- ☐ Commit agent's Docker container
- ☐ Define a Docker template
- ☐ Discard old builds
- ☐ GitHub project
- ☐ This build requires lockable resources
- ☐ This project is parameterized
- ☐ Throttle builds
- ☐ Disable this project
- ☐ Execute concurrent builds if necessary
- ☐ Restrict where this project can be run

[Advanced...](#)

Source Code Management

☒ None

☐ Git

Build Triggers

☐ Trigger builds remotely (e.g., from scripts)

☒ Build after other projects are built

Projects to watch:

☐ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

[Save](#) [Apply](#)

Build Environment

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s)
- ☐ Abort the build if it's stuck
- ☐ Add timestamps to the Console Output
- ☐ Inspect build log for published Gradle build scans
- ☐ With Ant

Build

Execute shell

Command:

```
sudo cat /tweaking_task/code_file/mnist.py | grep keras
if sudo docker ps -a | grep retrain_OS
then
sudo docker rm -f retrain_OS
else
sudo docker run --name retrain_OS -v /tweaking_task/code_file:/mlops tensorflow:v1
fi
```

[See the list of available environment variables](#)

[Advanced...](#)

[Add build step](#)

Post-build Actions

Build other projects

Projects to build:

☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

[Save](#) [Apply](#)

Here is the Output after retraining the model , but I have got approx 92% accuracy , you can wait for more time to get better accuracy

```
Epoch 9/10
32/60000 [.....] - ETA: 54s - loss: 0.0640 - accuracy: 0.9688
128/60000 [.....] - ETA: 54s - loss: 0.2279 - accuracy: 0.9297
192/60000 [.....] - ETA: 52s - loss: 0.2148 - accuracy: 0.9375
288/60000 [.....] - ETA: 45s - loss: 0.2253 - accuracy: 0.9410
384/60000 [.....] - ETA: 42s - loss: 0.5540 - accuracy: 0.9401
512/60000 [.....] - ETA: 40s - loss: 0.6697 - accuracy: 0.9336
608/60000 [.....] - ETA: 39s - loss: 0.5803 - accuracy: 0.9375
736/60000 [.....] - ETA: 37s - loss: 0.5362 - accuracy: 0.9402
864/60000 [.....] - ETA: 36s - loss: 0.5190 - accuracy: 0.9433
992/60000 [.....] - ETA: 35s - loss: 0.4905 - accuracy: 0.9385
1120/60000 [.....] - ETA: 34s - loss: 0.4690 - accuracy: 0.9357
1216/60000 [.....] - ETA: 33s - loss: 0.4858 - accuracy: 0.9367
1344/60000 [.....] - ETA: 33s - loss: 0.4807 - accuracy: 0.9368
1472/60000 [.....] - ETA: 32s - loss: 0.4909 - accuracy: 0.9361
1568/60000 [.....] - ETA: 32s - loss: 0.4859 - accuracy: 0.9337
1664/60000 [.....] - ETA: 32s - loss: 0.4660 - accuracy: 0.9351
1728/60000 [.....] - ETA: 33s - loss: 0.4715 - accuracy: 0.9346
1856/60000 [.....] - ETA: 33s - loss: 0.5137 - accuracy: 0.9327
1984/60000 [.....] - ETA: 32s - loss: 0.4862 - accuracy: 0.9345
2112/60000 [>.....] - ETA: 32s - loss: 0.4753 - accuracy: 0.9332
2208/60000 [>.....] - ETA: 32s - loss: 0.4561 - accuracy: 0.9352
2304/60000 [>.....] - ETA: 32s - loss: 0.4397 - accuracy: 0.9366
2336/60000 [>.....] - ETA: 33s - loss: 0.4355 - accuracy: 0.9371
2432/60000 [>.....] - ETA: 33s - loss: 0.4525 - accuracy: 0.9367
2496/60000 [>.....] - ETA: 34s - loss: 0.4442 - accuracy: 0.9375
2592/60000 [>.....] - ETA: 34s - loss: 0.4392 - accuracy: 0.9375
2688/60000 [>.....] - ETA: 34s - loss: 0.4842 - accuracy: 0.9371
2784/60000 [>.....] - ETA: 34s - loss: 0.4779 - accuracy: 0.9368
2880/60000 [>.....] - ETA: 34s - loss: 0.4683 - accuracy: 0.9378
2976/60000 [>.....] - ETA: 34s - loss: 0.4615 - accuracy: 0.9382
3072/60000 [>.....] - ETA: 34s - loss: 0.4617 - accuracy: 0.9368
3168/60000 [>.....] - ETA: 34s - loss: 0.4572 - accuracy: 0.9369
3264/60000 [>.....] - ETA: 34s - loss: 0.4800 - accuracy: 0.9366
3360/60000 [>.....] - ETA: 34s - loss: 0.4815 - accuracy: 0.9366
3456/60000 [>.....] - ETA: 33s - loss: 0.4799 - accuracy: 0.9346
3552/60000 [>.....] - ETA: 33s - loss: 0.4726 - accuracy: 0.9344
3648/60000 [>.....] - ETA: 33s - loss: 0.4651 - accuracy: 0.9342
```

```
retraining_the_model #11
30000/60000 [.....] - ETA: 1s - loss: 0.5195 - accuracy: 0.9300
38000/60000 [.....] - ETA: 1s - loss: 0.5188 - accuracy: 0.9309
58304/60000 [.....] - ETA: 1s - loss: 0.5182 - accuracy: 0.9309
58400/60000 [.....] - ETA: 0s - loss: 0.5178 - accuracy: 0.9309
58496/60000 [.....] - ETA: 0s - loss: 0.5182 - accuracy: 0.9308
58524/60000 [.....] - ETA: 0s - loss: 0.5179 - accuracy: 0.9308
58720/60000 [.....] - ETA: 0s - loss: 0.5172 - accuracy: 0.9308
58816/60000 [.....] - ETA: 0s - loss: 0.5164 - accuracy: 0.9309
58912/60000 [.....] - ETA: 0s - loss: 0.5167 - accuracy: 0.9309
59008/60000 [.....] - ETA: 0s - loss: 0.5176 - accuracy: 0.9310
59040/60000 [.....] - ETA: 0s - loss: 0.5174 - accuracy: 0.9310
59104/60000 [.....] - ETA: 0s - loss: 0.5170 - accuracy: 0.9310
59200/60000 [.....] - ETA: 0s - loss: 0.5173 - accuracy: 0.9309
59328/60000 [.....] - ETA: 0s - loss: 0.5177 - accuracy: 0.9310
59456/60000 [.....] - ETA: 0s - loss: 0.5232 - accuracy: 0.9310
59552/60000 [.....] - ETA: 0s - loss: 0.5256 - accuracy: 0.9310
59648/60000 [.....] - ETA: 0s - loss: 0.5262 - accuracy: 0.9310
59712/60000 [.....] - ETA: 0s - loss: 0.5258 - accuracy: 0.9310
59808/60000 [.....] - ETA: 0s - loss: 0.5251 - accuracy: 0.9310
59904/60000 [.....] - ETA: 0s - loss: 0.5244 - accuracy: 0.9311
60000/60000 [.....] - 36s 596us/step - loss: 0.5237 - accuracy: 0.9311
Epoch 10/10
32/60000 [.....] - ETA: 40s - loss: 0.0059 - accuracy: 1.0000
128/60000 [.....] - ETA: 35s - loss: 0.5437 - accuracy: 0.9609
224/60000 [.....] - ETA: 34s - loss: 0.3908 - accuracy: 0.9509
352/60000 [.....] - ETA: 32s - loss: 0.3298 - accuracy: 0.9574
448/60000 [.....] - ETA: 32s - loss: 0.2968 - accuracy: 0.9531
576/60000 [.....] - ETA: 31s - loss: 0.2540 - accuracy: 0.9601
704/60000 [.....] - ETA: 31s - loss: 0.2740 - accuracy: 0.9602
832/60000 [.....] - ETA: 30s - loss: 0.3511 - accuracy: 0.9531
960/60000 [.....] - ETA: 30s - loss: 0.3134 - accuracy: 0.9552
1056/60000 [.....] - ETA: 30s - loss: 0.3050 - accuracy: 0.9527
1152/60000 [.....] - ETA: 31s - loss: 0.2862 - accuracy: 0.9531
1216/60000 [.....] - ETA: 32s - loss: 0.2857 - accuracy: 0.9507
1312/60000 [.....] - ETA: 32s - loss: 0.3003 - accuracy: 0.9436
1440/60000 [.....] - ETA: 32s - loss: 0.2911 - accuracy: 0.9451
1536/60000 [.....] - ETA: 32s - loss: 0.2979 - accuracy: 0.9434
1632/60000 [.....] - ETA: 32s - loss: 0.3146 - accuracy: 0.9418
1760/60000 [.....] - ETA: 31s - loss: 0.3655 - accuracy: 0.9392
1856/60000 [.....] - ETA: 31s - loss: 0.3856 - accuracy: 0.9343
1952/60000 [.....] - ETA: 31s - loss: 0.4153 - accuracy: 0.9308
2016/60000 [>.....] - ETA: 32s - loss: 0.4162 - accuracy: 0.9301
2144/60000 [>.....] - ETA: 32s - loss: 0.4655 - accuracy: 0.9282
2240/60000 [>.....] - ETA: 32s - loss: 0.4545 - accuracy: 0.9277
2336/60000 [>.....] - ETA: 32s - loss: 0.4448 - accuracy: 0.9277
2432/60000 [>.....] - ETA: 32s - loss: 0.4520 - accuracy: 0.9252
2528/60000 [>.....] - ETA: 32s - loss: 0.4448 - accuracy: 0.9244
2624/60000 [>.....] - ETA: 32s - loss: 0.4473 - accuracy: 0.9234
2752/60000 [>.....] - ETA: 31s - loss: 0.4429 - accuracy: 0.9230
2880/60000 [>.....] - ETA: 31s - loss: 0.4651 - accuracy: 0.9219
```

```
32/10000 [.....] - ETA: 2:09
288/10000 [.....] - ETA: 15s
544/10000 [>.....] - ETA: 9s
864/10000 [=>.....] - ETA: 6s
1216/10000 [==>.....] - ETA: 4s
1504/10000 [===>.....] - ETA: 3s
1760/10000 [====>.....] - ETA: 3s
2112/10000 [=====>.....] - ETA: 2s
2464/10000 [=====>.....] - ETA: 2s
2848/10000 [=====>.....] - ETA: 2s
3232/10000 [=====>.....] - ETA: 1s
3616/10000 [=====>.....] - ETA: 1s
4000/10000 [=====>.....] - ETA: 1s
4352/10000 [=====>.....] - ETA: 1s
4704/10000 [=====>.....] - ETA: 1s
5088/10000 [=====>.....] - ETA: 1s
5472/10000 [=====>.....] - ETA: 1s
5728/10000 [=====>.....] - ETA: 0s
6016/10000 [=====>.....] - ETA: 0s
6272/10000 [=====>.....] - ETA: 0s
6592/10000 [=====>.....] - ETA: 0s
6944/10000 [=====>.....] - ETA: 0s
7328/10000 [=====>.....] - ETA: 0s
7680/10000 [=====>.....] - ETA: 0s
8064/10000 [=====>.....] - ETA: 0s
8416/10000 [=====>.....] - ETA: 0s
8640/10000 [=====>.....] - ETA: 0s
8928/10000 [=====>.....] - ETA: 0s
9280/10000 [=====>...] - ETA: 0s
9696/10000 [=====>.] - ETA: 0s
10000/10000 [=====] - 2s 200us/step
```

Using TensorFlow backend.

acc=91.63%

Triggering a new build of [restart_container](#)

Triggering a new build of [restart_container](#)

Finished: SUCCESS

JOB 6 ->> This Job will do monitoring , if docker goes down then it will restart it and will trigger the JOB1

restart_container

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Description: This Job will do monitoring and restrant the container

[Plain text] [Preview](#)

- ☐ Commit agent's Docker container
- ☐ Define a Docker template
- ☐ Discard old builds
- ☐ Github project
- ☐ This build requires lockable resources
- ☐ This project is parameterized
- ☐ Throttle builds
- ☐ Disable this project
- ☐ Execute concurrent builds if necessary
- ☐ Restrict where this project can be run

[Advanced...](#)

Source Code Management

☒ None

☐ Git

Build Triggers

☐ Trigger builds remotely (e.g., from scripts)

☒ Build after other projects are built

Projects to watch: retraining_the_model

☐ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

[Save](#) [Apply](#)

restart_container

General Source Code Management Build Triggers Build Environment Build Post-build Actions

☐ Github hook trigger for GITScm polling

☐ Poll SCM

Build Environment

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s)
- ☐ Abort the build if it's stuck
- ☐ Add timestamps to the Console Output
- ☐ Inspect build log for published Gradle build scans
- ☐ With Ant

Build

☒ Execute shell

Command: `sudo systemctl start docker`

[See the list of available environment variables](#)

[Advanced...](#)

[Add build step](#)

Post-build Actions

☒ Build other projects

Projects to build: Train_the_model

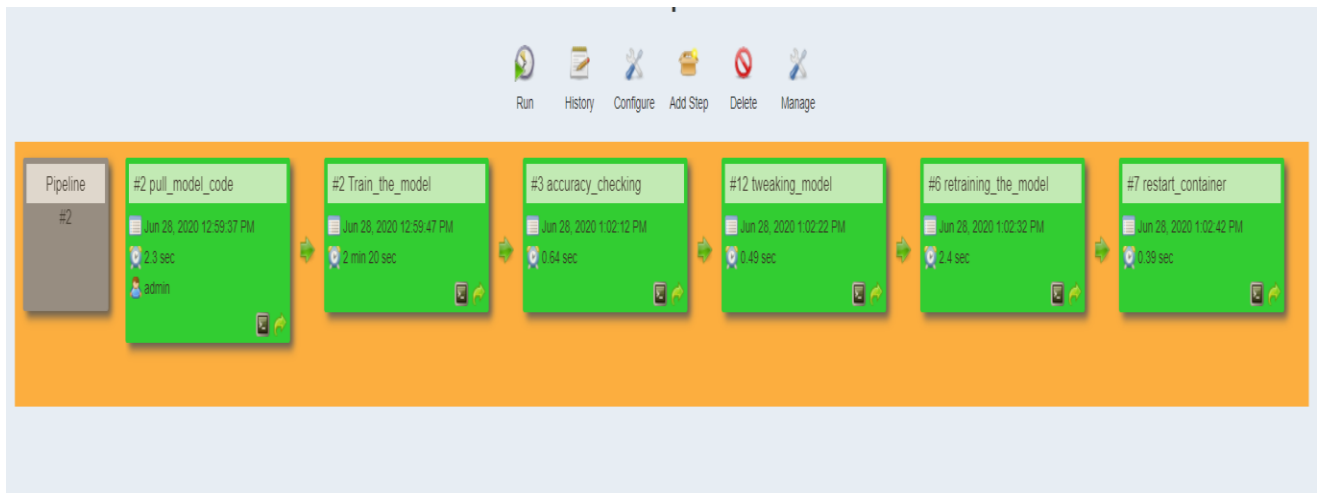
☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

[Save](#) [Apply](#)

BUILD PIPELINE VIEW OF WHOLE TASK



THANK YOU!!!

For any query whatsapp -9690881508