Name - Arpit Gupta
Student Id - 169572237

```
tf fmt
tf init
tf plan
tf apply

connect to instance -

sudo apt update -y
sudo apt install -y docker.io
sudo systemctl start docker
sudo systemctl enable docker
docker --version
sudo systemctl status docker
sudo usermod -aG docker $USER   (adding user mod to docker image)

exit and login again
---------------------

#custom network of type bridge
docker network create -d bridge --subnet 10.0.0.0/24 --gateway 10.0.0.1 my-network

----------------
sudo apt install awscli
sudo apt install awscli
aws ecr get-login-password --region us-east-1 | docker login -u AWS 135893829551.dkr.ecr.us-east-1.amazonaws.com/mysql-repo:mysql-latest --
password-stdin


ecr_db=135893829551.dkr.ecr.us-east-1.amazonaws.com/mysql-repo:mysql-latest
aws ecr get-login-password --region us-east-1 | docker login -u AWS 135893829551.dkr.ecr.us-east-1.amazonaws.com/mysql-repo:mysql-latest --
password-stdin

docker run -d -e MYSQL_ROOT_PASSWORD=pw --network my-network --name mysql-db $ecr_db

docker run -d -e MYSQL_ROOT_PASSWORD=pw --network my-network --name mysql-db 135893829551.dkr.ecr.us-east-1.amazonaws.com/mysql-repo:mysql-
latest
----------------

#for pulling webapp_repo image
b6f57f8d30c1541b1c7f282a47f5e577b85f092e686ccc9ff66184e492487f9f
ecr_app=135893829551.dkr.ecr.us-east-1.amazonaws.com/webapp-repo:app-latest
aws ecr get-login-password --region us-east-1 | docker login -u AWS 135893829551.dkr.ecr.us-east-1.amazonaws.com/webapp-repo:app-latest --
password-stdin
aws ecr get-login-password --region us-east-1 | docker login -u AWS $ecr_app --password-stdin
ecr_db=135893829551.dkr.ecr.us-east-1.amazonaws.com/mysql-repo:mysql-latest

docker run -d -e MYSQL_ROOT_PASSWORD=pw --network my-network --
docker run -d -e MYSQL_ROOT_PASSWORD=pw --network my-network --name mysql-db ecr_db=844188451376.dkr.ecr.us-east-1.amazonaws.com/mysql_repo:
mysql-latest

it will show us the container id

do docker inspect
docker inspect 48cbdecd6b31211d5813e85871608a0cf994e77aab3d63127506d2ad53815b4a
---------------
#test MySQL db connection and showing created database
docker exec -it mysql-db /bin/bash
mysql -uroot -ppw -e "SHOW DATABASES;"

exit
------------------
export DBHOST=10.0.0.2
export DBPORT=3306
export DBUSER=root
export DATABASE=employees
export DBPWD=pw
----------------

#for pulling webapp_repo image

ecr_app=135893829551.dkr.ecr.us-east-1.amazonaws.com/webapp-repo:app-latest
aws ecr get-login-password --region us-east-1 | docker login -u AWS 135893829551.dkr.ecr.us-east-1.amazonaws.com/webapp-repo:app-latest --
password-stdin

-------------------------
#running containers in detach mode for
docker run -d -p 8081:8080 -e DBHOST=$DBHOST -e DBPORT=$DBPORT -e DBUSER=$DBUSER -e DATABASE=$DATABASE -e DBPWD=$DBPWD -e APP_COLOR=blue --
network my-network --name webappblue $ecr_app
docker run -d -p 8082:8080 -e DBHOST=$DBHOST -e DBPORT=$DBPORT -e DBUSER=$DBUSER -e DATABASE=$DATABASE -e DBPWD=$DBPWD -e APP_COLOR=pink --
network my-network --name webapppink $ecr_app
docker run -d -p 8083:8080 -e DBHOST=$DBHOST -e DBPORT=$DBPORT -e DBUSER=$DBUSER -e DATABASE=$DATABASE -e DBPWD=$DBPWD -e APP_COLOR=lime --
network my-network --name webapplime $ecr_app
----------------
Containers can ping each other using their host names. For example, the following should work from inside the blue container: ping pink ping
```

```
lime
-----------------------
docker exec -it blue-app /bin/bash
docker exec -it pink-app /bin/bash
docker exec -it lime-app /bin/bash

#install ping
apt-get update && apt-get install iputils-ping -y
-------------
ping webapplime
ping webapppink
ping webappblue
```