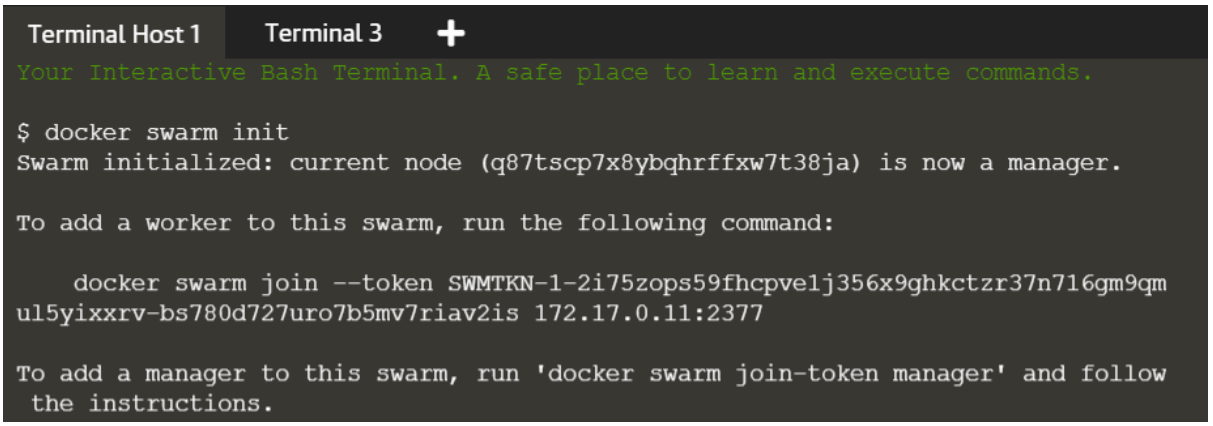
Anurag Pachauri
500069505
R171218026

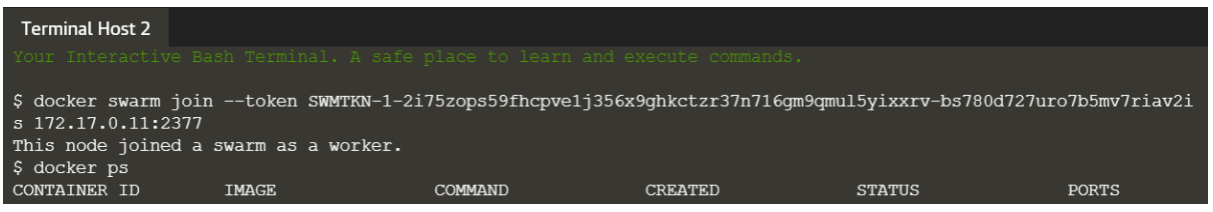
Aim: To create Docker swarm cluster.

Procedure:

1) Use command **docker swarm init** on a node to initialize it as master node.



2) Copy this docker swarm join --token command and run it on the another node to join it in the swarm clusterand configure it as worker node.



3) On the master node run command docker node ls to see all the nodes that have joined the swarm cluster.

```
$ docker node ls
ID                            HOSTNAME            STATUS              AVAILABILI
TY        MANAGER STATUS      ENGINE VERSION
q87tscp7x8ybqhrffxw7t38ja *   host01              Ready               Active
          Leader              19.03.13
2929ie1z9deqjk04y9nk7gwzz     host02              Ready               Active
                              19.03.13
```

4) To create an overlay network via swarm manager node, use command **docker network create -d overlay app1-network**.

```
$ docker network create -d overlay app1-network
m6k4kmr8xbowazpaodcpm0e6w
$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
m6k4kmr8xbow        app1-network        overlay             swarm
a64124b9c3a8        bridge              bridge              local
991a7e8fd0ce        docker_gwbridge     bridge              local
8b89e3388c32        host                host                local
lhgcs9hl6xoz        ingress             overlay             swarm
b3dc159371bf        none                null                local
```

5) To create a service, use command
docker service create --network <network-name> -p <port-forwading> --replicas <replicas> --name

<service-name> <image-name>.

```
$ docker service create \
>     --network app1-network -p 80:3000 \
>     --replicas 1 --name app1-web \
>     katacoda/redis-node-docker-example
image katacoda/redis-node-docker-example:latest could not be accessed on a regis
try to record
its digest. Each node will access katacoda/redis-node-docker-example:latest inde
pendently,
possibly leading to different nodes running different
versions of the image.

j8op4dm9hp2c7htgq7expw6bk
overall progress: 1 out of 1 tasks
1/1: running   [==================================================>]
verify: Waiting 2 seconds to verify that tasks are stable...
verify: Service converged
```

6) Use command docker service ls to check all the services running.

```
$ docker service ls
ID                  NAME                MODE                REPLICAS
IMAGE                                          PORTS
j8op4dm9hp2c        app1-web            replicated          1/1
katacoda/redis-node-docker-example:latest   *:80->3000/tcp
bqh3u0uzxb4e        redis               replicated          1/1
redis:alpine
```

7) Use command **docker service ps <service-name>** to check the state of the service.

```
$ docker service ps app1-web
ID                  NAME                IMAGE
    NODE                DESIRED STATE       CURRENT STATE            ERROR
         PORTS
t1adji1ke1e2        app1-web.1          katacoda/redis-node-docker-example:lates
t   host02             Running             Running 50 minutes ago
```