Experiment 7

Roll no: R171218009

AIM: Docker swarm

initialize Swarm Mode

```
$ docker swarm init
Swarm initialized: current node (labw4681jnq9pqwvoetkp1y1c) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-2gyzs45eo59moqrpu2epfnd0psjyhorgwesa7wy0t
jrnn3dj2z-cedyi0rjo1sfzpmgv2p0u6wjr 172.17.0.62:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow
 the instructions.
```

Store token required to add a worker to the cluster

```
$ token=$(ssh -o StrictHostKeyChecking=no 172.17.0.62 "docker swarm join-token -q worker") && echo $token
Warning: Permanently added '172.17.0.62' (ECDSA) to the list of known hosts.
SWMTKN-1-2gyzs45eo59moqrpu2epfnd0psjyhorgwesa7wy0tjrnn3dj2z-cedyi0rjo1sfzpmgv2p0u6wjr
```

On the second host, join the cluster by requesting access via the manager

```
$ docker swarm join 172.17.0.62:2377 --token $token
This node joined a swarm as a worker.
```

view all nodes in the cluster

```
$ docker node ls
ID                            HOSTNAME            STATUS              AVAILABILI
TY        MANAGER STATUS      ENGINE VERSION
labw4681jnq9pqwvoetkp1y1c *   host01              Ready               Active
          Leader              19.03.13
o7a1yrxtmf5g4sptu0sw1u9r9     host02              Ready               Active
                              19.03.13
```

create a new overlay network called *Skynet*

```
$ docker network create -d overlay skynet
ngvrgy6fm4axb44h9cu0mk0wn
```

Load balance these two containers together on port *80*. Sending an HTTP request to any of the nodes in the cluster will process the request by one of the containers within the cluster.

```
$ docker service create --name http --network skynet --replicas 2 -p 80:80 katac
oda/docker-http-server
le4pjgsl09rq9tioga8cr5wfk
overall progress: 0 out of 2 tasks
overall progress: 2 out of 2 tasks
1/2: running   [==================================================>]
2/2: running   [==================================================>]
verify: Waiting 5 seconds to verify that tasks are stable...
verify: Waiting 2 seconds to verify that tasks are stable...
verify: Service converged
```

view the services running on the cluster

```
$ docker service ls
ID                  NAME                MODE                REPLICAS
IMAGE                               PORTS
le4pjgsl09rq        http                replicated          2/2
katacoda/docker-http-server:latest   *:80->80/tcp
```

List containers on the first host

```
$ docker ps
CONTAINER ID        IMAGE                                 COMMAND             CRE
ATED             STATUS              PORTS               NAMES
497a813903a5        katacoda/docker-http-server:latest   "/app"              6 s
econds ago       Up 5 seconds        80/tcp              http.2.t6sxw6z8v51p0vun
fr33xubcg
```

List containers on the second host

```
$ docker ps
CONTAINER ID        IMAGE                                 COMMAND             CREATED             STATUS
    PORTS               NAMES
a13584ad258f        katacoda/docker-http-server:latest   "/app"              2 seconds ago       Up 2 seconds
    80/tcp              http.1.c1dh64u9baqod19rovbbu6gb3
```

issue an HTTP request to the public port

```
$ curl host01
<h1>This request was processed by host: a13584ad258f</h1>
$ docker service ps http
ID                  NAME                IMAGE                                 NOD
E                   DESIRED STATE       CURRENT STATE           ERROR
  PORTS
c1dh64u9baqo        http.1              katacoda/docker-http-server:latest   hos
t02                 Running             Running 11 seconds ago

t6sxw6z8v51p        http.2              katacoda/docker-http-server:latest   hos
t01                 Running             Running 11 seconds ago
```

view the list of all the tasks associated with a service across the cluster

```
$ docker service ps http
ID                  NAME                IMAGE                                     NOD
E                   DESIRED STATE       CURRENT STATE            ERROR
  PORTS
c1dh64u9baqo        http.1              katacoda/docker-http-server:latest   hos
t02                 Running             Running 11 seconds ago

t6sxw6z8v51p        http.2              katacoda/docker-http-server:latest   hos
t01                 Running             Running 11 seconds ago
```

view the details and configuration of a service

```
$ docker service inspect --pretty http

ID:             le4pjgsl09rq9tioga8cr5wfk
Name:           http
Service Mode:   Replicated
 Replicas:      2
Placement:
UpdateConfig:
 Parallelism:   1
 On failure:    pause
 Monitoring Period: 5s
 Max failure ratio: 0
 Update order:      stop-first
RollbackConfig:
 Parallelism:   1
 On failure:    pause
 Monitoring Period: 5s
 Max failure ratio: 0
 Rollback order:    stop-first
ContainerSpec:
 Image:         katacoda/docker-http-server:latest@sha256:76dc8a47fd019f80f2a316
3aba789faf55b41b2fb06397653610c754cb12d3ee
 Init:          false
Resources:
Networks: skynet
Endpoint Mode:  vip
Ports:
 PublishedPort = 80
  Protocol = tcp
  TargetPort = 80
  PublishMode = ingress
```