### Prerequisites

For this task you will need:

* 3 x Ubuntu VM with Docker Installed with the following names:
  + swarm-manager
  + swarm-worker
  + nginx
* Port 9000 on swarm-manager and swarm-worker should be open.
* Port 80 on the NGINX VM should also be open.

### Initialising the Swarm

SSH into your swarm-manager VM and enter the following command:

docker swarm init

This will now initialise the manager node and output a worker token similar to the token below:

docker swarm join --token SWMTKN-1-2tcnqv3awppiy848ho9cewqop1g3vvuxfk5mn8po27pjkpx304d-4l8vpniuvzzt22jd1k6q64t5p 192.168.12.2:2477

**Note this Command down as you will need to run it on the "worker node" VM to add a worker node to the swarm.**

### Join the Worker to the Swarm

On your swarm-worker VM, paste the join command that was generated by the manager node earlier.

If you forgot to note down the join token run this command on the Manager VM to generate another worker token:

docker swarm join-token worker

While on the manager node VM list all nodes in the swarm with the following command:

docker node ls

### Create a Service

On the Manager node VM run the following command to initialise the Python HTTP Server application:

docker service create --replicas 2 --publish 9000:9000 --name python-http-server bobcrutchley/python-http-server

This will create a HTTP server on port 9000. Navigate to your VM's public IP address on your browser on port 9000 to see this in action!

Run the command below to see if the service is running on the swarm:

docker service ls

### Creating our External Load Balancer

Now we will need to create an NGINX configuration file in our nginx VM to work as a reverse proxy/load balancer.

Create the file with the following command:

touch nginx.conf

Open the file in your text editor of choice and paste the following code into the file:

events{}

http {

upstream python-http-server {

server "[SWARM\_MANAGER\_PRIVATE\_IP]":9000;

server "[SWARM\_WORKER\_PRIVATE\_IP]":9000;

}

server {

location / {

proxy\_pass http://python-http-server;

}

}

}

Be sure to replace [SWARM\_MANAGER\_PRIVATE\_IP] and [SWARM\_WORKER\_PRIVATE\_IP] with the private IP addresses of your swarm-manager and swarm-worker VMs.

On the nginx VM, make sure your nginx.conf file is in your current working directory and run the following command to run an NGINX container with our coniguration:

docker run -d -p 80:80 --name nginx-swarm-ingress --mount type=bind,source=$(pwd)/nginx.conf,target=/etc/nginx/nginx.conf nginx:alpine

Navigate to your nginx VM's public IP address on port 80. You should see the web server hosted in your Swarm, confirming that our nginx VM is proxy passing to your Swarm cluster.

Refresh the page a few times. If the hostname is changing each time you refresh, NGINX is also successfully load balancing between the manager and worker node.

### Clean Up

Remove the service:

docker service rm python-http-server

In your swarm-worker VM, make the node leave the Swarm:

docker swarm leave

In your swarm-manager VM, stop the Swarm:

docker swarm leave --force

In your nginx VM, remove the NGINX container:

docker rm nginx-swarm-ingress