Module-4: Orchestration in Docker

Demo Document - 2

DEMO-2: Starting Docker in Swarm Mode

Note: All commands are executed as root.

1. On your manager node run the following command to initialize a swarm cluster

```
$ docker swarm init --advertise-addr <MANAGER-IP>
```

```
root@docker-1:~# docker swarm init --advertise-adder 34.72.112.157
unknown flag: --advertise-adder
See 'docker swarm init --help'.
root@docker-1:~# docker swarm init --advertise-addr 34.72.112.157
Swarm initialized: current node (tu1ukz7mmibu3a80b70bn4nfa) is now a manager.

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

    docker swarm join --token SWMTKN-1-4p7i7n56ytjf8gcyi3qz2xreyxjzg3vigfp15falkb50nbmvc7-3y1qfpqoug2d4hmkqnpe4nsoq 34.72.112.1
57:2377

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

2. Copy the output of the init command (marked red in the image above) and use it on the worker nodes
   In case, you have lost the output, the join command can be obtained by running the following command

```
$ docker swarm join-token worker
```

```
root@docker-3:~# docker swarm join --token SWMTKN-1-4p7i7n5
q 34.72.112.157:2377
This node joined a swarm as a worker.
```

3. On the manager node run the node ls command to check all the connected nodes in the cluster

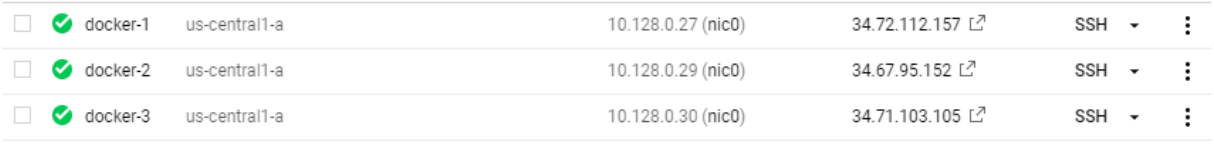
```
$ docker node ls
```

```
root@docker-1:~# docker node ls
ID                            HOSTNAME            STATUS              AVAILABILITY        MANAGER STATUS      ENGINE VERSION
tu1ukz7mmibu3a80b70bn4nfa *   docker-1            Ready               Active              Leader              19.03.11
m1y4s9m41tl8msdk2rqg1zdue     docker-2            Ready               Active                                  19.03.12
5aof5ybchhvxdq7klvu7ukqv2     docker-3            Ready               Active                                  19.03.12
```

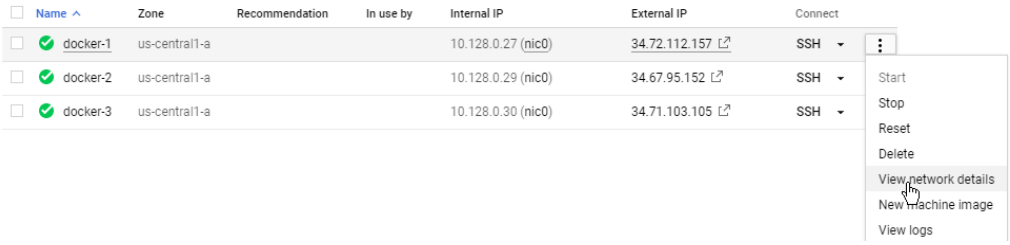
In case you want to create your own swarm cluster in a cloud environment, please enable the following ingress and egress ports on your instances: TCP 2377, TCP and UDP 7946, UDP 4789

Following is an example of how to do it on gcloud:

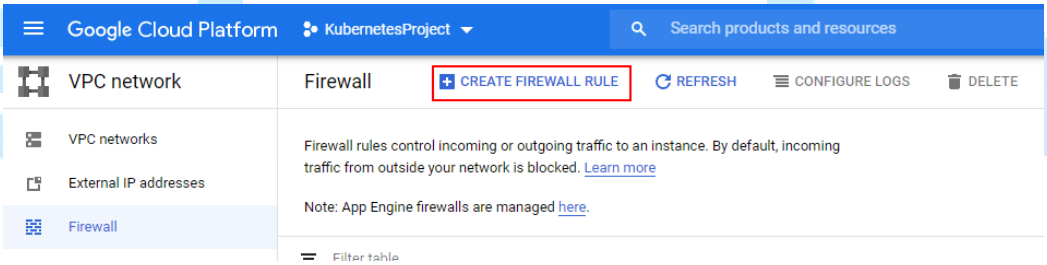
1. Create 3 ubuntu instance on GCP with Docker engine installed on them



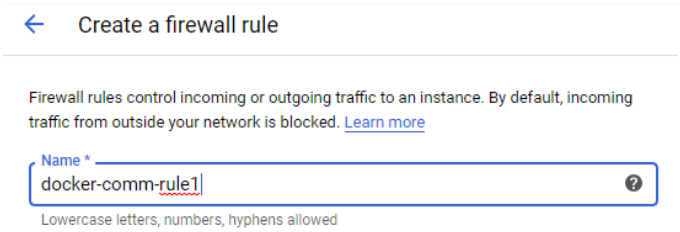
2. Before initializing with the init command set firewall rules for all the instances
   Create ingress and egress rules for TCP 2377, TCP and UDP 7946, and UDP 4789 ports
   Click on view network details and then Firewall



3. Click on CREATE FIREWALL RULE



4. Add a name for your rule



5. Select egress or ingress. Select all instances in network for Targets. In Source IP Ranges add the IPs for your instances

Set the port and protocol and click on tcp