

**UNIVERSITY OF PETROLEUM & ENERGY STUDIES**

**Dehradun**

**Application Containerization**

**Experiment 3**

**Name: Shashwat Kumar**

**Course: B-Tech CSE DevOps (2018-22)**

**Roll number: R171218093**

**Sap ID: 500068392**

**Docker Networks**

**Creating Networks in Docker**

* List the available networks using the command:

docker network ls

* Create a network using the command:

docker network create mynet1

* Check if the network was created using the command:

docker network ls

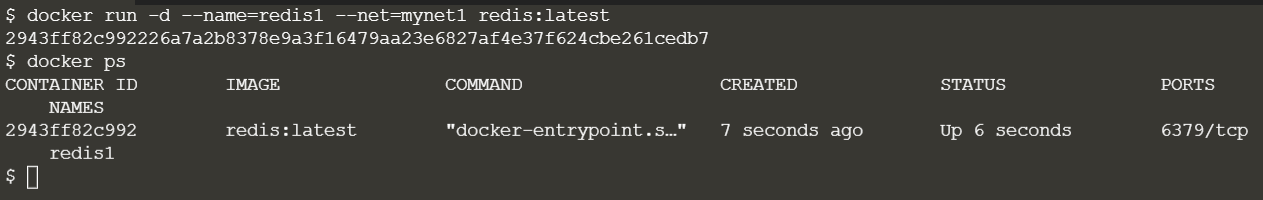


* Run a docker container in the network that was created using the command:

docker run -d –name=redis1 –net=mynet1 redis:latest

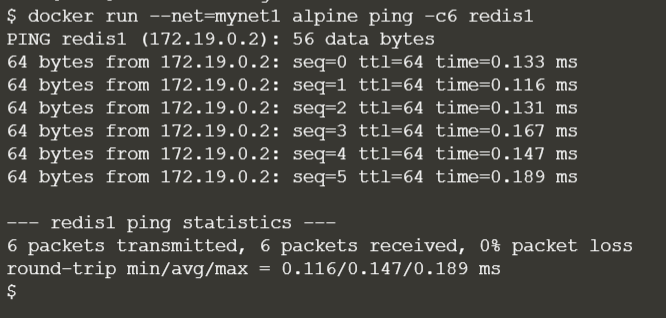
* Check if the container is running using the command:

docker ps



* Run an alpine image in the created network and try pinging the redis image in that network

docker run –net=mynet1 alpine ping -c6 redis1



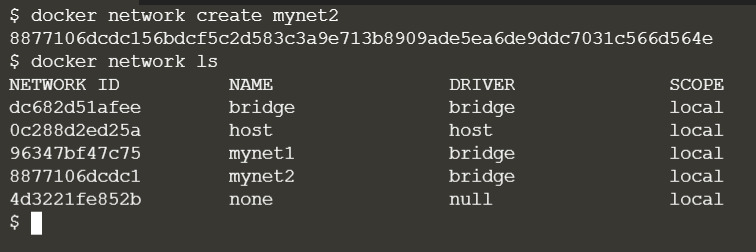
**Connecting new networks to existing containers**

* Create a new network using the command:

docker network create mynet2

* See the created network using the command:

docker network ls

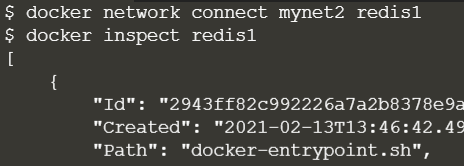


* Connect the running redis image with the newly created network using the command:

docker network connect mynet2 redis1

* Inspect the container using the command:

docker inspect redis1

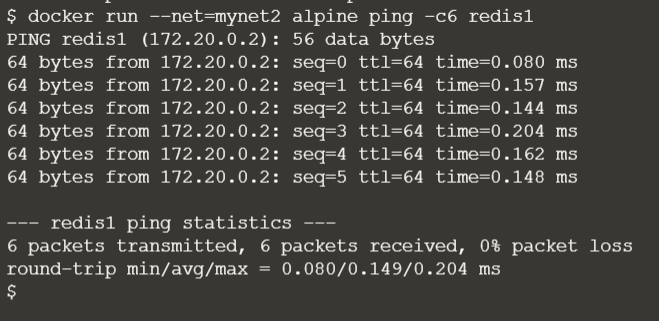


* The inspect command reveals that the running container is connected to two networks



* Run an alpine image in the newly created network and try pinging the redis image in that network

docker run –net=mynet2 alpine ping -c6 redis1



**Creating Networks using aliases**

* Create a new network using the command:

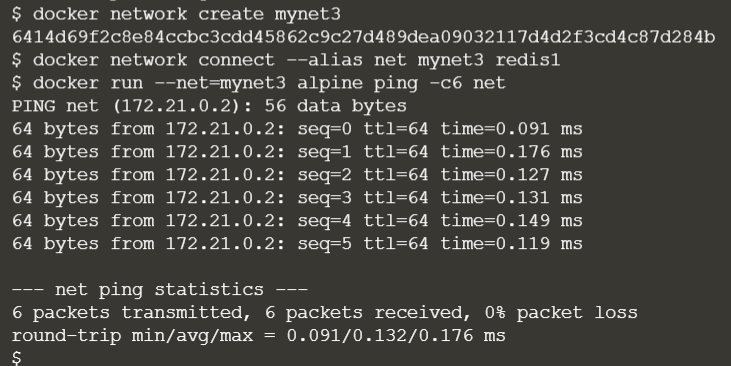
docker network create mynet3

* Connect the running redis container to the third network using an alias name using the command:

docker network connect -–alias net mynet3 redis1

* Run an alpine image in the third network and try pinging the redis image using the command:

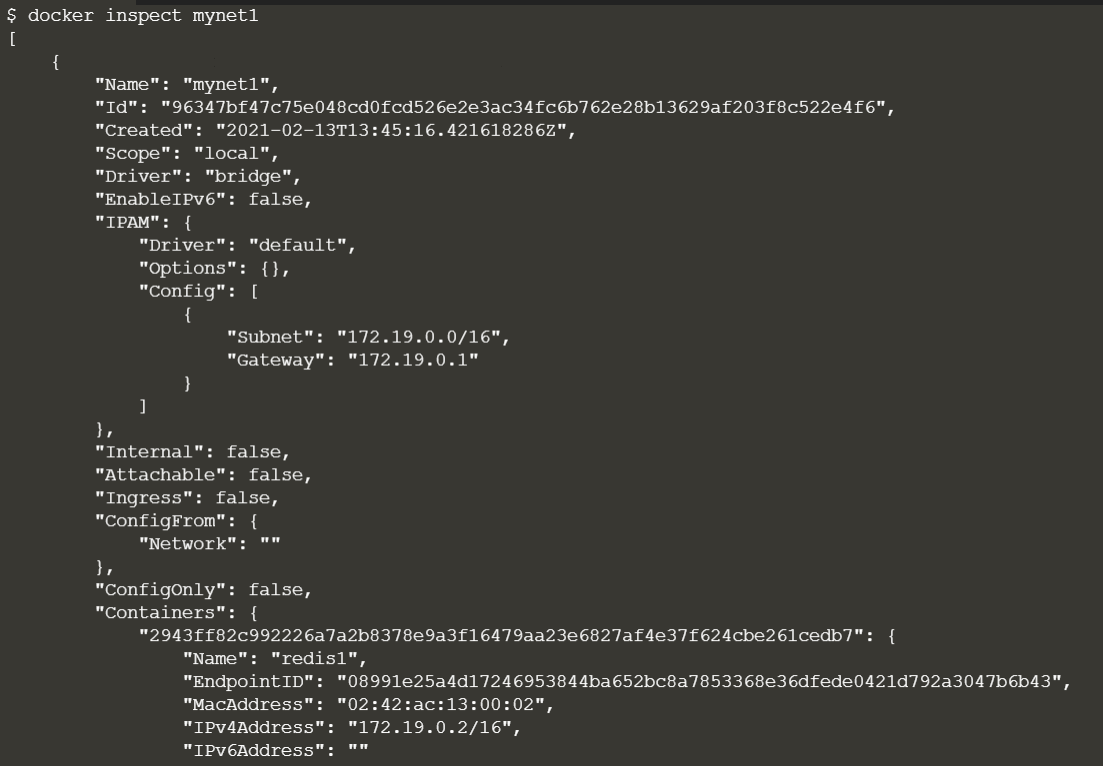
docker run –net=mynet3 alpine ping -c6 net



**Inspecting Networks**

* We can also inspect the networks to get its information using the following command:

docker inspect mynet1



**Disconnecting Networks**

* Disconnect the running container from the first created network using the command:

docker network disconnect mynet1 redis1

* Try pinging the running redis container from an alpine image in the disconnected network:

docker run –net=mynet1 alpine ping -c6 redis1

