

**UNIVERSITY OF PETROLEUM & ENERGY STUDIES**

**Dehradun**

**APPLICATION CONTAINERISATION**

**Name: Bhavini Mishra**

**B. TECH CSE DevOps (2018-22)**

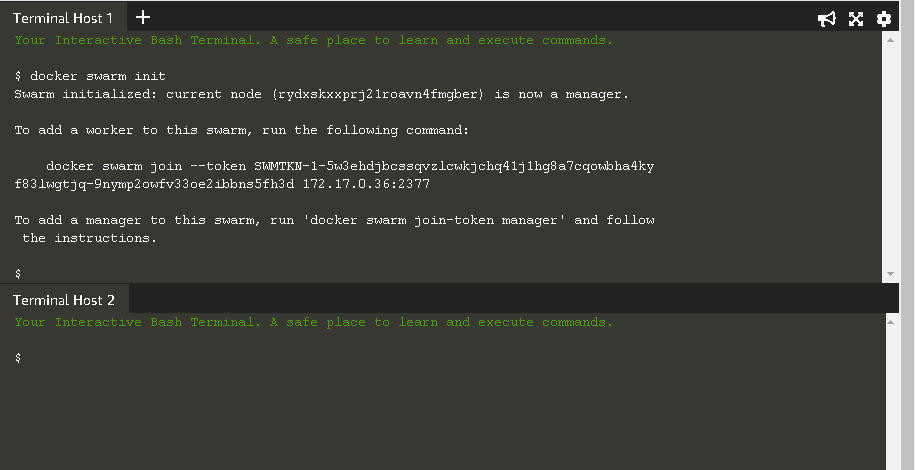
**Roll no.: R171218033**

**Experiment - 7**

**Docker Swarm**

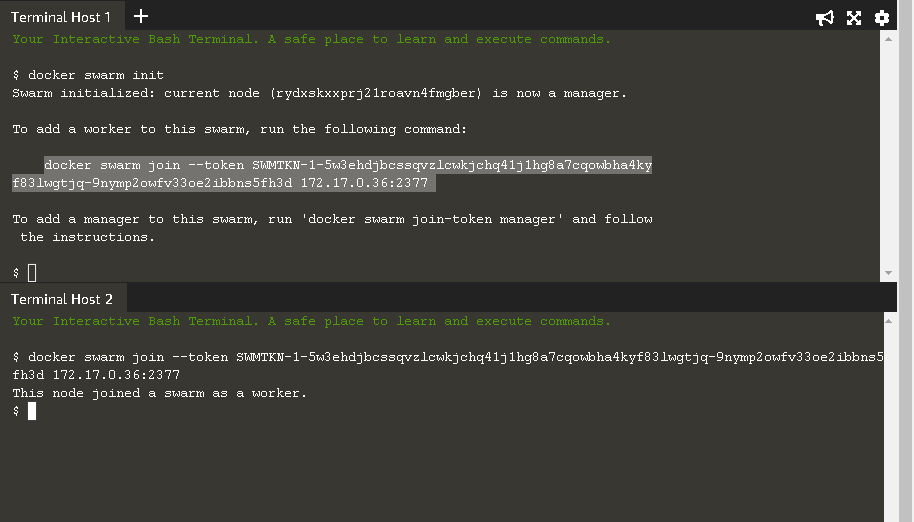
**• Initialize the Swarm Cluster into one of the terminal or virtual machine by using the following command.**

**$ docker swarm init**



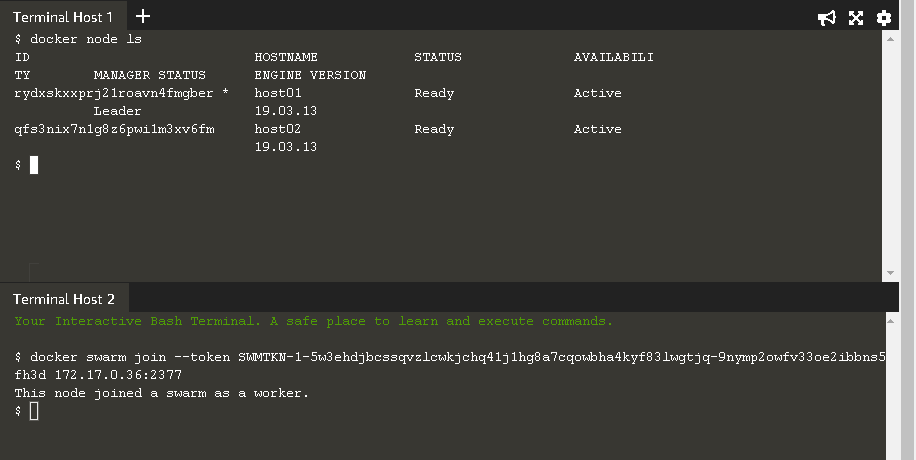
**Join the Cluster : -**

* **To add a worker to this swarm, run the following command to join the node to this swarm.**



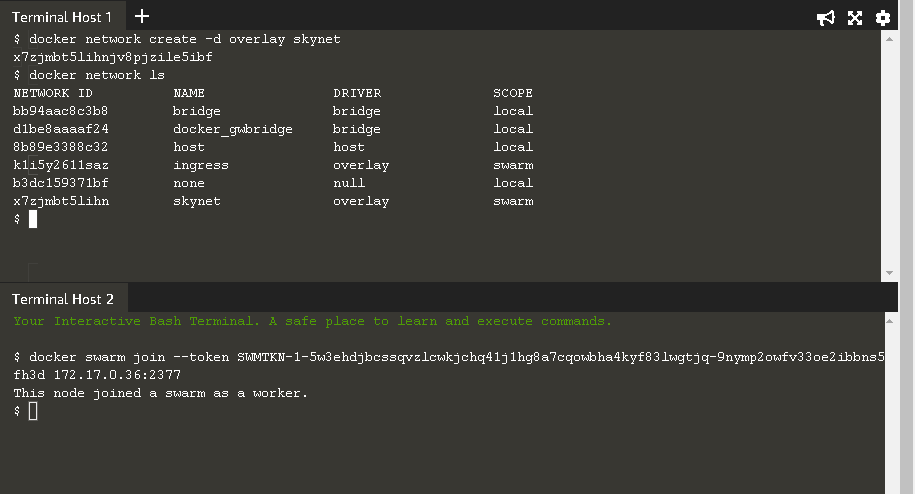
* **To see that how many nodes are joined in this Cluster by using the following command.**

**$ docker node ls**



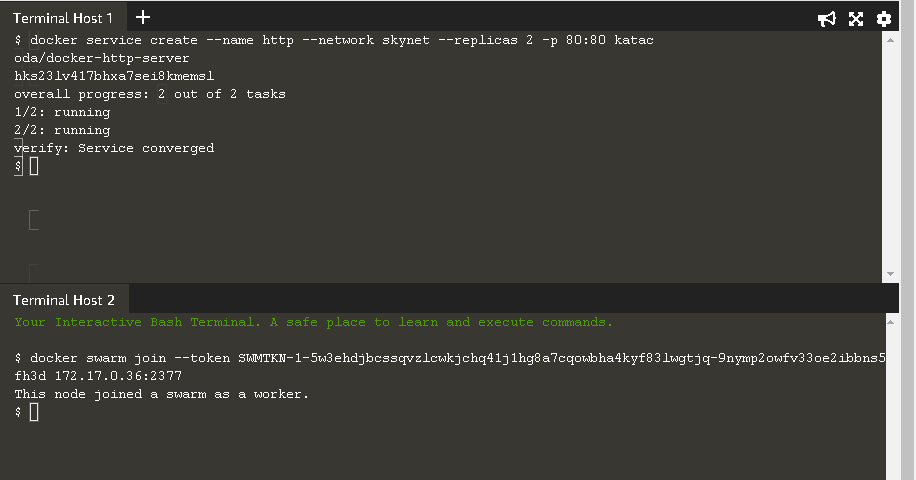
* **The following command will create a new overlay network called skynet. All containers registered to this network can communicate with each other, regardless of which node they are deployed onto.**

**$ docker network create -d overlay Skynet**



* **Now we are deploying the Docker Image katacoda/docker-http-server. We are defining a friendly name of a service called http and that it should be attached to the newly created skynet network.**

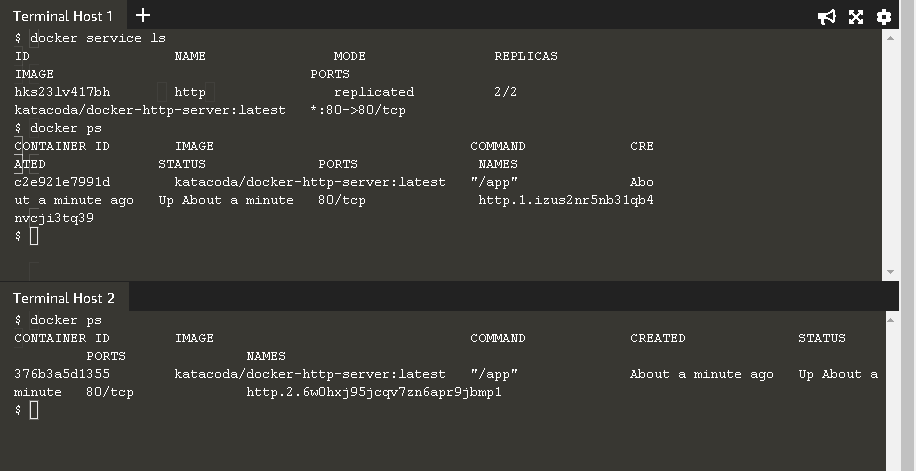
**$ docker service create --name http --network skynet --replicas 2 -p 80:80 katacoda/docker-httpserver**



* **You can view the services running on the cluster using the CLI command.**

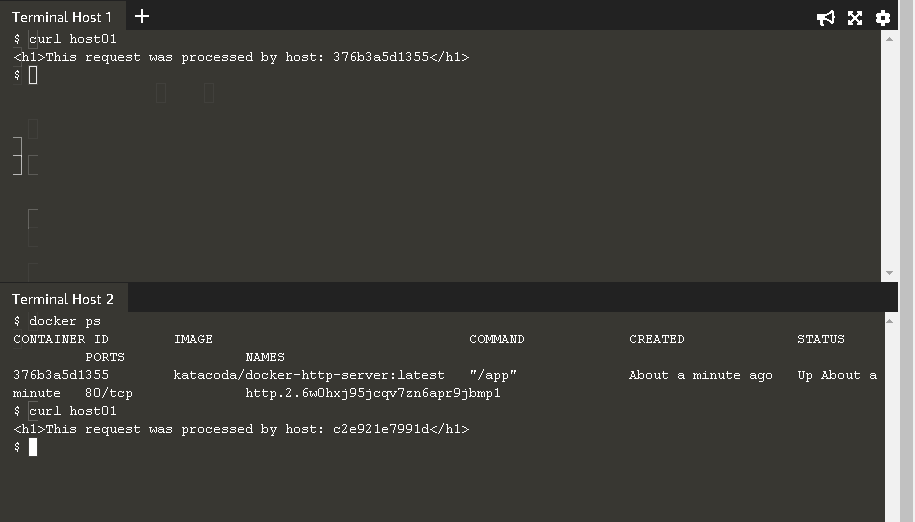
**$ docker service ls**

**As containers are started you will see them using the docker ps command. You should see one instance of the container on each host.**



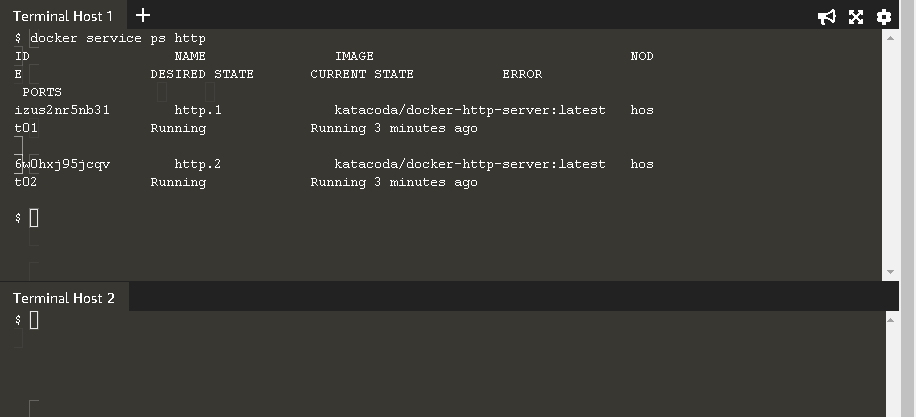
**• If we issue an HTTP request to the public port, it will be processed by the two containers**

**. $ curl host01**



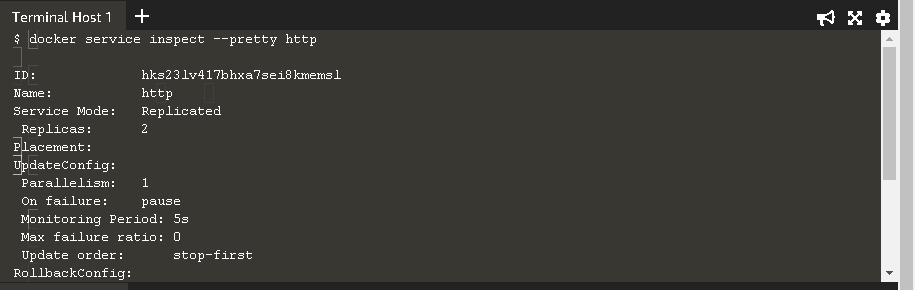
* **You can view the list of all the tasks associated with a service across the cluster. In this case, each task is a container.**

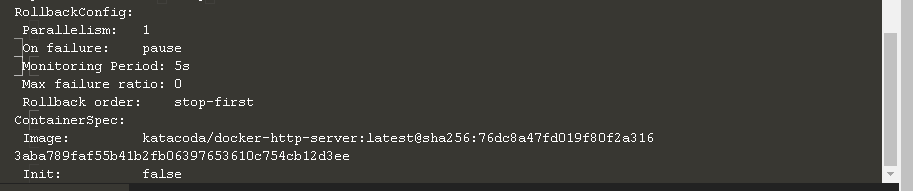
**$ docker service ps http**

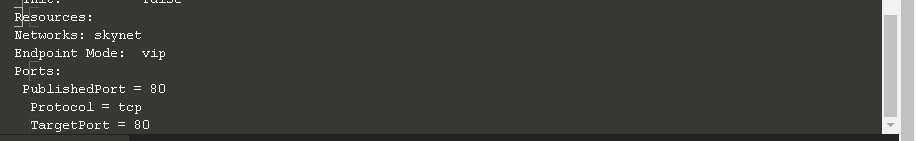


* **You can view the details and configuration of a service via**

**$ docker service inspect --pretty http**

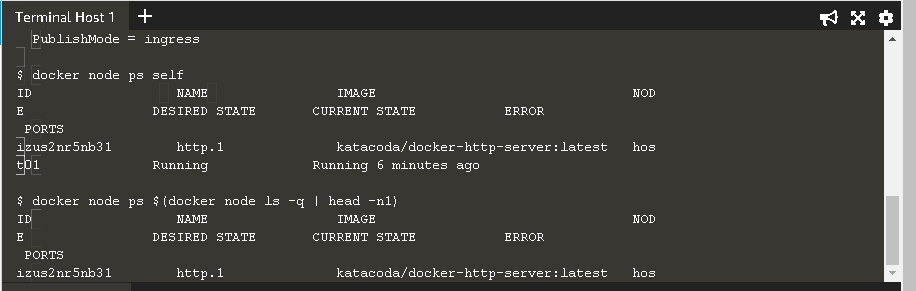






* **On each node, you can ask what tasks it is currently running. Self refers to the manager node Leader:**

**$ docker node ps self**



* **The command below will scale our http service to be running across five containers**

