Docker Swarm:

Docker Swarm is a cluster , grp of m/c where all container are running .

It is basically a collection of either virtual machines or physical machines that run the Docker Application. This group of several machines is configured to make a cluster.

create 4 instances

step 1:enable docker swarm at 192.168.0.11 (swarm node)

docker swarm init --advertise-addr=<ip>

docker swarm init --advertise-addr=192.168.0.28

docker node ls

step 2 :add other nodes to docker swarm (192.168.0.13)

cmd 1:

docker swarm join-token worker

o/p:o add a worker to this swarm, run the following command:

docker swarm join --token SWMTKN-1-52pv2bec08tk7k3915qoxroclmksoesa1mmo8mpzcdun641ok3-ckj5b86q4kgu7uc5sb9mptkjr 192.168.0.11:2377

docker swarm join --token SWMTKN-1-3zb8zb4ko3cnpqk674n78q1v48xrsjk8gfbnkx2puzrotlvm1i-7jmsun0wxq34gqzleawuqcr71 192.168.0.8:2377

TCP port 2377 . This port is used for communication between the nodes of a Docker Swarm or cluster. It only needs to be opened on manager nodes. TCP and UDP port 7946 for communication among nodes (container network discovery

step 3:

docker node ls

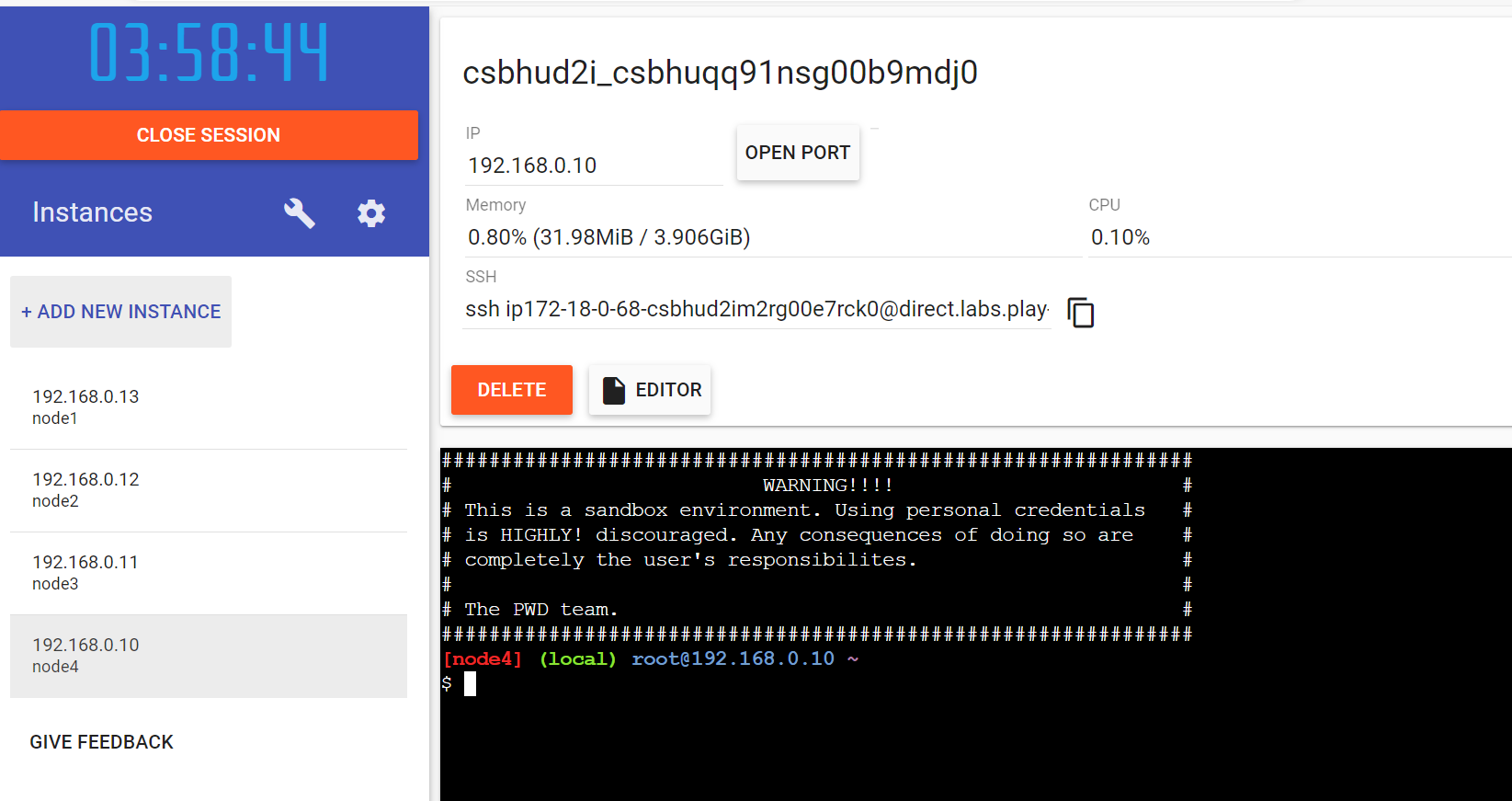
step 4:

docker network ls

step 5:

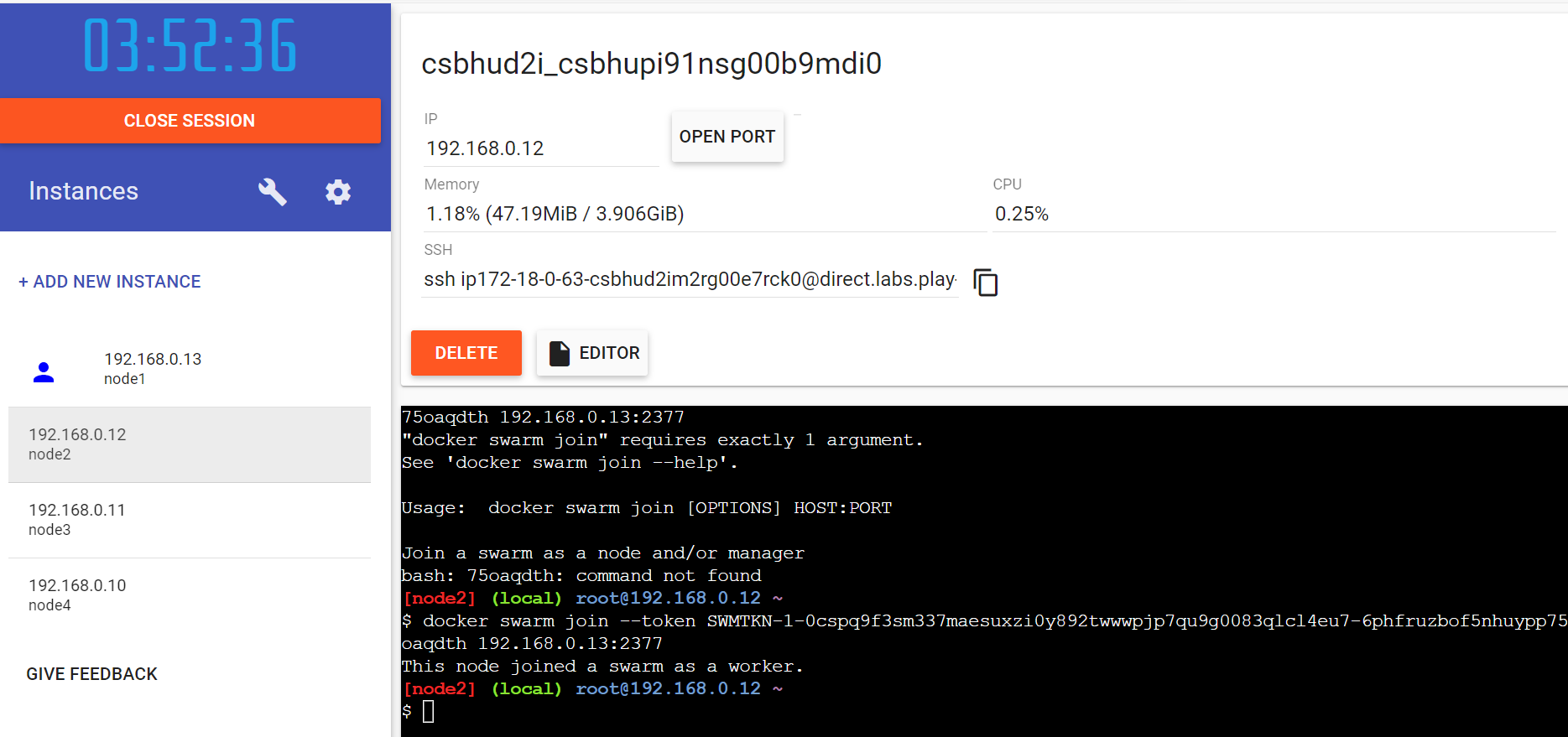
instead of run we can use docker service command.

Demo screens:

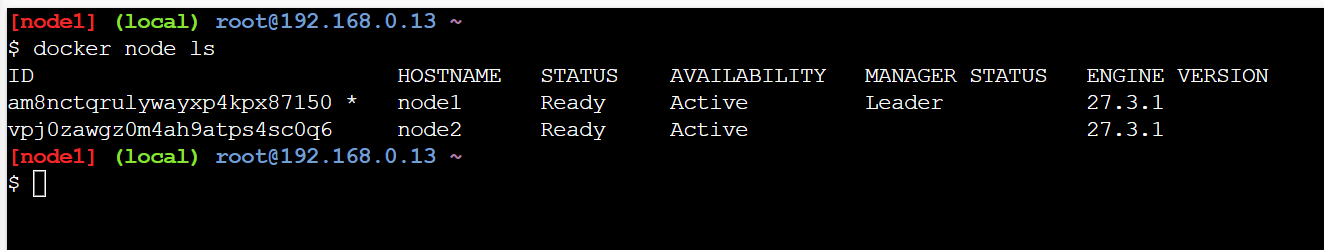


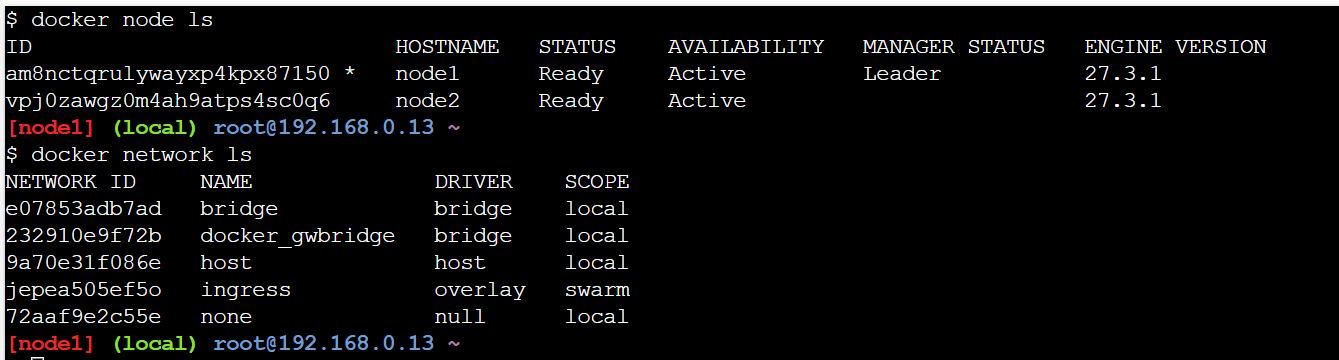
Docker swarm init --advertise-addr=<ip>

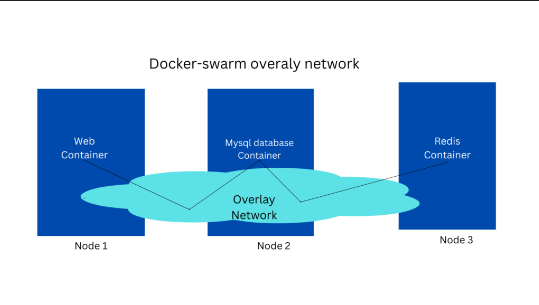


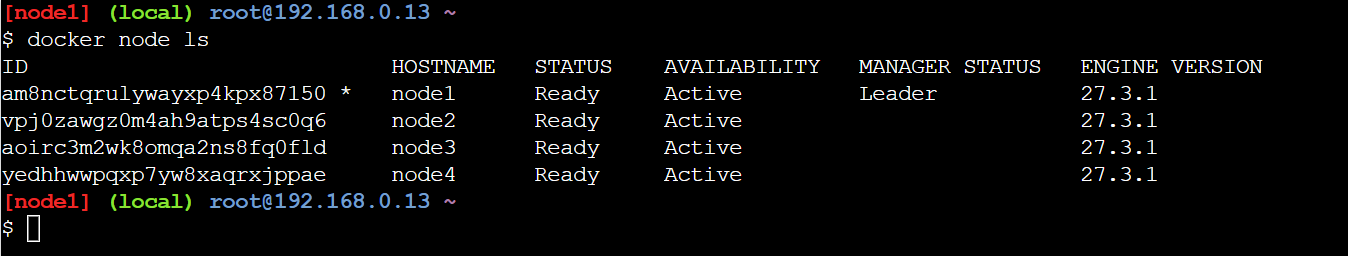


Docker node ls





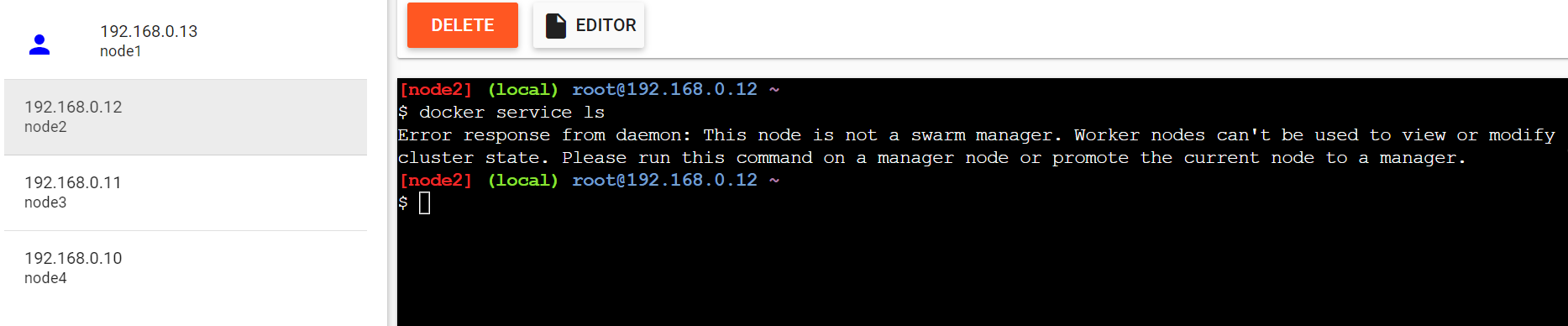




Docker Service

A docker service is a way of managing and deploying docker containers in a distributed environment,

such as a swarm of nodes. A docker service acts as a template for creating one or more tasks, which are instances of the service running on different nodes. A docker service can have various options, such as the number of replicas, the network, the port, the CPU and memory limits, and the rolling update policy. A docker service can be created, updated, scaled, removed, or inspected using the docker service.

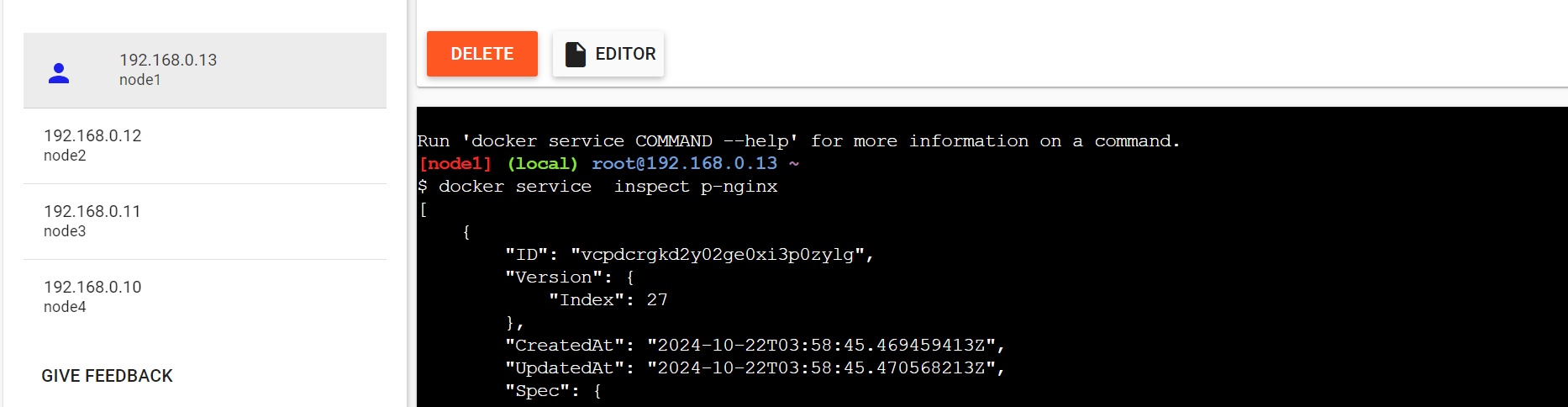


Docker service ls

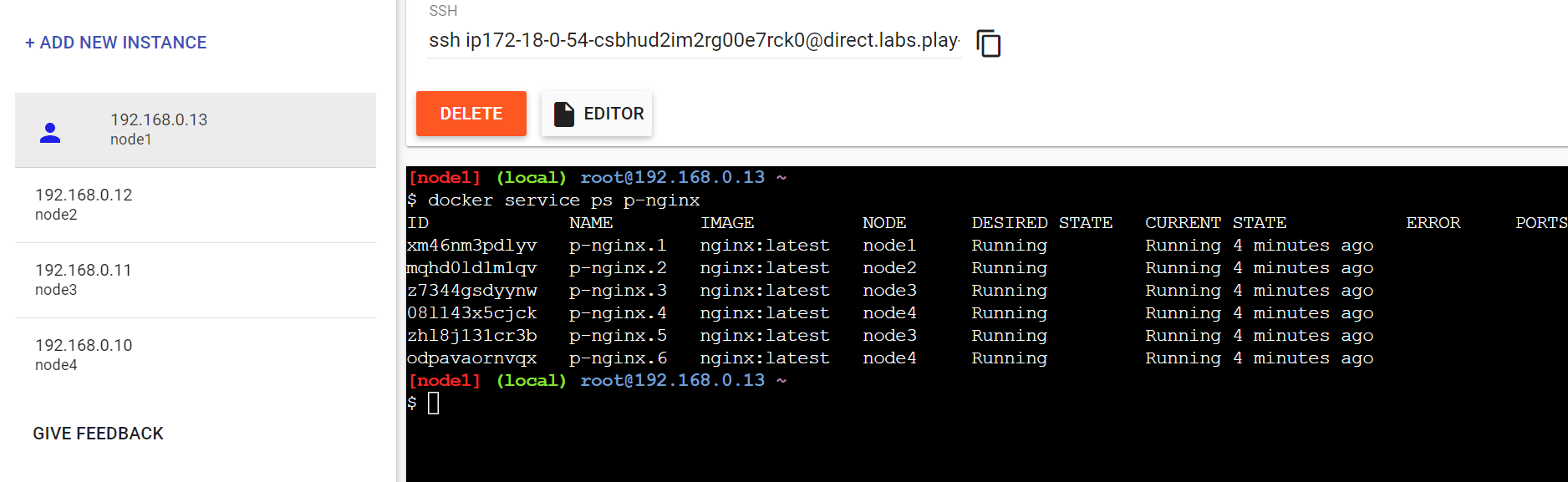
Docker service –help

docker service create --name p-nginx --replicas=6 -p 4444:80 nginx

Docker service inspect p-nginx



Docker service ps p-nginx

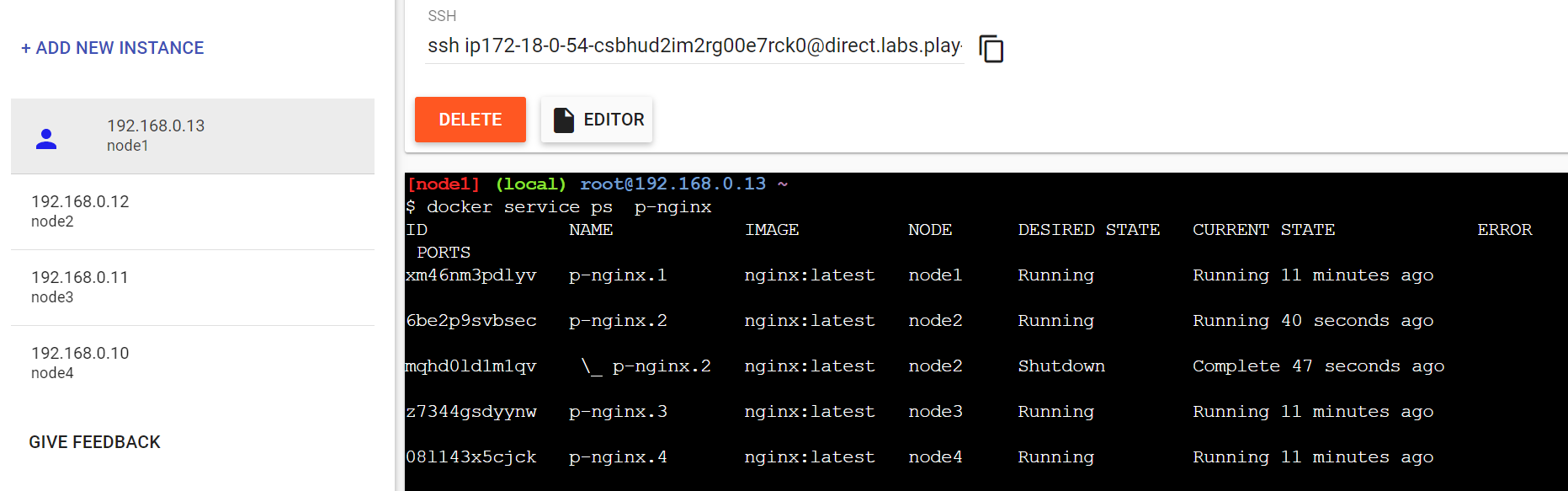




docker exec -it a-nginx bash echo "<h1>Hello A-NgInx Server </h1>" > /usr/share/nginx/html/index.html

stop container at any of the node





we can access container from any node

scaling(create 4 nodes)--no limit to scale

docker service scale p-nginx=2

docker service scale allows you to change the number of replicas for more than one service, compared to docker service update --replicas which allows you to change only one service at a time.

docker service scale p-nginx=2 p-nginx1=4

docker service rm my\_service