**Docker Network**

In Docker, a network allows containers to communicate with each other and with external resources. Docker networking provides various options and functionalities to manage and control the connectivity of containers.

**Types of Docker Networks:**

1. **Bridge Network:**

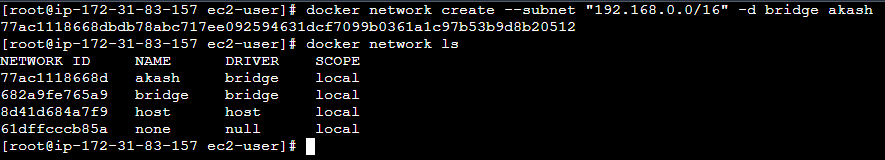
* The default network type.
* Containers connected to the same bridge network can communicate with each other using container names.
* Containers connected to different bridge networks cannot communicate by default.

1. **Host Network:**

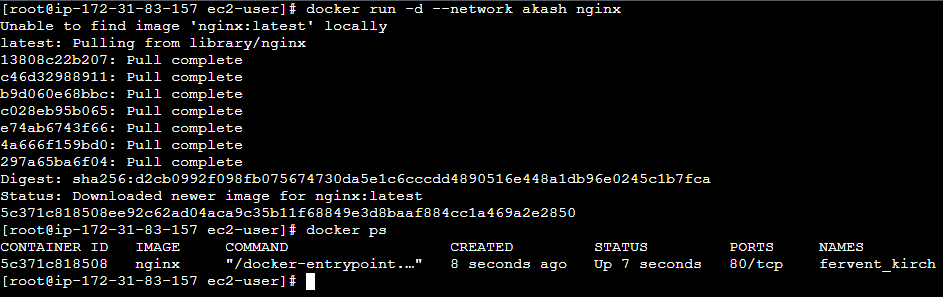
* Containers share the network stack with the Docker host.
* This means the container uses the host's IP address and ports.
* Best used when you want to remove network isolation between the container and the host.

1. **None Network:**

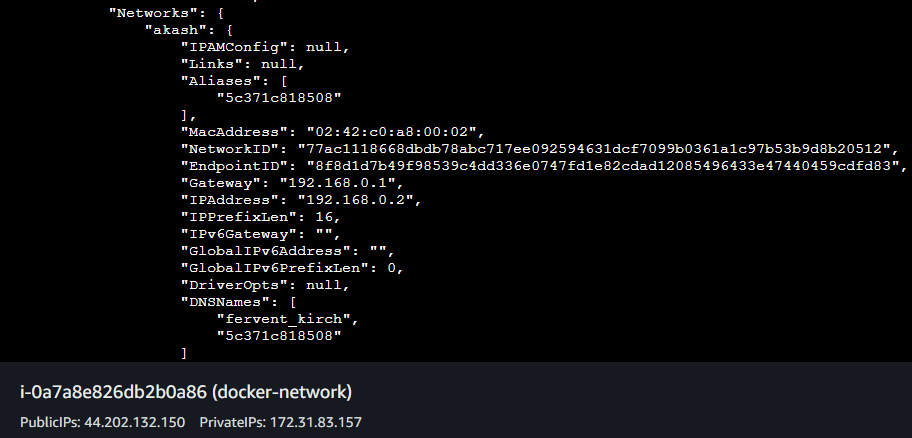
* The container will have no networking.
* This network type is used for isolating a container from any network.
* Creating a docker network.
* First we will start with creating a bridge network.
* Hit command “docker network create –subnet “192.168.0.0/16” -d bridge akash”.
* Here -d is the driver name and akash is the name for the network.
* Hit command “docker network ls” to check the networks.



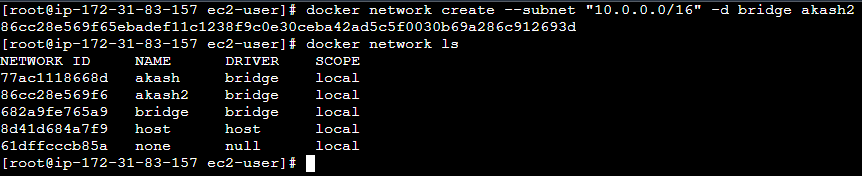
* Our network is created with name akash and driver bridge.
* Now we have to connect the network to the container.
* Hit command “docker run -d –network akash nginx”.
* This command will pull the nginx image if not available locally and run in detached mode and assign the network akash to it.
* Hit command “docker ps” to check the container is running or not.



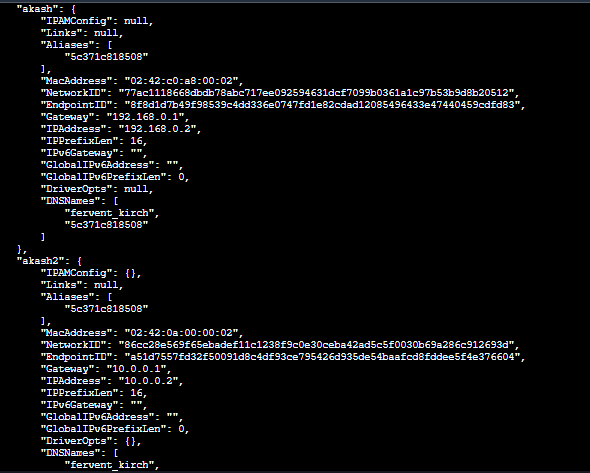
* Now the container is running we need to inspect the container to check which network is assigned to the container.
* Hit command “docker inspect container\_id”.



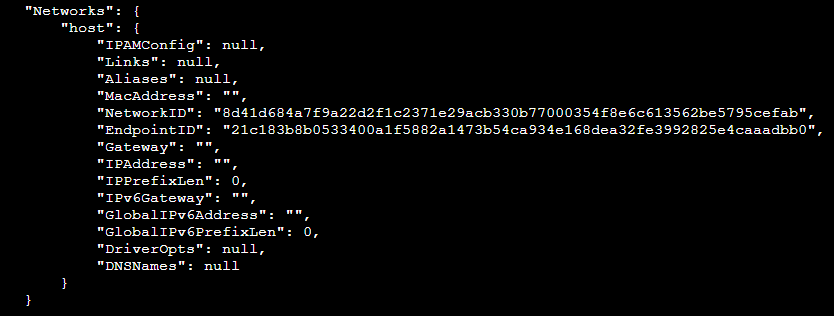
* We can also assign multiple networks to the container.
* We will now create another bridge network.
* Hit command “docker network create --subnet “10.0.0.0/16” -d bridge akash2”
* Hit command “docker network ls” to list all the networks.



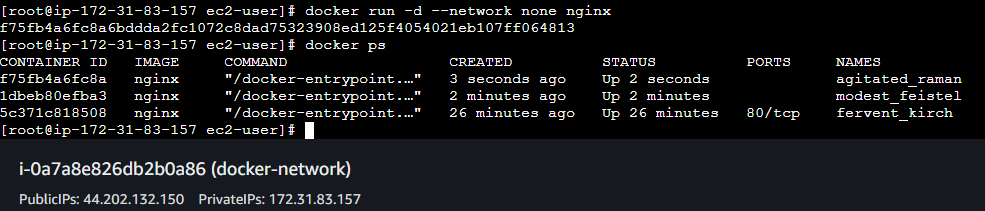
* Now we need to connect the network with our container.
* For that we need network\_id and containe\_id.
* Hit command “docker ps” to see the container id.
* Hit command “docker network connect network\_id container\_id”.



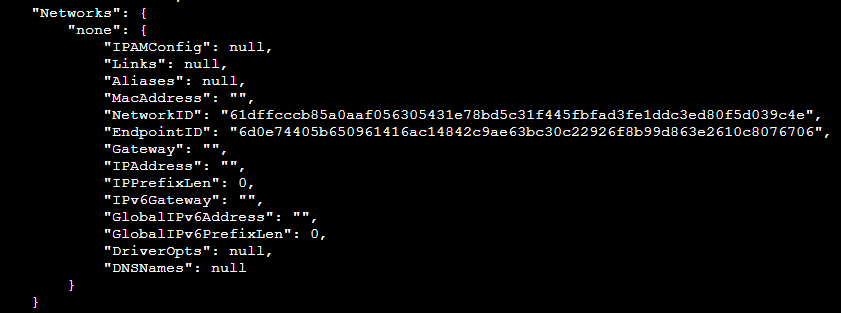
* We have attached two networks to the container.
* Now we will move forward with the host network.
* We cannot create a host network explicitly we can directly assign the host network to the container.
* Hit command “docker run -d --network host nginx”.
* Hit command “docker ps” to check the running containers.
* Hit command “docker inspect container\_id”.



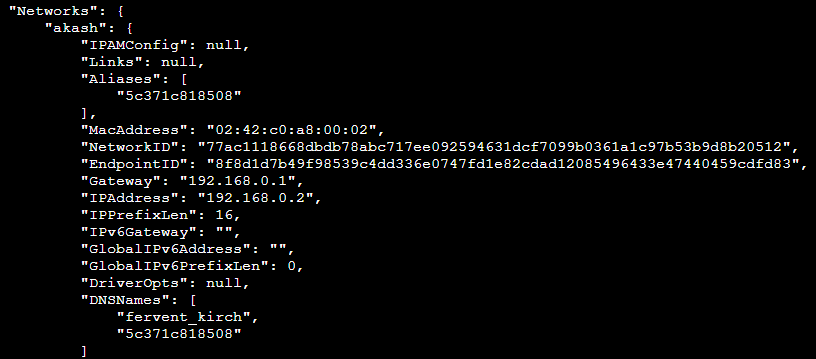
* We can also give the null network i.e if we don’t want to give network to the container then we can use null network.



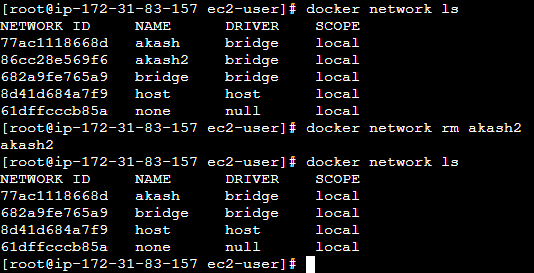
* Hit command “docker inspect container\_id”.



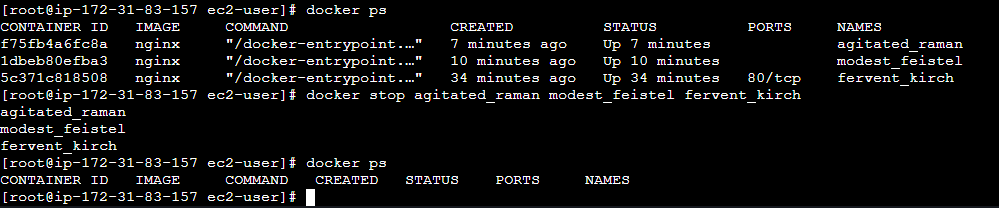
* We can also disconnect the network assigned to the container.
* Hit command “docker network disconnect network\_id container\_id”.
* Now we will disconnect the network akash2.
* Hit command “docker inspect container\_id”.



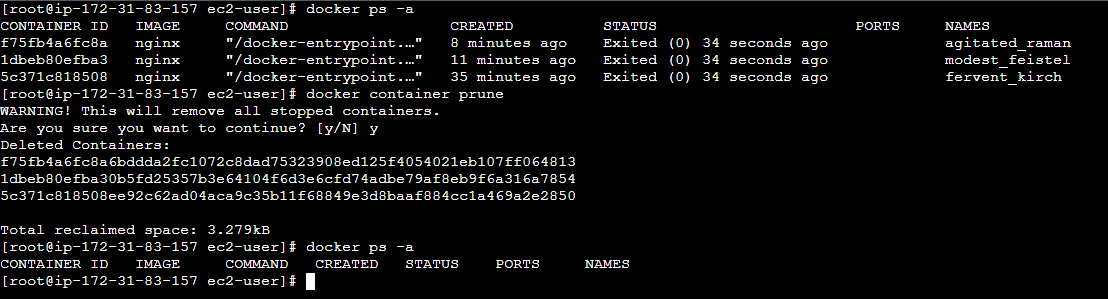
* Now the network akash2 is disconnected.
* We can also remove the network.
* Hit command “docker network ls” to list all the networks.
* Hit command “docker rm network\_id”



* Also there is a command prune which removes all the unused networks.
* First we will stop all the containers.
* Hit command “docker stop container\_id..”



* Hit command “docker container prune”
* This command will remove all the stopped containers.



* Now hit command “docker network prune”
* This command will remove all the unused netwoks.

