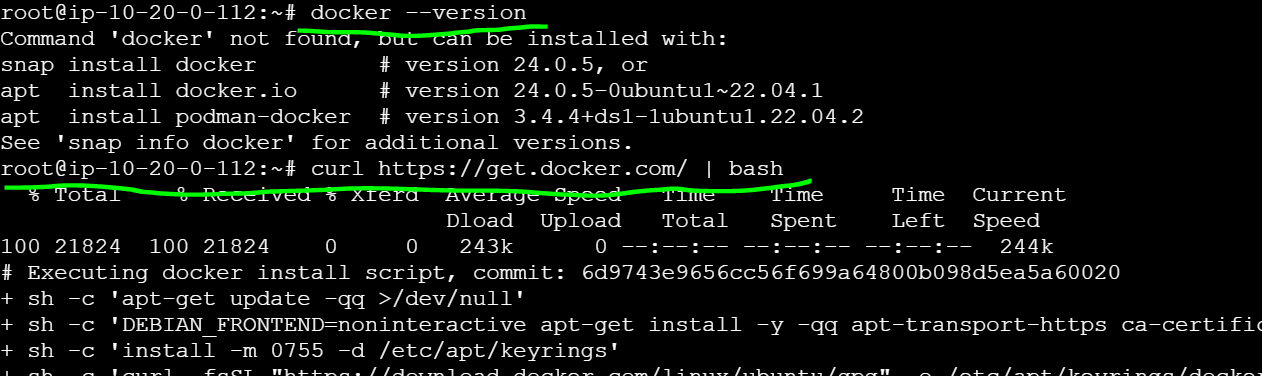
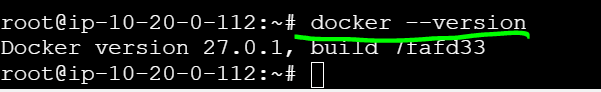
#Install tools

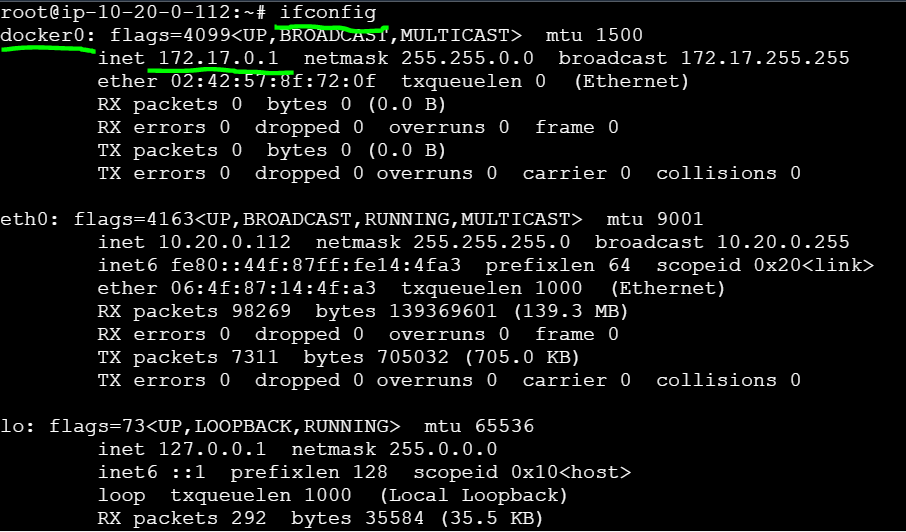
apt update && apt install jq net-tools

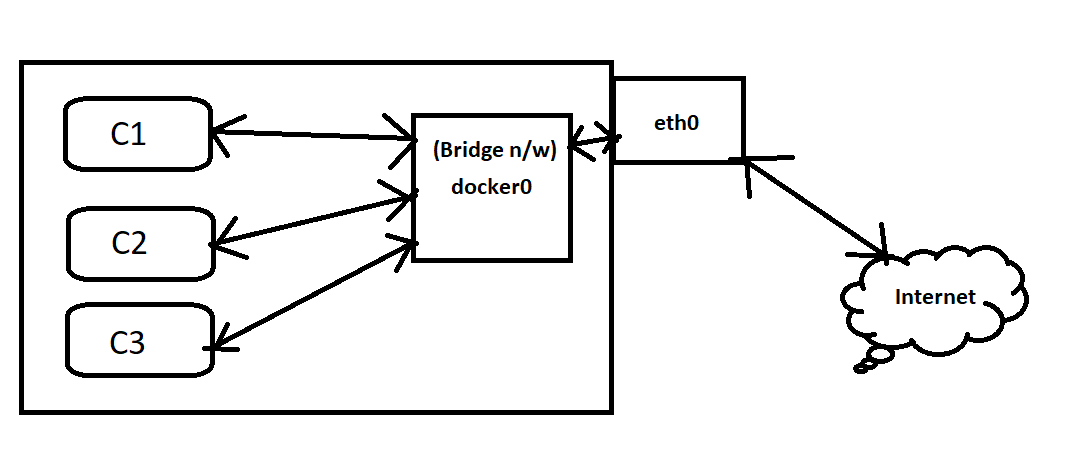
#Docker Installation

curl https://get.docker.com/ |bash

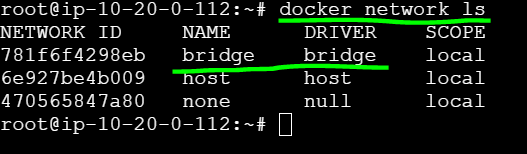


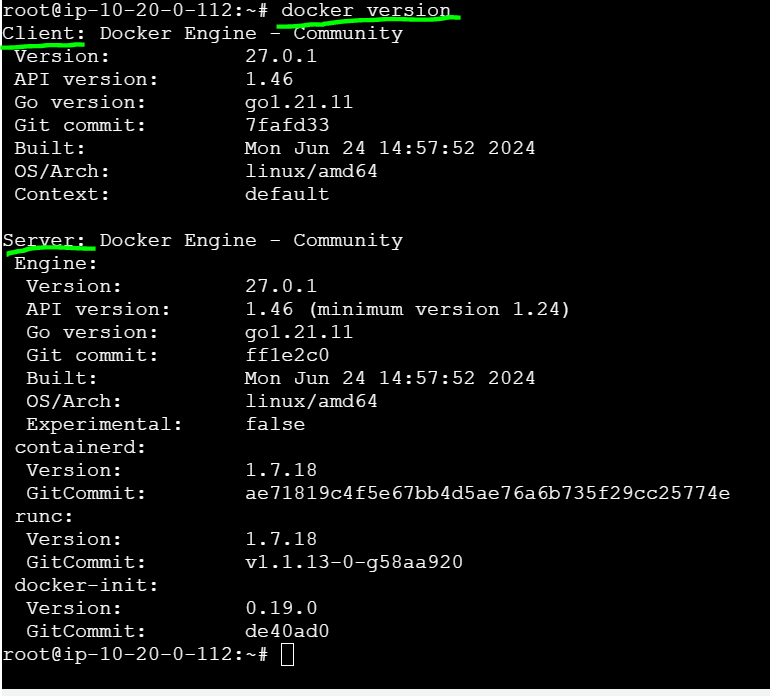


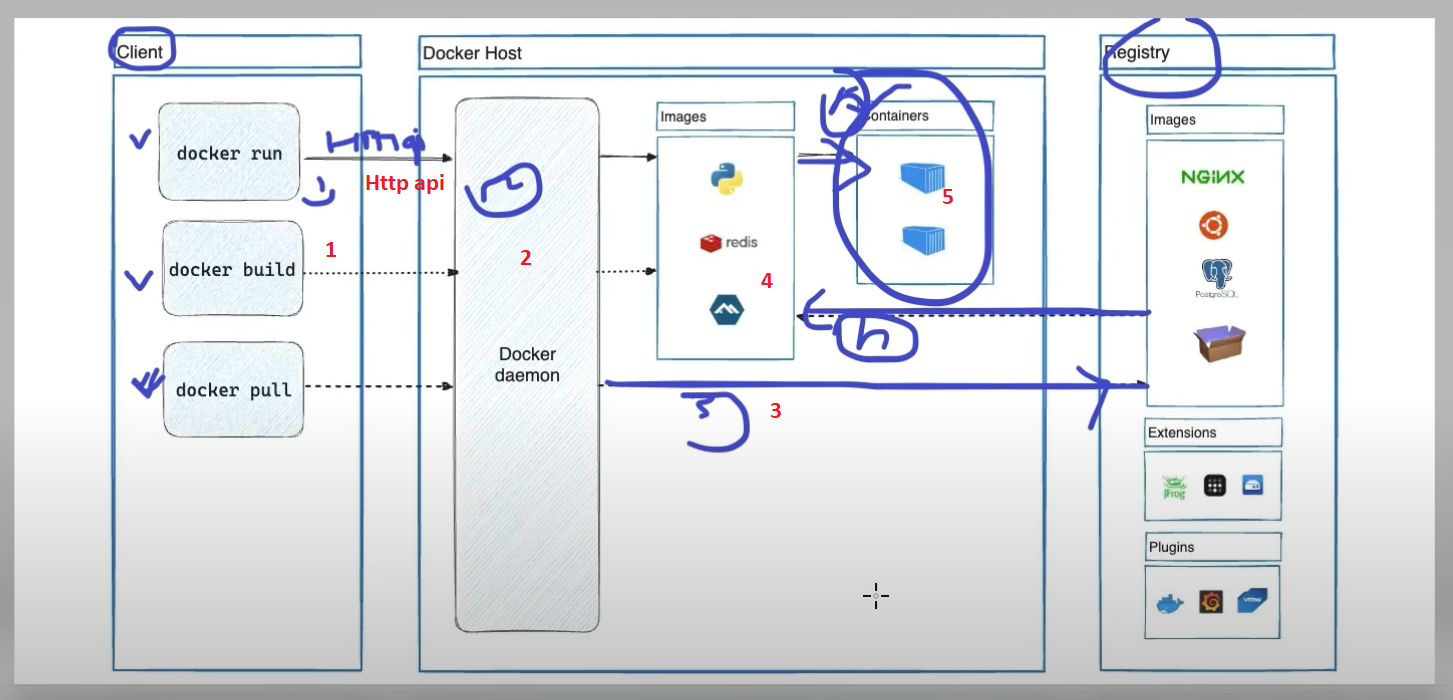




**Docker0 (172.17.0.1) acts as a Bridge N/W between containers & eth0**







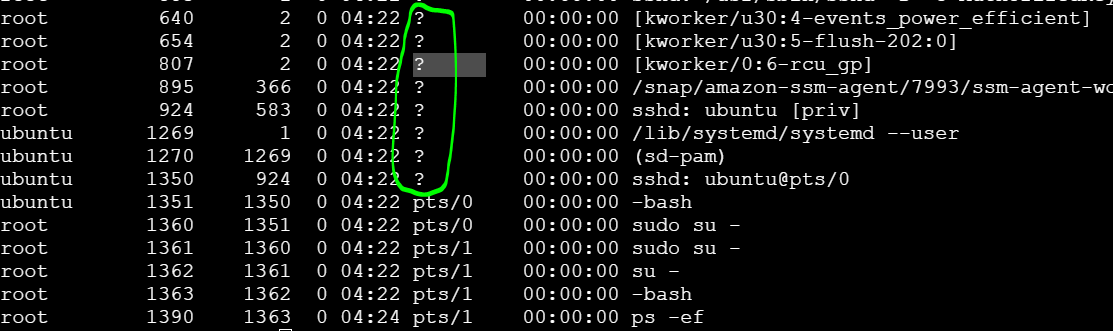
**Process of creating containers**

Docker Run / Build / Pull 🡪 Docker daemon 🡪Registry 🡪 image download 🡪 Container

Docker daemon -- > is a process ( ps -ef you will get the results with ? )

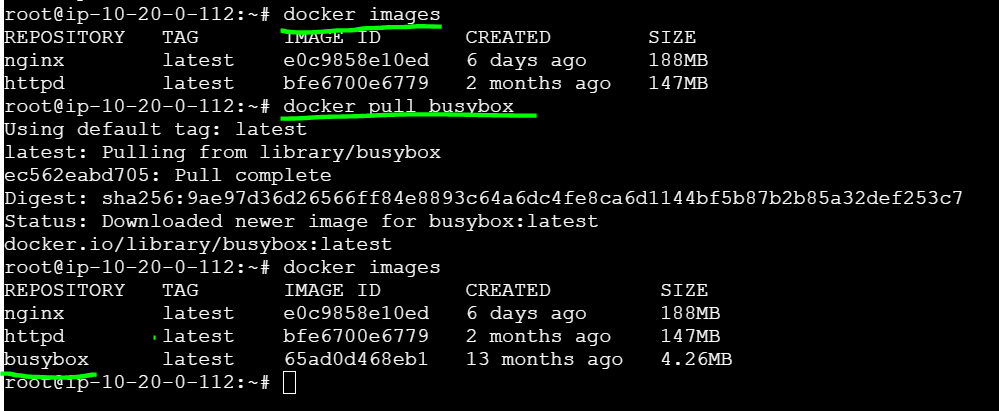
? 🡪 Docker daemon

pts/1 🡪 user



**Downloading images**

Docker pull nginx/busybox/httpd

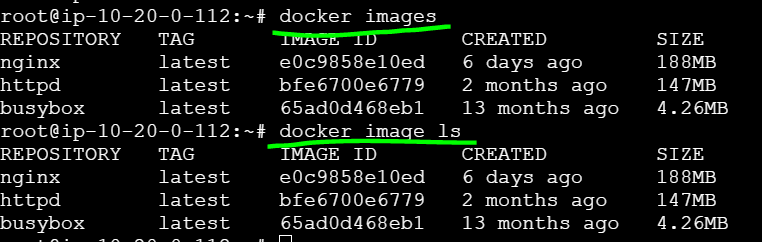


#list available images

docker images



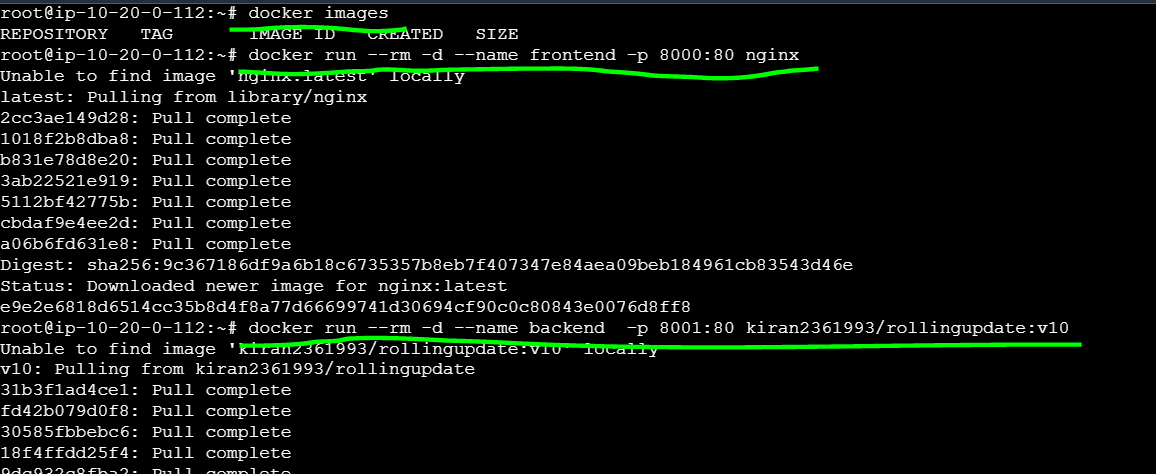


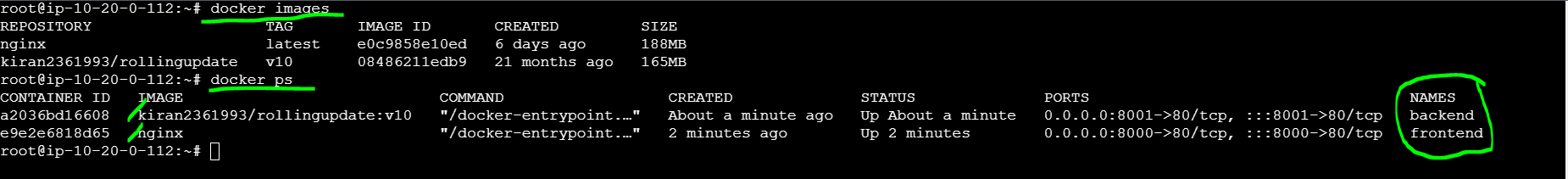


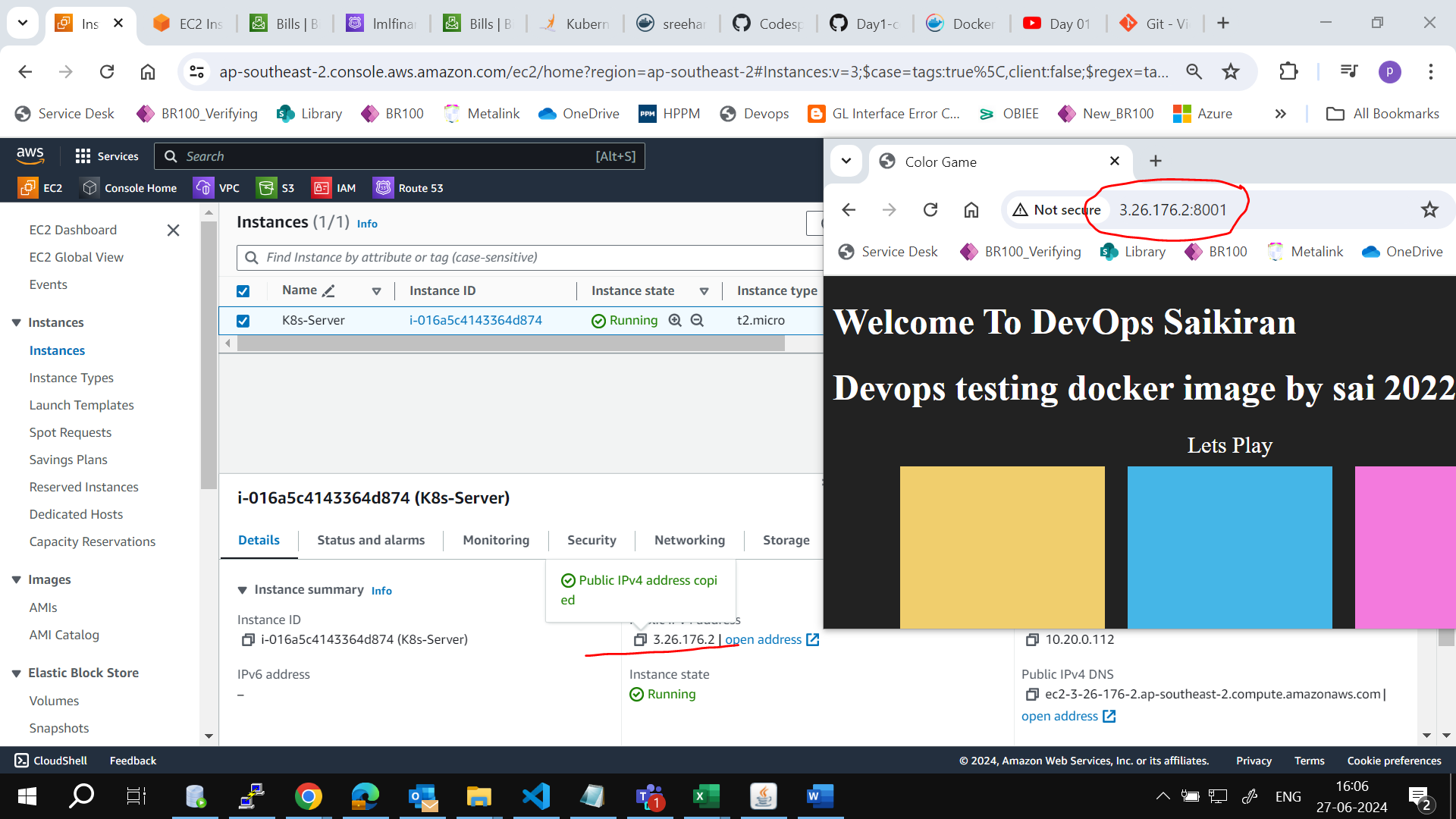
Port-forwarding

docker run --rm -d --name frontend -p 8000:80 nginx

docker run --rm -d --name backend -p 8001:80 kiran2361993/rollingupdate:v10







#containers path

cd /var/lib/docker/containers/

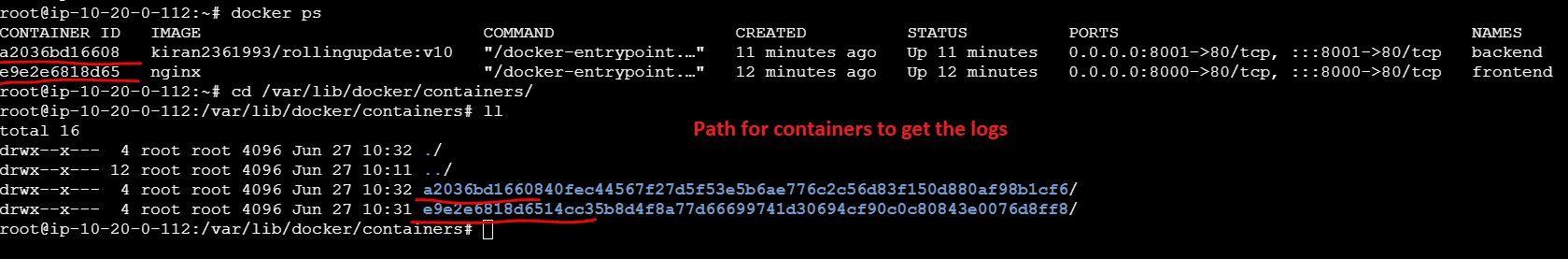
#Once containers are created login to it & get the logs

cd /var/lib/docker/containers/<container\_id>

ll

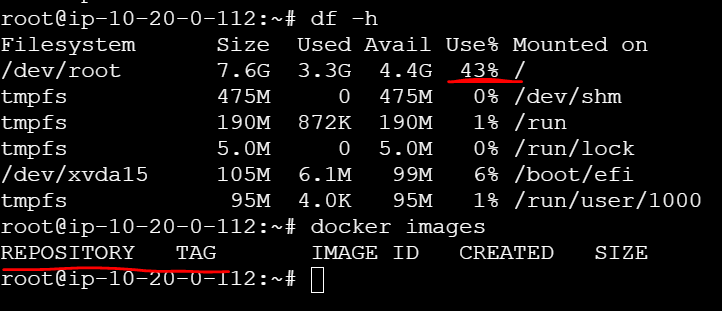
cat a2036bd1660840fec44567f27d5f53e5b6ae776c2c56d83f150d880af98b1cf6-json.log | jq

cat a2036bd1660840fec44567f27d5f53e5b6ae776c2c56d83f150d880af98b1cf6-json.log | jq > INC01478528.log



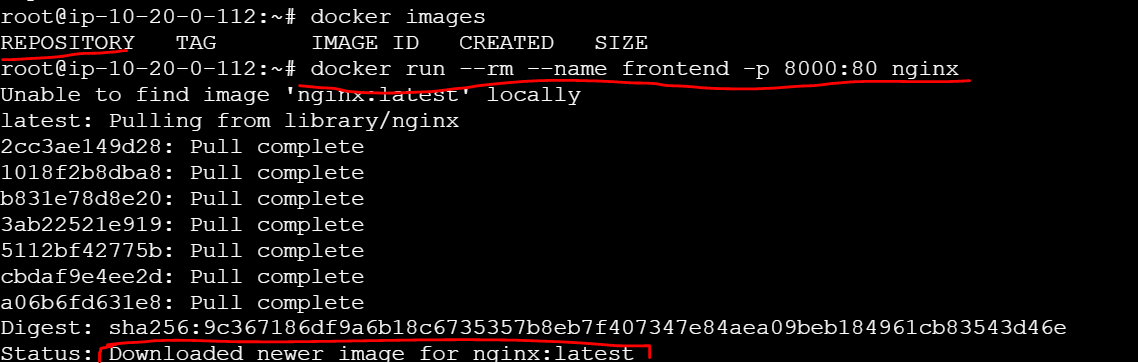
Day 02 Docker Dir Change | Custom Network | Host | AWS EBS Volume |

**Before without images – size usage of the root volume is 43%**

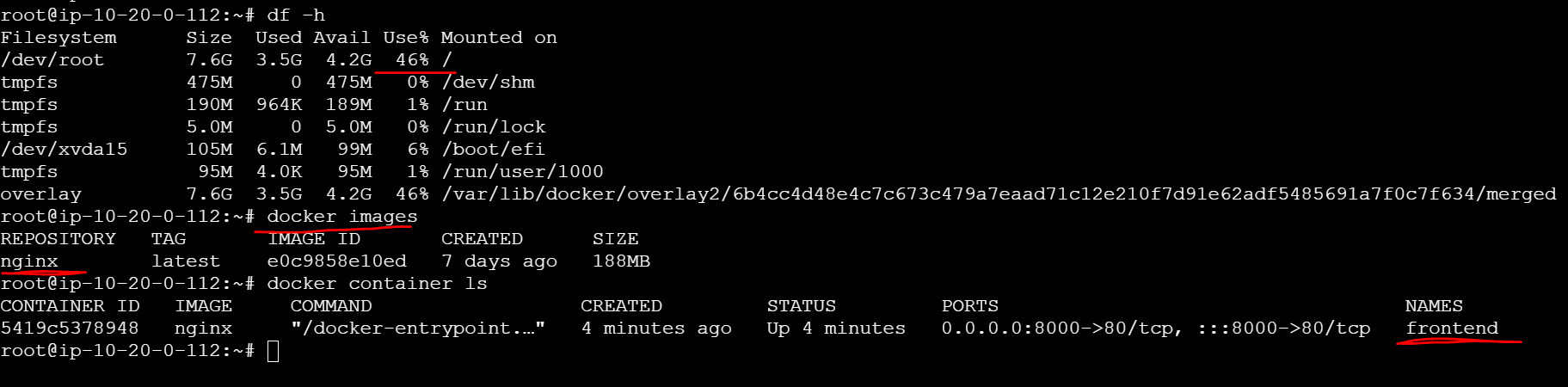


Download image

docker run --rm --name frontend -p 8000:80 nginx



**After downloading the images -- size usage of the root volume is 46%**



#check the current root volume usage  (43%)

df -h

#download image/create a container & check the current root volume usage (46%)

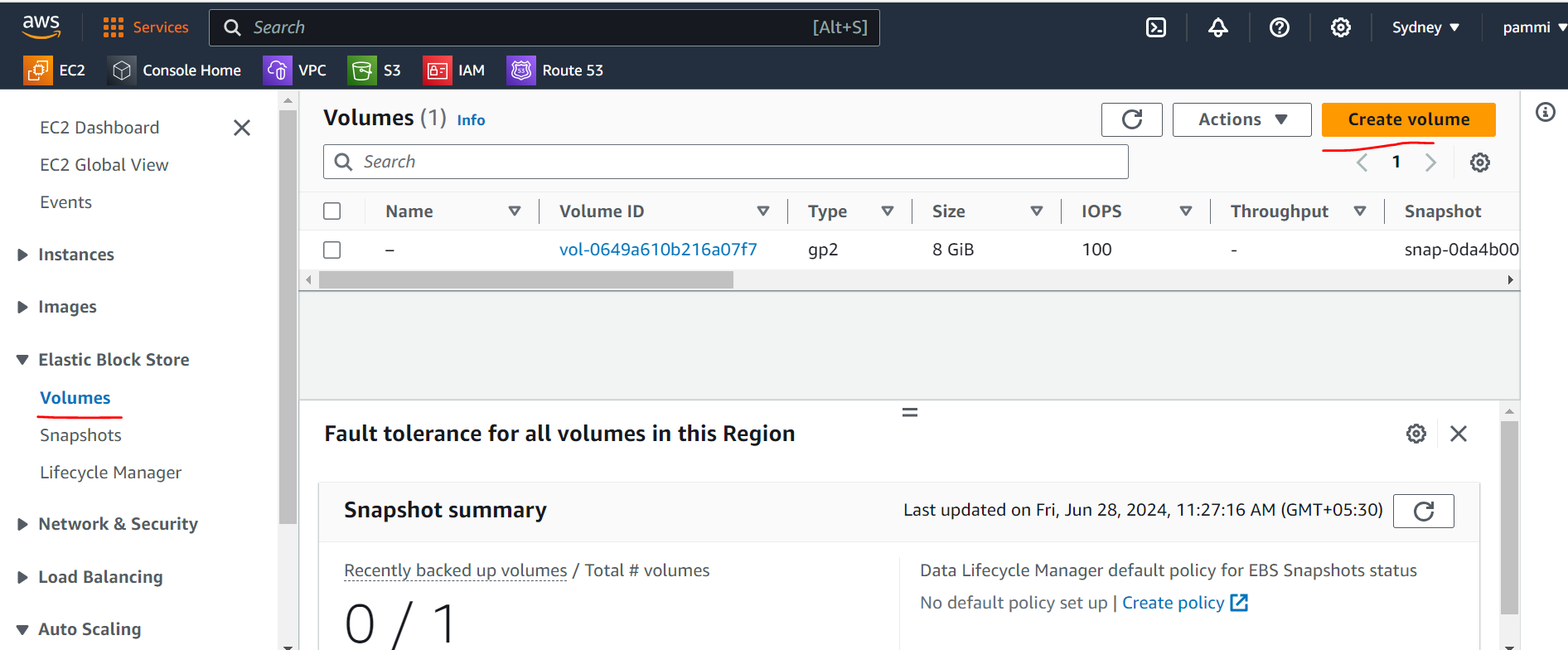
docker run --rm --name frontend -p 8000:80 nginx

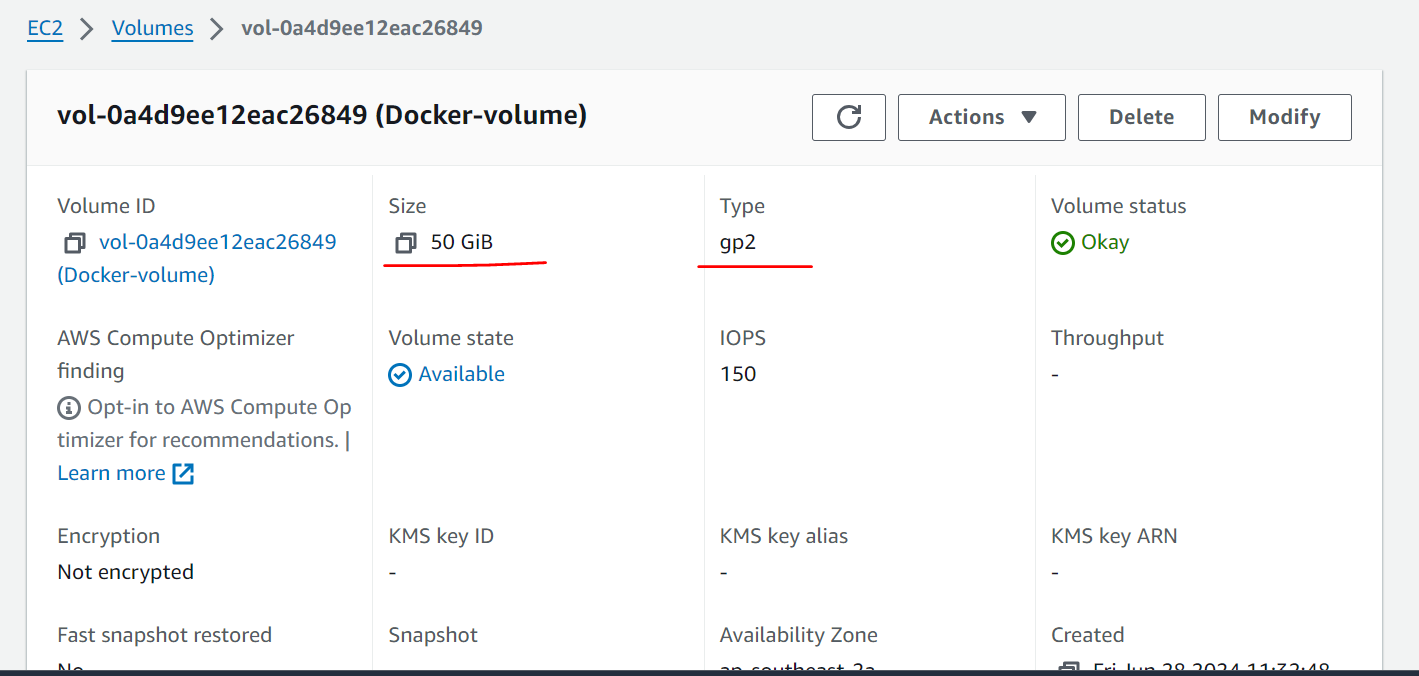
df -h

#if you keep on downloading images/creating containers the root volume reaches to 100%

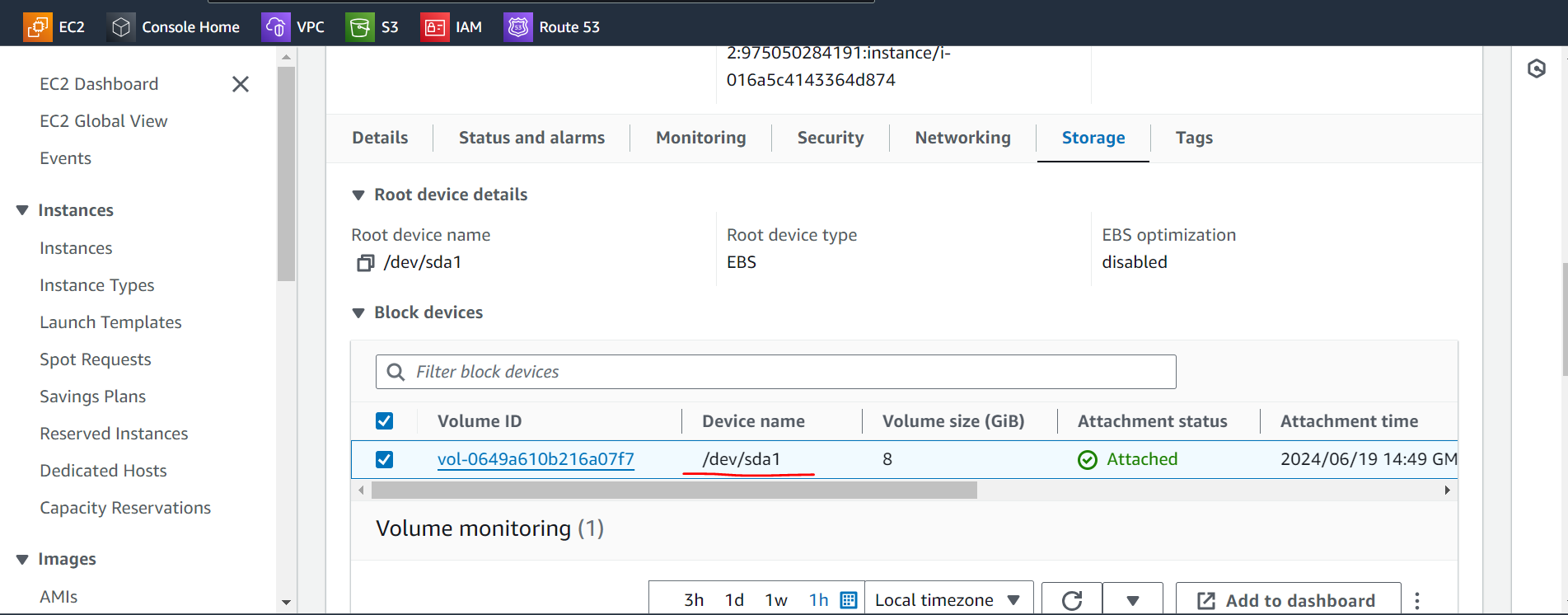
**#Process to change the docker directory**

**Volume creation**

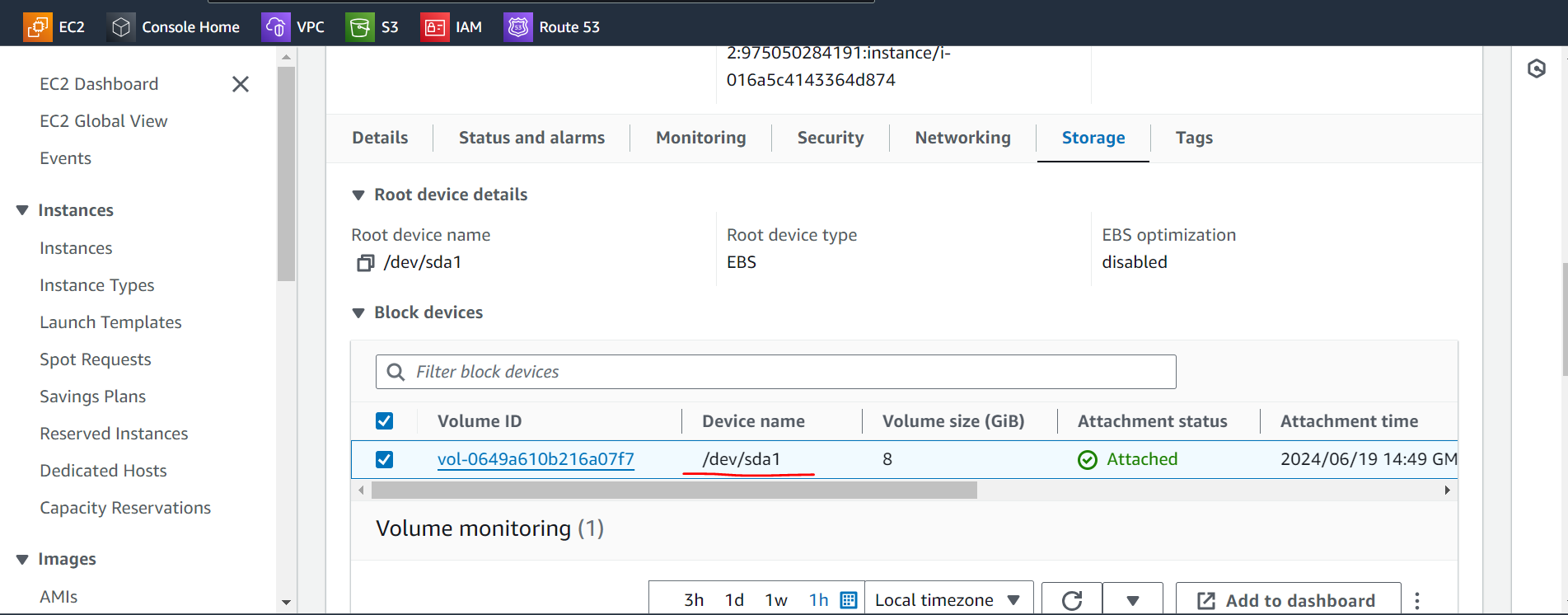




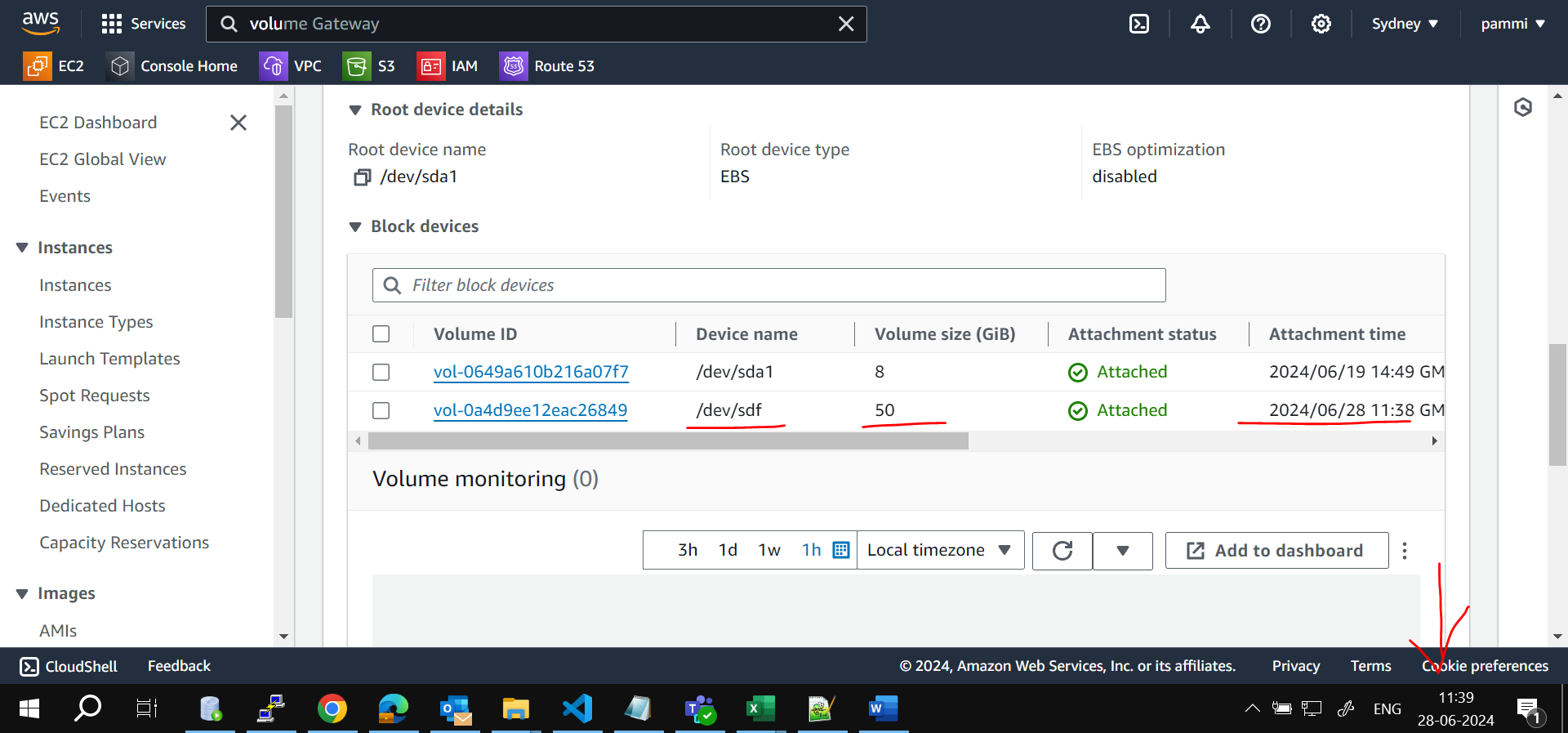
**Attach newly created volume to the instance.**



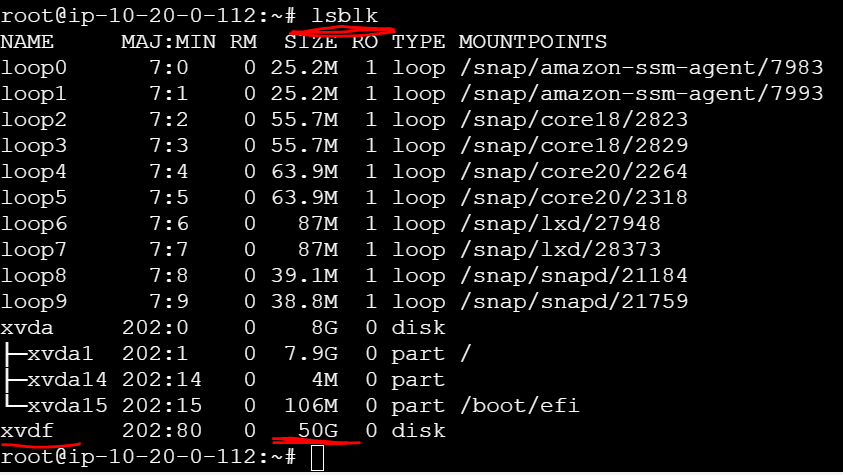
**Before Attaching volume to an instance**



**After attaching a volume to an instance**

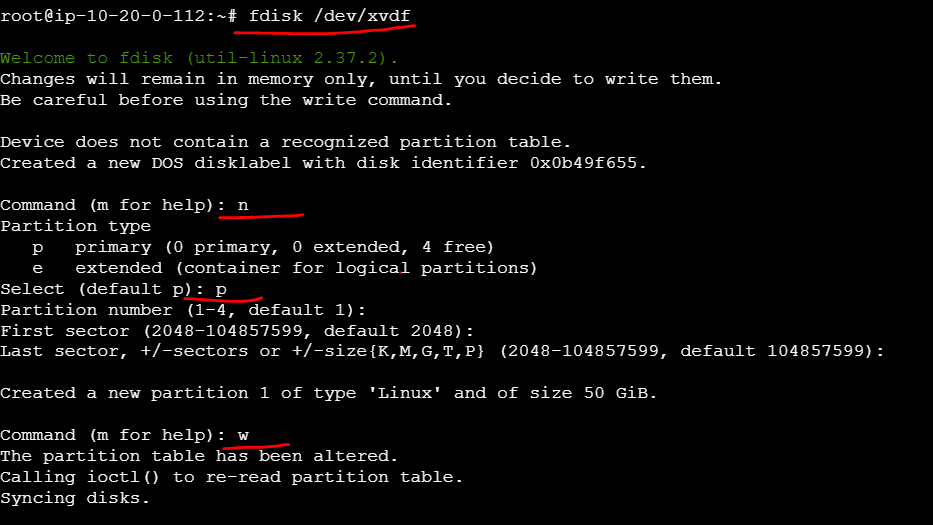


**Validate new volume in linux**

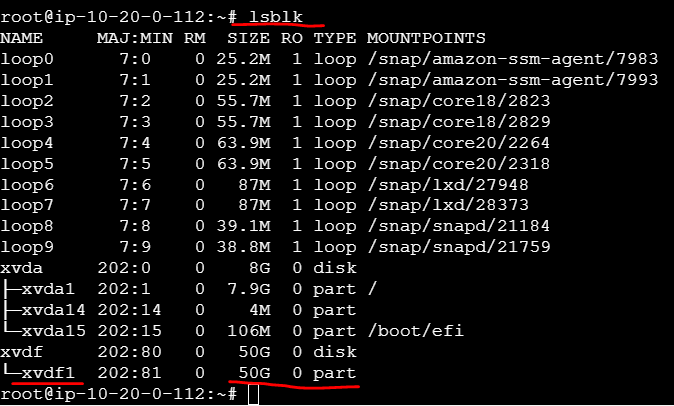


**Format the newly created volume (xvdf)**

**fdisk /dev/xvdf**



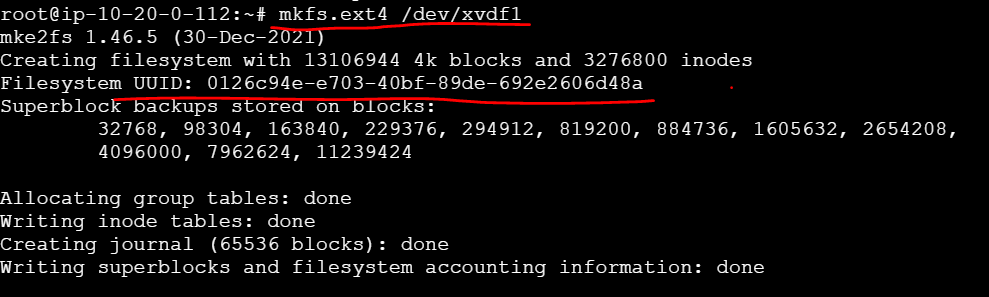
**Validate formatted volume.**



**Create a file system for the newly created volume (/dev/xvdf1)**

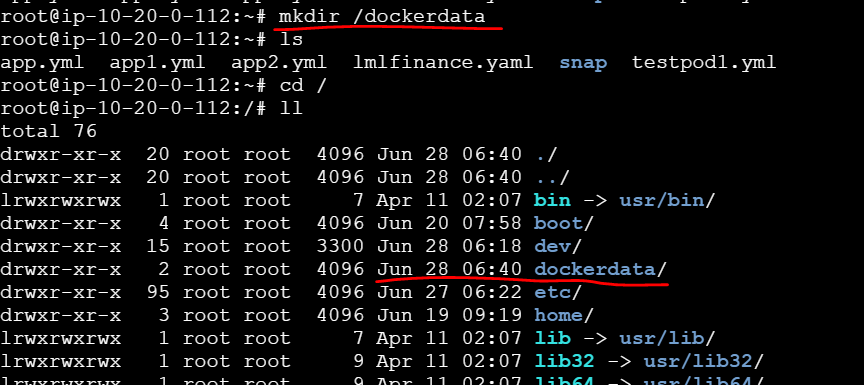
**mkfs.ext4 /dev/xvdf1**

**copy Filesystem UUID: 0126c94e-e703-40bf-89de-692e2606d48a**



**Create directory**

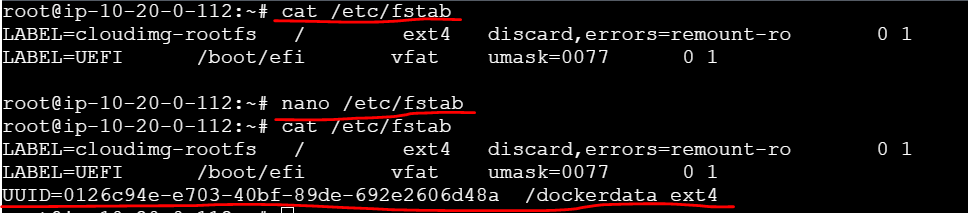
**mkdir /dockerdata**



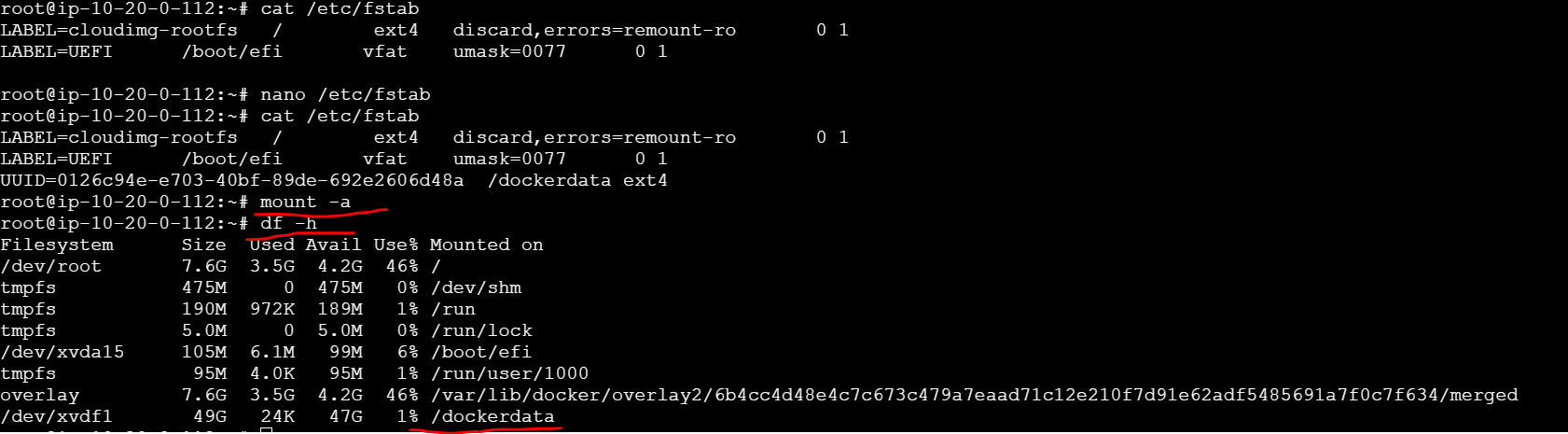
**Add Filesystem UUID: 0126c94e-e703-40bf-89de-692e2606d48a into the below path**

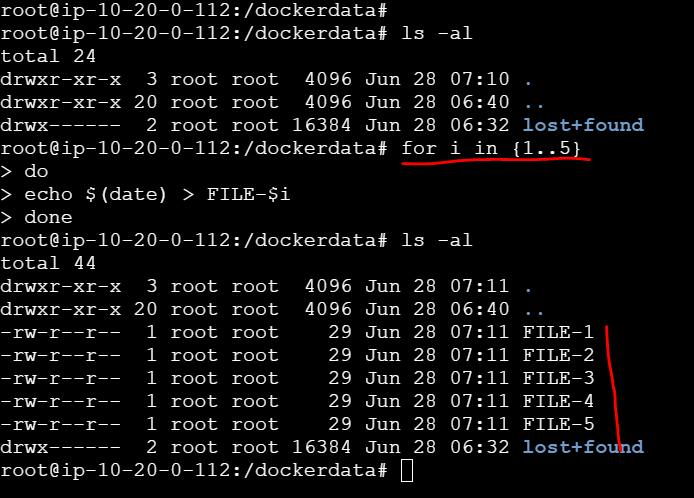
**nano /etc/fstab**

UUID=0126c94e-e703-40bf-89de-692e2606d48a  /dockerdata ext4



**mount -a**





**Change service directory file**

**Create additional EBS Volume , fdisk and mount it.**

**sudo systemctl stop docker.service**

**sudo systemctl stop docker.socket**

**sudo nano /lib/systemd/system/docker.service Add the following line with the custom directo**

**#ExecStart=/usr/bin/dockerd -H fd:// -- containerd=/run/containerd/containerd.sock**

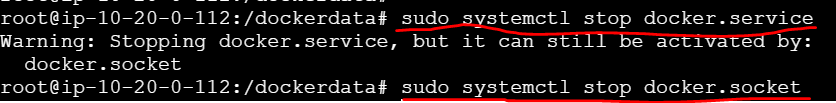
**# ExecStart=/usr/bin/dockerd --data-root /dockerdata -H fd:// -- containerd=/run/containerd/containerd. sock**

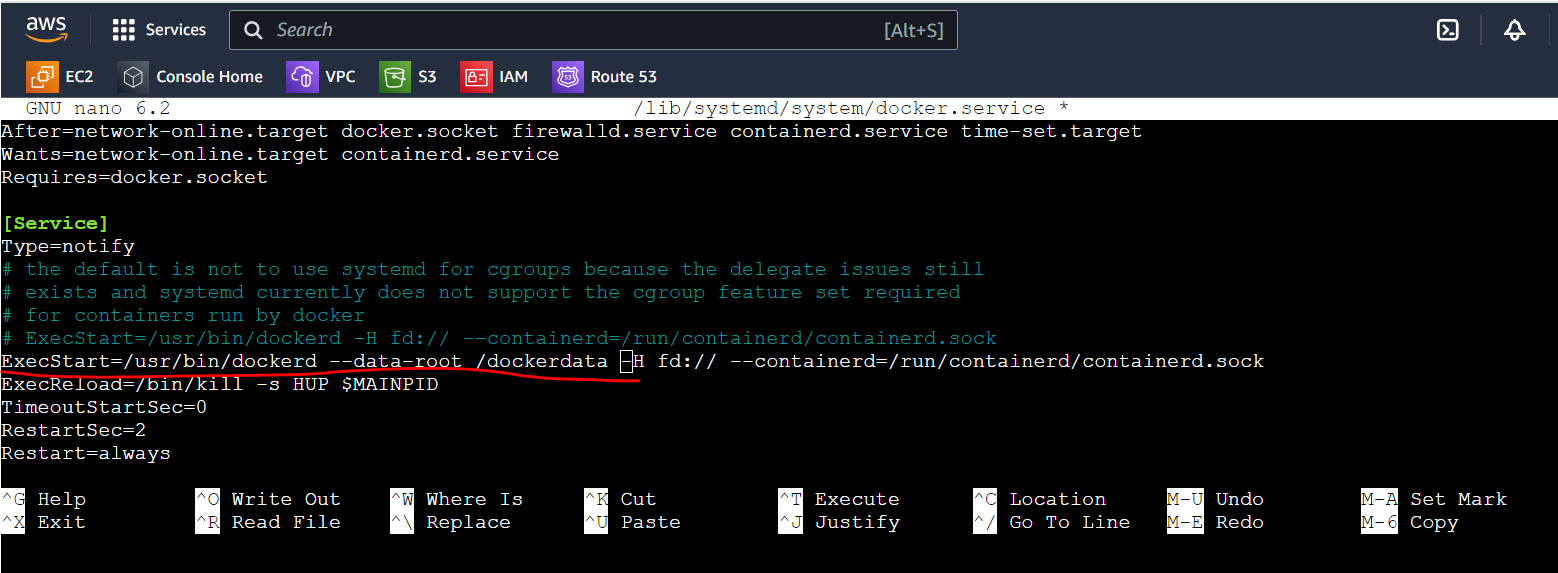
**sudo rsync -aqxP /var/lib/docker/ /dockerdata**

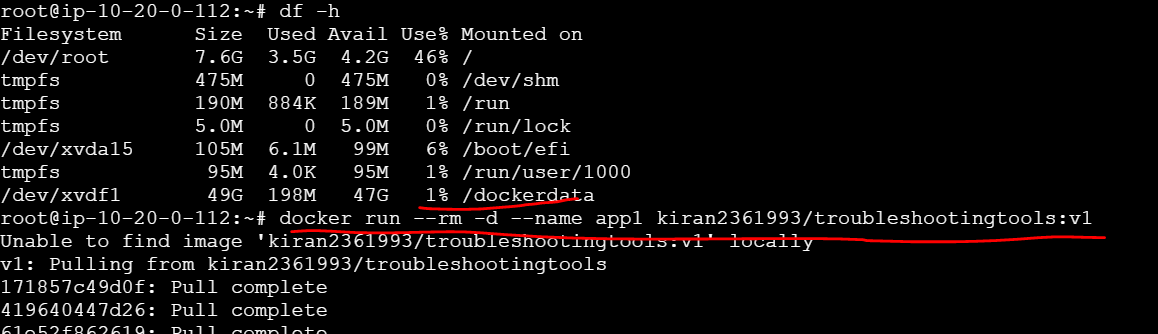
**sudo systemctl daemon-reload && sudo systemctl start docker**

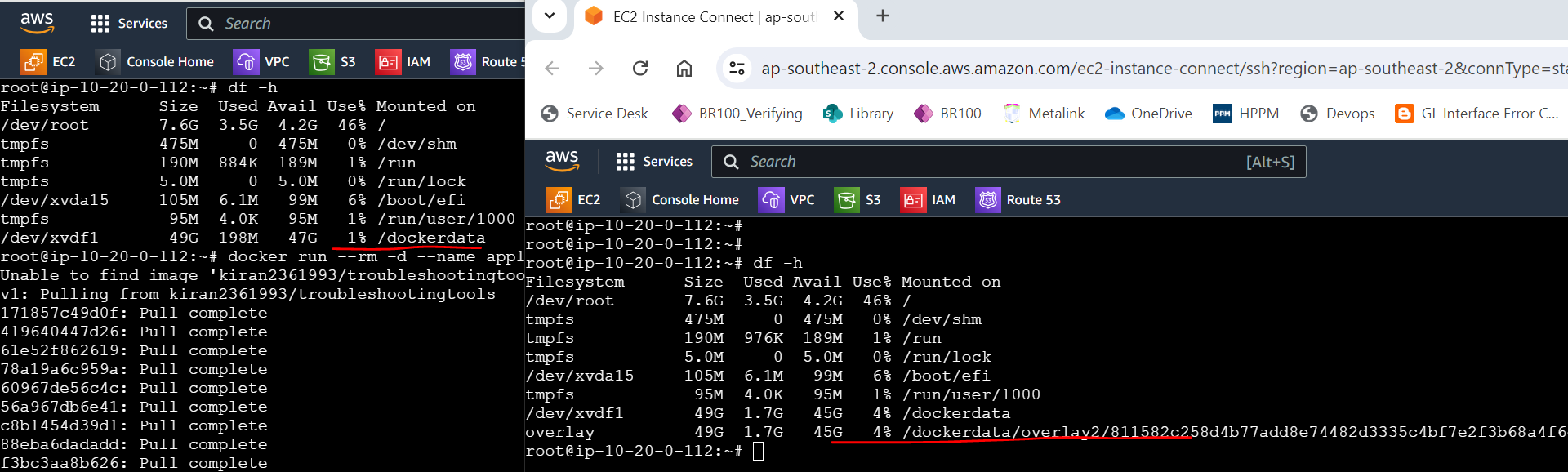
**sudo systemctl status docker --no-pager**

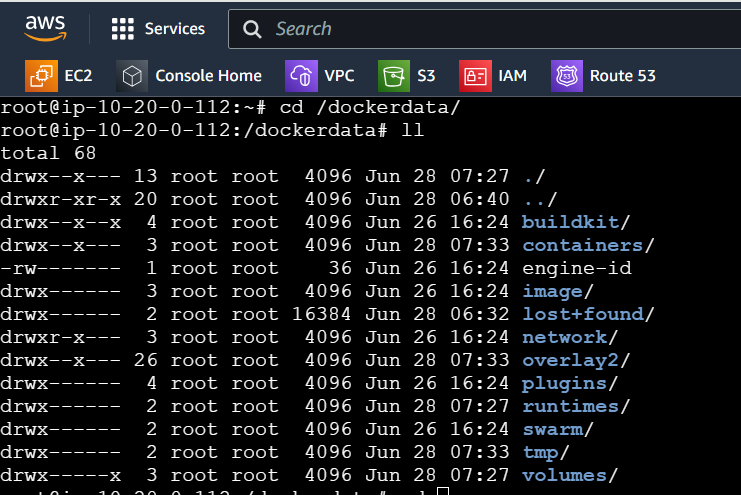
**ps aux | grep -i docker | grep -v grep**

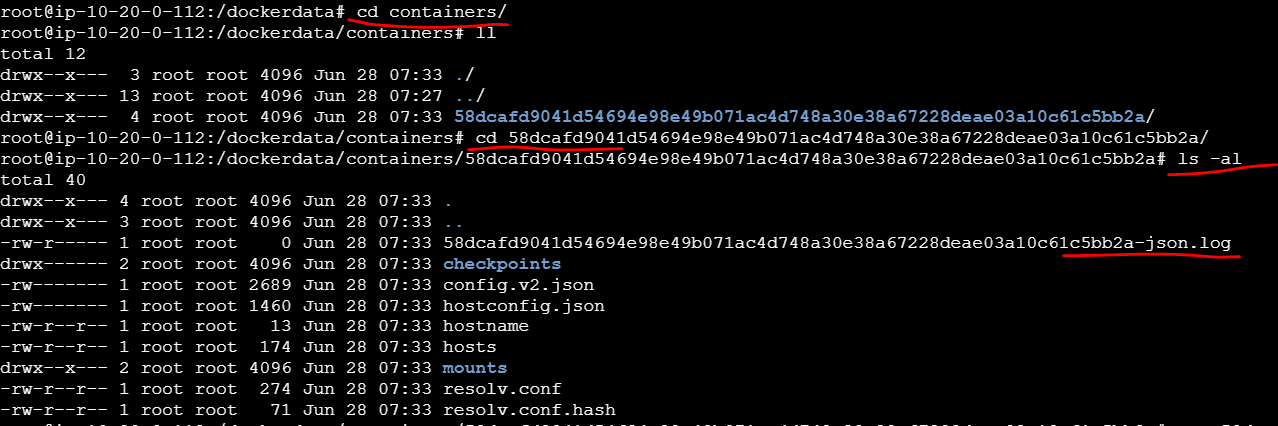








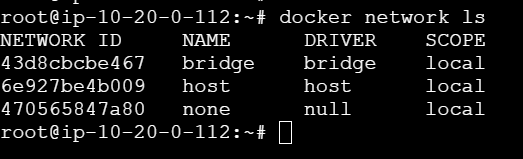




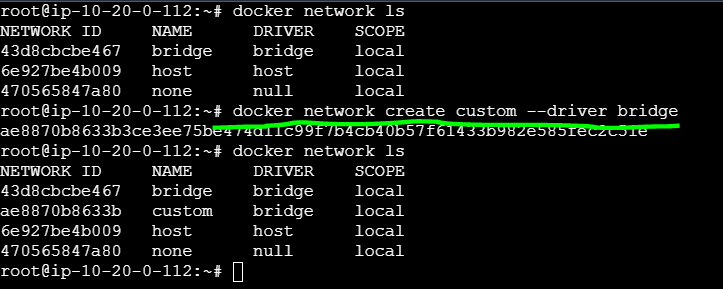
**Docker Network**

#Docker Network

docker network ls

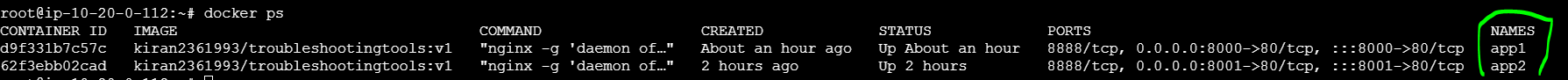


**Custom network creation**

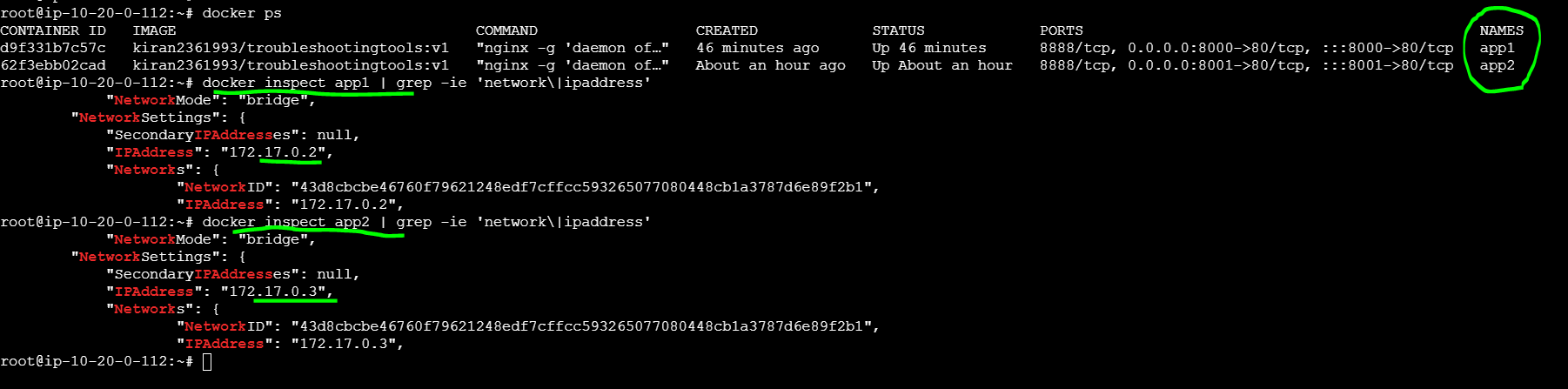


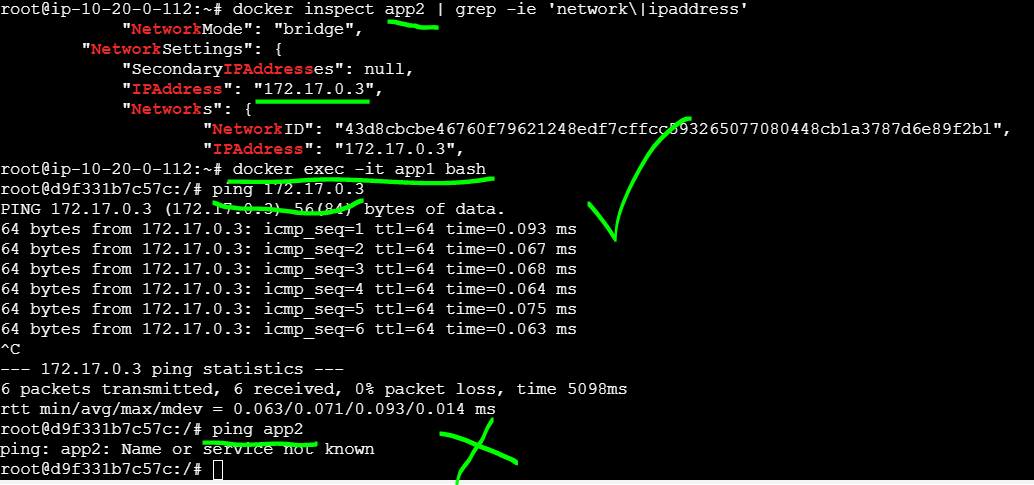
**Check the running containers & get the ipaddress & network details**

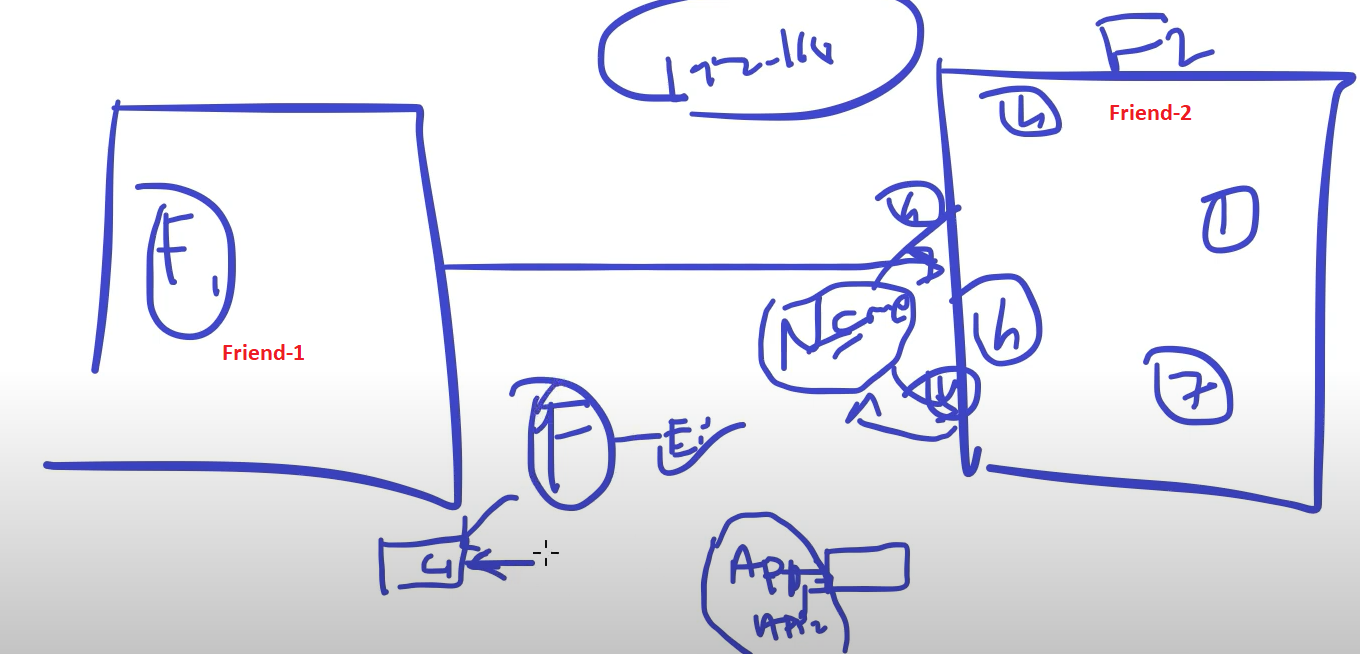
Docker ps



docker inspect app1 | grep -ie 'network\|ipaddress'

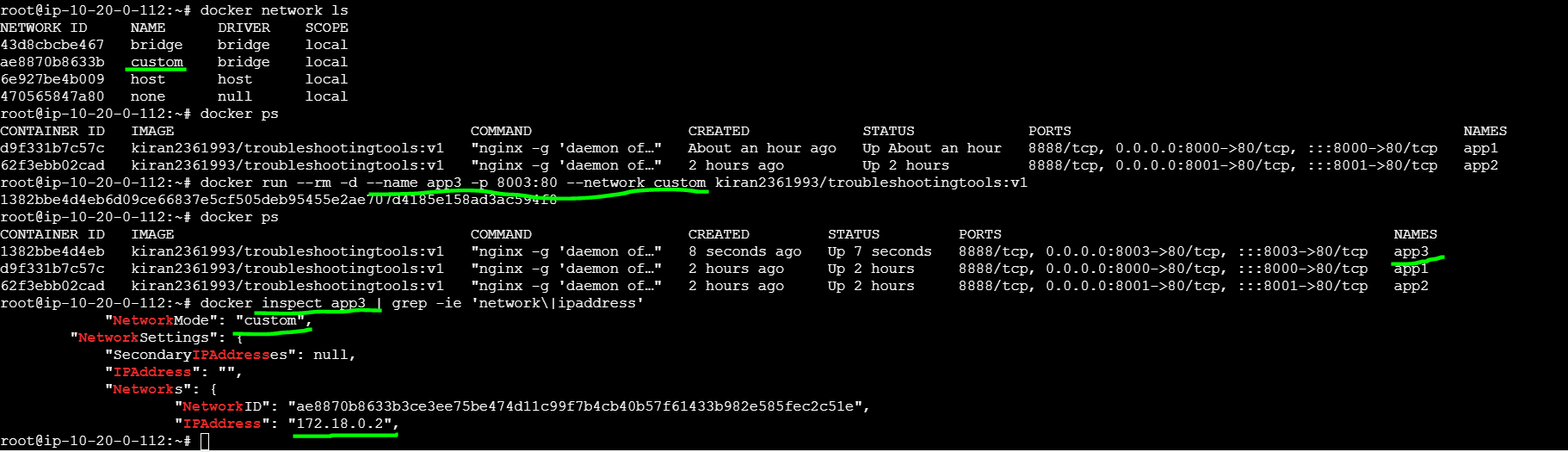




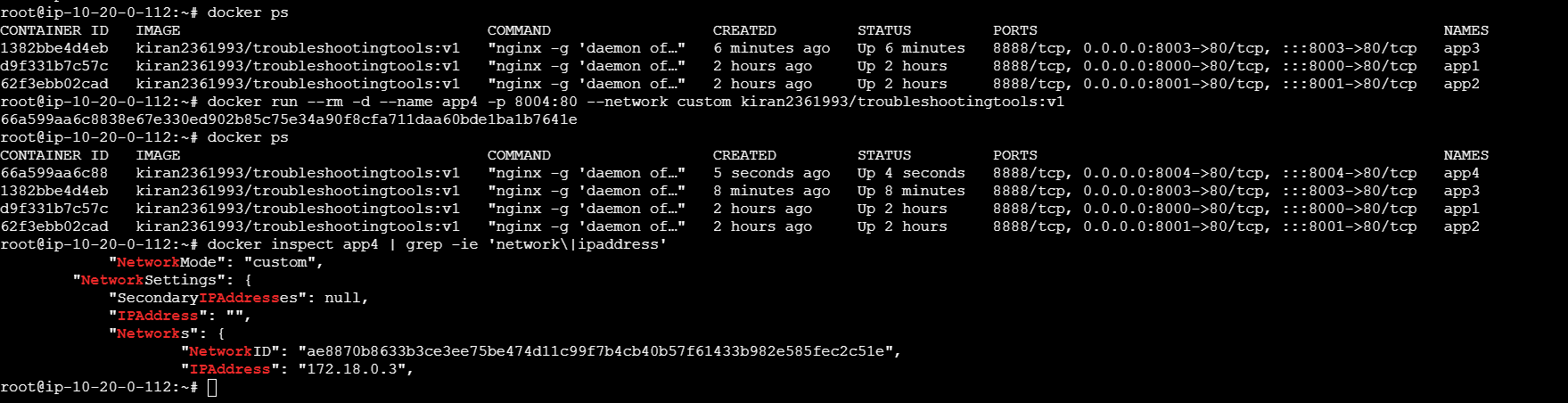


**Create container with custom network**

**docker inspect app3 | grep -ie 'network\|ipaddress'**



**docker inspect app4 | grep -ie 'network\|ipaddress'**



**Created 2 custom networks & 2 bridge networks**

