Deployments in swarm

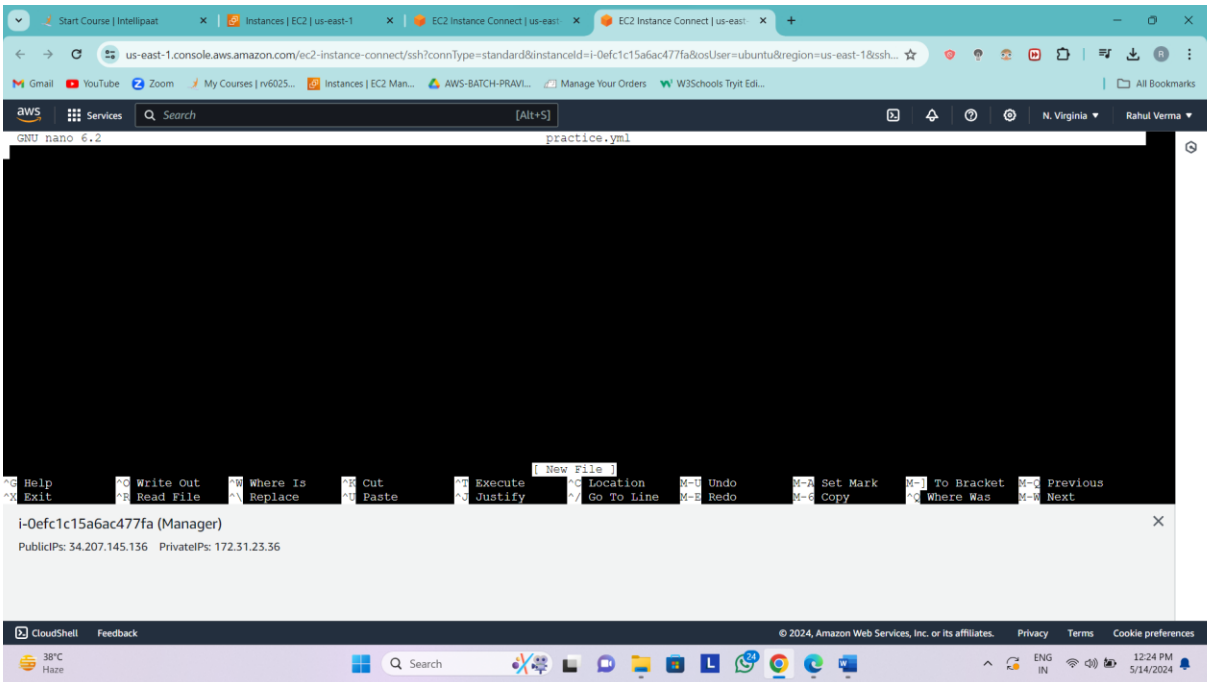
We will use Docker stack deploy

Docker stack function make use of YAML file to deploy multiple services at once.

1. Create one .yml file

nano <file_name>.yml

e.g- nano practice.yml



```
GNU nano 6.2                              practice.yml

                                          [ New File ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket M-Q Previous
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was   M-W Next
```

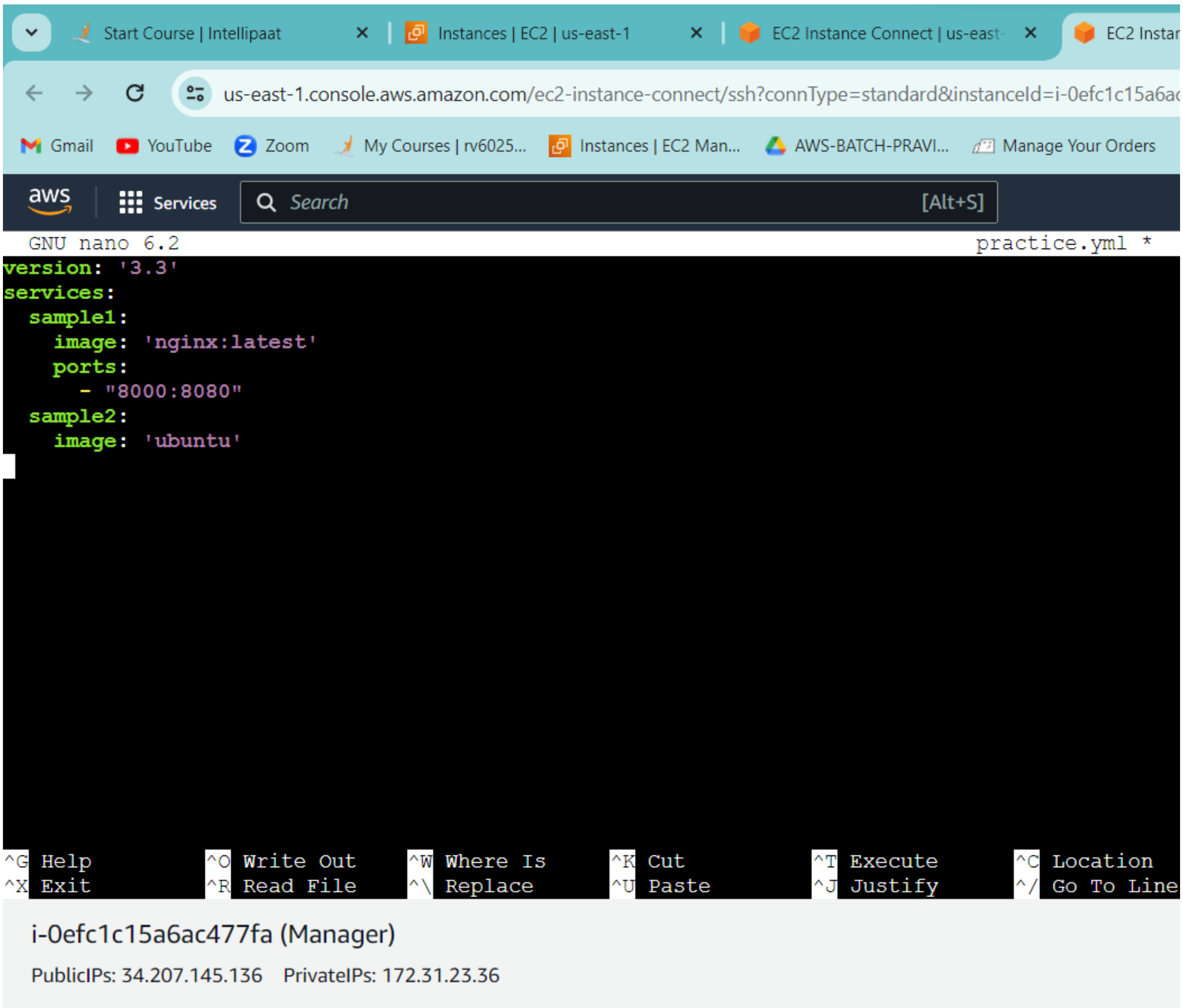
i-0efc1c15a6ac477fa (Manager)

PublicIPs: 34.207.145.136 PrivateIPs: 172.31.23.36

Start Course | Intellipaat
Instances | EC2 | us-east-1
EC2 Instance Connect | us-east-
EC2 Instance Connect | us-east-
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0efc1c15a6ac477fa&osUser=ubuntu&region=us-east-1&ssh...
Services
Search
[Alt+S]
N. Virginia
Rahul Verma
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
12:24 PM
5/14/2024

Paste the below commands

```
version: '3.3'
services:
  sample1:
    image: 'nginx:latest'
    ports:
      - "8000:8080"
  sample2:
    image: 'ubuntu'
```



Start Course | Intellipaat
Instances | EC2 | us-east-1
EC2 Instance Connect | us-east-
EC2 Instar
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0efc1c15a6ac
Gmail
YouTube
Zoom
My Courses | rv6025...
Instances | EC2 Man...
AWS-BATCH-PRAVI...
Manage Your Orders
aws
Services
Search
[Alt+S]

```
  GNU nano 6.2                                                  practice.yml *
version: '3.3'
services:
  sample1:
    image: 'nginx:latest'
    ports:
      - "8000:8080"
  sample2:
    image: 'ubuntu'

^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line
```

i-0efc1c15a6ac477fa (Manager)

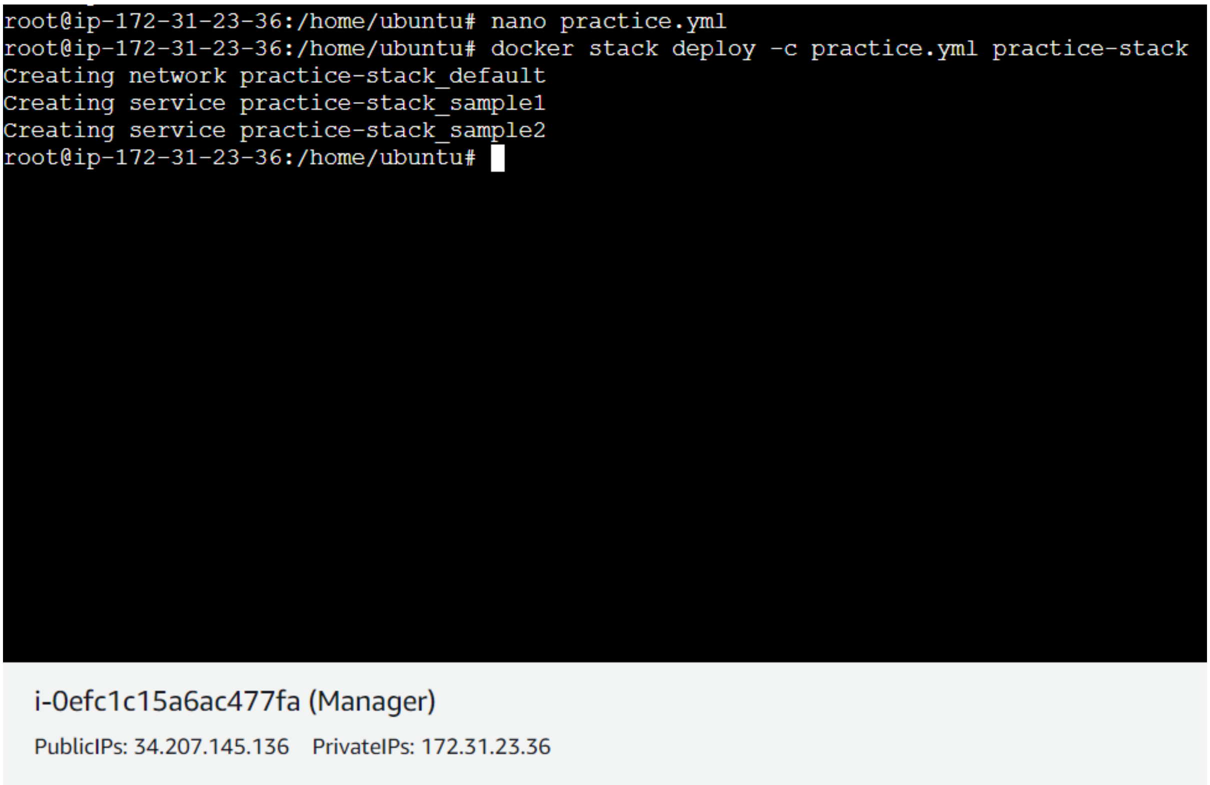
PublicIPs: 34.207.145.136 PrivateIPs: 172.31.23.36

Save and exit (ctrl+s and ctrl +x)

2. To launch services in our docker swarm using yaml file, we have to use stack function

docker stack deploy -c <name_of_the_yaml_file> <name_of_the_stack>

e.g- docker stack deploy -c practice.yml practice-stack



```
root@ip-172-31-23-36:/home/ubuntu# nano practice.yml
root@ip-172-31-23-36:/home/ubuntu# docker stack deploy -c practice.yml practice-stack
Creating network practice-stack_default
Creating service practice-stack_sample1
Creating service practice-stack_sample2
root@ip-172-31-23-36:/home/ubuntu#
```

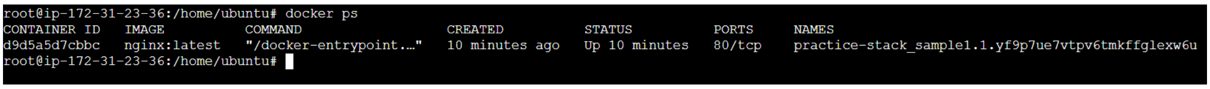
i-0efc1c15a6ac477fa (Manager)

PublicIPs: 34.207.145.136   PrivateIPs: 172.31.23.36

It has created two services because we mentioned two services in our yaml file

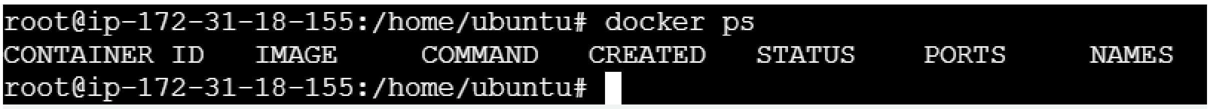
Now if we check there should be one container

docker ps

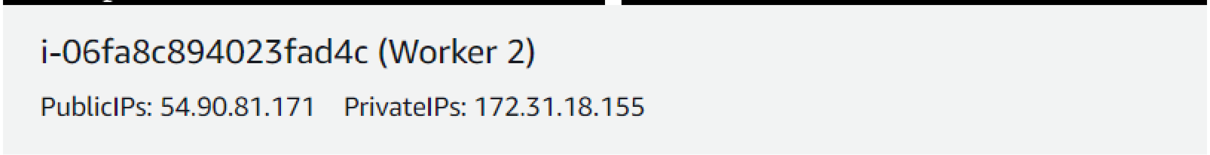


```
root@ip-172-31-23-36:/home/ubuntu# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS     NAMES
d9d5a5d7cbbc   nginx:latest   "/docker-entrypoint.…"   10 minutes ago   Up 10 minutes   80/tcp    practice-stack_sample1.1.yf9p7ue7vtpv6tmkffglexw6u
root@ip-172-31-23-36:/home/ubuntu#
```

If you check your worker nodes there will be no containers

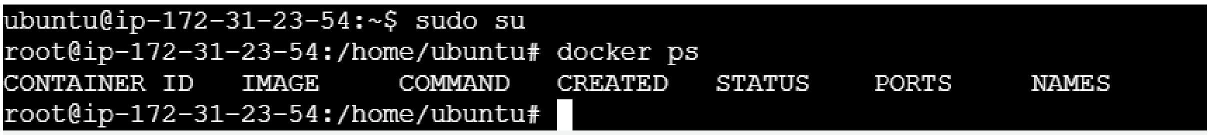


```
root@ip-172-31-18-155:/home/ubuntu# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@ip-172-31-18-155:/home/ubuntu# 
```

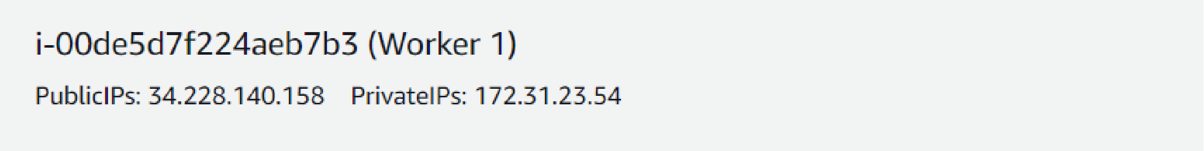


i-06fa8c894023fad4c (Worker 2)

PublicIPs: 54.90.81.171   PrivateIPs: 172.31.18.155



```
ubuntu@ip-172-31-23-54:~$ sudo su
root@ip-172-31-23-54:/home/ubuntu# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@ip-172-31-23-54:/home/ubuntu# 
```



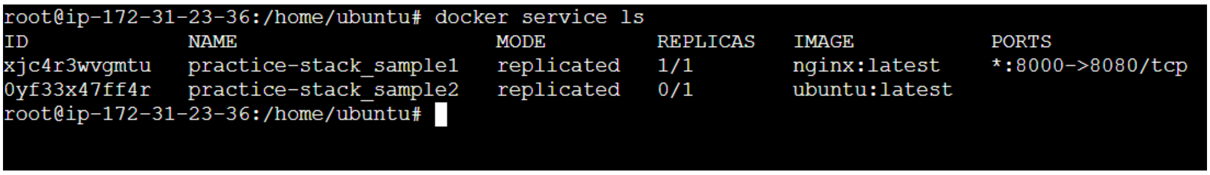
i-00de5d7f224aeb7b3 (Worker 1)

PublicIPs: 34.228.140.158   PrivateIPs: 172.31.23.54

Because we haven't mentioned any replicas in our yaml file so by default it is taking replica as 1

3. so our service is running

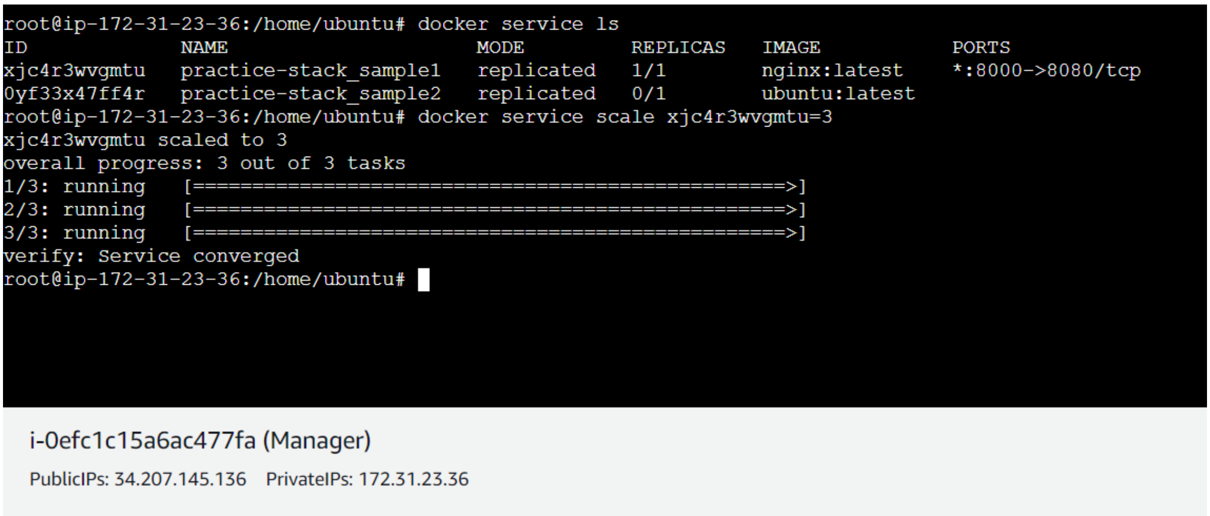
docker service ls



```
root@ip-172-31-23-36:/home/ubuntu# docker service ls
ID             NAME                    MODE         REPLICAS   IMAGE           PORTS
xjc4r3wvgmtu   practice-stack_sample1   replicated   1/1        nginx:latest    *:8000->8080/tcp
0yf33x47ff4r   practice-stack_sample2   replicated   0/1        ubuntu:latest
root@ip-172-31-23-36:/home/ubuntu# 
```

We can see replicated is showing as 1/1

So to scale up or scale down replica we will use the following command docker service scale <service-id>=replicas



```
root@ip-172-31-23-36:/home/ubuntu# docker service ls
ID             NAME                     MODE         REPLICAS   IMAGE           PORTS
xjc4r3wvgmtu   practice-stack_sample1   replicated   1/1        nginx:latest    *:8000->8080/tcp
0yf33x47ff4r   practice-stack_sample2   replicated   0/1        ubuntu:latest
root@ip-172-31-23-36:/home/ubuntu# docker service scale xjc4r3wvgmtu=3
xjc4r3wvgmtu scaled to 3
overall progress: 3 out of 3 tasks
1/3: running   [==================================================>]
2/3: running   [==================================================>]
3/3: running   [==================================================>]
verify: Service converged
root@ip-172-31-23-36:/home/ubuntu# █
```

i-0efc1c15a6ac477fa (Manager)

PublicIPs: 34.207.145.136   PrivateIPs: 172.31.23.36