Distributing the app across swarm cluster

Using AWS EC2 Instances:

1. Create two EC2 instances one instance act as manager and other instance act as working node.
2. Installing docker on two instances:-

   With commands:

   IN AWS you can install docker and work with all the hands-on:

   ```
   #JDK INSTALLATION
   sudo yum update
   sudo  yum install java-17-amazon-corretto-devel -y

   #GIT INSTALLATION
   sudo yum update
   sudo yum install git -y

   #MAVEN INSTALLATION
   sudo wget https://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-maven.repo -O /etc/yum.repos.d/epel-apache-maven.repo
   sudo sed -i s/\$releasever/6/g /etc/yum.repos.d/epel-apache-maven.repo
   sudo yum install -y apache-maven

   #DOCKER INSTALLATION
   sudo yum update
   sudo yum install docker -y
   sudo usermod -a -G docker ec2-user
   ```

```
newgrp docker
sudo systemctl start docker
```

NOTE: after installation docker, if you see this error

```
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
```

To fix the above issue

```
sudo systemctl start docker
sudo chmod 777 /var/run/docker.sock
```

3. Working with swam:

   DOCTOR SWARM:

   STEP 1 Create the Hub:

   Go to an EC2 instance

   Install Docker and start the docker service.

   To start the Swarm Manager (Hub)

   ```
   docker swarm init
   ```

   OUTPUT:

   ```
   To add a worker to this swarm, run the following command:

       docker swarm join --token SWMTKN-1-3c9n3mu6hthnlia97povq67yj8l0ozgpcrob900dmfgwqrihvz-36dxaqm9swvnyf8uwerej8oyz 172.31.28.175:2377
   ```

Before Step 2, open the port 2377 in the firewall of Hub EC2 instance.

STEP 2 Create the worker Node:
Go to another EC2 instance
Install Docker and start the docker service.

To start the Swarm Node

```
docker swarm join --token SWMTKN-1-3c9n3mu6hthnlia97povq67yj8l0ozgpcrob900dmfgwqrihvz-36dxaqm9swvnyf8uwerej8oyz 172.31.28.175:2377
```

STEP 3: Deploy your docker image on the hub. It will automatically go and get deployed(run) in all worker nodes.

Deploy the docker image as a docker swarm service

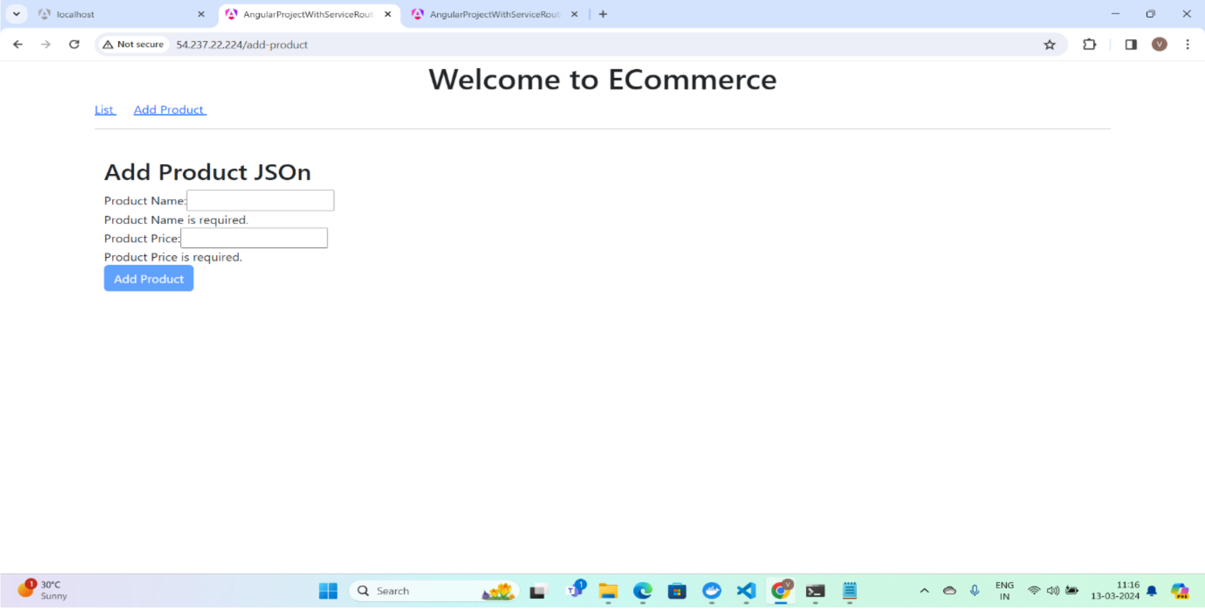
```
docker service create -p 80:80 --name my-ang-app-service  aw1app/my-ang-app
```

and scale it so that it goes to the worker nodes

```
docker service scale my-ang-app-service=2
```

4. Check the app is working or not in the browser use public IP addresses of two instances.

Manager instance output:


localhost
AngularProjectWithServiceRout
AngularProjectWithServiceRout
Not secure
54.237.22.224/add-product
30°C
Sunny
Search
ENG
IN
11:16
13-03-2024

Welcome to ECommerce

List Add Product

Add Product JSOn

Product Name:

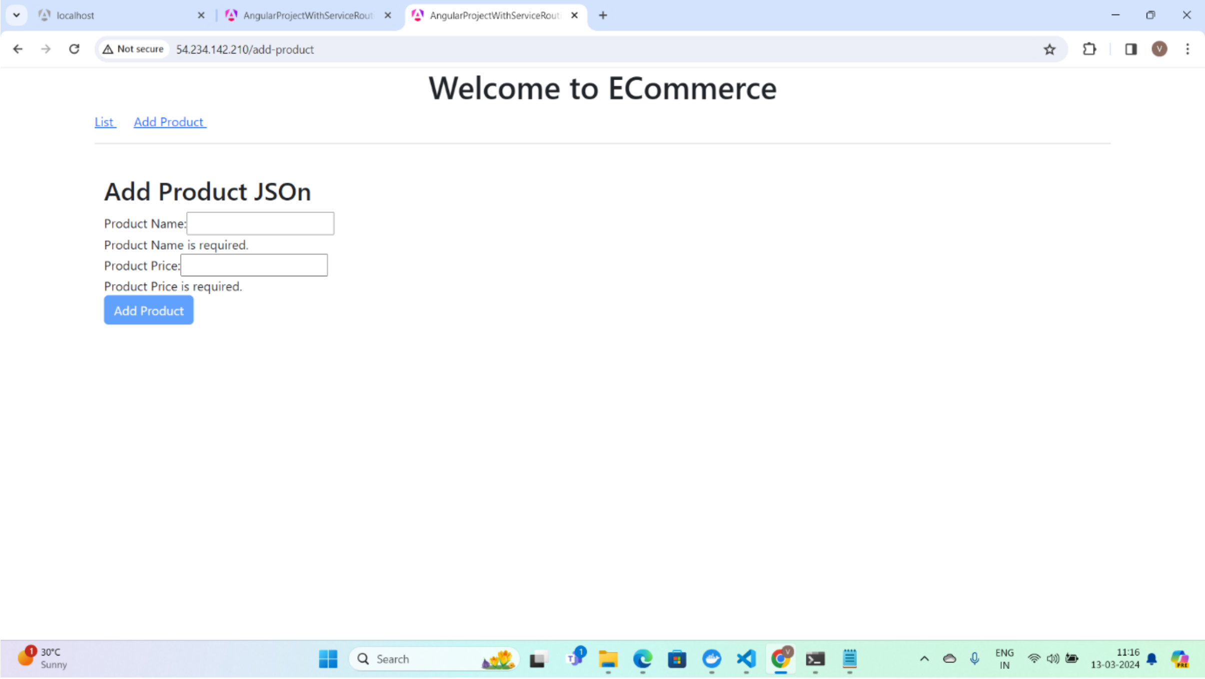
Product Name is required.

Product Price:

Product Price is required.

Add Product

Working node instance output:


localhost
AngularProjectWithServiceRout
AngularProjectWithServiceRout
Not secure
54.234.142.210/add-product
30°C
Sunny
Search
ENG
IN
11:16
13-03-2024

Welcome to ECommerce

List Add Product

Add Product JSOn

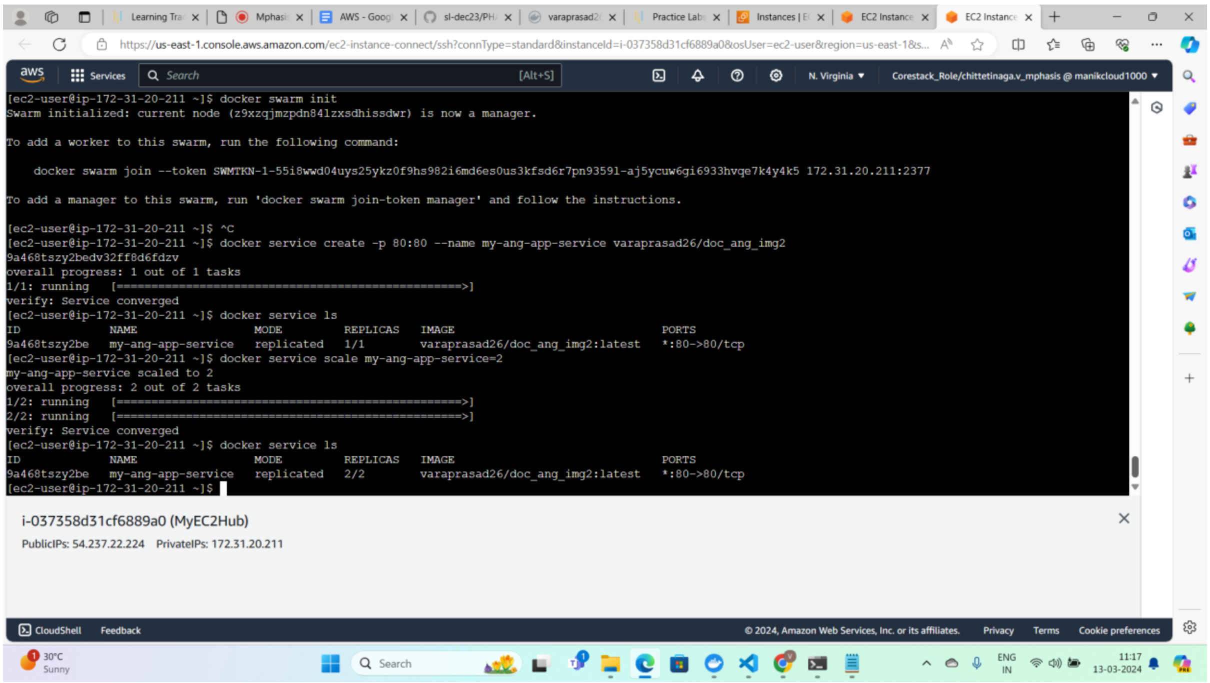
Product Name:

Product Name is required.

Product Price:

Product Price is required.

Add Product

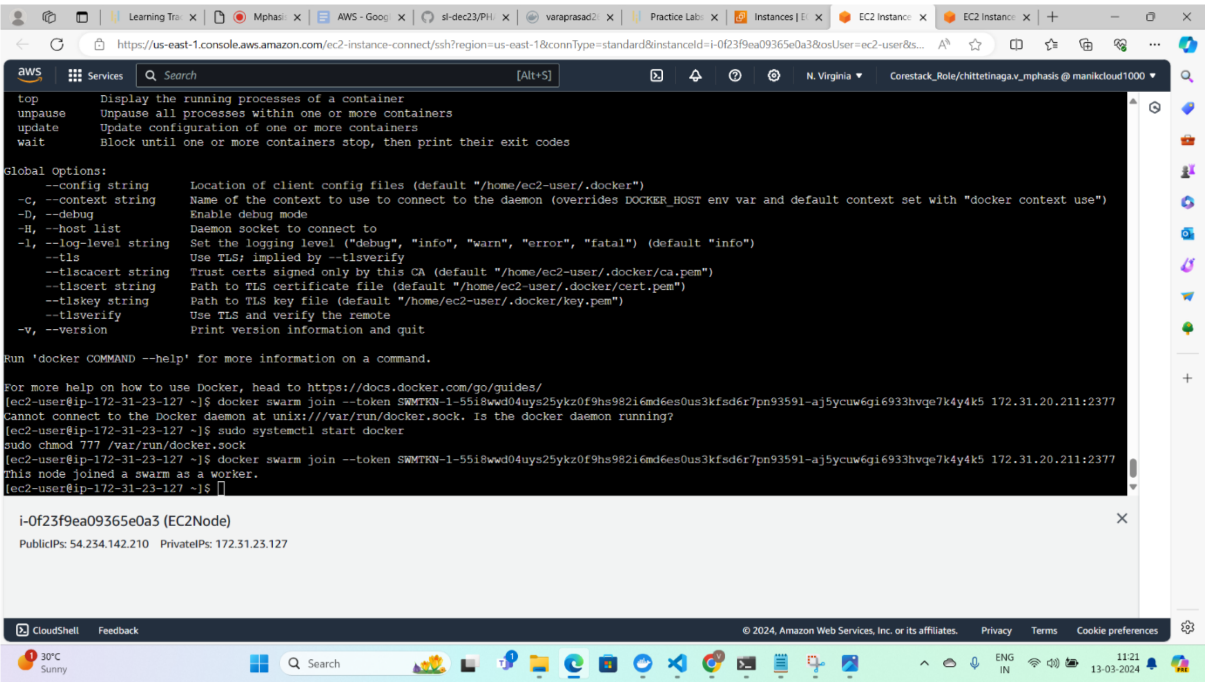


```
[ec2-user@ip-172-31-20-211 ~]$ docker swarm init
Swarm initialized: current node (z9xzqjmzpdn84lzxsdhissdwr) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-55i8wwd04uys25ykz0f9hs982i6md6es0us3kfsd6r7pn93591-aj5ycuw6gi6933hvqe7k4y4k5 172.31.20.211:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

[ec2-user@ip-172-31-20-211 ~]$ ^C
[ec2-user@ip-172-31-20-211 ~]$ docker service create -p 80:80 --name my-ang-app-service varaprasad26/doc_ang_img2
9a468tszy2bedv32ff8d6fdzv
overall progress: 1 out of 1 tasks
1/1: running   [==================================================>]
verify: Service converged
[ec2-user@ip-172-31-20-211 ~]$ docker service ls
ID             NAME                 MODE         REPLICAS   IMAGE                              PORTS
9a468tszy2be   my-ang-app-service   replicated   1/1        varaprasad26/doc_ang_img2:latest   *:80->80/tcp
[ec2-user@ip-172-31-20-211 ~]$ docker service scale my-ang-app-service=2
my-ang-app-service scaled to 2
overall progress: 2 out of 2 tasks
1/2: running   [==================================================>]
2/2: running   [==================================================>]
verify: Service converged
[ec2-user@ip-172-31-20-211 ~]$ docker service ls
ID             NAME                 MODE         REPLICAS   IMAGE                              PORTS
9a468tszy2be   my-ang-app-service   replicated   2/2        varaprasad26/doc_ang_img2:latest   *:80->80/tcp
[ec2-user@ip-172-31-20-211 ~]$
```

i-037358d31cf6889a0 (MyEC2Hub)

PublicIPs: 54.237.22.224   PrivateIPs: 172.31.20.211

Learning Tra
Mphasis
AWS - Goog
sl-dec23/PH
varaprasad2
Practice Labs
Instances | E
EC2 Instance
EC2 Instance
https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-037358d31cf6889a0&osUser=ec2-user&region=us-east-1&s...
aws
Services
Search
[Alt+S]
N. Virginia
Corestack_Role/chittetinaga.v_mphasis @ manikcloud1000
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
30°C
Sunny
ENG
IN
11:17
13-03-2024



```
  top         Display the running processes of a container
  unpause     Unpause all processes within one or more containers
  update      Update configuration of one or more containers
  wait        Block until one or more containers stop, then print their exit codes

Global Options:
      --config string      Location of client config files (default "/home/ec2-user/.docker")
  -c, --context string     Name of the context to use to connect to the daemon (overrides DOCKER_HOST env var and default context set with "docker context use")
  -D, --debug              Enable debug mode
  -H, --host list          Daemon socket to connect to
  -l, --log-level string   Set the logging level ("debug", "info", "warn", "error", "fatal") (default "info")
      --tls                Use TLS; implied by --tlsverify
      --tlscacert string   Trust certs signed only by this CA (default "/home/ec2-user/.docker/ca.pem")
      --tlscert string     Path to TLS certificate file (default "/home/ec2-user/.docker/cert.pem")
      --tlskey string      Path to TLS key file (default "/home/ec2-user/.docker/key.pem")
      --tlsverify          Use TLS and verify the remote
  -v, --version            Print version information and quit

Run 'docker COMMAND --help' for more information on a command.

For more help on how to use Docker, head to https://docs.docker.com/go/guides/
[ec2-user@ip-172-31-23-127 ~]$ docker swarm join --token SWMTKN-1-55i8wwd04uys25ykz0f9hs982i6md6es0us3kfsd6r7pn93591-aj5ycuw6gi6933hvqe7k4y4k5 172.31.20.211:2377
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
[ec2-user@ip-172-31-23-127 ~]$ sudo systemctl start docker
sudo chmod 777 /var/run/docker.sock
[ec2-user@ip-172-31-23-127 ~]$ docker swarm join --token SWMTKN-1-55i8wwd04uys25ykz0f9hs982i6md6es0us3kfsd6r7pn93591-aj5ycuw6gi6933hvqe7k4y4k5 172.31.20.211:2377
This node joined a swarm as a worker.
[ec2-user@ip-172-31-23-127 ~]$
```

i-0f23f9ea09365e0a3 (EC2Node)

PublicIPs: 54.234.142.210   PrivateIPs: 172.31.23.127

Learning Tra
Mphasis
AWS - Goog
sl-dec23/PH
varaprasad2
Practice Labs
Instances | E
EC2 Instance
EC2 Instance
https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-0f23f9ea09365e0a3&osUser=ec2-user&s...
aws
Services
Search
[Alt+S]
N. Virginia
Corestack_Role/chittetinaga.v_mphasis @ manikcloud1000
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
30°C
Sunny
ENG
IN
11:21
13-03-2024