

Q) Deploy a static website through Docker.

A)

Create an EC2 instance with Ubuntu AMI and create a .pem key pair

Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type


☒ RSA
RSA encrypted private and public key pair

☐ ED25519
ED25519 encrypted private and public key pair

Private key file format

☒ .pem
For use with OpenSSH

☐ .ppk
For use with PuTTY

 When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#)

Cancel

Create key pair

EC2

New EC2 Experience

EC2 Dashboard

EC2 Global View

Events

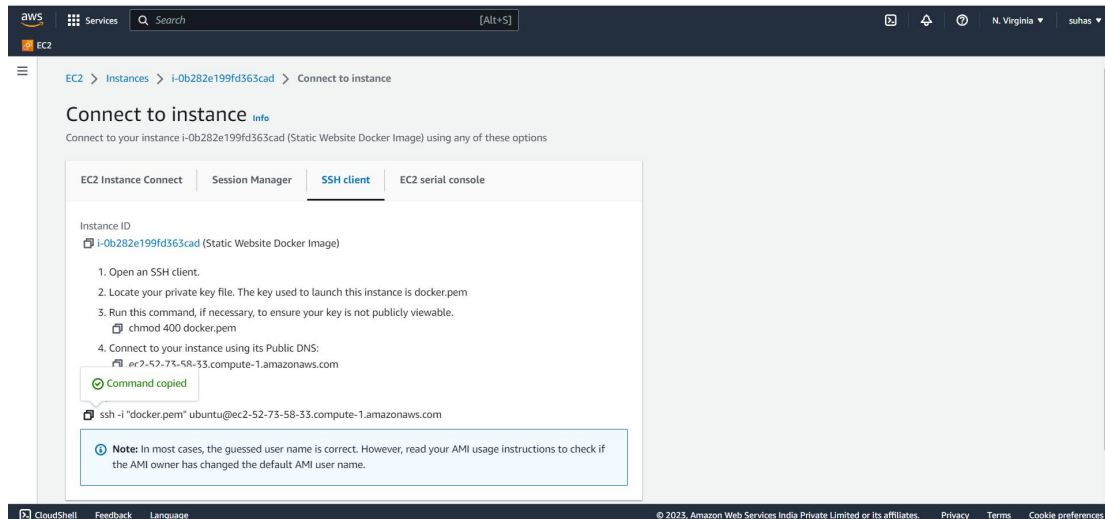
Instances

Instances (1) Info

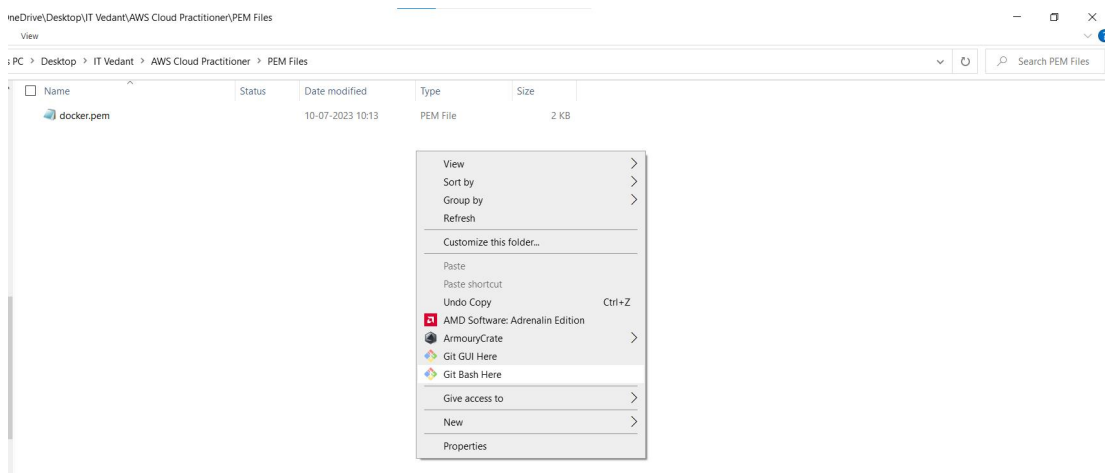
Find instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	Static Website...	i-0b282e199fd363cad	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-52-73-58-33

Once Ubuntu instance has been created, copy the SSH URL in SSH client tab of EC2 instance connect



Go to folder where your .pem file associated to the EC2 instance is located. Right click in that folder and select Git Bash Here option to SSH into your EC2 instance.



Execute the following commands in your Git Bash to create docker image of your static website -

1) `$ssh -i "docker.pem" ubuntu@ec2-52-73-58-33.compute-1.amazonaws.com` - Paste copied text to connect to your EC2 instance

```
ubuntu@ip-172-31-93-202: ~
suhas@LAPTOP-UJ0334FQ MINGW64 ~/OneDrive/Desktop/IT Vedant/AWS Cloud Practitione
r/PEM Files
$ ssh -i "docker.pem" ubuntu@ec2-52-73-58-33.compute-1.amazonaws.com
The authenticity of host 'ec2-52-73-58-33.compute-1.amazonaws.com (52.73.58.33)'
can't be established.
ED25519 key fingerprint is SHA256:5X07n9Wd7FBxWA2w/BFuADB0JrQ/VkOs9CwJKE03iu.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-73-58-33.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Mon Jul 10 04:52:12 UTC 2023

System load:  0.0               Processes:    97
Usage of /:   20.6% of 7.57GB   Users logged in: 0
Memory usage: 24%              IPv4 address for eth0: 172.31.93.202
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-93-202:~$
```

2) **\$sudo apt-get update** - updates your instance to get the latest dependencies

3) **\$Sudo apt-get install docker.io -y** - now the Ubuntu machine acts as a docker host (It is now possible to create docker images to run on containers)

4) **\$docker ps** - list all the running containers in the docker host

Will not show anything (permission denied)

5) **\$sudo usermod -aG docker ubuntu** - this will give the instance permission to view the containers

6) **\$exit** - to exit git bash

\$clear - to clear screen

7) Enter the same SSH URL of EC2 instance again (step 1)

\$ssh -i "docker.pem" [ubuntu@ec2-52-73-58-33.compute-1.amazonaws.com](https://console.aws.amazon.com/ec2/home?instances=ec2-52-73-58-33)

8) **\$docker ps** -now we can see empty list of containers

9) **\$mkdir myexamples**

\$cd myexamples (create and move to a directory)

10) **\$vi index.html** - opens editor to host the HTML code inside a html type file

Enter this sample HTML code into the file

```
<!DOCTYPE html>
<html>
<style>
table, th, td {
    border:1px solid black;
}
</style>
<body>
```

<h2>A basic HTML table</h2>

```
<table style="width:100%">
  <tr>
    <th>Name</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Suhas Meda</td>
    <td>123456789</td>
    <td>India</td>
  </tr>
  <tr>
    <td>Aniket Nandi</td>
    <td>987654321</td>
    <td>Mexico</td>
  </tr>
</table>
</body>
</html>
```

ubuntu@ip-172-31-93-202: ~/myexamples

```
<!DOCTYPE html>
<html>
<style>
table, th, td {
  border:1px solid black;
}
</style>
<body>

<h2>A basic HTML table</h2>

<table style="width:100%">
  <tr>
    <th>Name</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Suhas Meda</td>
    <td>123456789</td>
    <td>India</td>
  </tr>
  <tr>
    <td>Aniket Nandi</td>
    <td>987654321</td>
    <td>Mexico</td>
  </tr>
</table>
</body>
</html>
```

11) \$ls - list all files (index.html should be displayed)

12) \$vi Dockerfile - opens editor where we enter command to allow nginx server access to our index.html file to host on its server

FROM nginx

COPY . /usr/share/nginx/html

13) `$docker build -t suhasmeda/staticimage:latest .`

suhasmeda - username in dockerhub

staticimage - name of image we have provided

This command is used to build our docker image of the static website we have created

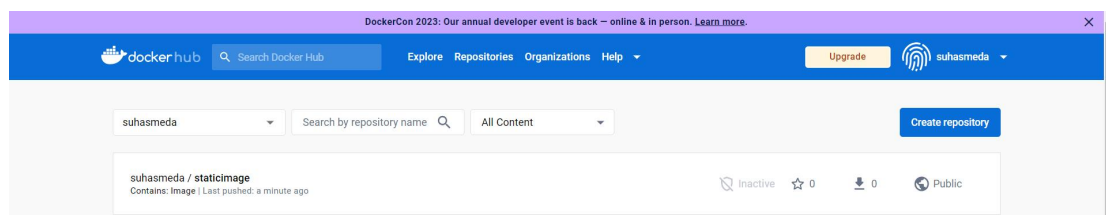
14) `$docker images` - used to list all docker images in the docker host that we have created/pulled (suhasmeda/staticimage should be listed)

15) `$docker login` - used to login to our dockerhub account by providing our username and password of dockerhub

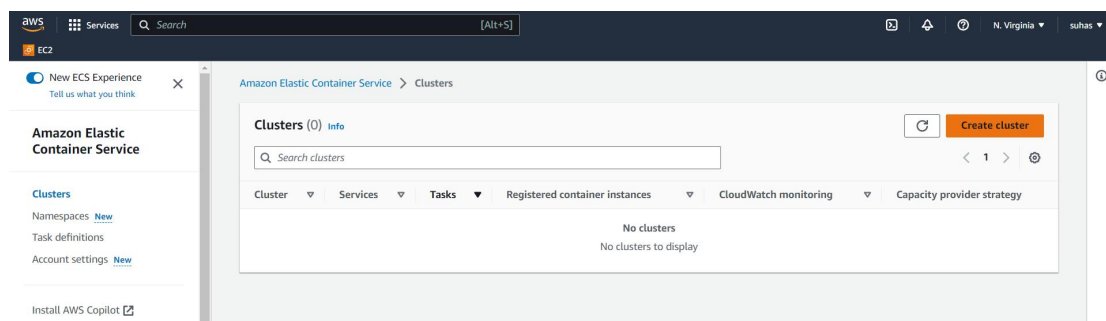
16) `$docker push suhasmeda/staticimage`

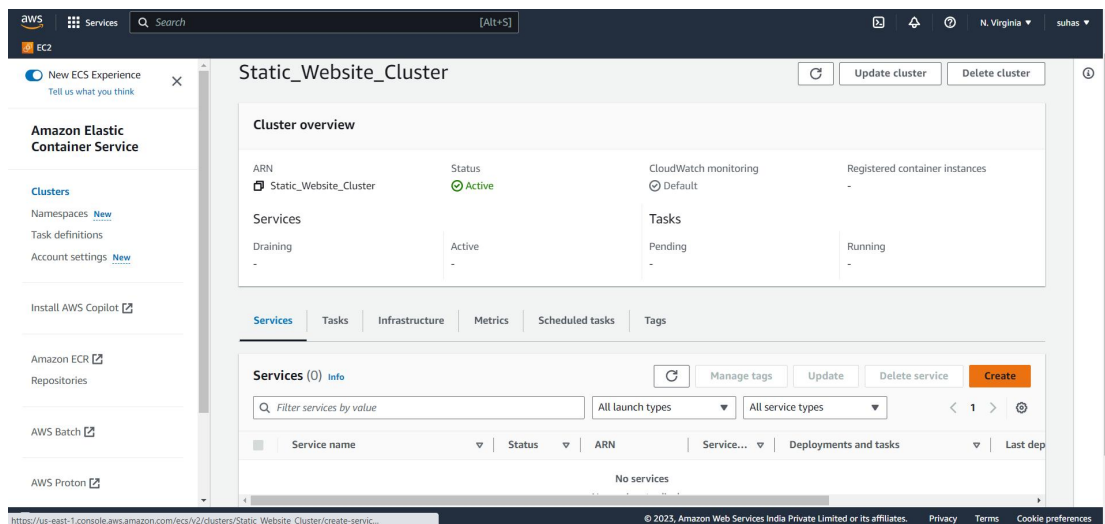
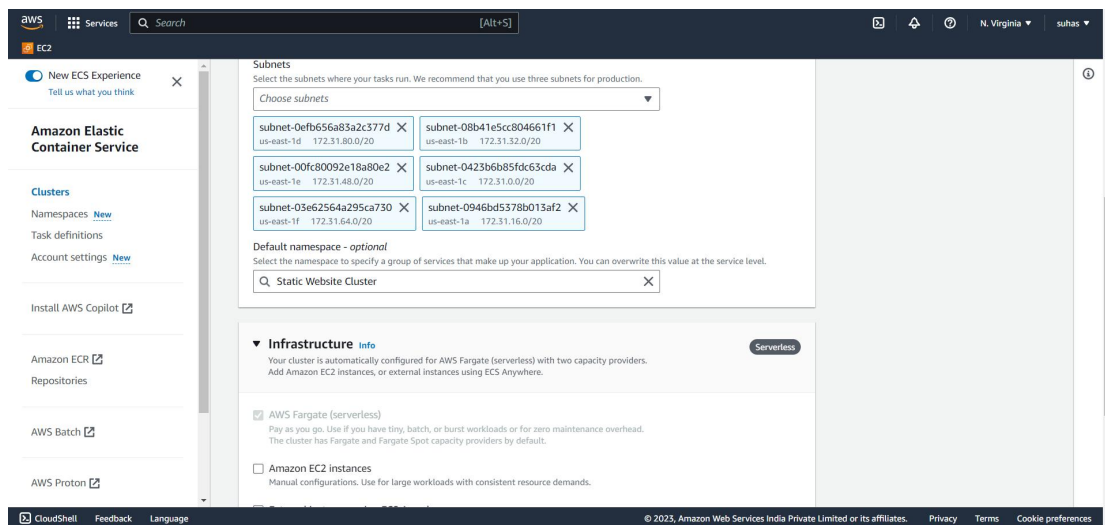
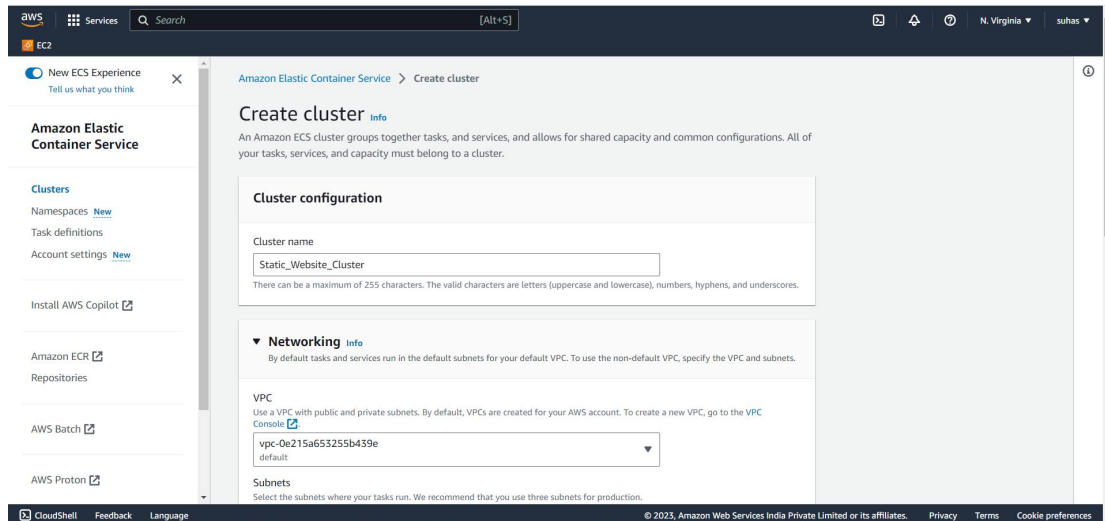
This is to push our docker image to dockerhub so that we can access it in our AWS Console.

Here we can see that the docker image we have created has been pushed into our dockerhub account

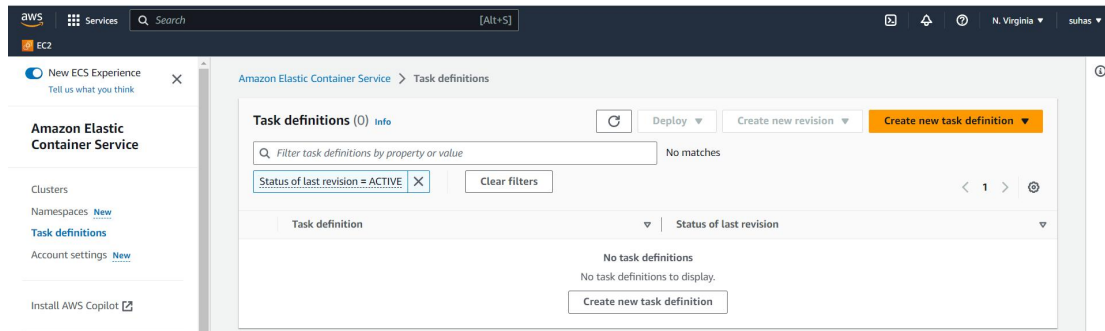


Now go to the AWS console, ECS ->Clusters ->Create cluster

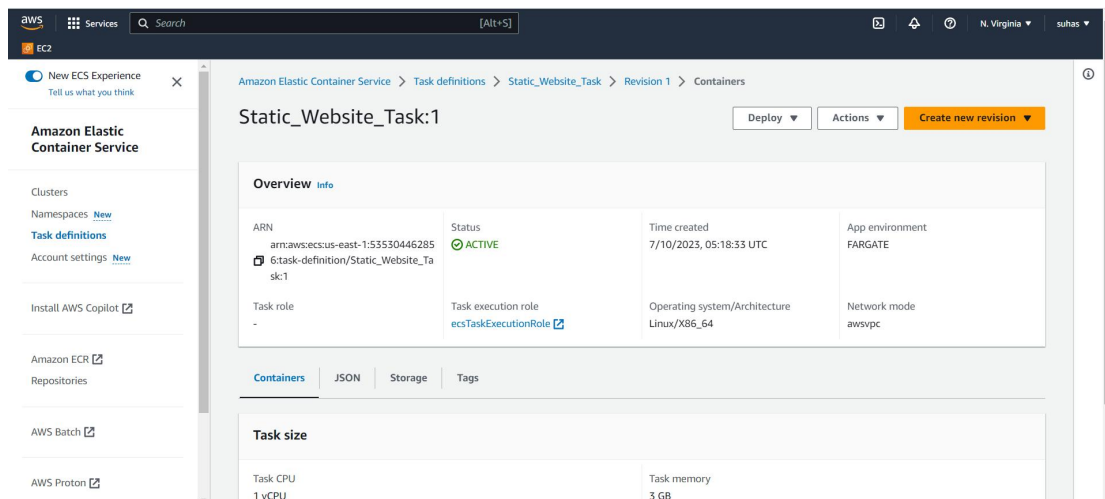
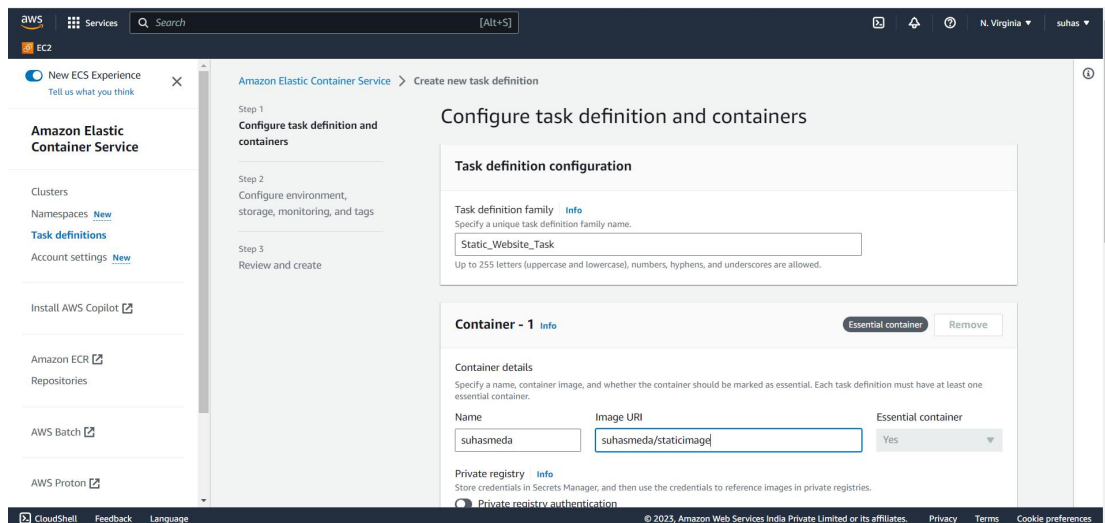




Once ECS cluster with above configurations in created, go to Task definitions section and create a Task Definition with below configurations



For, Name -suhasmeda(our dockerhub username)
Image URI -suhasmeda/staticimage(name of image we have created)



Once Task definition has been created, go to EC2 -> Security Groups -> Create Security Group

(make sure its inside same VPC as your EC2 instance)

Edit inbound rules -> Type - All traffic
Source - Anywhere (0.0.0.0/0)

Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name [Info](#)

Name cannot be edited after creation.

Description [Info](#)

VPC [Info](#)

Inbound rules [Info](#)

Type	Protocol	Port range	Source	Description - optional	
All traffic	All	All	Anywh...		Delete
			0.0.0.0/0		

© 2023, Amazon Web Services India Private Limited or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Now go back to your ECS Cluster you have created and create a Service for it
 (During create Service configuration , make sure to select task, Security Group we have created
 & Create an ALB and target group)

Create [Info](#)

Environment [AWS Fargate](#)

Existing cluster
 Select an existing cluster. To create a new cluster, go to [Clusters](#).

Compute configuration (advanced)

Compute options [Info](#)
 To ensure task distribution across your compute types, use appropriate compute options.

☐ Capacity provider strategy
 Select a launch strategy to distribute your tasks across one or more capacity providers.

☒ Launch type
 Launch tasks directly without the use of a capacity provider strategy.

Launch type [Info](#)
 Select either managed capacity (Fargate), or custom capacity (EC2 or user-managed, External instances). External instances are registered to your cluster using the ECS Anywhere capability.

Platform version [Info](#)
 Specify the platform version on which to run your service.

Deployment configuration

Application type [Info](#)
 Specify what type of application you want to run.

☒ Service
 Launch a group of tasks handling a long-running computing work that can be stopped and restarted. For example, a web application.

☐ Task
 Launch a standalone task that runs and terminates. For example, a batch job.

Task definition
 Select an existing task definition. To create a new task definition, go to [Task definitions](#).
☐ Specify the revision manually
 Manually input the revision instead of choosing from the 100 most recent revisions for the selected task definition family.

Family

Revision

Service name
 Assign a unique name for this service.

Service type [Info](#)
 Specify the service type that the service scheduler will follow.

☒ Replica
 Place and maintain a desired number of tasks across your cluster.

☐ Daemon
 Place and maintain one copy of your task on each container instance.

Desired tasks
 Specify the number of tasks to launch.

► Deployment options

► Deployment failure detection [Info](#)

► Service Connect - optional
 Configure this service in a namespace to create and resolve endpoints. Services can resolve endpoints within the same namespace without task or application configuration.

Networking

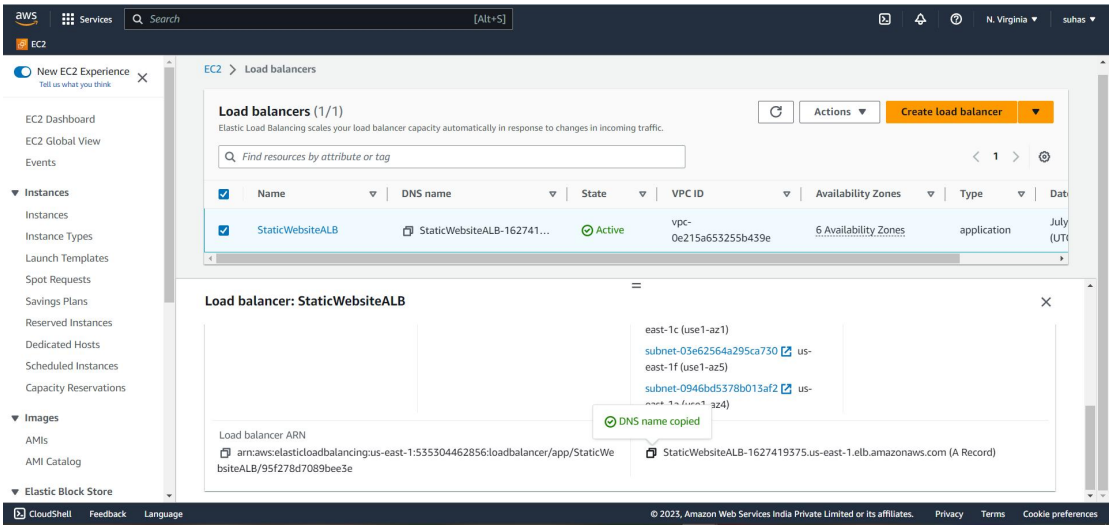
VPC [Info](#)
 Choose the Virtual Private Cloud to use.

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

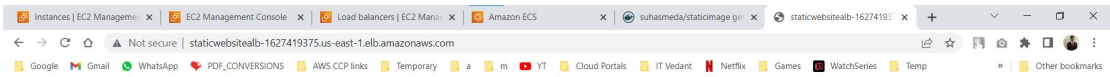
Now go to EC2 -> Load balancers ->

Select the load balancer we have created through the cluster and copy its DNS name



Enter the copied DNS Name to web browser and we can see our static website we have created from deployed Docker Image

OUTPUT -



A basic HTML table

Name	Contact	Country
Suhas Meda	123456789	India
Aniket Nandi	987654321	Mexico