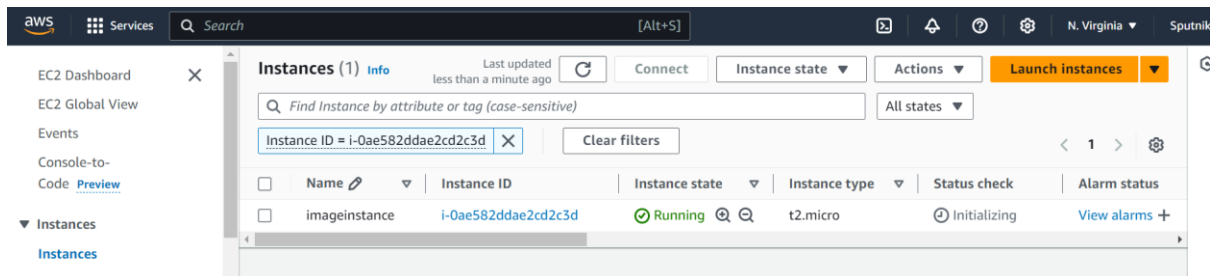


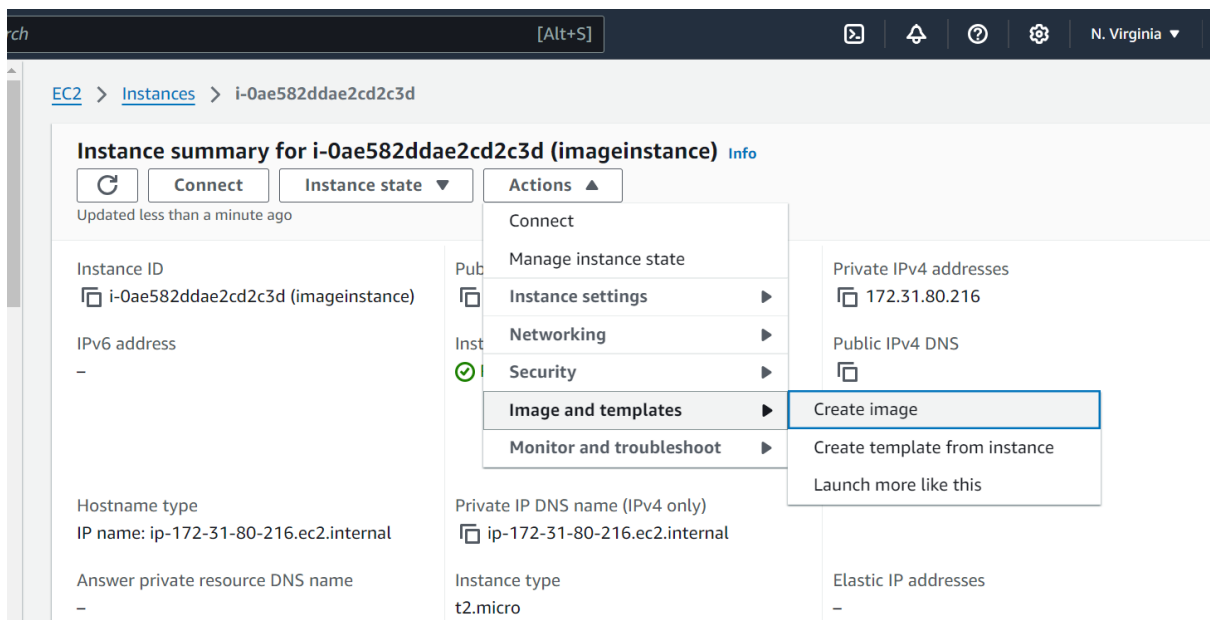
<AbhinavGupta><Milestone3><10747883>

Ans1.

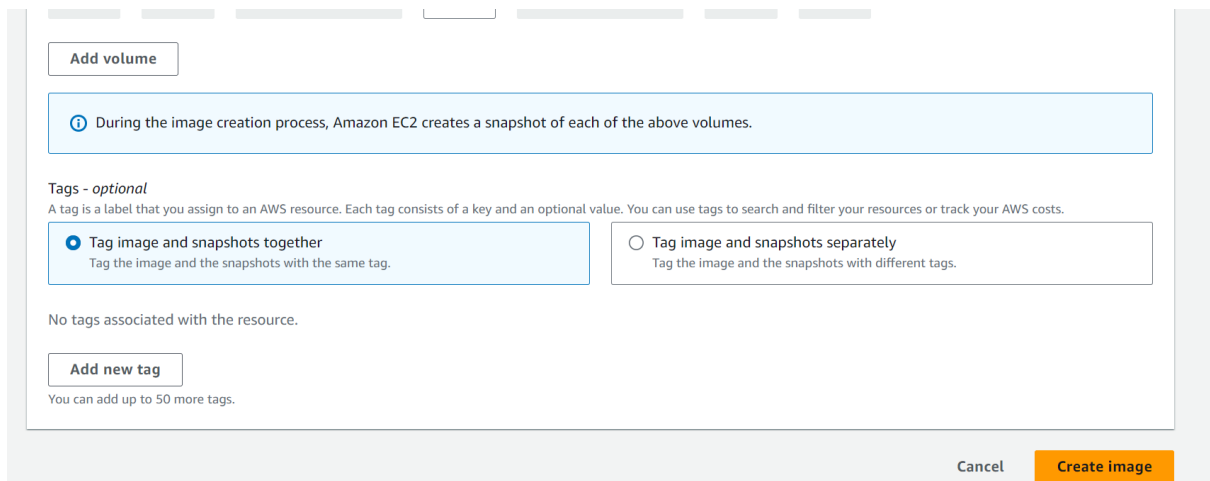
As we can see our running instance in N.Virginia.



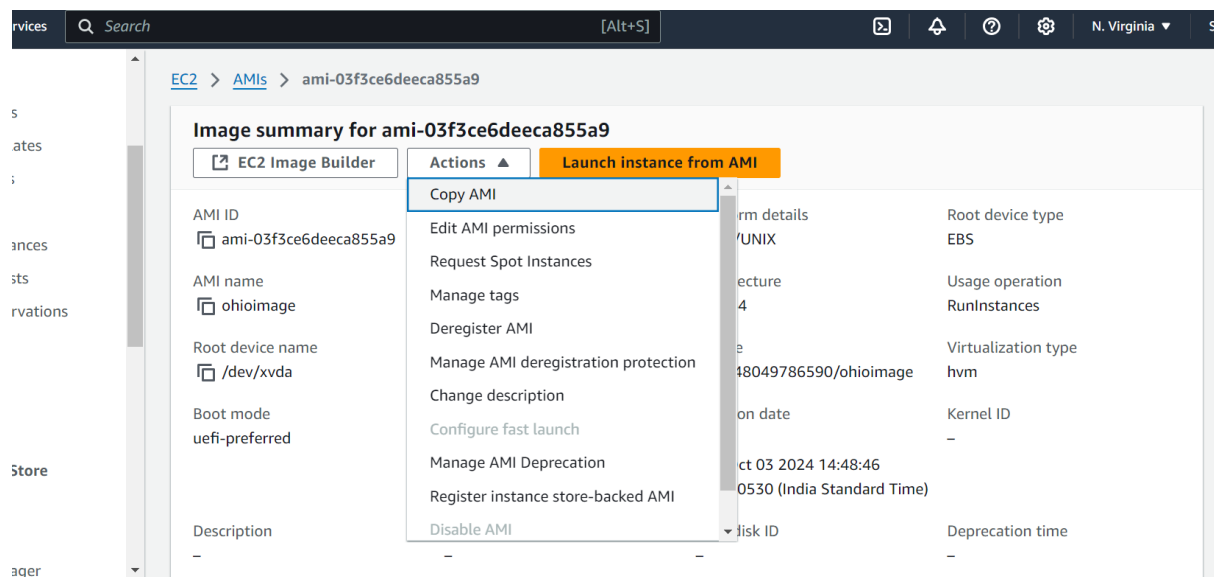
Now go to actions>create image




Select create image



Then go to AMI's and in actions select copy image.



Select required region and copy it.

Original AMI ID
 [ami-03f3ce6deeca855a9](#)

AMI copy name

AMI copy description

Destination Region
A copy of the original AMI will be created in the destination Region.

☐ Copy tags
Includes your user-defined AMI tags when copying the AMI.


☐ Encrypt EBS snapshots of AMI copy
Encrypts all snapshots in the AMI copy with the same key.

Tags - optional

Now go to Ohio region.

And you can see the required image.

Now launch the instance with my AMI's


 Search our full catalog including 1000s of application and OS images

My AMIs

Quick Start

☒ Owned by me

☐ Shared with me


Browse more AMIs

Including AMIs from
AWS, Marketplace and
the Community

Here you can see the running instance

First Create an instance and install terraform in it.


```
ingress {  
  from_port = 22  
  to_port   = 22  
  protocol  = "tcp"  
  cidr_blocks = ["0.0.0.0/0"]  
}
```

```
ingress {  
  from_port = 443  
  to_port   = 443  
  protocol  = "tcp"  
  cidr_blocks = ["0.0.0.0/0"]  
}
```

```
egress {  
  from_port = 0  
  to_port   = 0  
  protocol  = "-1"  
  cidr_blocks = ["0.0.0.0/0"]  
}
```

```
tags = {  
  Names = "dev_access"  
}  
}
```

```
resource "aws_instance" "web-server" {  
  ami           = "ami-08718895af4dfa033"  
  availability_zone = "ap-south-1a"  
  associate_public_ip_address = true
```

```

instance_type      = "t2.micro"

key_name           = "mumbai"

security_groups    = ["${aws_security_group.dev_access.name}"]

tags = {

    Name   = "new-instance"

    Stage  = "testing "

    Location = "INDIA"

}

}

```

Now after creating the we

Terraform init :We need to initiate the directory to use terraform.

Terraform fmt :Then we need to check the format of the file if it is correct.

Terraform validate : to validate the configuration of the files in directory.

Terraform plan : Plan will generate the actions you need to take in order to start the file.

Terraform apply : Creates the infrastructure.

```

[root@ip-172-31-45-82 mspro]# terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.69.0...
- Installed hashicorp/aws v5.69.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
[root@ip-172-31-45-82 mspro]# terraform fmt
provider.tf
[root@ip-172-31-45-82 mspro]# terraform validate
Success! The configuration is valid.

[root@ip-172-31-45-82 mspro]# terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
+ create

Terraform will perform the following actions:

```

Here you can see our new instance

[EC2](#) > [Instances](#) > i-079038680e2cc8166

Instance summary for i-079038680e2cc8166 (new-instance) [Info](#)

Connect

Instance state ▼


Actions ▼

Updated less than a minute ago

<div>Instance ID</div> <div> i-079038680e2cc8166 (new-instance)</div>	<div>Public IPv4 address</div> <div> 13.232.66.171 open address </div>	<div>Private IPv4 addresses</div> <div> 172.31.32.117</div>
<div>IPv6 address</div> <div>—</div>	<div>Instance state</div> <div> Running</div>	<div>Public IPv4 DNS</div> <div> ec2-13-232-66-171.ap-south-1.compute.amazonaws.com open address </div>

With our configuration in the security settings

Security groups

 [sg-03d85362cf740f310 \(dev_access\)](#)

▼ Inbound rules

Filter rules				
Name	Security group rule ID	Port range	Protocol	Source
–	sgr-0eabe6494b229b35a	443	TCP	0.0.0.0/0
–	sgr-0078e91c10e97ff0b	80	TCP	0.0.0.0/0
–	sgr-0b5c86264b9ca36f3	22	TCP	0.0.0.0/0

◀ 1 ▶

Outbound rules

Here you can see our running instance.

```
PS C:\Users\10747883\Downloads> ssh -i "mumbai.pem" ec2-user@ec2-13-232-66-171.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-232-66-171.ap-south-1.compute.amazonaws.com (13.232.66.171)' can't be established.
ED25519 key fingerprint is SHA256:rniNah0LqoHadd0xltRUzcUJx7fMnIb9+LrbT8seJfw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-232-66-171.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.

A newer release of "Amazon Linux" is available.
  Version 2023.5.20241001:
Run "/usr/bin/dnf check-release-update" for full release and version update info

  #
  ~\##### Amazon Linux 2023
  ~\#####
  ~\###
  ~\#|
  ~\V-! -> https://aws.amazon.com/linux/amazon-linux-2023
  ~\
  ~\
  ~\
  ~\m/

[ec2-user@ip-172-31-32-117 ~]$ sudo su
[root@ip-172-31-32-117 ec2-user]# cd
[root@ip-172-31-32-117 ~]#
```

Ans3.

Create two instances one controller and one host

<input type="checkbox"/>	controller	i-0b6584c33b5c43d66	Running	t2.micro	Initializing	View alarms +
<input checked="" type="checkbox"/>	host1	i-0474e8c3acff02557	Running	t2.micro	Initializing	View alarms +

i-0474e8c3acff02557 (host1)

Now configure `sshd_config` in host file to allow root access to controller with the help of `ssh`

```

root@ip-172-31-18-55:~
te.amazonaws.com (18.143.65.110)' can't be established.
ED25519 key fingerprint is SHA256:Y4CoZzCeISP7sS+ziPGk6NZ8GxB9Vw
wBn5PCrbvjew.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-143-65-110.ap-southeast-1.com
pute.amazonaws.com' (ED25519) to the list of known hosts.

A newer release of "Amazon Linux" is available.
Version 2023.5.20241001:
Run "/usr/bin/dnf check-release-update" for full release and ver
sion update info

#####
#_##### Amazon Linux 2023
#####
#####
##### \###|
##### \##/ ____ https://aws.amazon.com/linux/amazon-linux-2
023
##### V~! !->
##### _ _ _
##### _ _ _
##### _ _ _

[ec2-user@ip-172-31-18-55 ~]$ hostnamectl set-hostname controlle
r
Could not set static hostname: Access denied
[ec2-user@ip-172-31-18-55 ~]$ sudo su
[root@ip-172-31-18-55 ec2-user]# cd
[root@ip-172-31-18-55 ~]# hostnamectl set-hostname controller
[root@ip-172-31-18-55 ~]# bash
[root@controller ~]#

48 # but this is overridden so installations will only check .s
sh/authorized_keys
49 AuthorizedKeysFile .ssh/authorized_keys
50
51 #AuthorizedPrincipalsFile none
52
53
54 # For this to work you will also need host keys in /etc/ssh/
ssh_known_hosts
55 #HostbasedAuthentication no
56 # Change to yes if you don't trust ~/.ssh/known_hosts for
57 # HostbasedAuthentication
58 #IgnoreUserKnownHosts no
59 # Don't read the user's ~/.rhosts and ~/.shosts files
60 #IgnoreRhosts yes
61
62 # Explicitly disable PasswordAuthentication. By presettin
g, we
63 # avoid the cloud-init set_passwords module modifying sshd_c
onfig and
64 # restarting sshd in the default instance launch configurati
on.
65 PasswordAuthentication yes
66 PermitEmptyPasswords yes
67
68 # Change to no to disable s/key passwords
69 #KbdInteractiveAuthentication yes
70
71 # Kerberos options
72 #KerberosAuthentication no
73 #KerberosOrLocalPasswd yes
-- INSERT -- 66,25 43%
```

After this just restart and enable `sshd`

After configuration send the public key of controller to host

```

[root@controller ~]# ssh-copy-id root@172.31.24.255
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/
root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new ke
y(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- i
f you are prompted now it is to install the new keys
root@172.31.24.255's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@172.31.24.2
55'"
and check to make sure that only the key(s) you wanted were adde
d.

[root@controller ~]#
```

Here in `.ssh` authorized key we can see the added key


```
[root@host1 .ssh]# cat authorized_keys
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command=
"echo 'Please login as the user \"ec2-user\" rather than the us
er \"root\".'" ;echo;sleep 10;exit 142" ssh-rsa AAAAB3NzaC1yc2EAAA
ADAQABAAQDA10SZusXE5FHyAZIHdUXQ80tfeBYCd6ygreLsDXgBd+qSeCmgx1
CYjv3sS76pksYoSyRZML89cwGjD5Qj429s5X1iB3oi7uprnXG9HF8ExQ1ck319qs
Me5cNm48+sODhFN0JRzwgUEQdeMpJ4ILGN/hKJZJ0Xq09HHM0TERhXeSbNb4znVE
vCyHofDwIrXEAYE7zJ/njUgoqC1omZrkYjbCB24ife1yC70ECVsa9jNYSag7U9vo
3K9s834WluFm+lg0zq6qjce+d27sMkyUOR2g248ll8bS73MiOrVTnXOIUnDsZepq
z5g8B7X4rrZOXObfA82Agy7gqIk2lRnR3l Singapore1
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC6VYBLPs0YW8vwPcp5vEiiPaT3
L+hZm02XzKpA4ZJKo+Ff8WUQ9WRSZZphhtZiIHY8qAXdC7fsuzT5SB/HypzyFVjS
4vkE/Hgwe000pp0rrNiSq11fUFYKT+Lh1k/86Pxpr1Ws4K4xl0Js/m208H+zQ2SA
/Jcdw1rnvoaiJCxLaBKmN1PbwQ4QXuJbTTSnvq8DRkQuCEv+Ln3l5Ln1U0yxFTVo
uA8Hwe7VAo6VVeydFmGr52IKIYW2x80JXoqJP6KMhwHRk2vkntaPfjXAwTH4nlCR
EFAw0DsqaqMnPXfBASurN0pGIjtelM4Ce3Cr1m4BvNLj0b3aE4lgaZedruXzndBf
5y1o+aNdmMGy0M4ya+kEkwculeT1GRumU+0cI2unGECJJzmKwADD9zYH4jEHFpkg
PLmRq80zzInJ7BTvboVE03ibNAdmaZyohfBzF9lsWWI0ip6hWfe9SdiWzwTU43iG
It15n2KeVwlmSpwUo+wi1ZMPv+T8VgdhUuQvkzc= root@controller
[root@host1 .ssh]#
```

Now install ansible in controller instance

```
[root@controller ~]# yum install ansible* -y
Last metadata expiration check: 0:12:35 ago on Thu Oct 3 10:16:
14 2024.
Dependencies resolved.
=====
Package                Arch    Version                               Repository    Size
=====
Installing:
ansible                noarch  8.3.0-1.amzn2023.0.1                amazonlinux   32 M
ansible-core           x86_64  2.15.3-1.amzn2023.0.4                amazonlinux   2.5 M
ansible-packaging      noarch  1-11.amzn2023.0.1                    amazonlinux   14 k
ansible-packaging-tests noarch  1-11.amzn2023.0.1                    amazonlinux   9.7 k
ansible-srpm-macros    noarch  1-11.amzn2023.0.1                    amazonlinux   22 k
Installing dependencies:
ansible-test           x86_64  2.15.3-1.amzn2023.0.4                amazonlinux   705 k
```

And after this go to ansible folder , make and configure ansible.cfg file

```

# file socket names (108 characters for most platforms). In that
# case, you
# may wish to shorten the string below.
#
# Example:
# control_path = %(directory)s/%%h-%%r
#control_path = %(directory)s/ansible-ssh-%%h-%%p-%%r

# Enabling pipelining reduces the number of SSH operations requi
red to
# execute a module on the remote server. This can result in a si
gnificant
# performance improvement when enabled, however when using "sudo
:" you must
# first disable 'requiretty' in /etc/sudoers
#
# By default, this option is disabled to preserve compatibility
with
# sudoers configurations that have requiretty (the default on ma
ny distros).
#
#pipelining = False

# if True, make ansible use scp if the connection type is ssh
# (default is sftp)
#scp_if_ssh = True

[accelerate]
accelerate_port = 5099
accelerate_timeout = 30
accelerate_connect_timeout = 5.0

```

166,32

Bot

Update vim hosts file



A terminal window with a dark background. The title bar shows 'root@ip-172-31-18-55:/etc/an' and a close button. The terminal content shows the command '[new-server]' followed by the IP address '172.31.24.255'. Below this, there are three tilde '~' characters, likely representing the output of a command or a list of hosts.

Here we can see the host

```
[root@controller ansible]# vim hosts
[root@controller ansible]# ansible all --list-hosts
[WARNING]: Invalid characters were found in group names but not
replaced, use
-vvvv to see details
  hosts (1):
    172.31.24.255
[root@controller ansible]#
```

After successful connection we can now create a play book and see the results

This is my playbook

```
---
- name: creating a user and installing http
  hosts: all

  tasks:

    - name: create a user bob
      user:
        name: bob
        uid: 1100
        shell: /bin/bash
        home: /home/ltimindtree
        state: present

    - name: install http
      yum:
        name: httpd
        state: present

    - name: start service and enable it
      systemd:
        name: httpd
        state: started
        enabled: true

~
~
~
~
```

```

[root@controller ansible]# ansible-playbook playbook.yml
[WARNING]: Invalid characters were found in group names but not replaced, use
-vvvv to see details

PLAY [creating a user and installing http] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 172.31.24.255 is using the discovered Python
interpreter at /usr/bin/python3.9, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more informat
ion.
ok: [172.31.24.255]

TASK [create a user bob] *****
changed: [172.31.24.255]

TASK [install http] *****
changed: [172.31.24.255]

TASK [start service and enable it] *****
changed: [172.31.24.255]

PLAY RECAP *****
172.31.24.255 : ok=4    changed=3    unreachable=0    failed=0
               skipped=0    rescued=0    ignored=0

[root@controller ansible]#

```

Now we can bob user

```

in
systemd-timesync:x:995:995:systemd Time Synchronization:/:/usr/s
bin/nologin
chrony:x:994:994:chrony system user:/var/lib/chrony:/sbin/nologi
n
ec2-instance-connect:x:993:993:/:home/ec2-instance-connect:/sbin
/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
tcpdump:x:72:72:/:/sbin/nologin
ec2-user:x:1000:1000:EC2 Default User:/home/ec2-user:/bin/bash
bob:x:1100:1100:/:home/ltimindtree:/bin/bash
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
[root@host1 ~]#

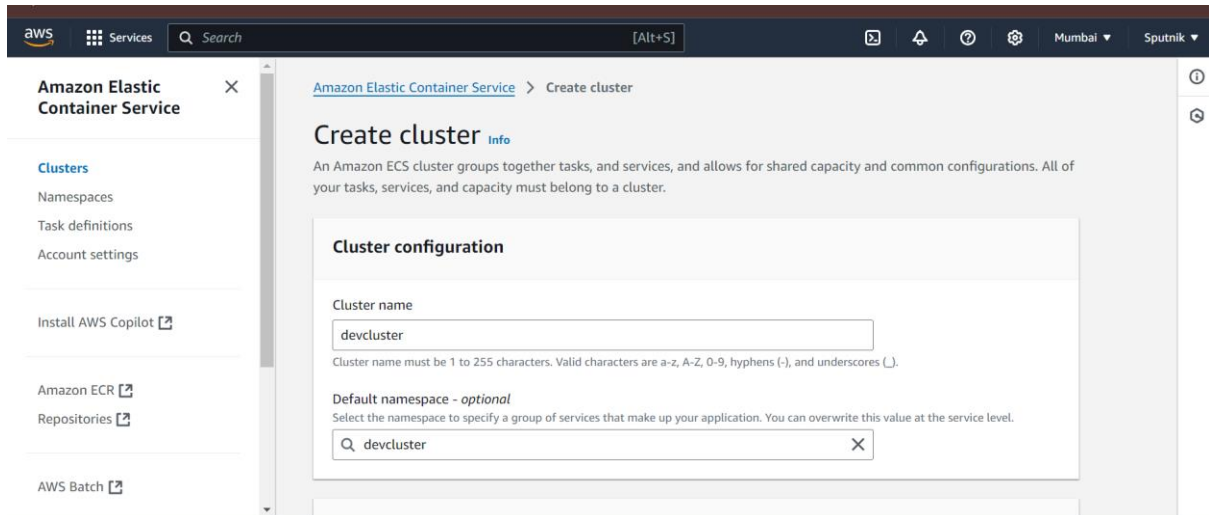
```

And we successfully install httpd

```
[root@host1 ~]# rpmquery httpd
httpd-2.4.62-1.amzn2023.x86_64
[root@host1 ~]#
```

Ans4.

Create one cluster with fargate in Mumbai region.



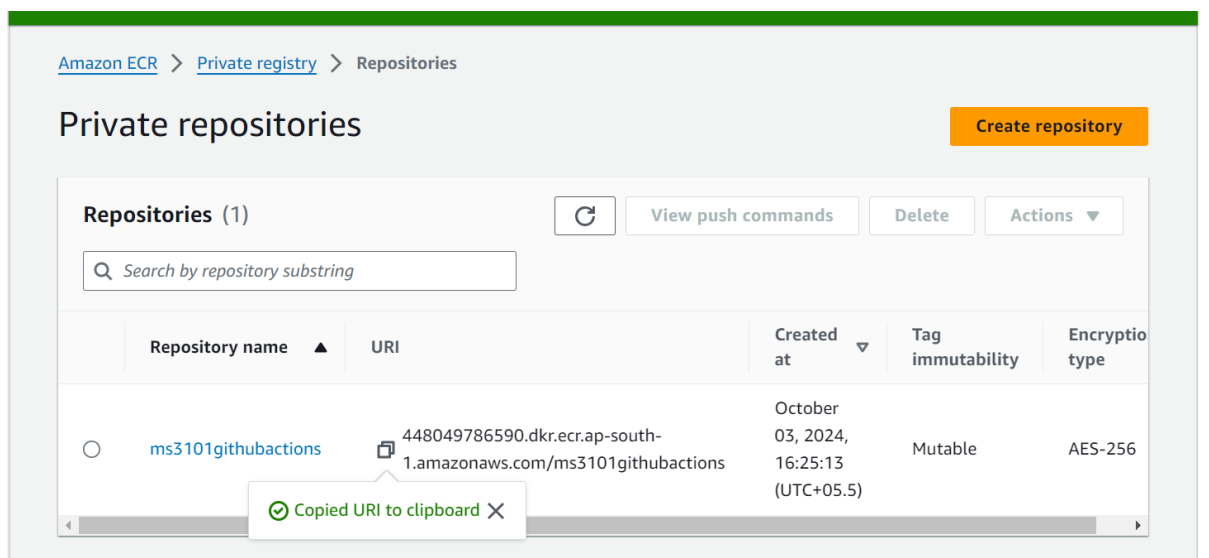
Then create role for task definition with these policy.

AdministratorAccess

amazonec2containerregistryfullaccess

amazonecstaskexecutionrolepolicy

then create ECR and copy it's uri



Task placement = optional

Task placement constraints are not supported for AWS Fargate launch type.

Container - 1 [Info](#) Essential container Remove

Container details
Specify a name, container image, and whether the container should be marked as essential. Each task definition must have at least one essential container.

Name: Image URI: Essential container:

Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

Up to 255 letters (uppercase and lowercase), numbers, hyphens, underscores, colons, periods, forward slashes, and number signs are allowed.

Private registry [Info](#)
Store credentials in Secrets Manager, and then use the credentials to reference images in private registries.

☒ Private registry authentication

Now we have to create service

ms3service deployment is in progress. It takes a few minutes. [View in CloudFormation](#)

[Amazon Elastic Container Service](#) > [Clusters](#) > [devcluster](#) > [Services](#)

devcluster Refresh Update cluster Delete cluster

Cluster overview

ARN arn:aws:ecs:ap-south-1:448049786590:cluster/devcluster	Status Active	CloudWatch monitoring Default	Registered container instances -
Services Draining	Tasks Pending		

Provide task definition and service name in this cluster services

Services (1) [Info](#) Refresh Manage tags Update Delete service

Filter launch type: Filter service type:

<input type="checkbox"/>	Service name	ARN	Status	Service
<input type="checkbox"/>	ms3service	arn:aws:ec...	Active	REPLICA

Configure security settings in services

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-0926f7ed5a411789a	All traffic ▼	All	All	Cus... ▼	<input type="text"/>	<input type="text"/> <input type="button" value="Delete"/>
-	HTTP ▼	TCP	80	An... ▼	<input type="text"/>	<input type="text"/> <input type="button" value="Delete"/>
-	Custom TCP ▼	TCP	8080	An... ▼	<input type="text"/>	<input type="text"/> <input type="button" value="Delete"/>

Copy this json from task definition

X

JSON

Download JSON
Download AWS CLI input
Copy to clipboard










```

1 {
2   "taskDefinitionArn": "arn:aws:ecs:ap-south-1:448049786590:task-definition/ms3taskdef:1",
3   "containerDefinitions": [
4     {
5       "name": "reg-app",
6       "image": "448049786590.dkr.ecr.ap-south-1.amazonaws.com/ms3101githubactions",
7       "cpu": 0,
8       "portMappings": [
9         {
10          "name": "reg-app-80-tcp",
11          "containerPort": 80,
12          "hostPort": 80,
13          "protocol": "tcp",
14          "appProtocol": "http"
15        }
16      ],
17      "essential": true

```

Paste this file in reg-app-task-def.json

In the repository

 voyag3r Update main.yml ✓		
 .github/workflows		Update main.yml
 server		Add files via upload
 webapp		Update index.jsp
 Dockerfile		Update Dockerfile
 Jenkinsfile		Update Jenkinsfile
 README.md		test
 pom.xml		Add files via upload
 reg-app-task-def.json		Create reg-app-task-def.json