Cycle 08 Homework Notes

Name: A Chetan Varma

Date: 21-07-2025

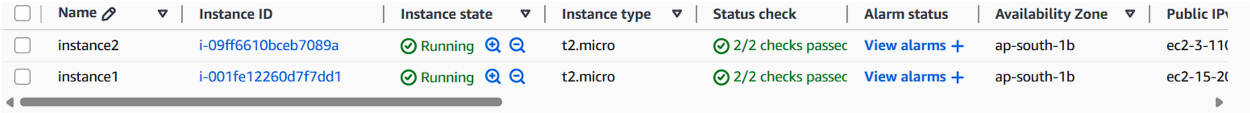
1. EBS & EFS Practical Exercise

EBS (Elastic Block Store):

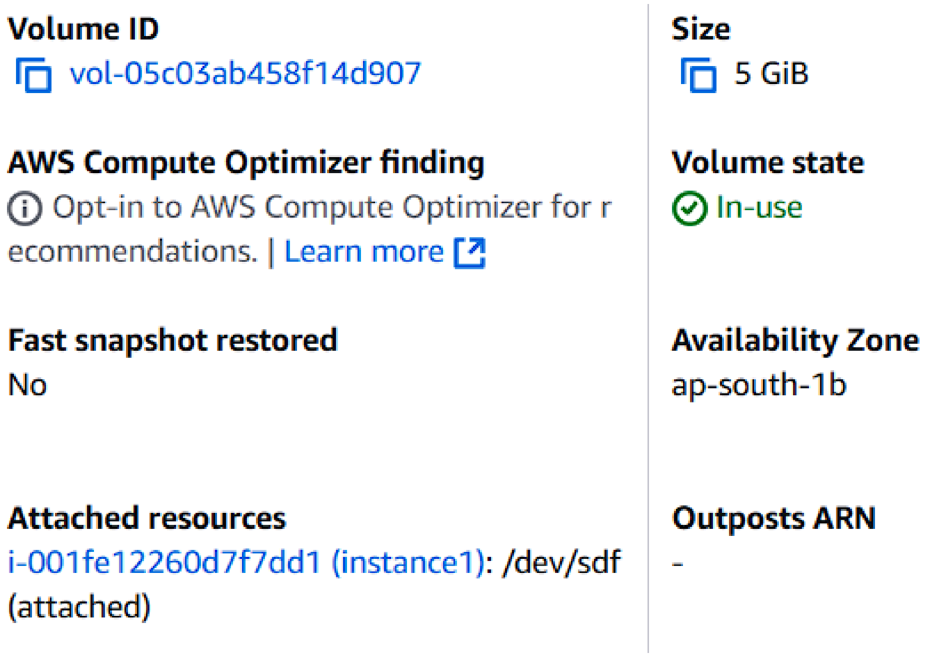
- Created a new EBS volume in the same Availability Zone as the EC2 instance.
- Attached the volume using EC2 console.
- On the instance:

```
bash
CopyEdit
sudo mkfs -t ext4 /dev/xvdf
sudo mkdir /ebs-data
sudo mount /dev/xvdf /ebs-data
```

- Created a test file and verified data persistence.
- Took a snapshot from the volume.
- Created a new volume from the snapshot.
- Attached to another EC2 instance and verified the file.



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP\
□	instance2	i-09ff6610bceb7089a	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-south-1b	ec2-3-11(
□	instance1	i-001fe12260d7f7dd1	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-south-1b	ec2-15-2(



Volume ID
vol-05c03ab458f14d907

Size
5 GiB

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

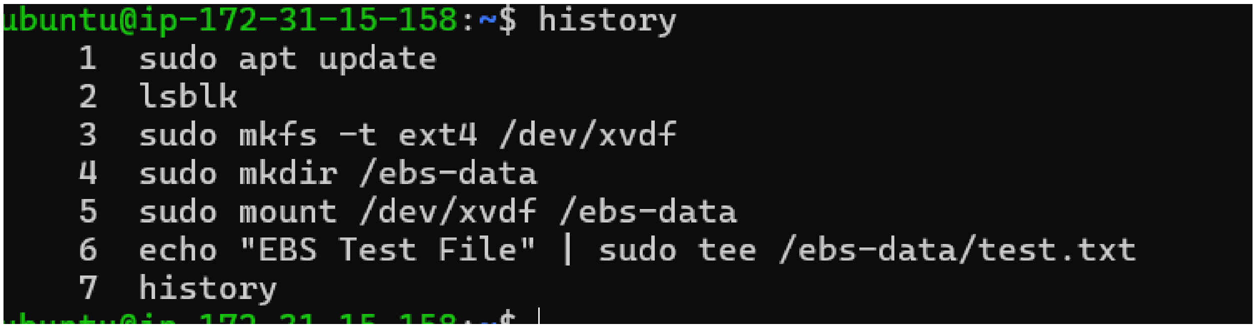
Volume state
In-use

Fast snapshot restored
No

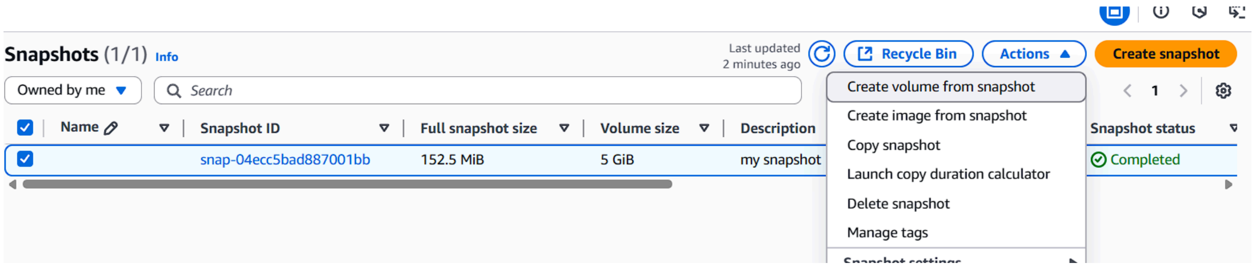
Availability Zone
ap-south-1b

Attached resources
i-001fe12260d7f7dd1 (instance1): /dev/sdf (attached)

Outposts ARN
-

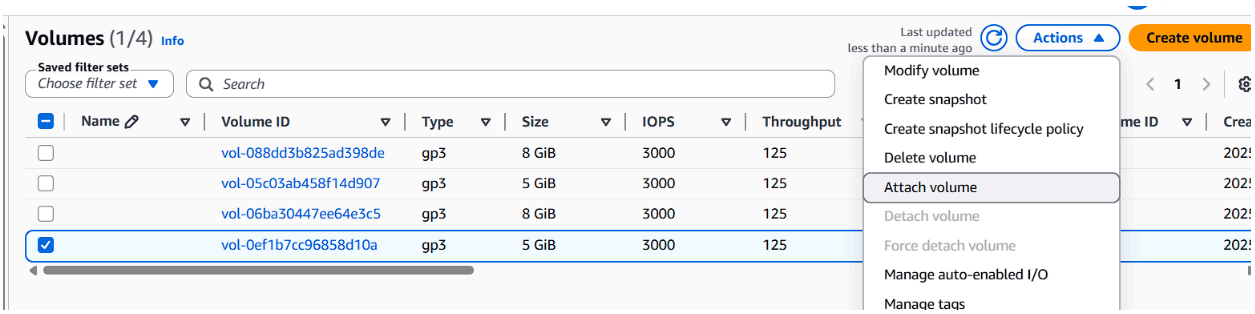


```
ubuntu@ip-172-31-15-158:~$ history
    1  sudo apt update
    2  lsblk
    3  sudo mkfs -t ext4 /dev/xvdf
    4  sudo mkdir /ebs-data
    5  sudo mount /dev/xvdf /ebs-data
    6  echo "EBS Test File" | sudo tee /ebs-data/test.txt
    7  history
ubuntu@ip-172-31-15-158:~$
```



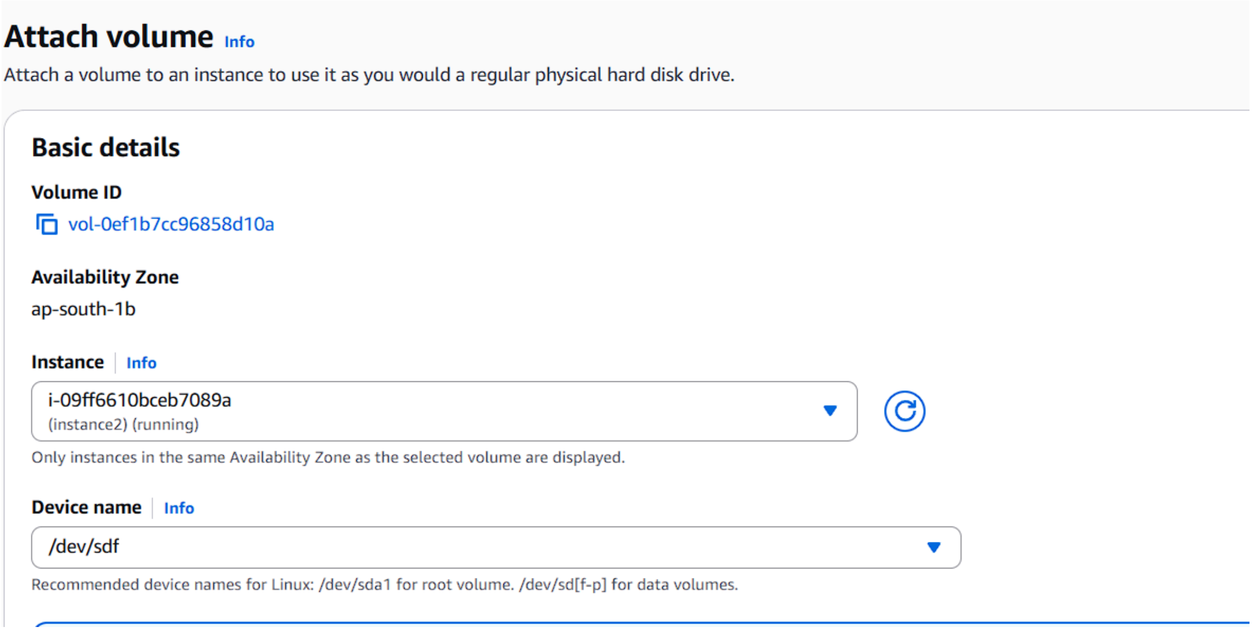
Snapshots (1/1) Info

☑	Name	Snapshot ID	Full snapshot size	Volume size	Description	Snapshot status
☑		snap-04ecc5bad887001bb	152.5 MiB	5 GiB	my snapshot	Completed



Volumes (1/4) Info

☐	Name	Volume ID	Type	Size	IOPS	Throughput	me ID	Crea
☐		vol-088dd3b825ad398de	gp3	8 GiB	3000	125		202
☐		vol-05c03ab458f14d907	gp3	5 GiB	3000	125		202
☐		vol-06ba30447ee64e3c5	gp3	8 GiB	3000	125		202
☑		vol-0ef1b7cc96858d10a	gp3	5 GiB	3000	125		202



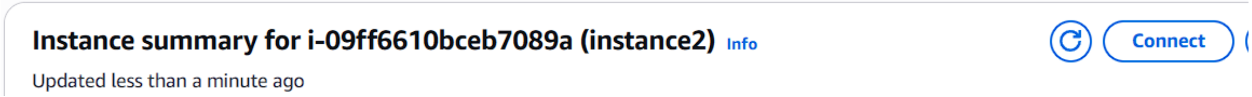
Attach a volume to an instance to use it as you would a regular physical hard disk drive.

Volume ID
vol-0ef1b7cc96858d10a

Availability Zone
ap-south-1b

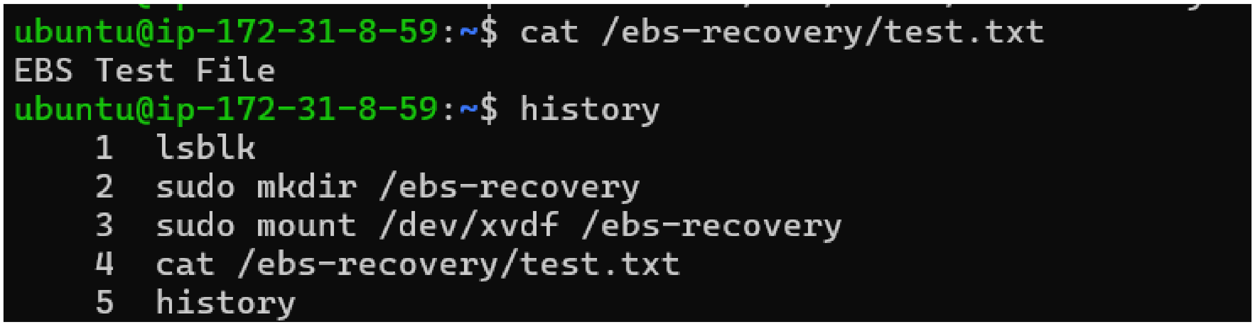
Instance | Info
i-09ff6610bceb7089a
(instance2) (running)
Only instances in the same Availability Zone as the selected volume are displayed.

Device name | Info
/dev/sdf
Recommended device names for Linux: /dev/sda1 for root volume. /dev/sd[f-p] for data volumes.

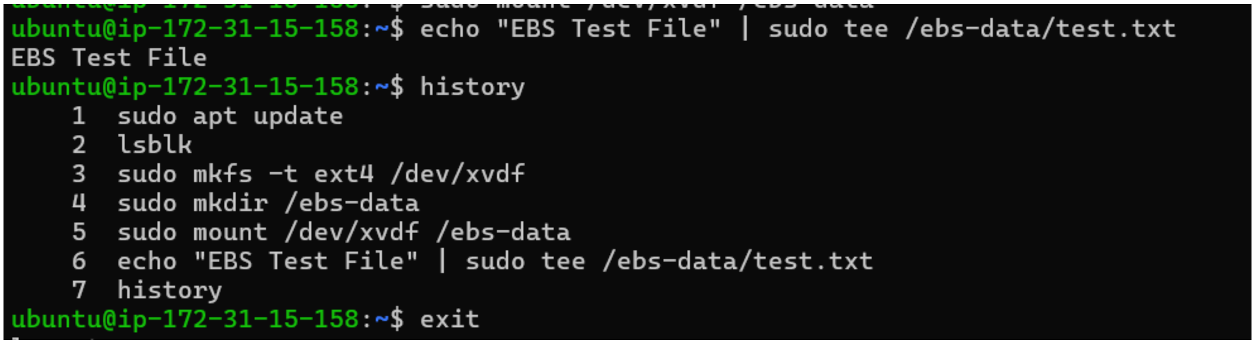


Instance summary for i-09ff6610bceb7089a (instance2)

Updated less than a minute ago



```
ubuntu@ip-172-31-8-59:~$ cat /ebs-recovery/test.txt
EBS Test File
ubuntu@ip-172-31-8-59:~$ history
    1  lsblk
    2  sudo mkdir /ebs-recovery
    3  sudo mount /dev/xvdf /ebs-recovery
    4  cat /ebs-recovery/test.txt
    5  history
```



```
ubuntu@ip-172-31-15-158:~$ sudo mount /dev/xvdf /ebs-data
ubuntu@ip-172-31-15-158:~$ echo "EBS Test File" | sudo tee /ebs-data/test.txt
EBS Test File
ubuntu@ip-172-31-15-158:~$ history
    1  sudo apt update
    2  lsblk
    3  sudo mkfs -t ext4 /dev/xvdf
    4  sudo mkdir /ebs-data
    5  sudo mount /dev/xvdf /ebs-data
    6  echo "EBS Test File" | sudo tee /ebs-data/test.txt
    7  history
ubuntu@ip-172-31-15-158:~$ exit
```

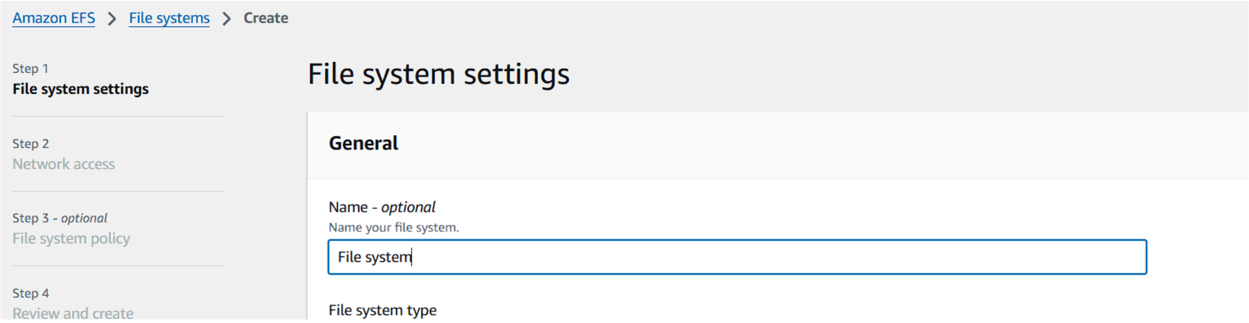
EFS (Elastic File System):

- Created an EFS file system with mount targets in each AZ.

- Installed EFS utilities on EC2 instances:

```
bash
CopyEdit
sudo yum install -y amazon-efs-utils
sudo mkdir /efs-data
sudo mount -t efs -o tls fs-xxxx:/ /efs-data
```

- Mounted EFS on multiple EC2s.
- Verified real-time file sharing across instances.



Amazon EFS > File systems > Create

Step 1
File system settings

Step 2
Network access

Step 3 - *optional*
File system policy

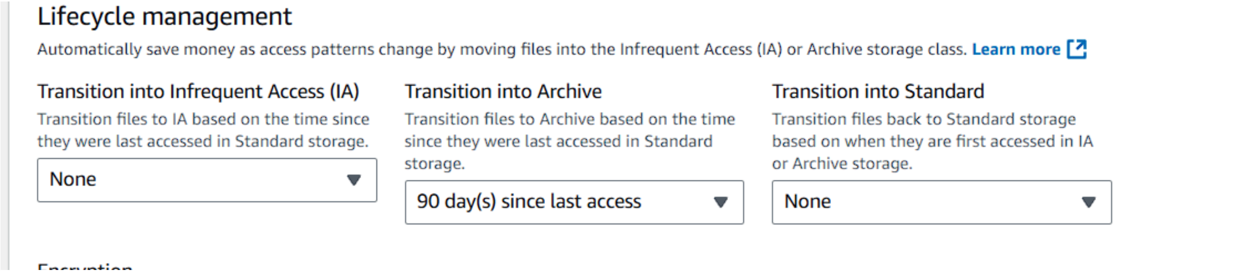
Step 4
Review and create

File system settings

General

Name - *optional*
Name your file system.
File system

File system type



Lifecycle management

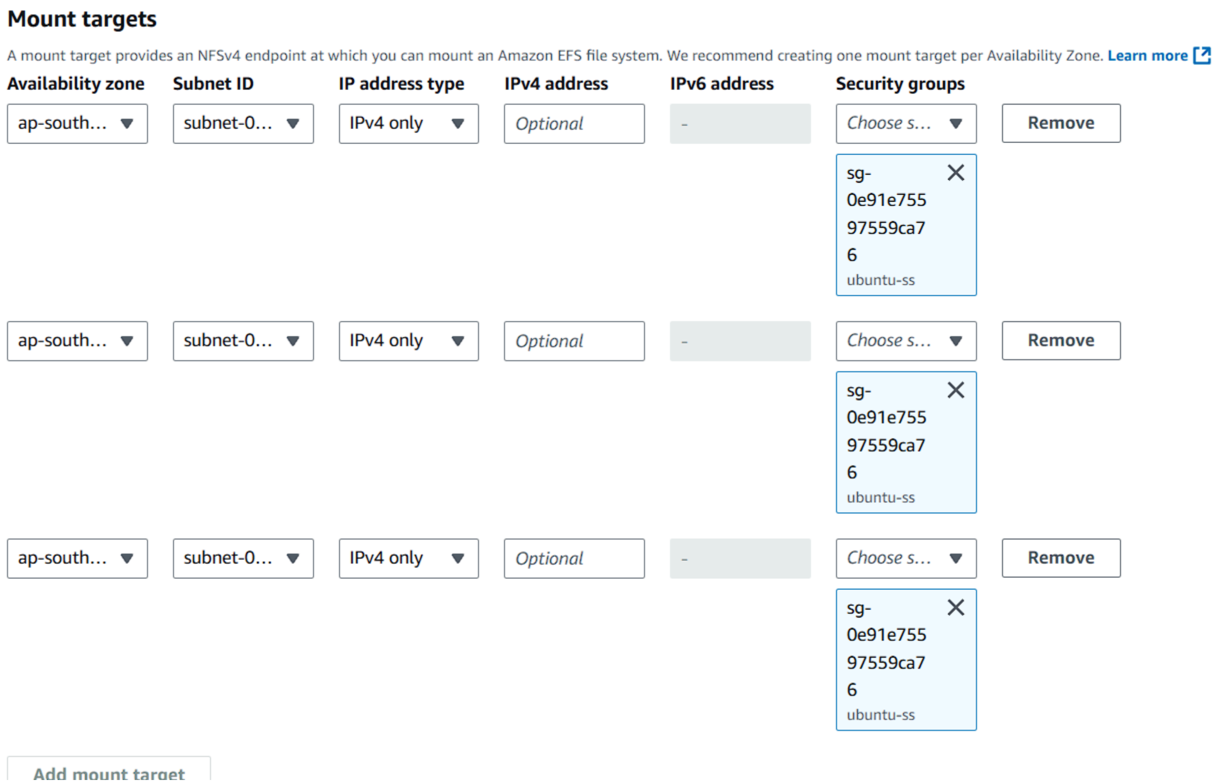
Automatically save money as access patterns change by moving files into the Infrequent Access (IA) or Archive storage class. **Learn more**

Transition into Infrequent Access (IA)
Transition files to IA based on the time since they were last accessed in Standard storage.
None

Transition into Archive
Transition files to Archive based on the time since they were last accessed in Standard storage.
90 day(s) since last access

Transition into Standard
Transition files back to Standard storage based on when they are first accessed in IA or Archive storage.
None

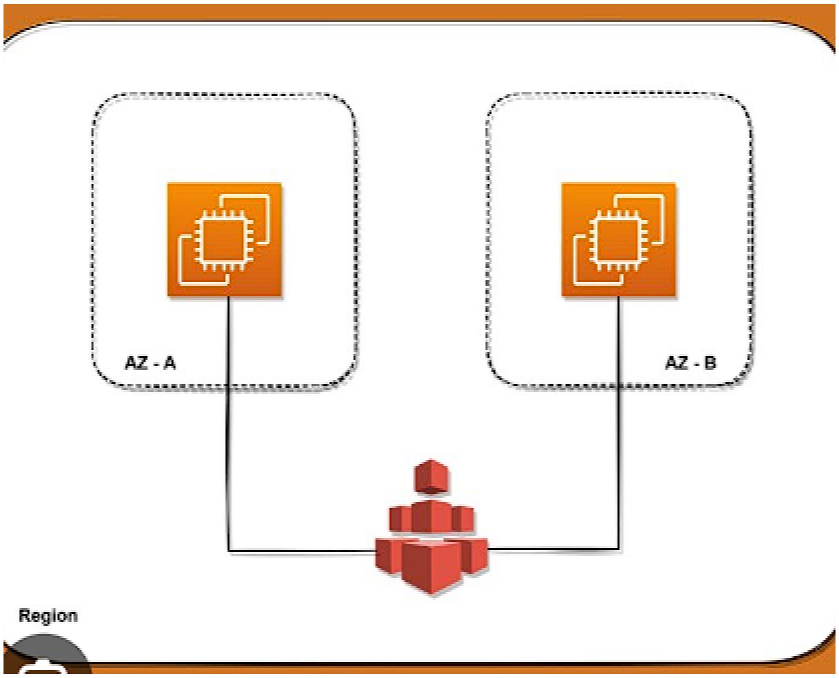
Encryption

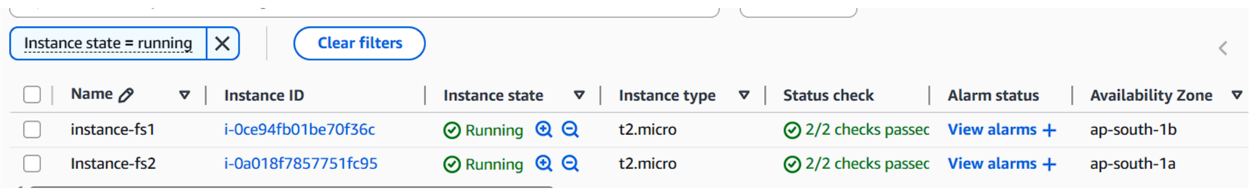


Mount targets

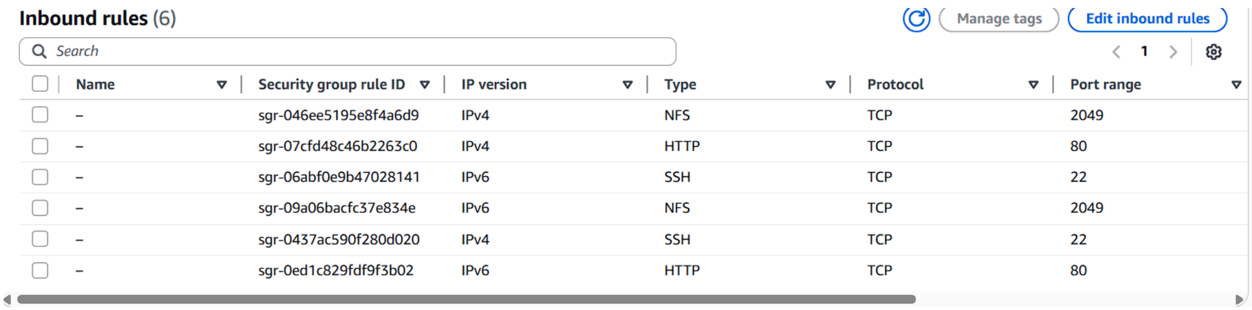
A mount target provides an NFSv4 endpoint at which you can mount an Amazon EFS file system. We recommend creating one mount target per Availability Zone. Learn more

Availability zone	Subnet ID	IP address type	IPv4 address	IPv6 address	Security groups	
ap-south... ▼	subnet-0... ▼	IPv4 only ▼	*Optional*	-	*Choose s...* ▼ sg-0e91e75597559ca76 ✕ ubuntu-ss	Remove
ap-south... ▼	subnet-0... ▼	IPv4 only ▼	*Optional*	-	*Choose s...* ▼ sg-0e91e75597559ca76 ✕ ubuntu-ss	Remove
ap-south... ▼	subnet-0... ▼	IPv4 only ▼	*Optional*	-	*Choose s...* ▼ sg-0e91e75597559ca76 ✕ ubuntu-ss	Remove



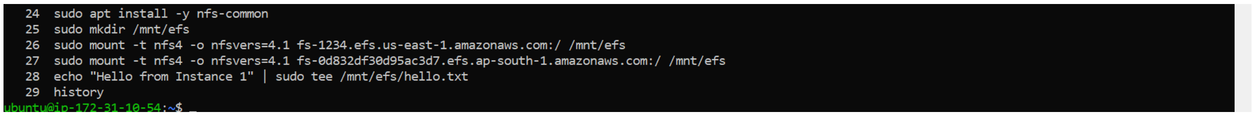


□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
□	instance-fs1	i-0ce94fb01be70f36c	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-south-1b
□	Instance-fs2	i-0a018f7857751fc95	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-south-1a

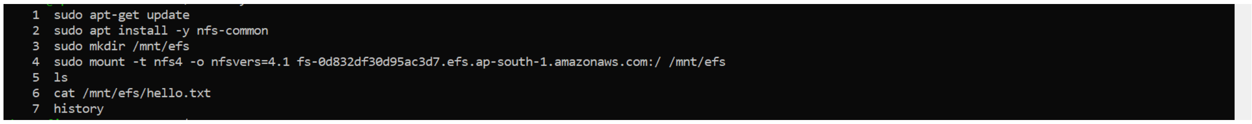


Inbound rules (6)

□	Name	Security group rule ID	IP version	Type	Protocol	Port range
□	–	sgr-046ee5195e8f4a6d9	IPv4	NFS	TCP	2049
□	–	sgr-07cfd48c46b2263c0	IPv4	HTTP	TCP	80
□	–	sgr-06abf0e9b47028141	IPv6	SSH	TCP	22
□	–	sgr-09a06bacfc37e834e	IPv6	NFS	TCP	2049
□	–	sgr-0437ac590f280d020	IPv4	SSH	TCP	22
□	–	sgr-0ed1c829fdf9f3b02	IPv6	HTTP	TCP	80



```
   24  sudo apt install -y nfs-common
   25  sudo mkdir /mnt/efs
   26  sudo mount -t nfs4 -o nfsvers=4.1 fs-1234.efs.us-east-1.amazonaws.com:/ /mnt/efs
   27  sudo mount -t nfs4 -o nfsvers=4.1 fs-0d832df30d95ac3d7.efs.ap-south-1.amazonaws.com:/ /mnt/efs
   28  echo "Hello from Instance 1" | sudo tee /mnt/efs/hello.txt
   29  history
ubuntu@ip-172-31-10-54:~$ _
```

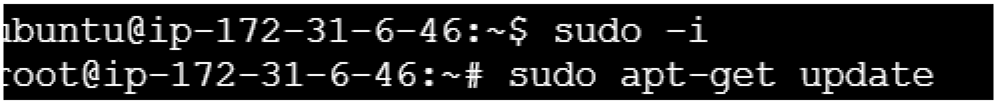


```
    1  sudo apt-get update
    2  sudo apt install -y nfs-common
    3  sudo mkdir /mnt/efs
    4  sudo mount -t nfs4 -o nfsvers=4.1 fs-0d832df30d95ac3d7.efs.ap-south-1.amazonaws.com:/ /mnt/efs
    5  ls
    6  cat /mnt/efs/hello.txt
    7  history
```

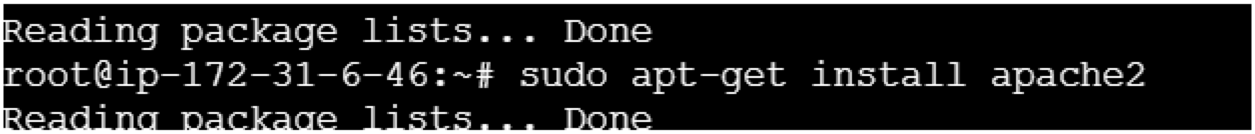
2. Autoscaling & Load Balancer Setup

- Created a Launch Template from a configured EC2 instance.
- Configured an Auto Scaling Group:
    - Minimum: 1
    - Desired: 1

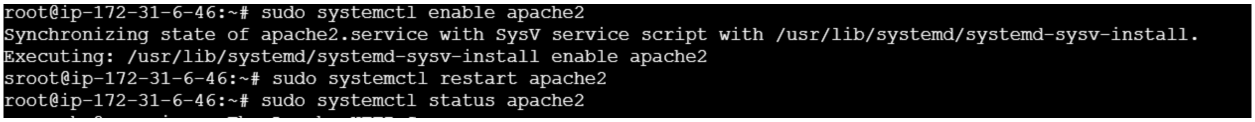
  Maximum: 2
- Attached Application Load Balancer for distributing traffic.
- Terminated one instance to test auto-recovery by ASG.
- ASG successfully launched a replacement instance.



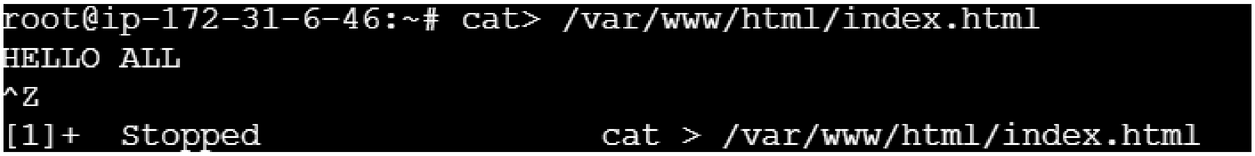
```
ıbuntu@ip-172-31-6-46:~$ sudo -i
root@ip-172-31-6-46:~# sudo apt-get update
```



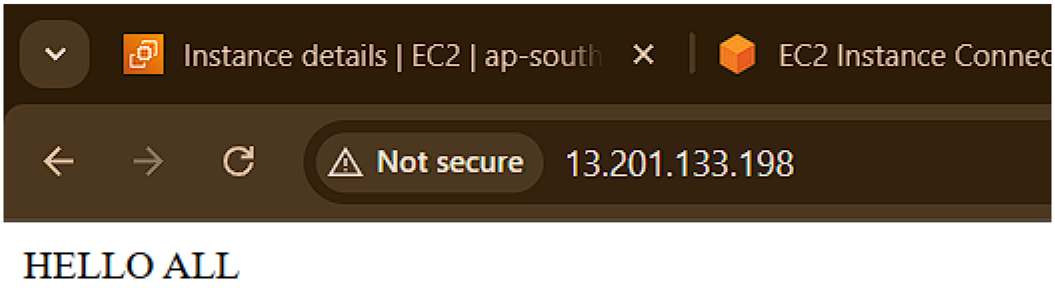
```
Reading package lists... Done
root@ip-172-31-6-46:~# sudo apt-get install apache2
Reading package lists... Done
```



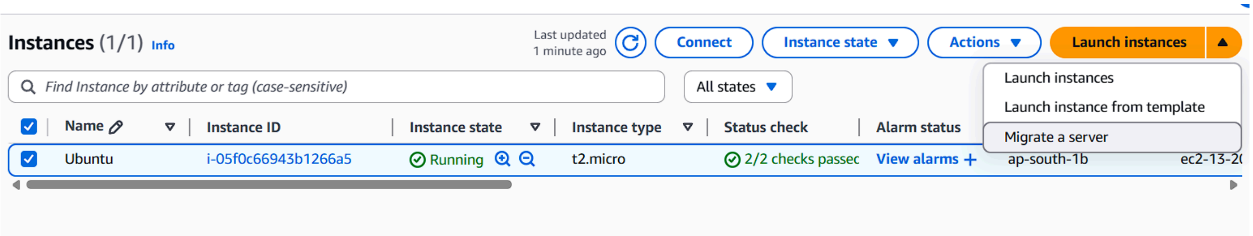
```
root@ip-172-31-6-46:~# sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
sroot@ip-172-31-6-46:~# sudo systemctl restart apache2
root@ip-172-31-6-46:~# sudo systemctl status apache2
```



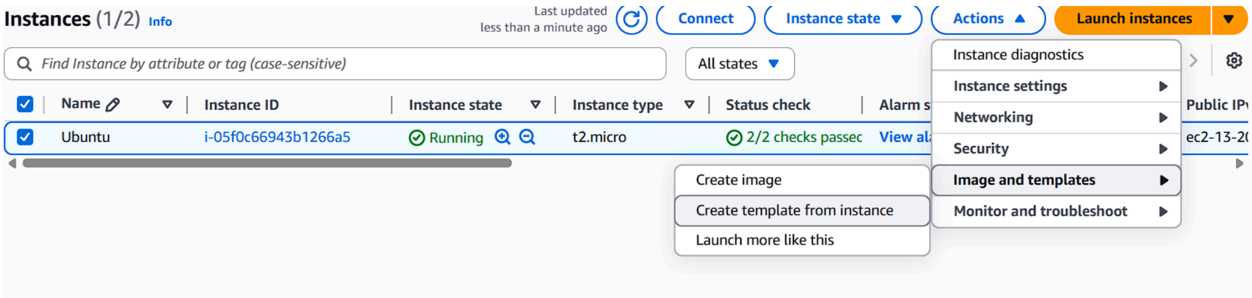
```
root@ip-172-31-6-46:~# cat> /var/www/html/index.html
HELLO ALL
^Z
[1]+  Stopped                 cat > /var/www/html/index.html
```



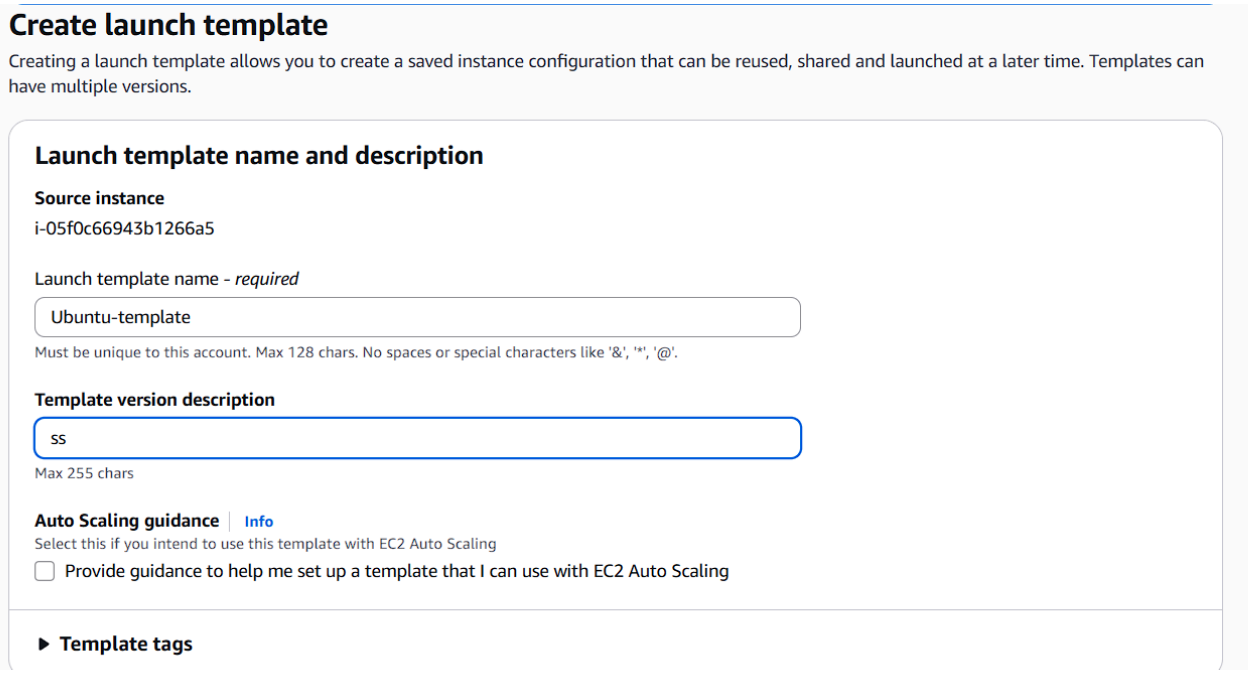
HELLO ALL



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status		
☑	Ubuntu	i-05f0c66943b1266a5	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-south-1b	ec2-13-2(



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm s	Public IP\
☑	Ubuntu	i-05f0c66943b1266a5	✓ Running	t2.micro	✓ 2/2 checks passec	View al	ec2-13-2(



Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Source instance

i-05f0c66943b1266a5

Launch template name - *required*

Ubuntu-template

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

ss

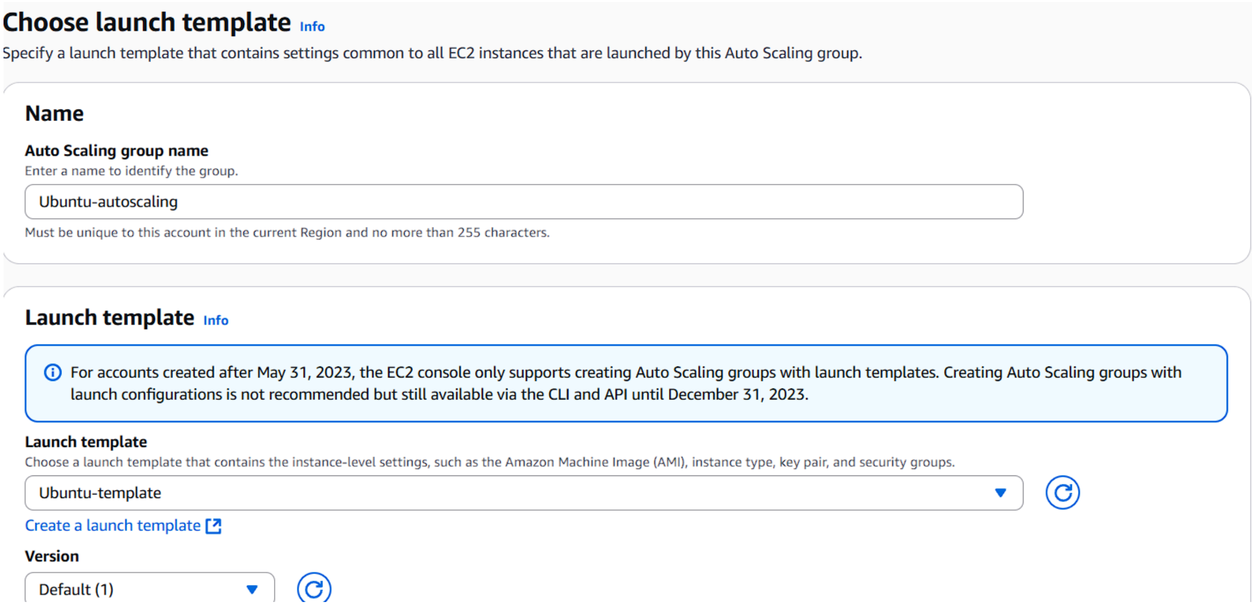
Max 255 chars

Auto Scaling guidance

Select this if you intend to use this template with EC2 Auto Scaling

□ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

▶ **Template tags**



Choose launch template

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name

Auto Scaling group name

Enter a name to identify the group.

Ubuntu-autoscaling

Must be unique to this account in the current Region and no more than 255 characters.

Launch template

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

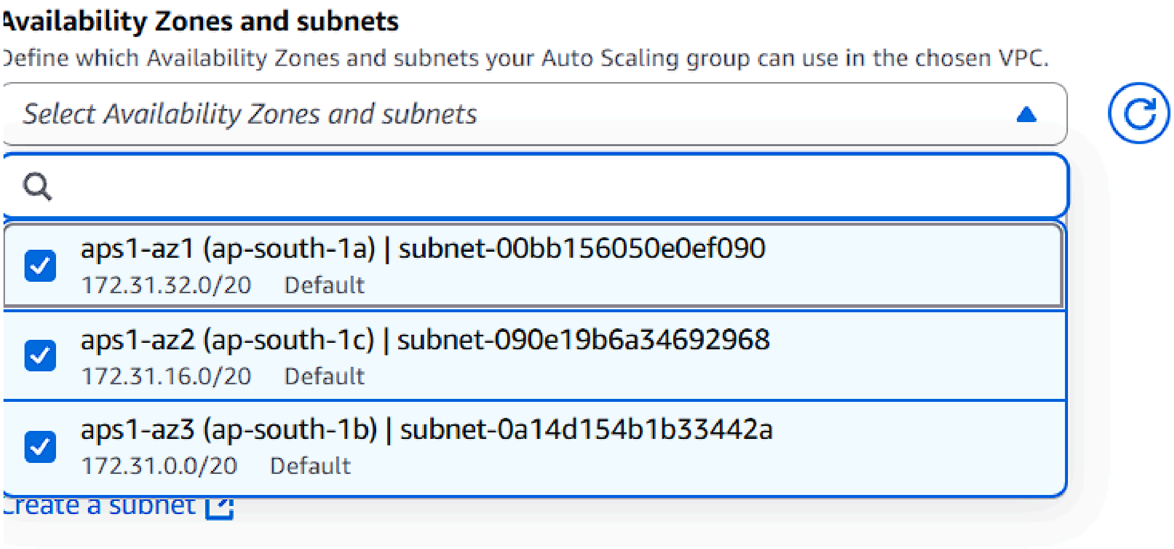
Launch template

Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Ubuntu-template ▼

Version

Default (1) ▼



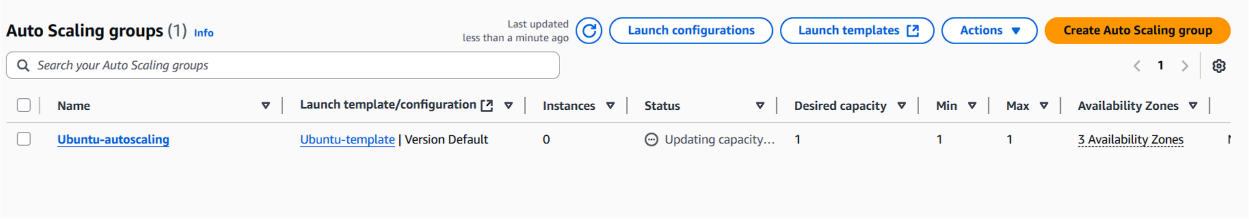
Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

☑ aps1-az1 (ap-south-1a) | subnet-00bb156050e0ef090
172.31.32.0/20 Default

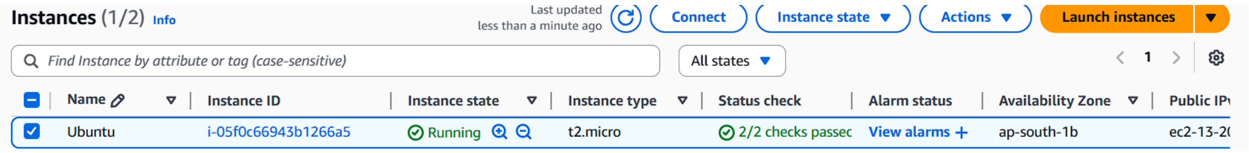
☑ aps1-az2 (ap-south-1c) | subnet-090e19b6a34692968
172.31.16.0/20 Default

☑ aps1-az3 (ap-south-1b) | subnet-0a14d154b1b33442a
172.31.0.0/20 Default



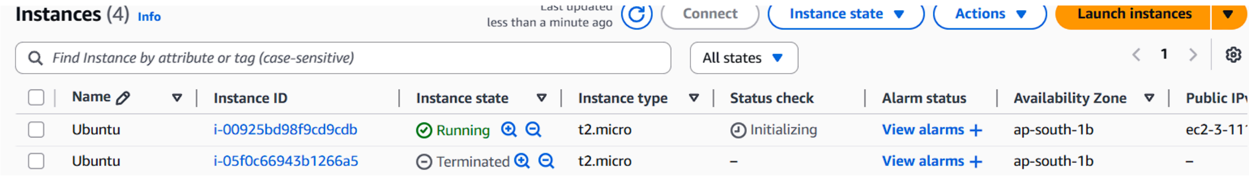
Auto Scaling groups (1) Info

□	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
□	Ubuntu-autoscaling	Ubuntu-template | Version Default	0	Updating capacity…	1	1	1	3 Availability Zones



Instances (1/2) Info

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv
☑	Ubuntu	i-05f0c66943b1266a5	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-13-2(



Instances (4) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv
□	Ubuntu	i-00925bd98f9cd9cdb	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-3-11
□	Ubuntu	i-05f0c66943b1266a5	Terminated	t2.micro	–	View alarms +	ap-south-1b	–

3. Comparison: EBS vs EFS vs S3

Feature	EBS	EFS	S3
Type	Block Storage	Network File System (NFS)	Object Storage
Access	Single EC2 instance	Multiple EC2s (shared access)	Accessible via HTTP/S3 API

Feature	EBS	EFS	S3
Use Case	Databases, boot volumes	Web servers, shared home dirs	Backup, data lake, media files
Scalability	Resizing requires manual	Automatically scalable	Infinitely scalable
Pricing	Cheaper than EFS for regular use	More expensive, billed per GB and I/O	Very cost-effective, pay-per-GB
Durability	High	High	99.999999999% (11 nines)

4. Vertical vs Horizontal Scaling

Vertical Scaling

- **Definition:** Increasing instance size (CPU, RAM)
- **Example:** Upgrading from t2.micro to m5.large
- **Pros:**
  - Simpler to implement
  - No code changes required
- **Cons:**
  - Limited by hardware
  - Downtime during upgrade

Horizontal Scaling

- **Definition:** Adding more instances to share the load
- **Example:** Netflix adds EC2s behind a load balancer
- **Pros:**
  - Better fault tolerance
  - High availability
- **Cons:**

  - Needs load balancing
  - More complex architecture

5. Troubleshooting Autoscaling Issues

- **Tested Error:** ASG created with wrong Launch Template ID
- **Error Observed:** ASG fails to launch new instances
- **Fix:** Updated ASG with the correct Launch Template version
- **Lesson Learned:** Always double-check Launch Template settings, including AMI, user data, and security groups

6. AWS Lambda with S3 Trigger

- Created a Lambda function in Python.
- Set trigger: S3 bucket → On file upload.
- Code summary:

  ```
  python
  CopyEdit
  def lambda_handler(event, context):
      print("New file uploaded:", event)
  ```

- Uploaded a file to S3 → Lambda triggered successfully.

7. AWS Cost Monitoring

- Opened **AWS Cost Explorer**
- Filtered data by service (EC2, EBS, EFS)
- Observed:
  - Highest cost from EFS due to I/O-based pricing

- EBS had predictable cost per GB

- Set budget alerts for Free Tier tracking

---

8. GitHub Portfolio

- Created GitHub repository: `aws-automation-scripts`
- Uploaded:
    - EC2 provisioning script using Boto3
    - Auto Scaling Group automation
    - EBS snapshot backup script
- Each script includes README with usage instructions