

Assignment: EC2 Instance Creation

Name: Uday Kiran Ailapaka

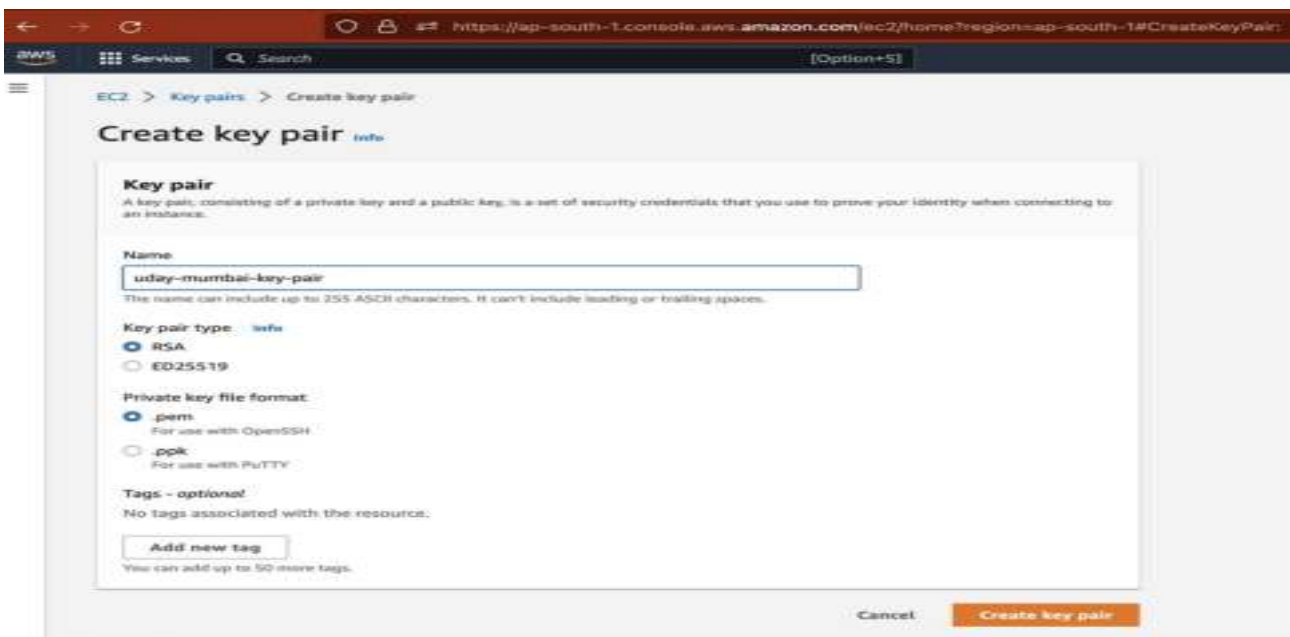
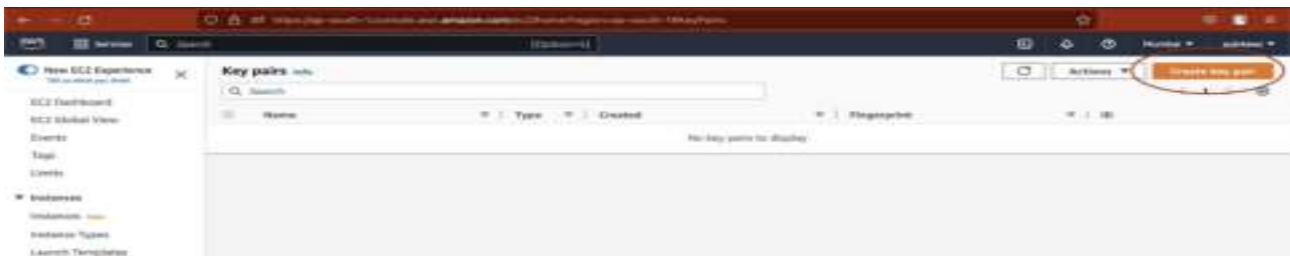
AWS Region: Mumbai (ap-south-1)

Creation of a key pair:

A key pair is required to connect to the EC2 instances created. We have to associate a key pair while creating an EC2 instance.

Navigation: EC2 -> Network & Security -> Key Pairs

Key pair name: **uday-mumbai-key-pair.pem**

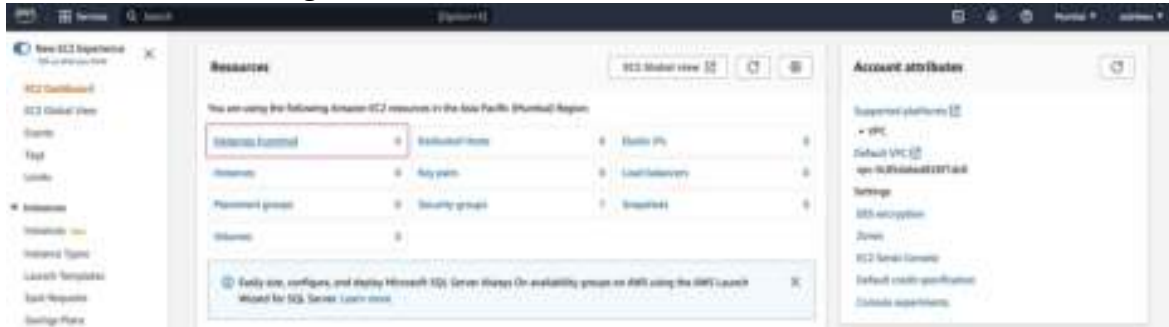


Note:- Then a new key pair is downloaded 'uday-mumbai-key-pair.pem'

Creation of a EC2 Instance:

We are going to create an Amazon Linux AWS EC2 instance.

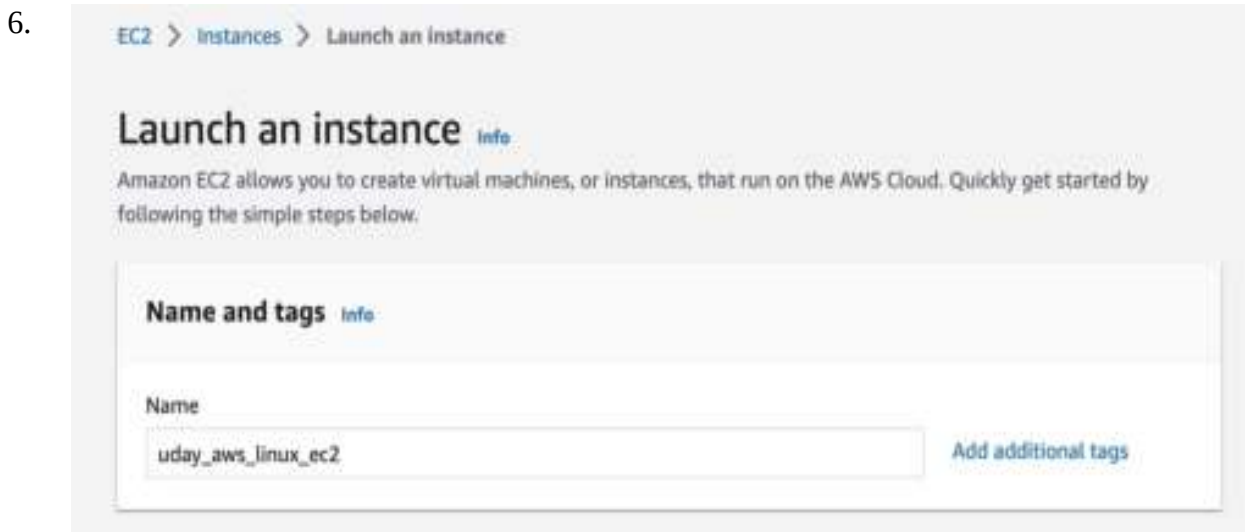
1. Login to AWS console.
2. Navigate to Services -> Compute -> EC2
3. Select Instances running
- 4.



Click Launch instances



Name the EC2 instance: uday_aws_linux_ec2



Select OS image and the version: (Amazon Linux 2 Kernel 5.10 AMI)

7.

Application and OS Images (Amazon Machine Image) [Info](#)
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including T000s of application and OS images.

Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE

Amazon Machine Image (AMI)
Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
ami-074dc0a6f6c764218 (64-bit x86) / ami-019774eScaffd1585 (64-bit Arm)
Virtualization: hvm ENA enabled: true Root device type: ebs **Free tier eligible**

Description
Amazon Linux 2 Kernel 5.10 AMI 2.0.20221103.3 x86_64 HVM gp2

Architecture: **64-bit (x86)** AMI ID: **ami-074dc0a6f6c764218** **Verified provider**

Select Instance type as t2.micro (free tier eligible):

8. Select the key pair created in the first step above (uday-mumbai-key-pair)

9.

Use

Instance type [Info](#)

Instance type

t2.micro **Free tier eligible**
Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0124 USD per Hour
On-Demand Windows pricing: 0.017 USD per Hour [Compare instance types](#)

Key pair (login) [Info](#)
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*
uday-mumbai-key-pair [Create new key pair](#)

the existing default settings for **Network Settings**:

▼ **Network settings** [Info](#) Edit

Network [Info](#)
vpc-0c85da6ed826f1dc8

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

Allow SSH traffic from Helps you connect to your instance Anywhere
0.0.0.0/0

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. ×

10. Select the required storage configuration: (8 GB of storage here)

▼ **Configure storage** [Info](#) Advanced

1x 8 GiB gp2 Root volume (Not encrypted)

ℹ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage ×

Add new volume

0 x File systems Edit

11. Select number of instances and click Launch Instances: (1 here)

▼ Summary

Number of instances [Info](#)

1

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...[read more](#)
ami-074dc0a6f6c764218

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel **Launch instance**

12. Success message is displayed:

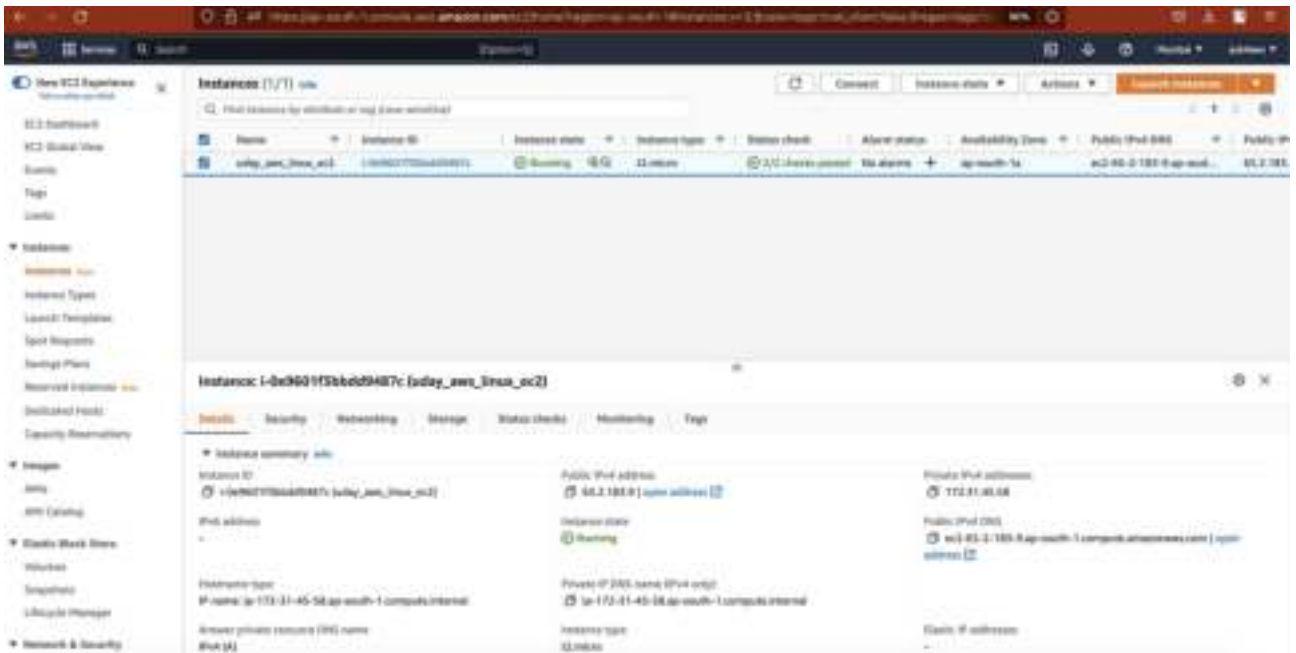
13.
Go

EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance i-015E7T0448H8214

[View Launch log](#)

to EC2 Dashboard to see the status of the newly created EC2 instance:



Connecting to EC2 Instances:

1. Click on the instance id of the EC2 instance created above.
2. Click on SSH client section and follow the instructions given.
3. Copy the ssh command given under example. ***'ssh -i "uday-mumbai-key-pair.pem" ec2-user@ec2-65-2-183-9.ap-south-1.compute.amazonaws.com'***
4. In MacOSX, launch terminal and cd to the location where the pem file is downloaded. Now run the copied command.

Connect to instance [Info](#)

Connect to your instance `i-0e9601f3b8dd9487c` (uday_aws_linux_ec2) using any of these options

EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Instance ID

`i-0e9601f3b8dd9487c` (uday_aws_linux_ec2)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is `uday-mumbai-key-pair.pem`
3. Run this command, if necessary, to ensure your key is not publicly viewable.
`chmod 400 uday-mumbai-key-pair.pem`
4. Connect to your instance using its Public DNS:
`ec2-65-2-183-9.ap-south-1.compute.amazonaws.com`

Example:

```
ssh -i "uday-mumbai-key-pair.pem" ec2-user@ec2-65-2-183-9.ap-south-1.compute.amazonaws.com
```

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

```
ec2-user@ip-172-31-45-58:~  
> ls  
uday-mumbai-key-pair.pem  
  
Desktop/Cloud&Devops/Workspace on ☁ (ap-south-1)  
> chmod 400 uday-mumbai-key-pair.pem  
  
Desktop/Cloud&Devops/Workspace on ☁ (ap-south-1)  
> ssh -i "uday-mumbai-key-pair.pem" ec2-user@ec2-65-2-183-9.ap-south-1.compute.a  
mazonaws.com  
  
  _ | _ | _ )  
  _ | ( _ /   Amazon Linux 2 AMI  
  _ | \ _ | _ |  
  
https://aws.amazon.com/amazon-linux-2/  
1 package(s) needed for security, out of 1 available  
Run "sudo yum update" to apply all updates.  
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file  
or directory  
[ec2-user@ip-172-31-45-58 ~]$
```

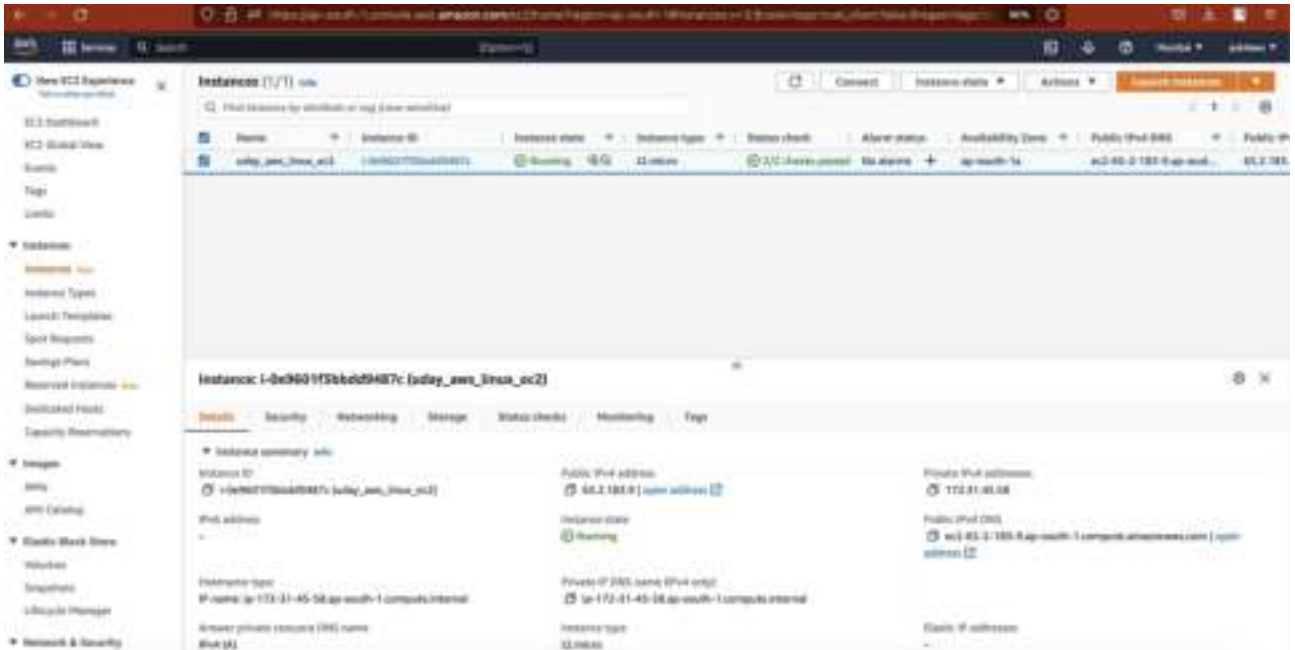
The EC2 Instance is successfully created and connect from MacOSX terminal using ssh

Create Elastic IP and Associate with EC2 instance:

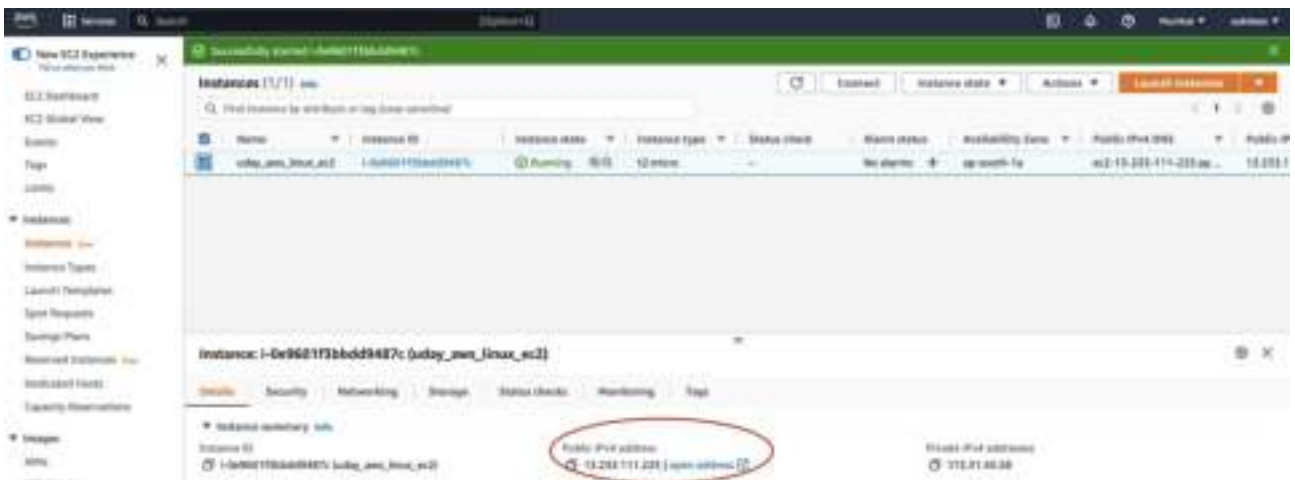
An EC2 instance has public and private ipv4 addressess. Private IP we use to connect within internal AWS services like Connect b/w multiple EC2 instances. Public IP we use to expose the system to Public connections. But, public address is dynamic i.e it changes when instance is restarted. So, we cannot use it to host a website. For that we can have a static IP address and

associate it with an EC2 instance. This static IP can be used to host a website with the DNS. This static IP is called as Elastic IP in AWS.

1. Public IP before restarting the EC2 instance: (65.2.183.9)



2. Public IP changed after restarting the same EC2 instance: (13.253.111.225)



3. Navigate to **EC2 -> Network & Security -> Elastic IPs -> Allocate Elastic IP address** and click allocate



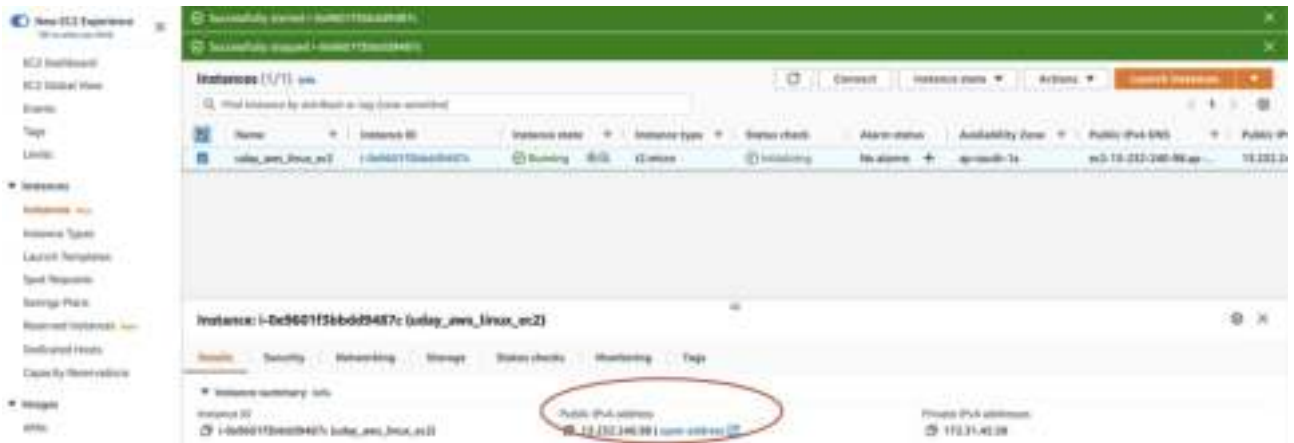
4. Select the created Elastic IP address, expand the actions and select 'Associate Elastic IP address'



5. Choose the EC2 instance and click Associate



6. The newly created static IP address is associated with the EC2 instance even after restarting.



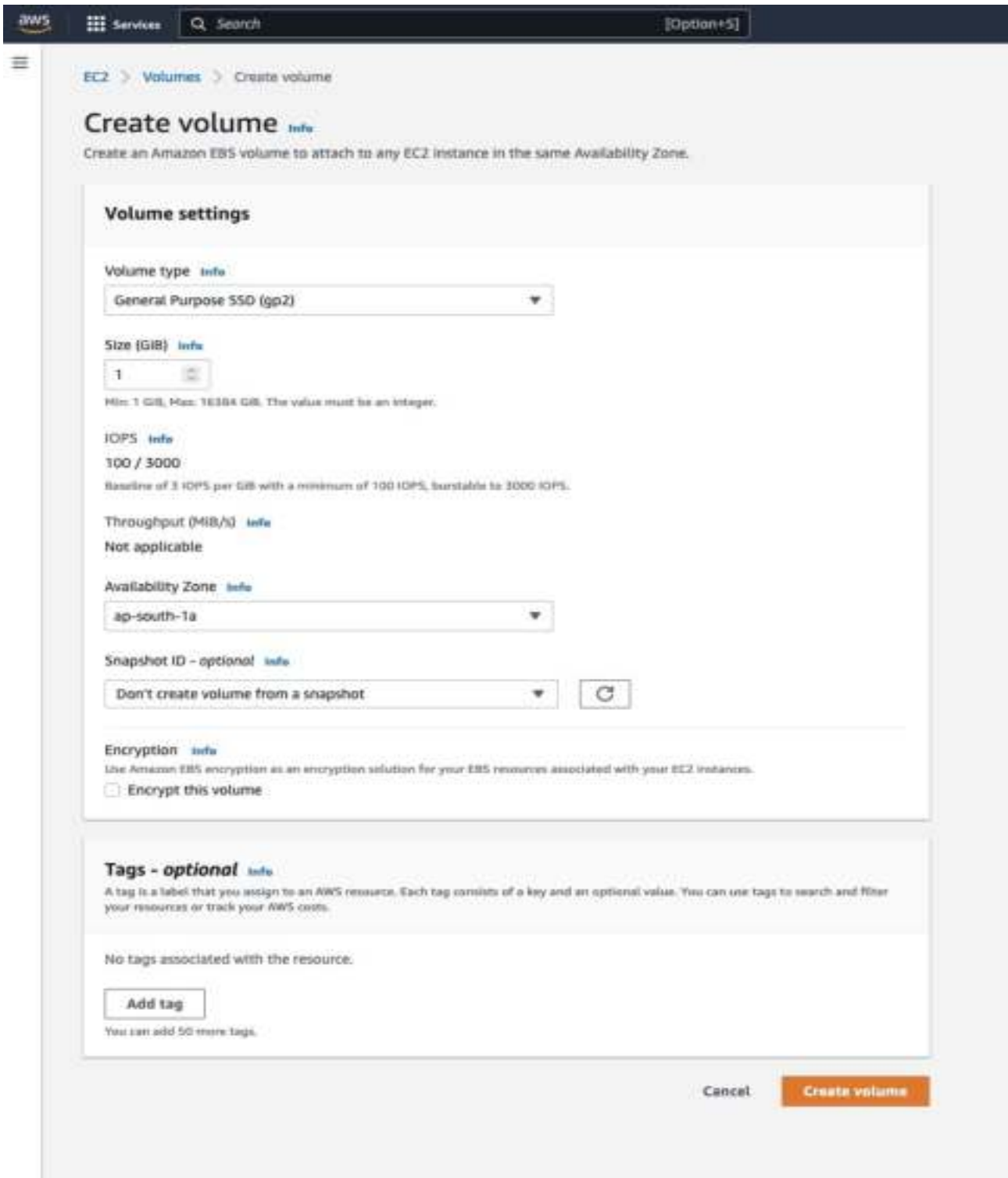
Create Elastic Block Store:

Used to extend the storage attached with EC2 instances without the need for downtime. You can mount this persistent storage with any EC2 instance.

1. Navigate to **EC2 -> Elastic Block Store -> Volumes** and click 'Create volume'



2. Select the size and create volume



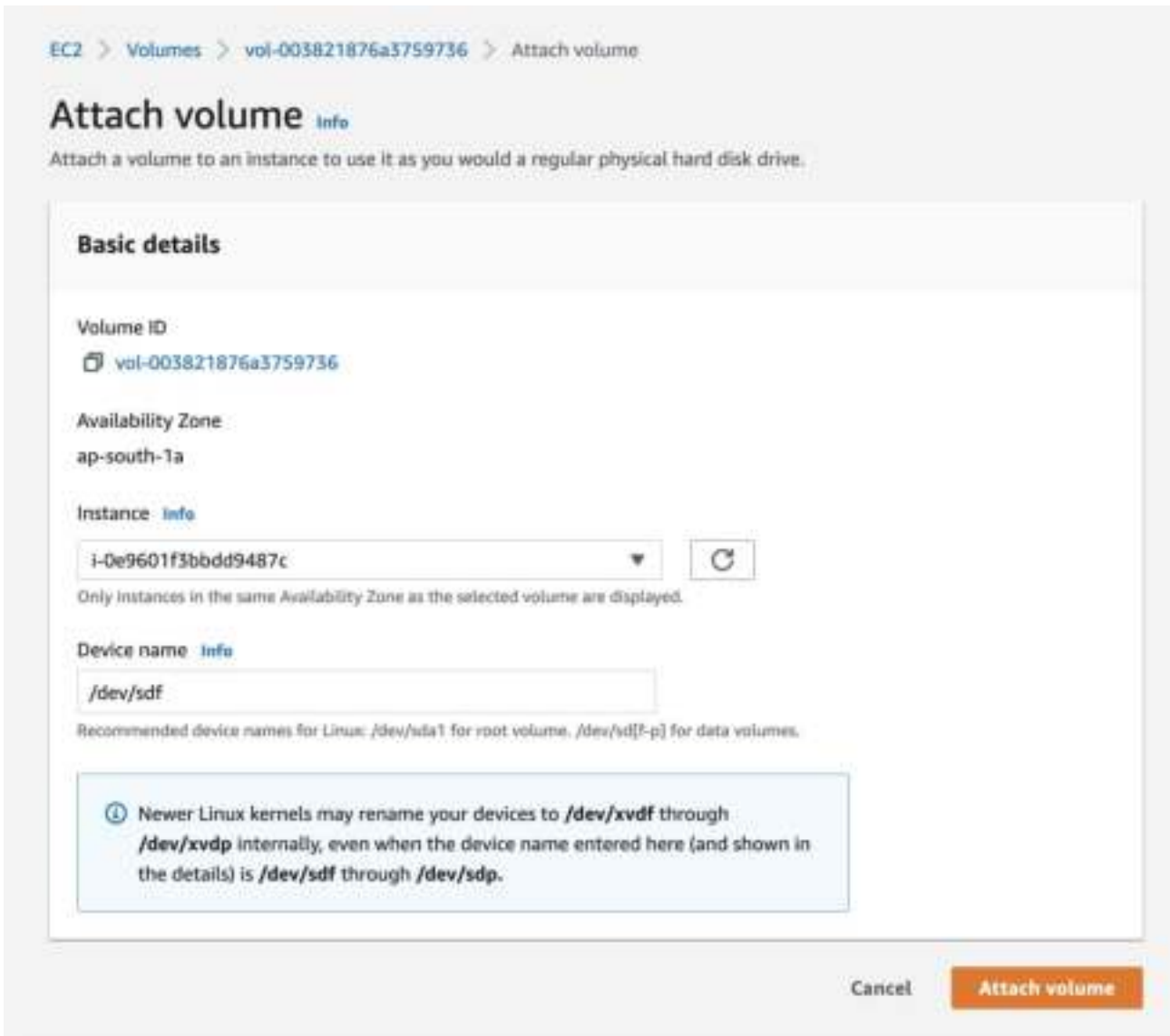
4. The newly created EBS volume state is 'Available'



5. Select the newly create EBS volume and choose 'Attach Volume'



6. Choose the required instance and click attach volume



7. The volume state of EBS store is now changed to 'In Use'



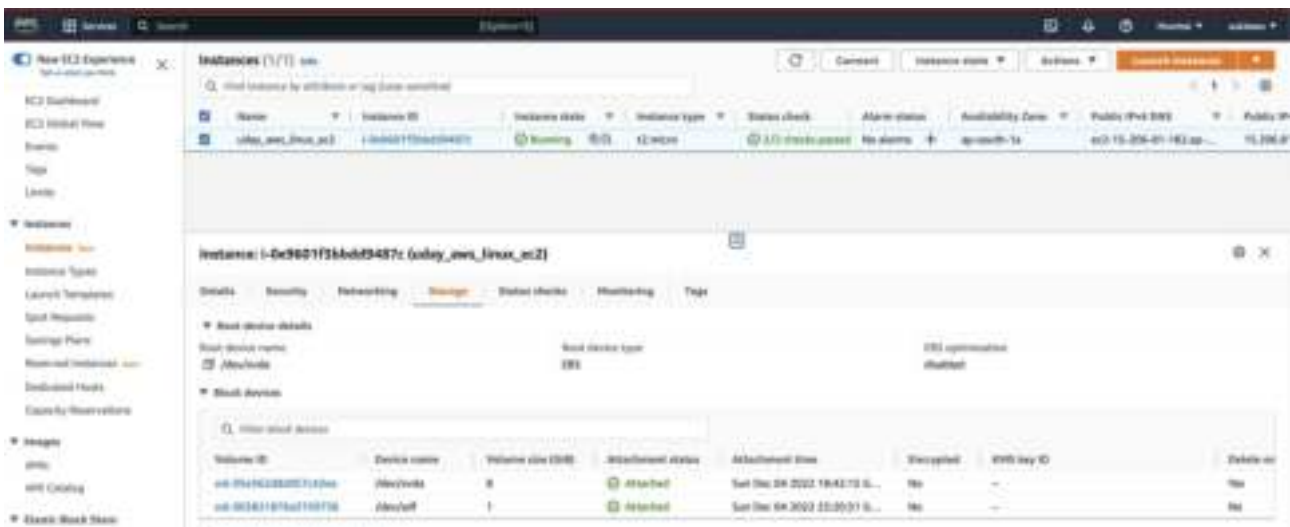
8. The EC2 disk allocation before new EBS is attached

```
[ec2-user@ip-172-31-45-58 ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO  TYPE MOUNTPOINT
xvda        202:0    0   8G  0  disk
`-xvda1     202:1    0   8G  0  part /
[ec2-user@ip-172-31-45-58 ~]$
```

9. The EC2 disk allocation after new EBS is attached

```
[ec2-user@ip-172-31-45-58 ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO  TYPE MOUNTPOINT
xvda        202:0    0   8G  0  disk
`-xvda1     202:1    0   8G  0  part /
xvdf        202:80    0   1G  0  disk
[ec2-user@ip-172-31-45-58 ~]$
```

10. The newly created EBS is seen attached to the EC2 instance



11. Connect to EC2 instance and Mount the new EBS:

- (a) sudo su

(b) lsblk

```
[ec2-user@ip-172-31-45-58 ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda        202:0    0   8G  0 disk
└─xvda1     202:1    0   8G  0 part /
xvdf        202:80   0   1G  0 disk
```

(c) mkdir storage

(d) fdisk -l

```
[root@ip-172-31-45-58 ec2-user]# fdisk -l
Disk /dev/xvda: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: DA90AB95-8C11-4BAE-9157-C48213FBAD0C

Device            Start      End  Sectors  Size Type
/dev/xvda1        4096 16777182 16773087   8G Linux filesystem
/dev/xvda128      2048      4095     2048    1M BIOS boot

Partition table entries are not in disk order.

Disk /dev/xvdf: 1 GiB, 1073741824 bytes, 2097152 sectors
```

(e) mkfs -t xfs /dev/xvdf (makes file system)

(f) mount -t xfs /dev/xvdf /home/ec2-user/storage/

(g) lsblk

```
[root@ip-172-31-45-58 ec2-user]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda        202:0    0   8G  0 disk
└─xvda1     202:1    0   8G  0 part /
xvdf        202:80   0   1G  0 disk /home/ec2-user/storage
```

12. Create somefiles in the filesystem

```

[ec2-user@ip-172-31-45-58 ~]$ sudo su
[root@ip-172-31-45-58 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-45-58 ec2-user]# cd storage/
[root@ip-172-31-45-58 storage]# pwd
/home/ec2-user/storage
[root@ip-172-31-45-58 storage]# touch {1..10}.txt
[root@ip-172-31-45-58 storage]# ls
1.txt 10.txt 2.txt 3.txt 4.txt 5.txt 6.txt 7.txt 8.txt 9.txt

```

13. Unmount the file system and detach the volume from the EC2 instance

```

[root@ip-172-31-45-58 storage]# cd ..
[root@ip-172-31-45-58 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-45-58 ec2-user]# umount /home/ec2-user/storage/
bash: umount: command not found
[root@ip-172-31-45-58 ec2-user]# mount /home/ec2-user/storage/
[root@ip-172-31-45-58 ec2-user]# df -hT

```

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
devtmpfs	devtmpfs	474M	0	474M	0%	/dev
tmpfs	tmpfs	483M	0	483M	0%	/dev/shm
tmpfs	tmpfs	483M	528K	483M	1%	/run
tmpfs	tmpfs	483M	0	483M	0%	/sys/fs/cgroup
/dev/xvda1	xfs	8.0G	1.6G	6.5G	20%	/
tmpfs	tmpfs	97M	0	97M	0%	/run/user/1000
tmpfs	tmpfs	97M	0	97M	0%	/run/user/0

14. Create another machine **uday_aws_linux_ec2_A**.



15. Attach the volume storage to the new machine

Attach volume Info

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

Basic details


Volume ID

 vol-003821876a3759736

Availability Zone

ap-south-1a

Instance Info


i-022bd625a4d63acc8 

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.

 Newer Linux kernels may rename your devices to **/dev/xvdf** through **/dev/xvdp** internally, even when the device name entered here (and shown in the details) is **/dev/sdf** through **/dev/sdp**.

Cancel

Attach volume

16. Mount the file system and access the already created files.

```

[ec2-user@ip-172-31-46-167 ~]$ sudo su
[root@ip-172-31-46-167 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-46-167 ec2-user]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda        202:0    0   8G  0 disk
└─xvda1    202:1    0   8G  0 part /
xvdf        202:80   0   1G  0 disk
[root@ip-172-31-46-167 ec2-user]# fdisk -l
Disk /dev/xvda: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: DA90AB95-8C11-4BAE-9157-C48213FBAD0C

Device            Start          End      Sectors  Size Type
/dev/xvda1        4096          16777182 16773087   8G Linux filesystem
/dev/xvda128      2048           4095      2048    1M BIOS boot

Partition table entries are not in disk order.

Disk /dev/xvdf: 1 GiB, 1073741824 bytes, 2097152 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
[root@ip-172-31-46-167 ec2-user]# mkdir storage
[root@ip-172-31-46-167 ec2-user]# df -hT
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  474M   0    474M   0% /dev
tmpfs           tmpfs     483M   0    483M   0% /dev/shm
tmpfs          tmpfs     483M  412K  483M   1% /run
tmpfs          tmpfs     483M   0    483M   0% /sys/fs/cgroup
/dev/xvda1      xfs       8.0G  1.6G  6.5G  20% /
tmpfs          tmpfs     97M   0    97M   0% /run/user/1000

```

```
[root@ip-172-31-46-167 ec2-user]# mkfs -t xfs /dev/xvdf
mkfs.xfs: /dev/xvdf appears to contain an existing filesystem (xfs).
mkfs.xfs: Use the -f option to force overwrite.
[root@ip-172-31-46-167 ec2-user]# mount -t xfs /dev/xvdf /home/ec2-user/storage/
[root@ip-172-31-46-167 ec2-user]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda        202:0    0   8G  0 disk
├─xvda1     202:1    0   8G  0 part /
└─xvdf      202:80   0   1G  0 disk /home/ec2-user/storage
[root@ip-172-31-46-167 ec2-user]# df -hT
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  474M   0  474M   0% /dev
tmpfs           tmpfs     483M   0  483M   0% /dev/shm
tmpfs           tmpfs     483M  412K  483M   1% /run
tmpfs           tmpfs     483M   0  483M   0% /sys/fs/cgroup
/dev/xvda1      xfs       8.0G  1.6G  6.5G  20% /
tmpfs           tmpfs     97M   0   97M   0% /run/user/1000
/dev/xvdf       xfs       1014M  34M  981M   4% /home/ec2-user/storage
[root@ip-172-31-46-167 ec2-user]# cd storage/
[root@ip-172-31-46-167 storage]# ls
1.txt 10.txt 2.txt 3.txt 4.txt 5.txt 6.txt 7.txt 8.txt 9.txt
[root@ip-172-31-46-167 storage]# █
```

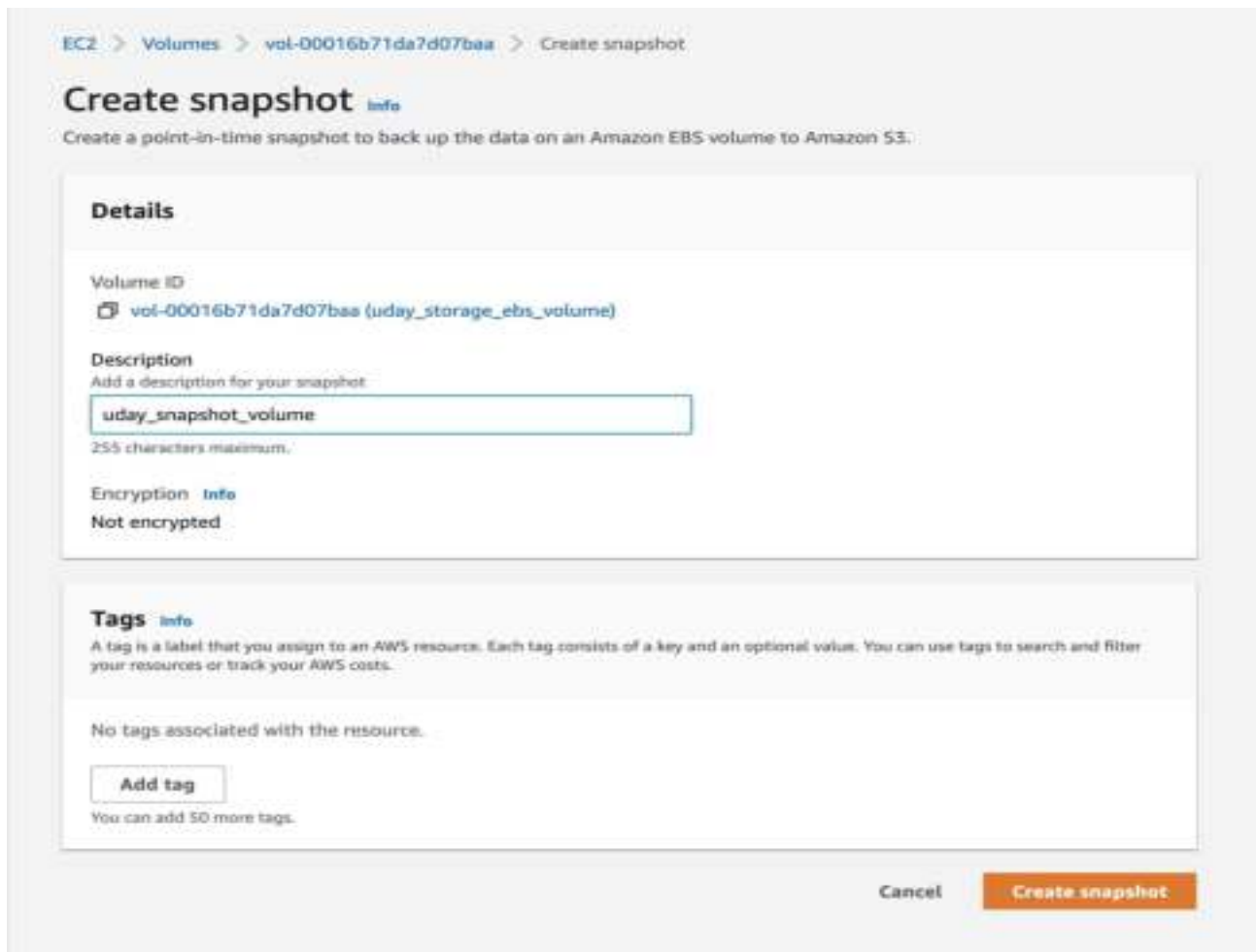
**EBS volumes are created
successfully and attached with
different EC2 instances without
loosing data**

Create a snapshot of the Volume or Instance:

Creates a prototype of the service which can be used to in other availability zone.

Snapshot can be copied to multiple regions. Deleted snapshots reside in recycle bin for the period defined in retention policy.

1. Create a snapshot of the existing EBS volume



2. Create a volume from the new snapshot in a different region (us-east-1)



3. Create a new EC2 instance in different region (us-east-1)



4. Attach the new volume with the new EC2 instances

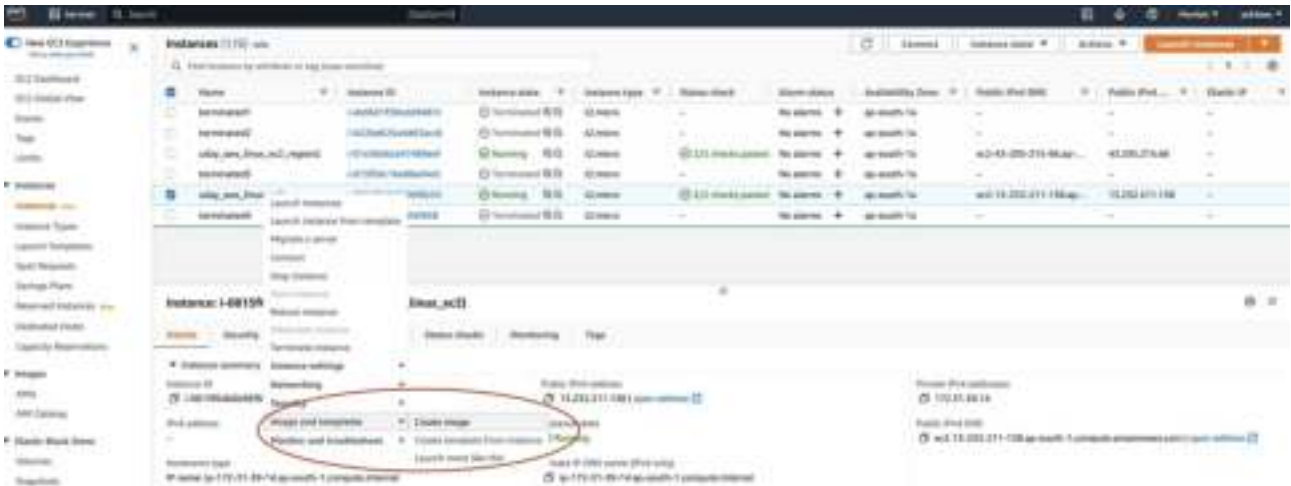


The Snapshot of the volume is created successfully, recreated the volume in different region and attached to the EC2 instance with data persisting

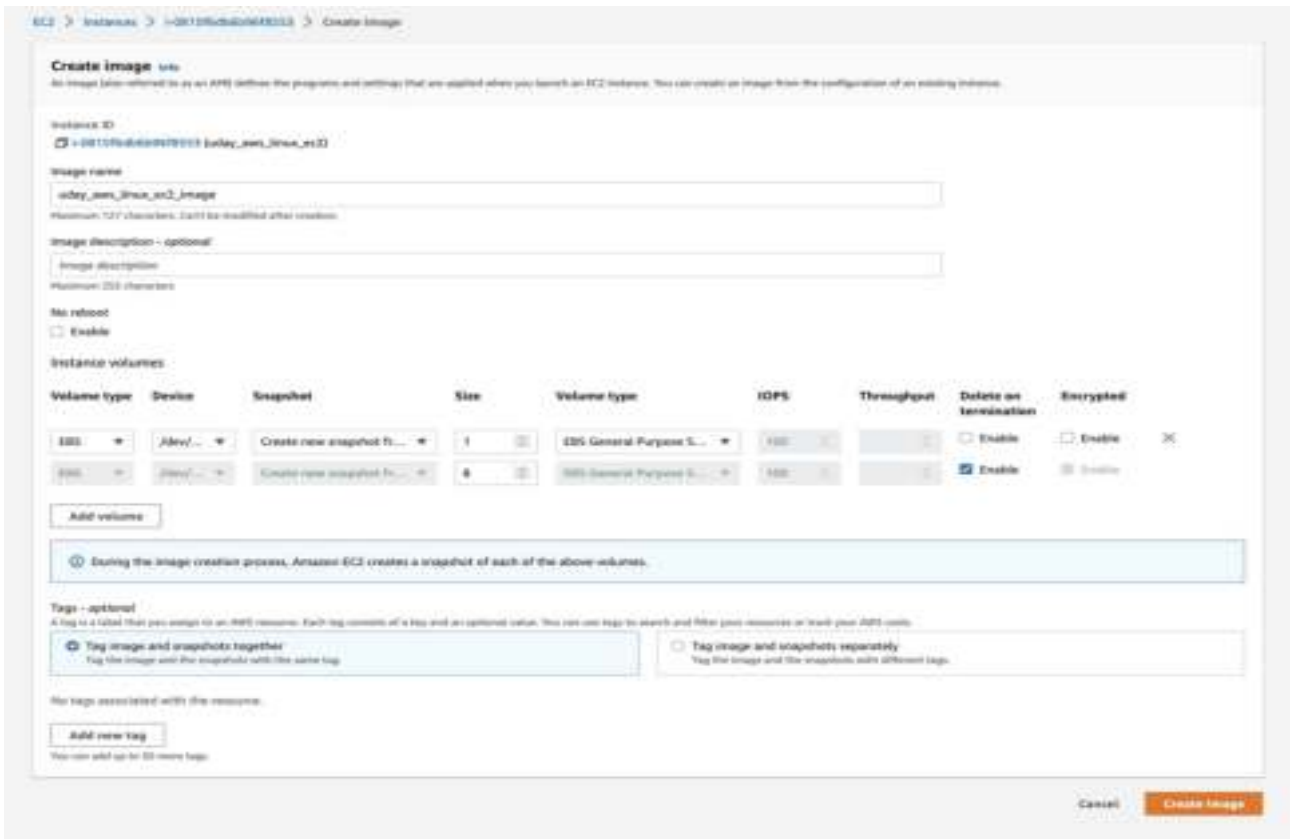
Create an AMI from an Amazon EC2 Instance

From the **Amazon EC2 Instances** view, you can create Amazon Machine Images (AMIs) from either running or stopped instances.

1. Right-click the instance you want to use as the basis for your AMI, and choose **Create Image** from the context menu.



2. In the **Create Image** dialog box, type a unique name and description, and then choose **Create Image**. By default, Amazon EC2 shuts down the instance, takes snapshots of any attached volumes, creates and registers the AMI, and then reboots the instance. Choose **No reboot** if you don't want your instance to be shut down.



3. Navigate:- EC2 -> Images -> AMIs.

You can Launch a new EC2 instance from the AMI



Successfully created an Image from existing EC2 instance.

Create Load Balancer:

1. Navigate to EC2 -> Load Balancing -> Load Balancers. And click create Load Balancers

Load balancer types

Application Load Balancer [info](#)

Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Create

Network Load Balancer [info](#)

Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Create

Gateway Load Balancer [info](#)

Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Create

▶ **Classic Load Balancer - previous generation**

Close

Create Application Load Balancer [info](#)

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

► How Elastic Load balancing works

Basic configuration

Load balancer name [info](#)

Name must be unique within your AWS account and cannot be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [info](#)

Scheme cannot be changed after the load balancer is created.

Internet-facing

An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#) 

Internal

An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type [info](#)

Select the type of IP addresses that your subnets use.

IPv4

Recommended for internal load balancers.

Dualstack

Includes IPv4 and IPv6 addresses.

Network mapping [info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [info](#)

Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

IPv4: 172.31.0.0/16

Mappings [info](#)

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

ap-south-1a

Subnet

IPv4 settings

Assigned by AWS

ap-south-1b

Subnet

IPv4 settings

Assigned by AWS

ap-south-1c

Security groups [info](#)

A security group is a set of firewall rules that control the traffic to your load balancer.

Security groups

Select up to 5 security groups

[Create new security group](#)

default sg-07de06485612f176b ✕
VPC: vpc-0c5166e8826f1a28

Listeners and routing [info](#)

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80

[Remove](#)

Protocol

HTTP

Port

80

1-65535

Default action [info](#)

Forward to: Select a target group

[Create target group](#)

Listener tags - optional

Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add listener tag](#)

You can add up to 10 more tags.

[Add listener](#)

▼ Add-on services - optional

Additional AWS services can be integrated with this load balancer at launch. You can also add these and other services after your load balancer is created by reviewing the "Integrated Services" tab for the selected load balancer.

AWS Global Accelerator [View](#)

- Create an accelerator to get static IP addresses and improve the performance and availability of your applications. Additional charges apply [View](#)

► Tags - optional

Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them. The "Key" is required, but "Value" is optional. For example, you can have Key = production-webserver, or Key = webserver, and Value = production.

Summary

Review and confirm your configurations. [Estimate cost](#) [View](#)

Basic configuration [View](#)

- us1ay_aws_app_loadbalancer
- Internet-facing
 - IPv4

Security groups [View](#)

- default
sg-071de06481612f176b [View](#)

Network mapping [View](#)

- VPC vpc-0x35dafe8826f1d28 [View](#)
- ap-south-1a
subnet-021d86157943e6d4 [View](#)
 - ap-south-1b
subnet-05094c333d09bc71f [View](#)

Listeners and routing [View](#)

- HTTP80 defaults to
Target group not defined

Add-on services [View](#)

None

Tags [View](#)

None

Attributes

[View](#) Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.

[Cancel](#)

[Create load balancer](#)

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section cannot be changed after the target group is created.

Choose a target type

Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice-based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv6 to IPv4 NAT.

Lambda function

- Facilitates routing to a single Lambda function.
- Available to Application Load Balancers only.

Application Load Balancer

- Offers the flexibility for a Network Load Balancer to assign and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol

Port

VPC

Select the VPC with the instances that you want to include in the target group.

Protocol version

HTTP1

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTPS.

HTTP2

Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

gRPC

Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

Health check path

Use the default path of "/" to ping the root, or specify a custom path if preferred.

Up to 1024 characters allowed.

▶ Advanced health check settings

Attributes



Application Load Balancer is created and registered available EC2 instances successfully