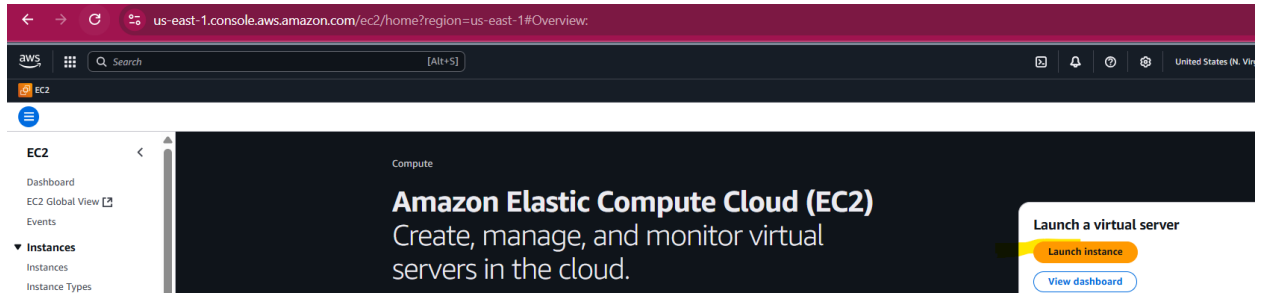
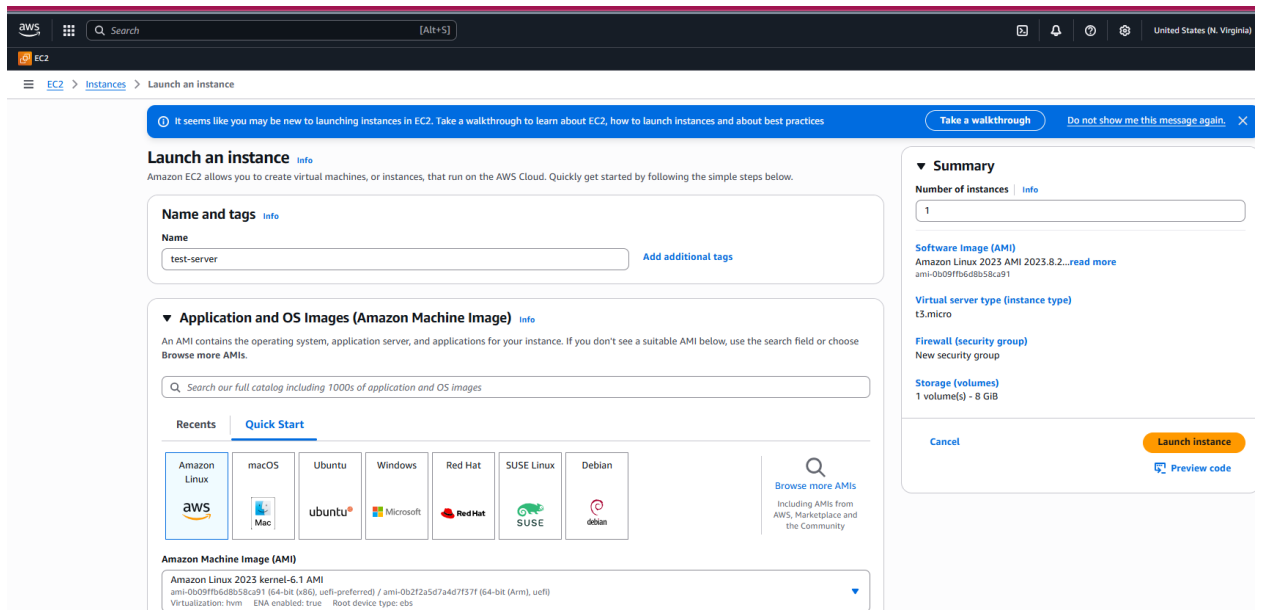


1. Launch one EC2 using Amazon Linux 2 image and add a script in user data to install Apache.

Login to aws account and goto ec2 and click on Launch instance:

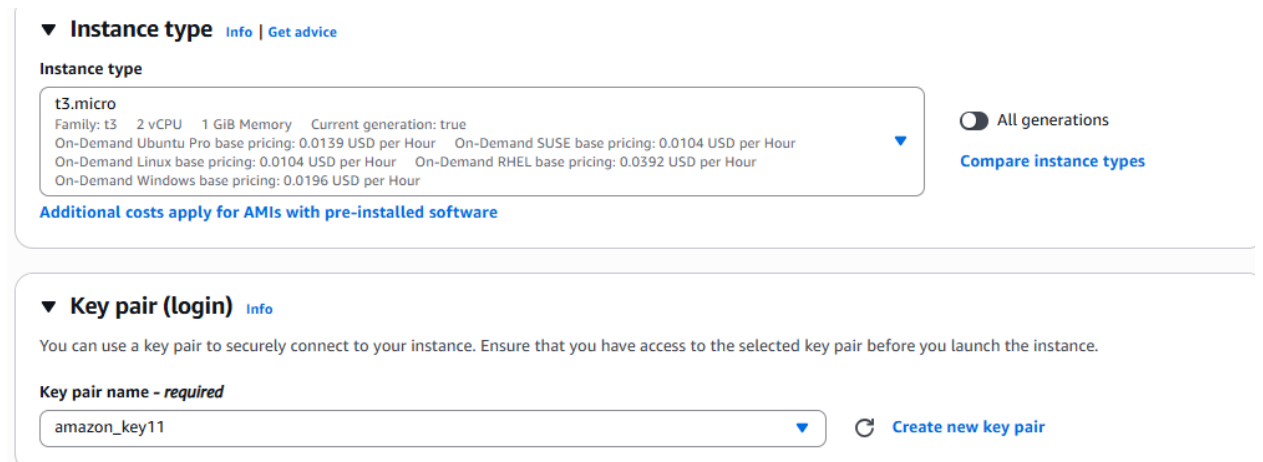


And provide the Name tag and select AMI 'Amazon Linux 2023':



Select the instance type as t3.micro and create a 'key pair'.

Note: I have previously created key pair so I just selected existing key pair:



Create the Security group,

Note: I have used the default Security group:

The screenshot shows the AWS Management Console for configuring an EC2 instance. The 'Network settings' section is expanded, showing the 'Firewall (security groups)' tab. The 'Create security group' button is selected. The 'Common security groups' dropdown shows 'default sg-0e64ae6a346f0ad43'. The 'Configure storage' section is also visible, showing '1x 8 GiB gp3' storage configuration.

Network settings [Info](#) [Edit](#)

Network [Info](#)
vpc-05f40aba9acea42ad

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

Common security groups [Info](#)
Select security groups

default sg-0e64ae6a346f0ad43 [X](#)
VPC: vpc-05f40aba9acea42ad

[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

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

1x 8 GiB gp3 Root volume, 3000 IOPS, Not encrypted

Number of instances [Info](#)
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.8.2...[read more](#)
ami-0b09ffb6d8b58ca91

Virtual server type (instance type)
t3.micro

Firewall (security group)
default

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

[Cancel](#) [Launch instance](#) [Preview code](#)

And click on Launch Template:

The screenshot shows the 'Advanced details' section of the AWS Management Console. The 'User data' field is visible, with a 'Choose file' button and a text area for entering the script.

Advanced details [Info](#)

and write the script in the user_data:

The screenshot shows the 'User data' section of the AWS Management Console. The 'User data - optional' section is expanded, showing the 'Choose file' button and a text area for entering the script. The script content is as follows:

```
#!/bin/bash
sudo dnf install httpd.x86_64 -y
sudo systemctl start httpd.service
```

☐ User data has already been base64 encoded

And Launch the instance by clicking on 'Launch instance' in right hand side.

Then login to the server using ssh and pem key and then check the apache is running or not status:

```
[ec2-user@ip-172-31-21-198 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: d
   Active: active (running) since Wed 2025-09-17 07:30:08 UTC; 5min ago
     Docs: man:httpd.service(8)
  Main PID: 3202 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes
      Tasks: 177 (limit: 1057)
     Memory: 13.3M
        CPU: 416ms
    CGroup: /system.slice/httpd.service
            └─3202 /usr/sbin/httpd -DFOREGROUND
              └─3213 /usr/sbin/httpd -DFOREGROUND
                └─3215 /usr/sbin/httpd -DFOREGROUND
                  └─3216 /usr/sbin/httpd -DFOREGROUND
                    └─3313 /usr/sbin/httpd -DFOREGROUND
```

Check the script which has been written in the user_data while launching the instance:

```
[ec2-user@ip-172-31-21-198 ~]$ sudo cat /var/lib/cloud/instance/user-data.txt
#!/bin/bash
sudo dnf install httpd.x86_64 -y
sudo systemctl start httpd.service
```

2. Launch one EC2 using Ubuntu image and add a script in user data to install Nginx.

Launch the ec2 instance and give the name and select AMI as ubuntu:

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

nginx-server

[Add additional tags](#)

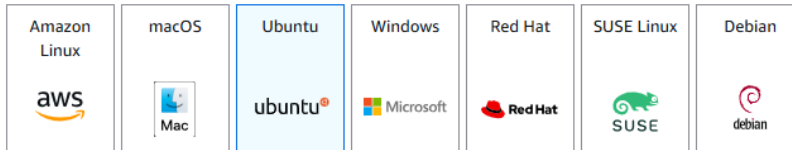
▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

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

Recents

Quick Start



[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-0360c520857e3138f (64-bit (x86)) / ami-026fccd88446aa0bf (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Select the instance type as 't3.micro' and create a key pair by giving the name:

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t3.micro

Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour On-Demand SUSE base pricing: 0.0104 USD per Hour
On-Demand Linux base pricing: 0.0104 USD per Hour On-Demand RHEL base pricing: 0.0392 USD per Hour
On-Demand Windows base pricing: 0.0196 USD per Hour

☐ All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

▼ 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*

ubuntu_key11



[Create new key pair](#)

Select the Default security group:

▼ Network settings [Info](#)

[Edit](#)

Network [Info](#)

vpc-05f40aba9acea42ad

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

Common security groups [Info](#)

Select security groups

default sg-0e64ae6a346f0ad43 ✕
VPC: vpc-05f40aba9acea42ad

🔄 [Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Expand the Advanced details:

► Advanced details [Info](#)

Write the script in user_data:

User data - optional [Info](#)

Upload a file with your user data or enter it in the field.

📁 Choose file

```
#!/bin/bash
sudo apt update -y
sudo apt install nginx -y
sudo systemctl enable nginx
sudo systemctl start nginx
```

☐ User data has already been base64 encoded

And launch the instance:

Metadata response hop limit
[info](#)

Allow tags in metadata
[info](#)

Select

User data - optional
[info](#)

Upload a file with your user data or enter it in the field.

Choose file

```
#!/bin/bash
sudo apt update -y
sudo apt install nginx -y
sudo systemctl enable nginx
sudo systemctl start nginx
```

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd64...[read more](#)

ami-0360c520857e3138f

Virtual server type (instance type)

t3.micro

Firewall (security group)

default

Storage (volumes)

1 volume(s) - 8 GiB

Cancel

Launch instance

[Preview code](#)

Login to ubuntu server using pem key and check nginx is running or not:

```
ubuntu@ip-172-31-17-175:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: en
   Active: active (running) since wed 2025-09-17 07:48:47 UTC; 5min ago
     Docs: man:nginx(8)
    Main PID: 1600 (nginx)
      Tasks: 3 (limit: 1008)
     Memory: 2.5M (peak: 5.3M)
```

To check the script which is written in the user_data while launching the instance:

```
ubuntu@ip-172-31-17-175:~$ sudo cat /var/lib/cloud/instance/user-data.txt
#!/bin/bash
sudo apt update -y
sudo apt install nginx -y
sudo systemctl enable nginx
sudo systemctl start nginx
```

3. Launch one Windows server and install Tomcat on Windows.

Launch the ec2 instance with Windows 2025 base Server:

Name and tags [Info](#)

Name

Windows-Server

[Add additional tags](#)

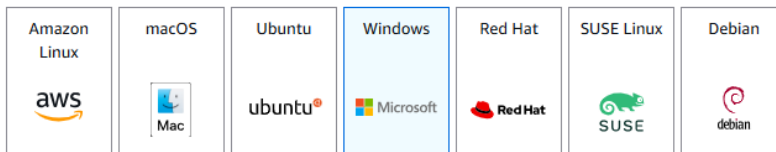
▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

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

Recents

[Quick Start](#)



[Browse more AMIs](#)

Including AMIs from
AWS, Marketplace and
the Community

Amazon Machine Image (AMI)

Microsoft Windows Server 2025 Base
ami-0e3c2921641a4a215 (64-bit (x86))
Virtualization: hvm ENA enabled: true Root device type: ebs

Provide the instance type and create a key pair:

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t3.micro

Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour On-Demand SUSE base pricing: 0.0104 USD per Hour
On-Demand Linux base pricing: 0.0104 USD per Hour On-Demand RHEL base pricing: 0.0392 USD per Hour
On-Demand Windows base pricing: 0.0196 USD per Hour

☐ All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

▼ 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*

windows_key



[Create new key pair](#)

For Windows instances, you use a key pair to decrypt the administrator password. You then use the decrypted password to connect to your instance.

And storage give 30gb, gp2

▼ Network settings [Info](#)

Network [Info](#)

vpc-05f40aba9acea42ad

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

Common security groups [Info](#)

Select security groups

default sg-0e64ae6a346f0ad43 [X](#)

VPC: vpc-05f40aba9acea42ad

[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

▼ Configure storage [Info](#)

Advanced

1x GiB

Root volume, Not encrypted

Create security group and open ports: 3389 for rdp, 80 for http, 8080 for tomcat

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

Security group name - *required*

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and `./:/()#,@!+=&:{}|_$*`

Description - *required* [Info](#)

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 3389, 0.0.0.0/0)

[Remove](#)

Type [Info](#)

Protocol [Info](#)

Port range [Info](#)

Source type [Info](#)

Source [Info](#)

[X](#)

Description - *optional* [Info](#)

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0)

[Remove](#)

Type [Info](#)

Protocol [Info](#)

Port range [Info](#)

Source type [Info](#)

Source [Info](#)

Description - *optional* [Info](#)

Type | [Info](#)
 Custom TCP ▼

Protocol | [Info](#)
 TCP

Port range | [Info](#)
 8080

Source type | [Info](#)
 Custom ▼

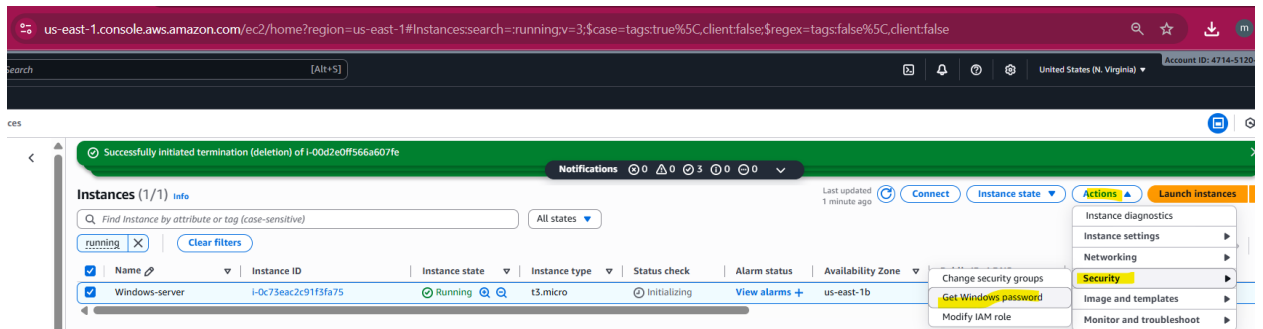
Source | [Info](#)
 🔍 Add CIDR, prefix list or security group

Description - optional | [Info](#)
 e.g. SSH for admin desktop

⚠️ 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.

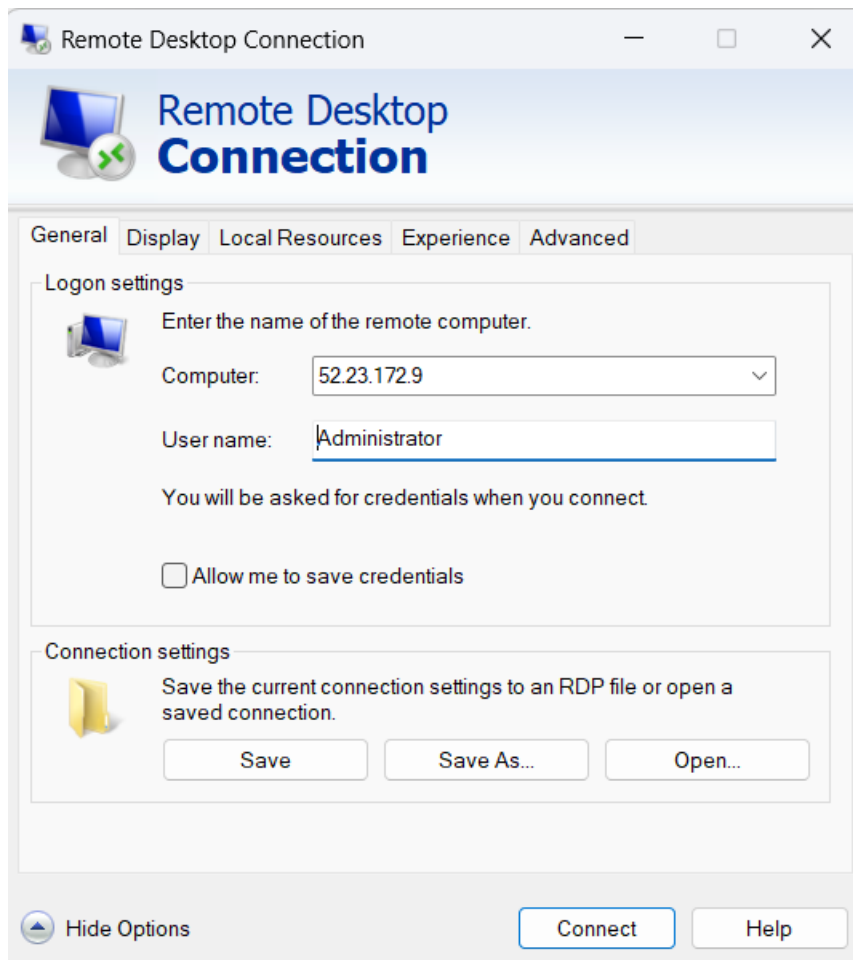
[Add security group rule](#)

And get Windows password by clicking on ->Actions->Security->Get Windows Password

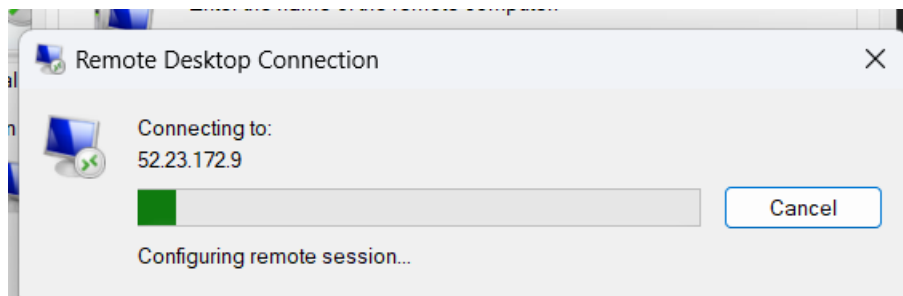


And then decrypt the password by providing the generated .pem key.

And to connect to windows : mstsc->ip, username: Administrator, password: copied password:



And click on connect



4. Take a snapshot of the instance created in Task 1.

In the Task1 Launched the ec2 instance with apache installed on it using user_data .

Click on snapshot and select the instance from which want to create snapshot:

Create snapshot Info

Create a point-in-time snapshot of an EBS volume and use it as a baseline for new volumes or for data backup. You can create snapshots from an individual volume, or you can create multi-volume snapshots from all of the volumes attached to an instance.

Source

Resource type Info

☐ Volume
Create a snapshot from a specific volume.

☒ Instance
Create multi-volume snapshots from an instance.

Instance ID
The instance from which to create multi-volume snapshots.

i-044674baebd05c561 (test)

Snapshot details

Description
Add a description for your snapshot.

apache-tomcat-snapshot

255 characters maximum

Click on create snapshot:

Tags Info

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key

Q name

Value - optional

Q apache-snapshot

Remove

Use "apache-snapshot"

Add tag

You can add 49 more tags.

Cancel Create snapshot

Snapshots (2) Info

Owned by me

Search

	Name	Snapshot ID	Full snapshot size	Volume size	Description	Storage tier	Snapshot status	Started
<input type="checkbox"/>		snap-05e331185f7b34	1.71 GiB	8 GiB	apache-tomcat-snapshot	Standard	Completed	2025/09/17 18:3

Create an Image from instance:

Successfully initiated termination (deletion) of i-0587339C7d6dcf78

Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

All states

running

Clear filters

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input checked="" type="checkbox"/>	test	i-044674baebd05c561	Running	t3.micro	3/3 checks passed	View alarms	us-east-1b	

Create image

Create template from instance

Launch more like this

Instance diagnostics

Instance settings

Networking

Security

Image and templates

Monitor and troubleshoot

Provide the image name and description:

Create image [Info](#)

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Image details

Instance ID

i-044674baebd05c561 (test)

Image name

apache-image

Maximum 127 characters. Can't be modified after creation.

Image description - optional

apache-image

Maximum 255 characters

☒ Reboot instance

When selected, Amazon EC2 reboots the instance so that data is at rest when snapshots of the attached volumes are taken. This ensures data consistency.

Instance volumes

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/xv...	Create new snapshot from v...	8	EBS General Purpose SSD - ...	3000		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

[Add volume](#)

Click on create image:

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/xv...	Create new snapshot from v...	8	EBS General Purpose SSD - ...	3000		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

[Add volume](#)

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

☒ Tag image and snapshots together

Tag the image and the snapshots with the same tag.

☐ Tag image and snapshots separately

Tag the image and the snapshots with different tags.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags.

[Cancel](#)

[Create image](#)

5. Assign passwordless authentication for the EC2 created in Task 2.

In task2 we have launched an ec2 instance with ubuntu and install nginx in user_data.

Generate ssh-keygen:

```
dell@DESKTOP-6ORBKUF MINGW64 ~/Downloads
$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/dell/.ssh/id_ed25519):
/c/Users/dell/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase for "/c/Users/dell/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/dell/.ssh/id_ed25519
Your public key has been saved in /c/Users/dell/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:w5l/G3r1CJ+y32tlQ7Yt6npjP3p0UC+XuS1cSbDoc21E dell@DESKTOP-6ORBKUF
The key's randomart image is:
```

And copy the public key:

```
de11@DESKTOP-6ORBKUF MINGW64 ~/Downloads
$ cat /c/Users/de11/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAII3H17TSMIosON5kZL6sFvXFxcLKjlUG2Vp5JjTC7sii
de11@DESKTOP-6ORBKUF
```

Into the ~/.ssh/authorized_keys:

```
ec2-user@ip-172-31-46-215 ~]$ cat ~/.ssh/authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAII3H17TSMIosON5kZL6sFvXFxcLKjlUG2Vp5JjTC7sii
de11@DESKTOP-6ORBKUF
```

Change the ownership means it should be ec2-user not the root:

```
[root@ip-172-31-46-215 ~]# chown ec2-user:ec2-user ~/.ssh
[root@ip-172-31-46-215 ~]# chown ec2-user:ec2-user ~/.ssh/authorized_keys
```

```
[ec2-user@ip-172-31-46-215 ~]$ ls -ld ~/.ssh
drwx-----. 2 ec2-user ec2-user 61 Sep 17 16:58 /home/ec2-user/.ssh
[ec2-user@ip-172-31-46-215 ~]$ ls -l ~/.ssh/authorized_keys
-rw-----. 1 ec2-user ec2-user 102 Sep 17 16:58 /home/ec2-user/.ssh/authorized_
keys
```

And also change the permission:

```
drwx-----. 2 root root 29 Sep 17 16:37 .ssh
-rw-r--r--. 1 root root 129 Feb 2 2023 .tcshrc
-rw-----. 1 root root 2172 Sep 17 16:37 .viminfo
[root@ip-172-31-46-215 ~]# chmod 700 ~/.ssh
[root@ip-172-31-46-215 ~]# ls -lA
total 28
-rw-----. 1 root root 37 Sep 17 16:37 .bash_history
-rw-r--r--. 1 root root 18 Feb 2 2023 .bash_logout
-rw-r--r--. 1 root root 141 Feb 2 2023 .bash_profile
-rw-r--r--. 1 root root 429 Feb 2 2023 .bashrc
-rw-r--r--. 1 root root 100 Feb 2 2023 .cshrc
drwx-----. 2 root root 29 Sep 17 16:37 .ssh
-rw-r--r--. 1 root root 129 Feb 2 2023 .tcshrc

[root@ip-172-31-46-215 ~]# chmod 600 ~/.ssh/authorized_keys
```

```
dell@DESKTOP-60RBKUF MINGW64 ~/Downloads
$ ssh ec2-user@13.60.105.19

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

Last login: wed Sep 17 16:53:44 2025 from 103.143.169.218

6. Launch any EC2 using the spot purchasing option.

During ec2 instance Launch click on Advanced details:

▼ **Advanced details** [Info](#)

In purchasing options select 'Spot instances':

Purchasing option | Info

☐ None

- Capacity Blocks

Launch instances for your active capacity blocks

- Spot instances

Request Spot Instances at the Spot price, capped at the On-Demand price

Customize Spot instance options

Click on customize spot instance options, and select Request type as Persistent, and Interruption behavior as Stop:

Spot Instance Options | [Info](#)
Specify Spot Instance Options such as Maximum Price, Request type, expiration date and interruption behavior

Maximum price | [Info](#)

☒ No maximum price
Request Spot Instances at the Spot price, capped at the On-Demand price

☐ Set your maximum price (per instance/hour)

Request type | [Info](#)

Persistent ▼

Valid to | [Info](#)

☒ No request expiry date
The default value is no expiry date

☐ Set your request expiry date

Interruption behavior | [Info](#)

Stop ▼

Click on Launch instance:

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

Cancel

Launch instance

[Preview code](#)

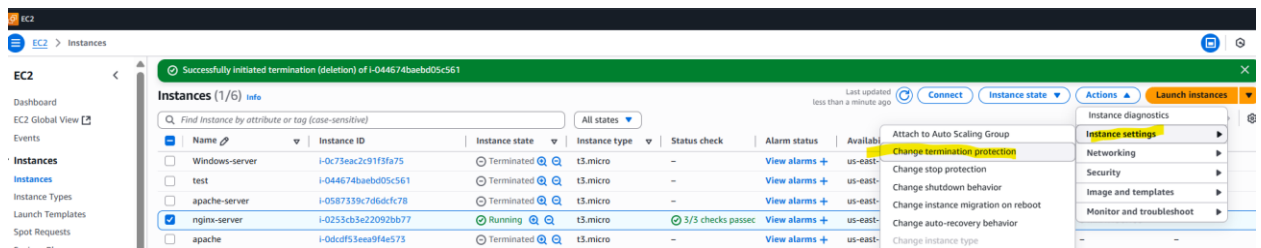
the created spot instance we can see in spot Request:

<div>EC2</div> <div>Dashboard</div> <div>EC2 Global View</div> <div>Events</div> <div>▼ Instances</div> <div>Instances</div> <div>Instance Types</div> <div>Launch Templates</div> <div>Spot Requests</div>	Spot Requests (1)									
	Q Search for requests									
	<input type="checkbox"/>	Request ID	Request type	Instance t...	State	Capacity	Status	Persistence	Created	
	<input type="checkbox"/>	si-4jq643p	instance	t3.micro	active	i-0956c823a33f82c63	fulfilled	persistent	a few seconds ago	

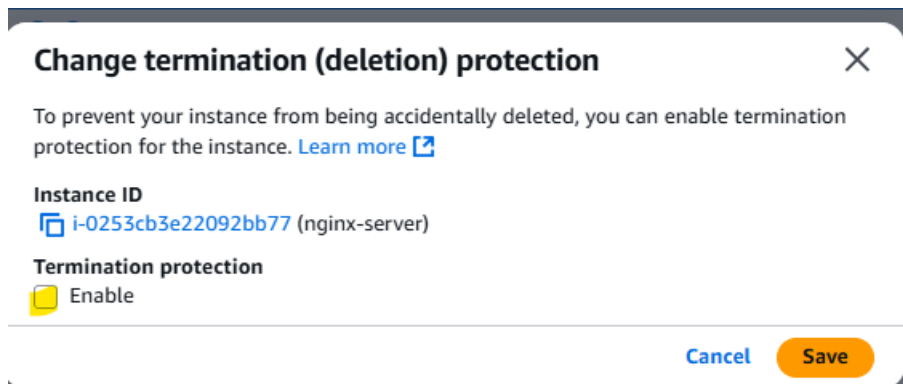
7. Enable termination policy on the EC2 created in Task 2.

In task2 we have launched an ec2 instance with ubuntu and install nginx in user_data.

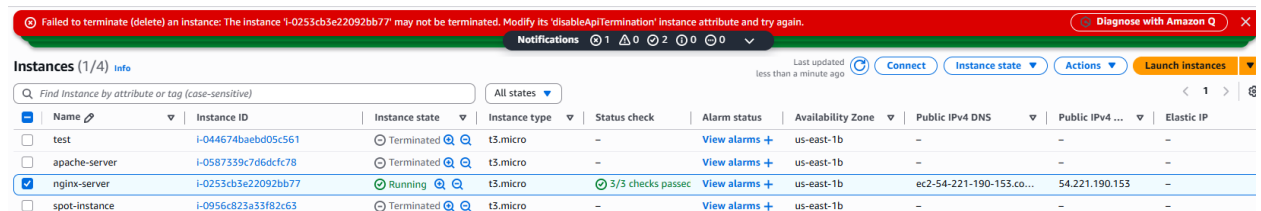
Select the instance and click on Actions->instance Settings->change termination protection:



Enable the Termination protection:



Then try to delete the instance, it won't allow us to delete because of this Termination protection policy:



8. Launch one EC2 using AWS CLI.

Download aws cli:

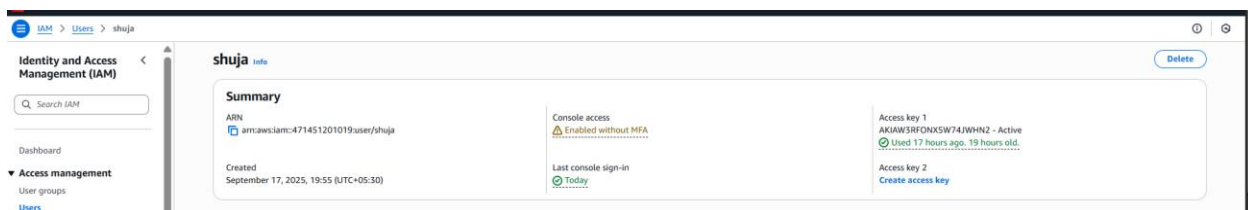
For various parameters that can be used with `msiexec`, see [msiexec](#) on the *Microsoft Docs* website. For example, you can use the `/qn` flag for a silent installation.

```
C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi /qn
```

To confirm the installation, open the **Start** menu, search for `cmd` to open a command prompt window, and at the command prompt use the `aws --version` command.

```
C:\> aws --version
aws-cli/2.27.41 Python/3.11.6 Windows/10 exe/AMD64 prompt/off
```

Get the access key and secret key by generating from IAM user:



Provide **Access Key, Secret Key, Region, Output format**.

```
de11@DESKTOP-6ORBKUF MINGW64 ~
$ aws configure
AWS Access Key ID [*****WHN2]:
AWS Secret Access Key [*****]xte]:
Default region name [us-east-1]:
Default output format [None]:
```

Create the pem key using aws cli:

```
de11@DESKTOP-6ORBKUF MINGW64 ~
$ aws ec2 create-key-pair --key-name my-key --query 'KeyMaterial' --output text
> my-key.pem
chmod 400 my-key.pem
```

Create an ec2 instance using aws cli:

```
de11@DESKTOP-6ORBKUF MINGW64 ~
$ aws ec2 run-instances --image-id ami-08982f1c5bf93d976 --count 1 --insta
nce-type t3.micro --key-name my-key --security-group-ids sg-0e64ae6a346f0ad4
3 --subnet-id subnet-0242f46eea951417e --tag-specifications 'ResourceType=in
stance,Tags=[{Key=Name,Value=MyFirstEC2}]'
{
```

```
{
  "ReservationId": "r-08b16b2cc8299abd6",
  "OwnerId": "471451201019",
  "Groups": [],
  "Instances": [
    {
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientToken": "4077c1e3-26d2-4c17-9e68-2897357d8d3b",
      "EbsOptimized": false,
      "EnaSupport": true,
      "Hypervisor": "xen",
      "NetworkInterfaces": [
```

We can see the created instance from the aws cli:

```
de11@DESKTOP-60RBKUF MINGW64 ~
$ aws ec2 describe-instances
{
  "Reservations": [
    {
      "ReservationId": "r-08b16b2cc8299abd6",
      "OwnerId": "471451201019",
      "Groups": [],
      "Instances": [
        {
          "Architecture": "x86_64",
          "BlockDeviceMappings": [
            {
              "DeviceName": "/dev/xvda",
              "Ebs": {
                "AttachTime": "2025-09-18T09:51:01+00:00",
                "DeleteOnTermination": true,
```

-----Completed-----