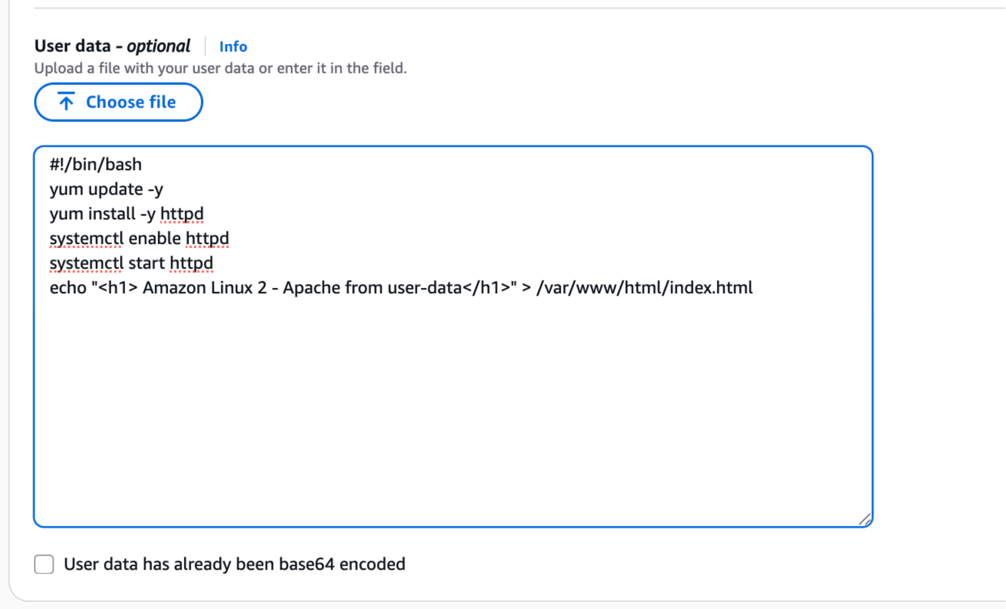
EC2 Task

— — — — — — — —

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

>> Here, we are creating a instance, before creating it we are adding the script to install httpd along with launching the server.



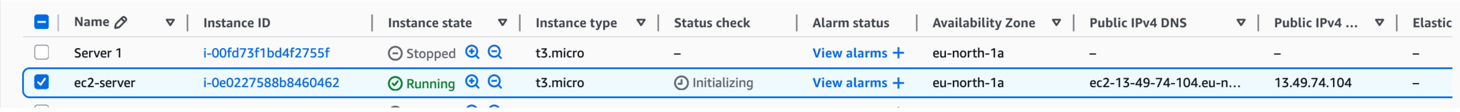
User data - *optional* | Info

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

```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl enable httpd
systemctl start httpd
echo "<h1> Amazon Linux 2 - Apache from user-data</h1>" > /var/www/html/index.html
```

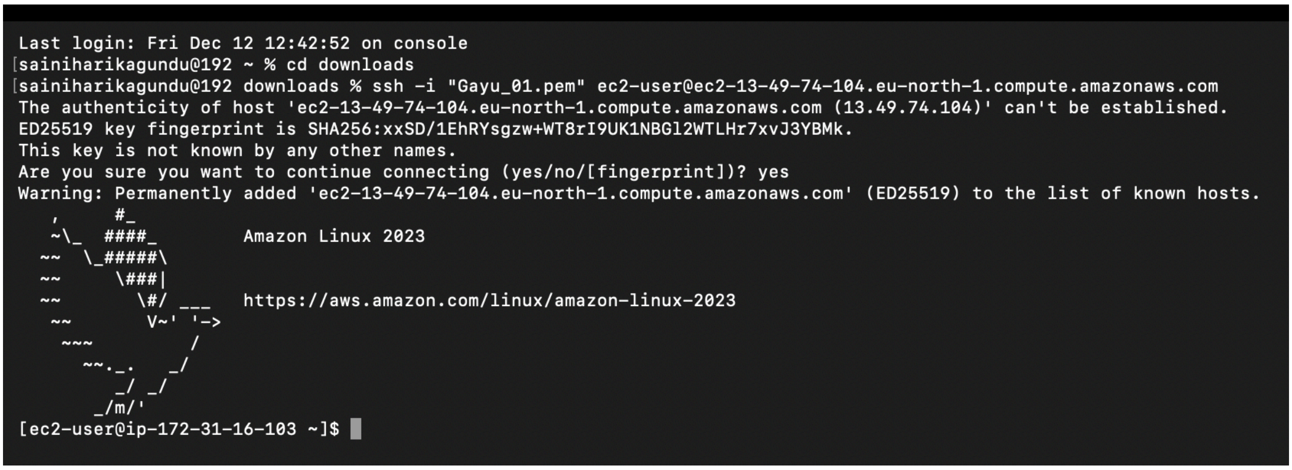
□ User data has already been base64 encoded

>> launched the server



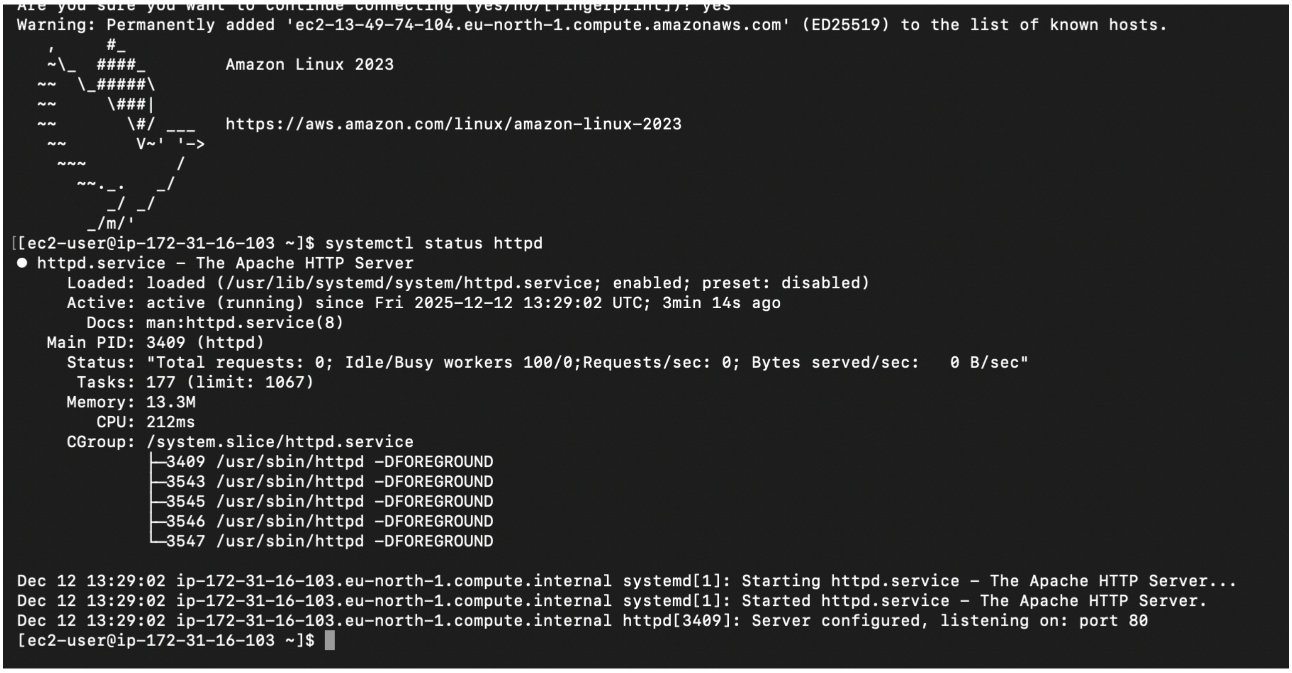
▣	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic
□	Server 1	i-00fd73f1bd4f2755f	Stopped	t3.micro	–	View alarms +	eu-north-1a	–	–	–
☑	ec2-server	i-0e0227588b8460462	Running	t3.micro	Initializing	View alarms +	eu-north-1a	ec2-13-49-74-104.eu-n...	13.49.74.104	–

>> now open the server in your local terminal



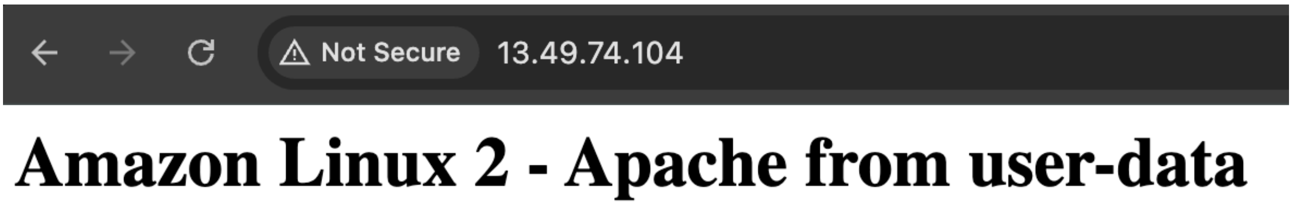
```
Last login: Fri Dec 12 12:42:52 on console
[sainiharikagundu@192 ~ % cd downloads
[sainiharikagundu@192 downloads % ssh -i "Gayu_01.pem" ec2-user@ec2-13-49-74-104.eu-north-1.compute.amazonaws.com
The authenticity of host 'ec2-13-49-74-104.eu-north-1.compute.amazonaws.com (13.49.74.104)' can't be established.
ED25519 key fingerprint is SHA256:xxSD/1EhRYsgzw+WT8rI9UK1NBGl2WTLHr7xvJ3YBMk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-49-74-104.eu-north-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-16-103 ~]$ 
```

>> here, we can see the httpd server is running



```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-49-74-104.eu-north-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-16-103 ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Fri 2025-12-12 13:29:02 UTC; 3min 14s ago
       Docs: man:httpd.service(8)
   Main PID: 3409 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 1067)
     Memory: 13.3M
        CPU: 212ms
     CGroup: /system.slice/httpd.service
             ├─3409 /usr/sbin/httpd -DFOREGROUND
             ├─3543 /usr/sbin/httpd -DFOREGROUND
             ├─3545 /usr/sbin/httpd -DFOREGROUND
             ├─3546 /usr/sbin/httpd -DFOREGROUND
             └─3547 /usr/sbin/httpd -DFOREGROUND

Dec 12 13:29:02 ip-172-31-16-103.eu-north-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Dec 12 13:29:02 ip-172-31-16-103.eu-north-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Dec 12 13:29:02 ip-172-31-16-103.eu-north-1.compute.internal httpd[3409]: Server configured, listening on: port 80
[ec2-user@ip-172-31-16-103 ~]$ 
```

>> here, we can see httpd is active



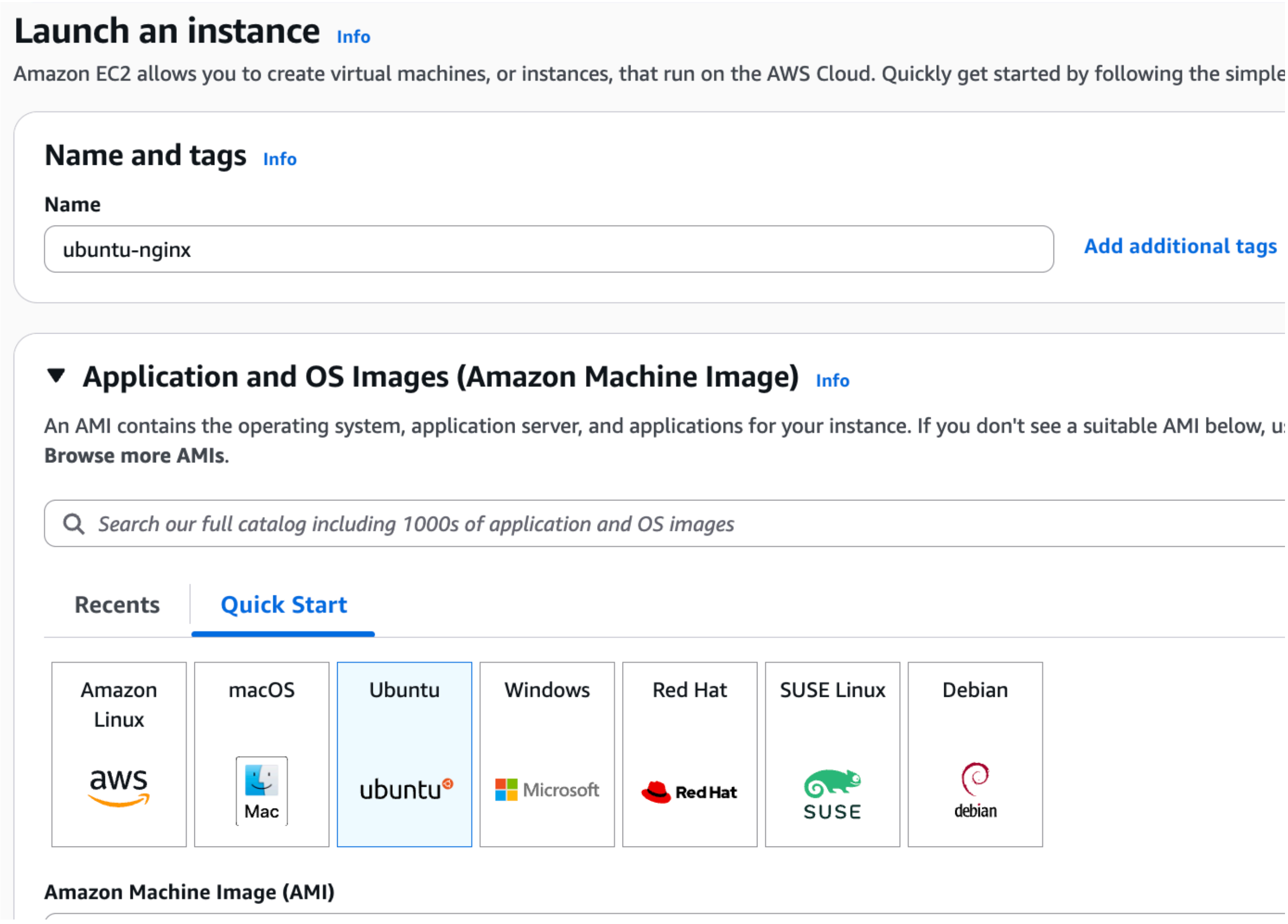
Amazon Linux 2 - Apache from user-data

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

>> Here, we are launching the instance by giving the name as “ubuntu-nginx” as we’re installing nginx (for my understanding)

>> and choosing Ubuntu as my AMI (Amazon Machine Image)

>> here, we are adding the bash script to install nginx along while launching the server



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

Name and tags Info

Name

ubuntu-nginx

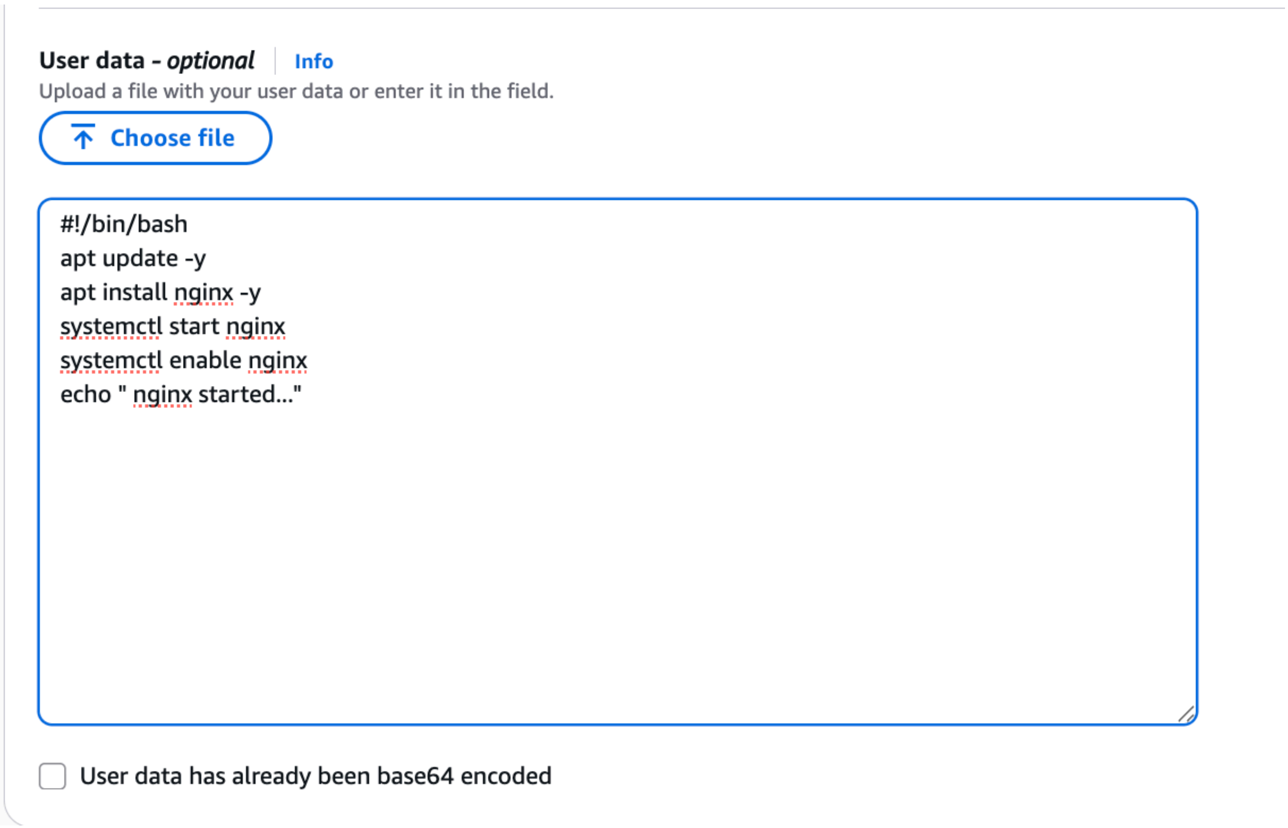
Add additional tags

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, u: **Browse more AMIs**.

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

Amazon Machine Image (AMI)



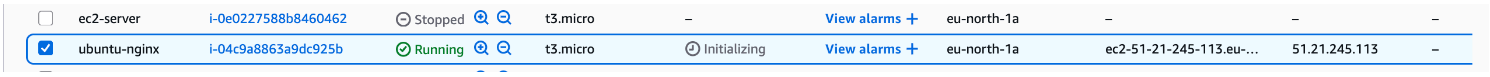
User data - *optional* Info

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

```
#!/bin/bash
apt update -y
apt install nginx -y
systemctl start nginx
systemctl enable nginx
echo " nginx started..."
```

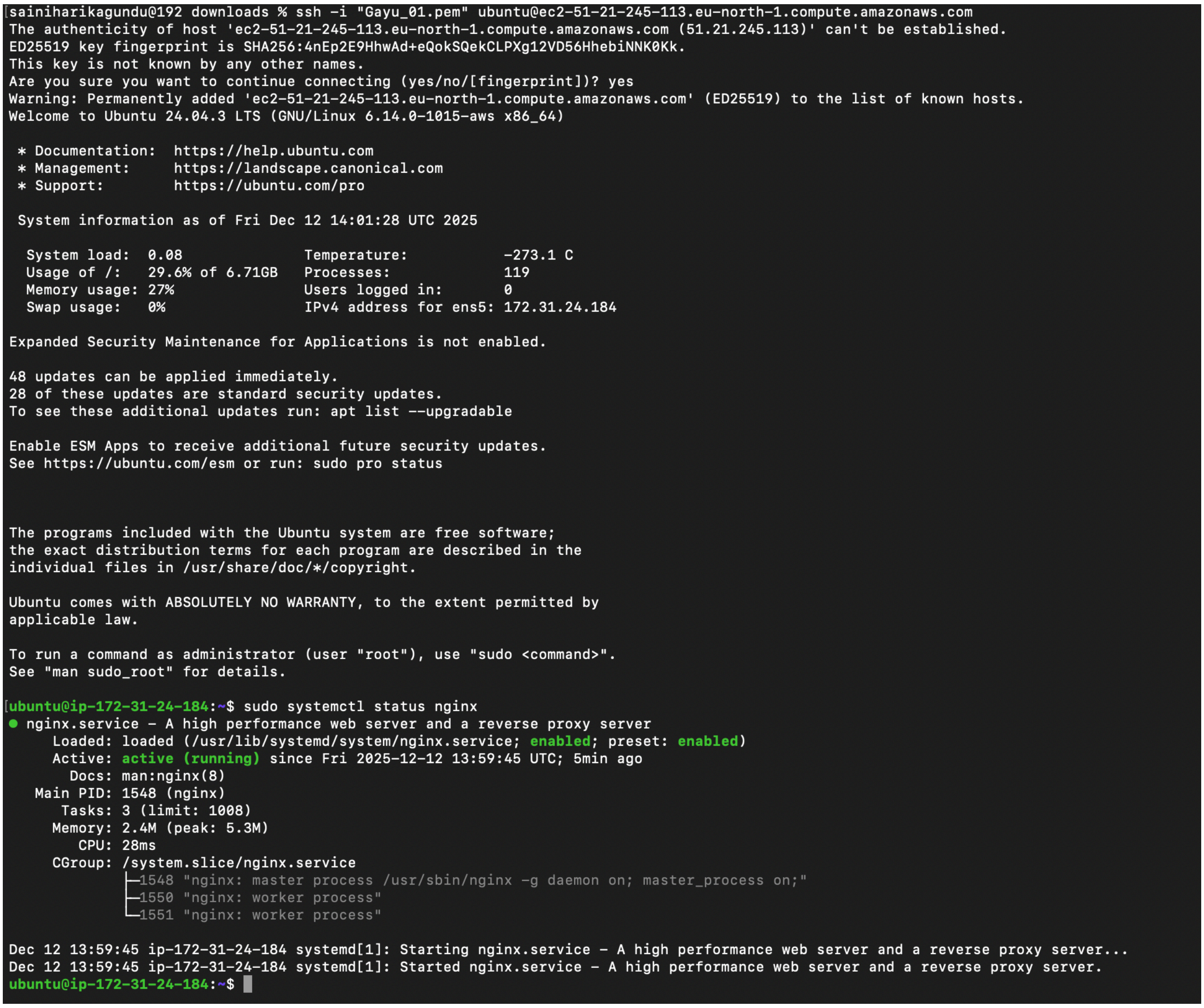
□ User data has already been base64 encoded

>> launching the instance

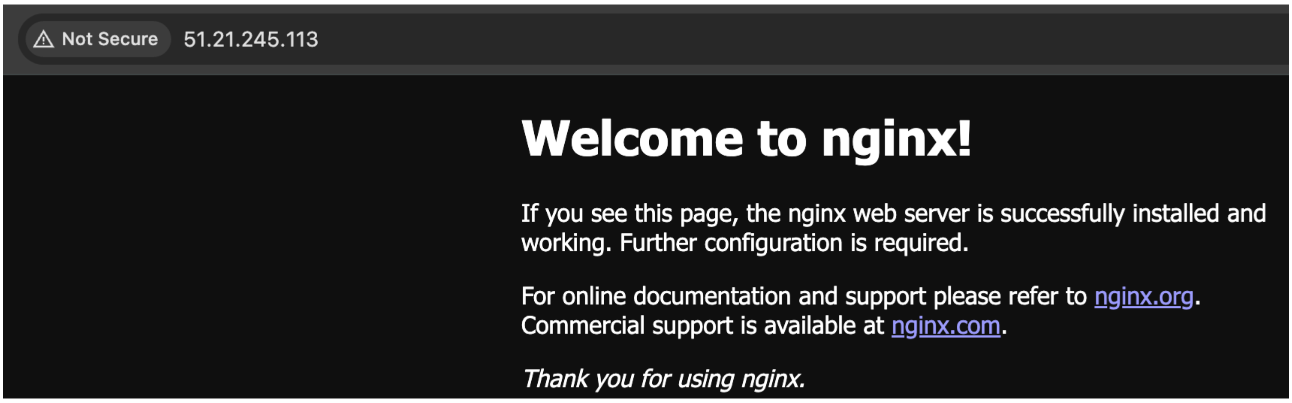


□	ec2-server	i-0e0227588b8460462	Stopped	t3.micro	–	View alarms +	eu-north-1a	–	–	–
☑	ubuntu-nginx	i-04c9a8863a9dc925b	Running	t3.micro	Initializing	View alarms +	eu-north-1a	ec2-51-21-245-113.eu-...	51.21.245.113	–

>> Here, we are adding the server to local and checking whether nginx is running successfully.
>> could confirm that nginx is running



```
sainiharikagundu@192 downloads % ssh -i "Gayu_01.pem" ubuntu@ec2-51-21-245-113.eu-north-1.compute.amazonaws.com
The authenticity of host 'ec2-51-21-245-113.eu-north-1.compute.amazonaws.com (51.21.245.113)' can't be established.
ED25519 key fingerprint is SHA256:4nEp2E9HhwAd+eQokSQekCLPXg12VD56HhebiNNK0Kk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-51-21-245-113.eu-north-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

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

 System information as of Fri Dec 12 14:01:28 UTC 2025

  System load:  0.08              Temperature:           -273.1 C
  Usage of /:   29.6% of 6.71GB   Processes:             119
  Memory usage: 27%               Users logged in:       0
  Swap usage:   0%                IPv4 address for ens5: 172.31.24.184

Expanded Security Maintenance for Applications is not enabled.

48 updates can be applied immediately.
28 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

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



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-172-31-24-184:~$ 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: enabled)
     Active: active (running) since Fri 2025-12-12 13:59:45 UTC; 5min ago
       Docs: man:nginx(8)
   Main PID: 1548 (nginx)
      Tasks: 3 (limit: 1008)
     Memory: 2.4M (peak: 5.3M)
        CPU: 28ms
     CGroup: /system.slice/nginx.service
             ├─1548 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─1550 "nginx: worker process"
             └─1551 "nginx: worker process"

Dec 12 13:59:45 ip-172-31-24-184 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Dec 12 13:59:45 ip-172-31-24-184 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
ubuntu@ip-172-31-24-184:~$ 
```



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

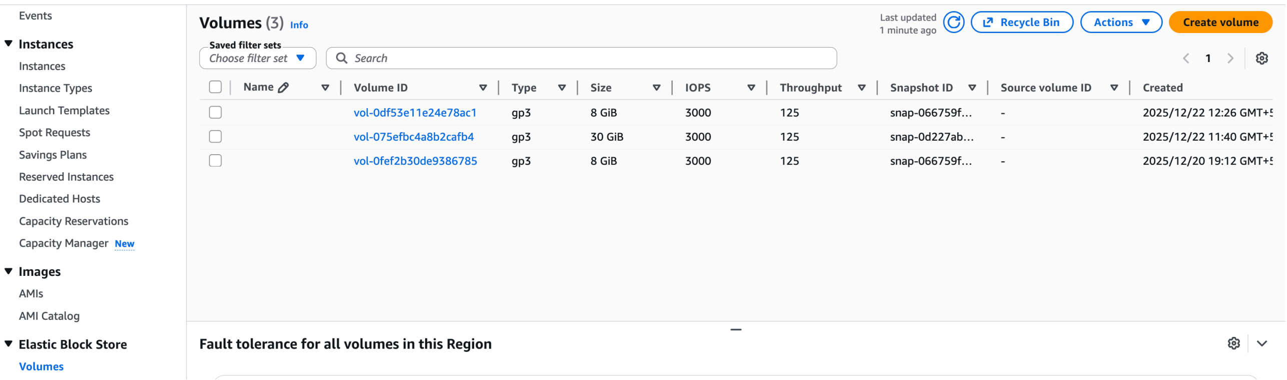
4. Take a snapshot of the instance created in Task 1.

A **Snapshot** is a **backup** of an EBS Volume
(where your EC2 instance’s storage / disk lives).
Snapshot = **Photo / Backup of your server’s disk**
It contains:

- Operating system (Amazon Linux)
- Apache installation
- Your user-data installed files
- Any files you had on the server

So if your instance is deleted or corrupted, you can **restore** a new instance using the snapshot.

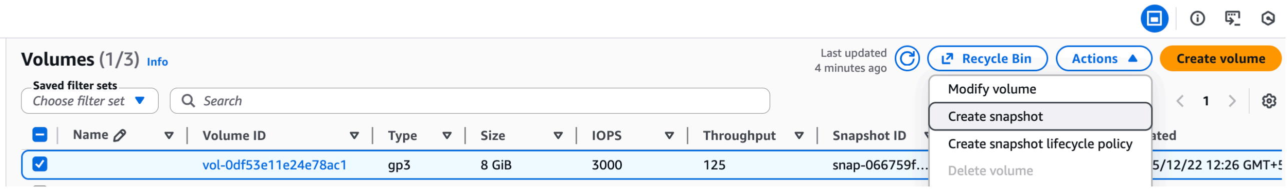
>> scroll to Volumes in EBS and select a instance



Volumes (3) Info

□	Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID	Source volume ID	Created
□		vol-0df53e11e24e78ac1	gp3	8 GiB	3000	125	snap-066759f…	-	2025/12/22 12:26 GMT+5
□		vol-075efbc4a8b2cafb4	gp3	30 GiB	3000	125	snap-0d227ab…	-	2025/12/22 11:40 GMT+5
□		vol-0fef2b30de9386785	gp3	8 GiB	3000	125	snap-066759f…	-	2025/12/20 19:12 GMT+5

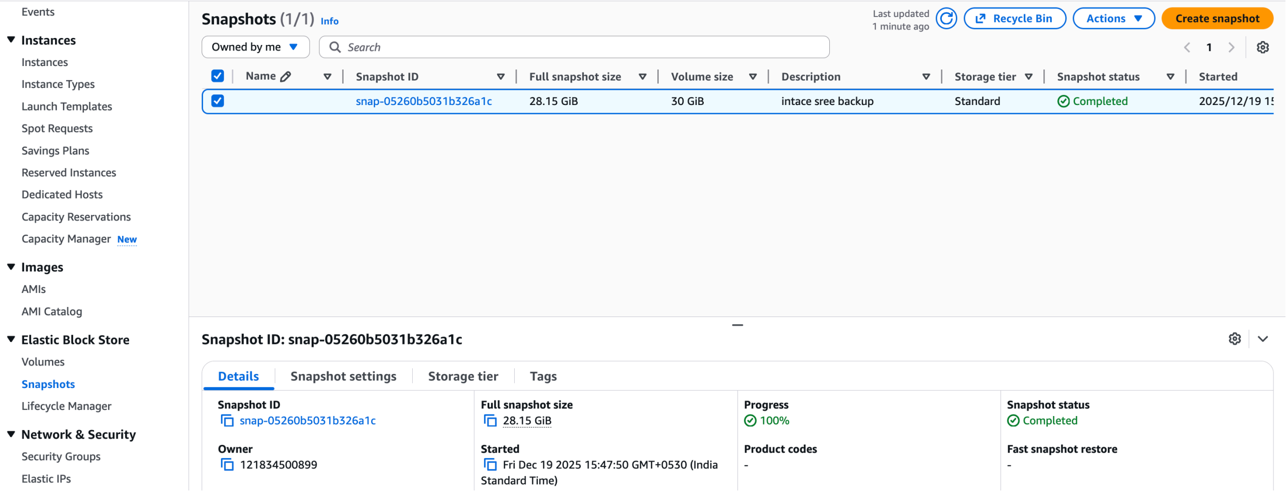
>> under actions creating snapshot



Volumes (1/3) Info

☑	Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID	…ated
☑		vol-0df53e11e24e78ac1	gp3	8 GiB	3000	125	snap-066759f…	5/12/22 12:26 GMT+5

>> now go to snapshots in EBS.
>> we can see the snapshot got created.



Snapshots (1/1) Info

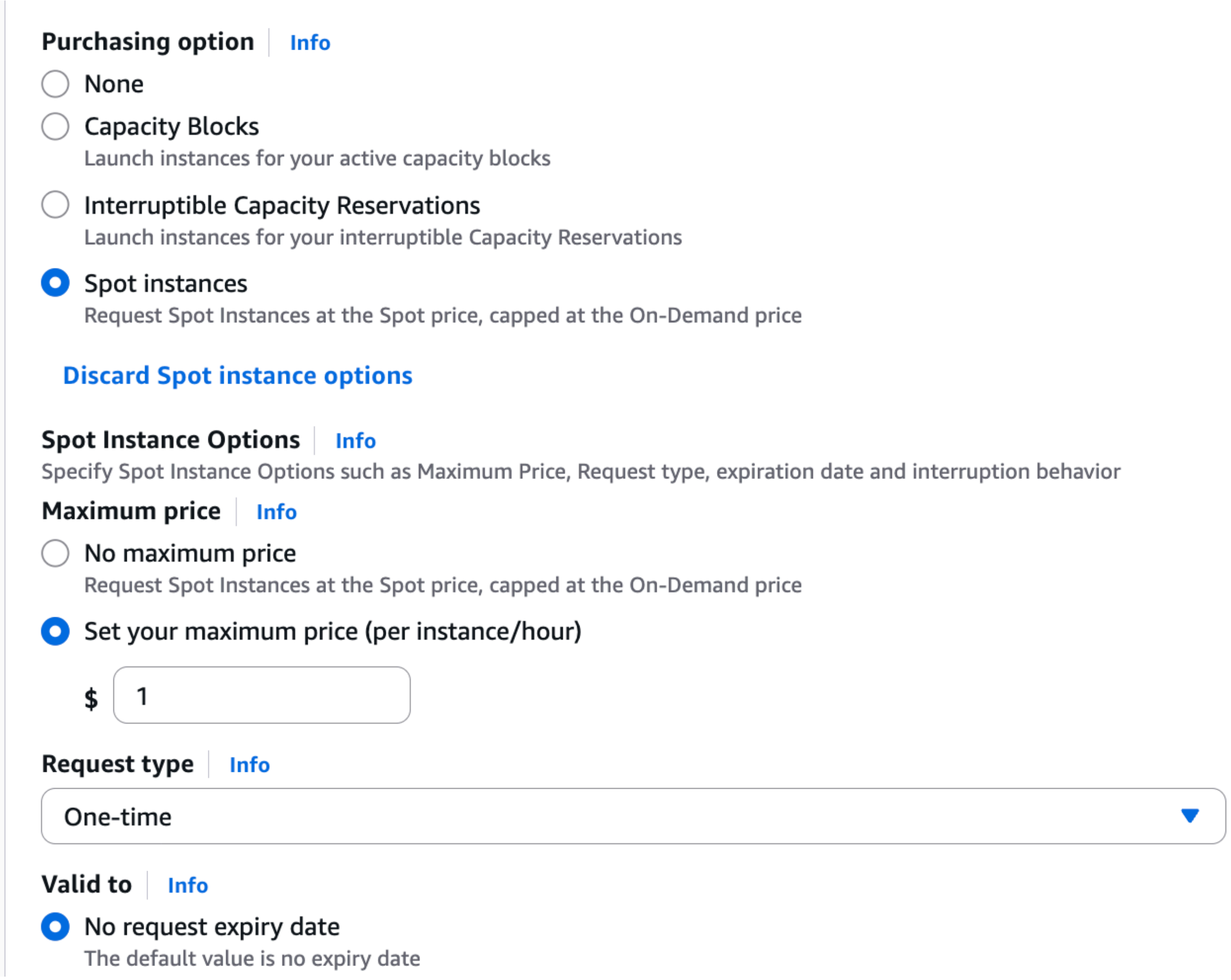
☑	Name	Snapshot ID	Full snapshot size	Volume size	Description	Storage tier	Snapshot status	Started
☑		snap-05260b5031b326a1c	28.15 GiB	30 GiB	intace sree backup	Standard	✓ Completed	2025/12/19 15

Snapshot ID: snap-05260b5031b326a1c

Snapshot ID	Full snapshot size	Progress	Snapshot status
snap-05260b5031b326a1c	28.15 GiB	✓ 100%	✓ Completed
Owner	**Started**	**Product codes**	**Fast snapshot restore**
121834500899	Fri Dec 19 2025 15:47:50 GMT+0530 (India Standard Time)	-	-

6. Launch Any EC2 Using the Spot Purchasing Option.

>> Before creating instance add details inside advanced details scrolldown and add spot instances



Purchasing option | Info

○ None

○ Capacity Blocks
Launch instances for your active capacity blocks

○ Interruptible Capacity Reservations
Launch instances for your interruptible Capacity Reservations

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

Discard Spot instance options

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)

$ 1

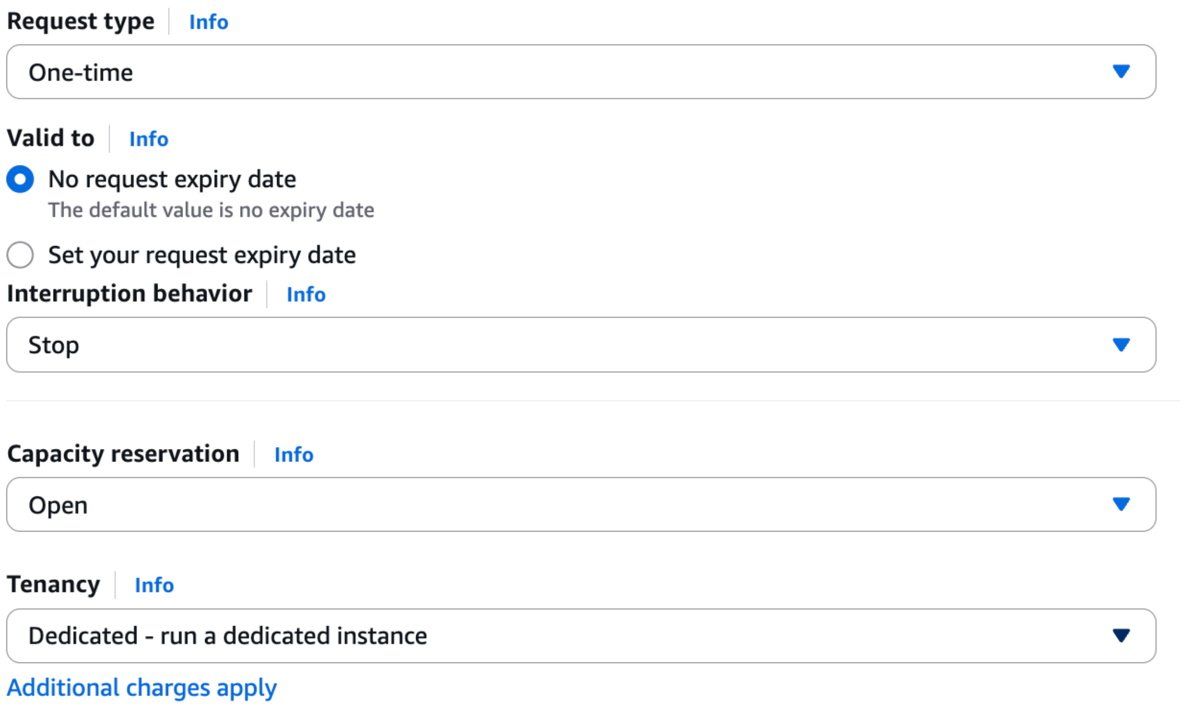
Request type | Info

One-time ▼

Valid to | Info

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

>> changed request type, instance behaviour and tenancy as per my requirement and successfully launched the instance.



Request type | Info

One-time ▼

Valid to | Info

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

○ Set your request expiry date

Interruption behavior | Info

Stop ▼

Capacity reservation | Info

Open ▼

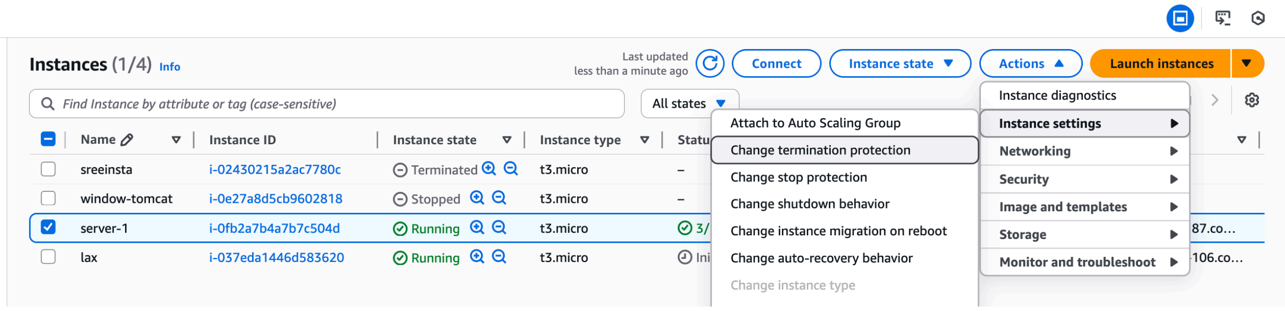
Tenancy | Info

Dedicated - run a dedicated instance ▼

Additional charges apply

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

>> here, we need to select a instance click on actions > instance settings > change termination protection.

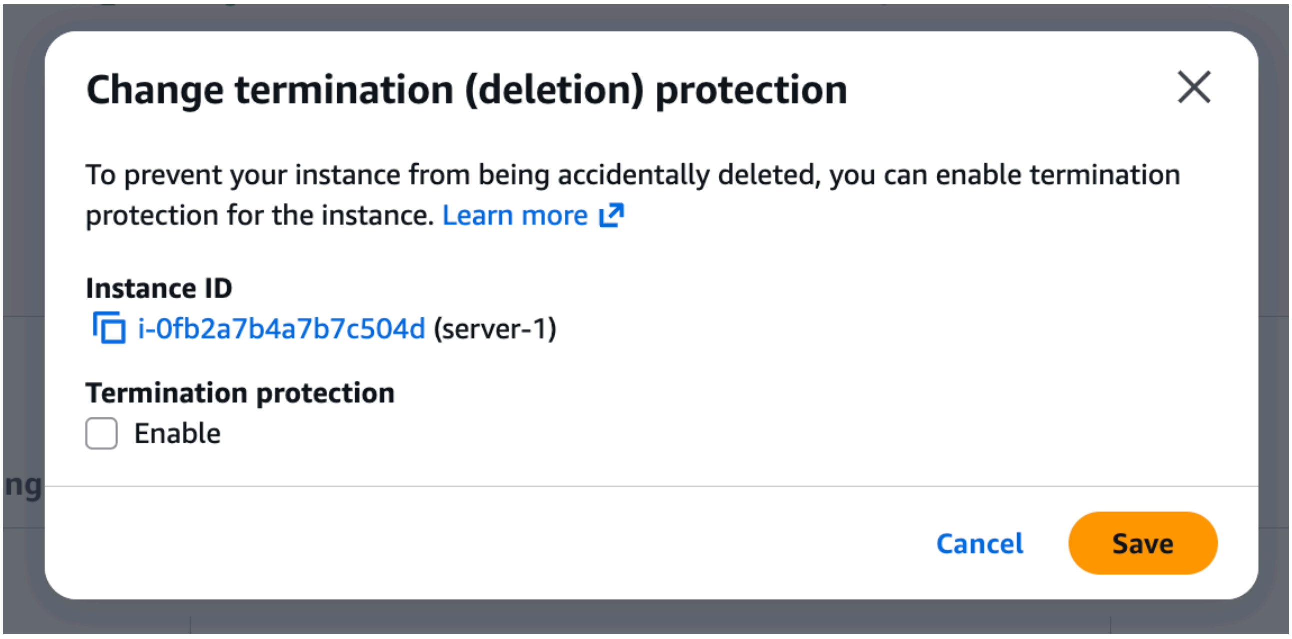


Instances (1/4) Info

Last updated less than a minute ago

	Name	Instance ID	Instance state	Instance type	Statu	
□	sreeinsta	i-02430215a2ac7780c	Terminated	t3.micro	–	
□	window-tomcat	i-0e27a8d5cb9602818	Stopped	t3.micro	–	
☑	server-1	i-0fb2a7b4a7b7c504d	Running	t3.micro	3/	87.co…
□	lax	i-037eda1446d583620	Running	t3.micro	Ini	106.co…

>> by clicking on save, now our EC2 instance cannot be terminated accidentally. If anyone tries to terminate it, AWS will show an error



Change termination (deletion) protection

To prevent your instance from being accidentally deleted, you can enable termination protection for the instance. Learn more

Instance ID

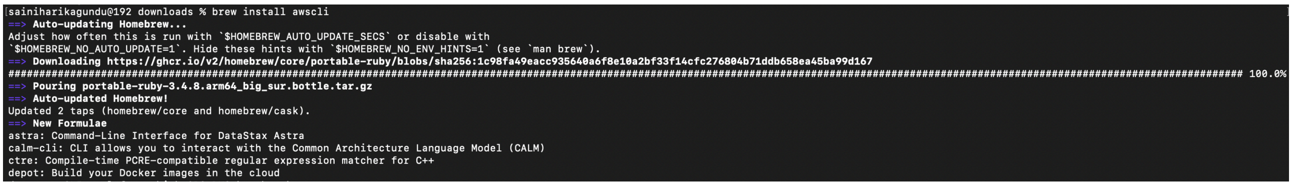
i-0fb2a7b4a7b7c504d (server-1)

Termination protection

□ Enable

8. Launch one EC2 using AWS CLI.

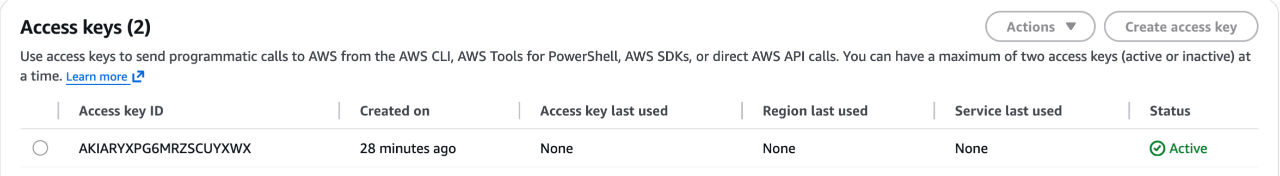
>> here, we have installed awscli in local terminal



```
sainiharikagundu@192 downloads % brew install awscli
==> Auto-updating Homebrew...
Adjust how often this is run with `$HOMEBREW_AUTO_UPDATE_SECS` or disable with
`$HOMEBREW_NO_AUTO_UPDATE=1`. Hide these hints with `$HOMEBREW_NO_ENV_HINTS=1` (see `man brew`).
==> Downloading https://ghcr.io/v2/homebrew/core/portable-ruby/blobs/sha256:1c98fa49eacc935640a6f8e10a2bf33f14cfc276804b71ddb658ea45ba99d167
######################################################################################################################################################################################################## 100.0%
==> Pouring portable-ruby-3.4.8.arm64_big_sur.bottle.tar.gz
==> Auto-updated Homebrew!
Updated 2 taps (homebrew/core and homebrew/cask).
==> New Formulae
astra: Command-Line Interface for DataStax Astra
calm-cli: CLI allows you to interact with the Common Architecture Language Model (CALM)
ctre: Compile-time PCRE-compatible regular expression matcher for C++
depot: Build your Docker images in the cloud
```

>> created a access key

>> here, created access key in access key section

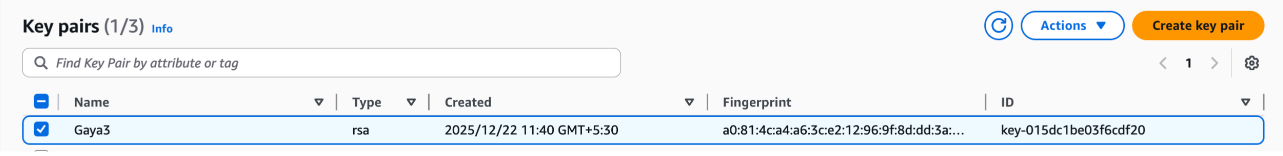


Access keys (2)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

	Access key ID	Created on	Access key last used	Region last used	Service last used	Status
○	AKIARYXPG6MRZSCUYXWX	28 minutes ago	None	None	None	✓ Active

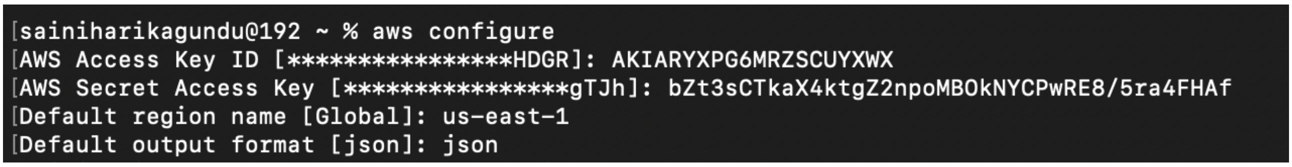
>> created key pair



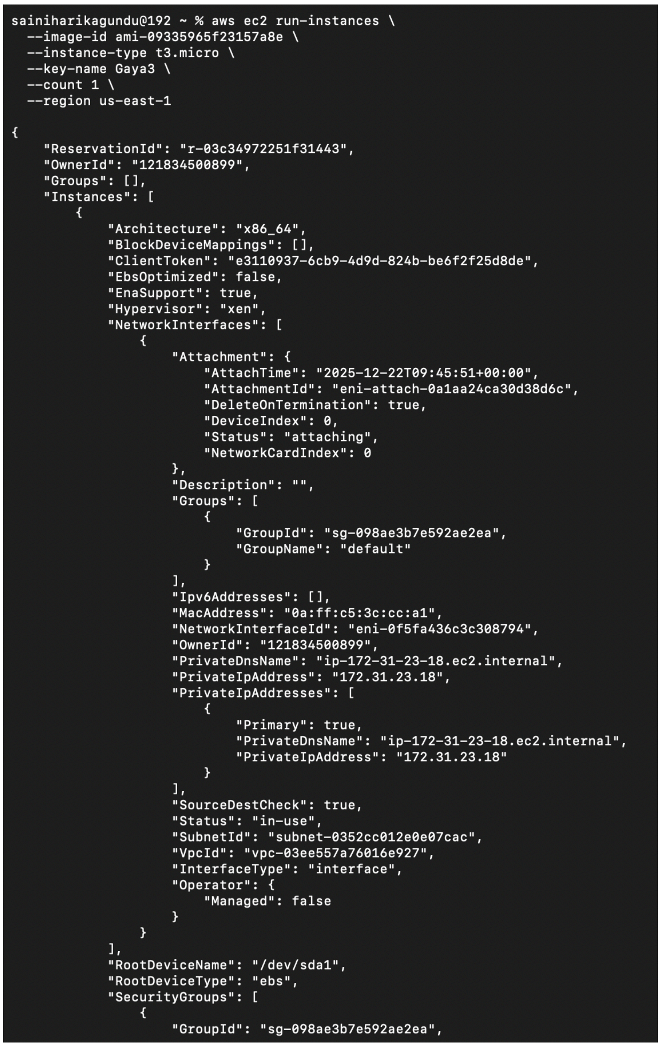
Key pairs (1/3) Info

☐	Name	Type	Created	Fingerprint	ID
☑	Gaya3	rsa	2025/12/22 11:40 GMT+5:30	a0:81:4c:a4:a6:3c:e2:12:96:9f:8d:dd:3a:...	key-015dc1be03f6cdf20

>> here, we used aws configure command for configuring key and password from above access key and added json as default output

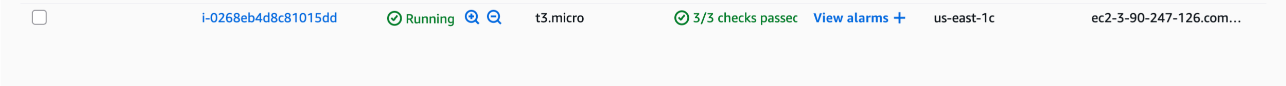


```
[sainiharikagundu@192 ~ % aws configure
[AWS Access Key ID [****************HDGR]: AKIARYXPG6MRZSCUYXWX
[AWS Secret Access Key [****************gTJh]: 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
[Default region name [Global]: us-east-1
[Default output format [json]: json
```



```
sainiharikagundu@192 ~ % aws ec2 run-instances \
  --image-id ami-09335965f23157a8e \
  --instance-type t3.micro \
  --key-name Gaya3 \
  --count 1 \
  --region us-east-1

{
    "ReservationId": "r-03c34972251f31443",
    "OwnerId": "121834500899",
    "Groups": [],
    "Instances": [
        {
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "e3110937-6cb9-4d9d-824b-be6f2f25d8de",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2025-12-22T09:45:51+00:00",
                        "AttachmentId": "eni-attach-0a1aa24ca30d38d6c",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching",
                        "NetworkCardIndex": 0
                    },
                    "Description": "",
                    "Groups": [
                        {
                            "GroupId": "sg-098ae3b7e592ae2ea",
                            "GroupName": "default"
                        }
                    ],
                    "Ipv6Addresses": [],
                    "MacAddress": "0a:ff:c5:3c:cc:a1",
                    "NetworkInterfaceId": "eni-0f5fa436c3c308794",
                    "OwnerId": "121834500899",
                    "PrivateDnsName": "ip-172-31-23-18.ec2.internal",
                    "PrivateIpAddress": "172.31.23.18",
                    "PrivateIpAddresses": [
                        {
                            "Primary": true,
                            "PrivateDnsName": "ip-172-31-23-18.ec2.internal",
                            "PrivateIpAddress": "172.31.23.18"
                        }
                    ],
                    "SourceDestCheck": true,
                    "Status": "in-use",
                    "SubnetId": "subnet-0352cc012e0e07cac",
                    "VpcId": "vpc-03ee557a76016e927",
                    "InterfaceType": "interface",
                    "Operator": {
                        "Managed": false
                    }
                }
            ],
            "RootDeviceName": "/dev/sda1",
            "RootDeviceType": "ebs",
            "SecurityGroups": [
                {
                    "GroupId": "sg-098ae3b7e592ae2ea",
```

>> we could see that we have created a ec2 server though our terminal



□	i-0268eb4d8c81015dd	✓ Running 🔍 🔍	t3.micro	✓ 3/3 checks passec	View alarms +	us-east-1c	ec2-3-90-247-126.com…