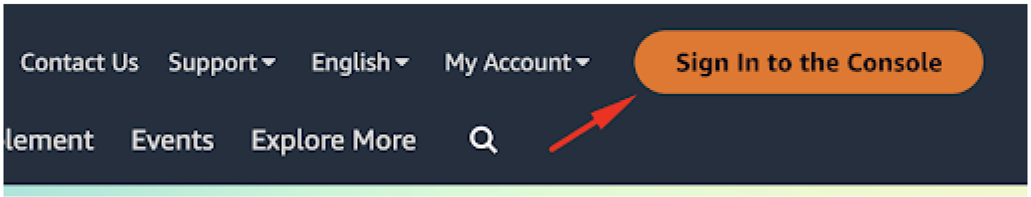
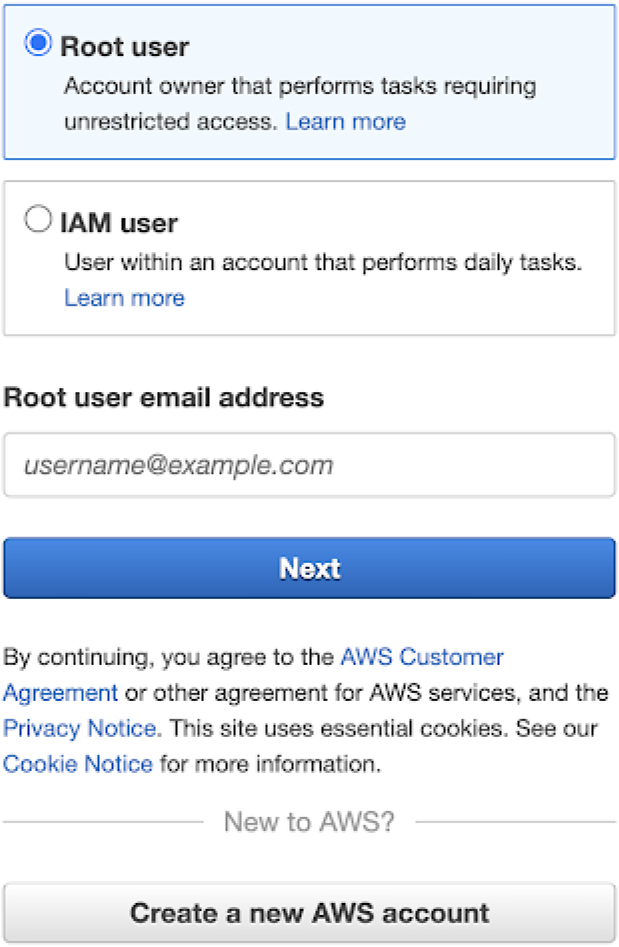
Step 1: Sign in to your AWS account

Before you can set up an EC2 instance with Amazon Linux, you need an AWS account.

First, go to the URL https://aws.amazon.com and click on the button that says “Sign in to the console.



Here you can select whether you want to log in as a Root user or an IAM user:



● **Root user**
Account owner that performs tasks requiring unrestricted access. Learn more

○ **IAM user**
User within an account that performs daily tasks. Learn more

Root user email address

username@example.com

By continuing, you agree to the AWS Customer Agreement or other agreement for AWS services, and the Privacy Notice. This site uses essential cookies. See our Cookie Notice for more information.

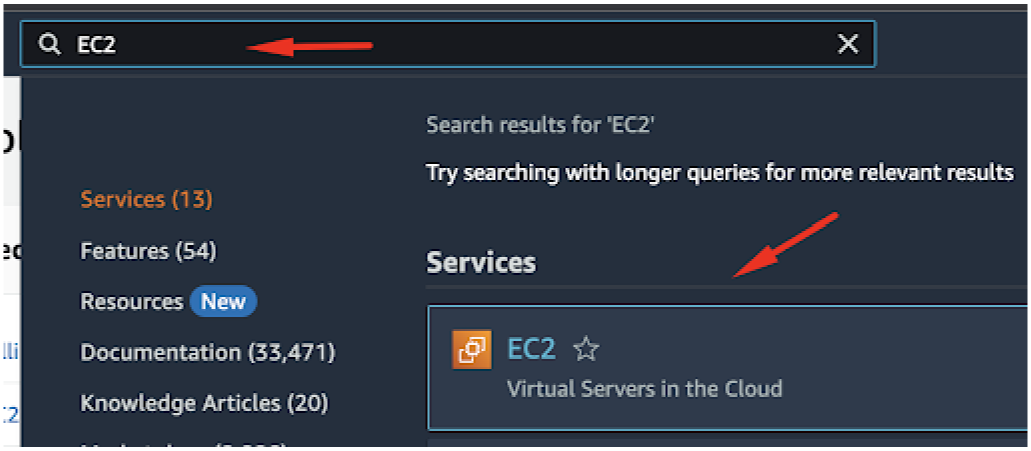
New to AWS?

Your choice of user login should consider your account's configuration and the existing user permissions. However, it is recommended to log in as a user with the lowest level of access privileges for best security practices.

If you don't have an AWS account, you can create one as you are often entitled to free tier benefits that you can use.

Step 2: Launch an EC2 instance

Once you have logged into your AWS account, in the search bar you can type EC2 in order to access the main page for provisioning and management.

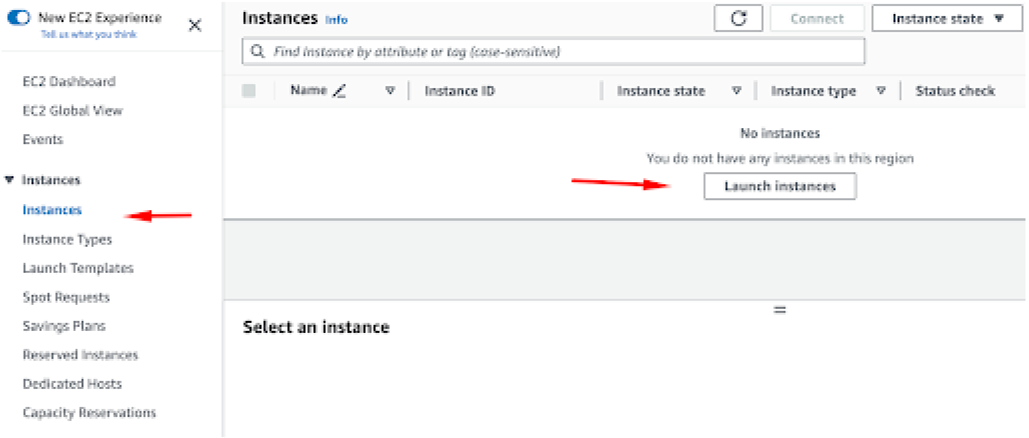


EC2

Search results for 'EC2'

Try searching with longer queries for more relevant results

Next, under Instances, you can select the option "Launch Instances."

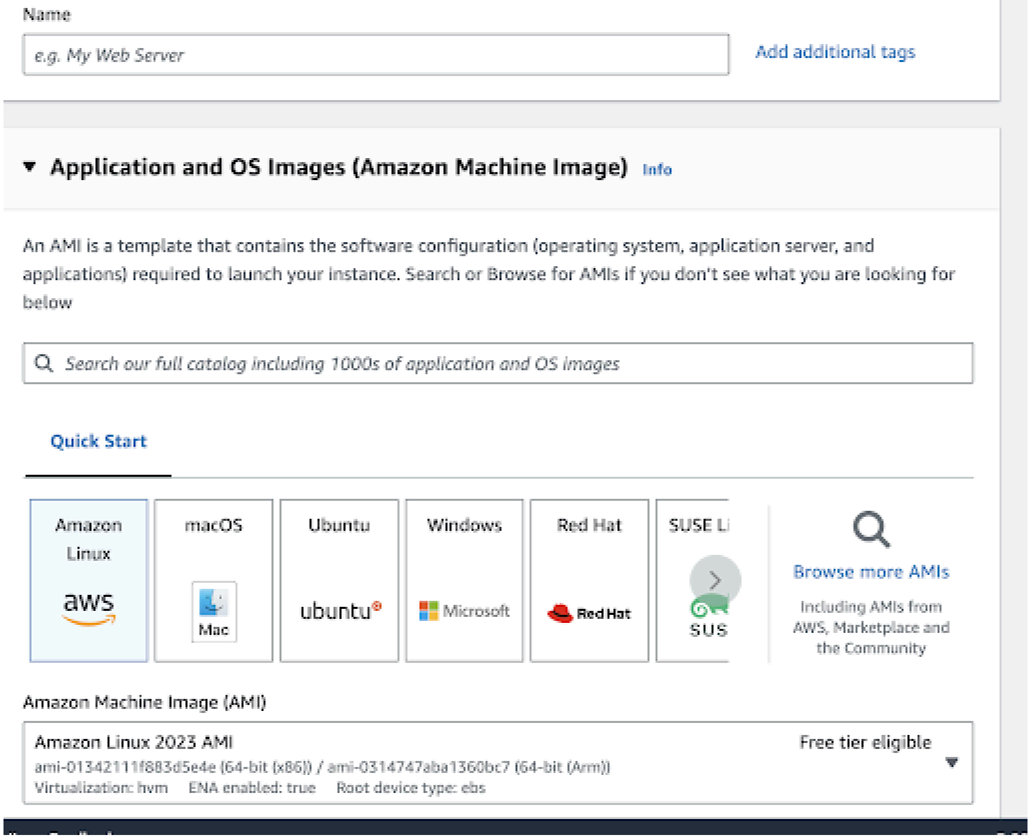


No instances

You do not have any instances in this region

Select an instance

We will land on the new page where we can select additional options, such as the Name of our instance, Application, and OS Images for our instance and we will select the default option — Amazon Linux.



Name

e.g. My Web Server

Add additional tags

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

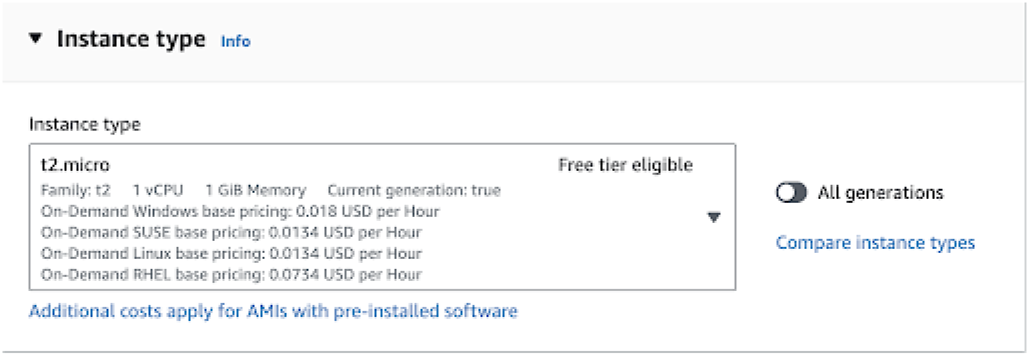
Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible
ami-01342111f883d5e4e (64-bit (x86)) / ami-0314747aba1360bc7 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

The “Free tier eligible” means that AWS offers a limited amount of its services at no cost for a specified period or up to certain usage limits. It’s a valuable opportunity for users to get started with AWS, but it’s crucial to be aware of the limitations and to monitor your usage to avoid unexpected charges once you exceed the free tier limits or duration.

When we scroll below we can choose the type of our instance. Amazon Linux has a minimum requirement of 512MB of RAM and 1 CPU core, so if we are testing things, we can select the t2.micro instance which fits into the free tier in AWS.



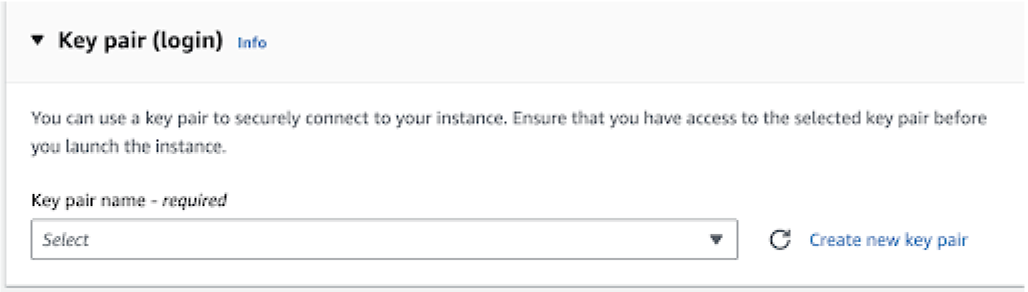
Instance type

t2.micro Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.018 USD per Hour
On-Demand SUSE base pricing: 0.0134 USD per Hour
On-Demand Linux base pricing: 0.0134 USD per Hour
On-Demand RHEL base pricing: 0.0734 USD per Hour

Additional costs apply for AMIs with pre-installed software

When you have a project and instance that should be production-ready, you can select instances with more RAM and CPU power, by choosing from the dropdown list.

Next, we will make sure to create a new security group for our instance.

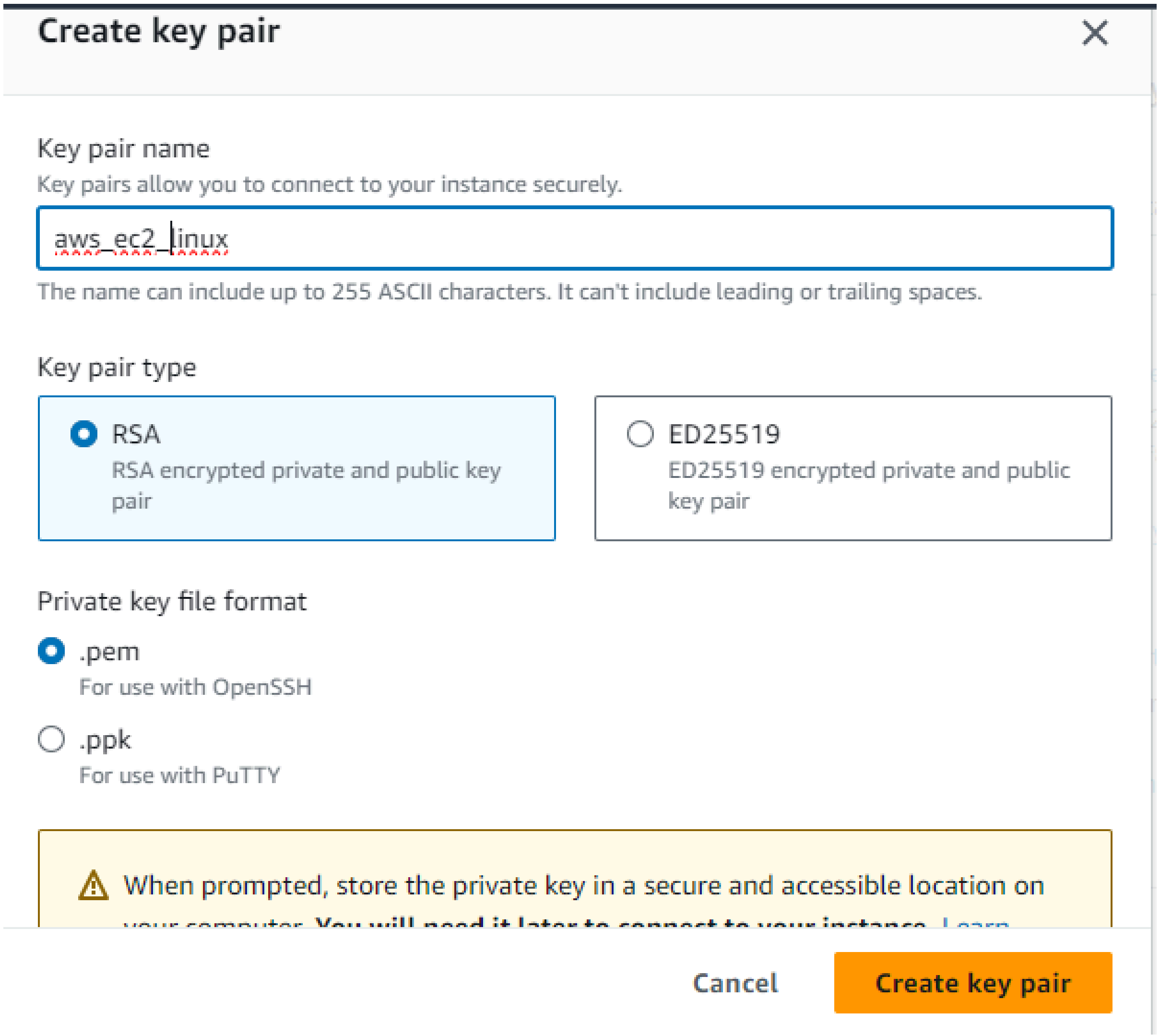


You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Select

Creating a new security group for your new EC2 instance is a fundamental security best practice in AWS. It allows you to define and enforce customized network access controls, adhere to the least privilege principle, and maintain better isolation and security for your EC2 instances.



Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

aws_ec2_linux

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

● RSA
RSA encrypted private and public key pair

○ ED25519
ED25519 encrypted private and public key pair

Private key file format

● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. Learn

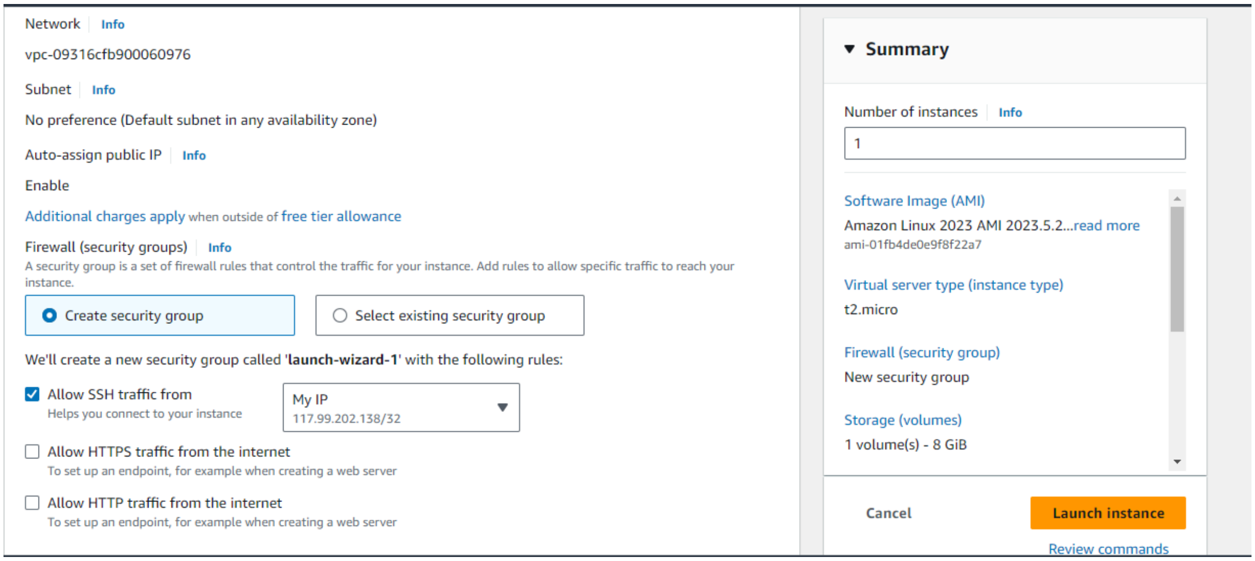
You should add the Key pair name and also the Key pair type which can be RSA or ED25519.

In general, both RSA and ED25519 are strong choices for SSH key authentication. The decision between them should consider your security requirements, compatibility needs, and the level of trust you place in modern cryptographic algorithms. For most use cases, either option will provide secure authentication when used correctly. RSA is one of the oldest and most widely supported SSH key algorithms, making it compatible with a wide range of SSH servers and clients while ED25519 keys are a good choice when security is a top priority and compatibility with older systems is not a concern.
We can select Private key file format which can be either in .pem format or .ppk in case you use a Windows SSH client such as PuTTy.

In our case, we will select the .pem format since we will be using the terminal.

Finally, click on the “Create key pair” option in order to generate the SSH key and this will prompt the download in your browser. Also, there is a warning from AWS that we should store our private keys in a secure and accessible location on our computers.

Further in our setup process, we would need to select our Network Settings.



Network | Info

vpc-09316cfb900060976

Subnet | Info

No preference (Default subnet in any availability zone)

Auto-assign public IP | Info

Enable

Additional charges apply when outside of free tier allowance

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
My IP
117.99.202.138/32

□ 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

▼ **Summary**

Number of instances | Info

1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
ami-01fb4de0e9f8f22a7

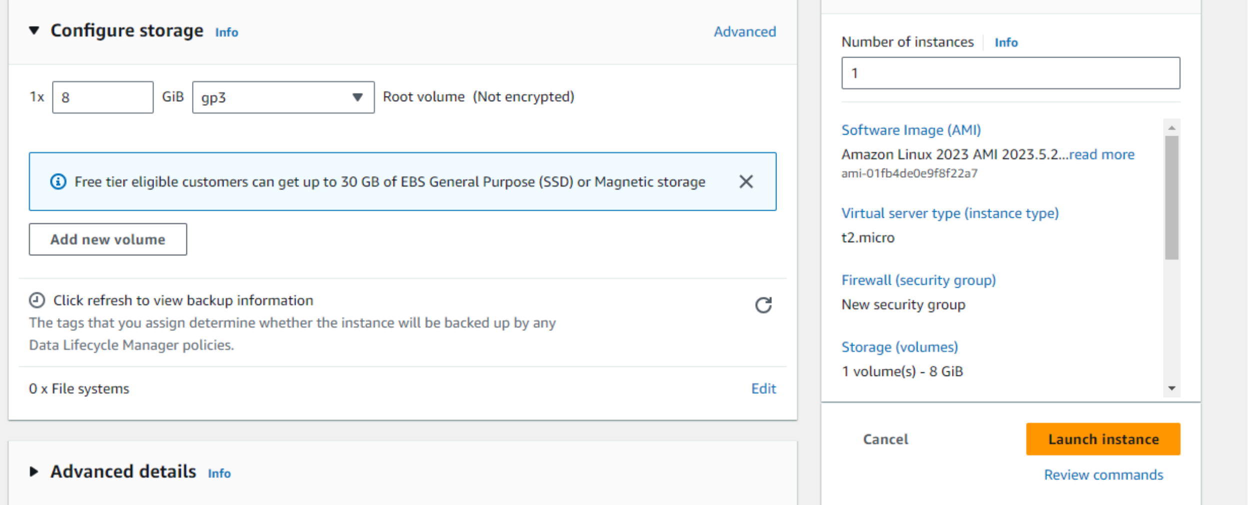
Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

For the first option, we need to access our EC2 instance through SSH but instead of allowing SSH traffic from “0.0.0.0/0” (which means from anywhere), restrict it to only allow connections from specific trusted IP addresses or ranges. This limits the exposure of your SSH port to the internet and reduces the risk of unauthorized access. You can also use CIDR notation to specify the IP ranges

Next, we will select our storage options from the menu:



▾ Configure storage Info

Advanced

1x 8 GiB gp3 Root volume (Not encrypted)

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

Add new volume

Click refresh to view backup information

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems Edit

▸ Advanced details Info

Number of instances Info

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.5.2...read more

ami-01fb4de0e9f8f22a7

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

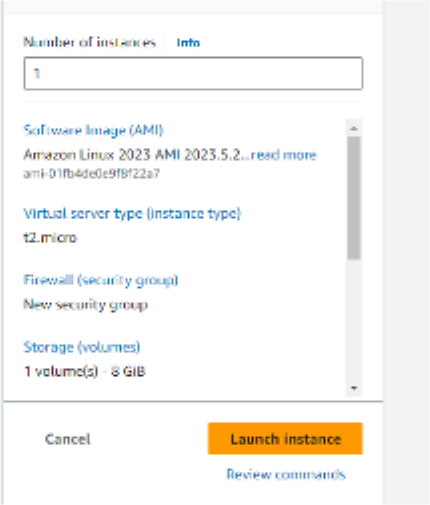
Cancel

Launch instance

Review commands

In this case, we have selected a new volume with 8 GB of storage with the general purpose, gp3 type of volume. For production purposes, make sure to increase the size of the volume based on your needs. In general, gp3 EBS volumes offer more flexibility, burst performance, and often a lower cost per gigabyte compared to gp2 volumes. When selecting between them, consider your application’s specific I/O and performance needs, as well as your budget constraints.

Our final step is confirming the selected options in the summary and we can proceed to launch the instance.



Number of instances | Info

1

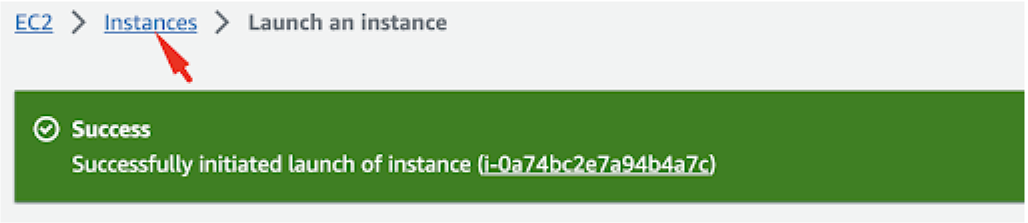
Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
ami-01fb4de0e9f8f22a7

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

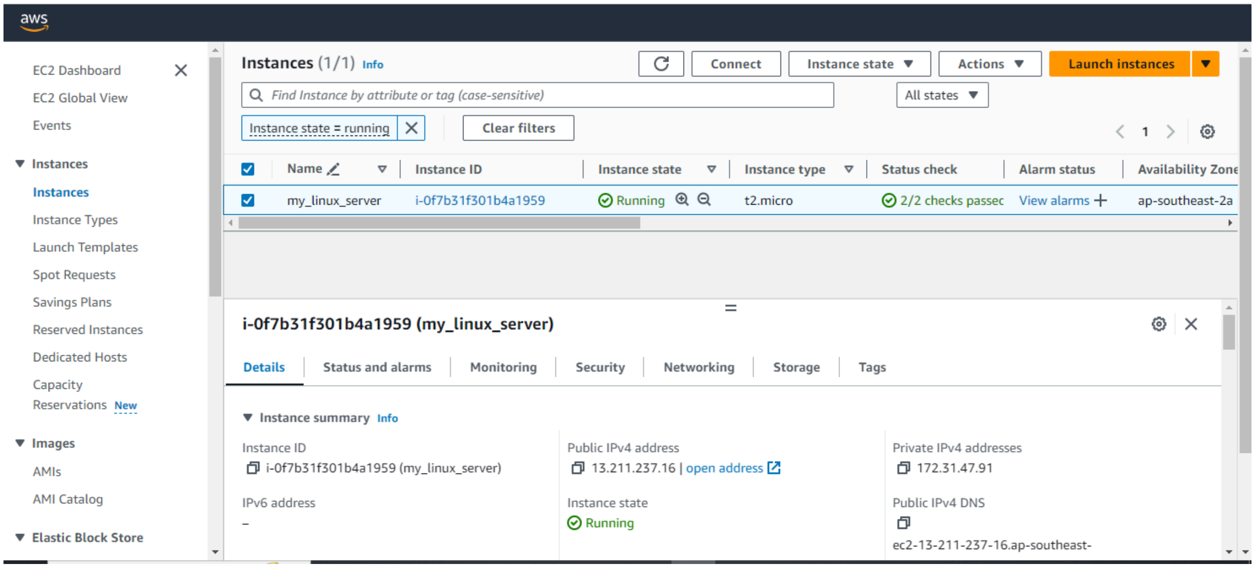
You will get the notification that the instance is successfully launched and you'll return to your Instances list.



EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-0a74bc2e7a94b4a7c)

After a short wait, typically with a fast provisioning time, you'll be able to locate your instance in the list.



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☑	my_linux_server	i-0f7b31f301b4a1959	Running	t2.micro	2/2 checks passec	View alarms +	ap-southeast-2a

i-0f7b31f301b4a1959 (my_linux_server)

Instance summary Info

Instance ID
i-0f7b31f301b4a1959 (my_linux_server)

Public IPv4 address
13.211.237.16 | open address

Private IPv4 addresses
172.31.47.91

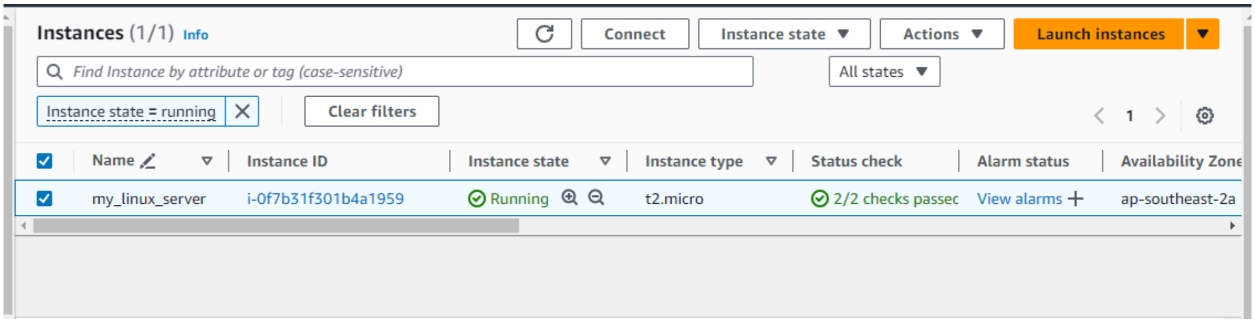
IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-13-211-237-16.ap-southeast-

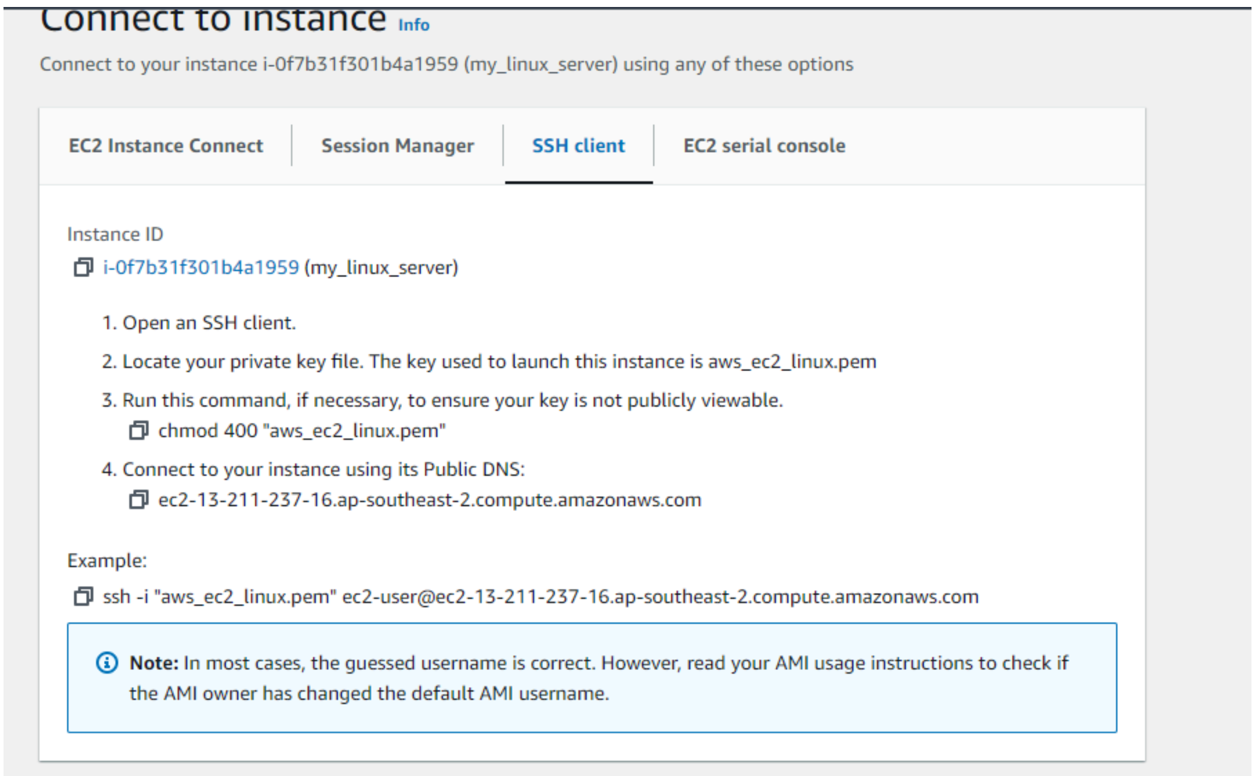
Step 3: Connect to your Amazon Linux instance

Now the next step is to connect to the instance via SSH. We can do so by using our Instances menu and clicking **Connect:**



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☑	my_linux_server	i-0f7b31f301b4a1959	Running	t2.micro	2/2 checks passec	View alarms +	ap-southeast-2a

Here we will see a new menu where we will select the **SSH client** menu item:



Connect to instance Info

Connect to your instance i-0f7b31f301b4a1959 (my_linux_server) using any of these options

Instance ID

i-0f7b31f301b4a1959 (my_linux_server)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is aws_ec2_linux.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   ```
   chmod 400 "aws_ec2_linux.pem"
   ```
4. Connect to your instance using its Public DNS:
   ```
   ec2-13-211-237-16.ap-southeast-2.compute.amazonaws.com
   ```

Example:

```
ssh -i "aws_ec2_linux.pem" ec2-user@ec2-13-211-237-16.ap-southeast-2.compute.amazonaws.com
```

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

After opening your terminal and finding where you have downloaded the .pem key, it is important to change permissions to it, for security reasons, so that the key is not publicly viewable, and in our case, we will run the following command:

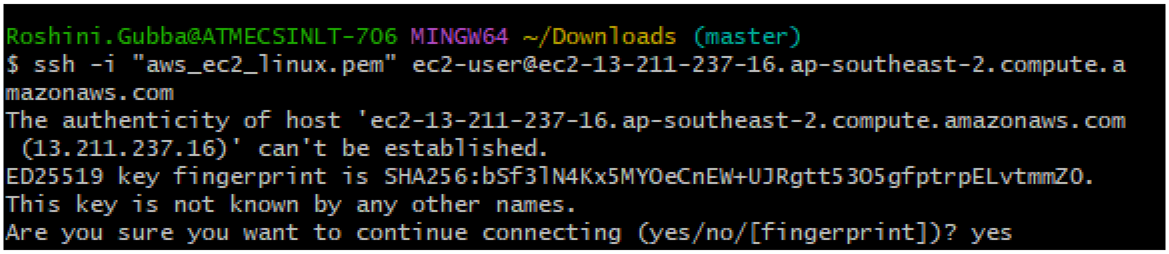
```
chmod 400 jumpcloud_amazon_limux.pem
```

Next, we will use the SSH key to connect to our instance:

```
ssh -i “jumpcloud_amazon_linux.pem”
ec2-user@ec2-3-79-150-186.eu-central-1.compute.amazonaws.com
```

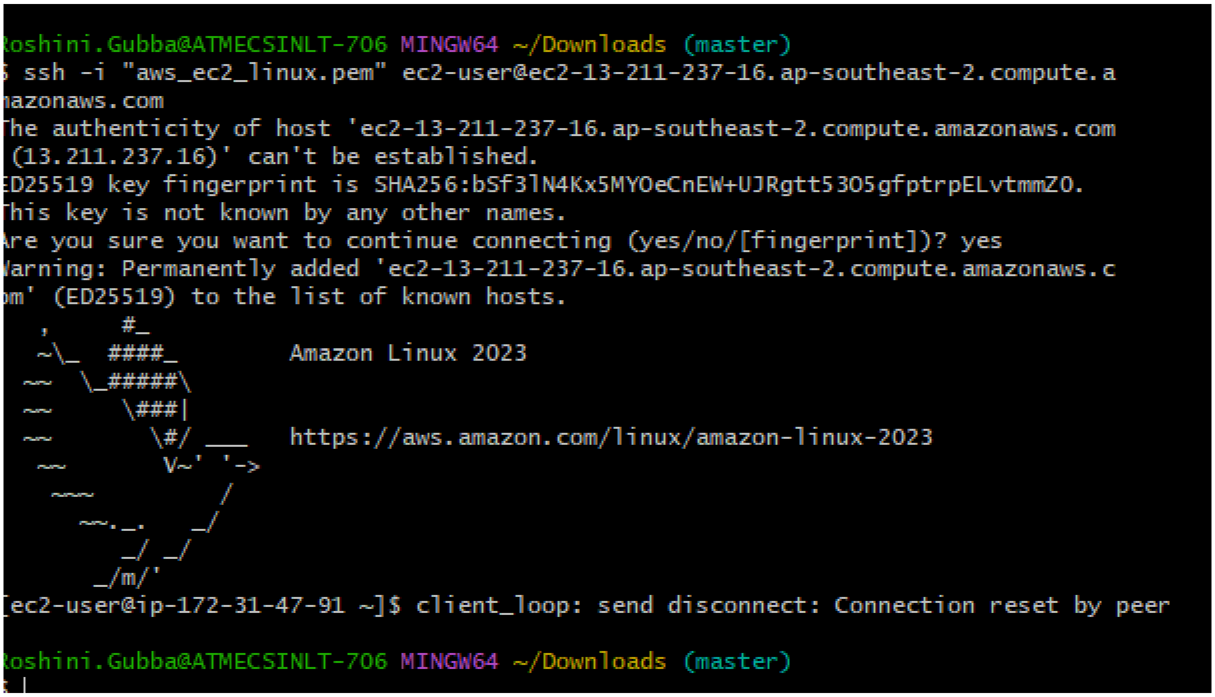
Here, the -i flag in SSH is used to specify the path to the private key file to be used for authentication when connecting to a remote server. It allows you to choose a specific key file when you have multiple key pairs or non-standard key file names and locations.

When we log into our Amazon Linux instance for the first time, we need to confirm the authenticity of the host. Here you can type yes and press Enter.



```
Roshini.Gubba@ATMECSINLT-706 MINGW64 ~/Downloads (master)
$ ssh -i "aws_ec2_linux.pem" ec2-user@ec2-13-211-237-16.ap-southeast-2.compute.a
mazonaws.com
The authenticity of host 'ec2-13-211-237-16.ap-southeast-2.compute.amazonaws.com
 (13.211.237.16)' can't be established.
ED25519 key fingerprint is SHA256:bSf3lN4Kx5MYOeCnEW+UJRgtt53O5gfptrpELvtmmZO.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

After this action, we will be logged into our instance

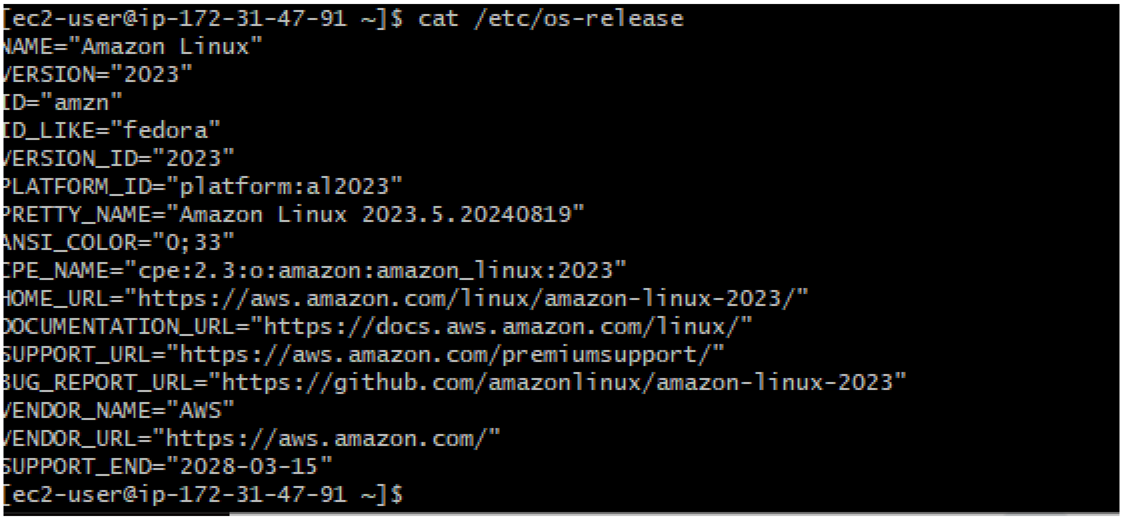


```
oshini.Gubba@ATMECSINLT-706 MINGW64 ~/Downloads (master)
 ssh -i "aws_ec2_linux.pem" ec2-user@ec2-13-211-237-16.ap-southeast-2.compute.a
azonaws.com
he authenticity of host 'ec2-13-211-237-16.ap-southeast-2.compute.amazonaws.com
(13.211.237.16)' can't be established.
D25519 key fingerprint is SHA256:bSf3lN4Kx5MYOeCnEW+UJRgtt53O5gfptrpELvtmmZO.
his key is not known by any other names.
re you sure you want to continue connecting (yes/no/[fingerprint])? yes
arning: Permanently added 'ec2-13-211-237-16.ap-southeast-2.compute.amazonaws.c
m' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
ec2-user@ip-172-31-47-91 ~]$ client_loop: send disconnect: Connection reset by peer

oshini.Gubba@ATMECSINLT-706 MINGW64 ~/Downloads (master)
 |
```

You can always verify the version of your Amazon Linux by typing:

cat /etc/os-release



```
[ec2-user@ip-172-31-47-91 ~]$ cat /etc/os-release
NAME="Amazon Linux"
VERSION="2023"
ID="amzn"
ID_LIKE="fedora"
VERSION_ID="2023"
PLATFORM_ID="platform:al2023"
PRETTY_NAME="Amazon Linux 2023.5.20240819"
ANSI_COLOR="0;33"
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2023"
HOME_URL="https://aws.amazon.com/linux/amazon-linux-2023/"
DOCUMENTATION_URL="https://docs.aws.amazon.com/linux/"
SUPPORT_URL="https://aws.amazon.com/premiumsupport/"
BUG_REPORT_URL="https://github.com/amazonlinux/amazon-linux-2023"
VENDOR_NAME="AWS"
VENDOR_URL="https://aws.amazon.com/"
SUPPORT_END="2028-03-15"
[ec2-user@ip-172-31-47-91 ~]$
```