EC2 (Elastic Compute Cloud)

> **EC2 lets you "rent" virtual computers in the cloud.**

Think of EC2 instances as **virtual machines (VMs)** that you can:

- **Spin up instantly:** Get a new computer ready in minutes.
- **Choose the size:** Pick from tiny ones (like a small laptop) to massive ones with many CPUs and powerful GPUs (like a supercomputer).
- **Pay only for what you use:** You pay by the hour (or even by the second) for the time the virtual computer is running. When you shut it down, you stop paying for the compute time.
- **Customize:** You can choose the operating system (Linux, Windows), the software installed, and even attach extra storage.

🔧 Simple Analogy:

Imagine you walk into a **cloud computer shop**. You say:

> “Give me a computer with Ubuntu, 8 GB RAM, 4 CPUs, and a GPU... but I only want it for 3 hours.”

AWS says:

> “Sure. We’ll spin up that computer instantly, charge you just for the time, and shut it down when you're done.”

That’s EC2.

💡 **The "Elastic" in EC2 means it's super flexible – you can easily scale up (get bigger computers or more of them) or scale down (shut them down) based on your needs.**

💡

⚙️ EC2 = Virtual Machine = Instance

🧱 EC2 Instance Has 3 Key Components:

Component	Meaning
AMI (Amazon Machine Image)	Like a pre-installed **OS** (e.g. Ubuntu 22, Amazon Linux 2)
Instance Type	Hardware power (RAM, CPU, GPU) — like `t2.micro` , `t3.large` , `p3.2xlarge`
Key Pair	SSH access (your password to connect securely)

🔍 What Can You Do With EC2?

Use Case	Description
🧪 **Train machine learning models**	Use powerful CPU or GPU machines to run large datasets
🌐 **Host websites/apps**	Flask, FastAPI, Streamlit, Django, etc.
📊 **Run Python scripts on schedule**	Automate tasks without keeping your laptop on
📁 **Transfer and preprocess large files**	Do heavy data operations in the cloud
🧠 **Deploy AI models**	Serve models as APIs with fast response

How is EC2 useful for Data Science?

EC2 is a powerhouse for data scientists because it provides the raw compute power needed for demanding tasks that your local machine might not handle:

- **Big Data Processing:** Running Spark jobs, Dask clusters, or custom Python scripts on datasets that are too large for your laptop's memory.
- **Machine Learning Model Training:** Especially for deep learning, training models can require specialized hardware like **GPUs**. EC2 offers instances specifically designed with powerful GPUs (e.g., P, G, or Inf instances).
- **Custom Environments:** You need a very specific version of Python, TensorFlow, PyTorch, or a custom library? You can set up your EC2 instance exactly how you need it.
- **Experimentation & Development:** Spin up an instance, try out a new library or framework, and then terminate it when you're done, without cluttering your local machine.
- **Hosting APIs:** After training a model, you might want to deploy it as a web service (API) for real-time predictions. An EC2 instance can host this API.
- **Running Jupyter Notebooks:** You can set up a Jupyter Notebook server on an EC2 instance, allowing you to access a powerful coding environment from your web browser, even on a less powerful local machine.

✅ Free Tier for Beginners

You can use **EC2 completely free** for up to **750 hours/month** using:

- Instance Type: `t2.micro` or `t3.micro`
- OS: Amazon Linux 2, Ubuntu, etc.
- Region: Any (e.g. `ap-south-1` = Mumbai)

🟢 **This is great for:**

- Practicing AWS
- Hosting small apps

- Running lightweight ML tasks

🚀 Step-by-Step: Launch Your First EC2 Instance

🔷 Step 1: Go to EC2

- Login to AWS Console
- In the search bar, type **EC2**
- Click: **EC2 Dashboard → Launch Instance**

🔷 Step 2: Fill Instance Details

Field	What to Select
Name	`my-first-ec2`
AMI	`Ubuntu Server 22.04` ✅ beginner friendly
Instance Type	`t2.micro` (Free Tier)
Key Pair	Create new → Download `.pem` file (keep it safe!)
Storage	Default (8 GB is fine)
Network Settings	Allow SSH (`port 22`), HTTP (`port 80` if hosting site)
Click	✅ "Launch instance"

In 1–2 minutes, your instance is running!

🔷 Step 3: Connect to EC2 (via SSH)

1. Open your terminal (CMD, PowerShell, or Git Bash)
2. Navigate to where you saved the `.pem` file
3. Make sure the file has proper permissions:

```
chmod 400 my-key.pem
```

4. Connect using SSH:

```
ssh -i my-key.pem ubuntu@<EC2-Public-IP>
```

Now you're **inside your cloud computer**.

You can now:

- Run Python code
- Install packages (`sudo apt update && sudo apt install python3-pip`)
- Upload scripts or notebooks
- Train models

💰 EC2 Pricing (Simplified)

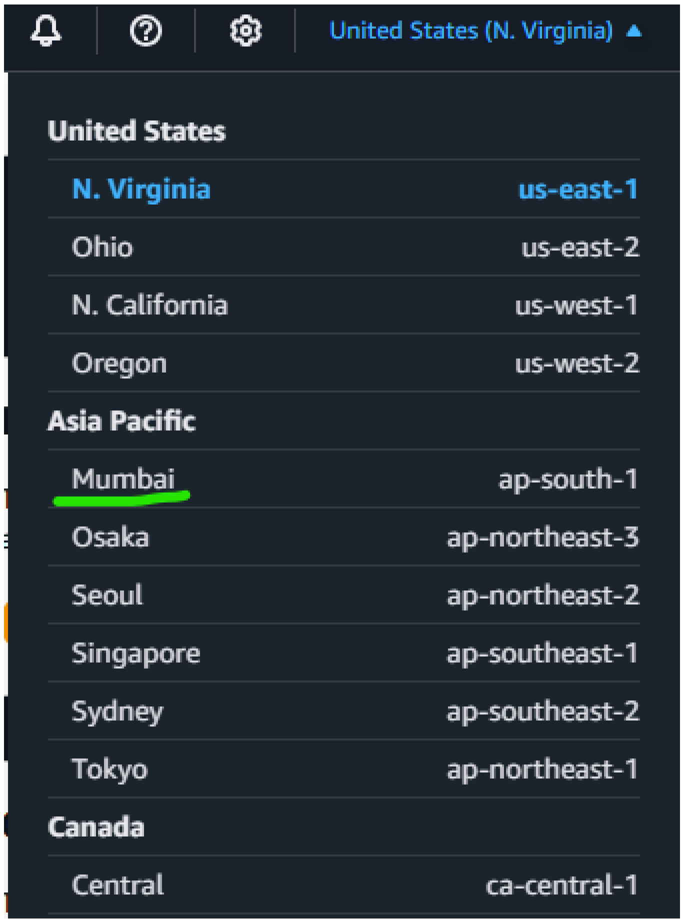
Type	Cost (India)	Best For
t2.micro	₹8/hour (~₹500/month)	Free tier eligible (750 hrs free).
g4dn.xlarge (GPU)	₹84/hour	AI/ML training.
Spot Instances	~70% cheaper	Non-urgent jobs (e.g., batch processing).

Key Point: **Stop instances when unused** to avoid bills!

Practical

Open EC2

💡 **FIRST, SELECT THE RESION**



United States (N. Virginia)

United States

N. Virginia	us-east-1
Ohio	us-east-2
N. California	us-west-1
Oregon	us-west-2

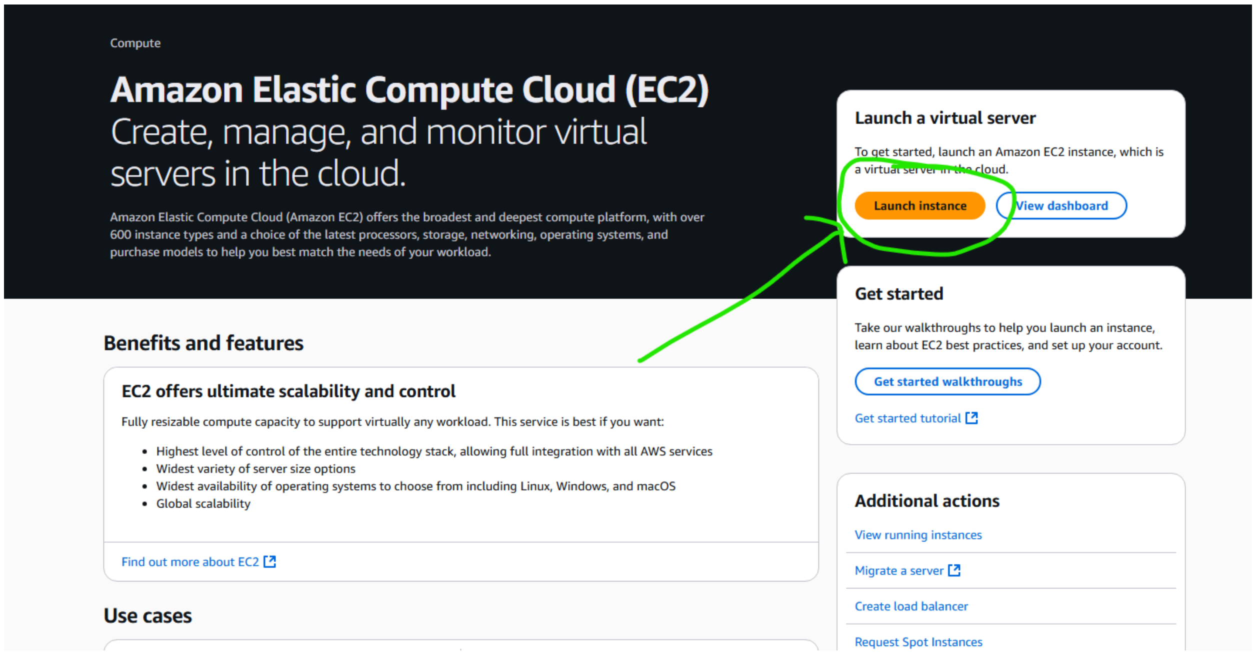
Asia Pacific

Mumbai	ap-south-1
Osaka	ap-northeast-3
Seoul	ap-northeast-2
Singapore	ap-southeast-1
Sydney	ap-southeast-2
Tokyo	ap-northeast-1

Canada

Central	ca-central-1

Click Launch Instance



Compute

Amazon Elastic Compute Cloud (EC2)
Create, manage, and monitor virtual servers in the cloud.

Amazon Elastic Compute Cloud (Amazon EC2) offers the broadest and deepest compute platform, with over 600 instance types and a choice of the latest processors, storage, networking, operating systems, and purchase models to help you best match the needs of your workload.

Launch a virtual server

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance | View dashboard

Get started

Take our walkthroughs to help you launch an instance, learn about EC2 best practices, and set up your account.

Get started walkthroughs

Get started tutorial

Additional actions

View running instances

Migrate a server

Create load balancer

Request Spot Instances

Benefits and features

EC2 offers ultimate scalability and control

Fully resizable compute capacity to support virtually any workload. This service is best if you want:

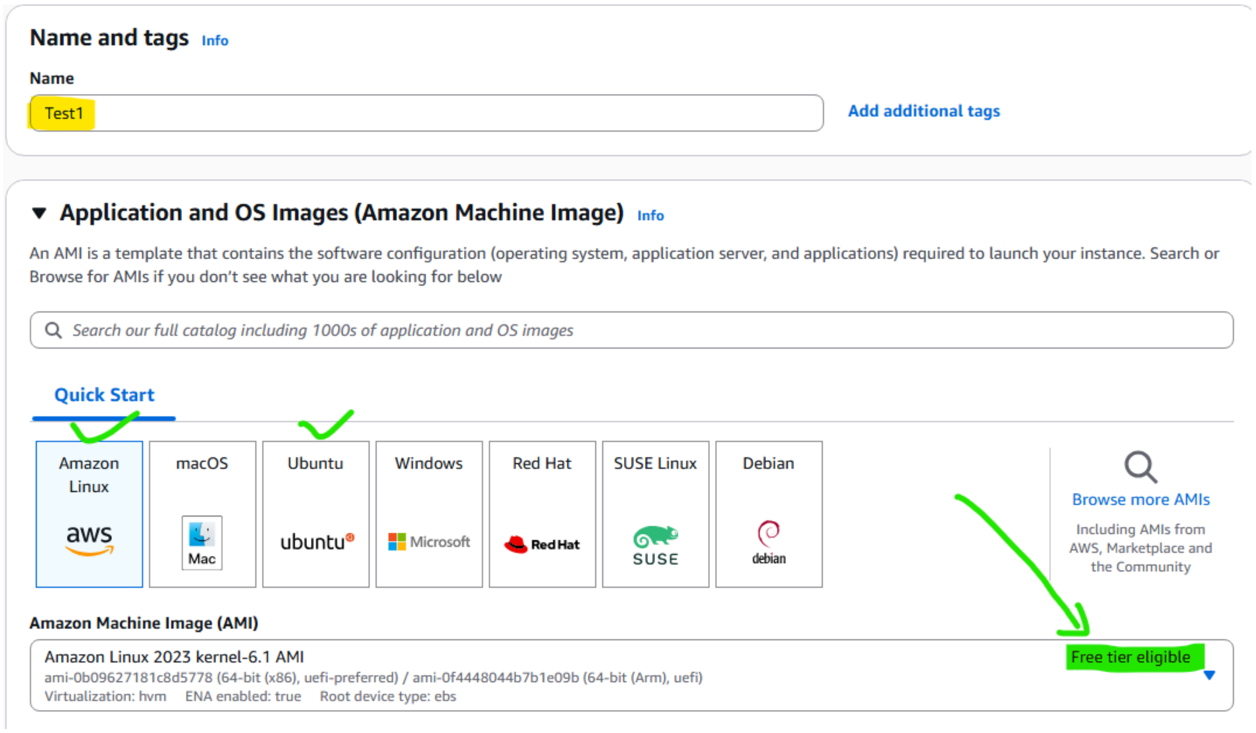
- Highest level of control of the entire technology stack, allowing full integration with all AWS services
- Widest variety of server size options
- Widest availability of operating systems to choose from including Linux, Windows, and macOS
- Global scalability

Find out more about EC2

Use cases

Enter name & Select the AMI

- Check the free tier eligibility



Name and tags Info

Name: Test1

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

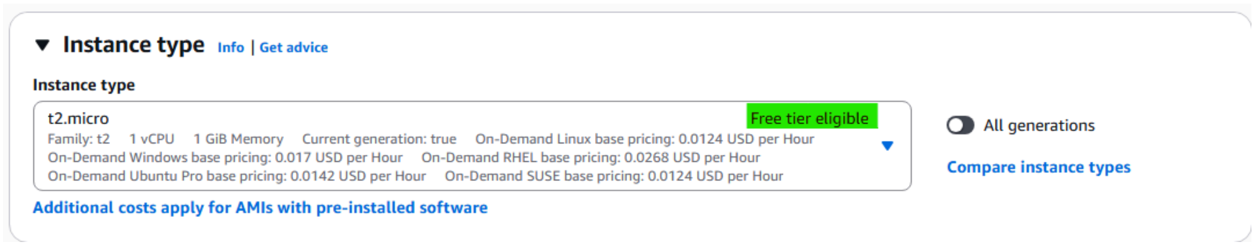
Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI
ami-0b09627181c8d5778 (64-bit (x86), uefi-preferred) / ami-0f4448044b7b1e09b (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Set Instance type



Instance type Info | Get advice

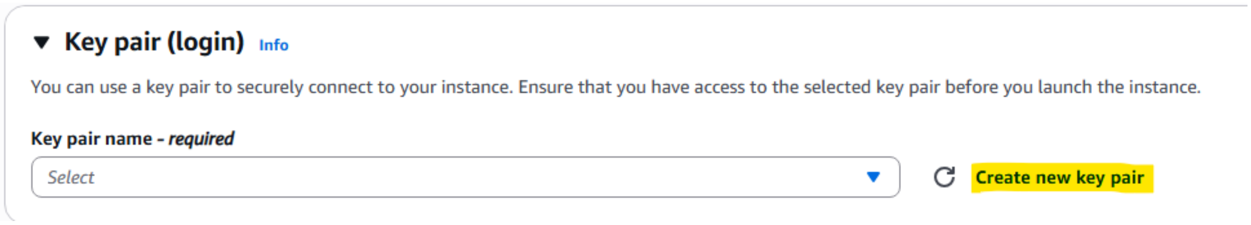
Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true On-Demand Linux base pricing: 0.0124 USD per Hour
On-Demand Windows base pricing: 0.017 USD per Hour On-Demand RHEL base pricing: 0.0268 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour On-Demand SUSE base pricing: 0.0124 USD per Hour

Additional costs apply for AMIs with pre-installed software

Key pair (login)

- Required to access this with your personal computer
- Create the key-pair

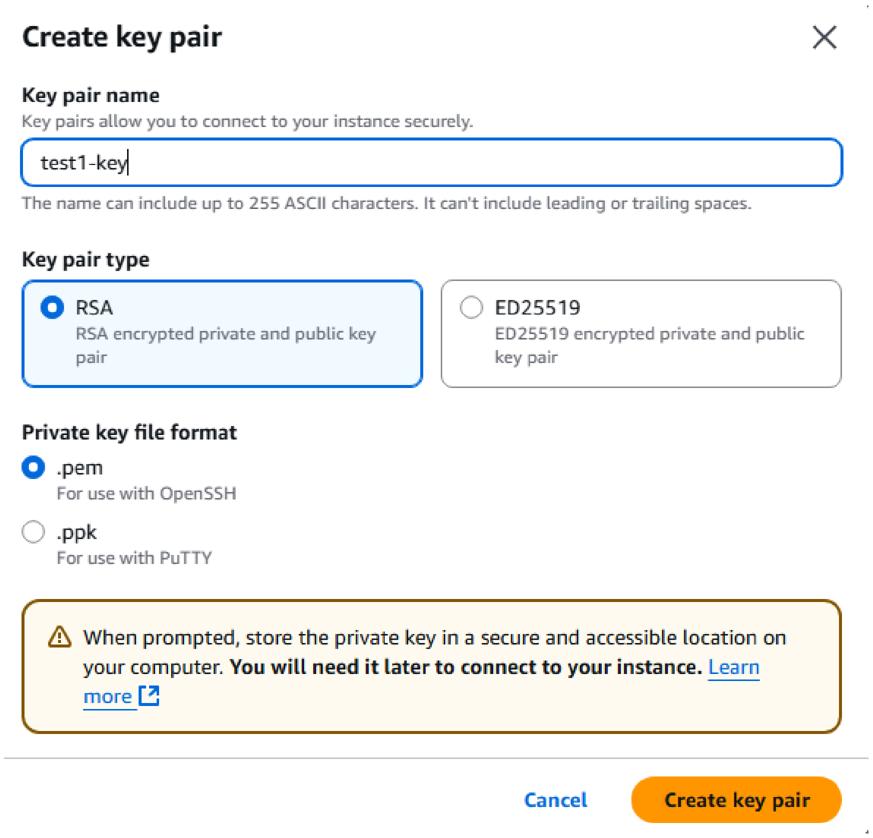

▼ Key pair (login) Info

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*

Select

Create new key pair


Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

test1-key

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

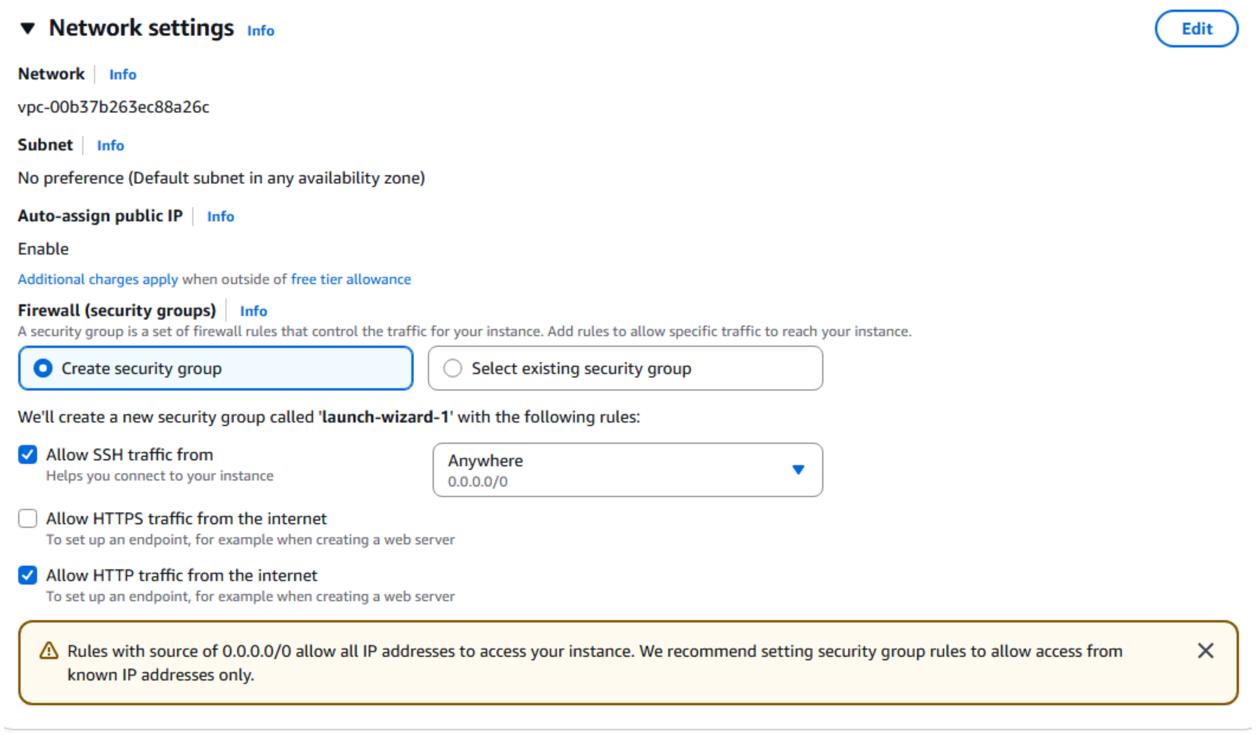
.ppk For use with PuTTY

⚠ 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 more

Cancel   Create key pair

💡 **.pem file will be downloaded.**

Network settings


▼ Network settings Info

Edit

Network | Info

vpc-00b37b263ec88a26c

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:

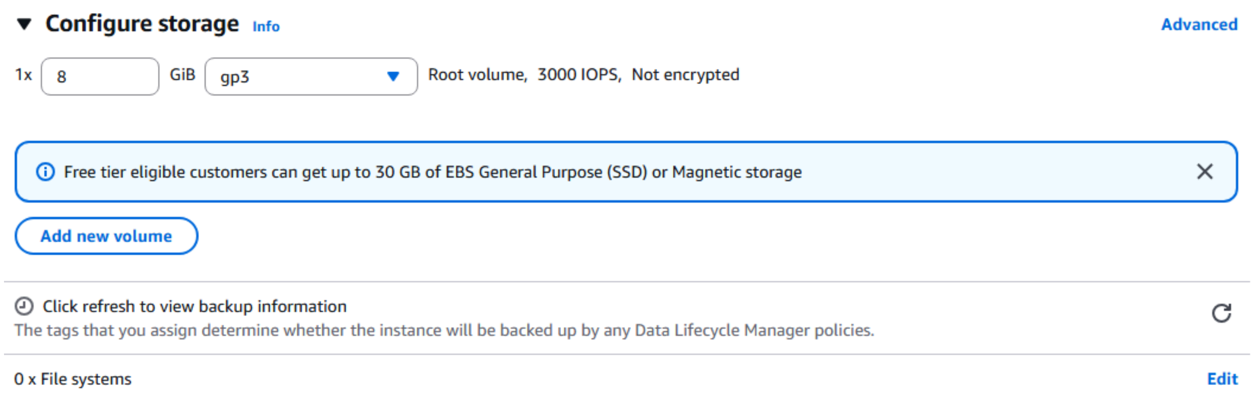
- [x] Allow SSH traffic from — Anywhere 0.0.0.0/0
  Helps you connect to your instance
- [ ] Allow HTTPS traffic from the internet
  To set up an endpoint, for example when creating a web server
- [x] Allow HTTP traffic from the internet
  To set up an endpoint, for example when creating a web server

⚠ 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.

- 0.0.0.0 is accessible from anywhere
- If you want to use from your own PC, you can give YOUR IP address

Configure storage

💡 **This is your C drive**


▼ Configure storage Info

Advanced

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

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

Add new volume

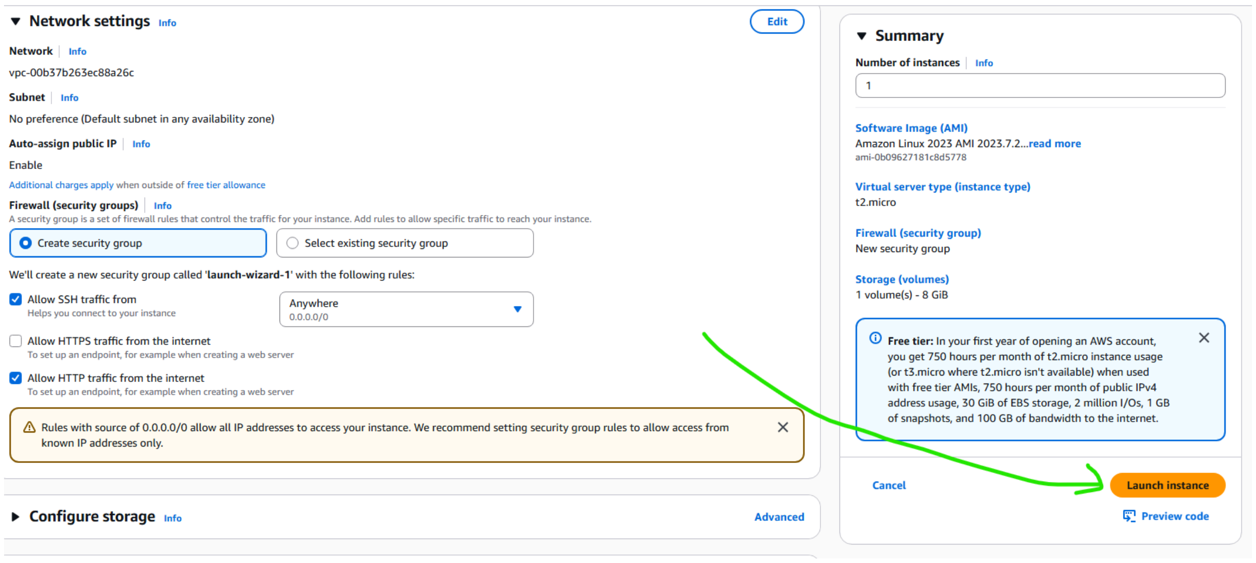
Click refresh to view backup information

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

0 x File systems

Edit

Launch Instance



Network Info
vpc-00b37b263ec88a26c

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 — Anywhere 0.0.0.0/0
  Helps you connect to your instance
- ☐ 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.

Summary

Number of instances Info
1

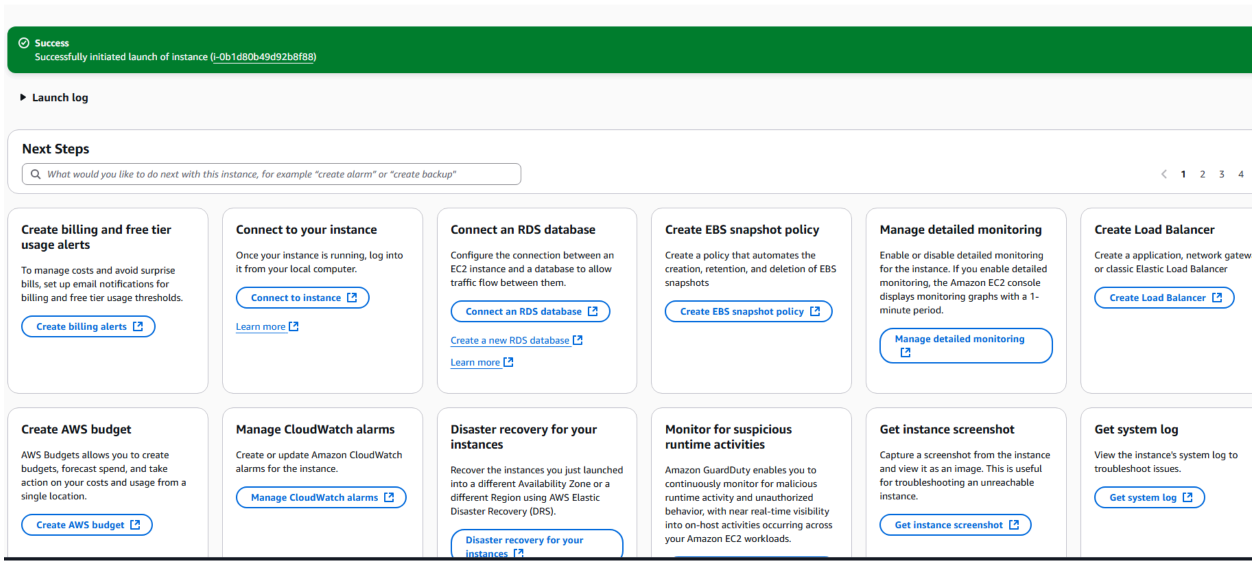
Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-0b09627181c8d5778

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 of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.


Success
Successfully initiated launch of instance (i-0b1d80b49d92b8f88)

Next Steps
What would you like to do next with this instance, for example "create alarm" or "create backup"

Create billing and free tier usage alerts
To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.
Create billing alerts

Connect to your instance
Once your instance is running, log into it from your local computer.
Connect to instance
Learn more

Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.
Connect an RDS database
Create a new RDS database
Learn more

Create EBS snapshot policy
Create a policy that automates the creation, retention, and deletion of EBS snapshots
Create EBS snapshot policy

Manage detailed monitoring
Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period.
Manage detailed monitoring

Create Load Balancer
Create a application, network gatew
or classic Elastic Load Balancer
Create Load Balancer

Create AWS budget
AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location.
Create AWS budget

Manage CloudWatch alarms
Create or update Amazon CloudWatch alarms for the instance.
Manage CloudWatch alarms

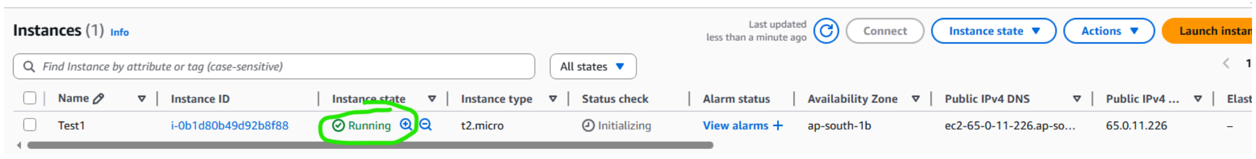
Disaster recovery for your instances
Recover the instances you just launched into a different Availability Zone or a different Region using AWS Elastic Disaster Recovery (DRS).
Disaster recovery for your instances

Monitor for suspicious runtime activities
Amazon GuardDuty enables you to continuously monitor for malicious runtime activity and unauthorized behavior, with near real-time visibility into on-host activities occurring across your Amazon EC2 workloads.

Get instance screenshot
Capture a screenshot from the instance and view it as an image. This is useful for troubleshooting an unreachable instance.
Get instance screenshot

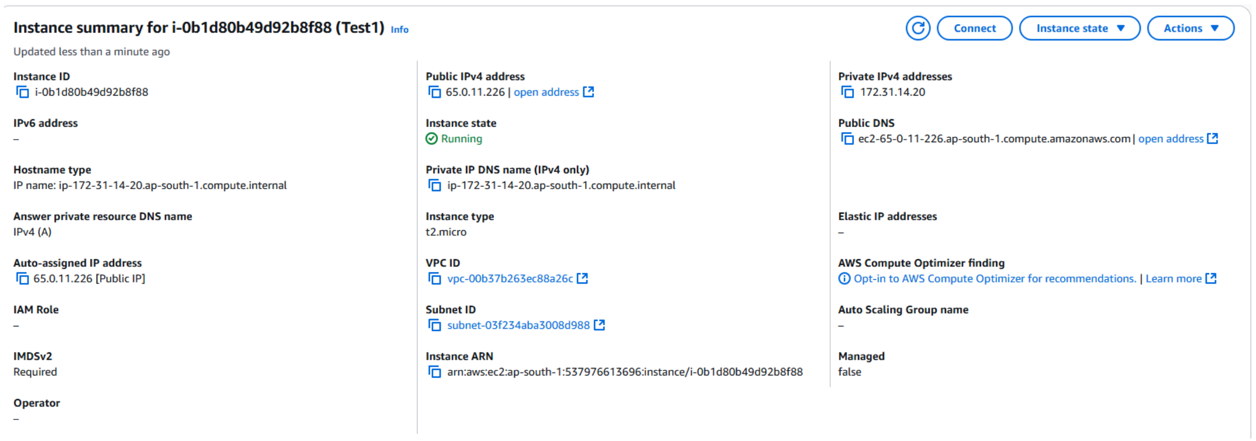
Get system log
View the instance's system log to troubleshoot issues.
Get system log

The instance is running


Instances (1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elast
Test1	i-0b1d80b49d92b8f88	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-65-0-11-226.ap-so...	65.0.11.226	–

Click on the ID to get detailed info


Instance summary for i-0b1d80b49d92b8f88 (Test1) Info

Updated less than a minute ago

Instance ID
i-0b1d80b49d92b8f88

IPv6 address
–

Hostname type
IP name: ip-172-31-14-20.ap-south-1.compute.internal

Answer private resource DNS name
IPv4 (A)

Auto-assigned IP address
65.0.11.226 [Public IP]

IAM Role
–

IMDSv2
Required

Operator
–

Public IPv4 address
65.0.11.226 | open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-14-20.ap-south-1.compute.internal

Instance type
t2.micro

VPC ID
vpc-00b37b263ec88a26c

Subnet ID
subnet-03f234aba3008d988

Instance ARN
arn:aws:ec2:ap-south-1:537976613696:instance/i-0b1d80b49d92b8f88

Private IPv4 addresses
172.31.14.20

Public DNS
ec2-65-0-11-226.ap-south-1.compute.amazonaws.com | open address

Elastic IP addresses
–

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

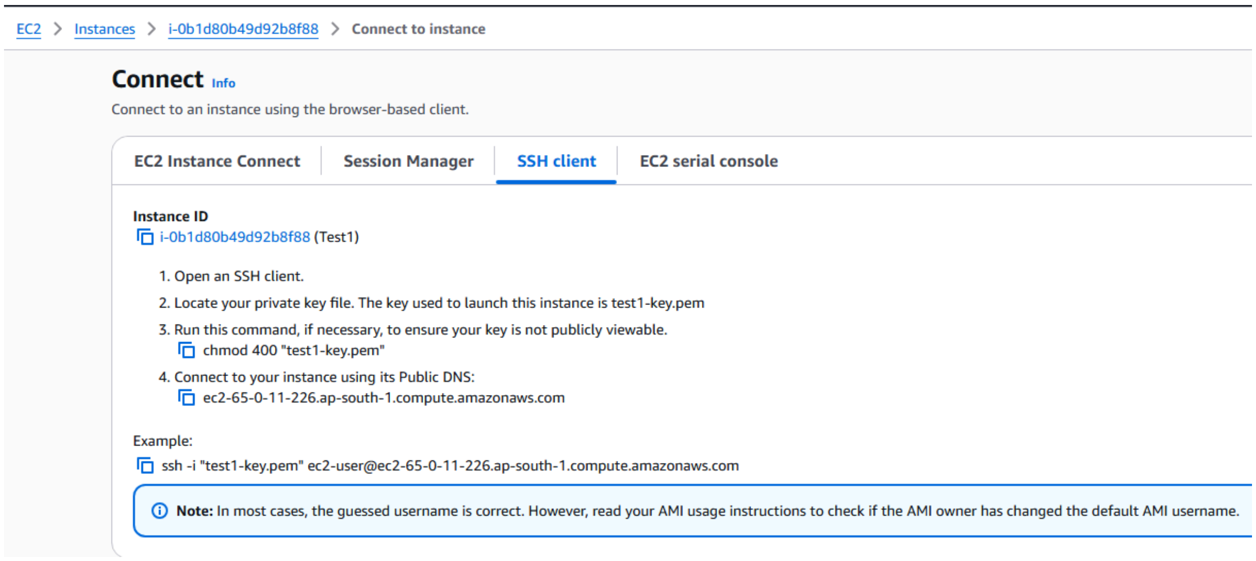
Auto Scaling Group name
–

Managed
false

SSH connection to your EC2 instance

1. Open `cmd` from download folder (where .pem file is located)

2. Run this command


EC2 > Instances > i-0b1d80b49d92b8f88 > Connect to instance

Connect Info

Connect to an instance using the browser-based client.

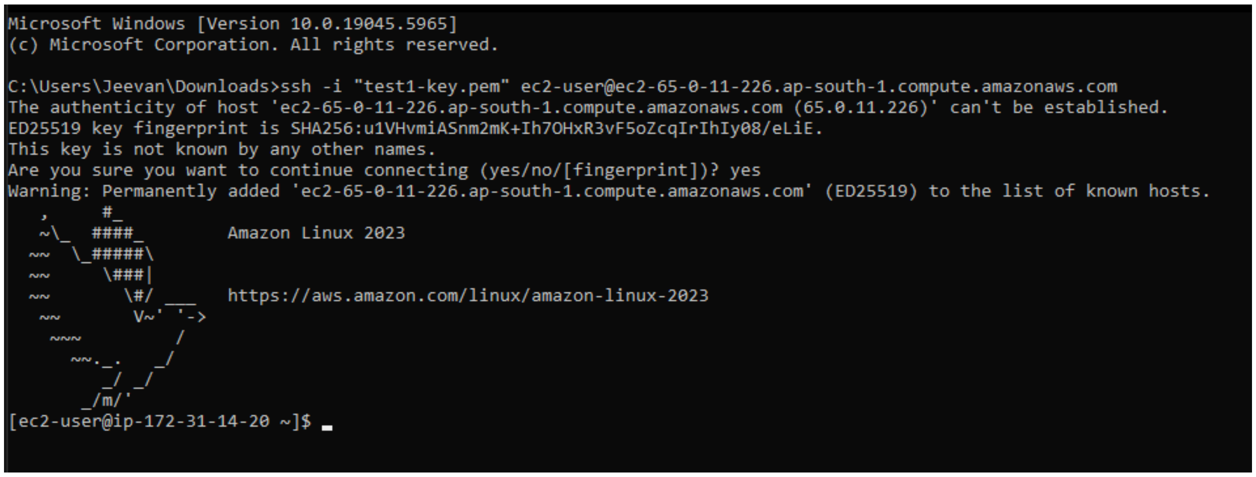
Instance ID
i-0b1d80b49d92b8f88 (Test1)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is test1-key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   `chmod 400 "test1-key.pem"`
4. Connect to your instance using its Public DNS:
   ec2-65-0-11-226.ap-south-1.compute.amazonaws.com

Example:

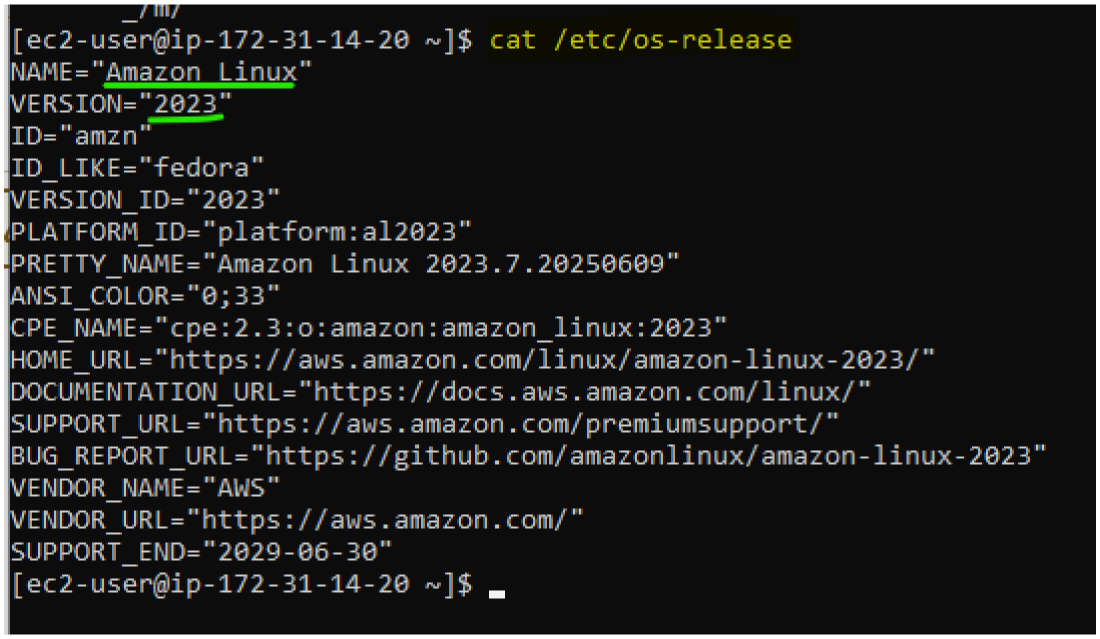
```
ssh -i "test1-key.pem" ec2-user@ec2-65-0-11-226.ap-south-1.compute.amazonaws.com
```

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.



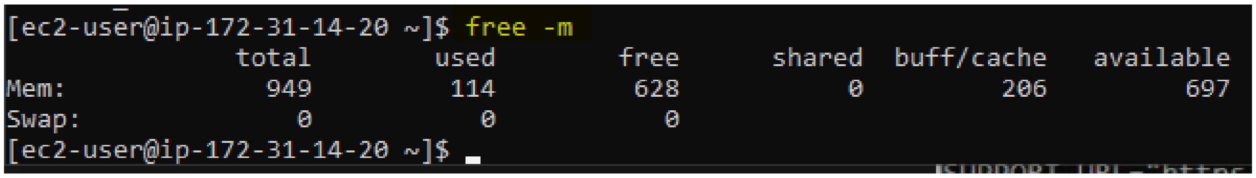
```
Microsoft Windows [Version 10.0.19045.5965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jeevan\Downloads>ssh -i "test1-key.pem" ec2-user@ec2-65-0-11-226.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-65-0-11-226.ap-south-1.compute.amazonaws.com (65.0.11.226)' can't be established.
ED25519 key fingerprint is SHA256:u1VHvmiASnm2mK+Ih7OHxR3vF5oZcqIrIhIy08/eLiE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-65-0-11-226.ap-south-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-172-31-14-20 ~]$ 
```

You have
successfully connected to your EC2 instance! This is a fantastic step. You are now controlling a virtual computer in the AWS cloud from your Windows command prompt.



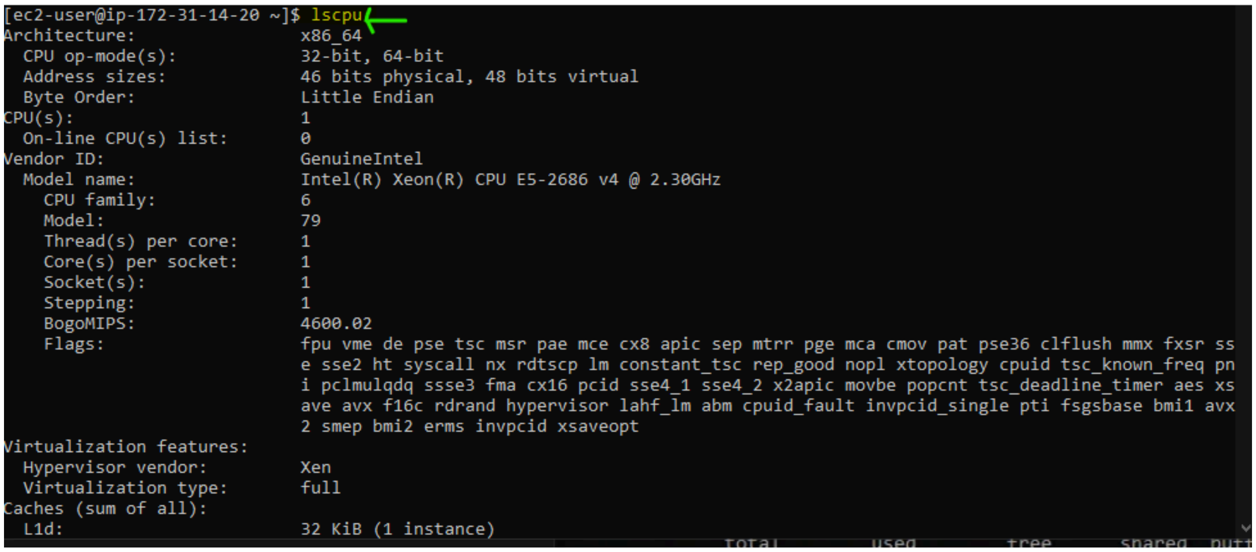
```
[ec2-user@ip-172-31-14-20 ~]$ cat /etc/os-release
NAME="Amazon Linux"
VERSION="2023"
ID="amzn"
ID_LIKE="fedora"
VERSION_ID="2023"
PLATFORM_ID="platform:al2023"
PRETTY_NAME="Amazon Linux 2023.7.20250609"
ANSI_COLOR="0;33"
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2023"
HOME_URL="https://aws.amazon.com/linux/amazon-linux-2023/"
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="2029-06-30"
[ec2-user@ip-172-31-14-20 ~]$ 
```

RAM:



```
[ec2-user@ip-172-31-14-20 ~]$ free -m
               total        used        free      shared  buff/cache   available
Mem:             949         114         628           0         206         697
Swap:              0           0           0
[ec2-user@ip-172-31-14-20 ~]$ _
```

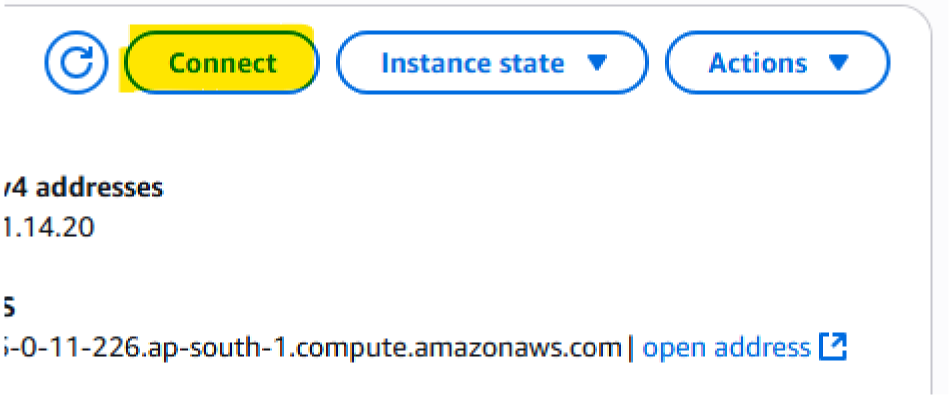
CPU:



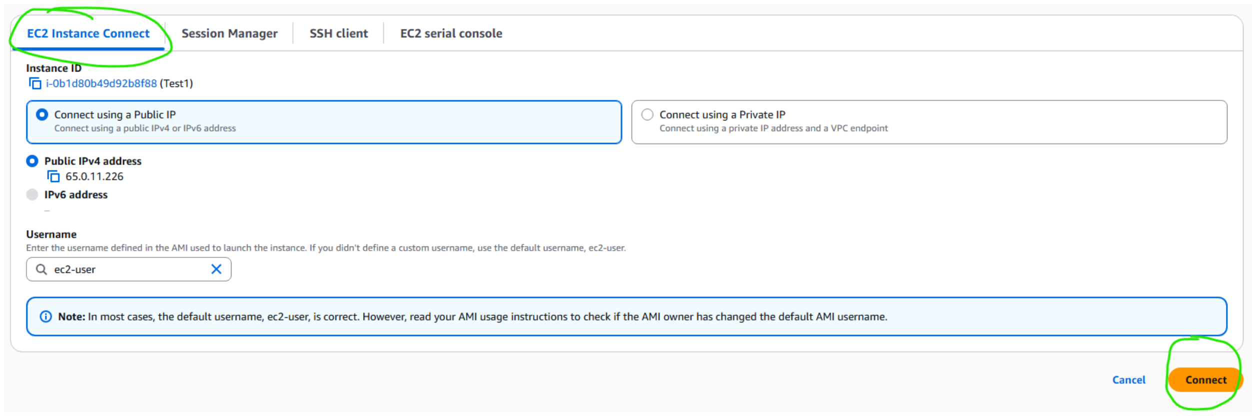
```
[ec2-user@ip-172-31-14-20 ~]$ lscpu
Architecture:            x86_64
  CPU op-mode(s):        32-bit, 64-bit
  Address sizes:         46 bits physical, 48 bits virtual
  Byte Order:            Little Endian
CPU(s):                  1
  On-line CPU(s) list:   0
Vendor ID:               GenuineIntel
  Model name:            Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz
    CPU family:          6
    Model:               79
    Thread(s) per core:  1
    Core(s) per socket:  1
    Socket(s):           1
    Stepping:            1
    BogoMIPS:            4600.02
    Flags:               fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr ss
                         e sse2 ht syscall nx rdtscp lm constant_tsc rep_good nopl xtopology cpuid tsc_known_freq pn
                         i pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xs
                         ave avx f16c rdrand hypervisor lahf_lm abm cpuid_fault invpcid_single pti fsgsbase bmi1 avx
                         2 smep bmi2 erms invpcid xsaveopt
Virtualization features:
  Hypervisor vendor:     Xen
  Virtualization type:   full
Caches (sum of all):
  L1d:                   32 KiB (1 instance)
```

Connect SSH from AWS website

- Click connect



```
/4 addresses
1.14.20

5
5-0-11-226.ap-south-1.compute.amazonaws.com | open address
```



Instance ID
i-0b1d80b49d92b8f88 (Test1)

● Connect using a Public IP
Connect using a public IPv4 or IPv6 address

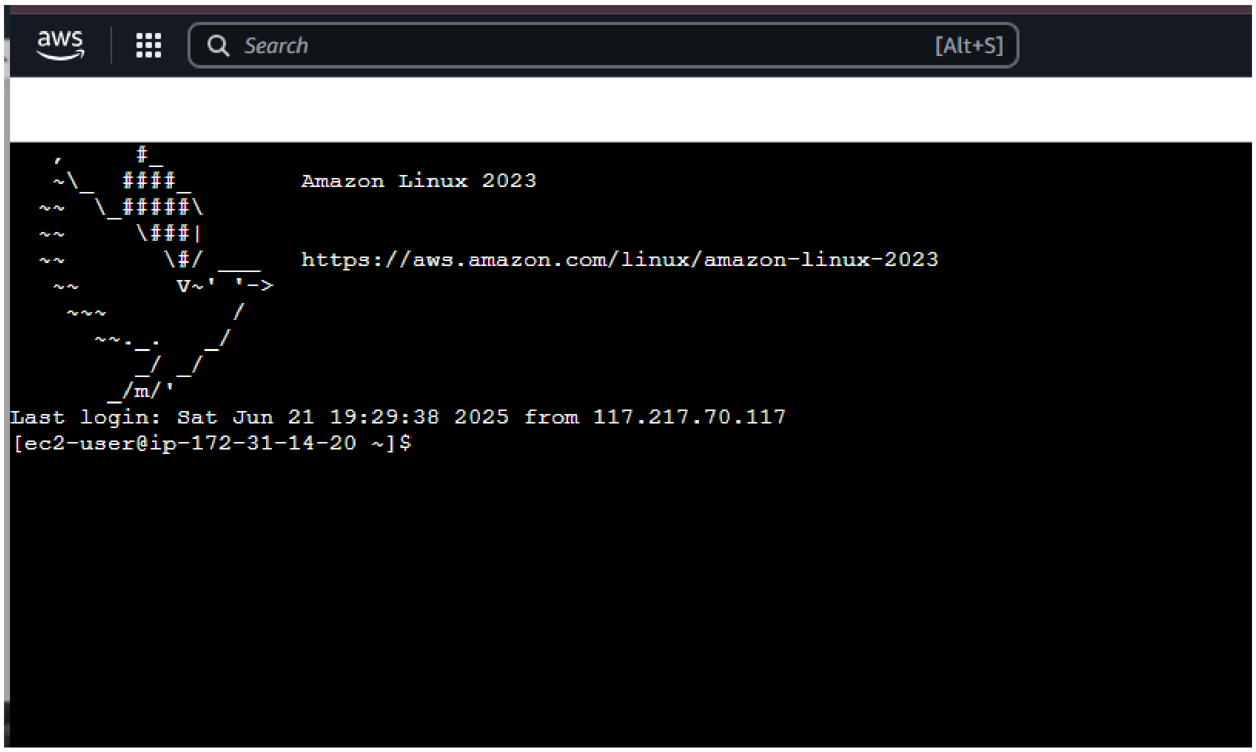
○ Connect using a Private IP
Connect using a private IP address and a VPC endpoint

● Public IPv4 address
65.0.11.226

○ IPv6 address
–

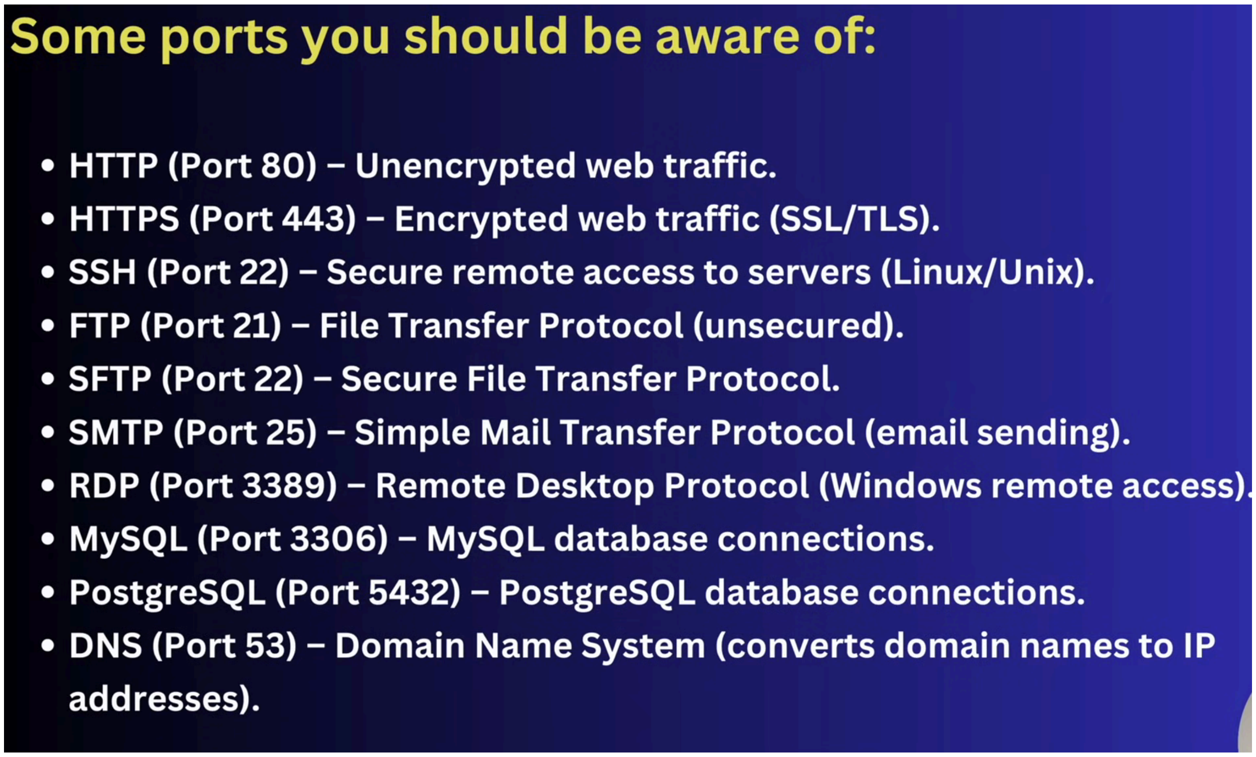
Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.
ec2-user

Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sat Jun 21 19:29:38 2025 from 117.217.70.117
[ec2-user@ip-172-31-14-20 ~]$
```

Security Groups



Some ports you should be aware of:

- HTTP (Port 80) – Unencrypted web traffic.
- HTTPS (Port 443) – Encrypted web traffic (SSL/TLS).
- SSH (Port 22) – Secure remote access to servers (Linux/Unix).
- FTP (Port 21) – File Transfer Protocol (unsecured).
- SFTP (Port 22) – Secure File Transfer Protocol.
- SMTP (Port 25) – Simple Mail Transfer Protocol (email sending).
- RDP (Port 3389) – Remote Desktop Protocol (Windows remote access).
- MySQL (Port 3306) – MySQL database connections.
- PostgreSQL (Port 5432) – PostgreSQL database connections.
- DNS (Port 53) – Domain Name System (converts domain names to IP addresses).

Host HTML/CSS Website

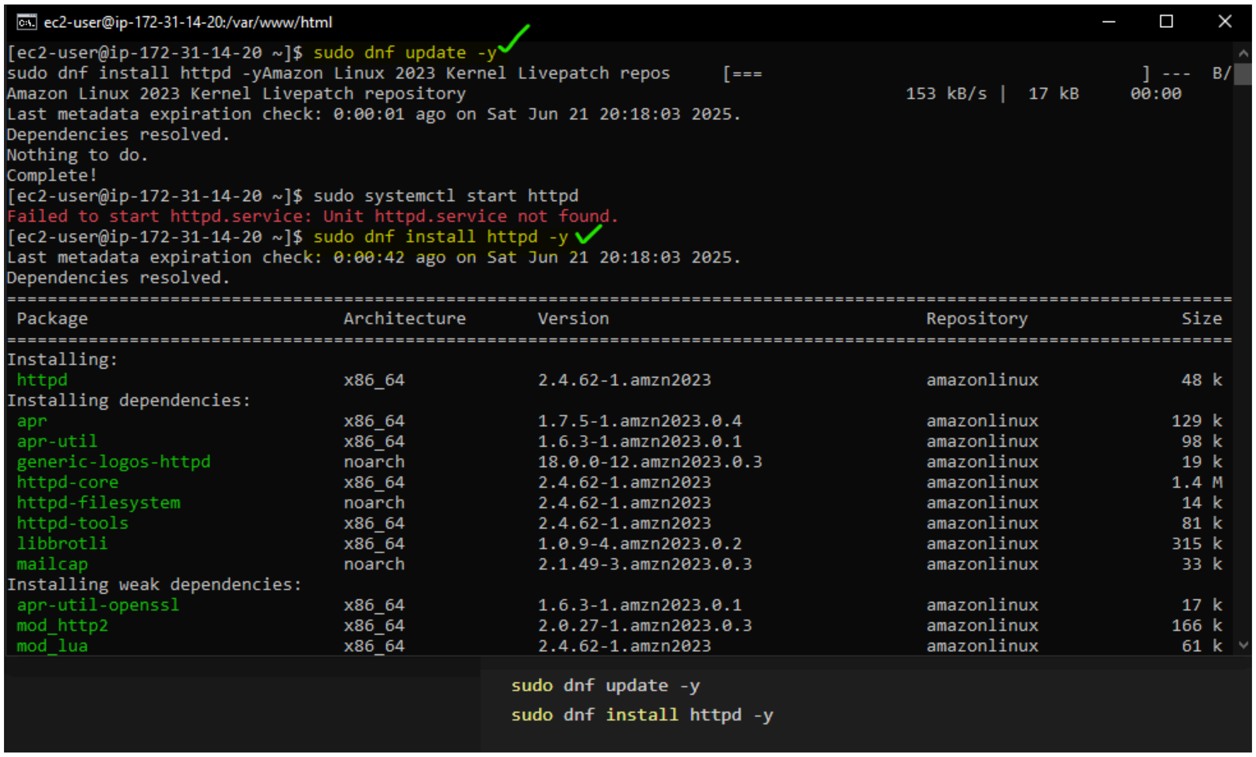
💡 Run the code line by line. Do not run 2 lines at same time

Connect to Your EC2 (SSH):

```
ssh -i your-key.pem ec2-user@your-ec2-public-ip
```

Install Apache:

```
sudo dnf update -y
sudo dnf install httpd -y
```

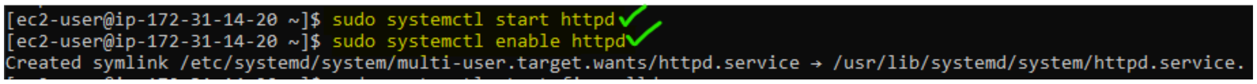


```
[ec2-user@ip-172-31-14-20 ~]$ sudo dnf update -y
sudo dnf install httpd -yAmazon Linux 2023 Kernel Livepatch repos     [===                                    ] ---  B/
Amazon Linux 2023 Kernel Livepatch repository                                    153 kB/s |  17 kB     00:00
Last metadata expiration check: 0:00:01 ago on Sat Jun 21 20:18:03 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-14-20 ~]$ sudo systemctl start httpd
Failed to start httpd.service: Unit httpd.service not found.
[ec2-user@ip-172-31-14-20 ~]$ sudo dnf install httpd -y
Last metadata expiration check: 0:00:42 ago on Sat Jun 21 20:18:03 2025.
Dependencies resolved.
========================================================================================================================
 Package                        Architecture        Version                               Repository               Size
========================================================================================================================
Installing:
 httpd                          x86_64              2.4.62-1.amzn2023                     amazonlinux              48 k
Installing dependencies:
 apr                            x86_64              1.7.5-1.amzn2023.0.4                  amazonlinux             129 k
 apr-util                       x86_64              1.6.3-1.amzn2023.0.1                  amazonlinux              98 k
 generic-logos-httpd            noarch              18.0.0-12.amzn2023.0.3                amazonlinux              19 k
 httpd-core                     x86_64              2.4.62-1.amzn2023                     amazonlinux             1.4 M
 httpd-filesystem               noarch              2.4.62-1.amzn2023                     amazonlinux              14 k
 httpd-tools                    x86_64              2.4.62-1.amzn2023                     amazonlinux              81 k
 libbrotli                      x86_64              1.0.9-4.amzn2023.0.2                  amazonlinux             315 k
 mailcap                        noarch              2.1.49-3.amzn2023.0.3                 amazonlinux              33 k
Installing weak dependencies:
 apr-util-openssl               x86_64              1.6.3-1.amzn2023.0.1                  amazonlinux              17 k
 mod_http2                      x86_64              2.0.27-1.amzn2023.0.3                 amazonlinux             166 k
 mod_lua                        x86_64              2.4.62-1.amzn2023                     amazonlinux              61 k
```

```
sudo dnf update -y
sudo dnf install httpd -y
```

Start the server:

```
sudo systemctl start httpd
sudo systemctl enable httpd
```

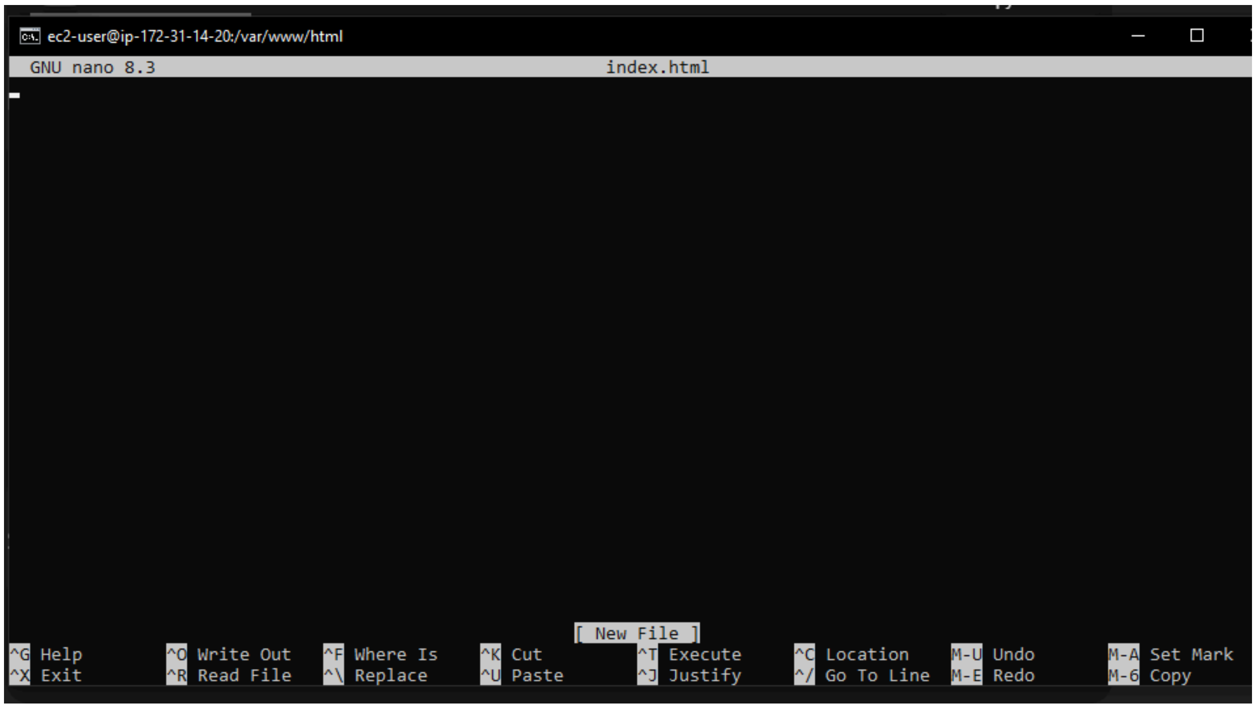


```
[ec2-user@ip-172-31-14-20 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-14-20 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
```

Create the HTML Page:

```
cd /var/www/html
sudo nano index.html
```

Paste your HTML content:



```
ec2-user@ip-172-31-14-20:/var/www/html
 GNU nano 8.3                          index.html

                                    [ New File ]
^G Help       ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy
```

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>My First AWS Website!</title>
   <style>
      body {
         font-family: Arial, sans-serif;
         background-color: #f0f8ff;
         color: #333;
         text-align: center;
         padding: 50px;
      }
      h1 {
         color: #4CAF50;
```

```
        }
        p {
            font-size: 1.2em;
        }
        .container {
            background-color: #ffffff;
            border-radius: 10px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            padding: 30px;
            max-width: 600px;
            margin: 20px auto;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Hello from AWS EC2!</h1>
        <p>This is my very first website, hosted on an Amazon Linux 2023 instan
ce.</p>
        <p>AWS is awesome for Data Science and Web Hosting!</p>
        <p>Date and Time: <span id="datetime"></span></p>
    </div>

    <script>
        function updateDateTime() {
            const now = new Date();
            document.getElementById('datetime').textContent = now.toLocaleStrin
g();
        }
        setInterval(updateDateTime, 1000); // Update every second
        updateDateTime(); // Initial call
    </script>
</body>
</html>
```

Ctrl +O → Save & exit

To Exit `nano` (Save and Exit, or Discard Changes and Exit):

Look at the bottom of your `nano` screen. It shows you the common commands. The `^` symbol means `Ctrl`.

1. **To Save and Exit:**
    - Press `Ctrl + O` (Ctrl + O for "Write Out" or Save).
        - Nano will then ask you "File Name to Write: index.html" (or whatever file you are editing). Just press `Enter` to confirm the current file name.
    - Press `Ctrl + X` (Ctrl + X for "Exit").
2. **To Discard Changes and Exit:**
    - Press `Ctrl + X` (Ctrl + X for "Exit").
    - Nano will then ask you "Save modified buffer (y/n/d)?"
        - Press `N` (for "No") if you do NOT want to save the changes you've made.
        - Press `Y` (for "Yes") if you want to save. (This will then prompt for the filename as in step 1).
        - Press `D` (for "Discard") to discard without asking to save.

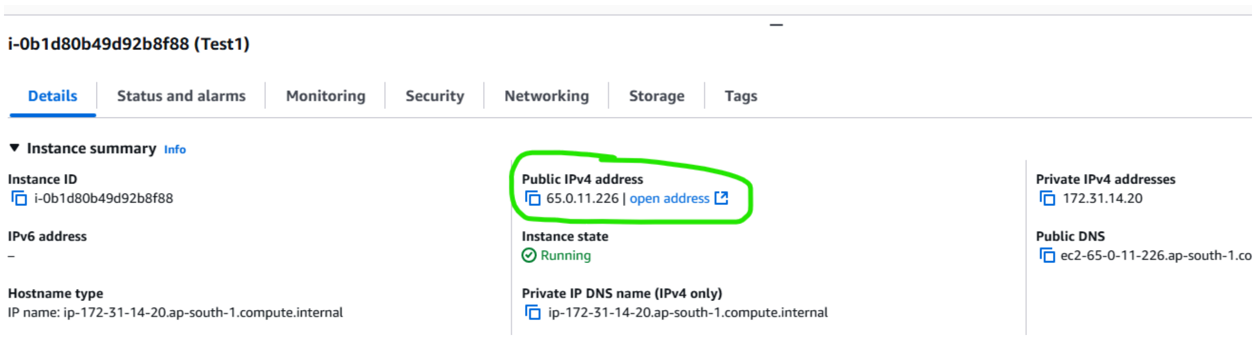
So, the commands would be:

1. **Press** **`Ctrl + O`**
2. **Press** **`Enter`** (to confirm saving to `index.html`)
3. **Press** **`Ctrl + X`**

```
sudo systemctl restart httpd
```

View Website:

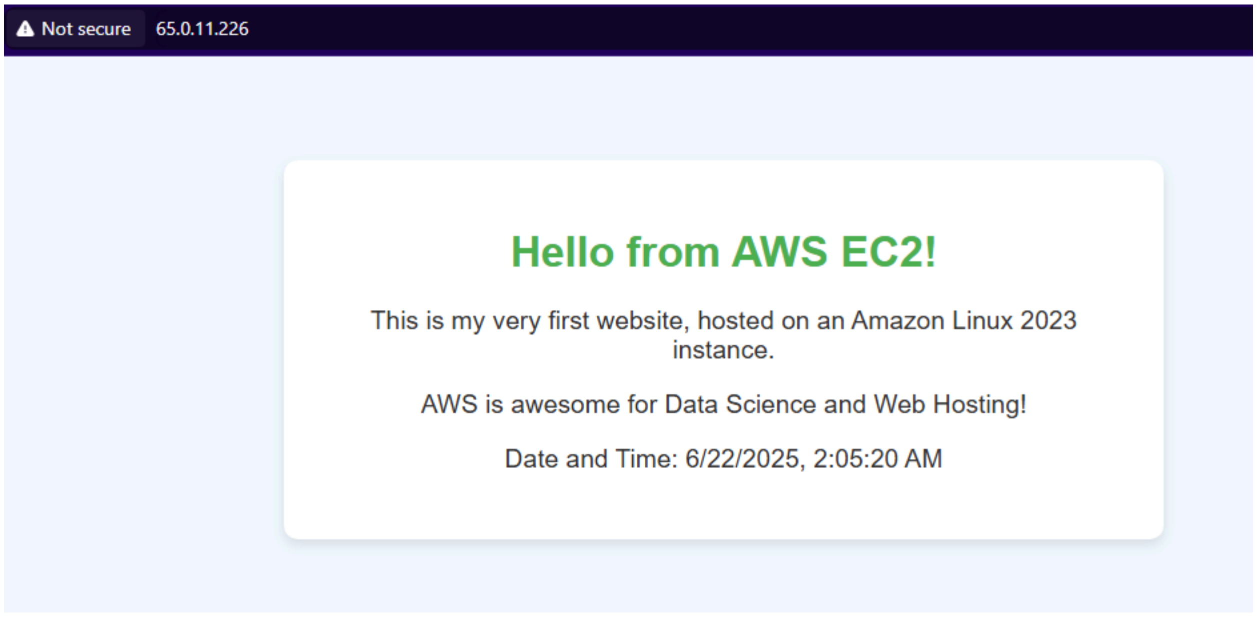
Copy & paste the public IP



i-0b1d80b49d92b8f88 (Test1)

▼ Instance summary Info

Instance ID i-0b1d80b49d92b8f88	Public IPv4 address 65.0.11.226 | open address	Private IPv4 addresses 172.31.14.20
IPv6 address –	Instance state Running	Public DNS ec2-65-0-11-226.ap-south-1.co
Hostname type IP name: ip-172-31-14-20.ap-south-1.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-14-20.ap-south-1.compute.internal	



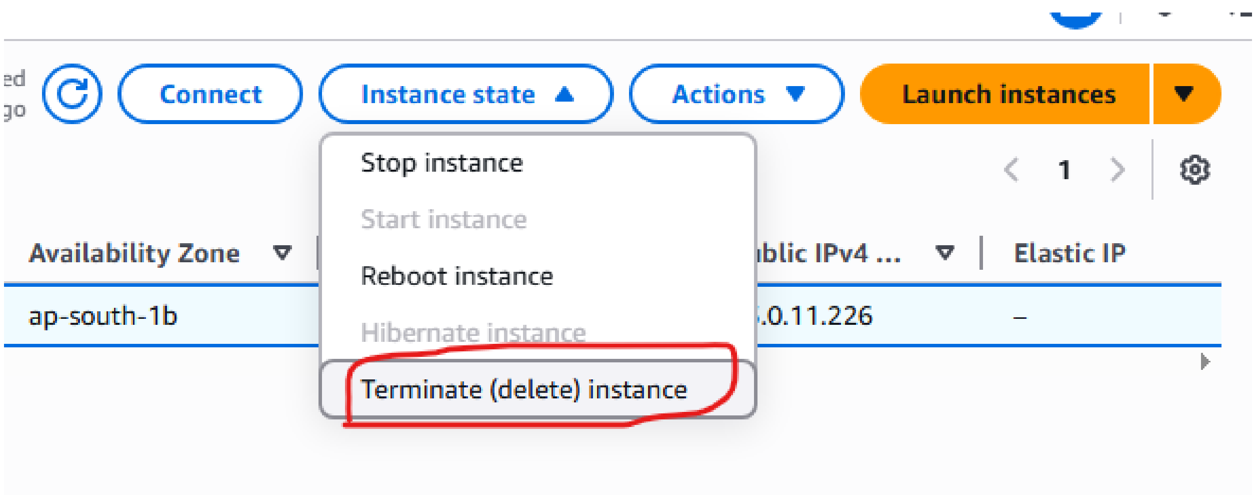
Hello from AWS EC2!

This is my very first website, hosted on an Amazon Linux 2023 instance.

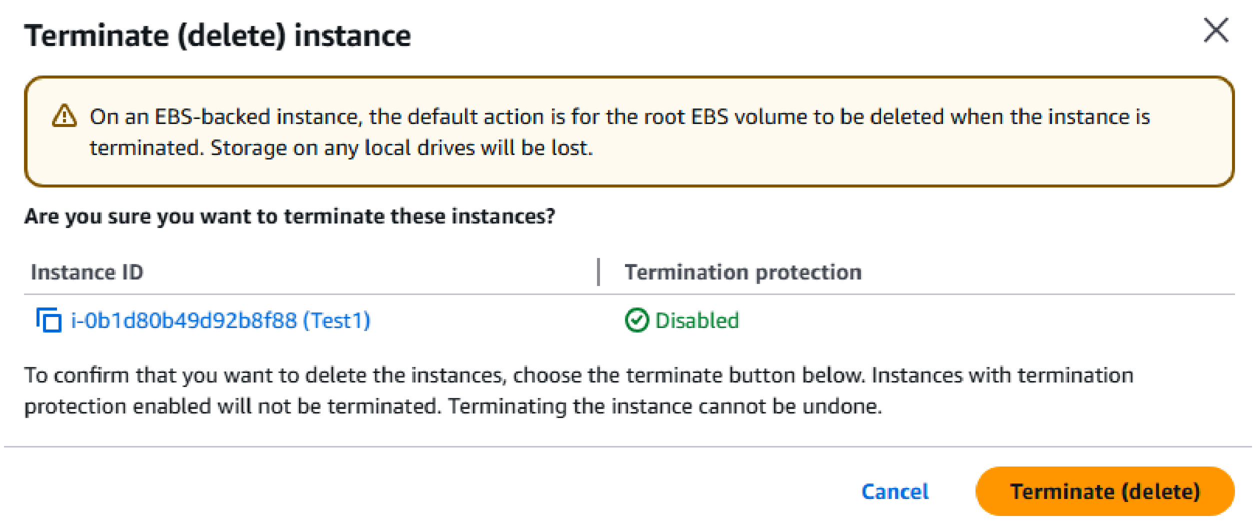
AWS is awesome for Data Science and Web Hosting!

Date and Time: 6/22/2025, 2:05:20 AM

Terminate Instance



💡 **Do not just stop**



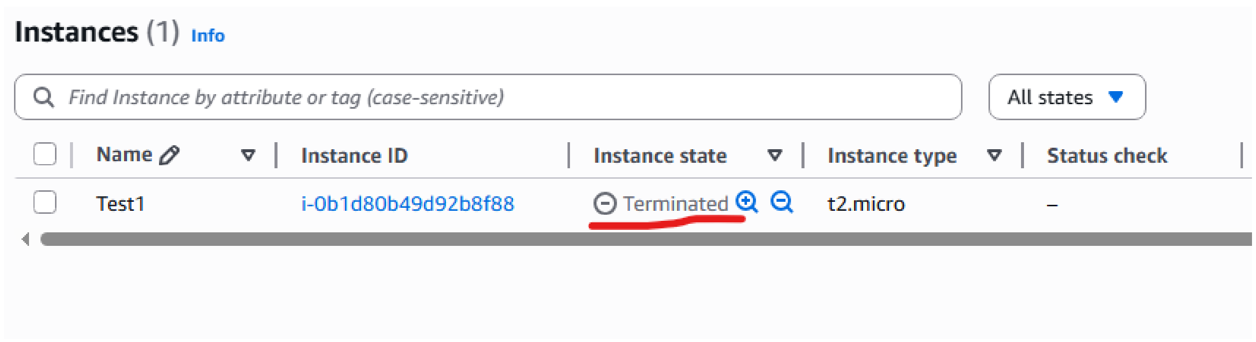
Terminate (delete) instance

⚠ On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

Are you sure you want to terminate these instances?

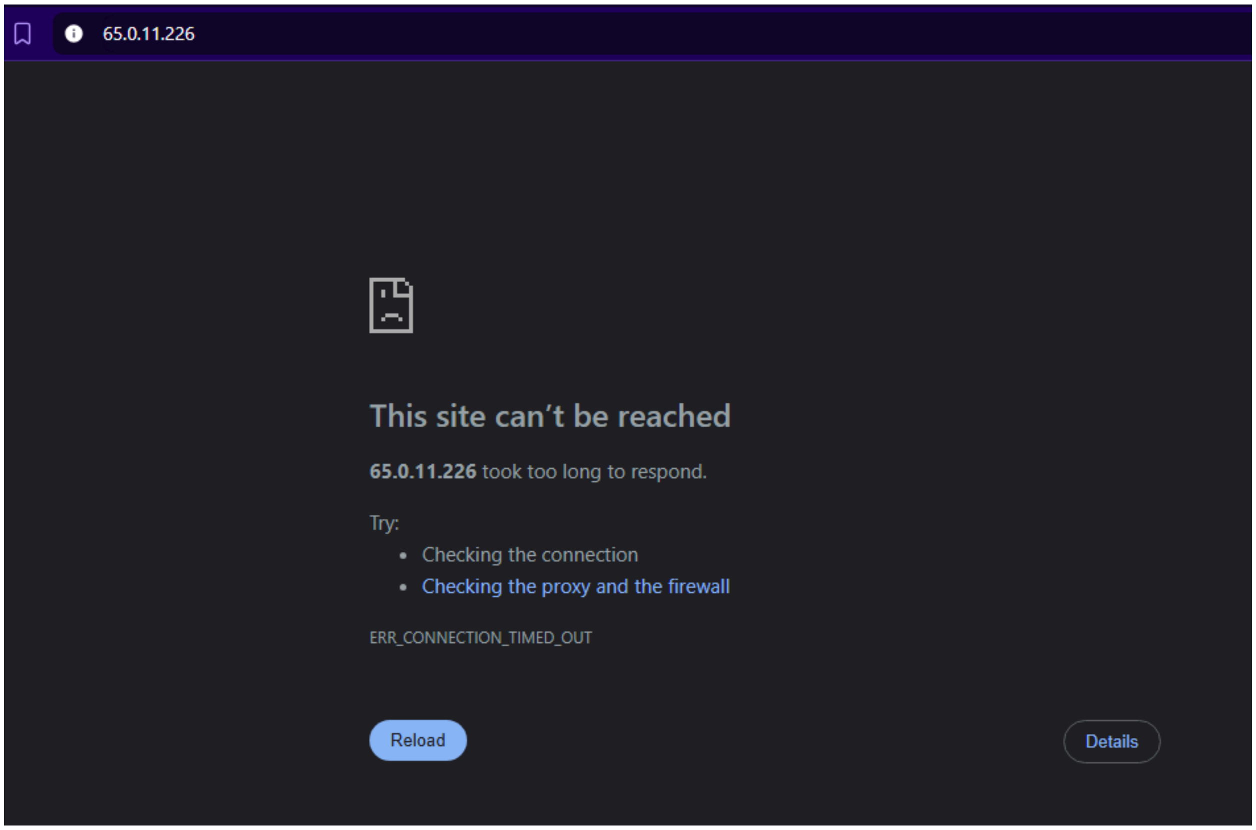
Instance ID	Termination protection
i-0b1d80b49d92b8f88 (Test1)	Disabled

To confirm that you want to delete the instances, choose the terminate button below. Instances with termination protection enabled will not be terminated. Terminating the instance cannot be undone.



Instances (1) Info

Name	Instance ID	Instance state	Instance type	Status check
Test1	i-0b1d80b49d92b8f88	Terminated	t2.micro	–


65.0.11.226

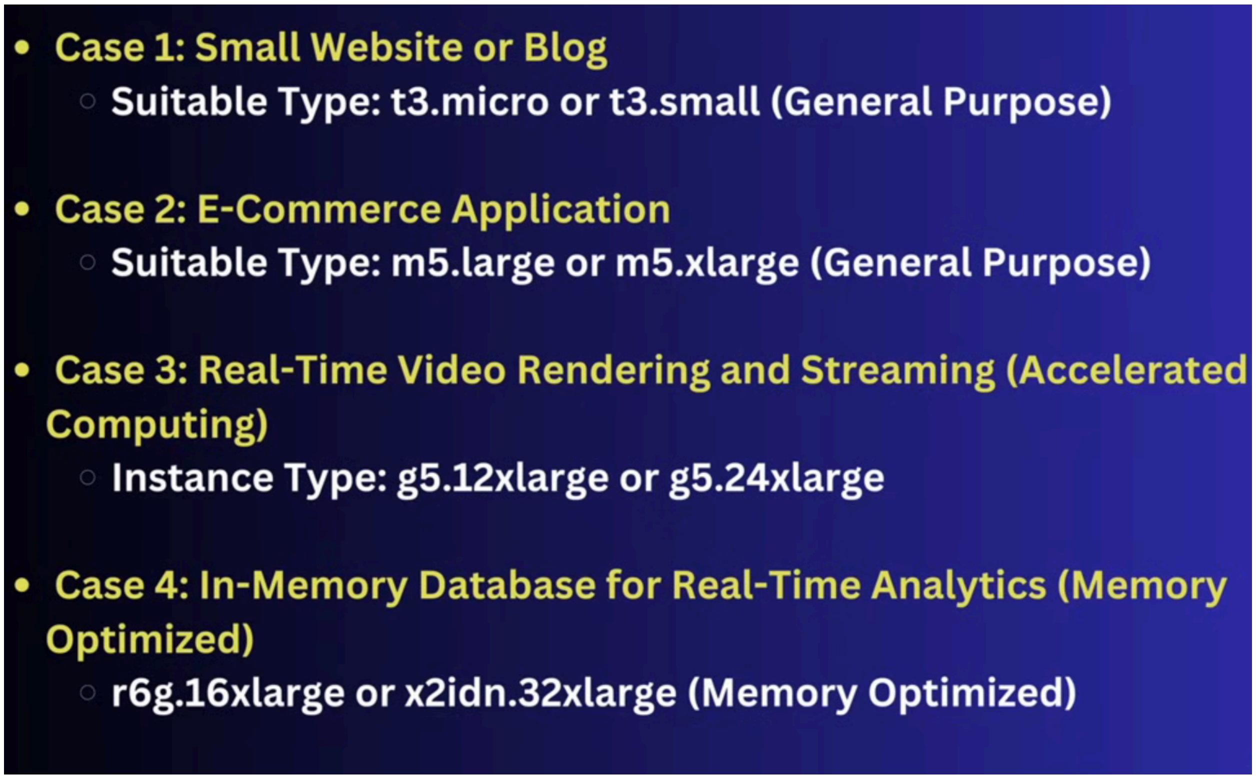
This site can't be reached

65.0.11.226 took too long to respond.

Try:
- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_TIMED_OUT

Instance Types:



- **Case 1: Small Website or Blog**
  - **Suitable Type: t3.micro or t3.small (General Purpose)**
- **Case 2: E-Commerce Application**
  - **Suitable Type: m5.large or m5.xlarge (General Purpose)**
- **Case 3: Real-Time Video Rendering and Streaming (Accelerated Computing)**
  - **Instance Type: g5.12xlarge or g5.24xlarge**
- **Case 4: In-Memory Database for Real-Time Analytics (Memory Optimized)**
  - **r6g.16xlarge or x2idn.32xlarge (Memory Optimized)**