Hands-On Guide to AWS EC2 Instances and Web Server Setup

Introduction to EC2

EC2 stands for **Elastic Compute Cloud**, a core service of AWS that allows you to create virtual servers (instances) running Windows or Linux OS. It offers the ability to scale up or down instances as needed.

- EC2 is **region-specific** — instances created in one region (e.g., Mumbai) won't be visible in another region. Common use cases include hosting websites, running applications, and testing environments.

Steps to Create a New EC2 Instance

1. Name and Tags

- Tags act as labels to identify the instance.
- Example: Name = Linux-machine1

2. Select AMI (Amazon Machine Image)

- AMI is a predefined OS template.
- Examples: Red Hat Linux, SUSE Linux, Windows Server 2016.

3. Select Instance Type

- Choose the CPU and RAM configuration.
- Default is usually 1 CPU, 1 GiB RAM.

4. Select Key Pair

- Used for secure SSH (Linux) or RDP (Windows) connection.
- .pem file for Linux; .ppk file for PuTTY connection.

5. Network Settings

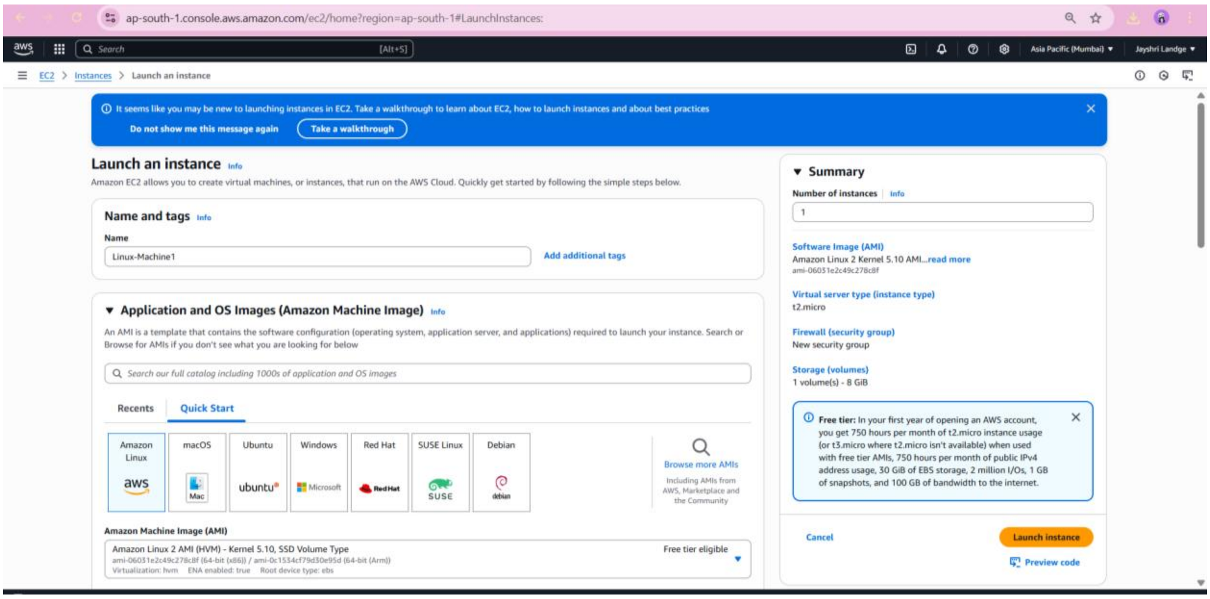
- Assign Security Groups (firewall rules).
- Default open ports: 22 for Linux, 3389 for Windows.

6. Configure Storage

- Default storage: 8 GiB for Linux, 30 GiB for Windows.

7. Configure Instance

- Specify the number of instances to launch.



ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:

Search [Alt+S] | Asia Pacific (Mumbai) ▾ | Jayshri Landge ▾

EC2 > Instances > Launch an instance

ⓘ It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices

Do not show me this message again | Take a walkthrough

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: Linux-Machine1 | 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

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian | Browse more AMIs — Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type — Free tier eligible
ami-06031e2c49c278c8f (64-bit (x86)) / ami-0c1534cf79d30e95d (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

▼ Summary

Number of instances | Info: 1

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more
ami-06031e2c49c278c8f

Virtual server type (instance type): t2.micro

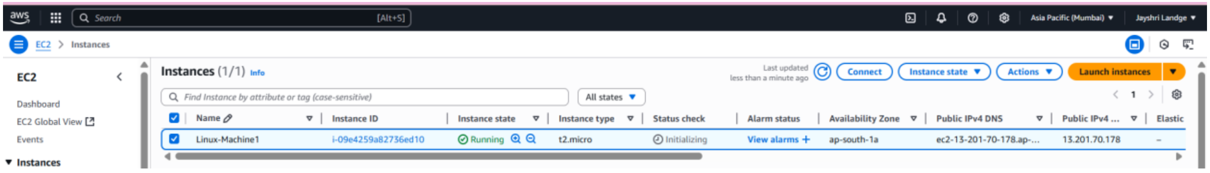
Firewall (security group): New security group

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.

Cancel | Launch instance | Preview code

Screenshot 1: EC2 Creation Step - Name and Tags



Search [Alt+S] | Asia Pacific (Mumbai) ▾ | Jayshri Landge ▾

EC2 > Instances

EC2: Dashboard, EC2 Global View, Events, ▼ Instances

Instances (1/1) Info — Last updated less than a minute ago | Connect | Instance state ▾ | Actions ▾ | Launch instances ▾

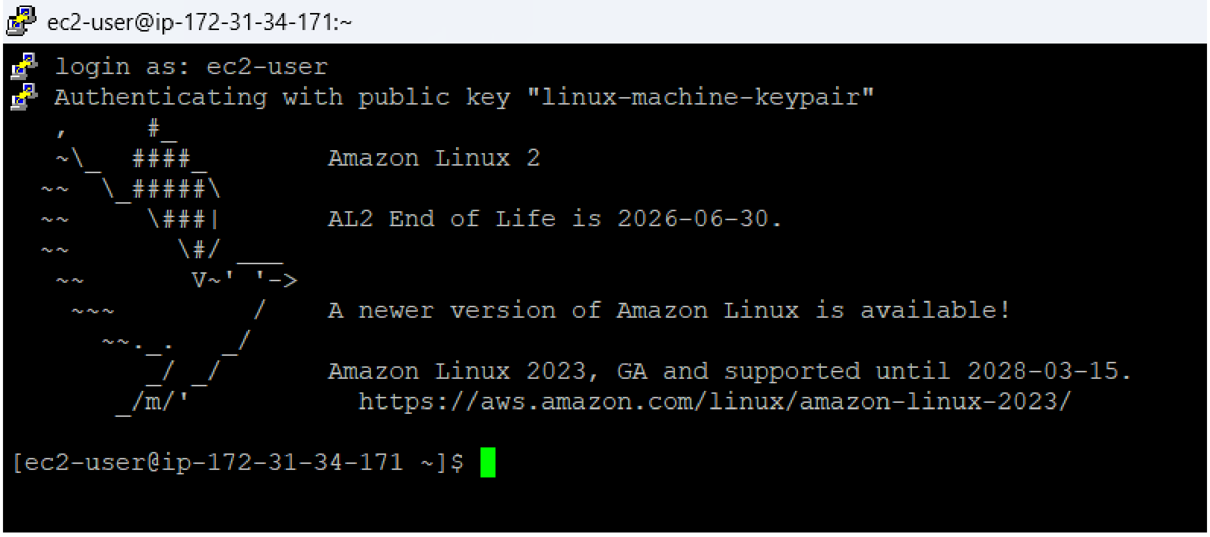
Find Instance by attribute or tag (case-sensitive) | All states ▾

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic
☑	Linux-Machine1	i-09e4259a82736ed10	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-13-201-70-178.ap-...	13.201.70.178	–

Screenshot 2: Show the running instance dashboard

Connecting to Your Instance

- **Linux**: Use **PuTTY** with .ppk file for SSH connection.
- **Windows**: Use **Remote Desktop Connection** with .pem file to connect.

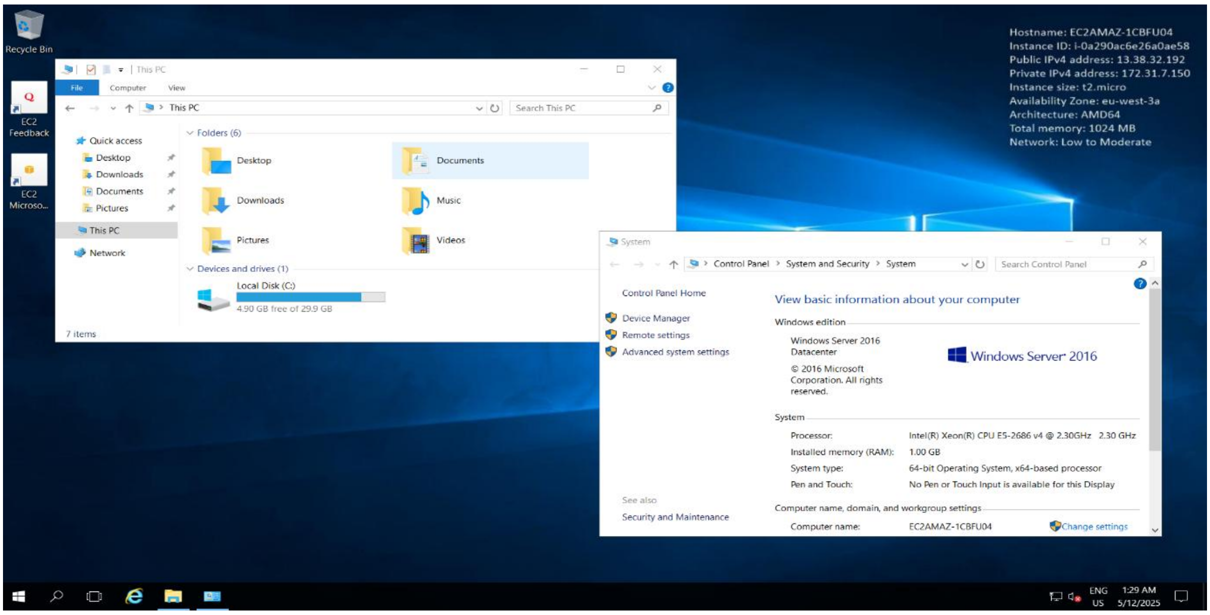


ec2-user@ip-172-31-34-171:~

```
login as: ec2-user
Authenticating with public key "linux-machine-keypair"
       ,     #_
       ~\_  ####_        Amazon Linux 2
      ~~  \_#####\
      ~~     \###|       AL2 End of Life is 2026-06-30.
      ~~       \#/ ___
       ~~       V~' '->
        ~~~         /    A newer version of Amazon Linux is available!
          ~~._.   _/
             _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
           _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-34-171 ~]$
```

Screenshot 3: PuTTY



Hostname: EC2AMAZ-1CBFU04
Instance ID: i-0a290ac6e26a0ae58
Public IPv4 address: 13.38.32.192
Private IPv4 address: 172.31.7.150
Instance size: t2.micro
Availability Zone: eu-west-3a
Architecture: AMD64
Total memory: 1024 MB
Network: Low to Moderate

View basic information about your computer

Windows edition

Windows Server 2016 Datacenter

© 2016 Microsoft Corporation. All rights reserved.

System

Processor: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz 2.30 GHz
Installed memory (RAM): 1.00 GB
System type: 64-bit Operating System, x64-based processor
Pen and Touch: No Pen or Touch Input is available for this Display

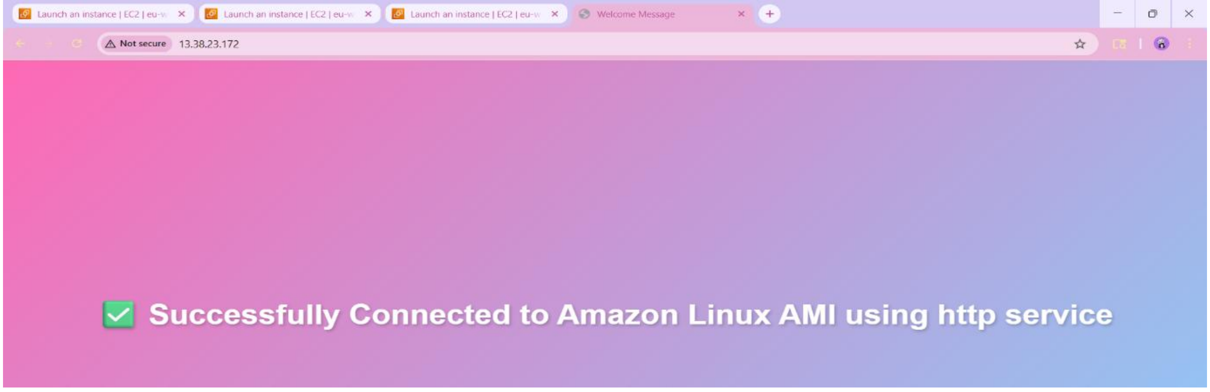
Computer name, domain, and workgroup settings

Computer name: EC2AMAZ-1CBFU04

Screenshot 4: Remote Desktop Connection

Creating a Web Server on EC2 (Linux Example)

1. Install httpd package:
2. yum install httpd -y
3. Start the httpd service:
4. systemctl start httpd
5. Create a simple webpage:
6. vim /var/www/html/index.html
7. Allow port 80 in Security Group for HTTP access.
8. Access the web page in browser via: http://<Public_IP>



Launch an instance | EC2 | eu-w
Launch an instance | EC2 | eu-w
Launch an instance | EC2 | eu-w
Welcome Message
Not secure
13.38.23.172

✅ Successfully Connected to Amazon Linux AMI using http service

Screenshot 5: Creating a Web Server on EC2

Running Web Server on Custom Port

1. Edit the httpd config file:
2. vi /etc/httpd/conf/httpd.conf

Change the port number (e.g., 90).

3. Restart httpd:
4. systemctl restart httpd
5. Open custom port (90) in Security Group.
6. Access via:
   http://<Public_IP>:90

Bootstrap Script for Automation

Use bootstrap scripts to automate instance setup during launch:

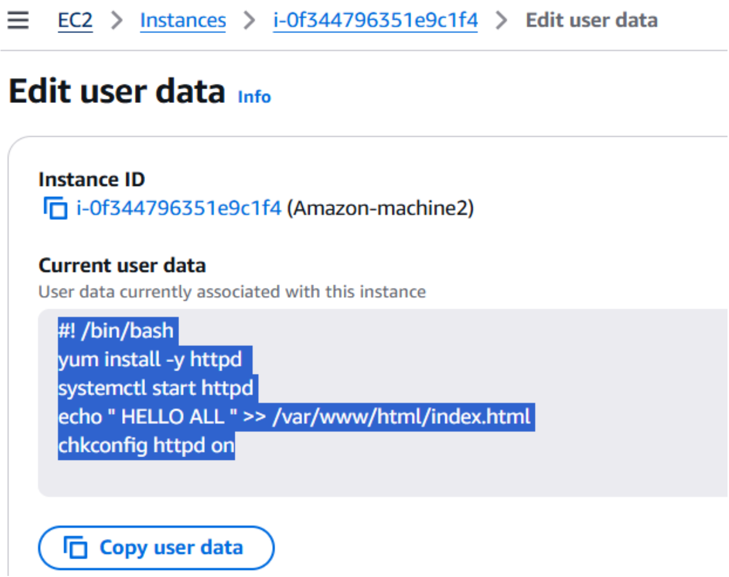
#! /bin/bash

yum install httpd -y

systemctl start httpd

chkconfig httpd on

echo "Hello All" > /var/www/html/index.html



Instance ID

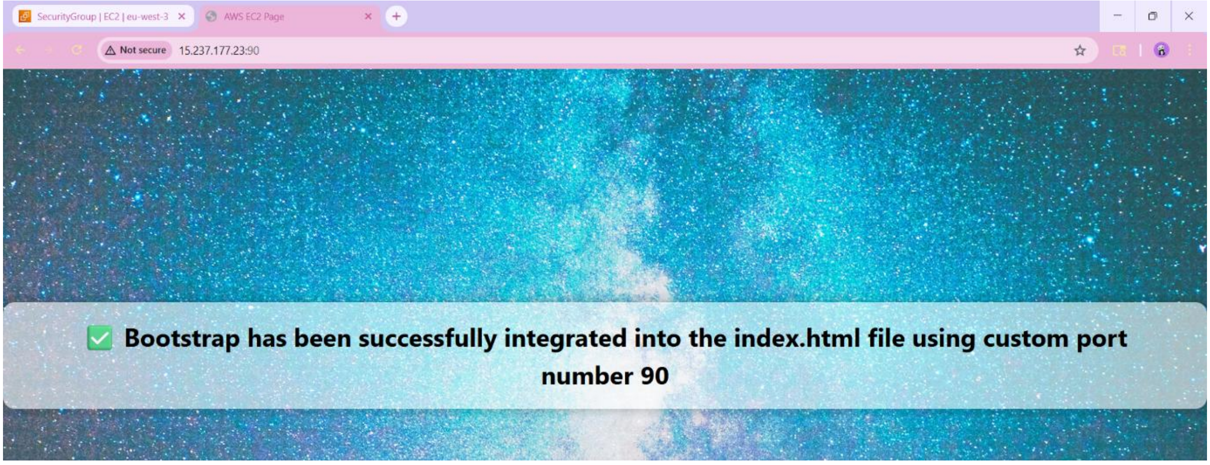
i-0f344796351e9c1f4 (Amazon-machine2)

Current user data

User data currently associated with this instance

```
#! /bin/bash
yum install -y httpd
systemctl start httpd
echo " HELLO ALL " >> /var/www/html/index.html
chkconfig httpd on
```

Screenshot 6: Bootstrap Script in EC2 User Data

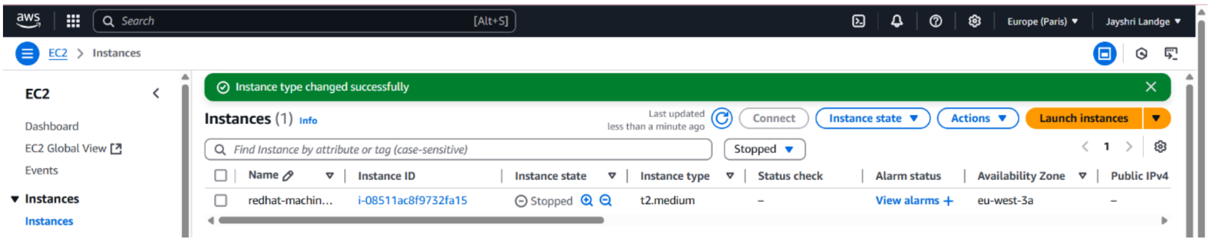


✅ Bootstrap has been successfully integrated into the index.html file using custom port number 90

Screenshot 7: Custom Web Server Running on Port 90

Managing EC2 Instances

- **Change Instance Type:**
  - Instance must be stopped.
  - Go to Actions > Instance Settings > Change Instance Type.
- **Elastic IP:**
  - Static public IP that remains fixed after instance restart.
  - Allocate, Associate, Disassociate, and Release options available.



Instance type changed successfully

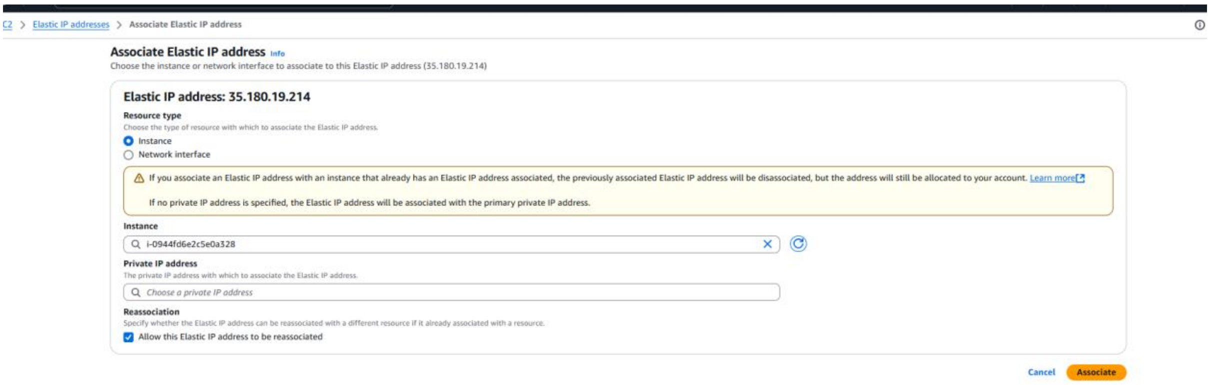
Instances (1) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
□	redhat-machin...	i-08511ac8f9732fa15	Stopped	t2.medium	–	View alarms +	eu-west-3a	–

Screenshot 8: Changing Instance Type

Is Elastic IP free or paid?

- If we allocate an Elastic IP from AWS and do not attach it to a server, we need to pay for it. In this case, we can say that the Elastic IP is paid.
- If we allocate an Elastic IP from AWS and attach it to a server, we do not need to pay for it. In this case, we can say that the Elastic IP is free.



Associate Elastic IP address Info

Choose the instance or network interface to associate to this Elastic IP address (35.180.19.214)

Elastic IP address: 35.180.19.214

Resource type

Choose the type of resource with which to associate the Elastic IP address.

● Instance

○ Network interface

⚠ If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account. Learn more

If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.

Instance

i-0944fd6e2c5e0a328

Private IP address

The private IP address with which to associate the Elastic IP address.

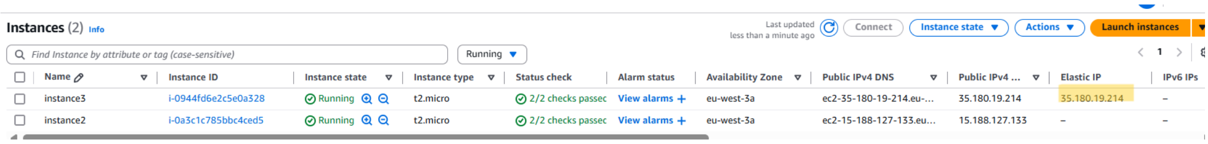
Choose a private IP address

Reassociation

Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.

☑ Allow this Elastic IP address to be reassociated

Screenshot 9: Elastic IP Reassociation with EC2



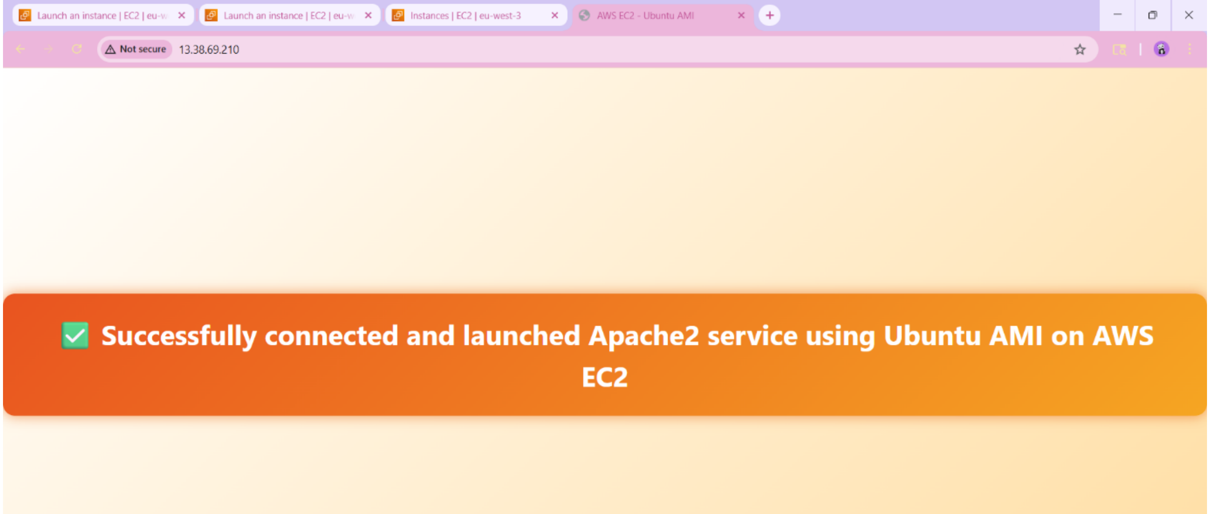
Instances (2) Info
Last updated less than a minute ago
Connect
Instance state ▼
Actions ▼
Launch instances
Find Instance by attribute or tag (case-sensitive)
Running ▼
1

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs
□	instance3	i-0944fd6e2c5e0a328	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	eu-west-3a	ec2-35-180-19-214.eu-...	35.180.19.214	35.180.19.214	–
□	instance2	i-0a3c1c785bbc4ced5	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	eu-west-3a	ec2-15-188-127-133.eu...	15.188.127.133	–	–

Screenshot 10: Elastic IP Allocation

Apache Web Server on Ubuntu EC2

- Install Apache:
- sudo apt-get update -y
- sudo apt-get install apache2 -y
- sudo systemctl start apache2
- Create index.html and access via browser.



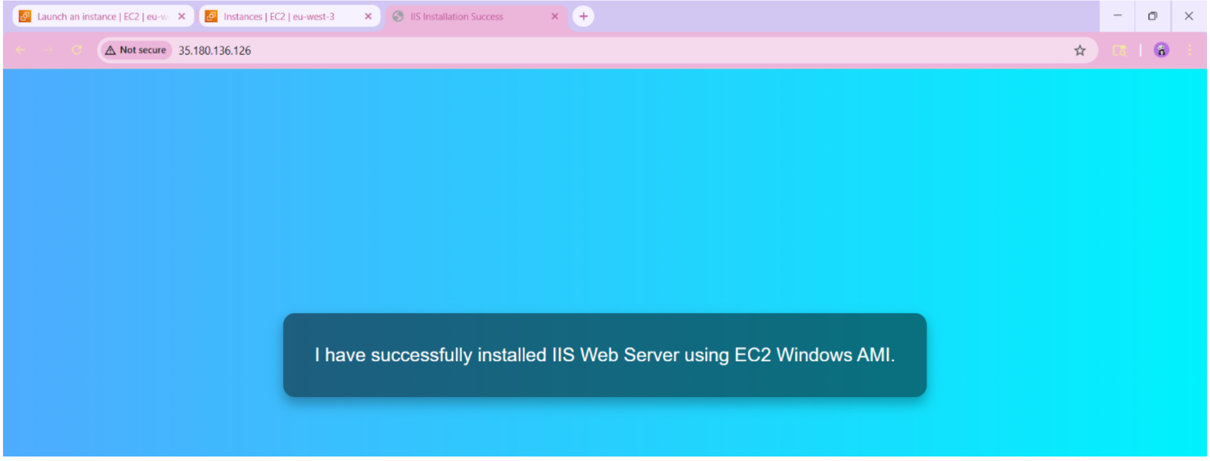
Launch an instance | EC2 | eu-w
Launch an instance | EC2 | eu-w
Instances | EC2 | eu-west-3
AWS EC2 - Ubuntu AMI
Not secure
13.38.69.210

✅ Successfully connected and launched Apache2 service using Ubuntu AMI on AWS EC2

Screenshot 11: Web Server Installation on Ubuntu

IIS Web Server on Windows EC2

1. Use Server Manager > Add Roles and Features > Web Server (IIS).
2. Install and add content to C:\inetpub\wwwroot\index.html.
3. Open port 80 in Security Group.
4. Access via browser using instance Public IP.

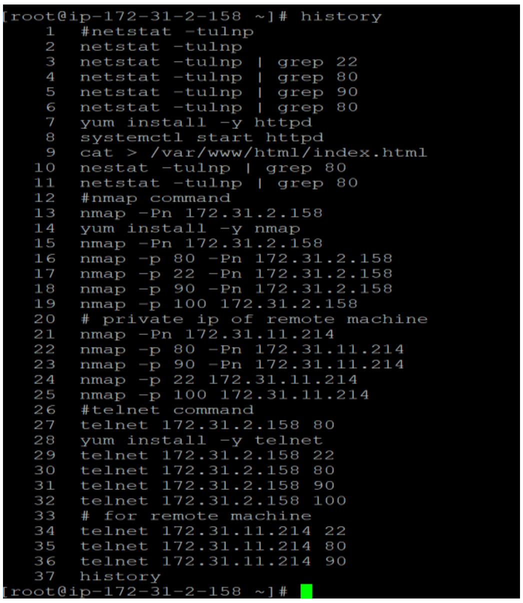


I have successfully installed IIS Web Server using EC2 Windows AMI.

Screenshot 12: IIS Web Server Installation on Windows EC2

Useful Commands

- **netstat** – to check open ports:- netstat -tulnp
- **nmap** – to scan open ports:- nmap -Pn <IP_Address>
- **telnet** – to test port connectivity:- telnet <IP_Address> <Port>



```
[root@ip-172-31-2-158 ~]# history
    1  #netstat -tulnp
    2  netstat -tulnp
    3  netstat -tulnp | grep 22
    4  netstat -tulnp | grep 80
    5  netstat -tulnp | grep 90
    6  netstat -tulnp | grep 80
    7  yum install -y httpd
    8  systemctl start httpd
    9  cat > /var/www/html/index.html
   10  nestat -tulnp | grep 80
   11  netstat -tulnp | grep 80
   12  #nmap command
   13  nmap -Pn 172.31.2.158
   14  yum install -y nmap
   15  nmap -Pn 172.31.2.158
   16  nmap -p 80 -Pn 172.31.2.158
   17  nmap -p 22 -Pn 172.31.2.158
   18  nmap -p 90 -Pn 172.31.2.158
   19  nmap -p 100 172.31.2.158
   20  # private ip of remote machine
   21  nmap -Pn 172.31.11.214
   22  nmap -p 80 -Pn 172.31.11.214
   23  nmap -p 90 -Pn 172.31.11.214
   24  nmap -p 22 172.31.11.214
   25  nmap -p 100 172.31.11.214
   26  #telnet command
   27  telnet 172.31.2.158 80
   28  yum install -y telnet
   29  telnet 172.31.2.158 22
   30  telnet 172.31.2.158 80
   31  telnet 172.31.2.158 90
   32  telnet 172.31.2.158 100
   33  # for remote machine
   34  telnet 172.31.11.214 22
   35  telnet 172.31.11.214 80
   36  telnet 172.31.11.214 90
   37  history
[root@ip-172-31-2-158 ~]#
```

Screenshot 13: Command Outputs

Types of AMIs (Amazon Machine Images) in AWS

1. **Quickstart AMIs**

   - Free-tier eligible and ideal for practice or experimentation.
   - Pre-configured and ready to use with minimal setup.
   - Not verified by AWS — not recommended for production use.

2. **AWS Marketplace AMIs**

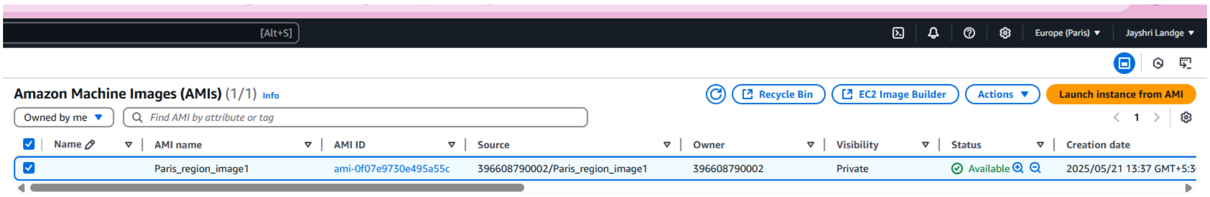
   - Trusted and verified images curated by AWS partners.
   - Available in the AWS Marketplace with user ratings and reviews.
   - Suitable for production workloads with available support and updates.

3. **My AMIs**

   - Custom AMIs that you or other AWS users have created.
     - **Owned by me**: AMIs you personally created from existing instances.
     - **Shared with me:** AMIs shared by other AWS accounts.
   - Useful for deploying consistent configurations across environments.

4. **Community AMIs**

   - Public AMIs created and shared by other AWS users.
   - Free to use but not verified by AWS.
   - May contain untrusted or outdated configurations — use with caution.



Amazon Machine Images (AMIs) (1/1) Info

☑	Name	AMI name	AMI ID	Source	Owner	Visibility	Status	Creation date
☑		Paris_region_image1	ami-0f07e9730e495a55c	396608790002/Paris_region_image1	396608790002	Private	✓ Available	2025/05/21 13:37 GMT+5:3

Screenshot 14: Create My AMI in paris region

Instance Types

- General Purpose (M, T series)
- Compute Optimized (C series)
- Memory Optimized (R series)
- Storage Optimized (I, D series)
- GPU Instances (G, P series)

Types of EC2 Purchasing Options in AWS

1. **On-Demand Instances**
    - Pay-as-you-go with no long-term commitment.
    - Billed by the second or hour, depending on instance type.
    - Ideal for short-term, unpredictable workloads or testing environments.
2. **Spot Instances**
    - Bid for unused EC2 capacity at reduced rates (up to 90% off).
    - Instances can be terminated by AWS when capacity is needed or price exceeds your bid.
    - Best suited for fault-tolerant, flexible applications like batch processing or data analysis.
3. **Reserved Instances**
    - Commit to using instances for a 1- or 3-year term for significant savings (up to 75%).
    - Offers capacity reservation in a specific Availability Zone.
    - Suitable for steady-state workloads or applications with predictable usage.

Conclusion

This hands-on experience with AWS EC2 has helped me understand instance creation, server setup, security, and networking basics in the cloud.
Looking forward to leveraging this knowledge for scalable cloud deployments!