AWS EC2 Deployment with Custom VPC, User Data & AMI Creation

This document explains the complete process of creating AWS infrastructure and deploying an application using EC2 User Data and AMI.

----

1. Create Custom VPC

Step 1: Create VPC

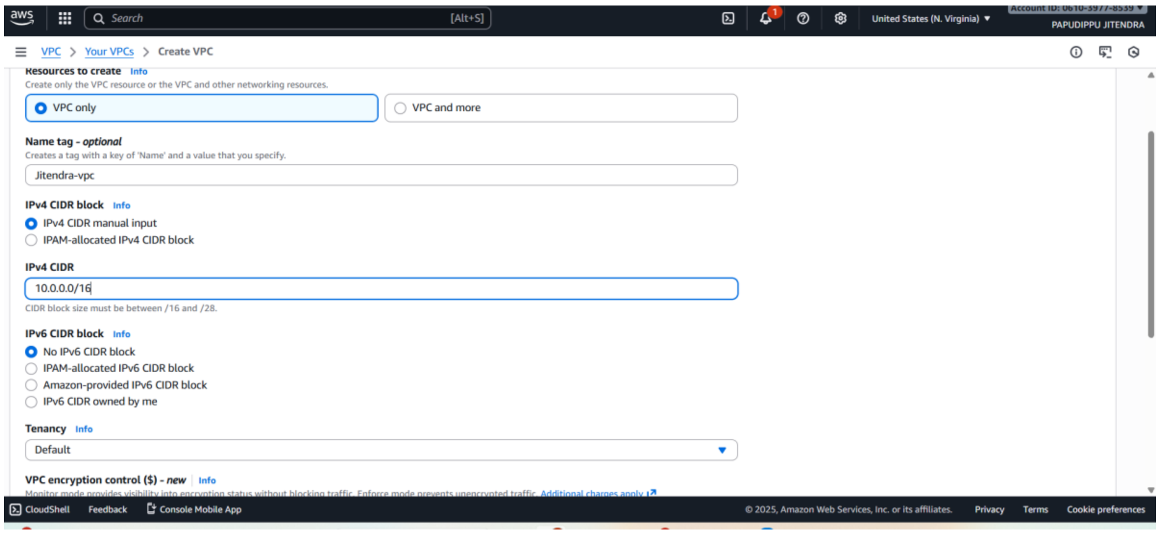
- Go to AWS Console → VPC

- Click Create VPC

- Select VPC only

- CIDR block: 10.0.0.0/16

- Name: Custom-VPC


aws
Search [Alt+S]
United States (N. Virginia)
Account ID: 0610-3977-8539
PAPUDIPPU JITENDRA
VPC > Your VPCs > Create VPC
CloudShell
Feedback
Console Mobile App
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.
[✓ VPC only] VPC and more

Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
Jitendra-vpc

IPv4 CIDR block Info
[✓ IPv4 CIDR manual input]
IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/16
CIDR block size must be between /16 and /28.

IPv6 CIDR block Info
[✓ No IPv6 CIDR block]
IPAM-allocated IPv6 CIDR block
Amazon-provided IPv6 CIDR block
IPv6 CIDR owned by me

Tenancy Info
Default

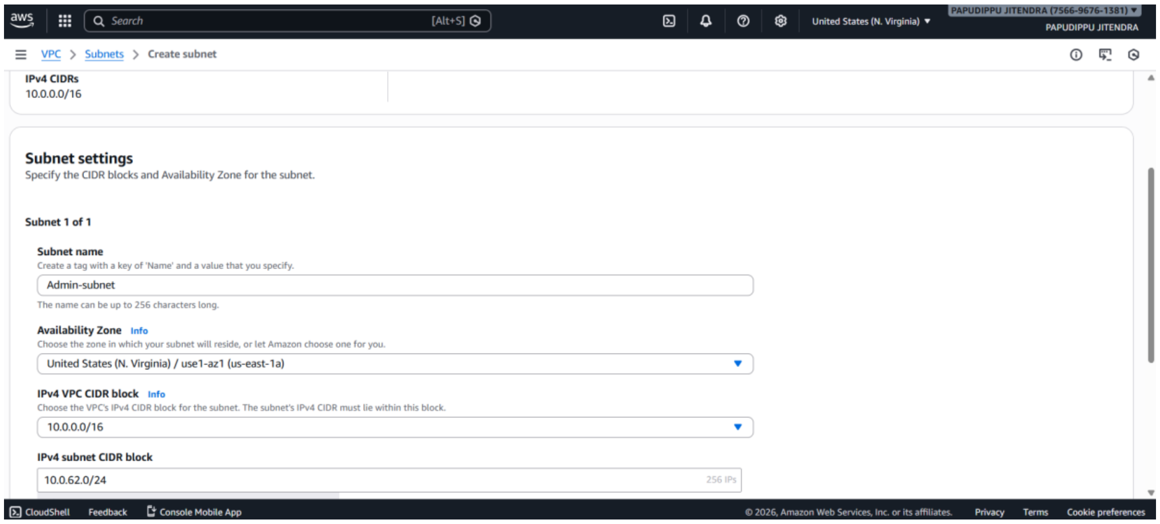
VPC encryption control ($) - *new* Info
Monitor mode provides visibility into encryption status without blocking traffic. Enforce mode prevents unencrypted traffic. Additional charges apply

Step 2: Create Subnet

- VPC: Custom-VPC

- Subnet name: Public-Subnet

- CIDR: 10.0.1.0/24



IPv4 CIDRs
10.0.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
Admin-subnet
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
United States (N. Virginia) / use1-az1 (us-east-1a)

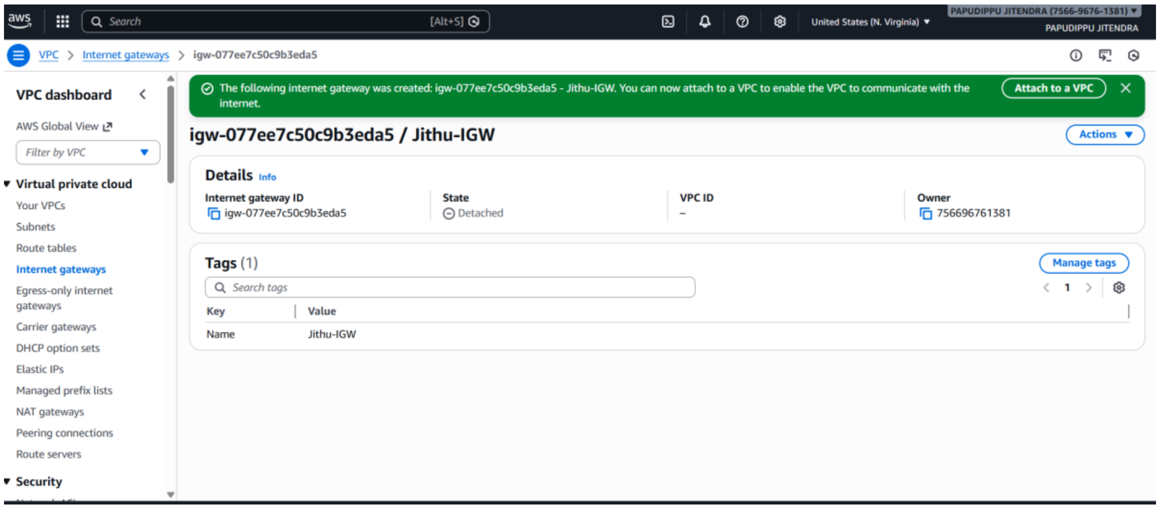
IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
10.0.0.0/16

IPv4 subnet CIDR block
10.0.62.0/24 256 IPs

Step 3: Create Internet Gateway

- Create IGW: Custom-IGW

- Attach IGW to Custom-VPC



The following internet gateway was created: igw-077ee7c50c9b3eda5 - Jithu-IGW. You can now attach to a VPC to enable the VPC to communicate with the internet.

igw-077ee7c50c9b3eda5 / Jithu-IGW

Details Info

Internet gateway ID	State	VPC ID	Owner
igw-077ee7c50c9b3eda5	Detached	–	756696761381

Tags (1)

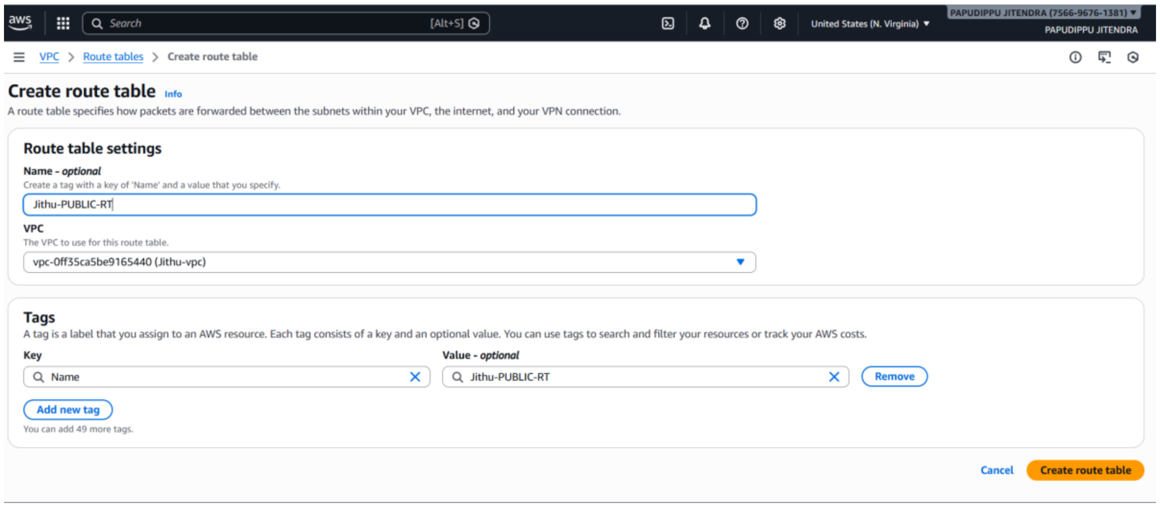
Key	Value
Name	Jithu-IGW

Step 4: Create Route Table

- Route Table name: Public-Route-Table

- Add route: 0.0.0.0/0 → IGW

- Associate Public-Subnet


Search [Alt+S]
United States (N. Virginia)
PAPUDIPPU JITENDRA (7566-9676-1381)
PAPUDIPPU JITENDRA
VPC > Route tables > Create route table

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - *optional*

Create a tag with a key of 'Name' and a value that you specify.

Jithu-PUBLIC-RT

VPC

The VPC to use for this route table.

vpc-0ff35ca5be9165440 (Jithu-vpc)

Tags

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	Value - *optional*
Name	Jithu-PUBLIC-RT

You can add 49 more tags.

----

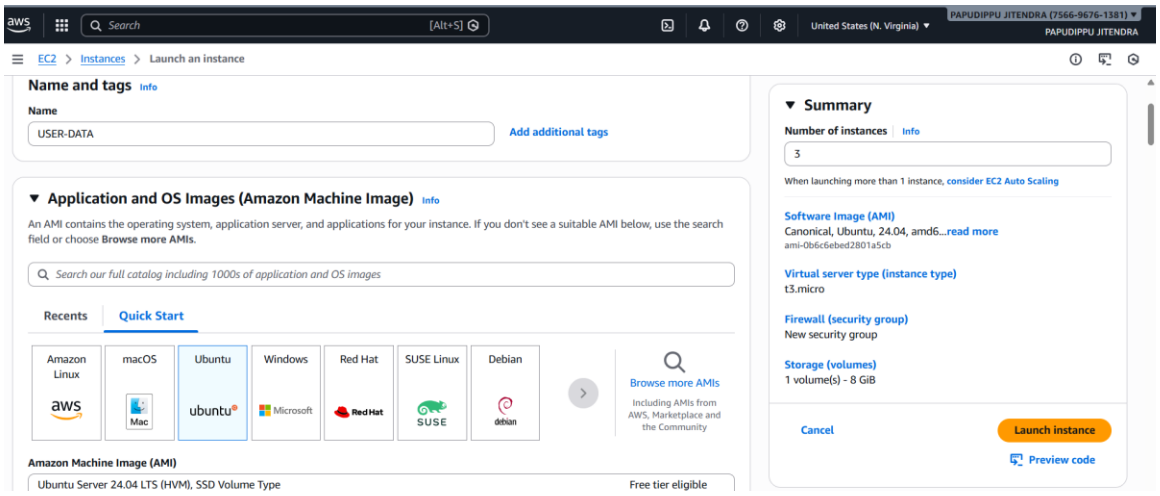
2. Launch EC2 Instance with User Data

Step 5: Launch Instance

- Name: Apache-Web-Server

- AMI: Ubuntu Server 20.04 / 22.04

- Instance type: t2.micro



EC2 > Instances > Launch an instance

Name and tags

Name: USER-DATA

Application and OS Images (Amazon Machine Image)

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**.

Quick Start: Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Amazon Machine Image (AMI): Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible

Summary

Number of instances: 3

When launching more than 1 instance, consider EC2 Auto Scaling

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more — ami-0b6c6ebed2801a5cb

Virtual server type (instance type): t3.micro

Firewall (security group): New security group

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

Network:

- VPC: Custom-VPC

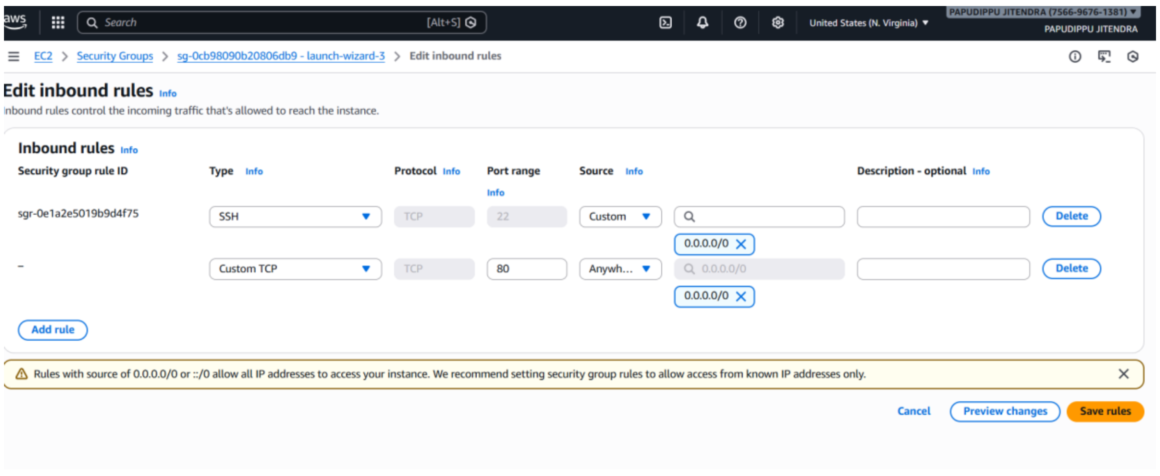
- Subnet: Public-Subnet

- Auto-assign Public IP: Enable

Security Group:

- SSH (22) – My IP

- HTTP (80) – Anywhere



EC2 > Security Groups > sg-0cb98090b20806db9 - launch-wizard-3 > Edit inbound rules

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-0e1a2e5019b9d4f75	SSH	TCP	22	Custom	0.0.0.0/0	
–	Custom TCP	TCP	80	Anywh...	0.0.0.0/0	

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Step 6: User Data Script

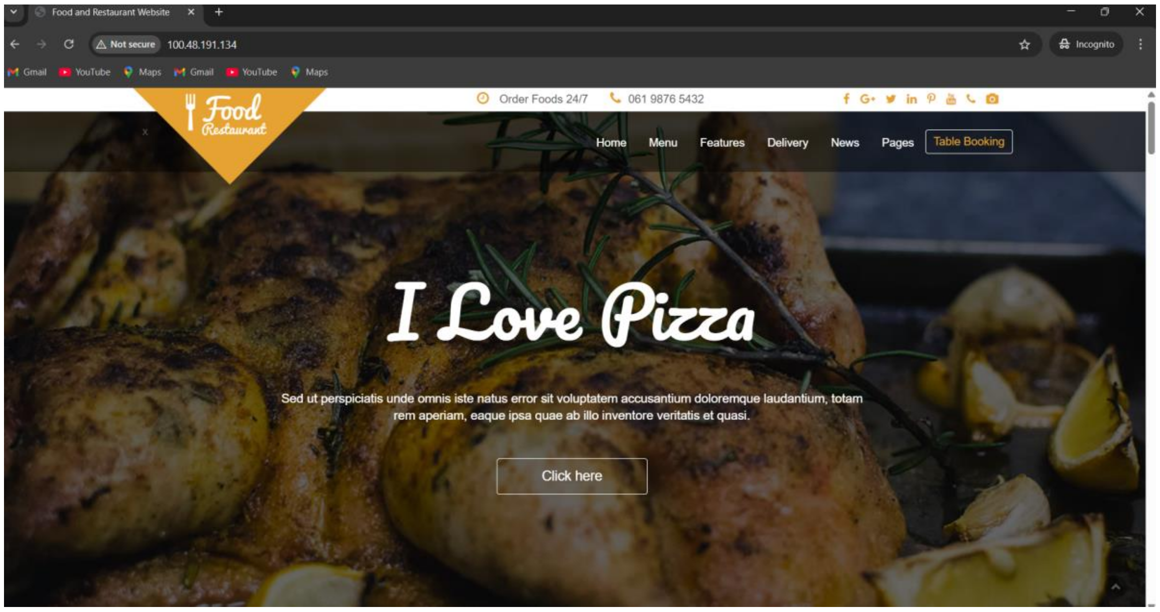
```
#!/bin/bash

sudo apt update -y

sudo apt install apache2 -y

sudo systemctl start apache2

sudo systemctl enable apache2

sudo rm -rf /var/www/html/*

sudo mkdir -p /var/www/html/

cd /var/www/html/

sudo git clone <YOUR_GIT_REPOSITORY_URL> .
```

----

3. Verify Application

- Copy Public IPv4 address

- Open browser: http://PUBLIC_IP



Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi.

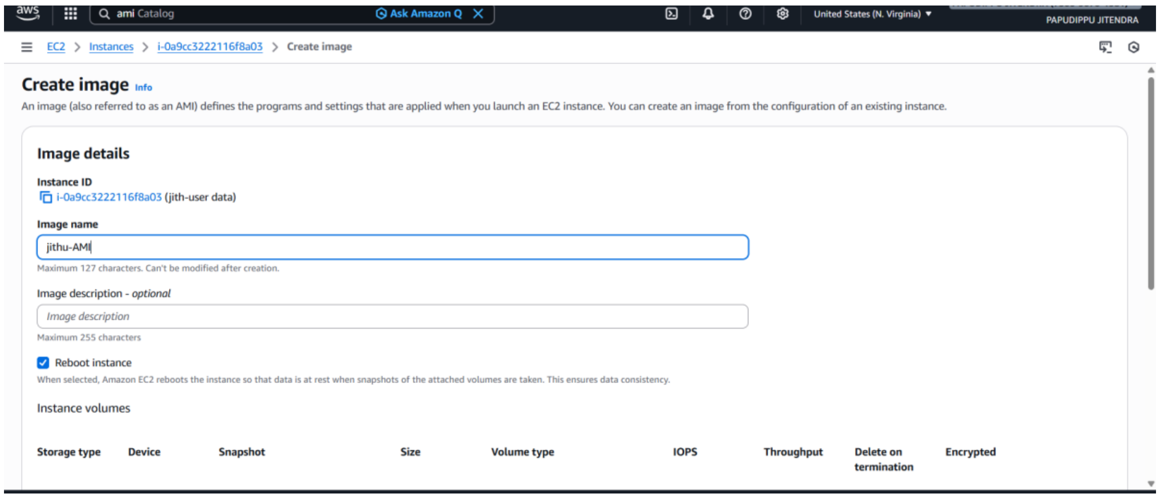
----

4. Create AMI

Step 7: Stop Instance

Step 8: Create Image

- Name: Apache-App-AMI


aws
ami Catalog
Ask Amazon Q
United States (N. Virginia)
PAPUDIPPU JITENDRA
EC2 > Instances > i-0a9cc3222116f8a03 > Create image

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-0a9cc3222116f8a03 (jith-user data)

Image name
jithu-AMI
Maximum 127 characters. Can't be modified after creation.

Image description - *optional*
Image description
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

Step 9: Verify AMI

- EC2 → AMIs → Status Available

----

5. Launch Instance Using AMI

- My AMIs → Select Apache-App-AMI

- Launch instance

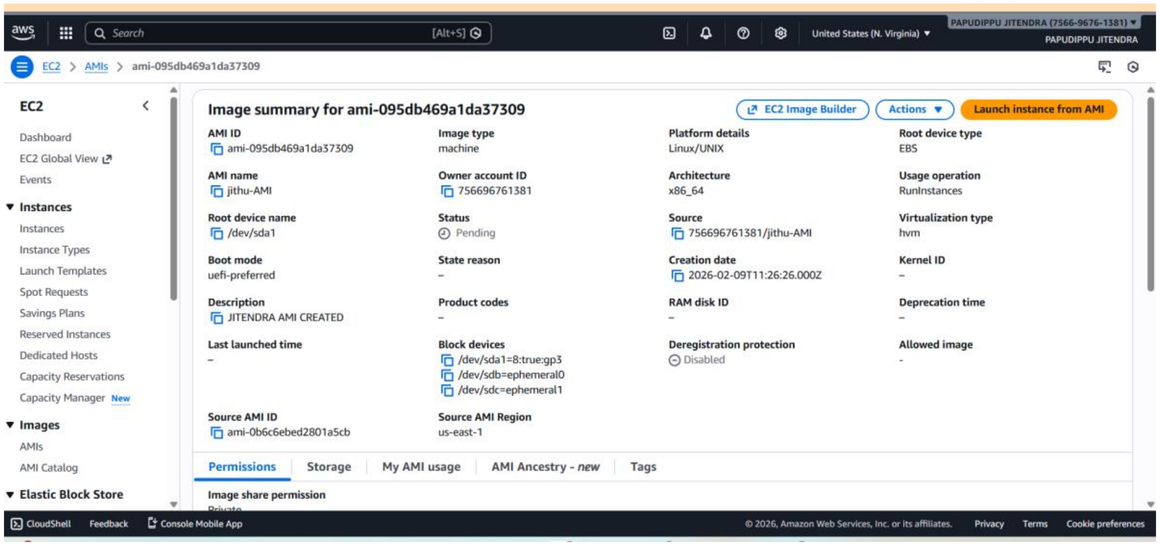


Image summary for ami-095db469a1da37309

AMI ID ami-095db469a1da37309	Image type machine	Platform details Linux/UNIX	Root device type EBS
AMI name jithu-AMI	Owner account ID 756696761381	Architecture x86_64	Usage operation RunInstances
Root device name /dev/sda1	Status Pending	Source 756696761381/jithu-AMI	Virtualization type hvm
Boot mode uefi-preferred	State reason –	Creation date 2026-02-09T11:26:26.000Z	Kernel ID –
Description JITENDRA AMI CREATED	Product codes –	RAM disk ID –	Deprecation time –
Last launched time –	Block devices /dev/sda1=8:true:gp3 /dev/sdb=ephemeral0 /dev/sdc=ephemeral1	Deregistration protection Disabled	Allowed image –
Source AMI ID ami-0b6c6ebed2801a5cb	Source AMI Region us-east-1		

Image share permission

Conclusion

This document demonstrated the creation of a custom VPC, deployment of an EC2 instance using User Data for automated Apache setup, and creation of an AMI for reusable infrastructure. These steps enable secure, automated, and scalable deployments in AWS.

End of Document