AWS VPC CIDR & Subnet Design with Enterprise Route Table Architecture

Project Overview

This project focuses on designing and deploying a **robust, enterprise-grade AWS VPC** that can support both current workloads and future expansion. The network architecture is built to:

- Use **different subnet sizes** tailored to workload requirements
- Enforce **deliberate and restricted internet connectivity**
- Apply **structured CIDR planning** aligned with industry standards
- Maintain **strong security, scalability, and traceability**

Importance of This Approach

- Network design decisions are **difficult to change later**
- Poor IP planning leads to **address exhaustion and rework**
- Clearly defined routing improves **security control and visibility**

Business Use Case

The organization is establishing a **centralized cloud networking layer** to host multiple types of services, including:

- Administrative and operational systems
- Internet-facing web applications
- Backend application services
- Shared internal utilities
- Containerized, high-growth workloads

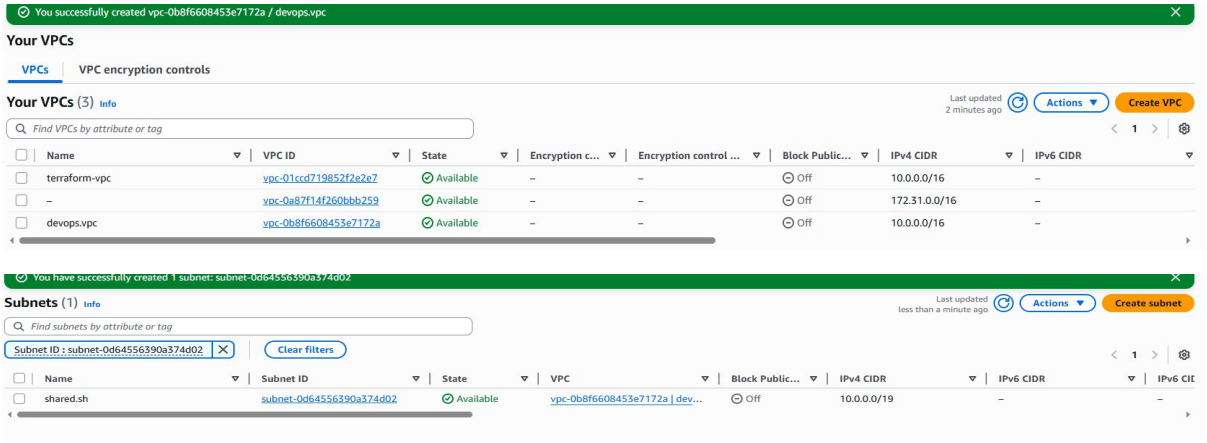
Rationale for a Shared VPC Model

- Enables **central policy enforcement**
- Simplifies **logging, auditing, and monitoring**
- Reduces operational overhead and **optimizes costs**

TASK 1: VPC CIDR & Capacity Planning

1. Choose a private IP range suitable for enterprise workloads.
2. Allocate a CIDR block **not smaller than /16** to allow future growth.
3. Create a VPC with the following CIDR:
    - **VPC CIDR:** 10.0.0.0/16
4. Enable DNS resolution and DNS hostnames for internal communication.

Result:
A VPC with 65,536 IP addresses, sufficient for current and future subnet expansion.



STEP 2: Plan Subnet Allocation (Largest First)

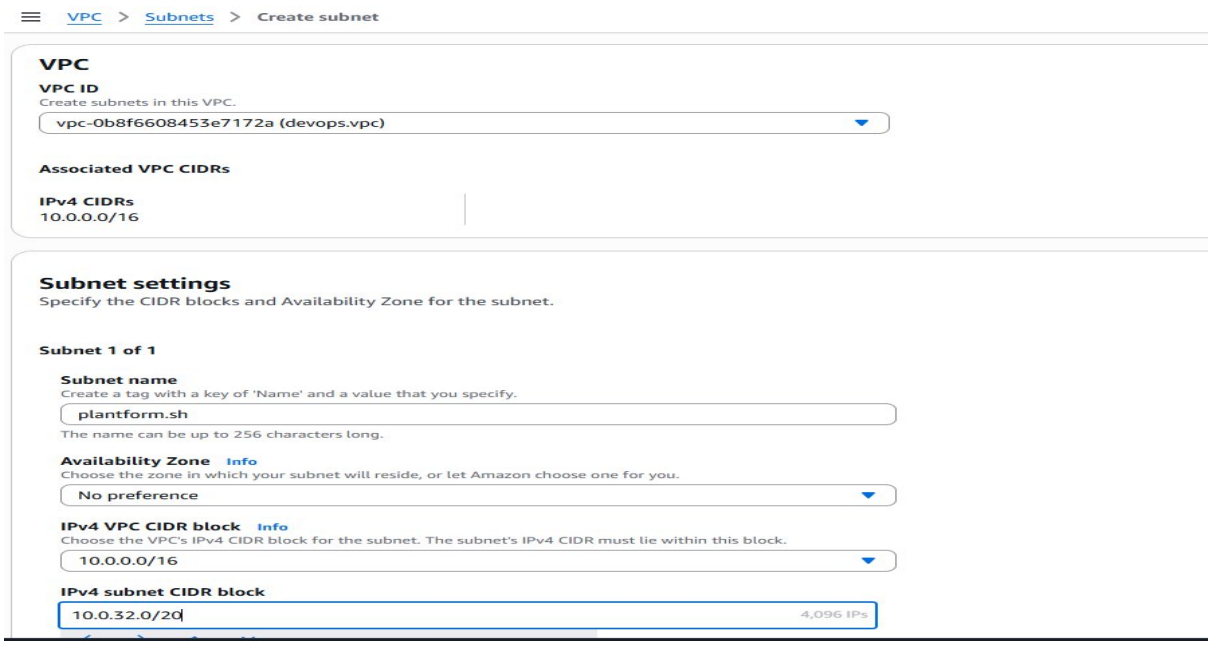
1. List subnet requirements in descending order of IP capacity.
2. Start allocating CIDRs from the beginning of the VPC range.
3. Ensure:
    - No CIDR overlap
    - Correct network boundaries
    - All subnets remain within 10.0.0.0/16

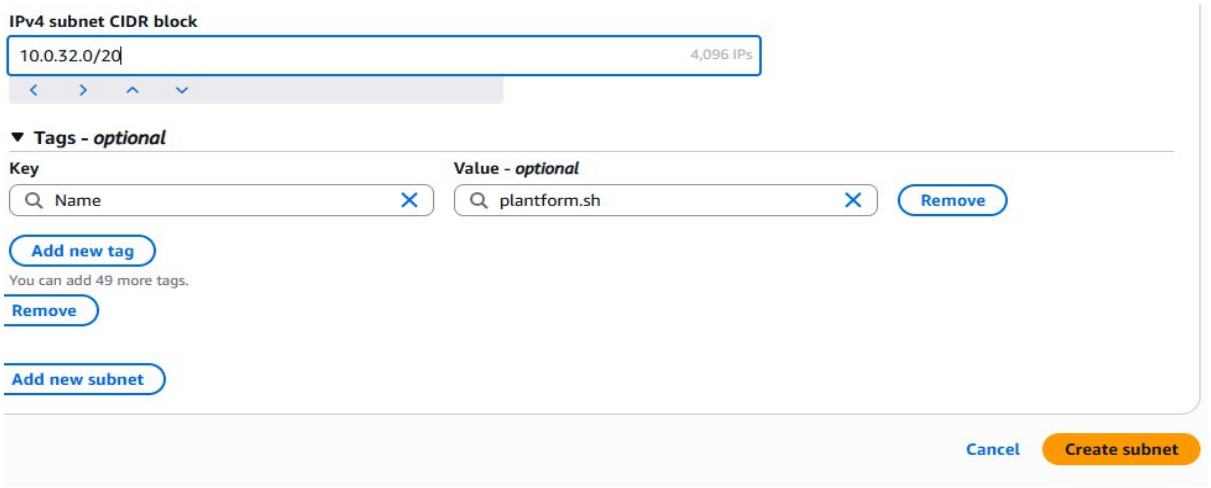
STEP 3: Create Subnets

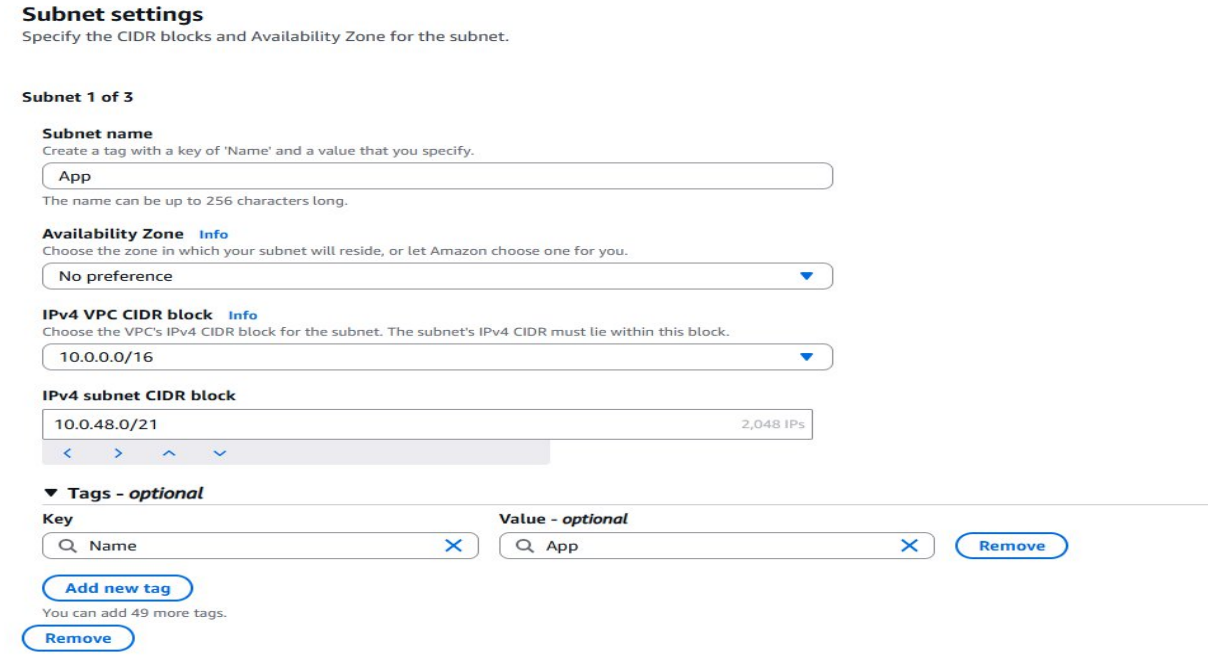
Create the following six subnets in order:

1. **Shared Subnet**
    - CIDR: 10.0.0.0/19
    - Purpose: Large internal services
2. **Platform Subnet**
    - CIDR: 10.0.32.0/20
    - Purpose: Containers and internal tools
3. **App Subnet**
    - CIDR: 10.0.48.0/21
    - Purpose: Application tier
4. **Web Subnet**
    - CIDR: 10.0.56.0/22
    - Purpose: Web tier
5. **Edge Subnet**
    - CIDR: 10.0.60.0/23
    - Purpose: Load balancers / ingress
6. **Admin Subnet**
    - CIDR: 10.0.62.0/24
    - Purpose: Bastion hosts and operations

Result:
Six non-overlapping subnets with different capacities, all inside the VPC.


VPC > Subnets > Create subnet
VPC
VPC ID
Create subnets in this VPC.
vpc-0b8f6608453e7172a (devops.vpc)
Associated VPC CIDRs
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.
plantform.sh
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.
No preference
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
10.0.32.0/20
4,096 IPs


IPv4 subnet CIDR block
10.0.32.0/20
4,096 IPs
Tags - optional
Key
Value - optional
Name
plantform.sh
Remove
Add new tag
You can add 49 more tags.
Remove
Add new subnet
Cancel
Create subnet


Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.
Subnet 1 of 3
Subnet name
Create a tag with a key of 'Name' and a value that you specify.
App
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.
No preference
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
10.0.48.0/21
2,048 IPs
Tags - optional
Key
Value - optional
Name
App
Remove
Add new tag
You can add 49 more tags.
Remove

Subnet 2 of 3

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
web
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.
No preference

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
10.0.56.0/22 1,024 IPs

▼ **Tags - *optional***

Key	Value - *optional*	
Name	web	Remove

Add new tag
You can add 49 more tags.
Remove

Subnet 3 of 4

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
App
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.
No preference

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
10.0.56.0/22 1,024 IPs

▼ **Tags - *optional***

Key	Value - *optional*	
Name	App	Remove

Add new tag
You can add 49 more tags.
Remove

Subnet 4 of 5

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
Edge
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.
No preference

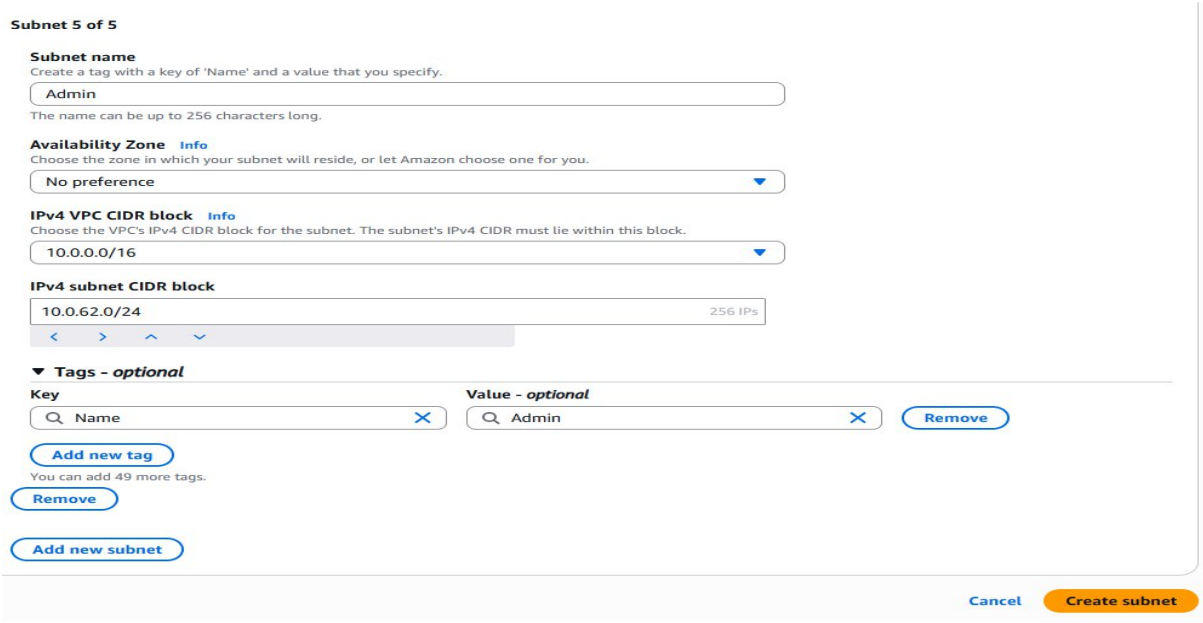
IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
10.0.0.0/16

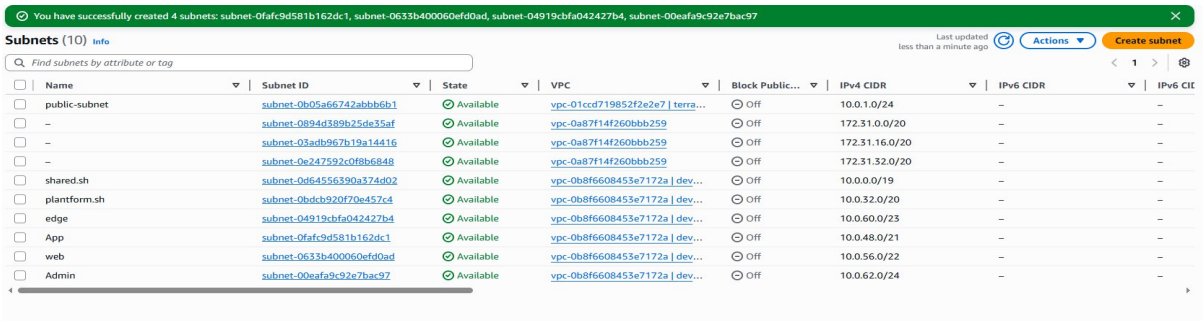
IPv4 subnet CIDR block
10.0.60.0/23 512 IPs

▼ **Tags - *optional***

Key	Value - *optional*	
Name	Edge	Remove

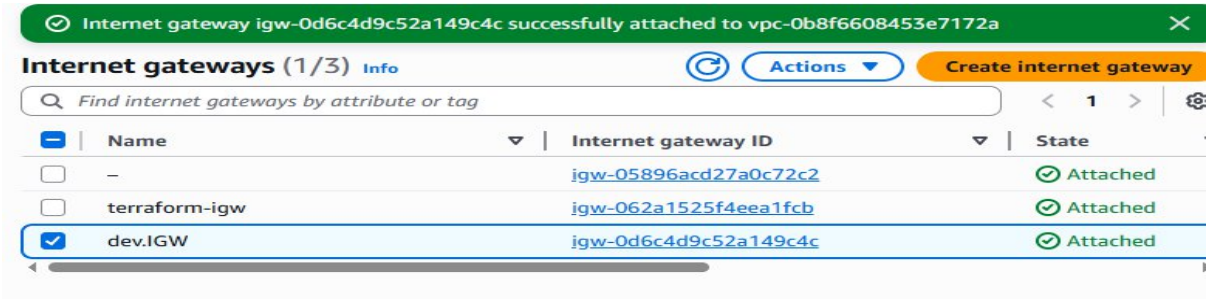
Add new tag
You can add 49 more tags.
Remove





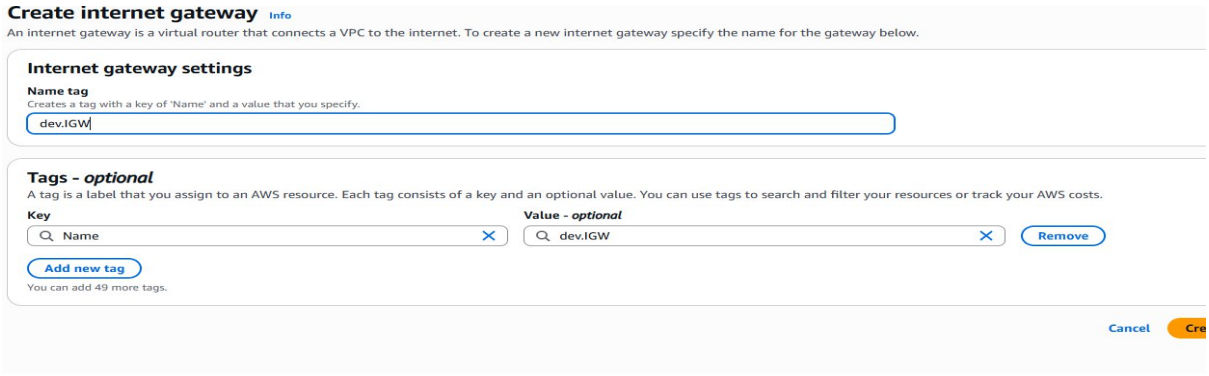
STEP 4: Create and Attach Internet Gateway

Internet gateway creations (dev.igw)

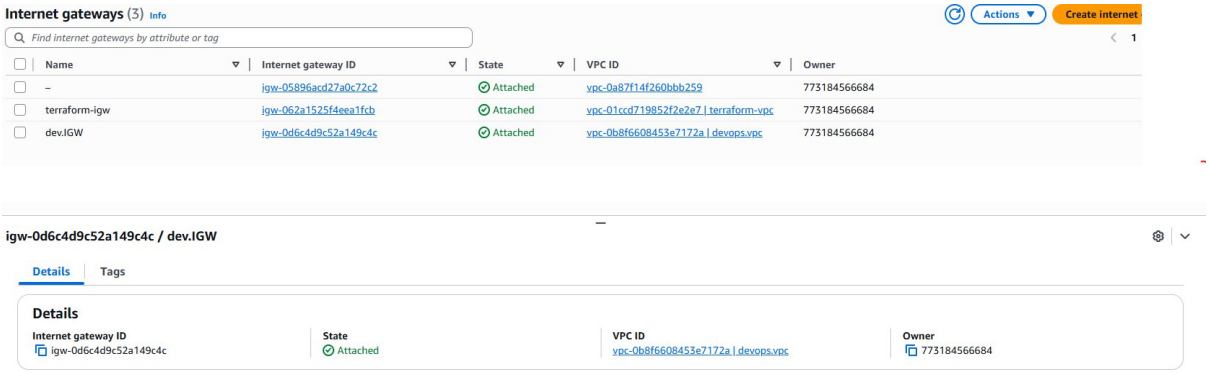


Why IGW is required:

- Enables communication between VPC and the internet
- Does nothing unless routing explicitly allows it
- Central control point for outbound/inbound traffic



➢ **Internet gateway Attached to vpc**



Why attachment matters:

- Without attachment, public subnets cannot access internet
- Ensures intentional connectivity

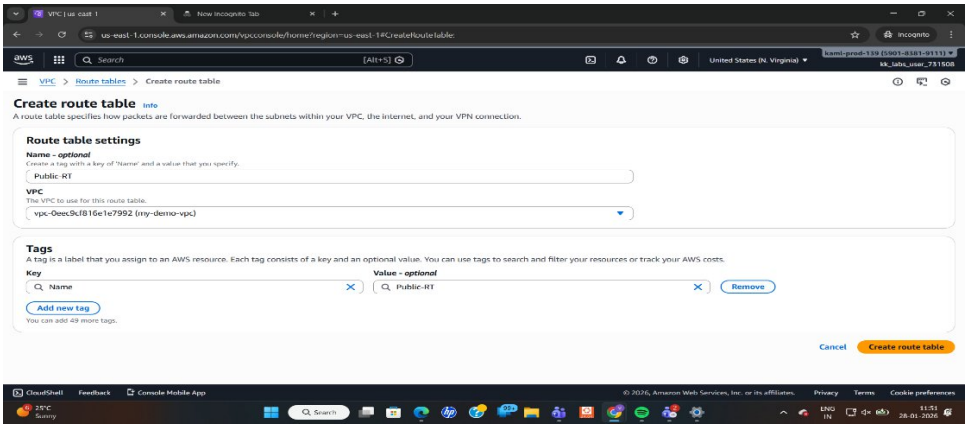
STEP 5: Create Route Tables

Step 5.1: Create Public Route Table (Public-RT)

1. Create a route table named **Public-RT**.
2. Add the following routes:
   - `10.0.0.0/16` → Local
   - `0.0.0.0/0` → Internet Gateway

Why this is used:

- Allows controlled internet access
- Used only by Admin and Edge subnets

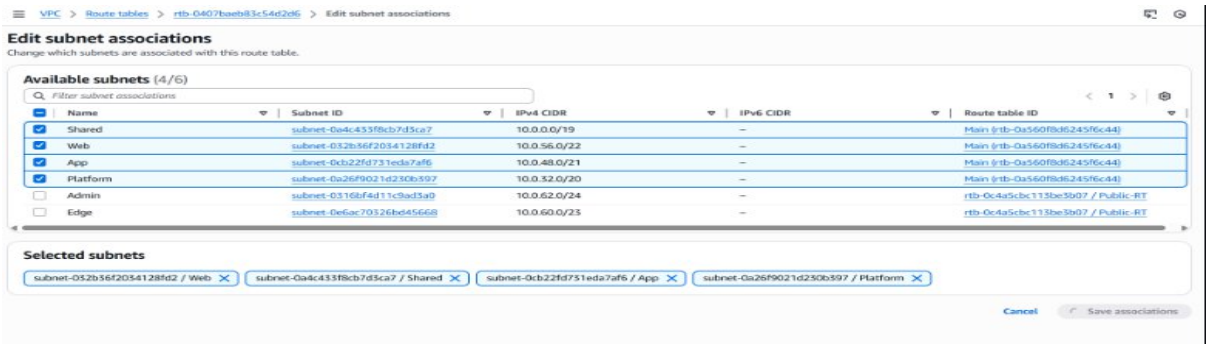


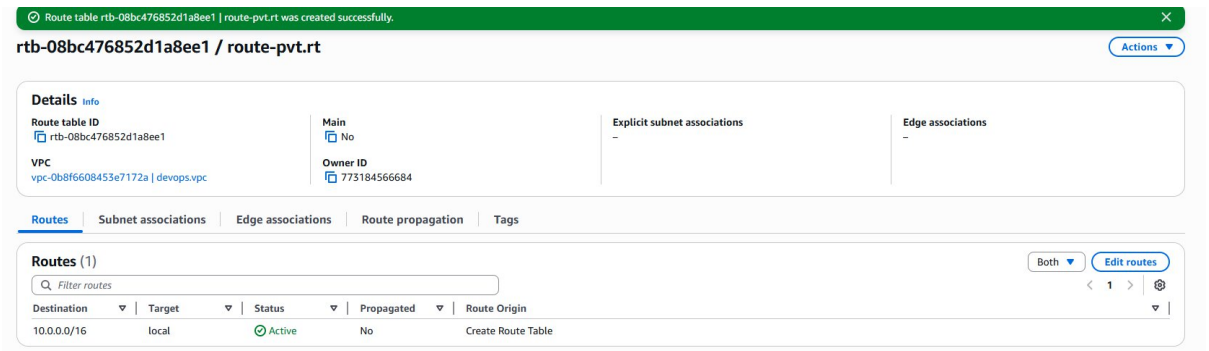
Step 5.2: Create Private Route Table (Private-RT)

1. Create a route table named **Private-RT**.
2. Ensure only the default route exists:
    - `10.0.0.0/16` → Local

Why no internet route:

- Prevents accidental internet exposure
- Enforces zero-trust networking





Route Table Associations

Subnet	Route Table	Why
Admin	Public-RT	Ops access
Edge	Public-RT	Ingress traffic
Web	Private-RT	Internal only
App	Private-RT	Backend isolation
Platform	Private-RT	Secure tooling
Shared	Private-RT	Internal services

📸 **Image-12:** Route table associations

Why explicit association:

- Prevents use of default main route table
- Improves audit visibility

8. Security-Driven Network Behavior (Task)

Subnet Type	Internet Access	Reason
Admin	Yes	IGW route
Edge	Yes	IGW route
Others	No	No default route

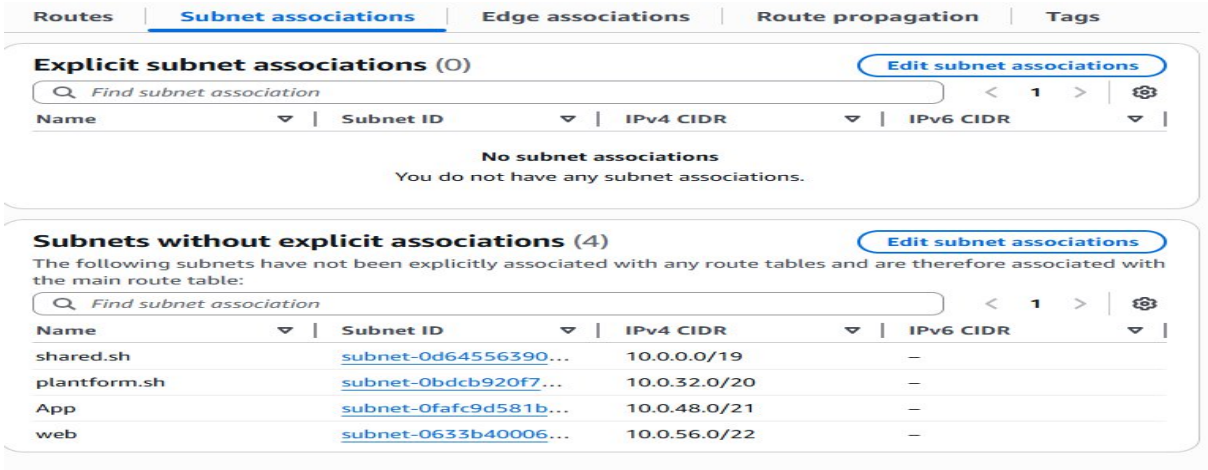
Why this works:

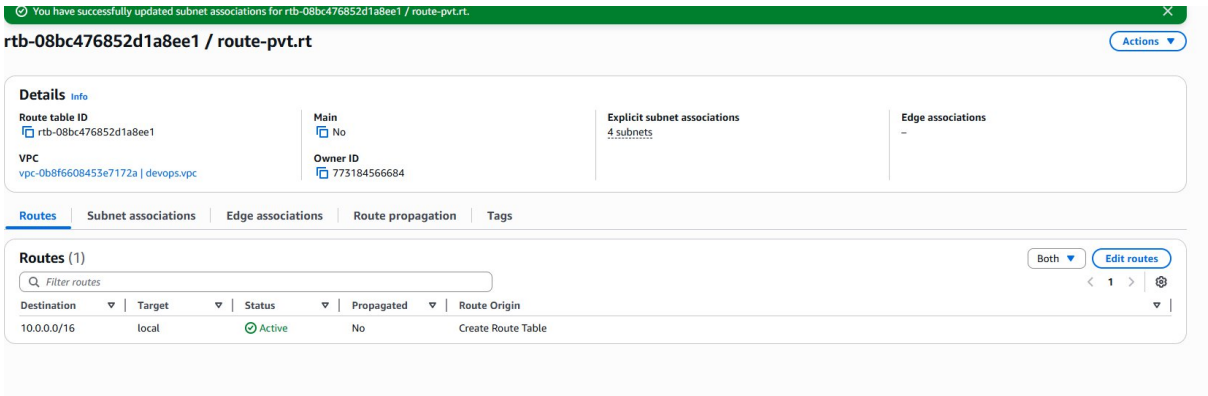
- AWS local routing enables internal communication
- No security groups needed to prove isolation

STEP 6: Associate Route Tables with Subnets

1. Explicitly associate **Admin** subnet with **Public-RT**.
2. Explicitly associate **Edge** subnet with **Public-RT**.
3. Explicitly associate the following subnets with **Private-RT**:
    - Web
    - App
    - Platform
    - Shared
4. Verify no subnet is using the **main route table**.

Result:
Public and private access is strictly controlled.



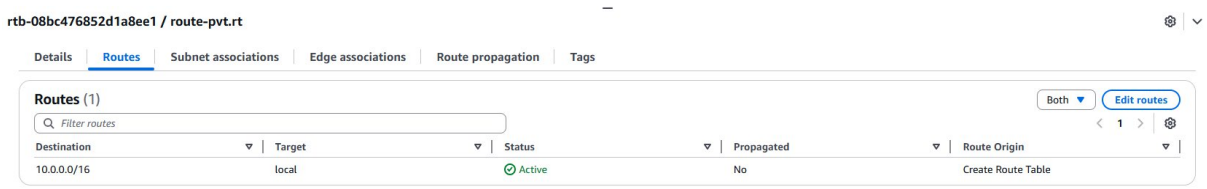


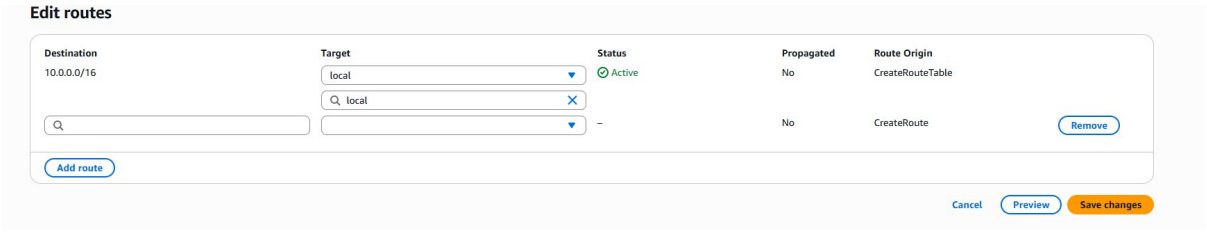
STEP 7: Enforce Security-Driven Network Behavior

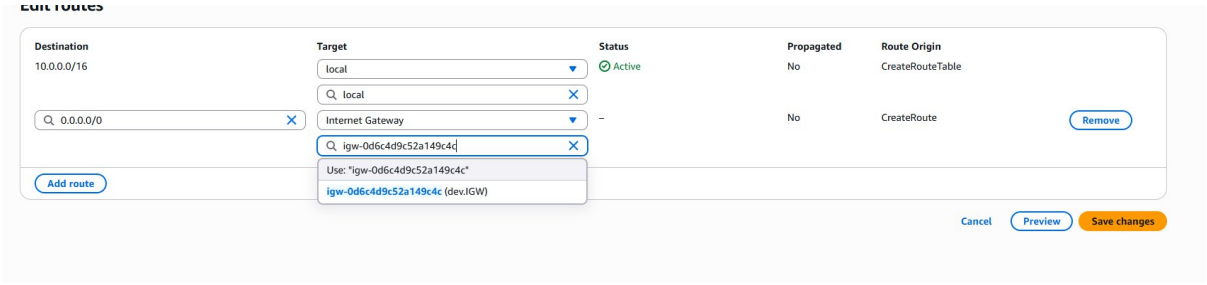
1. Confirm only Public-RT has a route to the Internet Gateway.
2. Verify private subnets have no `0.0.0.0/0` route.
3. Ensure all subnets retain the local VPC route.

Outcome:

- Admin & Edge → Internet access allowed
- Web, App, Platform, Shared → Internet blocked
- Internal communication → Allowed







STEP 8: Validate Network Behavior

Step 8.1: Test Public Subnets

1. Launch an EC2 instance in Admin or Edge subnet.
2. Attempt to access an external website.

Expected Result:
Internet access succeeds due to IGW route.

Step 8.2: Test Private Subnets

1. Launch an EC2 instance in any private subnet.
2. Attempt to access the internet.

Expected Result:
Connection fails due to absence of internet route.

Step 8.3: Test Internal Communication

1. Test connectivity between instances in different subnets.

Expected Result:
Communication succeeds via local VPC routing.

STEP 9: Failure & Audit Scenarios

Scenario 1: Internet Gateway Detached

1. Detach IGW from the VPC.

Result:

- All internet traffic fails
- Internal traffic remains functional

Scenario 2: Private Subnet Associated with Public-RT

1. Associate a private subnet with Public-RT.

Result:

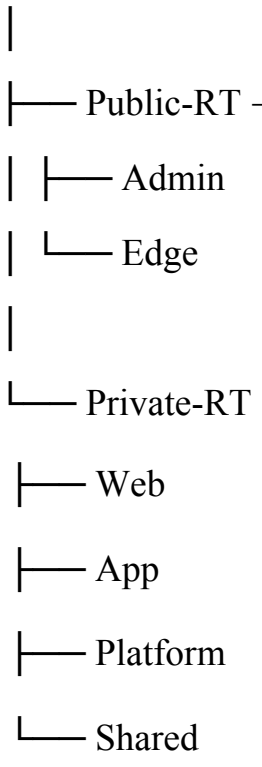
- Subnet becomes internet-accessible
- Violates security and compliance policies

\

Architecture Diagram (Logical)

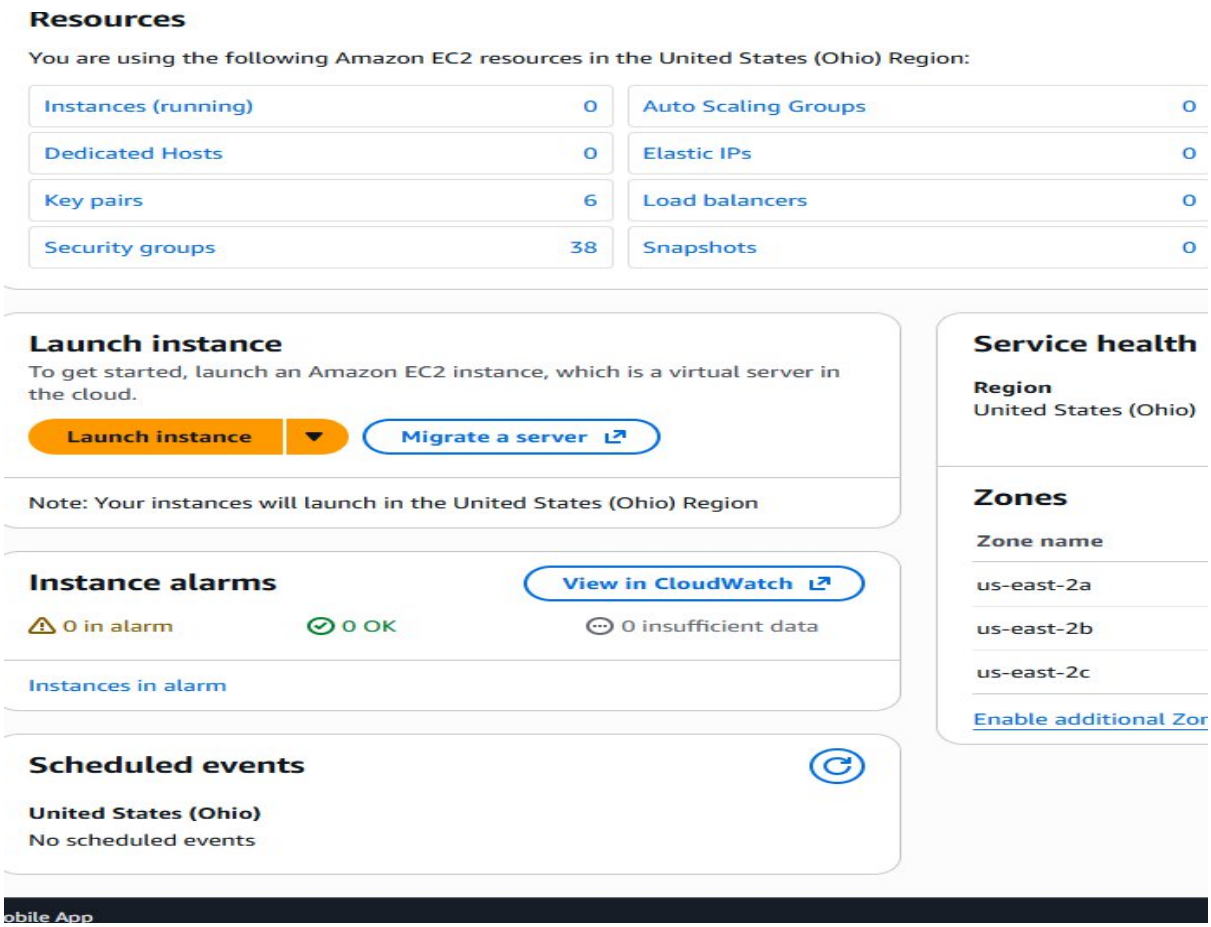
📸 **Image-15:** Logical VPC architecture diagram

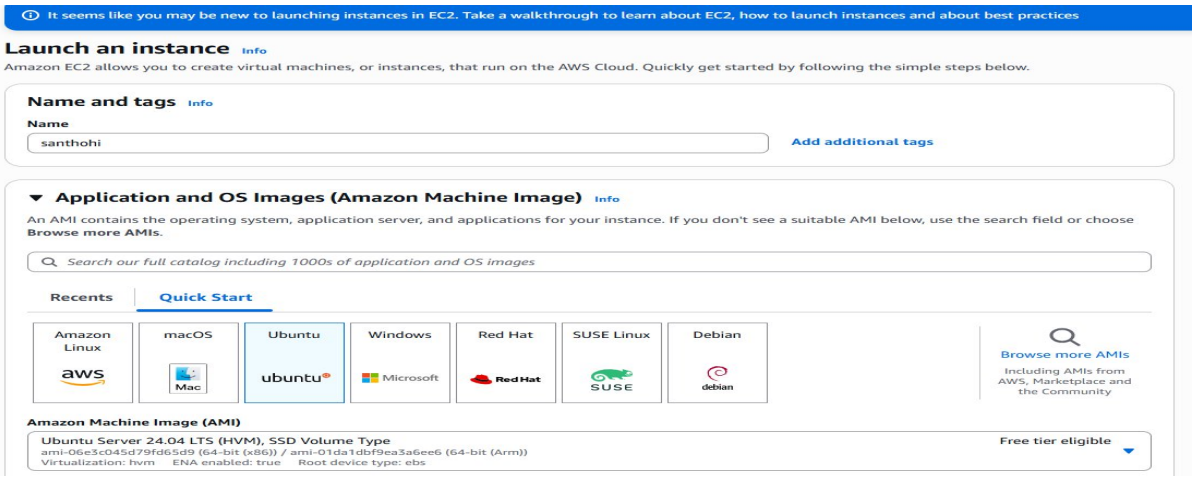
VPC 10.0.0.0/16



Conclusion

This VPC design follows **enterprise-grade networking principles**, ensures **strong security boundaries**, and provides a **future-proof IP addressing strategy**. The architecture is scalable, auditable, and suitable for long-term production use.





Description
Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture	AMI ID	Publish Date	Username	
64-bit (x86)	ami-06e3c045d79fd65d9	2025-12-12	ubuntu	Verified provider

Instance type Info | Get advice

Instance type

t3.micro — Free tier eligible
Family: t3 2 vCPU 1 GiB Memory Current generation: true On-Demand RHEL base pricing: 0.0392 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour On-Demand Windows base pricing: 0.0196 USD per Hour
On-Demand SUSE base pricing: 0.0104 USD per Hour On-Demand Linux base pricing: 0.0104 USD per Hour

All generations
Compare instance types

Additional costs apply for AMIs with pre-installed software

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*

sandeep — Create new key pair

VPC - *required* Info

vpc-0b8f6608453e7172a (devops.vpc)
10.0.0.0/16

Subnet Info

subnet-00eafa9c92e7bac97 — Admin
VPC: vpc-0b8f6608453e7172a Owner: 773184566684 Availability Zone: us-east-2b (use2-az2)
Zone type: Availability Zone IP addresses available: 251 CIDR: 10.0.62.0/24)

Create new subnet

Auto-assign public IP Info

Enable

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

Security group name - *required*

launch-wizard-36

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Description - *required* Info

launch-wizard-36 created 2026-01-28T07:31:09.948Z

Inbound Security Group Rules

Security group rule 1 (TCP, 22, 0.0.0.0/0) — Remove

Type Info	Protocol Info	Port range Info
ssh	TCP	22

Source type Info	Source Info	Description - *optional* Info
Anywhere	Add CIDR, prefix list or security group — 0.0.0.0/0	e.g. SSH for admin desktop

Connect Info

Connect to an instance using the browser-based client.

EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

Instance ID
i-03348593fdcf087a5 (devops)

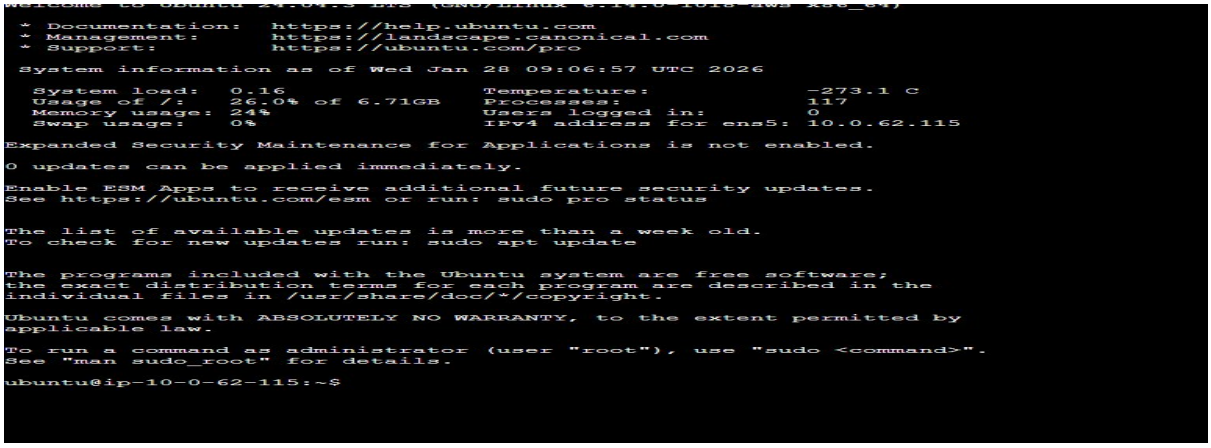
1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is saikiran.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 "saikiran.pem"
4. Connect to your instance using its Public IP:
   3.23.114.78

Command copied

ssh -i "saikiran.pem" ubuntu@3.23.114.78

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.

Cancel


* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro
System information as of Wed Jan 28 09:06:57 UTC 2026
System load: 0.16
Usage of /: 26.0% of 6.71GB
Memory usage: 24%
Swap usage: 0%
Temperature: -273.1 C
Processes: 117
Users logged in: 0
IPv4 address for ens5: 10.0.62.115
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
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-10-0-62-115:~$