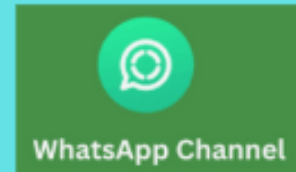


@devopschallengehub



## How do you create VPC in AWS?

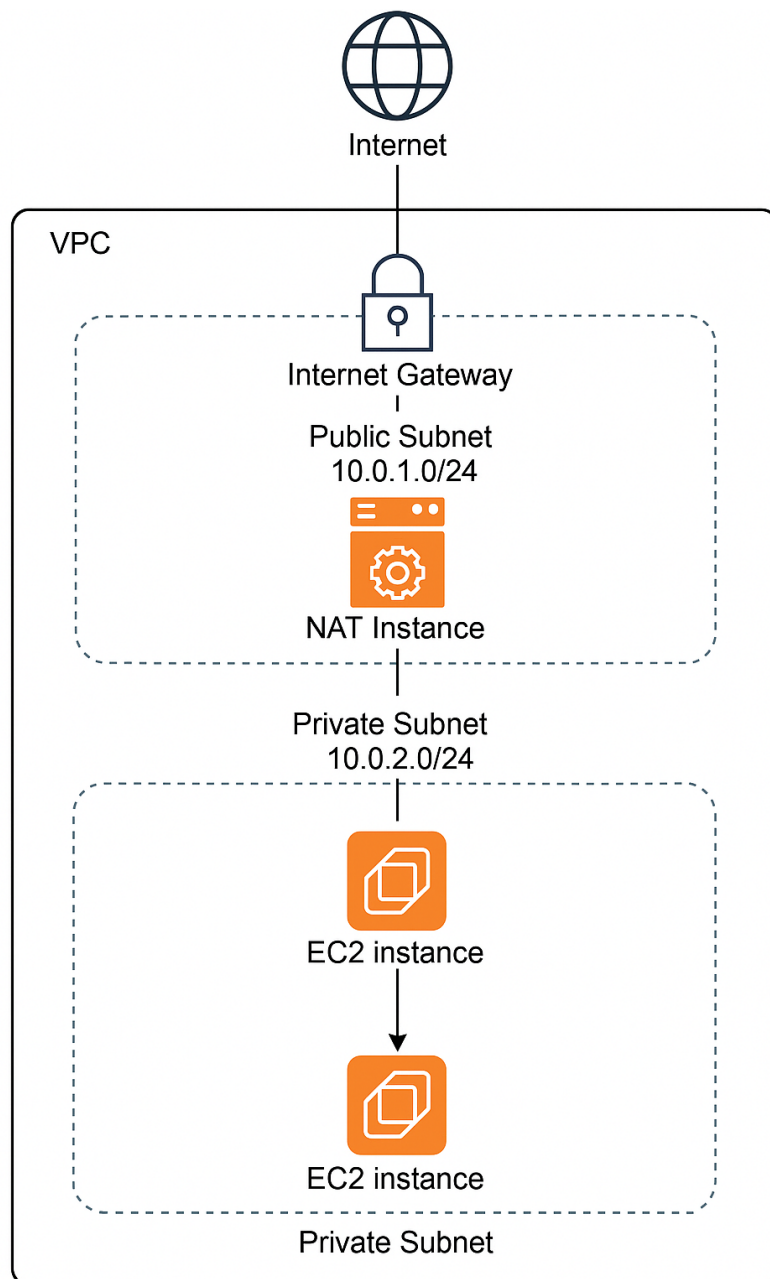
### Concept

VPC	Custom VPC creation
Subnet	Public & private subnets
Internet Gateway	Public internet access
NAT instance	Private instance internet access
Route Tables	Routing traffic from/to subnets
Security Groups	Controlling instance access
EC2	One public, one private

---

### Prerequisites

- AWS Free Tier account
  - Key pair created
  - Basic IAM permissions to create EC2, VPC, etc.
  - Use **us-east-1** or **any region with AMIs available**
-



## ✅ Step-by-Step Setup

### 1. Create a VPC

- CIDR block: 10.0.0.0/16
- Enable DNS hostname: ✅

### 2. Create Subnets

- **Public Subnet:** 10.0.1.0/24 (e.g., AZ: us-east-1a)
- **Private Subnet:** 10.0.2.0/24 (e.g., AZ: us-east-1a)

### 3. Create an Internet Gateway (IGW)

- Attach it to the VPC

### 4. Create a Route Table for Public Subnet

- Add route: 0.0.0.0/0 → IGW
- Associate with **Public Subnet**

### 5. Launch a NAT Instance

- Use a NAT AMI (e.g., amzn-ami-vpc-nat)

- Place it in **Public Subnet**
- Enable Source/Dest Check = Disabled
- Add Elastic IP to it

#### 6. Create a Route Table for Private Subnet

- Add route: 0.0.0.0/0 → NAT Instance (not IGW)
- Associate with **Private Subnet**

#### 7. Launch EC2 in Public Subnet

- Use Amazon Linux
- Assign public IP
- Security Group: Allow SSH from your IP

#### 8. Launch EC2 in Private Subnet

- No public IP
- Security Group: Allow SSH from Public EC2's private IP

---

### Live Demo

#### Part 1: VPC + Subnet + IGW + Public EC2

- Show how EC2 in public subnet gets internet
- SSH into it from your local machine

#### Part 2: NAT + Private EC2


- SSH from public EC2 → private EC2
- Show curl google.com or yum update on private EC2 to prove internet via NAT

#### Part 3: Troubleshoot Scenarios (Good for Engagement)

- Remove NAT route → internet fails
- Enable/disable Source/Dest check on NAT → show the effect
- Attach IGW but forget route → demo broken internet


### 1. You want to host a public web server on EC2 that users can access over the internet. Which setup is MOST appropriate?

- A. Launch the EC2 instance in a private subnet with a NAT Gateway
- B. Launch the EC2 instance in a public subnet with an Internet Gateway and assign a public IP**
- C. Launch the EC2 instance in a private subnet with no route to the internet
- D. Launch the EC2 instance in a public subnet without any route table changes

 **Correct Answer:** B. Launch the EC2 instance in a public subnet with an Internet Gateway and assign a public IP


### 2. Your backend EC2 instance should not be accessible from the internet but needs to download OS patches. What setup should you use?

- A. Place the instance in a public subnet and attach an Elastic IP
- B. Place the instance in a private subnet with no internet access
- C. Place the instance in a private subnet with a route to a NAT Gateway in a public subnet
- D. Place the instance in a private subnet and use a security group allowing inbound internet access

 **Correct Answer:** C. Place the instance in a private subnet with a route to a NAT Gateway in a public subnet

**3. A database in your private subnet should only accept traffic from application servers in the same VPC. How do you achieve this?**

- A. Use a Security Group that allows access from the internet
- B. Assign a public IP to the database and control access via NACL
- C. Use a Security Group that allows inbound traffic only from the app server's security group
- D. Use a Route Table entry for 0.0.0.0/0

 **Correct Answer:** C. Use a Security Group that allows inbound traffic only from the app server's security group