AWS VPC Setup with 2 Ubuntu EC2 Instances

One public instance (bastion/frontend) and one private instance (backend) — step by step

1. Overview

This guide creates a secure AWS Virtual Private Cloud (VPC) with:
- VPC: 10.0.0.0/16
- Public Subnet (10.0.1.0/24): Ubuntu EC2 with public IP (bastion/frontend)
- Private Subnet (10.0.2.0/24): Ubuntu EC2 without public IP (backend)
- Internet Gateway for public subnet
- NAT Gateway for private instance outbound internet

2. Create VPC & Subnets

Steps (Console):
1. Go to VPC → Create VPC. Enter 10.0.0.0/16. Enable DNS hostnames.
2. Create 2 subnets:

Name	CIDR	Subnet type	Note
demo-public-a	10.0.1.0/24	Public	Enable auto-assign public IP
demo-private-a	10.0.2.0/24	Private	No public IPs

3. Internet & NAT Gateways

- Create an Internet Gateway, attach to VPC
- Create Public Route Table: 0.0.0.0/0 → IGW; associate to public subnet
- Allocate Elastic IP, create NAT Gateway in public subnet
- Create Private Route Table: 0.0.0.0/0 → NAT; associate to private subnet

4. Security Groups

demo-web-sg (Public EC2):
- Inbound: SSH (22) from Your IP
- Inbound: HTTP (80) from 0.0.0.0/0
demo-app-sg (Private EC2):
- Inbound: SSH (22) from demo-web-sg
- Inbound: app ports (e.g., 4000) from demo-web-sg

5. Launch EC2 Instances

Public EC2 (bastion/frontend): Ubuntu 22.04, subnet demo-public-a, public IP enabled, SG demo-web-sg
Private EC2 (backend): Ubuntu 22.04, subnet demo-private-a, no public IP, SG demo-app-sg

6. Connect to Instances (Agent Forwarding)

Use the public EC2 as bastion, keep your PEM locally, forward agent:
eval "$(ssh-agent -s)" ssh-add /path/to/key.pem ssh -A -i /path/to/key.pem ubuntu@ ssh ubuntu@10.0.2.74

7. Verify Connectivity

- From private EC2: ping 8.8.8.8 (via NAT)
- From public EC2: ssh ubuntu@10.0.2.74

8. Cleanup

Terminate EC2s, delete NAT Gateway (billable), release Elastic IP, delete IGW, route tables, subnets, and finally VPC.