**Shaikh Ateeb Ahmed**
**22-08-2025**

---

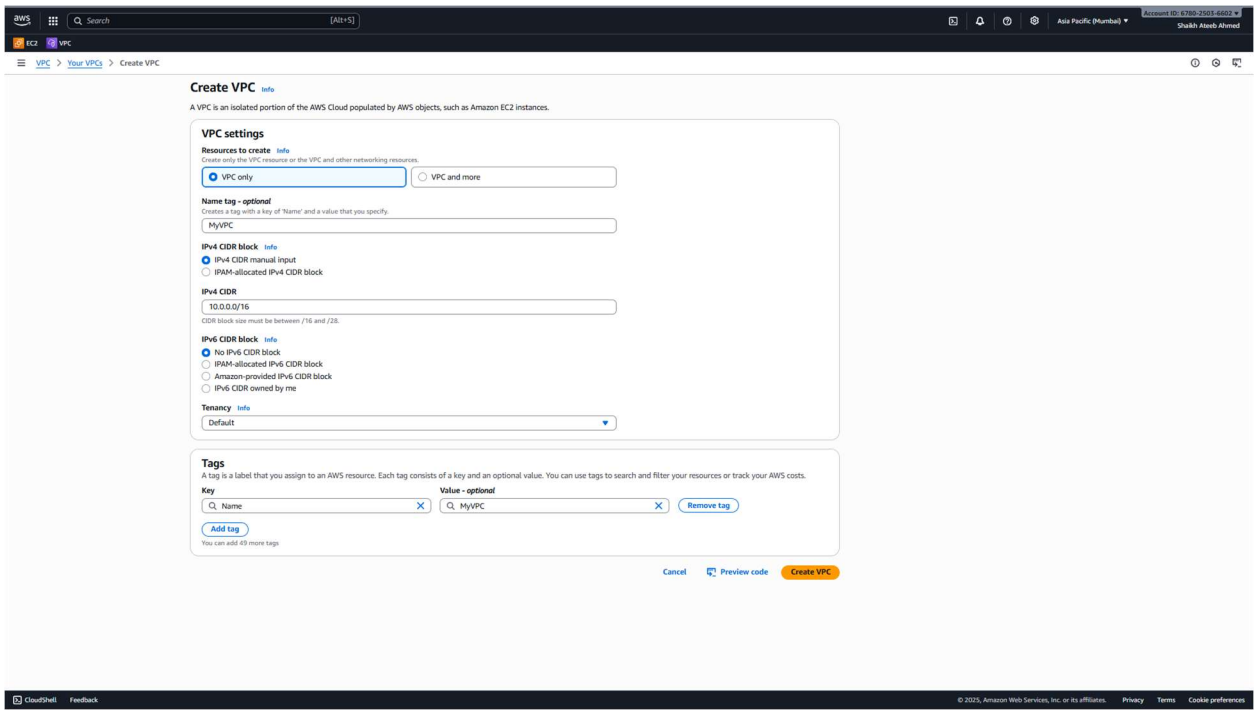
# Secure a Web Server using EC2, Subnets, and Firewalls

### Step 1: Create a VPC and subnet

1. Sign in to the AWS Console and open the **VPC** service.
2. Create the VPC

- Navigation: **VPC → Your VPCs → Create VPC**.
- Settings:

- **Name tag**: MyVPC
- **IPv4 CIDR block**: 10.0.0.0/16
- **Tenancy**: Default



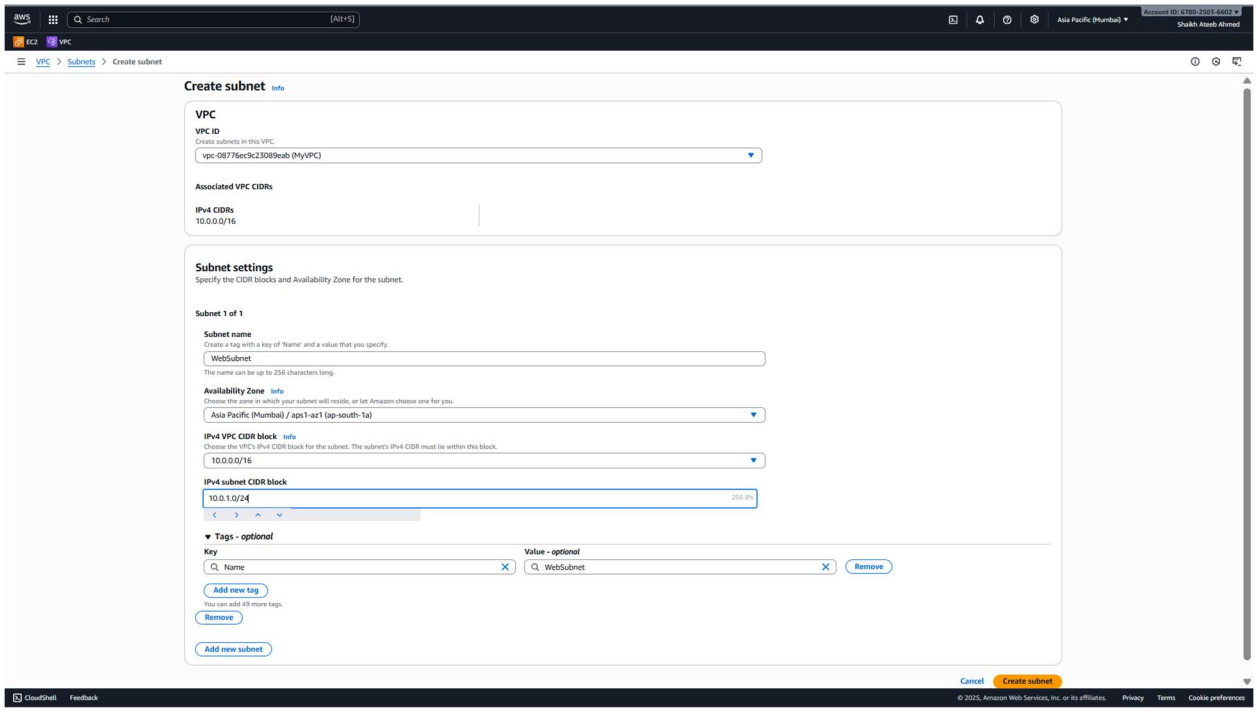

- Click **Create VPC** (or **Create**).

3. Create the subnet

- Navigation: **VPC → Subnets → Create subnet**.
- Settings:

- **VPC**: select MyVPC
- **Name tag**: WebSubnet

- **Availability Zone**: choose one (example: ap-south-1a)
- **IPv4 CIDR block**: 10.0.1.0/24

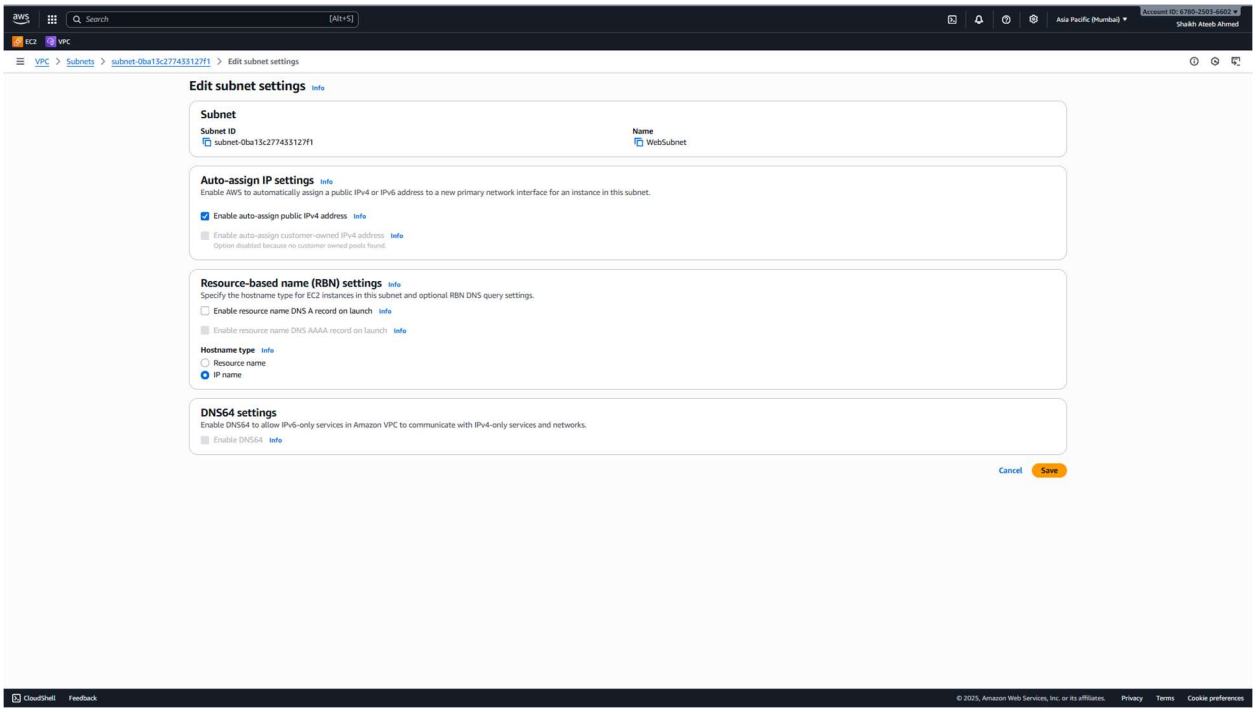



- Click **Create subnet**.

4. Enable auto-assign public IPv4 on the subnet

- Navigation: **VPC → Subnets** → click the row for WebSubnet.
- Actions: **Actions → Edit subnet settings**.
- In the dialog: check **Enable auto-assign public IPv4** → **Save**.

**Shaikh Ateeb Ahmed**
**22-08-2025**

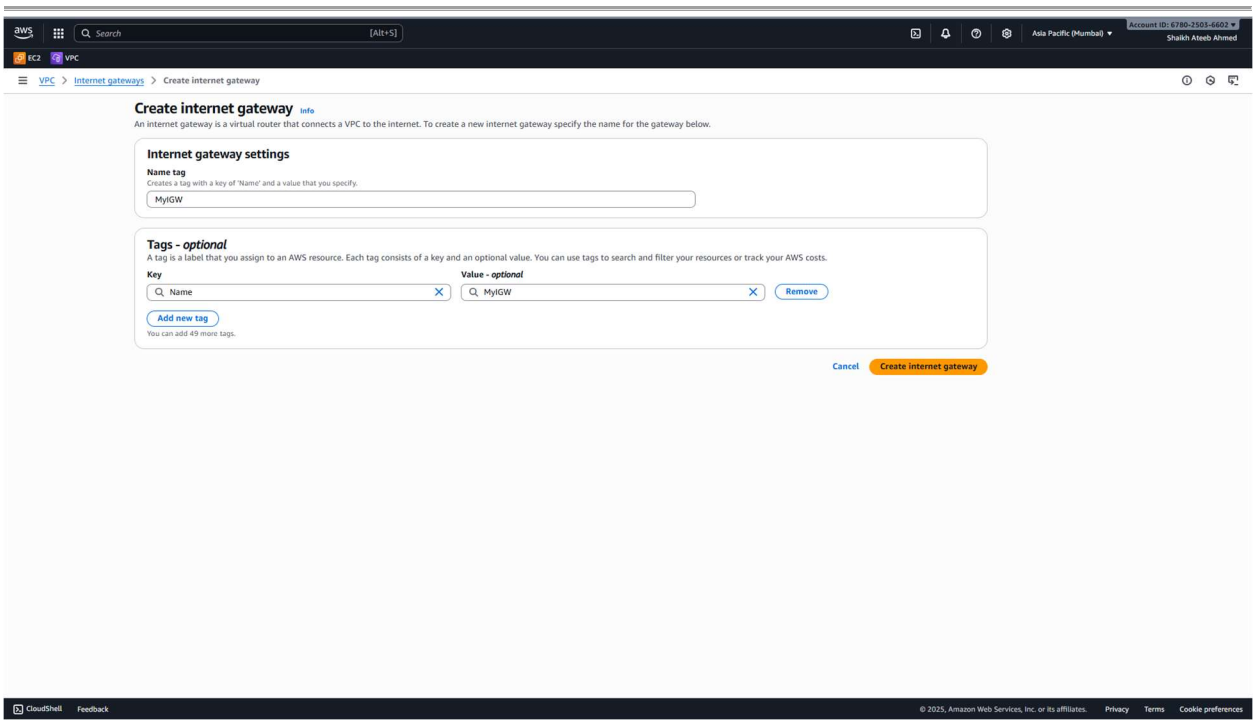



## Step 2: Create an Internet Gateway & Attach to VPC

1. **Open the VPC service in the AWS Management Console.**
2. **Create the Internet Gateway**

- **Navigation: VPC → Internet Gateways → Create internet gateway.**
- **In the dialog: set Name tag to MyIGW.**

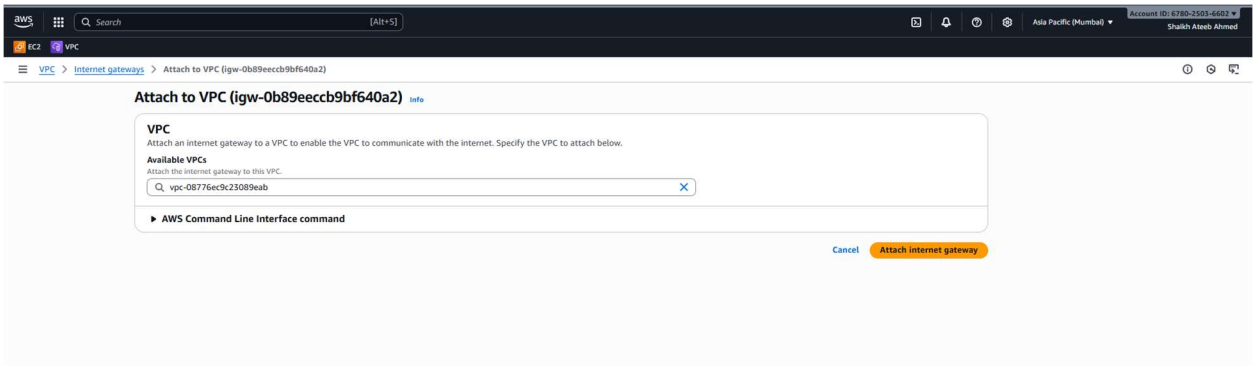
**Shaikh Ateeb Ahmed**
**22-08-2025**




- Click Create internet gateway (or Create).

3. Attach the IGW to your VPC

- Still under Internet Gateways, select the row for MyIGW.
- Click Actions → Attach to VPC.
- Choose MyVPC from the list and click Attach internet gateway.




- Verify the VPC column for MyIGW shows MyVPC (or the VPC ID).

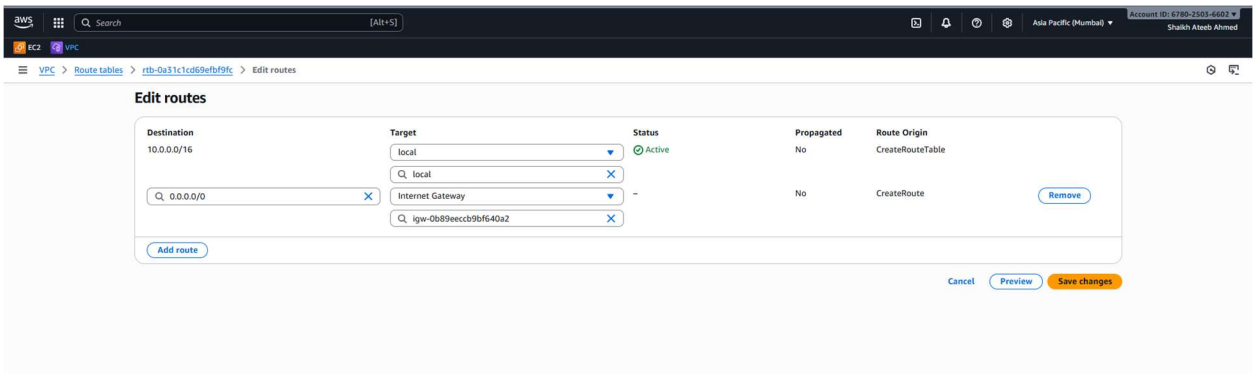
4. Add a route to the IGW

- Navigation: VPC → Route Tables.
- Select the route table that MyVPC uses.

**Shaikh Ateeb Ahmed**
**22-08-2025**

---

- **Click Routes → Edit routes → Add route.**
- **Destination: 0.0.0.0/0**
- **Target: choose Internet gateway and select MyIGW (it will show igw-xxxxx with the name).**




- **Click Save routes.**

---

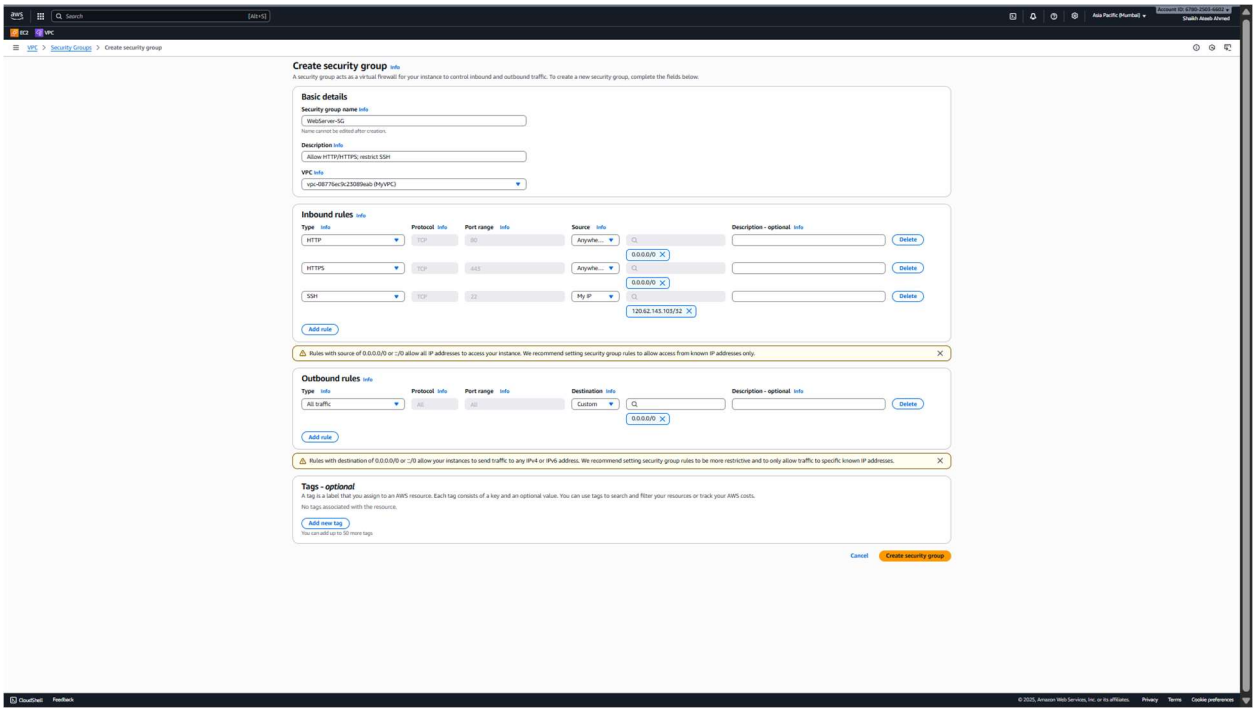
**Step 3: Create a Security Group (SG)**

1. **Open the VPC service in the AWS Console.**
2. **Go to Security Groups**
   - **Navigation: VPC → Security Groups.**
3. **Create the security group**
   - **Click Create security group.**
   - **Name tag: WebServer-SG**
   - **Description: Allow HTTP/HTTPS; restrict SSH**
   - **VPC: select MyVPC**
4. **Add inbound rules**
   - **Add rule for each of the following:**
   - **Rule 1 — Type: HTTP**
   - **Protocol: TCP (auto)**
   - **Port range: 80**
   - **Source: Anywhere → enter 0.0.0.0/0**

**Shaikh Ateeb Ahmed**
**22-08-2025**

---

- **Rule 2 — Type: HTTPS**
- **Protocol: TCP**
- **Port range: 443**
- **Source: Anywhere → 0.0.0.0/0**
- **Rule 3 — Type: SSH**
- **Protocol: TCP**
- **Port range: 22**
- **Source: My IP (recommended) — your public IP.**




- **Click Create security group (or Create).**

5. **Verify outbound rules**

- **By default Security Groups allow all outbound traffic. Confirm Outbound rules tab shows a rule: All traffic → 0.0.0.0/0.**

---

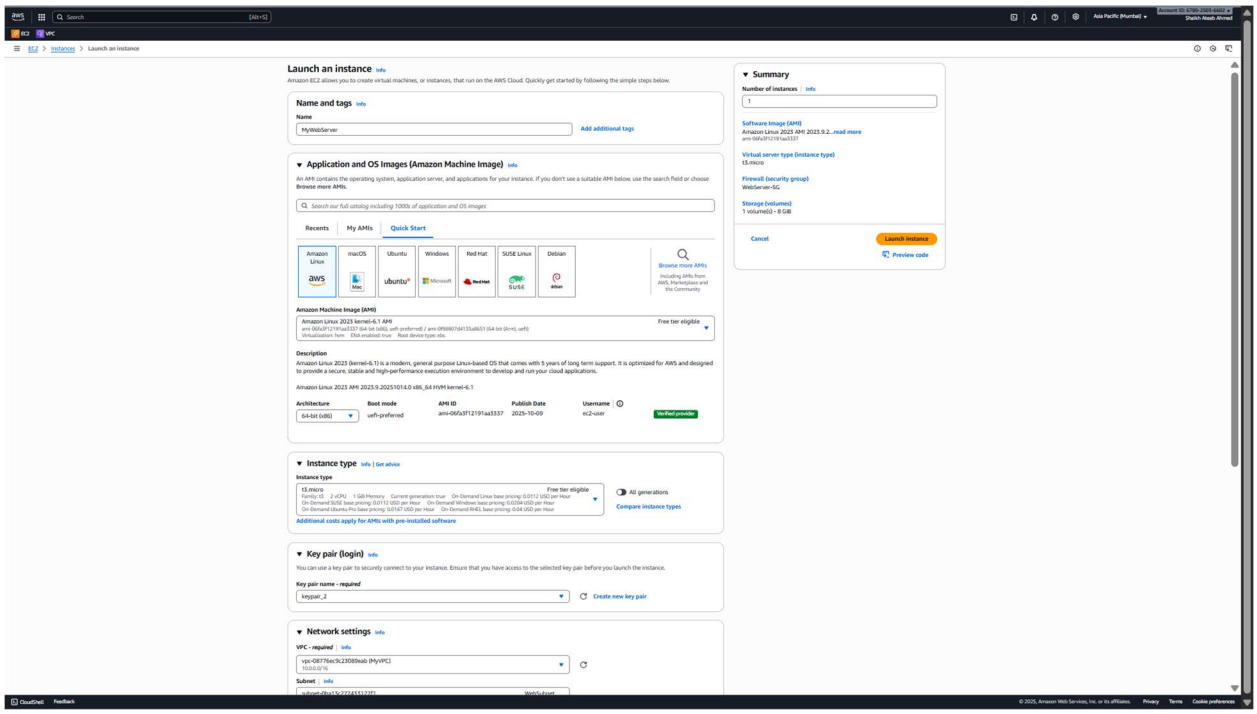
**Step 4: Launch an EC2 instance in the subnet**

1. **EC2 → Launch Instance.**

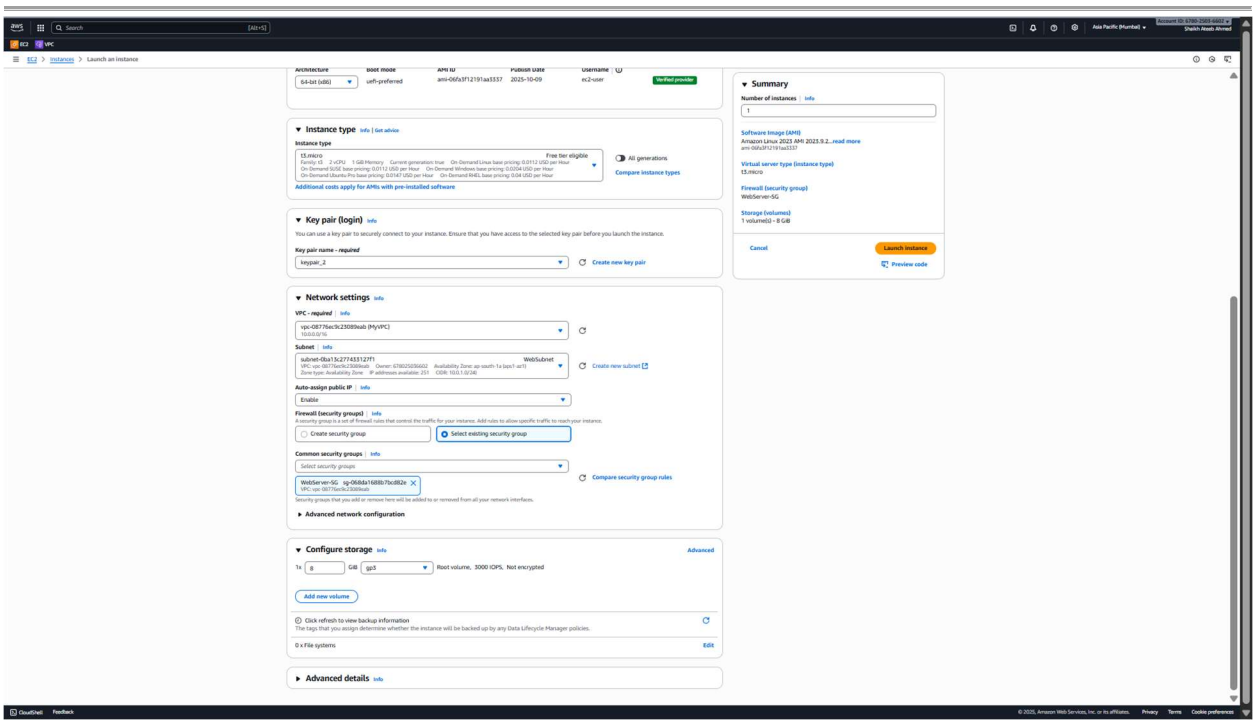
**Shaikh Ateeb Ahmed**
**22-08-2025**

---

- **Name tag: MyWebServer.**
- **AMI: Amazon Linux 2 (free tier).**
- **Instance type: t3.micro.**
- **Network: MyVPC; Subnet: WebSubnet.**
- **Auto-assign Public IP: Enabled (should be set by subnet attribute).**
- **Security Group: choose WebServer-SG.**
- **Key pair: Create new key pair or select existing. Download .pem (store it safely).**



**Shaikh Ateeb Ahmed**
**22-08-2025**



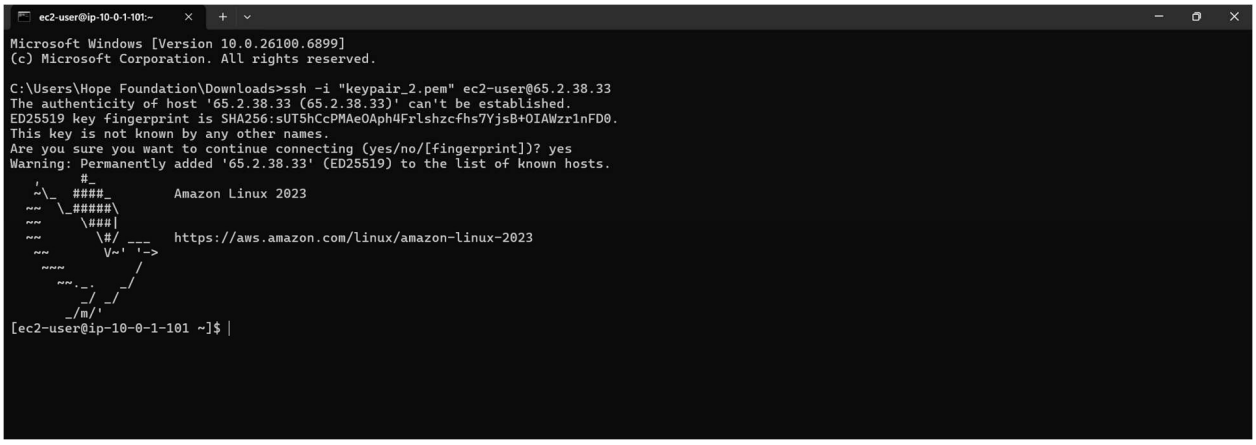

2. **Launch.**

---

## Step 5: Install a Web Server (Apache)

1. **SSH into the instance (replace <public-ip>):**

   **ssh -i "keypair_2.pem" ec2-user@65.2.38.33**



**Shaikh Ateeb Ahmed**
**22-08-2025**

---

2. **On the EC2 instance, update and install Apache:**

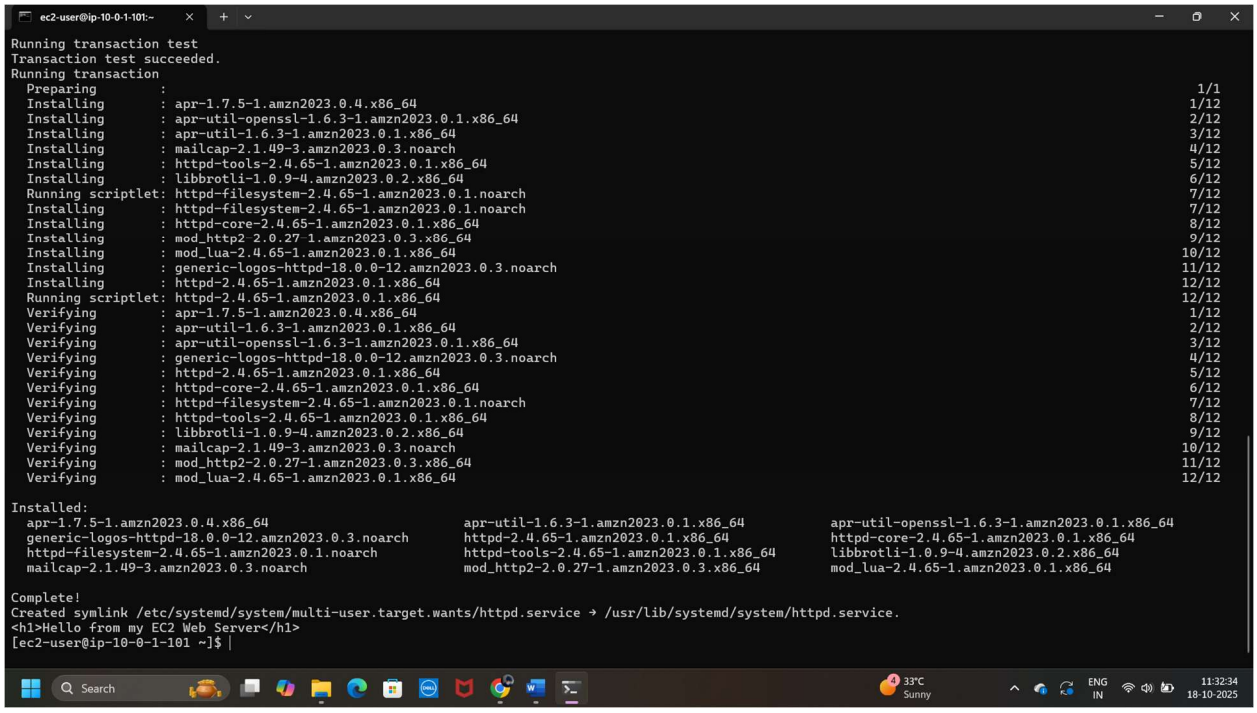
   **sudo yum update -y**

   **sudo yum install -y httpd**

   **sudo systemctl start httpd**

   **sudo systemctl enable httpd**

   **echo "<h1>Hello from my EC2 Web Server</h1>" | sudo tee /var/www/html/index.html**



```
ec2-user@ip-10-0-1-101:~
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                    1/1
  Installing       : apr-1.7.5-1.amzn2023.0.4.x86_64                                                                    1/12
  Installing       : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64                                                       2/12
  Installing       : apr-util-1.6.3-1.amzn2023.0.1.x86_64                                                               3/12
  Installing       : mailcap-2.1.49-3.amzn2023.0.3.noarch                                                               4/12
  Installing       : httpd-tools-2.4.65-1.amzn2023.0.1.x86_64                                                           5/12
  Installing       : libbrotli-1.0.9-4.amzn2023.0.2.x86_64                                                              6/12
  Running scriptlet: httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch                                                      7/12
  Installing       : httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch                                                      7/12
  Installing       : httpd-core-2.4.65-1.amzn2023.0.1.x86_64                                                            8/12
  Installing       : mod_http2-2.0.27-1.amzn2023.0.3.x86_64                                                             9/12
  Installing       : mod_lua-2.4.65-1.amzn2023.0.1.x86_64                                                              10/12
  Installing       : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                                 11/12
  Installing       : httpd-2.4.65-1.amzn2023.0.1.x86_64                                                                12/12
  Running scriptlet: httpd-2.4.65-1.amzn2023.0.1.x86_64                                                                12/12
  Verifying        : apr-1.7.5-1.amzn2023.0.4.x86_64                                                                    1/12
  Verifying        : apr-util-1.6.3-1.amzn2023.0.1.x86_64                                                               2/12
  Verifying        : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64                                                       3/12
  Verifying        : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                                  4/12
  Verifying        : httpd-2.4.65-1.amzn2023.0.1.x86_64                                                                 5/12
  Verifying        : httpd-core-2.4.65-1.amzn2023.0.1.x86_64                                                            6/12
  Verifying        : httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch                                                      7/12
  Verifying        : httpd-tools-2.4.65-1.amzn2023.0.1.x86_64                                                           8/12
  Verifying        : libbrotli-1.0.9-4.amzn2023.0.2.x86_64                                                              9/12
  Verifying        : mailcap-2.1.49-3.amzn2023.0.3.noarch                                                              10/12
  Verifying        : mod_http2-2.0.27-1.amzn2023.0.3.x86_64                                                            11/12
  Verifying        : mod_lua-2.4.65-1.amzn2023.0.1.x86_64                                                              12/12

Installed:
  apr-1.7.5-1.amzn2023.0.4.x86_64                      apr-util-1.6.3-1.amzn2023.0.1.x86_64          apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch    httpd-2.4.65-1.amzn2023.0.1.x86_64            httpd-core-2.4.65-1.amzn2023.0.1.x86_64
  httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch        httpd-tools-2.4.65-1.amzn2023.0.1.x86_64      libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  mailcap-2.1.49-3.amzn2023.0.3.noarch                 mod_http2-2.0.27-1.amzn2023.0.3.x86_64        mod_lua-2.4.65-1.amzn2023.0.1.x86_64

Complete!
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
<h1>Hello from my EC2 Web Server</h1>
[ec2-user@ip-10-0-1-101 ~]$
```

4. **On your browser open: http://<public-ip> → You should see the Hello message.**


**Hello from my EC2 Web Server**

**Shaikh Ateeb Ahmed**
**22-08-2025**

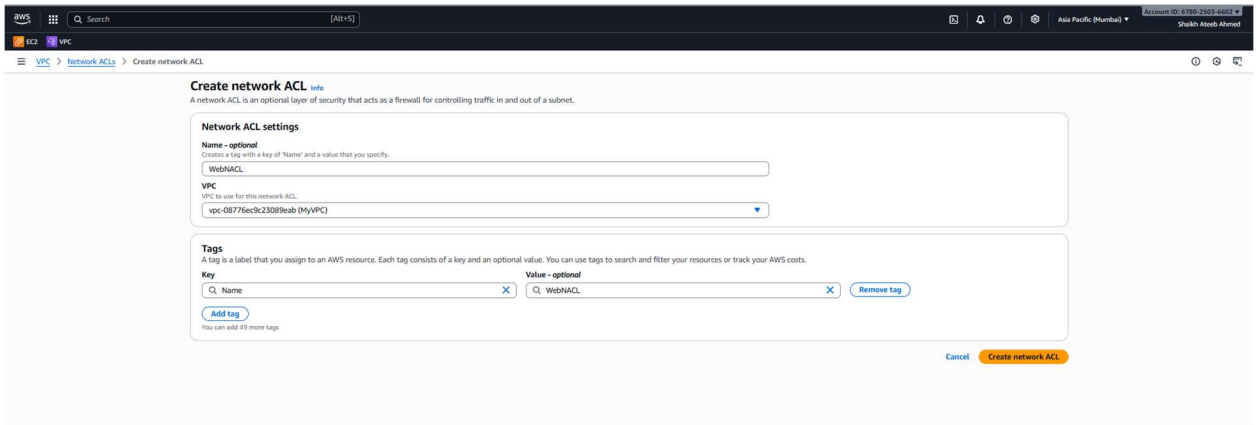
---

---

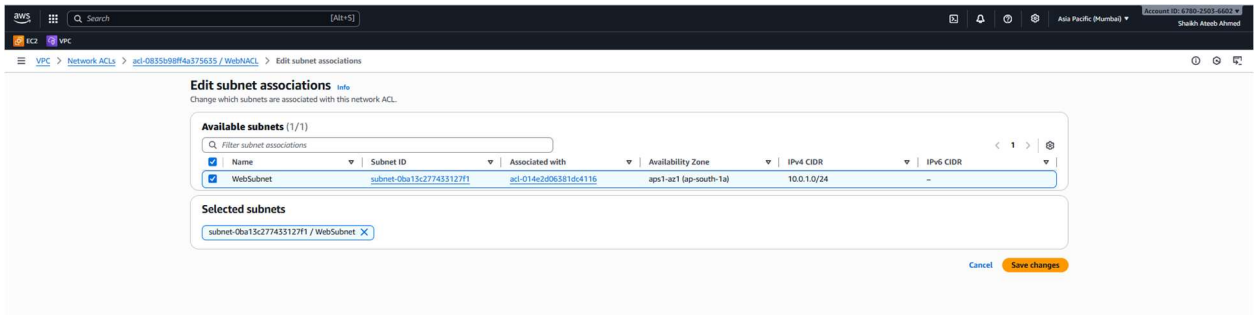
## Step 6: Add NACL Rules

1. **VPC → Network ACLs → Create network ACL.**

- **Name: WebNACL.**
- **VPC: MyVPC.**




2. **Associate the NACL with WebSubnet (Subnet Associations → Edit → add WebSubnet).**

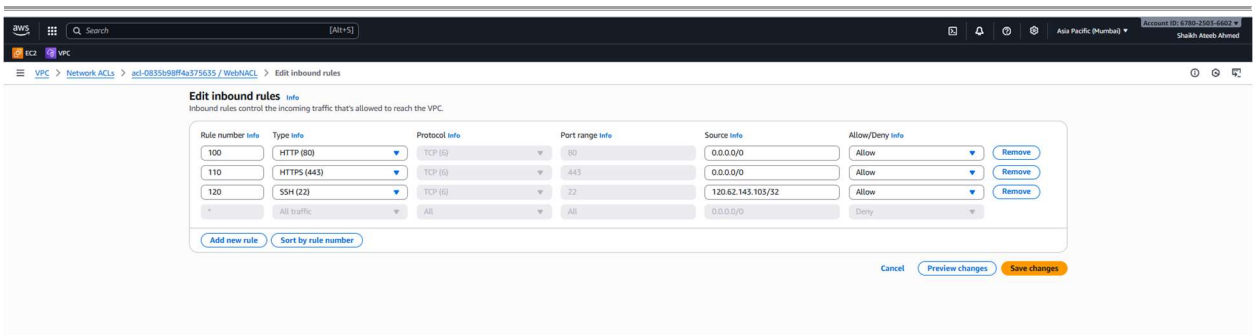



3. **Add entries (NACL uses rule numbers; processed lowest → highest):**

- **Inbound:**

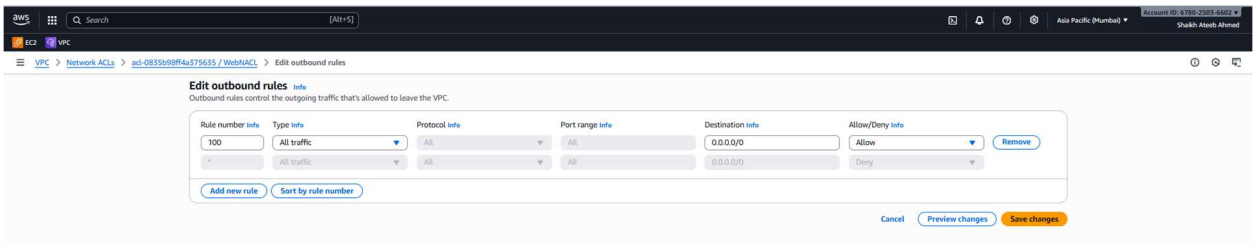
- **Rule 100: Allow TCP 80 (HTTP 80) (Source 0.0.0.0/0)**
- **Rule 110: Allow TCP 443 (HTTPS 443) (Source 0.0.0.0/0)**
- **Rule 120: Allow TCP 22 (SSH 22) (Source YOUR_IP/32) (120.62.143.103/32)**

**Shaikh Ateeb Ahmed**
**22-08-2025**




- **Outbound:**

- **Rule 100: Allow All TCP/UDP/ICMP to 0.0.0.0/0**




4. **Save.**

5. **Test again → Website should still be reachable only on allowed ports.**


**Hello from my EC2 Web Server**