

AWS

Day1

Nada Mohamed Ahmed Hassan Eleshmawy

Mansoura Open Source

Date : 30/5/2025

LAB 1:

To launch your first EC2 instance of Linux OS, install httpd package and run a simple html webpage allowing for public access from anywhere. + VPC

1. Create VPC

1. Go to **VPC** in AWS Console.
2. Click **Create VPC**.
3. Choose: **VPC only**
4. Fill in:
 - **Name:** nada mohamed ahmed vpc
 - **IPv4 CIDR block:** 10.0.0.0/16
 - Leave the rest default
5. Click **Create VPC**

The screenshot displays the AWS Management Console interface for the VPC service. The left sidebar shows the navigation menu with 'VPC' selected. The main content area shows the 'Your VPCs' list, which includes a table with columns for Name, VPC ID, State, Block Public Access, IPv4 CIDR, and IPv6 CIDR. The VPC 'nada mohamed ahmed vpc' is highlighted. Below the table, the details for this VPC are shown, including its ID, state (Available), DNS resolution (Enabled), Main network ACL, IPv6 CIDR (Network border group), Block Public Access (Disabled), DHCP option set, IPv4 CIDR, Route 53 Resolver DNS Firewall rule groups, DNS hostnames (Disabled), Main route table, IPv6 pool, and Owner ID.

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
MyVPC	vpc-0e77ce732b69a31c0	Available	Off	10.0.0.0/16	-
nada mohamed ahmed vpc	vpc-0be378d435a08f41d	Available	Off	10.0.0.0/16	-
-	vpc-099f07hrr4dea5f9d3	Available	Off	172.31.0.0/16	-

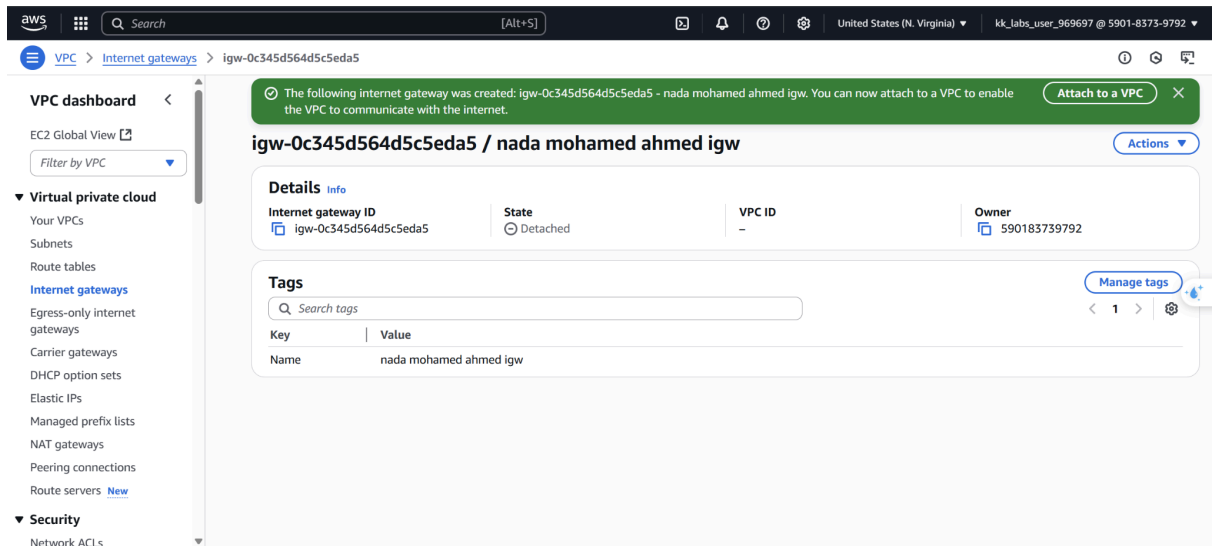
vpc-0be378d435a08f41d / nada mohamed ahmed vpc

Details

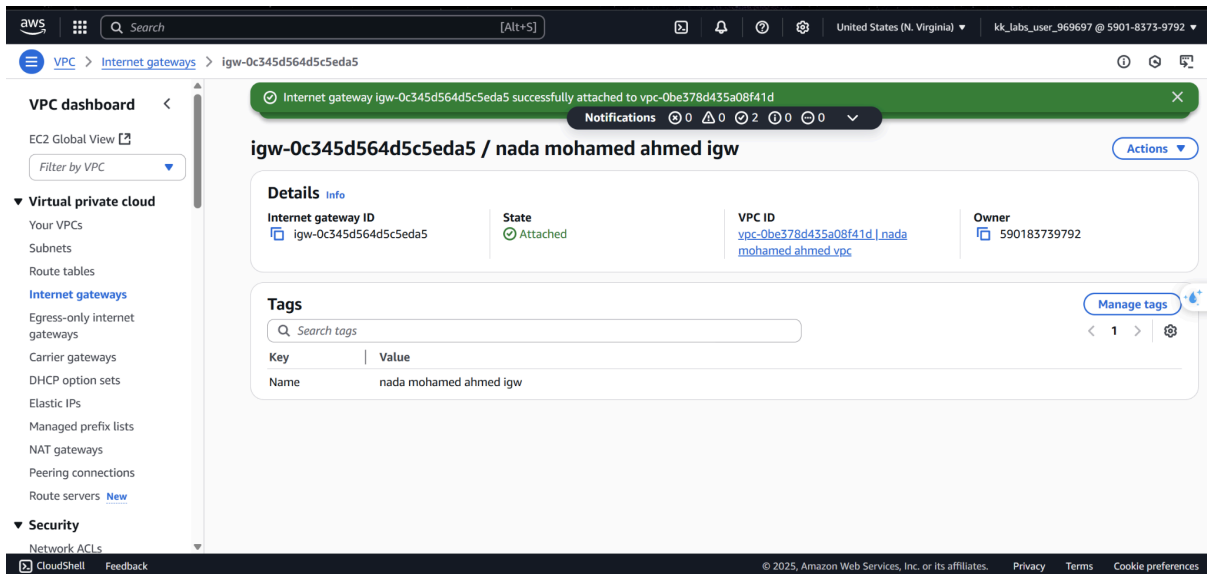
VPC ID vpc-0be378d435a08f41d	State Available	Block Public Access Off	DNS hostnames Disabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-066407fa4e2822341	Main route table rtb-0e12c44c3cea8b724
Main network ACL acl-067c82f59de21094a	Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -
IPv6 CIDR (Network border group) -	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups	Owner ID 590183739792

2. Create Internet Gateway (IGW)

1. From the left menu, click **Internet Gateways**.
2. Click **Create internet gateway**.
3. Name: **nada mohamed ahmed igw**
4. Click **Create**

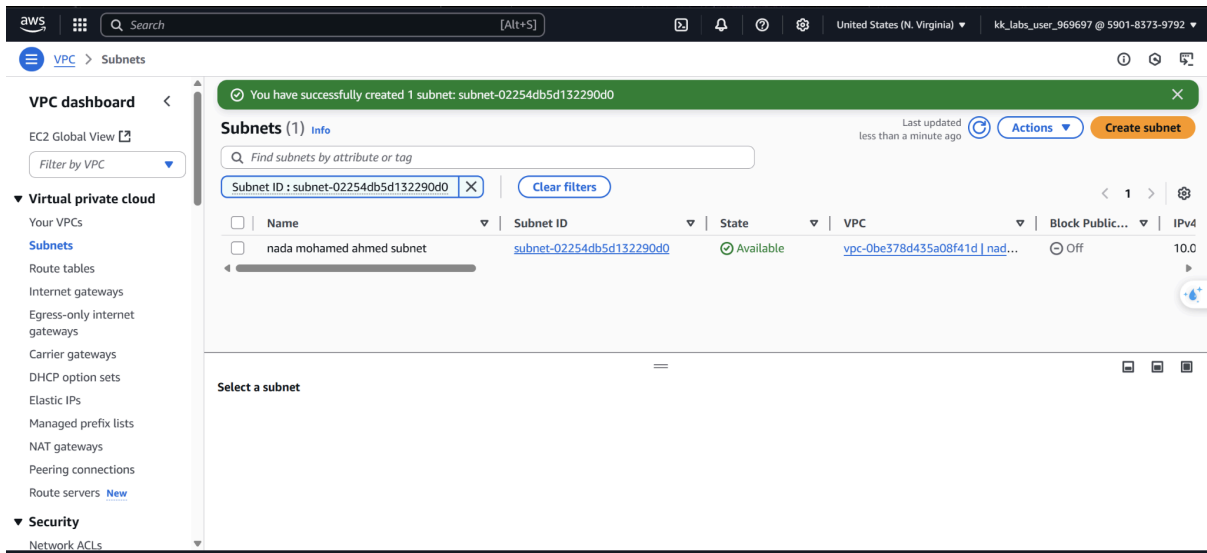


5. Then click **Attach to VPC**
 - Select **nada mohamed ahmed vpc**
 - Click **Attach**



3. Create Subnet

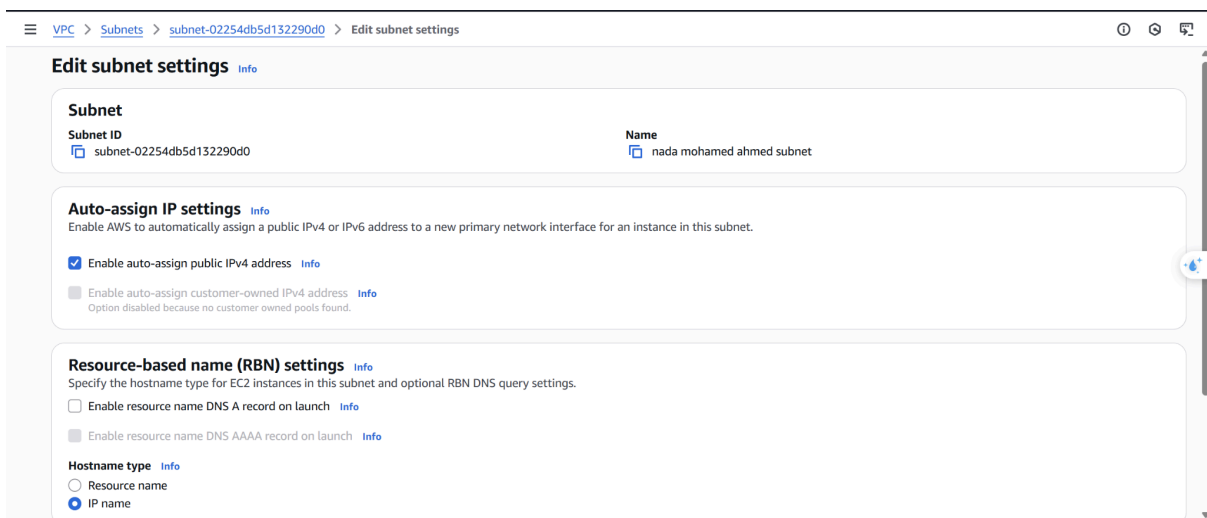
1. Go to **Subnets**
2. Click **Create Subnet**
3. Select:
 - **VPC:** nada mohamed ahmed vpc
 - **Subnet name:** nada mohamed ahmed subnet
 - **Availability Zone:** us-east-1a
 - **CIDR block:** 10.0.1.0/24
4. Click **Create subnet**



5. Select **nada mohamed ahmed subnet**.

6. Click Edit subnet settings.

7. Enable Auto-assign public IPv4 address



aws [Search] [Alt+S] United States (N. Virginia) kk_labs_user_969697 @ 5901-8373-9792

VPC > Subnets > subnet-02254db5d132290d0

VPC dashboard < EC2 Global View [?] Filter by VPC

▼ Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Route servers [New](#)

▼ Security
Network ACLs

You have successfully changed subnet settings:
◦ Enable auto-assign public IPv4 address

subnet-02254db5d132290d0 / nada mohamed ahmed subnet [Actions](#)

Details

Subnet ID subnet-02254db5d132290d0	Subnet ARN arn:aws:ec2:us-east-1:590183739792:subnet/subnet-02254db5d132290d0	State Available	Block Public Access Off
IPv4 CIDR 10.0.1.0/24	Available IPv4 addresses 251	IPv6 CIDR -	IPv6 CIDR association ID -
Availability Zone us-east-1a	Availability Zone ID use1-az6	Network border group us-east-1	VPC vpc-0be378d435a08f41d nada mohamed ahmed vpc
Route table -	Network ACL -	Default subnet No	Auto-assign public IPv4 address Yes
Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No	Customer-owned IPv4 pool -	Outpost ID -
IPv4 CIDR reservations -	IPv6 CIDR reservations -	IPv6-only No	Hostname type IP name
Resource name DNS A record Disabled	Resource name DNS AAAA record -	DNS64 Disabled	Owner 590183739792

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

4. Create Route Table

1. Go to **Route Tables**
2. Click **Create route table**
3. Name: **nada mohamed ahmed route table**
4. VPC: **nada mohamed ahmed vpc**
5. Click **Create**

us-east-1.console.aws.amazon.com/vpconsole/home?region=us-east-1#RouteTableDetails:RouteTableId=rtb-0cd928d17cbdb8d92

aws [Search] [Alt+S] United States (N. Virginia) kk_labs_user_969697 @ 5901-8373-9792

VPC > Route tables > rtb-0cd928d17cbdb8d92

VPC dashboard < EC2 Global View [?] Filter by VPC

▼ Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Route servers [New](#)

▼ Security
Network ACLs

Route table rtb-0cd928d17cbdb8d92 | nada mohamed ahmed route table was created successfully.

rtb-0cd928d17cbdb8d92 / nada mohamed ahmed route table [Actions](#)

Details [Info](#)

Route table ID rtb-0cd928d17cbdb8d92	Main No	Explicit subnet associations -	Edge associations -
VPC vpc-0be378d435a08f41d nada mohamed ahmed vpc	Owner ID 590183739792		

Routes | Subnet associations | Edge associations | Route propagation | Tags

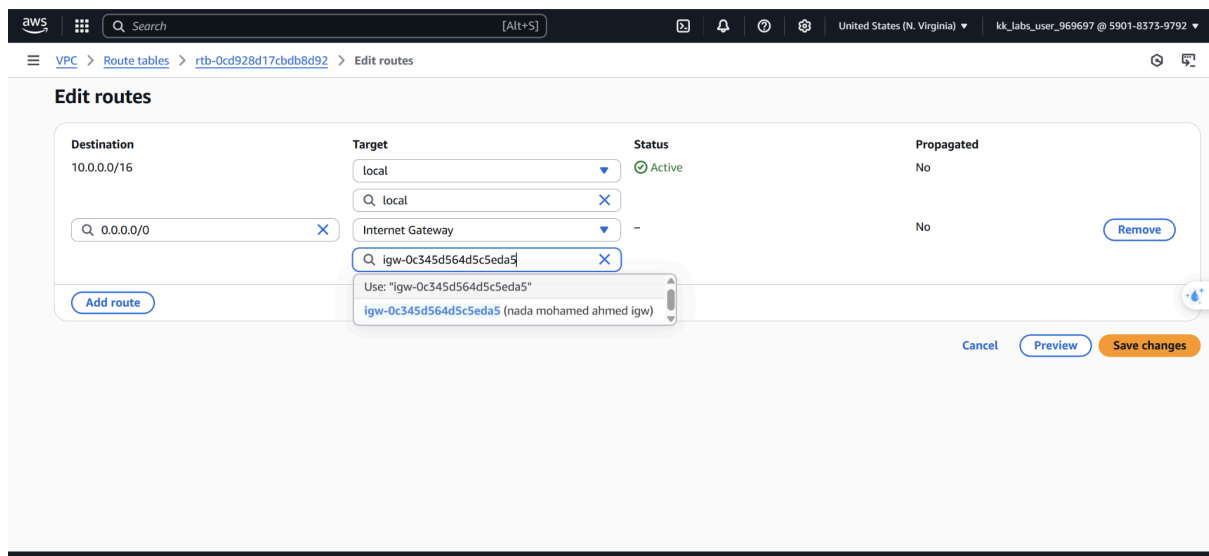
Routes (1) [Filter routes](#) [Both](#) [Edit routes](#)

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

6. After creation:

- Go to **Routes** → **Edit routes**
- Click **Add route**
 - Destination: **0.0.0.0/0**
 - Target: Internet Gateway → **nada mohamed ahmed igw**
- Click **Save changes**



7. Go to Subnet Associations

- Click **Edit subnet associations**
- Select **nada mohamed ahmed subnet**
- Click **Save**

rtb-0cd928d17cddb8d92 / nada mohamed ahmed route table

Details

Route table ID rtb-0cd928d17cddb8d92	Main No	Explicit subnet associations subnet-02254db5d132290d0 / nada mohamed ahmed subnet	Edge associations -
VPC vpc-0be378d435a08f41d nada mohamed ahmed vpc	Owner ID 590183739792		

Subnet associations

Explicit subnet associations (1)

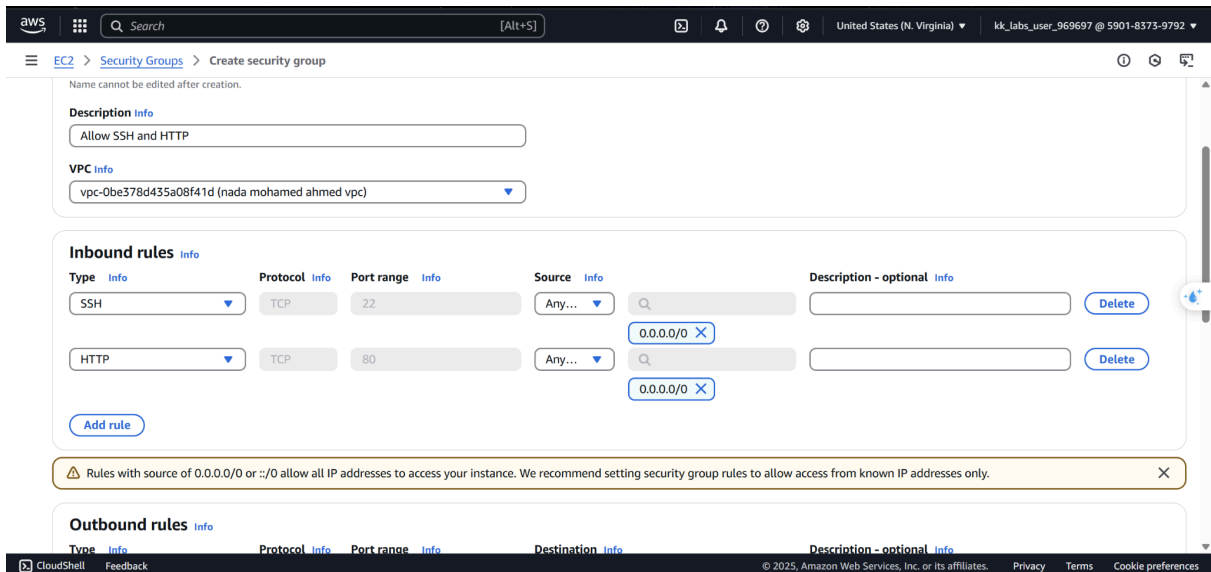
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
nada mohamed ahmed subnet	subnet-02254db5d132290d0	10.0.1.0/24	-

Subnets without explicit associations (0)

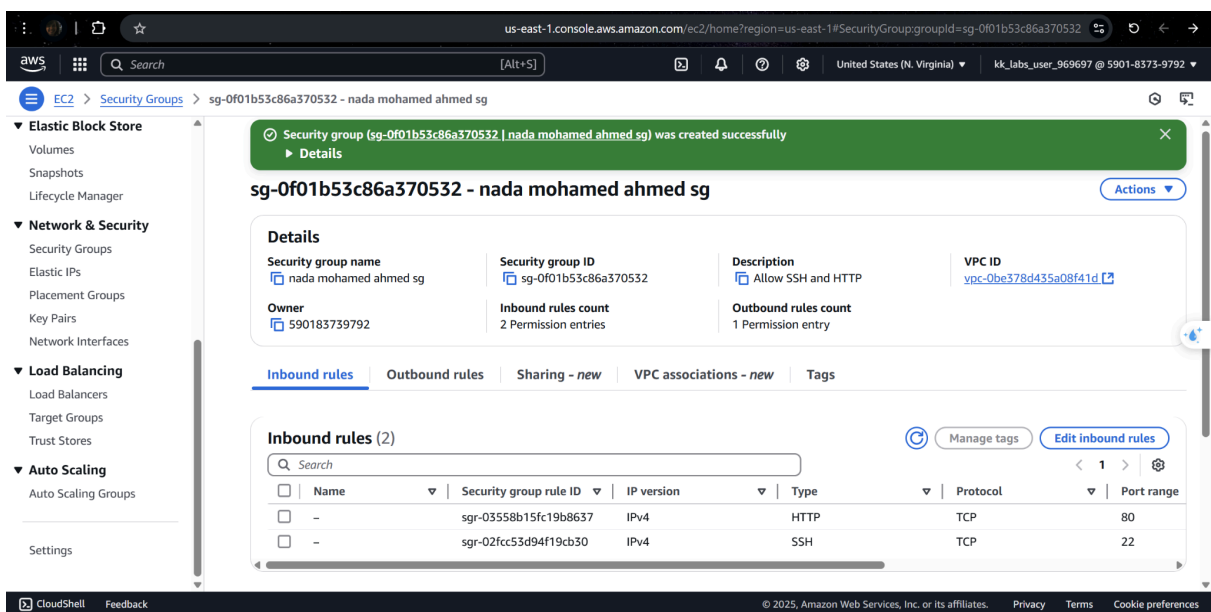
The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

5. Create Security Group

1. Go to **EC2** → **Security Groups**
2. Click **Create security group**
3. Fill:
 - Name: **nada mohamed ahmed sg**
 - Description: **Allow SSH and HTTP**
 - VPC: **nada mohamed ahmed vpc**
4. Under **Inbound rules**, click **Add rule**:
 - Type: **SSH** | Port: **22** | Source: **Anywhere (0.0.0.0/0)**
 - Type: **HTTP** | Port: **80** | Source: **Anywhere (0.0.0.0/0)**



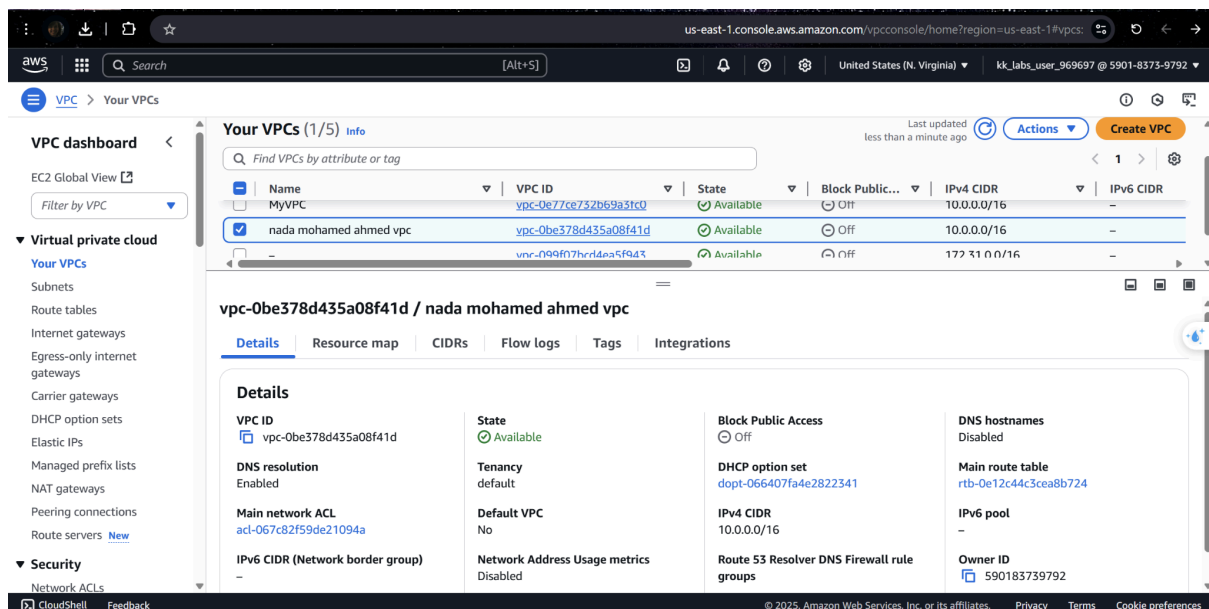
5. Click **Create security group**



6. Launch EC2 Instance

1. Go to **EC2** → **Instances** → **Launch instance**
2. Name: **nada mohamed ahmed ec2**
3. Amazon Machine Image (AMI): **Amazon Linux 2023**

4. Instance Type: **t2.micro**
5. Key pair: create (**nada-key**)
6. Network settings:
 - VPC: **nada mohamed ahmed vpc**
 - Subnet: **nada mohamed ahmed subnet**
 - Auto-assign Public IP: **Enable**
 - Firewall: Select **Existing security group** → **nada mohamed ahmed sg**
7. Click **Launch instance**



7. Connect to EC2 via SSH

Use terminal or Git Bash:

```
ssh -i "nada-key.pem" ec2-user@54.80.198.241
```

```
nadam@NadaMohamed MINGW64 ~/Downloads/open_source/AWS/day1/lab
○ $ ssh -i "nada-key.pem" ec2-user@54.80.198.241
The authenticity of host '54.80.198.241 (54.80.198.241)' can't be established.
ED25519 key fingerprint is SHA256:5llcUu9LcCovA1h1+mBLtgC0Q9DpVRgf390JPRCuLOA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.80.198.241' (ED25519) to the list of known hosts.
```

The screenshot shows a Windows command prompt where a user named nadam connects via SSH to an EC2 instance at IP 54.80.198.241 using a private key named nada-key.pem. The terminal output displays the standard SSH warning about the host's fingerprint, which the user accepts. After the connection is established, the user runs a series of commands to set up a development environment: they update the system packages, install Node.js and npm, create a new directory named 'open-source', move into it, and initialize a new Git repository. Finally, they clone the 'aws/aws-cdk-js' repository from GitHub. The terminal also shows the user navigating through their file system, listing directories like 'cdk' and 'node_modules', and checking the contents of a file named 'README.md'.

```
'_#_
~\_#####_      Amazon Linux 2023
~~\#####\
~~\###|
~~\#/_____
~~V~' '-> https://aws.amazon.com/linux/amazon-linux-2023
~~~~
~~~.-.
~/ \_/ \_/
_/m/'
```

Only ensure there is internet connection

Ping google.com

```
[ec2-user@ip-10-0-1-144 ~]$ Ping google.com
-bash: Ping: command not found
[ec2-user@ip-10-0-1-144 ~]$ ping google.com
PING google.com (64.233.180.113) 56(84) bytes of data.
64 bytes from on-in-f113.1e100.net (64.233.180.113): icmp_seq=1 ttl=106 time=2.18 ms
64 bytes from on-in-f113.1e100.net (64.233.180.113): icmp_seq=2 ttl=106 time=2.56 ms
64 bytes from on-in-f113.1e100.net (64.233.180.113): icmp_seq=3 ttl=106 time=2.23 ms
```

8. Install and Configure Apache (httpd)

Run the following commands:

Update packages

```
sudo yum update -y
```

```
[ec2-user@ip-10-0-1-144 ~]$ sudo yum update -y
Amazon Linux 2023 Kernel Livepatch repository                143 kB/s | 16 kB      00:00
Dependencies resolved.
Nothing to do.
Complete!
```

Install Apache

```
sudo yum install -y httpd
```

```
[ec2-user@ip-10-0-1-144 ~]$ sudo yum install -y httpd
Last metadata expiration check: 0:00:15 ago on Fri May 30 14:22:10 2025.
Dependencies resolved.
=====
Package                                Architecture    Version                                Repository      Size
=====
Installing:
httpd                                   x86_64          2.4.62-1.amzn2023                     amazonlinux      48 k
Installing dependencies:
apr                                     x86_64          1.7.5-1.amzn2023.0.4                  amazonlinux      129 k
apr-util                               x86_64          1.6.3-1.amzn2023.0.1                  amazonlinux      98 k
generic-logos-httpd                   noarch          18.0.0-12.amzn2023.0.3                amazonlinux      19 k
httpd-core                             x86_64          2.4.62-1.amzn2023                     amazonlinux      1.4 M
httpd-filesystem                      noarch          2.4.62-1.amzn2023                     amazonlinux      14 k
httpd-tools                            x86_64          2.4.62-1.amzn2023                     amazonlinux      81 k
libbrotli                              x86_64          1.0.9-4.amzn2023.0.2                  amazonlinux      315 k
mailcap                                noarch          2.1.49-3.amzn2023.0.3                amazonlinux      33 k
Installing weak dependencies:
apr-util-openssl                      x86_64          1.6.3-1.amzn2023.0.1                  amazonlinux      17 k
mod_http2                             x86_64          2.0.27-1.amzn2023.0.3                amazonlinux      166 k
mod_lua                                x86_64          2.4.62-1.amzn2023                     amazonlinux      61 k
Transaction Summary
=====
Install 12 Packages
```

Start Apache service

```
sudo systemctl start httpd
```

Enable it on reboot

```
sudo systemctl enable httpd
```

```
[ec2-user@ip-10-0-1-144 ~]$ sudo systemctl start httpd
[ec2-user@ip-10-0-1-144 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
```

Create a simple webpage

```
echo "<h1>Hello from Nada Mohamed Ahmed EC2 instance\!</h1>" | sudo tee /var/www/html/index.html
```

```
[ec2-user@ip-10-0-1-144 ~]$ echo "<h1>Hello from Nada Mohamed Ahmed EC2 instance\!</h1>" | sudo tee /var/www/html/index.html
<h1>Hello from Nada Mohamed Ahmed EC2 instance\!</h1>
[ec2-user@ip-10-0-1-144 ~]$
```

And in Browser

<http://54.80.198.241/>



Hello from Nada Mohamed Ahmed EC2 instance\!



For Delete:

- We must delete resources in the correct order. For example we **cannot delete a VPC if it still has subnets, gateways, or active resources.**

Step 1: Terminate our EC2 instance

1. Go to **EC2 Dashboard** in AWS Console.
 2. Click **Instances** > find your instance
 3. Select the instance > click **Actions** > **Instance State** > **Terminate instance.**
 4. Confirm termination.
-

Step 2: Delete Security Groups

1. Go to **VPC Dashboard** (or EC2 > Security Groups).
 2. Find the security group named **nada mohamed ahmed sg.**
 3. Select and **Delete Security Group.**
-

Step 3: Delete Subnets

1. In the **VPC Dashboard**, go to **Subnets.**
 2. Find your subnet named **nada mohamed ahmed subnet.**
 3. Select it and click **Delete subnet.**
-

Step 5: Delete Route Tables (except main)

1. Go to **VPC Dashboard** > **Route Tables.**
2. Find the route table named **nada mohamed ahmed route table.**

Step 6: Detach and Delete Internet Gateway

1. In the **VPC Dashboard**, go to **Internet Gateways**.
2. Find the IGW attached to our VPC.
3. Select it > click **Actions** > **Detach from VPC**.
4. Once detached, select again and **Delete Internet Gateway**.

Step 7: Delete VPC

1. Go to **VPC Dashboard** > **Your VPCs**.
2. Select your VPC named **nada mohamed ahmed vpc**.
3. Click **Delete VPC**.