

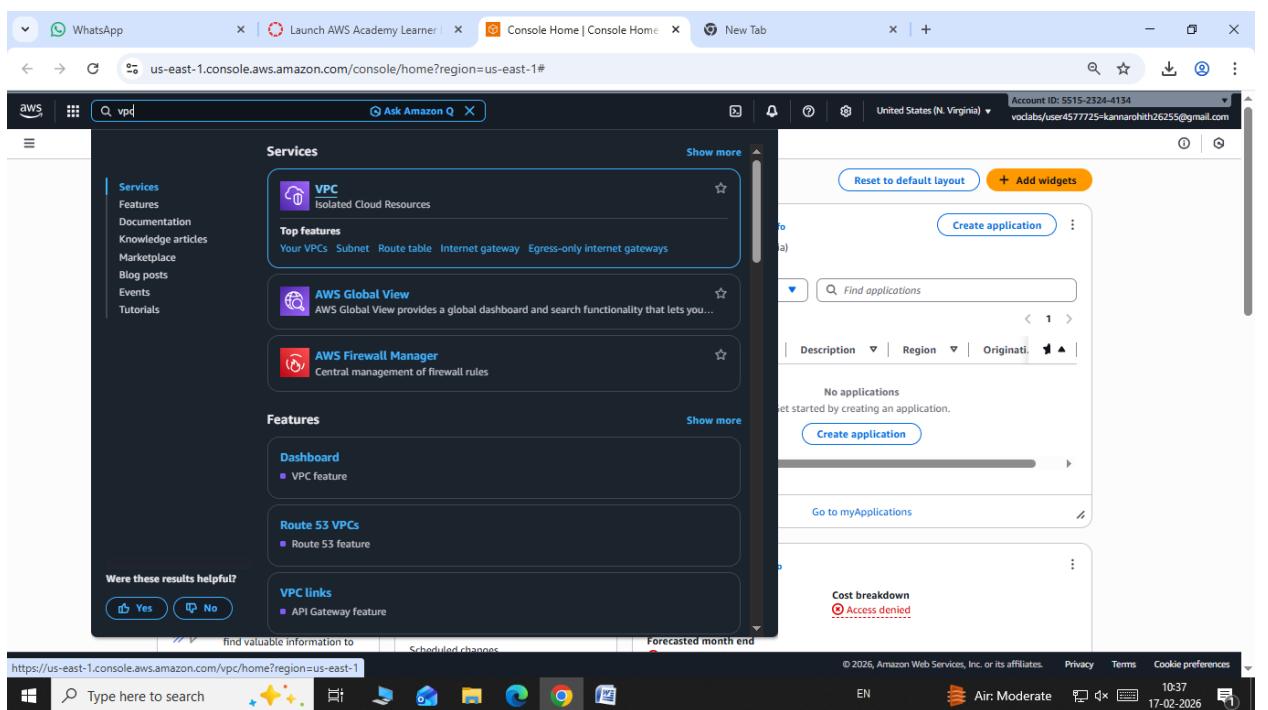
Name: k.Rohith

Rollno.: 23bd1a1291

## Week 5: VPC Creation and Ec2 Instance Connection

### ◆Step 1: Create the VPC

1. Go to AWS Management Console → VPC
2. Click **Create VPC**
3. Choose **VPC only**
4. Enter:
  - o **Name:** Custom-VPC
  - o **IPv4 CIDR block:** e.g. 10.0.0.0/16
5. Click **Create VPC**



The screenshot shows the 'Create VPC' configuration page in the AWS Cloud console. The 'VPC settings' section is visible, including fields for 'IPv4 CIDR block' (set to 10.0.0.0/16) and 'Tenancy' (set to 'Default'). The 'IPv6 CIDR block' section is collapsed. The 'Tags' section shows a single tag named 'Name' with value 'custom-VPN'. The bottom right corner features a large orange 'Create VPC' button.

## ◆ Step 2: Create Subnets (Public & Private)

Create subnets in same **Availability Zones** for high availability.

**Example:**

- **Public Subnet:** 10.0.1.0/24 (AZ-A)
- **Private Subnet:** 10.0.2.0/24 (AZ-A)

Steps:

1. Go to **Subnets** → **Create subnet**
2. Select your VPC
3. Choose AZ
4. Enter CIDR
5. Create subnet

The screenshot shows the AWS VPC Subnets page. On the left, there's a navigation sidebar with options like VPC dashboard, AWS Global View, Virtual private cloud (with Subnets selected), Security, PrivateLink and Lattice, CloudShell, and Feedback. The main area displays a table titled "Subnets (6) info" with the following data:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR
-	subnet-0dc9b8d069b931153	Available	vpc-08e6f3ebe3f57f6d6	Off	172.31.80.0/20	-
-	subnet-0ca26ba5d3fb87451	Available	vpc-08e6f3ebe3f57f6d6	Off	172.31.32.0/20	-
-	subnet-0d7a03052448b6930	Available	vpc-08e6f3ebe3f57f6d6	Off	172.31.0.0/20	-
-	subnet-0050e2917f24e87fa	Available	vpc-08e6f3ebe3f57f6d6	Off	172.31.16.0/20	-
-	subnet-028fb647ba46c845	Available	vpc-08e6f3ebe3f57f6d6	Off	172.31.64.0/20	-
-	subnet-07386b25fcfa765066	Available	vpc-08e6f3ebe3f57f6d6	Off	172.31.48.0/20	-

Below the table, there's a section titled "Select a subnet". At the bottom of the page, there are links for "CloudShell", "Feedback", and "Type here to search". The status bar at the bottom right shows "EN", "PM Modi inaugurates...", "10:43", "17-02-2026", and a battery icon.

Screenshot of the AWS VPC Subnet creation interface in a browser window.

**VPC ID:** vpc-09d577c85b87e742 (custom-VPC)

**Associated VPC CIDRs:** 10.0.0.0/16

**Subnet settings:**

- Subnet name:** cc\_vpc\_subnet
- Availability Zone:** United States (N. Virginia) / us-east-1a
- IPv4 VPC CIDR block:** 10.0.0.0/16
- IPv4 subnet CIDR block:** 10.0.1.0/24

**Tags - optional:**

- Key: Name, Value: public\_subnet

**Create subnet** button at the bottom right.

Screenshot of the AWS VPC Subnet creation interface in a browser window.

**VPC ID:** vpc-09d577c85b87e742 (custom-VPC)

**Associated VPC CIDRs:** 10.0.0.0/16

**Subnet settings:**

- Subnet name:** public\_subnet
- Availability Zone:** United States (N. Virginia) / us-east-1a
- IPv4 VPC CIDR block:** 10.0.0.0/16
- IPv4 subnet CIDR block:** 10.0.1.0/24

**Tags - optional:**

- Key: Name, Value: public\_subnet

**Add new tag** and **Remove** buttons.

**Add new subnet** button at the bottom left.

**Create subnet** button at the bottom right.

Private:

The screenshot shows the 'Create subnet' wizard in the AWS VPC console. In the 'VPC' section, the VPC ID is set to 'vpc-0a9d577c85b87e742 (custom-VPC)'. Under 'Associated VPC CIDRs', the IPv4 CIDR is specified as '10.0.0.0/16'. The 'Subnet settings' section contains 'Subnet 1 of 1' with a 'Subnet name' of 'cc\_vpc\_subnet2' and an 'Availability Zone' of 'United States (N. Virginia) / us-east-1a'. The 'IPv4 VPC CIDR block' is set to '10.0.0.0/16'. A 'Tags - optional' section includes a tag named 'Name' with the value 'private\_subnet'. The bottom right corner features a 'Create subnet' button.

## ◆ Step 3: Create an Internet Gateway (IGW)

1. Go to **Internet Gateways**
2. Click **Create internet gateway**

3. Name it Custom-IGW
4. Attach it to your VPC

**Create internet gateway** Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

**Internet gateway settings**

**Name tag**  
Creates a tag with a key of 'Name' and a value that you specify.

**Tags - optional**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="custom_IGW"/> <span style="color: blue;">X</span> <span style="color: blue;">Remove</span>

**Add new tag**  
You can add 49 more tags.

Cancel Create internet gateway

**Attach to VPC (igw-0d40e36688e756ab4)** Info

**VPC**  
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

**Available VPCs**  
Attach the internet gateway to this VPC.  
 X

**AWS Command Line Interface command**

Cancel Attach internet gateway

CloudShell Feedback

Type here to search

## ◆ Step 4: Create Route Tables

You need **separate route tables** for public and private subnets.

## Public Route Table

1. Go to **Route Tables** → **Create route table**
2. Select VPC
3. Add route:
  - o Destination: 0.0.0.0/0
  - o Target: **Internet Gateway**
4. Associate with **public subnet**

The screenshot shows the 'Create route table' wizard in the AWS VPC console. The 'Route table settings' step is active. It includes fields for 'Name - optional' (set to 'public\_route\_table'), 'VPC' (set to 'vpc-0a9d577c85b87e742 (custom-VPC)'), and 'Tags'. A single tag is added with key 'Name' and value 'public\_route\_table'. The 'Create route table' button is at the bottom right.

**Create route table** Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

**Route table settings**

**Name - optional**  
Create a tag with a key of 'Name' and a value that you specify.

**VPC**  
The VPC to use for this route table.

**Tags**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="public_route_table"/> <span style="color: blue;">Remove</span>

Add new tag  
You can add 49 more tags.

Cancel Create route table

CloudShell Feedback © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences EN 23°C Sunny 11:08 17-02-2026

Screenshot of the AWS VPC Route Table Details page for route table rtb-098657d0c8f22f82b.

**Details**

Route table ID	rtb-098657d0c8f22f82b
Main	No
VPC	vpc-0a9d577c85b87e742   custom-VPC
Owner ID	551523244134

**Actions**

- Set main route table
- Edit subnet associations
- Edit edge associations
- Edit route propagation
- Edit routes
- Manage tags
- Delete

**Routes**

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	CreateRouteTable

Screenshot of the AWS VPC Edit Routes page for route table rtb-098657d0c8f22f82b.

**Edit routes**

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	-	No	CreateRoute
0.0.0.0/0	igw-0d40e36680e756ab	-	-	-

**Actions**

- Add route
- Remove

**Buttons**

- Cancel
- Preview
- Save changes



The screenshot shows the AWS VPC Route Tables page for a public route table named 'rtb-098657d0c8f22f82b'. The 'Routes' tab is selected, displaying two entries:

Destination	Target	Status	Propagated	Route Origin
0.0.0.0/0	igw-0d40c3668be756ab4	Active	No	Create Route
10.0.0.16	local	Active	No	Create Route Table

The 'Actions' menu on the right includes options like Set main route table, Edit subnet associations, Edit edge associations, Edit route propagation, Edit routes, Manage tags, and Delete.

The screenshot shows the 'Edit subnet associations' page for the same route table. It lists available subnets and selected subnets. The 'Selected subnets' section contains one item: 'subnet-062d708d1e84c0504 / public\_subnet'.

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
public_subnet	subnet-062d708d1e84c0504	10.0.1.0/24	-	Main (rtb-0aec5f45a9894086)
private_subnet	subnet-0ea466fb8c8c81453	10.0.2.0/24	-	Main (rtb-0aec5f45a9894086)

Buttons at the bottom include 'Cancel' and 'Save associations'.

## Private Route Table

- Keep default local route only (no IGW)
- Associate with **private subnet**

WhatsApp | Launch AWS Academy Learner | VPC | us-east-1

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateRouteTable:

aws | Search [Alt+S] | United States (N. Virginia) | Account ID: 5515-2324-4134 vodlabs/user4577725=kannaroth26255@gmail.com

VPC > Route tables > Create route table

### Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

**Route table settings**

**Name - optional**  
Create a tag with a key of 'Name' and a value that you specify.

**VPC**  
The VPC to use for this route table.

**Tags**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="private_route_table"/> <span style="color: blue;">X</span> <span style="color: blue;">Remove</span>
<a href="#">Add new tag</a>	

You can add 49 more tags.

[Cancel](#) [Create route table](#)

CloudShell Feedback © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences EN 23°C Sunny 11:12 17-02-2026

WhatsApp | Launch AWS Academy Learner | VPC | us-east-1

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#RouteTableDetails:RouteTableId=rtb-0e211c732d711dc87

aws | Search [Alt+S] | United States (N. Virginia) | Account ID: 5515-2324-4134 vodlabs/user4577725=kannaroth26255@gmail.com

VPC > Route tables > rtb-0e211c732d711dc87

**Route table rtb-0e211c732d711dc87 / private\_route\_table was created successfully.**

**rtb-0e211c732d711dc87 / private\_route\_table**

**Details** Info

Route table ID <a href="#">rtb-0e211c732d711dc87</a>	Main <input type="checkbox"/> No	Explicit subnet associations -	Edge associations -
VPC <a href="#">vpc-0a9d577c85b87e742   custom-VPC</a>	Owner ID <a href="#">551523244134</a>	<a href="#">Actions ▾</a> <ul style="list-style-type: none"> <li><a href="#">Set main route table</a></li> <li><a href="#">Edit subnet associations</a></li> <li><a href="#">Edit edge associations</a></li> <li><a href="#">Edit route propagation</a></li> <li><a href="#">Edit routes</a></li> <li><a href="#">Manage tags</a></li> <li><a href="#">Delete</a></li> </ul>	

**Routes** (1)

Destination	Target	Status	Propagated	Route Origin
10.0.0.1/16	local	Active	No	<a href="#">Create Route Table</a>

[Both](#) [Edit routes](#)

CloudShell Feedback © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences EN 23°C Sunny 11:12 17-02-2026

**Edit subnet associations**

Change which subnets are associated with this route table.

Available subnets (1/2)					
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID	
public_subnet	subnet-062d708d1e84c0504	10.0.1.0/24	-	rtb-098657d0c8f22f82b / public_route...	
<input checked="" type="checkbox"/> private_subnet	subnet-0ea466f88c8c81453	10.0.2.0/24	-	Main (rtb-0aec5f45a9894086)	

**Selected subnets**

subnet-0ea466f88c8c81453 / private\_subnet X

Cancel **Save associations**



## ◆ Step 5: Enable Auto-Assign Public IP (Public Subnet)

1. Select **Public Subnet**
2. Go to **Edit subnet settings**
3. Enable **Auto-assign public IPv4 address**

**Subnets (1/8) Info**

Name	Subnet ID	State	VPC	Block Public Access
-	subnet-0dc9b8d069b931153	Available	vpc-08e6f3eb3f57f6d6	Off
-	subnet-0ca26ba5d3fb87451	Available	vpc-08e6f3eb3f57f6d6	Off
-	subnet-0d7ab03052448b6930	Available	vpc-08e6f3eb3f57f6d6	Off
-	subnet-0050e2917f24e87f9	Available	vpc-08e6f3eb3f57f6d6	Off
<input checked="" type="checkbox"/> public_subnet	subnet-062d708d1e84c0504	Available	vpc-0a9d577e85b87e742   cust...	Off
private_subnet	subnet-0ea466f88c8c81453	Available	vpc-0a9d577e85b87e742   cust...	Off
-	subnet-028fb6479a46c845	Available	vpc-08e6f3eb3f57f6d6	Off
-	subnet-0738693fcfa765066	Available	vpc-08e6f3eb3f57f6d6	Off

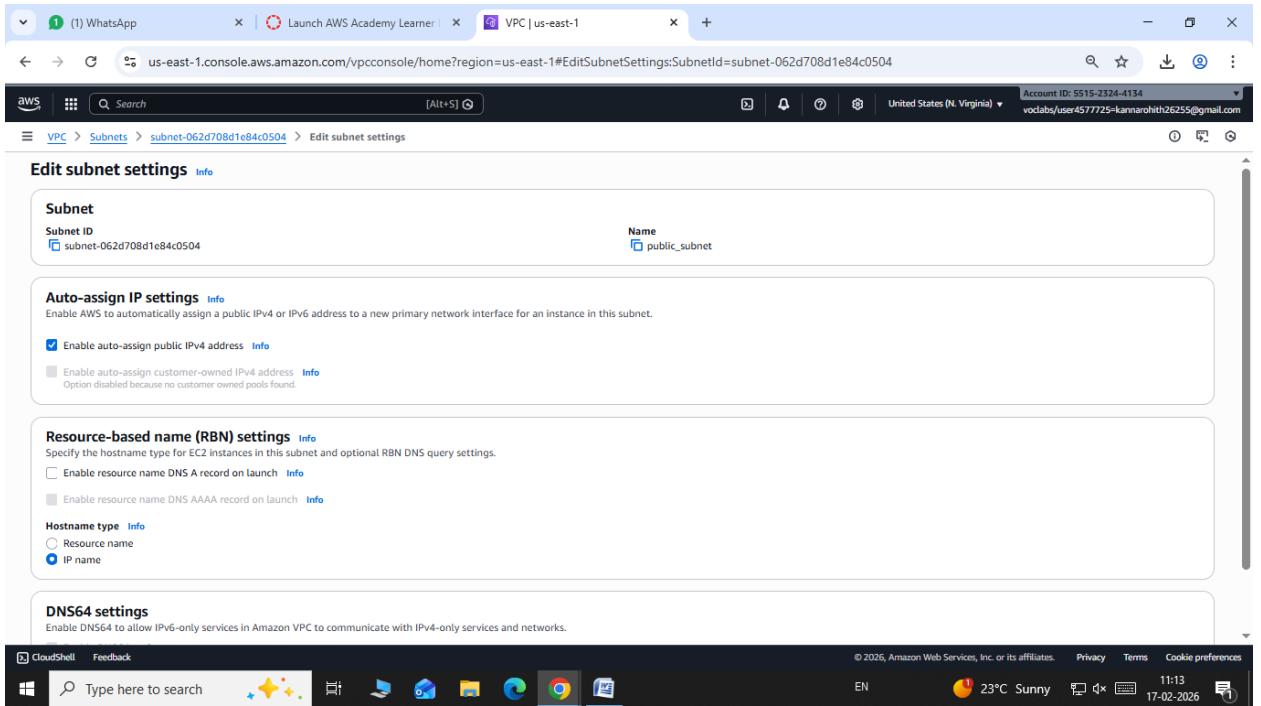
**Actions**

- Create subnet**
- View details**
- Create flow log**
- Edit subnet settings** (highlighted)
- Edit IPv6 CIDs**
- Edit network ACL association**
- Edit route table association**
- Edit CIDR reservations**
- Share subnet**
- Manage tags**
- Delete subnet**

**subnet-062d708d1e84c0504 / public\_subnet**

**Details**

Subnet ID <a href="#">subnet-062d708d1e84c0504</a>	Subnet ARN <a href="#">arn:aws:ec2:us-east-1:551523244134:subnet/subnet-062d708d1e84c0504</a>	State <b>Available</b>	Block Public Access Off
IPv4 CIDR <a href="#">10.0.1.0/24</a>	IPv6 CIDR -	IPv6 CIDR association ID -	IPv6 CIDR association ID -
Availability Zone <a href="#">use-az1 (us-east-1a)</a>	Available IPv4 addresses <a href="#">251</a>	VPC <a href="#">vpc-0a9d577e85b87e742   custom VPC</a>	Route table <a href="#">rtb-098657d0c8f22f82b   public_route_table</a>



## ◆ Step 6: Configure Security Groups

1. Create a **Security Group**
2. Add inbound rules:
  - o HTTP (80) / HTTPS (443)
  - o SSH (22) from trusted IP
3. Attach to EC2 instances

## ◆ Step 7: Launch EC2 Instances

- Public EC2 → Public subnet

Screenshot of the AWS CloudShell interface showing the AWS Management Console. The user is navigating through the EC2 service to launch a new instance.

The browser tabs are:

- WhatsApp
- Launch AWS Academy Learner
- Launch an instance | EC2 | us-east-1
- SecurityGroups | VPC Console

The URL in the address bar is: [us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances](https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances)

The AWS Management Console navigation bar shows:

- Search
- [Alt+S]
- Ask Amazon Q
- United States (N. Virginia)
- Account ID: 5515-2324-4134
- vodlabs/user4577725=kannarohith26255@gmail.com

The main content area displays the "Launch an instance" wizard:

- Name and tags**: A text input field contains "public\_EC2". A "Add additional tags" button is available.
- Application and OS Images (Amazon Machine Image)**: A search bar and a "Quick Start" tab are present. Recent AMIs include Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. A "Browse more AMIs" link is available.
- Amazon Machine Image (AMI)**: A specific AMI entry for "Amazon Linux 2023 kernel-6.1 AMI" is shown, with details: ami-0c1fe732b5494dc14 (64-bit (x86), uefi-preferred) / ami-02b86da1e539b4dd0 (64-bit (Arm), uefi). It is marked as "Free tier eligible".
- Summary**: Shows 1 instance. Configuration details include:
  - Software Image (AMI)**: Amazon Linux 2023 AMI 2023.10... (read more)
  - Virtual server type (instance type)**: t2.micro
  - Firewall (security group)**: default
  - Storage (volumes)**: 1 volume(s) - 8 GiB
- Buttons**: "Cancel", "Launch instance" (highlighted in orange), and "Preview code".

Screenshot of the AWS CloudShell interface showing the AWS Management Console. The user is navigating through the EC2 service to launch a new instance.

The browser tabs are:

- WhatsApp
- Launch AWS Academy Learner
- Launch an instance | EC2 | us-east-1
- SecurityGroups | VPC Console

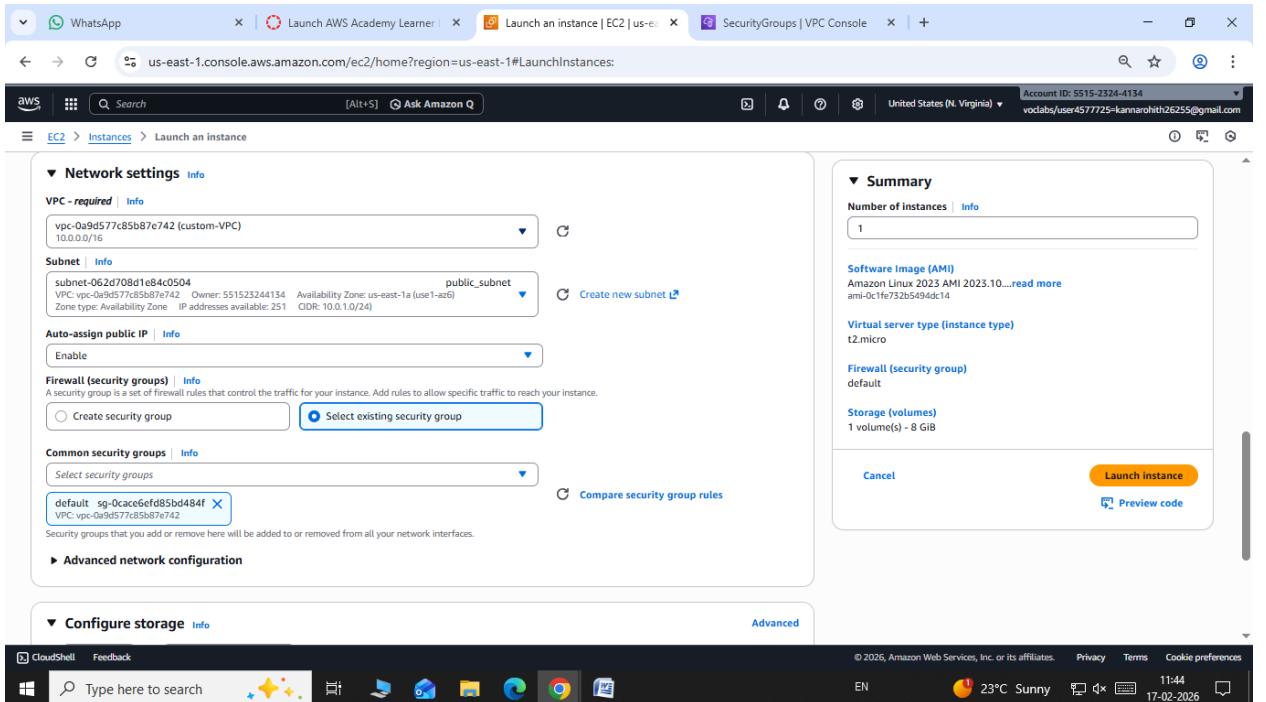
The URL in the address bar is: [us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances](https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances)

The AWS Management Console navigation bar shows:

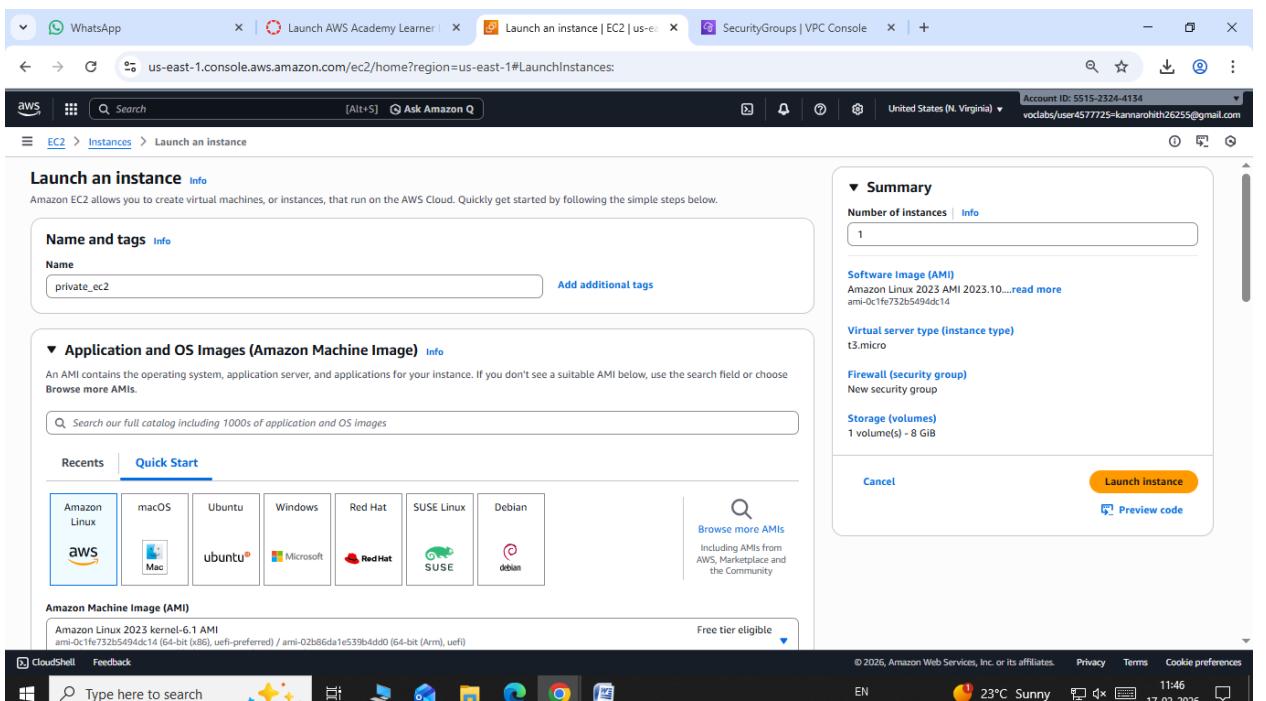
- Search
- [Alt+S]
- Ask Amazon Q
- United States (N. Virginia)
- Account ID: 5515-2324-4134
- vodlabs/user4577725=kannarohith26255@gmail.com

The main content area displays the "Launch an instance" wizard:

- Amazon Machine Image (AMI)**: A search bar and a "Quick Start" tab are present. Recent AMIs include Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. A "Browse more AMIs" link is available.
- Amazon Linux 2023 kernel-6.1 AMI**: A specific AMI entry is shown, with details: ami-0c1fe732b5494dc14 (64-bit (x86), uefi-preferred) / ami-02b86da1e539b4dd0 (64-bit (Arm), uefi). It is marked as "Free tier eligible".
- Description**: Describes Amazon Linux 2023 (kernel-6.1) as a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.
- AMI Details**:
  - Architecture**: 64-bit (x86)
  - Boot mode**: uefi-preferred
  - AMI ID**: ami-0c1fe732b5494dc14
  - Publish Date**: 2026-02-03
  - Username**: ec2-user
  - Verified provider**
- Instance type**:
  - t2.micro**: Family: t2, 1 vCPU, 1 GiB Memory. Current generation: true. On-Demand Linux base pricing: 0.0116 USD per Hour
  - All generations
  - Compare instance types
- Summary**: Shows 1 instance. Configuration details include:
  - Software Image (AMI)**: Amazon Linux 2023 AMI 2023.10... (read more)
  - Virtual server type (instance type)**: t2.micro
  - Firewall (security group)**: default
  - Storage (volumes)**: 1 volume(s) - 8 GiB
- Buttons**: "Cancel", "Launch instance" (highlighted in orange), and "Preview code".

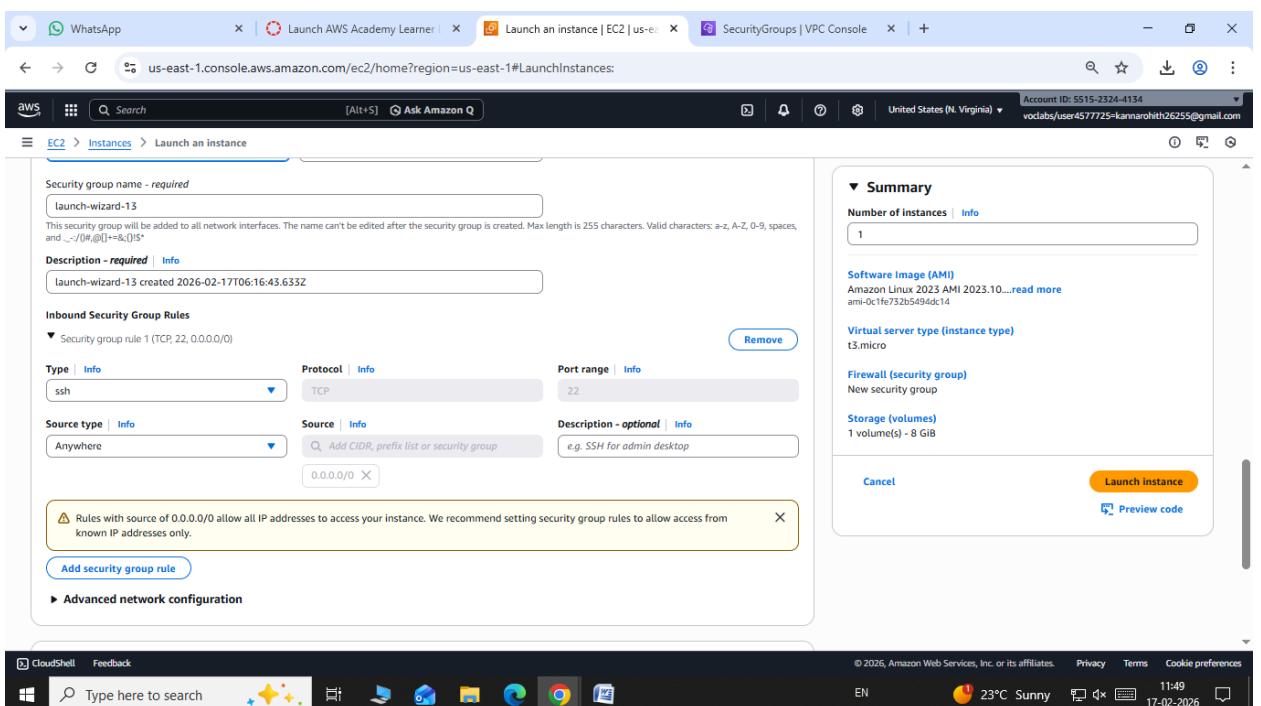
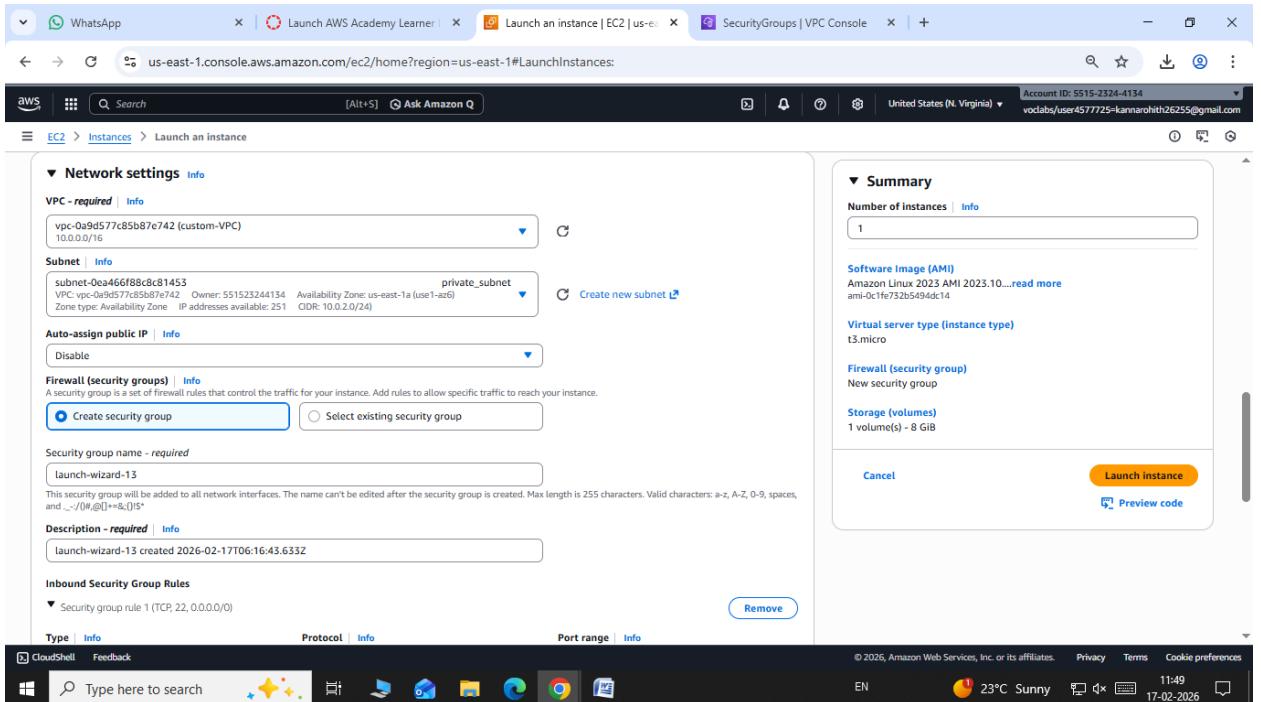


- Private EC2 → Private subnet  
Attach correct security groups.



The screenshot shows the AWS EC2 Launch Instance wizard. On the left, there's a sidebar with 'Recent' and 'Quick Start' tabs. Under 'Quick Start', several AMI icons are displayed: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. A search bar labeled 'Search' is at the top. The main area shows the selected 'Amazon Linux 2023 kernel-6.1 AMI'. It includes a detailed description, a table with instance details (Architecture: 64-bit (x86), Boot mode: uefi-preferred, AMI ID: ami-0c1fe732b549dc14, Publish Date: 2026-02-03, Username: ec2-user, Verified provider), and a note that it's 'Free tier eligible'. To the right is a 'Summary' panel with fields for 'Number of instances' (set to 1) and 'Software Image (AMI)', along with buttons for 'Launch instance' and 'Preview code'.

This screenshot continues the AWS EC2 Launch Instance wizard. The 'Instance type' tab is selected, showing the 't3.micro' instance type. It lists base pricing for various demand and on-demand options. Below this, the 'Additional costs apply for AMIs with pre-installed software' section is visible. The 'Key pair (login)' tab is shown next, where a key pair named 'exp5' is selected. The 'Network settings' tab is also visible, showing network and subnet configurations. The right side of the screen features a 'Summary' panel with instance details and a 'Launch instance' button.

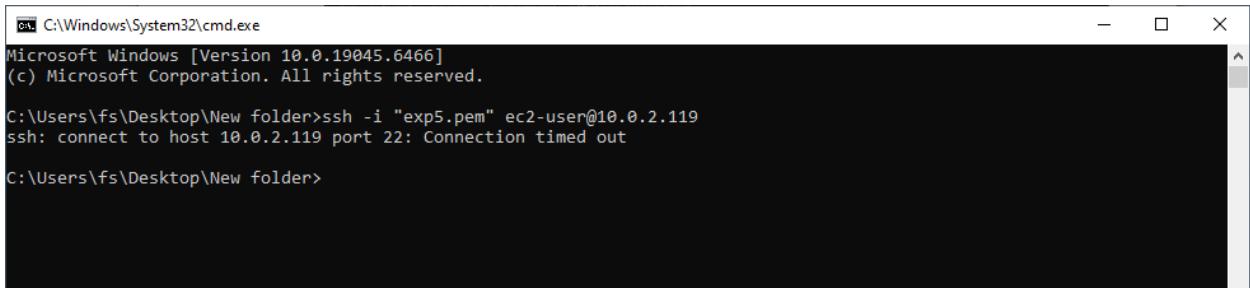


## ◆ Step 8: Test the Setup

- Public EC2 → Internet access ✓

```
ec2-user@ip-10-0-1-63:~  
_/m/  
Last login: Tue Feb 17 06:22:11 2026 from 183.82.97.138  
[ec2-user@ip-10-0-1-63 ~]$ ping google.com  
PING google.com (192.178.218.102) 56(84) bytes of data.  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=1 ttl=100 time=1.82 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=2 ttl=100 time=2.31 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=3 ttl=100 time=2.08 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=4 ttl=100 time=2.08 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=5 ttl=100 time=1.84 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=6 ttl=100 time=1.83 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=7 ttl=100 time=2.06 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=8 ttl=100 time=1.83 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=9 ttl=100 time=1.82 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=10 ttl=100 time=1.83 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=11 ttl=100 time=1.82 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=12 ttl=100 time=1.83 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=13 ttl=100 time=1.84 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=14 ttl=100 time=2.06 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=15 ttl=100 time=2.06 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=16 ttl=100 time=2.47 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=17 ttl=100 time=2.32 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=18 ttl=100 time=1.82 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=19 ttl=100 time=1.86 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=20 ttl=100 time=1.85 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=21 ttl=100 time=2.22 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=22 ttl=100 time=2.12 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=23 ttl=100 time=2.11 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=24 ttl=100 time=1.84 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=25 ttl=100 time=1.81 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=26 ttl=100 time=1.83 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=27 ttl=100 time=2.05 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=28 ttl=100 time=2.28 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=29 ttl=100 time=1.92 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=30 ttl=100 time=1.98 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=31 ttl=100 time=1.82 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=32 ttl=100 time=2.05 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=33 ttl=100 time=2.08 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=34 ttl=100 time=1.86 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=35 ttl=100 time=1.83 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=36 ttl=100 time=1.81 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=37 ttl=100 time=2.22 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=38 ttl=100 time=1.95 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=39 ttl=100 time=1.83 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=40 ttl=100 time=2.64 ms  
64 bytes from yuadtnq-in-f102.1e100.net (192.178.218.102): icmp_seq=41 ttl=100 time=2.80 ms  
%C  
--- google.com ping statistics ---  
41 packets transmitted, 41 received, 0% packet loss, time 40069ms  
rtt min/avg/max/mdev = 1.819/2.009/2.801/0.232 ms  
[ec2-user@ip-10-0-1-63 ~]$ ping google.com
```

- Private EC2 → Internet via NAT ✓
  - Private EC2 → No direct inbound access ✓



A screenshot of a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window shows the following text:

```
Microsoft Windows [Version 10.0.19045.6466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\fs\Desktop>New folder>ssh -i "exp5.pem" ec2-user@10.0.2.119
ssh: connect to host 10.0.2.119 port 22: Connection timed out

C:\Users\fs\Desktop>New folder>
```

### **Scenario Based Questions:**

1. You have a web application where users access a website, but the database should not be exposed to the internet. How would you design the VPC? Which resources go into public and private subnets?

Answer: Place ALB/Web servers in public subnets and the database in private subnets; allow DB access only from the application security group.

2. An EC2 instance in a public subnet cannot access the internet. What VPC components would you check and why?

Answer: Check Internet Gateway attachment, route table (0.0.0.0/0 → IGW), public IP assignment, and Security Group/NACL rules.

3. Your application server needs to connect to an RDS database securely. How would you configure security groups and subnets?

Answer: Deploy RDS in a private subnet and allow inbound DB port access only from the application server's security group.

4. You are asked to design a VPC for 500 servers today, but it should scale to 2,000 servers in the future. How would you choose the CIDR block?

Answer: Choose a larger CIDR block (e.g., /20) to accommodate future scaling without IP exhaustion.

5. A company wants secure connectivity between its on-premises data center and AWS VPC. Which AWS services would you choose and why (VPN vs Direct Connect)?

Answer: Use VPN for quick, cost-effective setup; choose Direct Connect for dedicated, high-bandwidth, low-latency connectivity.

6. You need to connect two VPCs, but both use the same CIDR range. How would you solve this problem?

Answer: Modify one VPC's CIDR range or use NAT/PrivateLink to handle overlapping IP addresses.

7. Multiple microservices running in different subnets need to communicate securely. How would you design routing and security groups?

Answer: Use private subnets with proper route tables and configure security groups referencing each other for controlled access.

8. Design a VPC for a 3-tier application (Web, App, DB) with high security and scalability. Explain subnets, route tables, gateways, and security groups.

Answer: Use public subnets for ALB, private subnets for App, isolated private subnets for DB; attach IGW/NAT appropriately and restrict access via security groups.

9. You suspect unusual traffic inside your VPC. How would you monitor and analyze network traffic?

Answer: Enable VPC Flow Logs and analyze using CloudWatch, GuardDuty, or Traffic Mirroring.

10. Why can't a private subnet have an Internet Gateway directly attached?

Answer: An Internet Gateway attaches to the VPC, not to individual subnets; private subnets simply lack a route to it.

11. What happens if route tables are misconfigured in a VPC?

Answer: Traffic will fail to reach intended destinations, causing connectivity issues or outages.

12. What happens if a route table has no local route?

Answer: Instances within the VPC will not be able to communicate with each other.

13. An EC2 instance allows traffic on port 80 in the security group, but traffic is still blocked. What could be the reason?

Answer: Possible causes include NACL blocking, OS firewall rules, missing public IP, incorrect routing, or the web service not running.

14. You need to create a VPC that will host 1,000+ EC2 instances across multiple AZs. How do you decide the CIDR block?

Answer: Select a sufficiently large CIDR (e.g., /21 or /20) and divide it into multiple subnets across AZs for scalability.

15. Only a company's corporate IP should be able to SSH into EC2 instances. How would you implement this securely in AWS VPC?

Answer: Configure the security group to allow port 22 access only from the corporate public IP address (/32).

16. An EC2 instance can send traffic out but cannot receive responses. Which VPC component might be misconfigured and why?

Answer: Likely a Network ACL blocking inbound return traffic (ephemeral ports) or asymmetric routing issue.

