

Network Mastery with AWS VPC

Author: Oluwaseun Osunsola

Environment: AWS

Project Link: <https://github.com/Oluwaseunoa/DevOps-Projects/tree/main>

Project Overview

In this session, we explore the core concepts of Amazon Web Services (AWS), focusing specifically on Virtual Private Clouds (VPCs). Our objective is to understand the fundamental components of VPC infrastructure, including subnets, gateways, and routing tables. Through practical demonstrations and interactive exercises, we navigate the AWS Management Console to deploy and manage these critical components effectively.

Before proceeding with setting up VPCs, ensure a solid understanding of cloud networking basics.

Project Goals

- Understand the fundamentals of Virtual Private Cloud (VPC) and its components.
- Gain hands-on experience in setting up and configuring VPC, subnets, Internet Gateway, NAT Gateway, and VPC peering connections.
- Learn how to enable internet connectivity securely within a VPC.
- Implement outbound internet access through the NAT Gateway.
- Establish direct communication between VPCs using VPC peering.

Learning Outcomes

- Acquired knowledge about VPC and its essential components, such as subnets, gateways, and route tables.
- Developed skills in creating and configuring VPC resources using the AWS Management Console.
- Learned how to set up routing tables to enable internet connectivity and outbound internet access securely.
- Gained understanding of VPC peering and its significance in connecting multiple VPCs within the same or different regions.
- Enhanced understanding of network security principles and best practices for cloud environments.

Key Concepts

What is VPC, Subnets, Internet Gateway, and NAT Gateway?

Imagine building a virtual space for the company GatoGrowFast.com so that its computers can communicate securely. That's what a **Virtual Private Cloud (VPC)** is all about—a private room in the cloud for GatoGrowFast.com's use.

Example: Think of GatoGrowFast.com's office building with different departments like HR, Finance, and IT, each with its own area and access rules. In a VPC, GatoGrowFast.com creates different sections, called **subnets**, for different parts of the business.

To connect its office to the internet, GatoGrowFast.com uses a router to control data flow. In a VPC, this is achieved with an **Internet Gateway**, allowing secure communication with the internet.

A **NAT (Network Address Translation) Gateway** acts as a secret agent between GatoGrowFast.com's computers and the internet. When a computer in the virtual office communicates with the internet, the NAT Gateway translates the message and hides the sender's identity, similar to sending a letter without a return address. This keeps GatoGrowFast.com's computers safe and anonymous online.

Note: A **router** directs data packets between networks, acting like a traffic cop for the internet. Data is broken into packets, and the router uses **routing tables**—like maps of the internet—to determine the best path for these packets based on destination IP addresses.

What is an IP Address?

An **IP address** is like a phone number for a computer, a unique set of numbers that helps devices find and communicate with each other on a network, like the internet.

Types of IP Addresses:

- **Public IP Address:** Like a home address, it's unique and allows other computers on the internet to find your device. Assigned by an Internet Service Provider (ISP), it enables global communication. Public IPs can be **dynamic** (changing periodically) or **static** (constant, used for servers or consistent connectivity).
- **Private IP Address:** Like an internal extension number in an office, used for communication within a specific network (e.g., home Wi-Fi or office network). Assigned by a router or DHCP server, private IPs are not routable over the internet and can be reused across different private networks without conflict.

IP Address Versions:

- **IPv4:** The most common type, a 32-bit numeric address written in decimal format (e.g., 192.168.0.1). Each octet ranges from 0 to 255, divided into classes A, B, and C for host addressing.
- **IPv6:** Designed to replace IPv4 due to address exhaustion, a 128-bit hexadecimal address (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334), offering a vast address space.

What is CIDR?

CIDR (Classless Inter-Domain Routing) simplifies talking about groups of IP addresses. Instead of listing each address, CIDR uses a shorthand, like saying "all houses on Main Street" instead of naming each house.

Example: For the IP address 192.168.1.0, CIDR notation like 192.168.1.0/24 refers to all addresses from 192.168.1.0 to 192.168.1.255.

Calculating Available IP Addresses in a CIDR Block: Use the formula: $2^{(32 - \text{CIDR notation})} - 2$ (subtracting 2 for the network and broadcast addresses).

Example: For 192.168.1.0/24:

- $2^{(32 - 24)} - 2 = 2^8 - 2 = 256 - 2 = 254$ available IP addresses.

What is a Gateway?

Gateways are like doorways between networks, enabling data to travel between a local network and others, like the internet. They act as a traffic cop, directing data to its destination.

Example: In a city with neighborhoods, to visit a friend in Neighborhood B from Neighborhood A, you pass through a gateway connecting the two. Similarly, a network gateway connects your local network to the internet.

What is a Route Table?

A **route table** is like a map guiding data around a network. It lists destinations and paths (routes) to reach them. Devices consult the route table to determine where to send data packets.

Example: To send data to a website, a computer checks its route table to find the gateway to the internet. The router then forwards the data to the next stop, ensuring it reaches its destination.

Connection Between Gateway and Route Table

- **Gateways:** Devices (e.g., routers, firewalls) serving as entry/exit points between networks with different IP ranges. They receive packets and determine the next destination based on routing information.
- **Route Tables:** Maintained by networking devices, they list destination networks and the next hop (gateway) to reach them. Devices use route tables to find the best path for packets.
- **Connection:** When a device sends data outside its local network, it checks the route table to identify the gateway. The packet is forwarded to the gateway, which continues routing until the packet reaches its destination.

Difference Between Internet Gateway and NAT Gateway

- **Internet Gateway:** A door to the internet for a subnet, allowing resources (e.g., EC2 instances) to send and receive internet traffic. It enables two-way communication.
- **NAT Gateway:** A one-way street for subnet traffic, allowing resources to access the internet without allowing incoming internet traffic, enhancing security.

What is VPC Peering?

VPC peering connects two virtual offices (VPCs) in the cloud for direct communication, like neighboring offices sharing files without a middleman. By default, EC2 instances in different VPCs cannot communicate. VPC peering establishes a direct network connection between VPCs.

Why VPC Peering?

It enables different parts of a cloud network (e.g., development and marketing VPCs) to share data securely and efficiently.

Key Points:

- VPCs require peering, VPN, or AWS Direct Connect for connectivity.
- Subnets within the same VPC communicate by default via AWS-configured route tables.
- EC2 instances in the same or different subnets within a VPC communicate if security group rules and route tables allow.
- **VPC Peering Basics:**

- Allows direct communication using private IP addresses.
- Supports same or different regions and AWS accounts.
- CIDR blocks must not overlap.
- Requires proper Security Group and Network Access Control List (NACL) configurations.
- No transitive traffic (traffic cannot flow through a peered VPC to another VPC).
- Route tables must include routes for the peer VPC's CIDR block.
- Limits exist on the number of peering connections and route entries.

What is a VPC Endpoint?

A **VPC endpoint** is a secure, dedicated tunnel between a VPC and an AWS service (e.g., S3), bypassing the public internet. It ensures private, safe access to resources.

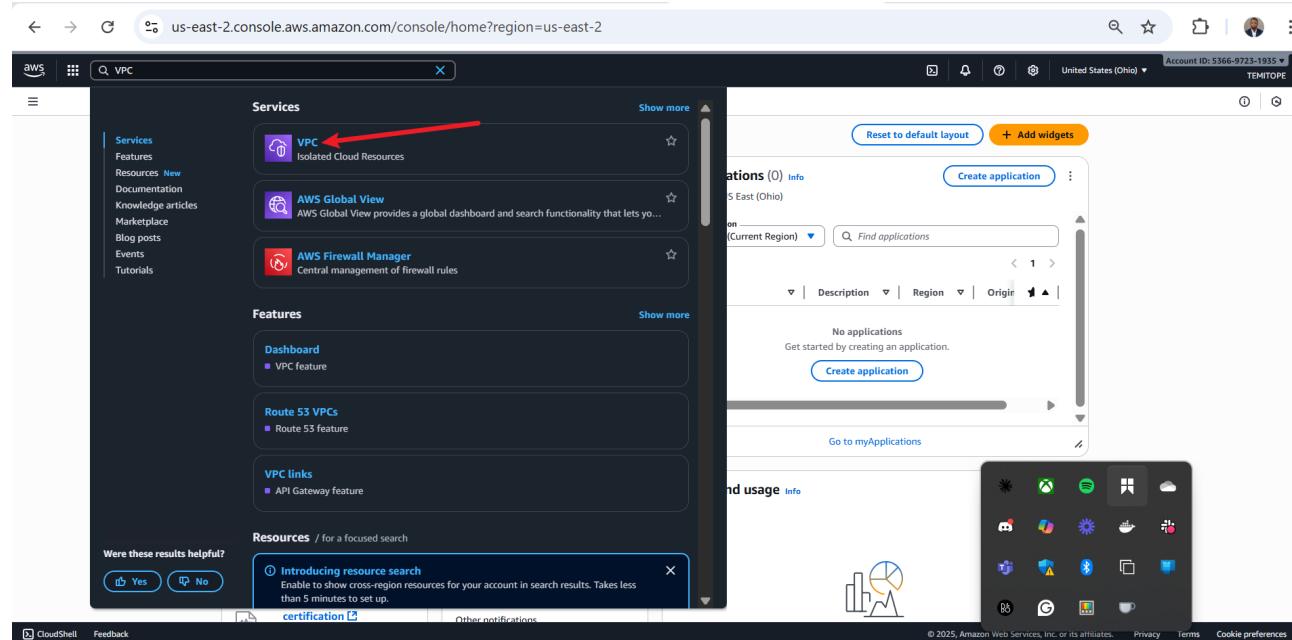
Problem Context: Backing up data from an EC2 instance to an S3 bucket typically goes over the internet, risking sensitive data exposure. A VPC endpoint creates a private connection, keeping data secure.

Note: An **EC2 instance** is a virtual server in AWS for running applications, offering scalable computing power for hosting websites, software, or data processing.

Practical Steps

Part 1: Setting Up a Virtual Private Cloud (VPC)

1. Navigate to the search bar, enter 'VPC', and click on the VPC service.



2. On the VPC dashboard, click **Create VPC**.

The screenshot shows the AWS VPC dashboard. At the top center, there is a prominent blue button labeled "Create VPC". A thick red arrow points directly at this button. Below the button, there is another blue button labeled "Launch EC2 Instances". The dashboard displays various resources by region, including VPCs, Subnets, Route Tables, Internet Gateways, and NAT Gateways, all currently set to Ohio. On the right side, there are sections for "Service Health", "Settings", and "Additional Information". The bottom of the screen shows standard AWS navigation links like CloudShell and Feedback.

3. Select **VPC only**, set the IPv4 CIDR block (e.g., 10.0.0.0/16), and click **Create VPC**.

This screenshot shows the "Create VPC" configuration page. In the "Resources to create" section, the radio button for "VPC only" is selected, indicated by a red arrow. Below it, the "Name tag - optional" field contains "my-vpc-01". Under "IPv4 CIDR block", the input field is set to "10.0.0.0/16", also highlighted with a red box. In the "Tags" section, there is a note about adding tags and a "Create VPC" button at the bottom right. A large red arrow points from the "Create VPC" button on the left towards the "Create VPC" button on the right, indicating the final step.

4. VPC successfully created.

The screenshot shows the AWS VPC console interface. A green success message at the top states: "You successfully created vpc-059fcaa09dbb14e12". Below this, the VPC details are listed:

VPC ID vpc-059fcaa09dbb14e12	State Available	Block Public Access Off	DNS hostnames Disabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-09149d5972dcad56b	Main route table rtb-01b8c1967af12ebbo
Main network ACL acl-0dd802252bcd3adcb	Default VPC	IPv4 CIDR 10.0.0.0/16	IPv6 pool -
IPv6 CIDR -	Network Address Usage metrics	Route 53 Resolver DNS Firewall rule groups -	Owner ID 536697231935

Below the details, there are tabs for Resource map, CIDRs, Flow logs, Tags, and Integrations. The Resource map tab is selected, showing a summary of resources: VPC (Your AWS virtual network), Subnets (0), Route tables (1), and Network Connect (0). A modal window titled "Resource map" is open, displaying a grid of icons representing various AWS services.

5. **Note:** If a CIDR block size error occurs, ensure the block size is between /16 and /28.

IPv4 CIDR

10.0.0.0/8

⚠ CIDR block size must be between /16 and /28.

CIDR block size must be between /16 and /28.

IPv4 CIDR

10.0.0.0/16

CIDR block size must be between /16 and /28.

Part 2: Configuring Subnets within the VPC

1. From the created VPC, click **Subnets** to create a subnet.

The screenshot shows the AWS VPC console interface. In the left sidebar, under 'Virtual private cloud', the 'Your VPCs' section is selected, and the 'Subnets' link is highlighted with a red box. At the top right, there is a 'Actions' dropdown menu with a red arrow pointing to the 'Create VPC' button. The main content area displays a table titled 'Your VPCs (1/2) Info' with two rows of VPC information. The first VPC, 'vpc-059fcaa09ddb14e12', has its 'Actions' dropdown open, showing options like 'Edit', 'Delete', and 'Associate route table'. The second VPC, 'vpc-0946dc2d5bf4cbe', also has its 'Actions' dropdown open. The bottom of the screen includes standard AWS navigation links like CloudShell and Feedback, and a footer with copyright information.

2. On the subnet dashboard, click **Create subnet**.

The screenshot shows the AWS VPC console interface. In the left sidebar, under 'Virtual private cloud', the 'Subnets' section is selected, highlighted with a red box. At the top right, there is a 'Actions' dropdown menu with a red arrow pointing to the 'Create subnet' button. The main content area displays a table titled 'Subnets (3) Info' with three rows of subnet information. The first subnet, 'subnet-0774bf3662e7df059', is associated with VPC 'vpc-0946dc2d5bf4cbe'. The second subnet, 'subnet-03ab326585afffc0b', is associated with VPC 'vpc-0946dc2d5bf4cbe'. The third subnet, 'subnet-08834c04364b5df9e', is associated with VPC 'vpc-0946dc2d5bf4cbe'. The bottom of the screen includes standard AWS navigation links like CloudShell and Feedback, and a footer with copyright information.

3. Select the VPC created in Part 1.

The screenshot shows the 'Create subnet' interface in the AWS VPC console. The 'VPC' section is active, showing a dropdown menu for selecting a VPC. The option 'vpc-059fcaa09dbb14e12' is highlighted with a red box. Below the dropdown, there is a note: 'Select a VPC first to create new subnets.' At the bottom right of the form are 'Cancel' and 'Create subnet' buttons.

4. Enter subnet name (e.g., **my-public-subnet-1**), set availability zone, specify IPv4 CIDR block (e.g., 10.0.1.0/24), and click **Add new subnet**.

The screenshot shows the 'Subnet Settings' interface for creating a new subnet. It includes fields for 'Subnet name' (with 'my-public-subnet-1' entered), 'Availability Zone' (set to 'United States (Ohio) / use2-ad1 (us-east-2a)'), and 'IPv4 subnet CIDR block' (set to '10.0.6.0/24'). There is also a 'Tags - optional' section with a single tag 'Name: my-public-subnet-1'. A red arrow points to the 'Add new subnet' button at the bottom left of the form.

5. Enter subnet name (e.g., **my-private-subnet-1**), set availability zone, specify IPv4 CIDR block (e.g., 10.0.2.0/24), and click **Create subnet**.

Subnet 2 of 2

Subnet name
Create a tag with the key of 'Name' and a value that you specify.
 The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
 256 IPs
◀ ▶ ▲ ▼

Tags - optional

Key Value - optional

You can add 49 more tags.

6. View the architecture.



7. Subnets **my-public-subnet-1** and **my-private-subnet-1** successfully created.

The screenshot shows the AWS VPC Subnets page. The left sidebar has sections for Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers), Security (Network ACLs, Security groups), and PrivateLink and Lattice (Getting started, Endpoints, Endpoint services, Service networks). The main content area shows a table of subnets:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv6 C...
my-private-subnet-1	subnet-0bffd09a345bd2e1f	Available	vpc-059fcfa09dbb14e12	Off	10.0.7.0/24	-	-
my-public-subnet-1	subnet-04e9903eb59663bdb	Available	vpc-059fcfa09dbb14e12	Off	10.0.6.0/24	-	-

Part 3: Creating Internet Gateway and Attaching it to VPC

1. Click **Internet Gateways** to connect the public subnet to the internet.

The screenshot shows the AWS VPC Subnets page. The left sidebar has sections for Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers), Security (Network ACLs, Security groups), and PrivateLink and Lattice (Getting started, Endpoints, Endpoint services, Service networks). The right sidebar shows a grid of icons for various AWS services. A red arrow points to the 'Internet gateways' link in the sidebar.

2. Click **Create Internet Gateway**.

The screenshot shows the AWS VPC console with the URL us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#igws. The left sidebar is expanded, showing sections like Virtual private cloud, Security, and PrivateLink and Lattice. The main area displays a table of Internet gateways with one entry: Name igw-037e56e68b137c83b, Internet gateway ID igw-037e56e68b137c83b, State Attached, VPC ID vpc-0a946cd2d5bf4cbe, and Owner 536697231935. Below the table, a message says 'Select an internet gateway above'. In the top right, there is an 'Actions' dropdown with a red arrow pointing to the 'Create internet gateway' button.

3. Name the Internet Gateway and click **Create Internet Gateway**.

The screenshot shows the 'Create internet gateway' wizard with the URL us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#CreateInternetGateway. The form has two sections: 'Internet gateway settings' (Name tag: my-internet-gw-1) and 'Tags - optional' (Key: Name, Value: optional, my-internet-gw-1). At the bottom right, there is a 'Create internet gateway' button with a red arrow pointing to it.

4. Internet Gateway created, note it is detached.

The screenshot shows the AWS VPC console with the URL <https://us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#InternetGateway:id=igw-05cba0c2e0b35021e>. The page displays a message: "The following internet gateway was created: igw-05cba0c2e0b35021e - my-internet-gw-1. You can now attach to a VPC to enable the VPC to communicate with the internet." Below this, the internet gateway details are shown: ID (igw-05cba0c2e0b35021e), Name (my-internet-gw-1), State (Detached, highlighted with a red box), VPC ID (-), Owner (536697231935), and Tags (my-internet-gw-1). The Actions button is also highlighted with a red box.

5. Click Actions and Attach to VPC.

The screenshot shows the same AWS VPC console page as above. A red arrow points to the 'Actions' button in the top right corner of the main content area. The 'Attach to VPC' option under the 'Attach to VPC' section is also highlighted with a red arrow.

6. Select the VPC and click Attach Internet Gateway.

The screenshot shows the 'Attach to VPC' dialog box. It has a 'VPC' section with a sub-section 'Available VPCs' containing a search bar with the value 'vpc-059fcaa09dbb14e12'. At the bottom right of the dialog box, there is a 'Cancel' button and a large red arrow pointing to the 'Attach internet gateway' button.

7. Internet Gateway successfully attached to VPC.

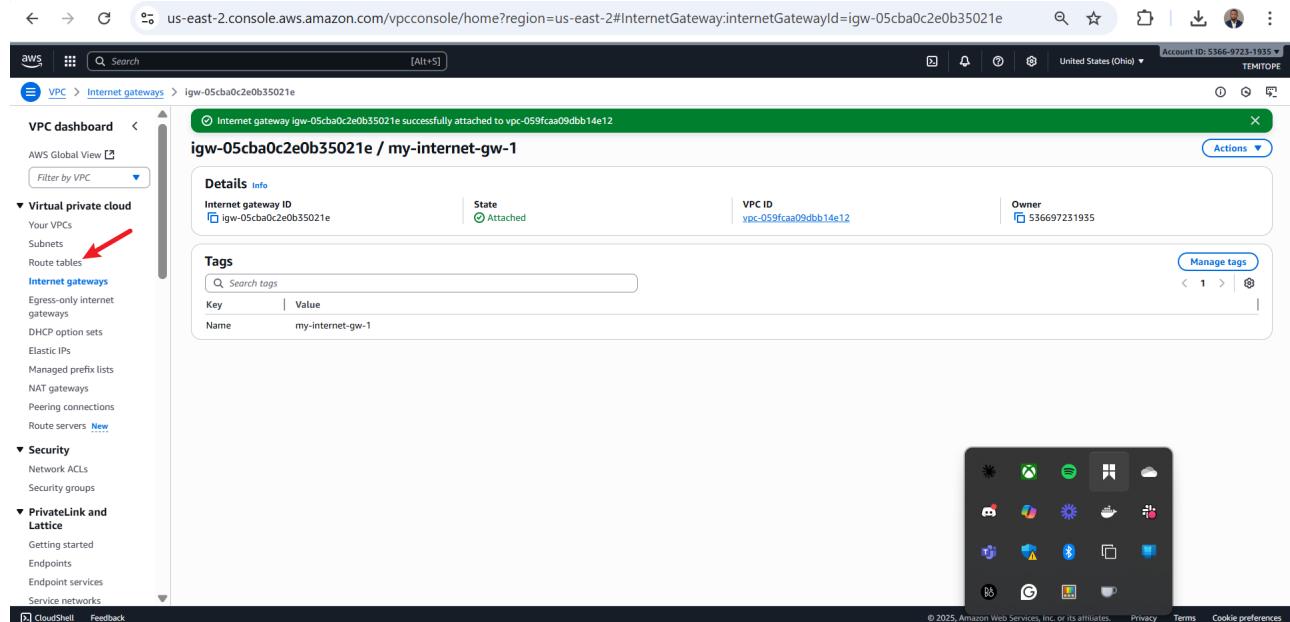
The screenshot shows the AWS VPC console with the URL us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#InternetGateway:id=igw-05cba0c2e0b35021e. The main content area displays an Internet gateway named "igw-05cba0c2e0b35021e / my-internet-gw-1". A green banner at the top indicates that the gateway has been successfully attached to a VPC. The "State" field is highlighted with a red box and shows "Attached". The "VPC ID" is listed as "vpc-059fcaa09dbb14e12". The "Owner" is listed as "536697231935". On the left sidebar, the "Internet gateways" section is selected. A small screenshot of a Windows desktop environment is visible in the bottom right corner.

Part 4: Enabling Internet Connectivity with the Internet Gateway by Setting Up Routing Tables

1. View the current architecture.

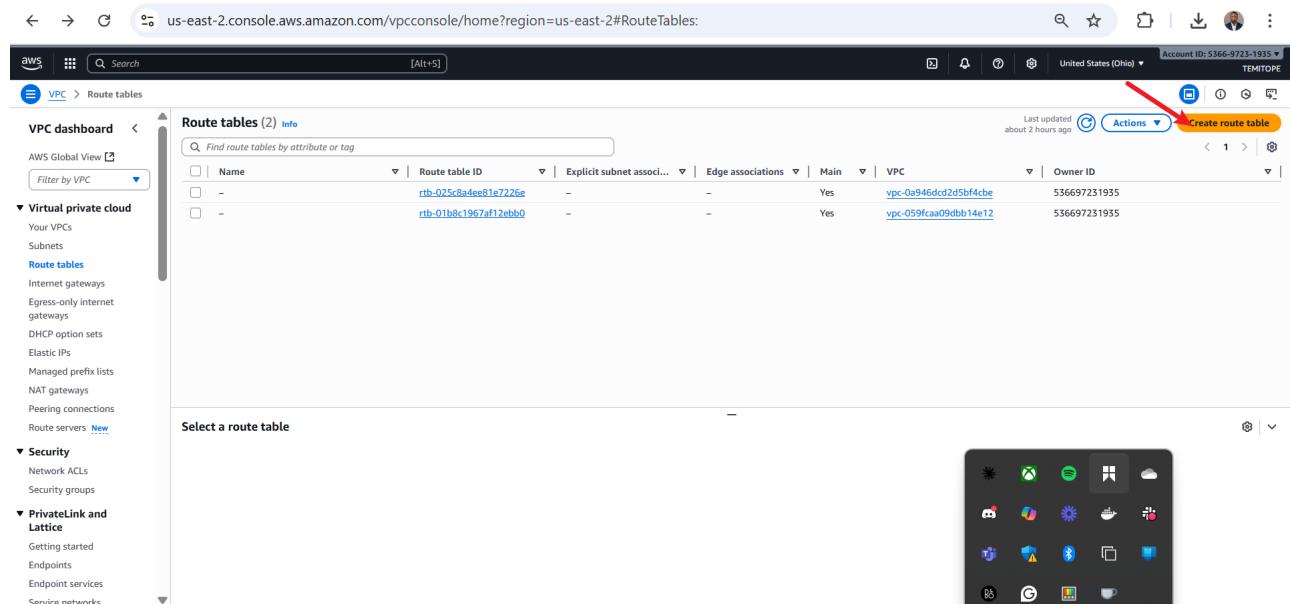


2. Click Route Tables.



The screenshot shows the AWS VPC console with the URL us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#InternetGateway:internetGatewayId=igw-05cba0c2e0b35021e. The main content area displays an Internet gateway named 'igw-05cba0c2e0b35021e / my-internet-gw-1'. It shows the gateway is successfully attached to a VPC with VPC ID 'vpc-059fcaa09dbb14e12'. The 'Details' tab is selected, and the 'Tags' section contains a single tag 'Name: my-internet-gw-1'. The left sidebar under 'Virtual private cloud' has 'Route tables' highlighted with a red arrow. The top right corner shows the account ID '536697231935' and the region 'United States (Ohio)'.

3. Click Create route table.



The screenshot shows the AWS VPC console with the URL us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#RouteTables. The main content area displays a table titled 'Route tables (2) Info' showing two existing route tables. The top right corner of the actions menu has a red arrow pointing to the 'Create route table' button. The left sidebar under 'Virtual private cloud' has 'Route tables' highlighted with a red arrow. The top right corner shows the account ID '536697231935' and the region 'United States (Ohio)'.

Name	Route table ID	Explicit subnet associ...	Main	VPC	Owner ID
-	rtb-025c8adeeb1e7226e	-	Yes	vpc-0a946cd2d5bf4cbe	536697231935
-	rtb-01b8c1967af12eb0	-	Yes	vpc-059fcaa09dbb14e12	536697231935

4. Name the route table, select the VPC, and click **Create route table**.

aws [Alt+S]

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

You can add 49 more tags.

5. Click **Subnet associations**, then **Edit subnet associations**.

aws [Alt+S]

VPC > Route tables > rtb-0d41bba2ed11c8401

rtb-0d41bba2ed11c8401 / my-route-table-1

Details Info

Route table ID:
Main: No
Owner ID:

Explicit subnet associations: None

Edge associations: None

Routes **Subnet associations** **Edge associations** **Route propagation** **Tags**

Explicit subnet associations (0)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
<small>No subnet associations</small> You do not have any subnet associations.			

Edit subnet associations

Subnets without explicit associations (2)

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

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
my-public-subnet-1	subnet-0480c4a5151b0c74c	10.0.7.0/24	-
my-public-subnet-1	subnet-0ff5b95c2043b1a6f	10.0.6.0/24	-

Edit subnet associations

6. Select the public subnet and click **Save associations**.

The screenshot shows the 'Edit subnet associations' page for a route table. In the 'Available subnets' section, 'my-public-subnet-1' is selected. In the 'Selected subnets' section, 'subnet-04e9903eb59663bdb / my-public-subnet-1' is listed. A red arrow points to the 'Save associations' button at the bottom right.

7. Under **Routes** tab, click **Edit routes**.

The screenshot shows the details page for route table 'rtb-0d41bba2ed11c8401'. The 'Routes' tab is selected, showing one route entry: 'Destination: 10.0.0.0/16, Target: local, Status: Active'. A red arrow points to the 'Edit routes' button at the top right of the routes table.

8. Click **Add route**.

The screenshot shows the 'Edit routes' page for route table 'rtb-0d41bba2ed11c8401'. A new route entry is being added: 'Destination: 10.0.0.0/16, Target: local, Status: Active'. A red arrow points to the 'Add route' button at the bottom left of the form.

9. Set Destination to **0.0.0.0/0**, Target to the created Internet Gateway, and save changes.

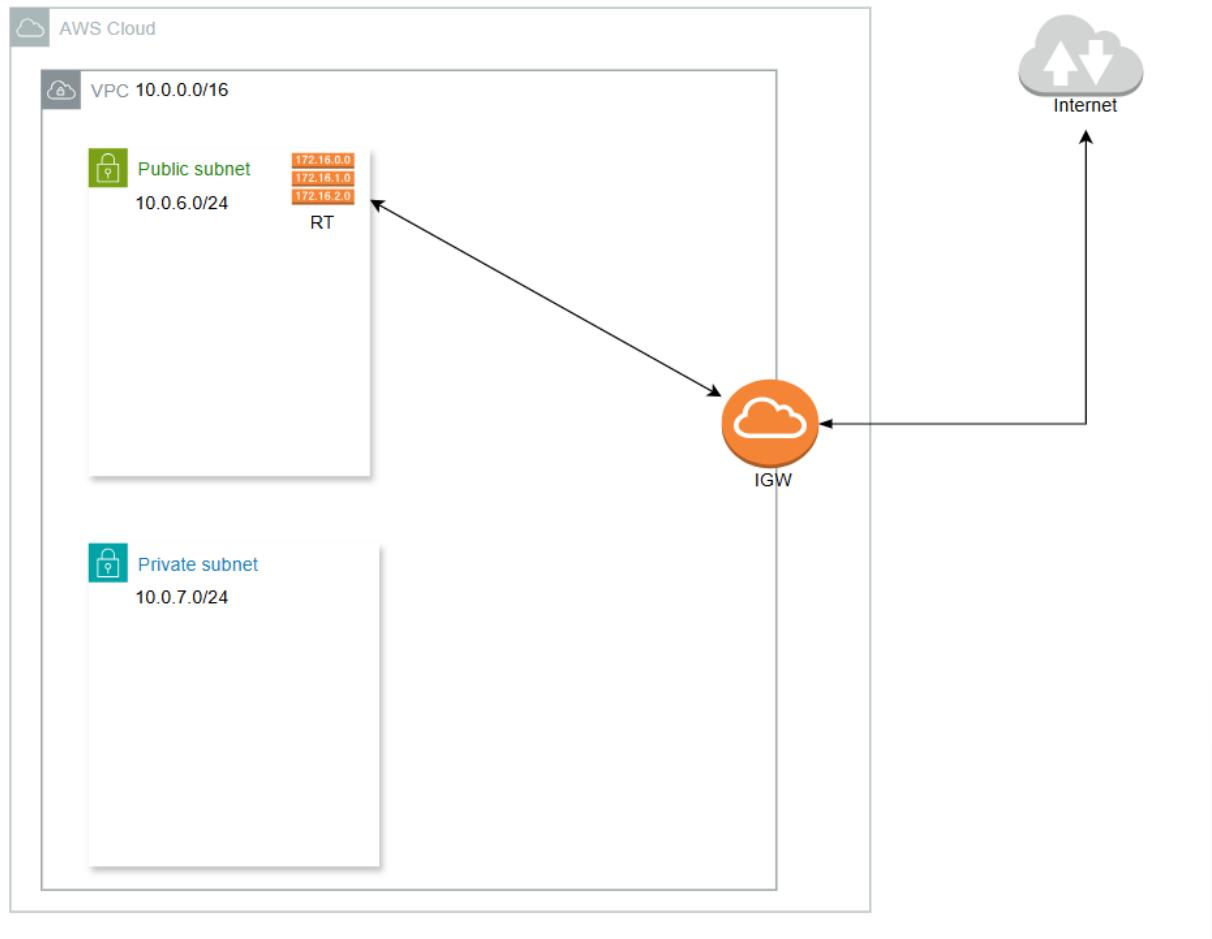
The screenshot shows the 'Edit routes' page for route table rtb-0d41bba2ed11c8401. The 'Destination' field is set to '0.0.0.0/0'. The 'Target' dropdown is set to 'Internet Gateway', and the list shows 'igw-05cba0c2e0b35021e'. The 'Save changes' button is highlighted with a red arrow.

10. Route table updated successfully.

The screenshot shows the details for route table rtb-0d41bba2ed11c8401. The 'Routes' tab is selected, displaying two routes: one for '0.0.0.0/0' targeting the internet gateway 'igw-05cba0c2e0b35021e', and another for '10.0.0.16' targeting 'local'. The 'Actions' dropdown menu is visible on the right.

Part 5: Enabling Outbound Internet Access through NAT Gateway

1. View the current VPC architecture.



2. Click NAT Gateways.

Screenshot of the AWS VPC Route Tables page for route table `rtb-0d41bba2ed11c8401 / my-route-table-1`. The page shows the following details:

- Details**: Route table ID `rtb-0d41bba2ed11c8401`, Main: No, Owner ID: `536697231935`.
- Explicit subnet associations**: `subnet-04e9903eb59663bdb / my-public-subnet-1`.
- Edge associations**: None.
- Routes** (2):

Destination	Target	Status	Propagated	Route Origin
0.0.0.0/0	igw-05cba0c2e0b35021e	Active	No	Create Route
10.0.0.16	local	Active	No	Create Route

A red arrow points to the 'NAT gateways' link in the left sidebar under the 'Route tables' section.

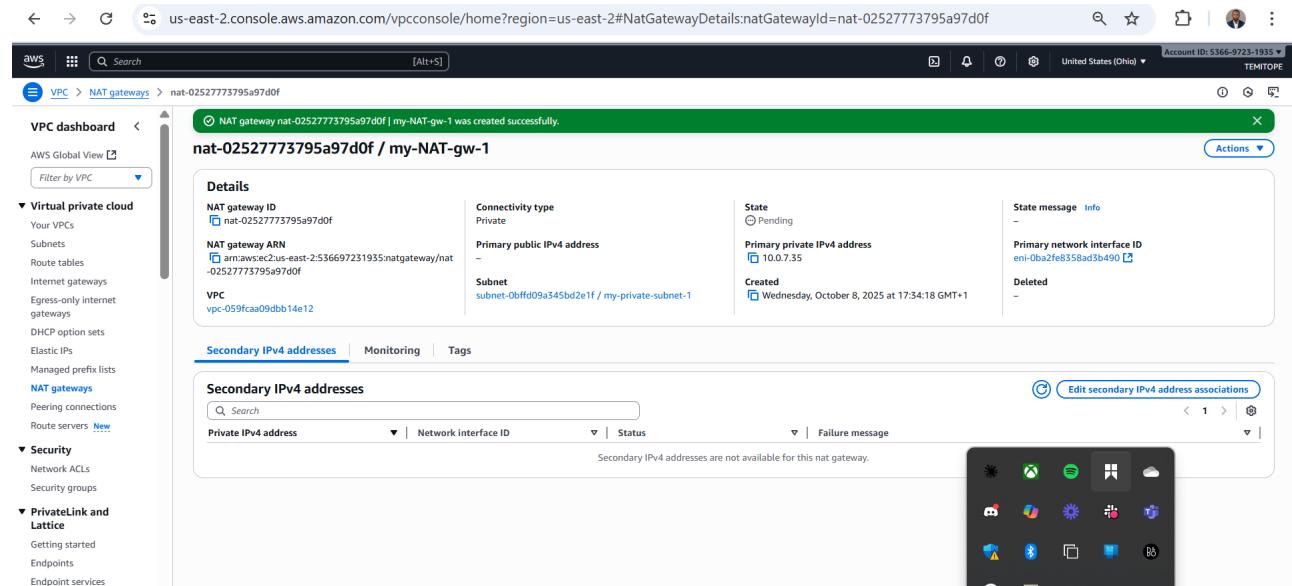
3. Click **Create NAT Gateway**.

The screenshot shows the AWS VPC NAT gateways page. On the left, there's a navigation sidebar with sections like 'Virtual private cloud', 'Security', and 'PrivateLink and Lattice'. The main area is titled 'NAT gateways info' and shows a table with columns: Name, NAT gateway ID, Connectivity..., State, State message, Primary public I..., Primary private I..., Primary network..., and VPC. A search bar at the top says 'Find NAT gateways by attribute or tag'. At the top right, there are 'Actions' and 'Create NAT gateway' buttons. A red arrow points to the 'Create NAT gateway' button.

4. Name the NAT Gateway, select the private subnet, set connectivity type to **Private**, and click **Create NAT Gateway**.

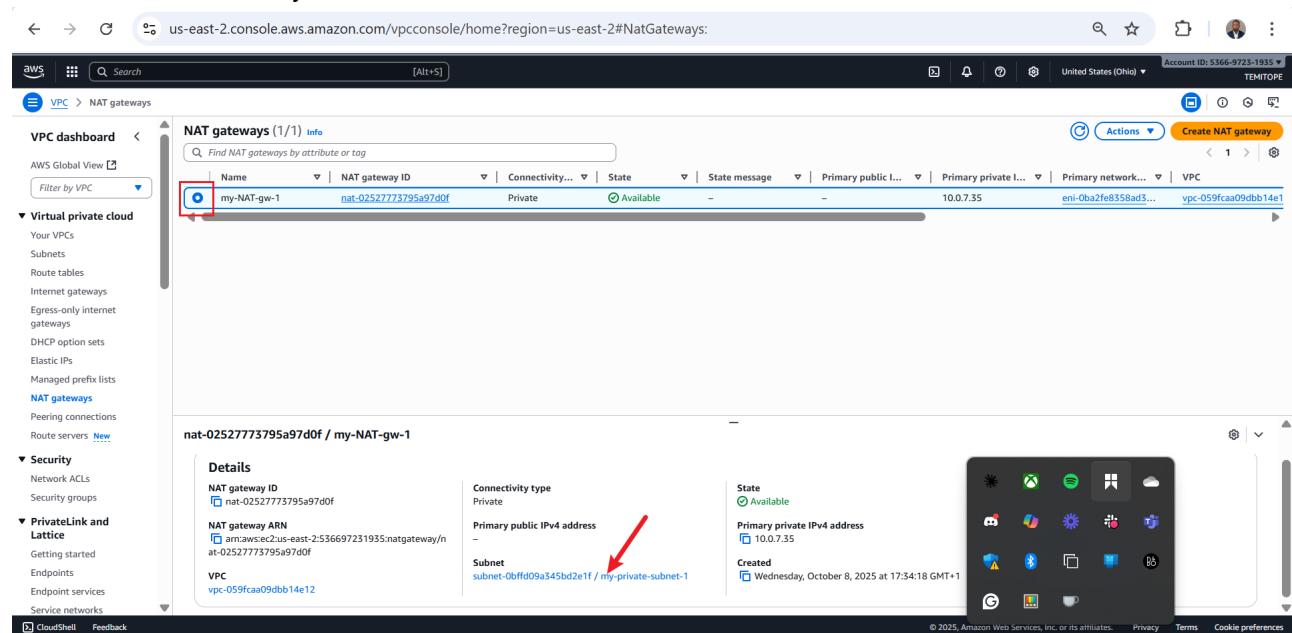
The screenshot shows the 'Create NAT gateway' wizard. The first step is 'NAT gateway settings'. It has fields for 'Name - optional' (with 'my-NAT-gw-1' entered), 'Subnet' (selected as 'subnet-0bffd09a345bd2e1f (my-private-subnet-1)'), and 'Connectivity type' (set to 'Private'). A note below says 'Private NAT gateway traffic can't reach the internet.' Below this is an 'Additional settings' section and a 'Tags' section where a tag 'Name: my-NAT-gw-1' is added. At the bottom right is a 'Create NAT gateway' button, which is highlighted with a red arrow.

5. NAT Gateway created successfully.



The screenshot shows the AWS VPC console with the URL us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#NatGatewayDetails:natGatewayId=nat-02527773795a97d0f. The main page displays a success message: "NAT gateway nat-02527773795a97d0f | my-NAT-gw-1 was created successfully." Below this, the "Details" tab is selected for the NAT gateway "nat-02527773795a97d0f / my-NAT-gw-1". The "Subnet" field shows "subnet-0bffd09a345bd2e1f / my-private-subnet-1". The "State" is listed as "Pending". Other tabs include "Secondary IPv4 addresses", "Monitoring", and "Tags". A sidebar on the left lists various VPC and security-related options. The bottom right corner shows the Windows taskbar with several pinned icons.

6. Select the NAT Gateway, locate the subnet ID in the Details tab, and click it.



The screenshot shows the AWS VPC console with the URL us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#NatGateways. The "NAT gateways (1/1) info" table lists one entry: "my-NAT-gw-1" with NAT gateway ID "nat-02527773795a97d0f". The "Actions" button is visible. Below the table, the "Details" tab is selected for the same NAT gateway. The "Subnet" field shows "subnet-0bffd09a345bd2e1f / my-private-subnet-1". A red arrow points from the "my-private-subnet-1" text to the "my-private-subnet-1" link. The bottom right corner shows the Windows taskbar with pinned icons.

7. Navigate to the **Route Table** section and click the route table ID.

Screenshots of the AWS VPC Subnets page:

- Subnets (1/1) Info** table:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR
my-private-subnet-1	subnet-0bffd09a345bd2e1f	Available	vpc-059fcaa09dbb14e12	Off	10.0.0.0/24	-	-
- subnet-0bffd09a345bd2e1f / my-private-subnet-1** details:
 - Route table:** rtb-01b8c1967af12ebbo (highlighted with a red box)
 - Routes (1):** Destination 10.0.0.0/16 Target local

8. Click **Routes**, then **Edit routes**.

Screenshots of the AWS Route Tables page:

- Route tables (1/1) Info** table:

Name	Route table ID	Explicit subnet assoc...	Edge associations	Main	VPC	Owner ID
-	rtb-01b8c1967af12ebbo	-	-	Yes	vpc-059fcaa09dbb14e12	536697231935
- rtb-01b8c1967af12ebbo** details:
 - Routes (1):** Destination 10.0.0.0/16 Target local Status Active Propagated No Route Origin Create Route Table
 - Edit routes** button (highlighted with a red box)

9. Click Add route.

The screenshot shows the 'Edit routes' section of a VPC route table. It displays one existing route: a destination of 10.0.0.0/16 pointing to a target of 'local'. The status is 'Active', propagation is 'No', and the route origin is 'CreateRouteTable'. At the bottom left of the table, there is a blue 'Add route' button, which is highlighted with a red arrow. On the right side of the screen, there is a standard Windows taskbar with various pinned icons.

10. Set Destination to **0.0.0.0/0**, Target to **NAT Gateway**, select the created NAT Gateway, and save changes.

The screenshot shows the 'Edit routes' section of a VPC route table. The 'Destination' field is set to 0.0.0.0/0. In the 'Target' dropdown, 'NAT Gateway' is selected, and a list of available NAT Gateways is shown. The 'Save changes' button at the bottom right is highlighted with a red arrow. The rest of the route table interface is visible, including the existing route for 10.0.0.0/16. On the right side of the screen, there is a standard Windows taskbar with various pinned icons.

11. Route table updated successfully, click **Subnet associations**, then **Edit subnet associations**.

Updated routes for rtb-01b8c1967af12eb0 successfully

rtb-01b8c1967af12eb0

Details **Info**

Route table ID: rtb-01b8c1967af12eb0
VPC: vpc-059fcaa09dbb14e12

Routes **Subnet associations** **Edge associations** **Route propagation** **Tags**

Explicit subnet associations (0)

No subnet associations

Subnets without explicit associations (1)

my-private-subnet-1 (subnet-0bffd09a345bd2e1f) 10.0.7.0/24

Edit subnet associations

12. Select the private subnet and click **Save associations**.

Edit subnet associations

Available subnets (1/2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/> my-private-subnet-1	subnet-0bffd09a345bd2e1f	10.0.7.0/24	-	Main (rtb-01b8c1967af12eb0)
<input type="checkbox"/> my-public-subnet-1	subnet-04e9903eb59663bdb	10.0.6.0/24	-	rtb-0d41bb2e

Selected subnets

subnet-0bffd09a345bd2e1f / my-private-subnet-1

Save associations

13. Subnet successfully attached to the route table.

You have successfully updated subnet associations for rtb-01b8c1967af12eb0.

rtb-01b8c1967af12eb0

Details **Info**

Route table ID: rtb-01b8c1967af12eb0
VPC: vpc-059fcaa09dbb14e12

Routes **Subnet associations** **Edge associations** **Route propagation** **Tags**

Explicit subnet associations

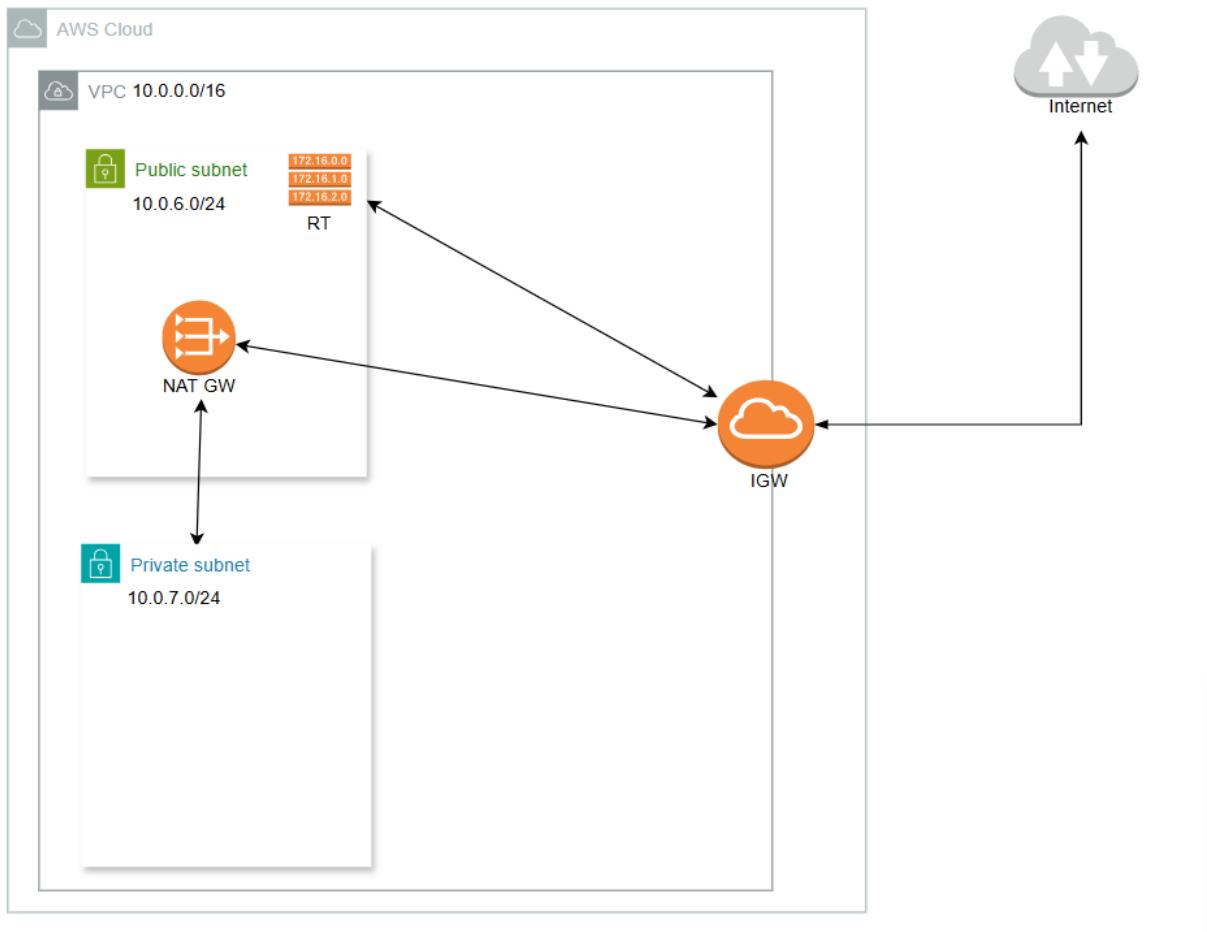
subnet-0bffd09a345bd2e1f / my-private-subnet-1

Routes (2)

Destination	Target	Status	Propagated	Route Origin
0.0.0.0/0	nat-02527773795a97d0f	Active	No	Create Route
10.0.0.0/16	local	Active	No	Create Route Table

Edit routes

14. View the current VPC architecture.



15. Create an Ubuntu EC2 instance named **Public-EC2**.

<https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstances>

The screenshot shows the 'Launch an instance' wizard. In the 'Name and tags' section, the instance is named 'Public-EC2'. Under 'Application and OS Images (Amazon Machine Image)', the 'Quick Start' tab is selected, showing options for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. An 'Amazon Machine Image (AMI)' section details the selection of 'Ubuntu Server 24.04 LTS (HVM), SSD Volume Type' (ami-0cfddealed312d4). The 'Summary' section indicates 1 instance will be launched. The 'Free tier' information notes 750 hours per month of t2.micro usage. The bottom of the screen includes standard AWS navigation links like cloudShell, Feedback, Privacy, Terms, and Cookie preferences.

16. In network settings, click **Edit**.

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Network settings' section, there is a 'Key pair name - required' dropdown set to 'temskey1'. Below it, under 'Network', is a dropdown for 'VPC' set to 'vpc-0a94646cd2d5bf4cbe'. Under 'Subnet', it shows 'my-public-subnet-1' with 'CIDR: 10.0.0.0/16'. The 'Auto-assign public IP' checkbox is checked. In the bottom right of the 'Network settings' box, there is an 'Edit' button with a red arrow pointing to it. To the right of the main form is a 'Summary' panel showing 'Number of instances: 1'. Below that are sections for 'Software Image (AMI)', 'Virtual server type (instance type)', 'Firewall (security group)', and 'Storage (volumes)'. At the bottom right of the summary panel is a 'Launch' button.

17. Set VPC to 10.0.0.0/16, subnet to the public subnet, enable **Auto-assign public IP**, and launch instance.

This screenshot continues the 'Launch an instance' wizard. The 'Network settings' section has been updated: the 'VPC' dropdown now shows 'vpc-059ca09dbb14e12' with 'CIDR: 10.0.0.0/16', and the 'Subnet' dropdown shows 'my-public-subnet-1' with 'CIDR: 10.0.0.0/16'. The 'Auto-assign public IP' section has the 'Enable' checkbox checked. A red box highlights the 'VPC' and 'Subnet' fields. A red arrow points to the 'Enable' checkbox. In the bottom right of the 'Network settings' box is a 'Launch' button. To the right of the main form is a 'Summary' panel with the same information as the previous screenshot, including the 'Free tier' note and the 'Launch' button.

18. Create an Ubuntu EC2 instance named Private-EC2.

The screenshot shows the 'Launch an instance' wizard on the AWS EC2 console. In the 'Name and tags' step, a tag 'Private-EC2' is added. The 'Summary' panel on the right shows the instance configuration: Software Image (AMI) is Canonical, Ubuntu, 24.04, amd64; Virtual server type is t2.micro; Storage (volumes) is 1 volume(s) - 8 GiB. A tooltip points to the 'Edit' button in the summary panel, which is part of a larger callout bubble.

19. In network settings, click Edit.

The screenshot shows the 'Network settings' step of the 'Launch an instance' wizard. It includes fields for Network (set to 'vpc-0a946dd2d25bf4cbe'), Subnet (no preference), Auto-assign public IP (enabled), and Firewall (security groups). A red arrow points to the 'Edit' button in the summary panel, which is part of a larger callout bubble.

20. Set VPC to 10.0.0.0/16, subnet to the private subnet, and launch instance.

Network settings

VPC: vpc-059fcaa09dbb14e12 (10.0.0.0/16)

Subnet: my-private-subnet-1 (VPC: vpc-059fcaa09dbb14e12, Owner: 536697231935, Availability Zone: us-east-2a (use2-az1), Zone type: Availability Zone, IP addresses available: 250, CIDR: 10.0.7.0/24)

Auto-assign public IP: Disable

Firewall (security groups): Create security group

Description - required: launch-wizard-6

Inbound Security Group Rules:

- Security group rule 1 (TCP: 22, 0.0.0.0/0)

Type	Protocol	Port range
ssh	TCP	22

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd64... (ami-0cfde0a8ed312c04)

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volume): 1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs. 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet. Data transfer charges are not included as part of the free tier allowance.

Launch instance

21. Both EC2 instances created, click the Public-EC2 ID.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
Public-EC2	i-0616ddda467c914e62	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a	-	-	-
Private-EC2	i-096caf021a83bda8	Running	t2.micro	-	View alarms +	us-east-2a	-	-	-

Select an instance

22. Copy the public IP for remote connection.

The screenshot shows the AWS Management Console with the EC2 service selected. In the left sidebar, 'Instances' is expanded, showing options like Instances, Instance Types, Launch Templates, and so on. The main pane displays the 'Instance summary for i-0e182c5c5bd39ce3a (Public-EC2)'. Key details shown include:

- Public IPv4 address:** 3.145.105.141 (with a link to open address)
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-10-0-6-194.us-east-2.compute.internal
- Instance type:** t2.micro
- VPC ID:** vpc-059fcaa09dbb14e12
- Subnet ID:** subnet-04e9903eb59663bdb (my-public-subnet-1)
- Instance ARN:** arn:aws:ec2:us-east-2:536697231935:instance/i-0e182c5c5bd39ce3a

Below the summary, there are tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The Monitoring tab is currently selected, showing 'disabled'.

23. Connect via SSH.

The terminal window title is 'MINGW64/c/Users/HP/Downloads'. The session output is as follows:

```

ED25519 key fingerprint is SHA256:xeXdkC7+gbfubqTs9ztyx+zyiThUI+z9PgbCNICpk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.145.105.141' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:   https://landscape.canonical.com
 * Support:      https://ubuntu.com/pro

System information as of Thu Oct  9 16:29:41 UTC 2025

  System load:  0.0          Processes:           108
  Usage of /:   25.6% of 6.71GB   Users logged in:    0
  Memory usage: 20%            IPv4 address for enx0: 10.0.6.194
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

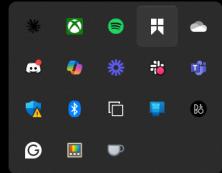
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-6-194:~$ 

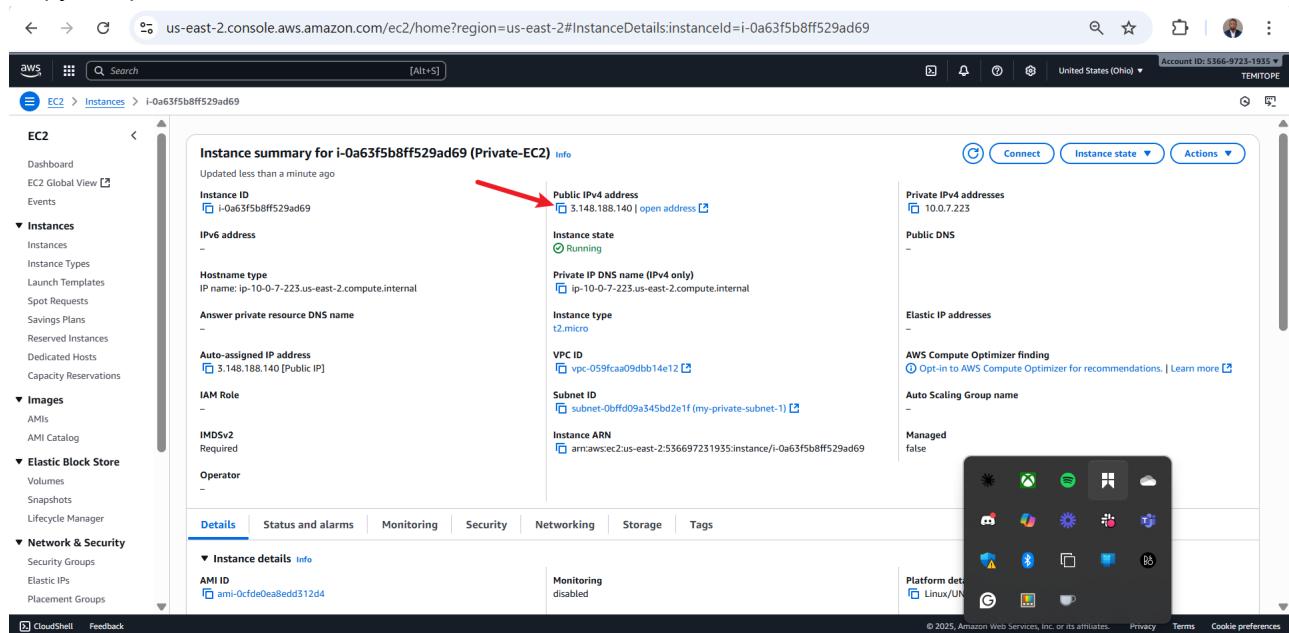
```

24. Test outbound traffic with `git clone`.



```
ubuntu@ip-10-0-6-194:~$ git --version
git version 2.43.0
ubuntu@ip-10-0-6-194:~$ git clone https://github.com/oluwaseunoa/hng-stage-1-task.git
Cloning into 'hng-stage-1-task'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 27 (delta 12), reused 20 (delta 8), pack-reused 0 (from 0)
Receiving objects: 100% (27/27), 17.05 KiB | 3.41 MiB/s, done.
Resolving deltas: 100% (12/12), done.
ubuntu@ip-10-0-6-194:~$ ls hng-stage-1-task/
README.md index.js package-lock.json package.json
ubuntu@ip-10-0-6-194:~$
```

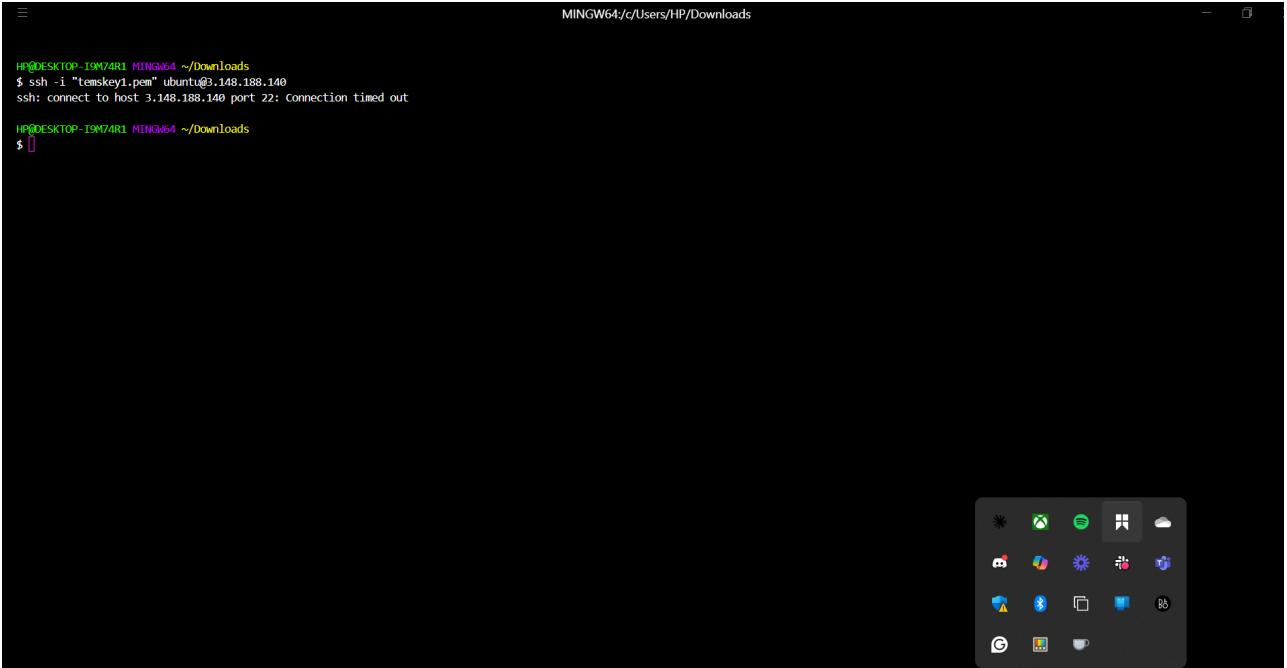
25. Copy the public IP of `Private-EC2` for remote connection (note: connection should fail).



Instance summary for i-0a63f5b8ff529ad69 (Private-EC2)

- Public IPv4 address: 3.148.188.140
- Instance ID: i-0a63f5b8ff529ad69
- Hostname type: ip-10-0-7-223.us-east-2.compute.internal
- Auto-assigned IP address: 3.148.188.140 [Public IP]
- VPC ID: vpc-059fcaa09dbb14e12
- Subnet ID: subnet-0bffd09a345bd2e1f (my-private-subnet-1)
- Instance ARN: arn:aws:ec2:us-east-2:536697231935:instance/i-0a63f5b8ff529ad69

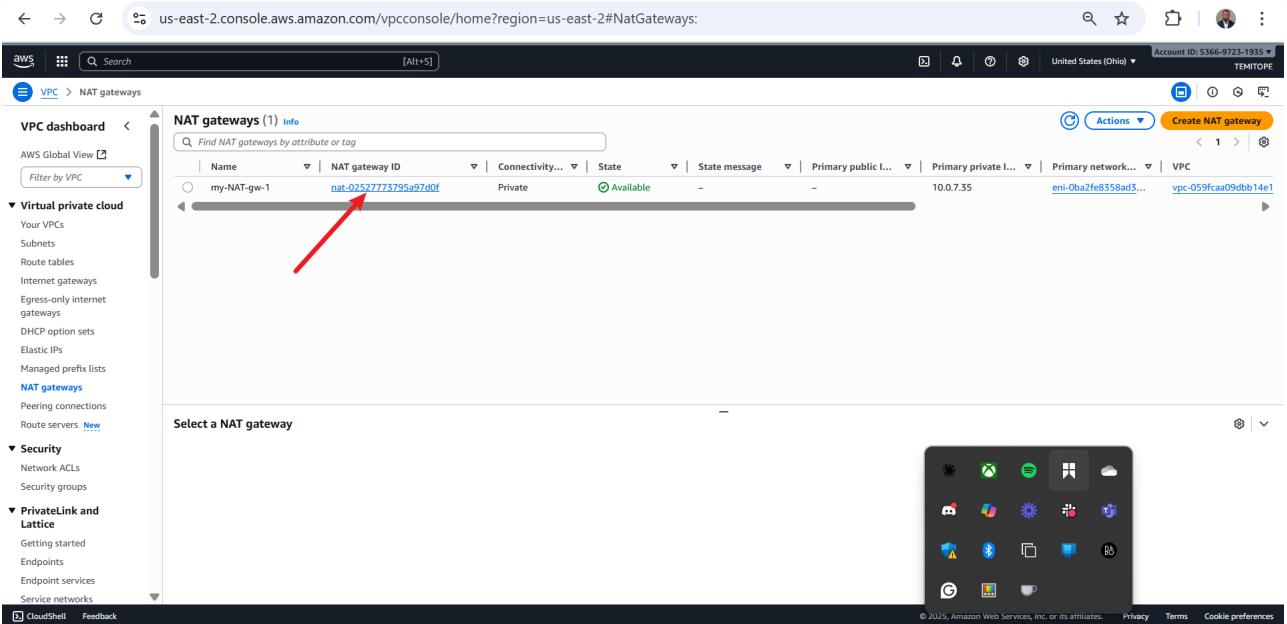
26. Unable to connect to **Private-EC2** over the internet.



```
HPDESKTOP-I9M74R1 MINGW64 ~/Downloads
$ ssh -i "tenskey1.pem" ubuntu@3.148.188.140
ssh: connect to host 3.148.188.140 port 22: Connection timed out

HPDESKTOP-I9M74R1 MINGW64 ~/Downloads
$ [ ]
```

27. Verify NAT Gateway functionality by navigating to **NAT Gateways** and selecting the created NAT Gateway.



The screenshot shows the AWS VPC console with the 'NAT gateways' page. The left sidebar includes sections for VPC dashboard, Virtual private cloud, Security, PrivateLink and Latency, and CloudShell/Feedback. The main area displays a table titled 'NAT gateways (1) Info' with one entry:

Name	NAT gateway ID	Connectivity...	State	Primary public I...	Primary private I...	Primary network...
my-NAT-gw-1	nat-02527773795a97d0f	Private	Available	-	10.0.7.55	eni-0ba2fe8358ad3... vpc-059fcaa09dbb14e1

A red arrow points to the NAT gateway ID 'nat-02527773795a97d0f'. The bottom right corner of the screen shows the Windows taskbar with various pinned icons.

28. Observe outbound traffic data in **Packet Out to Destination** and **Byte Out to Destination**.

The screenshot shows the AWS VPC NAT gateway details page for a specific gateway. The left sidebar includes sections for VPC dashboard, Virtual private cloud, Security, PrivateLink and Lattice, and CloudShell. The main content area displays the NAT gateway ID (nat-02527773795a97d0f), ARN (arn:aws:ec2:us-east-2:536697231935:natgateway/nat-02527773795a97d0f), and VPC (vpc-059fcaa9dbb14e12). The 'Monitoring' tab is selected, showing two line charts: 'Packets out to destination (Count)' and 'Bytes out to destination (Bytes)'. Both charts show data from 16:15 to 17:00. The 'Packets out to destination' chart has values around 168, 336, and 180. The 'Bytes out to destination' chart has values around 10.9k, 21.9k, and 11.9k.

29. Create another EC2 instance to test connectivity within the same subnet.

The screenshot shows the 'Launch an instance' wizard. The 'Name and tags' step shows a name 'Public-EC2-2'. The 'Application and OS Images (Amazon Machine Image)' step shows a search bar and a grid of AMI icons for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. A tooltip for the Ubuntu AMI provides details: 'Ubuntu Server 24.04 LTS (HVM), SSD Volume Type' and 'Free tier eligible'. The 'Virtual server type (instance type)' step shows a tooltip for the t2.micro instance type: 'Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs. 750 hours of compute usage, 700 hours of address usage, 30 GiB of snapshots, and 10 GiB of data transfer chargeable at the standard tier allowance.' The right sidebar shows a summary of the instance configuration, including the number of instances (1), software image (Canonical, Ubuntu, 24.04, amd64), virtual server type (t2.micro), firewall (New security group), and storage (1 volume(s) - 8 GiB).

30. Set VPC to 10.0.0.0/16, subnet to the public subnet, and launch instance.

Network settings

VPC - required Info
vpc-059caa09dbb14e12 10.0.0.0/16

Subnet Info
subnet-04e920d8eb29630db my-public-subnet-1
VPC: vpc-059caa09dbb14e12 Owner: 536697231935 Availability Zone: us-east-2a (use2-az1) Zone type: Availability Zone IP addresses available: 250 CIDR: 10.0.6.0/24

Create new subnet

Auto-assign public IP Info
Enable Additional charges apply when outside of free tier allowance

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
 Create security group Select existing security group

Security group name - required
launch-wizard-9

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-~!@#\$%^&*()

Description - required Info
launch-wizard-9 created 2025-10-09T21:47:33.068Z

Inbound Security Group Rules
▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)
Type | Info Protocol | Info Port range | Info
ssh TCP 22

Remove

Summary
Number of instances | Info
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64... [read more](#)
ami-0cfde0eaed6312c4

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs. 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GiB of bandwidth to the internet. Data transfer charges are not included as part of the free tier allowance.

Launch Instance [Preview code](#)

31. Two EC2 instances now in the public subnet.

Instances (1/3) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
Private-EC2	i-063f5b8ff529ad69	Stopped	t2.micro	-	View alarms +	us-east-2a	-	-	-
Public-EC2-2	i-076053c6f003a065e	Pending	t2.micro	-	View alarms +	us-east-2a	-	3.148.164.208	-
Public-EC2	i-0e182c5c5bd39ce3a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a	-	3.136.233.109	-

i-0e182c5c5bd39ce3a (Public-EC2)

Details [Status and alarms](#) [Monitoring](#) [Security](#) [Networking](#) [Storage](#) [Tags](#)

Instance summary [Info](#)

Instance ID: i-0e182c5c5bd39ce3a

Public IPv4 address: 3.136.233.109 [open address](#)

Private IPv4 address: 10.0.6.1

32. Click the new Public-EC2.

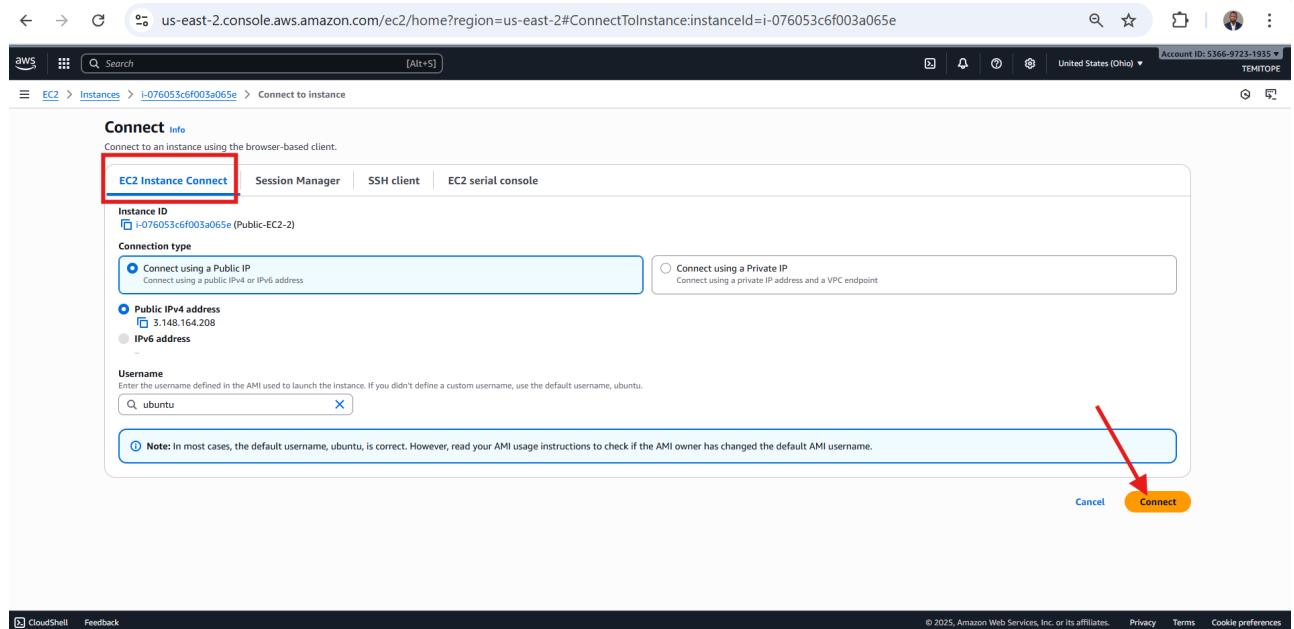
The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances (selected), Images, Elastic Block Store, Network & Security, and more. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
Private-EC2	i-0a63f5b8ff529ad69	Stopped	t2.micro	-	View alarms	us-east-2a	-	-	-
Public-EC2-2	i-076053c6f003a065e	Running	t2.micro	Initializing	View alarms	us-east-2a	-	3.148.164.208	-
Public-EC2	i-0e182c5c5bd39ce3a	Running	t2.micro	2/2 checks passed	View alarms	us-east-2a	-	3.136.233.109	-

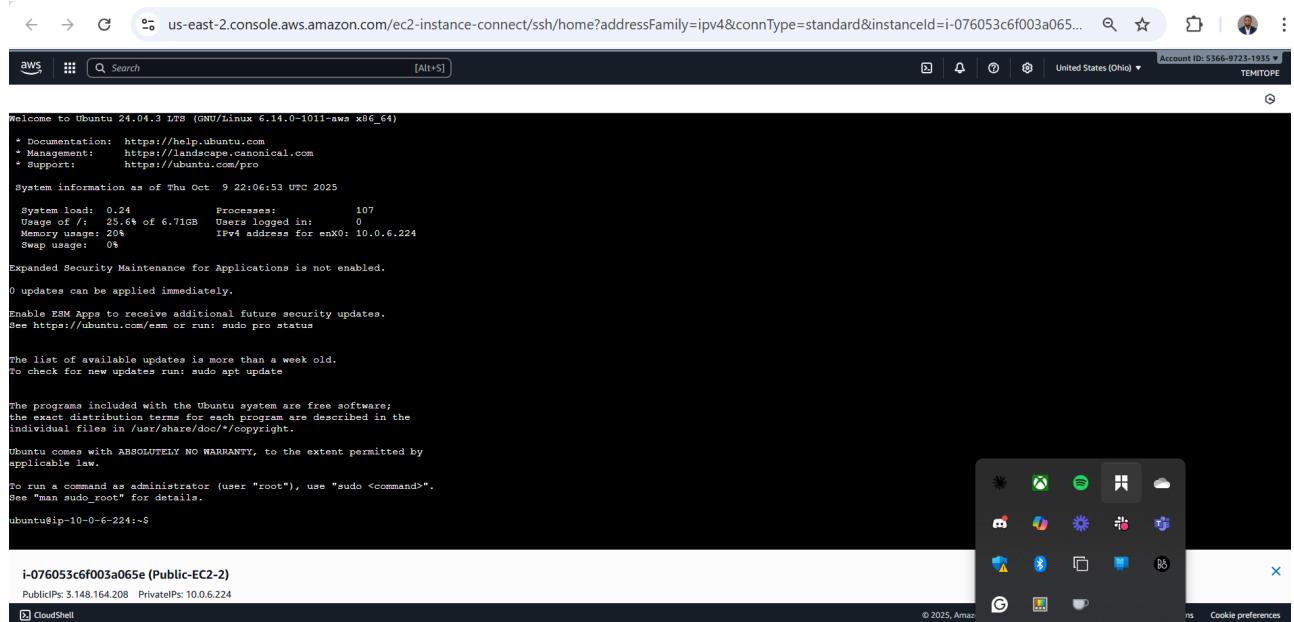
33. Click Connect.

The screenshot shows the AWS EC2 Instance summary page for instance i-076053c6f003a065e. The left sidebar is identical to the previous screenshot. The main content area is titled "Instance summary for i-076053c6f003a065e (Public-EC2-2)" and contains detailed information about the instance, including its public and private IP addresses, instance state, and various identifiers. At the top right of this summary page, there is a "Connect" button, which is highlighted with a red arrow.

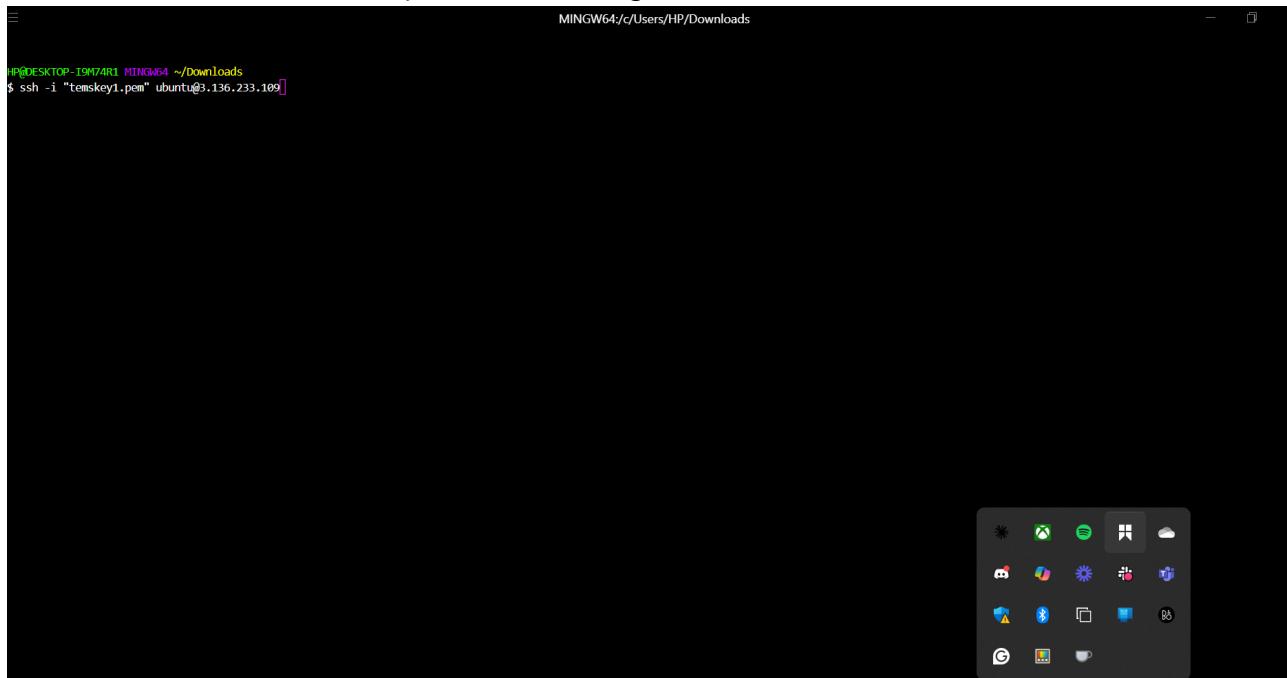
34. Select EC2 Instance Connect tab and click Connect.



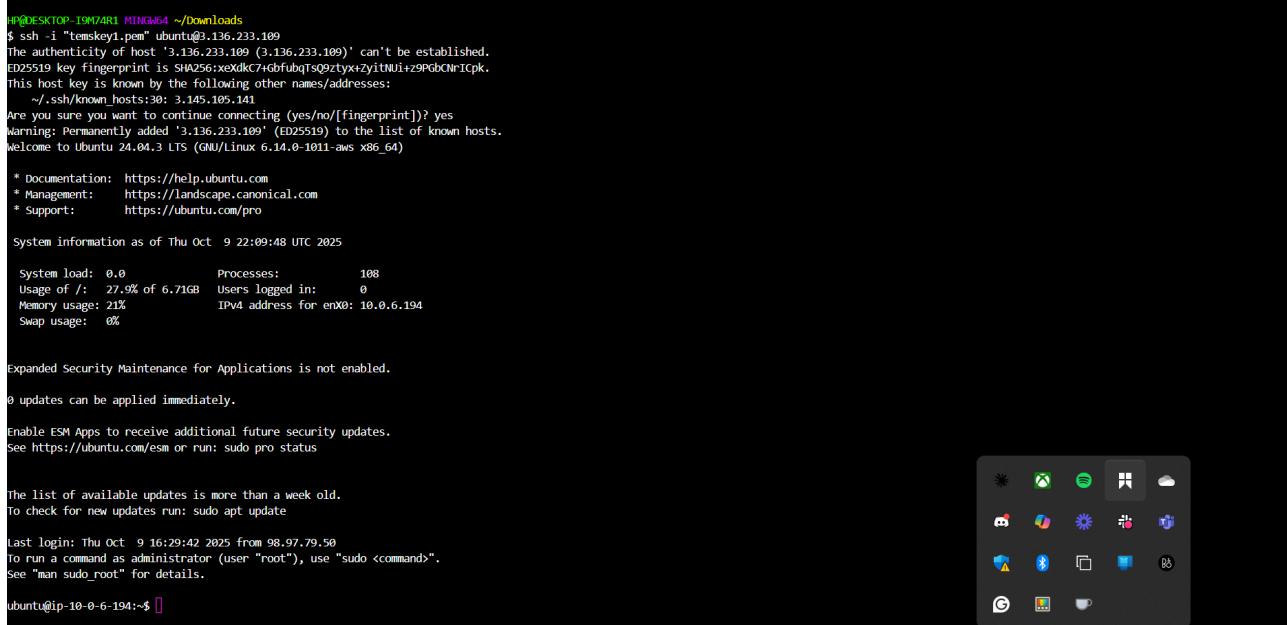
35. Successfully connected via browser.



36. Connect to the other EC2 in the public subnet using a terminal.



37. Successfully connect via terminal.



```
HPDESKTOP-T9M74R1 MINGW64 ~/Downloads
$ ssh -i "temskey1.pem" ubuntu@3.136.233.109
The authenticity of host '3.136.233.109 (3.136.233.109)' can't be established.
ED25519 key fingerprint is SHA256:xeXkC7+gbfubqTsQ9zytxyzitNUi+z9PgbcNRIcpk.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:30: 3.145.105.141
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.136.233.109' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Thu Oct 9 22:09:48 UTC 2025

System load: 0.0          Processes:      108
Usage of /: 27.9% of 6.71GB  Users logged in: 0
Memory usage: 21%          IPv4 address for enx0: 10.0.6.194
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

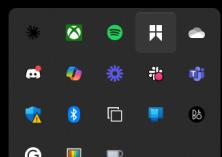
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Thu Oct 9 16:29:42 2025 from 98.97.79.50
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

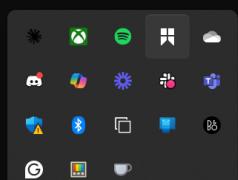
ubuntu@ip-10-0-6-194:~$
```

38. Ping the second EC2 to ensure reachability.



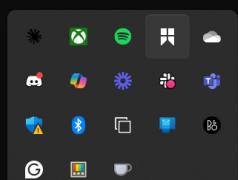
```
ubuntu@ip-10-0-6-194:~$ ping 3.148.164.208
```

39. Move the key pair to EC2 using SCP to SSH into the second EC2.



```
C:\Users\HP\Downloads>scp -i "temskey1.pem" "temskey1.pem" ubuntu@3.136.233.109:/home/ubuntu/
temskey1.pem
100% 1678      6.4KB/s   00:00
C:\Users\HP\Downloads>
```

40. Confirm key pair copied and SSH into [Public-EC2-2](#) from [Public-EC2](#).



```
ubuntu@ip-10-0-6-194:~$ ls temskey1.pem
temskey1.pem
ubuntu@ip-10-0-6-194:~$ ssh -i "temskey1.pem" ubuntu@3.148.164.208
```

41. Successfully SSH into Public-EC2-2.

```
WARNING: UNPROTECTED PRIVATE KEY FILE! @
Permissions 0444 for 'temsky1.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "temsky1.pem": bad permissions
ubuntu@3.148.164.208: Permission denied (publickey).
ubuntu@ip-10-0-6-194:~$ chmod 400 temsky1.pem
ubuntu@ip-10-0-6-194:~$ ssh -i "temsky1.pem" ubuntu@3.148.164.208
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Thu Oct  9 22:35:34 UTC 2025

 System load:  0.0          Processes:           114
 Usage of:   26.2% of 6.71GB  Users logged in:      1
 Memory usage: 22%          IPv4 address for eth0: 10.0.6.224
 Swap usage:   0%          

Expanded Security Maintenance for Applications is not enabled.

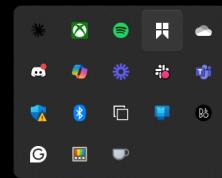
0 updates can be applied immediately.

Enable ESM apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

last login: Thu Oct  9 22:06:54 2025 from 3.16.146.4
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-6-224:~$ █
```



42. Log out of both instances.

```
ubuntu@ip-10-0-6-194:~$ ssh -l "temsky1.pem" ubuntu@3.148.164.208
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Thu Oct  9 22:35:34 UTC 2025

 System load: 0.0          Processes:           114
 Usage of /: 26.2% of 6.71GB  Users logged in:      1
 Memory usage: 22%
 Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

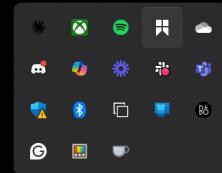
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Thu Oct  9 22:06:54 2025 from 3.16.146.4
To run a command as administrator (user "root"), use "sudo command"
see "man sudo root" for details.

ubuntu@ip-10-0-6-224:~$ ^C
ubuntu@ip-10-0-6-224:~$ exit
logout
Connection to 3.148.164.208 closed.
ubuntu@ip-10-0-6-194:~$ exit
logout
Connection to 3.136.233.109 closed.

[IPADDRESS] MINGW64 ~/Downloads
```



Part 6: Establishing VPC Peering Connections

1. Create the requester VPC (192.168.0.0/16).

The screenshot shows the 'Create VPC' wizard on the AWS VPC console. The 'VPC settings' section is selected. Under 'Resources to create', 'VPC only' is chosen. A tag 'requester-VPC' is added under 'Name tag - optional'. The 'IPv4 CIDR' field contains '192.168.0.0/16'. The 'Tags' section shows a tag 'requester-VPC' being added. A red arrow points to the 'Create VPC' button at the bottom right.

2. Create the accepter VPC (172.16.0.0/16).

The screenshot shows the 'Create VPC' wizard on the AWS VPC console. The 'VPC settings' section is selected. Under 'Resources to create', 'VPC only' is chosen. A tag 'accepter-VPC' is added under 'Name tag - optional'. The 'IPv4 CIDR' field contains '172.16.0.0/16'. The 'Tags' section shows a tag 'accepter-VPC' being added. A red arrow points to the 'Create VPC' button at the bottom right.

3. Both VPCs created, navigate to Peering Connections.

The screenshot shows the AWS VPC console with the URL <https://us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#vpcs>. The left sidebar is expanded to show 'Your VPCs'. The main table lists four VPCs:

Name	VPC ID	Status	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table	Main network ACL	Tenancy	Default VPC	On
-	vpc-039fca09d0a1e12	Available	Off	10.0.0.0/16	-	dopt-09149d5972cad...	rb-011b1967af12eb90	ad-046802252bea5akb	default	No	5.3t
-	vpc-0a346cd2dfbfbfe	Available	Off	172.31.0.0/16	-	dopt-09149d5972cad...	rb-025cb4ee01e7229e	ad-0438bcfa99ea7%8	default	Yes	5.3t
requester-VPC	vpc-0777a0164ff1ab4	Available	Off	192.168.0.0/16	-	dopt-09149d5972cad...	-	-	default	No	5.3t
accepter-VPC	vpc-0d05919128bc0f3ed	Available	Off	172.16.0.0/16	-	dopt-09149d5972cad...	-	-	default	No	5.3t

4. Click Create Peering Connection.

The screenshot shows the AWS VPC console with the URL <https://us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#PeeringConnections>. The left sidebar is expanded to show 'Peering connections'. The main table shows no existing peering connections:

Name	Peering connection ID	Status	Requester VPC	Acceptor VPC	Requester CIDRs	Acceptor CIDRs	Requester c
No peering connection found							

5. Name the peering connection, select requester VPC, choose **My account**, select accepter VPC, and click **Create Peering Connection**.

Create peering connection

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately. [Info](#)

Peering connection settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.
my-first-vpc-peering

Select a local VPC to peer with

VPC ID (Requester)
vpc-0777a0c1649ff1ab4 (requester-VPC)

VPC CIDRs for vpc-0777a0c1649ff1ab4 (requester-VPC)

CIDR	Status	Status reason
192.168.0.0/16	Associated	-

Select another VPC to peer with

Account
 My account
 Another account

Region
 This Region (us-east-2)
 Another Region

VPC ID (Acceptor)
vpc-0c059193282cd03e6 (accepter-VPC)

VPC CIDRs for vpc-0c059193282cd03e6 (accepter-VPC)

CIDR	Status	Status reason
172.16.0.0/16	Associated	-

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 Name my-first-vpc-peering

Add new tag

You can add 49 more tags.

6. Scroll down and create the peering connection.

Select another VPC to peer with

Account
 My account
 Another account

Region
 This Region (us-east-2)
 Another Region

VPC ID (Acceptor)
vpc-0c059193282cd03e6 (accepter-VPC)

VPC CIDRs for vpc-0c059193282cd03e6 (accepter-VPC)

CIDR	Status	Status reason
172.16.0.0/16	Associated	-

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 Name my-first-vpc-peering

Add new tag

You can add 49 more tags.

7. VPC peering connection created, click Actions and Accept Request.

VPC dashboard <

AWS Global View

Virtual private cloud

- Your VPCs
- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections
- Route servers

Security

- Network ACLs
- Security groups

PrivateLink and Lattice

- Getting started
- Endpoints
- Endpoint services
- Service networks

Peering connections

Route servers

Details

Requester owner ID 536697231935

Peer connection ID pxc-0f6e1f02cfce7b81e

Status Pending acceptance by 536697231935

Expiration time Friday, October 17, 2025 at 00:07:05 GMT+1

DNS
Route tables
Tags

pxc-0f6e1f02cfce7b81e / my-first-vpc-peering

Pending acceptance
You can accept or reject this peering connection request using the 'Actions' menu. You have until Friday, October 17, 2025 at 00:07:05 GMT+1 to accept or reject the request, otherwise it expires.

Actions

- Accept request**
- Reject request
- Edit DNS settings
- Manage tags
- Delete peering connection

VPC Peering connection ARN arn:aws:ec2:us-east-2:536697231935:pxc-0f6e1f02cfce7b81e

Acceptor VPC vpc-0c059193282cd03e6 / accepter-VPC

Requester CIDRs 192.168.0.0/16

Requester Region Ohio (us-east-2)

Acceptor CIDRs

Acceptor Region Ohio (us-east-2)

8. Click Accept Request in the pop-up.

AWS Global View Actions ▾

VPC dashboard < Virtual private cloud Your VPCs Subnets Route tables Internet gateways Egress-only internet gateways DHCP option sets Elastic IPs Managed prefix lists NAT gateways Peering connections Route servers New Security Network ACLs Security groups PrivateLink and Lattice Getting started Endpoints Endpoint services Service networks CloudShell Feedback

aws Search [Alt+S] Account ID: 5366-9723-1935 United States (Ohio) TEMPORARY

Peering connections > pcx-0f6e1f02cfce7b81e

A VPC peering connection pcx-0f6e1f02cfce7b81e / my-first-vpc-peering has been requested.

pcx-0f6e1f02cfce7b81e / my-first-vpc-peering

Pending acceptance You can accept or reject this peering connection request using the 'Actions' menu. You have until Friday, October 17, 2025 at 00:07:05 GMT+1 to accept or reject the request, otherwise it expires.

Details Info

Requester owner ID [536697231935](#)
Peer connection ID [pcx-0f6e1f02cfce7b81e](#)
Status .Pending Acceptance by 536697231935
Expiration time Friday, October 17, 2025 at 00:07:05 GMT+1

Accepter owner ID [536697231935](#) **VPC Peering connection ARN** [us-east-2:536697231935:vpc-peering-connection/pcx-0f6e1f02cfce7b81e](#)

Accept VPC peering connection request Info

Are you sure you want to accept this VPC peering connection request? (pcx-0f6e1f02cfce7b81e / my-first-vpc-peering)

Requester VPC [vpc-0777a0c1649ff1ab4 / requester-VPC](#) **Acceptor VPC** [vpc-0c059193282cd03e6 / accepter-VPC](#)
Requester CIDRs [192.168.0.0/16](#) **Acceptor CIDRs** [-](#)
Requester Region Ohio (us-east-2) **Acceptor Region** Ohio (us-east-2)
Requester owner ID [536697231935](#) (This account) **Acceptor owner ID** [536697231935](#) (This account)

Cancel **Accept request** Edit DNS settings

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

9. VPC peering connection established, click VPC.

AWS Global View Actions ▾

VPC dashboard < Virtual private cloud Your VPCs Subnets Route tables Internet gateways Egress-only internet gateways DHCP option sets Elastic IPs Managed prefix lists NAT gateways Peering connections Route servers New Security Network ACLs Security groups PrivateLink and Lattice Getting started Endpoints Endpoint services Service networks CloudShell Feedback

aws Search [Alt+S] Account ID: 5366-9723-1935 United States (Ohio) TEMPORARY

Peering connections > pcx-0f6e1f02cfce7b81e

Your VPC peering connection (pcx-0f6e1f02cfce7b81e | my-first-vpc-peering) has been established.

To send and receive traffic across this VPC peering connection, you must add a route to the peered VPC in one or more of your VPC route tables. [Info](#) [Modify my route tables now](#)

pcx-0f6e1f02cfce7b81e / my-first-vpc-peering

Details Info

Requester owner ID [536697231935](#)
Peer connection ID [pcx-0f6e1f02cfce7b81e](#)
Status Active
Expiration time [-](#)

Accepter owner ID [536697231935](#) **VPC Peering connection ARN** [arn:aws:ec2:us-east-2:536697231935:vpc-peering-connection/pcx-0f6e1f02cfce7b81e](#)

Requester VPC [vpc-0777a0c1649ff1ab4 / requester-VPC](#) **Acceptor VPC** [vpc-0c059193282cd03e6 / accepter-VPC](#)
Requester CIDRs [192.168.0.0/16](#) **Acceptor CIDRs** [172.16.0.0/16](#)
Requester Region Ohio (us-east-2) **Acceptor Region** Ohio (us-east-2)

DNS Route tables Tags Edit DNS settings

DNS settings

Requester VPC ([vpc-0777a0c1649ff1ab4 / requester-VPC](#)) Info
Allow accepter VPC to resolve DNS of hosts in requester VPC to private IP addresses Enabled

Acceptor VPC ([vpc-0c059193282cd03e6 / accepter-VPC](#)) Info
Allow requester VPC to resolve DNS of hosts in accepter VPC to private IP addresses Enabled

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

10. Click the accepter VPC's main route table ID.

Your VPCs (1/4) [Info](#)

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table	Main network ACL	Tenancy	Default V
accepter-VPC	vpc-0c059193282cd035e6	Available	<input type="radio"/> Off	172.16.0.0/16	-	dopt-09149d5972cad...	rtb-0d6e66b374058101b	acl-096a1bf0d05d5a4ea	default	No
-	vpc-059fcac09dbb14e12	Available	<input type="radio"/> Off	10.0.0.0/16	-	dopt-09149d5972cad...	rtb-01ba8c1967af12eb0	acl-0dd802252bcd3adcb	default	No
requester-VPC	vpc-0777a0c1649ff1ab4	Available	<input type="radio"/> Off	192.168.0.0/16	-	dopt-09149d5972cad...	rtb-0a7c459e61b1aa146	acl-0cc1fa0ffe65727d8	default	No
-	vpc-0a946dcf2d5bf4ce	Available	<input type="radio"/> Off	172.31.0.0/16	-	dopt-09149d5972cad...	rtb-025c8a4ee81e7226e	acl-0c388cfaf59ea79c8	default	Yes

vpc-0777a0c1649ff1ab4 / requester-VPC

[Details](#) | [Resource map](#) | [CIDRs](#) | [Flow logs](#) | [Tags](#) | [Integrations](#)

Details

VPC ID: vpc-0777a0c1649ff1ab4 State: Available

DNS resolution: Enabled

Block Public Access: OFF

DHCP option set: dopt-09149d5972cad5d6b

Tenancy: default

Owner ID: 536697231955

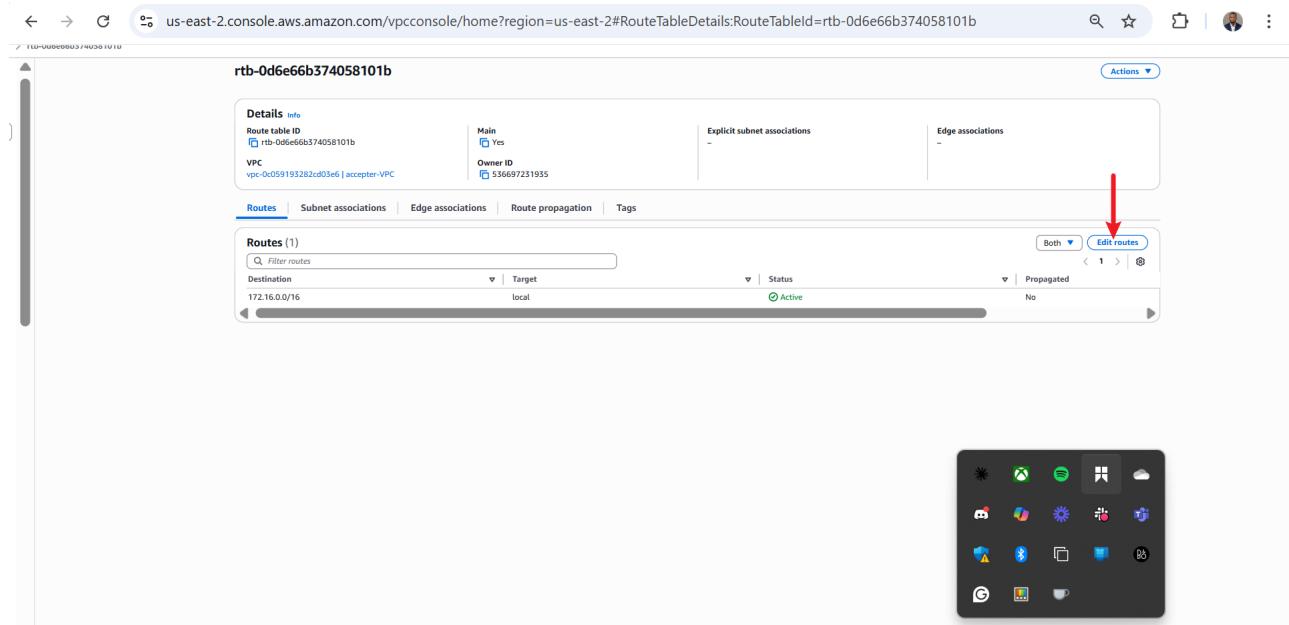
11. Click the route table ID.

Route tables (1) [Info](#)

Name	Route table ID	Explicit subnet assoc...	Edge associations	Main	VPC	Owner ID
-	rtb-0d6e66b374058101b	-	-	Yes	vpc-0c059193282cd035e6 acce...	536697231955

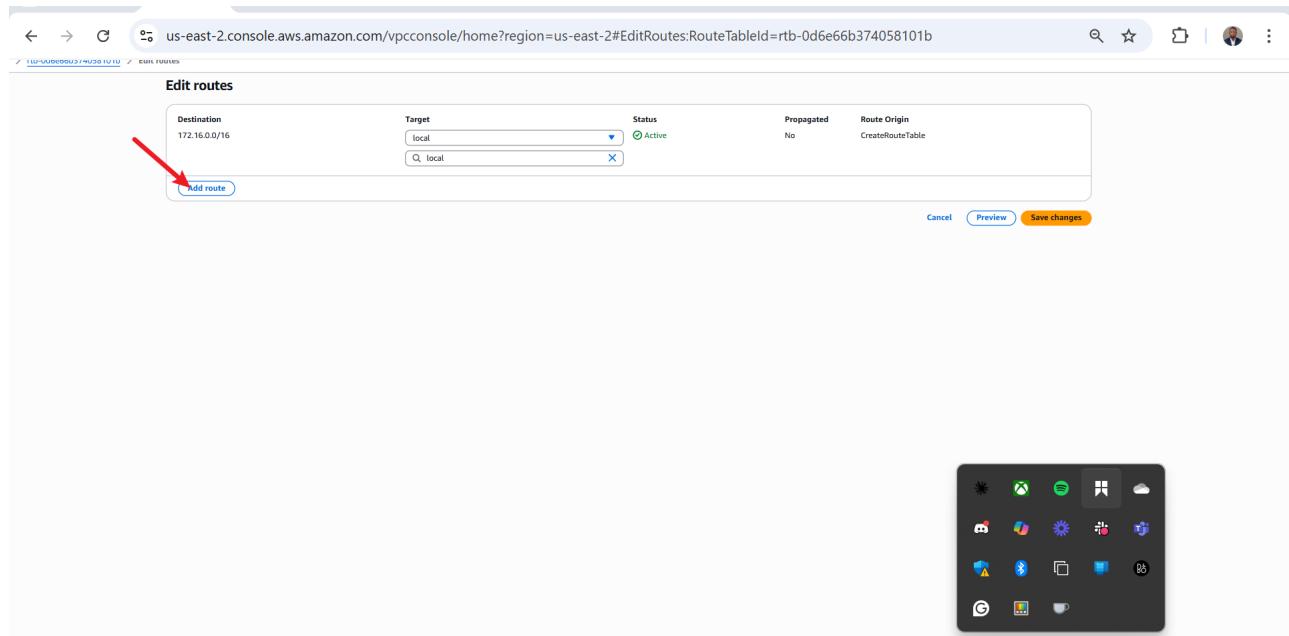
Select a route table

12. Click **Edit routes**.



The screenshot shows the AWS VPC console with the URL <https://us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#RouteTableDetails:RouteTableId=rtb-0d6e66b374058101b>. The page displays route table details for 'rtb-0d6e66b374058101b'. The 'Routes' tab is selected, showing one route entry: Destination 172.16.0.0/16, Target local, Status Active, Propagated No, and Route Origin CreateRouteTable. Below the table is an 'Edit routes' button, which is highlighted with a red arrow.

13. Click **Add route**.



The screenshot shows the 'Edit routes' dialog from the AWS VPC console. It includes fields for Destination (172.16.0.0/16), Target (local), Status (Active), Propagated (No), and Route Origin (CreateRouteTable). At the bottom left of the dialog, there is a blue 'Add route' button, which is highlighted with a red arrow.

14. Copy the IPv4 CIDR of the requester VPC (192.168.0.0/16).

The screenshot shows the AWS VPC console with the 'Your VPCs' page. The requester-VPC (vpc-0777a0c1649f1aa146) is selected, and its IPv4 CIDR (192.168.0.0/16) is highlighted with a red box.

Name	VPC ID	State	Block Public...	IPv4 CIDR	DHCP option set	Main route table	Main network ACL	Tenancy	Default VPC	Owns
accepter-VPC	vpc-0e059193202c0d3ed	Available	Off	172.16.0.0/16	dopt-09149d5972cad...	rtb-0d6e66b374058101b	aci-09da1bf6005a04ea	default	No	S3I
-	vpc-059fb9a09bb014e172	Available	Off	10.0.0.0/16	dopt-09149d5972cad...	rtb-016c1967af12e660	aci-0e8b022b2bd5a4cb	default	No	S3I
requester-VPC	vpc-0777a0c1649f1aa146	Available	Off	192.168.0.0/16	dopt-09149d5972cad...	rtb-0a7c459e61b1aa146	aci-0ec1bb0ff65727d8	default	No	S3I
	vpc-0946cd72df0f4cbe	Available	Off	172.31.0.0/16	dopt-09149d5972cad...	rtb-025c844e81e7229e	aci-0c38bc499ea79c8	default	Yes	S3I

15. Add the requester VPC's CIDR to the **Destination**, set **Target** to the peering connection, and save changes.

The screenshot shows the 'Edit routes' section of the AWS VPC console. A red box highlights the 'Target' dropdown, which is set to 'Peering Connection'. An arrow points from the 'Save changes' button at the bottom right towards the highlighted area.

Destination	Target	Status	Propagated	Route Origin
192.168.0.0/16	local	Active	No	CreateRouteTable
172.31.0.0/16	Peering Connection	-	No	CreateRoute
	Q pcx-0f6e1f02fc7b81e			

16. Route table updated successfully, return to VPC.

The screenshot shows the AWS VPC Route Tables page. The URL is us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#RouteTableDetails:RouteTableId=rtb-0d6e66b374058101b. The page title is "rtb-0d6e66b374058101b". A green banner at the top says "Updated routes for rtb-0d6e66b374058101b successfully". The left sidebar shows the VPC dashboard and various VPC-related options like Route tables, Subnets, and NAT gateways. The main content area displays the "Details" tab for the route table, showing its ID, Main status (Yes), and Owner ID (536697231935). Below this is a table titled "Routes (2)" with columns for Destination, Target, Status, Propagated, and Route Origin. Two routes are listed: one to "local" and another to "pce-0f6e1f02cfce7b81e". A red arrow points from the left sidebar to the "Route tables" link.

17. Copy the IPv4 CIDR of the accepter VPC (172.16.0.0/16) and click the requester VPC's main route table ID.

The screenshot shows the AWS VPC Your VPCs page. The URL is us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#vpcs. The page title is "Your VPCs (1 / 4)". It lists two VPCs: "accepter-VPC" and "requester-VPC". The "requester-VPC" row is selected, and its details are shown in a modal. The modal title is "vpc-0777a0c1649ff1ab4 / requester-VPC". The "Details" tab is selected, showing the VPC ID (vpc-0777a0c1649ff1ab4), State (Available), Block Public Access (Off), IPv4 CIDR (172.31.0.0/16), and Main route table (rtb-0a7ca49e61b1aa146). A red arrow points from the right side of the requester VPC row to the "Main route table" column of the requester VPC row in the table.

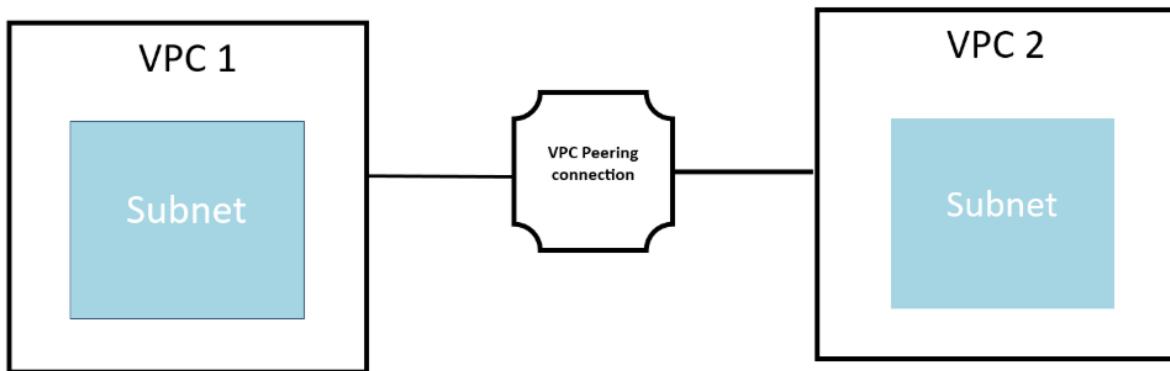
18. Click **Edit route**.

The screenshot shows the AWS VPC Route Tables console. On the left, there's a navigation sidebar with sections like 'VPC dashboard', 'Virtual private cloud', 'Route tables', 'Security', 'PrivateLink and Lattice', and 'DNS firewall'. The main area is titled 'rtb-0a7c459e61b1aa146'. It displays 'Details' for the route table, including 'Route table ID' (rtb-0a7c459e61b1aa146), 'Main' status, and 'Owner ID' (vpc-0777a0c1649ff1ab4 | requester-VPN). A red box highlights the 'Routes' tab in the navigation bar. Below it, a table shows a single route entry: 'Destination' (192.168.0.0/16) and 'Target' (local). The 'Status' is 'Active'. In the bottom right corner of the main area, there's a blue button labeled 'Both' with a dropdown arrow and 'edit routes' next to it. A red arrow points to this button.

19. Add the accepter VPC's CIDR (172.16.0.0/16) to the **Destination**, set **Target** to the peering connection, select the created peering connection, and save changes.

The screenshot shows the 'Edit routes' dialog box. It has fields for 'Destination' (172.16.0.0/16), 'Target' (local), and 'Status' (Active). Below these, there's a section for 'Propagated' and 'Route Origin'. A red box highlights the 'Add route' button at the bottom left. At the bottom right, there are three buttons: 'Cancel', 'Preview', and a large orange 'Save changes' button. A red arrow points to the 'Save changes' button.

20. Peering connection successfully established.



The VPC peering connection has been successfully established. Resources in the accepter VPC (172.16.0.0/16) can now communicate with resources in the requester VPC (192.168.0.0/16), and vice versa, using private IP addresses, ensuring secure and direct inter-VPC communication.

Project Validation

To validate the setup, we performed connectivity tests:

- **Public Subnet Connectivity:** Connected to the [Public-EC2](#) instance via SSH using its public IP and tested outbound internet access by cloning a repository, confirming the Internet Gateway's functionality.
- **Private Subnet Isolation:** Attempted to connect to the [Private-EC2](#) instance using its public IP, which failed as expected, verifying that the private subnet is not directly accessible from the internet.
- **NAT Gateway Verification:** Observed outbound traffic metrics (Packet Out to Destination and Byte Out to Destination) in the NAT Gateway dashboard, confirming that the private subnet can access the internet securely.
- **Intra-Subnet Communication:** Created a second EC2 instance ([Public-EC2-2](#)) in the public subnet, established SSH connectivity between the two public EC2 instances, and verified reachability using ping, confirming default intra-subnet communication within the VPC.
- **VPC Peering:** Configured route tables for both VPCs to include each other's CIDR blocks, enabling direct communication between resources in the requester and accepter VPCs.

Project Reflection

- **Achievements:**
 - Successfully completed all project tasks, including setting up a VPC, configuring subnets, attaching an Internet Gateway, enabling outbound access via a NAT Gateway, and establishing VPC peering.
 - Gained hands-on expertise in navigating the AWS Management Console to deploy and manage VPC resources.
 - Demonstrated the ability to configure secure network architectures, ensuring resources in public subnets have internet access while private subnets remain isolated yet capable of outbound

connectivity.

- Established and validated VPC peering, enabling seamless communication between two VPCs.

- **Challenges and Solutions:**

- Encountered CIDR block size limitations during VPC creation, resolved by adjusting the CIDR block to fall within the recommended range (/16 to /28).
- Ensured non-overlapping CIDR blocks for VPC peering to avoid conflicts, reinforcing the importance of proper IP address planning.
- Configured security group rules and route tables meticulously to enable connectivity while maintaining security best practices.

- **Key Learnings:**

- Developed a deeper understanding of cloud networking concepts, including the roles of subnets, gateways, and route tables in AWS VPCs.
- Learned the significance of network security measures like NAT Gateways and VPC endpoints for protecting sensitive data.
- Gained practical experience in troubleshooting network configurations and validating connectivity through real-world tests.

Conclusion

This project provided a comprehensive exploration of AWS VPC fundamentals, from creating and configuring a VPC to enabling secure internet access and inter-VPC communication. The hands-on exercises reinforced theoretical knowledge with practical application, enhancing proficiency in cloud network architecture. The successful setup of public and private subnets, Internet and NAT Gateways, and VPC peering connections demonstrates the ability to design secure and scalable cloud environments. The project also highlighted the importance of network security and proper configuration to prevent unauthorized access while enabling necessary connectivity.

Overall, this mini-project offered valuable insights into cloud networking and practical skills in deploying VPC infrastructure on AWS, equipping participants with the knowledge and confidence to tackle more complex cloud networking challenges in the future.
