

Security Groups and NACLs Project

Introduction

This project, titled "Security Groups and NACLs," explores the core concepts of Amazon Web Services (AWS) networking security, with a focus on Security Groups and Network Access Control Lists (NACLs). The objective is to understand how Security Groups control inbound and outbound traffic to EC2 instances, and how NACLs serve as subnet-level firewalls to regulate traffic entering and exiting subnets. Through hands-on demonstrations in the AWS Management Console, we deploy and manage these components to demonstrate their functionality.

Before proceeding, a basic understanding of cloud networking is assumed. If terms like "Security Groups" or "NACLs" are unfamiliar, review foundational cloud computing materials.

Project Goals

- Understand the concepts of Security Groups and Network Access Control Lists (NACLs) in AWS.
- Explore how Security Groups and NACLs function as virtual firewalls to control inbound and outbound traffic.
- Gain hands-on experience with configuring Security Groups and NACLs to allow or deny specific types of traffic.

Learning Outcomes

- Gain proficiency in configuring Security Groups and NACLs to control network traffic within AWS environments.
- Understand the differences between Security Groups and NACLs, including their scope, statefulness, and rule configurations.
- Learn how to troubleshoot network connectivity issues by analyzing Security Group and NACL configurations.
- Develop a deeper understanding of AWS networking concepts and best practices for securing cloud environments.

Key Terms and Concepts

Security Group (SG)

- **Inbound Rules:** Rules that control the incoming traffic to an AWS resource, such as an EC2 instance or an RDS database.
- **Outbound Rules:** Rules that control the outgoing traffic from an AWS resource.
- **Stateful:** Security groups automatically allow return traffic initiated by the instances to which they are attached.
- **Port:** A communication endpoint that processes incoming and outgoing network traffic. Security groups use ports to specify the types of traffic allowed.
- **Protocol:** The set of rules that governs the communication between different endpoints in a network. Common protocols include TCP, UDP, and ICMP.

In simple terms, a Security Group acts like a bouncer at a party, deciding who can enter (inbound traffic) and who can leave (outbound traffic). For example, you might allow only port 80 traffic for a web server, keeping it safe from unauthorized access.

Network Access Control List (NACL)

- **Subnet-level Firewall:** NACLs act as a firewall at the subnet level, controlling traffic entering and exiting the subnet.
- **Stateless:** Unlike security groups, NACLs are stateless, meaning they do not automatically allow return traffic. You must explicitly configure rules for both inbound and outbound traffic.
- **Allow/Deny:** NACL rules can either allow or deny traffic based on the specified criteria.
- **Ingress:** Refers to inbound traffic, i.e., traffic entering the subnet.
- **Egress:** Refers to outbound traffic, i.e., traffic exiting the subnet.
- **CIDR Block:** Specifies a range of IP addresses in CIDR notation (e.g., 10.0.0.0/24) that the NACL rule applies to.

NACLs are like neighborhood security guards, controlling access at the subnet level rather than for individual resources. They require separate rules for inbound and outbound traffic due to their stateless nature.

Default Settings

- **Default Security Group:** Every VPC comes with a default security group that allows all outbound traffic and denies all inbound traffic by default.
- **Default NACL:** Every subnet within a VPC is associated with a default NACL that allows all inbound and outbound traffic by default.

Differences Between Security Groups and NACLs

Security Groups operate at the instance level, are stateful, and implicitly deny traffic not explicitly allowed (no explicit "deny" rules). They protect individual resources by filtering based on protocols, ports, and IP addresses.

NACLs operate at the subnet level, are stateless, and can explicitly allow or deny traffic. They filter based on IP addresses and protocol numbers, applying independently to inbound and outbound traffic.

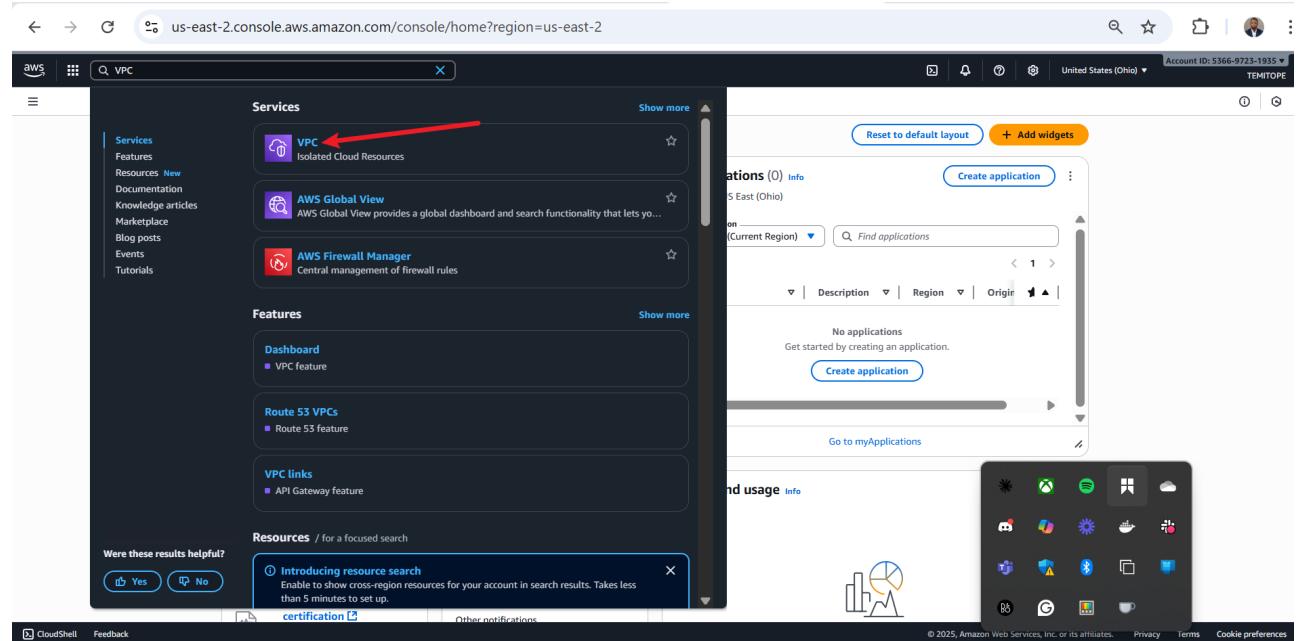
Note: In security groups, there's no explicit "deny" option as seen in NACLs; any rule configured within a security group implies permission.

Practical Implementation

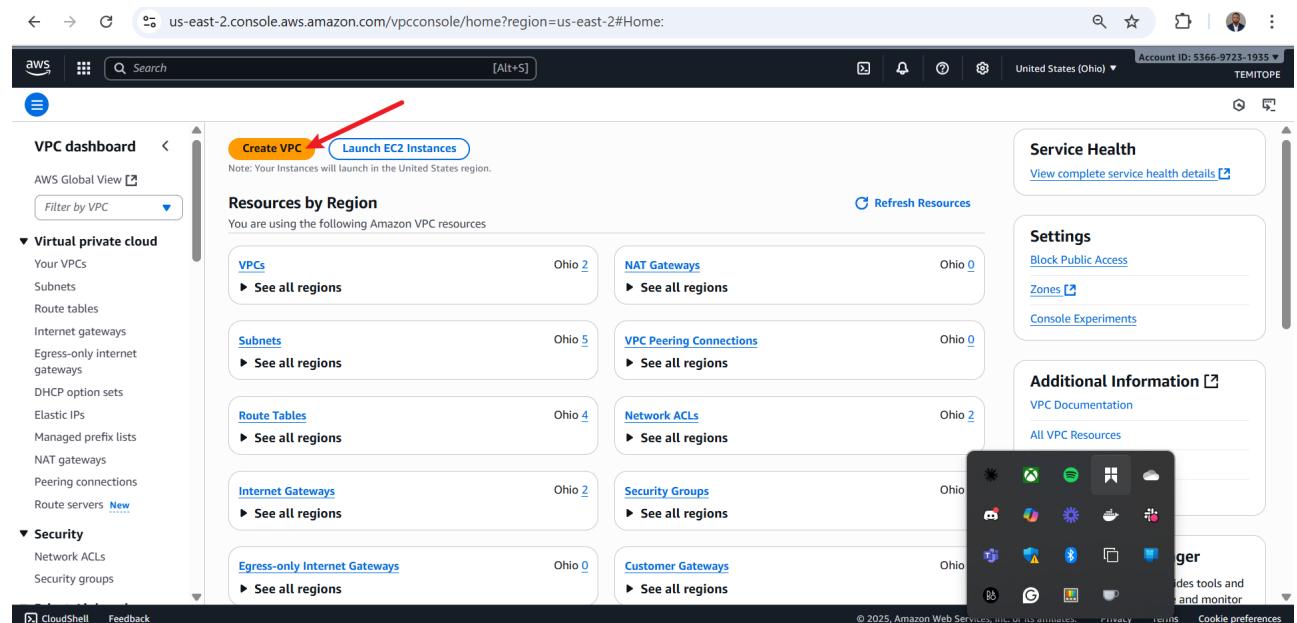
The practical part is divided into setup and two main sections: Security Groups and NACLs. We begin by setting up the foundational VPC environment, including subnets, internet gateways, route tables, NAT gateways, and an EC2 instance. Then, we configure and test Security Groups and NACLs in various scenarios.

Setting Up the VPC Environment

1. Navigate to the search bar and enter "VPC", then click on the VPC service that appears.



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



3. On the Create VPC page, select "VPC only", set the IPv4 CIDR block, and click "Create VPC".

The screenshot shows the 'Create VPC' page in the AWS Management Console. The 'VPC settings' section is highlighted with a red arrow pointing to the 'Resources to create' dropdown, which has 'VPC only' selected. Another red arrow points to the 'IPv4 CIDR' input field, which contains '10.0.0.0/16'. A third red arrow points to the 'Create VPC' button at the bottom right of the page.

VPC settings

Resources to create: VPC only VPC and more

Name tag - optional: my-vpc-01

IPv4 CIDR: 10.0.0.0/16

IPv6 CIDR block: No IPv6 CIDR block

Tenancy: Default

Tags: No tags associated with the resource. Add tag. You can add 50 more tags.

Create VPC

4. VPC successfully created.

The screenshot shows the 'VPC dashboard' in the AWS Management Console. A red box highlights the success message 'You successfully created vpc-059fcaa09dbb14e12'. Below it, the 'vpc-059fcaa09dbb14e12' card displays the following details:

- Details**:
 - VPC ID**: vpc-059fcaa09dbb14e12
 - DNS resolution**: Enabled
 - Main network ACL**: acl-0dd802252bcd3adcb
 - IPv4 CIDR**: 10.0.0.0/16
 - State**: Available
 - Tenancy**: default
 - Default VPC**: No
 - Network Address Usage metrics**: Disabled
 - Block Public Access**: Off
 - DHCP option set**: dopt-09140d5972cad56b
 - IPv4 CIDR**: 10.0.0.0/16
 - Route 53 Resolver DNS Firewall rule groups**: None
 - DNS hostnames**: Disabled
 - Main route table**: rtb-01b8c1967af12eb0
 - IPv6 pool**: None
 - Owner ID**: 536697231935
- Resource map** tab is selected, showing the VPC structure.

5. Note: In case of CIDR block size error, let your block size fall within the specified range.

IPv4 CIDR

10.0.0.0/8

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.

6. Here is the created VPC; click "Subnets" to create a subnet.

aws → us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#vpcs:

Your VPCs (1/2) Info

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table
-	vpc-059fcaa09dbb14e12	Available	Off	10.0.0.0/16	-	dopt-09149d5972cad...	rtb-01b8c1967af...
-	vpc-0a946dc2d5bf4cbe	Available	Off	172.31.0.0/16	-	dopt-09149d5972cad...	rtb-0258a4ee81e...

vpc-059fcaa09dbb14e12

Details Resource map CIDs Flow logs Tags Integrations

Details

VPC ID vpc-059fcaa09dbb14e12	State Available	Block Public Access Off
DNS resolution Enabled	Tenancy default	DHCP option set dopt-09149d5972cad56b
Main network ACL	Default VPC	IPv4 CIDR

7. On the subnet dashboard, click "Create subnet".

aws → us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#subnets:

Subnets (3) Info

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR
-	subnet-0774bf3662e7df059	Available	vpc-0a946dc2d5bf4cbe	Off	172.31.0.0/20	-	-
-	subnet-03ab326585affc0b	Available	vpc-0a946dc2d5bf4cbe	Off	172.31.16.0/20	-	-
-	subnet-08834c04364b5df9e	Available	vpc-0a946dc2d5bf4cbe	Off	172.31.32.0/20	-	-

Select a subnet

8. Select the Subnet VPC (created in the first step).

The screenshot shows the 'Create subnet' page in the AWS VPC console. In the 'VPC' section, there is a dropdown menu labeled 'Select a VPC'. Below it, two options are listed: 'vpc-059fcaa09dbb14e12' (selected) and 'vpc-0946gddc2d5bf4cbe'. A red box highlights the selected VPC. At the bottom right of the page are 'Cancel' and 'Create subnet' buttons.

9. Enter subnet name (my-public-subnet-1), set availability zone, subnet IPv4 CIDR block, and click "Add new subnet".

The screenshot shows the 'Subnet settings' page for creating a public subnet. The 'Subnet 1 of 1' section is filled out as follows:

- Subnet name:** my-public-subnet-1 (highlighted with a red box)
- Availability Zone:** United States (Ohio) / use2-az1 (us-east-2a) (highlighted with a red box)
- IPv4 subnet CIDR block:** 10.0.6.0/24 (highlighted with a red box)
- Tags - optional:** A tag named 'Name' with value 'my-public-subnet-1' is added.

A red arrow points from the 'Add new subnet' button at the bottom left of the form area towards the 'Add new subnet' button at the bottom right of the page.

10. Enter subnet name (my-private-subnet-1), set availability zone, subnet IPv4 CIDR block, 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

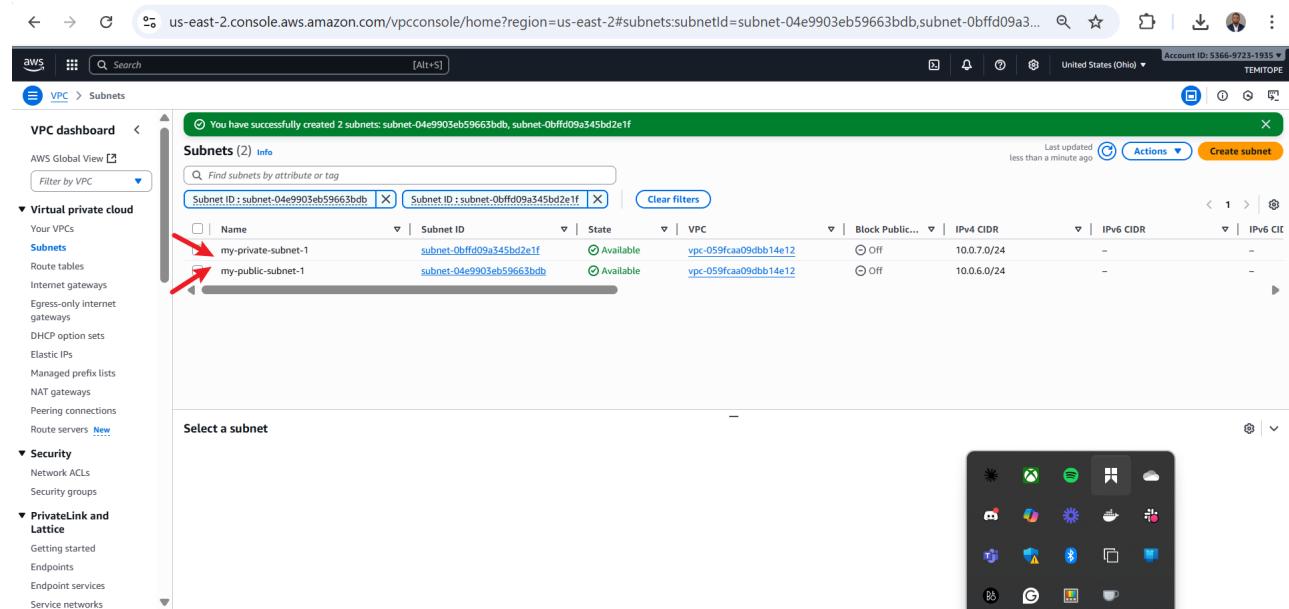
Key Value - optional

You can add 49 more tags.

11. The architecture.



12. my-public-subnet-1 and my-private-subnet-1 successfully created.

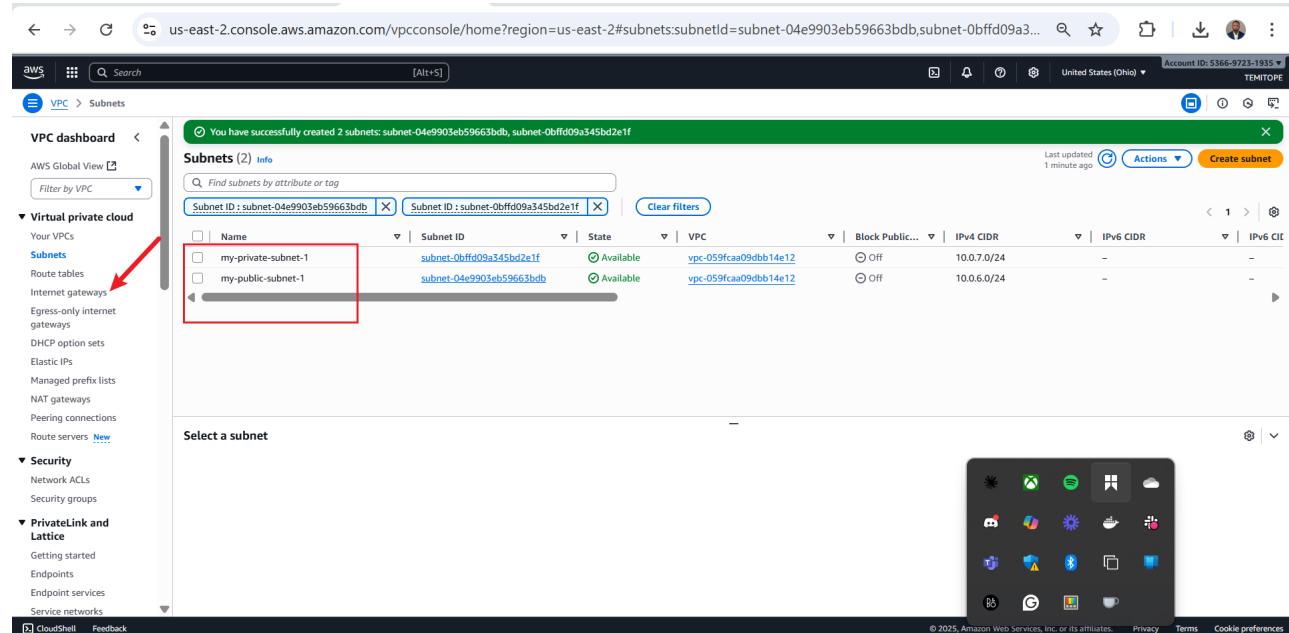


The screenshot shows the AWS VPC Subnets page. The left sidebar is titled "Virtual private cloud" and includes sections for Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, and Route servers. The "Subnets" section is currently selected. The main content area displays a table of subnets:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv6 CIC
my-private-subnet-1	subnet-0bffd09a345bd2e1f	Available	vpc-059fcaa09dbb14e12	Off	10.0.7.0/24	-	-
my-public-subnet-1	subnet-04e9903eb59663bdb	Available	vpc-059fcaa09dbb14e12	Off	10.0.6.0/24	-	-

A red arrow points to the "my-public-subnet-1" row.

13. Since my-public-subnet-1 and my-private-subnet-1 successfully created, now click on "Internet gateways" to connect public subnet to the internet.



The screenshot shows the same AWS VPC Subnets page as the previous one. A red arrow points to the "Internet gateways" link in the "Virtual private cloud" sidebar. The main content area shows the same subnet table as before.

14. On the internet gateway dashboard, click "Create internet gateway" button.

The screenshot shows the AWS VPC Internet Gateways dashboard. On the left, there's a navigation sidebar with sections like VPC dashboard, Virtual private cloud, Security, and PrivateLink and Lattice. The main area displays a table of existing Internet Gateways, with one entry: Name igw-037e56e68b137c83b, Internet gateway ID igw-037e56e68b137c83b, State Attached, VPC ID vpc-0a946cd2d5bf4cbe, and Owner 536697231935. At the top right, there are 'Actions' and 'Create internet gateway' buttons, with a red arrow pointing to the 'Create internet gateway' button.

15. Name internet gateway and click "Create internet gateway".

The screenshot shows the 'Create internet gateway' wizard. The first step, 'Internet gateway settings', has a 'Name tag' input field containing 'my-internet-gw-1', which is highlighted with a red box. The second step, 'Tags - optional', shows a single tag named 'Name' with value 'my-internet-gw-1'. At the bottom right of the wizard, there is a large orange 'Create internet gateway' button, which is also highlighted with a red arrow.

16. Internet gateway successfully created; note that the state shows detached.

The screenshot shows the AWS VPC console interface. On the left, there's a navigation sidebar with sections like 'Virtual private cloud', 'Security', and 'PrivateLink and Lattice'. The main area displays a green banner stating: '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: 'igw-05cba0c2e0b35021e / my-internet-gw-1'. The 'State' field is highlighted with a red box and contains the value 'Detached'. To the right, there are tabs for 'Details' and 'Info', and a 'Tags' section. At the top right, there are buttons for 'Attach to a VPC' and 'Actions'. A small icon bar with various service icons is visible at the bottom right.

17. Click "Actions" and then "Attach to VPC".

This screenshot is similar to the previous one, showing the same internet gateway details. However, a red arrow points to the 'Actions' button in the top right corner of the main content area. The 'Actions' menu is open, showing options like 'Attach to VPC', 'Detach from VPC', 'Manage tags', and 'Delete'.

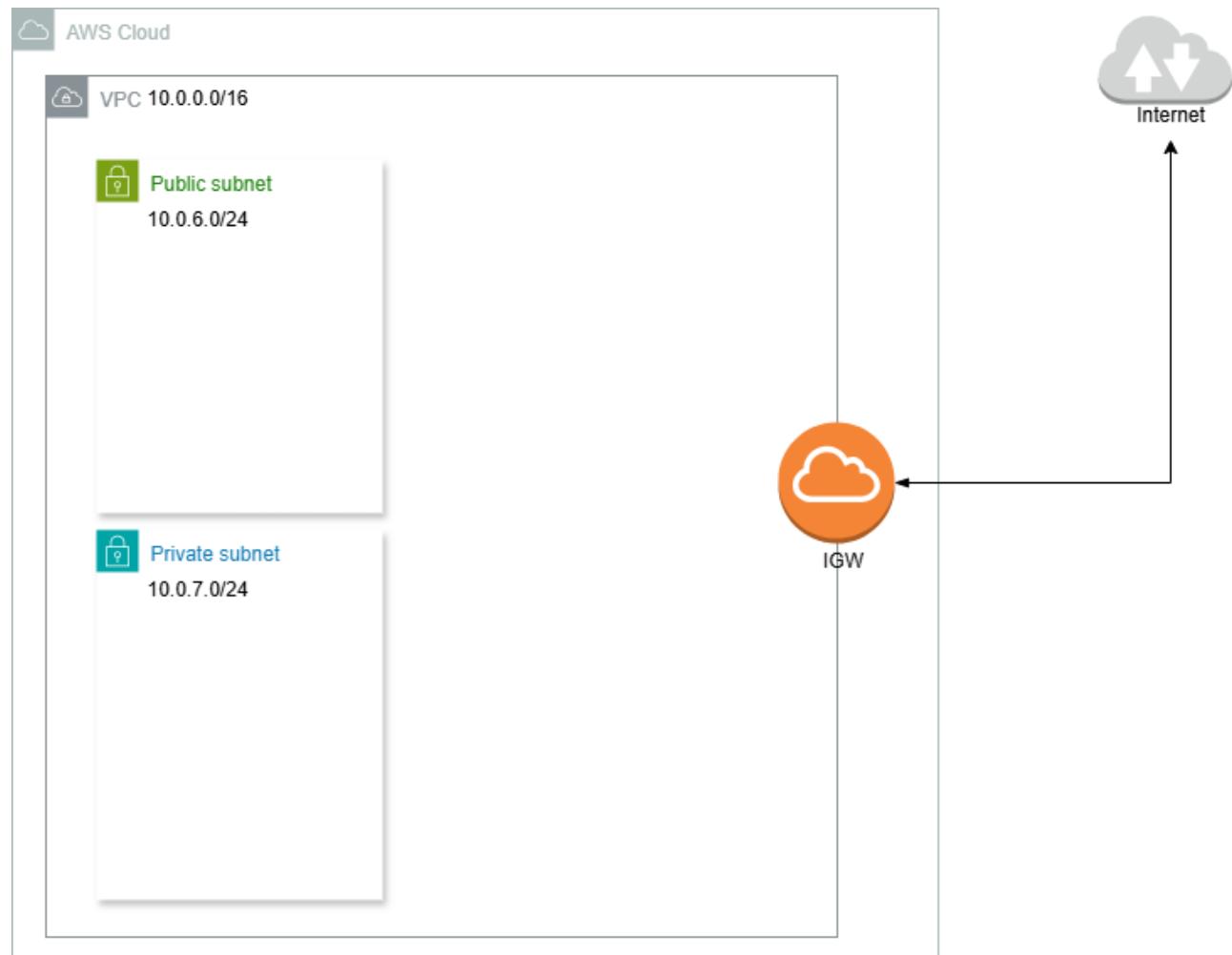
18. Select VPC and click "Attach internet gateway".

This screenshot shows the 'Attach to VPC' dialog box. It has a 'VPC' section where a VPC ID 'vpc-059fcaa09dbb14e12' is selected from a dropdown. Below it is an 'AWS Command Line Interface command' section. A red arrow points to the 'Attach internet gateway' button at the bottom right of the dialog. The background shows the AWS VPC console interface with a small icon bar at the bottom right.

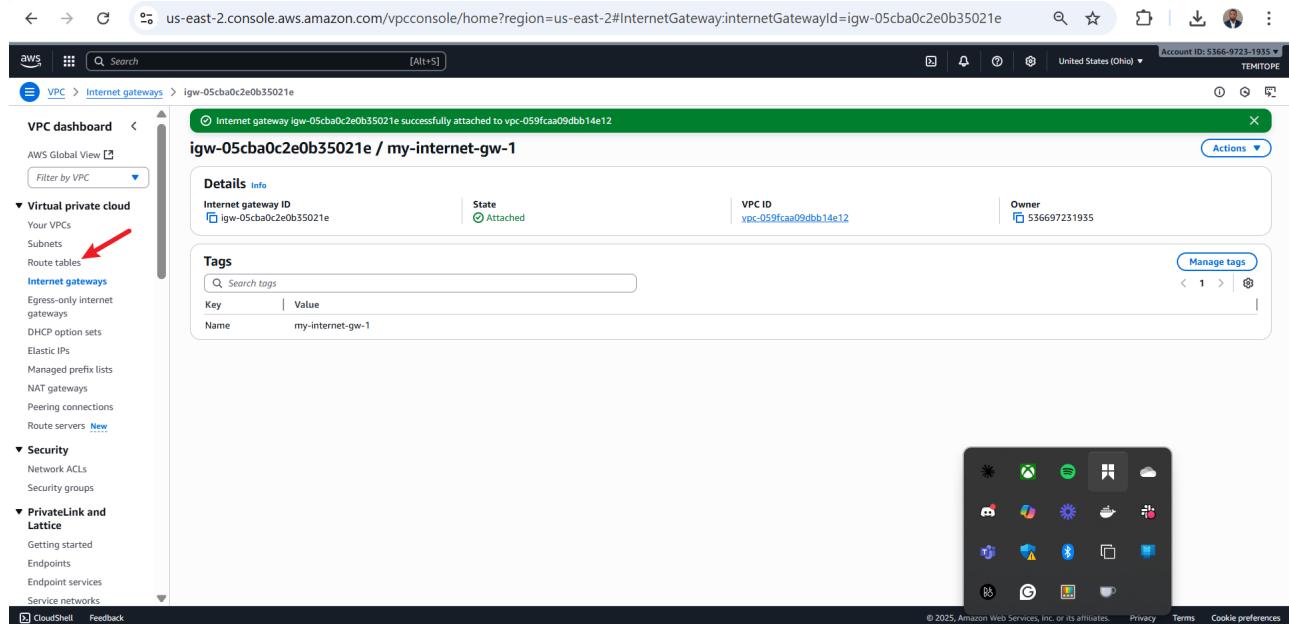
19. Internet gateway successfully attached to VPC.

The screenshot shows the AWS VPC console interface. On the left, there's a navigation sidebar with options like 'Virtual private cloud', 'Internet gateways' (which is selected), and 'Security'. The main content area displays a green success message: 'Internet gateway igw-05cba0c2e0b35021e successfully attached to vpc-059fcaa09dbb14e12'. Below this, it shows the 'igw-05cba0c2e0b35021e / my-internet-gw-1' entry. A red box highlights the 'State' column, which shows 'Attached'. Other details include the 'Internet gateway ID' (igw-05cba0c2e0b35021e), 'VPC ID' (vpc-059fcaa09dbb14e12), and 'Owner' (Account ID: 536697231935). There are also 'Details' and 'Info' tabs, and a 'Tags' section with one tag named 'my-internet-gw-1'. On the right side of the screen, there's a small screenshot of a Windows desktop with various icons.

20. Present architecture.

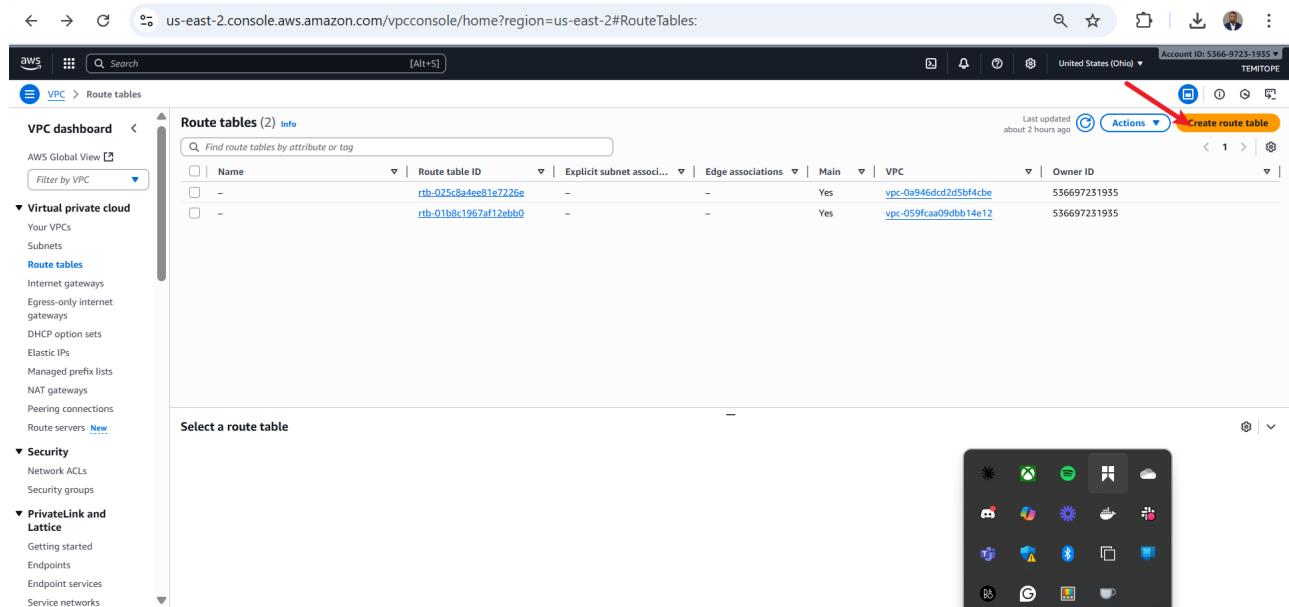


21. Since IGW is now attached to VPC, click on "Route tables".



The screenshot shows the AWS VPC console. In the left sidebar under 'Virtual private cloud', 'Route tables' is highlighted with a red arrow. The main content area displays the details for an Internet gateway named 'igw-05cba0c2e0b35021e'. It shows the gateway is successfully attached to a VPC ('vpc-059fcaa09dbb14e12'). There are no tags listed.

22. On the route table dashboard, click "Create route table".



The screenshot shows the AWS VPC Route tables dashboard. The left sidebar shows 'Route tables' is selected. The main area displays two existing route tables: 'rtb-025c8adeeb1e7226e' and 'rtb-01b8c1967af12eb0'. In the top right corner of the actions menu, there is a 'Create route table' button, which is highlighted with a red arrow.

23. Name route table, select 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 Add new tag Remove

You can add 49 more tags.

Cancel Create route table

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

24. Route table created; now click on "Subnet associations" then "Edit subnet association".

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 (0)

Find subnet association IPv4 CIDR IPv6 CIDR

No subnet associations
You do not have any 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

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

25. Select public subnet and click "Save associations".

The screenshot shows the 'Edit subnet associations' page in the AWS VPC console. 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.

26. Subnet association successfully saved; now under "Routes" tab, click "Edit routes".

The screenshot shows the 'rtb-0d41bba2ed11c8401 / my-route-table-1' details page in the AWS VPC console. Under the 'Routes' tab, there is 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.

27. Click "Add route" to add new route.

The screenshot shows the 'Edit routes' page in the AWS VPC console. Under the 'Routes' tab, there is one route entry: '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 routes table.

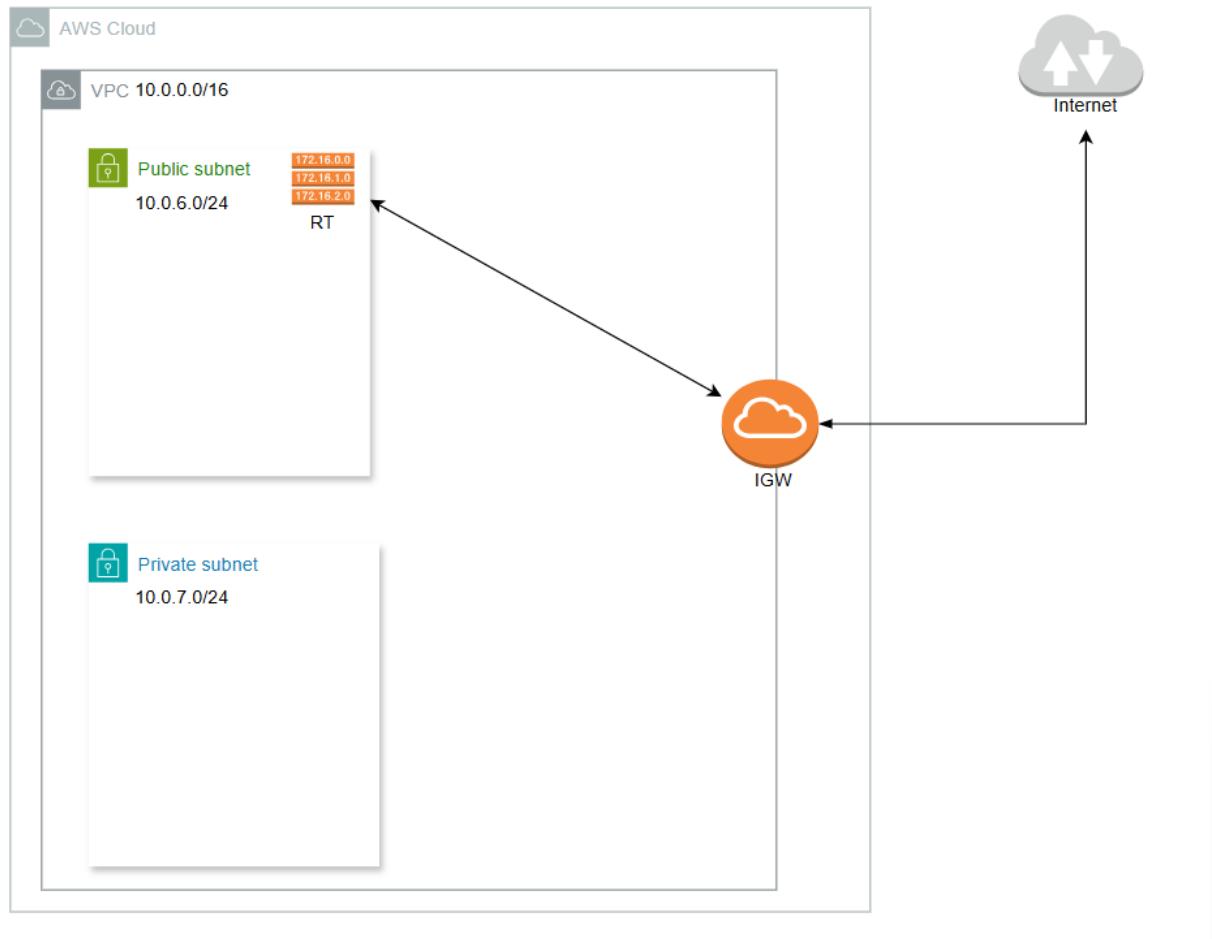
28. Set route destination to 0.0.0.0/0, target to the created IGW, and save changes.

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 specific gateway 'igw-05cba0c2e0b35021e' is selected. The 'Status' is 'Active'. The 'Propagated' status is 'No'. The 'Route Origin' is 'CreateRoute'. The 'Save changes' button is highlighted with a red arrow.

29. Route table updated successfully.

The screenshot shows the details for route table rtb-0d41bba2ed11c8401 / my-route-table-1. The 'Details' tab is selected, showing the route table ID as rtb-0d41bba2ed11c8401, Main status as 'No', and Owner ID as 536697231935. The 'Routes' tab is selected, showing two routes: one to '0.0.0.0/0' targeting 'igw-05cba0c2e0b35021e' and another to '10.0.0.16' targeting 'local'. The 'Actions' button is visible at the top right.

30. Present VPC or network architecture.



31. Now that route tables have been updated, click on "NAT gateways".

The screenshot shows the AWS VPC Route Tables page. The route table 'rtb-0d41bba2ed11c8401 / my-route-table-1' is selected. The 'Details' tab is active, showing the route table ID, VPC, and explicit subnet associations. The 'Routes' tab displays two routes: one to 'igw-05cba0c2e0b35021e' and another to 'local'. A red arrow points to the 'NAT gateways' link in the left sidebar.

32. Click "Create NAT gateway".

The screenshot shows the AWS VPC dashboard with the 'NAT gateways' section selected. The top navigation bar includes the AWS logo, search bar, and account information. On the left, a sidebar lists various VPC-related services like 'Your VPCs', 'Subnets', 'Route tables', and 'NAT gateways'. The main content area displays a table header for 'NAT gateways info' with columns for Name, NAT gateway ID, Connectivity..., State, and others. A message at the bottom states 'No NAT gateways found'. In the top right, there's a 'Actions' dropdown and a prominent orange 'Create NAT gateway' button.

33. Name NATgw, select private subnet, and set connectivity type to private.

This screenshot shows the 'Create NAT gateway' wizard. The first step, 'NAT gateway settings', is displayed. It requires a 'Name - optional' (set to 'my-NAT-gw-1'), a 'Subnet' (selected as 'subnet-0bffd095345bd2e1f (my-private-subnet-1)'), and a 'Connectivity type' (set to 'Private'). A note below the connectivity type says 'Private NAT gateway traffic can't reach the internet.' The second step, 'Additional settings', is partially visible. At the bottom right of the first step, there is a 'Create NAT gateway' button, which is highlighted with a red arrow.

34. NAT gateway created successfully.

The screenshot shows the AWS VPC console with the URL <https://us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#NatGatewayDetails:natGatewayId=nat-02527773795a97d0f>. The main content area displays a success message: "NAT gateway nat-02527773795a97d0f | my-NAT-gw-1 was created successfully." Below this, the NAT gateway details are shown:

NAT gateway ID	nat-02527773795a97d0f	Connectivity type	Private	State	Pending
NAT gateway ARN	arn:aws:ec2:us-east-2:536697231935:natgateway/nat-02527773795a97d0f	Primary public IPv4 address	-	Primary private IPv4 address	10.0.7.35
VPC	vpc-059fcaa09dbb14e12	Subnet	subnet-0bffd09a345bd2e1f / my-private-subnet-1	Created	Wednesday, October 8, 2025 at 17:34:18 GMT+1

The "Secondary IPv4 addresses" tab is selected, showing a message: "Secondary IPv4 addresses are not available for this nat gateway." A small Windows taskbar icon is visible in the bottom right corner.

35. Select the created NAT gateway and locate subnet ID link at details section and click on it.

The screenshot shows the AWS VPC console with the URL <https://us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#NatGateways>. The main content area shows a table of NAT gateways:

Name	NAT gateway ID	Connectivity...	State	Primary public I...	Primary private I...	Primary network...	VPC
my-NAT-gw-1	nat-02527773795a97d0f	Private	Available	-	10.0.7.35	eni-0ba2fe8358ad3...	vpc-059fcaa09dbb14e1

A red box highlights the first row in the table. Below the table, the details for the selected NAT gateway ("my-NAT-gw-1") are shown:

NAT gateway ID	nat-02527773795a97d0f	Connectivity type	Private	State	Available
NAT gateway ARN	arn:aws:ec2:us-east-2:536697231935:natgateway/nat-02527773795a97d0f	Primary public IPv4 address	-	Primary private IPv4 address	10.0.7.35
VPC	vpc-059fcaa09dbb14e12	Subnet	subnet-0bffd09a345bd2e1f / my-private-subnet-1	Created	Wednesday, October 8, 2025 at 17:34:18 GMT+1

A red arrow points to the "my-private-subnet-1" link in the Subnet field. A small Windows taskbar icon is visible in the bottom right corner.

36. In the subnet page, navigate to "Route Table" section and click one route table ID (rtb-...).

Screenshots of the AWS VPC Subnets page:

- Subnets (1/1) Info**: Shows a table with one row for 'my-private-subnet-1'. Columns include Name, Subnet ID, State, VPC, Block Public..., IPv4 CIDR, IPv6 CIDR, and IPv6 CIDR.
- subnet-0bffd09a345bd2e1f / my-private-subnet-1**: Shows the subnet details. A red arrow points to the 'Route table' tab.
- Route table: rtb-01b8c1967af12ebb0**: Shows the route table configuration with one route to '10.0.0.0/16' target 'local'.

37. On route table page, click on "Routes" tab and then "Edit routes".

Screenshots of the AWS Route Tables page:

- Route tables (1/1) Info**: Shows a table with one row for 'rtb-01b8c1967af12ebb0'. Columns include Name, Route table ID, Explicit subnet assoc..., Edge associations, Main, VPC, and Owner ID.
- rtb-01b8c1967af12ebb0**: Shows the route table details. A red arrow points to the 'Routes' tab.
- Routes (1)**: Shows the route table configuration with one route to '10.0.0.0/16' target 'local'. A red arrow points to the 'Edit routes' button.

38. On the edit route page, click "Add route".

The screenshot shows the 'Edit routes' page for a specific route table. A single route entry is listed:

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

At the bottom left, there is a blue 'Add route' button with a red arrow pointing to it.

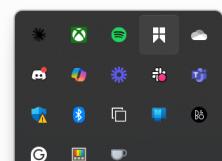


39. Select 0.0.0.0/0 as destination, NAT Gateway as the target, and the created NAT gateway as the NAT Gateway, then save changes.

The screenshot shows the 'Edit routes' page for a specific route table. A route entry is being configured:

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

The 'Destination' field (0.0.0.0/0) and 'Target' field (NAT Gateway) are highlighted with a red box. At the bottom right, there is a blue 'Save changes' button with a red arrow pointing to it.



40. Route table updated successfully; now click "Subnet association" and "Edit subnet association".

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. You do not have any subnet associations.

Subnets without explicit associations (1)

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

Edit subnet associations

41. Click on the private subnet and then "Save associations" button.

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
<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-04e9903eb59663b0	10.0.6.0/24	-	rtb-0d41bb02e

Selected subnets

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

Save associations

42. The subnet has been successfully attached with 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

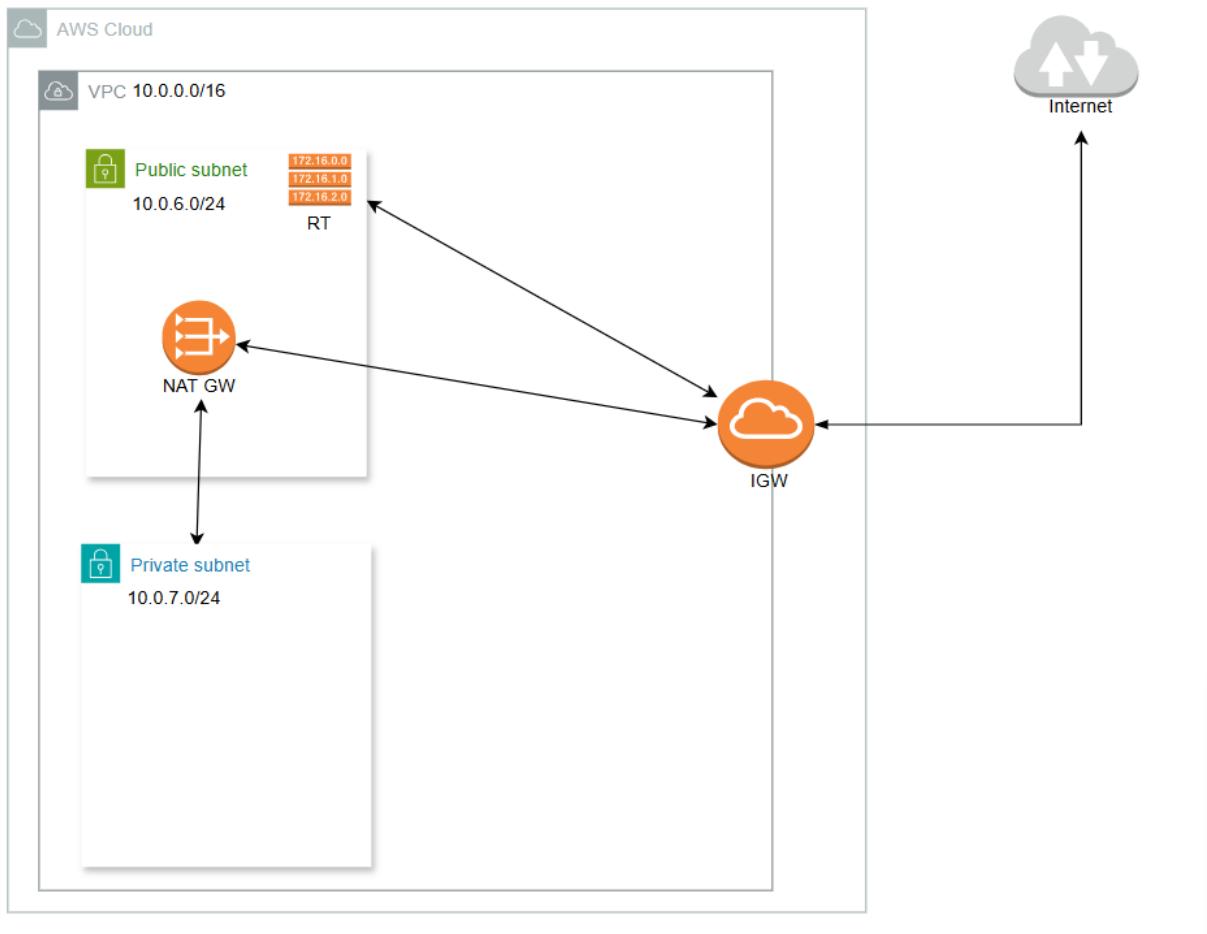
Edge associations

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

43. Present VPC architecture or status.



Part 1: Security Groups

We now create an EC2 instance and configure Security Groups, exploring scenarios such as allowing specific inbound/outbound traffic, removing rules, and testing connectivity.

44. Create an Ubuntu EC2 instance and name it "my-test-instance" and scroll down.

The screenshot shows the AWS EC2 Instances launch wizard. The current step is "Launch an instance". The "Name and tags" section shows a single tag named "my-test-instance". The "Application and OS Images (Amazon Machine Image)" section lists various AMIs, with "Ubuntu Server 24.04 LTS (HVM), SSD Volume Type" selected. The "Summary" section on the right shows 1 instance being launched, using the Canonical, Ubuntu, 24.04, amd64 AMI. A tooltip for the free tier is visible, stating: "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 snapshots, and 10 GiB of data transfer charge-free per month." The bottom of the screen shows the AWS navigation bar and footer.

45. After setting the key pair, on the network settings click on "Edit".

The screenshot shows the AWS EC2 console with the URL us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstances. The 'Network settings' section is open, showing a dropdown for 'Key pair name - required' set to 'temskey1'. Below it, the 'Network' and 'Subnet' sections are visible. To the right, the 'Summary' section shows 'Number of instances' set to 1. A red arrow points to the 'Edit' button located next to the 'Create new key pair' link.

46. Set its VPC to the created VPC (10.0.0.0/16) and subnet to the public subnet. Enable Auto-assign public IP, then launch instance.

The screenshot shows the AWS EC2 console with the same URL as the previous screenshot. The 'Network settings' section is open, with the 'VPC' dropdown set to 'vpc-059fcaa09dbb14e12' and the 'Subnet' dropdown set to 'subnet-04e9903eb59663bdb my-public-subnet-1'. Both dropdowns are highlighted with a red box. A red arrow points to the 'Enable' link under the 'Auto-assign public IP' section. Another red arrow points to the 'Launch instance' button at the bottom right of the summary section.

47. We have created a public EC2 in the public subnet which hosts our website.

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key name
my-test-instance	i-0d25eb835f9f56b9b	Running	t2.micro	Initializing	View alarms +	us-east-2a	-	18.119.133.18	-	-	disabled	launch-wizard-5	temsky1

i-0d25eb835f9f56b9b (my-test-instance)

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary

Public IPv4 address: 18.119.133.18 [open address]

Private IP address: 10.0.6.209

Public DNS: -

Private IP addresses: 10.0.6.209

Elastic IP addresses: -

AWS Compute Optimizer Finding: Opt-in to AWS Compute Optimizer for recommendations. [Learn more]

Auto Scaling Group name: -

Managed: False

48. To see the instance security group rule, click on "Security" tab and the security groups link.

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key name
my-test-instance	i-0d25eb835f9f56b9b	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a	-	18.119.133.18	-	-	disabled	launch-wizard-5	temsky1

i-0d25eb835f9f56b9b (my-test-instance)

Security Details Status and alarms Monitoring Networking Storage Tags

Security details

Security groups: sg-099675804954285d4 (launch-wizard-5)

Inbound rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups	Description
sg-08fe6807fbaf047	22	TCP	0.0.0.0/0	launch-wizard-5	-	-

Outbound rules

Name	Security group rule ID	Port range	Protocol	Destination	Description
-	-	-	-	-	-

49. Note that the security group inbound rule allows only IPv4 SSH connection using TCP port 22 from anywhere.

sg-099675804954285d4 - launch-wizard-5

Details

Security group name: launch-wizard-5

Owner: 536697231935

Security group ID: sg-099675804954285d4

Description: Launch-wizard-5 created 2025-10-16T03:17:22.687Z

VPC ID: vpc-0ef7c16e6038b0ffef

Inbound rules count: 1 Permission entry

Outbound rules count: 1 Permission entry

Inbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sg-08fe6807fbaf047	IPv4	SSH	TCP	22	0.0.0.0/0	-

50. Security group outbound rule however allows only IPv4 All traffic connection using all protocol on all ports from anywhere.

The screenshot shows the AWS CloudWatch Metrics console. A single metric named "AWS/CloudWatch Metrics" is displayed with a value of 1. The metric is described as "CloudWatch Metrics Metrics". The "Metrics" tab is selected, and the "CloudWatch Metrics Metrics" series is highlighted.

51. Back to instance, retrieve the public IP address of the instance from details tab.

The screenshot shows the AWS EC2 Instances page. A specific instance named "my-test-instance" is selected and highlighted with a red box. The instance details are shown, including its Public IPv4 address, which is 18.119.133.18, also highlighted with a red box.

52. Using the IP, the site is unreachable due to HTTP rule not defined in the security group.

The screenshot shows a browser window with the URL "18.119.133.18". An error message "This site can't be reached" is displayed, stating that "18.119.133.18 took too long to respond". Below the message, there is a list of troubleshooting steps:

- Checking the connection
- Checking the proxy and the firewall
- Running Windows Network Diagnostics

The error code "ERR_CONNECTION_TIMED_OUT" is visible at the bottom of the browser window. To the right, a portion of a Windows taskbar is visible with several icons.

53. Navigate to "Security groups" on the left side bar and click on it.

The screenshot shows the AWS EC2 Instances page. On the left sidebar, under the 'Network & Security' section, there is a 'Security Groups' link. A red arrow points to this link. The main content area displays an instance named 'my-test-instance' with its details: Instance ID (i-0d25eb835f9f56b9b), Public IPv4 address (18.119.133.18), Instance state (Running), and Instance type (t2.micro). The 'Details' tab is selected. A tooltip for the 'Public IPv4 address' field shows both the public and private IP addresses.

54. Click on "Create security group".

The screenshot shows the AWS EC2 Security Groups page. The left sidebar has the same 'Network & Security / Security Groups' link as the previous screenshot. The main content area lists several security groups, each with a name, VPC ID, description, and owner. At the top right of the table, there is a 'Create security group' button, which is highlighted with a red arrow.

55. Set security group name, select VPC, and under inbound rules click on "Add rule".

The screenshot shows the 'Create security group' wizard. In the 'Basic details' step, the security group name is set to 'my-first-security-group' and the description is 'creating security group that allows ssh and http'. The 'VPC info' dropdown is open, showing the selected VPC. In the 'Inbound rules' step, there is a red box around the 'Add rule' button, which is highlighted with a red arrow.

56. Add HTTP and SSH inbound rules respectively and set their source to be from anywhere.

The screenshot shows the 'Create security group' page in the AWS EC2 console. Under the 'Inbound rules' section, there are two entries: one for 'HTTP' (Protocol: TCP, Port range: 80, Source: Anywhere..., 0.0.0.0/0) and one for 'SSH' (Protocol: TCP, Port range: 22, Source: Anywhere..., 0.0.0.0/0). Both entries have a red box around them. Below the table is a note: '⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' An 'Add rule' button is visible at the bottom left of the table.

57. Keep the outbound rules the way it is.

The screenshot shows the 'Create security group' page in the AWS EC2 console. Under the 'Outbound rules' section, there is one entry: 'All traffic' (Protocol: All, Port range: All, Destination: Anywhere..., 0.0.0.0/0). Below the table is a note: '⚠ Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses.' An 'Add rule' button is visible at the bottom left of the table.

58. Now click "Create security group".

The screenshot shows the 'Create security group' page in the AWS EC2 console. It includes sections for 'Outbound rules' (with a note about destination 0.0.0.0/0), 'Tags - optional' (with a note about tags), and a summary of the security group configuration. A red arrow points to the 'Create security group' button at the bottom right of the page.

59. Security group created successfully.

The screenshot shows the 'Security Groups' page in the AWS EC2 console, displaying the details of the newly created security group 'sg-0012c41f2dcc7ac95'. A green banner at the top says 'Security group (sg-0012c41f2dcc7ac95 | my-first-security-group) was created successfully'. The main table shows the following details:

Details	Value
Security group name	my-first-security-group
Owner	AWS Account (536697231935)
Inbound rules count	2 Permission entries
Outbound rules count	1 Permission entry
VPC ID	vpc-0ef7c16d603bbdfef

The 'Inbound rules' section shows two entries:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
sgr-0a11f4d7e80cc4e16	sgr-0a11f4d7e80cc4e16	IPv4	SSH	TCP	22	0.0.0.0/0	-
sgr-09b26fe06ef059a0e	sgr-09b26fe06ef059a0e	IPv4	HTTP	TCP	80	0.0.0.0/0	-

60. Click on "Instances" to navigate to instances.

The screenshot shows the AWS EC2 Security Groups interface. A success message at the top states: "Security group (sg-0012c41f2dcc7ac95 | my-first-security-group) was created successfully". The left sidebar has a red arrow pointing to the "Instances" section. The main content area shows the details of the "sg-0012c41f2dcc7ac95 - my-first-security-group". The "Inbound rules" tab is active, listing two entries:

- sg-0012c41f2dcc7ac95 | my-first-security-group | sg-0012c41f2dcc7ac95 | 536697231935 | Inbound rules count: 2 | Outbound rules count: 1
- | sgr-0a11f4d7eb0cc4e16 | IPv4 | SSH | TCP | Port range: 22 | Source: - | Description: -
- | sgr-09b26fe06ef059a0e | IPv4 | HTTP | TCP | Port range: 80 | Source: - | Description: -

61. To attach created security group to my-test-instance, select it and click on "Actions" button then "Security" option.

The screenshot shows the AWS EC2 Instances interface. A red arrow points to the "Instances" section in the left sidebar. The main content area shows a table of instances with one entry: "my-test-instance" (i-0d25eb835f9f56b9b). A context menu is open over this row, with a red arrow pointing to the "Security" option under the "Actions" dropdown. The "Actions" dropdown menu includes options like "Connect", "Instance state", "Launch instances", "Instance diagnostics", "Instance settings", "Networking", and "Security".

62. Under security option, click on "Change security groups".

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), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager (New), Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, and Elastic IPs. The main area displays a table for 'Instances (1/1)'. The first row shows an instance named 'my-test-instance' with ID 'i-0d25eb835f9f56b9b', status 'Running', type 't2.micro', and availability zone 'us-east-2a'. A red arrow points to the 'Actions' dropdown menu at the top right of the table, specifically to the 'Change security groups' option.

63. On change security groups page, select the created security group.

The screenshot shows the 'Change security groups' page. At the top, it says 'Change security groups' and provides instructions: 'Amazon EC2 evaluates all the rules of the selected security groups to control inbound and outbound traffic to and from your instance. You can use this window to add and remove security groups.' Below this, there are two sections: 'Instance details' (Instance ID: i-0d25eb835f9f56b9b) and 'Associated security groups'. The 'Associated security groups' section contains a table with one row: 'Use: *sg-0012c41f2dcc7ac95* my-first-security-group (sg-0012c41f2dcc7ac95)'. A red arrow points to this row. At the bottom right of the page are 'Cancel' and 'Save' buttons.

64. Click on "Add security group".

The screenshot shows the AWS EC2 console with the URL us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#ChangeSecurityGroups:instanceId=i-0d25eb835f9f56b9b. The page displays 'Instance details' for instance ID `i-0d25eb835f9f56b9b` and network interface ID `eni-06456b62eb8efd6f8`. In the 'Associated security groups' section, there is a search bar containing `sg-0012c41f2dcc7ac95` and a blue 'Add security group' button. Below this, a table lists security groups associated with the network interface. The table has columns: Security group ID, Security group name, Description, and Owner ID. It shows two rows: one for `sg-099675804954285d4` named `launch-wizard-5` with description `launch-wizard-5 created 2025-10-16T03:17:22.687Z` and owner ID `536697231935`, and another row partially visible. At the bottom right of the interface is a toolbar with various icons, and a prominent orange 'Save' button.

65. See that the security group is now added; now click "Save".

The screenshot shows the same AWS EC2 console and URL as the previous screenshot. The 'Associated security groups' section now includes the newly added security group `my-first-security-group` (with ID `sg-0012c41f2dcc7ac95`), which is highlighted with a red box. The table below it remains the same. The bottom right of the interface still features the orange 'Save' button, which is also highlighted with a red arrow.

66. Security group now attached successfully; now copy the public address to test it on the browser again.

The screenshot shows the AWS EC2 Instances page. A green banner at the top indicates that security groups for the instance have changed successfully. The main table lists one instance: 'my-test-instance' (i-0d25eb835f9f56b9b), which is running and has an 't2.micro' instance type. It is associated with the 'us-east-2a' availability zone and has a public IPv4 address of 18.119.133.18. The 'Details' tab is selected, showing the instance's public IPv4 address (18.119.133.18) highlighted with a red box.

67. It is still unreachable because webserver is not automatically configured to serve webpage on a newly created instance.

The screenshot shows a browser window with the URL 18.119.133.18. The page displays an error message: "This site can't be reached" and "18.119.133.18 refused to connect." Below the message, there are troubleshooting steps: "Try:" followed by two bullet points: "• Checking the connection" and "• Checking the proxy and the firewall". At the bottom of the page, it says "ERR_CONNECTION_REFUSED".



This site can't be reached

18.119.133.18 refused to connect.

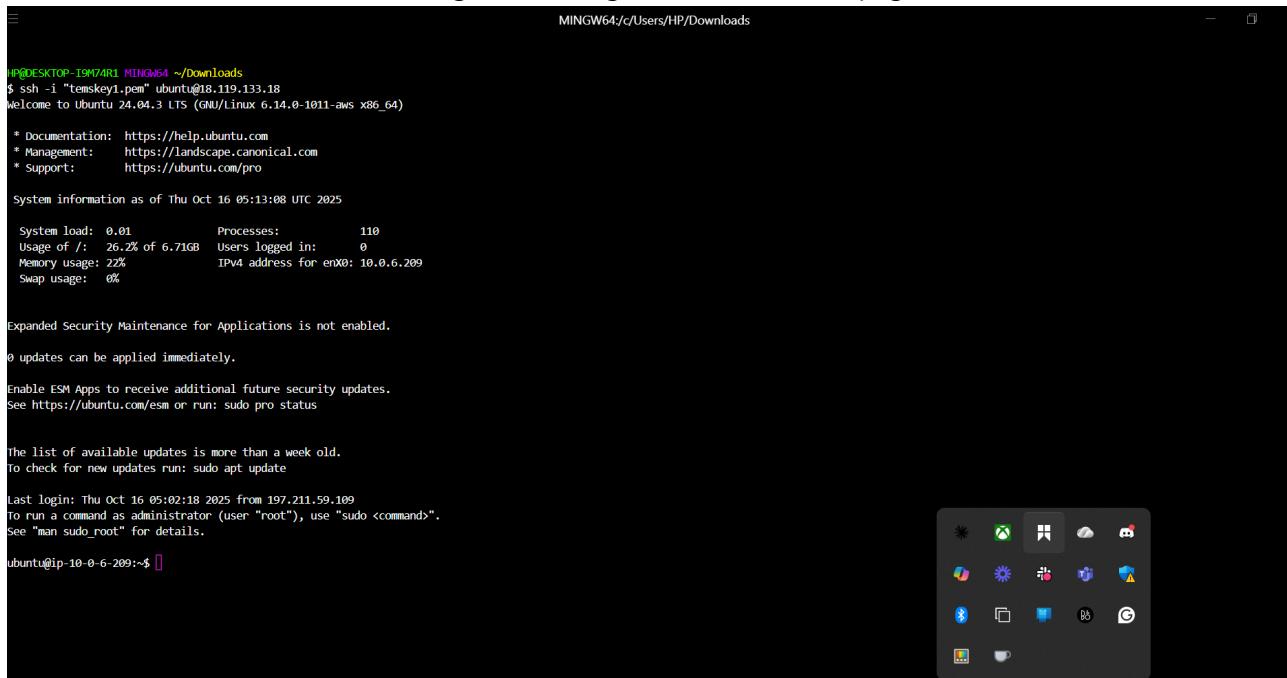
Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED



68. SSH into the Ubuntu server to manage and configure it to serve web page.



```

HPDESKTOP-T9M7AR1 MINGW64 ~ /Downloads
$ ssh -i "teamkey1.pem" ubuntu@18.119.133.18
Welcome to Ubuntu 24.04 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 16 05:13:08 UTC 2025

System load: 0.01 Processes: 110
Usage of /: 26.2% of 6.71GB Users logged in: 0
Memory usage: 22% IPv4 address for enx0: 10.0.6.209
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

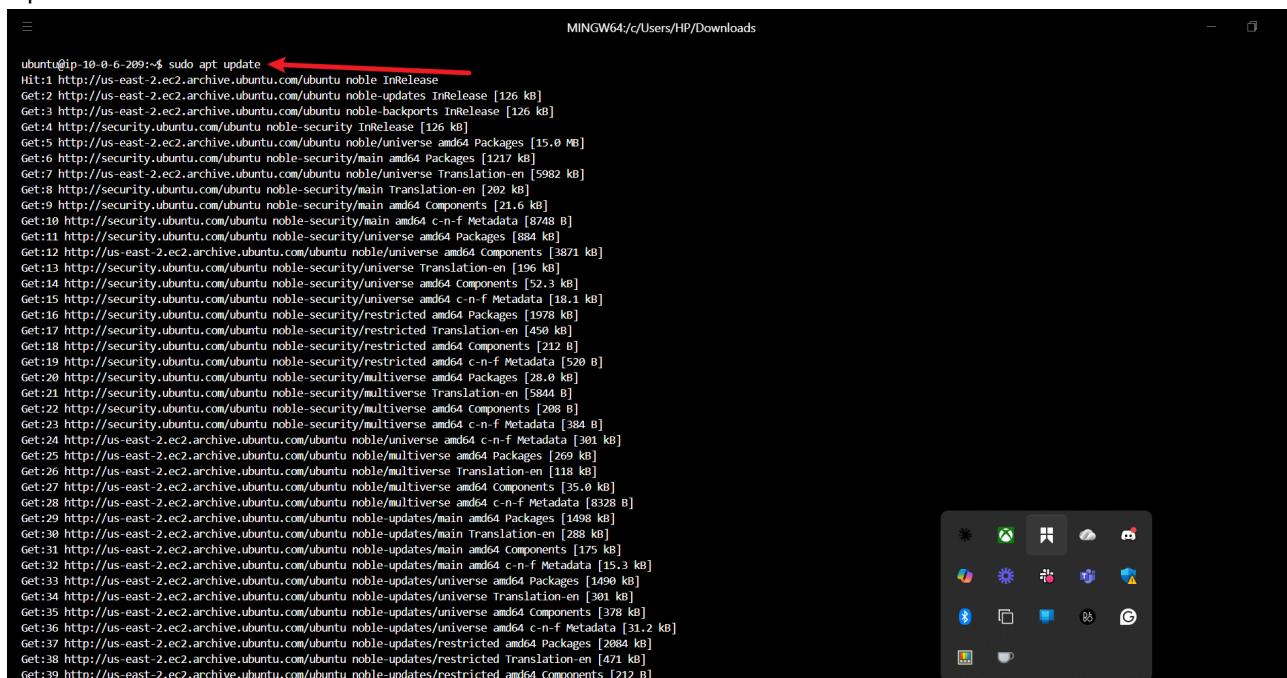
The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Thu Oct 16 05:02:18 2025 from 197.211.59.109
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-6-209:~$ 

```

69. Update Ubuntu server.



```

ubuntu@ip-10-0-6-209:~$ sudo apt update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu/noble-security/main amd64 Packages [217 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/universe Translation-en [5982 kB]
Get:8 http://security.ubuntu.com/ubuntu/noble-security/main Translation-en [202 kB]
Get:9 http://security.ubuntu.com/ubuntu/noble-security/main amd64 Components [21.6 kB]
Get:10 http://security.ubuntu.com/ubuntu/noble-security/main amd64 c-n-f Metadata [8748 B]
Get:11 http://security.ubuntu.com/ubuntu/noble-security/universe amd64 Packages [884 kB]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/noble/universe amd64 Components [3871 kB]
Get:13 http://security.ubuntu.com/ubuntu/noble-security/universe Translation-en [196 kB]
Get:14 http://security.ubuntu.com/ubuntu/noble-security/universe amd64 Components [52.3 kB]
Get:15 http://security.ubuntu.com/ubuntu/noble-security/universe amd64 c-n-f Metadata [18.1 kB]
Get:16 http://security.ubuntu.com/ubuntu/noble-security/restricted amd64 Packages [1978 kB]
Get:17 http://security.ubuntu.com/ubuntu/noble-security/restricted Translation-en [450 kB]
Get:18 http://security.ubuntu.com/ubuntu/noble-security/restricted amd64 Components [212 kB]
Get:19 http://security.ubuntu.com/ubuntu/noble-security/restricted amd64 c-n-f Metadata [520 kB]
Get:20 http://security.ubuntu.com/ubuntu/noble-security/multiverse amd64 Packages [28.0 kB]
Get:21 http://security.ubuntu.com/ubuntu/noble-security/multiverse Translation-en [5844 B]
Get:22 http://security.ubuntu.com/ubuntu/noble-security/multiverse amd64 Components [208 kB]
Get:23 http://security.ubuntu.com/ubuntu/noble-security/multiverse amd64 c-n-f Metadata [388 kB]
Get:24 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/noble/universe amd64 c-n-f Metadata [301 kB]
Get:25 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/noble/multiverse amd64 Packages [269 kB]
Get:26 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/noble/multiverse Translation-en [118 kB]
Get:27 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/noble/multiverse amd64 Components [35.0 kB]
Get:28 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:29 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/noble-updates/main amd64 Packages [1498 kB]
Get:30 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/noble-updates/main Translation-en [288 kB]
Get:31 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/noble-updates/main amd64 Components [175 kB]
Get:32 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/noble-updates/main amd64 c-n-f Metadata [15.3 kB]
Get:33 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/noble-updates/universe amd64 Packages [1490 kB]
Get:34 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/noble-updates/universe Translation-en [301 kB]
Get:35 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/noble-updates/universe amd64 Components [378 kB]
Get:36 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/noble-updates/universe amd64 c-n-f Metadata [31.2 kB]
Get:37 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/noble-updates/restricted amd64 Packages [2084 kB]
Get:38 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/noble-updates/restricted Translation-en [471 kB]
Get:39 http://us-east-2.ec2.archive.ubuntu.com/ubuntu/noble-updates/restricted amd64 Components [212 B]

```

70. Install apache2.

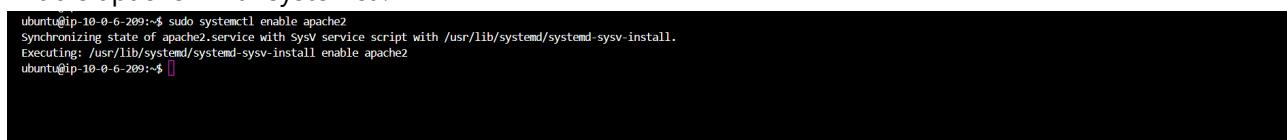


```

ubuntu@ip-10-0-6-209:~$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.8).
0 upgraded, 0 newly installed, 0 to remove and 53 not upgraded.

```

71. Enable apache2 with systemctl.



```

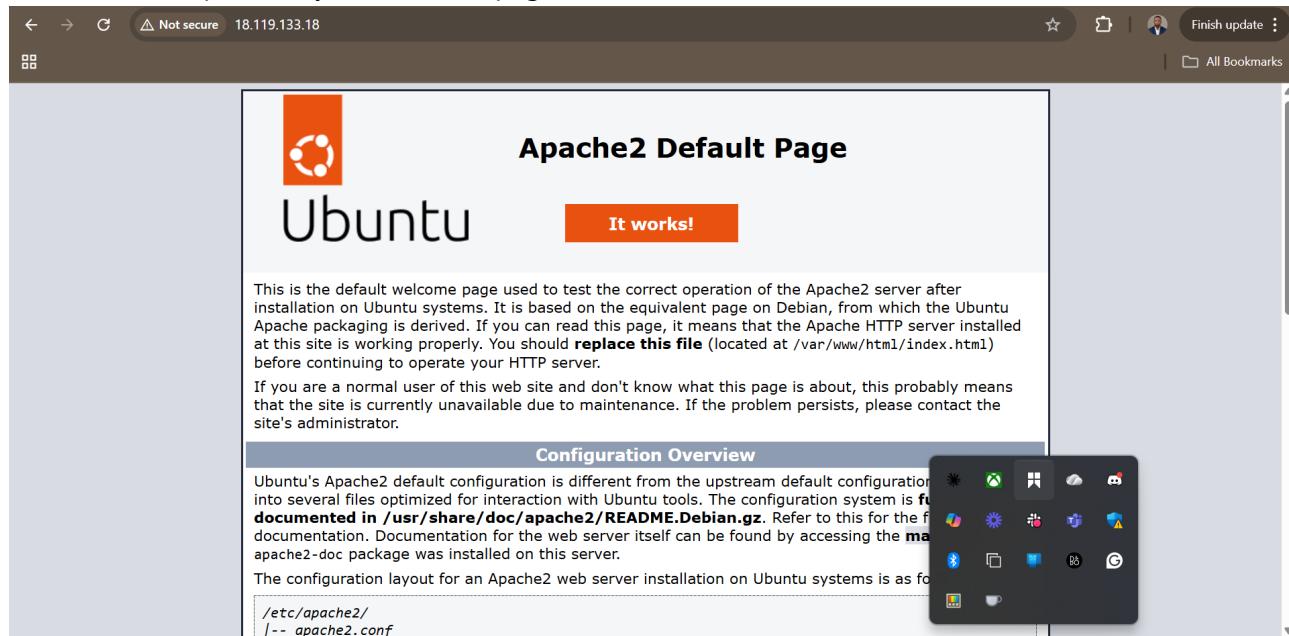
ubuntu@ip-10-0-6-209:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
ubuntu@ip-10-0-6-209:~$ 

```

72. Start apache2 with systemctl.

```
ubuntu:ip-10-0-6-209:~$ sudo systemctl start apache2
ubuntu:ip-10-0-6-209:~$
```

73. Reload and the previously unreachable page is now reachable on the browser.



74. SG Overview - Inbound rule allows SSH and HTTP traffic from any source hence webserver reachable over the internet via CLI (SSH) and browser (HTTP).

The screenshot shows the AWS EC2 Security Groups console. A security group named "my-first-security-group" is selected. The "Inbound rules" tab is active, displaying two rules:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sgr-0a11f4d7e80cc4e16	IPv4	SSH	TCP	22	0.0.0.0/0	-
-	sgr-09b26fe06ef059a0e	IPv4	HTTP	TCP	80	0.0.0.0/0	-

75. SG Overview - Outbound rules allows all traffic for all protocol through all port to exit the server.

The screenshot shows the AWS EC2 Security Groups console. The left sidebar navigation includes: EC2 Global View, Events, Instances, Images, Elastic Block Store, Network & Security, and a CloudShell link. The main content area displays the details for 'sg-0012c41f2dcc7ac95 - my-first-security-group'. The 'Outbound rules' tab is selected. A table lists one rule: Name: sgr-01396059a9344bbb7, IP version: IPv4, Type: All traffic, Protocol: All, Port range: All, Destination: 0.0.0.0/0. The 'Edit outbound rules' button is located at the top right of the table. A small screenshot of a Windows desktop interface is overlaid on the bottom right.

Scenario 1: Inbound Rules for HTTP/SSH, Outbound for All (Default Successful Access)

This is the initial configuration where inbound allows HTTP and SSH from anywhere, and outbound allows all traffic. The website is reachable.

Scenario 2: Inbound Rules for HTTP/SSH, No Outbound Rules

76. On the outbound rules tab, click "Edit outbound rules".

The screenshot shows the AWS EC2 Security Groups console, identical to the previous one but with a red arrow pointing to the 'Edit outbound rules' button in the 'Outbound rules' table. The table shows the same one rule as before: Name: sgr-01396059a9344bbb7, IP version: IPv4, Type: All traffic, Protocol: All, Port range: All, Destination: 0.0.0.0/0.

77. Click "Delete" to delete outbound rule and click "Save rules".

The screenshot shows the 'Edit outbound rules' interface for a security group. It lists one rule: '0.0.0.0/0' with 'All traffic' selected. At the bottom, there are buttons for 'Cancel', 'Preview changes', and a prominent orange 'Save rules' button.

78. Outbound security rules now deleted.

The screenshot shows the 'Details' page for the security group. It displays basic information like the security group name, ID, owner, and VPC ID. The 'Outbound rules' section shows a table with no data, indicating they have been successfully deleted.

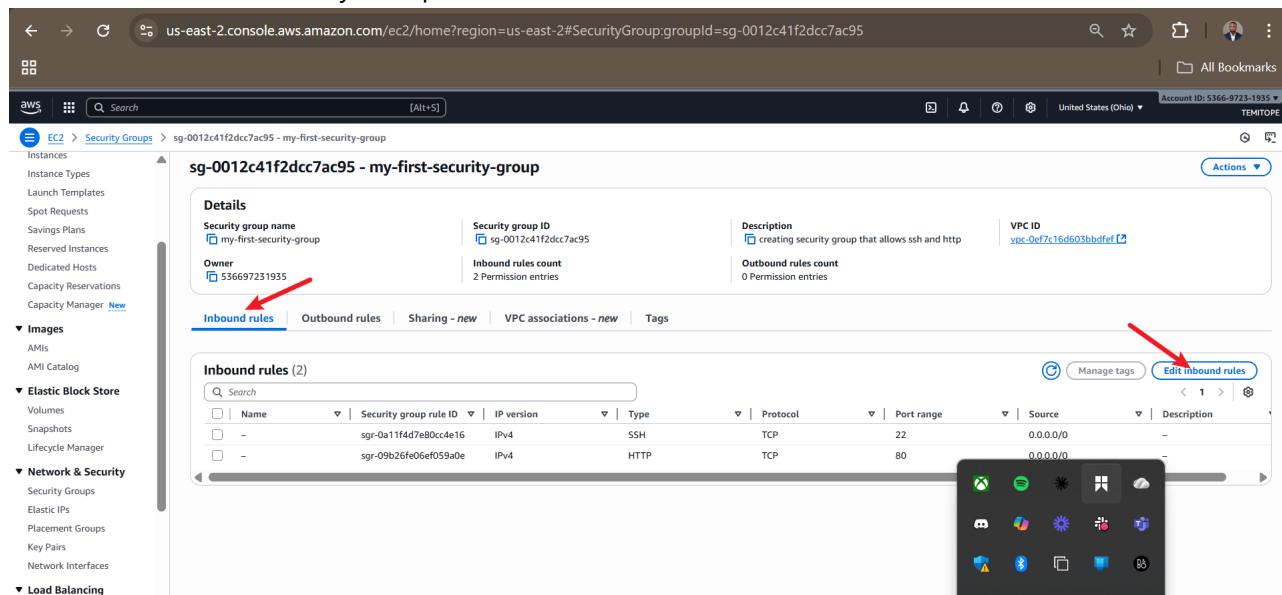
79. Website reachability tested after deleting the outbound rules and nothing breaks as website is still reachable.



(Website remains reachable due to the stateful nature of Security Groups.)

Scenario 3: Remove Both Inbound and Outbound Rules

80. Back to the created Security Group, click "Inbound rules" and then "Edit inbound rules".



81. Delete both inbound rules and click "Save rules".

82. Like outbound rules, inbound rules have now been deleted.

83. Web page now unreachable again.

(No access without inbound rules.)

Scenario 4: No Inbound Rules, Outbound Rules for HTTP

84. Back to the created Security Group, click "Outbound rules" and then "Edit outbound rules".

The screenshot shows the AWS EC2 Security Groups console. On the left, there's a navigation sidebar with categories like Instances, Launch Templates, and Network & Security. The main area shows a security group named 'sg-0012c41f2dcc7ac95 - my-first-security-group'. The 'Details' section shows the security group name, ID, owner, and VPC ID. Below it, tabs for 'Inbound rules' and 'Outbound rules' are present, with 'Outbound rules' being the active tab. A red arrow points to the 'Edit outbound rules' button located at the top right of the table header. The table below shows no security group rules found.

85. Click on "Add rules".

The screenshot shows the 'Edit outbound rules' page for the same security group. The 'Outbound rules info' section states that the security group has no outbound rules. A red arrow points to the 'Add rule' button at the bottom left of this section. At the bottom right of the page, there are 'Cancel', 'Preview changes', and 'Save rules' buttons.

86. Set type HTTP, protocol TCP, port 80, Destination Anywhere, and save rules.

The screenshot shows the 'Edit outbound rules' interface for a security group. The 'Type' dropdown is set to 'HTTP', 'Protocol' to 'TCP', and 'Port range' to '80'. The 'Destination' dropdown is set to 'Anywhere...'. A note at the bottom says: 'Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses.' The 'Save rules' button is highlighted with a red arrow.

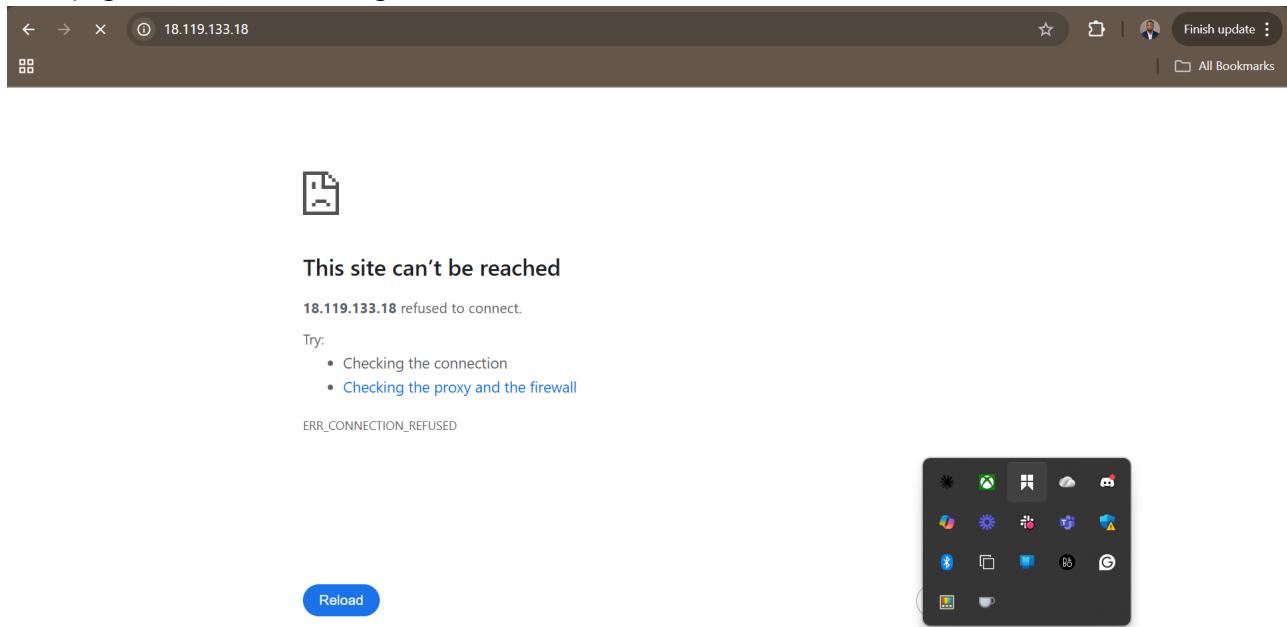
87. Outbound rules successfully modified.

The screenshot shows the 'sg-0012c41f2dcc7ac95 - my-first-security-group' details page. The 'Outbound rules' section is highlighted with a red box. It shows one rule named 'sgr-04ec66f62ea7d3116' with the following details: Type: HTTP, Port range: 80, Destination: 0.0.0.0/0.

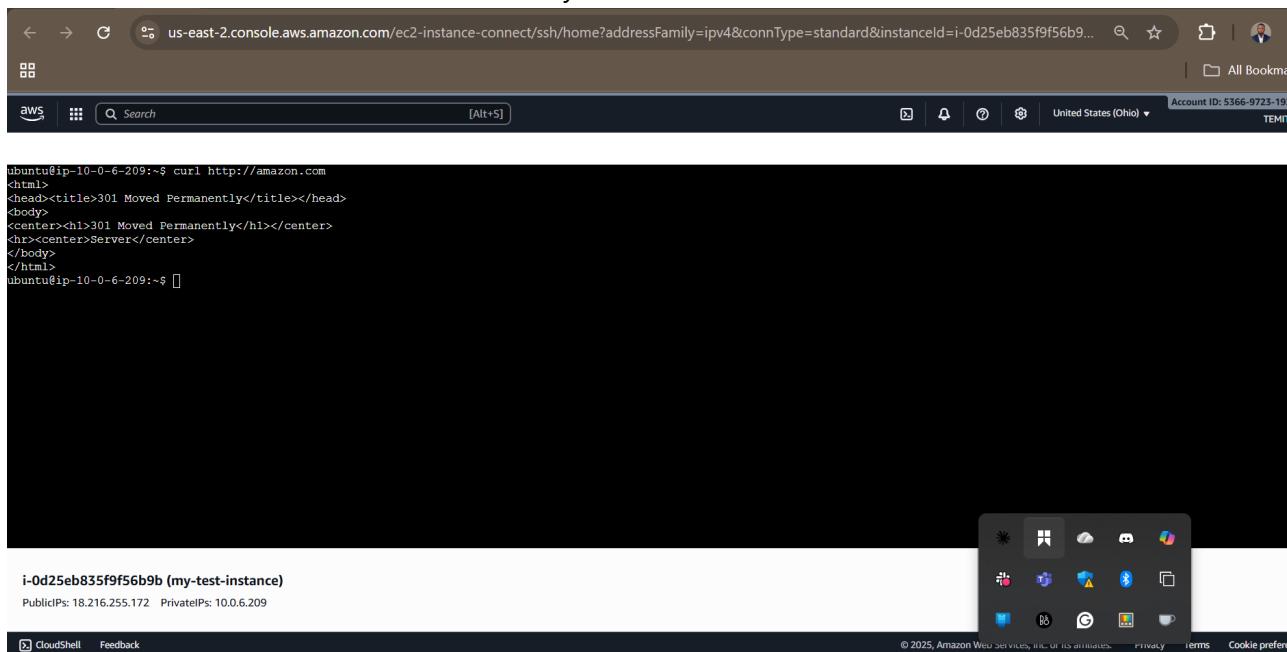
88. Note that inbound rules is still empty because we deleted it.

The screenshot shows the 'sg-0012c41f2dcc7ac95 - my-first-security-group' details page. The 'Inbound rules' section is highlighted with a red box. It displays the message: 'No security group rules found'.

89. Web page now unreachable again.



90. Note that the outbound traffic works normally when we curl amazon.com via the instance connect CLI.



90b. Reset the SG to allow inbound HTTP traffic; we need this to test the NACL as we go further.

The screenshot shows the AWS EC2 Security Groups console. A green success message at the top states: "Inbound security group rules successfully modified on security group (sg-0012c41f2dcc7ac95 | my-first-security-group)". Below this, the "Details" section for the security group "sg-0012c41f2dcc7ac95 - my-first-security-group" is displayed. The "Inbound rules" tab is selected, showing one rule: "sgr-08c8d512f02a38ab7" (Security group rule ID), IPv4 (IP version), TCP (Protocol), port 80 (Port range), and 0.0.0.0/0 (Source). The entire "Inbound rules" table is highlighted with a red box.

(Outbound allows traffic, but no inbound means the website is unreachable from outside.)

Part 2: NACLs

We now configure NACLs at the subnet level, examining defaults and modifying rules to allow/deny traffic, demonstrating their stateless nature.

91. Navigate to VPC using the console search.

The screenshot shows the AWS console search results for "vpc". The search bar at the top contains "vpc". The results are categorized under "Services" and "Features". Under "Services", the "VPC" card is highlighted with a red box. Other cards include "AWS Global View" and "AWS Firewall Manager". Under "Features", there are cards for "Dashboard", "Route 53 VPCs", and "VPC links". A sidebar on the left shows navigation links for EC2, VPC, Lambda, and other services. A "CloudShell" button is at the bottom left. The bottom right corner features a "Connect" button with a QR code.

92. Click on "Network ACLs" to navigate to it.

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Security' section, 'Network ACLs' is highlighted with a red arrow. The main content area displays 'Resources by Region' for the Ohio region, including sections for 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 groups, and Customer Gateways. Each section has a 'See all regions' link. A modal window titled 'Service Health' is partially visible on the right.

93. Click on "Create network ACL".

The screenshot shows the 'Network ACLs' list page. The left sidebar shows the 'VPC' icon selected. The main area displays a table of existing Network ACLs with two entries: 'acl-023c92d8018173470' associated with 2 Subnets and 'acl-0c388cfaf99ea79x8' associated with 3 Subnets. Both have 'Default' set to 'Yes' and 'VPC ID' set to 'vpc-0ef7c16d603bbdef'. The 'Actions' dropdown menu at the top right has a red arrow pointing to the 'Create network ACL' button. A modal window titled 'Select a network ACL' is visible at the bottom.

94. Name network ACL, select the created VPC, and click "Create network ACL".

The screenshot shows the 'Create network ACL' page in the AWS VPC console. The 'Network ACL settings' section includes fields for 'Name - optional' (set to 'my-first-NACL') and 'VPC' (set to 'vpc-0ef7c16d603bbdef'). The 'Tags' section shows a single tag 'Name: my-first-NACL'. At the bottom right of the main form area, there is a large red arrow pointing from the 'Create network ACL' button to another 'Create network ACL' button located at the top right of the page.

95. Note that it denies all inbound traffic by default.

The screenshot shows the 'Network ACLs (1/3)' page. It lists three Network ACLs: 'acl-023c92d8018173470' (2 Subnets, Default Yes, 2 Inbound rules), 'acl-0c388cf99ea79k8' (3 Subnets, Default Yes, 2 Inbound rules), and 'my-first-NACL' (selected, 1 Subnet, Default No, 1 Inbound rule). Below the list is a detailed view for 'acl-06b418e422ce85144 / my-first-NACL', specifically the 'Inbound rules' tab. A red box highlights the 'Edit inbound rules' button.

96. All outbound traffic are also denied by default.

The screenshot shows the 'Network ACLs (1/3)' page, identical to the previous one. It lists the same three Network ACLs. Below the list is a detailed view for 'acl-06b418e422ce85144 / my-first-NACL', specifically the 'Outbound rules' tab. A red box highlights the 'Edit outbound rules' button.

Examine Default Settings

Defaults deny all traffic (as shown above).

Modify Inbound to Permit All

97. Click on "Inbound rules" tab then "Edit inbound rules".

Name	Network ACL ID	Associated with	Default	VPC ID	Inbound rules count
acl-025c92d8018173470	2 Subnets		Yes	vpc-0ef7c16d603bbdef	2 Inbound rules
acl-0c388cfaf99ea79x8	3 Subnets		Yes	vpc-0a946dc2d5bf4cbe	2 Inbound rules
my-first-NACL	acl-06b418e422ce85144		No	vpc-0ef7c16d603bbdef	1 Inbound rule

my-first-NACL / acl-06b418e422ce85144

Inbound rules (1)

Edit inbound rules

98. Click on "Add new rule".

Rule number	Type	Protocol	Port range	Source	Allow/Deny
*	All traffic	All	All	0.0.0.0/0	Deny

Add new rule

Preview changes

99. Set rule number 1, all traffic, any source, then allow and click "Save changes".

Edit inbound rules Info
Inbound rules control the incoming traffic that's allowed to reach the VPC.

Rule number <small>Info</small>	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Allow/Deny <small>Info</small>
1	All traffic	All	All	0.0.0.0/0	Allow
<input type="button" value="Add new rule"/> <input type="button" value="Sort by rule number"/>					

Cancel

100. Successfully updated inbound rules for NACL; click "Actions" and "Subnet association" to associate NACL with subnet.

You have successfully updated inbound rules for acl-06b418e422ce85144 / my-first-NACL

Network ACLs (1/3) Info

Name	Network ACL ID	Associated with	Default	VPC ID
acl-023c92d8018173470	2 Subnets	Yes	vpc-0ef7c16d603bbdfef	
acl-0388cfae99ea79c8	3 Subnets	Yes	vpc-0946dcfd5bf4cbe	
my-first-NACL	acl-06b418e422ce85144	-	No	vpc-0ef7c16d603bbdfef

Actions

my-first-NACL

Details

Details

Network ACL ID <input checked="" type="checkbox" value="acl-06b418e422ce85144"/>	Associated with -	Default No
Owner <input checked="" type="checkbox" value="536697231935"/>		

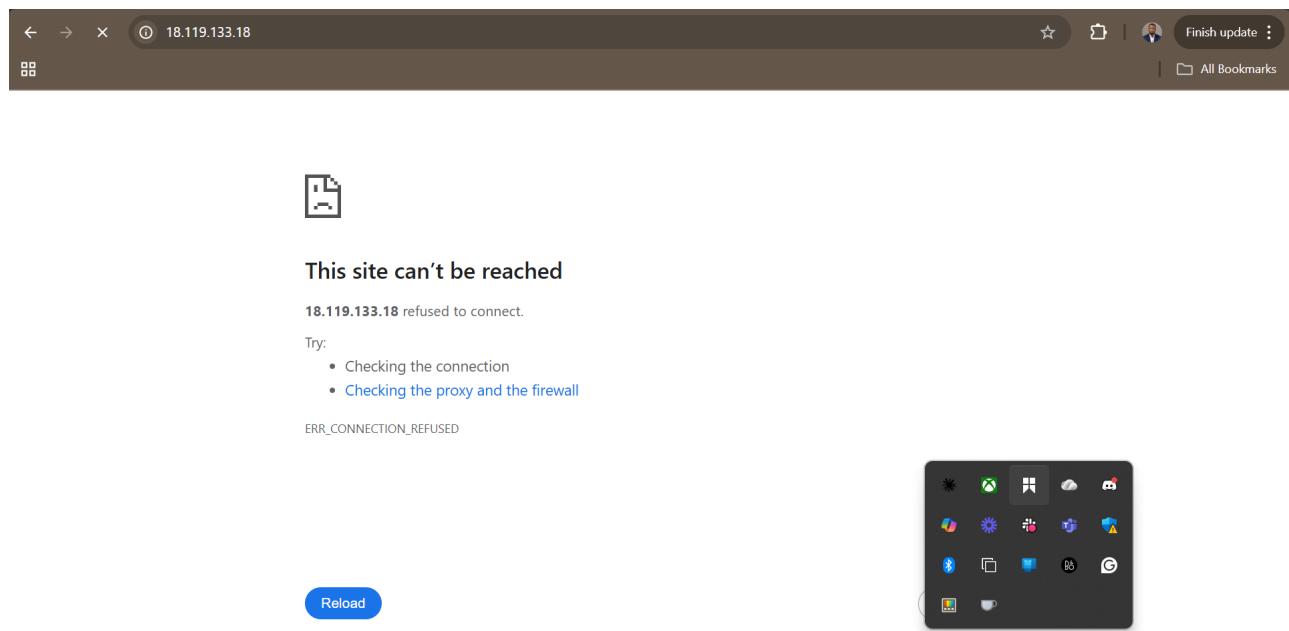
101. On edit subnet page, click public subnet and save changes.

The screenshot shows the 'Edit subnet associations' page for a specific Network ACL. In the 'Available subnets' table, the row for 'my-public-subnet-1' has its checkbox selected. In the 'Selected subnets' section, the same subnet is listed with a delete icon. At the bottom right, there are 'Cancel' and 'Save changes' buttons, with 'Save changes' highlighted by a red arrow.

102. Successfully updated subnet association.

The screenshot shows the 'Network ACLs' page. A green success message box at the top states: 'You have successfully updated subnet associations for acl-06b418e422ce85144 / my-first-NACL.' Below this, the 'Network ACLs' table lists three entries. The entry for 'my-first-NACL' is selected and shows its details in the 'acl-06b418e422ce85144 / my-first-NACL' card. The 'Subnet associations' tab is active, displaying the subnet it is associated with: 'subnet-06c7cfe7c9fb96a9b / my-public-subnet-1'. Other tabs include 'Details', 'Inbound rules', 'Outbound rules', and 'Tags'.

103. Web page unreachable even though we set inbound rules to allow all traffic; this is because NACL is stateless.



104. Even though the inbound rules allow traffic, the outbound rules deny.

Rule number	Type	Protocol	Port range	Source	Allow/Deny
1	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Allow
*	All traffic	All	All	0.0.0.0/0	<input type="checkbox"/> Deny

105. Here is outbound rules that denies.

The screenshot shows the AWS VPC Network ACLs console. A green success message at the top says "You have successfully updated subnet associations for acl-06b418e422ce85144 / my-first-NACL." Below it, the "Network ACLs (1/3)" table lists three entries:

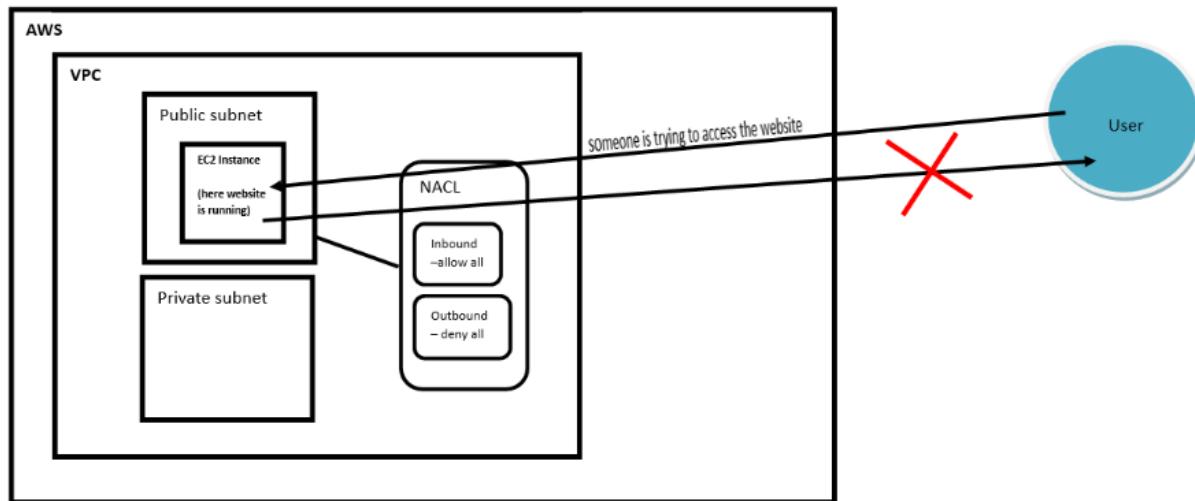
Name	Network ACL ID	Associated with	Default	VPC ID	Inbound rules count
acl-023c92d8018175470	subnet-0857e455da7700fea / my-private-subnet	Yes	ypc-0ef7c16d603bbdef	2 Inbound rules	
acl-0c388cfae99ea79c8	3 Subnets	Yes	ypc-0a946dec2d5bf4cbe	2 Inbound rules	
my-first-NACL	acl-06b418e422ce85144	subnet-06c7fce7c9fb96a9b / my-public-subnet-1	No	ypc-0ef7c16d603bbdef	2 Inbound rules

On the right, under "Outbound rules", there is a table with one row:

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
*	All traffic	All	All	0.0.0.0/0	Deny

A red arrow points to the "edit outbound rules" button.

106. Here is the architecture overview that makes the website unviewable.



Not able to see website because you are able to go inside of the subnet because of the inbound rule (allow all) but any traffic from subnet is not allowed to go outside due to the outbound rule (deny all).

Adjust Outbound to Allow All

107. Select created NACL, click the "Outbound rules" tab, and edit outbound rules.

The screenshot shows the AWS VPC Network ACLs page. In the left sidebar, under the 'Network ACLs' section, there is a table listing three Network ACLs. The third row, 'my-first-NACL', has its checkbox checked and is highlighted with a blue border. Below the table, a sub-section titled 'acl-06b418e422ce85144 / my-first-NACL' is displayed. It contains tabs for 'Details', 'Inbound rules', 'Outbound rules' (which is currently selected), 'Subnet associations', and 'Tags'. A red arrow points from the text above to the 'Outbound rules' tab. Another red arrow points to the 'Edit outbound rules' button in the top right corner of the sub-section.

108. Click "Add new rule".

The screenshot shows the 'Edit outbound rules' page for the 'my-first-NACL'. At the top, it says 'Edit outbound rules' and 'Outbound rules control the outgoing traffic that's allowed to leave the VPC.' Below this is a table header with columns: Rule number Info, Type Info, Protocol Info, Port range Info, Destination Info, and Allow/Deny Info. A red arrow points to the 'Add new rule' button at the bottom left of the table. At the very bottom of the page, there are buttons for 'Cancel', 'Preview changes', and 'Save changes'. A red arrow also points to the 'Add new rule' button on this line. The bottom of the screen features a standard AWS navigation bar with links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

109. Set rule number to 1, all traffic, all destination, allow, save changes.

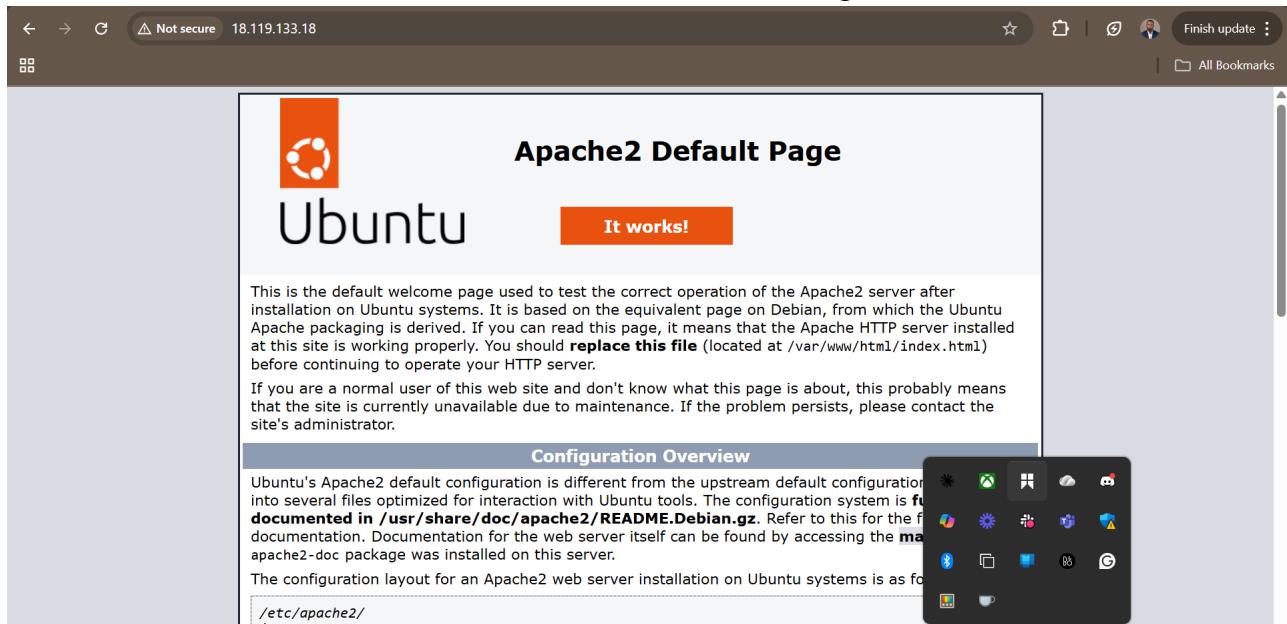
The screenshot shows the 'Edit outbound rules' interface. The 'Rule number' is set to 1, 'Type' is 'All traffic', 'Protocol' is 'All', 'Port range' is 'All', 'Destination' is '0.0.0.0/0', and 'Allow/Deny' is 'Allow'. The 'Save changes' button is highlighted with a red arrow.

110. Outbound traffic updated successfully allowing all traffic.

The screenshot shows the 'Network ACLs' list with a success message: 'You have successfully updated outbound rules for acl-06b418e422ce85144 / my-first-NACL'. The 'Outbound rules' tab is selected, displaying the following rules:

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
1	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

111. Website now reachable due to inbound and outbound rules allowing traffic.



Additional Scenario: Allow All in SG, Deny All in NACL

112. In this new scenario, set security group inbound rules to allow HTTP and SSH from anywhere.

The screenshot shows the AWS CloudWatch Metrics interface. A line graph titled "AWS Lambda Metrics" displays two metrics over time. The Y-axis represents the metric value, and the X-axis represents time. Two data series are shown: "Function Invocations" (blue line) and "Execution Duration" (orange line). Both metrics show a significant spike starting around October 1st, 2025. The "Function Invocations" metric reaches approximately 10,000,000 by October 10th, while the "Execution Duration" metric reaches approximately 100 seconds.

113. Also set security group outbound rules to allow all traffic to anywhere and save rules.

The screenshot shows the 'Edit outbound rules' page for a security group. It displays a single rule: 'Security group rule ID sgr-04ec66f62ea7d3116' with 'Type' set to 'All traffic'. The 'Destination' dropdown is set to 'Anywhere...'. A note at the bottom left says: '⚠️ Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses.' At the bottom right, there are 'Cancel', 'Preview changes', and 'Save rules' buttons. A red arrow points from the 'Save rules' button to the 'Preview changes' button.

114. All types of outbound traffic now allowed to any destination over the internet.

The screenshot shows the 'Outbound rules' section for the security group 'sg-0012c41f2dcc7ac95 - my-first-security-group'. It displays one outbound rule: 'sgr-066daa43f5a6f8ca1' with 'IP version' set to 'IPv4', 'Type' to 'All traffic', 'Protocol' to 'All', and 'Port range' to 'All'. The 'Destination' field is set to '0.0.0.0/0'. Below the table are 'Manage tags' and 'Edit outbound rules' buttons. The sidebar on the left includes sections for Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), and Load Balancing (Load Balancers, Target Groups).

115. Navigate back to NACL, remove NACL inbound rules that allow all traffic, and save rule.

The screenshot shows the 'Edit inbound rules' page for a specific Network ACL. The table contains one rule with the following details:

Rule number	Type Info	Protocol Info	Port range Info	Source Info	Allow/Deny Info
1	All traffic	All	All	0.0.0.0/0	Allow

At the bottom of the page, there are three buttons: 'Cancel', 'Preview changes', and 'Save changes'. A red arrow points from the 'Remove' button in the table to the 'Save changes' button.

116. NACL inbound rules now deny all traffic types of all protocols on all port from all source.

The screenshot shows the 'Inbound rules' page for the 'my-first-NACL'. The table displays one rule with the following configuration:

Rule number	Type	Protocol	Port range	Source	Allow/Deny
*	All traffic	All	All	0.0.0.0/0	Deny

At the top of the page, a green message box states: 'You have successfully updated inbound rules for acl-06b418e422ce85144 / my-first-NACL'. Below the table, there is an 'Edit inbound rules' button. A red box surrounds both the 'Edit inbound rules' button and the 'Deny' checkbox in the rule table, and a red arrow points from the 'Deny' checkbox to the 'Edit inbound rules' button.

117. Also remove outbound rules that allow all traffic to all destination and save changes.

The screenshot shows the 'Edit outbound rules' interface for a specific Network ACL. There is one rule listed:

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
1	All traffic	All	All	0.0.0.0/0	Allow

At the bottom right of the page, there are 'Cancel', 'Preview changes', and 'Save changes' buttons. A red arrow points to the 'Save changes' button.

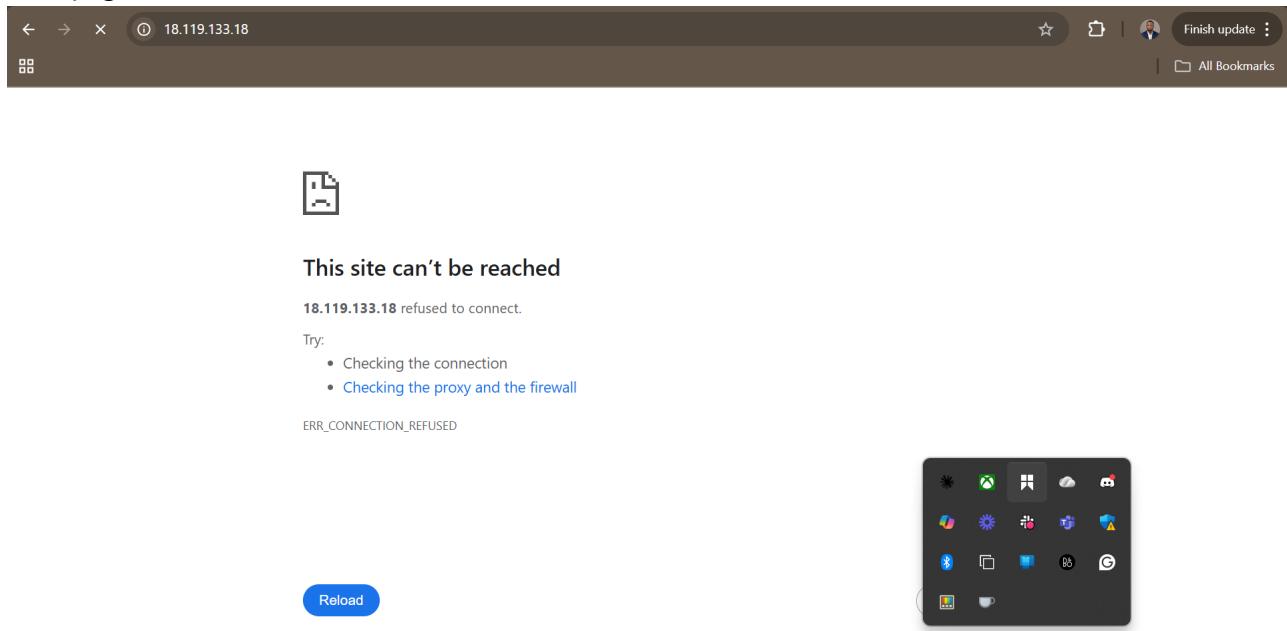
118. NACL outbound rules now deny all traffic types of all protocols on all port to all destinations.

The screenshot shows the Network ACLs list. One Network ACL is selected: 'my-first-NACL' (acl-06b418e422ce85144). The 'Outbound rules' tab is selected. The table shows one rule:

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
*	All traffic	All	All	0.0.0.0/0	Deny

A red box highlights the 'Outbound rules' tab, and another red box highlights the 'Deny' option in the 'Allow/Deny' dropdown.

119. Web page unreachable because NACL denies traffic.



(Even with permissive SG rules, NACL denials block traffic at the subnet level.)

Project Reflection

Successfully configured Security Groups and NACLs to control inbound and outbound traffic in AWS. Identified the differences between Security Groups and NACLs and their respective roles in network security. Explored various scenarios to understand how Security Groups and NACLs interact and impact network traffic. Learned valuable troubleshooting techniques for diagnosing and resolving network connectivity issues in AWS. Overall, gained practical experience and confidence in managing network security within AWS environments.