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.

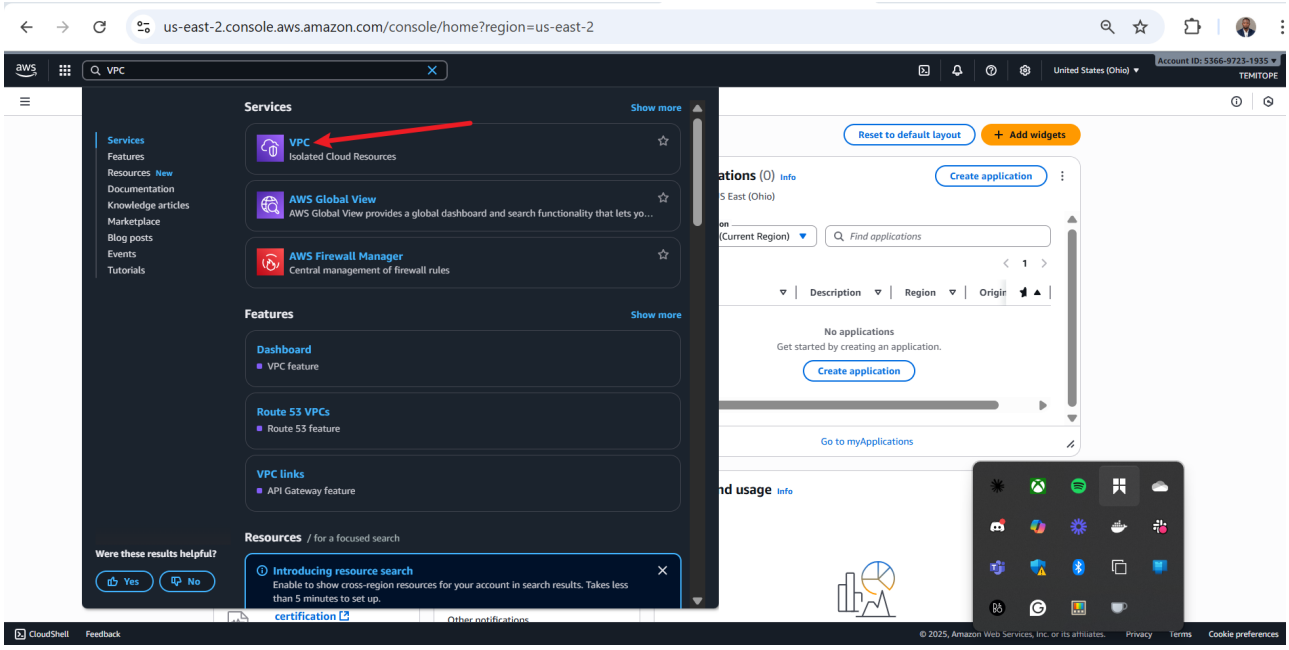
Note: In security groups, there's no explicit "deny" option as seen in NACLs; any rule configured within a security group implies permission.

Practical Implementation

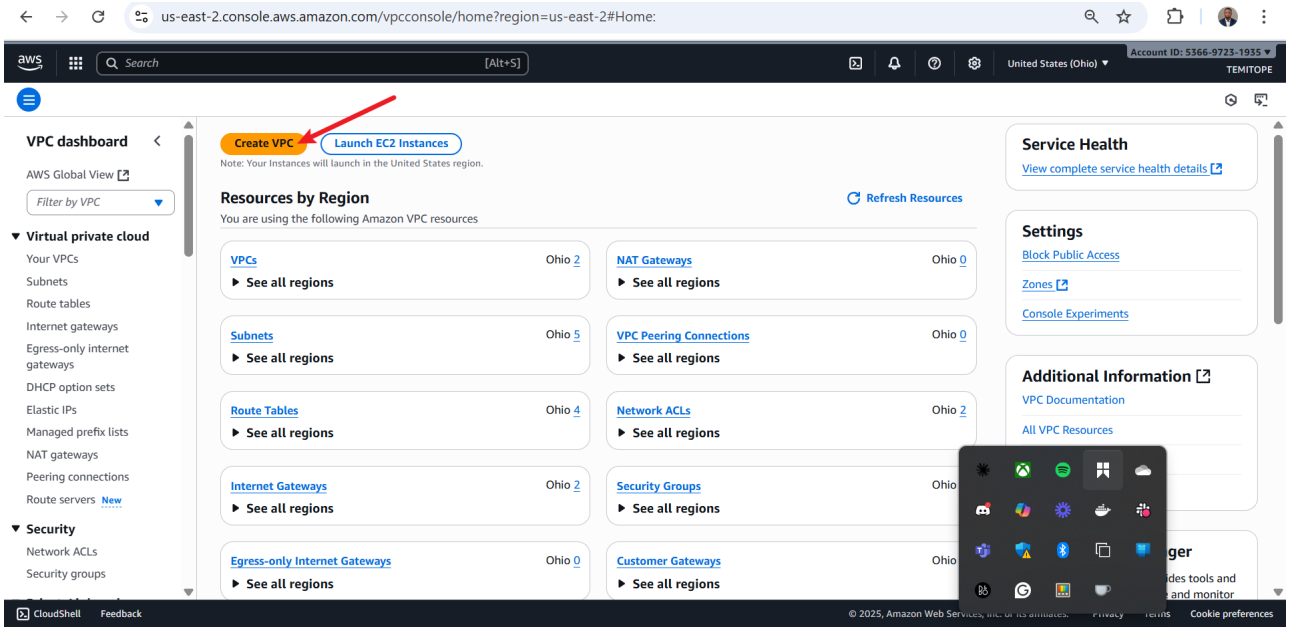
The practical part is divided into setup and two main sections: Security Groups and NACLs. We begin by setting up the foundational VPC environment, including subnets, internet gateways, route tables, NAT gateways, and an EC2 instance. Then, we configure and test Security Groups and NACLs in various scenarios.

Setting Up the VPC Environment

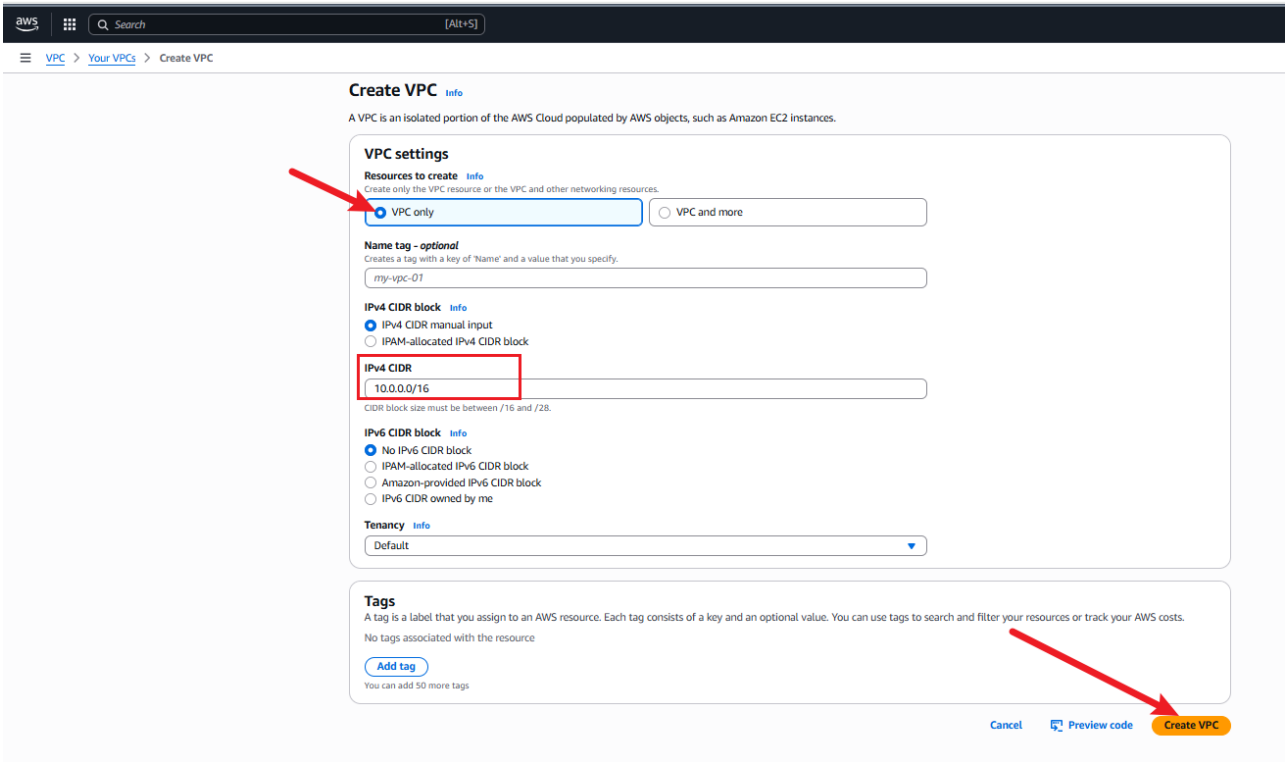
1. Navigate to the search bar and enter "VPC", then click on the VPC service that appears.



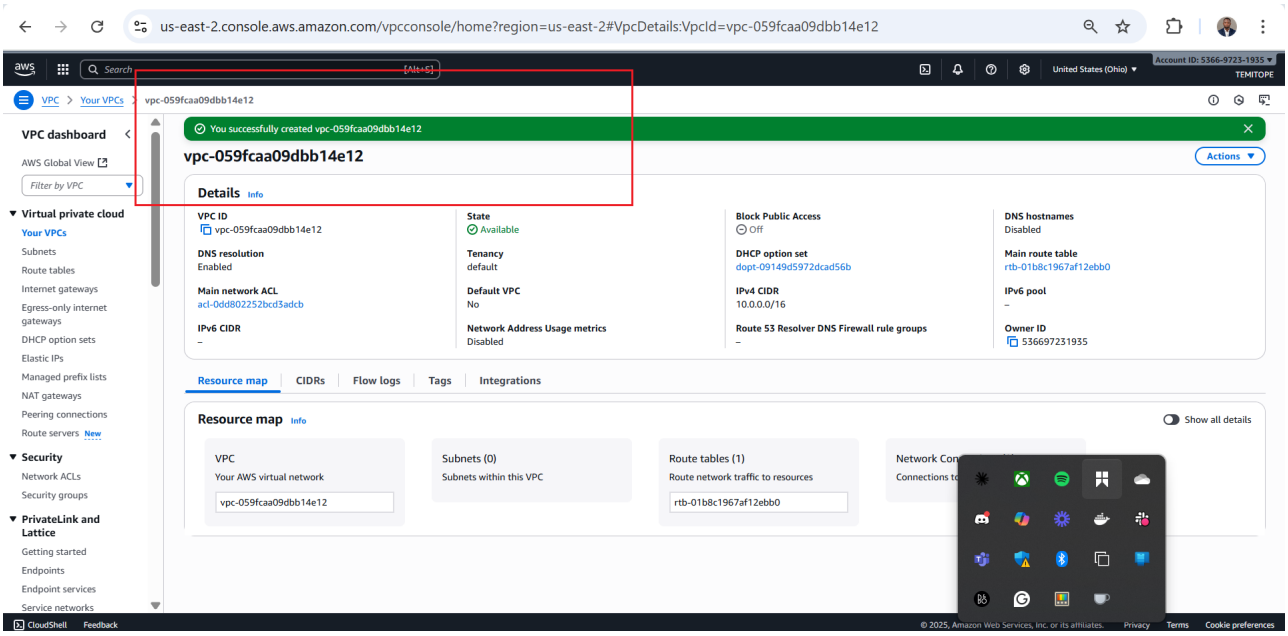
2. On the VPC dashboard, click "Create VPC".



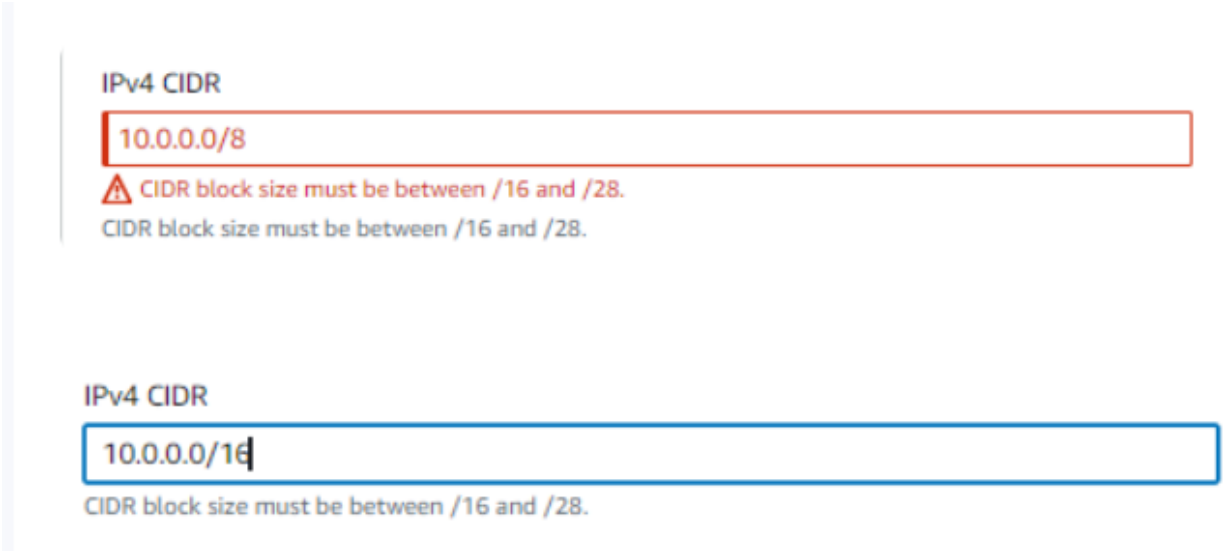
3. On the Create VPC page, select "VPC only", set the IPv4 CIDR block, and click "Create VPC".



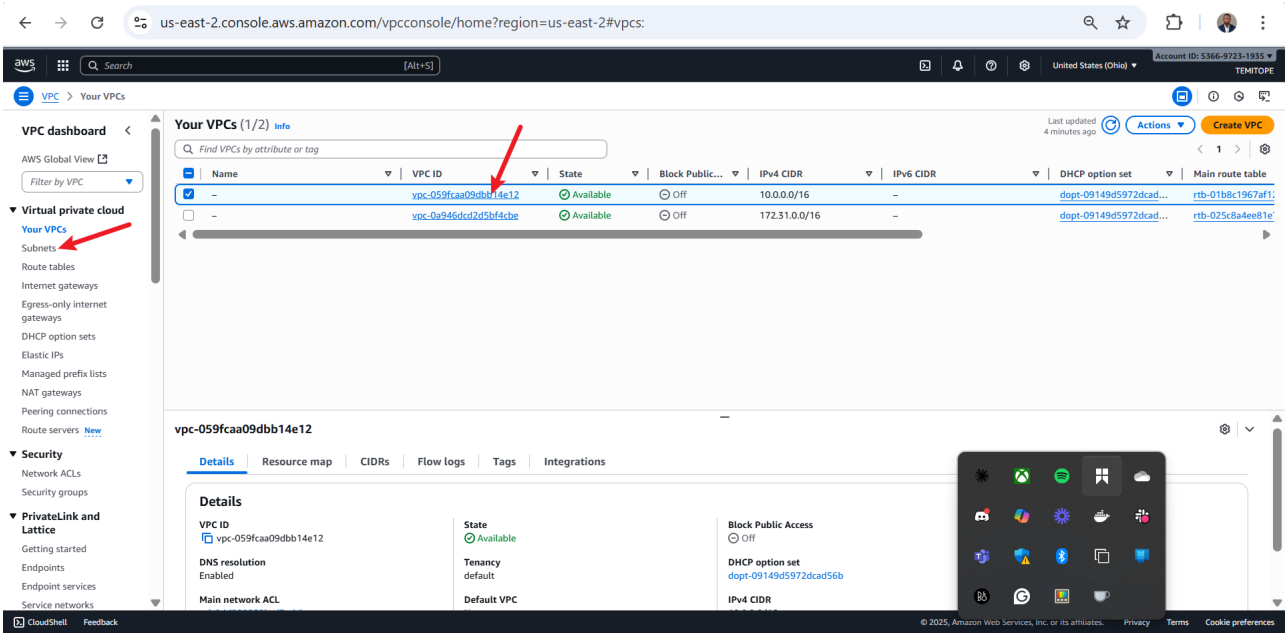
4. VPC successfully created.



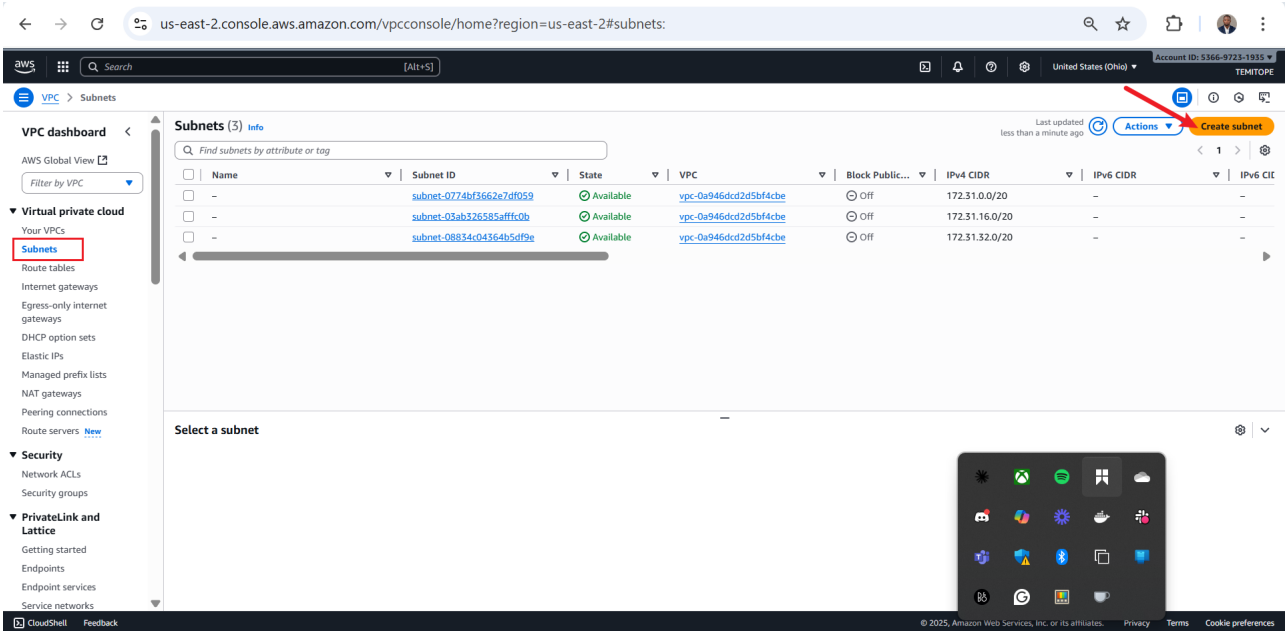
5. Note: In case of CIDR block size error, let your block size fall within the specified range.



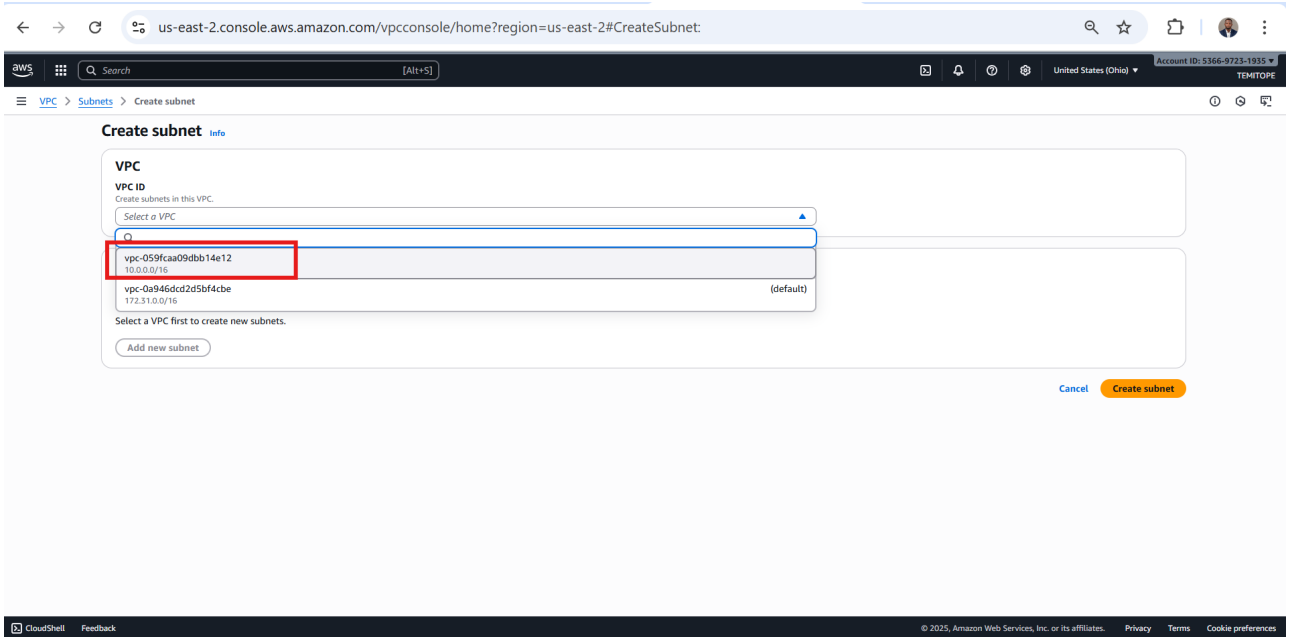
6. Here is the created VPC; click "Subnets" to create a subnet.



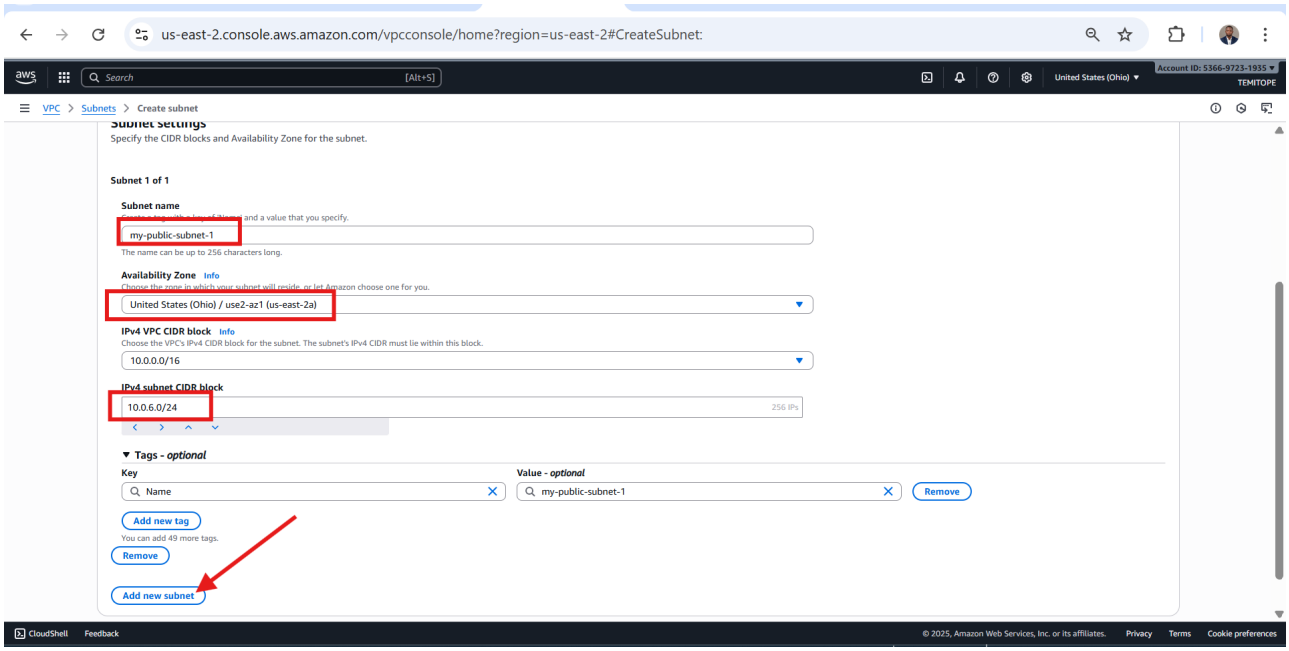
7. On the subnet dashboard, click "Create subnet".



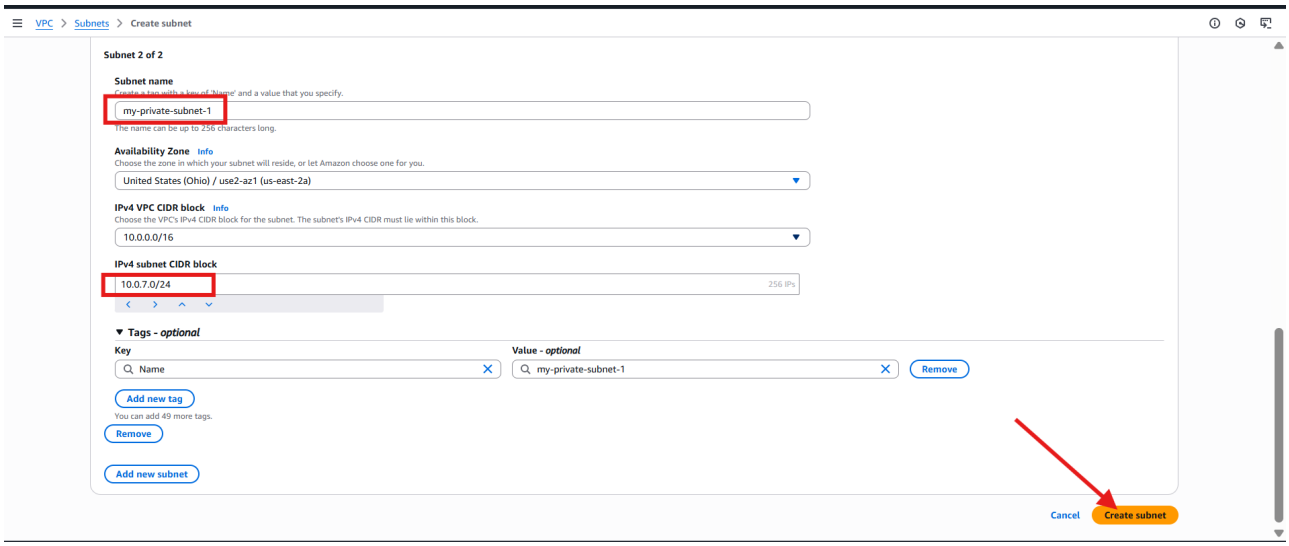
8. Select the Subnet VPC (created in the first step).



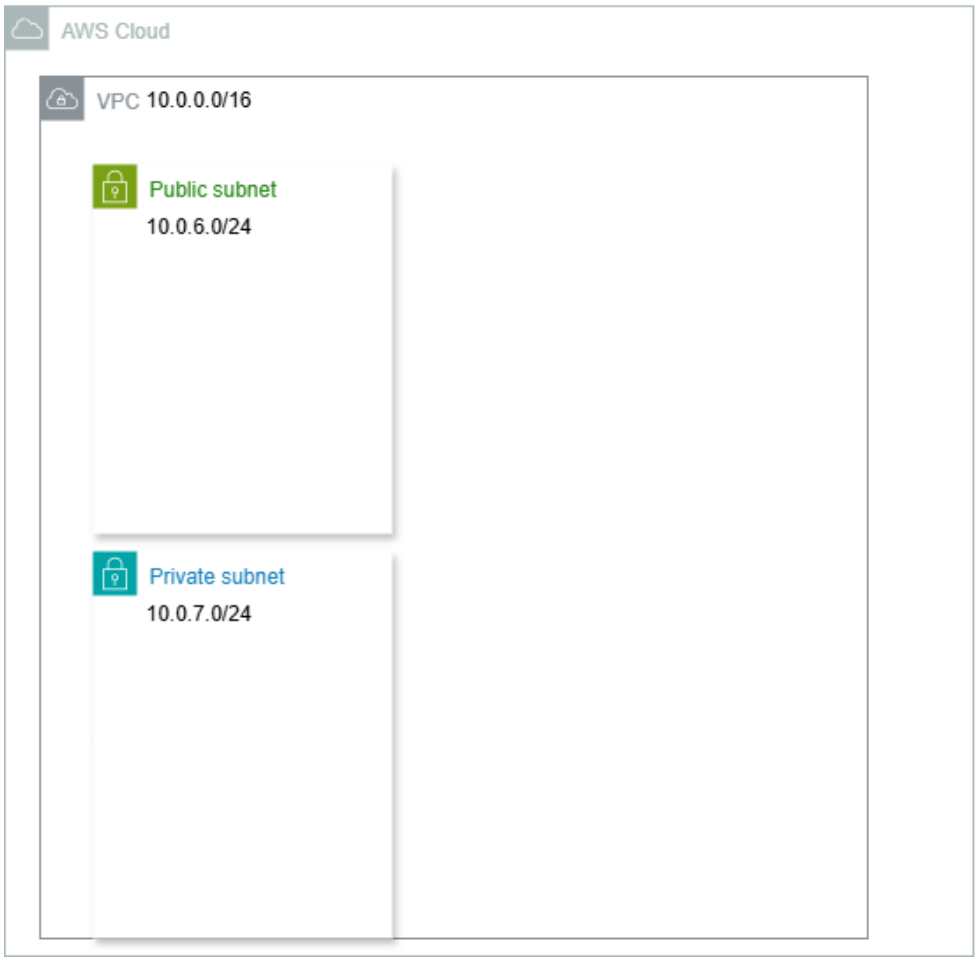
9. Enter subnet name (my-public-subnet-1), set availability zone, subnet IPv4 CIDR block, and click "Add new subnet".



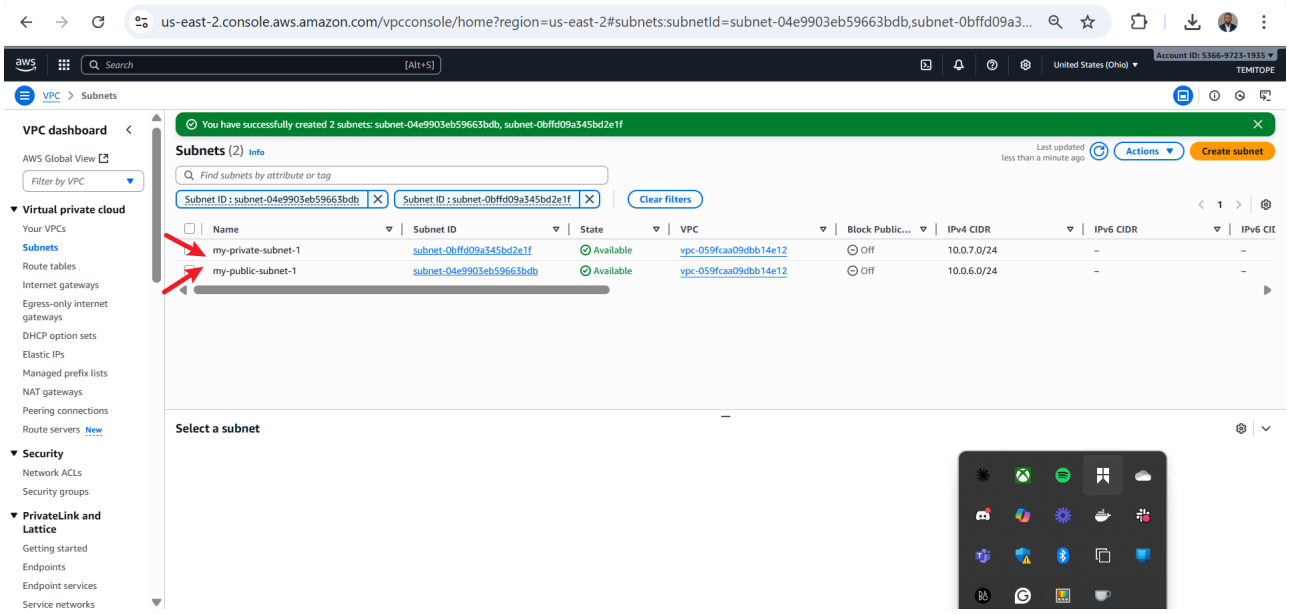
10. Enter subnet name (my-private-subnet-1), set availability zone, subnet IPv4 CIDR block, and click "Create subnet".



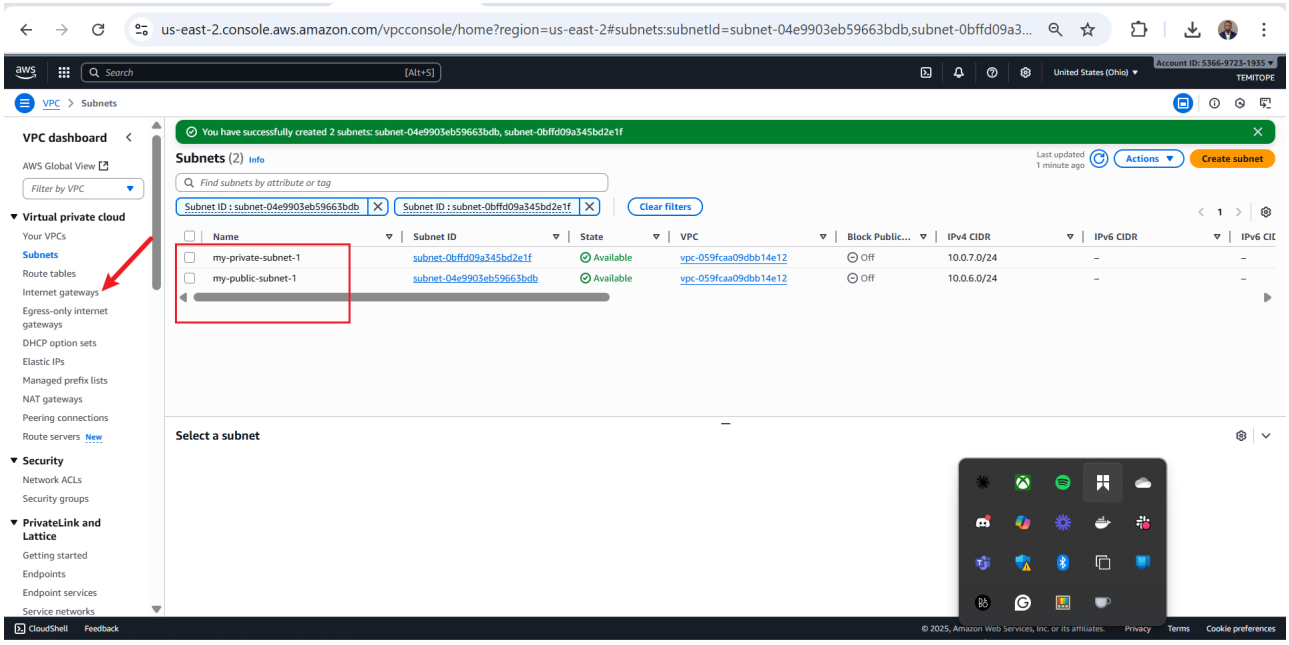
11. The architecture.



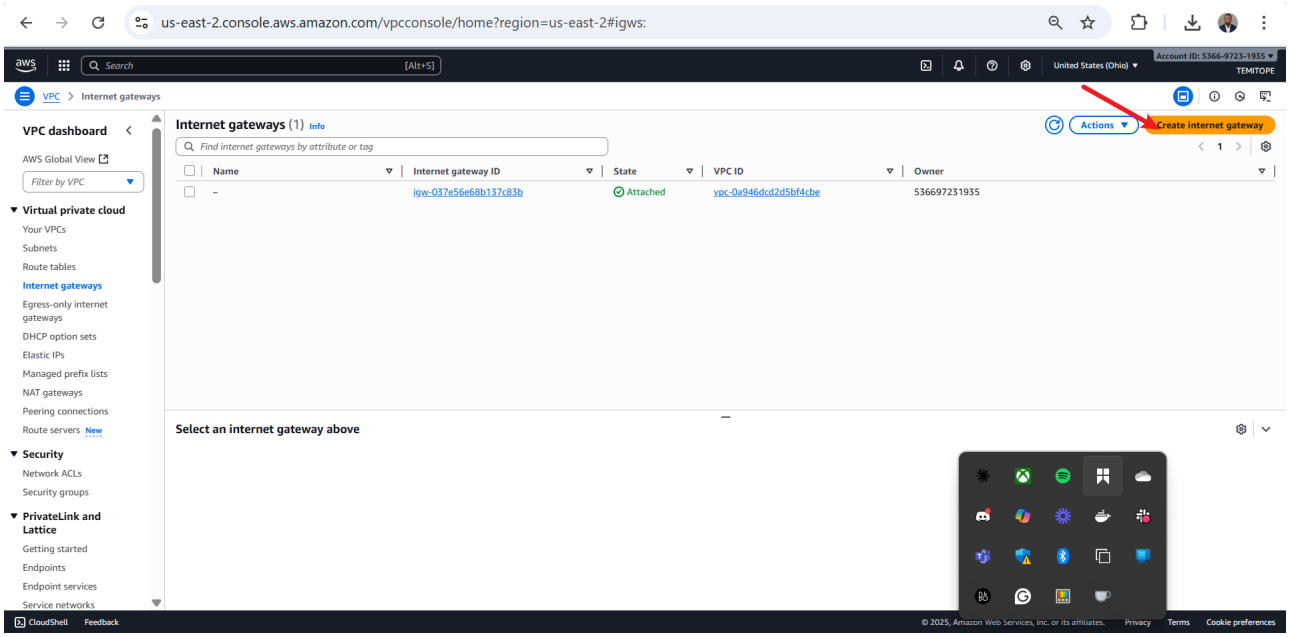
12. my-public-subnet-1 and my-private-subnet-1 successfully created.



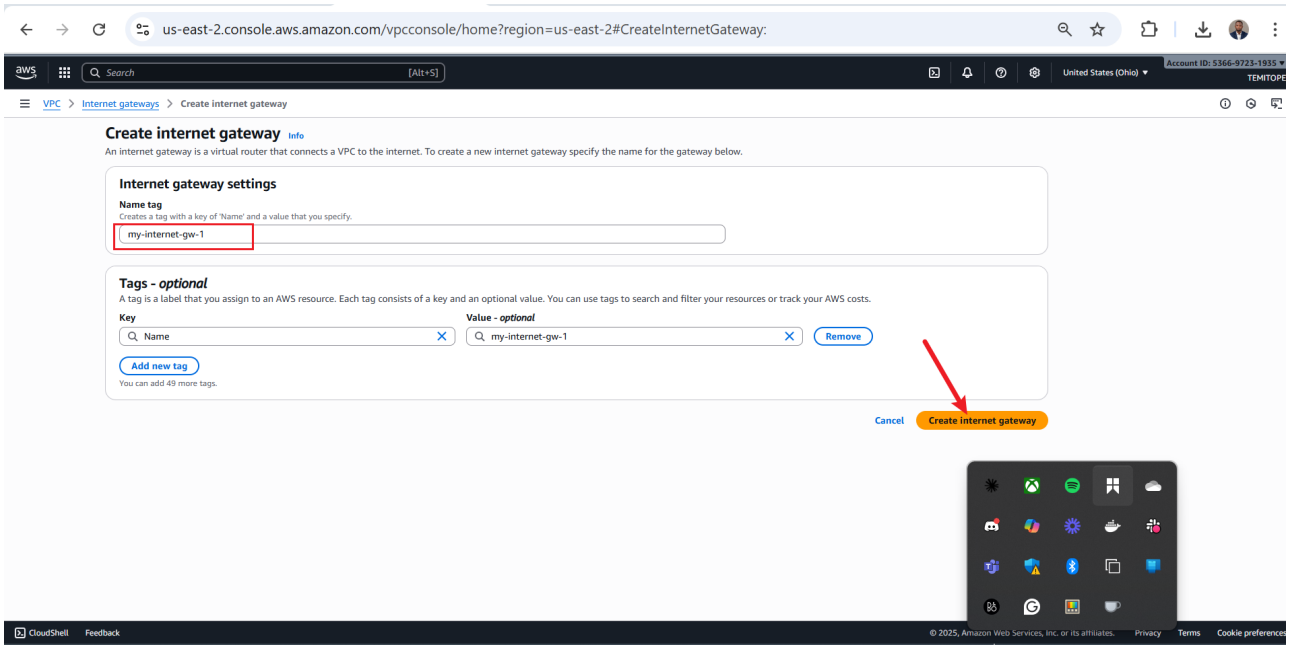
13. Since my-public-subnet-1 and my-private-subnet-1 successfully created, now click on "Internet gateways" to connect public subnet to the internet.



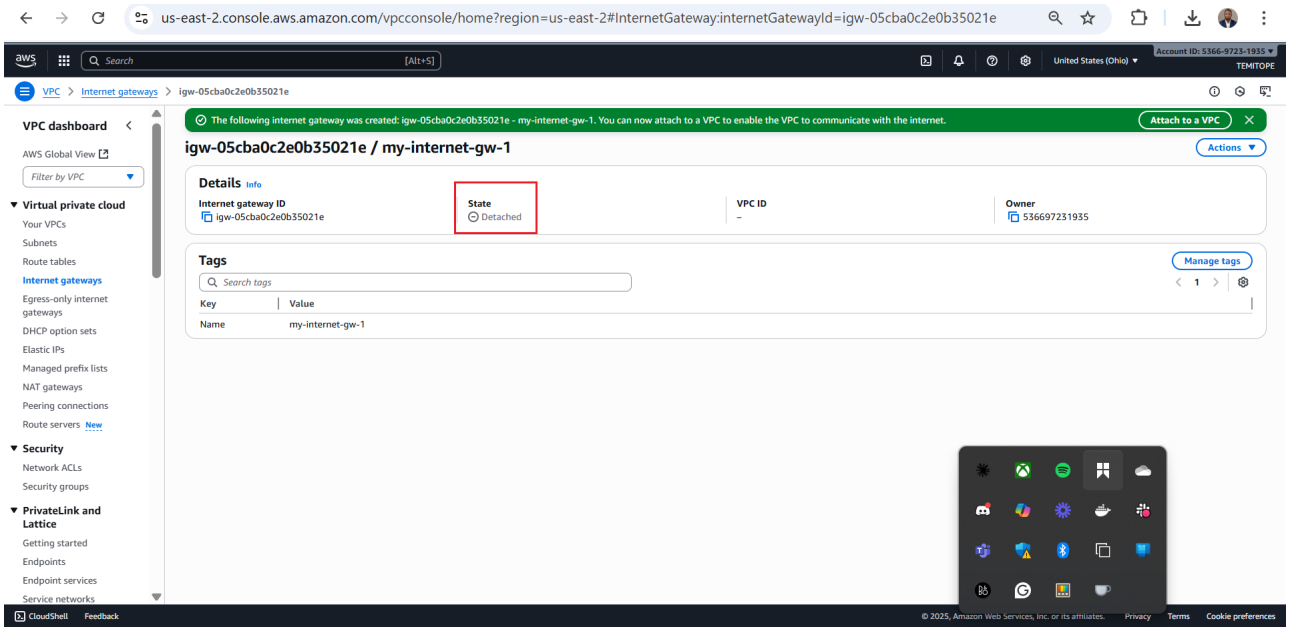
14. On the internet gateway dashboard, click "Create internet gateway" button.



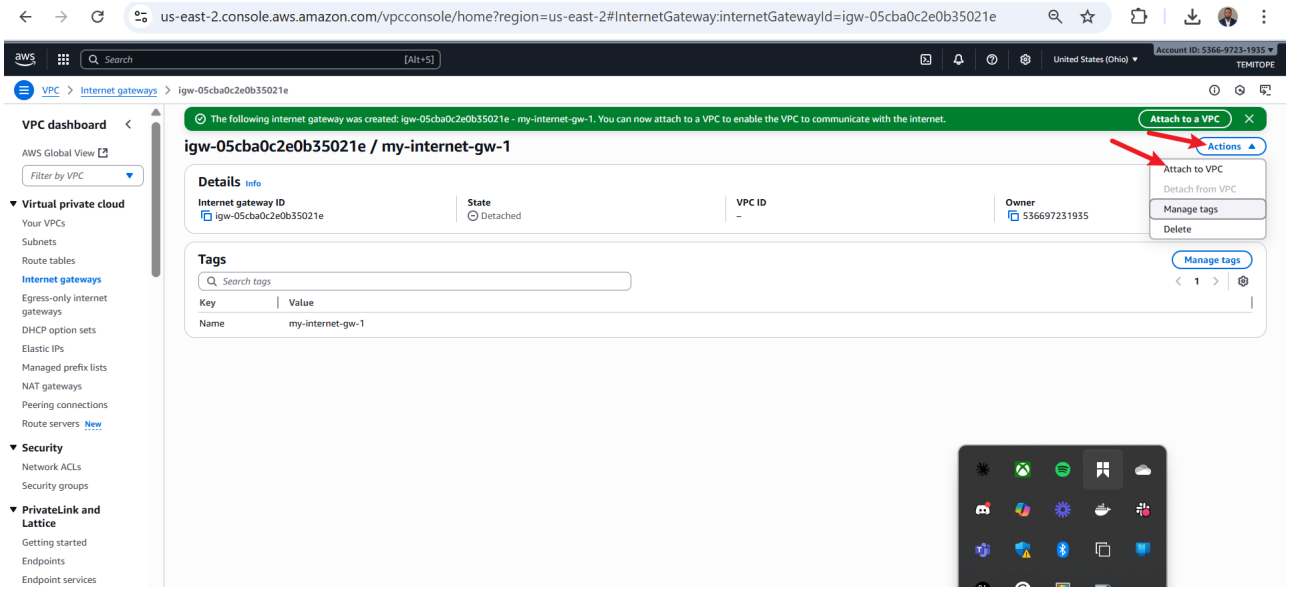
15. Name internet gateway and click "Create internet gateway".



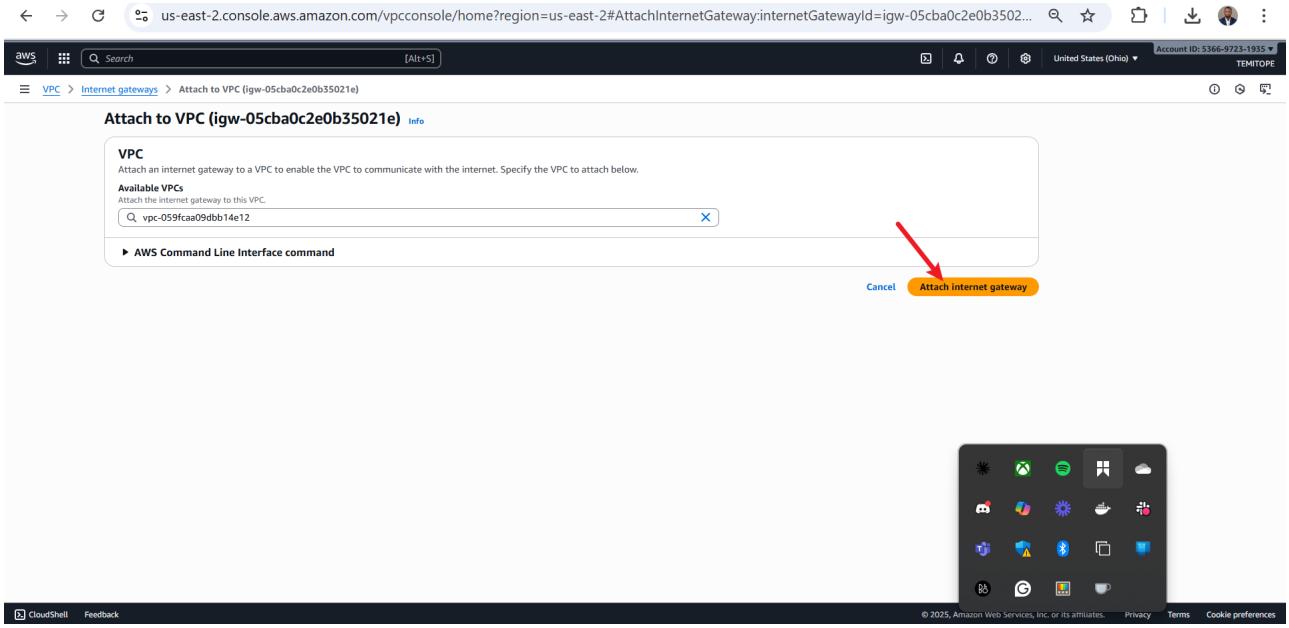
16. Internet gateway successfully created; note that the state shows detached.



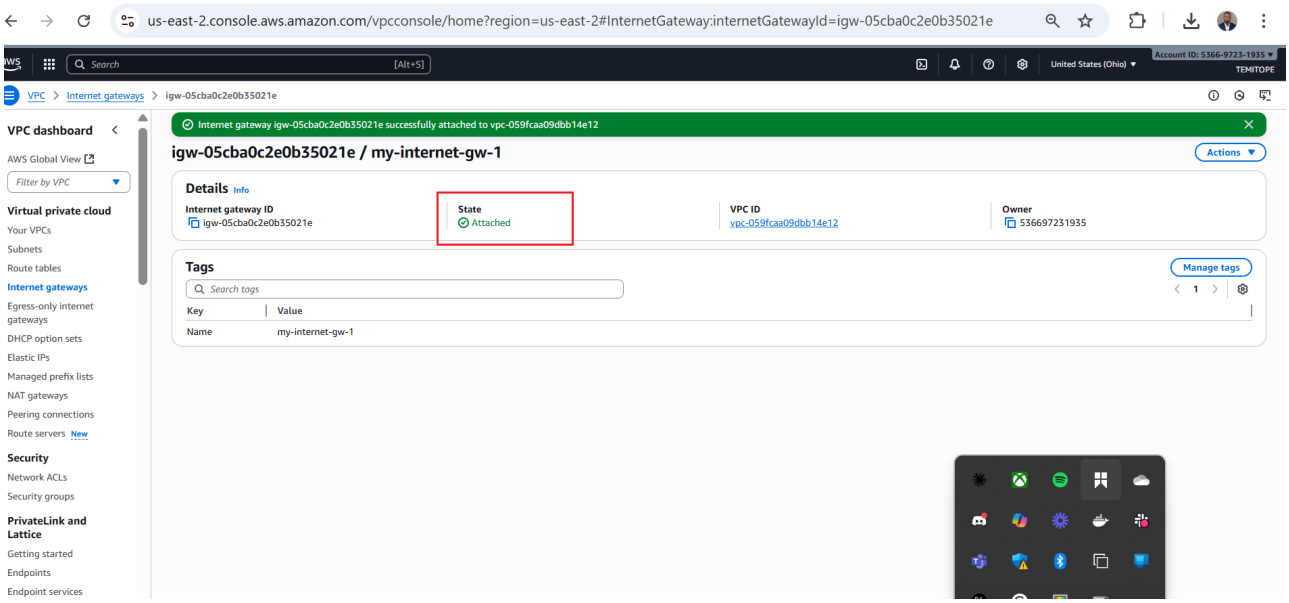
17. Click "Actions" and then "Attach to VPC".



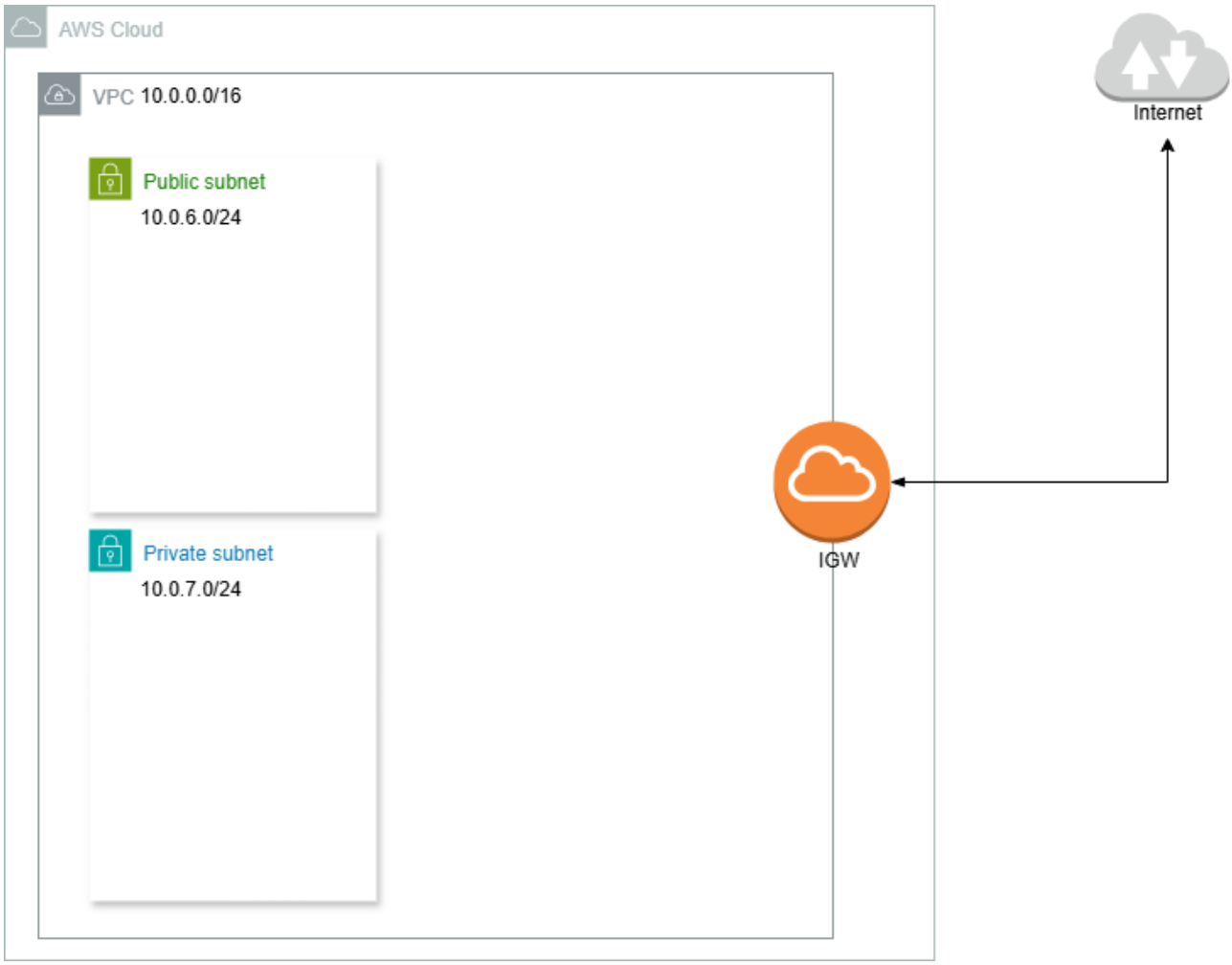
18. Select VPC and click "Attach internet gateway".



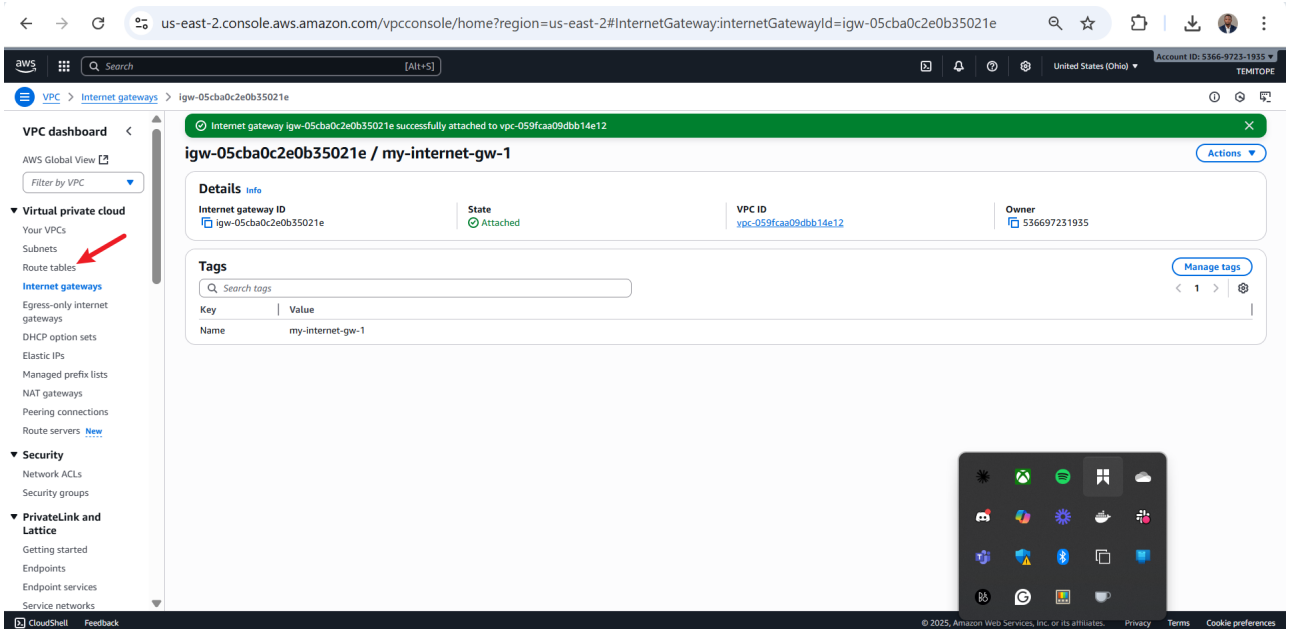
19. Internet gateway successfully attached to VPC.



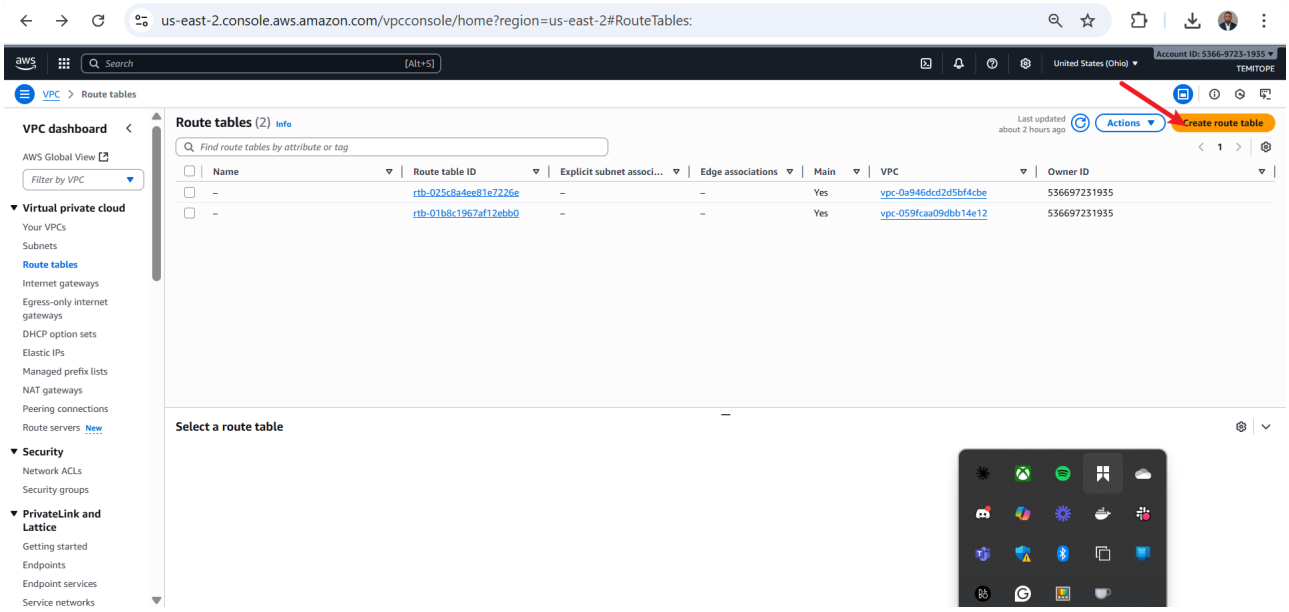
20. Present architecture.



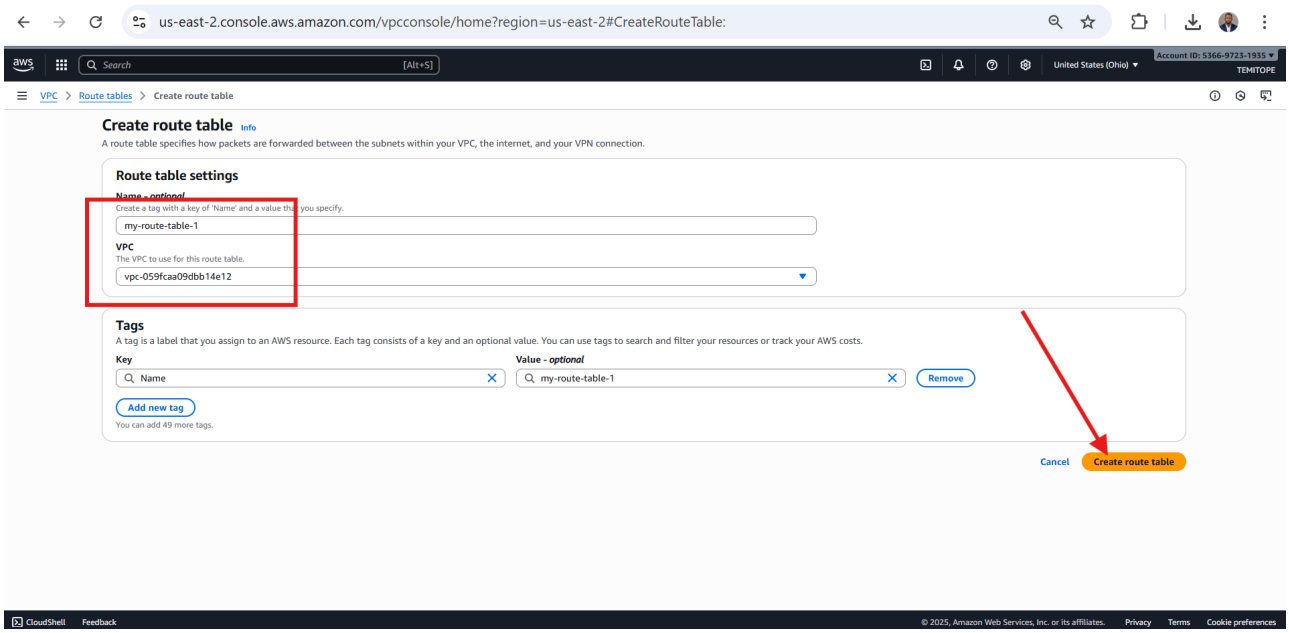
21. Since IGW is now attached to VPC, click on "Route tables".



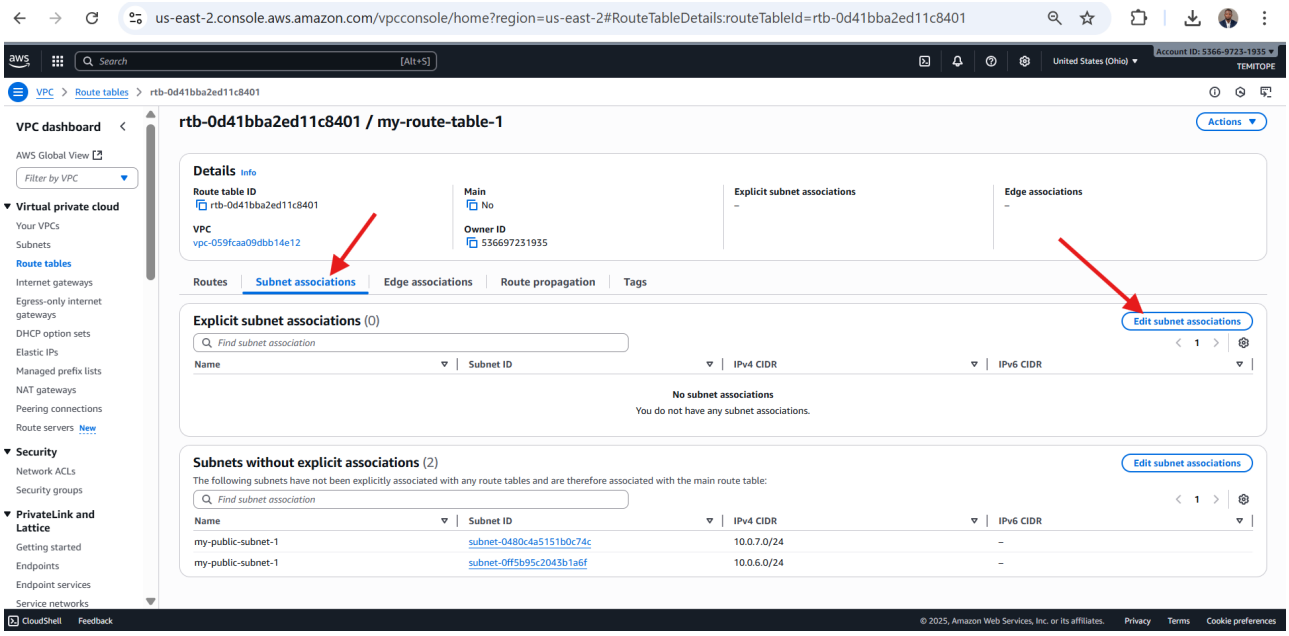
22. On the route table dashboard, click "Create route table".



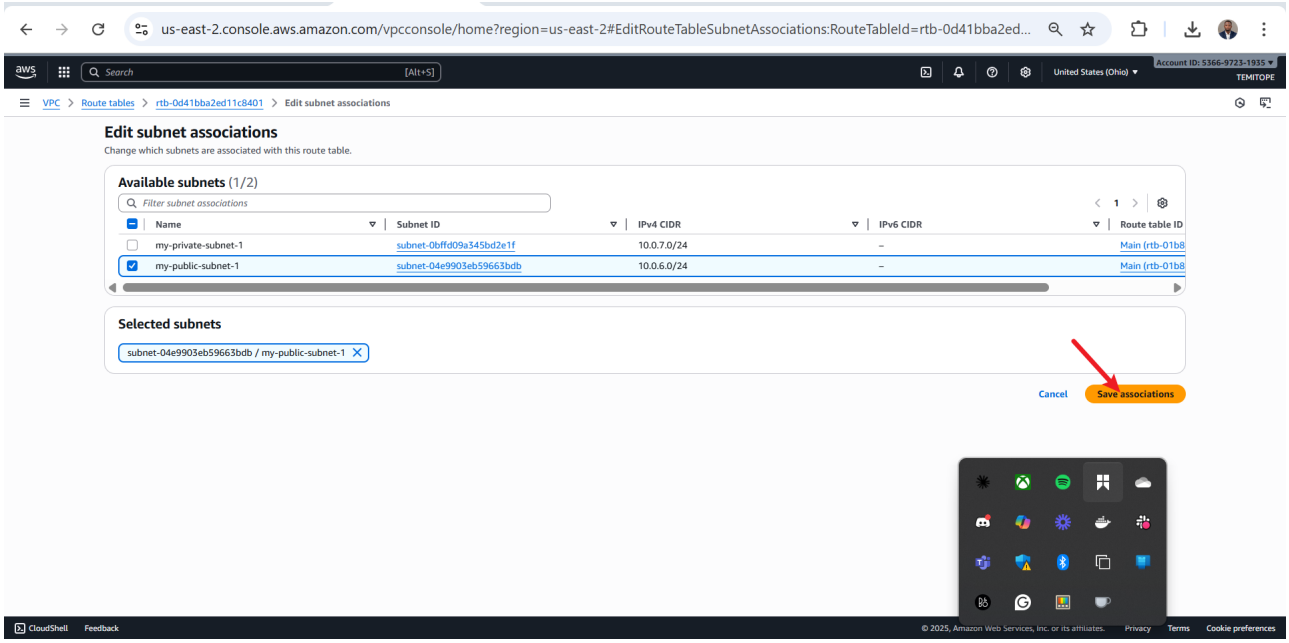
23. Name route table, select VPC, and click "Create route table".



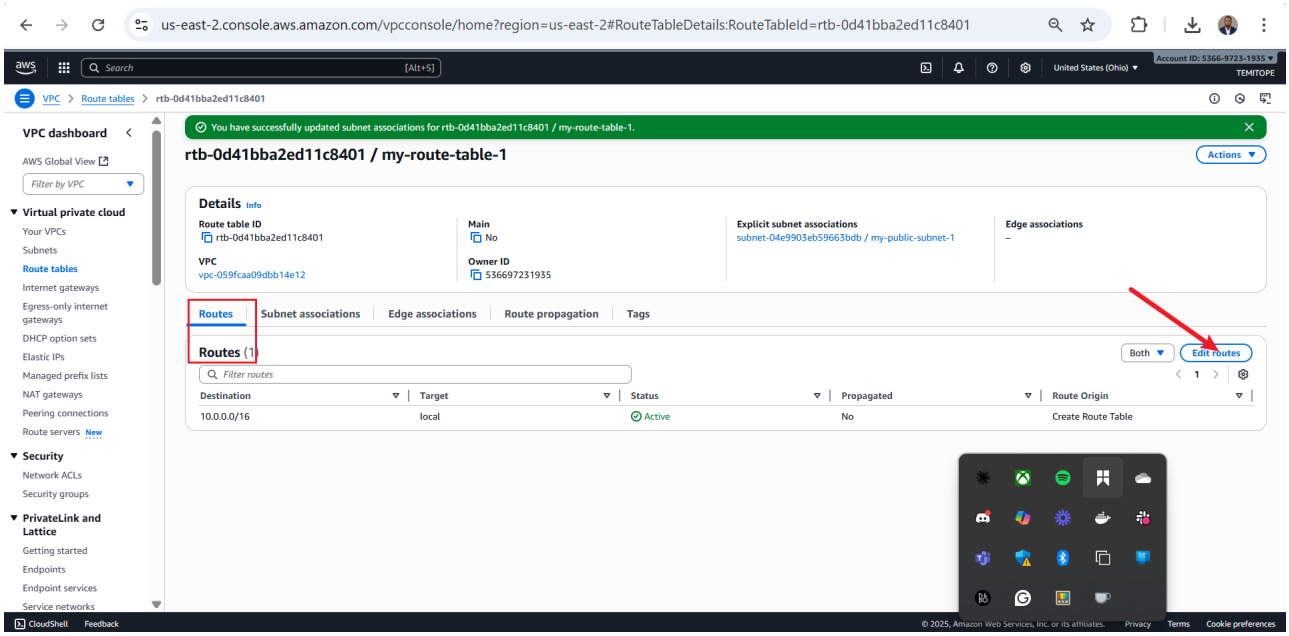
24. Route table created; now click on "Subnet associations" then "Edit subnet association".



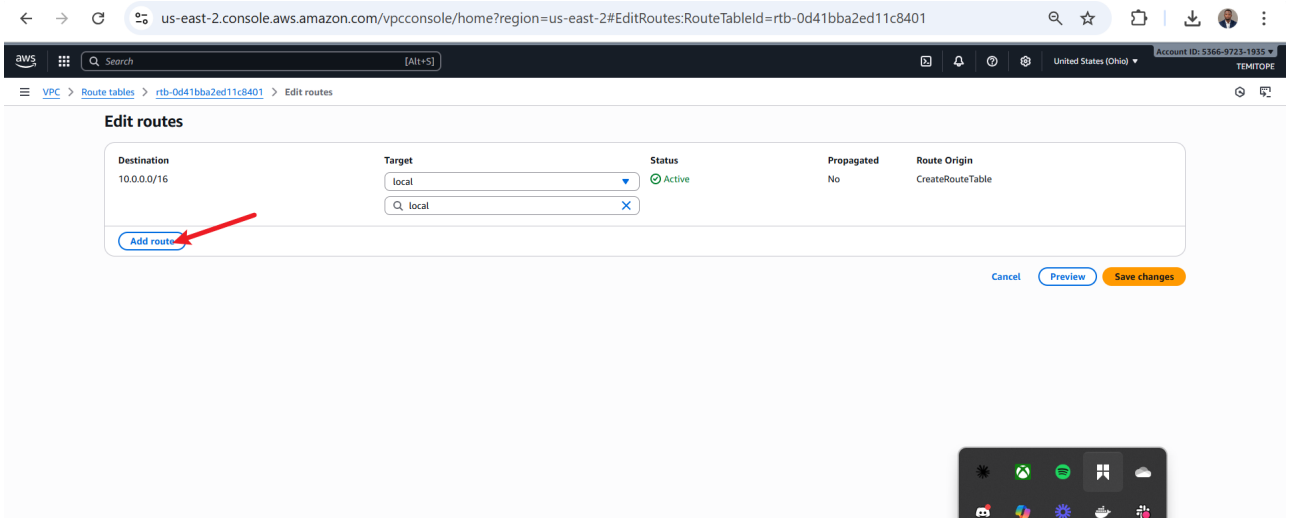
25. Select public subnet and click "Save associations".



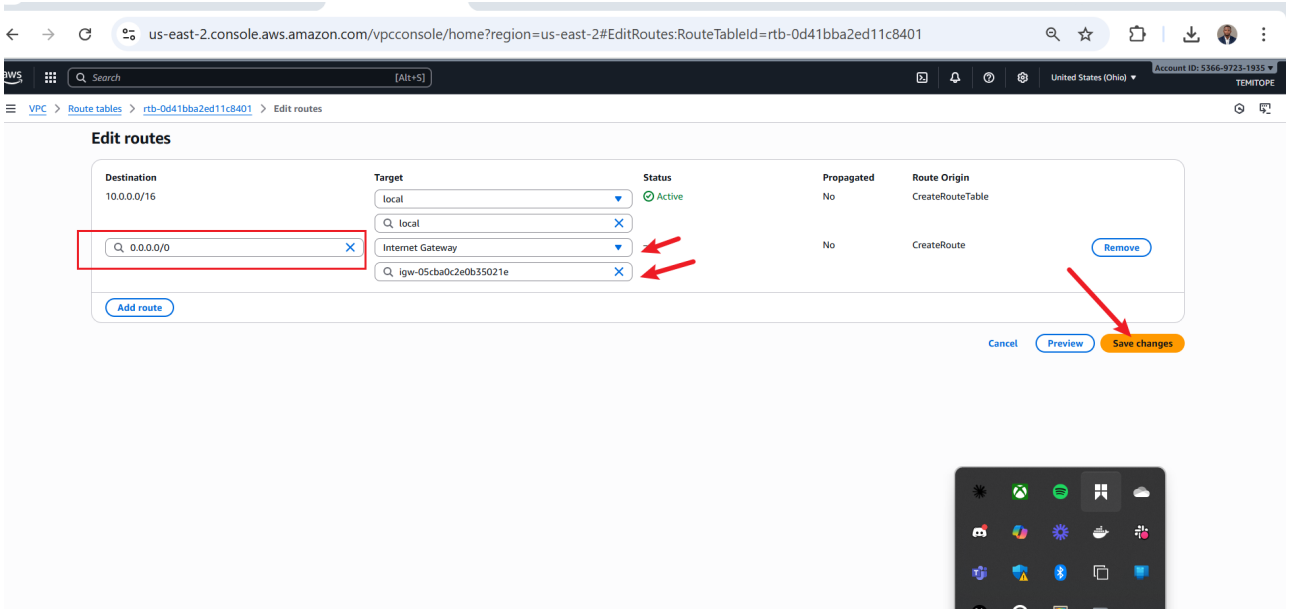
26. Subnet association successfully saved; now under "Routes" tab, click "Edit routes".



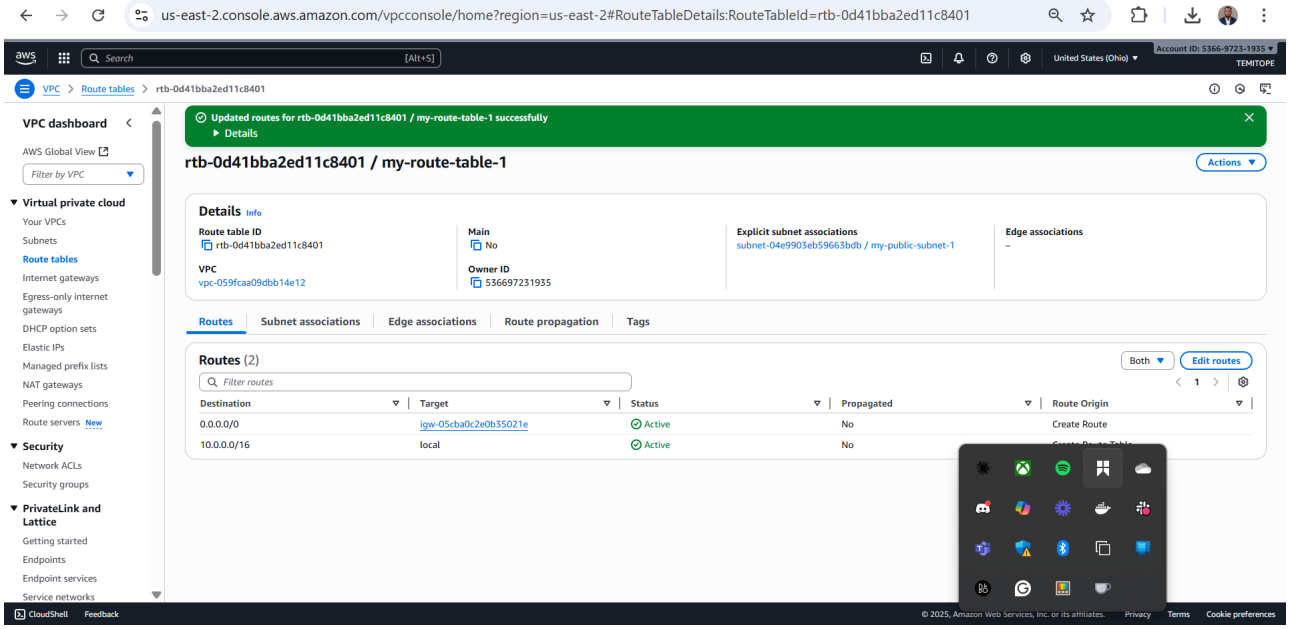
27. Click "Add route" to add new route.



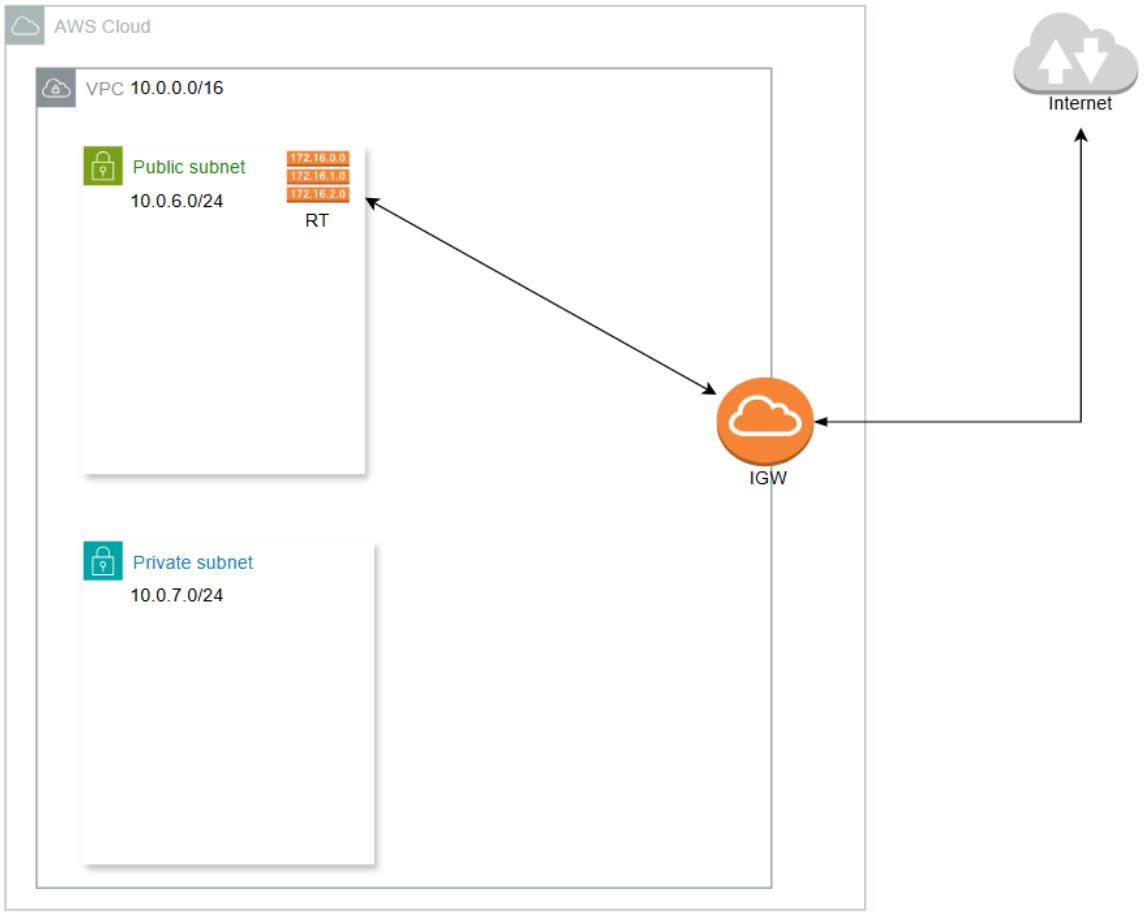
28. Set route destination to 0.0.0.0/0, target to the created IGW, and save changes.



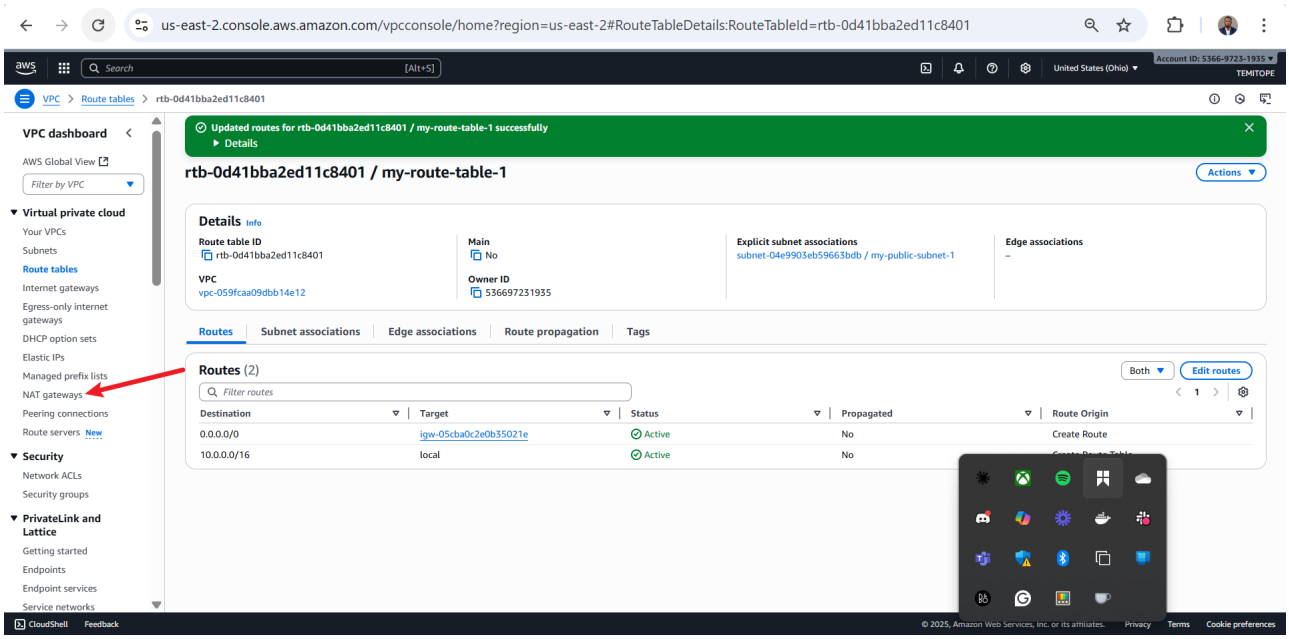
29. Route table updated successfully.



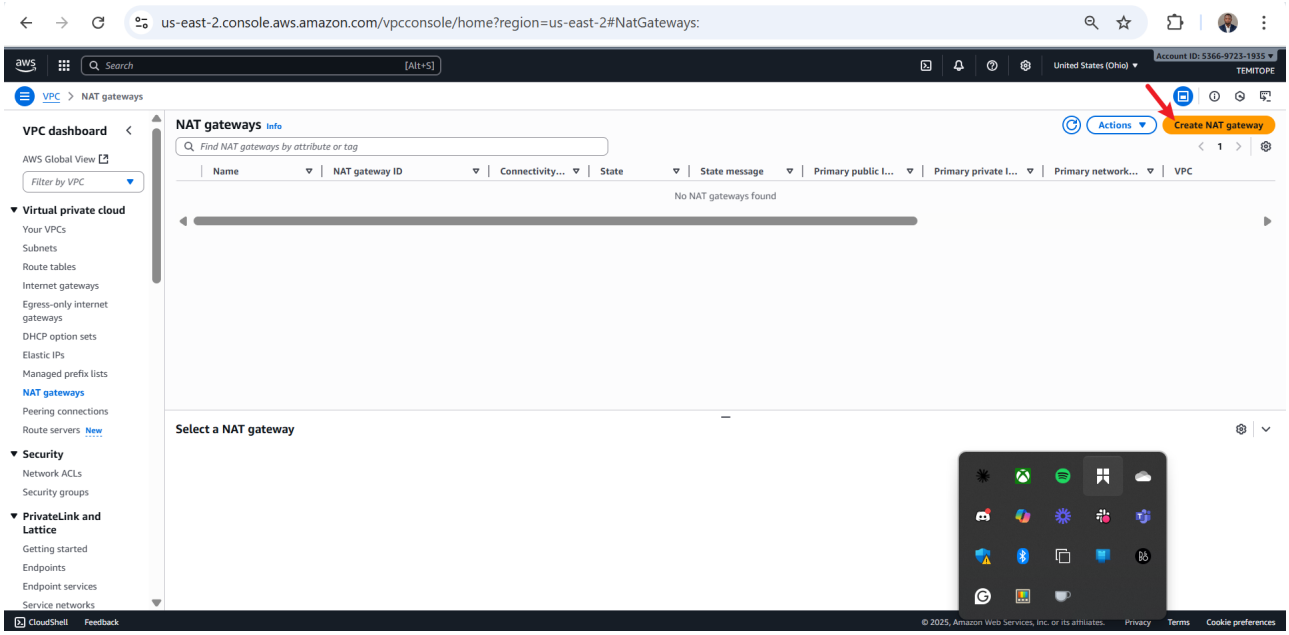
30. Present VPC or network architecture.



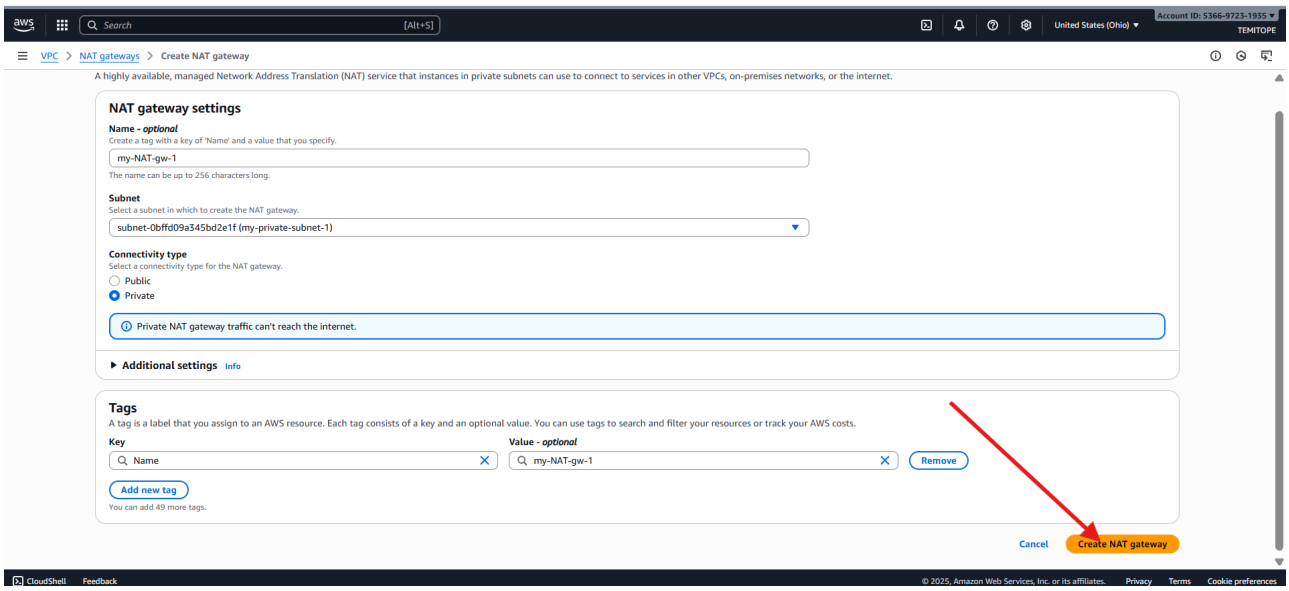
31. Now that route tables have been updated, click on "NAT gateways".



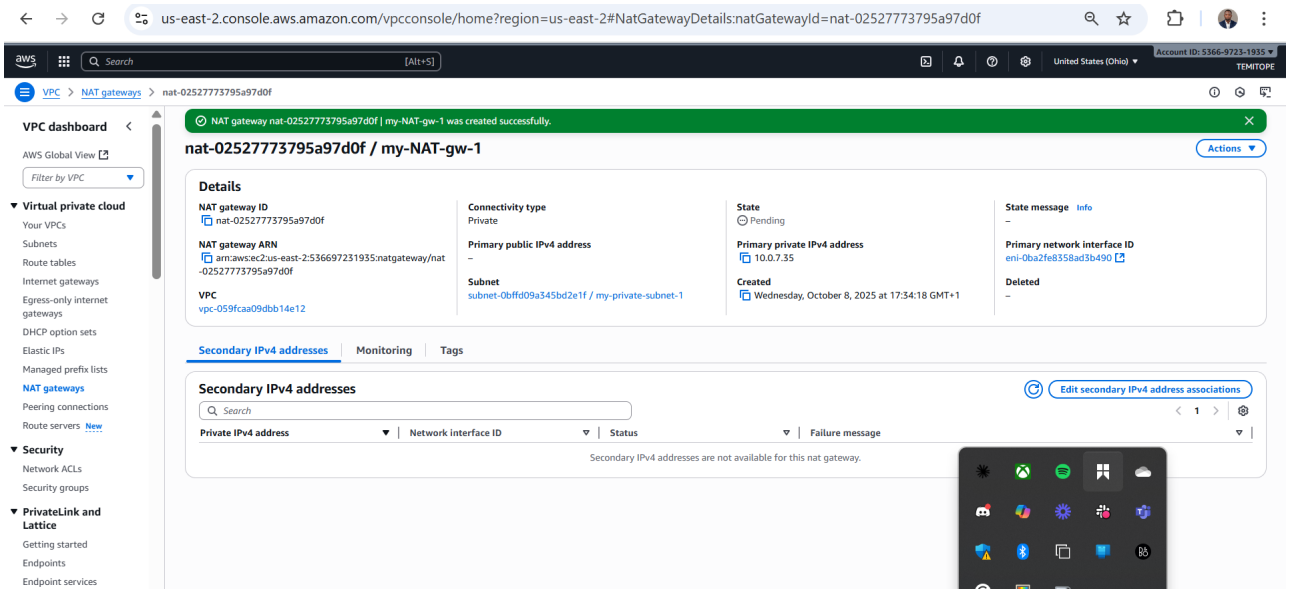
32. Click "Create NAT gateway".



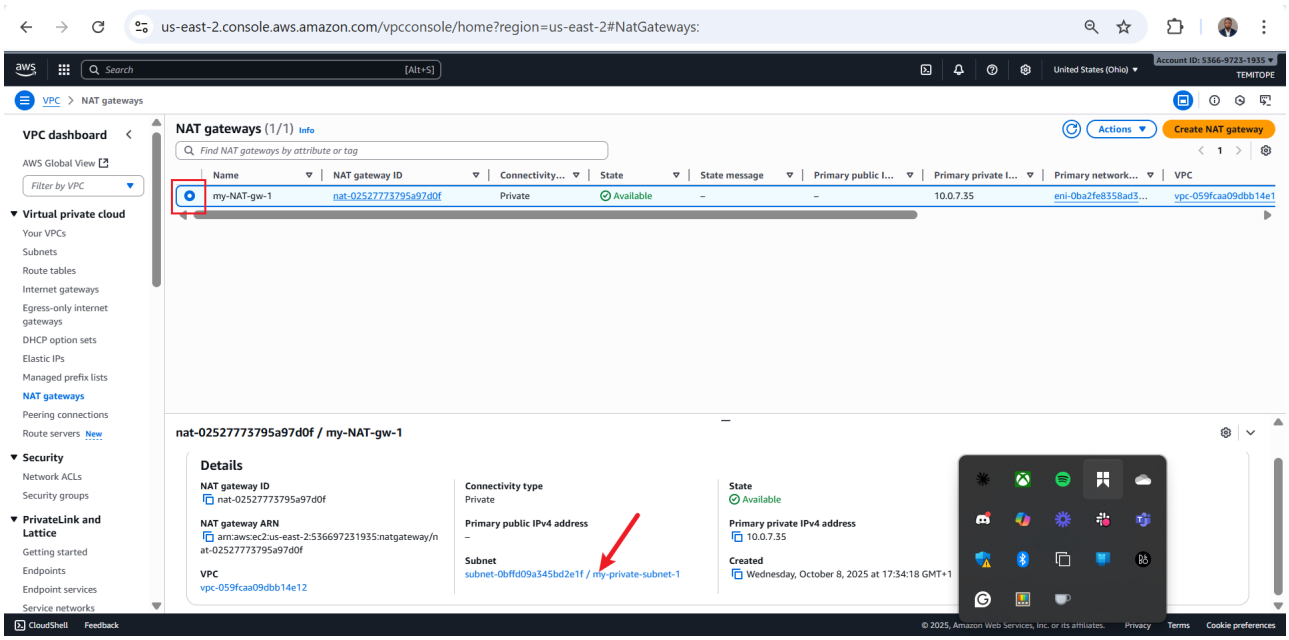
33. Name NATgw, select private subnet, and set connectivity type to private.



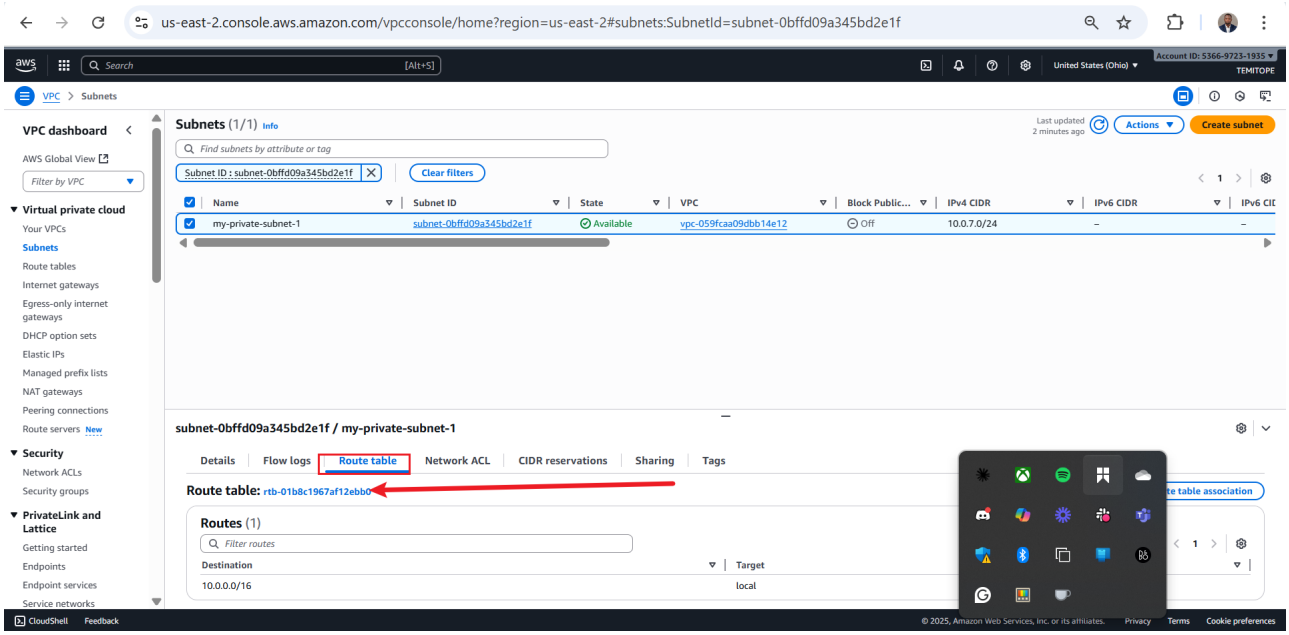
34. NAT gateway created successfully.



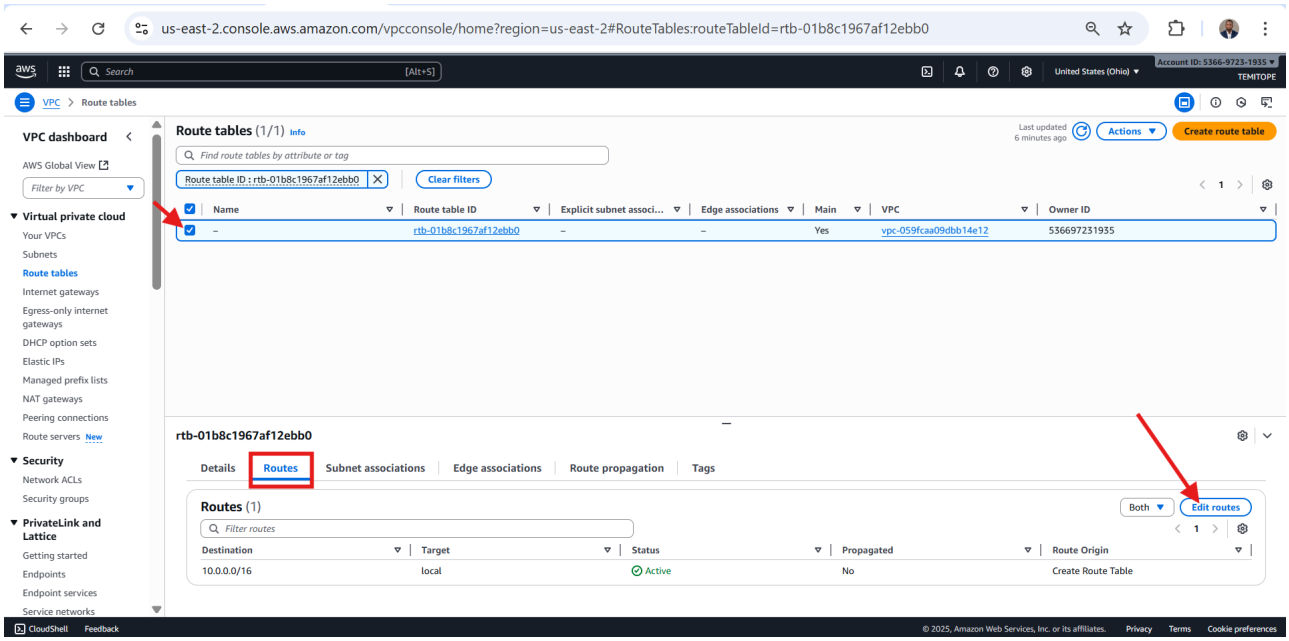
35. Select the created NAT gateway and locate subnet ID link at details section and click on it.



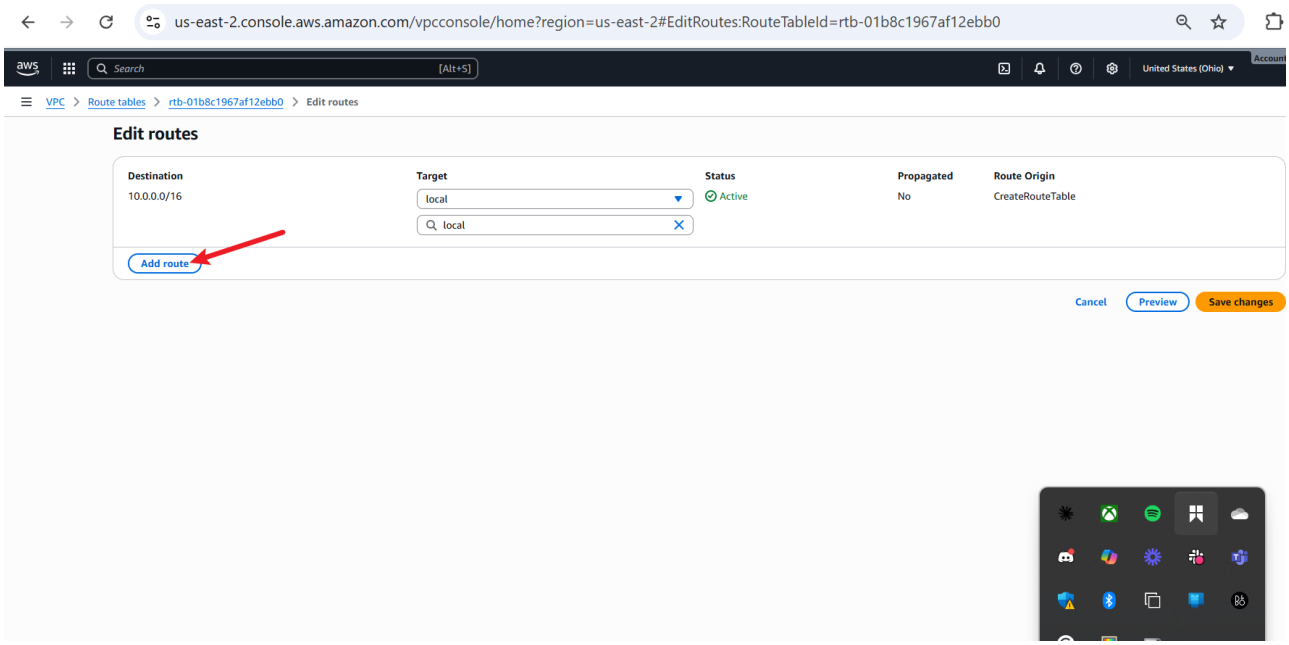
36. In the subnet page, navigate to "Route Table" section and click one route table ID (rtb-...).



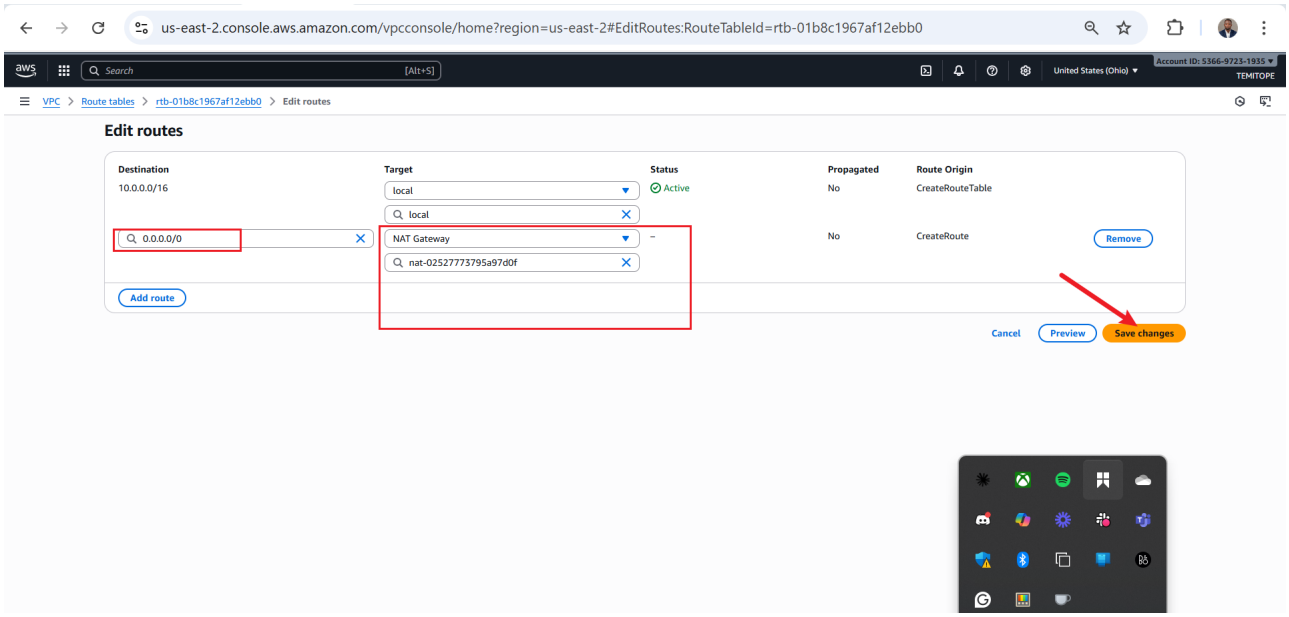
37. On route table page, click on "Routes" tab and then "Edit routes".



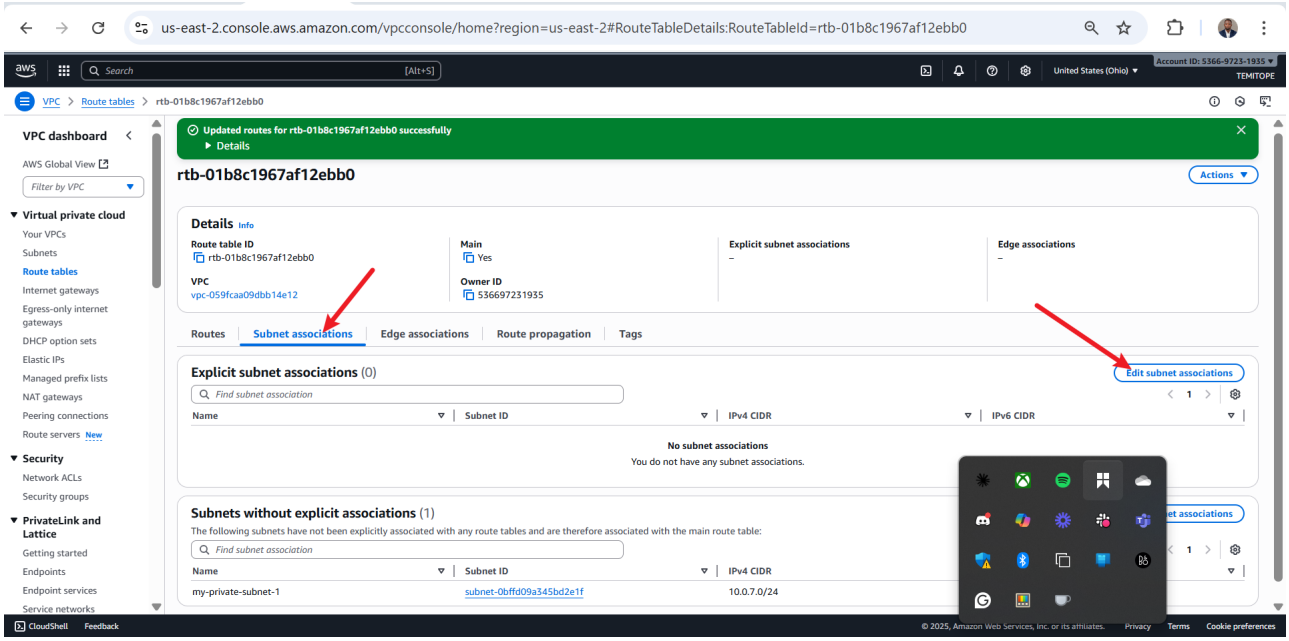
38. On the edit route page, click "Add route".



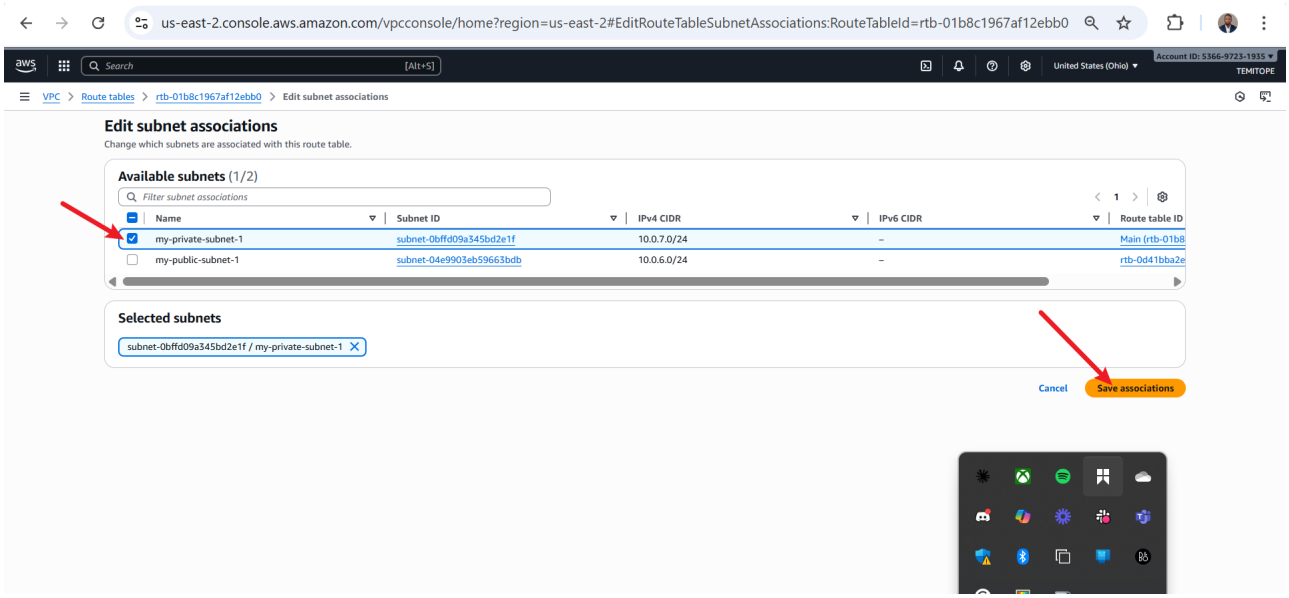
39. Select 0.0.0.0/0 as destination, NAT Gateway as the target, and the created NAT gateway as the NAT Gateway, then save changes.



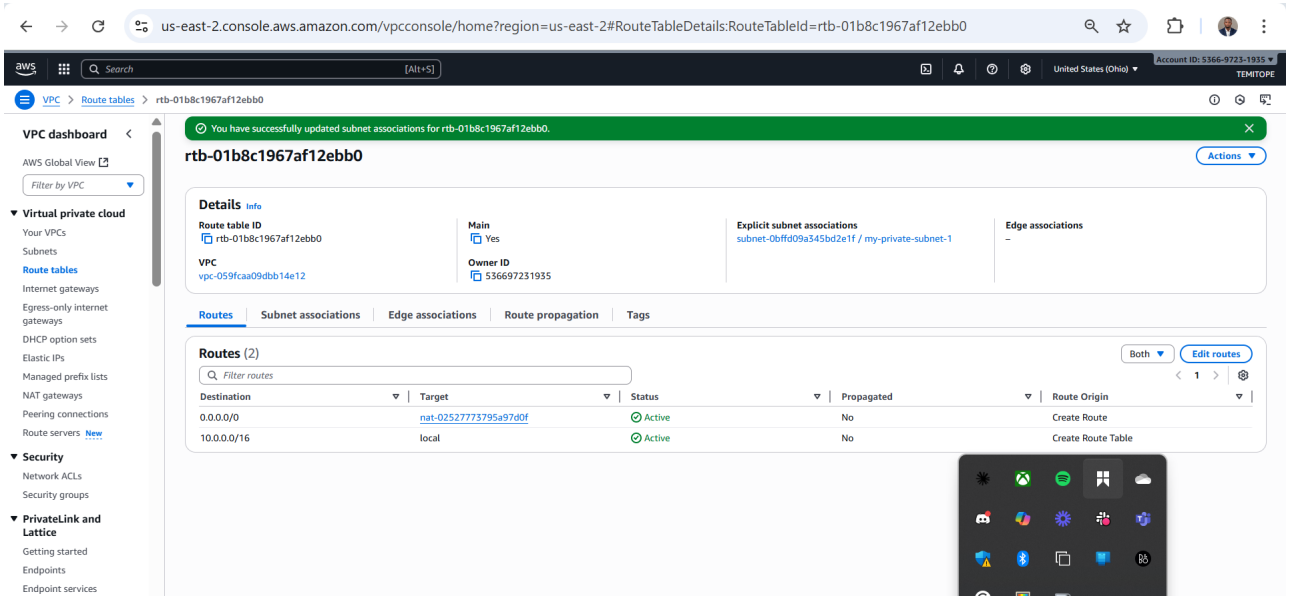
40. Route table updated successfully; now click "Subnet association" and "Edit subnet association".



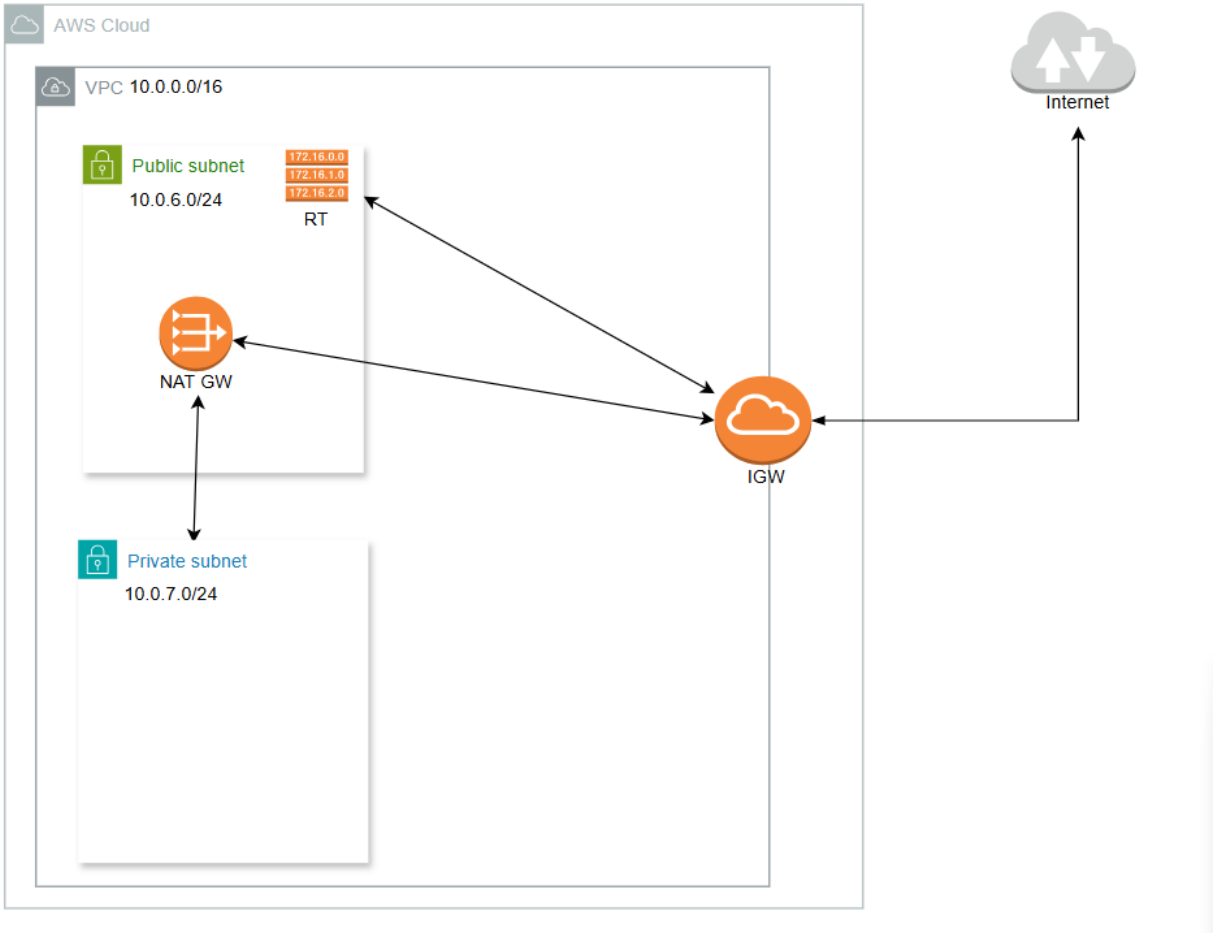
41. Click on the private subnet and then "Save associations" button.



42. The subnet has been successfully attached with the route table.



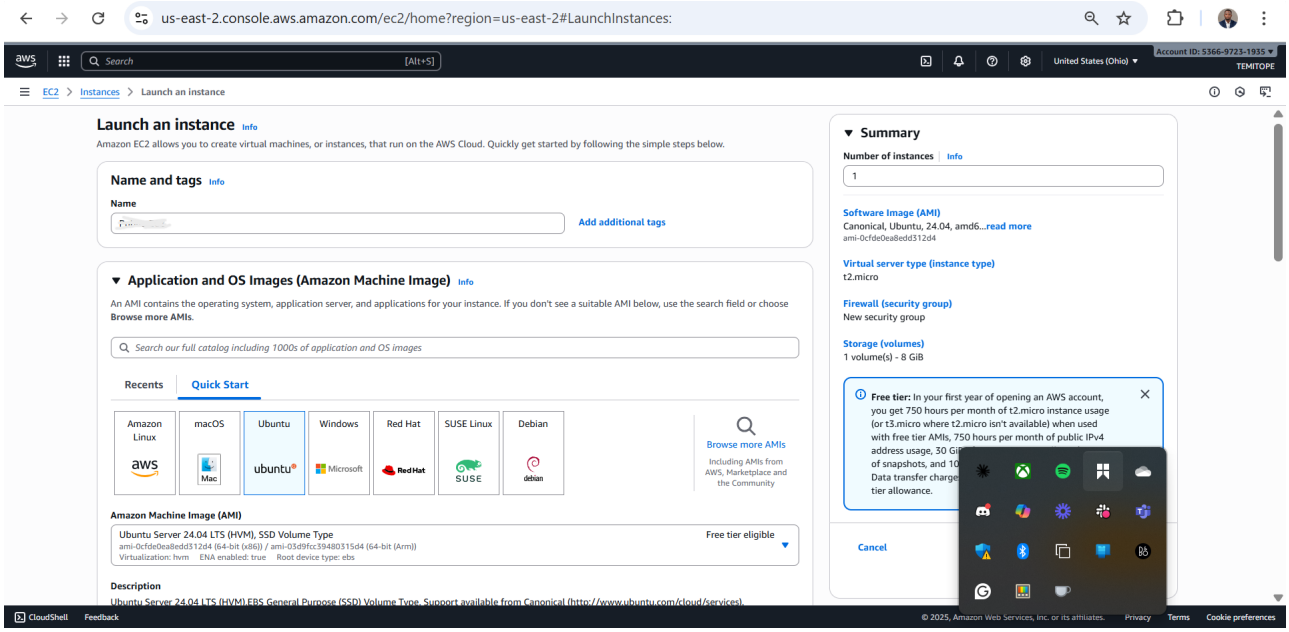
43. Present VPC architecture or status.



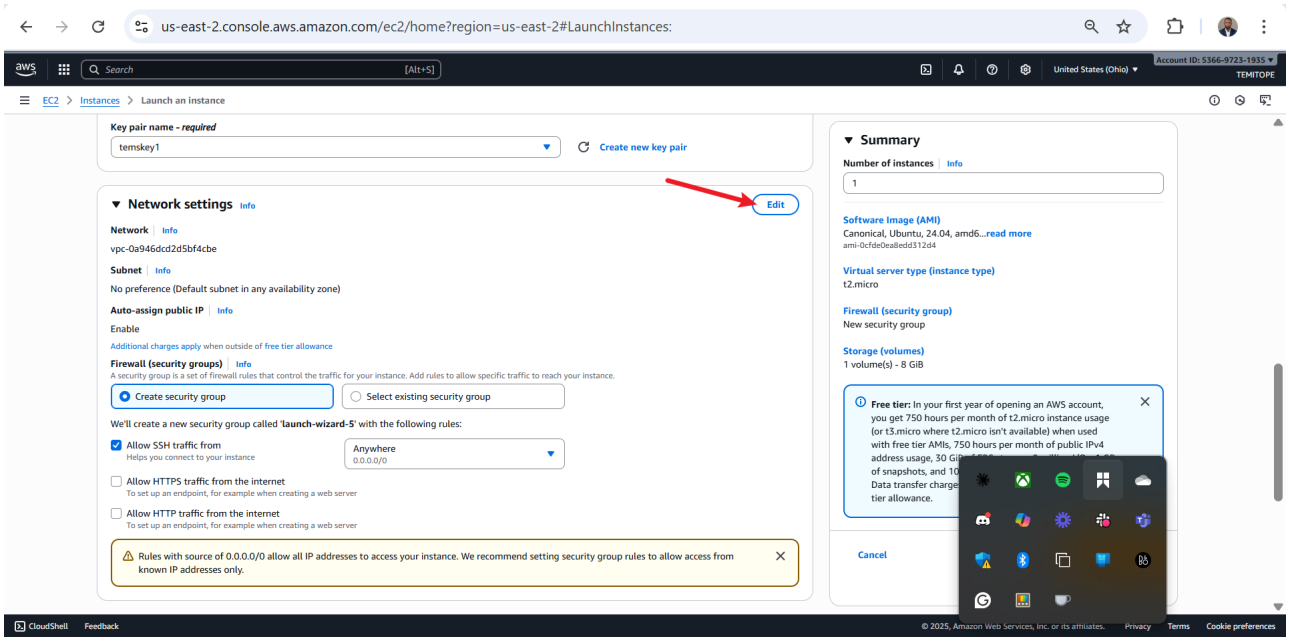
Part 1: Security Groups

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

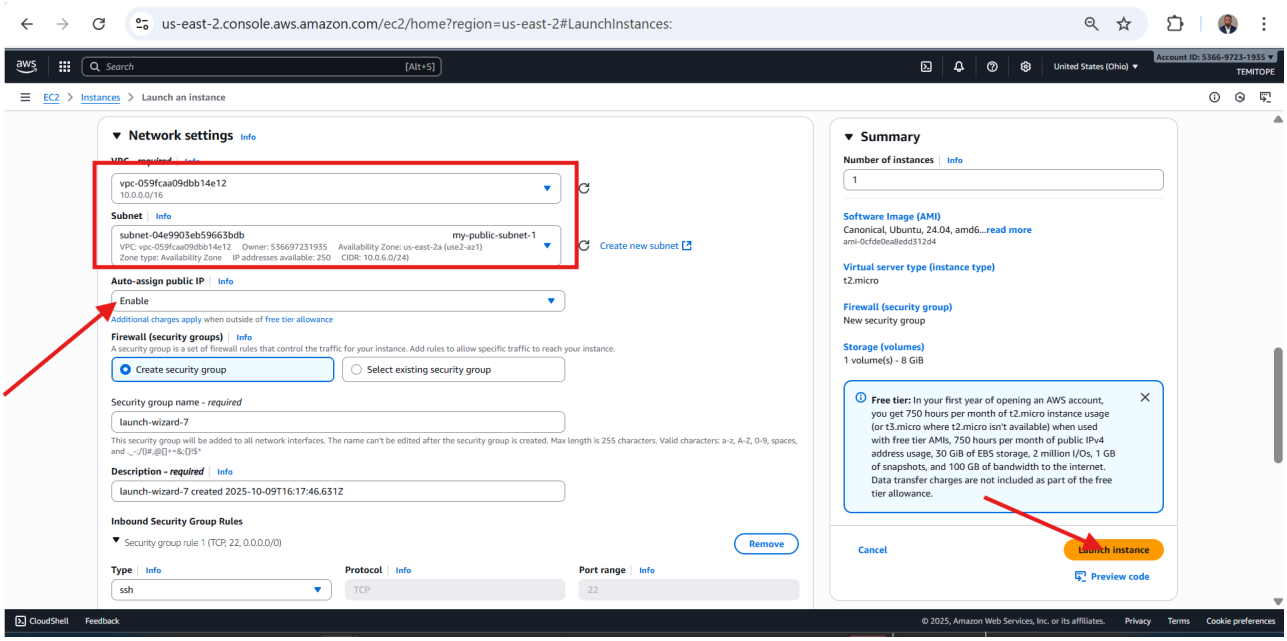
44. Create an Ubuntu EC2 instance and name it "my-test-instance" and scroll down.



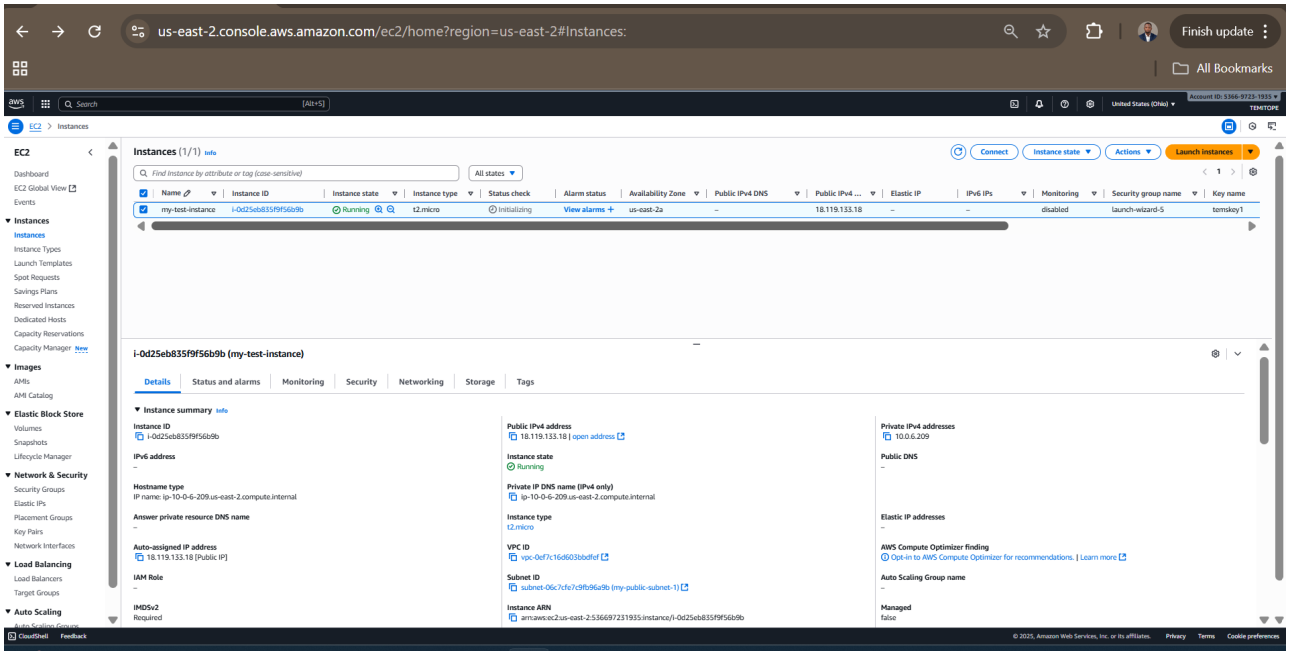
45. After setting the key pair, on the network settings click on "Edit".



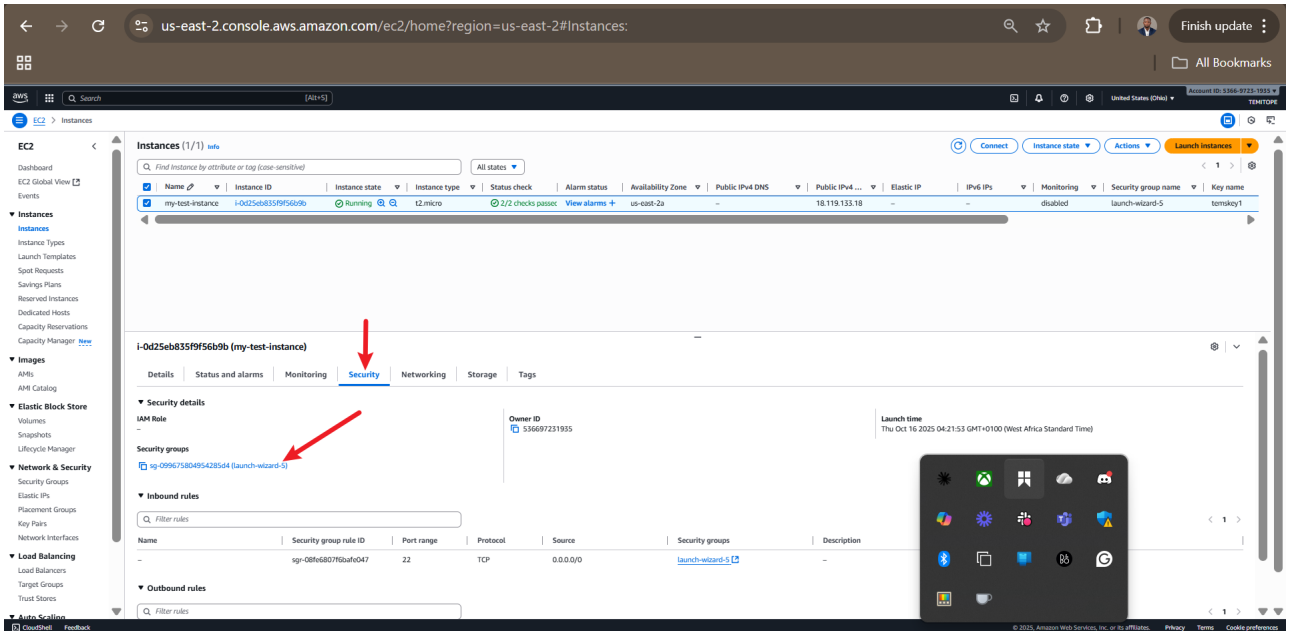
46. Set its VPC to the created VPC (10.0.0.0/16) and subnet to the public subnet. Enable Auto-assign public IP, then launch instance.



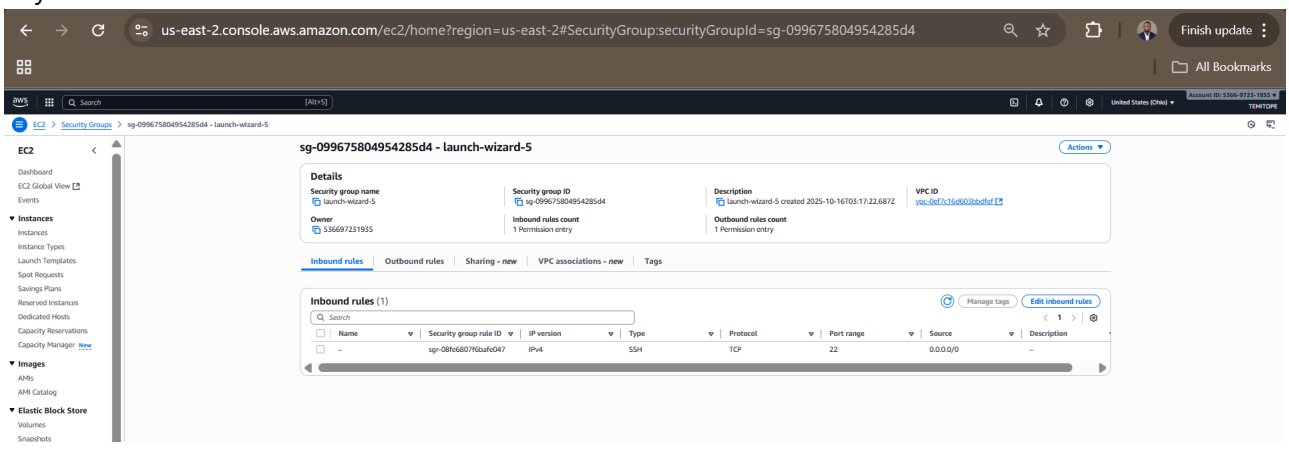
47. We have created a public EC2 in the public subnet which hosts our website.



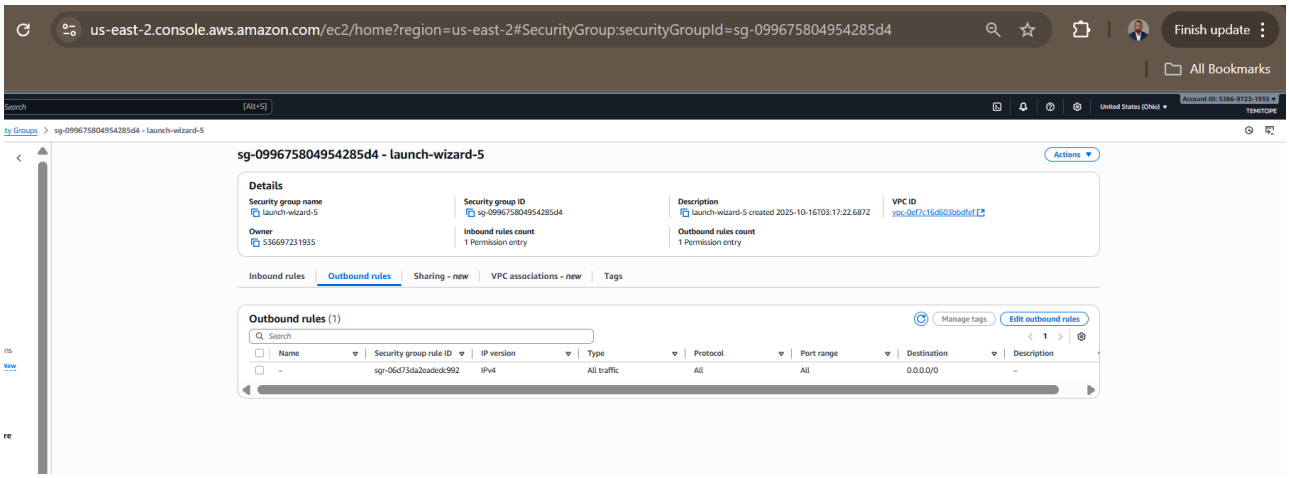
48. To see the instance security group rule, click on "Security" tab and the security groups link.



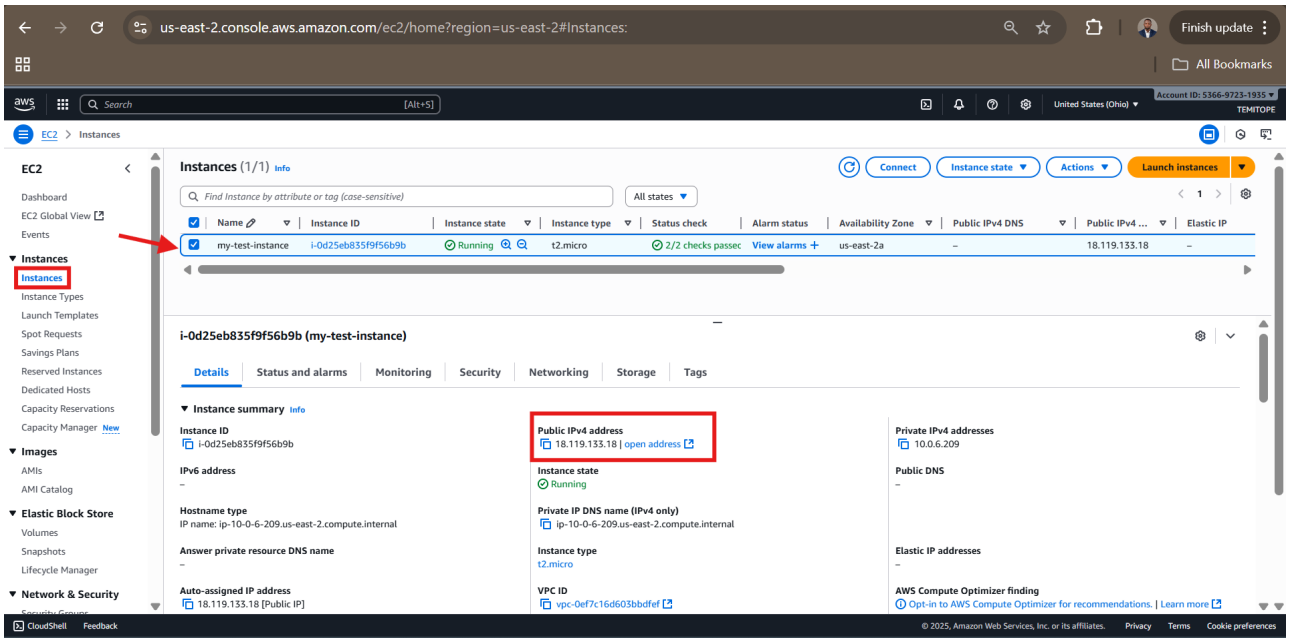
49. Note that the security group inbound rule allows only IPv4 SSH connection using TCP port 22 from anywhere.



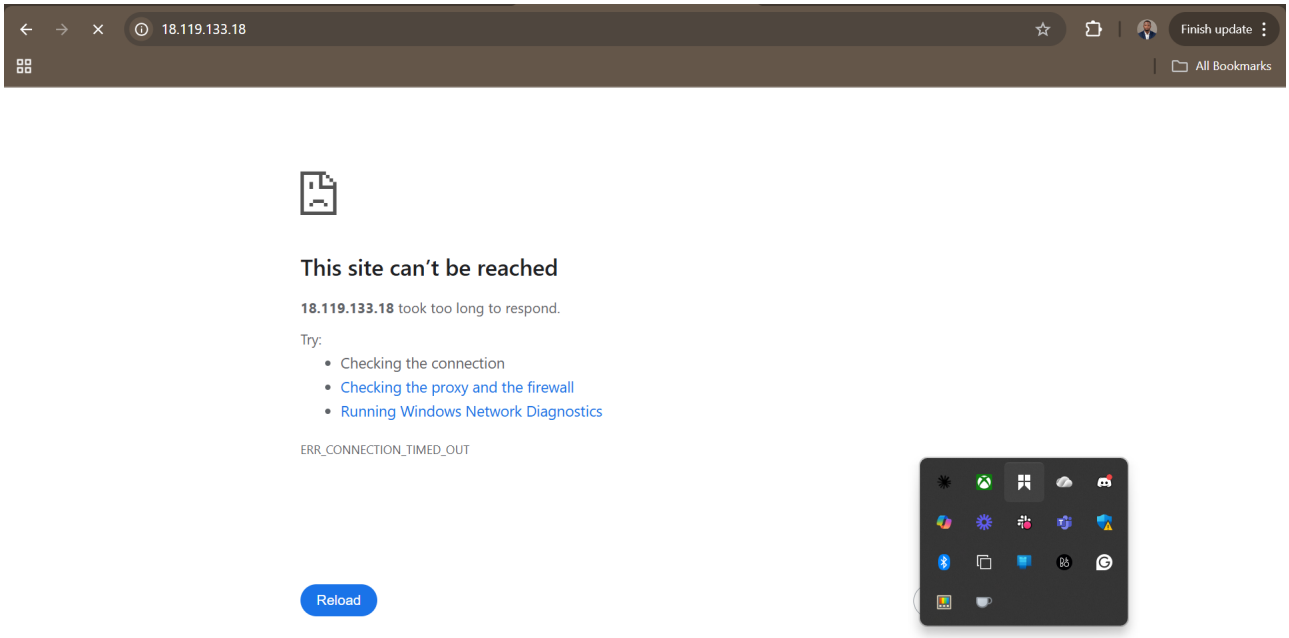
50. Security group outbound rule however allows only IPv4 All traffic connection using all protocol on all ports from anywhere.



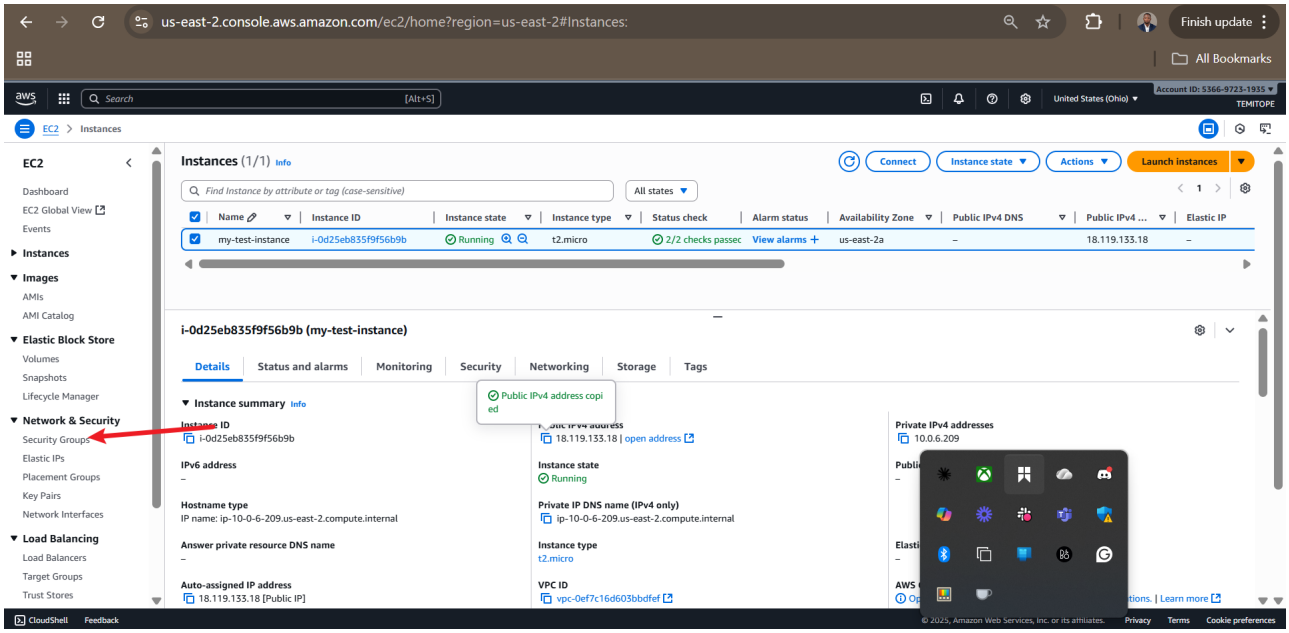
51. Back to instance, retrieve the public IP address of the instance from details tab.



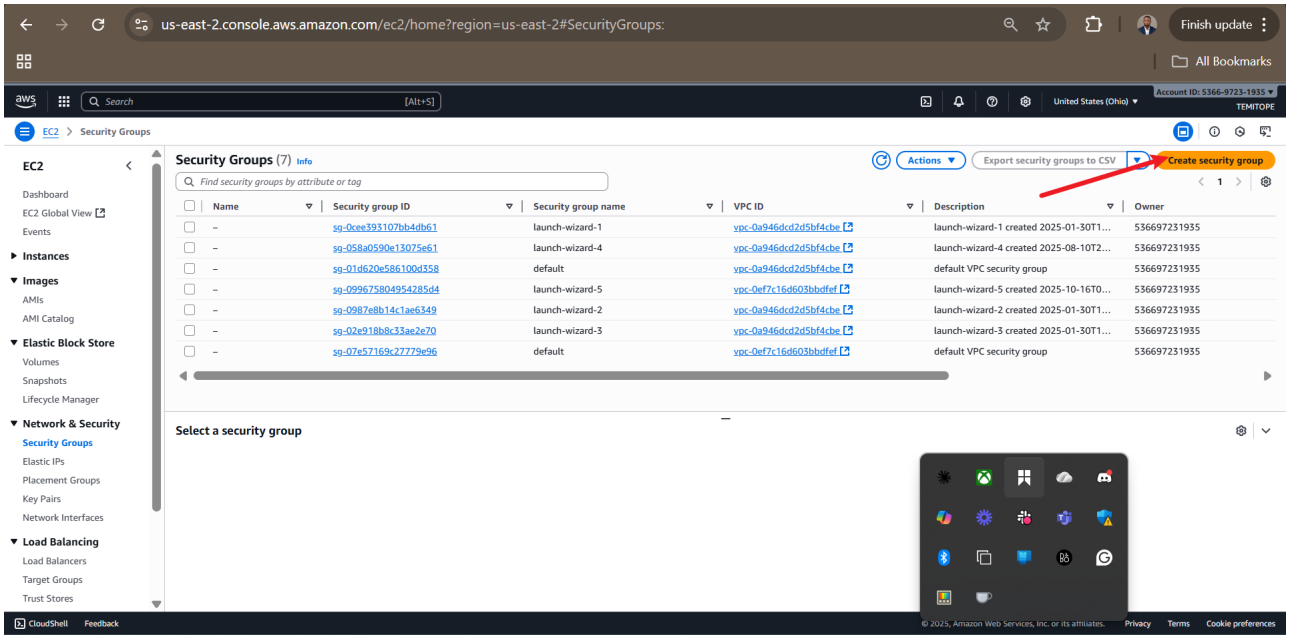
52. Using the IP, the site is unreachable due to HTTP rule not defined in the security group.



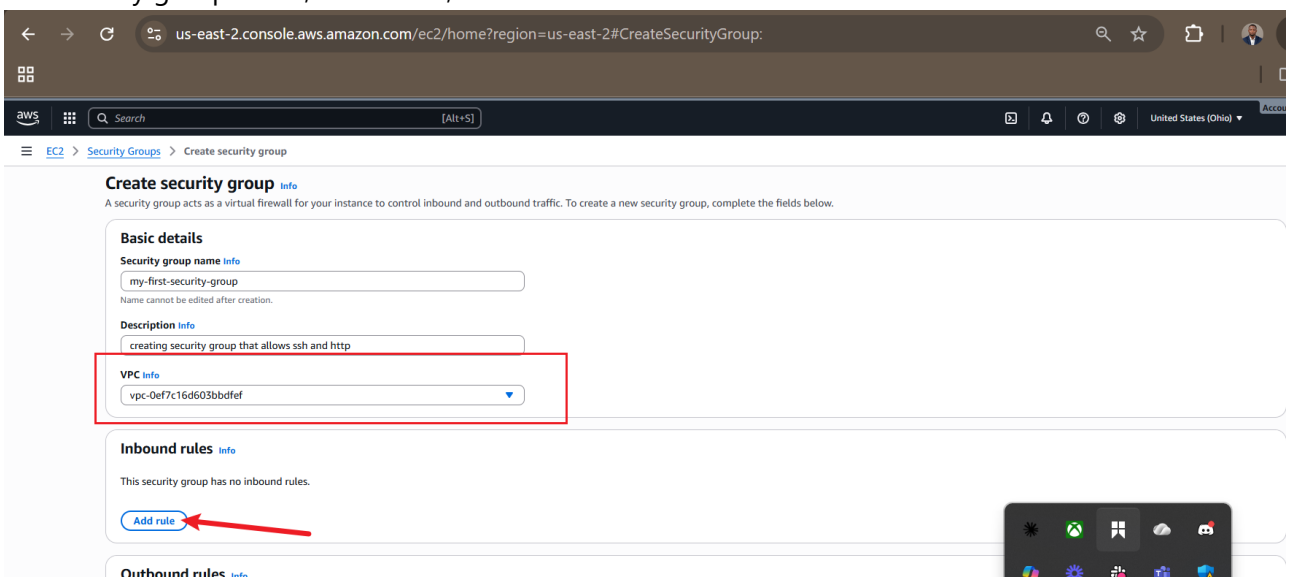
53. Navigate to "Security groups" on the left side bar and click on it.



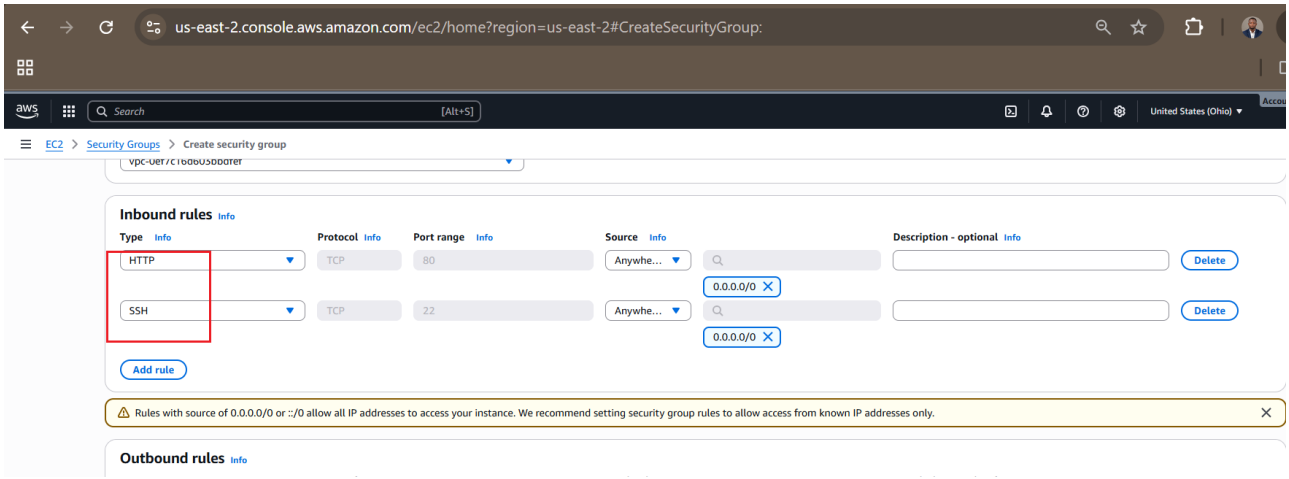
54. Click on "Create security group".



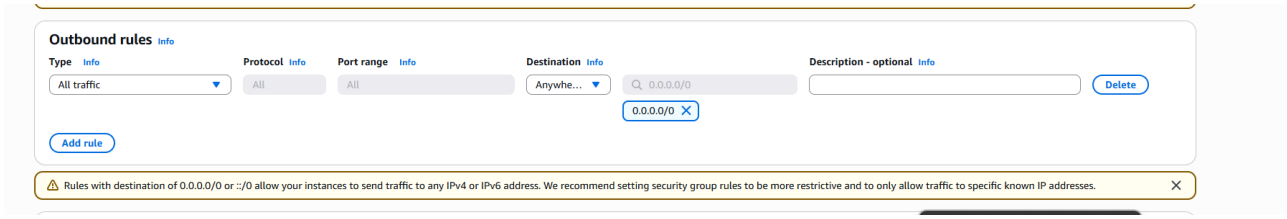
55. Set security group name, select VPC, and under inbound rules click on "Add rule".



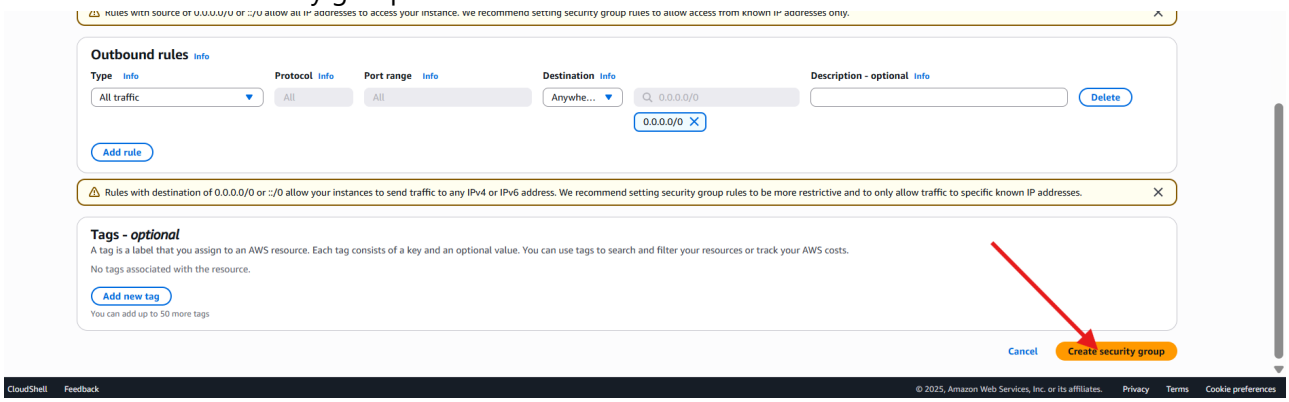
56. Add HTTP and SSH inbound rules respectively and set their source to be from anywhere.



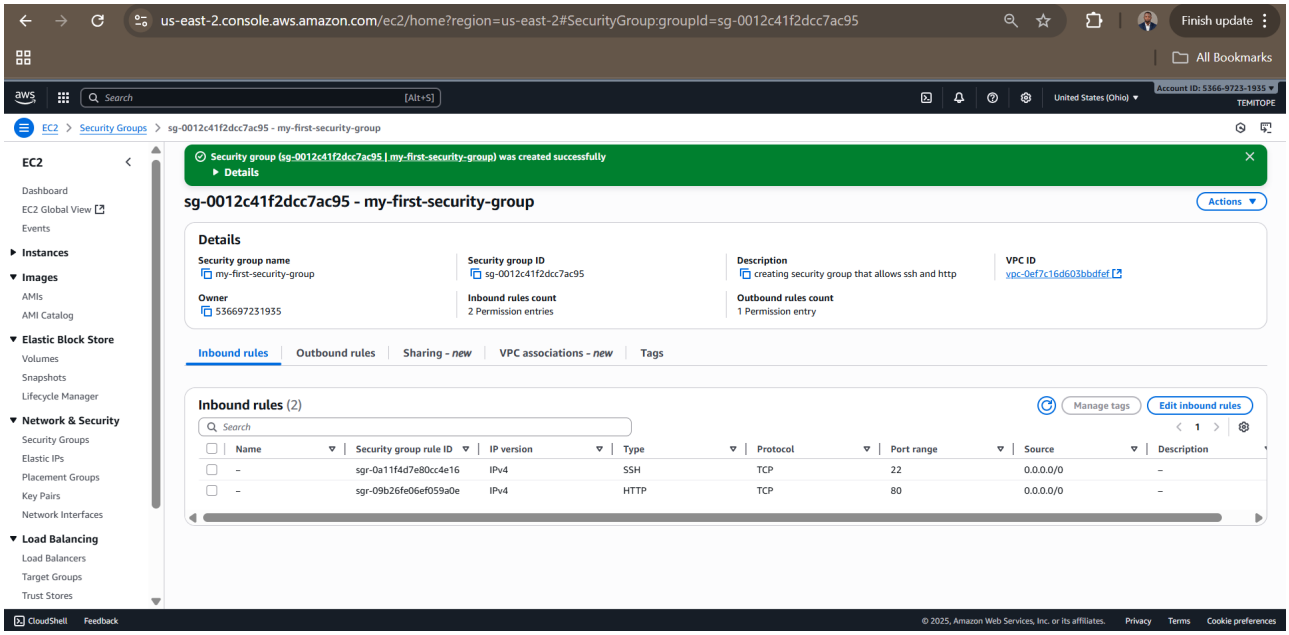
57. Keep the outbound rules the way it is.



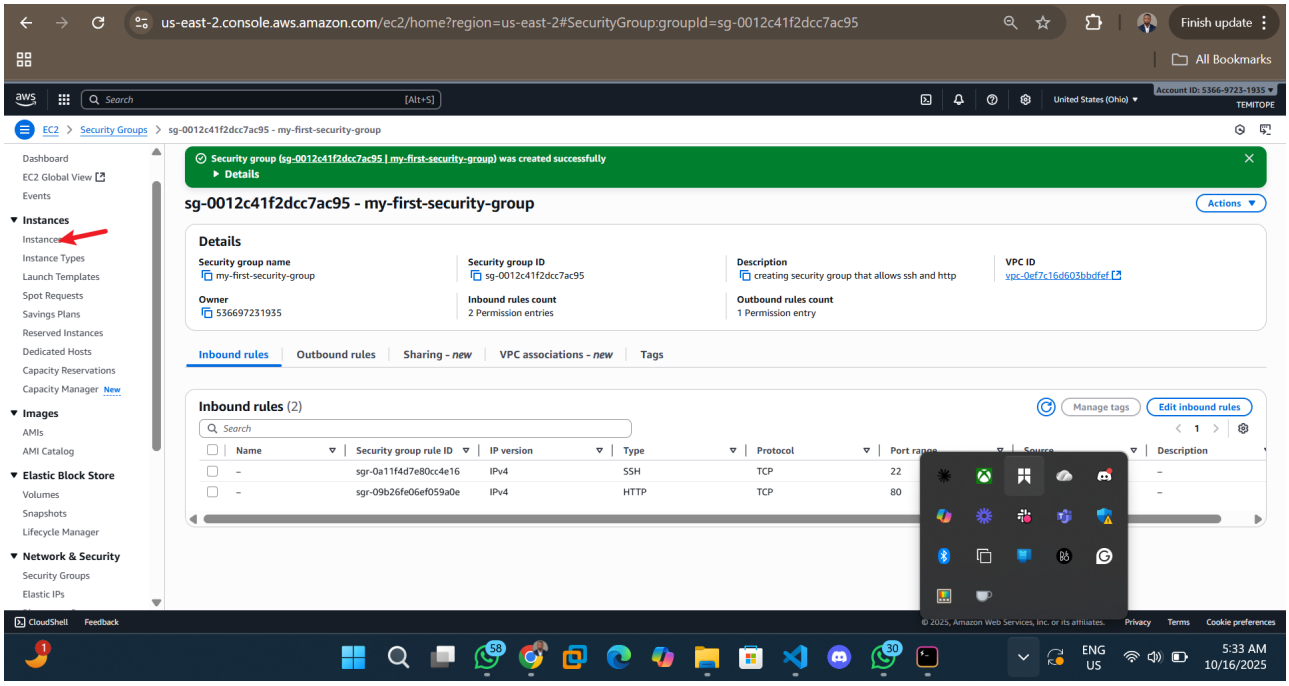
58. Now click "Create security group".



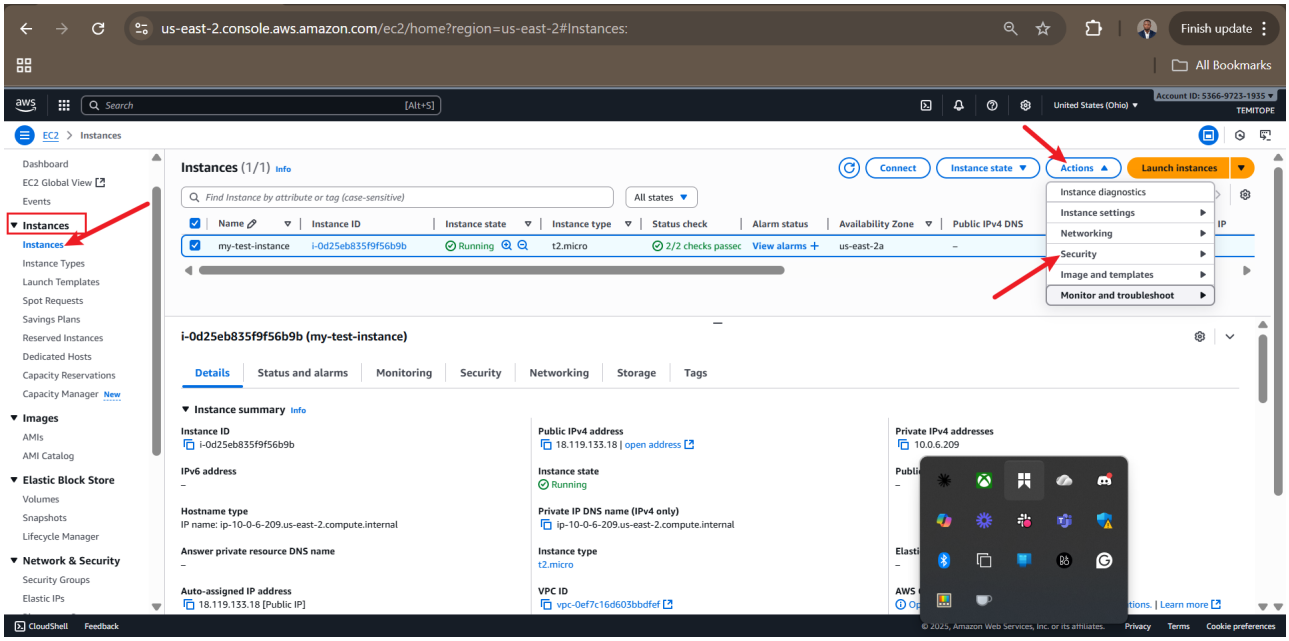
59. Security group created successfully.



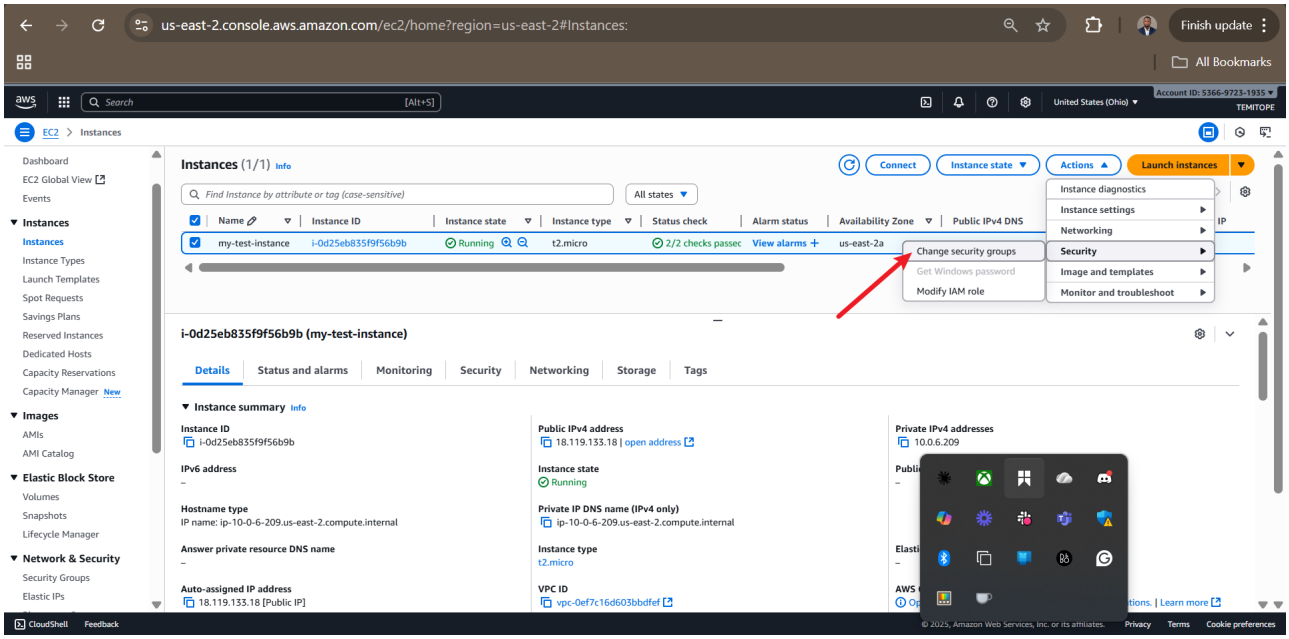
60. Click on "Instances" to navigate to instances.



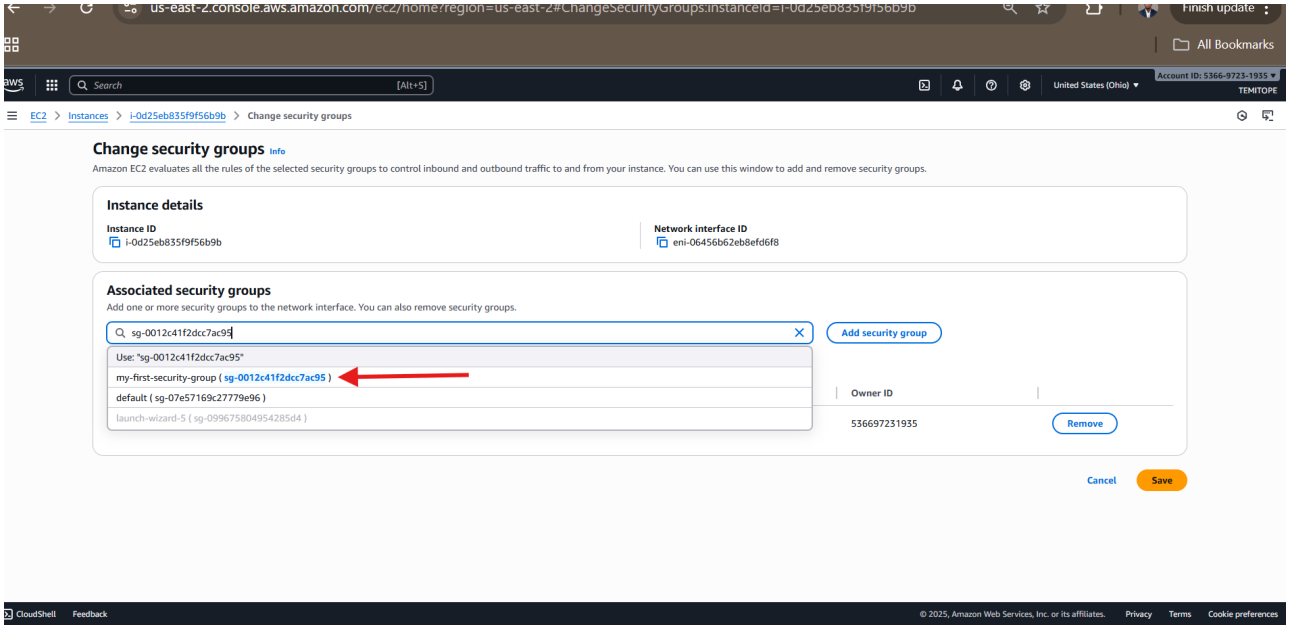
61. To attach created security group to my-test-instance, select it and click on "Actions" button then "Security" option.



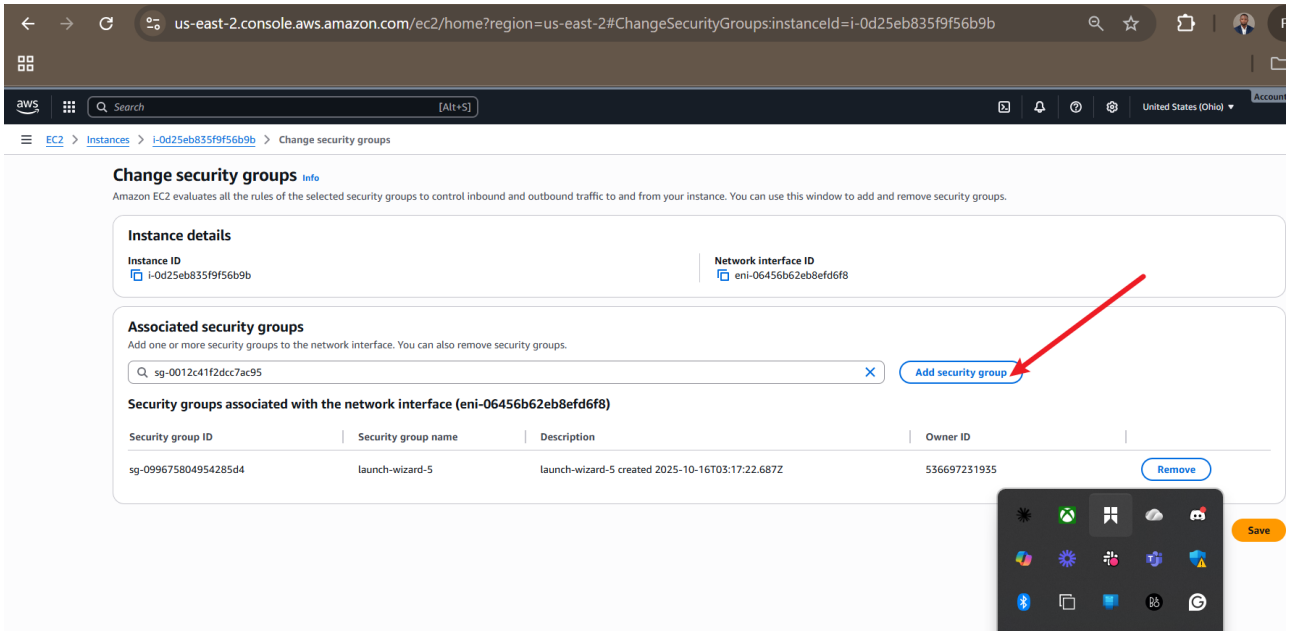
62. Under security option, click on "Change security groups".



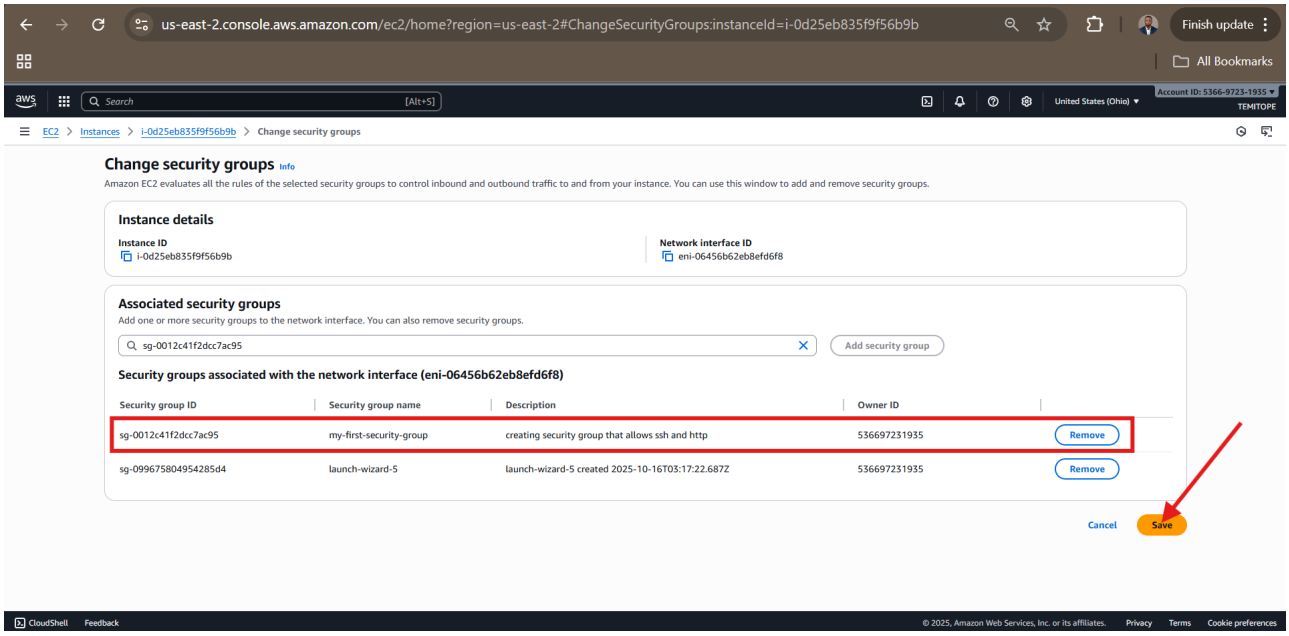
63. On change security groups page, select the created security group.



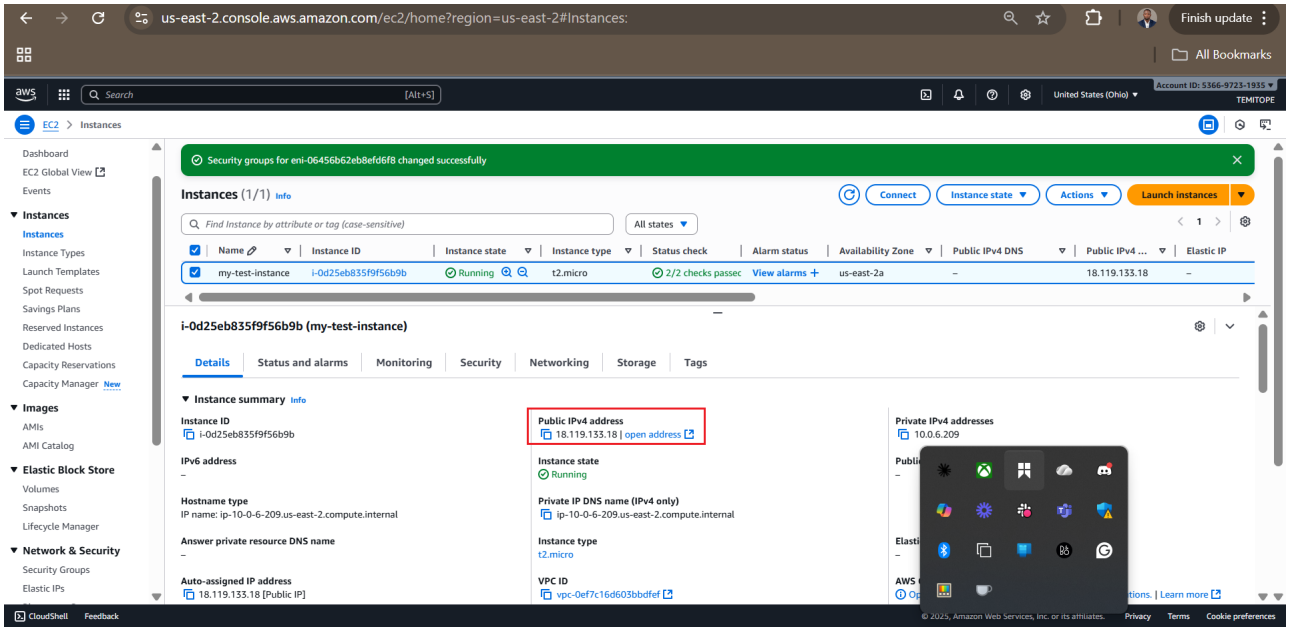
64. Click on "Add security group".



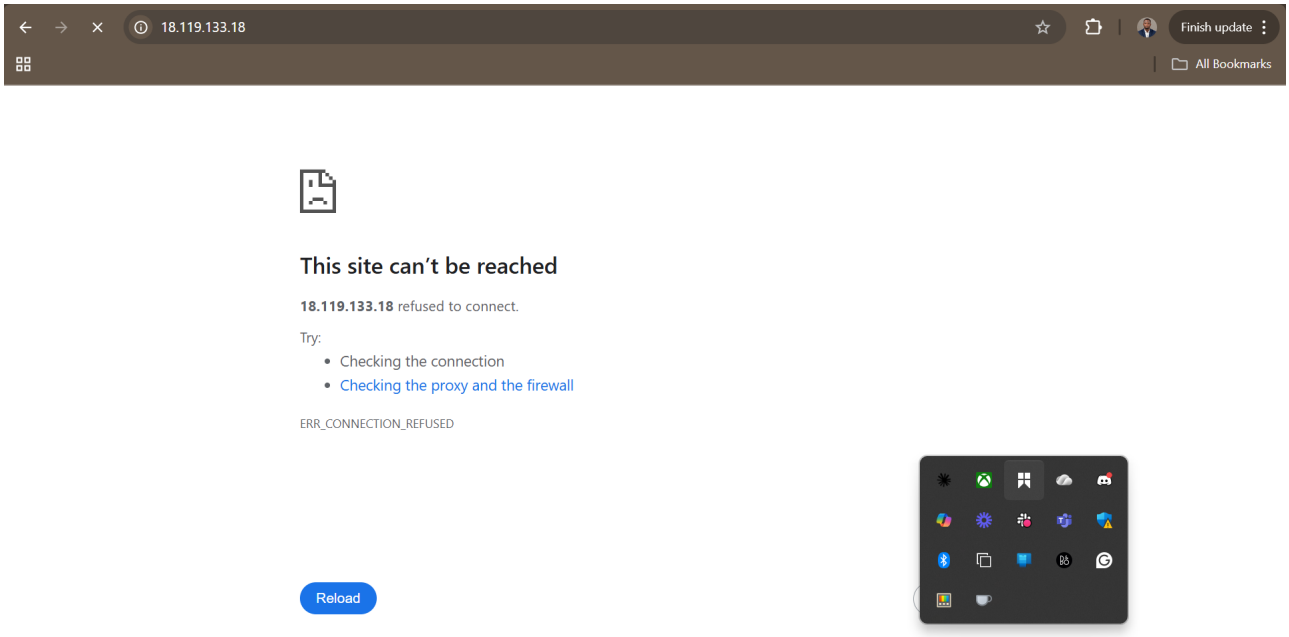
65. See that the security group is now added; now click "Save".



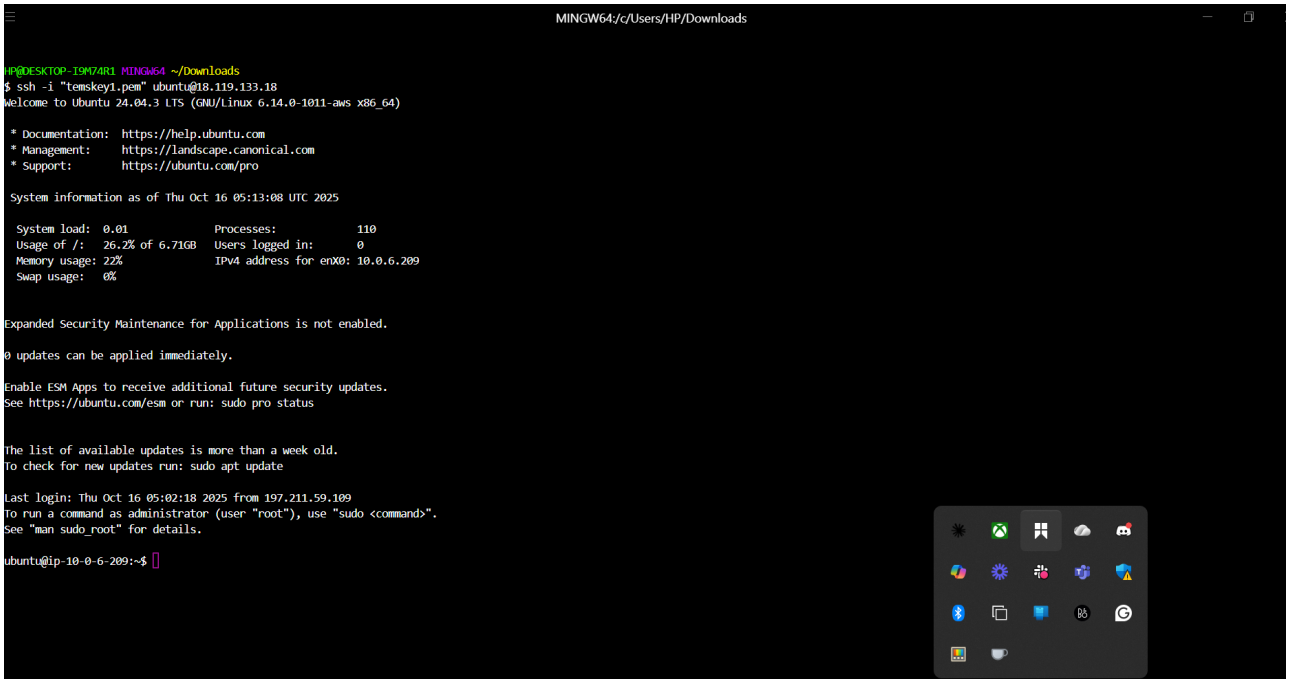
66. Security group now attached successfully; now copy the public address to test it on the browser again.



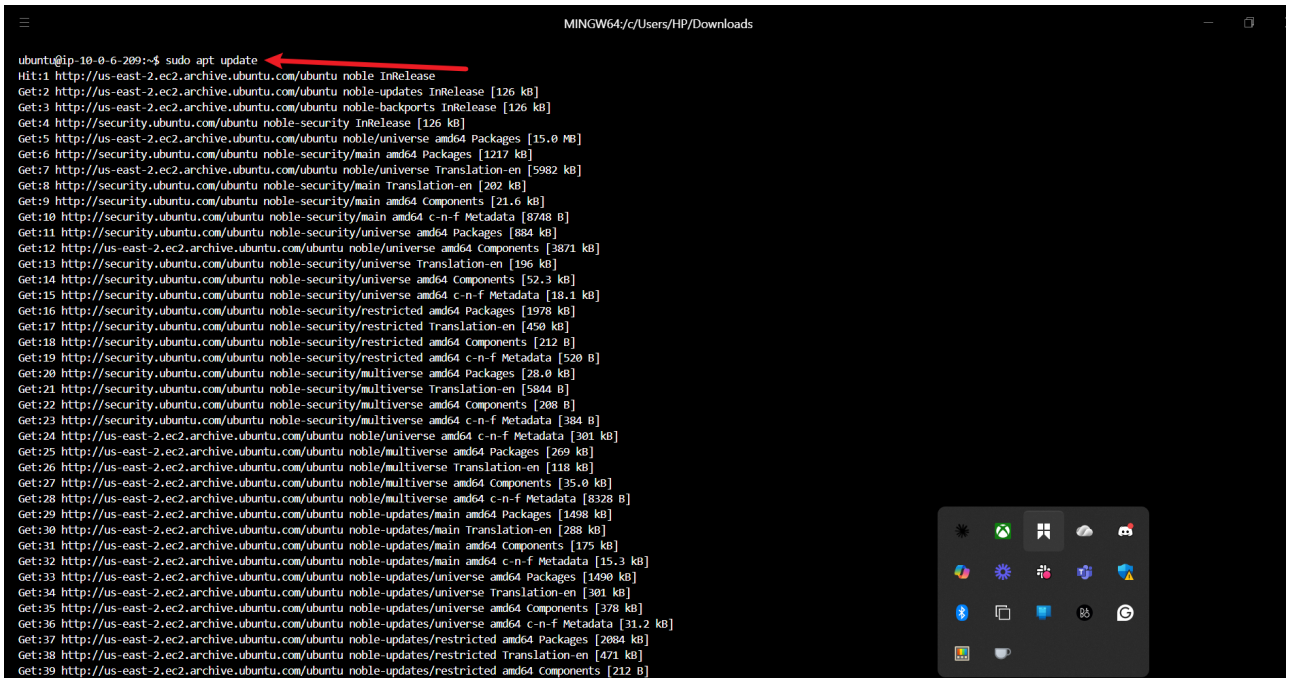
67. It is still unreachable because webserver is not automatically configured to serve webpage on a newly created instance.



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

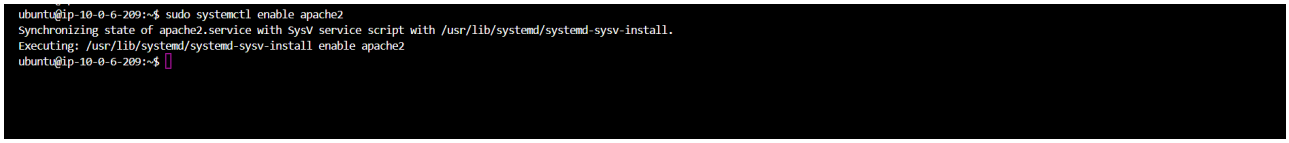


69. Update Ubuntu server.



70. Install apache2.

71. Enable apache2 with systemctl.

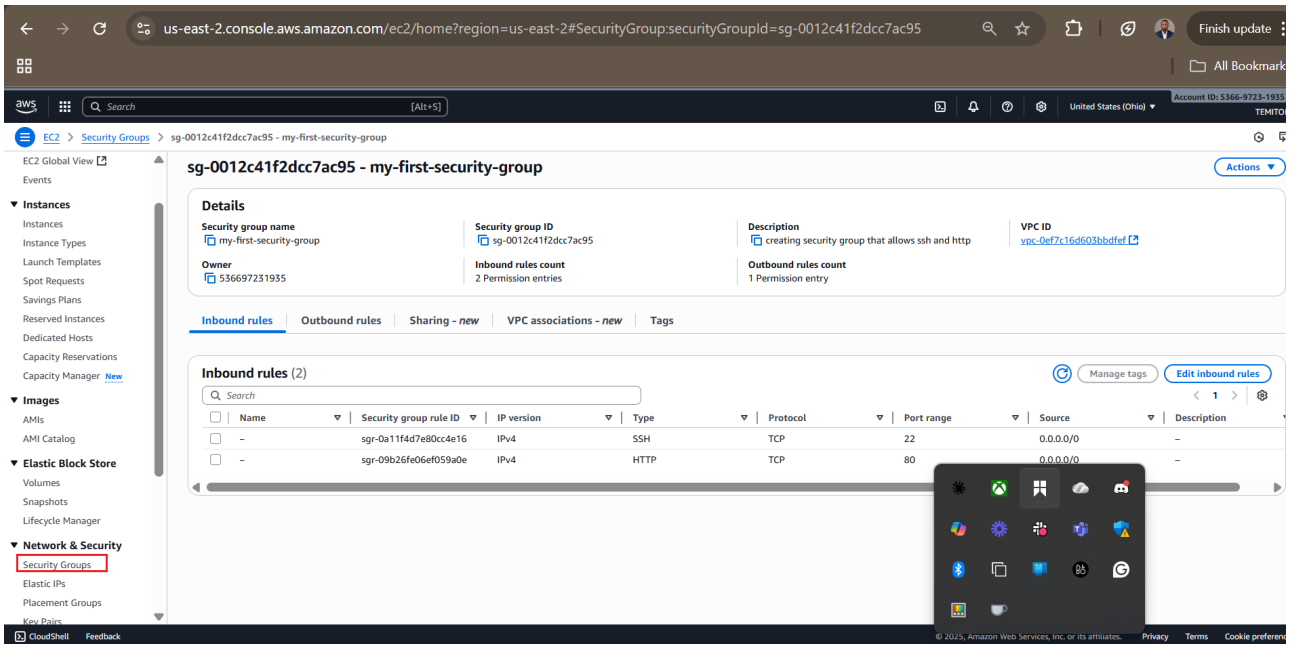


72. Start apache2 with systemctl.

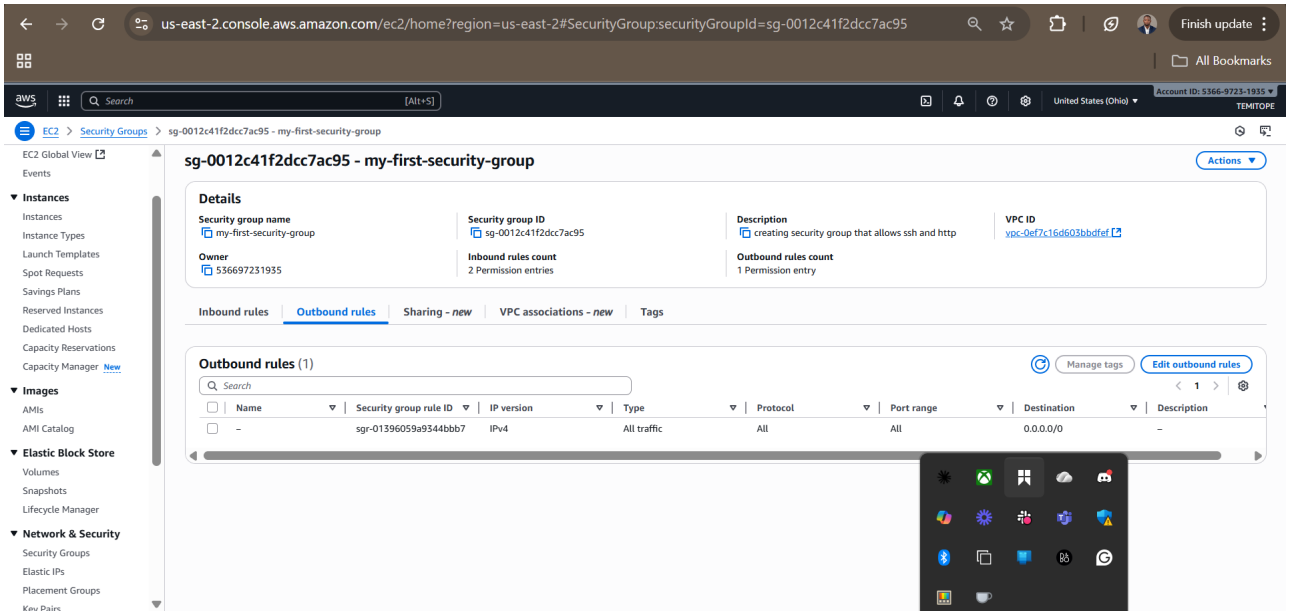
```
ubuntu@ip-10-0-6-209:~$ sudo systemctl start apache2
ubuntu@ip-10-0-6-209:~$
```

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).



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

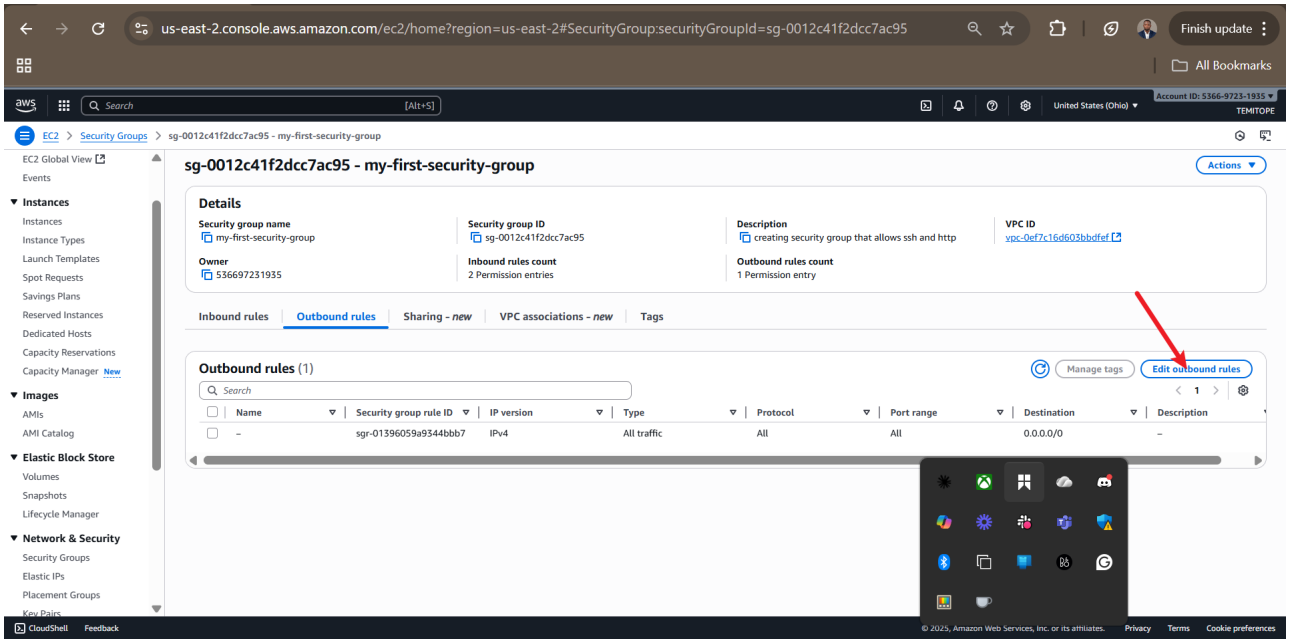


Scenario 1: Inbound Rules for HTTP/SSH, Outbound for All (Default Successful Access)

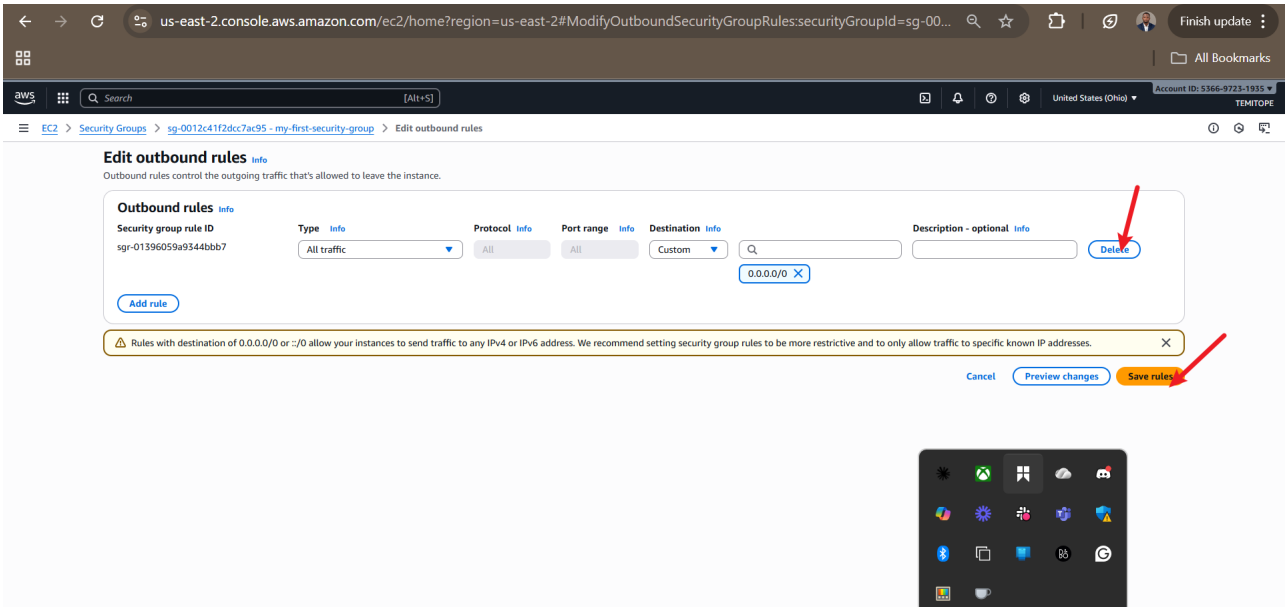
This is the initial configuration where inbound allows HTTP and SSH from anywhere, and outbound allows all traffic. The website is reachable.

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

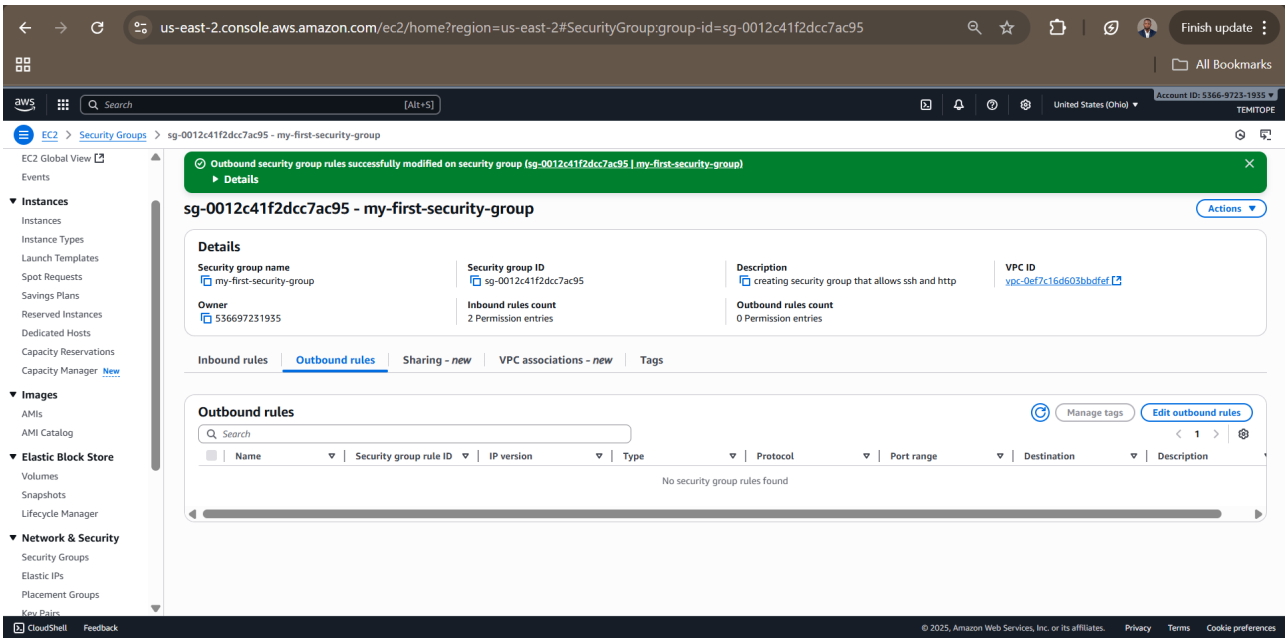
76. On the outbound rules tab, click "Edit outbound rules".



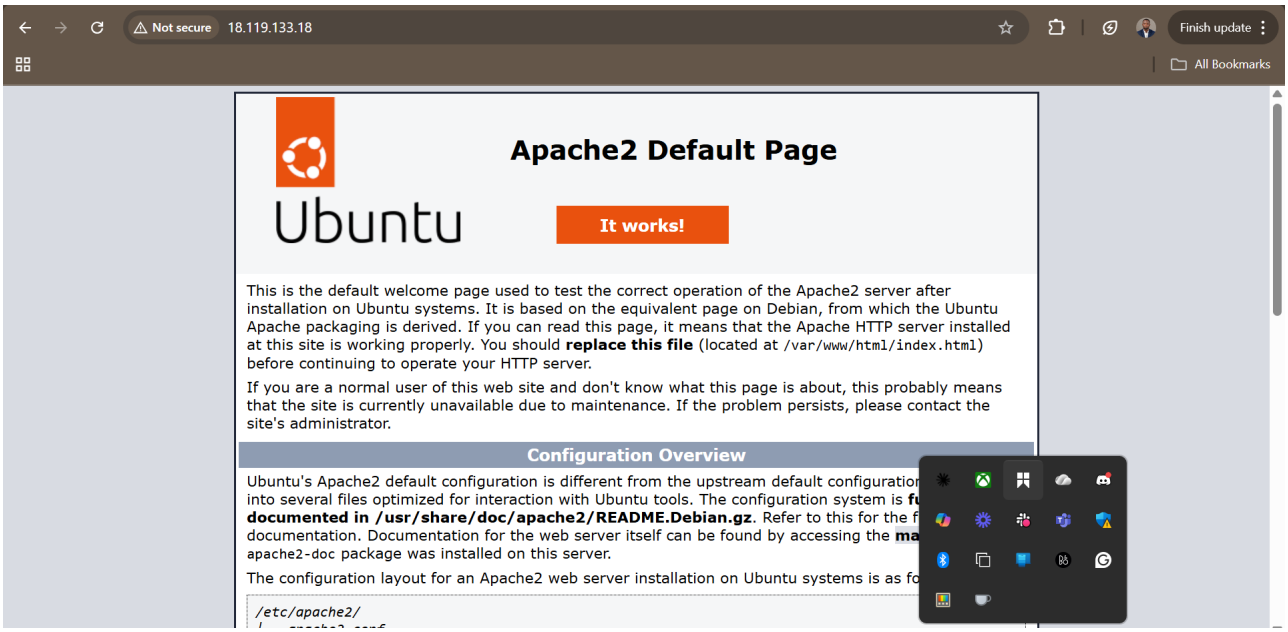
77. Click "Delete" to delete outbound rule and click "Save rules".



78. Outbound security rules now deleted.



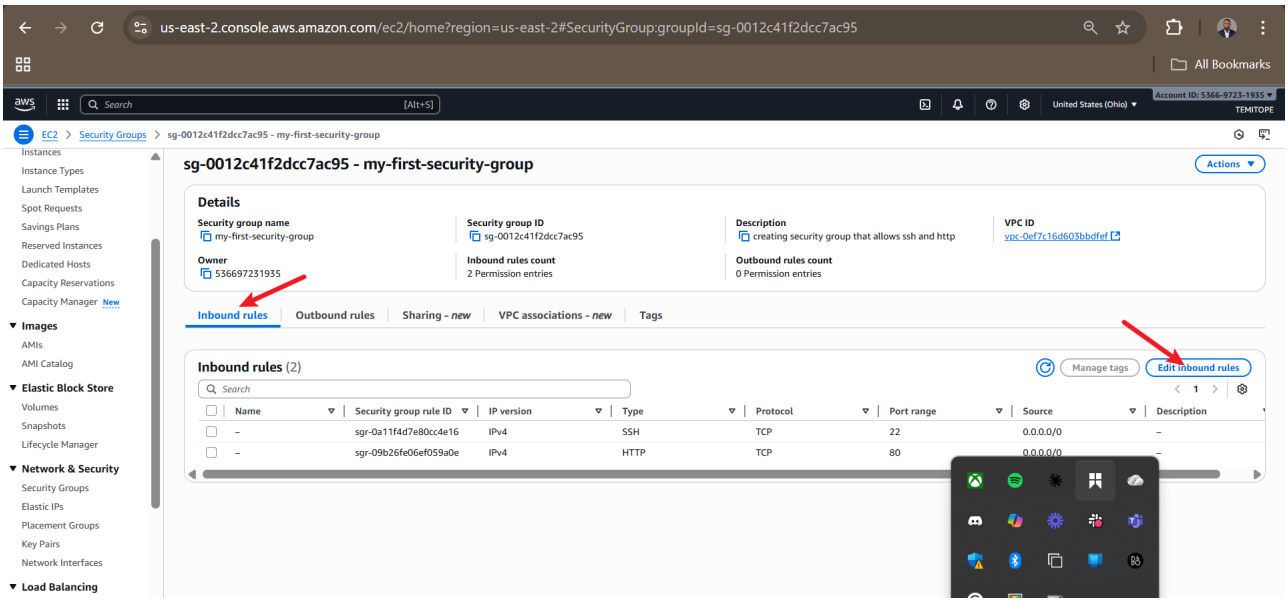
79. Website reachability tested after deleting the outbound rules and nothing breaks as website is still reachable.



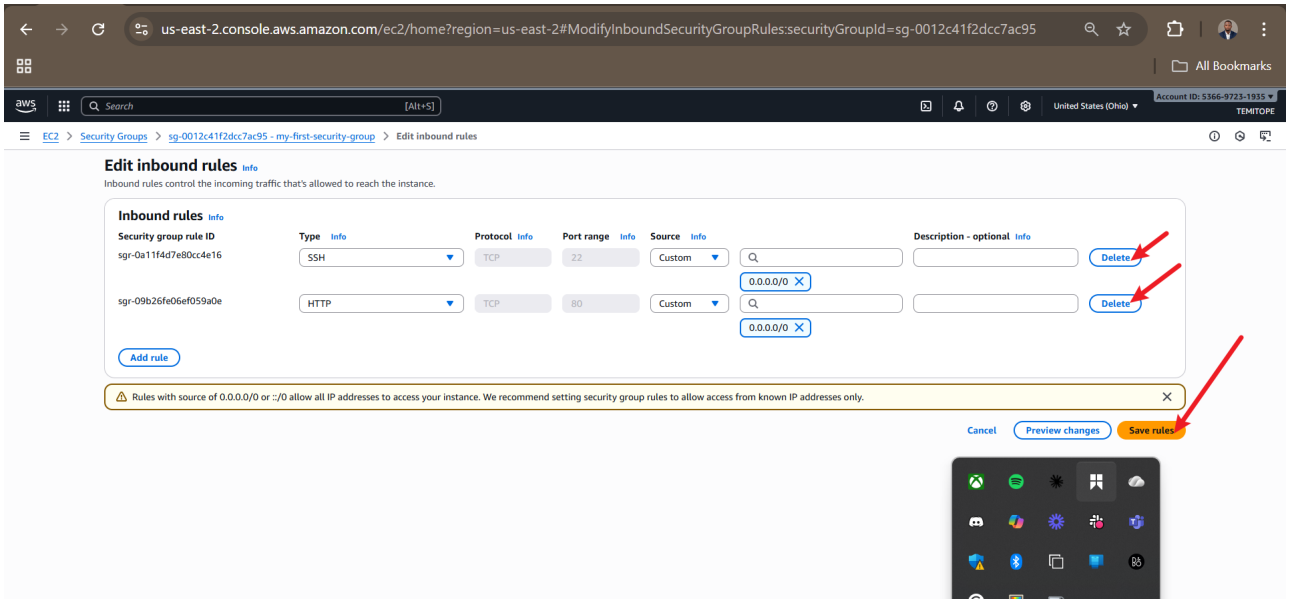
(Website remains reachable due to the stateful nature of Security Groups.)

Scenario 3: Remove Both Inbound and Outbound Rules

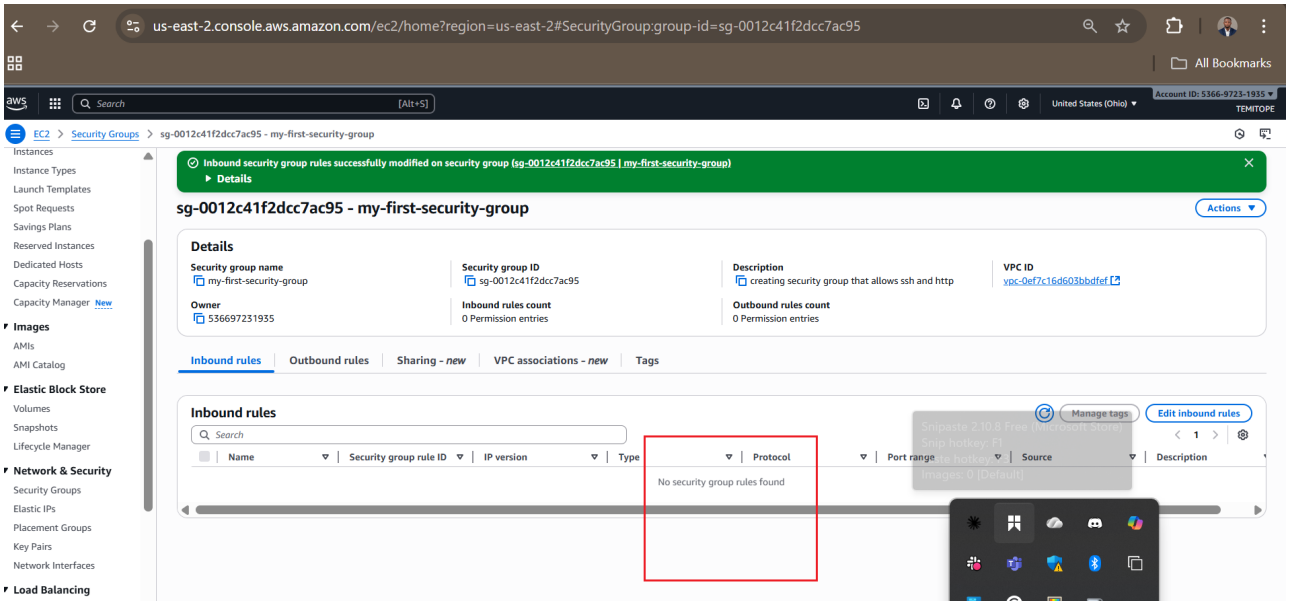
80. Back to the created Security Group, click "Inbound rules" and then "Edit inbound rules".



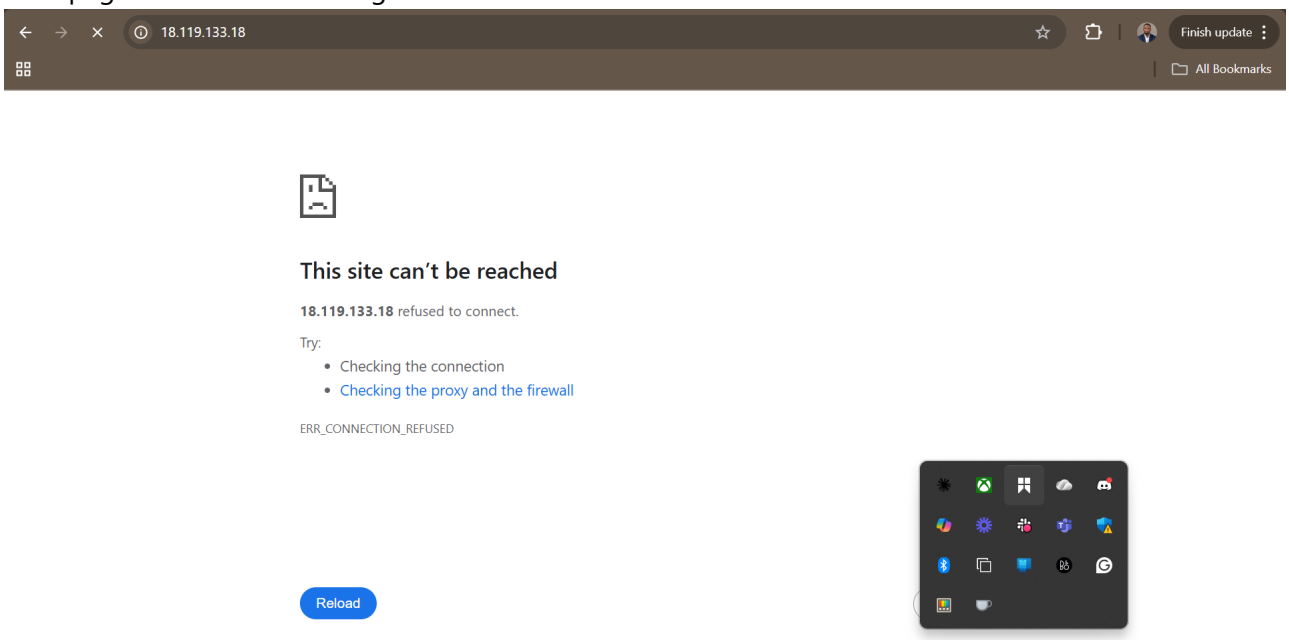
81. Delete both inbound rules and click "Save rules".



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



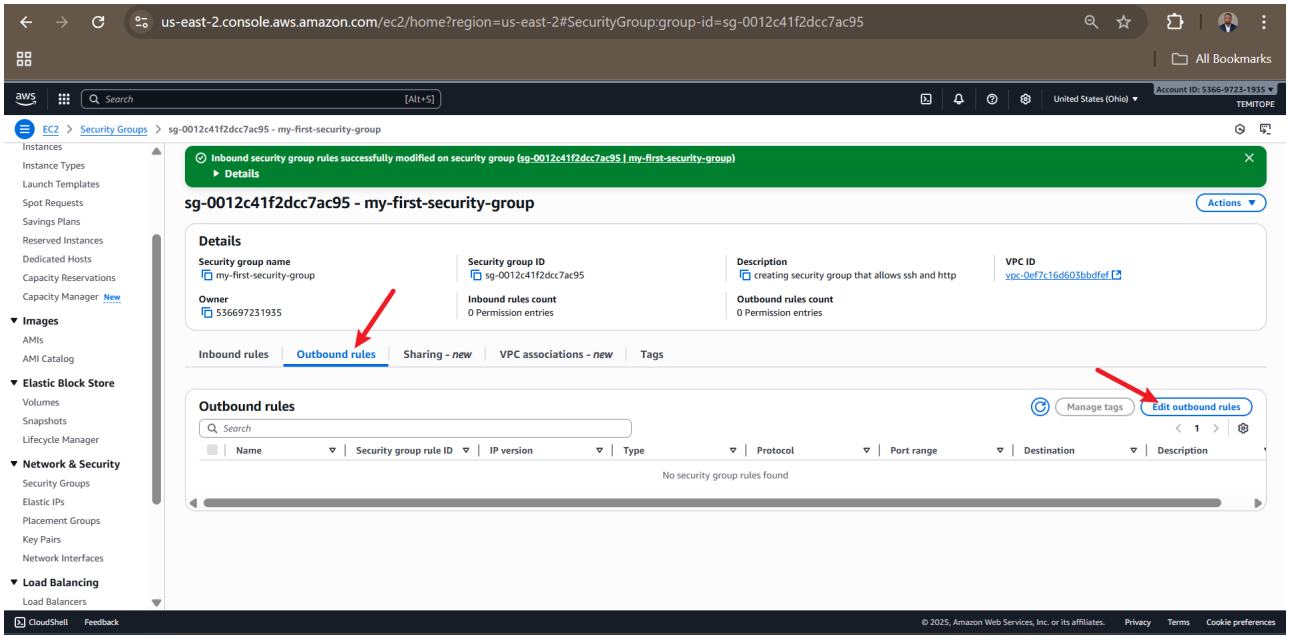
83. Web page now unreachable again.



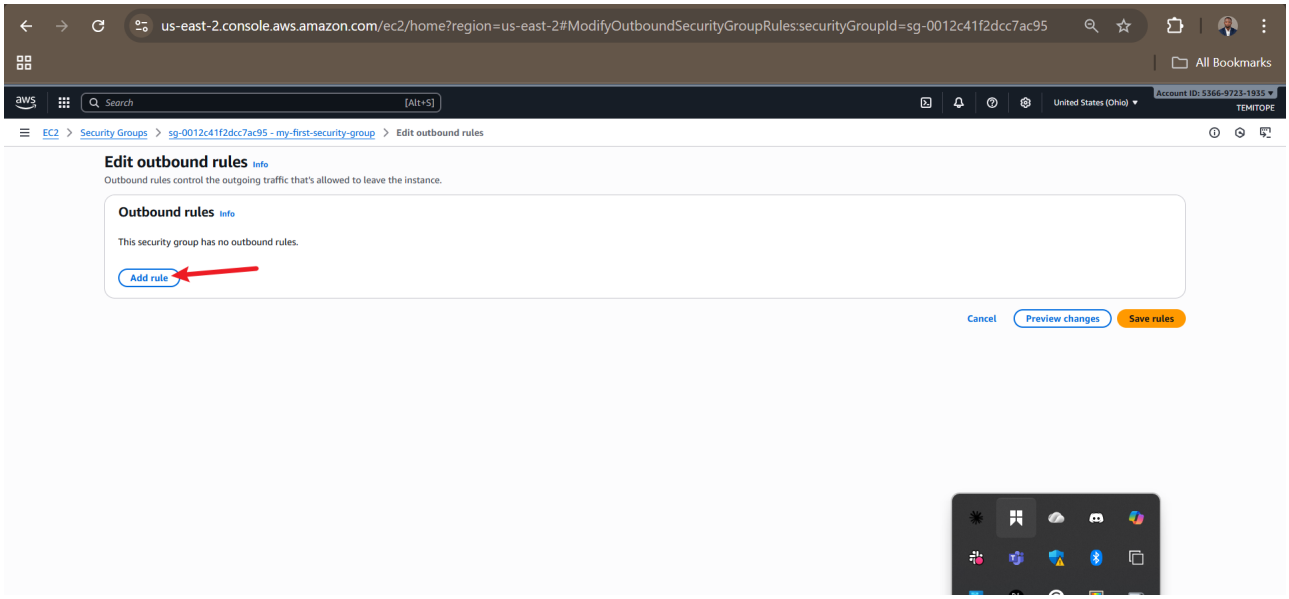
(No access without inbound rules.)

Scenario 4: No Inbound Rules, Outbound Rules for HTTP

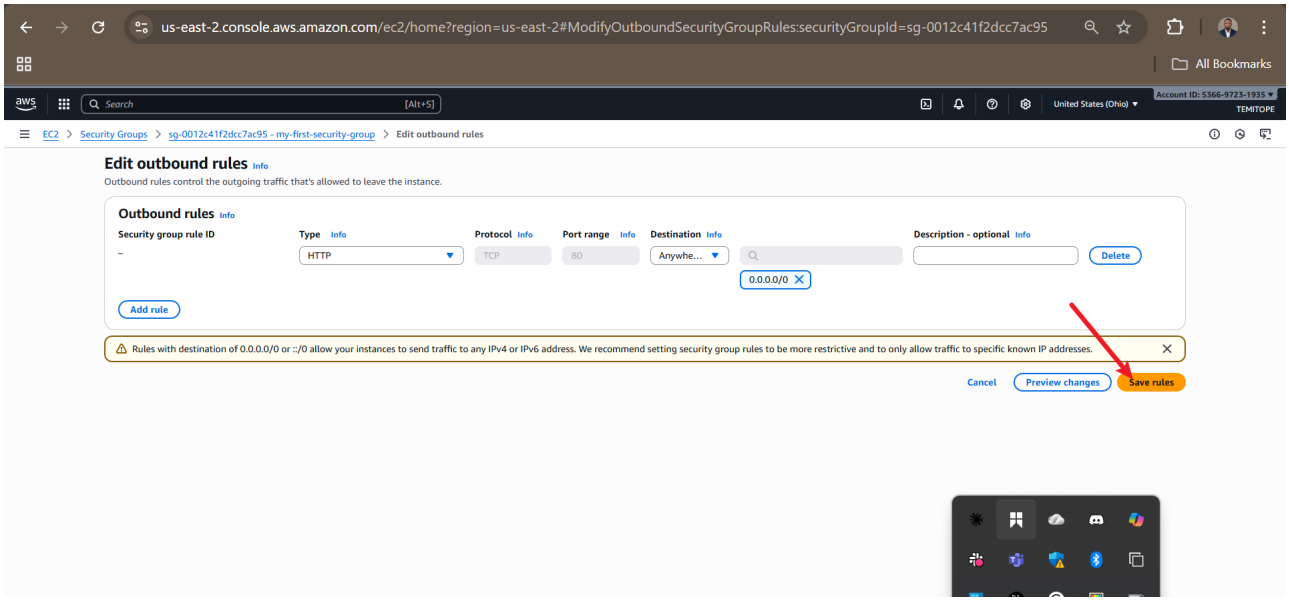
84. Back to the created Security Group, click "Outbound rules" and then "Edit outbound rules".



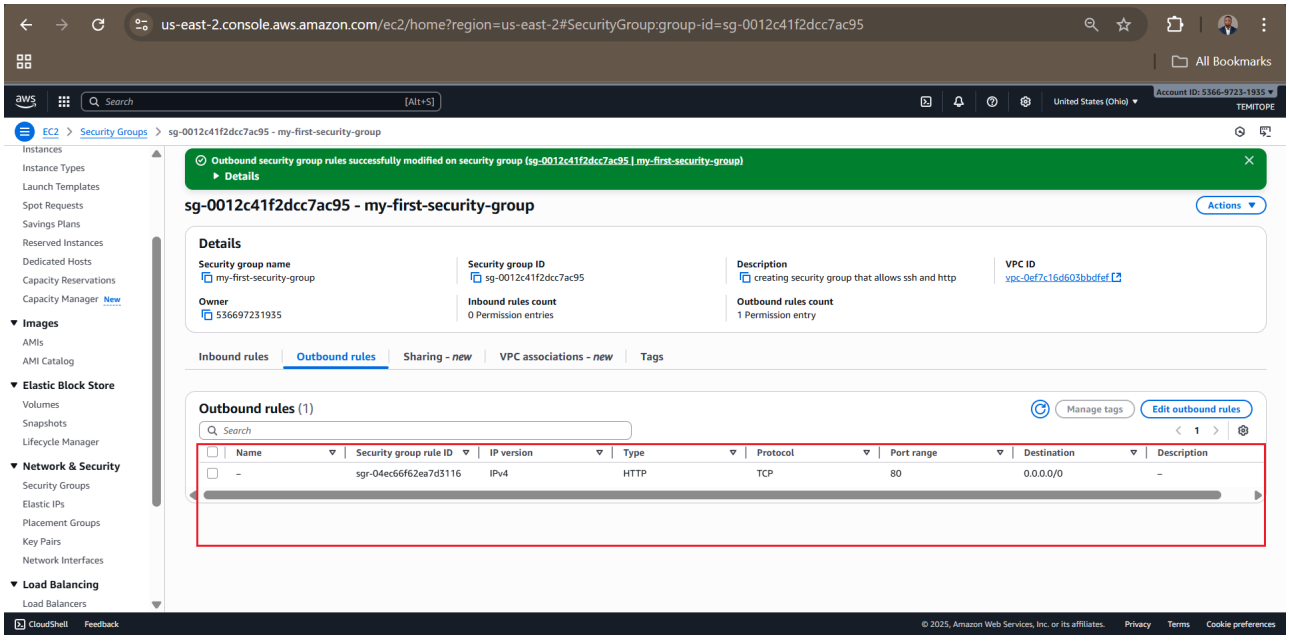
85. Click on "Add rules".



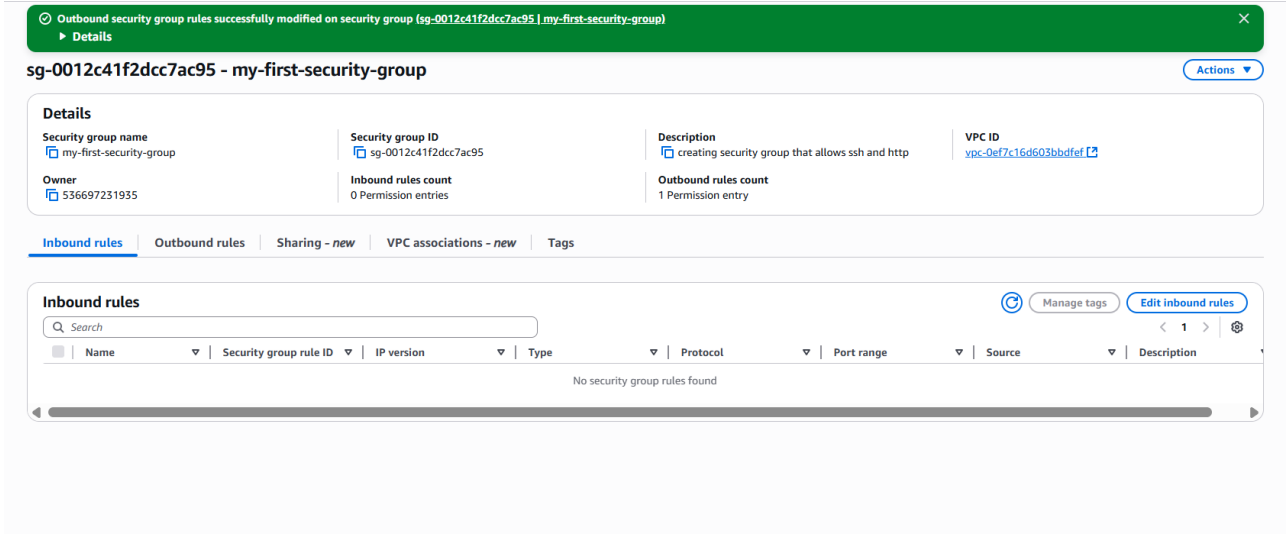
86. Set type HTTP, protocol TCP, port 80, Destination Anywhere, and save rules.



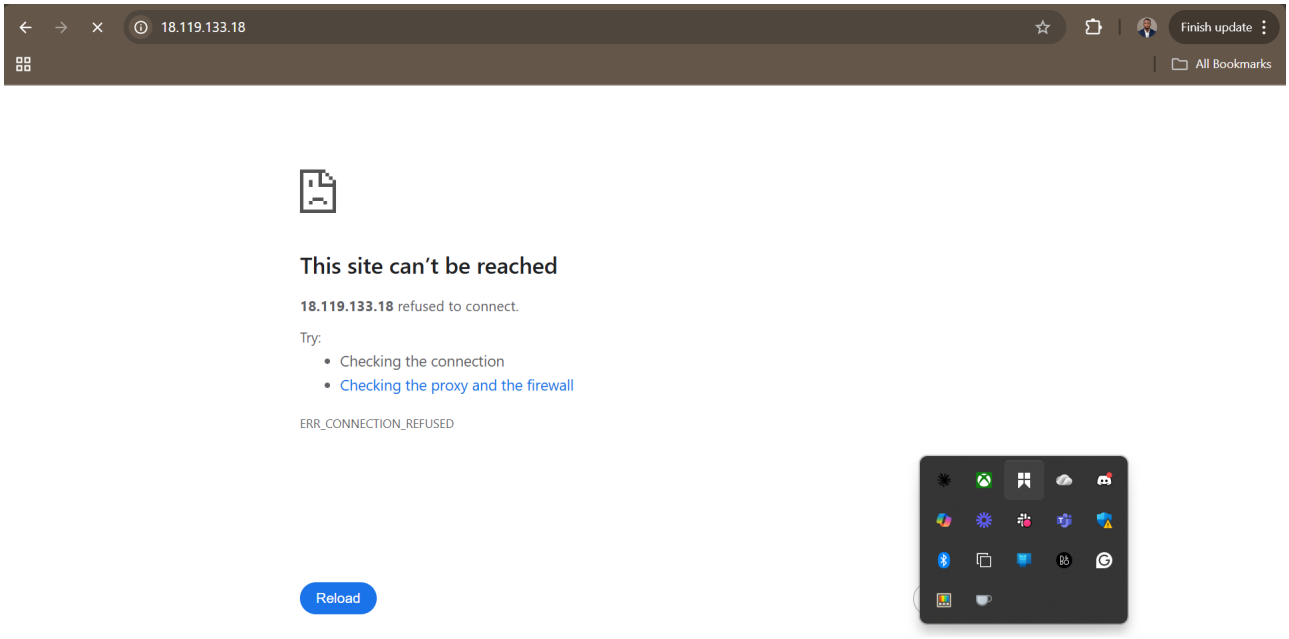
87. Outbound rules successfully modified.



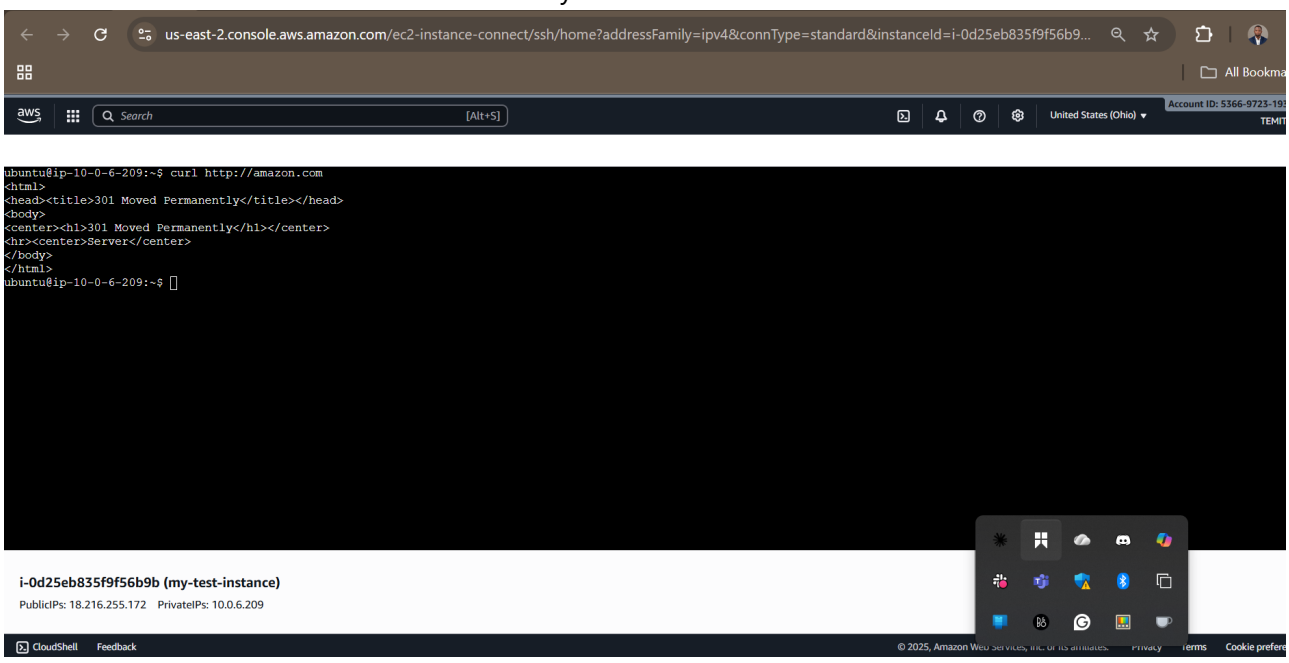
88. Note that inbound rules is still empty because we deleted it.



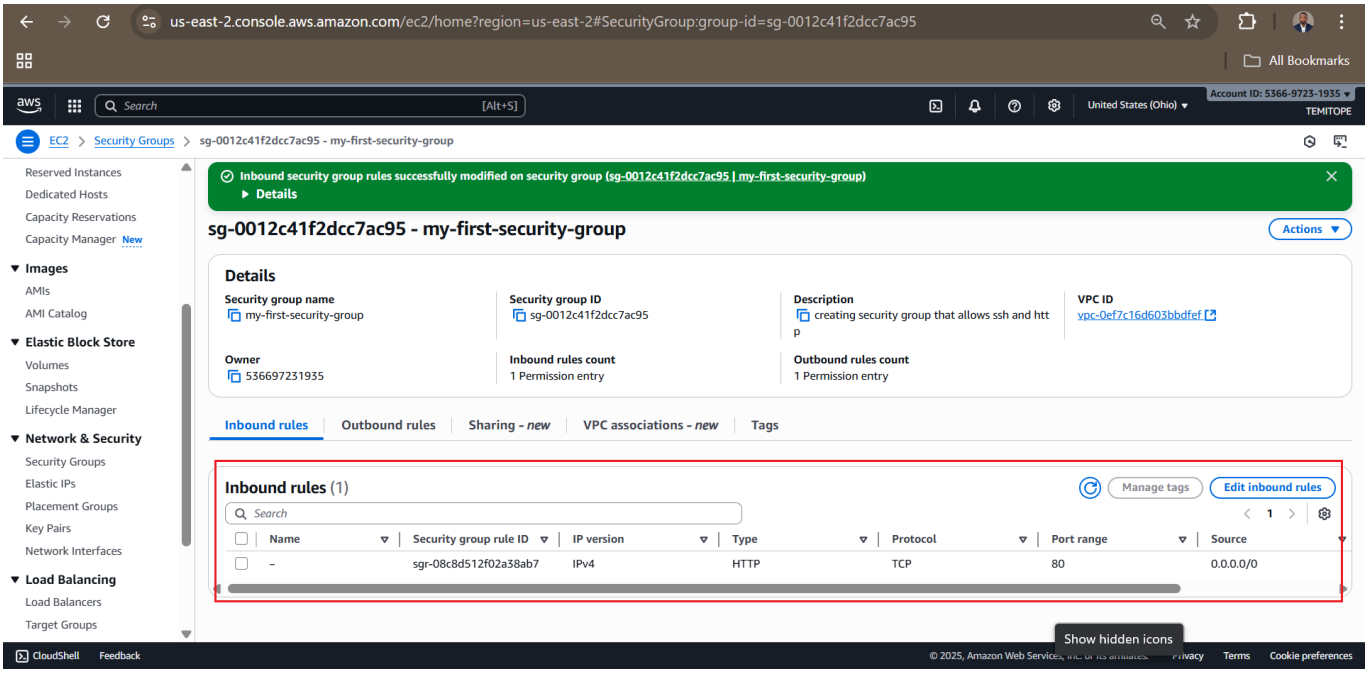
89. Web page now unreachable again.



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.

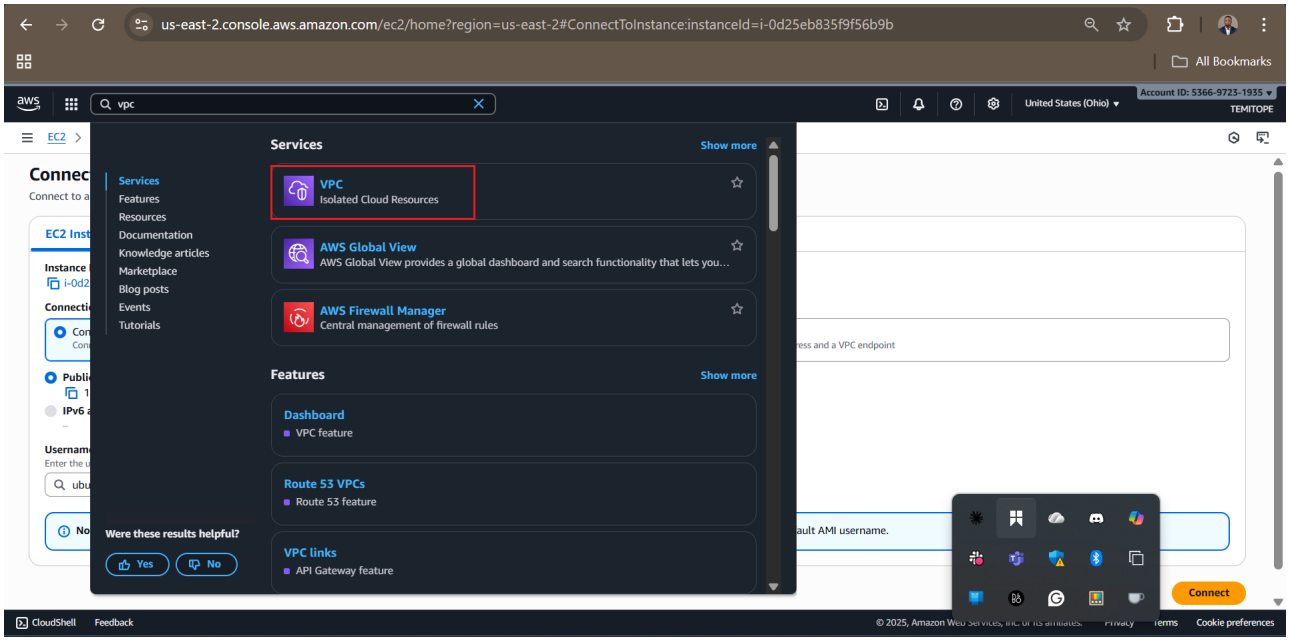


(Outbound allows traffic, but no inbound means the website is unreachable from outside.)

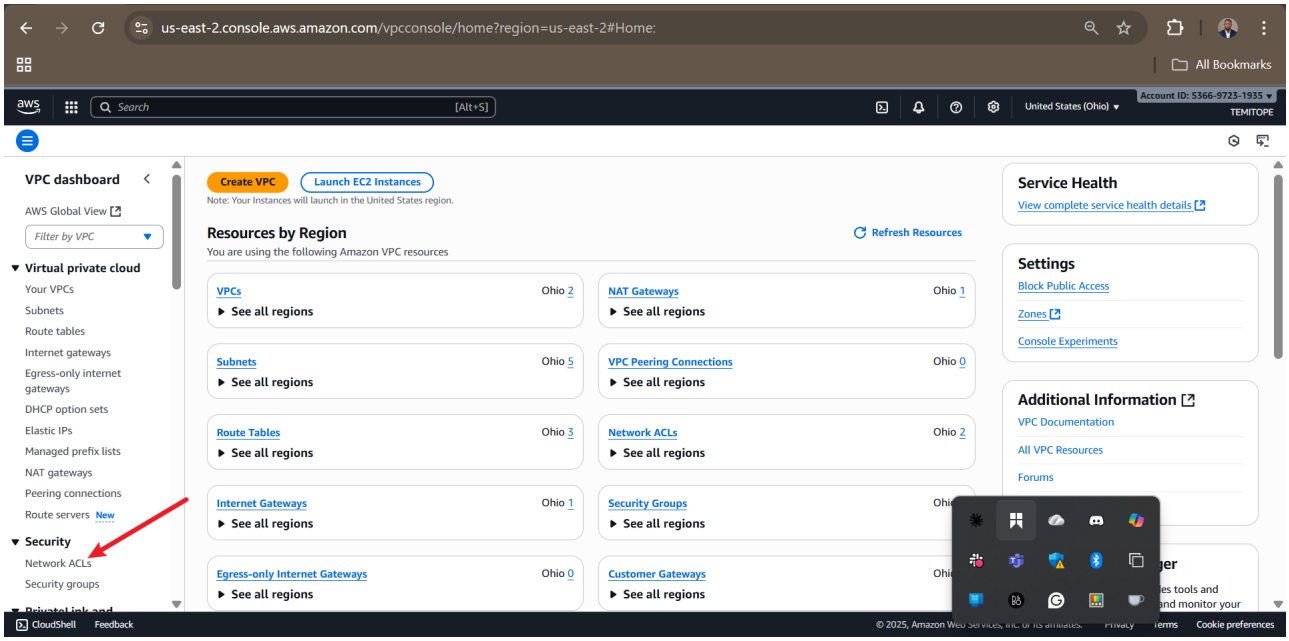
Part 2: NACLs

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

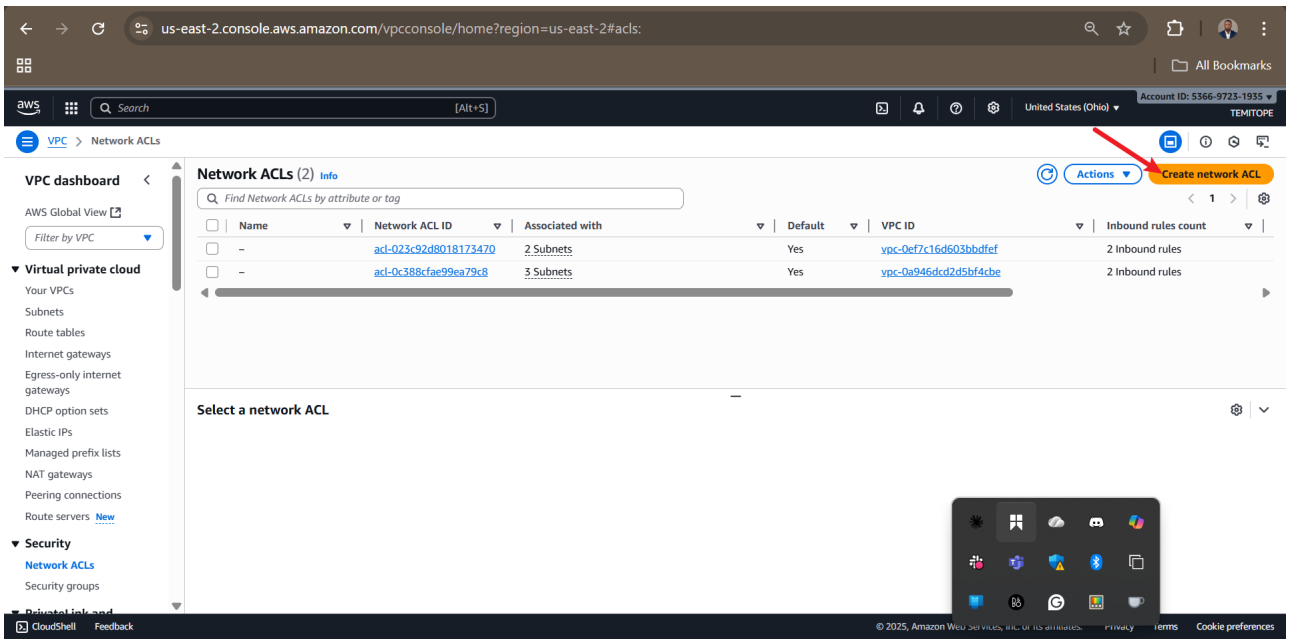
91. Navigate to VPC using the console search.



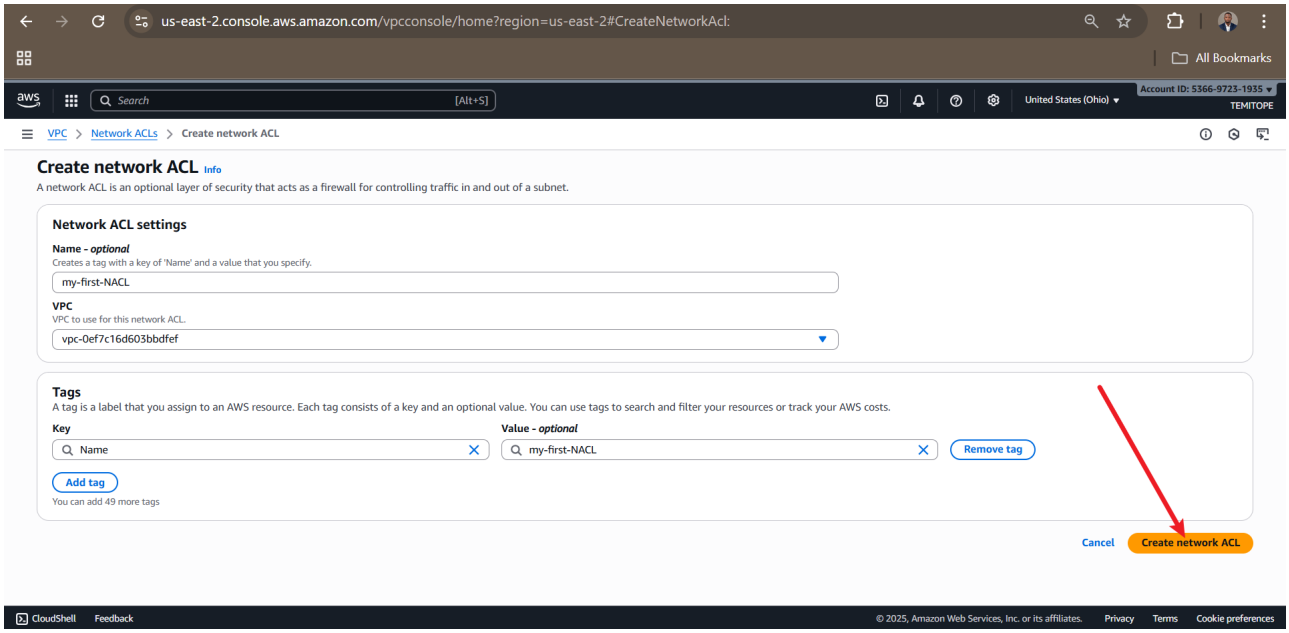
92. Click on "Network ACLs" to navigate to it.



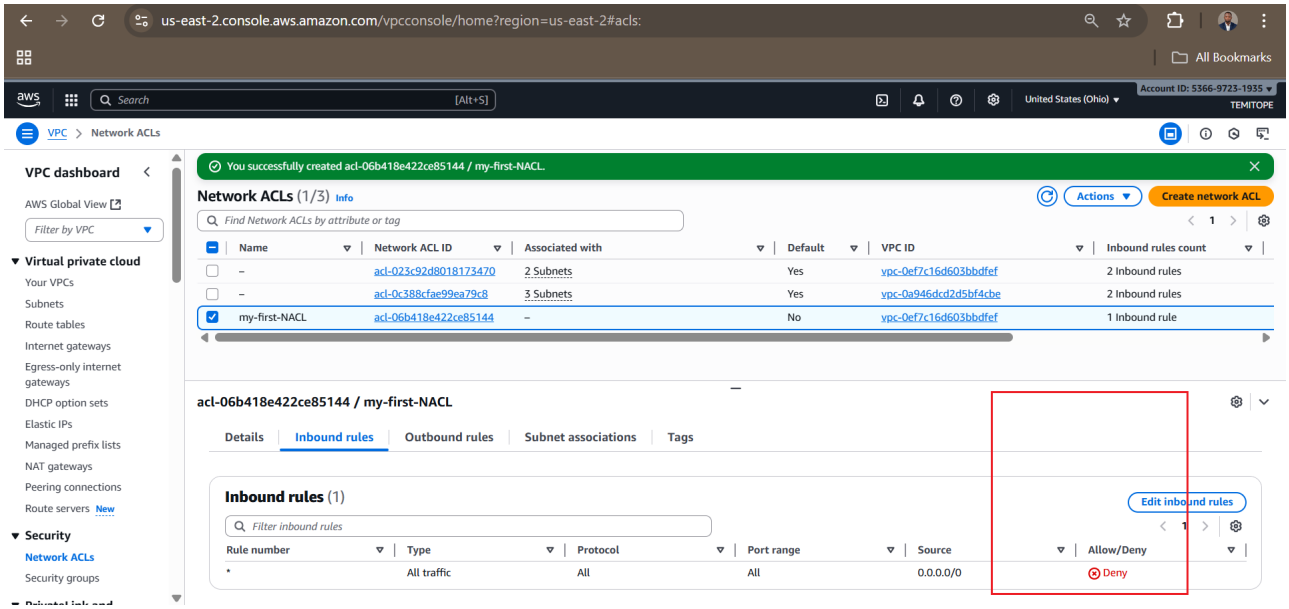
93. Click on "Create network ACL".



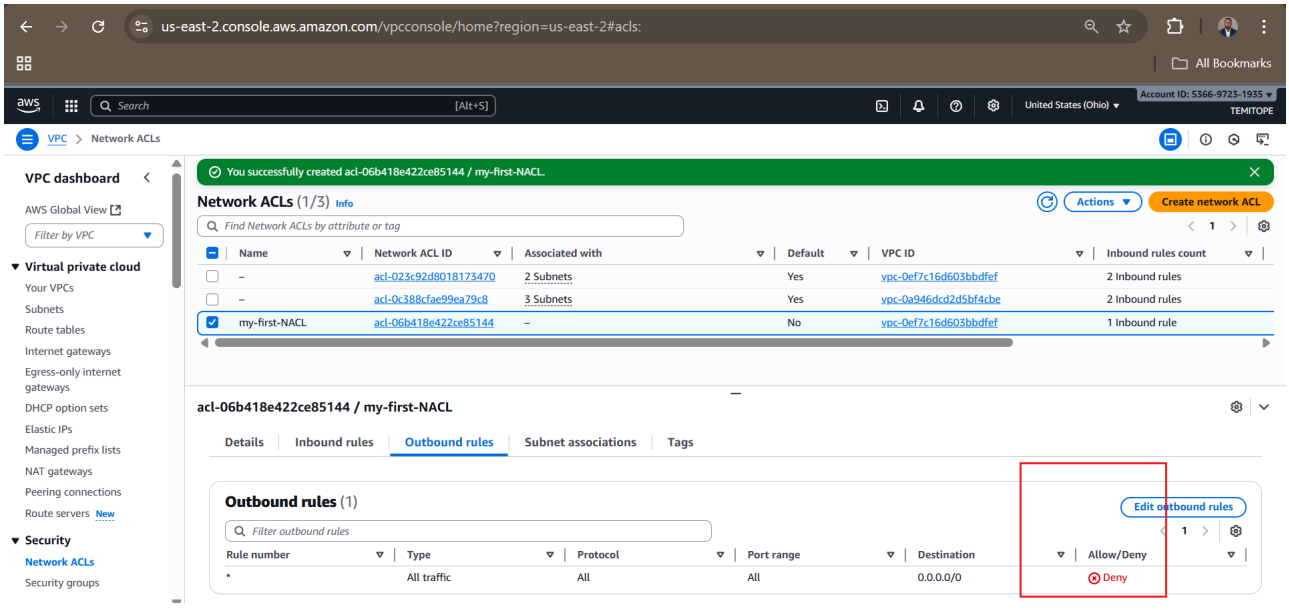
94. Name network ACL, select the created VPC, and click "Create network ACL".



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



96. All outbound traffic are also denied by default.

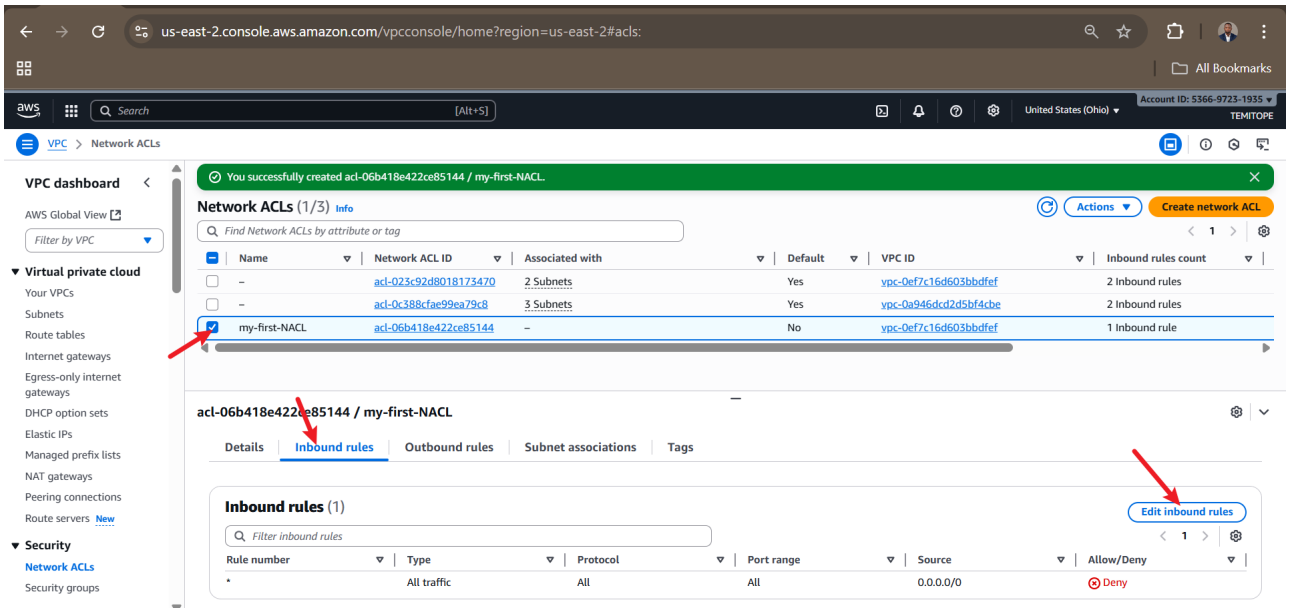


Examine Default Settings

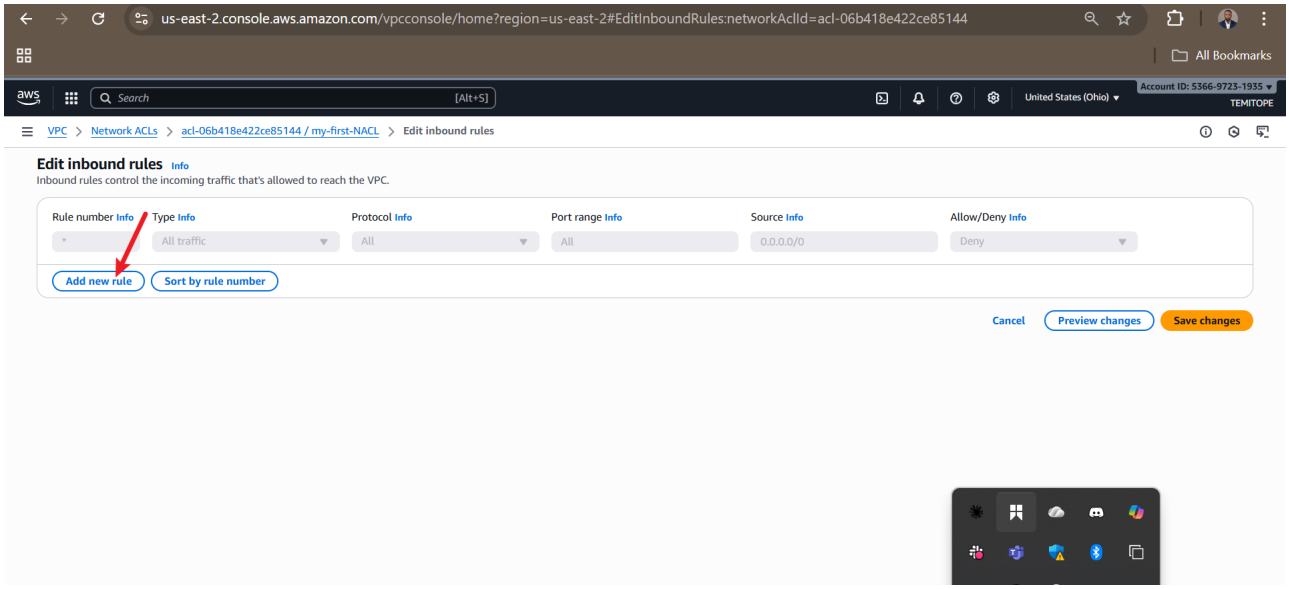
Defaults deny all traffic (as shown above).

Modify Inbound to Permit All

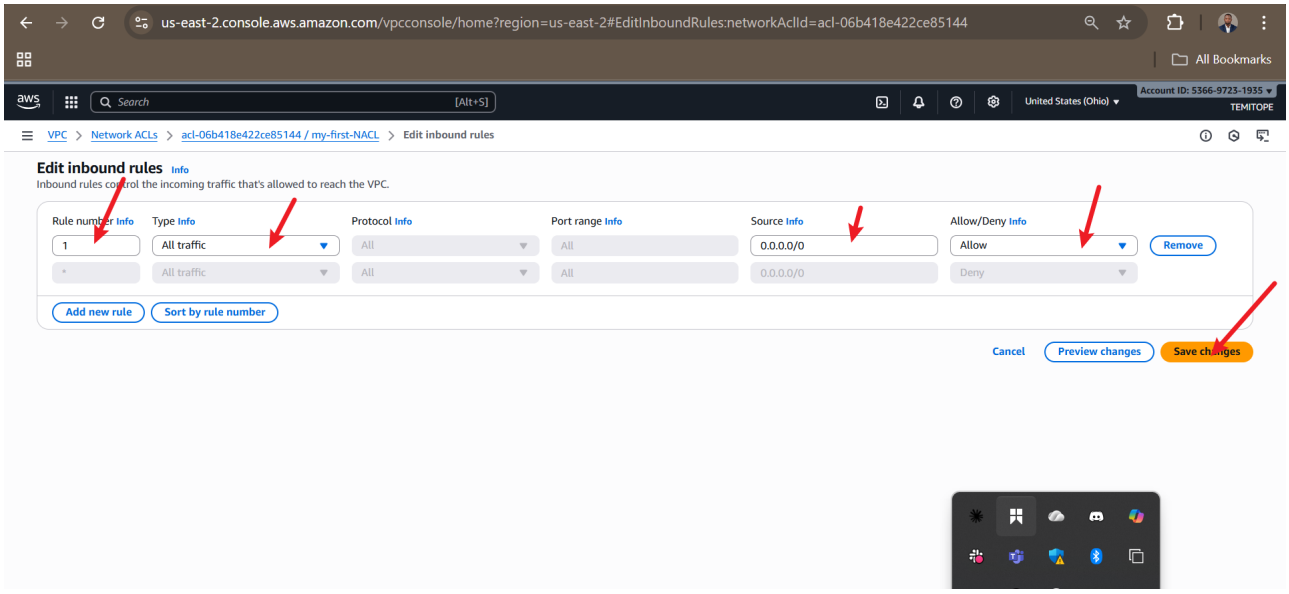
97. Click on "Inbound rules" tab then "Edit inbound rules".



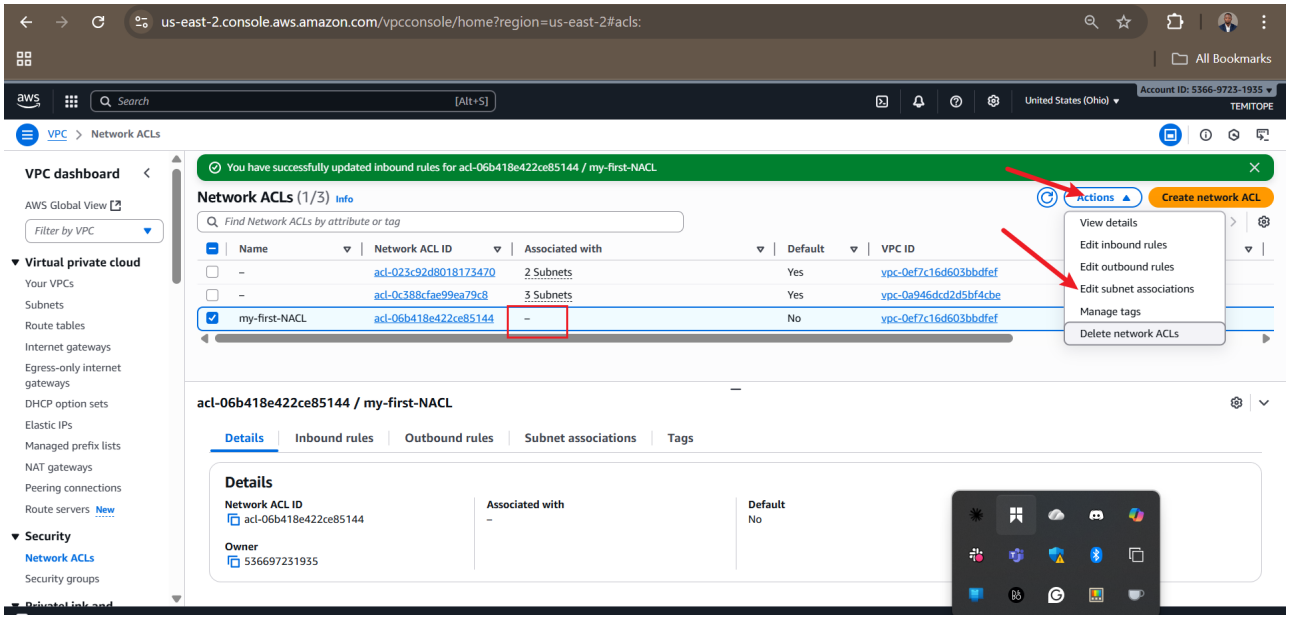
98. Click on "Add new rule".



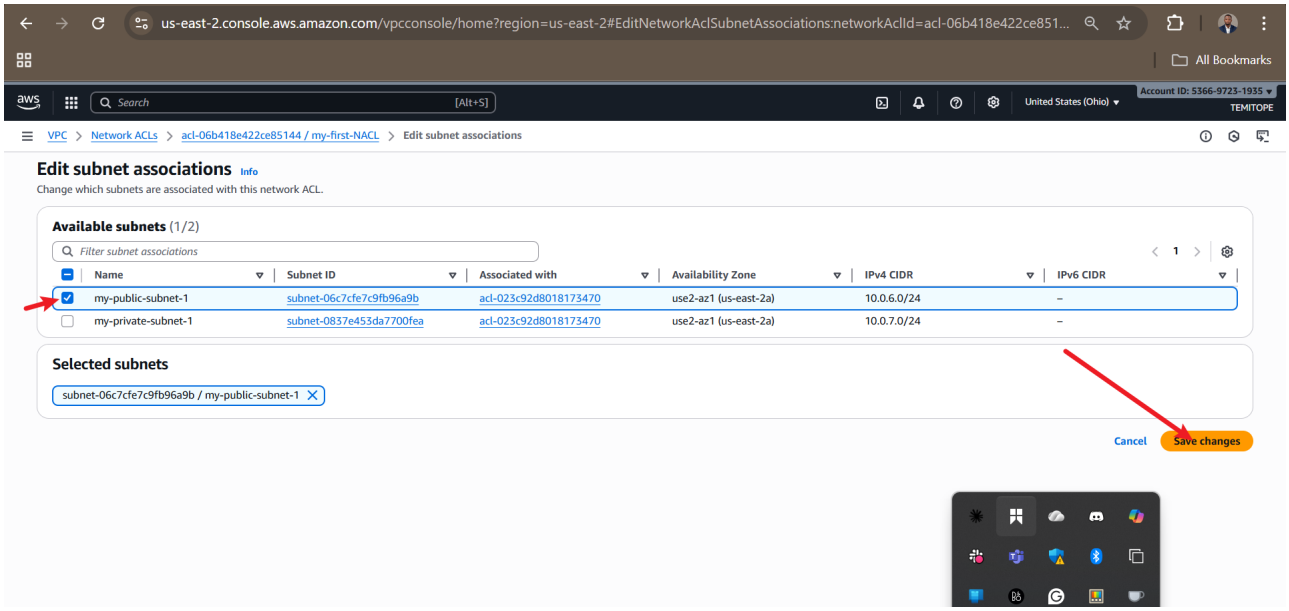
99. Set rule number 1, all traffic, any source, then allow and click "Save changes".



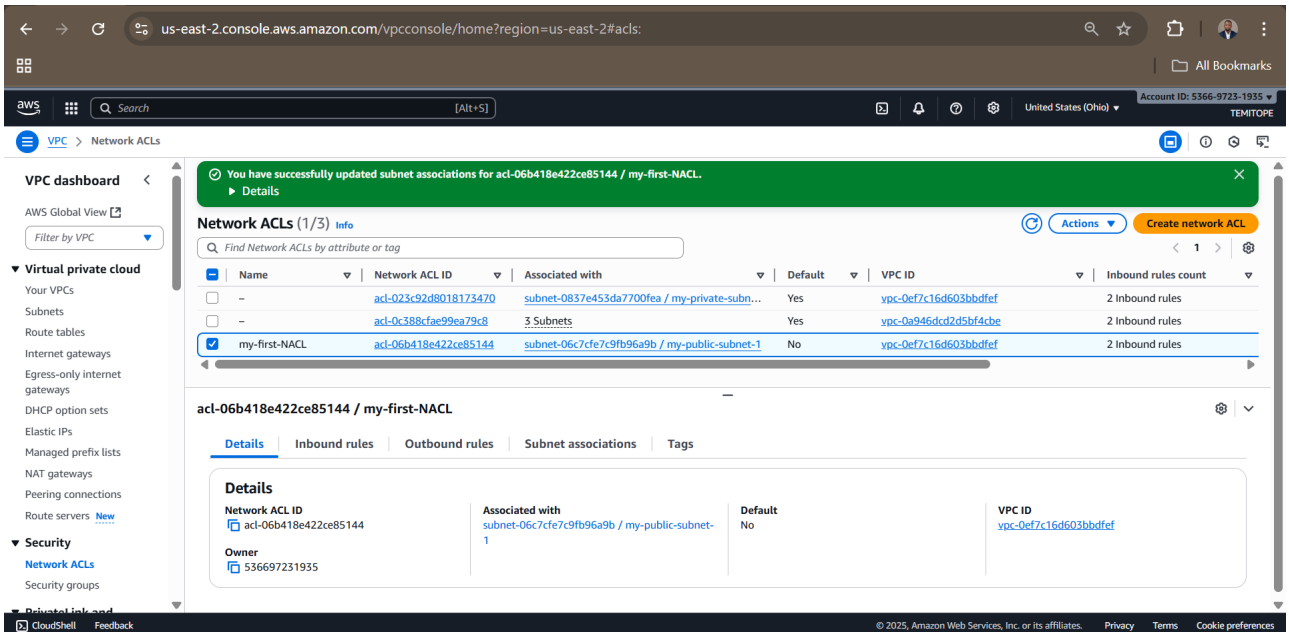
100. Successfully updated inbound rules for NACL; click "Actions" and "Subnet association" to associate NACL with subnet.



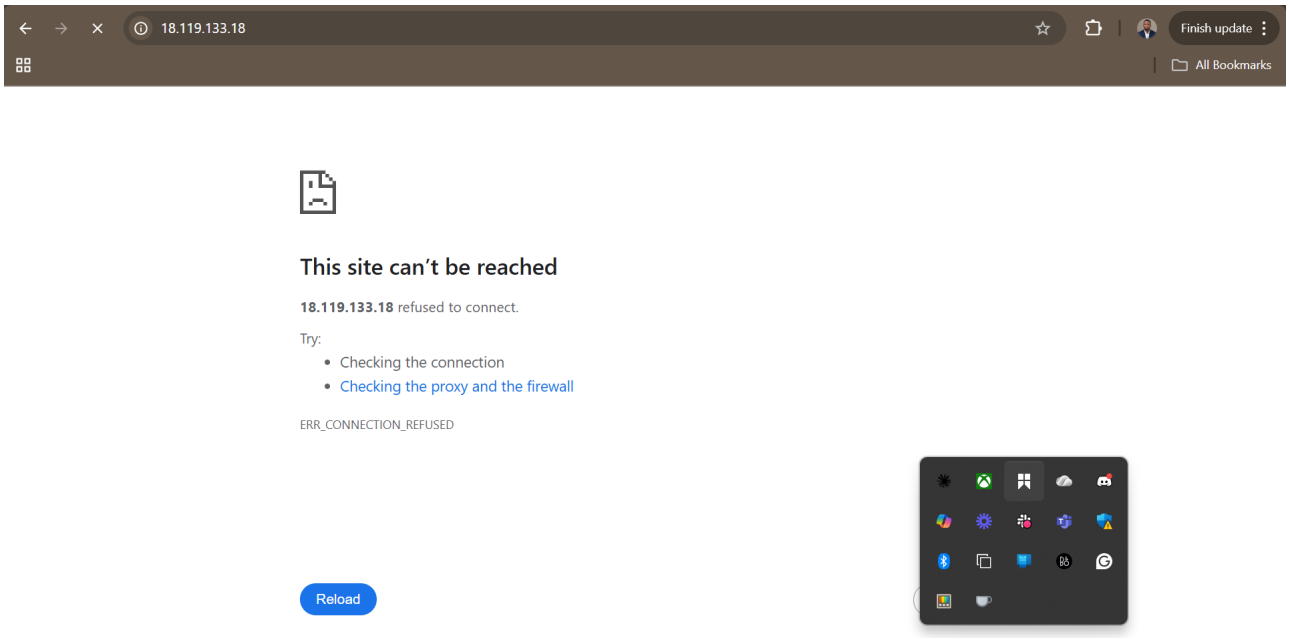
101. On edit subnet page, click public subnet and save changes.



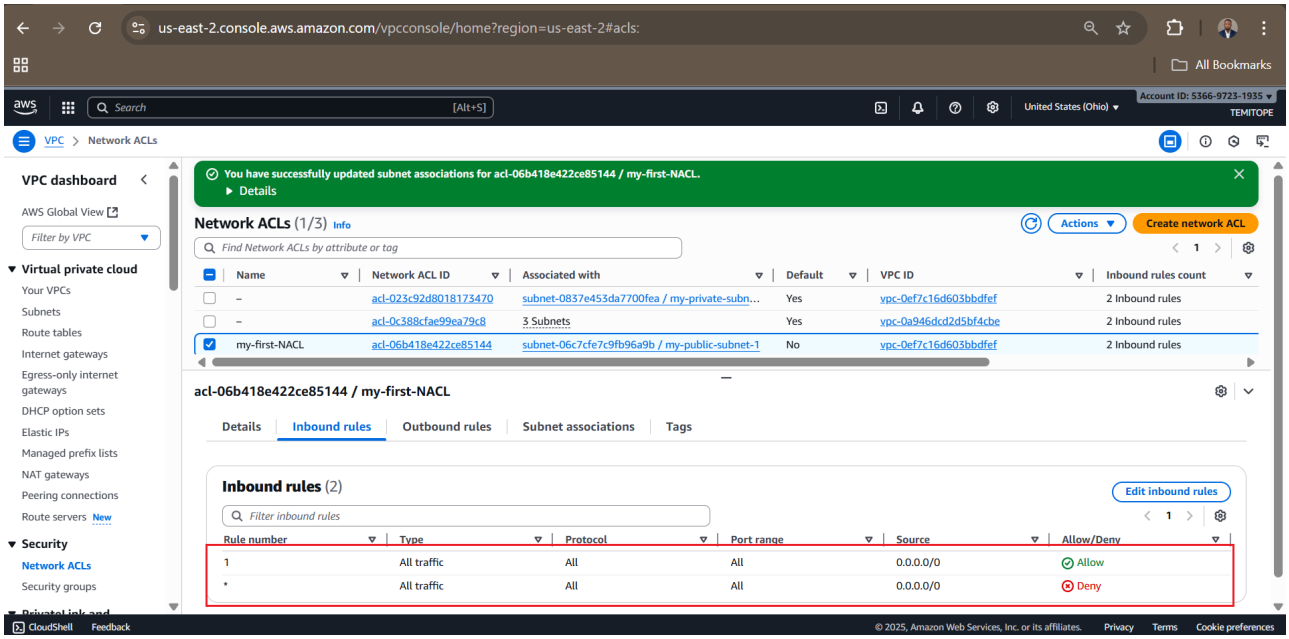
102. Successfully updated subnet association.



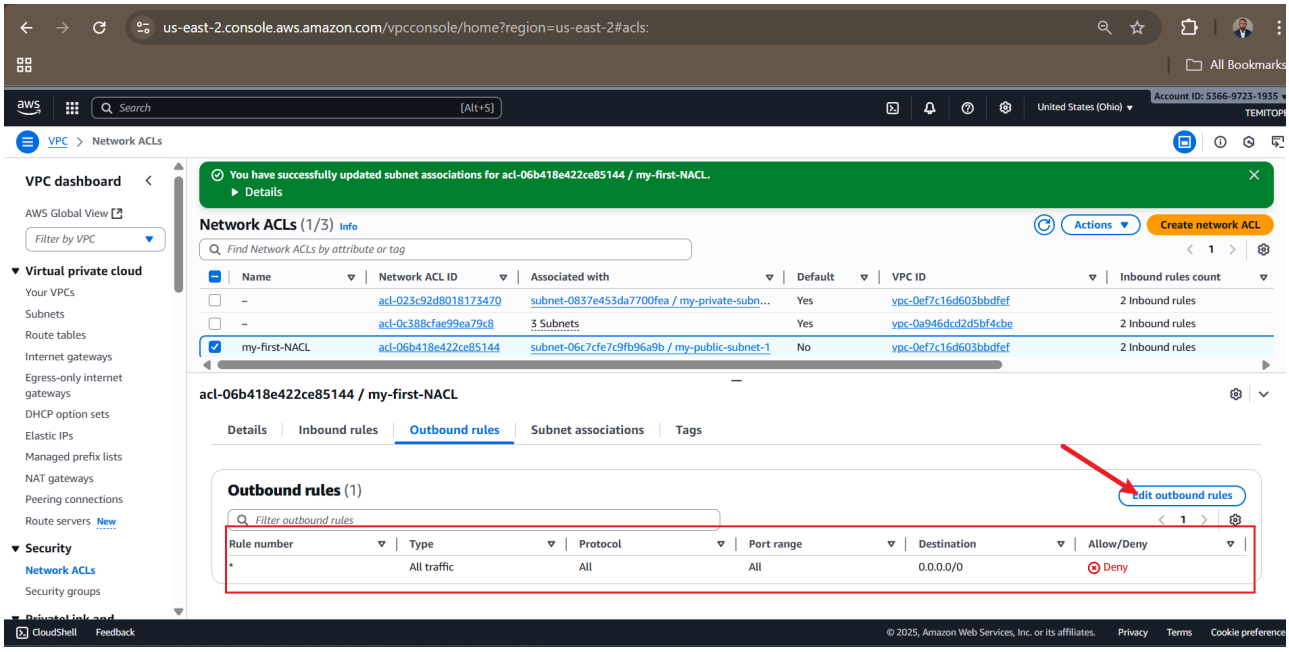
103. Web page unreachable even though we set inbound rules to allow all traffic; this is because NACL is stateless.



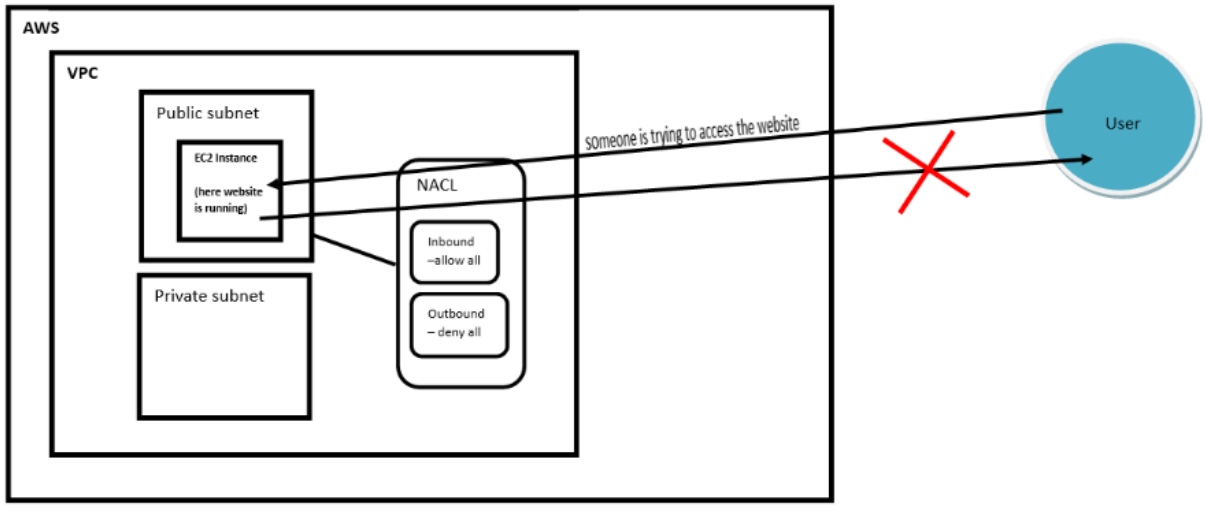
104. Even though the inbound rules allow traffic, the outbound rules deny.



105. Here is outbound rules that denies.



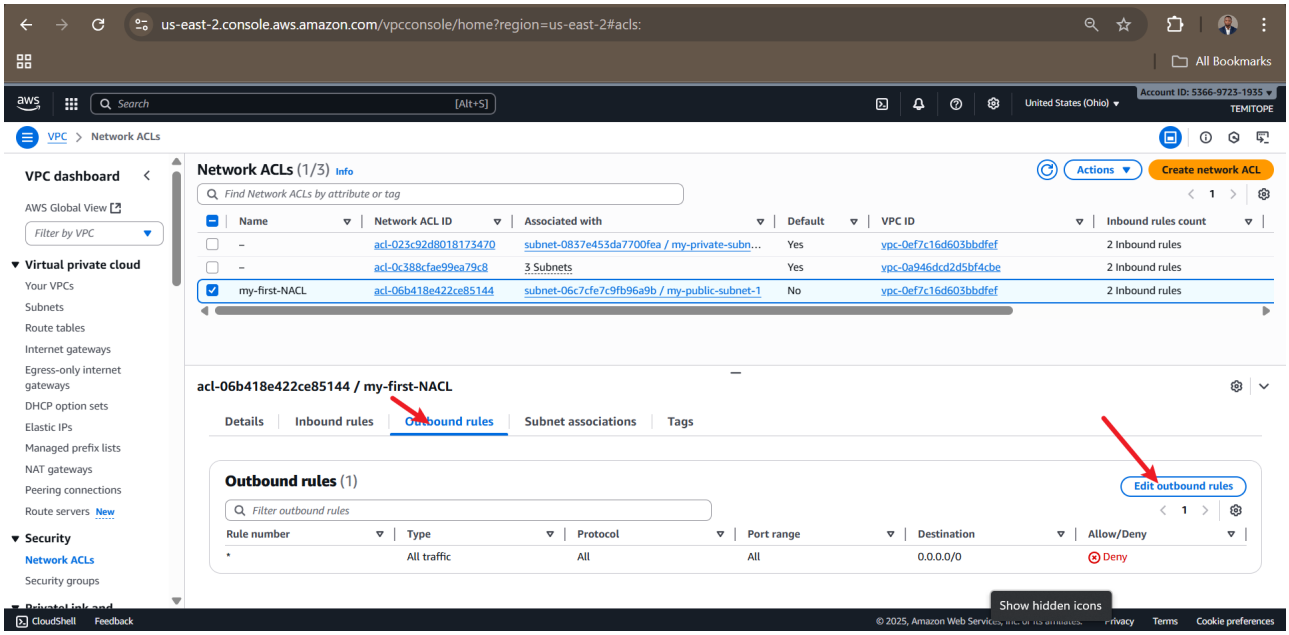
106. Here is the architecture overview that makes the website unviewable.



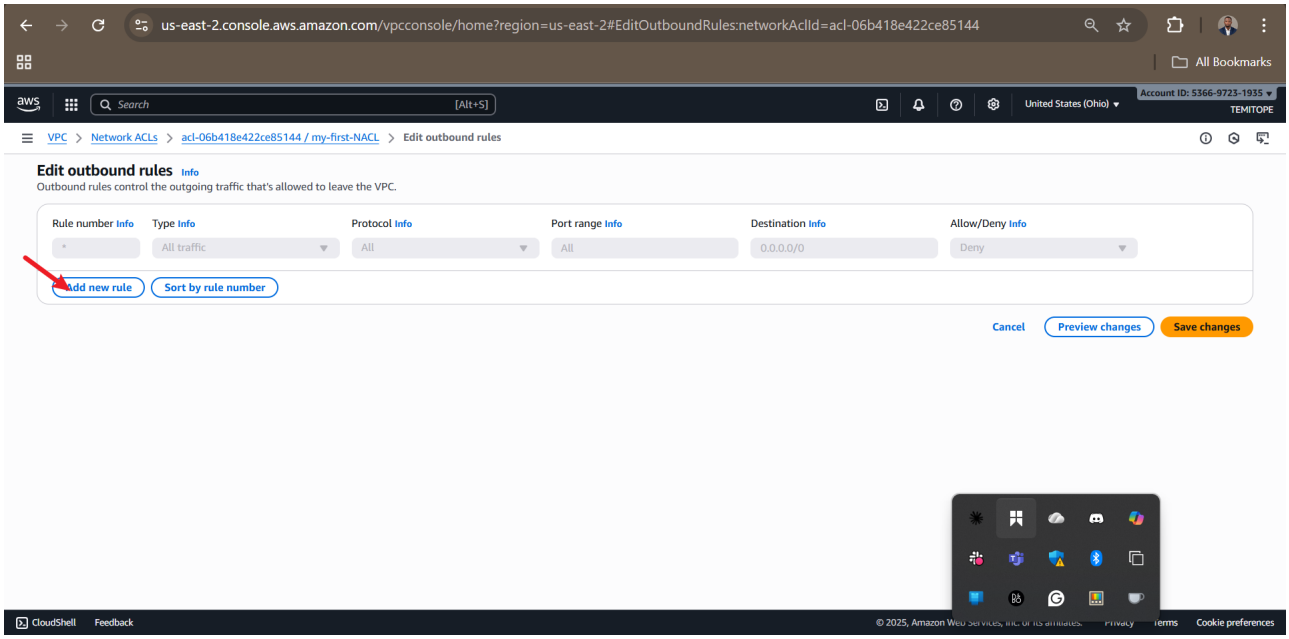
Not able to see website because you are able to go inside of the subnet because of the inbound rule (allow all) but any traffic from subnet is not allowed to go outside due to the outbound rule (deny all).

Adjust Outbound to Allow All

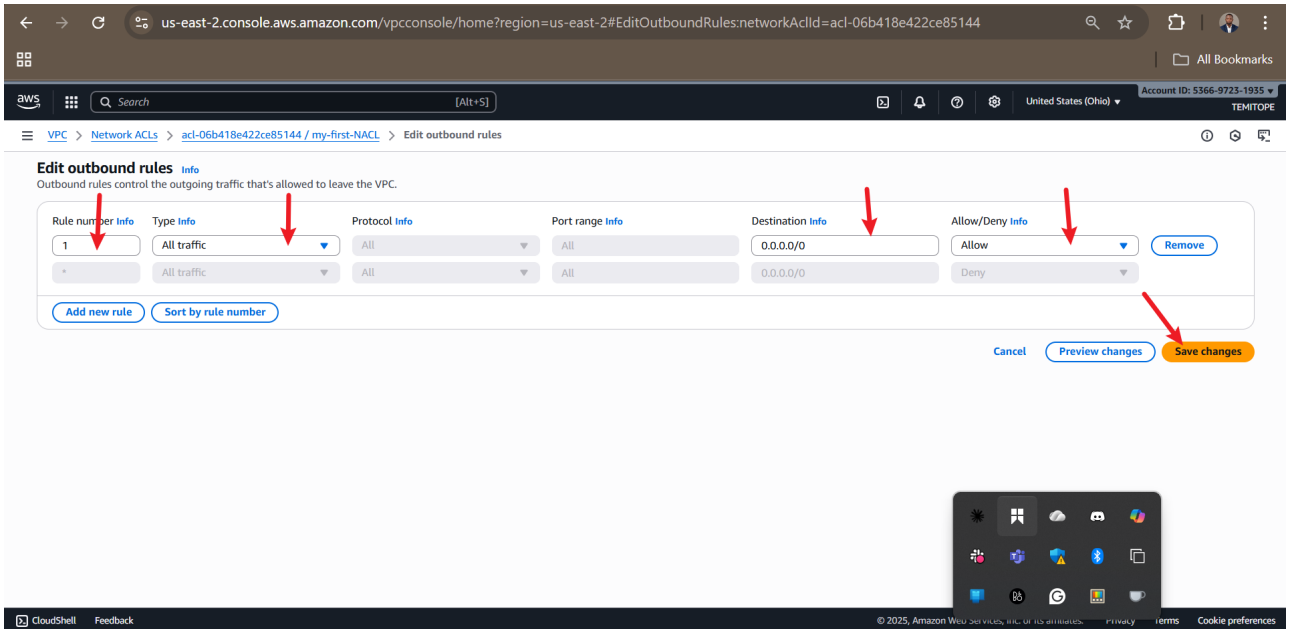
107. Select created NACL, click the "Outbound rules" tab, and edit outbound rules.



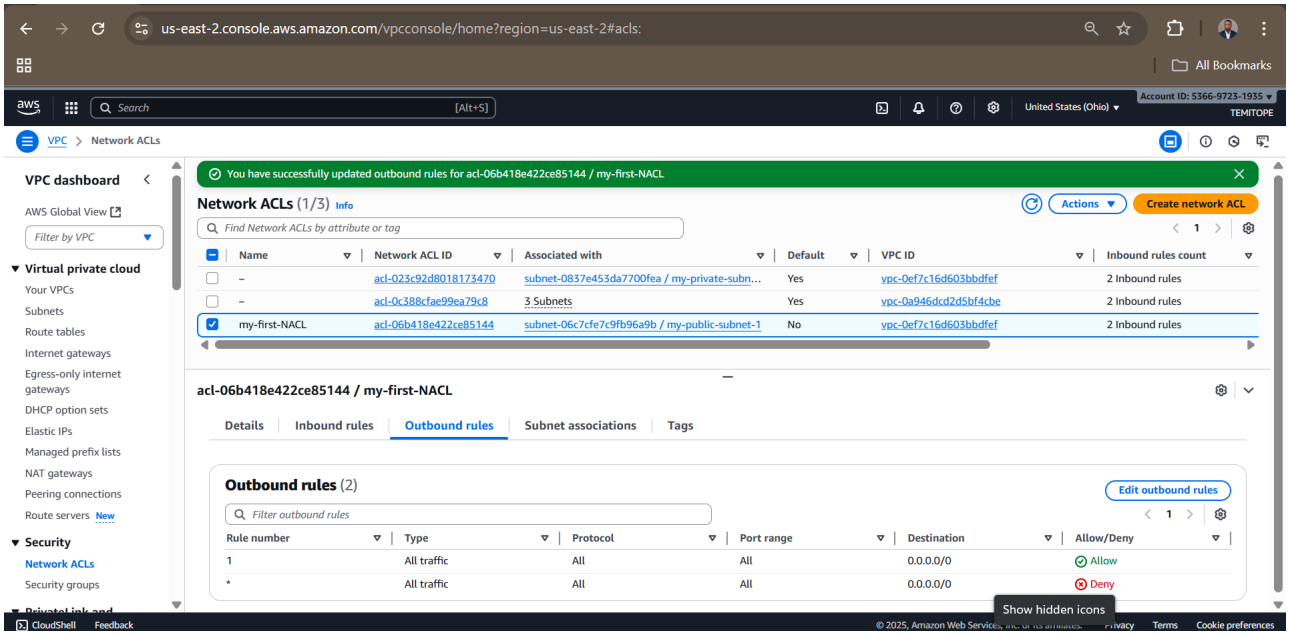
108. Click "Add new rule".



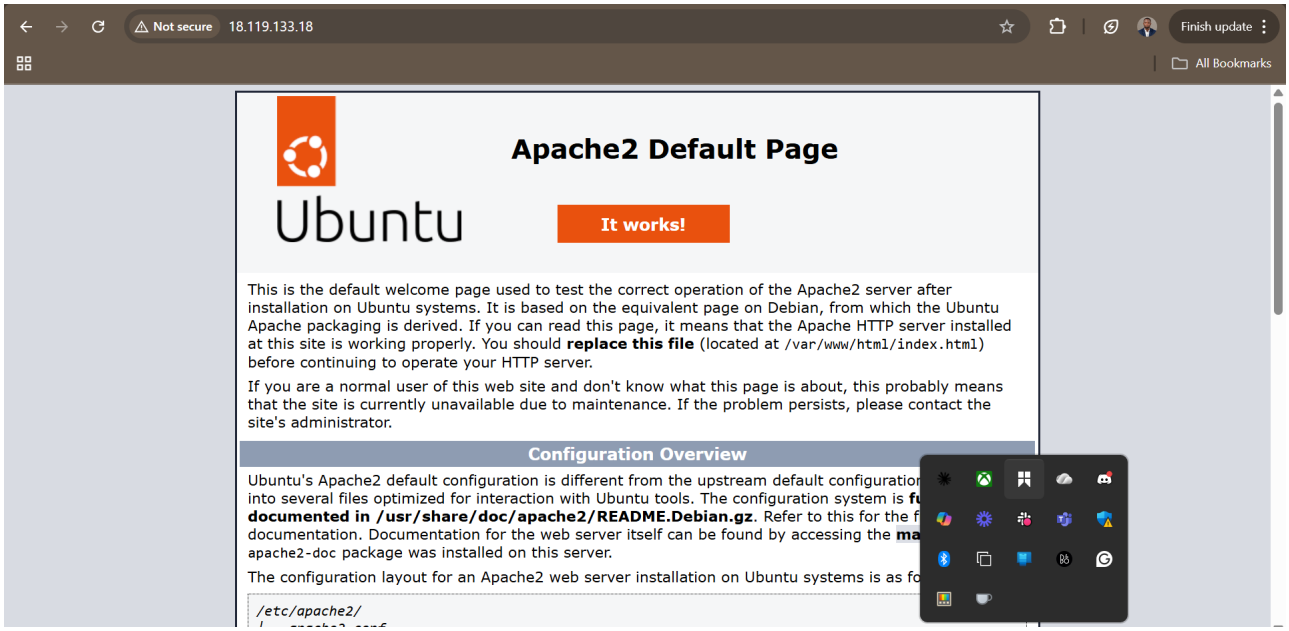
109. Set rule number to 1, all traffic, all destination, allow, save changes.



110. Outbound traffic updated successfully allowing all traffic.

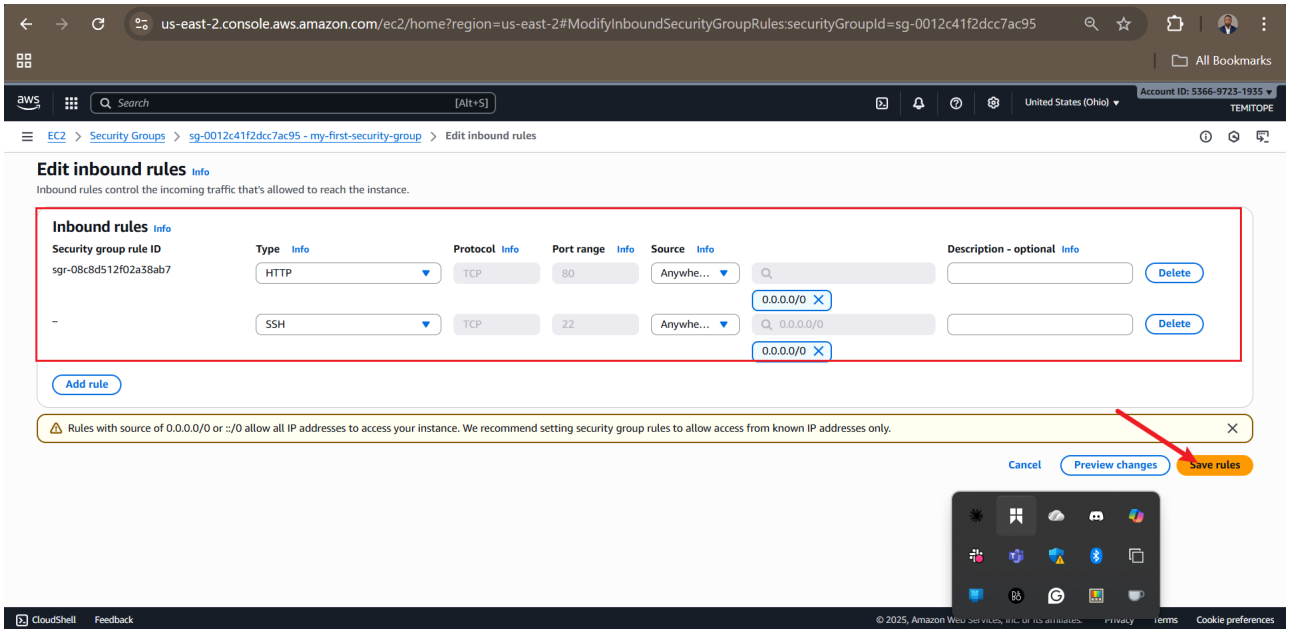


111. Website now reachable due to inbound and outbound rules allowing traffic.

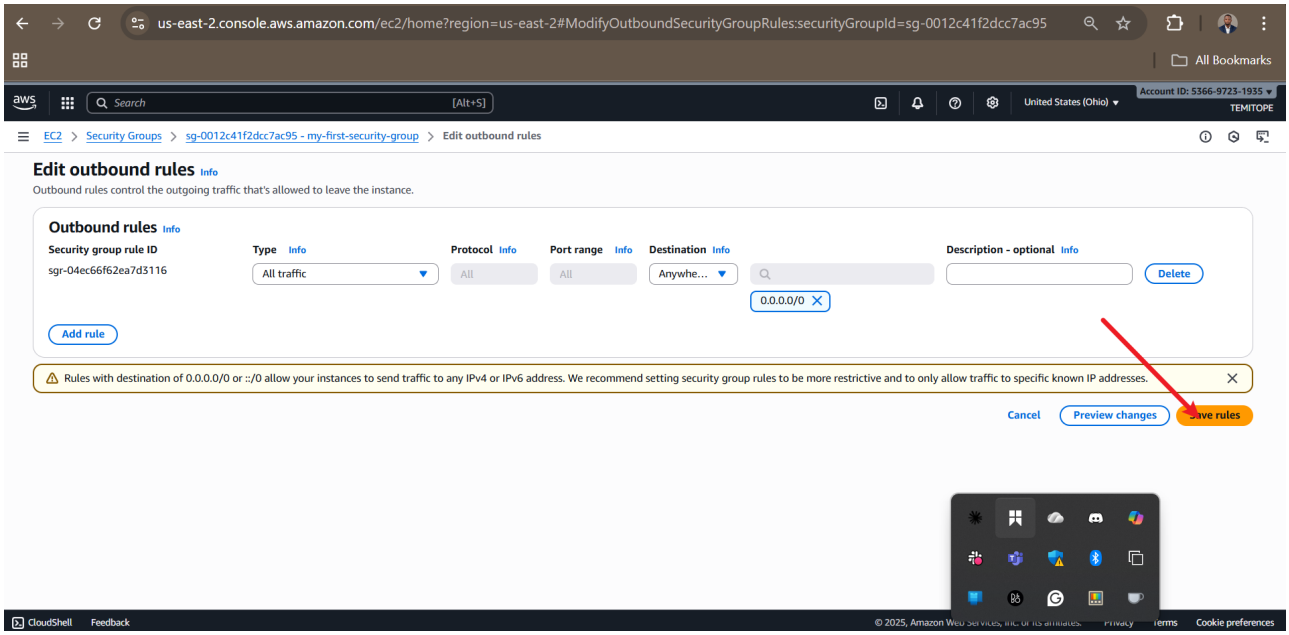


Additional Scenario: Allow All in SG, Deny All in NACL

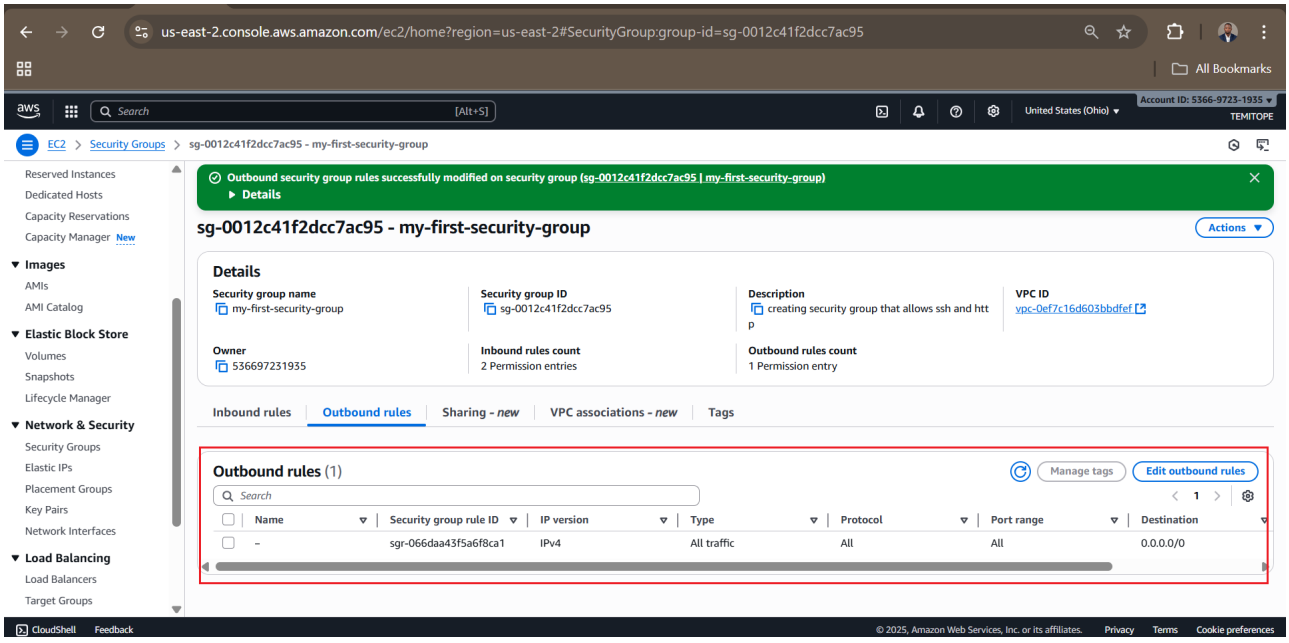
112. In this new scenario, set security group inbound rules to allow HTTP and SSH from anywhere.



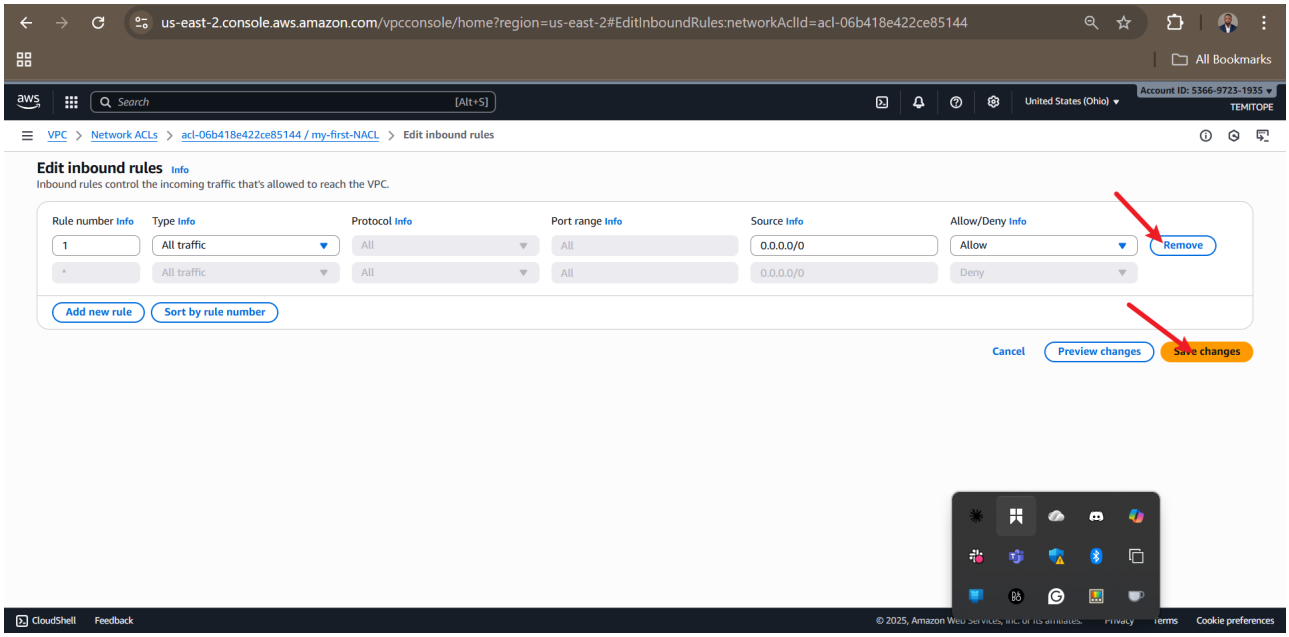
113. Also set security group outbound rules to allow all traffic to anywhere and save rules.



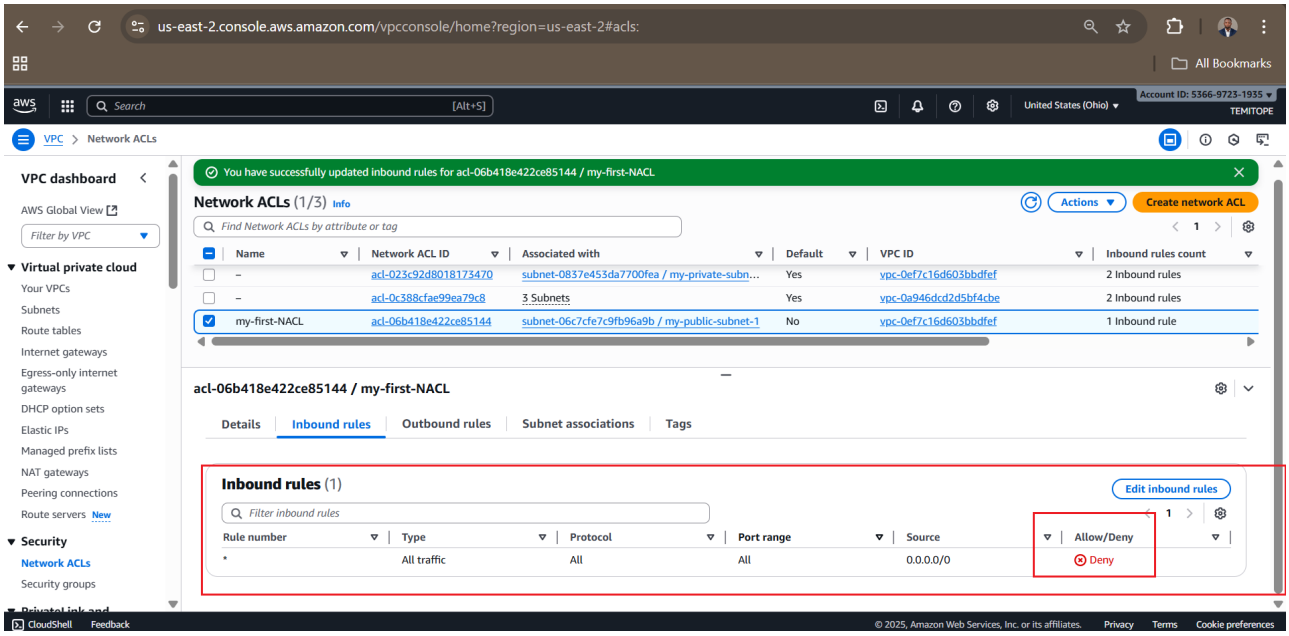
114. All types of outbound traffic now allowed to any destination over the internet.



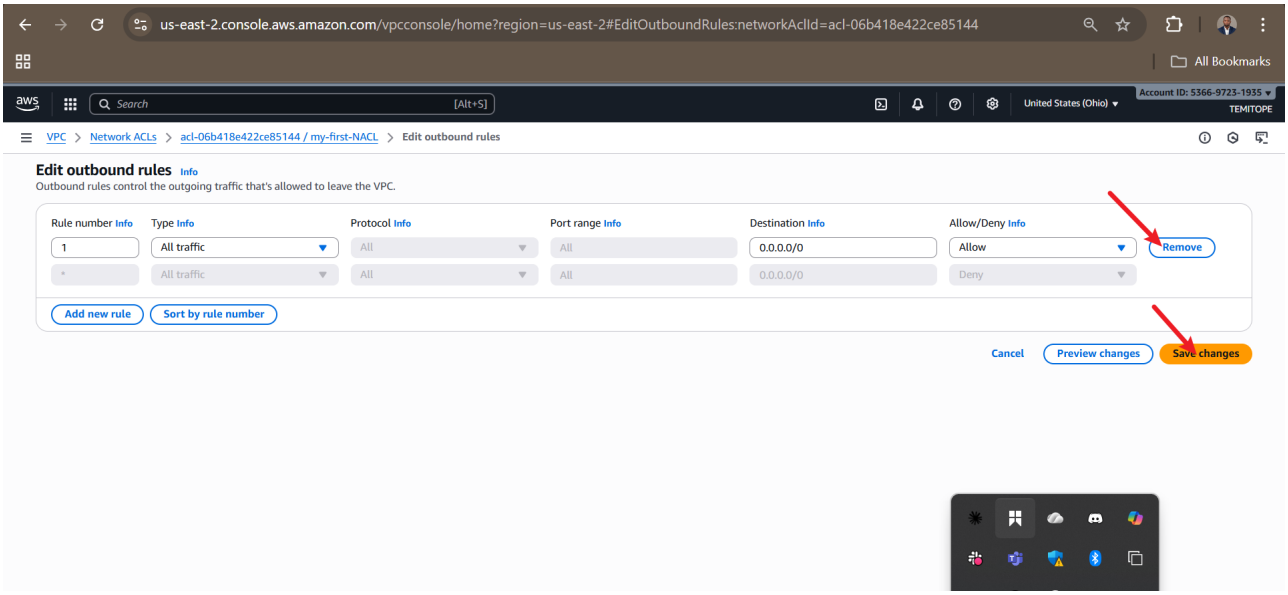
115. Navigate back to NACL, remove NACL inbound rules that allow all traffic, and save rule.



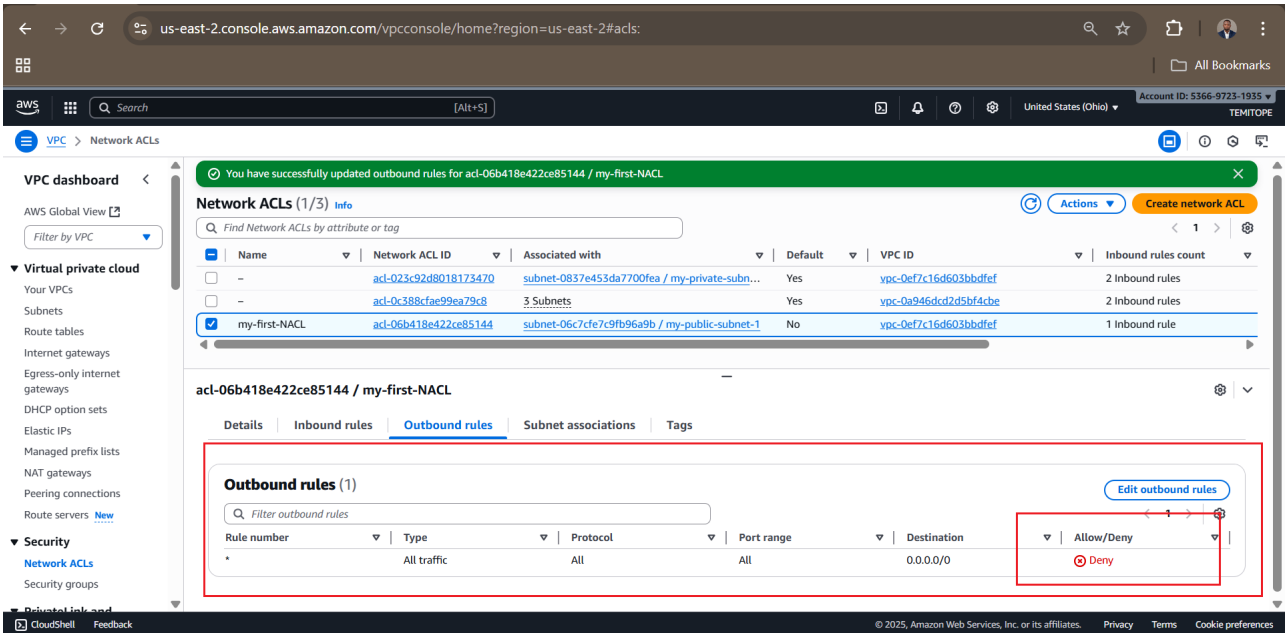
116. NACL inbound rules now deny all traffic types of all protocols on all port from all source.



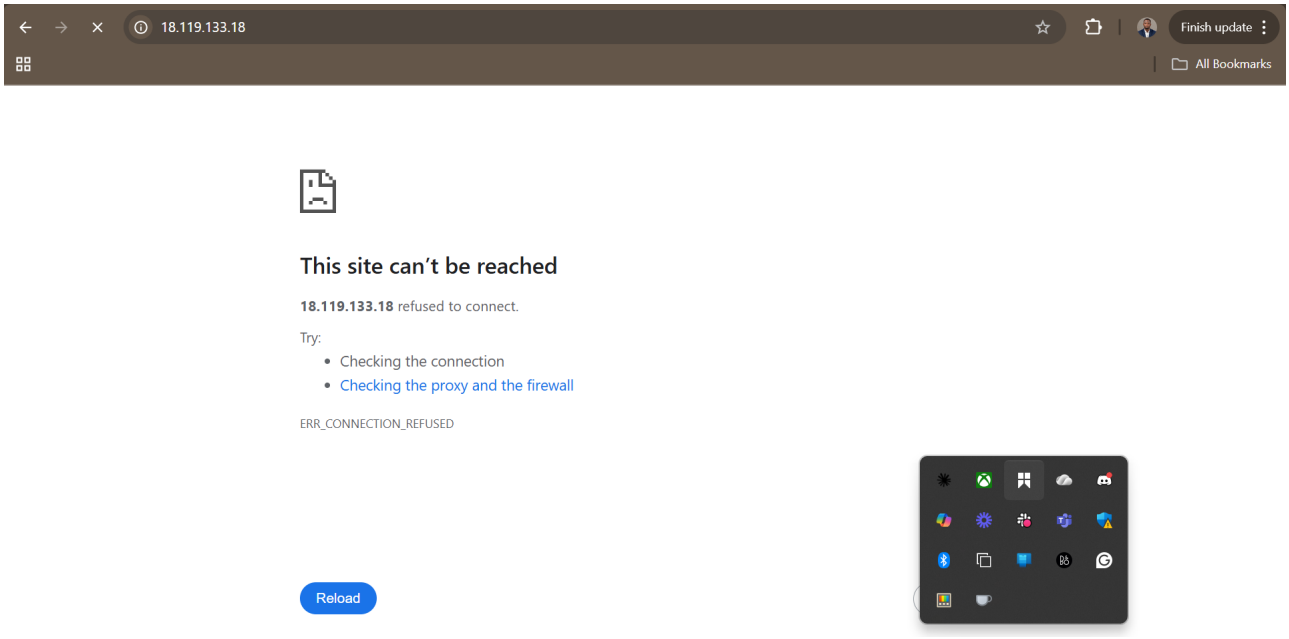
117. Also remove outbound rules that allow all traffic to all destination and save changes.



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



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.