Lab 2: Building your Amazon VPC Infrastructure

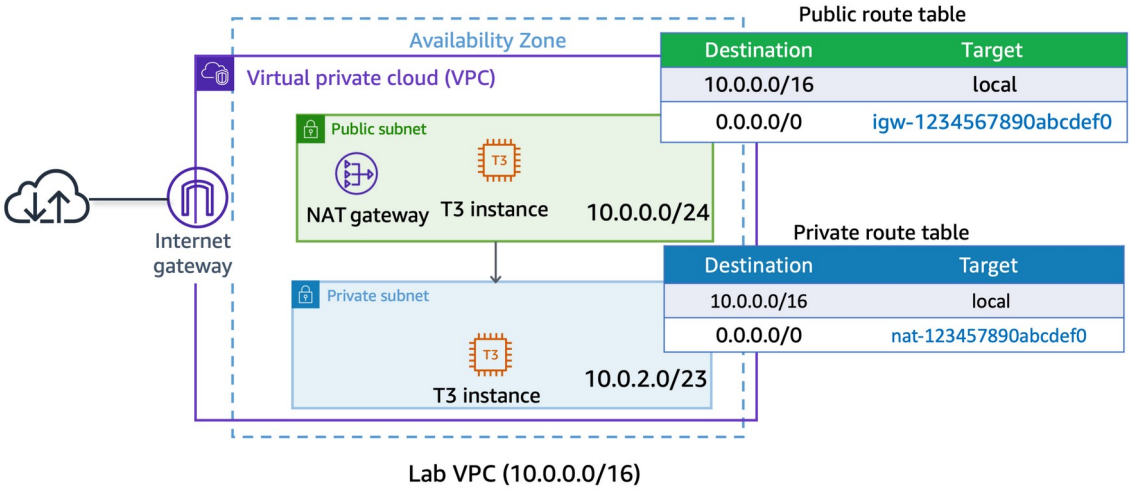
Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at *AWS Training and Certification*.

Lab overview

As an AWS solutions architect, it is important that you understand the overall functionality and capabilities of Amazon Web Service (AWS) and the relationship between the AWS networking components. In this lab, you create an Amazon Virtual Private Cloud (Amazon VPC), a public and a private subnet in a single Availability Zone, public and private routes, a NAT gateway, and an internet gateway. These services are the foundation of networking architecture inside of AWS. This architecture design covers concepts of infrastructure, design, routing, and security.

The following image shows the final architecture for this lab environment:



Objectives

After completing this lab, you should know how to do the following:

- Create an Amazon VPC.
- Create public and private subnets.
- Create an internet gateway.
- Configure a route table and associate it to a subnet.
- Create an Amazon Elastic Compute Cloud (Amazon EC2) instance and make the instance publicly accessible.

- Isolate an Amazon EC2 instance in a private subnet.
- Create and assign security groups to Amazon EC2 instances.
- Connect to Amazon EC2 instances using Session Manager, a capability of AWS Systems Manager.

Start lab

1. To launch the lab, at the top of the page, choose **Start Lab**.

   Caution: You must wait for the provisioned AWS services to be ready before you can continue.

2. To open the lab, choose **Open Console** .

   You are automatically signed in to the AWS Management Console in a new web browser tab.

   Warning: Do not change the **Region** unless instructed.

Common sign-in errors

Error: You must first sign out

If you see the message, **You must first log out before logging into a different AWS account:**

- Choose the **click here** link.
- Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
- Choose **Open Console** again.

Error: Choosing Start Lab has no effect

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

- Add the lab domain name to your pop-up or script blocker's allow list or turn it off.
- Refresh the page and try again.

Scenario

Your team has been tasked with prototyping an architecture for a new web-based application. To define your architecture, you need to have a better understanding of public and private subnets, routing, and Amazon EC2 instance options.

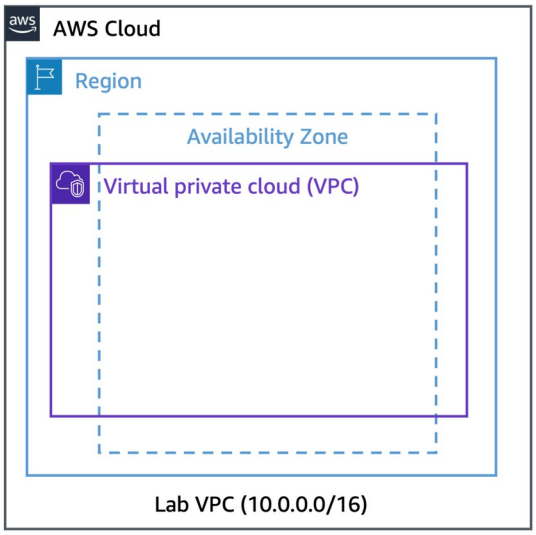
AWS services not used in this lab

AWS services not used in this lab are deactivated in the lab environment. In addition, the capabilities of the services used in this lab are limited to only what the lab requires. Expect errors when accessing other services or performing actions beyond those provided in this lab guide.

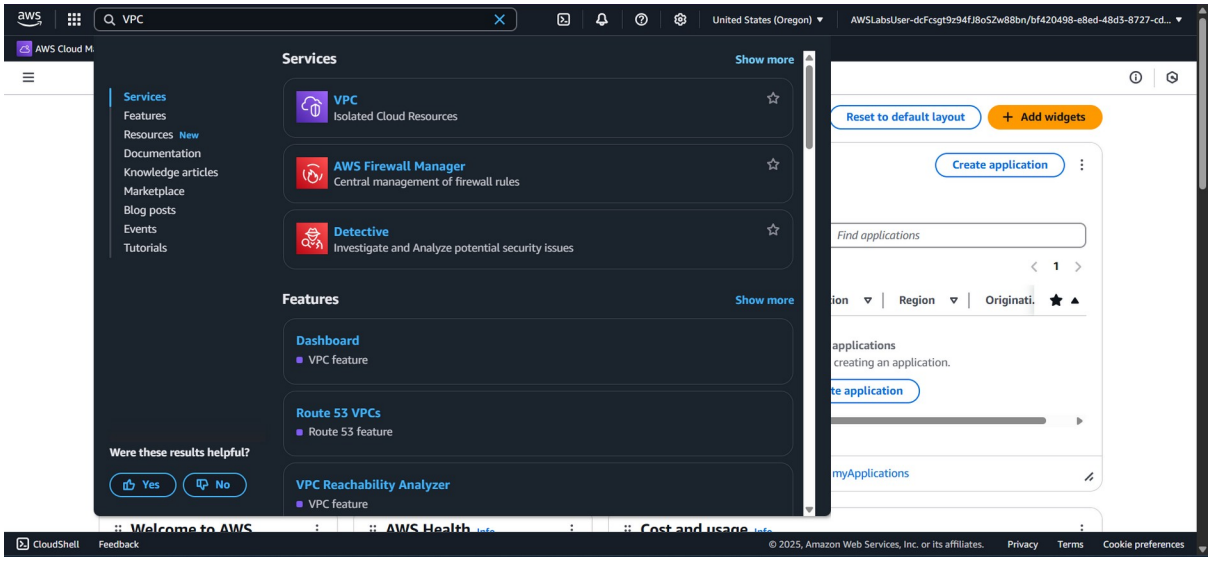
Task 1: Create an Amazon VPC in a Region

In this task, you create a new Amazon VPC in the AWS Cloud.

Learn more: With Amazon VPC, you can provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address ranges, creation of subnets, and configuration of route tables and network gateways. You can also use the enhanced security options in Amazon VPC to provide more granular access to and from the Amazon EC2 instances in your virtual network.



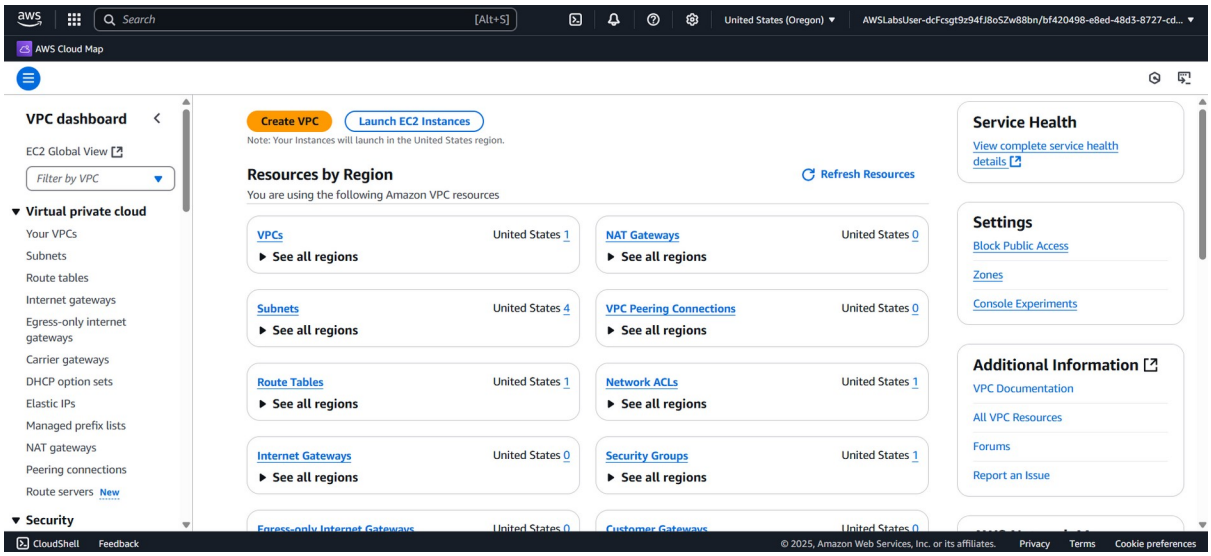
3. At the top of the AWS Management Console, in the search bar, search for and choose VPC.



Caution: Verify that the Region displayed in the top-right corner of the console is the same as the **Region** value on the left side of this lab page.

Note: The VPC management console offers a VPC Wizard, which can automatically create several VPC architectures. However, in this lab you create the VPC components manually.

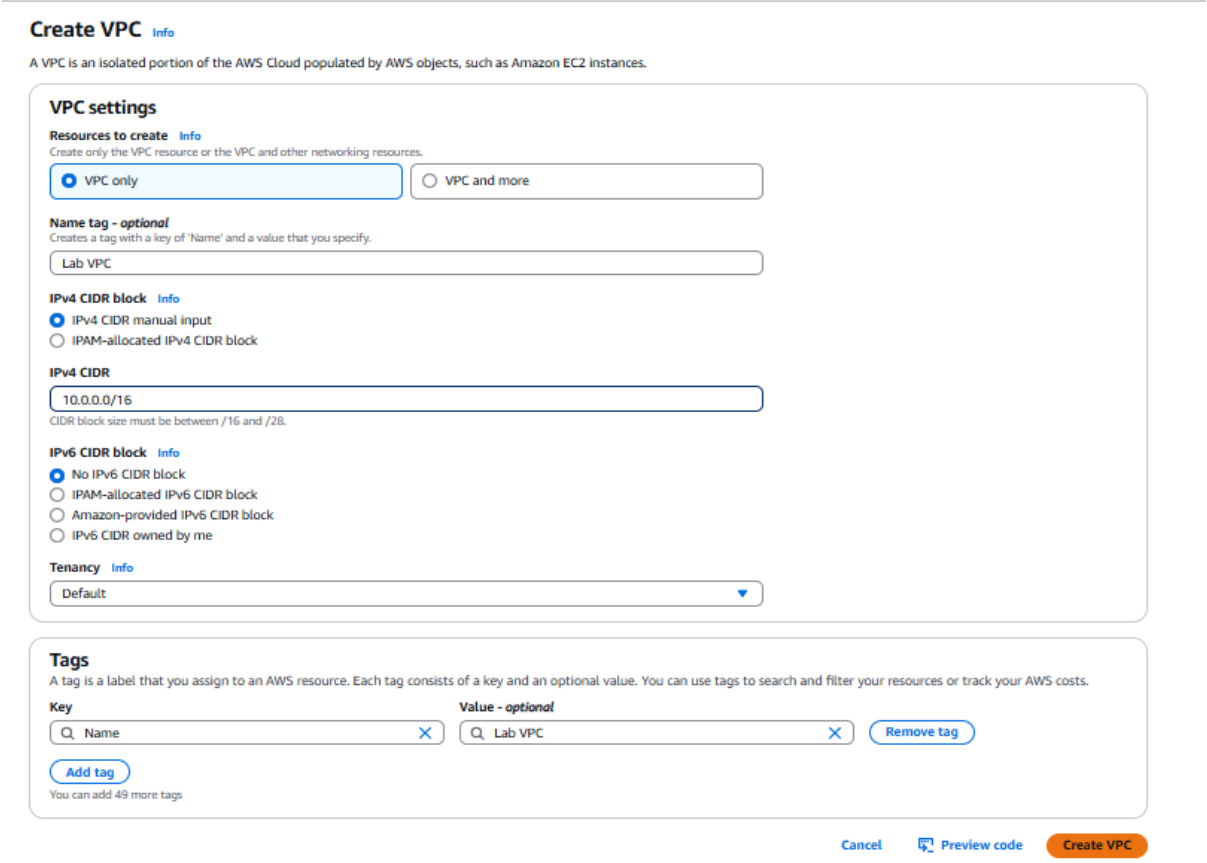
4. In the left navigation pane, choose **Your VPCs**.



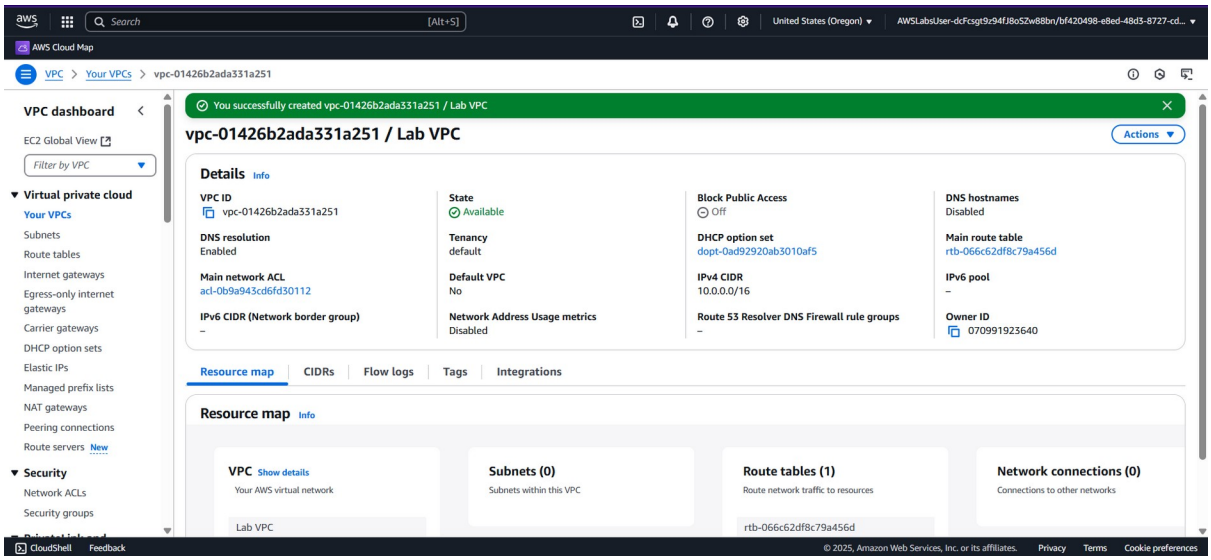
The console displays a list of your currently available VPCs. A default VPC is provided so that you can launch resources as soon as you start using AWS.

5. Choose **Create VPC** and configure the following:

- **Resources to create:** Choose *VPC only*.
- **Name tag - *optional*:** Enter Lab VPC
- **IPv4 CIDR:** Enter 10.0.0.0/16



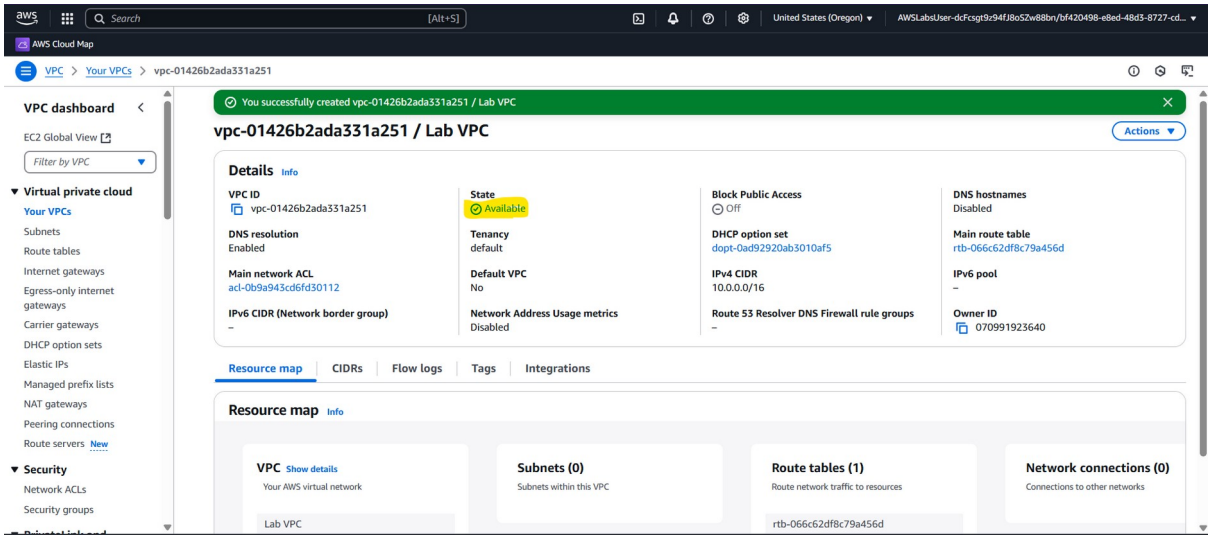
6. Choose **Create VPC**.



A (You successfully created vpc-xxxxxxxxxx / Lab VPC) message is displayed on top of the screen.

The **VPC Details** page is displayed.

7. Verify the state of the **Lab VPC**.

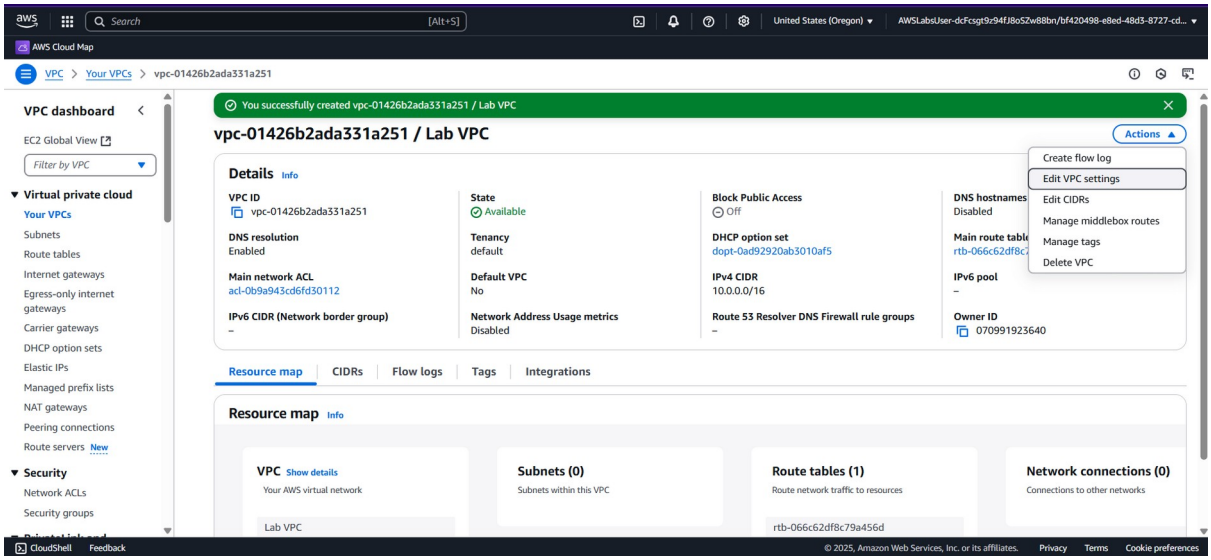


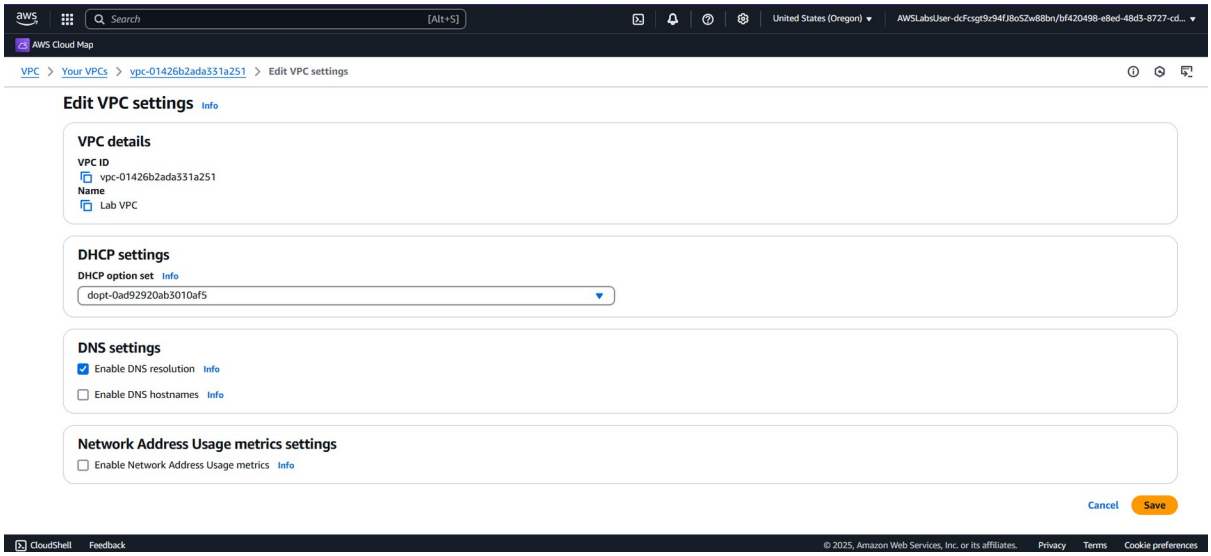
Expected output: It should display the following:

- **State:** Available

The lab VPC has a Classless Inter-Domain Routing (CIDR) range of **10.0.0.0/16**, which includes all IP addresses that start with **10.0.x.x**. This range contains over 65,000 addresses. You later divide the addresses into separate subnets.

8. From the same page, choose **Actions** and choose **Edit VPC settings**.



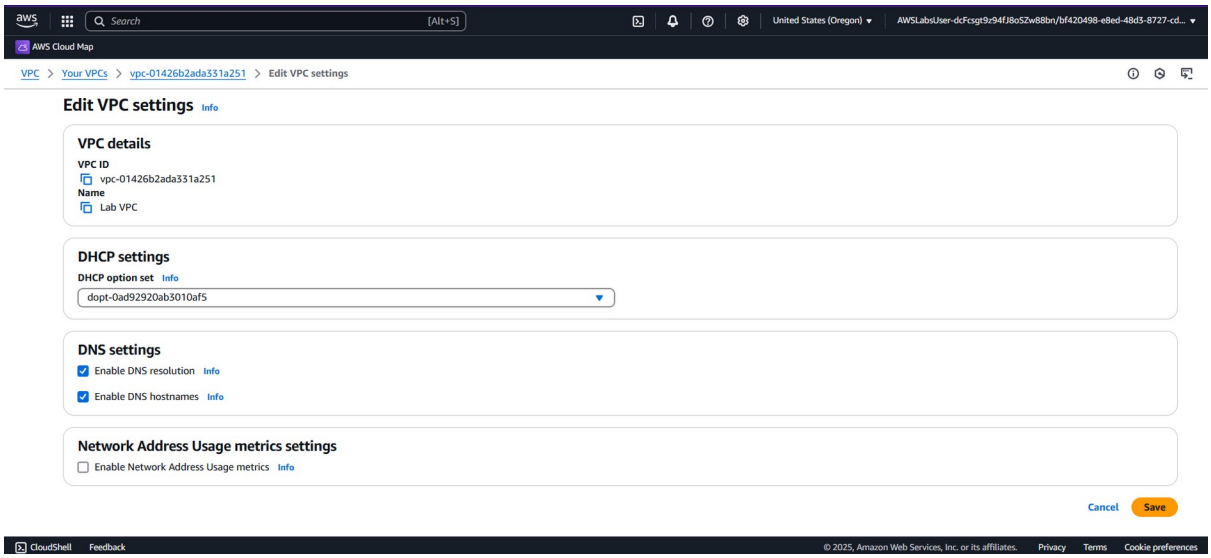


The **Edit VPC settings** page is displayed.

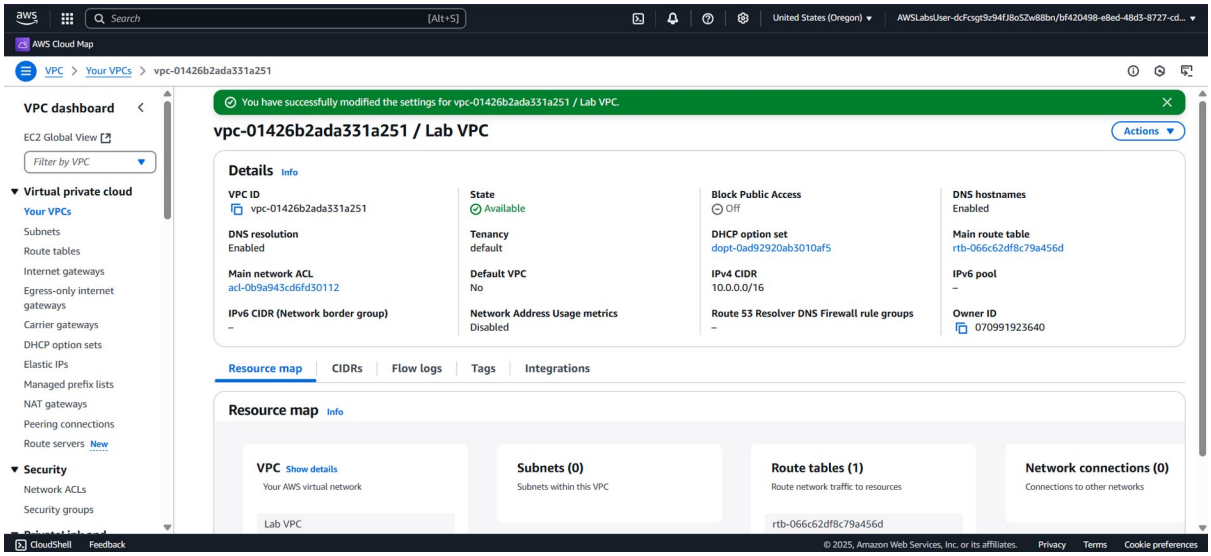
9. From the **DNS settings** section, select **Enable DNS hostnames**.

This option assigns a friendly Domain Name System (DNS) name to Amazon EC2 instances in the VPC, such as the following:

ec2-52-42-133-255.us-west-2.compute.amazonaws.com



10. Choose **Save**.



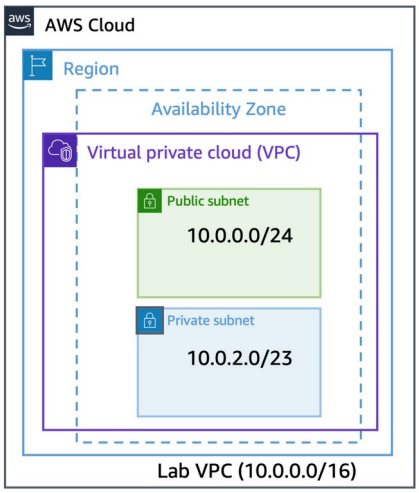
A (You have successfully modified the settings for vpc-xxxxxxxxxx / Lab VPC.) message is displayed on top of the screen.

Any Amazon EC2 instances launched into this Amazon VPC now automatically receive a DNS hostname. You can also create a more meaningful DNS name (for example, *app.company.com*) using records in Amazon Route 53.

Congratulations! You have successfully created your own VPC and now you can launch the AWS resources in this defined virtual network.

Task 2: Create public subnets and private subnets

In this task, you create a public subnet and a private subnet in the lab VPC. To add a new subnet to your VPC, you must specify an IPv4 CIDR block for the subnet from the range of your VPC. You can specify the Availability Zone in which you want the subnet to reside. You can have multiple subnets in the same Availability Zone.

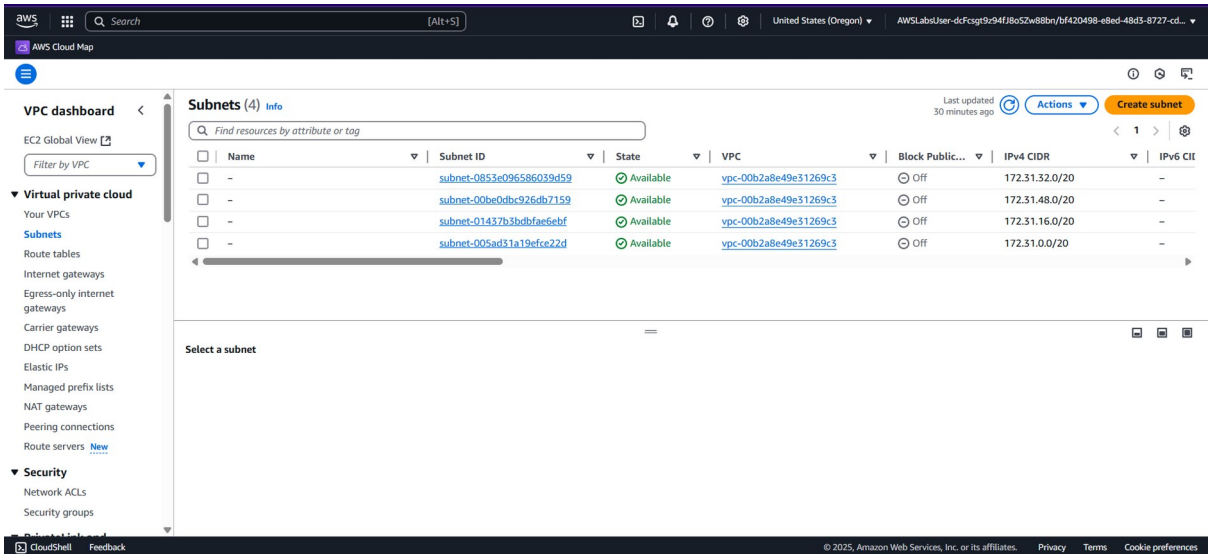


Note: A *subnet* is a sub-range of IP addresses within a network. You can launch AWS resources into a specified subnet. Use a *public subnet* for resources that must be connected to the internet, and use a *private subnet* for resources that are to remain isolated from the internet.

Task 2.1: Create your public subnet

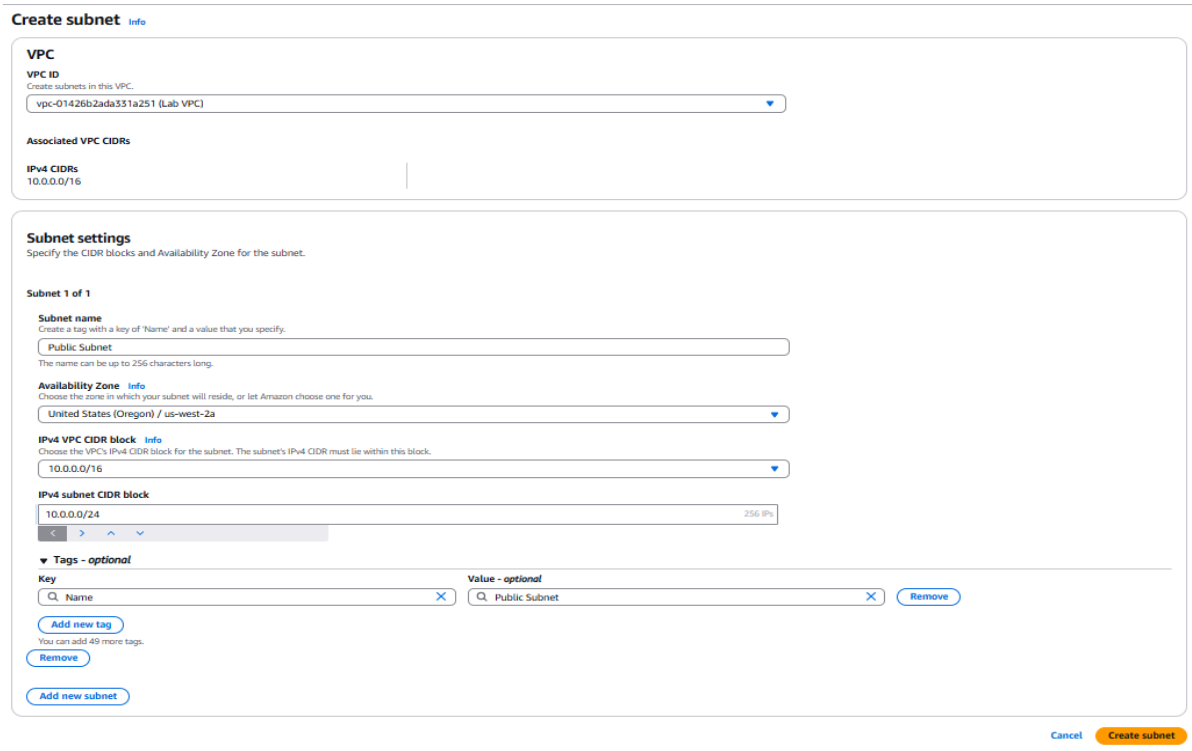
The public subnet is for internet-facing resources.

11. In the left navigation pane, choose **Subnets**.

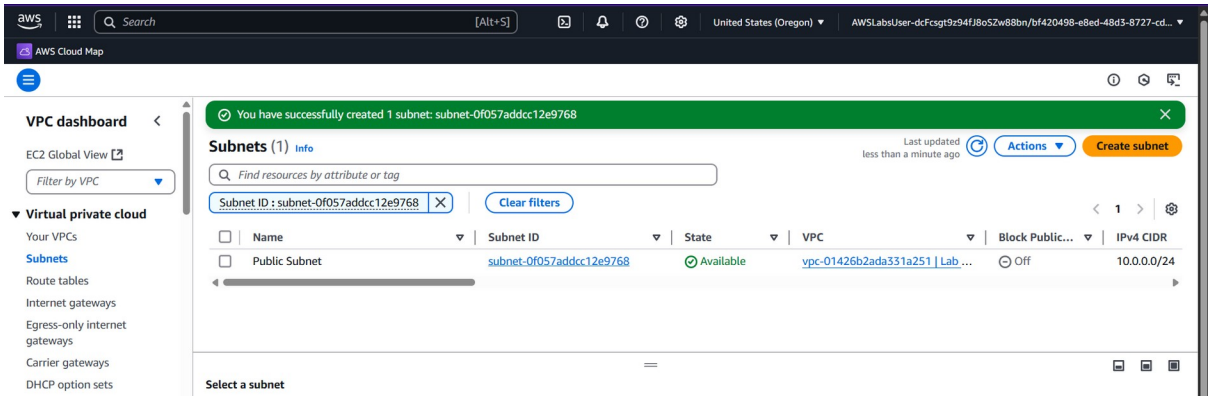


12. Choose **Create subnet** and configure the following:

- **VPC ID:** Select **Lab VPC** from the dropdown menu.
- **Subnet name:** Enter Public Subnet.
- **Availability Zone:** Select the **first** Availability Zone in the list. (Do **not** choose *No Preference*.)
- **IPv4 subnet CIDR block:** Enter 10.0.0.0/24.



13. Choose **Create subnet**.

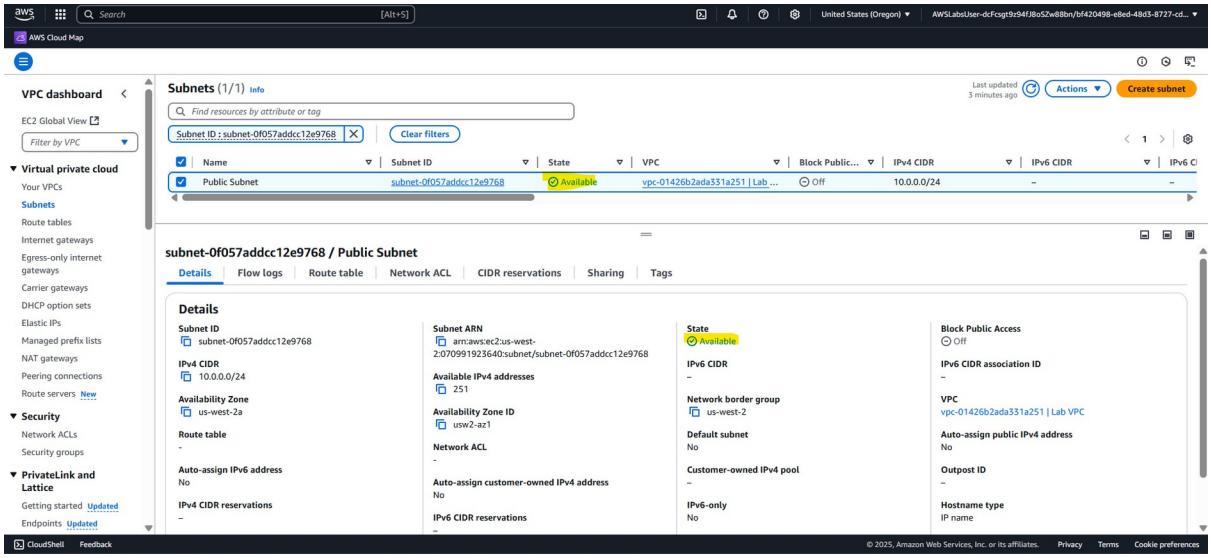


A You have successfully created 1 subnet: subnet-xxxxxx message is displayed on top of the screen.

14. Verify the state.

Expected output: It should display the following:

- **State:** Available

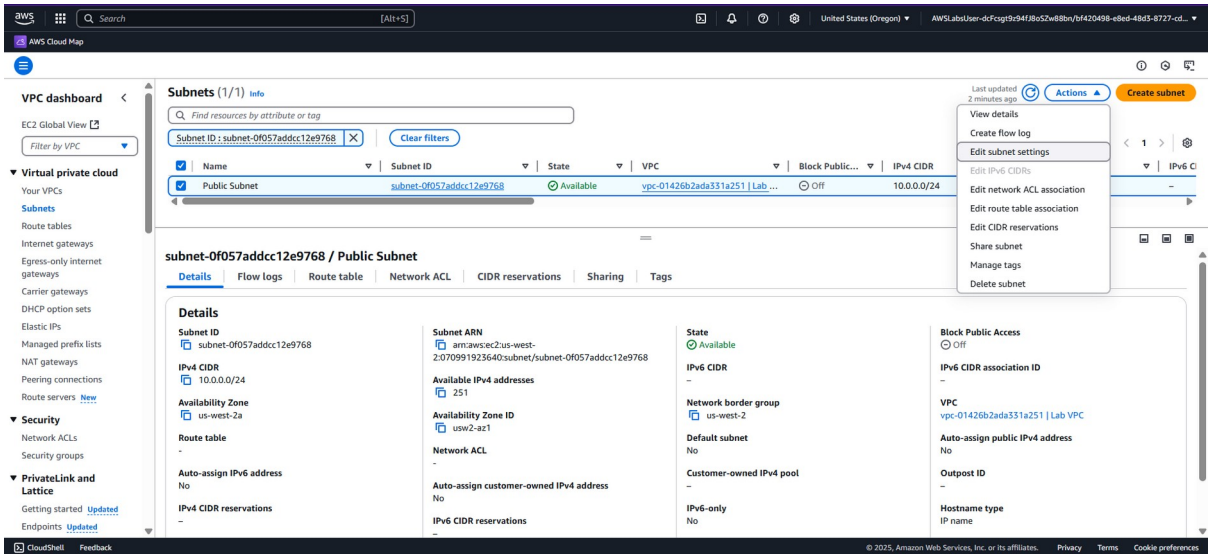


Note: The VPC has a CIDR range of **10.0.0.0/16**, which includes all **10.0.x.x** IP addresses. The subnet you just created has a CIDR range of **10.0.0.0/24**, which includes all **10.0.0.x** IP addresses. These ranges might look similar, but the subnet is smaller than the VPC because of the **/24** in the CIDR range.

Now, configure the subnet to automatically assign a public IP address for all instances launched within it.

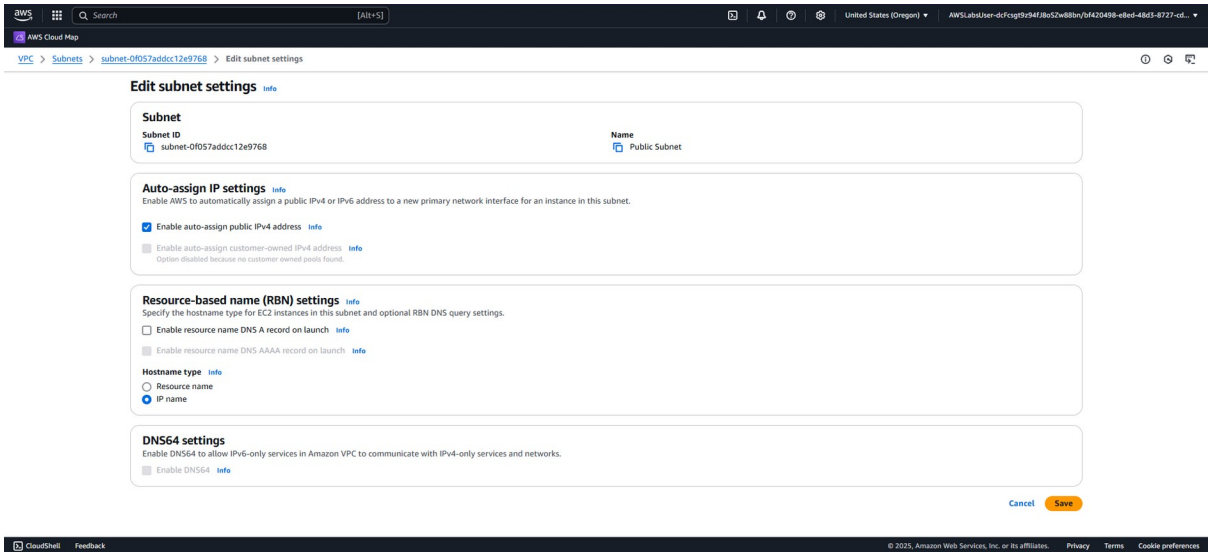
15. Select **Public Subnet**.

16. Choose **Actions** and choose **Edit subnet settings**.

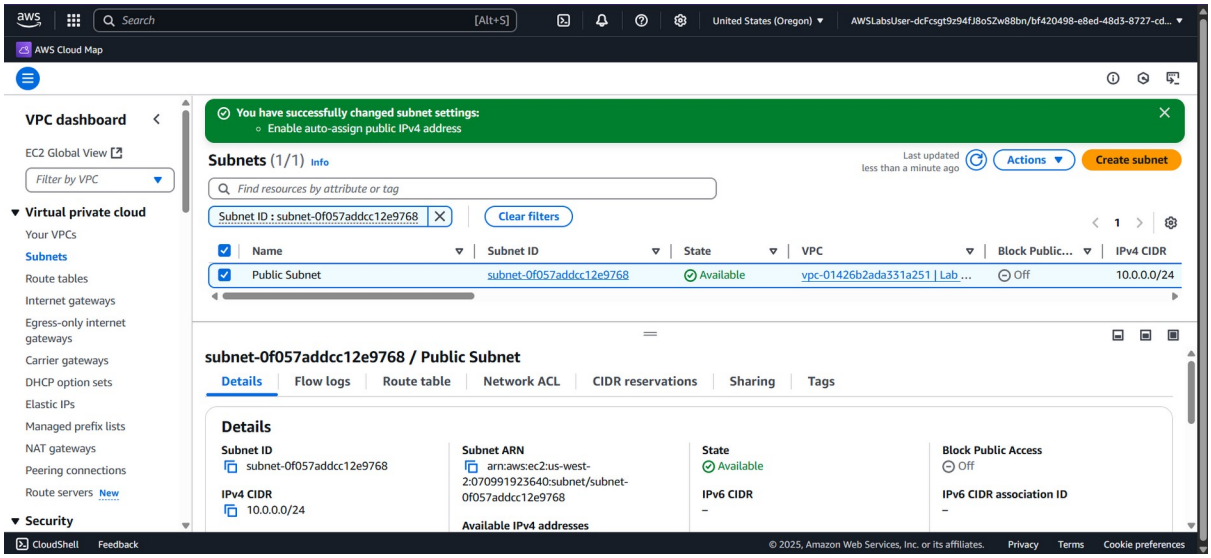


The **Edit subnet settings** page is displayed.

17. From the **Auto-assign IP settings** section, select **Enable auto-assign public IPv4 address**.



18. Choose **Save**.



A (**You have successfully changed subnet settings:** Enable auto-assign public IPv4 address) message is displayed on top of the screen.

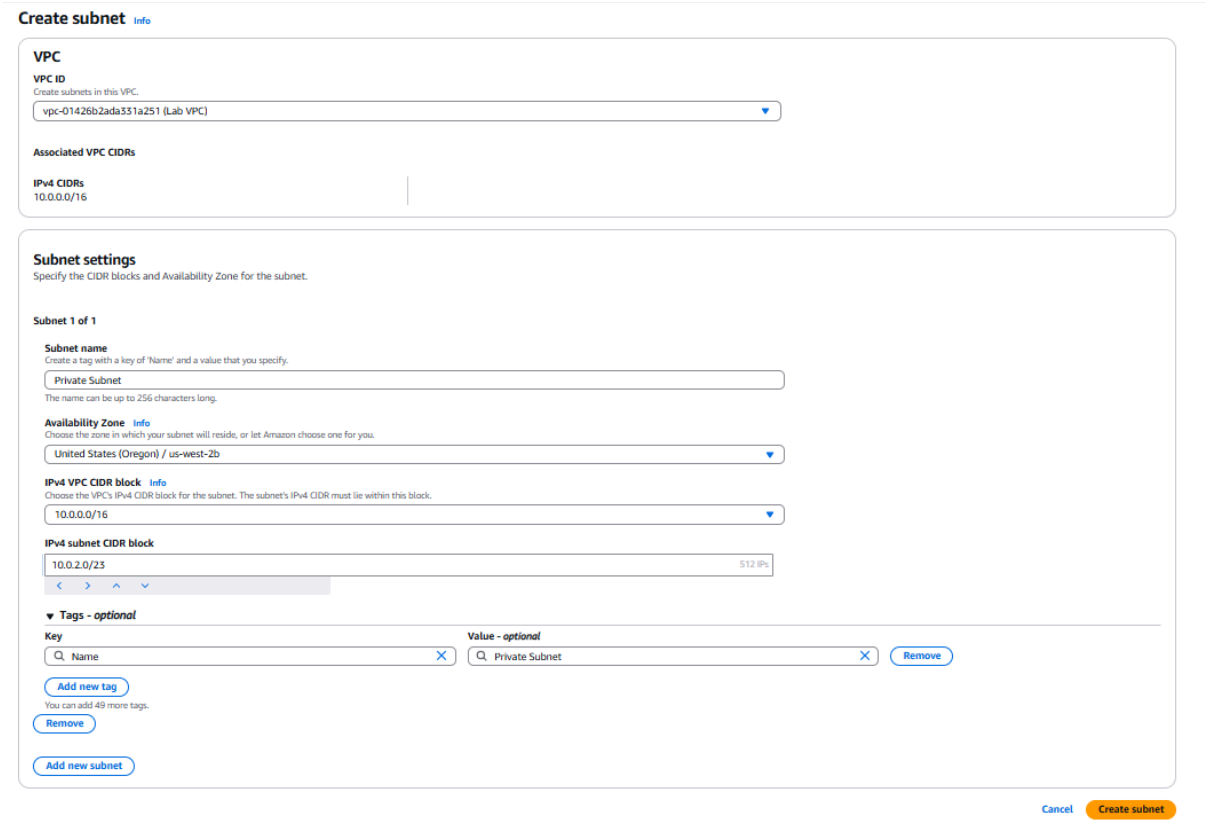
Note: Even though this subnet is named **Public Subnet**, it is not yet public. A public subnet must have an internet gateway and route to the gateway. You create and attach the internet gateway and route tables in this lab.

Task 2.2: Create your private subnet

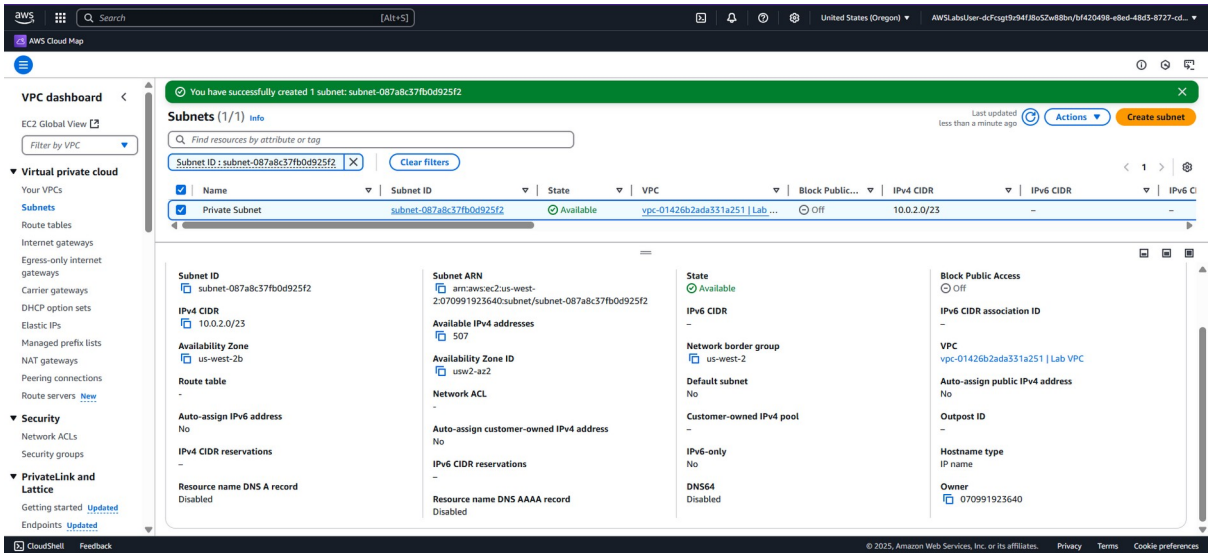
The private subnet is for resources that are to remain isolated from the internet.

19. Choose **Create subnet**, and then configure the following:

- **VPC ID:** Select **Lab VPC** from the dropdown menu.
- **Subnet name:** Enter Private Subnet.
- **Availability Zone:** Select the **first** Availability Zone in the list. (Do **not** choose *No Preference*.)
- **IPv4 subnet CIDR block:** Enter 10.0.2.0/23.



20. Choose **Create subnet**.

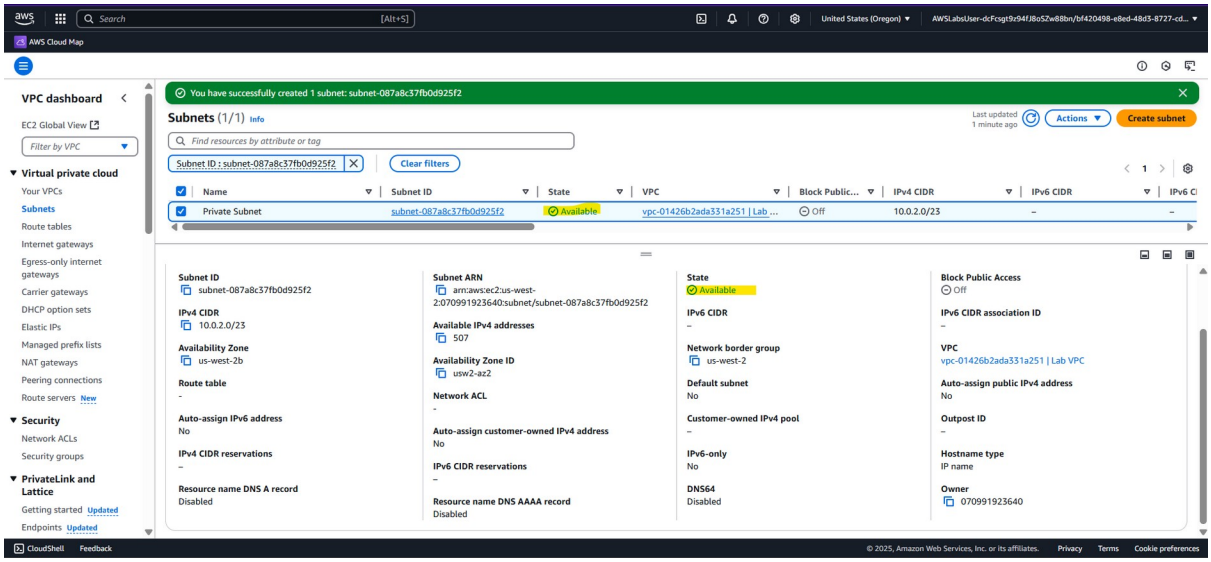


A (You have successfully created 1 subnet: subnet-xxxxxx) message is displayed on top of the screen.

21. Verify the state.

Expected output: It should display the following:

- **State:** Available



Note: The CIDR block of **10.0.2.0/23** includes all IP addresses that start with **10.0.2.x** and **10.0.3.x**. This is twice as large as the public subnet because most resources should be kept private, unless they specifically need to be accessible from the internet.

Your VPC now has two subnets. However, these subnets are isolated and cannot communicate with resources outside the VPC. Next, you configure the public subnet to connect to the internet through an internet gateway.

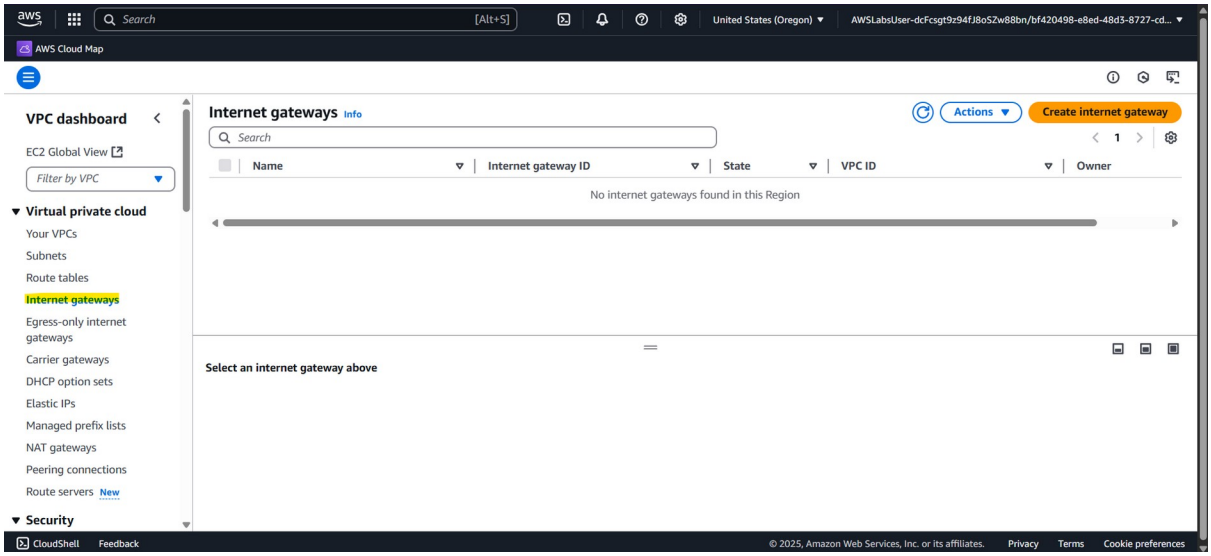
Congratulations! You have successfully created a public subnet and a private subnet in the lab VPC.

Task 3: Create an internet gateway

In this task, you create an internet gateway so that internet traffic can access the public subnet. To grant access to or from the internet for instances in a subnet in a VPC, you create an internet gateway and attach it to your VPC. Then you add a route to your subnet's route table that directs internet-bound traffic to the internet gateway.

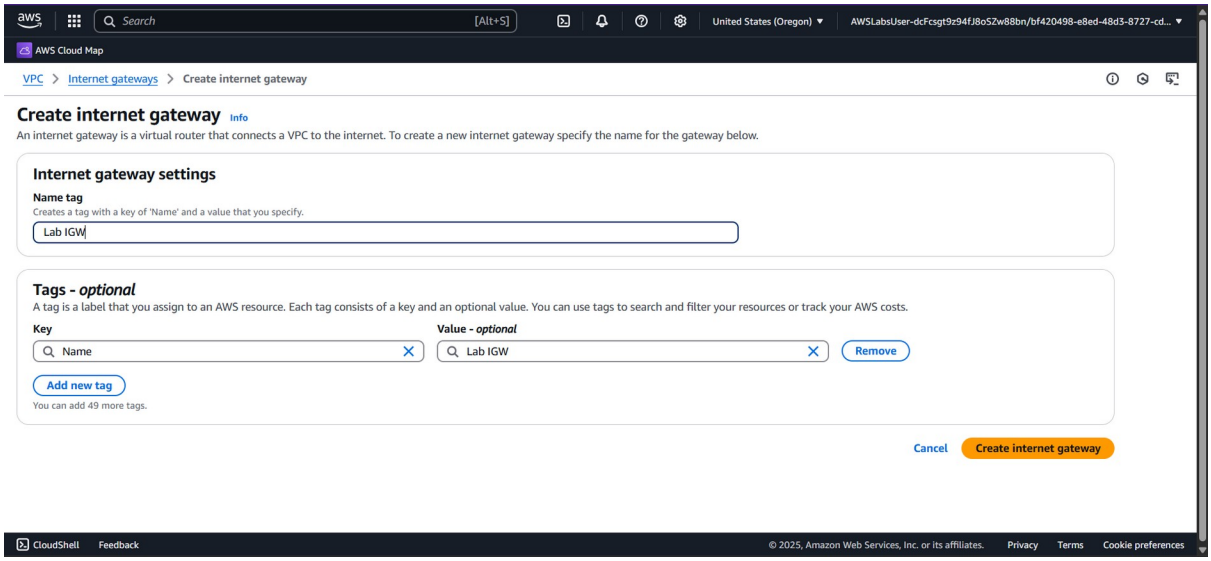
Learn more: An internet gateway serves two purposes: To provide a target in your VPC route tables for internet-bound traffic, and to perform network address translation (NAT) for instances that have been assigned public IPv4 addresses.

22. In the left navigation pane, choose **Internet gateways**.

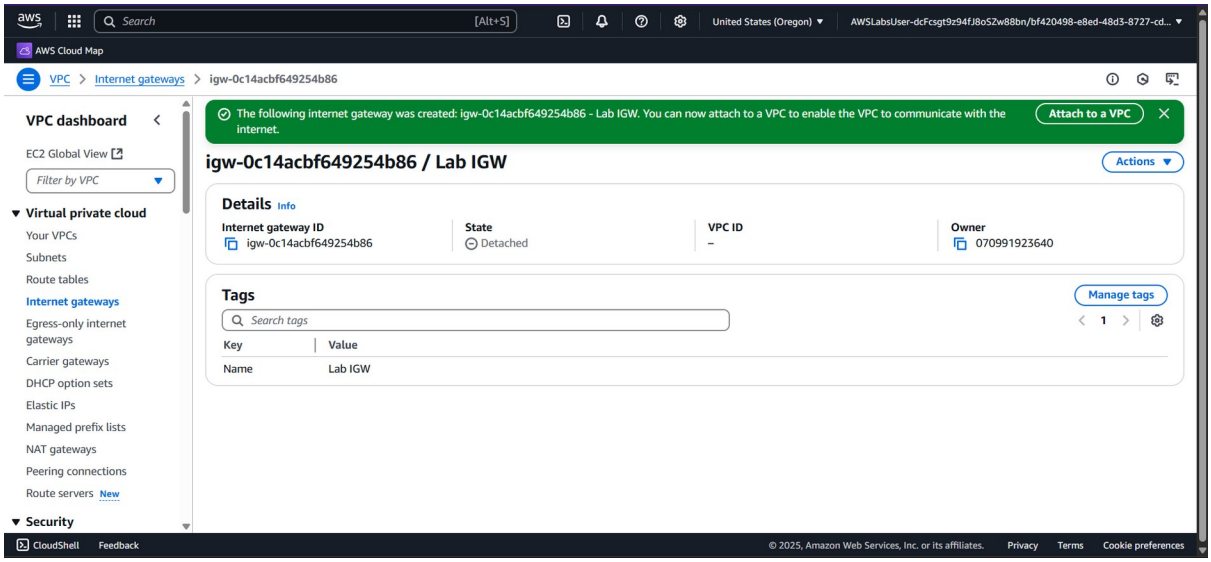


23. Choose **Create internet gateway** and configure the following:

    - **Name tag:** Enter Lab IGW.



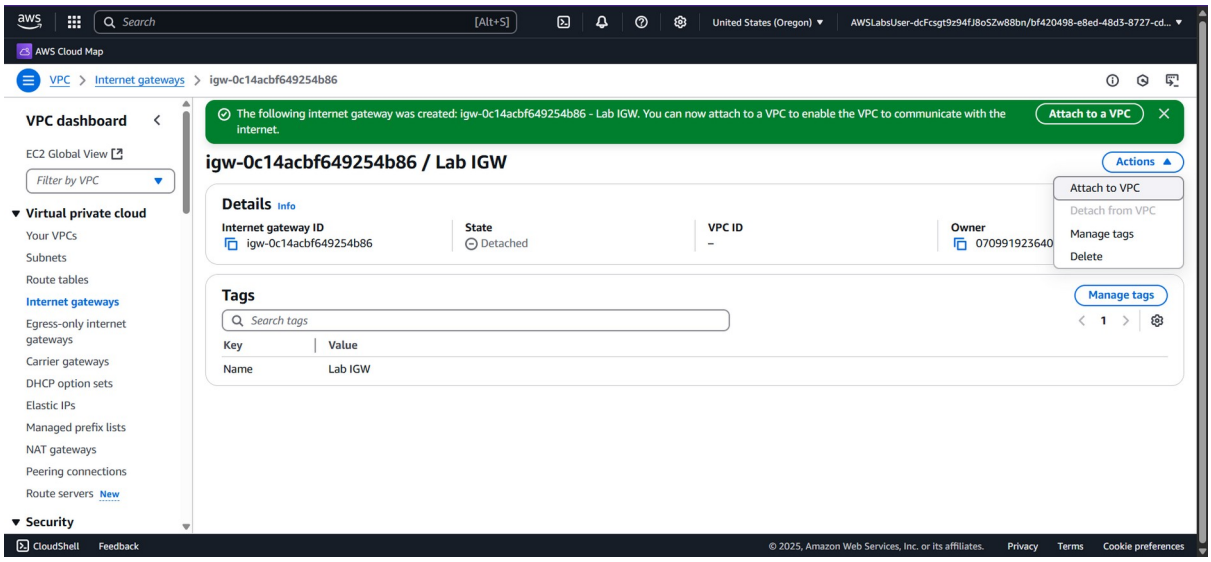
24. Choose **Create internet gateway**.



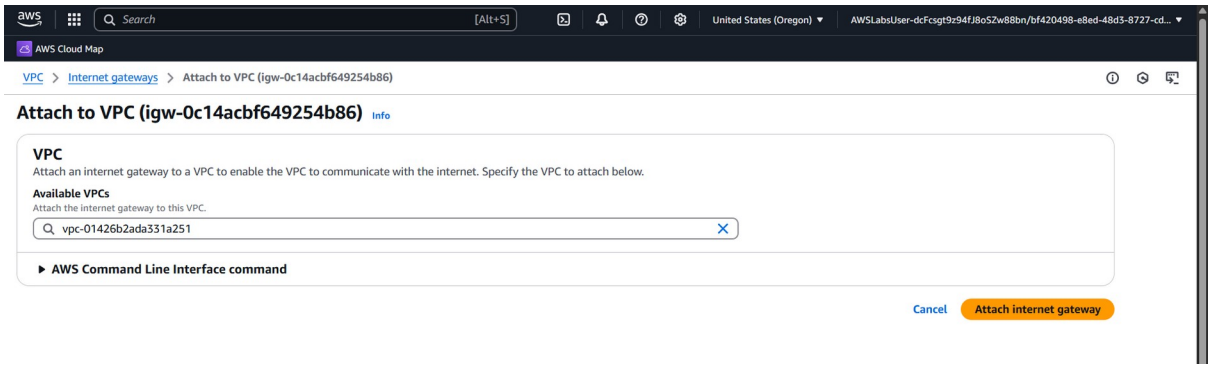
A (The following internet gateway was created: igw-xxxxxx - Lab IGW. You can now attach to a VPC to enable the VPC to communicate with the internet.) message is displayed on top of the screen.

You can now attach the internet gateway to your Lab VPC.

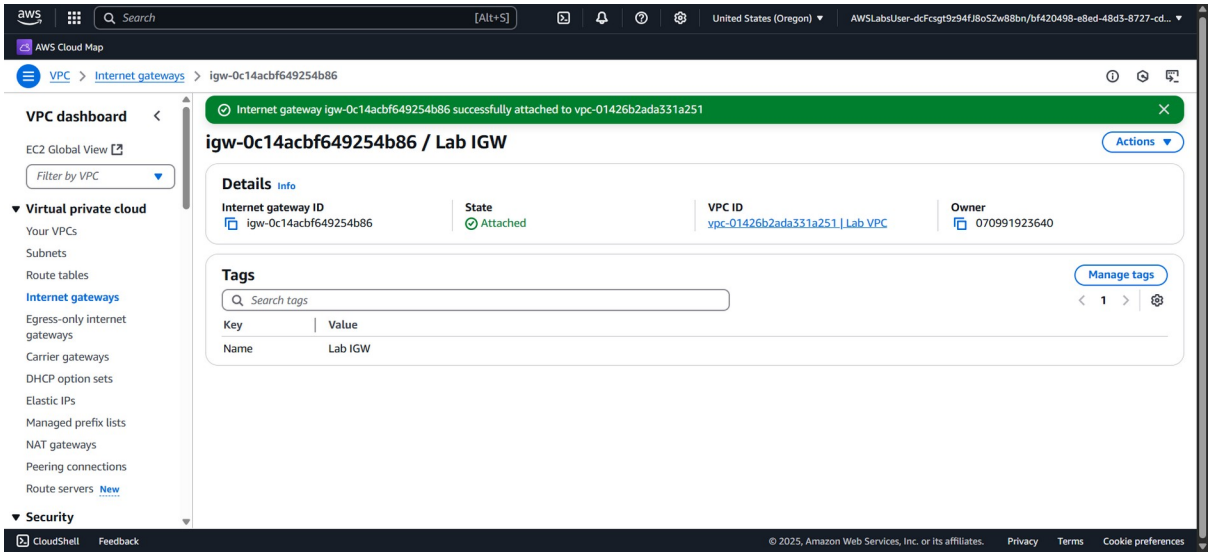
25. From the same page, choose **Actions** and choose **Attach to VPC**.



26. For **Available VPCs**, select **Lab VPC** from the dropdown menu.



27. Choose **Attach internet gateway**.

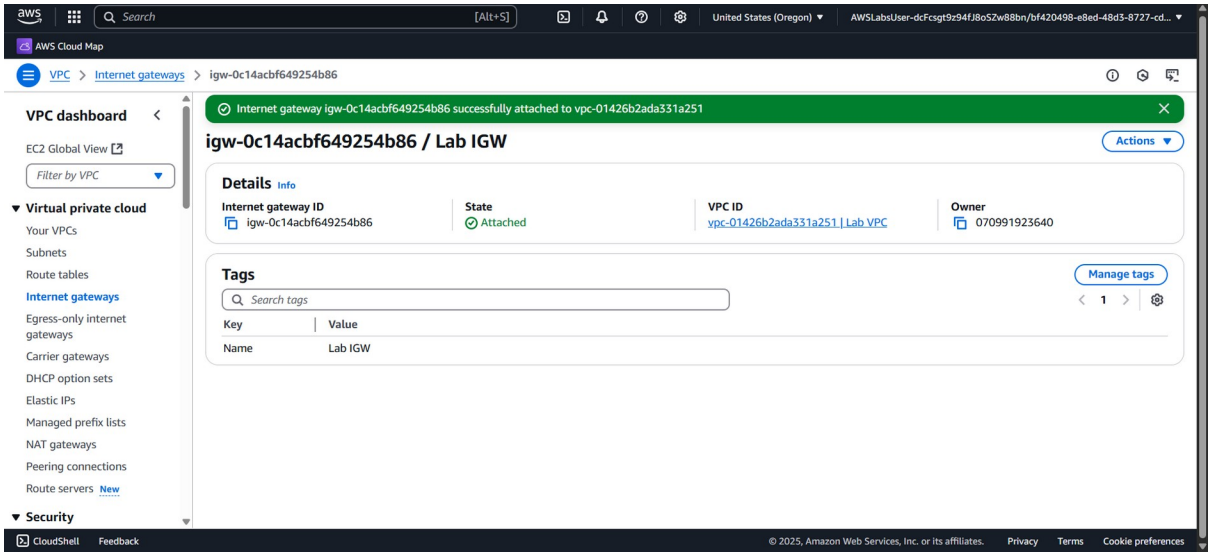


A (Internet gateway igw-xxxxx successfully attached to vpc-xxxxx) message is displayed on top of the screen.

28. Verify the state.

Expected output: It should display the following:

- **State:** Attached



The internet gateway is now attached to your Lab VPC. Even though you have created an internet gateway and attached it to your VPC, you must also configure the route table of the public subnet to use the internet gateway.

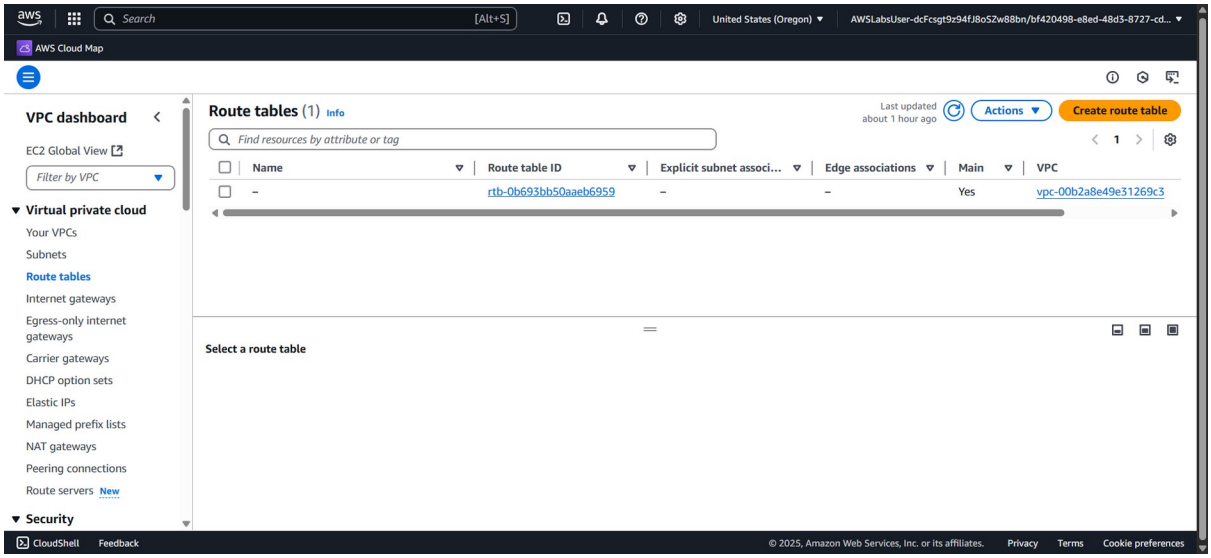
Congratulations! You have successfully created an internet gateway so that internet traffic can access the public subnet.

Task 4: Route internet traffic in the public subnet to the internet gateway

In this task, you create a route table and add a route to the route table to direct internet-bound traffic to your internet gateway and associate your public subnets with your route table. Each subnet in your VPC must be associated with a route table; the table controls the routing for the subnet. A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same route table.

Learn more: A route table contains a set of rules, called routes, that are used to determine where network traffic is directed. To use an internet gateway, your subnet's route table must contain a route that directs internet-bound traffic to the internet gateway. You can scope the route to all destinations not explicitly known to the route table (0.0.0.0/0 for IPv4 or ::/0 for IPv6), or you can scope the route to a narrower range of IP addresses. If your subnet is associated with a route table that has a route to an internet gateway, it's known as a public subnet.

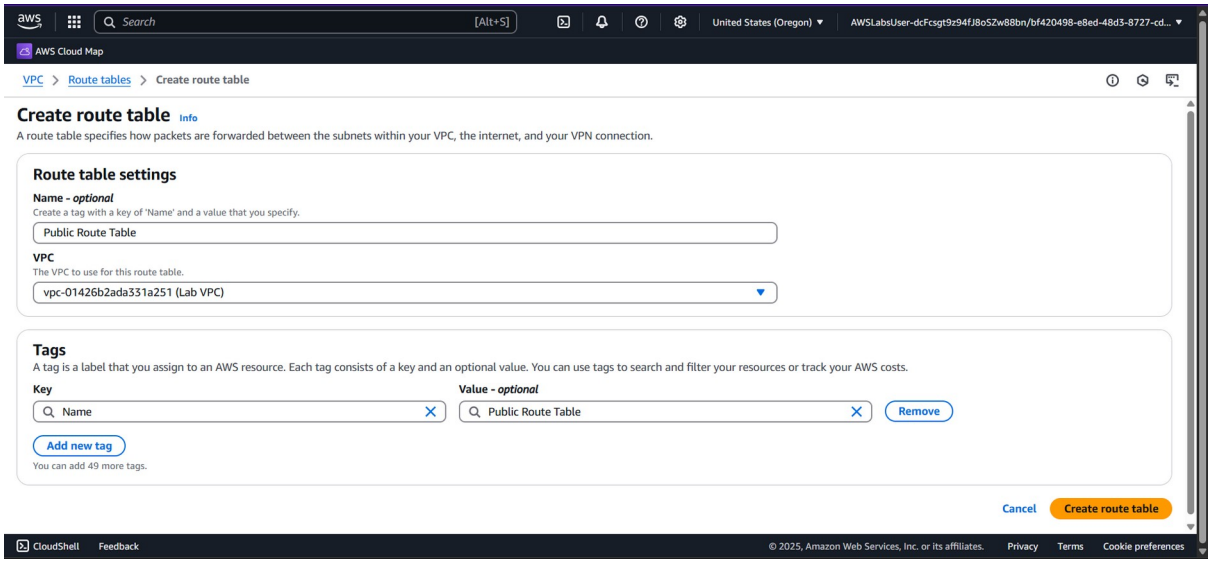
29. In the left navigation pane, choose **Route tables**.



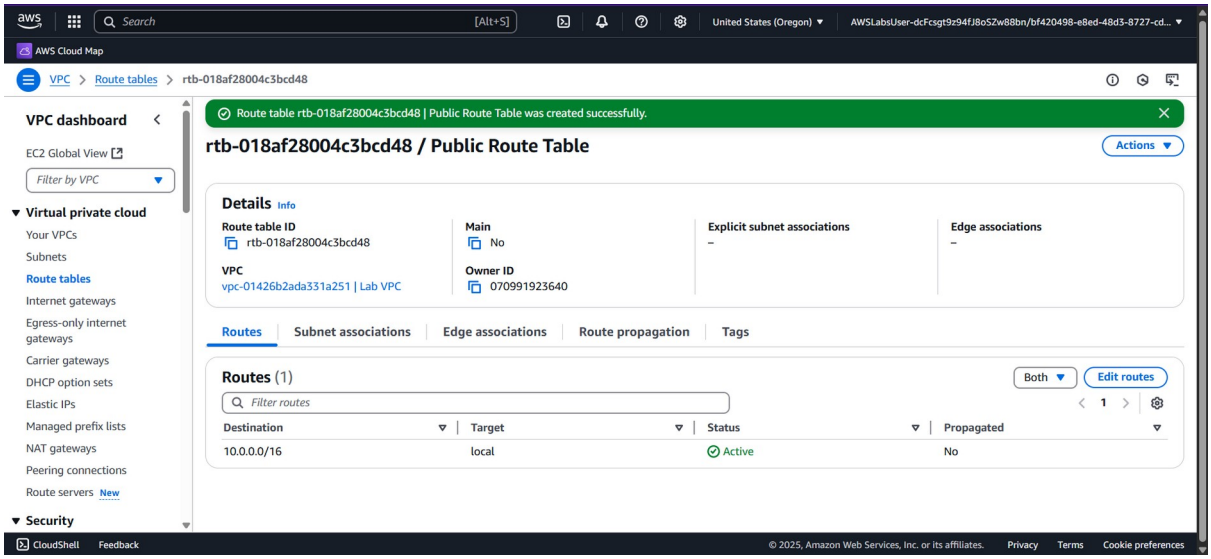
There is currently one default route table associated with the VPC, **Lab VPC**. This routes traffic locally. You now create an additional route table to route public traffic to your internet gateway.

30. Choose **Create route table**, and then configure the following:

    - **Name - *optional*:** Enter Public Route Table.
    - **VPC:** Select **Lab VPC** from the dropdown menu.



31. Choose **Create route table**.



A (Route table rtb-xxxxxxx | Public Route Table was created successfully.) message is displayed on top of the screen.

32. Choose the **Routes** tab in the lower half of the page.

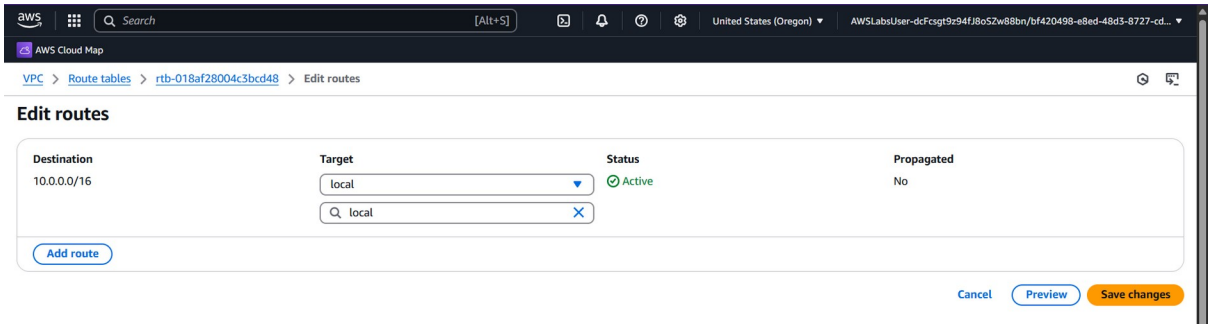
Note: There is one route in your route table that allows traffic within the 10.0.0.0/16 network to flow within the network, but it does not route traffic outside of the network.

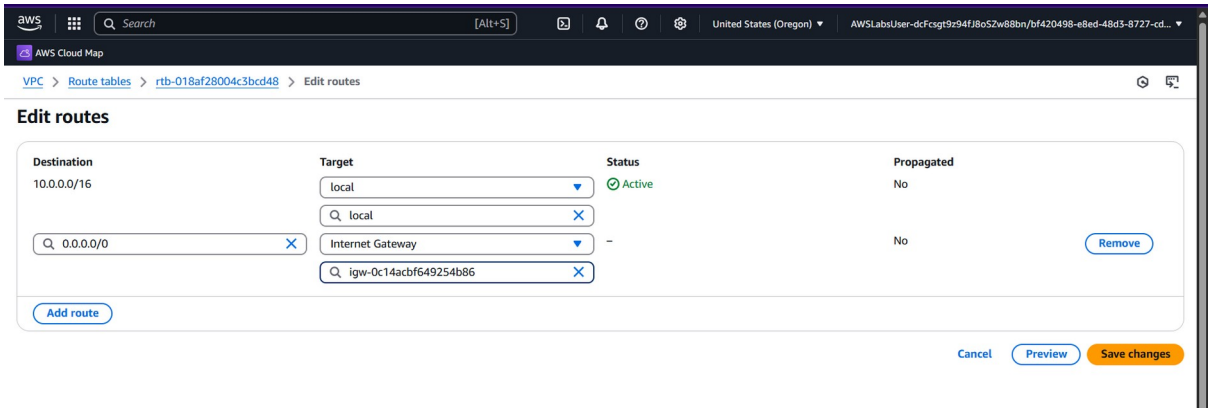
You now add a new route to permit public traffic.

33. Choose **Edit routes**.

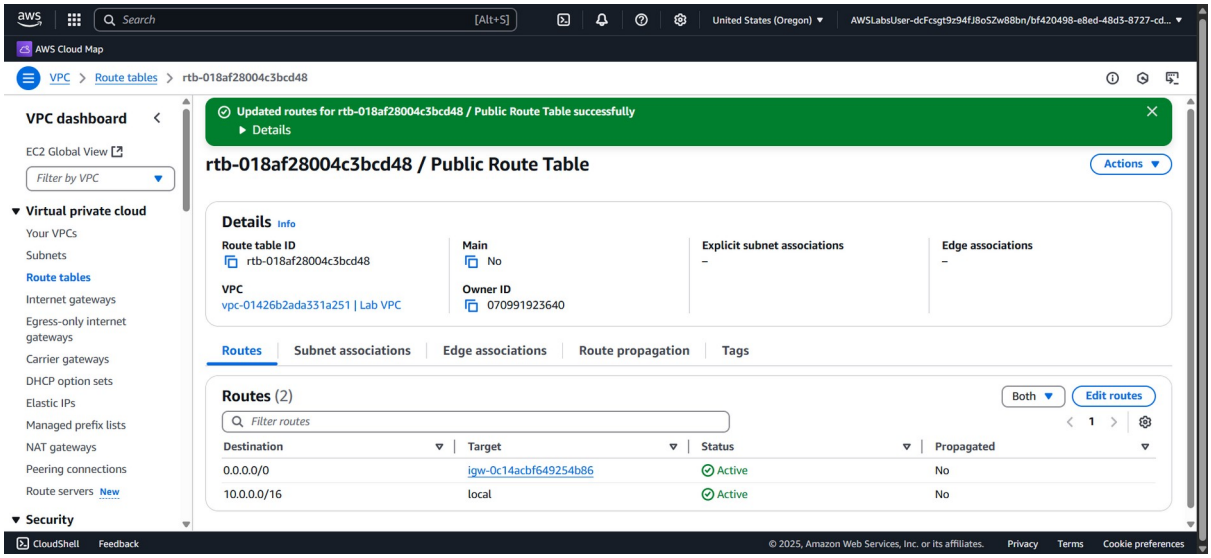
34. Choose **Add route**, and then configure the following:

- **Destination:** Enter 0.0.0.0/0.
- **Target:** Choose **Internet Gateway** in the dropdown menu, and then choose the displayed internet gateway ID.



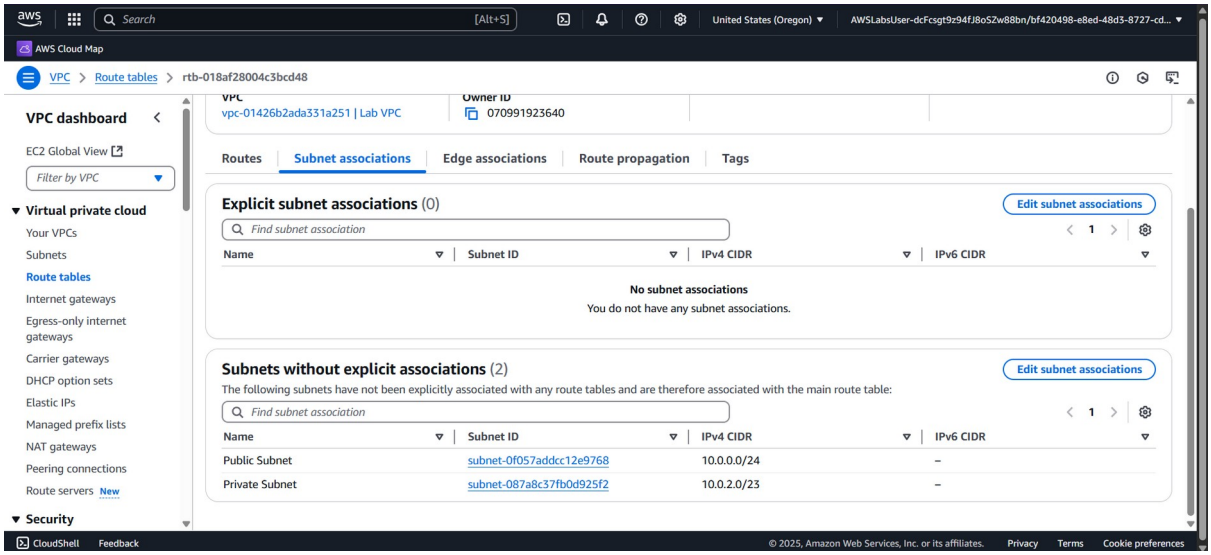


35. Choose **Save changes**.

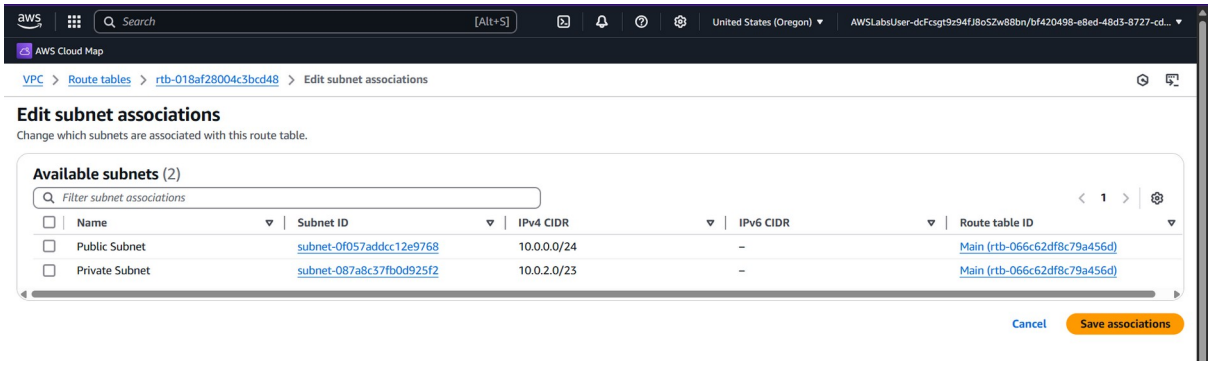


A **(Updated routes for rtb-xxxxxxx / Public Route Table successfully)** message is displayed on top of the screen.

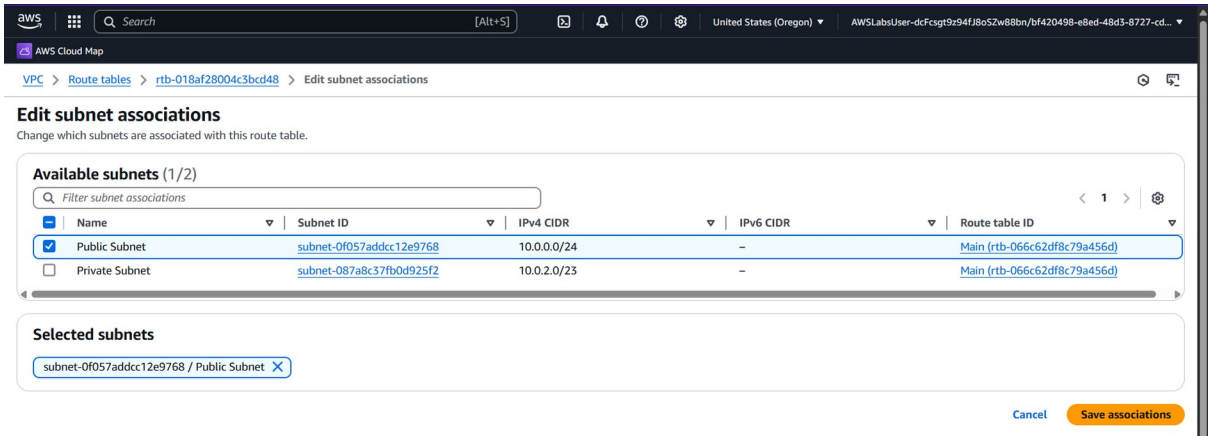
36. Choose the **Subnet associations** tab.



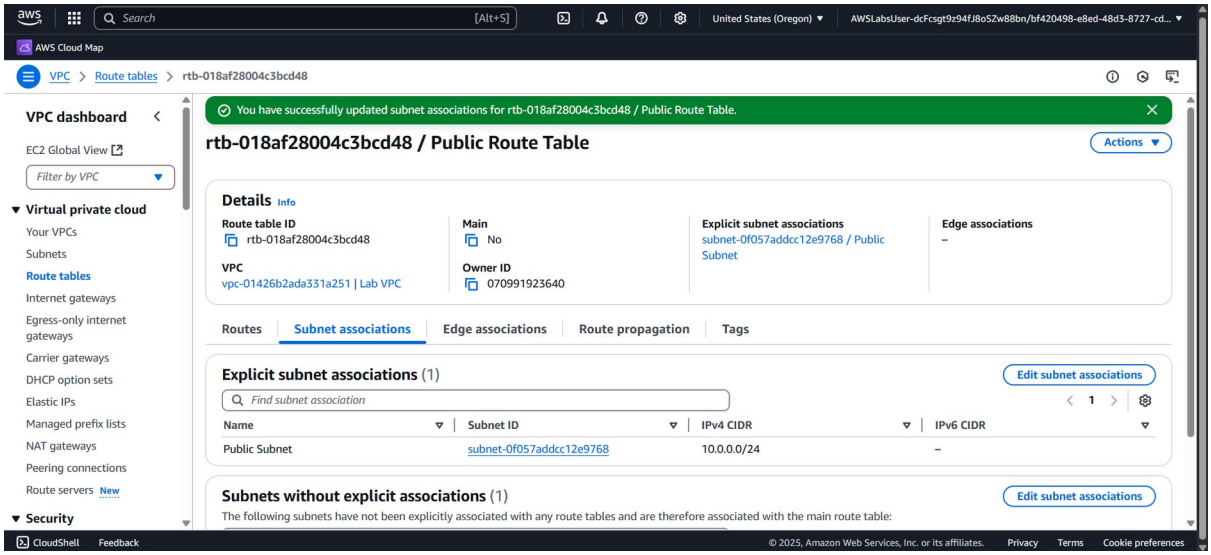
37. Choose **Edit subnet associations**.



38. Select **Public Subnet**



39. Choose **Save associations**.



A (You have successfully updated subnet associations for rtb-xxxxxxx / Public Route Table.) message is displayed on top of the screen.

Note: The subnet is now *public* because it has a route to the internet through the internet gateway.

Congratulations! You have successfully configured the route table.

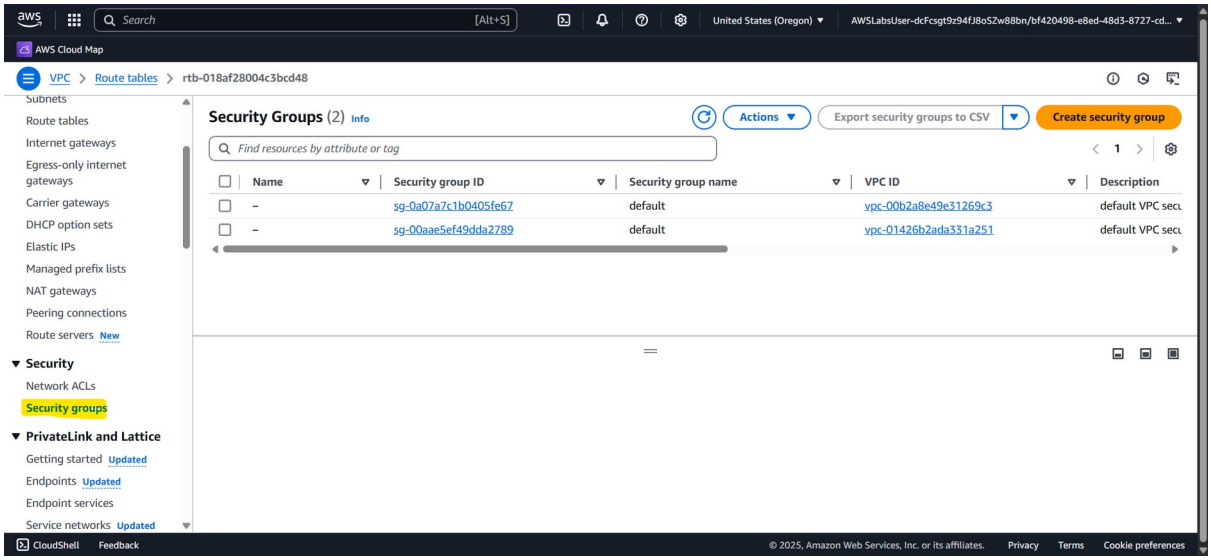
Task 5: Create a public security group

In this task, you create a security group so that users can access your Amazon EC2 instance. Security groups in a VPC specify which traffic is allowed to or from an Amazon EC2 instance.

Learn more: You can use Amazon EC2 security groups to help secure instances within an Amazon VPC. By using security groups in a VPC, you can specify both inbound and outbound network traffic that is allowed to or from each Amazon EC2 instance. Traffic that is not explicitly allowed to or from an instance is automatically denied.

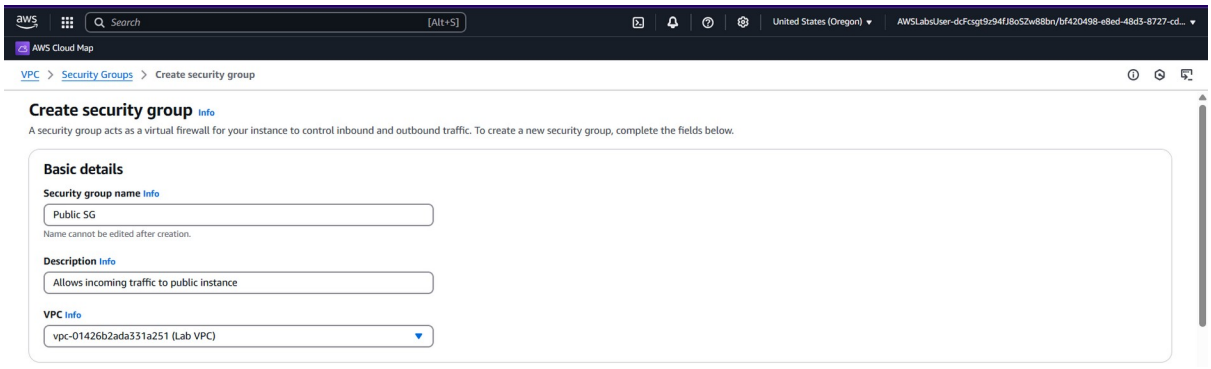
Security: It is recommended to use *HTTPS* protocol to improve web traffic security. However, to simplify this lab, only *HTTP* protocol is used.

40. In the left navigation pane, choose **Security groups**.



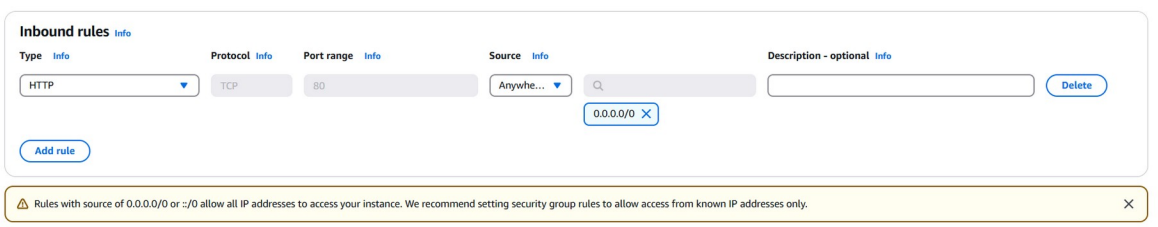
41. Choose **Create security group**, and then configure the following:

    - **Security group name:** Enter Public SG.
    - **Description:** Enter Allows incoming traffic to public instance.
    - **VPC:** Select **Lab VPC** from the dropdown menu.



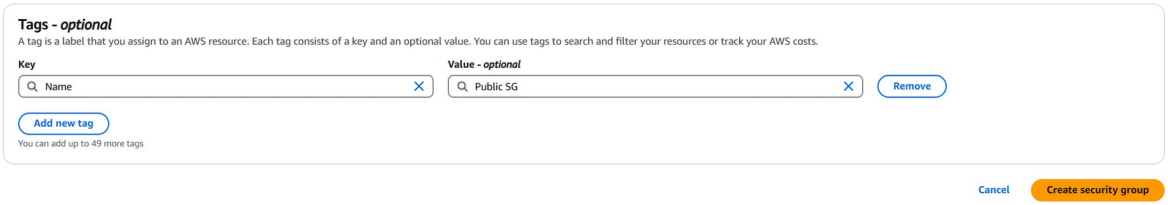
42. In the **Inbound rules** section, choose **Add rule** and configure the following:

    - **Type:** Select **HTTP** from the dropdown menu.
    - **Source:** Select **Anywhere-IPv4** from the dropdown menu.

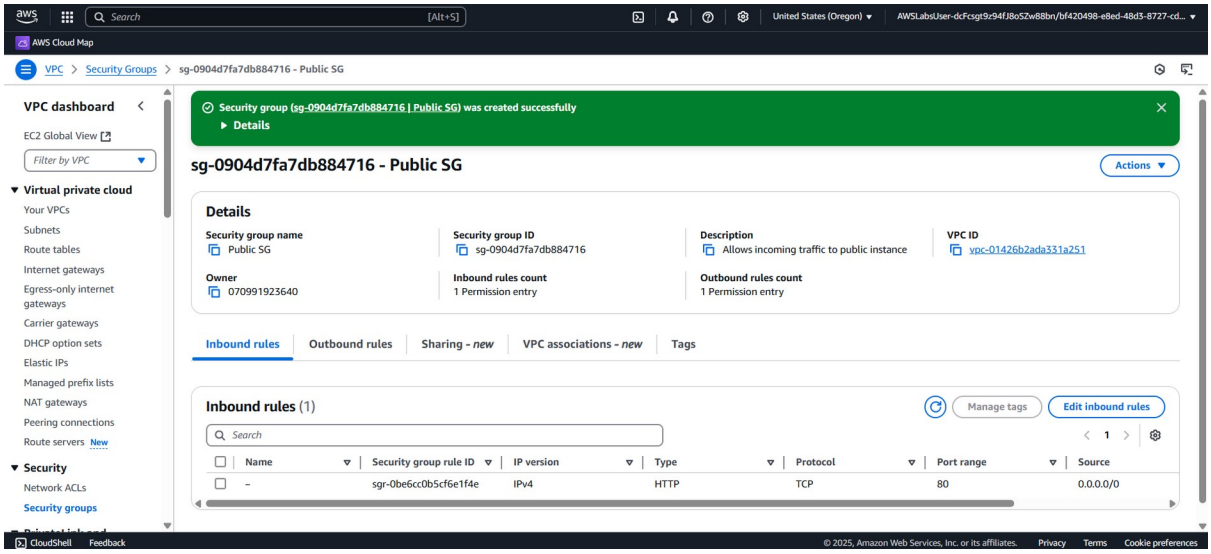


43. In the **Tags - *optional*** section, choose **Add new tag** and configure the following:

    - **Key:** Enter Name.
    - **Value:** Enter Public SG.



44. Choose **Create security group**.

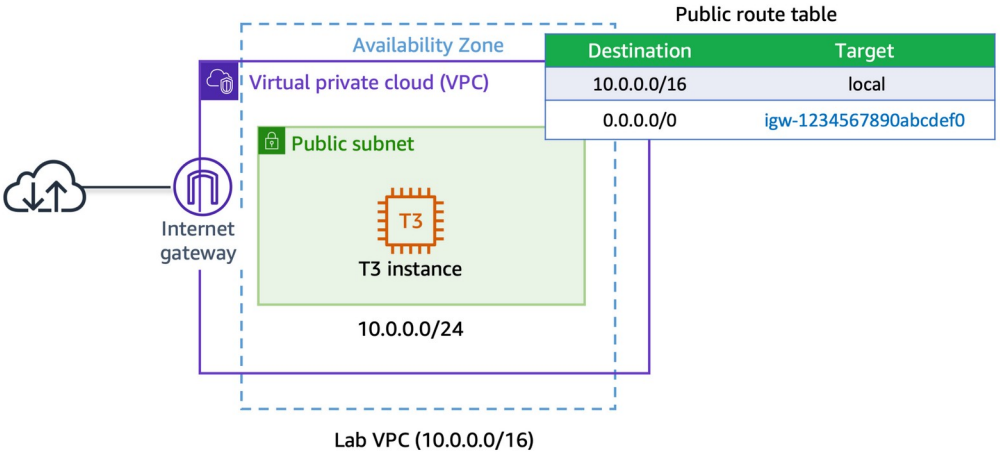


A **(Security group (sg-xxxxxxx | Public SG) was created successfully)** message is displayed on top of the screen.

Congratulations! You have successfully created a security group that allows HTTP traffic. You need this in the next task when you launch an Amazon EC2 instance in the public subnet.

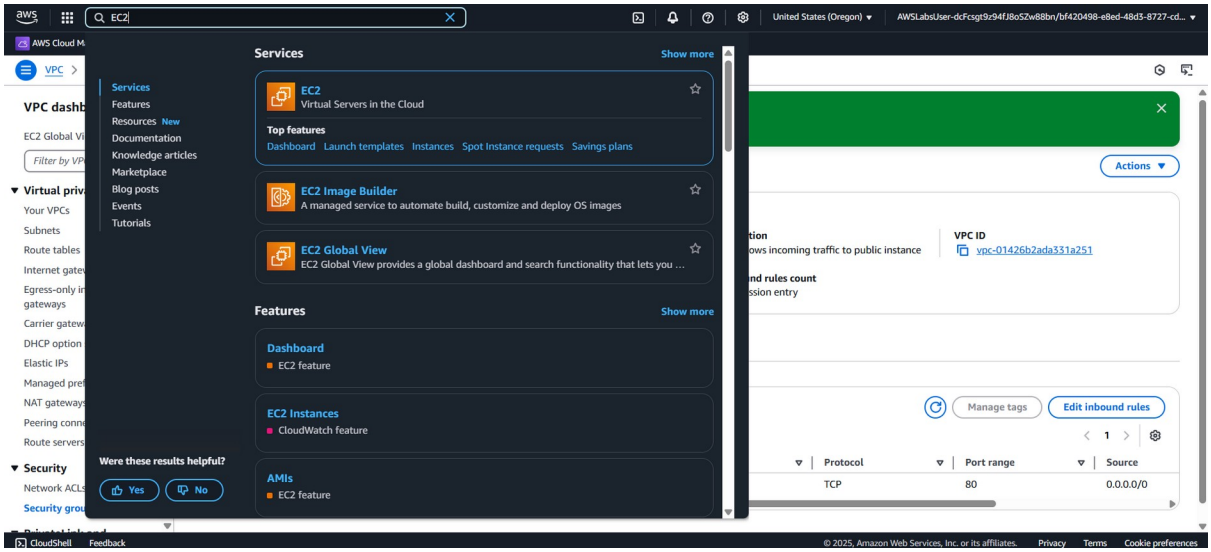
Task 6: Launch an Amazon EC2 instance into a public subnet

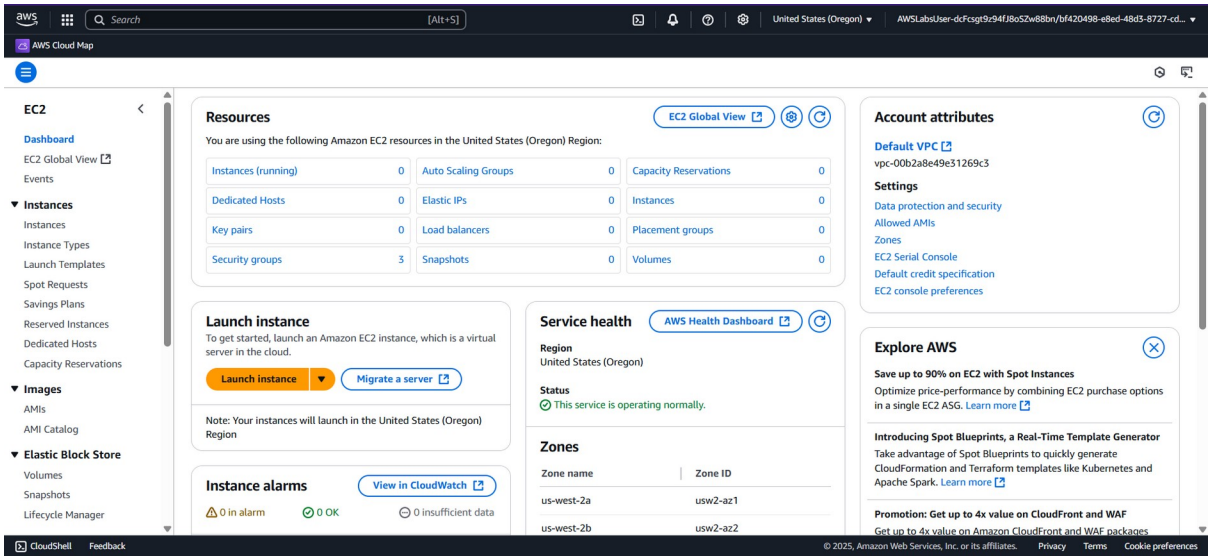
In this task, you launch an Amazon EC2 instance into a public subnet. To activate communication over the internet for IPv4, your instance must have a public IPv4 address that's associated with a private IPv4 address on your instance. By default, your instance is only aware of the private (internal) IP address space defined within the VPC and subnet.



Learn more: The internet gateway that you created logically provides the one-to-one NAT on behalf of your instance. So when traffic leaves your VPC subnet and goes to the internet, the reply address field is set to the public IPv4 address or Elastic IP address of your instance, and not its private IP address.

45. At the top of the AWS Management Console, in the search bar, search for and choose EC2.



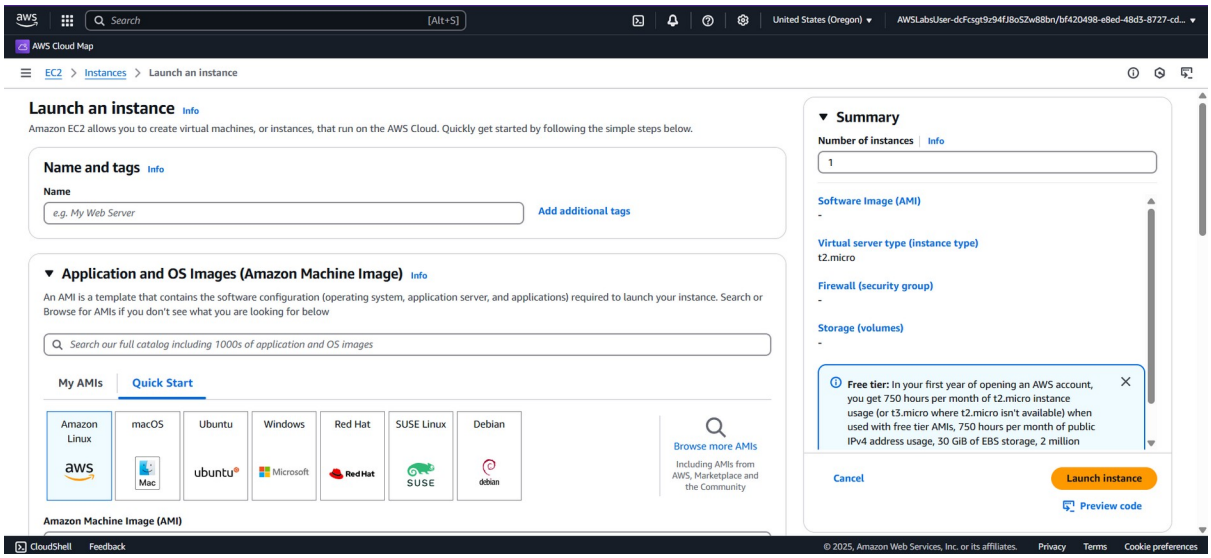


The **Amazon EC2 Management Console** is displayed.

Task 6.1: Begin the instance configuration

46. From the console navigation menu on the left, choose **EC2 Dashboard**.

47. From the **Launch instances** section, choose **Launch instances**.



The **Launch an instance** page is displayed.

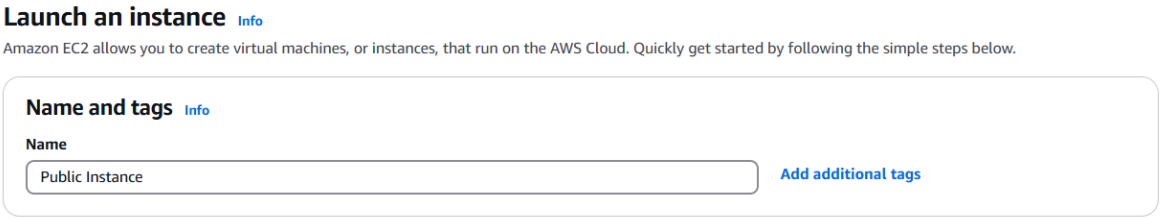
Task 6.2: Add tags to the instance

You can use tags to categorize your AWS resources in different ways, such as by purpose, owner, or environment. You can apply tags to most AWS Cloud resources. Each tag consists of a *key* and a *value*, both of which you define. One use of tags is for when you must manage many resources of the same type. You can quickly search for and identify a specific resource by the tag you have applied to it.

In this task, you add a tag to the Amazon EC2 instance.

48. Locate the **Name and tags** section.

49. In the **Name** field, enter Public Instance.



Note: No additional instance tags are required for this lab.

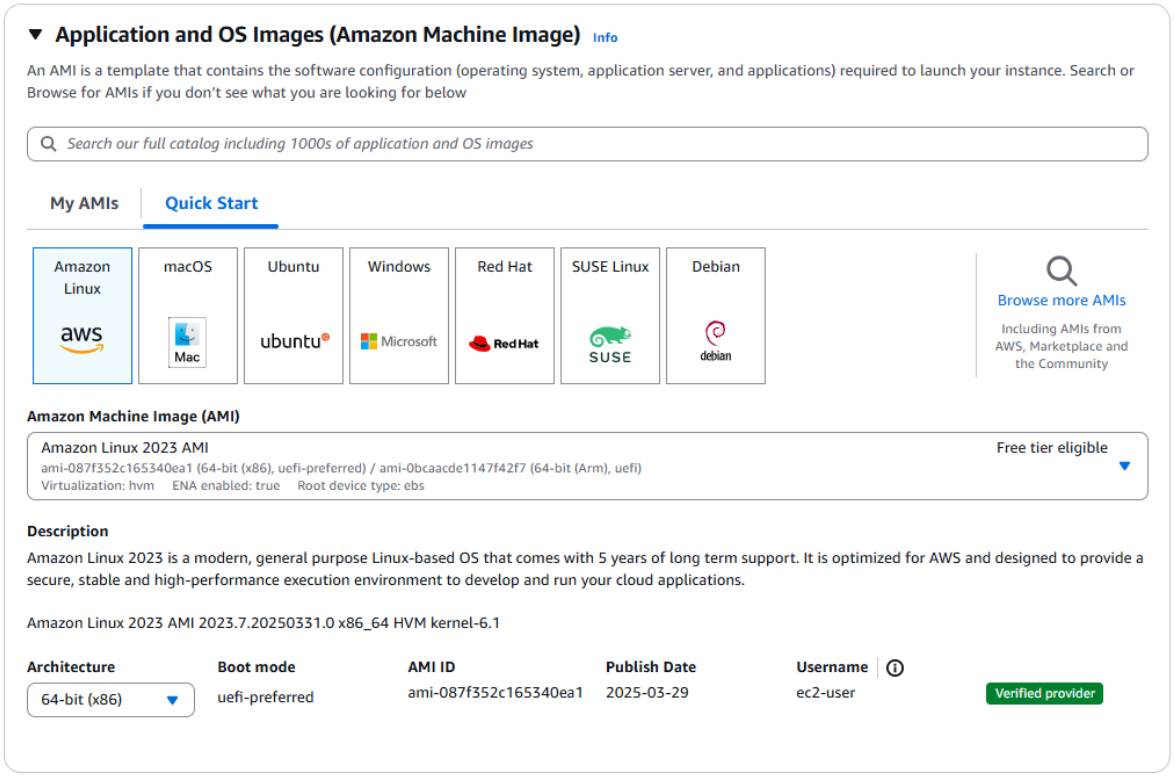
Task 6.3: Select an AMI

In this task, you choose an Amazon Machine Image (AMI). The AMI contains a copy of the disk volume used to launch the instance.

50. Locate the **Application and OS Images (Amazon Machine Image)** section.

51. Ensure that **Amazon Linux** is selected as the OS.

52. Ensure that **Amazon Linux 2023 AMI** is selected in the dropdown menu.



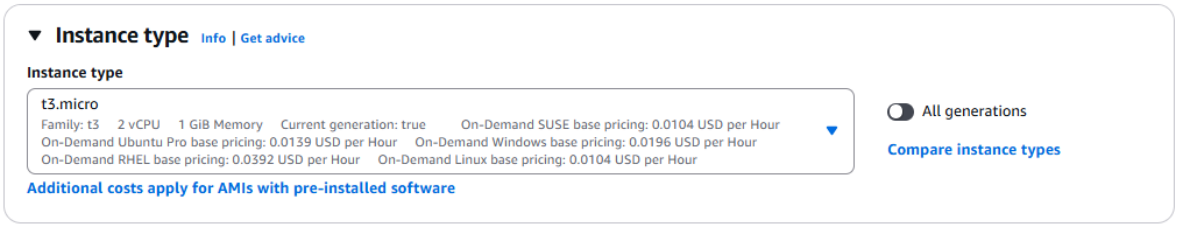
Task 6.4: Choose the Amazon EC2 instance type

Each instance type allocates a specific combination of virtual CPUs (vCPUs), memory, disk storage, and network performance.

For this lab, use a **t3.micro** instance type. This instance type has 2 vCPUs and 1 GiB of memory.

53. Locate the **Instance type** section.

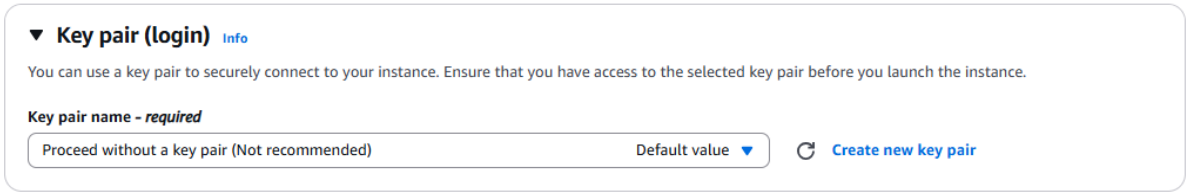
54. From the **Instance type** dropdown menu, choose **t3.micro**.



Task 6.5: Configure key pair for login

55. Locate the **Key pair (login)** section.

56. From the **Key pair name - *required*** dropdown menu, choose **Proceed without a key pair (Not recommended)**.



Task 6.6: Configure instance networking

57. Locate the **Network settings** section.

58. Choose **Edit**.

59. Configure the following settings from the dropdown menus:

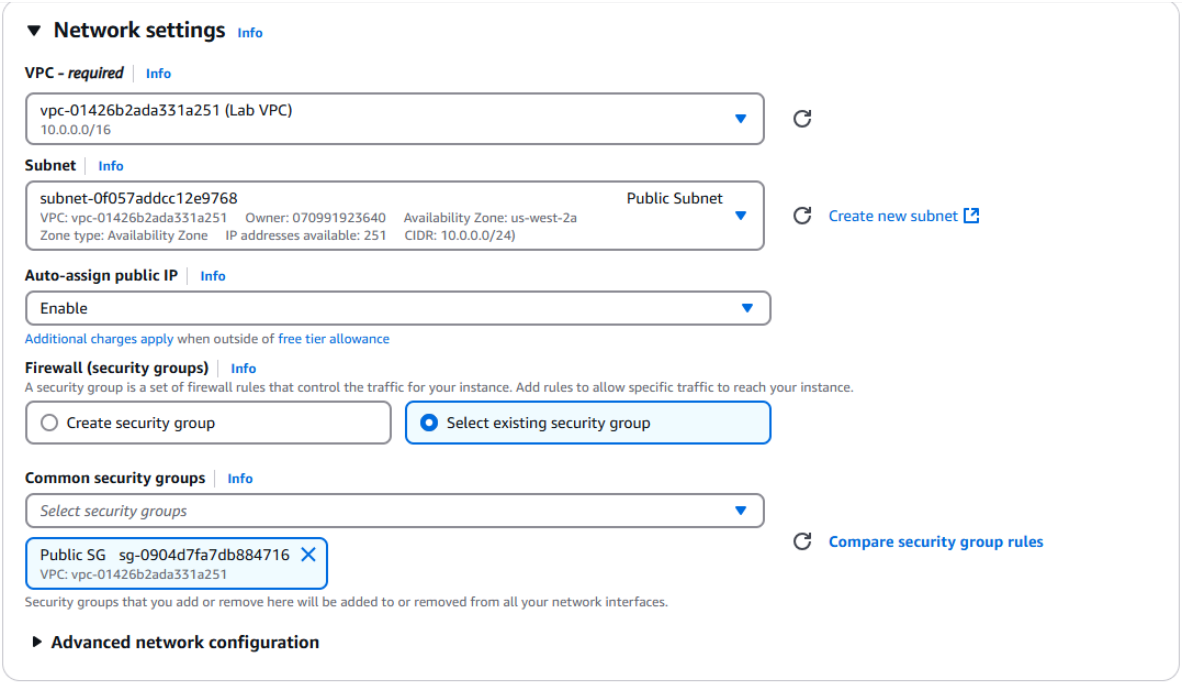
    - **VPC - *required:*** Select **Lab VPC**.
    - **Subnet:** Select **Public Subnet**.
    - **Auto-assign public IP:** Select **Enable**.

Task 6.7: Configure instance security groups

You can use security groups to define both the allowed/denied and the inbound/outbound traffic for the elastic network interface. The network interface is attached to an Amazon EC2 instance. Port 80 is the default port for HTTP traffic, and it is necessary for the web server you launch in this lab to work correctly.

60. For **Firewall (security groups)**, choose **Select existing security group**.

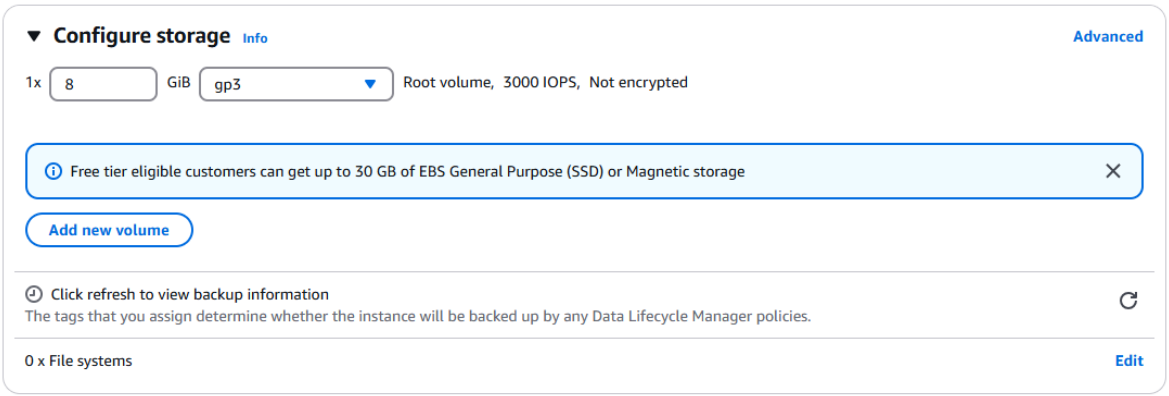
61. From the **Common security groups** dropdown menu, choose the security group that has a name like **Public SG**.



Task 6.8: Add storage

You can use the **Configure storage** section to specify or modify the storage options for the instance and add additional Amazon Elastic Block Store (Amazon EBS) disk volumes attached to the instance. The EBS volumes can be configured in both their size and performance.

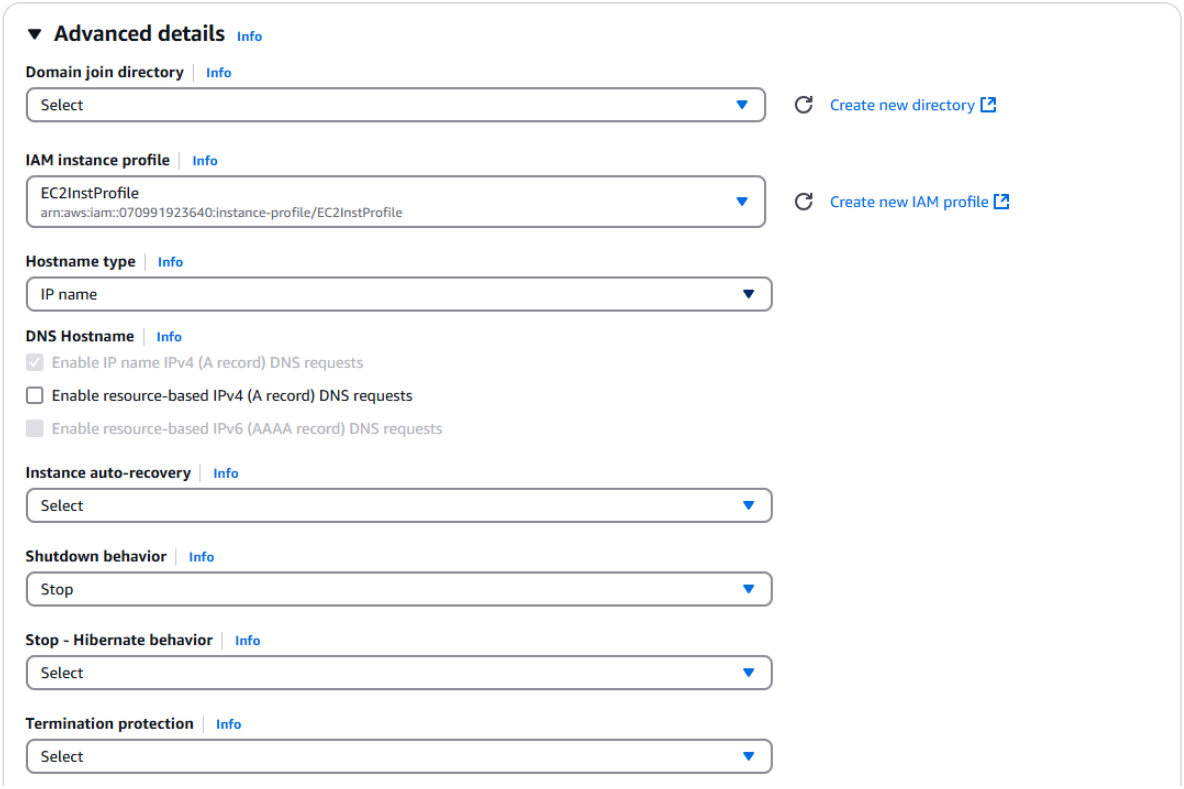
In this lab, the default storage settings are all that is needed. No changes are required.



Task 6.9: Configure user data

62. Locate and expand the **Advanced details** section.

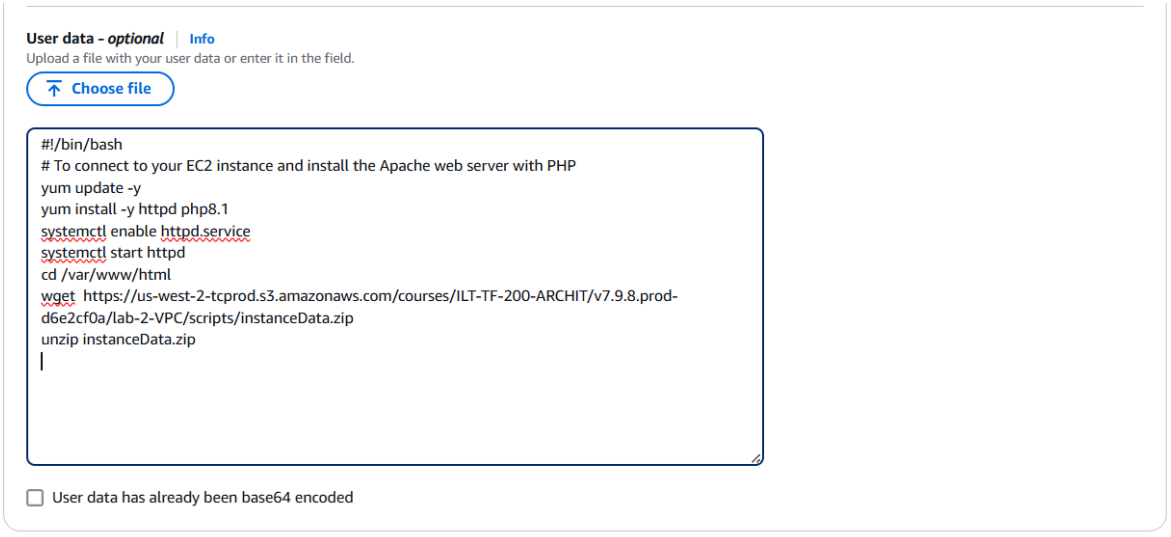
63. From the **IAM instance profile** dropdown menu, select the role that has a name like **EC2InstProfile**.



Note: To install and configure the new instance as a web server, you provide a user data script that automatically runs when the instance launches.

64. In the **User data - *optional*** section, copy and paste the following:

```
#!/bin/bash
# To connect to your EC2 instance and install the Apache web server with PHP
yum update -y
yum install -y httpd php8.1
systemctl enable httpd.service
systemctl start httpd
cd /var/www/html
wget  https://us-west-2-tcprod.s3.amazonaws.com/courses/ILT-TF-200-ARCHIT/v7.9.8.prod-d6e2cf0a/lab-2-VPC/scripts/instanceData.zip
unzip instanceData.zip
```



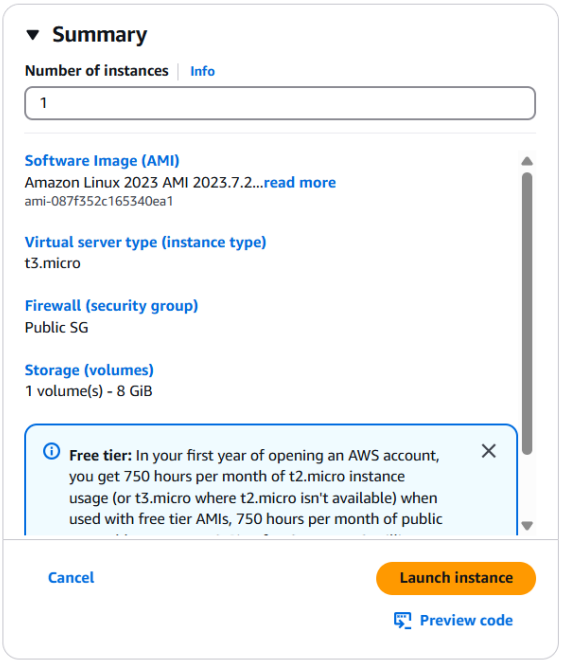
The remaining settings on the page can be left at their default values.

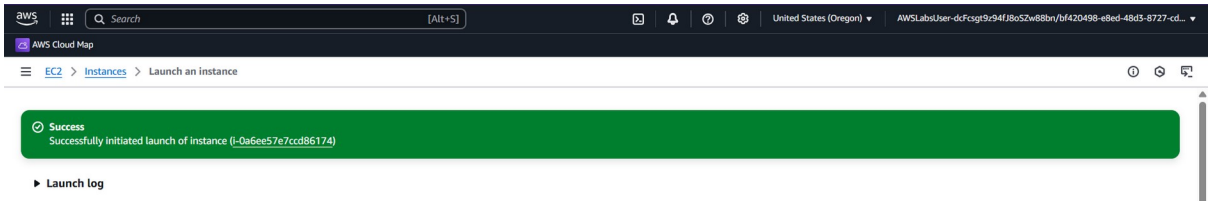
Task 6.10: Review the instance launch

Take a moment to review that the configuration for the Amazon EC2 instance you are about to launch is correct.

65. Locate the **Summary** section.

66. Choose **Launch instance**.

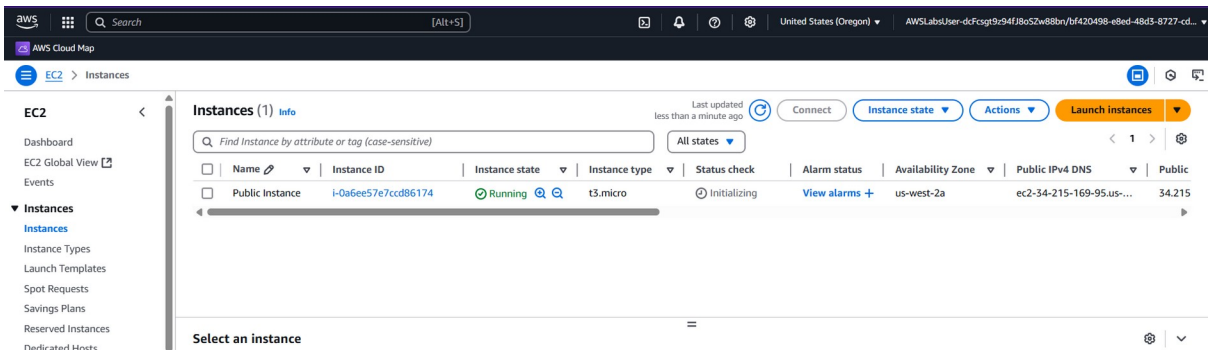




The **Launch an instance** page is displayed.

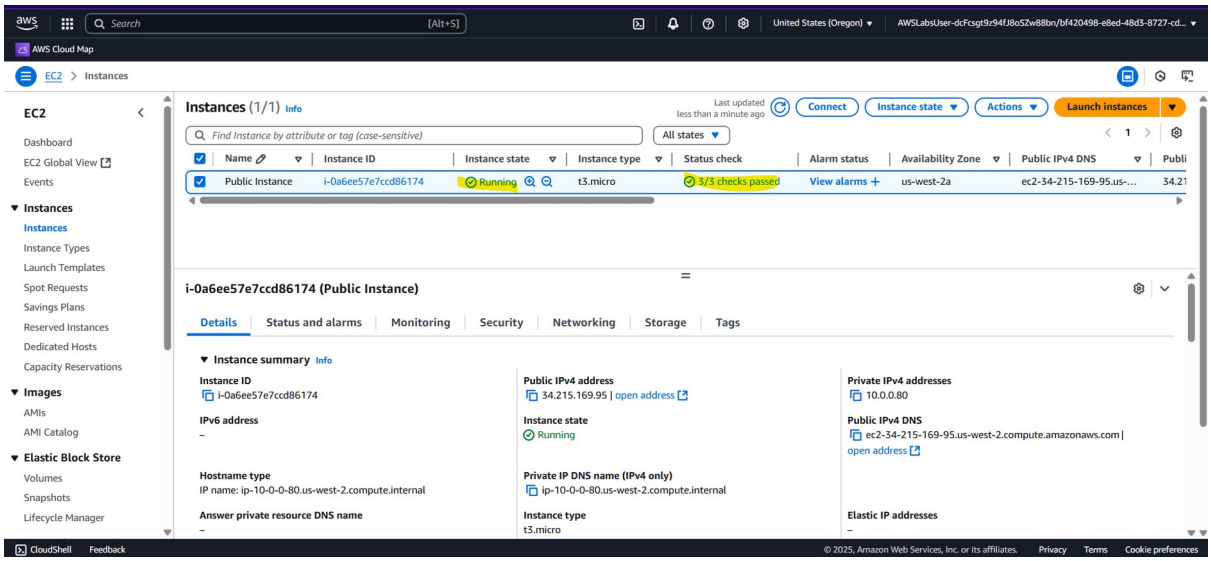
Your Amazon EC2 instance is now launched and configured as you specified.

67. Choose **View all instances**.



The **Amazon EC2 console** is displayed.

68. Occasionally choose the console refresh button and wait for **Public Instance** to display the **Instance state** as Running and wait for Status check to pass 3/3 checks passed.



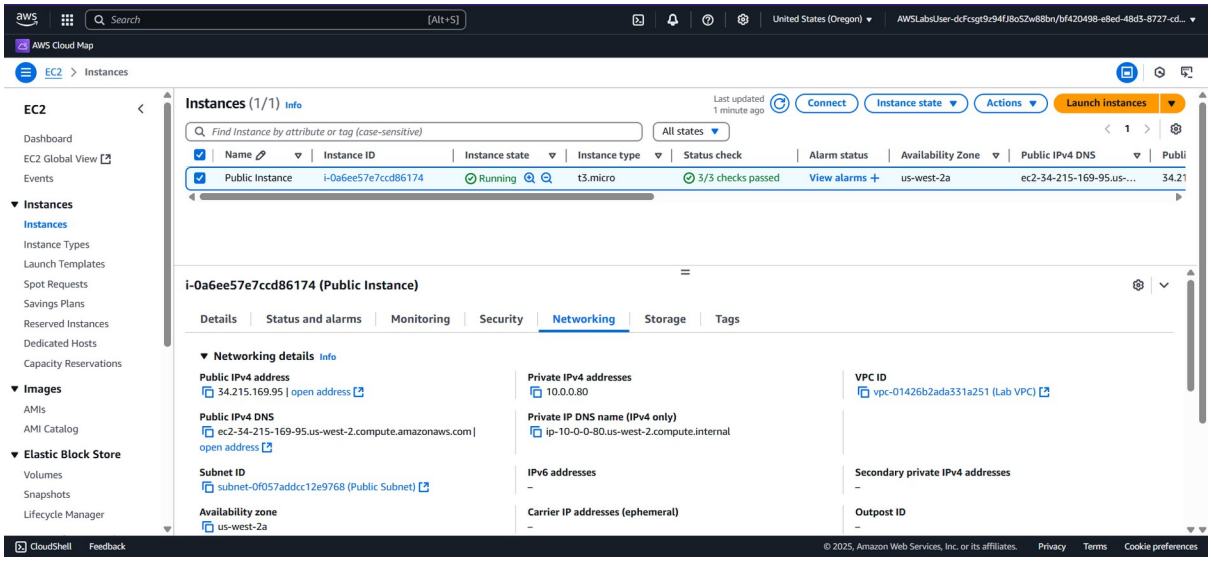
Note: The Amazon EC2 instance named Public Instance is initially in a *Pending* state. The instance state then changes to Running indicating that the instance has finished booting.

Congratulations! You have successfully launched an Amazon EC2 instance into a public subnet.

Task 7: Connect to a public instance through HTTP

In this task, you connect to the public instance and launch the basic Apache web server page. The inbound rules added earlier that allow HTTP access (port 80) allow you to connect to the web server running Apache.

69. Choose the **Networking** tab in the lower pane. In the left navigation pane, choose **Instances**.
70. Select **Public Instance**.
71. Choose the **Networking** tab in the lower pane.

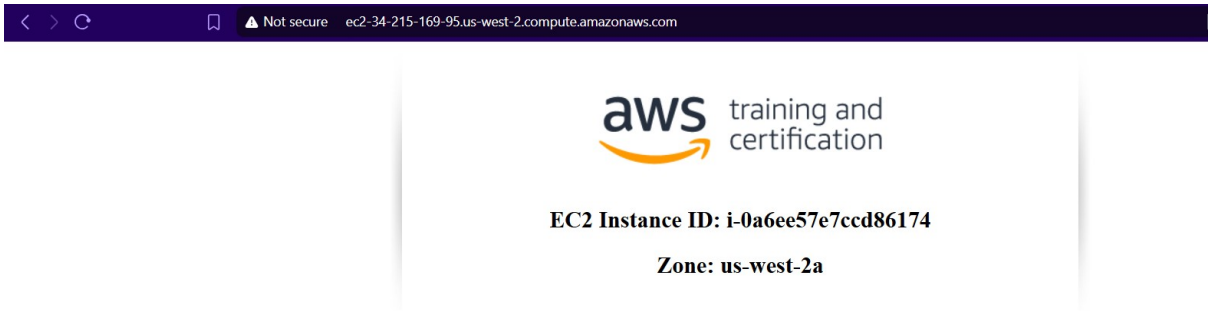


Note: If you need to make any section of the console larger, you can resize the horizontal edges of the containers displayed on the console.

72. Locate the **Public IPv4 DNS** value.

73. Copy the public DNS value. Do not choose the open address option, because HTTPS is not set up for this lab environment.

74. Open a new browser tab and paste the public DNS value for *Public Instance* in the URL address bar.



The web page hosted on the Amazon EC2 instance is displayed. The page displays the instance ID and the AWS Availability Zone where the Amazon EC2 instance is located.

75. Close the browser tab and return to the console.

Congratulations! You have successfully launched an Apache web server in the public subnet and tested the HTTP connection. You can safely close the tab and return to the console.

Task 8: Connect to the Amazon EC2 instance in the public subnet through Session Manager

In this task, you connect to your Amazon EC2 instance in the public subnet using Session Manager.

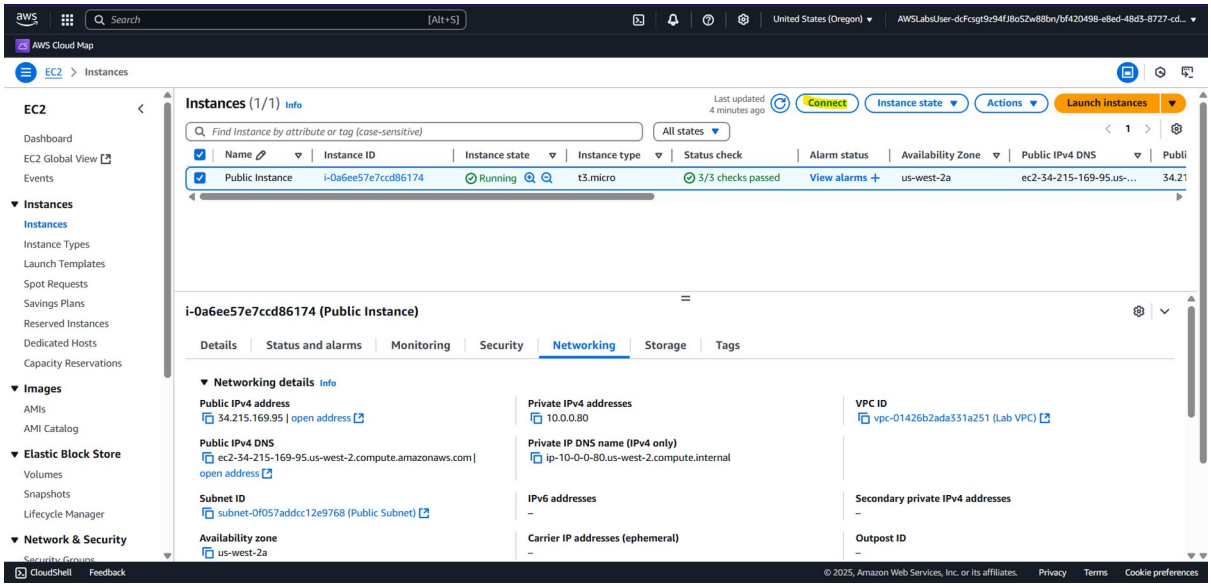
Learn more: Session Manager is a fully managed AWS Systems Manager capability that you use to manage your Amazon EC2 instances through an interactive one-click browser-based shell or through the AWS

Command Line Interface (AWS CLI). You can use Session Manager to start a session with an Amazon EC2 instance in your account. After starting the session, you can run bash commands as you would through any other connection type.

76. At the top of the AWS Management Console, in the search bar, search for and choose EC2.

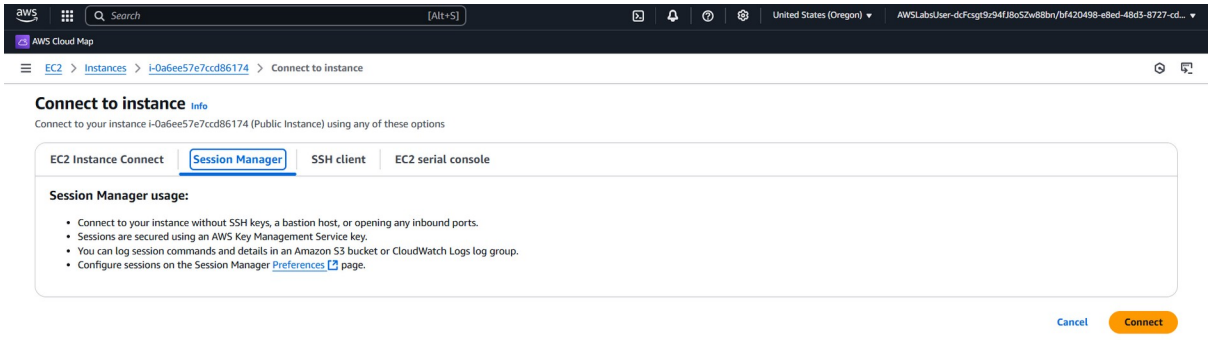
77. In the left navigation pane, choose **Instances**.

78. Select **Public Instance** and choose **Connect**.



The **Connect to instance** page is displayed.

79. Choose the **Session Manager** tab.



Learn more: With Session Manager, you can connect to Amazon EC2 instances without needing to expose the SSH port on your firewall or Amazon VPC security group. For more information, see AWS Systems Manager Session Manager.

80. Choose **Connect**.

A new browser tab or window opens with a connection to the **Public Instance**.

Note: The Session Manager service is not updated in real time. If you experience errors with Session Manager connecting to an Amazon EC2 instance you just launched, ensure that you have given the instance a few minutes to launch, pass health checks, and communicate with the Session Manager service before trying to open a session connection again.

81. **Command:** Enter the following command to change to the home directory (/home/ssm-user/) and test web connectivity using the cURL command:

```
cd ~
curl -I https://aws.amazon.com/training/
```

Expected output:

```
HTTP/2 200
content-type: text/html;charset=UTF-8
server: Server
date: Wed, 19 Apr 2023 14:43:47 GMT
x-amz-rid: 6HVPS1JY1XW2S1K34Q3Z
set-cookie: aws-priv=eyJ2IjoxLCJldSI6MCwic3QiOjB9; Version=1; Comment="Anonymous cookie for privacy regulations"; Domain=.aws.amazon.com; Max-Age=31536000; Expires=Thu, 18-Apr-2024 14:43:47 GMT; Path=/; Secure
set-cookie: aws_lang=en; Domain=.amazon.com; Path=/
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block
strict-transport-security: max-age=63072000
x-content-type-options: nosniff
x-amz-id-1: 6HVPS1JY1XW2S1K34Q3Z
last-modified: Thu, 30 Mar 2023 15:58:02 GMT
content-security-policy-report-only: default-src *; connect-src *; font-src * data:; frame-src *; img-src * data:; media-src *; object-src *; script-src *; style-src 'unsafe-inline' *; report-uri https://prod-us-west-2.csp-report.marketing.aws.dev/submit
vary: accept-encoding,Content-Type,Accept-Encoding,User-Agent
x-cache: Miss from cloudfront
via: 1.1 88c333921d5c405e037b84bb8c2dc33e.cloudfront.net (CloudFront)
x-amz-cf-pop: GRU3-P1
x-amz-cf-id: 89R1wtM9vYV0kIQXrEVkcoNzg_C3UfQJIEVkC5BA3xiIH3FD0nVnYw==
```

Congratulations! You have successfully connected to your public instance using Session Manager. You can safely close the tab and return to the console.

Task 9: Create a NAT gateway and configuring routing in the private subnet

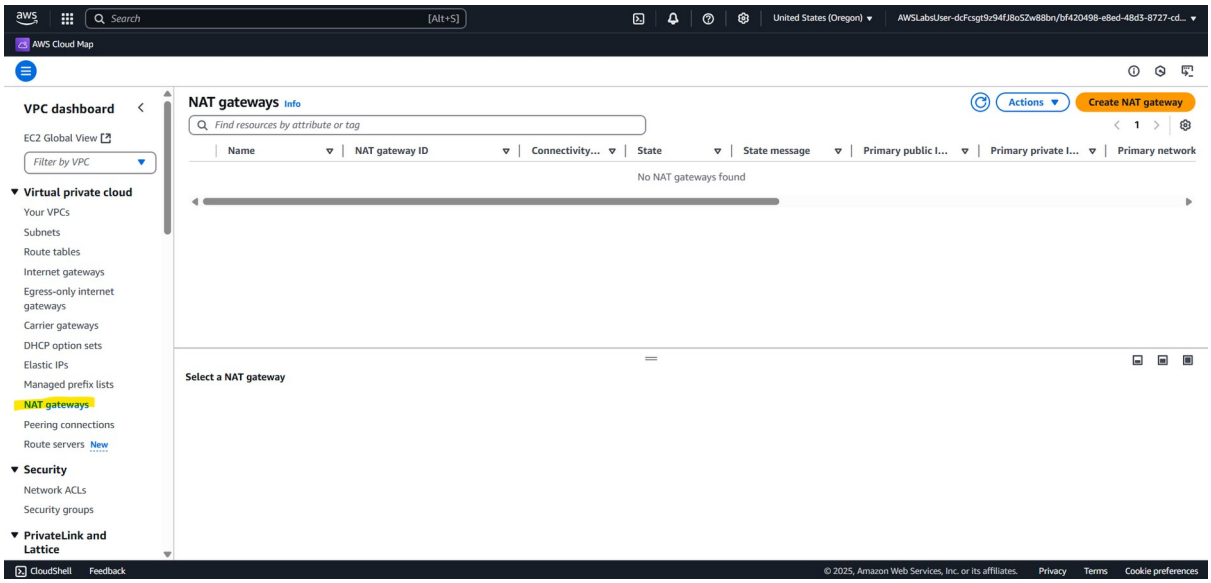
In this task, you create a NAT gateway and then create a route table to route non-local traffic to the NAT gateway. You then attach the route table to the private subnet. You can use a NAT gateway to allow instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.

Note: To create a NAT gateway, you must specify the public subnet in which the NAT gateway should reside. You must also specify an Elastic IP address to associate with the NAT gateway when you create it. You cannot change the Elastic IP address after you associate it with the NAT gateway. After you've created a NAT gateway, you must update the route table associated with one or more of your private subnets to point internet-bound traffic to the NAT gateway. This allows instances in your private subnets to communicate with the internet.

82. Return to the AWS Management Console browser tab.

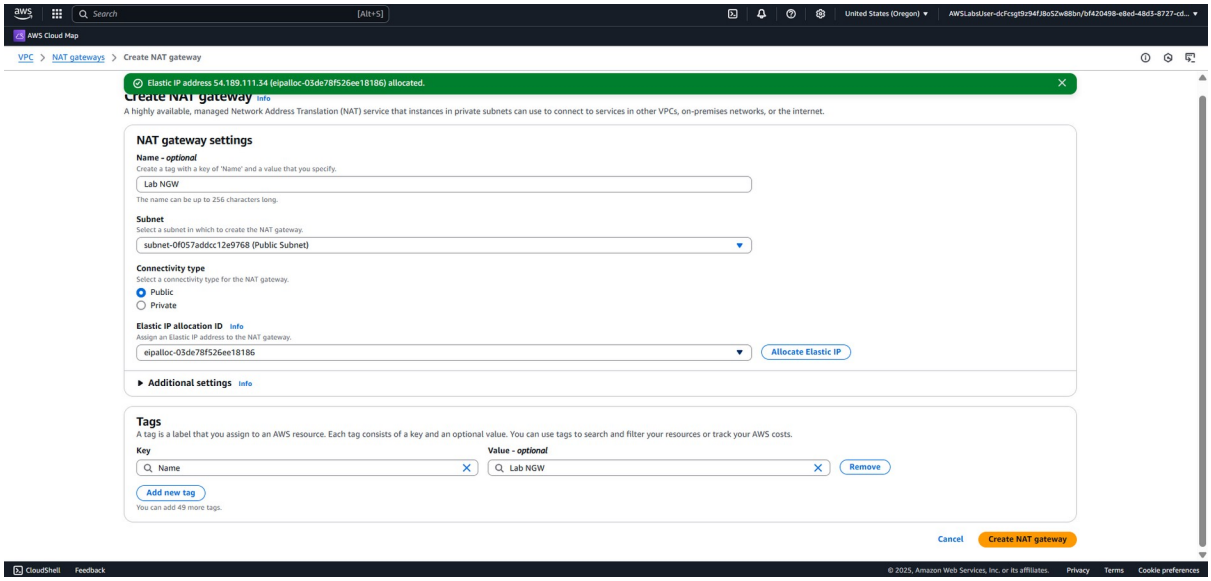
83. At the top of the AWS Management Console, in the search box, search for and choose VPC.

84. In the left navigation pane, choose **NAT gateways**.

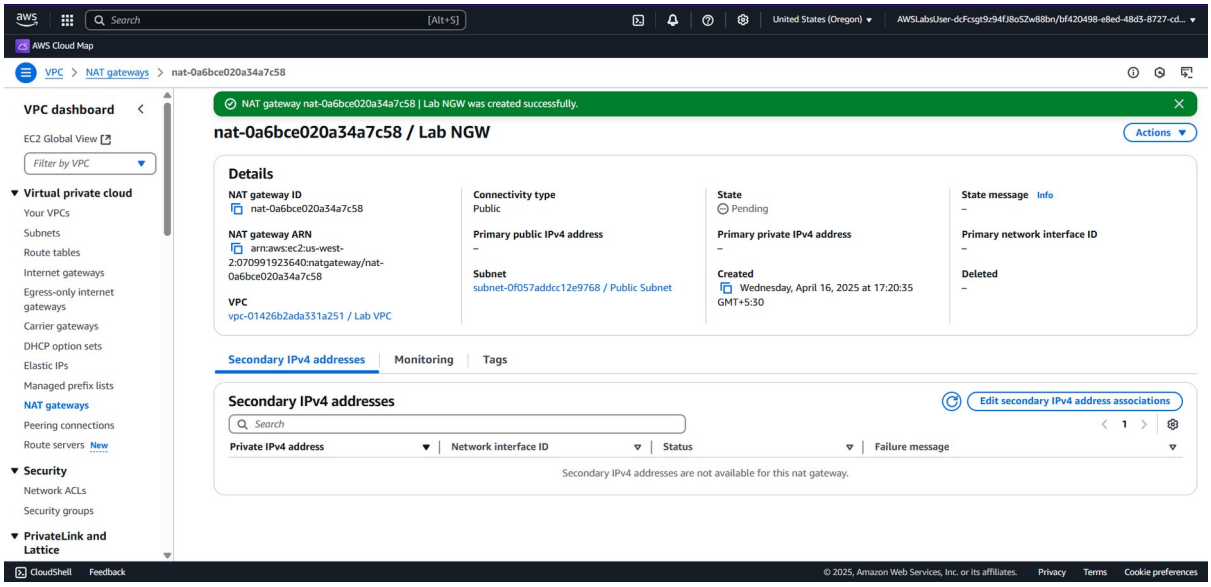


85. Choose **Create NAT gateway** and configure the following:

- **Name - *optional*:** Enter Lab NGW.
- **Subnet:** Select **Public Subnet** from the dropdown menu.
- For **Elastic IP allocation ID**, choose **Allocate Elastic IP**.



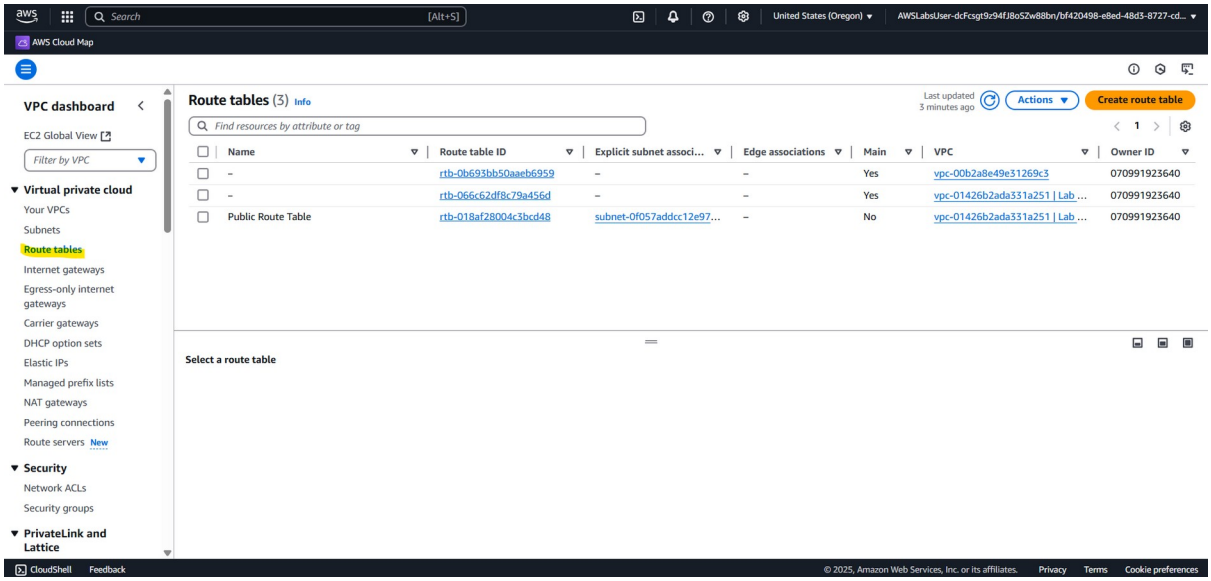
86. Choose **Create NAT gateway**.



A (NAT gateway nat-xxxxxxx | Lab NGW was created successfully.) message is displayed on top of the screen.

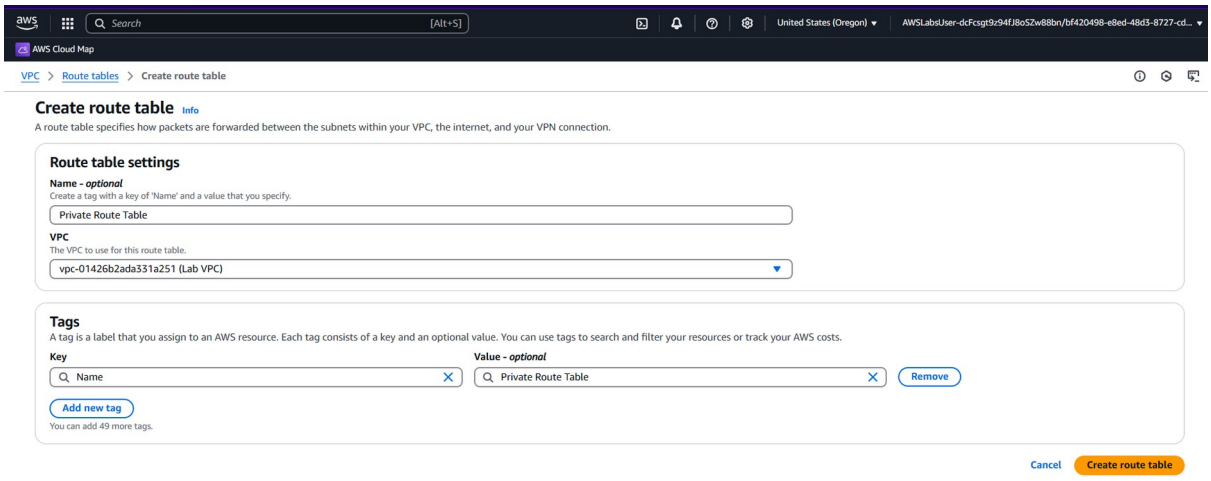
In the next step, you create a new route table for a private subnet that redirects non-local traffic to the NAT gateway.

87. In the left navigation pane, choose **Route tables**.

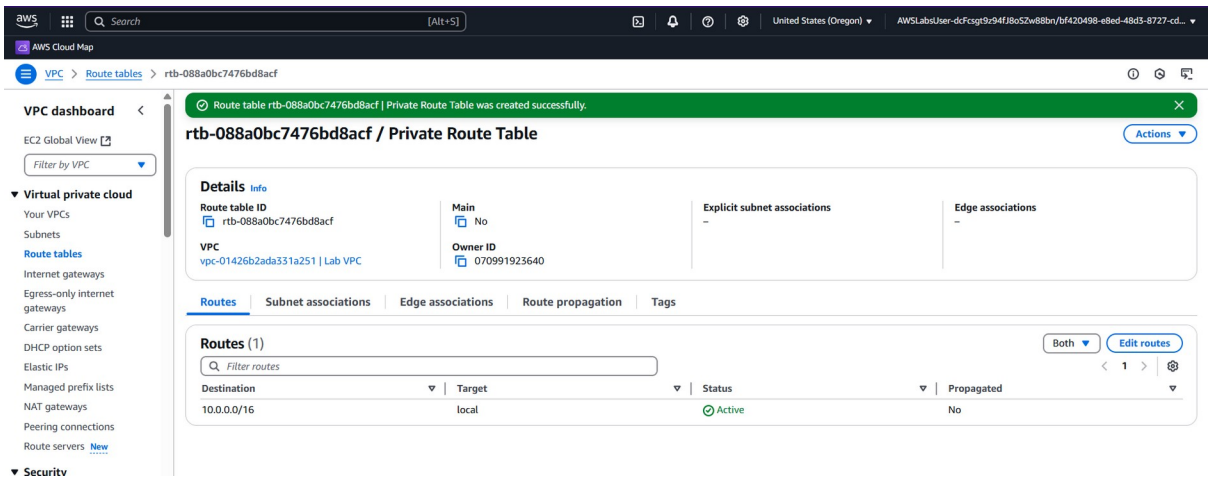


88. Choose **Create route table** and configure the following:

- **Name - *optional*:** Enter Private Route Table.
- **VPC:** Select **Lab VPC** from the dropdown menu.



89. Choose **Create route table**.



A (Route table rtb-xxxxxxx | Private Route Table was created successfully.) message is displayed on top of the screen.

The private route table is created and the details page for the private route table is displayed.

90. Choose the **Routes** tab.

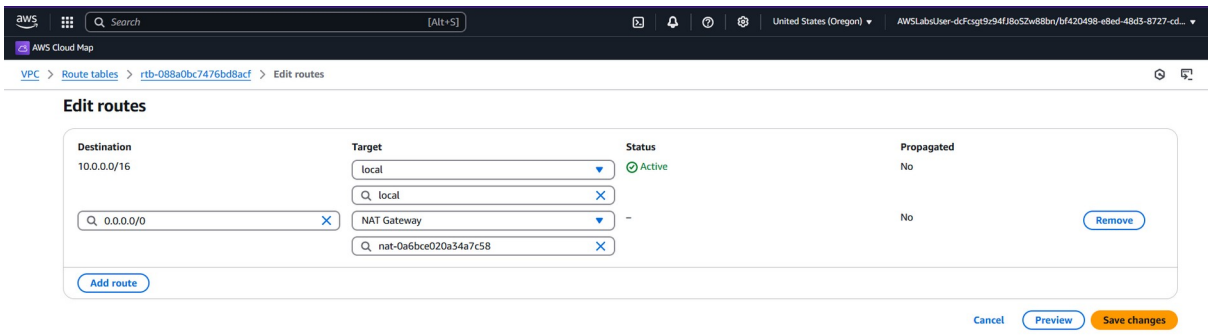
There is currently one route that directs all traffic *locally*.

You now add a route to send internet-bound traffic through the NAT gateway.

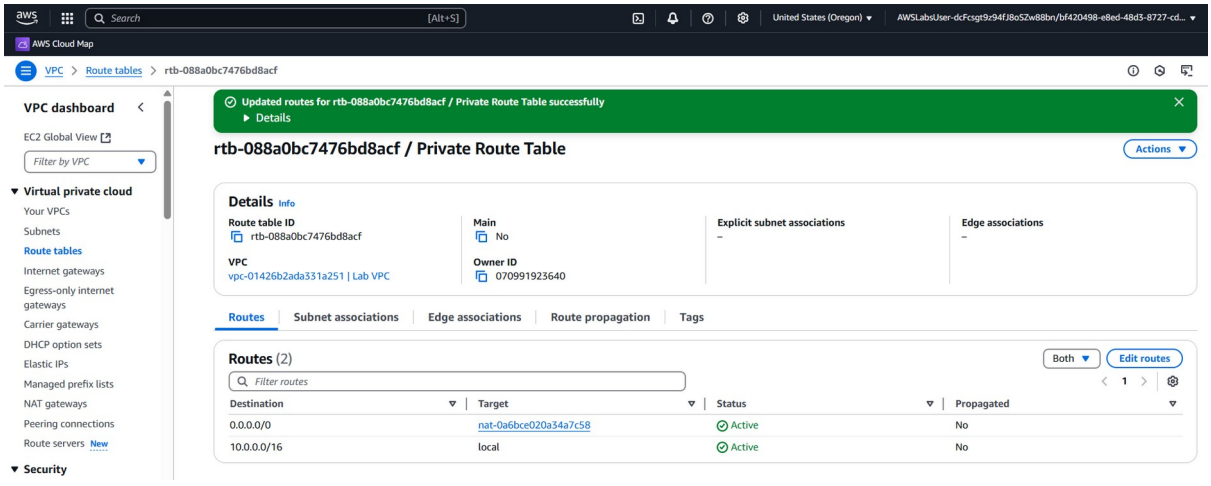
91. Choose **Edit routes**.

92. Choose **Add route** and then configure the following:

    - **Destination:** Enter 0.0.0.0/0.
    - **Target:** Choose **NAT Gateway** in the dropdown menu, and then choose the displayed NAT Gateway ID.



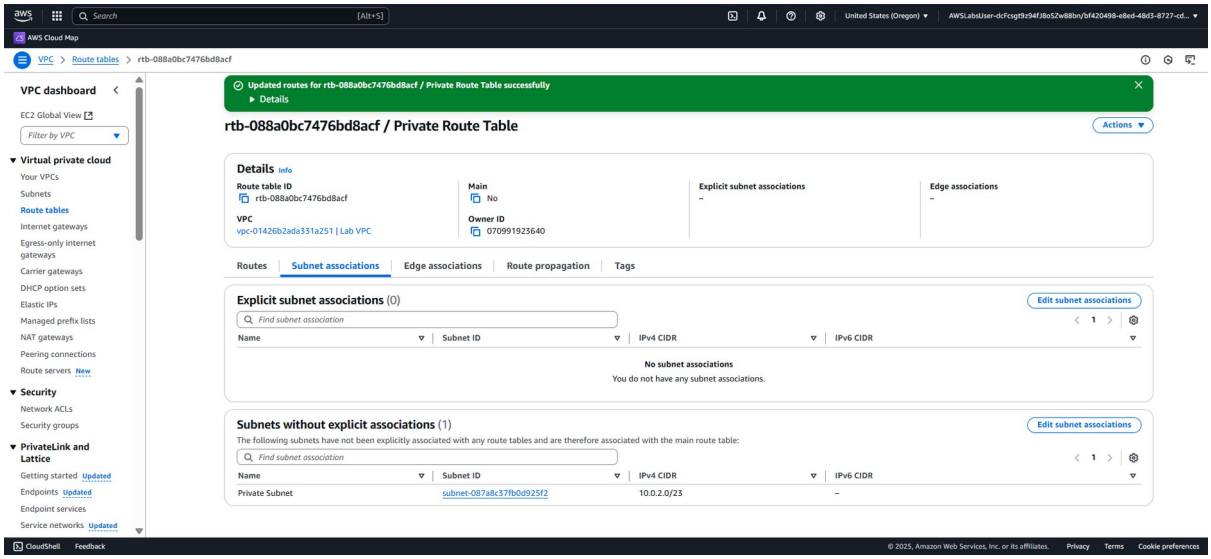
93. Choose **Save changes**.



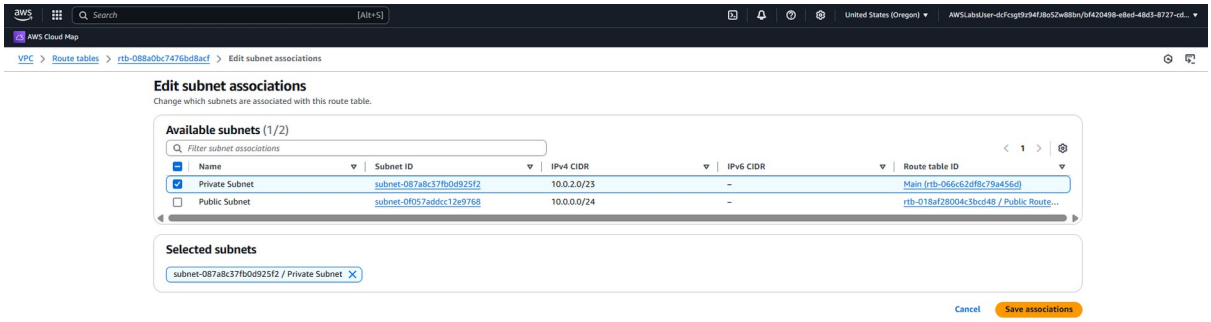
A **(Updated routes for rtb-xxxxxxx / Private Route Table successfully)** message is displayed on top of the screen.

94. Choose the **Subnet associations** tab.

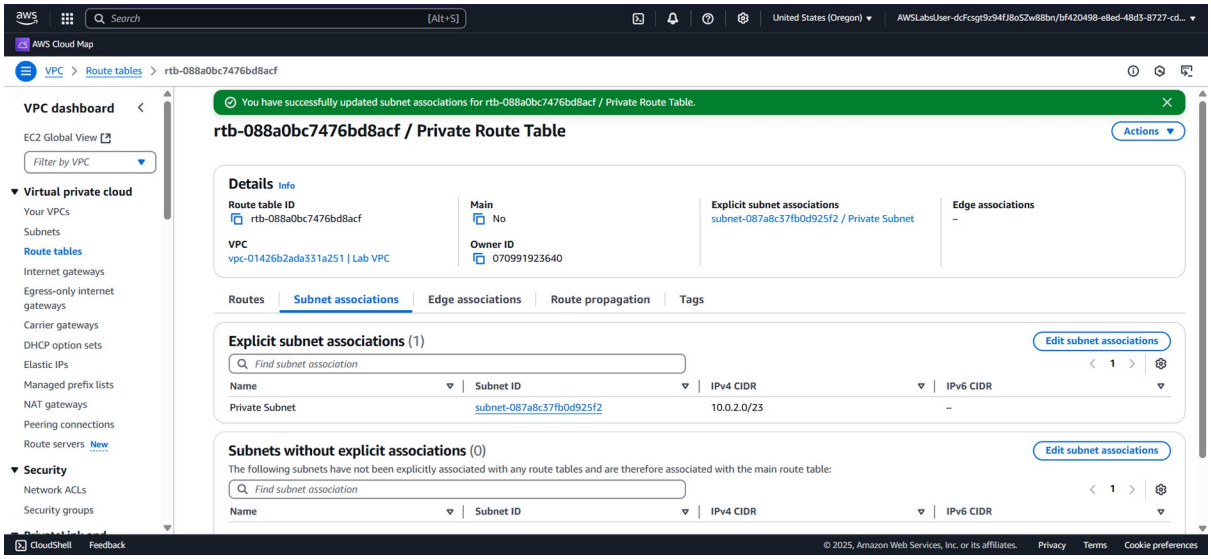
95. Choose **Edit subnet associations**.



96. Select **Private Subnet**.



97. Choose **Save associations**.



A (You have successfully updated subnet associations for rtb-xxxxxxx / Private Route Table.) message is displayed on top of the screen.

This route sends internet-bound traffic from the private subnet to the NAT gateway that is in the same Availability Zone.

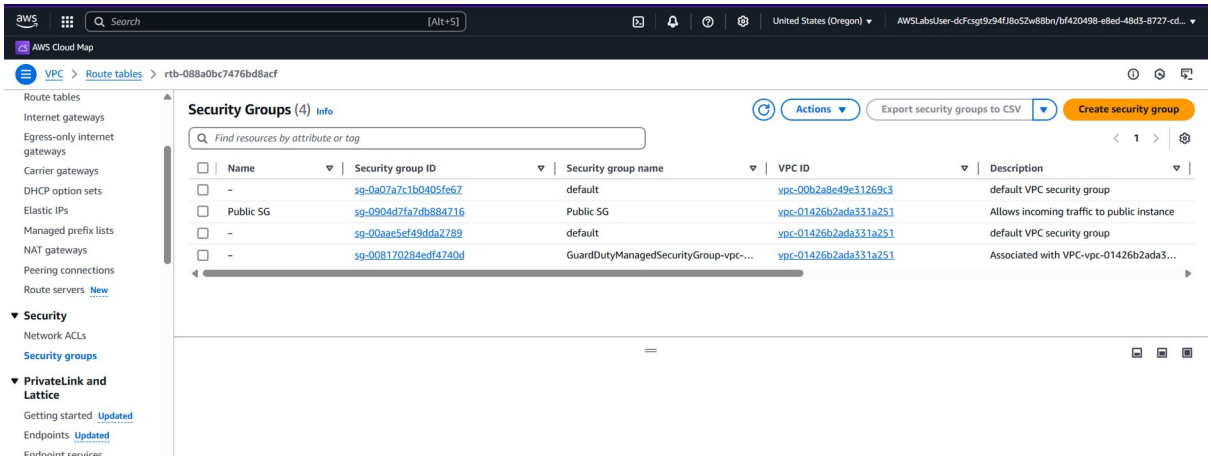
Congratulations! You have successfully created the NAT gateway and configured the private route table.

Task 10: Create a security group for private resources

In this task, you create a security group that allows incoming HTTP traffic from resources assigned to the public security group. In a multi-tiered architecture, resources in a private subnet are should not directly accessible from the internet, however there is a common use case to route web traffic from publicly accessible resources to private resources.

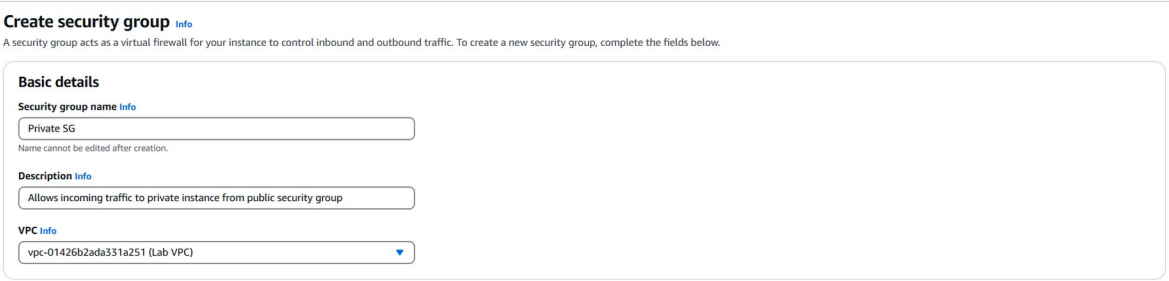
Learn more: When you specify a security group as the source for a rule, traffic is allowed from the network interfaces that are associated with the source security group for the specified port and protocol. Incoming traffic is allowed based on the private IP addresses of the network interfaces that are associated with the source security group (and not the public IP or Elastic IP addresses). Adding a security group as a source does not add rules from the source security group.

98. In the left navigation pane, choose **Security groups**.



99. Choose **Create security group**, and then configure the following:

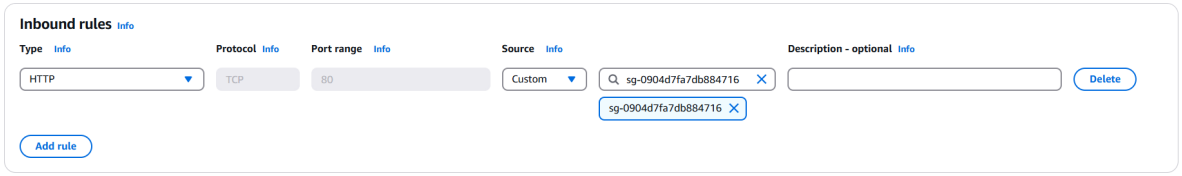
- **Security group name:** Enter Private SG.
- **Description:** Enter Allows incoming traffic to private instance from public security group.
- **VPC:** Select **Lab VPC** from the dropdown menu.



100. In the **Inbound rules** section, choose **Add rule** and configure the following:

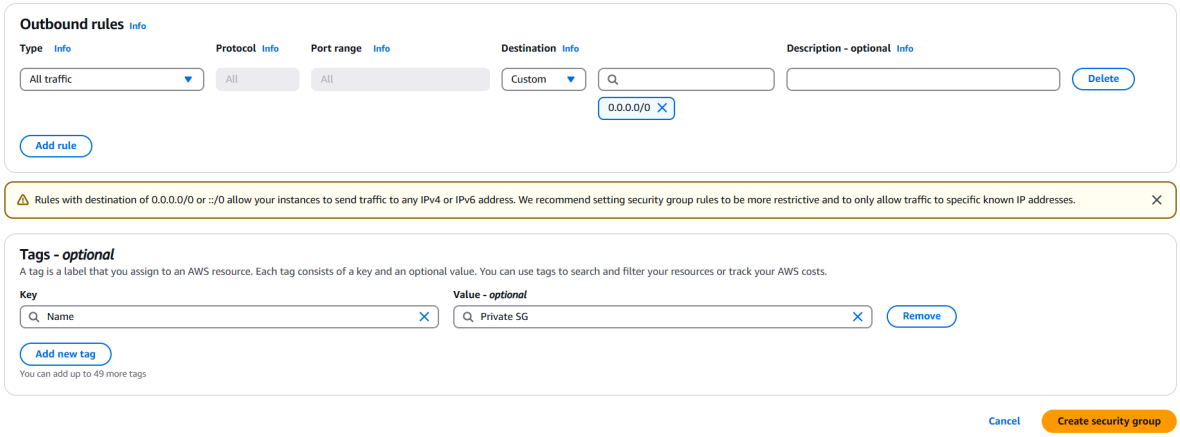
- **Type:** Select **HTTP**.

- **Source:** Select **Custom**.
    - In the box to the right of Custom, type sg.
    - Choose **Public SG** from the list.

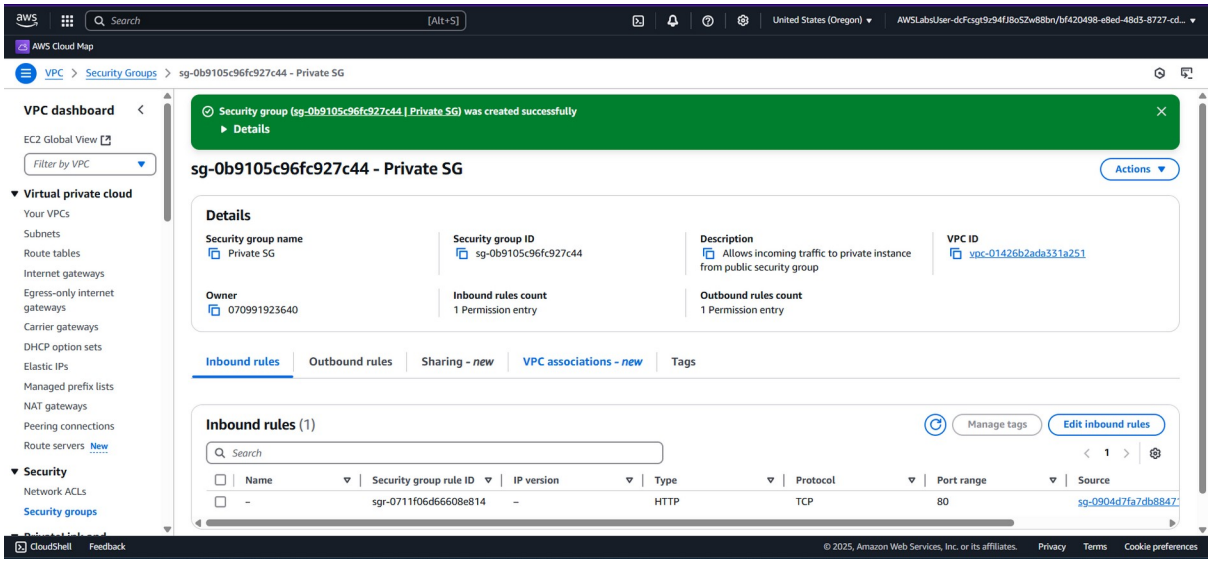


101. In the **Tags - *optional*** section, choose **Add new tag** and configure the following:

- **Key:** Enter Name.
- **Value:** Enter Private SG.



102. Choose **Create security group**.



A **(Security group (sg-xxxxxxx | Private SG) was created successfully)** message is displayed on top of the screen.

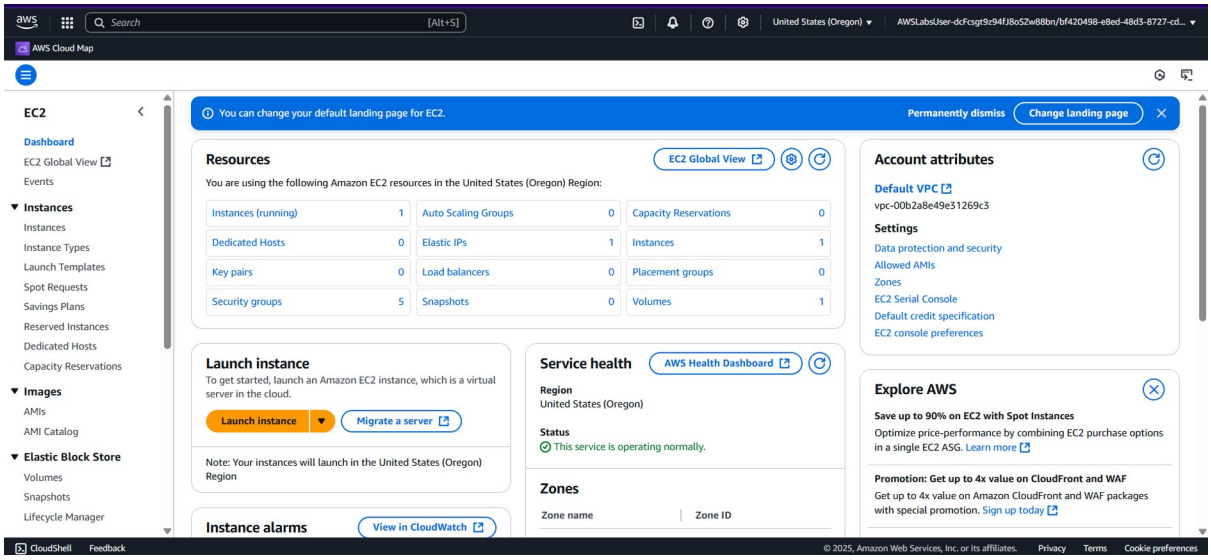
Congratulations! You have successfully created the private security group.

Task 11: Launch an Amazon EC2 instance into a private subnet

In this task, you launch an Amazon EC2 instance into a private subnet.

Learn more: Private instances can route their traffic through a NAT gateway or a NAT instance to access the internet. Private instances use the public IP address of the NAT gateway or NAT instance to traverse the internet. The NAT gateway or NAT instance allows outbound communication but doesn't allow machines on the internet to initiate a connection to the privately addressed instances.

103. At the top of the AWS Management Console, in the search bar, search for and choose EC2.

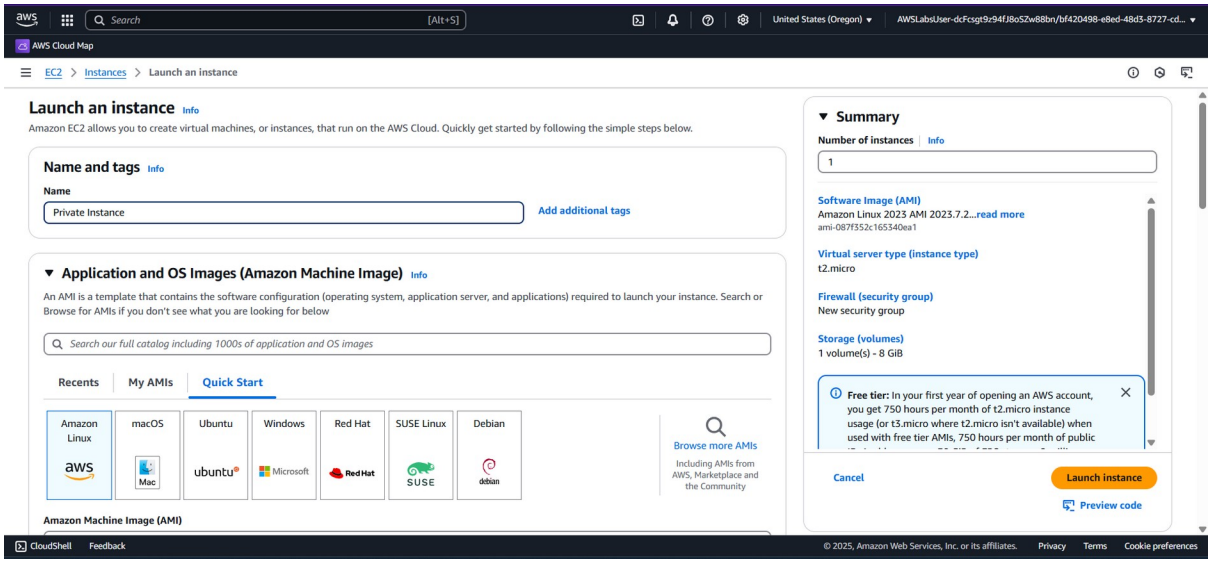


The **Amazon EC2 console** is displayed.

Task 11.1: Begin the instance configuration

104. Choose **EC2 Dashboard** from the console navigation menu on the left.

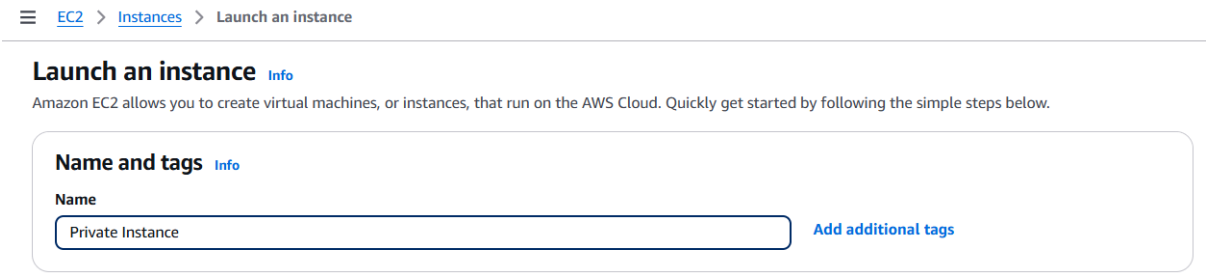
105. Choose **Launch instance** from the **Launch instance** section.



The **Launch an instance** page is displayed. In this task, you add a tag to the Amazon EC2 instance.

106. Locate the **Name and tags** section.

107. Enter Private Instance in the **Name** field.



Note: No additional instance tags are required for this lab.

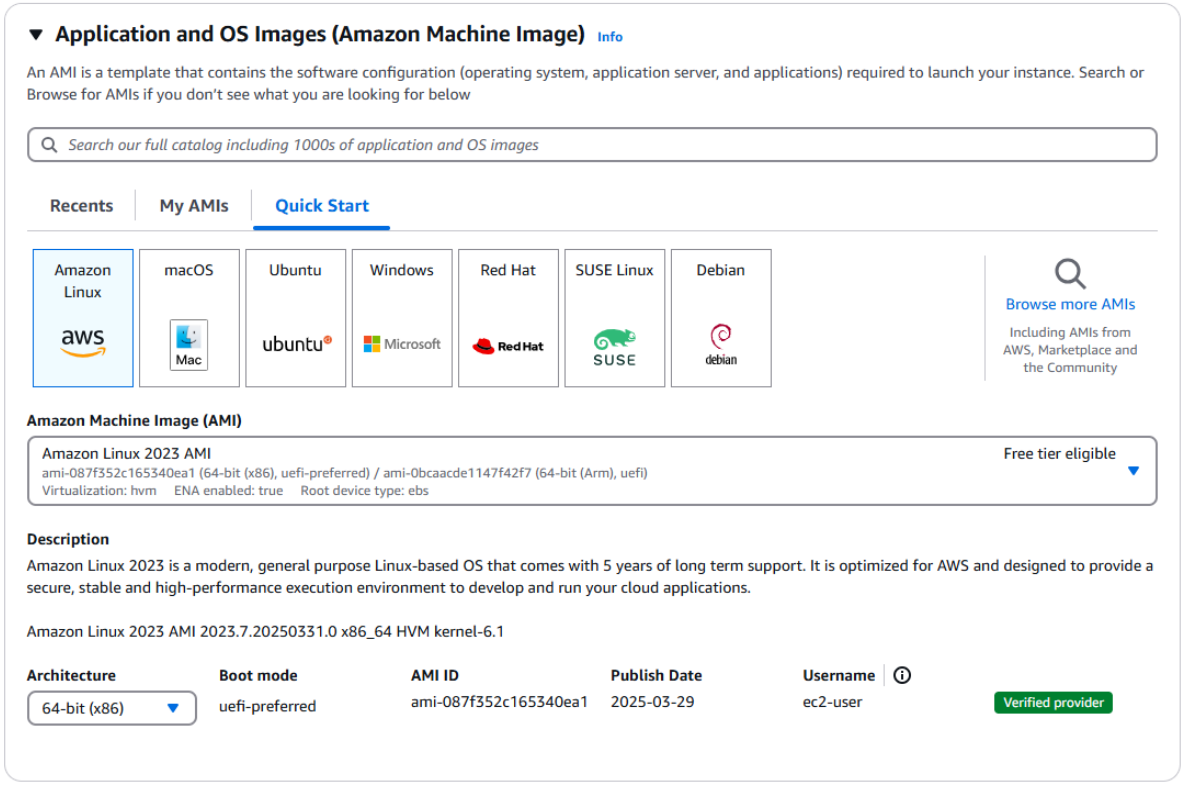
Task 11.3: Select an AMI

In this task, you choose an AMI. The AMI contains a copy of the disk volume used to launch the instance.

108. Locate the **Application and OS Images (Amazon Machine Image)** section.

109. Ensure that **Amazon Linux** is selected as the OS.

110. Ensure that **Amazon Linux 2023 AMI** is selected in the dropdown menu.



Task 11.4: Choose the Amazon EC2 instance type

Each instance type allocates a specific combination of vCPUs, memory, disk storage, and network performance.

For this lab, use a **t3.micro** instance type. This instance type has 2 vCPUs and 1 GiB of memory.

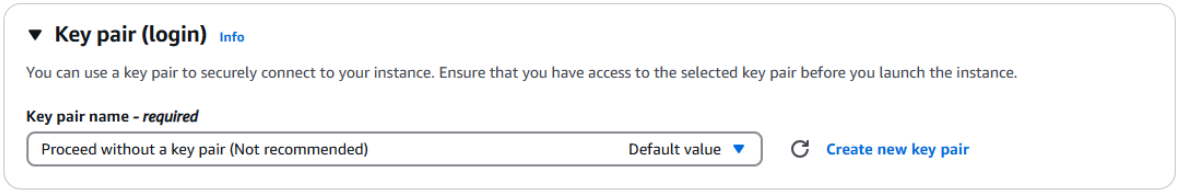
111. Locate the **Instance type** section.

112. Choose **t3.micro** from the **Instance type** dropdown menu.

Task 11.5: Configure key pair for login

113. Locate the **Key pair (login)** section.

114. Choose **Proceed without a key pair (Not recommended)** from the **Key pair name - *required*** dropdown menu.



Task 11.6: Configure instance networking

115. Locate the **Network settings** section.

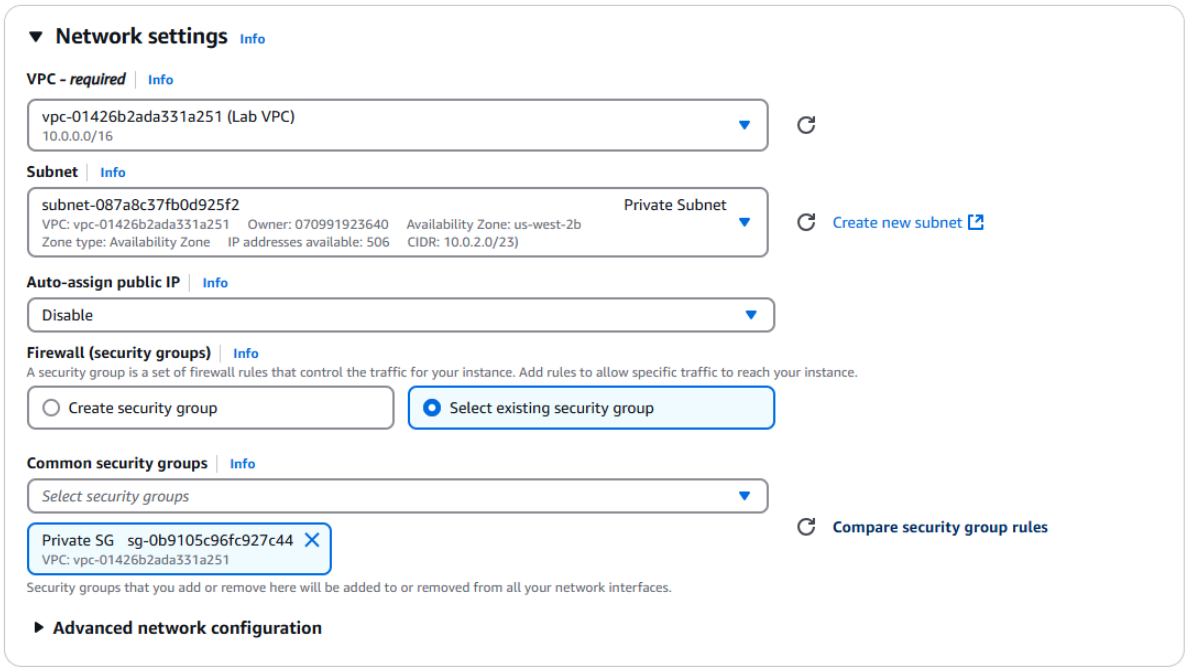
116. Choose **Edit** and configure the following settings from the dropdown menus:

- **VPC - *required:*** Select **Lab VPC**.
- **Subnet:** Select **Private Subnet**.
- **Auto-assign public IP:** Select **Disable**.

Task 11.7: Configure instance security groups

117. For **Firewall (security groups)**, choose **Select existing security group**

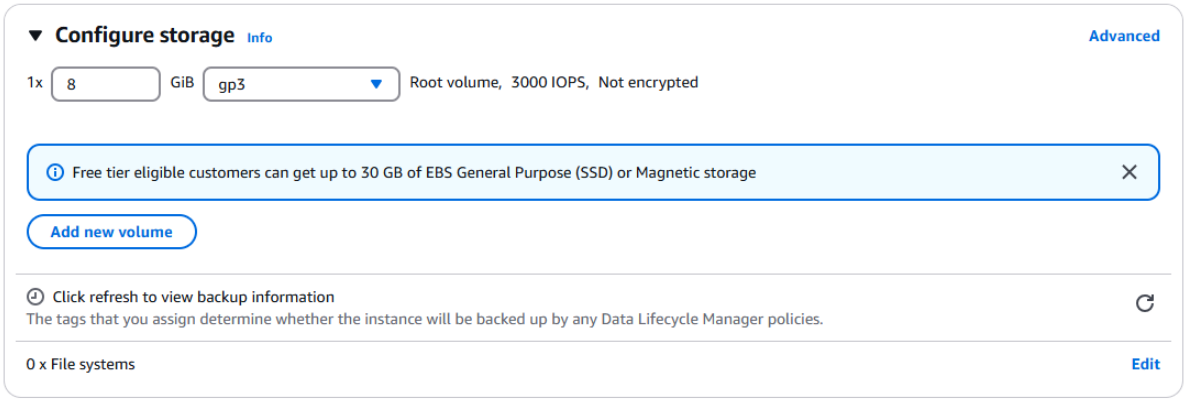
118. Choose the security group that has a name like **Private SG** from the **Common security groups** dropdown menu.



Task 11.8: Add storage

You can use the **Configure storage** section to specify or modify the storage options for the instance and add additional Amazon Elastic Block Store (Amazon EBS) disk volumes attached to the instance. The EBS volumes can be configured in both their size and performance.

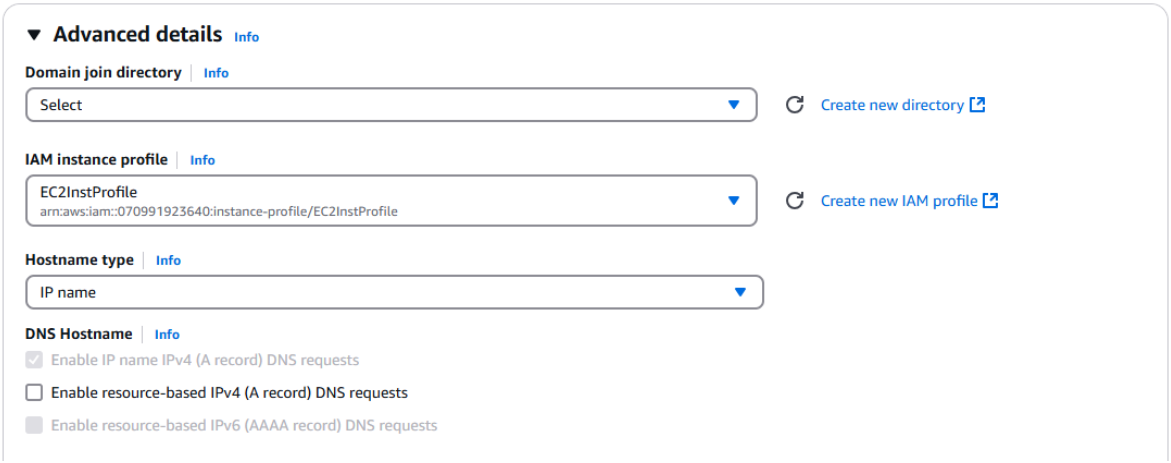
In this lab, the default storage settings are all that is needed. No changes are required.



Task 11.9: Configure the IAM instance profile

119. Locate and expand the **Advanced details** section.

120. From the **IAM instance profile** dropdown menu, select the role that has a name like **EC2InstProfile**.



The remaining settings on the page can be left at their default values.

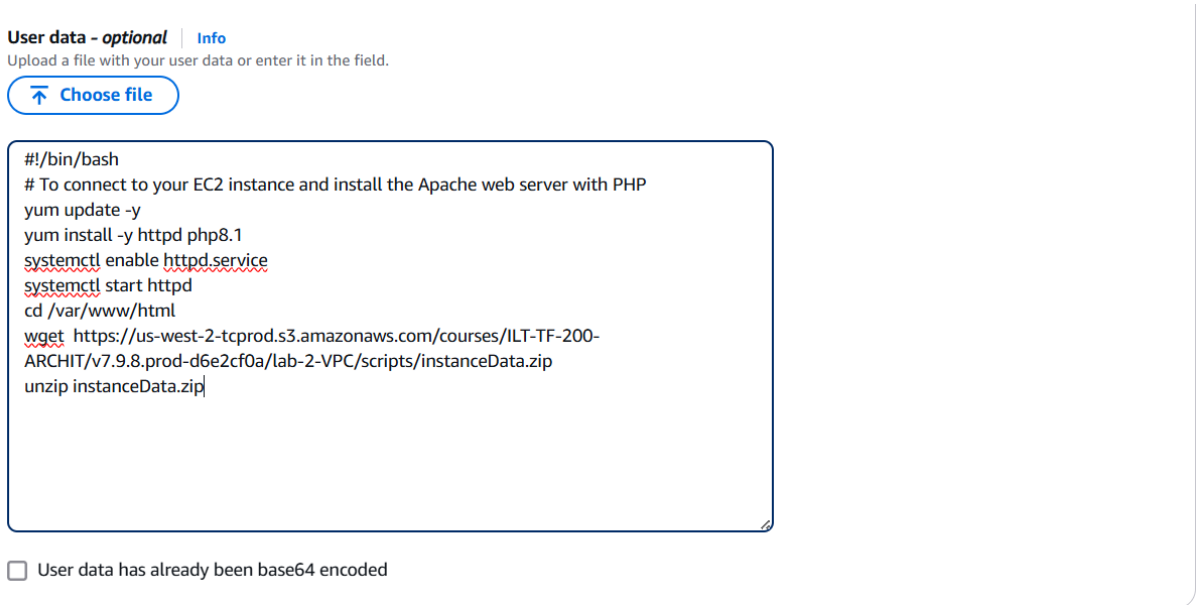
Task 11.10: Configure user data

121. Locate and expand the **Advanced details** section.

Note: To install and configure the new instance as a web server, you provide a user data script that automatically runs when the instance launches.

122. In the **User data - *optional*** section, copy and paste the following:

```
#!/bin/bash
# To connect to your EC2 instance and install the Apache web server with PHP
yum update -y
yum install -y httpd php8.1
systemctl enable httpd.service
systemctl start httpd
cd /var/www/html
wget  https://us-west-2-tcprod.s3.amazonaws.com/courses/ILT-TF-200-ARCHIT/v7.9.8.prod-d6e2cf0a/lab-2-VPC/scripts/instanceData.zip
unzip instanceData.zip
```



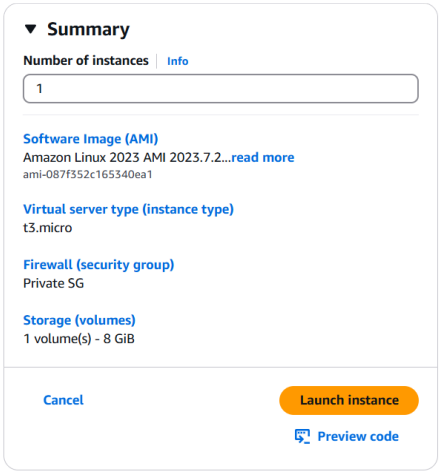
The remaining settings on the page can be left at their default values.

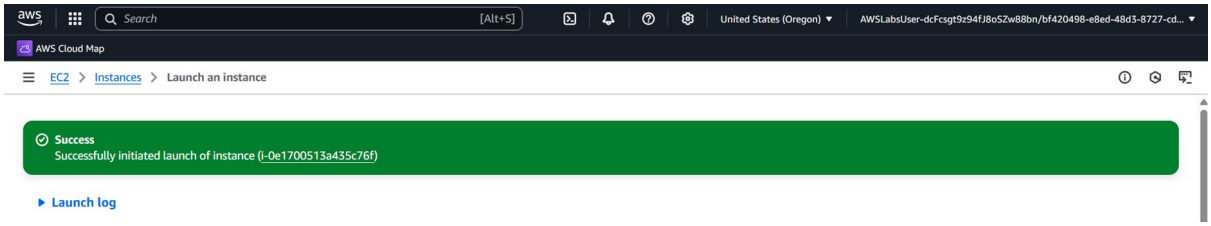
Task 11.11: Review the instance launch

Take a moment to review that the configuration for the Amazon EC2 instance you are about to launch is correct.

123. Locate the **Summary** section.

124. Choose **Launch instance**.

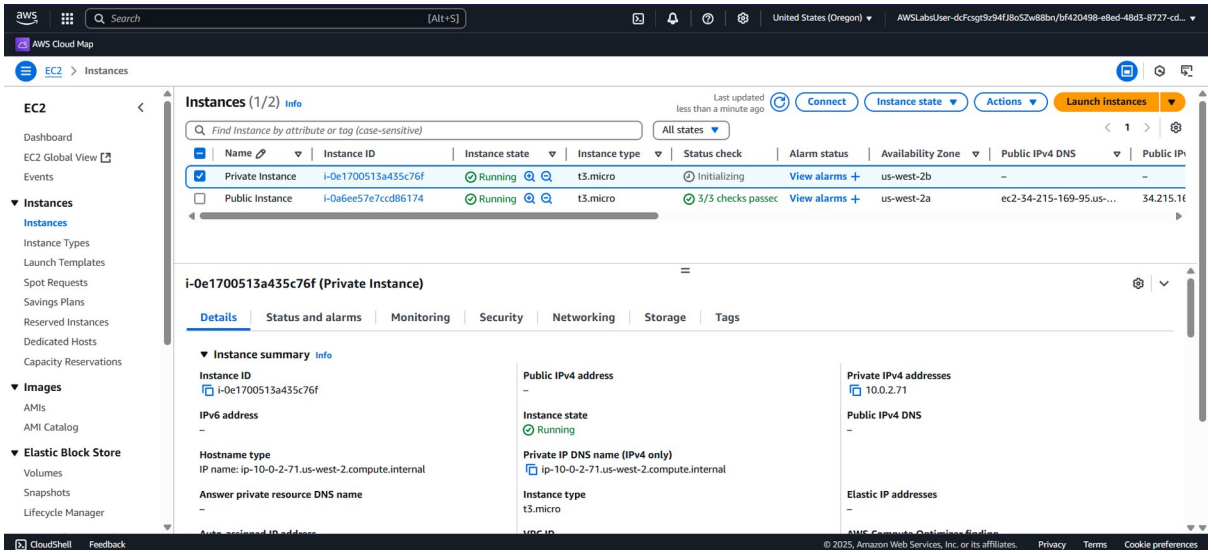




The **Launch an instance** page is displayed.

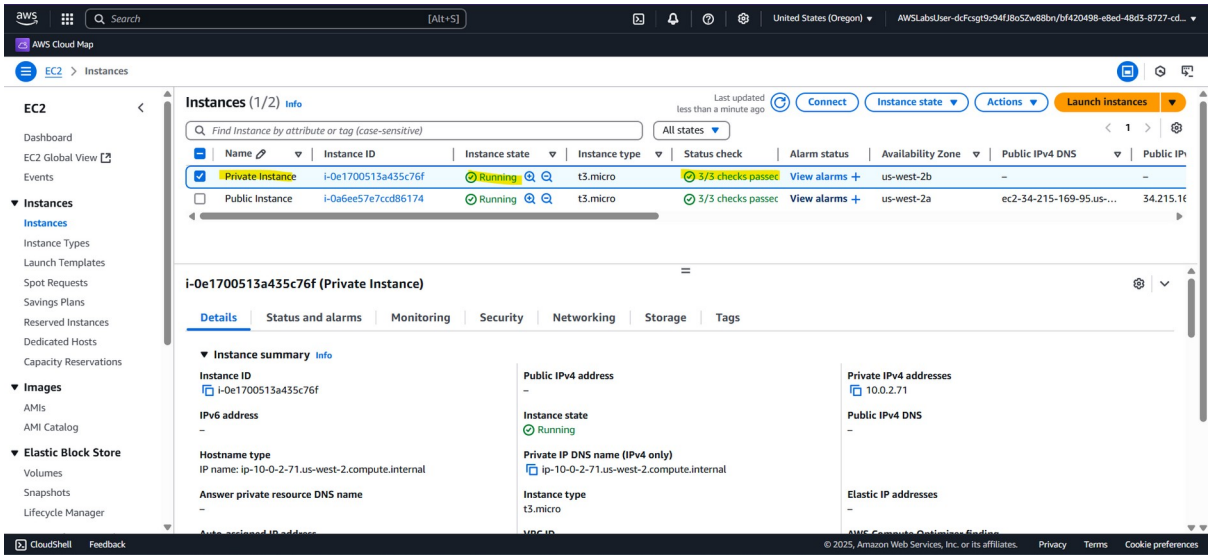
Your Amazon EC2 instance is now launched and configured as you specified.

125. Choose **View all instances**.



The **Amazon EC2 console** is displayed.

126. Occasionally choose the console refresh button and wait for **Private Instance** to display the **Instance state** as Running and wait for Status check to pass 3/3 checks passed.



Note: The Amazon EC2 instance named Private Instance is initially in a *Pending* state. The instance state then changes to Running indicating that the instance has finished booting.

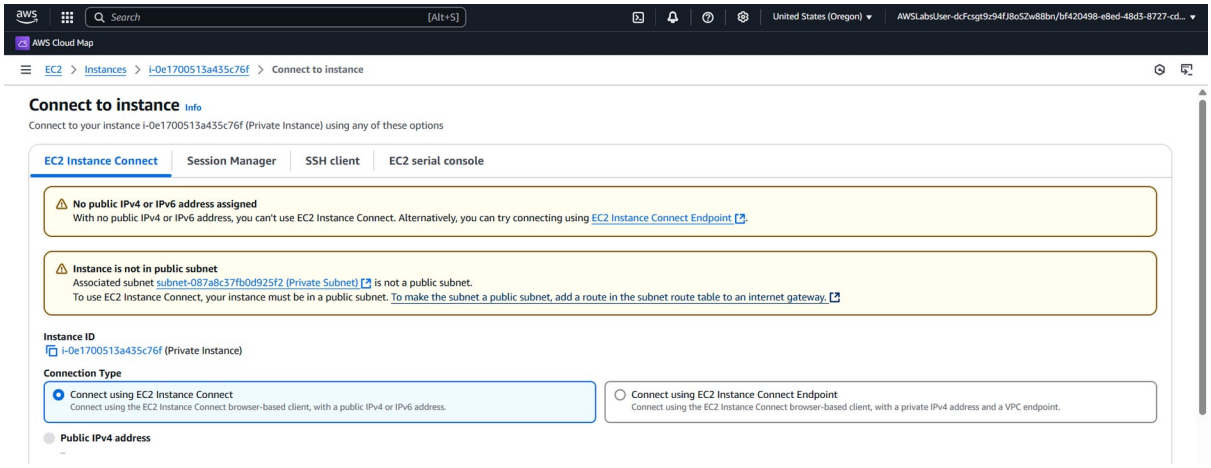
Congratulations! You have successfully launched an Amazon EC2 instance into a private subnet.

Task 12: Connect to the Amazon EC2 instance in the private subnet

In this task, you connect to the Amazon EC2 instance in the private subnet using Session Manager.

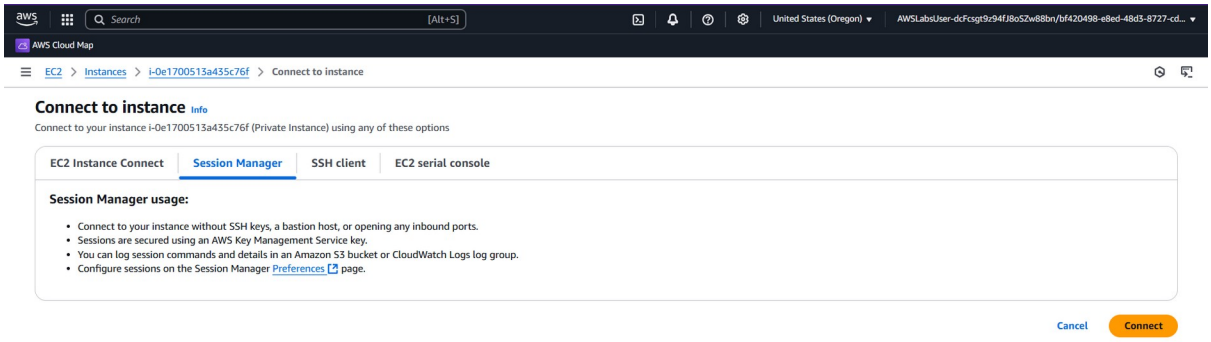
127. In the left navigation pane, choose **Instances**.

128. Select **Private Instance** and choose **Connect**.



The **Connect to instance** page is displayed.

129. Choose the **Session Manager** tab.



130. Choose **Connect**.

A new browser tab or window opens with a connection to the **Private Instance**.

Note: The Session Manager service is not updated in real time. If you experience errors with Session Manager connecting to an Amazon EC2 instance you just launched, ensure that you have given the instance a few minutes to launch, pass health checks, and communicate with the Session Manager service before trying to open a session connection again.

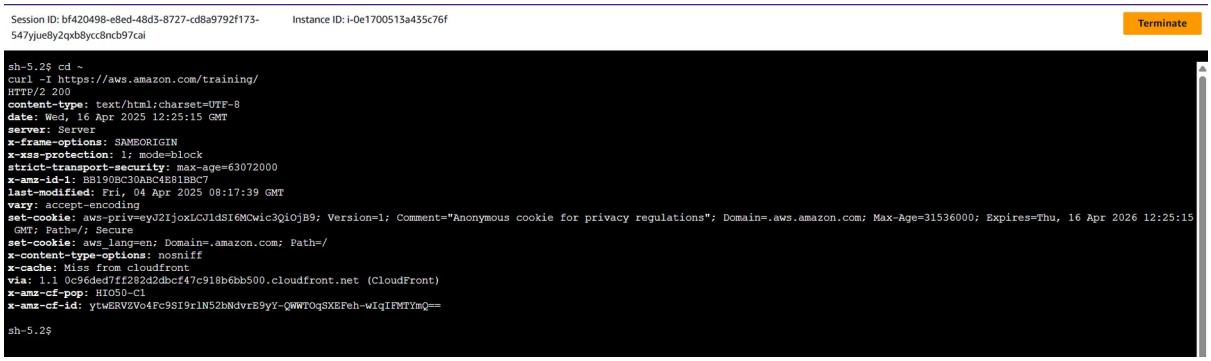
131. **Command:** Enter the following command to change to the home directory (/home/ssm-user/) and test web connectivity using the cURL command:

cd ~
curl -I https://aws.amazon.com/training/

Expected output:

HTTP/2 200

content-type: *text/html;charset=UTF-8*
server: *Server*
date: *Wed, 19 Apr 2023 14:59:09 GMT*
x-amz-rid: *AZPXJ57K93ERATZV588Z*
set-cookie: *aws-priv=eyJ2IjoxLCJldSI6MCwic3QiOjB9; Version=1; Comment="Anonymous cookie for privacy regulations"; Domain=.aws.amazon.com; Max-Age=31536000; Expires=Thu, 18-Apr-2024 14:59:08 GMT; Path=/; Secure*
set-cookie: *aws_lang=en; Domain=.amazon.com; Path=/*
x-frame-options: *SAMEORIGIN*
x-xss-protection: *1; mode=block*
strict-transport-security: *max-age=63072000*
x-content-type-options: *nosniff*
x-amz-id-1: *AZPXJ57K93ERATZV588Z*
last-modified: *Thu, 30 Mar 2023 15:58:02 GMT*
content-security-policy-report-only: *default-src *; connect-src *; font-src * data:; frame-src *; img-src * data:; media-src *; object-src *; script-src *; style-src 'unsafe-inline' *; report-uri https://prod-us-west-2.csp-report.marketing.aws.dev/submit*
vary: *accept-encoding,Content-Type,Accept-Encoding,User-Agent*
x-cache: *Miss from cloudfront*
via: *1.1 fb6a4eca9caced7b791557c24b8c6606.cloudfront.net (CloudFront)*
x-amz-cf-pop: *GRU3-P1*
x-amz-cf-id: *Tjphb1UhSXmtyHvybuq4QIFwzTurEI0g_saLB2nLjlYRiBbHbqn85Q==*



132. Close the Session Manager tab and return to the console.

Congratulations! You have successfully connected to a private instance using Session Manager.

(Optional) Task 1: Troubleshooting connectivity between the private instance and the public instance

In this optional task, you use the Internet Control Message Protocol (ICMP) to validate a private instance's network reachability from the public instance.

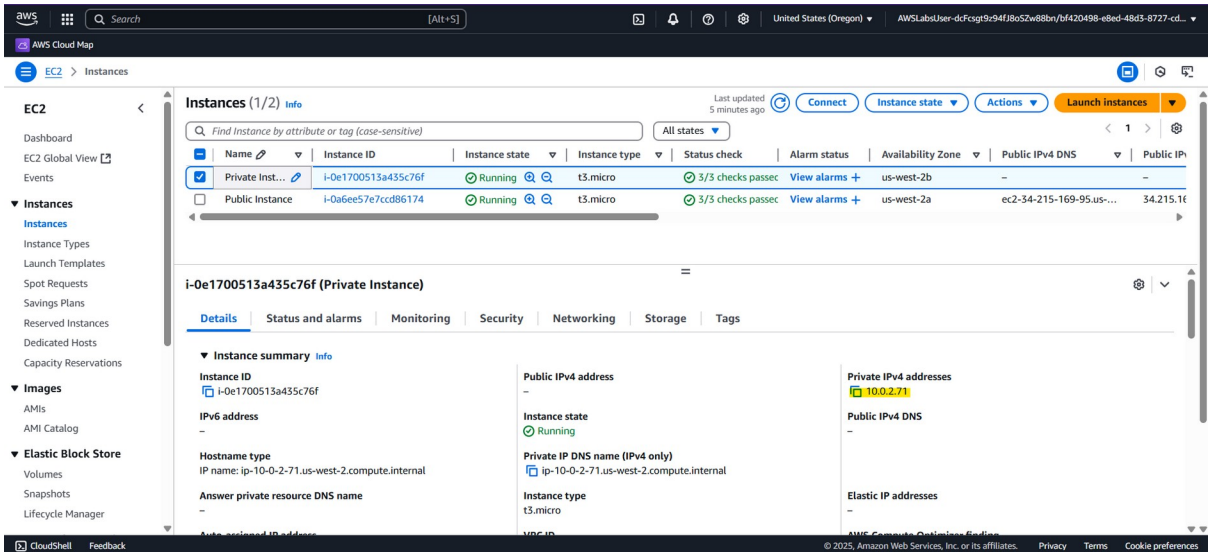
Note: This task is **optional** and is provided in case you have lab time remaining. You can complete this task or skip to the end of the lab.

133. Return to the AWS Management Console browser tab.

134. In the left navigation pane, choose **Instances**.

135. Select **Private Instance**.

136. On the **Details** tab, copy the value of **Private IPv4 addresses** to your clipboard.

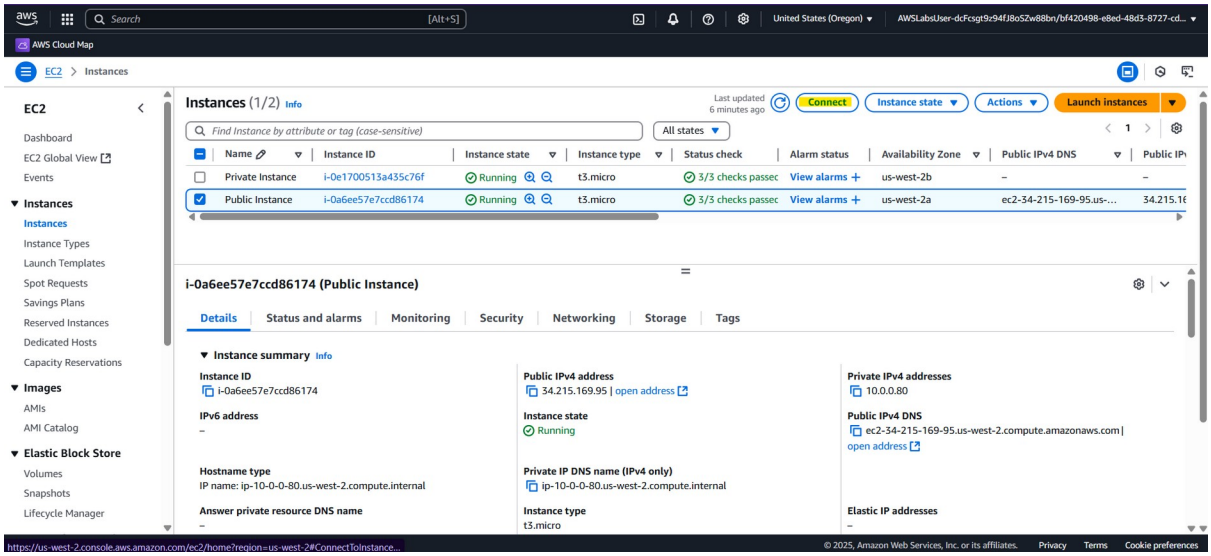


Note: To copy the private IPv4 address, hover over it and choose the copy icon.

137. Unselect **Private Instance**.

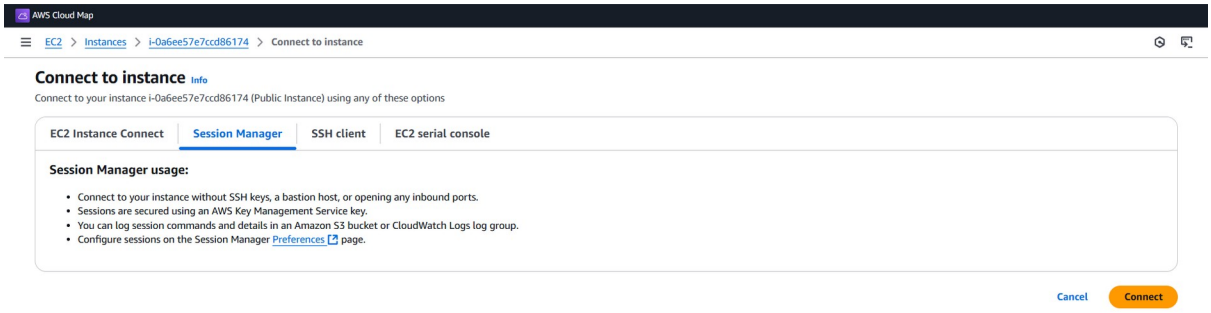
138. Select **Public Instance**.

139. Choose **Connect**.



The **Connect to instance** page is displayed.

140. Choose the **Session Manager** tab.



141. Choose **Connect**.

A new browser tab or window opens with a connection to the **Public Instance**.

First, use a *curl* command to retrieve a header file and confirm is the web app hosted on the private instance is reachable from the public instance.

142. **Command:** Copy the following command to your notepad. Replace **PRIVATE_IP** with the value of the **Private IPv4 address** for the **Private Instance**:

curl PRIVATE_IP

Example: *curl 10.0.2.71*

Expected output:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
    <html xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Amazon EC2 InstanceA</title>
    <link rel="stylesheet" href="css/screen.css" type="text/css" media="screen" title="default" />
    </head>
    <body>
    <div id="content-outer">
    <!-- start content -->
    <center>
    <div id="content">
    <table border="0" width="50%" cellpadding="0" cellspacing="0" id="content-table">
    <tr>
        <th rowspan="3" class="sized"><img src="images/shared/side_shadowleft.jpg" width="20"
height="300" alt="" /></th>
        <th class="topleft"></th>
        <td id="tbl-border-top"> </td>
        <th class="topright"></th>
        <th rowspan="3" class="sized"><img src="images/shared/side_shadowright.jpg" width="20"
height="300" alt="" /></th>
    </tr>
    <tr>
        <td id="tbl-border-left"></td>
        <td>
        <!--  start content-table-inner ...................................................................... START -->
```

```
        <div id="content-table-inner">

            <!--  start table-content  -->
            <div id="table-content">



            <center>
                <img src="images/reinvent_mark_white.png" width="300"/>
                <br/>
                <br/>
                <h2>EC2 Instance ID: i-01a5b999bd67645cf</h2>
                <h2>Zone: us-east-1a</h2>

            </center>

            </div>
            <!--  end table-content  -->

            <div class="clear"></div>

        </div>
        <!--  end content-table-inner ............................................END  -->
        </td>
        <td id="tbl-border-right"></td>
</tr>
<tr>
        <th class="sized bottomleft"></th>
        <td id="tbl-border-bottom"> </td>
        <th class="sized bottomright"></th>
</tr>
</table>
<div class="clear"> </div>
</div>
</center>
<!--  end content -->
<div class="clear"> </div>
</div>
<!--  end content-outer........................................................END -->
<div class="clear"> </div>
</body>
</html>sh-5.2$
```

Session ID: bf420498-e8ed-48d3-8727-cd8a9792f173-8cdd599ovkhnuxy3crpkbayjeq

Instance ID: i-0a6ee57e7ccd86174

```
sh-5.2$ curl 10.0.2.71
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Amazon EC2 InstanceA</title>
<link rel="stylesheet" href="css/screen.css" type="text/css" media="screen" title="default" />

</head>
<body>


<div id="content-outer">
<!-- start content -->
<center>
<div id="content">
       <table border="0" width="50%" cellpadding="0" cellspacing="0" id="content-table">
       <tr>
              <th rowspan="3" class="sized"><img src="images/shared/side_shadowleft.jpg" width="20" height="300" alt="" /></th>
              <th class="topleft"></th>
              <td id="tbl-border-top"> </td>
              <th class="topright"></th>
              <th rowspan="3" class="sized"><img src="images/shared/side_shadowright.jpg" width="20" height="300" alt="" /></th>
       </tr>
       <tr>
              <td id="tbl-border-left"></td>
              <td>
              <!--  start content-table-inner ...................................................................... START -->
              <div id="content-table-inner">

                     <!--  start table-content  -->
                     <div id="table-content">




                     <center>
                            <img src="images/reinvent_mark_white.png" width="300"/>
                            <br/>
                            <br/>
                            <h2>EC2 Instance ID: i-0e1700513a435c76f</h2>
                            <h2>Zone: us-west-2b</h2>

                     </center>

                     </div>
                     <!--  end table-content  -->
```

```
                     <!--  end table-content  -->

                     <div class="clear"></div>

              </div>
              <!--  end content-table-inner ............................................END  -->
              </td>
              <td id="tbl-border-right"></td>
       </tr>
       <tr>
              <th class="sized bottomleft"></th>
              <td id="tbl-border-bottom"> </td>
              <th class="sized bottomright"></th>
       </tr>
       </table>
       <div class="clear"> </div>

</div>
</center>
<!--  end content -->
<div class="clear"> </div>
</div>
<!--  end content-outer........................................................END -->

<div class="clear"> </div>
</body>
sh-5.2$
```

143. **Command:** Copy the following command to your notepad. Replace **PRIVATE_IP** with the value of the **Private IPv4 address** for the **Private Instance**:

ping PRIVATE_IP

Example: *ping 10.0.2.71*

144. **Command:** Copy and paste the updated command in your terminal and press **Enter**.
145. After a few seconds, stop the ICMP ping request by pressing CTRL+C.

The ping request to the private instance fails. Your challenge is to use the console and figure out the correct *inbound rule* required in the **Private SG** to be able to successfully ping the private instance.

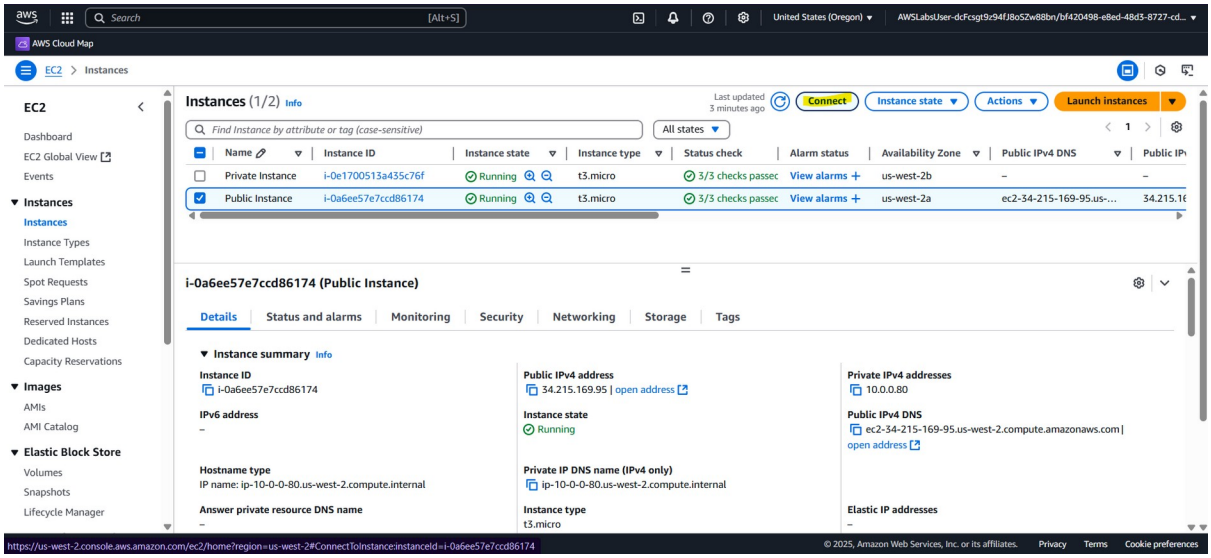
If you have trouble completing the optional task, refer to the Optional Task Solution section at the end of the lab.

(Optional) Task 2: Retrieving instance metadata

In this optional task, you run instance metadata commands on AWS CLI using a tool such as cURL. Instance metadata is available from your running Amazon EC2 instance. This can be helpful when you write scripts to run from your Amazon EC2 instance.

Note: This task is **optional** and is provided in case you have lab time remaining. You can complete this task or skip to the end of the lab .

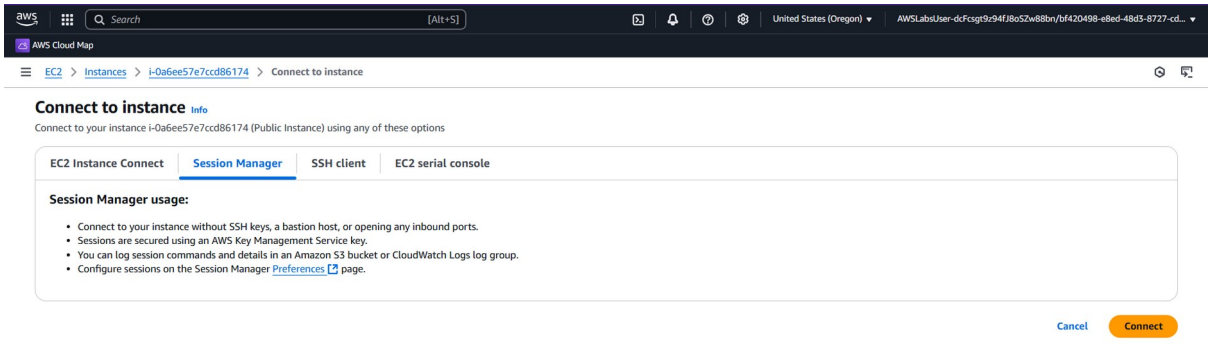
146. Return to the browser tab with the AWS Management Console open.
147. In the left navigation pane, choose **Instances**.
148. Select **Public Instance**.
149. Choose **Connect**.



The **Connect to instance** page is displayed.

150. Choose the **Session Manager** tab.

151. Choose **Connect**.



A new browser tab or window opens with a connection to the **Public Instance**.

152. **Command:** To view all categories of instance metadata from within a running instance, run the following command:

*TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` *
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/

That command is used to **retrieve instance metadata** on an **Amazon EC2 instance** using **IMDSv2 (Instance Metadata Service v2)**. Let's break it down step by step:

Command 1
*TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" *
-H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`

This part **requests a temporary token** for accessing metadata.

- *curl -X PUT*: This is an HTTP PUT request (required by IMDSv2 for tokens).
- *"http://169.254.169.254/latest/api/token":* The **IMDSv2 token endpoint**.

- *-H "X-aws-ec2-metadata-token-ttl-seconds: 21600":*
    - This header asks AWS to issue a token valid for **6 hours** (21,600 seconds).
- The command wraps the result with backticks (` `), so the token is stored in the TOKEN variable.

📦 **Result:** You now have a token saved in $TOKEN.

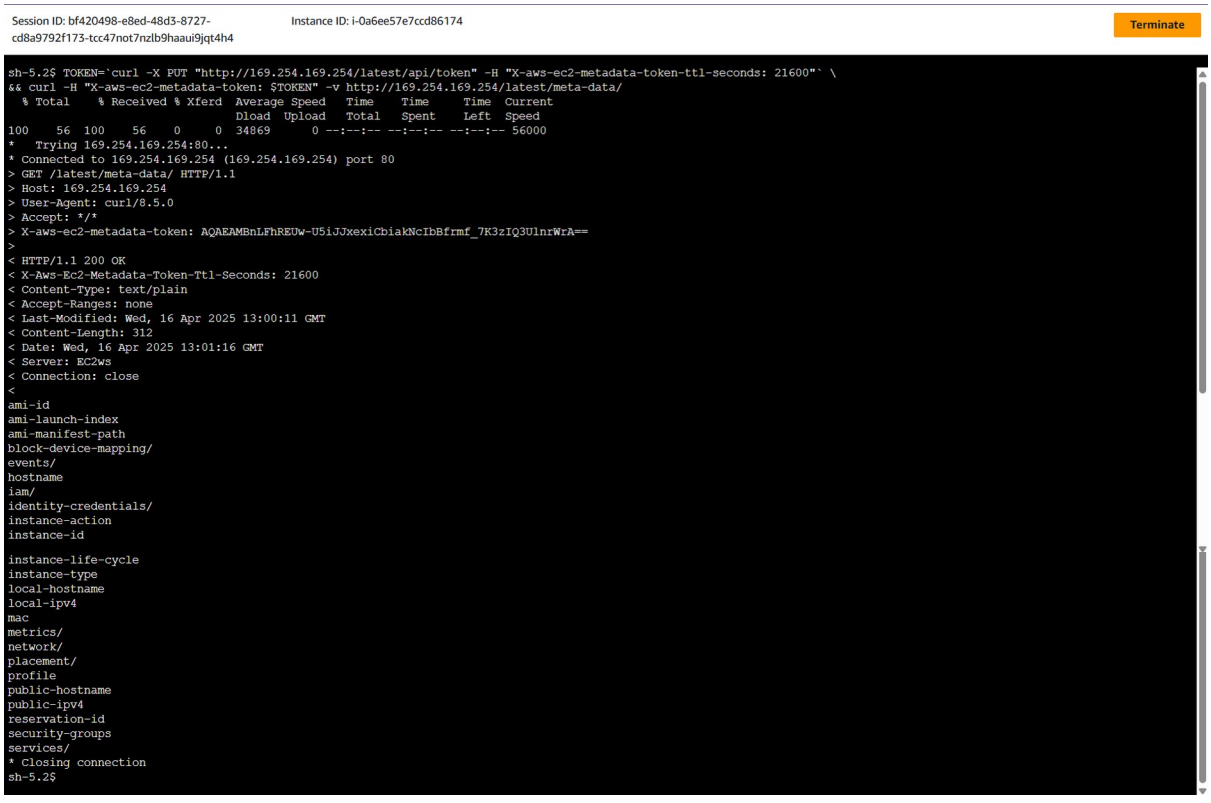
Command 2:

*curl -H "X-aws-ec2-metadata-token: $TOKEN" *

-v http://169.254.169.254/latest/meta-data/

- *-H "X-aws-ec2-metadata-token: $TOKEN":*
    - This header **authenticates** the metadata request using the token from step 1.
- *-v:* Verbose output, useful for debugging (shows headers, connection status, etc).
- *http://169.254.169.254/latest/meta-data/:*
    - This is the root path for metadata (like instance ID, IP, hostname, etc).

📥 **Result:** You'll get a list of metadata categories (e.g., instance-id, ami-id, public-ipv4, etc).



153. **Command:** Run the following command to retrieve the public-hostname (one of the top-level metadata items that were obtained in the preceding command):

curl http://169.254.169.254/latest/meta-data/public-hostname -H "X-aws-ec2-metadata-token: $TOKEN"

This command retrieves the **public DNS hostname** of your **EC2 instance**, using **IMDSv2** (Instance Metadata Service v2).

Part-by-Part Breakdown

curl

This is the tool used to make an HTTP request from the command line.

---

 http://169.254.169.254/latest/meta-data/public-hostname

This is the **metadata URL** for the instance.

- *169.254.169.254* is a **special internal IP** used by AWS for instance metadata access.
- */latest/meta-data/* is the base path for metadata information.
- public-hostname is the specific metadata item you're requesting — it gives you the **public DNS name** of your EC2 instance (e.g., ec2-3-91-25-123.compute-1.amazonaws.com).

📦 **So you're asking for the instance's public hostname**.

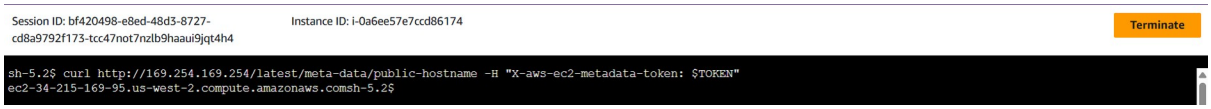
-H "X-aws-ec2-metadata-token: $TOKEN"

- This adds an HTTP header that passes in your **IMDSv2 session token**, which you've already stored in the $TOKEN variable from a previous command like:

*TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" *

-H "X-aws-ec2-metadata-token-ttl-seconds: 21600")

This token is **required** when IMDSv2 is enabled (or enforced) on your EC2 instance.



Note: The IP address 169.254.169.254 is a link-local address and is valid only from the instance.

Here's a list of **commonly used metadata fields** and the exact paths you can use with curl to fetch them via IMDSv2.

Assuming you already have the token stored in $TOKEN, like this:

*TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" *

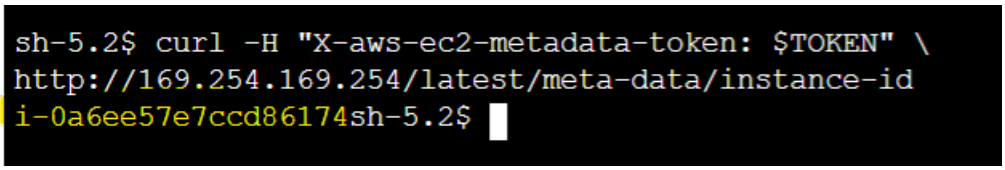
-H "X-aws-ec2-metadata-token-ttl-seconds: 21600")

 Curl commands for popular metadata fields

 Instance ID

*curl -H "X-aws-ec2-metadata-token: $TOKEN" *

http://169.254.169.254/latest/meta-data/instance-id



 AMI ID

*curl -H "X-aws-ec2-metadata-token: $TOKEN" *

http://169.254.169.254/latest/meta-data/ami-id

```
sh-5.2$ curl -H "X-aws-ec2-metadata-token: $TOKEN" \
http://169.254.169.254/latest/meta-data/ami-id \
ami-087f352c165340ea1sh-5.2$
```

Availability Zone

*curl -H "X-aws-ec2-metadata-token: $TOKEN" *

http://169.254.169.254/latest/meta-data/placement/availability-zone

```
sh-5.2$ curl -H "X-aws-ec2-metadata-token: $TOKEN" \
http://169.254.169.254/latest/meta-data/placement/availability-zone
us-west-2ash-5.2$
```

Public IPv4 Address

*curl -H "X-aws-ec2-metadata-token: $TOKEN" *

http://169.254.169.254/latest/meta-data/public-ipv4

```
sh-5.2$ curl -H "X-aws-ec2-metadata-token: $TOKEN" \
http://169.254.169.254/latest/meta-data/public-ipv4
34.215.169.95sh-5.2$
```

Local IPv4 Address (Private IP)

*curl -H "X-aws-ec2-metadata-token: $TOKEN" *

http://169.254.169.254/latest/meta-data/local-ipv4

```
sh-5.2$ curl -H "X-aws-ec2-metadata-token: $TOKEN" \
http://169.254.169.254/latest/meta-data/local-ipv4
10.0.0.80sh-5.2$
```

Hostname (Private DNS name)

*curl -H "X-aws-ec2-metadata-token: $TOKEN" *

http://169.254.169.254/latest/meta-data/hostname

```
sh-5.2$ curl -H "X-aws-ec2-metadata-token: $TOKEN" \
http://169.254.169.254/latest/meta-data/hostname
ip-10-0-0-80.us-west-2.compute.internalsh-5.2$
```

Public Hostname (Public DNS)

*curl -H "X-aws-ec2-metadata-token: $TOKEN" *

http://169.254.169.254/latest/meta-data/public-hostname

```
sh-5.2$ curl -H "X-aws-ec2-metadata-token: $TOKEN" \
http://169.254.169.254/latest/meta-data/public-hostname
ec2-34-215-169-95.us-west-2.compute.amazonaws.comsh-5.2$
```

🔹 IAM Role Name (if instance has one)

*curl -H "X-aws-ec2-metadata-token: $TOKEN" *

http://169.254.169.254/latest/meta-data/iam/security-credentials/

```
sh-5.2$ curl -H "X-aws-ec2-metadata-token: $TOKEN" \
http://169.254.169.254/latest/meta-data/iam/security-credentials/
EC2SSMRolesh-5.2$
```

This will return the IAM role name. Then you can query its credentials like this:

*curl -H "X-aws-ec2-metadata-token: $TOKEN" *

http://169.254.169.254/latest/meta-data/iam/security-credentials/<role-name>

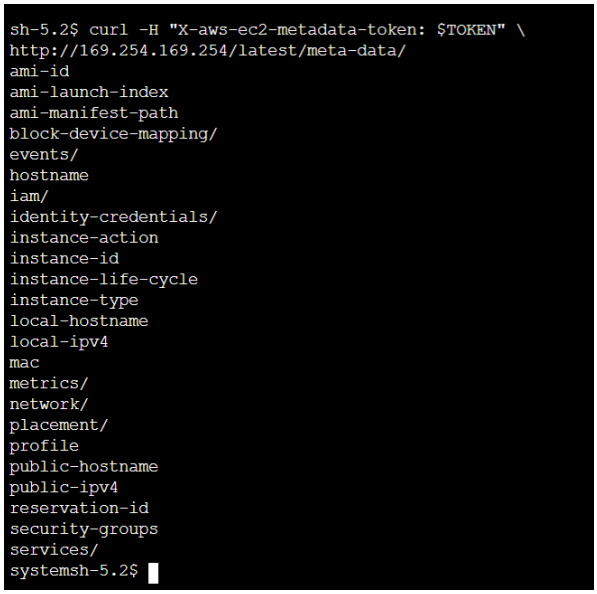

```
sh-5.2$ curl -H "X-aws-ec2-metadata-token: $TOKEN" \
http://169.254.169.254/latest/meta-data/iam/security-credentials/EC2SSMRole
{
  "Code" : "Success",
  "LastUpdated" : "2025-04-16T12:59:50Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASIARBB3SZG4GFUPGJJR",
  "SecretAccessKey" : "gn8garzqM030e57KltqMPffohKBhEzZT/JJrtdWc",
  "Token" : "IQoJb3JpZ2luX2VjEL3//////////wEaCXVzLXdlc3QtMiJHMEUCIQDzykiDdPF28eSHJWOGV8dWIc9eiU6oX7kBjfo4umy0dwIgOzgXzArme9BZ9M6pyluUEmZX8vgoxHqqZWyJxc5B8C0quAUIRhAEGgwwNzA5OTE5MjM2NDAiDDVUKuoVKlUhUk+b1iqVBTYjdXXeurMjol5GmW6ZOhx1IjR8McPtDl3BFeJt4cgkwGExTdo+432cW+QGzHmxqXNtpRlVYnSdkzk1PF1pZhFmi1yUgsu0bU0IWe8EkxNuCYWJQ1ja5c0hZGE1BUEddx2Fymggg W6I4oW4Z90Qgk1R96A3ISKunVxaifPNLGfy+UGlNsQzaLoOgZ++PfzVYUQv82FUaCfDlhjn23KHHNVG/bok2CAz6Um2PYvadktJf8VB3r0NePsexIYC+UQPTYqfzxpbEGeslqsNvzcklqtLz3iZEgxkoh+ZO55dPozy3I688jGn LzZWYqKOh2JW9NNXUMm7/q+v1ZobAoLYuj7+fI4HBMRJCO9z+3BwSP/7cOB7oA2q/ARrHeOJudDz4gYwf37GVWz1fmbpdE/lGrzVeMXhnOFN+6Iq7QteNHOSldHAzS8sBWQsG5Y8/hl5UqrKdfj9J4x5vWgrwZfEiYd2oI0wB2I zc3+ZulU2JhniaRxKuRu+cnIHsIepjTjoiqdNjvskScuku3x8tsrOu7w3ec+A4a+IpU+VG7FlePzhvUWOw/ZM1xz9wRd2cGtfznOTMmqu4lS5FO26DnZNC5vMmxzPFlPI8IdpJElC3DMFmn/eOKnaRcebmpaDSN33Dxa+hmbUj4 H9NUG75rm5Q84hy2UmxqCLQ3tGFavBZxWvvo4PC+Tm6NlbEb6Ie+5Ay1B5n/o7znm9MWkbM9QG5BTxTENmgH6/OAAvjBLFuVc83+O69neh8HvyPCMSW2qKjV4Ymx1h/Vn5cWUbdCDlfFRGXmJrQ9HsIBDl3WIYzQROoT14cXclL WmDlRrO7yysK1wfpuTK93YnR3jZcdiluvVczzdodo8IeRggP2q59ZNadjIk/tkw29P+vwY6sQFUg1rI8jtHxJMEy628+kMnnmHwBzbSvvV3exjtKJ8i2w069YtpmKyMOp3heZKuGi9QAapGqzm9Po/Oc/p+W0DBwlAm7mpcHlBw HFQcm6wN07KqhvOIjqvxluPQ8L0+YsmpyImjbUH46R0I9dmIa0r+SOtAVmWKmwiSb8CEA+JnKVBmkpAZ4o7lZdwFznMpvddFUeSbaxxE/3trkWaEvUYoyhonglI8fPBKCR7CqHTFG2A=",
  "Expiration" : "2025-04-16T19:02:22Z"
}sh-5.2$
```

🔹 Full metadata tree (to explore)

*curl -H "X-aws-ec2-metadata-token: $TOKEN" *

http://169.254.169.254/latest/meta-data/

This will list all the available metadata categories so you can explore more.



You have successfully learned how to retrieve instance metadata from your running Amazon EC2 instance.

Conclusion

Creating a VPC with both public and private subnets provides you the flexibility to launch tasks and services in either a public or private subnet. Tasks and services in the private subnets can access the internet through a NAT gateway.

Congratulations! You now have successfully:

- Created an Amazon VPC.
- Created public and private subnets.
- Created an internet gateway.
- Configured a route table and associated it to a subnet.
- Created an Amazon EC2 instance and made the instance publicly accessible.
- Isolated an Amazon EC2 instance in a private subnet.
- Created and assigned security groups to Amazon EC2 instances.
- Connected to Amazon EC2 instances using Session Manager.

End lab

Follow these steps to close the console and end your lab.

154. Return to the **AWS Management Console**.

155. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.

156. Choose **End Lab** and then confirm that you want to end your lab.

Additional resources

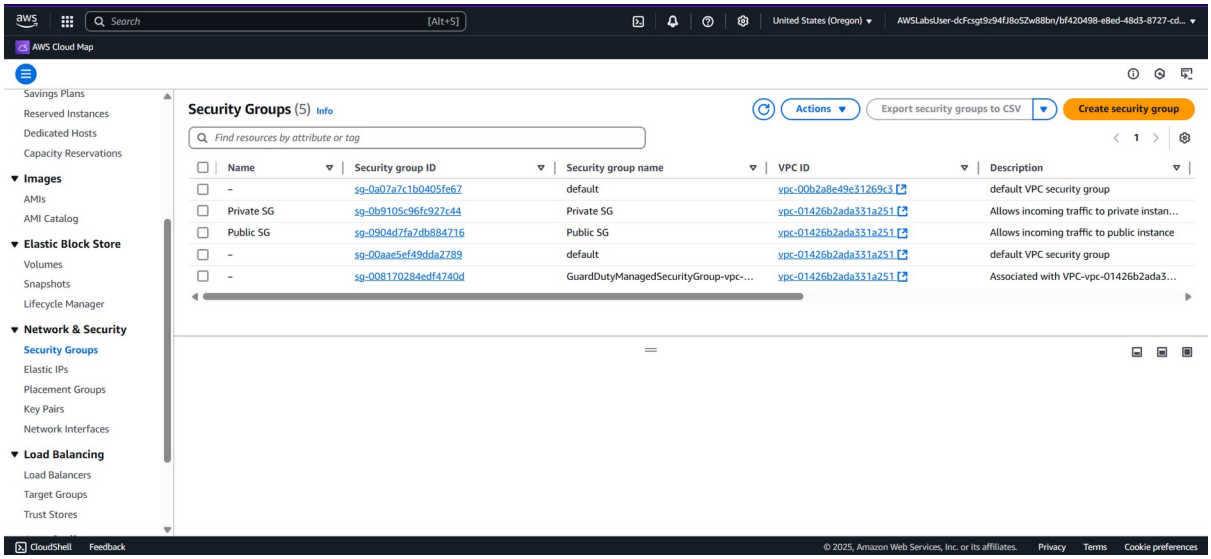
- What is Amazon VPC?
- Subnets for Your VPC
- Connect to the internet using an internet gateway
- Configure route tables

- Control traffic to resources using security groups
- NAT gateways
- Public IPv4 addresses
- Understanding the basics of IPv6 networking on AWS

Optional task solution

157. Return to the AWS Management Console browser tab.

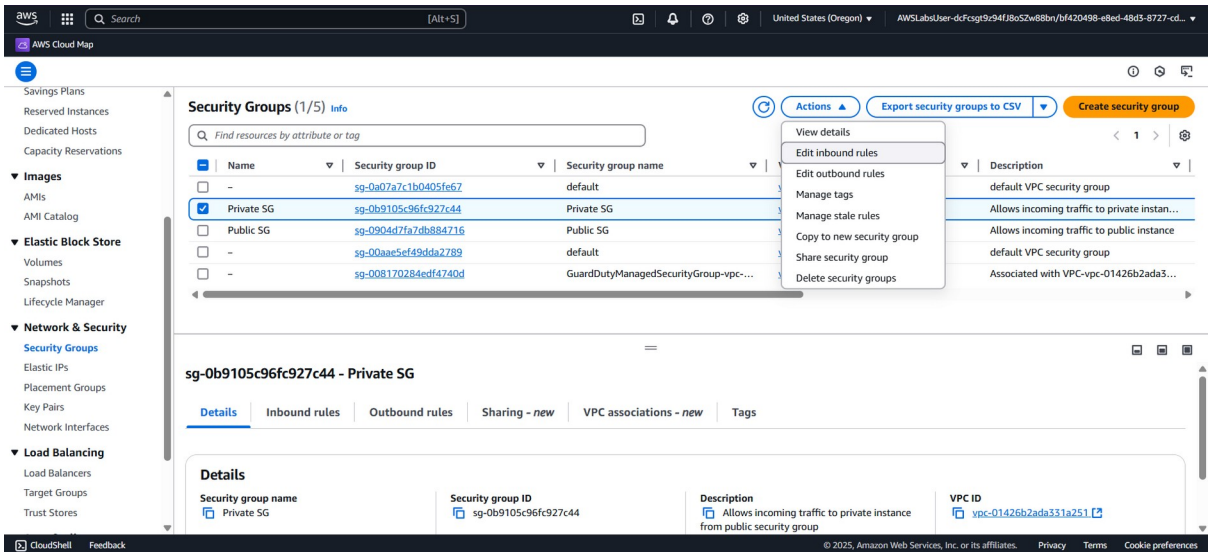
158. At the top of the AWS Management Console, in the search box, search for and choose EC2.



159. In the left navigation pane, choose **Security Groups**.

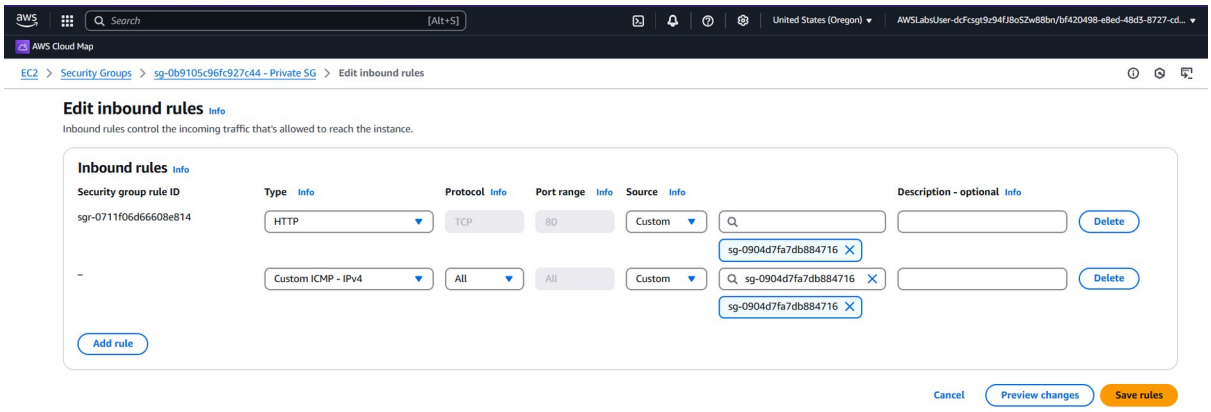
160. Select **Private SG**.

161. Choose **Actions** and then choose **Edit inbound rules**.

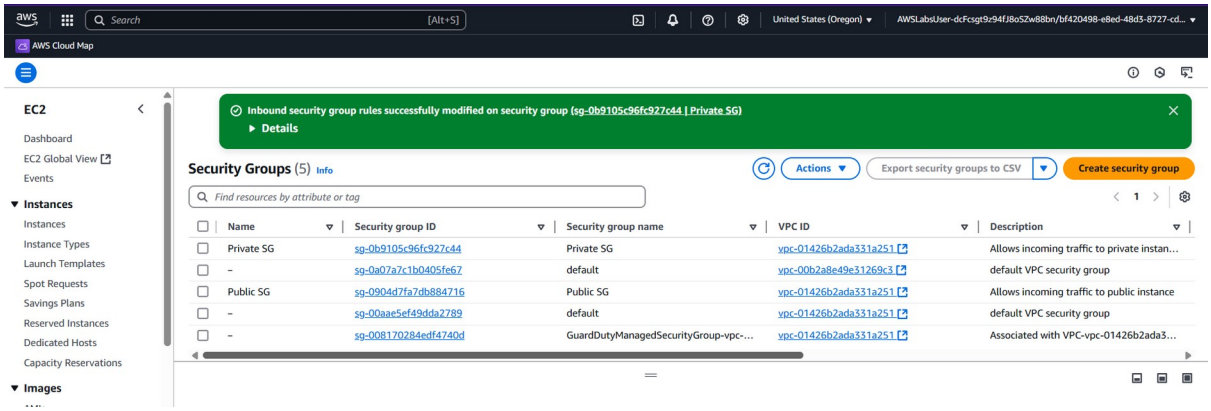


162. On the **Edit inbound rules** page, in the **Inbound rules**, choose **Add rule** and configure the following:

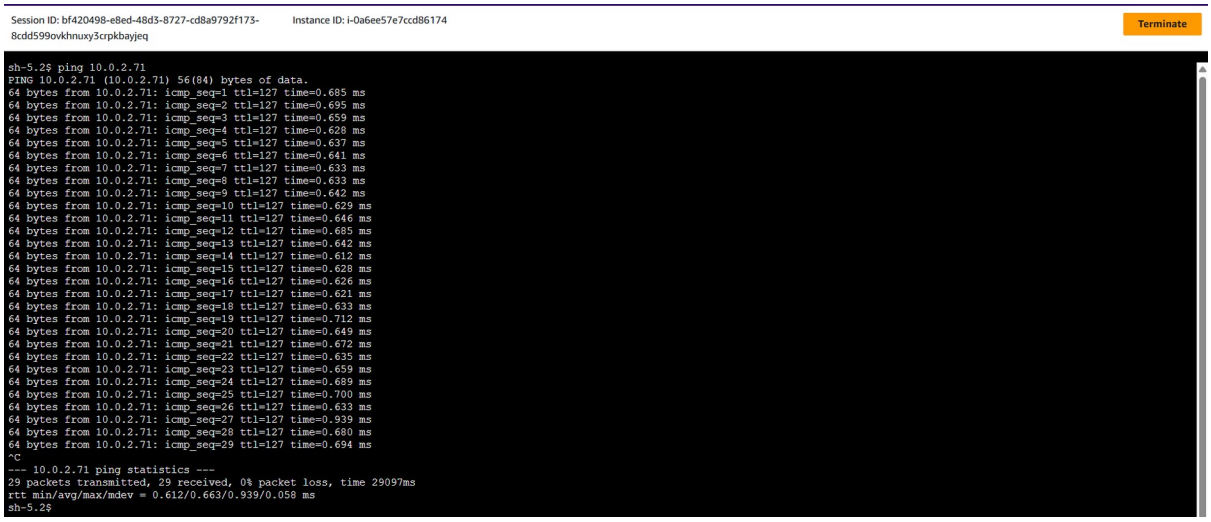
- **Type:** Select *Custom ICMP - IPV4*.
- **Source:** Select *Custom*.
    - In the box to the right of **Custom**, type sg.
    - Choose **Public SG** from the list.



163. Choose **Save rules**.



164. Select the Optional Task link to go to the **Optional Task** and re-run the steps. The *Public Instance* should now be able to successfully ping *Private Instance*.



For more information about AWS Training and Certification, see *https://aws.amazon.com/training/*.

Your feedback is welcome and appreciated.
If you would like to share any feedback, suggestions, or corrections, please provide the details in our *AWS Training and Certification Contact Form*.