

# Lab 2: Building your Amazon VPC Infrastructure

© 2025 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

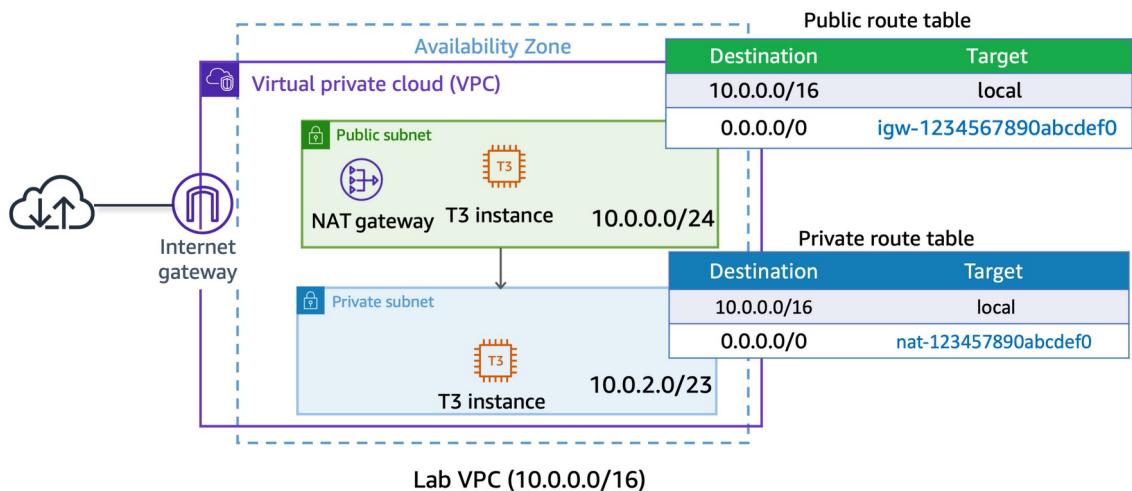
Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

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

## Amazon Web Services Sign In

You must first log out before logging into a different AWS account.

To logout, [click here](#)

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

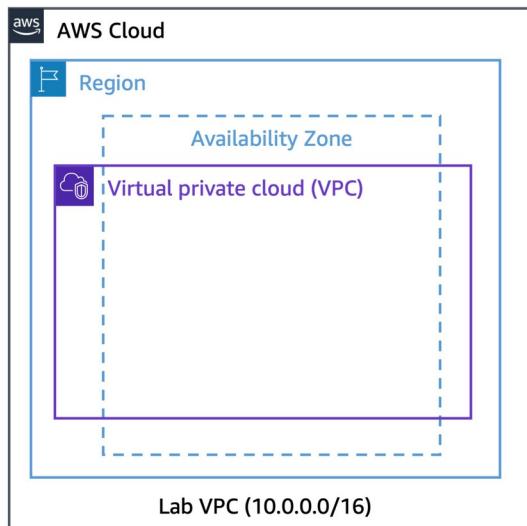
AWS services not used in this lab are deactivated in the lab environment. In addition, the capabilities of the services used in this lab are limited to only what the lab requires. Expect errors when accessing other services or performing actions beyond those provided in this lab guide.

---

### 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**.

The screenshot shows the AWS Cloud Map search results for 'VPC'. The search bar at the top has 'Q VPC'. The left sidebar has a 'Services' section with links like Features, Resources (New), Documentation, Knowledge articles, Marketplace, Blog posts, Events, and Tutorials. The main content area shows three services: 'VPC' (Isolated Cloud Resources), 'AWS Firewall Manager' (Central management of firewall rules), and 'Detective' (Investigate and Analyze potential security issues). Below these are sections for 'Features' (Dashboard, Route 53 VPCs, VPC Reachability Analyzer) and 'Were these results helpful?' (Yes/No buttons). To the right is a 'Create application' interface with a 'Find applications' search bar and a 'Create application' button. The bottom of the screen shows the AWS navigation bar with Welcome to AWS, AWS Health, Cost and usage, and other links.

**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 screenshot shows the AWS VPC dashboard. The left sidebar includes 'Virtual private cloud' (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers) and 'Security' (AWS Cloud Map). The main content area features a 'Create VPC' and 'Launch EC2 Instances' button. It displays 'Resources by Region' with counts for VPCs (1), Subnets (4), Route Tables (1), Internet Gateways (0), NAT Gateways (0), VPC Peering Connections (0), Network ACLs (1), Security Groups (1), and Customer Gateways (0). A 'Service Health' section shows 'View complete service health details'. On the right, there are 'Settings' (Block Public Access, Zones, Console Experiments) and 'Additional Information' (VPC Documentation, All VPC Resources, Forums, Report an Issue).

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**

## Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

**VPC settings**

Resources to create Info  
Create only the VPC resource or the VPC and other networking resources.

VPC only  VPC and more

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

**IPv4 CIDR block** Info  
 IPv4 CIDR manual input  IPAM-allocated IPv4 CIDR block

**IPv4 CIDR**  
10.0.0.0/16  
CIDR block size must be between /16 and /28.

**IPv6 CIDR block** Info  
 No IPv6 CIDR block  IPAM-allocated IPv6 CIDR block  Amazon-provided IPv6 CIDR block  IPv6 CIDR owned by me

**Tenancy** Info  
Default

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

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="Lab VPC"/> <span>X</span> <span>Remove tag</span>

Add tag  
You can add 49 more tags

Cancel Preview code Create VPC

## 6. Choose **Create VPC**.

The screenshot shows the AWS VPC Details page for a newly created VPC named "vpc-01426b2ada331a251 / Lab VPC". A green banner at the top indicates success: "You successfully created vpc-01426b2ada331a251 / Lab VPC". The main section displays VPC details:

Details	Info
VPC ID	vpc-01426b2ada331a251
DNS resolution	Enabled
Main network ACL	acl-0ba9945c6df30112
IPv6 CIDR (Network border group)	-
State	Available
Tenancy	default
Default VPC	No
Block Public Access	Off
DHCP option set	dopt-0ad92920ab3010af5
IPV4 CIDR	10.0.0.0/16
Network Address Usage metrics	Disabled
Route 53 Resolver DNS Firewall rule groups	-
DNS hostnames	Disabled
Main route table	rtb-066c62d8c79a456d
IPv6 pool	-
Owner ID	070991923640

Below the details, there's a "Resource map" section with four cards: "VPC Show details", "Subnets (0)", "Route tables (1)", and "Network connections (0)".

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**.

You successfully created vpc-01426b2ada331a251 / Lab VPC

**vpc-01426b2ada331a251 / Lab VPC**

**Details** **Info**

VPC ID vpc-01426b2ada331a251	State <span style="color: green;">Available</span>	Block Public Access <input type="radio"/> Off	DNS hostnames Disabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-0ad92920ab3010af5	Main route table rtb-066c62df8c79a456d
Main network ACL acl-0b9a943cd6fd30112	Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool —
IPv6 CIDR (Network border group) —	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups —	Owner ID 070991923640

**Resource map** **Info**

- VPC** Show details Your AWS virtual network
- Subnets (0)** Subnets within this VPC
- Route tables (1)** Route network traffic to resources
- Network connections (0)** Connections to other networks

**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**.

You successfully created vpc-01426b2ada331a251 / Lab VPC

**vpc-01426b2ada331a251 / Lab VPC**

**Details** **Info**

VPC ID vpc-01426b2ada331a251	State <span style="color: green;">Available</span>	Block Public Access <input type="radio"/> Off	DNS hostnames Disabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-0ad92920ab3010af5	Main route table rtb-066c62df8c79a456d
Main network ACL acl-0b9a943cd6fd30112	Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool —
IPv6 CIDR (Network border group) —	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups —	Owner ID 070991923640

**Resource map** **Info**

- VPC** Show details Your AWS virtual network
- Subnets (0)** Subnets within this VPC
- Route tables (1)** Route network traffic to resources
- Network connections (0)** Connections to other networks

**Actions**

- Create flow log
- Edit VPC settings**
- Edit CIDRs
- Manage middlebox routes
- Manage tags
- Delete VPC

The screenshot shows the 'Edit VPC settings' page for a VPC named 'Lab VPC'. In the 'DNS settings' section, the 'Enable DNS hostnames' checkbox is selected. The 'Save' button is highlighted in orange at the bottom right.

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*

The screenshot shows the 'Edit VPC settings' page for a VPC named 'Lab VPC'. In the 'DNS settings' section, both 'Enable DNS resolution' and 'Enable DNS hostnames' checkboxes are selected. The 'Save' button is highlighted in orange at the bottom right.

10. Choose **Save**.

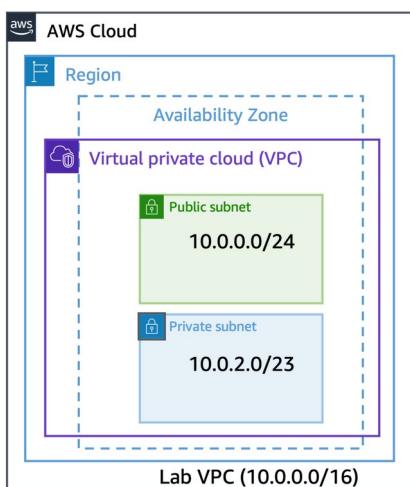
A (You have successfully modified the settings for vpc-xxxxxxxxxx / Lab VPC.) message is displayed on top of the screen.

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.

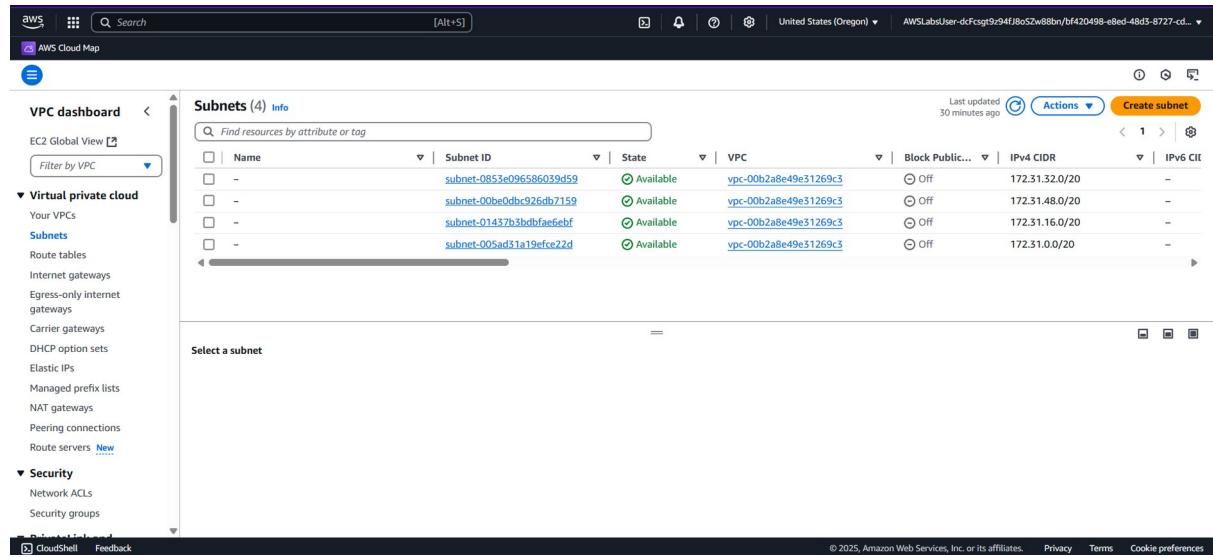


**Note:** A *subnet* is a sub-range of IP addresses within a network. You can launch AWS resources into a specified subnet. Use a *public subnet* for resources that must be connected to the internet, and use a *private subnet* for resources that are to remain isolated from the internet.

### Task 2.1: Create your public subnet

The public subnet is for internet-facing resources.

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



The screenshot shows the AWS VPC Subnets page. The left sidebar includes sections for EC2 Global View, Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers), Security (Network ACLs, Security groups), and CloudWatch Metrics (CloudWatch Metrics Home, Metrics Insights). The main content area displays a table titled "Subnets (4) Info" with the following data:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR
-	subnet-0855e096586039d59	Available	vpc-00b2a8e49e31269c3	Off	172.31.32.0/20	-
-	subnet-00be0db926db7159	Available	vpc-00b2a8e49e31269c3	Off	172.31.48.0/20	-
-	subnet-01437b3bdffae6ebf	Available	vpc-00b2a8e49e31269c3	Off	172.31.16.0/20	-
-	subnet-005ad31a19efce22d	Available	vpc-00b2a8e49e31269c3	Off	172.31.0.0/20	-

Below the table, there is a section titled "Select a subnet" with three small icons: a plus sign, a minus sign, and a question mark.

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**.

**Create subnet** Info

**VPC**

**VPC ID**  
Create subnets in this VPC.  
vpc-01426b2ada331a251 (Lab VPC)

**Associated VPC CIDRs**

**IPv4 CIDRs**  
10.0.0.0/16

---

**Subnet settings**  
Specify the CIDR blocks and Availability Zone for the subnet.

**Subnet 1 of 1**

**Subnet name**  
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

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

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

**IPv4 subnet CIDR block**  
 256 IPs

**Tags - optional**

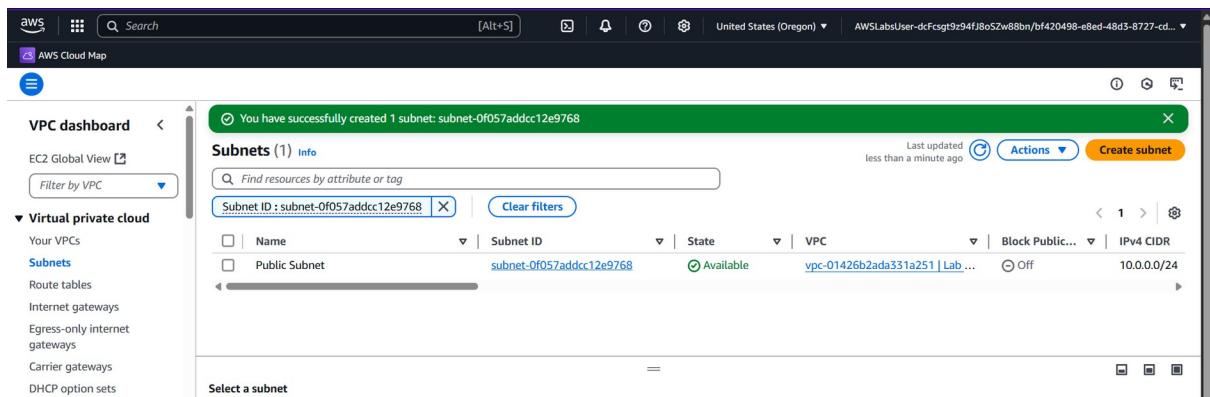
Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="Public Subnet"/>

[Add new tag](#)  
You can add 49 more tags.  
[Remove](#)

[Add new subnet](#)

[Cancel](#) [Create subnet](#)

13. Choose **Create subnet**.



You have successfully created 1 subnet: subnet-0f057addcc12e9768

**Subnets (1) Info**

Last updated less than a minute ago

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
Public Subnet	subnet-0f057addcc12e9768	Available	vpc-01426b2ada331a251   Lab ...	Off	10.0.0.0/24

Select a 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

The screenshot shows the AWS VPC dashboard with the 'Subnets' section selected. A single subnet, 'Public Subnet' (subnet-0f057addcc12e9768), is listed with a CIDR range of 10.0.0.0/24. The 'Actions' dropdown menu is open, showing options such as 'Edit subnet settings', 'Create flow log', and 'Edit IPv6 CIDs'. The subnet details page is also visible, showing various configuration parameters like Subnet ARN, State, Network border group, and Auto-assign IPv6 address.

**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.

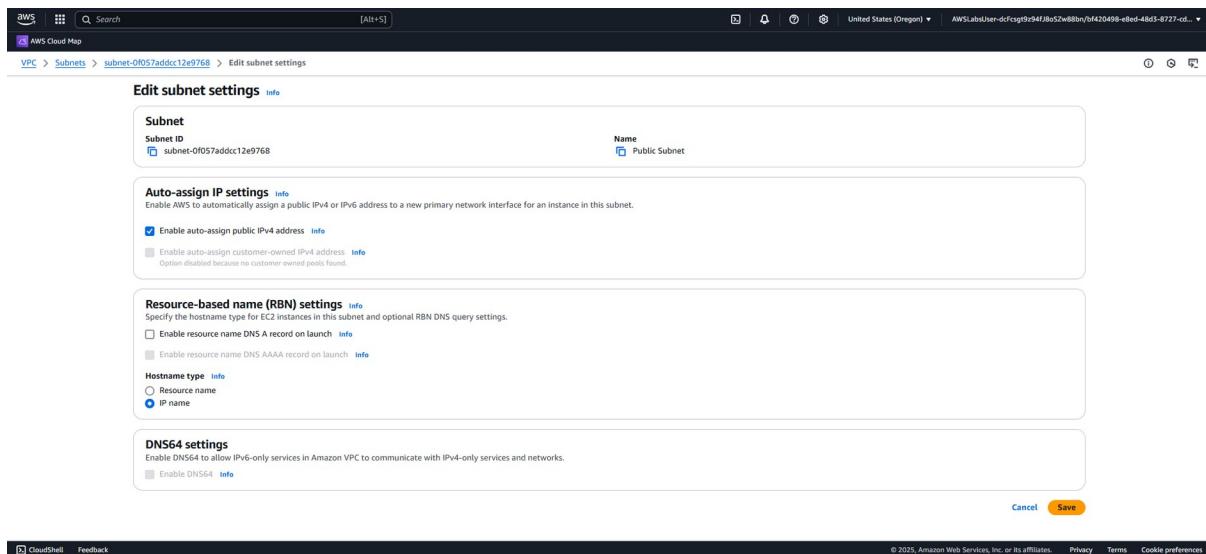
## 15. Select **Public Subnet**.

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

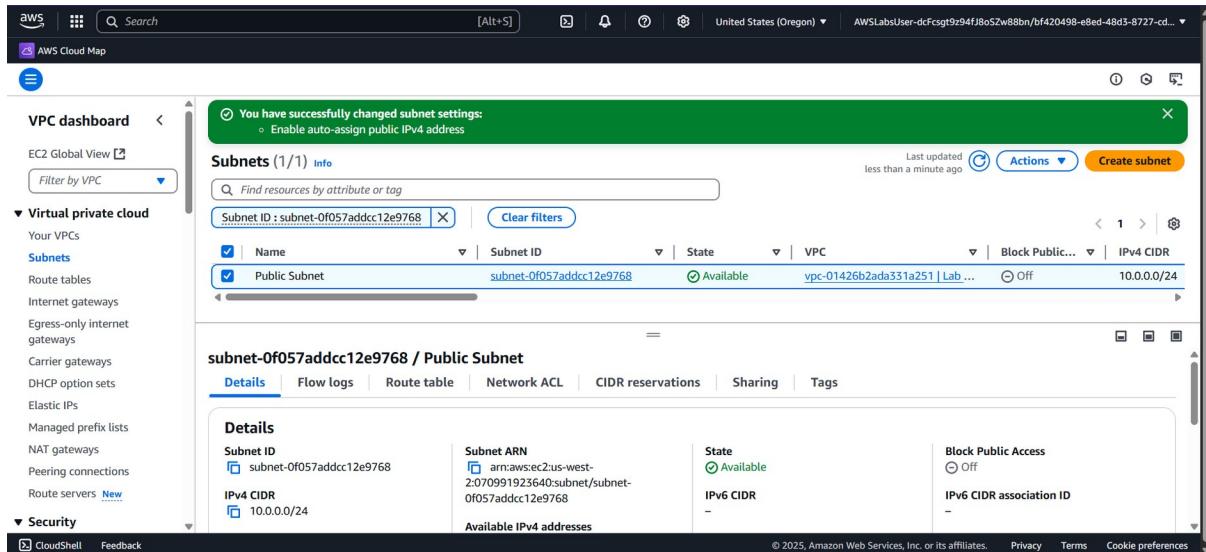
The screenshot shows the AWS VPC dashboard with the 'Subnets' section selected. A single subnet, 'Public Subnet' (subnet-0f057addcc12e9768), is listed with a CIDR range of 10.0.0.0/24. The 'Actions' dropdown menu is open, with 'Edit subnet settings' highlighted. The subnet details page is also visible, showing various configuration parameters like Subnet ARN, State, Network border group, and Auto-assign IPv6 address.

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**.

**Create subnet** Info

**VPC**  
VPC ID  
Create subnets in this VPC.  
▼

Associated VPC CIDRs  
IPv4 CIDRs  
10.0.0.0/16

**Subnet settings**  
Specify the CIDR blocks and Availability Zone for the subnet.

**Subnet 1 of 1**

**Subnet name**  
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

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

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

**IPv4 subnet CIDR block**  
 512 IPs  
◀ ▶ ⌂ ⌃ ⌄ ⌅

**Tags - optional**  
Key  Value - optional     
You can add 49 more tags.

20. Choose **Create subnet**.

**VPC dashboard** < [Alt+S]

**Subnets** (1/1) Info Last updated less than a minute ago

Subnet ID	Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv6 C...
<input type="text" value="subnet-087a8c37fb0d925f2"/>	<input checked="" type="checkbox" value="Private Subnet"/>	<input type="text" value="subnet-087a8c37fb0d925f2"/>	<input checked="" type="checkbox" value="Available"/>	vpc-01426b2ada331a251   Lab ...	<input checked="" type="checkbox" value="Off"/>	10.0.2.0/23	-	-

**Actions**

**Subnet ID:** subnet-087a8c37fb0d925f2  
**IPv4 CIDR:** 10.0.2.0/23  
**Availability Zone:** us-west-2b  
**Route table:** -  
**Auto-assign IPv6 address:** No  
**IPv4 CIDR reservations:** -  
**Resource name DNS A record:** Disabled

**Subnet ARN:** [arn:aws:ec2:us-west-2:070991923640:subnet/subnet-087a8c37fb0d925f2](#)  
**Available IPv4 addresses:** 507  
**Availability Zone ID:** usw2-az2  
**Network ACL:** -  
**Auto-assign customer-owned IPv4 address:** No  
**IPv6 CIDR reservations:** -  
**Resource name DNS AAAA record:** Disabled

**State:** Available  
**IPv6 CIDR:** -  
**Network border group:** us-west-2  
**Default subnet:** No  
**Customer-owned IPv4 pool:** -  
**IPv6-only:** No  
**DNS64:** Disabled

**Block Public Access:**  Off  
**IPv6 CIDR association ID:** -  
**VPC:** vpc-01426b2ada331a251 | Lab VPC  
**Auto-assign public IPv4 address:** No  
**Outpost ID:** -  
**Hostname type:** IP name  
**Owner:** [070991923640](#)

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

The screenshot shows the AWS VPC dashboard with a success message: "You have successfully created 1 subnet: subnet-087a8c37fb0d925f2". The left navigation pane includes sections for EC2 Global View, Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers), Security (Network ACLs, Security groups), PrivateLink and Lattice (Getting started, Updated Endpoints), and CloudShell/Feedback.

**Subnets (1/1) Info**

Name	Subnet ID	State	VPC	Block Public Access	IPv4 CIDR	IPv6 CIDR	IPv6 C
Private Subnet	subnet-087a8c37fb0d925f2	Available	vpc-01426b2ada331a251   Lab ...	Off	10.0.2.0/23	-	-

**Subnet ARN**: arn:aws:ec2:us-west-2:070991923640:subnet/subnet-087a8c37fb0d925f2

**State**: Available

**IPv4 CIDR**: 10.0.2.0/23

**Available IPv4 addresses**: 507

**Availability Zone ID**: usw2-az2

**Network ACL**: -

**Auto-assign IPv6 address**: No

**Auto-assign customer-owned IPv4 address**: No

**IPv6 CIDR reservations**: -

**Resource name DNS A record**: Disabled

**Customer-owned IPv4 pool**: -

**IPv6-only**: No

**DNS64**: Disabled

**Block Public Access**: Off

**IPv6 CIDR association ID**: -

**VPC**: vpc-01426b2ada331a251 | Lab VPC

**Auto-assign public IPv4 address**: No

**Outpost ID**: -

**Hostname type**: IP name

**Owner**: 070991923640

**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.

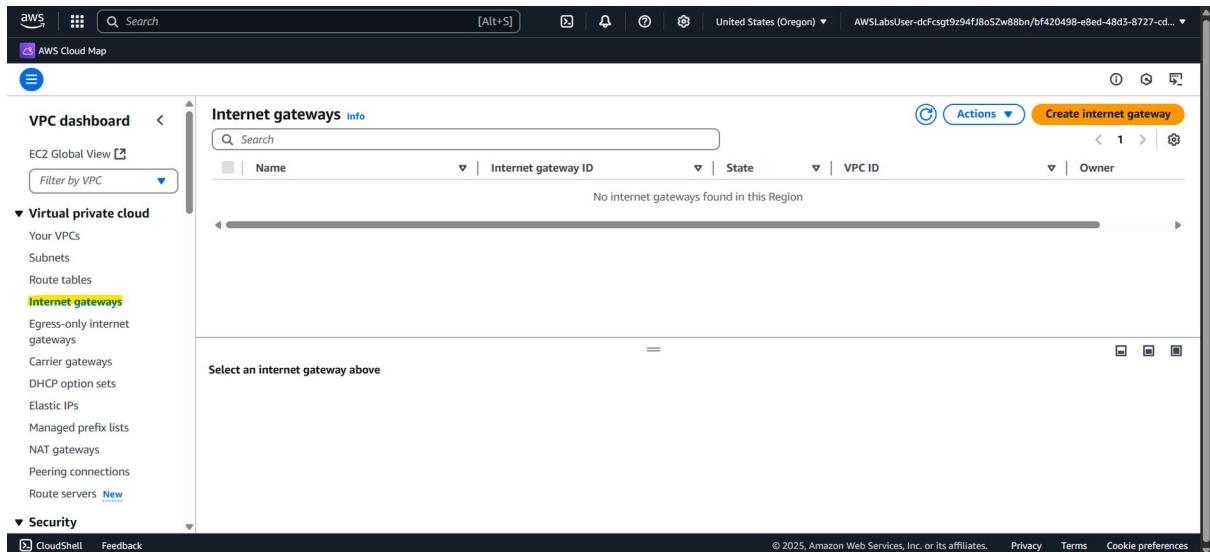
Congratulations! You have successfully created a public subnet and a private subnet in the lab VPC.

### 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.

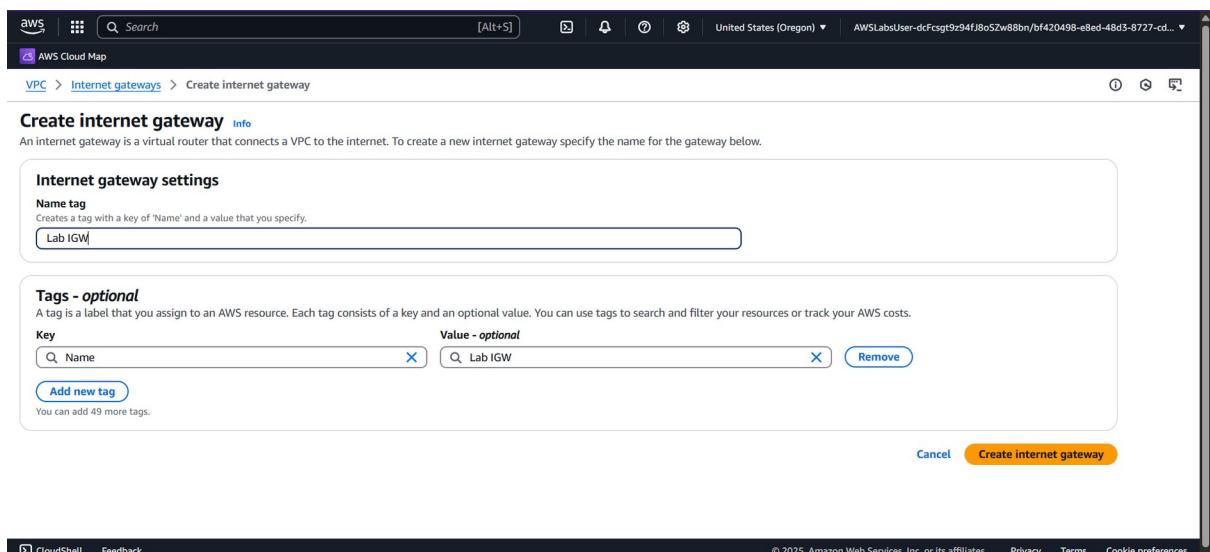
**Learn more:** An internet gateway serves two purposes: To provide a target in your VPC route tables for internet-bound traffic, and to perform network address translation (NAT) for instances that have been assigned public IPv4 addresses.

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**.

The screenshot shows the AWS VPC Internet Gateways page. A green banner at the top states: "The following internet gateway was created: igw-0c14acbf649254b86 - Lab IGW. You can now attach to a VPC to enable the VPC to communicate with the internet." Below the banner, the internet gateway is listed with the ID "igw-0c14acbf649254b86". The "State" is "Detached". The "VPC ID" and "Owner" information are shown as "-". The "Actions" dropdown menu includes "Attach to a VPC", "Actions", "Manage tags", and "Delete". The left sidebar shows the "Virtual private cloud" section with "Internet gateways" selected.

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**.

The screenshot shows the same AWS VPC Internet Gateways page as before, but the "Actions" dropdown menu is open, revealing options: "Attach to VPC", "Detach from VPC", "Manage tags", and "Delete". The "Attach to VPC" option is highlighted.

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

The screenshot shows the "Attach to VPC" confirmation dialog. It displays the message: "Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below." A dropdown menu titled "Available VPCs" lists "vpc-01426b2ada331a251". The "AWS Command Line Interface command" field contains the command "aws ec2 attach-internet-gateway --internet-gateway-id igw-0c14acbf649254b86 --vpc-id vpc-01426b2ada331a251". The "Attach internet gateway" button is highlighted in orange.

27. Choose **Attach internet gateway**.

The screenshot shows the AWS VPC Internet Gateways page. At the top, a green success message reads "Internet gateway igw-0c14acbf649254b86 successfully attached to vpc-01426b2ada331a251". Below this, the title "igw-0c14acbf649254b86 / Lab IGW" is displayed. The "Details" section shows the Internet gateway ID (igw-0c14acbf649254b86), State (Attached), VPC ID (vpc-01426b2ada331a251 | Lab VPC), and Owner (070991923640). The "Tags" section lists a single tag: Name (Lab IGW). The left sidebar shows the navigation path: VPC > Internet gateways > igw-0c14acbf649254b86.

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 screenshot shows the AWS VPC Internet Gateways page. A green success message at the top reads "Internet gateway igw-0c14acbf649254b86 successfully attached to vpc-01426b2ada331a251". The title "igw-0c14acbf649254b86 / Lab IGW" is displayed. The "Details" section shows the Internet gateway ID (igw-0c14acbf649254b86), State (Attached), VPC ID (vpc-01426b2ada331a251 | Lab VPC), and Owner (070991923640). The "Tags" section lists a single tag: Name (Lab IGW). The left sidebar shows the navigation path: VPC > Internet gateways > igw-0c14acbf649254b86.

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.

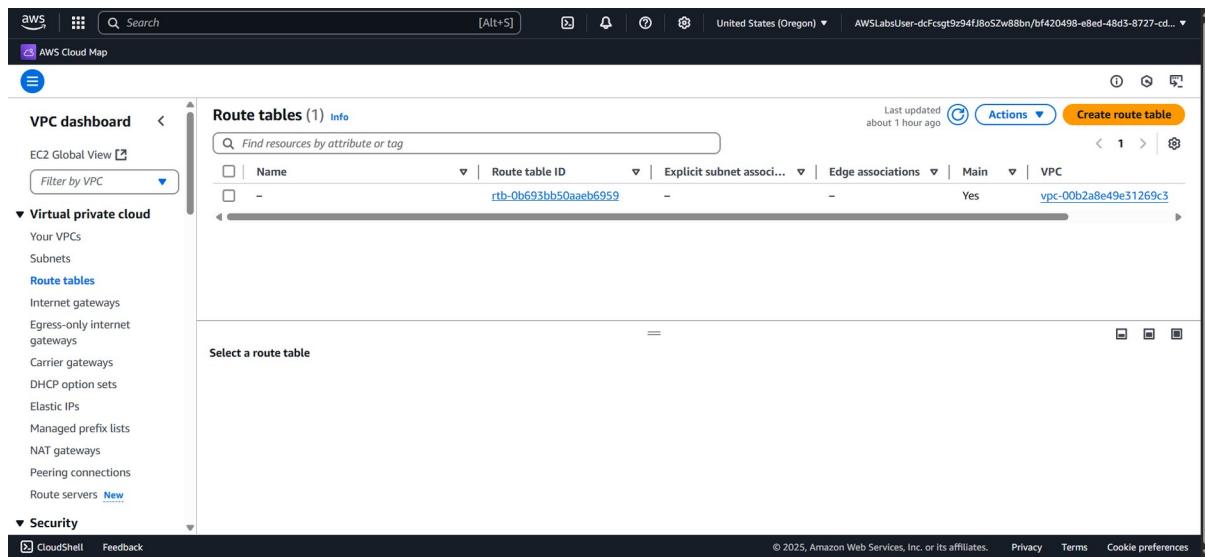
---

#### 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**.

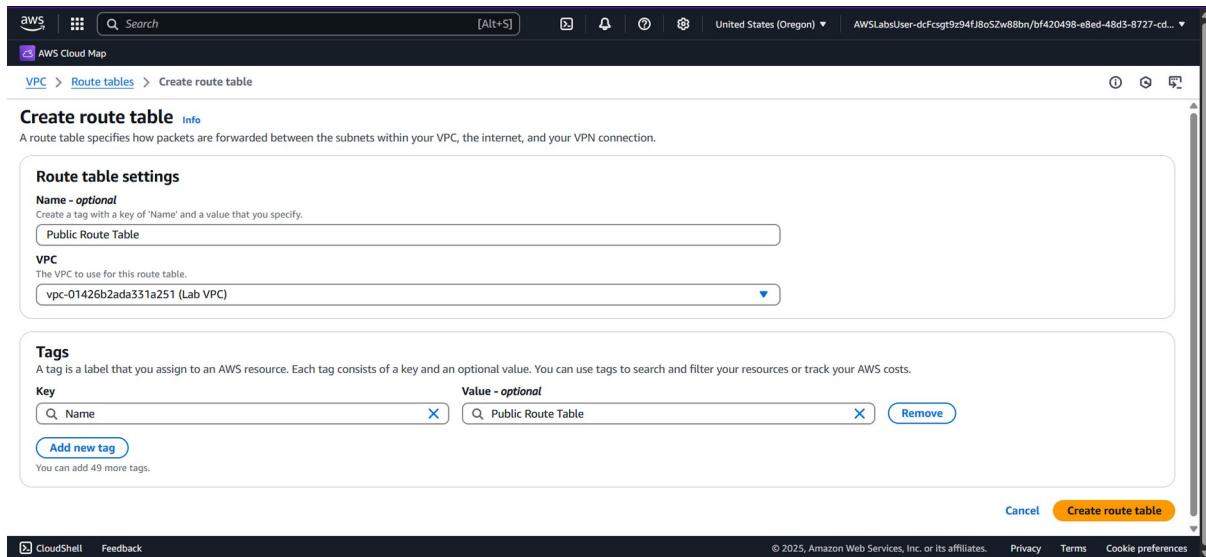


The screenshot shows the AWS Cloud Map interface with the 'Route tables' section selected. The left sidebar includes options like 'VPC dashboard', 'EC2 Global View', 'Virtual private cloud' (with 'Your VPCs', 'Subnets', and 'Route tables' listed), 'Internet gateways', 'Egress-only Internet gateways', 'Carrier gateways', 'DHCP option sets', 'Elastic IPs', 'Managed prefix lists', 'NAT gateways', 'Peering connections', 'Route servers', and 'Security'. The main area displays a table titled 'Route tables (1) Info' with one entry: 'rtb-0b693bb50aaeb6959'. The table columns include 'Name' (with a dropdown menu), 'Route table ID' (containing the value 'rtb-0b693bb50aaeb6959'), 'Explicit subnet associ...', 'Edge associations', 'Main', and 'VPC' (containing 'vpc-00b2a8e49e31269c3'). Below the table is a section titled 'Select a route table' with three icons: a magnifying glass, a plus sign, and a minus sign.

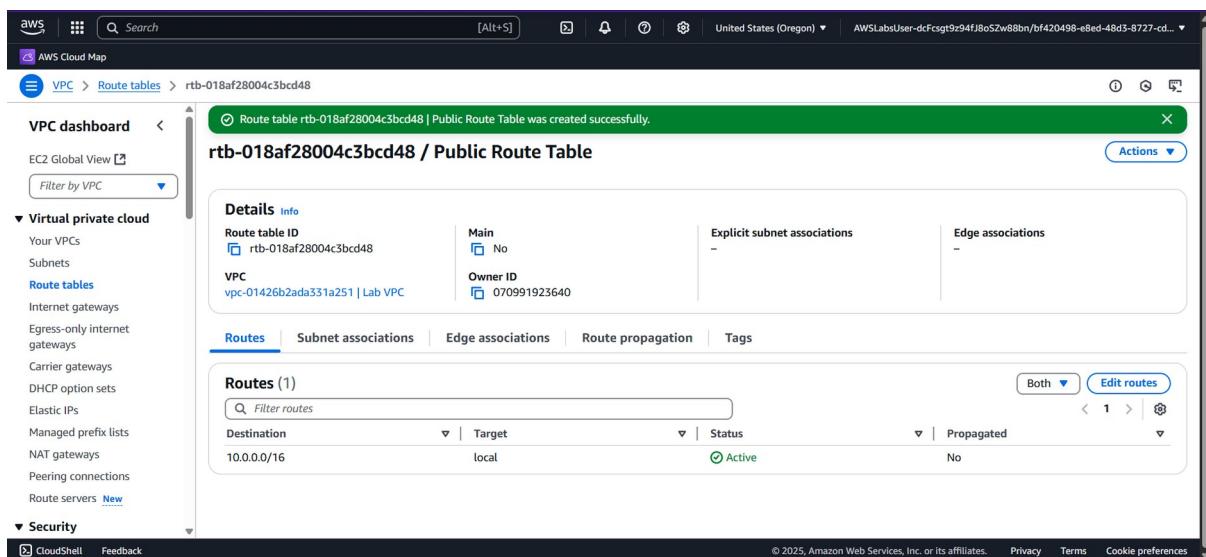
There is currently one default route table associated with the VPC, **Lab VPC**. This routes traffic locally. You now create an additional route table to route public traffic to your internet gateway.

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-xxxxxx | 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.

aws | Search [Alt+S] | United States (Oregon) | AWSLabUser-dcfsgt9z94fJ8oSzw8bn/bf420498-e8ed-48d3-8727-cd... | AWS Cloud Map

VPC > Route tables > rtb-018af28004c3bcd48 > Edit routes

**Edit routes**

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

Add route Cancel Preview Save changes

aws | Search [Alt+S] | United States (Oregon) | AWSLabUser-dcfsgt9z94fJ8oSzw8bn/bf420498-e8ed-48d3-8727-cd... | AWS Cloud Map

VPC > Route tables > rtb-018af28004c3bcd48 > Edit routes

**Edit routes**

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway igw-0c14acbf649254b86	-	No

Add route Remove Cancel Preview Save changes

35. Choose **Save changes**.

aws | Search [Alt+S] | United States (Oregon) | AWSLabUser-dcfsgt9z94fJ8oSzw8bn/bf420498-e8ed-48d3-8727-cd... | AWS Cloud Map

VPC dashboard < VPC Global View Filter by VPC

rtb-018af28004c3bcd48 / Public Route Table

**Details** Info

Route table ID rtb-018af28004c3bcd48	Main No	Explicit subnet associations -	Edge associations -
VPC vpc-01426b2ada331a251   Lab VPC	Owner ID 070991923640		

Routes Subnet associations Edge associations Route propagation Tags

**Routes (2)**

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0c14acbf649254b86	Active	No
10.0.0.0/16	local	Active	No

Both Edit routes < 1 > Actions

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

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

VPC dashboard < EC2 Global View Filter by VPC

Virtual private cloud Your VPCs Subnets Route tables Internet gateways Egress-only internet gateways Carrier gateways DHCP option sets Elastic IPs Managed prefix lists NAT gateways Peering connections Route servers New Security CloudShell Feedback

Owner ID: 070991923640

Routes Subnet associations Edge associations Route propagation Tags

Explicit subnet associations (0)

No subnet associations

Subnets without explicit associations (2)

Find subnet association

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Public Subnet	subnet-0f057addcc12e9768	10.0.0.0/24	-
Private Subnet	subnet-087a8c37fb0d925f2	10.0.2.0/23	-

Find subnet association

Cloud Shell Privacy Terms Cookie preferences

37. Choose **Edit subnet associations**.

Route tables > rtb-018af28004c3bcd48 > Edit subnet associations

Edit subnet associations

Available subnets (2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
Public Subnet	subnet-0f057addcc12e9768	10.0.0.0/24	-	Main (rtb-066c62df8c79a456d)
Private Subnet	subnet-087a8c37fb0d925f2	10.0.2.0/23	-	Main (rtb-066c62df8c79a456d)

Cancel Save associations

38. Select **Public Subnet**

Route tables > rtb-018af28004c3bcd48 > Edit subnet associations

Edit subnet associations

Available subnets (1/2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
Public Subnet	subnet-0f057addcc12e9768	10.0.0.0/24	-	Main (rtb-066c62df8c79a456d)
Private Subnet	subnet-087a8c37fb0d925f2	10.0.2.0/23	-	Main (rtb-066c62df8c79a456d)

Selected subnets

subnet-0f057addcc12e9768 / Public Subnet X

Cancel Save associations

39. Choose **Save associations**.

The screenshot shows the AWS VPC Route Tables page. In the top right corner, there is a green success message: "You have successfully updated subnet associations for rtb-018af28004c3bcd48 / Public Route Table." Below this, the title "rtb-018af28004c3bcd48 / Public Route Table" is displayed. On the left, a navigation pane lists "Virtual private cloud" and "Security". The main content area shows "Details" for the route table, including its ID (rtb-018af28004c3bcd48), VPC (vpc-01426b2ada331a251 | Lab VPC), and owner (070991923640). It also shows "Explicit subnet associations" (1) for a public subnet with CIDR 10.0.0.0/24 and "Subnets without explicit associations" (1) listed below.

A [You have successfully updated subnet associations for rtb-xxxxxx / 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**.

AWS Cloud Map

VPC > Route tables > rtb-018af28004c3bcd48

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

Security  
Network ACLs  
**Security groups**

PrivateLink and Lattice  
Getting started Updated  
Endpoints Updated  
Endpoint services  
Service networks Updated

CloudShell Feedback

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

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.

VPC > Security Groups > Create security group

**Create security group** Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

**Basic details**

Security group name Info  
Public SG

Name cannot be edited after creation.

Description Info  
Allows incoming traffic to public instance

VPC Info  
vpc-01426b2ada331a251 (Lab VPC)

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.

Inbound rules Info

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	Anywhere... 0.0.0.0/0	

Add rule

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

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**.

Details
Security group name: Public SG Security group ID: sg-0904d7fa7db884716 Owner: 070991923640 Inbound rules count: 1 Permission entry Description: Allows incoming traffic to public instance Outbound rules count: 1 Permission entry VPC ID: vpc-01426b2ada331a251

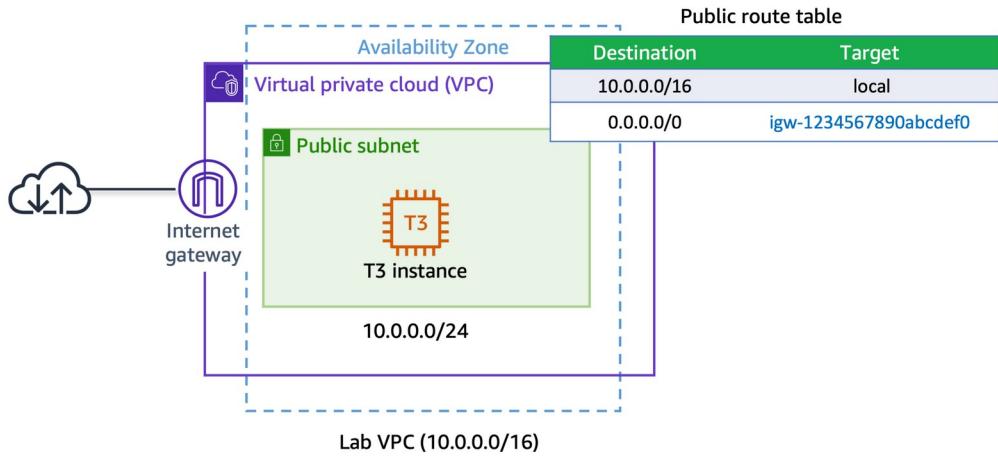
Inbound rules (1)
Name: sgr-0be6cc0b5cf6e1f4e Security group rule ID: sgr-0be6cc0b5cf6e1f4e IP version: IPv4 Type: HTTP Protocol: TCP Port range: 80 Source: 0.0.0.0/0

A **(Security group (sg-xxxxxx | 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**.

Protocol	Port range	Source
TCP	80	0.0.0.0/0

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**.

### Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

#### Name and tags Info

Name

Public Instance

Add additional tags

**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.

#### ▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

My AMIs

Quick Start



Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

#### Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

ami-087f352c165340ea1 (64-bit (x86), uefi-preferred) / ami-0bcaacde1147f42f7 (64-bit (Arm), uefi)  
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

#### Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.7.20250331.0 x86\_64 HVM kernel-6.1

Architecture

64-bit (x86)

Boot mode

uefi-preferred

AMI ID

ami-087f352c165340ea1

Publish Date

2025-03-29

Username

ec2-user

Verified provider

### 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**.

The screenshot shows the 'Instance type' configuration section. A dropdown menu is open, showing 't3.micro' as the selected option. Below the dropdown, there is descriptive text about the instance type, including its family (t3), vCPUs (2), memory (1 GiB), current generation status, and On-Demand pricing for SUSE, Ubuntu Pro, Windows, RHEL, and Linux. To the right of the dropdown, there is a 'All generations' toggle switch and a 'Compare instance types' link. At the bottom left, a note states 'Additional costs apply for AMIs with pre-installed software'.

### 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)**.

The screenshot shows the 'Key pair (login)' configuration section. A dropdown menu is open, showing 'Proceed without a key pair (Not recommended)' as the selected option. To the right of the dropdown, there is a 'Default value' button and a 'Create new key pair' link.

### 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**.

The screenshot shows the 'Network settings' section of an AWS instance configuration. It includes fields for VPC (selected: 'vpc-01426b2ada331a251 (Lab VPC)'), Subnet ('subnet-0f057addcc12e9768'), Auto-assign public IP ('Enable'), Firewall security groups ('Select existing security group' selected), and Common security groups ('Public SG sg-0904d7fa7db884716'). A note at the bottom states: 'Security groups that you add or remove here will be added to or removed from all your network interfaces.'

### 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.

The screenshot shows the 'Configure storage' section. It displays a root volume configuration: '1x 8 GiB gp3 Root volume, 3000 IOPS, Not encrypted'. A note indicates: 'Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage'. Below this is a button to 'Add new volume'. A note below the volume list says: 'Click refresh to view backup information'. At the bottom, it shows '0 x File systems' with an 'Edit' link.

### 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**.

**▼ Advanced details** [Info](#)

**Domain join directory** [Info](#)

Select [Create new directory](#)

**IAM instance profile** [Info](#)

EC2InstProfile  
arn:aws:iam::070991923640:instance-profile/EC2InstProfile

[Create new IAM profile](#)

**Hostname type** [Info](#)

IP name

**DNS Hostname** [Info](#)

Enable IP name IPv4 (A record) DNS requests

Enable resource-based IPv4 (A record) DNS requests

Enable resource-based IPv6 (AAAA record) DNS requests

**Instance auto-recovery** [Info](#)

Select

**Shutdown behavior** [Info](#)

Stop

**Stop - Hibernate behavior** [Info](#)

Select

**Termination protection** [Info](#)

Select

**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
```

**User data - optional** [Info](#)

Upload a file with your user data or enter it in the field.

[Choose file](#)

```
#!/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
```

User data has already been base64 encoded

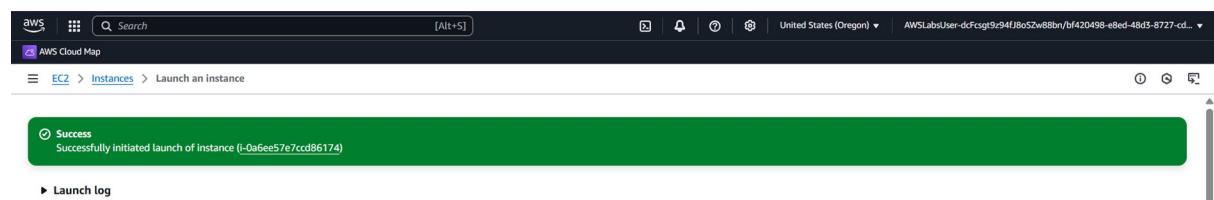
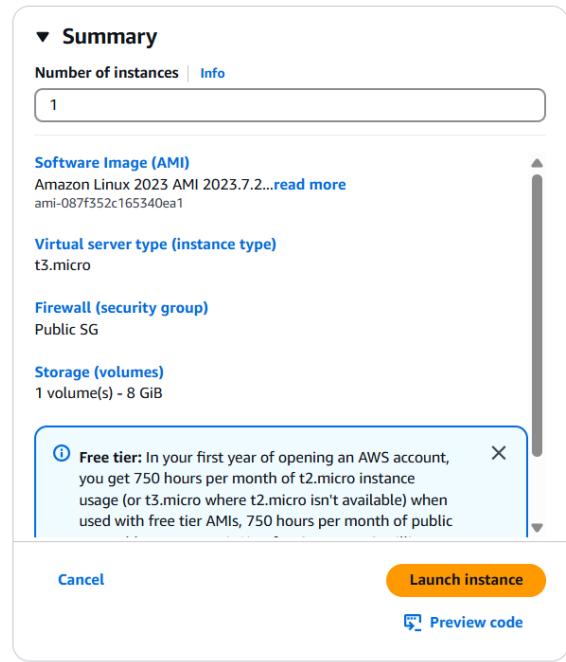
The remaining settings on the page can be left at their default values.

### 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.

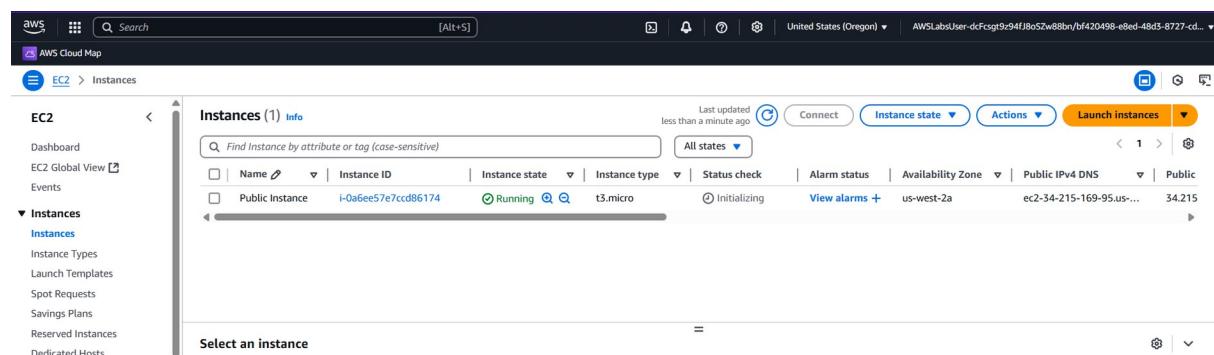
66. Choose **Launch instance**.



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**.

The screenshot shows the AWS EC2 Instances page. In the left navigation pane, under the 'Instances' section, 'Instances' is selected. The main content area displays a table of instances. One instance is selected, labeled 'Public Instance' with the ID 'i-0a6ee57e7cccd86174'. The instance status is shown as 'Running' with a green dot icon. Below the table, a detailed view for the selected instance is shown. The 'Details' tab is selected, displaying information such as Instance ID (i-0a6ee57e7cccd86174), Public IPv4 address (34.215.169.95), Instance state (Running), Hostname type (IP name: ip-10-0-0-80.us-west-2.compute.internal), Private IP DNS name (IPv4 only) (ip-10-0-0-80.us-west-2.compute.internal), and Instance type (t3.micro). The status check section indicates '3/3 checks passed'. The Networking tab is also visible in the details view.

**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.

The screenshot shows the AWS Cloud Map Instances page. On the left, there's a sidebar with navigation links for EC2, Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store. The main content area is titled 'Instances (1/1) Info' and shows a table with one row for the instance i-0a6ee57e7ccd86174. The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IPv4 address. The instance is listed as a 'Public Instance' with an 'Running' state, t3.micro instance type, 3/3 checks passed, and located in the us-west-2a Availability Zone with a Public IPv4 DNS of ec2-34-215-169-95.us-west-2.amazonaws.com. Below the table, there's a detailed view for the instance i-0a6ee57e7ccd86174, specifically the 'Networking' tab. It shows details like Public IPv4 address (34.215.169.95), Private IPv4 addresses (10.0.0.80), VPC ID (vpc-01426b2ada531a251), and various other networking configurations.

**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 screenshot shows a web browser window. The address bar displays 'Not secure' and the URL 'ec2-34-215-169-95.us-west-2.compute.amazonaws.com'. The page content includes the AWS logo and the text 'aws training and certification'. Below that, it displays 'EC2 Instance ID: i-0a6ee57e7ccd86174' and 'Zone: us-west-2a'.

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 screenshot shows the AWS Management Console with the EC2 service selected. The left sidebar shows various navigation options like Dashboard, EC2 Global View, Events, Instances, Images, Elastic Block Store, and Network & Security. The main content area displays a table titled 'Instances (1/1) Info' with one row for a 'Public Instance' with ID i-0a6ee57e7cc86174. The instance is listed as 'Running' with type 't3.micro'. The 'Networking' tab is selected in the details view for this instance, showing its public and private IP addresses, DNS names, subnet ID, and availability zone.

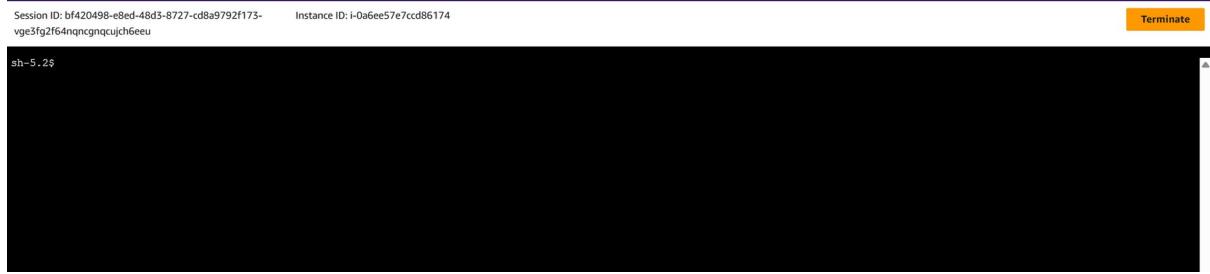
The **Connect to instance** page is displayed.

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

The screenshot shows the 'Connect to instance' page for the instance i-0a6ee57e7cc86174. The 'Session Manager' tab is selected. Below it, the 'Session Manager usage' section lists several bullet points about connecting to the instance without exposing the SSH port. At the bottom right of the page are 'Cancel' and 'Connect' buttons.

**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_C3UfQJIEVvkC5BA3xiIH3FD0nVnYw==
```

Session ID: bf420498-e8ed-48d5-8727-cd8a9792f173-vge3fg2f64nqncgnqcujhfeeu Instance ID: i-0a6ee57e7cd86174

**Terminate**

```
sh-5.2$ cd ~
curl -I https://aws.amazon.com/training/
HTTP/2.00
content-type: text/html; charset=UTF-8
date: Wed, 16 Apr 2025 11:37:46 GMT
x-content-type-options: nosniff
server: Server
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block
strict-transport-security: max-age=63072000
x-amz-id-2: 90b21fd56256eda6d1379e32829c4c446
last-modified: Fri, 04 Apr 2025 08:17:39 GMT
vary: accept-encoding
set-cookie: aws-priv=yJ2ijoxLCJldSI6MCwic3QlojB9; Version=1; Comment="Anonymous cookie for privacy regulations"; Domain=.aws.amazon.com; Max-Age=31536000; Expires=Thu, 16 Apr 2026 11:37:46
GMT; Path=/; Secure
set-cookie: aws_lang=en; Domain=.amazon.com; Path=/
x-cache: Miss from cloudfront
via: 1.1 9b21fd56256eda6d1379e32829c4c446.cloudfront.net (CloudFront)
x-amz-cf-pop: SEA73-P2
x-amz-cf-id: USLk5j6m5s4VQEHrs9PvbGsnG8Ep_keTvguqqWeZ_UZ5pe4siMZgQ==

sh-5.2$
```

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.

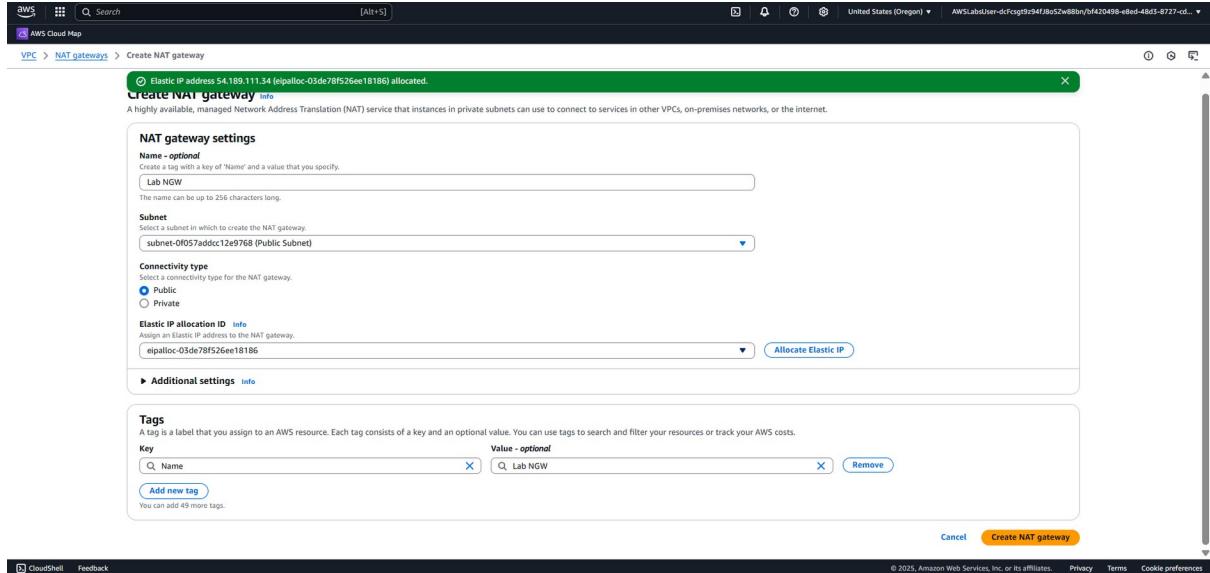
**Note:** To create a NAT gateway, you must specify the public subnet in which the NAT gateway should reside. You must also specify an Elastic IP address to associate with the NAT gateway when you create it. You cannot change the Elastic IP address after you associate it with the NAT gateway. After you've 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**.

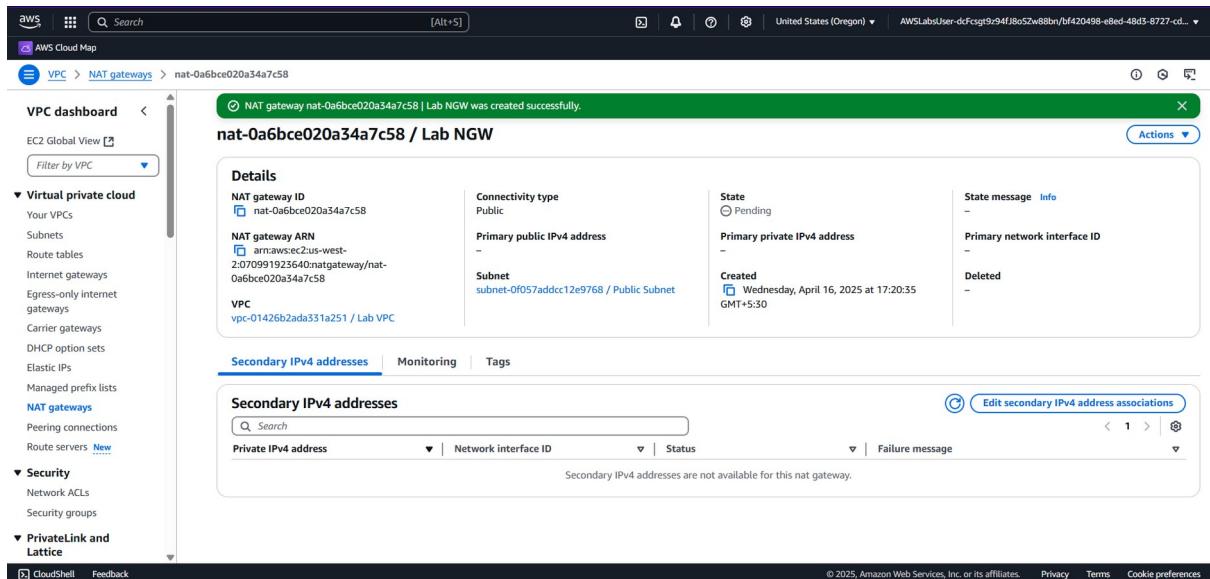
The screenshot shows the AWS Management Console interface for the VPC service. The left sidebar has a tree view with 'Virtual private cloud' expanded, showing 'Your VPCs', 'Subnets', 'Route tables', 'Internet gateways', 'Egress-only internet gateways', 'Carrier gateways', 'DHCP option sets', 'Elastic IPs', 'Managed prefix lists', and 'NAT gateways' (which is currently selected). Other collapsed sections include 'Security', 'PrivateLink and Lattice', and 'AWS Cloud Map'. The main content area is titled 'NAT gateways' and contains a table with the following columns: Name, NAT gateway ID, Connectivity..., State, State message, Primary public I..., Primary private I..., and Primary network. A search bar at the top of the table says 'Find resources by attribute or tag'. A message at the top right of the table area says 'No NAT gateways found'. Below the table, there is a section titled 'Select a NAT gateway' with three small icons. At the bottom of the page, there are links for 'cloudShell', 'Feedback', and copyright information: '© 2025, Amazon Web Services, Inc. or its affiliates.' and 'Privacy Terms Cookie preferences'.

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-xxxxxx | 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**.

**Route tables (3) Info**

Name	Route table ID	Explicit subnet associ...	Main	VPC	Owner ID
-	rtb-0693b50aaeb6959	-	-	ypc-00b2a8e49e31269c3	070991923640
-	rtb-066c62df8c79a456d	-	-	ypc-01426b2ada331a251   Lab ...	070991923640
Public Route Table	rtb-018af28004c3bcd48	subnet-0f057addcc12e97...	-	ypc-01426b2ada331a251   Lab ...	070991923640

Select a route table

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

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

**Create route table** Info

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

**Route table settings**

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

**VPC**  
The VPC to use for this route table.  
**vpc-01426b2ada331a251 (Lab VPC)**

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

**Value - optional**  
Key: Name Value: Private Route Table  
**Add new tag**  
You can add 49 more tags.

**Create route table**

89. Choose **Create route table**.

**Route table rtb-088a0bc7476bd8acf | Private Route Table was created successfully.**

**rtb-088a0bc7476bd8acf / Private Route Table**

**Details** Info

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-088a0bc7476bd8acf	No	-	-
VPC	Owner ID		
vpc-01426b2ada331a251   Lab VPC	070991923640		

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

**Routes (1)**

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

A **(Route table rtb-xxxxxx | 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.

Destination	Target	Status	Propagated
10.0.0.16	local	Active	No
0.0.0.0/0	NAT Gateway	-	No

93. Choose **Save changes**.

Updated routes for rtb-088a0bc7476bd8acf / Private Route Table successfully

Destination	Target	Status	Propagated
0.0.0.0/0	nat-0a6bce020a34a7c58	Active	No
10.0.0.16	local	Active	No

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

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

95. Choose **Edit subnet associations**.

The screenshot shows the AWS VPC dashboard with the 'Route tables' section selected. A success message at the top states: "Updated routes for rtb-088a0bc7476bd8acf / Private Route Table successfully". Below it, the "rtb-088a0bc7476bd8acf / Private Route Table" details are shown. The "Subnet associations" tab is active, displaying a table with one row: "Private Subnet" (subnet-087a8c37fb0d925f2) associated with "Main" route table ID rtb-018af28004c3bcd48. The "Explicit subnet associations" section is empty. The "Edit subnet associations" button is visible.

96. Select **Private Subnet**.

The screenshot shows the "Edit subnet associations" dialog box. Under "Available subnets (1/2)", the "Private Subnet" (subnet-087a8c37fb0d925f2) is selected. In the "Selected subnets" section, it is listed as "subnet-087a8c37fb0d925f2 / Private Subnet". At the bottom right are "Cancel" and "Save associations" buttons, with "Save associations" being highlighted.

97. Choose **Save associations**.

The screenshot shows the AWS VPC dashboard with the 'Route tables' section selected. A success message at the top states: "You have successfully updated subnet associations for rtb-088a0bc7476bd8acf / Private Route Table." Below it, the "rtb-088a0bc7476bd8acf / Private Route Table" details are shown. The "Subnet associations" tab is active, displaying a table with one row: "Private Subnet" (subnet-087a8c37fb0d925f2) associated with "Main" route table ID rtb-018af28004c3bcd48. The "Explicit subnet associations" section is empty. The "Edit subnet associations" button is visible.

A **(You have successfully updated subnet associations for rtb-xxxxxx / 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.

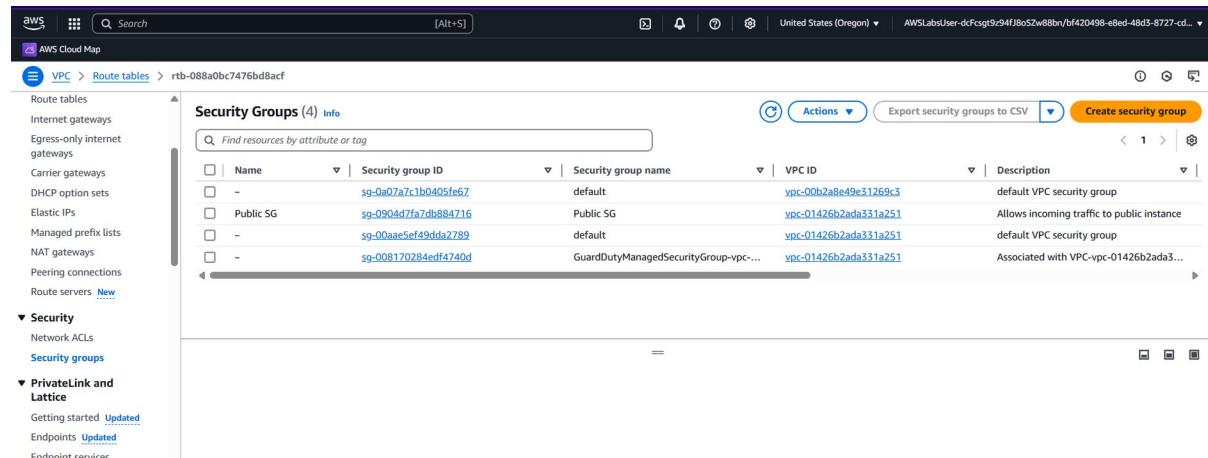
Congratulations! You have successfully created the NAT gateway and configured the private route table.

## 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 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.

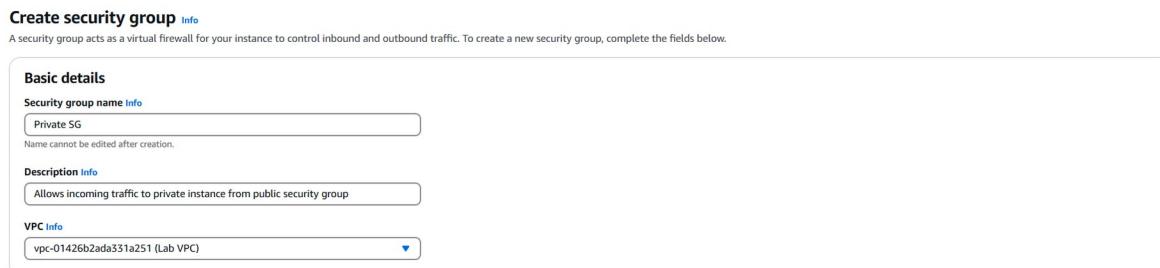
98. In the left navigation pane, choose **Security groups**.



The screenshot shows the AWS VPC Security Groups page. The left sidebar has sections for Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, and Route servers. Under the Security section, Network ACLs and Security groups are listed. The main area shows a table titled "Security Groups (4) Info" with columns: Name, Security group ID, Security group name, VPC ID, and Description. The table contains four rows: "default" (sg-0a07a7c1b0405fe67), "Public SG" (sg-0904d7fa7db884716), "default" (sg-00aae5ef49dd2789), and "GuardDutyManagedSecurityGroup-vpc-..." (sg-008170284edf4740d). Buttons for Actions, Export security groups to CSV, and Create security group are at the top right.

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.



The screenshot shows the "Create security group" configuration page. It has a "Basic details" section with fields for Security group name (Private SG), Description (Allows incoming traffic to private instance from public security group), and VPC (vpc-01426b2ada331a251 (Lab VPC)). A note at the top says: "A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below."

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.

Inbound rules [Info](#)

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
HTTP	TCP	80	Custom <a href="#">▼</a>	<input type="text" value="sg-0904d7fa7db884716"/> <a href="#">X</a> sg-0904d7fa7db884716 <a href="#">X</a>

[Delete](#)

[Add rule](#)

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

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

Outbound rules [Info](#)

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Destination <a href="#">Info</a>	Description - optional <a href="#">Info</a>
All traffic	All	All	Custom <a href="#">▼</a>	<input type="text" value="0.0.0.0/0"/> <a href="#">X</a> 0.0.0.0/0 <a href="#">X</a>

[Delete](#)

[Add rule](#)

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

Key	Value - optional
<input type="text" value="Name"/> <a href="#">X</a>	<input type="text" value="Private SG"/> <a href="#">X</a> <a href="#">Remove</a>

[Add new tag](#)  
You can add up to 49 more tags

[Cancel](#) [Create security group](#)

102. Choose **Create security group**.

aws [Search](#) [Alt+S]

AWS Cloud Map

VPC > Security Groups > sg-0b9105c96fc927c44 - Private SG

Security group (sg-0b9105c96fc927c44 | Private SG) was created successfully

sg-0b9105c96fc927c44 - Private SG

Actions [▼](#)

Details

Security group name <a href="#">Private SG</a>	Security group ID <a href="#">sg-0b9105c96fc927c44</a>	Description <a href="#">Allows incoming traffic to private instance from public security group</a>	VPC ID <a href="#">vpc-01426b2ada331a251</a>
Owner <a href="#">070991923640</a>	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules | Outbound rules | Sharing - new | VPC associations - new | Tags

Inbound rules (1)

Inbound rules (1)						
<a href="#">Edit inbound rules</a>						
<a href="#">Manage tags</a>						
<a href="#">Search</a>						
Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-0711f06d66608e814	-	HTTP	TCP	80	sg-0904d7fa7db884716

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

A **(Security group (sg-xxxxxx | 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 screenshot shows the AWS Management Console EC2 dashboard. The left sidebar has a navigation menu with sections like Dashboard, Instances, Images, and Elastic Block Store. The main content area displays various EC2 metrics and a 'Launch instance' button. On the right side, there's an 'Account attributes' section and an 'Explore AWS' section with promotional offers.

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 screenshot shows the 'Launch an instance' wizard. It consists of three main steps: 'Name and tags', 'Application and OS Images (Amazon Machine Image)', and 'Summary'. In the 'Summary' step, it shows the user is launching 1 instance using the 'Amazon Linux 2023 AMI 2023.7.2...' AMI, with an 't2.micro' instance type, a 'New security group' selected, and 1 volume(s) of 8 GiB storage. A note about the free tier is displayed at the bottom.

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.

EC2 > Instances > Launch an instance

### Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

#### Name and tags Info

Name

Private Instance

Add additional tags

**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.

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | My AMIs | **Quick Start**

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Browse more AMIs  
Including AMIs from AWS, Marketplace and the Community

**Amazon Machine Image (AMI)**

Amazon Linux 2023 AMI  
ami-087f352c165340ea1 (64-bit (x86), uefi-preferred) / ami-0bcaacde1147f42f7 (64-bit (Arm), uefi)  
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

**Description**  
Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.7.20250331.0 x86\_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Publish Date	Username	Verified provider
64-bit (x86)	uefi-preferred	ami-087f352c165340ea1	2025-03-29	ec2-user	

### 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.

The screenshot shows the 'Instance type' configuration section. A dropdown menu is open, showing 't3.micro' as the selected option. Below the dropdown, there is detailed information about the instance type, including its family (t3), 2 vCPU, 1 GiB Memory, and current generation status. It also lists On-Demand SUSE base pricing, On-Demand Ubuntu Pro base pricing, On-Demand Windows base pricing, and On-Demand RHEL base pricing. To the right of the dropdown, there are buttons for 'All generations' and 'Compare instance types'. A note at the bottom states 'Additional costs apply for AMIs with pre-installed software'.

### 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.

The screenshot shows the 'Key pair (login)' configuration section. A dropdown menu is open, showing 'Proceed without a key pair (Not recommended)' as the selected option. To the right of the dropdown, there are buttons for 'Default value' and 'Create new key pair'.

### 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.

**Network settings** [Info](#)

**VPC - required** [Info](#)

vpc-01426b2ada331a251 (Lab VPC)  
10.0.0.0/16

**Subnet** [Info](#)

subnet-087a8c37fb0d925f2 Private Subnet  
VPC: vpc-01426b2ada331a251 Owner: 070991923640 Availability Zone: us-west-2b  
Zone type: Availability Zone IP addresses available: 506 CIDR: 10.0.2.0/23

**Create new subnet** [Create new subnet](#)

**Auto-assign public IP** [Info](#)

Disable

**Firewall (security groups)** [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group  Select existing security group

**Common security groups** [Info](#)

Select security groups

Private SG sg-0b9105c96fc927c44 [X](#)  
VPC: vpc-01426b2ada331a251

**Compare security group rules**

Security groups that you add or remove here will be added to or removed from all your network interfaces.

**Advanced network configuration**

### 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.

**Configure storage** [Info](#) [Advanced](#)

1x  GiB  Root volume, 3000 IOPS, Not encrypted

**Add new volume**

**Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage** [X](#)

**Click refresh to view backup information** [Edit](#)  
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems [Edit](#)

### 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**.

**▼ Advanced details** [Info](#)

**Domain join directory** [Info](#)

Select [Create new directory](#)

**IAM instance profile** [Info](#)

EC2InstProfile [Create new IAM profile](#)

**Hostname type** [Info](#)

IP name

**DNS Hostname** [Info](#)

Enable IP name IPv4 (A record) DNS requests

Enable resource-based IPv4 (A record) DNS requests

Enable resource-based IPv6 (AAAA record) DNS requests

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
```

**User data - optional** [Info](#)  
Upload a file with your user data or enter it in the field.

[Choose file](#)

```
#!/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
```

User data has already been base64 encoded

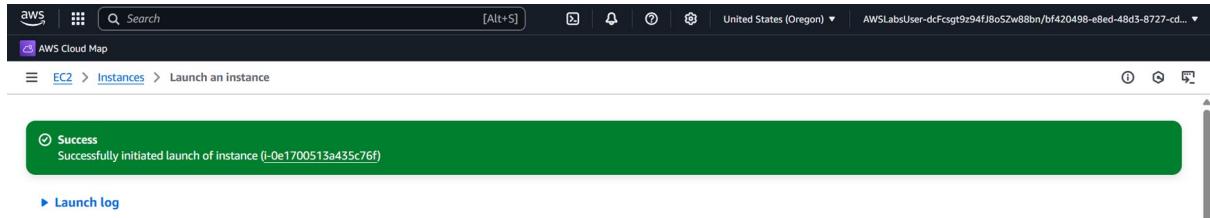
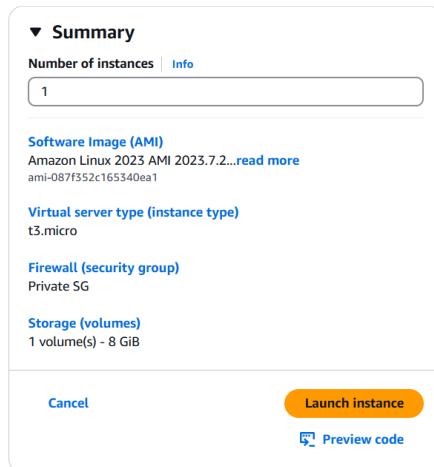
The remaining settings on the page can be left at their default values.

### 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 screenshot shows the AWS EC2 Instances page. The left navigation pane is visible with options like Dashboard, EC2 Global View, Events, Instances, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store. The main content area displays 'Instances (1/2) Info' with a search bar and filters for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IP. Two instances are listed: 'Private Instance' (i-0e1700513a435c76f) and 'Public Instance' (i-0a6ee57e7cd86174). The Private Instance is shown as 'Running' with '3/3 checks passed'. The Public Instance is also listed as 'Running'. Below the instance list, there is a detailed view for the Private Instance (i-0e1700513a435c76f), showing its details such as Instance ID, Public IPv4 address (10.0.2.71), Instance state (Running), Private IP DNS name (ip-10-0-2-71.us-west-2.compute.internal), Instance type (t3.micro), and VPC ID.

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**.

This screenshot is identical to the one above, showing the AWS EC2 Instances page. The Private Instance (i-0e1700513a435c76f) now has a green 'Running' status with '3/3 checks passed' highlighted. The Public Instance remains 'Running' with '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:** Tjphb1UhSXmyHvybuq4QIFwzTurEI0g\_saLB2nLjYRiBbHbqn85Q==

```

Session ID: bfa42049b-e8ed-48d3-8727-cd8a9792f173- Instance ID: i-0e1700513a435c76f
547yjue8y2qx8ycncnb97caj Terminate

sh-5.2$ cd ~
curl -I https://aws.amazon.com/training/
HTTP/2 200
content-type: text/html; charset=UTF-8
date: Wed, 16 Apr 2025 12:25:15 GMT
server: Server
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block
strict-transport-security: max-age=63072000
x-amz-id-1: E91908C10ABC4501BBC7
last-modified: Fri, 04 Apr 2025 08:17:39 GMT
vary: accept-encoding
set-cookie: aws-priv=eyJ2IjoxLCJldSI6MCwic3QiOjB9; Version=1; Comment="Anonymous cookie for privacy regulations"; Domain=.aws.amazon.com; Max-Age=31536000; Expires=Thu, 16 Apr 2026 12:25:15
GMT; Path=/; Secure
set-cookie: aws_lang=en; Domain=.amazon.com; Path=/
x-content-type-options: nosniff
x-cache: Miss from cloudfront
via: 1.1 0c96ded4ff282d2dbc47c918b6bb500.cloudfront.net (CloudFront)
x-amz-cf-pop: HIO30-C1
x-amz-cf-id: ytwERV2Vo4Fc9SI9rlN52bNdvrB9yY-QWWTOqSXEFeh-wIqIFMTYmQ==

sh-5.2$
```

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.

The screenshot shows the AWS EC2 Instances page. The left sidebar includes sections for EC2 (Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots, Lifecycle Manager). The main content area displays 'Instances (1/2) Info' with a table of instance details. A modal window for 'i-0e1700513a435c76f (Private Instance)' is open, showing the 'Details' tab with fields like Instance ID, Public IPv4 address, Instance state, Private IP DNS name, Instance type, VPC ID, and Network interface details. The 'Public IPv4 address' field has a copy icon next to it.

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

137. Unselect **Private Instance**.

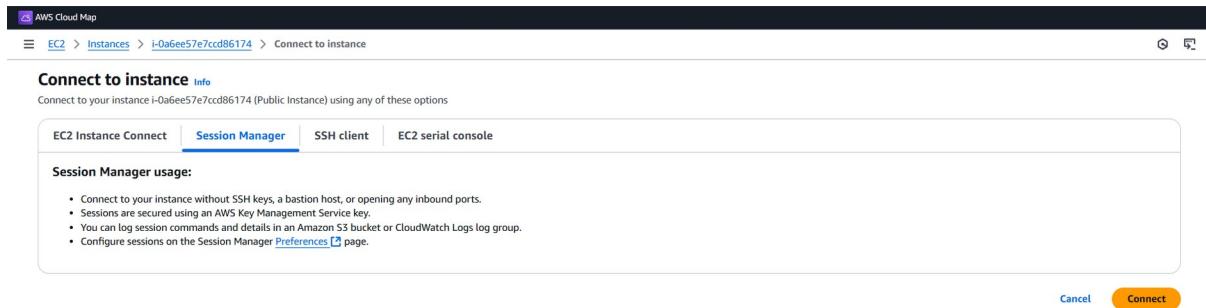
138. Select **Public Instance**.

139. Choose **Connect**.

This screenshot is identical to the previous one, but the 'Public Instance' (i-0a6ee57e7cd86174) is now selected in the list. The 'Connect' button is highlighted in yellow at the top of the page. The modal window for the Public Instance shows the 'Details' tab with fields like Instance ID, Public IPv4 address (34.215.169.95), Instance state (Running), Private IP DNS name (ip-10-0-0-80.us-west-2.compute.internal), Instance type (t3.micro), and Network interface details. The 'Public IPv4 address' field has a copy icon next to it.

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"></th>
<th class="topleft"></th>
<td id="tbl-border-top">&ampnbsp</td>
<th class="topright"></th>
<th rowspan="3" class="sized"></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>
            
            <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">&nbsp;</td>
    <th class="sized bottomright"></th>
</tr>
</table>
<div class="clear">&nbsp;</div>
</div>
</center>
<!-- end content -->
<div class="clear">&nbsp;</div>
</div>
<!-- end content-outer.....END -->
<div class="clear">&nbsp;</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"></th>
            <th class="topleft">&ampnbsp</th>
            <th id="tbl-border-top">&ampnbsp</th>
            <th class="topright"></th>
            <th rowspan="3" class="sized"></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>
                            
                            <br/>
                            <br/>
                            <h2>EC2 Instance ID: i-0e1700513a435c76f</h2>
                            <h2>Zone: us-west-2b</h2>
                        </center>
                    </div>
                    <!-- end table-content -->
                <!-- end content-table-inner ..... END -->
            </td>
            <td id="tbl-border-right"></td>
        </tr>
        <tr>
            <th class="sized bottomleft"></th>
            <td id="tbl-border-bottom">&ampnbsp</td>
            <th class="sized bottomright"></th>
        </tr>
    </table>
    <div class="clear">&ampnbsp</div>
</div>
</center>
<!-- end content -->
<div class="clear">&ampnbsp</div>
</div>
<!-- end content-outer.....END -->

<div class="clear">&ampnbsp</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**.

Session ID: bf420498-e8ed-48d5-8727-cd8a9792f173-  
6fdhzbcsbgv4yj5hv9zup7uvu      Instance ID: i-0a6ee57e7ccdb86174      Terminate

```
sh-5.2$ ping 10.0.2.71
PING 10.0.2.71 (10.0.2.71) 56(84) bytes of data.
```

Session ID: bf420498-e8ed-48d5-8727-cd8a9792f173-  
6fdhzbcsbgv4yj5hv9zup7uvu      Instance ID: i-0a6ee57e7ccdb86174      Terminate

```
sh-5.2$ ping 10.0.2.71
PING 10.0.2.71 (10.0.2.71) 56(84) bytes of data.
^C
--- 10.0.2.71 ping statistics ---
49 packets transmitted, 0 received, 100% packet loss, time 49895ms
sh-5.2$
```

**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 screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store. The main area displays a table of instances. One instance, i-0a6ee57e7cccd86174, is selected and shown in detail on the right. The 'Details' tab is selected, showing information such as Instance ID (i-0a6ee57e7cccd86174), Public IPv4 address (34.215.169.95), Private IP DNS name (ip-10-0-0-80.us-west-2.compute.internal), and Instance type (t3.micro). The 'Connect' button is highlighted in yellow at the top of the instance details.

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

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

151. Choose **Connect**.

The screenshot shows the 'Connect to instance' page. At the top, there's a breadcrumb trail: EC2 > Instances > i-0a6ee57e7cccd86174 > Connect to instance. Below that, there's a title 'Connect to instance' with a 'Info' link. A note says 'Connect to your instance i-0a6ee57e7cccd86174 (Public Instance) using any of these options'. There are four tabs: EC2 Instance Connect, Session Manager (selected), SSH client, and EC2 serial console. Under the Session Manager tab, there's a section titled 'Session Manager usage:' with a bulleted list: 'Connect to your instance without SSH keys, a bastion host, or opening any inbound ports.', 'Sessions are secured using an AWS Key Management Service key.', 'You can log session commands and details in an Amazon S3 bucket or CloudWatch Logs log group.', and 'Configure sessions on the Session Manager Preferences page.' At the bottom right, there are 'Cancel' and 'Connect' buttons, with 'Connect' being highlighted in yellow.

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.

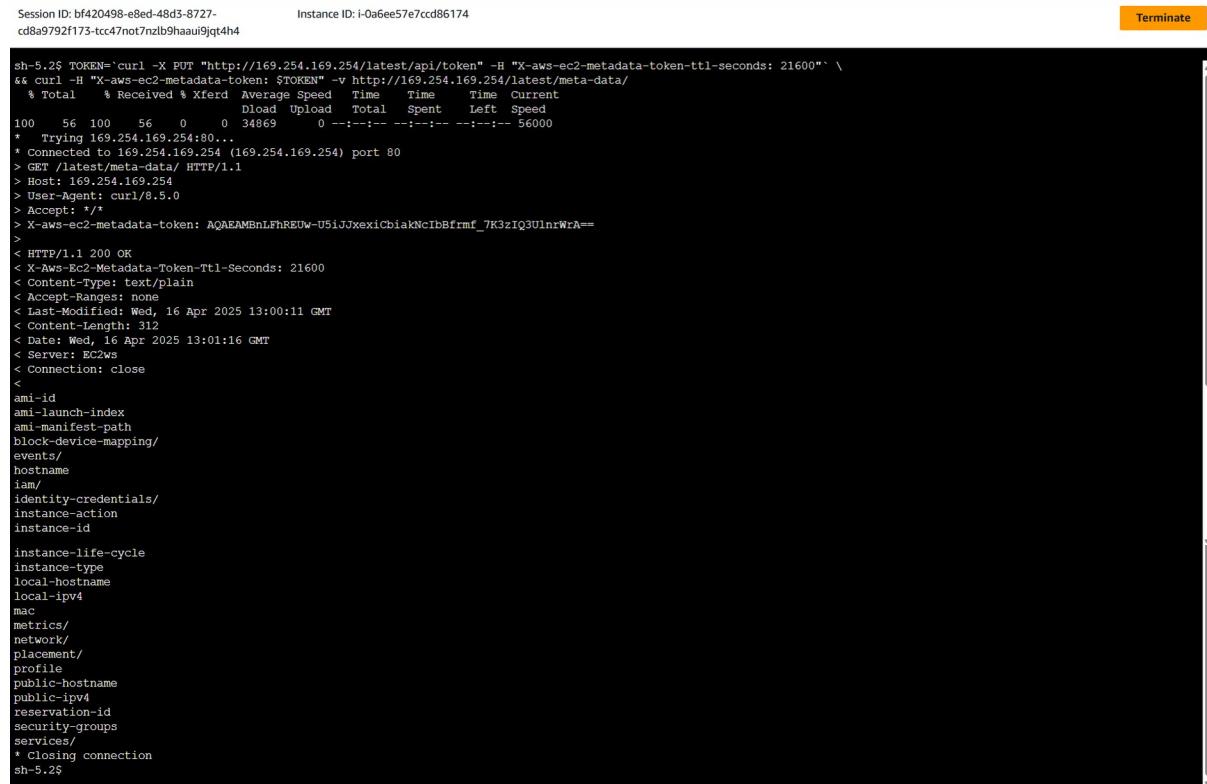
### Command 2:

```
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).



```
Session ID: bf420498-e8ed-48d3-8727-cc8a9792f173-tcc47not7nzb9haau9jqt4h4
Instance ID: i-0a6ee5e7cccd86174
Terminate

sh-5.2$ 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: $TOKEN" -v http://169.254.169.254/latest/meta-data/
% Total    % Received % Xferd  Average Speed   Time     Time   Current
          Dload  Upload Total Spent   Left Speed
100  56  100  56    0     0  34869  0 --:--:--:--:--:-- 56000
* Trying 169.254.169.254:80...
* Connected to 169.254.169.254 (169.254.169.254) port 80
> GET /latest/meta-data/ HTTP/1.1
> Host: 169.254.169.254
> User-Agent: curl/8.5.0
> Accept: */*
> X-aws-ec2-metadata-token: AQAEAMBnLFhREUw-U5iJJxexiCbiakNcIbBfrmf_7K3zIQ3UlnrWrA==
>
< HTTP/1.1 200 OK
< X-Aws-Ec2-Metadata-Token-Ttl-Seconds: 21600
< Content-Type: text/plain
< Accept-Ranges: none
< Last-Modified: Wed, 16 Apr 2025 13:00:11 GMT
< Content-Length: 312
< Date: Wed, 16 Apr 2025 13:01:16 GMT
< Server: EC2w
< Connection: close
<
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hostname
iam/
identity-credentials/
instance-action
instance-id
instance-life-cycle
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
reservation-id
security-groups
services/
* Closing connection
sh-5.2$
```

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.

---

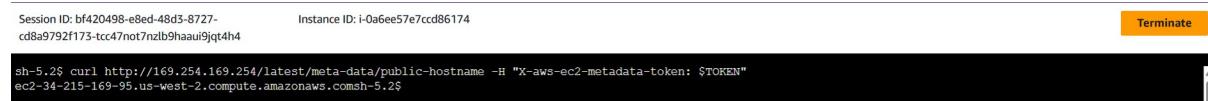
**-H "X-aws-ec2-metadata-token: \$TOKEN"**

- 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.



A screenshot of a terminal window titled 'Session ID: bf420498-e8ed-48d3-8727-cd8a9792f173-tcc47not7nzb9haau9jq4h4'. The window shows the command 'curl http://169.254.169.254/latest/meta-data/public-hostname -H "X-aws-ec2-metadata-token: \$TOKEN"' being run. The output shows the public DNS name 'ec2-34-215-169-95.us-west-2.compute.amazonaws.com'. There is a yellow 'Terminate' button in the top right corner.

**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")
```

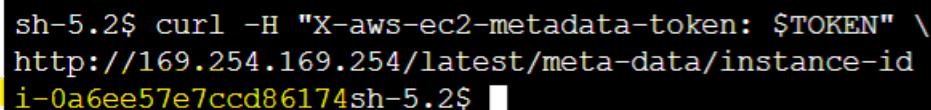
---

## □ **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
```



A screenshot of a terminal window showing the command 'curl -H "X-aws-ec2-metadata-token: \$TOKEN" http://169.254.169.254/latest/meta-data/instance-id'. The output shows the instance ID 'i-0a6ee57e7ccd86174'. The terminal prompt is 'sh-5.2\$'.

### □ **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: $TOKEN" \
http://169.254.169.254/latest/meta-data/iam/security-credentials/EC2SSMRole
{
  "Code": "Success",
  "LastUpdated": "2025-04-16T12:59:50Z",
  "Type": "AWS-HMAC",
  "AccessKeyId": "ASIAARB3S2G4GFURBQJXR",
  "SecretAccessKey": "gn8gqarzqM030e57KltQMpffohRBheZzT/JJrtdWc",
  "Token": "IQQJb3sp22lu2vZjEl5//wacxVzlxcl-3qtX1tHMEUC1QzykiDdf28eSHJWOGV9dWIC5eiU6ox7Kbjf04umyJdwlg0zqXArme9BZWMpyjuUJEmZX9VgvoxRqqZWjXc5BfCJquAUJRRAEGqwWzA50TE5MjM2NDA1DDVVURuovKLUhUk+b11qvBYjdXeurMj0l5gmW62ohx1j1R8MC-PtD13BFcJt4cgkwEXrTo-432cW+Qz2hmcpXNtp1vYnSdckxk1Pf1pzhFnfytfgsu0b0Tw88kxKnCYXWQjja5c0hZGE1BvEdd2fymgggW614W4290Qqk1R96A3tSkunraifNLGrf-UcnLsqzaLoogZ+fzvVUgv62pUacfD1hjn23RHNvNG/bok2caze0m0PyaduktJf0V3r(NePssexIVC-UgPTYqfxzpbGesalqsnVzccklqtLz31ZBgrkh+h+Z055dPozy3t68jJgnLzzZWYKch2JW9NNUKUm7/q+v12obAclUuj7+f14HMjC09z+3BwSP/7c0B7oA2g/ArHeOjucDz4gyf37GWz1fmbycds/1GrzVeXmnFn+Gig7qtENiosldHaZs8sBWQgSY8/h15uqrKdfj9.7x5wgrwf6f1yD2o10wB2Izcc3+zu1U23hmlaRxKuRu+cnHsTeP7joiqdNjskcuuk3x8t:sru7w3ec+A4+a1pU-VG7flePhvUWw/2M1xz9wRd2zGtfnDtmgu41s5F026nZN25yMmxzPf1p1dje1C3IMfmm/e0KnaRcehmpaDSN33Dxa+hmUj14H9NUG75msQ84hy2UmxcCLQ3tGavbzxWWvc4FC-Tm6nlbEd61e-5ay1B5-/zrnmMWkMsQ58TxTENngH6/0AAvjBLBuvc83-069m88VvPCM5WzqjV4Ymx1h/VnscWUdcd1fFRGxmJtQ9HsIBD13WtzyQRo0t14cx1LWmDlRr07yyssk1wP93Xn3j3Zcdiluvvczdodo81eRgj2g52Naqj1k/tkz29+rw6sQfUgl1t8j1bJMjM628+kmmmlbwzbzSvrV3exitJ5j1zW69YpmkMyMop:3he2KuG19QapGgzm9Po/Oc/p+W0D8wlAm7mpc11BwHFQcmewn07KghvO1jgxv1uPq810+YsmmpImbuH46R019dmiaor-SotAvmWrmwi5BSCA-JnKVbmkpA24o71ZdWrznMpvdfUeShaxxx/3trkwabvUyoyhong118ffPBKcr7cqHTFG2A=",
  "Expiration": "2025-04-16T19:02:22Z"
}sh-5.2$
```

## □ 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.

```
sh-5.2$ curl -H "X-aws-ec2-metadata-token: $TOKEN" \
http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hostname
iam/
identity-credentials/
instance-action
instance-id
instance-life-cycle
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
reservation-id
security-groups
services/
systemsh-5.2$ █
```

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**.

Name	Security group ID	Security group name	VPC ID	Description
-	sg-0a07a7c1b0405fe67	default	vpc-00b2a8e49e31269c3	default VPC security group
Private SG	sg-0b9105c96fc927c44	Private SG	vpc-01426b2ada331a251	Allows incoming traffic to private instance
Public SG	sg-0904d7fa7db884716	Public SG	vpc-01426b2ada331a251	Allows incoming traffic to public instance
-	sg-00aae5ef49dd2789	default	vpc-01426b2ada331a251	default VPC security group
-	sg-008170284edf4740d	GuardDutyManagedSecurityGroup-vpc-	vpc-01426b2ada331a251	Associated with VPC-vpc-01426b2ada3...

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

160. Select **Private SG**.

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

Security group ID	Security group name	Description	VPC ID
sg-0b9105c96fc927c44	Private SG	Allows incoming traffic to private instance from public security group	vpc-01426b2ada331a251

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.

**Edit inbound rules** Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0711f06d66608e814	HTTP	TCP	80	Custom	<input type="text"/> sg-0904d7fa7db884716 <small>(X)</small>
-	Custom ICMP - IPv4	All	All	Custom	<input type="text"/> sg-0904d7fa7db884716 <small>(X)</small>
					<input type="text"/> sg-0904d7fa7db884716 <small>(X)</small>

**Add rule** Cancel Preview changes Save rules

163. Choose **Save rules**.

**Inbound security group rules successfully modified on security group (sg-0b9105c96fc927c44 | Private SG)**

**Security Groups (5)** Info

Name	Security group ID	Security group name	VPC ID	Description
Private SG	sg-0b9105c96fc927c44	Private SG	vpc-01426b2ada331a251	Allows incoming traffic to private instan...
-	sg-0a07a7c1cb0405fe67	default	vpc-00b2a8e49e31269c3	default VPC security group
Public SG	sg-0904d7fa7db884716	Public SG	vpc-01426b2ada331a251	Allows incoming traffic to public instance
-	sg-00aae5ef49dd42789	default	vpc-01426b2ada331a251	default VPC security group
-	sg-000170284edf4740d	GuardDutyManagedSecurityGroup-vpc-...	vpc-01426b2ada331a251	Associated with VPC:vpc-01426b2ada3...

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*.

```
Session ID: bf420498-e8ed-48d5-8727-cd8a9792f173- Instance ID: i-0a6ee57e7cccd86174
8cd599ovkhnuxy3crpbayje0 Terminate
```

```
sh-5.2$ ping 10.0.2.71
PING 10.0.2.71 (10.0.2.71) 56(84) bytes of data.
64 bytes from 10.0.2.71: icmp_seq=1 ttl=127 time=0.685 ms
64 bytes from 10.0.2.71: icmp_seq=2 ttl=127 time=0.695 ms
64 bytes from 10.0.2.71: icmp_seq=3 ttl=127 time=0.659 ms
64 bytes from 10.0.2.71: icmp_seq=4 ttl=127 time=0.628 ms
64 bytes from 10.0.2.71: icmp_seq=5 ttl=127 time=0.637 ms
64 bytes from 10.0.2.71: icmp_seq=6 ttl=127 time=0.641 ms
64 bytes from 10.0.2.71: icmp_seq=7 ttl=127 time=0.633 ms
64 bytes from 10.0.2.71: icmp_seq=8 ttl=127 time=0.633 ms
64 bytes from 10.0.2.71: icmp_seq=9 ttl=127 time=0.642 ms
64 bytes from 10.0.2.71: icmp_seq=10 ttl=127 time=0.629 ms
64 bytes from 10.0.2.71: icmp_seq=11 ttl=127 time=0.646 ms
64 bytes from 10.0.2.71: icmp_seq=12 ttl=127 time=0.685 ms
64 bytes from 10.0.2.71: icmp_seq=13 ttl=127 time=0.642 ms
64 bytes from 10.0.2.71: icmp_seq=14 ttl=127 time=0.642 ms
64 bytes from 10.0.2.71: icmp_seq=15 ttl=127 time=0.628 ms
64 bytes from 10.0.2.71: icmp_seq=16 ttl=127 time=0.626 ms
64 bytes from 10.0.2.71: icmp_seq=17 ttl=127 time=0.621 ms
64 bytes from 10.0.2.71: icmp_seq=18 ttl=127 time=0.633 ms
64 bytes from 10.0.2.71: icmp_seq=19 ttl=127 time=0.714 ms
64 bytes from 10.0.2.71: icmp_seq=20 ttl=127 time=0.672 ms
64 bytes from 10.0.2.71: icmp_seq=21 ttl=127 time=0.672 ms
64 bytes from 10.0.2.71: icmp_seq=22 ttl=127 time=0.635 ms
64 bytes from 10.0.2.71: icmp_seq=23 ttl=127 time=0.659 ms
64 bytes from 10.0.2.71: icmp_seq=24 ttl=127 time=0.689 ms
64 bytes from 10.0.2.71: icmp_seq=25 ttl=127 time=0.700 ms
64 bytes from 10.0.2.71: icmp_seq=26 ttl=127 time=0.633 ms
64 bytes from 10.0.2.71: icmp_seq=27 ttl=127 time=0.939 ms
64 bytes from 10.0.2.71: icmp_seq=28 ttl=127 time=0.680 ms
64 bytes from 10.0.2.71: icmp_seq=29 ttl=127 time=0.694 ms
^C
sh-5.2$ ping 10.0.2.71
ping statistics --
29 packets transmitted, 29 received, 0% packet loss, time 29097ms
rtt min/avg/max/mdev = 0.612/0.663/0.939/0.058 ms
```

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](#).