aws.qwiklabs.com

Architecting on AWS - Lab 2 - Build your Amazon VPC infrastructure | Qwiklabs

Qwiklabs

31-39 minutes

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

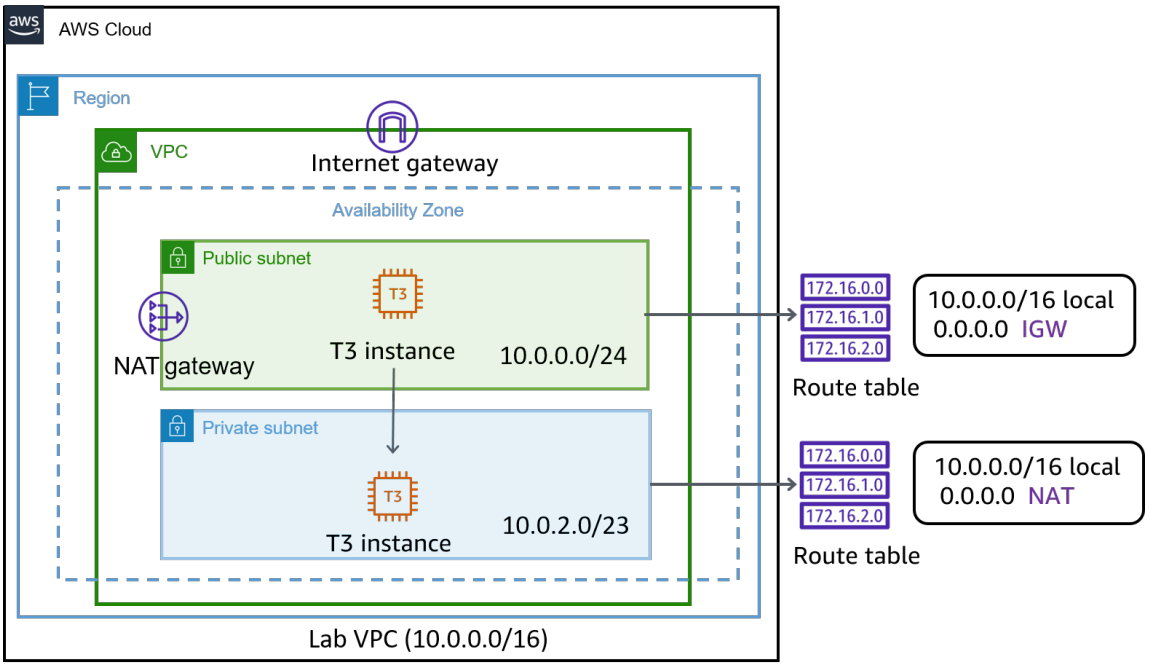
Corrections, feedback, or other questions? Contact us at *AWS Training and Certification*.

Lab overview

As an AWS solutions architect, it is important that you understand the overall functionality and capabilities of AWS, and the relationship between the AWS networking components. In this lab you create an Amazon Virtual Private Cloud (VPC),

subnets across multiple Availability Zones (AZs), public and private routes, a NAT gateway, and an Internet Gateway. These services are the foundation of networking architecture inside of AWS. This 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:

- 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 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

Prerequisites

This lab requires:

- Access to a notebook computer with Wi-Fi and Microsoft Windows, macOS, or Linux (Ubuntu, SuSE, or Red Hat)
- An internet browser such as Chrome, Firefox, or Microsoft Edge
- A plaintext editor

Duration

This lab requires up to 45 minutes to complete.

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.

Start Lab

1. At the top of your screen, launch your lab by choosing Start Lab

   This starts the process of provisioning your lab resources. An estimated amount of time to provision your lab resources is displayed. You must wait for your resources to be provisioned before continuing.

   If you are prompted for a token, use the one distributed to you (or credits you have purchased).

2. Open your lab by choosing Open Console

   This opens an AWS Management Console sign-in page.

3. On the sign-in page, configure:

- **IAM user name:**
- **Password:** Paste the value of **Password** from the left side of the lab page
- Choose Sign In

Do not change the Region unless instructed.

Common Login Errors

Error: You must first log out

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

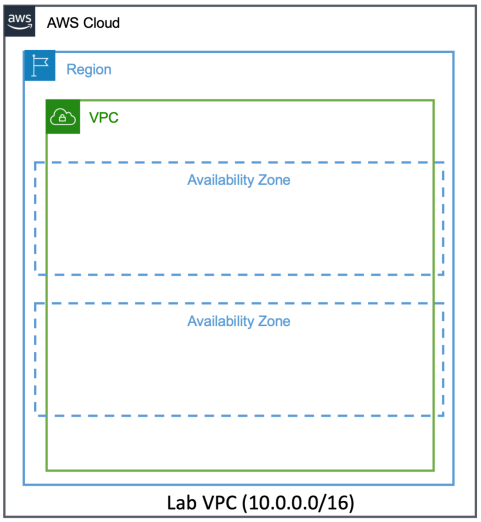
- Choose **here**
- Close your browser tab to return to your initial lab window
- Choose Open Console again

Task 1: Create an Amazon VPC in a region

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

Amazon VPC lets you provision a logically isolated section of the Amazon Web Services (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 leverage the enhanced security options

in Amazon VPC to provide more granular access to and from the Amazon EC2 instances in your virtual network.



4. In the AWS Management Console, on the Services menu, choose **VPC**.

   Note: You can also locate a service in the AWS Management Console by searching for it by name in the unified search bar at the top-center of the page. The unified search bar is located to the right of the Services menu, and it is labeled:

   Search for services, features, marketplace products, and docs

   Caution: This lab is designed to use the new VPC Console. If **New VPC Experience** is displayed at the top-left of your screen, ensure **New VPC Experience** is selected.

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

   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.

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

   A list of your VPCs displays. A default VPC is provided so that

you can launch resources as soon as you start using AWS.

6. Choose Create VPC and configure:

- **Name tag:**
- **IPv4 CIDR block:**

7. Choose the Create VPC button.
8. Verify the state. It should display the following:

- **State:** Available

The Lab VPC has a 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.

9. From the same page, choose Actions and select *Edit DNS hostnames*.

This option assigns a *friendly* DNS name to Amazon EC2 instances in the VPC, such as the following:

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

10. Select **Enable**.
11. Choose the Save changes button.

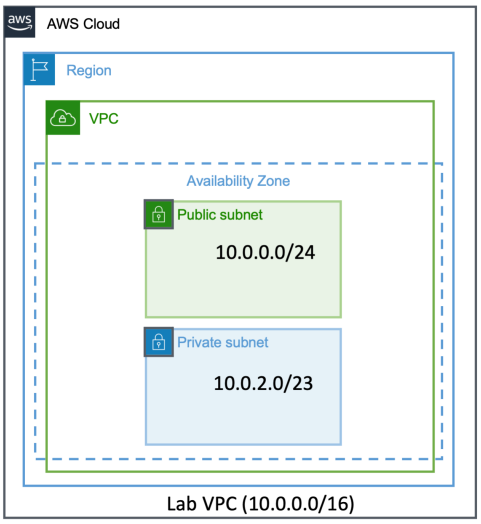
Any Amazon EC2 instances launched into this Amazon VPC now automatically receive a DNS hostname. You can also add a more meaningful DNS name (for example, *app.company.com*) by using Amazon Route 53.

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

Task 2: Create Public 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.



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.

12. In the left navigation pane, choose **Subnets**.
13. Choose Create subnet and configure:

- **VPC:** Select *Lab VPC*
- **Subnet name:**
- **Availability Zone:** Select the **first** Availability Zone in the list (Do **not** choose *No Preference*.)

- **IPv4 CIDR block:**

14. Choose the Create subnet button.
15. Verify the state . It should display the following:

- **State:** Available

  Note: The VPC has a CIDR range of **10.0.0.0/16**, which includes all **10.0.x.x** IP addresses. The subnet you just created has a CIDR range of **10.0.0.0/24**, which includes all **10.0.0.x** IP addresses. These ranges may 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.

16. Select **Public Subnet**.
17. Choose Actions and select **Edit subnet settings**.
18. Select **Enable auto-assign public IPv4 address**.
19. Choose the Save button.

    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.

20. Choose Create subnet then configure:

- **VPC:** Select *Lab VPC*
- **Subnet name:**
- **Availability Zone:** Select the **first** Availability Zone in the list

(Do **not** choose *No Preference*.)

- **IPv4 CIDR block:**

21. Choose the Create subnet button.
22. Verify the state. It should display the following:

- **State:** Available

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

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

Task 3: Create an Internet gateway

In this task, you create an internet gateway so that internet traffic can access the public subnet. To enable 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.

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

23. In the left navigation pane, choose **Internet Gateways**.
24. Choose Create internet gateway and configure:

- **Name tag:**

25. Choose the Create internet gateway button.

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

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

27. For **VPC**, select **Lab VPC**.

28. Choose the Attach internet gateway button.

29. Verify the state. It should display the following:

- **State: Attached**

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

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.

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.

30. In the left navigation pane, choose **Route Tables**.

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

31. Choose Create route table then configure:

- **Name:**
- **VPC:** Select *Lab VPC*
- Choose the Create route table button.

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

    Notice that 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 enable public traffic.

33. Choose Edit routes

34. Choose Add route then configure:

- **Destination:**
- **Target:** Select **Internet Gateway** in the drop down and then select the displayed **Internet Gateway ID**
- Choose the Save changes button.

35. Choose the **Subnet Associations** tab.

36. Choose the Edit subnet associations button.

37. Select **Public Subnet**

38. Choose the Save associations button.

    You have configured the route table. The subnet is now *public* because it has a route to the internet via the internet gateway.

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.

Amazon EC2 security groups can be used to help secure instances within an Amazon VPC. Security groups in a VPC enable you to specify both inbound and outbound network traffic that is allowed to or from each Amazon EC2 instance. Traffic which is not explicitly allowed to or from an instance is automatically denied.

HTTPS protocol is recommended for improved security, but to simplify the lab you use *HTTP* protocol.

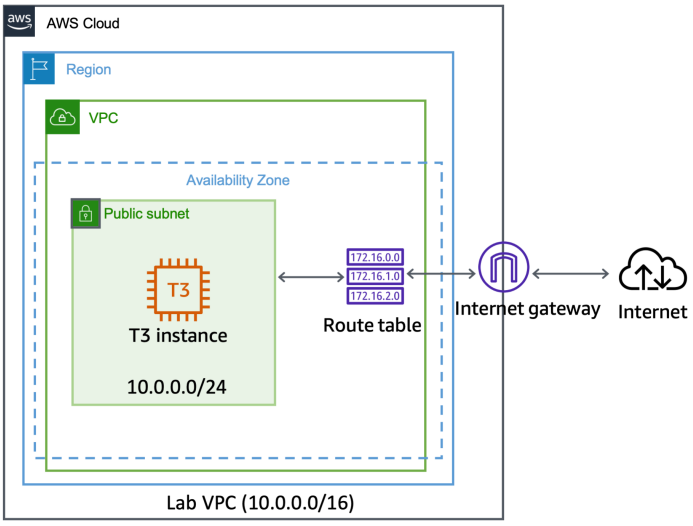
39. In the left navigation pane, choose **Security Groups**.
40. Choose Create security group then configure:
    - **Security group name:**
    - **Description:**
    - **VPC:** Select the **X** to clear the text box and then select *Lab VPC* from the drop-down list.
41. In the **Inbound rules** section:
    - Choose the Add rule button.
    - **Type:** HTTP
    - **Source:** *Anywhere-IPv4*

42. In the **Tags** section:
    - Choose the Add new tag button.
    - **Key:**
    - **Value:**
43. Choose the Create security group button.

    You have successfully created a security group which 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 enable 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.



The internet gateway that was created logically provides the one-to-one NAT on behalf of your instance, so that 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.

44. On the Services menu, choose **EC2**.

    Caution This lab is designed to use the new EC2 Console. If you see **New EC2 Experience** at the top-left of your screen, ensure **New EC2 Experience** is selected.

45. In the left navigation pane, choose **Instances**.
46. Choose the Launch instances button.
47. Choose Select next to **Amazon Linux 2 AMI**.
48. On the **Choose an Instance Type** page, select **t3.micro**.

    You are launching a t3.micro Amazon EC2 instance. This instance type has 2 vCPUs and 1 GiB of memory.

49. Choose the Next: Configure Instance Details button.

    To install and configure the new instance as web server, you provide a User Data script that will automatically run when the instance launches.

50. On the **Configure instance details** page, configure:

    - For **Network**, select **Lab VPC**
    - For **Subnet**, select **Public Subnet**
    - For **Auto-assign Public IP**, select **Use subnet setting (Enable)**
    - For **IAM role**, select the role named like **xxxx-EC2InstProfile-xxxx**
    - Scroll down to and expand **Advanced Details**. Copy and paste the following into **User data**:

      ```
      #!/bin/bash
      ```

```
# To connect to your EC2 instance and install
the Apache web server with PHP

yum update -y &&
amazon-linux-extras install -y lamp-
mariadb10.2-php7.2 php7.2 &&
yum install -y httpd &&
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/v7.1.1/lab-2-
VPC/scripts/instanceData.zip
unzip instanceData.zip
```

- Accept the default values for the remaining options

51. Choose the Next: Add Storage button.

    Default configurations on this page work for this lab.

52. Choose the Next: Add Tags button.
53. Choose Add Tag then configure:
    - **Key:**
    - **Value:**
54. Choose the Next: Configure Security Group button.
55. On the **Configure Security Group** page, choose the option **Select an existing security group**.
56. Select the security group with **Public SG** in the name.
57. Choose the Review and Launch button.

58. There is a warning box displayed regarding adding an open port 22 to the security group. Port 22 is not needed in this lab. This message box can safely be closed. Choose the Continue button.
59. Review the instance launch details, then choose the Launch button.
60. On the **Select an existing key pair or create a new key pair** window, configure the following:
    - Select **Proceed without a key pair**
    - Select **I acknowledge that without a key pair..**
    - Choose the Launch Instances button.
61. On the **Launch Status** page, choose the View Instances button.

    This brings you to the **Instances** window where you can watch your instance launch and view its details.
62. Wait for your **Public Instance** to fully launch. It should display the following:
    - **Instance State:** Running

    You can choose the refresh icon to refresh your instances status.

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

Task 7. Connect to a public instance via HTTP

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

63. In the left navigation pane, choose **Instances**.
64. Select **Public Instance**.
65. On the **Details** tab, copy the **Public IPv4 address** to your clipboard.

    Tip To copy the *Public IPv4 address* , hover on it and select the copy icon.

    Caution Do not select **open address** to open the IP Address.
66. Open a new web browser tab, paste the *Public IPv4 address* into the address bar, and press Enter.

    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.

    You have successfully launched an Apache web server in the public subnet and tested the HTTP connection.

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

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

Session Manager is a fully managed AWS Systems Manager capability that lets you manage your Amazon EC2 instances through an interactive one-click browser-based shell or through the AWS CLI. You can use Session Manager to start a session with an Amazon EC2 instance in your account. After the session is started, you can run bash commands as you would through any other connection type.

67. In the left navigation pane, choose **Instances**.
68. Select **Public Instance** and then choose the **Connect** button.

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

69. For **Connection method**, Select the **Session Manager** tab.

    With Session Manager, you can connect to Amazon EC2 instances without requiring exposing the SSH port on your firewall or Amazon Virtual Private Cloud (Amazon VPC) security group. Refer to AWS Systems Manager Session Manager for more information.

70. Choose the **Connect** button.

    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 you have allowed a few minutes time for the instance to launch, pass health checks, and communicate with the Sessions Manager service before trying to open a session connection once again.

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

    ```
    cd ~

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

    Sample Output:

    ```
    HTTP/2 200
    content-type: text/html;charset=UTF-8
    server: Server
    date: Fri, 14 May 2021 14:30:15 GMT
    x-amz-rid: 3WNTZRMBGMP8AP6HT4K7
    ```

```
set-cookie: aws-
priv=eyJ2IjoxLCJldSI6MCwic3QiOjB9; Version=1;
Comment="Anonymous cookie for privacy
regulations"; Domain=.aws.amazon.com; Max-
Age=31536000; Expires=Sat, 14-May-2022 14:30:15
GMT; Path=/
set-cookie: aws_lang=en; Domain=.amazon.com;
Path=/
x-frame-options: SAMEORIGIN
x-content-type-options: nosniff
x-amz-id-1: 3WNTZRMBGMP8AP6HT4K7
last-modified: Tue, 11 May 2021 17:39:32 GMT
vary: accept-encoding,Content-Type,Accept-
Encoding,X-Amzn-CDN-Cache,X-Amzn-AX-
Treatment,User-Agent
x-cache: Miss from cloudfront
via: 1.1
86561b4243b7d0478ca4582dd013e00e.cloudfront.net
(CloudFront)
x-amz-cf-pop: ATL52-C1
x-amz-cf-id: 3VxbxlST1HIeEd4sdFtNepfu7jnEjQB-
4gqet8mmAL5mbU0sQolzJw==
```

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 configure 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. Then you

attach the route table to the private subnet. You can use a NAT gateway to enable 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.

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. The Elastic IP address cannot be changed 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 enables instances in your private subnets to communicate with the internet.

72. Go to the tab with the AWS Management Console open.
73. On the Services menu, choose **VPC**.
74. In the left navigation pane, choose **NAT gateways**.
75. Choose Create NAT gateway and configure:
    - **Name:**
    - **Subnet:** Select *Public Subnet*
    - Select Allocate Elastic IP
76. Choose Create NAT gateway

    In the next step, you create a new Route Table for Private Subnet that redirects non-local traffic to the NAT gateway.
77. In the left navigation pane, choose **Route Tables**.
78. Choose Create route table and configure:
    - **Name:**
    - **VPC:** Select *Lab VPC*

- Choose the Create route table button.

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

79. 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.

80. Choose the Edit routes button then:

- Choose the Add route button.
- **Destination:**
- **Target:** Select **NAT Gateway** in the drop down and then select the displayed **NAT Gateway** id
- Choose then Save changes button.

81. Choose the **Subnet Associations** tab.
82. Choose the Edit subnet associations button.
83. Select **Private Subnet**.
84. Choose the Save associations button.

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

    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 HTTPS traffic from resources assigned to the public security

group.

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 protocol and port. 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.

85. In the left navigation pane, choose **Security Groups**.
86. Choose Create security group then configure:
    - **Security group name:**
    - **Description:**
    - **VPC:** Select the **X** to clear the text box and then select *Lab VPC* from the drop-down list.
87. In the **Inbound rules** section:
    - Choose the Add rule button.
    - **Type:** *HTTPS*
    - **Source type:** *Custom*
    - **Source:**
    - In the box to the right of Custom, type
    - Select *Public SG* from the list that appears
88. In the **Tags** section:
    - Choose the Add new tag button.
    - **Key:**
    - **Value:**

89. Choose the Create security group button.

    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.

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.

90. On the Services menu, choose **EC2**.

    If you see **New EC2 Experience** at the top-left of your screen, ensure **New EC2 Experience** is selected. This lab is designed to use the new EC2 Console.

91. In the left navigation pane, choose **Instances**.
92. Choose the Launch instances button.
93. Choose the Select button next to **Amazon Linux 2 AMI**.
94. On the **Choose an Instance Type** page, select **t3.micro**.

    You launch a t3.micro Amazon EC2 instance. This instance type has 2 vCPU and 1 GiB of memory.

95. Choose the Next: Configure Instance Details button.
96. On the **Configure instance details** page, configure:

- For **Network**, select **Lab VPC**

- For **Subnet**, select **Private Subnet**
- For **Auto-assign Public IP**, select **Use subnet setting (Disable)**
- For **IAM role:**, select the role named like **xxxx-EC2InstProfile-xxxx**
- Accept the default values for the remaining options

97. Choose the Next: Add Storage button.
98. Choose the Next: Add Tags button.
99. Choose the Add Tag button then configure:
    - **Key:**
    - **Value:**
100. Choose Next: Configure Security Group
101. On the **Configure Security Group** page, choose the option **Select an existing security group**.
102. Select the security group with **Private SG** in the name.
103. Choose Review and Launch
104. There is a warning box displayed regarding adding an open port 22 to the security group. Port 22 is not needed in this lab. This message box can safely be closed. Choose the Continue button.
105. Review the settings, then choose the Launch button.
106. On the **Select an existing key pair or create a new key pair** window, configure the following:
    - Select **Proceed without a key pair**
    - Select **I acknowledge that without a key pair...**
    - Choose Launch Instances

107. On the **Launch Status** page, choose View Instances

     This brings you to the **Instances** window where you can watch your **Private Instance** launch and view its details.

108. Wait for your private instance to fully launch. It should display the following:

- **Instance State:** Running

  You can choose the refresh icon to refresh your instances status.

  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.

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

110. Select **Private Instance** and then choose **Connect**

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

111. For **Connection method**, Select the **Session Manager** tab.

112. Choose the **Connect** button.

113. 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 you have allowed a few minutes time for the instance to launch, pass health checks, and communicate with the Sessions Manager

service before trying to open a session connection once again.

114. Enter the following commands to browse to the home directory (/home/ssm-user/) and test web connectivity using the *cURL* command: :

```
cd ~

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

Sample Output:

```
HTTP/2 200
content-type: text/html;charset=UTF-8
server: Server
date: Fri, 14 May 2021 14:30:15 GMT
x-amz-rid: 3WNTZRMBGMP8AP6HT4K7
set-cookie: aws-
priv=eyJ2IjoxLCJldSI6MCwic3QiOjB9; Version=1;
Comment="Anonymous cookie for privacy
regulations"; Domain=.aws.amazon.com; Max-
Age=31536000; Expires=Sat, 14-May-2022 14:30:15
GMT; Path=/
set-cookie: aws_lang=en; Domain=.amazon.com;
Path=/
x-frame-options: SAMEORIGIN
x-content-type-options: nosniff
x-amz-id-1: 3WNTZRMBGMP8AP6HT4K7
last-modified: Tue, 11 May 2021 17:39:32 GMT
vary: accept-encoding,Content-Type,Accept-
Encoding,X-Amzn-CDN-Cache,X-Amzn-AX-
Treatment,User-Agent
```

```
x-cache: Miss from cloudfront
via: 1.1
86561b4243b7d0478ca4582dd013e00e.cloudfront.net
(CloudFront)
x-amz-cf-pop: ATL52-C1
x-amz-cf-id: 3VxbxlST1HIeEd4sdFtNepfu7jnEjQB-
4gqet8mmAL5mbU0sQolzJw==
```

You have successfully connected to private instance using Session Manager. You can safely close the tab and return to the console.

Optional Task 1: Test connectivity to the private instance from 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 may complete this task or skip to the end of the lab here.

115. Go to the tab with the AWS Management Console open.
116. In the left navigation pane, choose **Instances**.
117. Select **Private Instance**.
118. On the **Details** tab, copy the Private IPv4 address to your clipboard.

     Tip To copy the *Private IPv4 addresses* , hover on it and choose the copy icon.
119. Unselect **Private Instance**.

120. Select **Public Instance**.
121. Choose the **Connect** button.

    The **Connect to instance** page is displayed.
122. Select the **Session Manager** tab.
123. Choose the **Connect** button.

    A new browser tab or window opens with a connection to the **Public Instance**.
124. Copy the following command to your notepad. Replace **<private_ip>** with the value of the the **Private IPv4 addresses**:

    ```
    ping <private_ip>
    ```

125. Copy and paste the updated command in your terminal and press Enter.

    Sample command: Do not use the following command.

    ```
    ping 10.0.2.131
    ```

126. After a few seconds, stop the ICMP ping request by pressing CTRL+C.

    The ping request to the private instance fails. Your challenge is to use the console and figure out the right *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: Retrieve 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 may complete this task or skip to the end of the lab here.

127. Go to the tab with the AWS Management Console open.
128. In the left navigation pane, choose **Instances**.
129. Select **Public Instance**.
130. Choose the **Connect** button.

    The **Connect to instance** page is displayed.
131. Select the **Session Manager** tab.
132. Choose the **Connect** button.

    A new browser tab or window opens with a connection to the **Public Instance**.
133. To view all categories of instance metadata from within a running instance, run the following command.

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

134. 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
    ```

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

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.

In this lab you learned how to:

- 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 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

End Lab

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

135. Return to the AWS Management Console.
136. On the navigation bar, choose **awsstudent@<AccountNumber>**, and then choose **Sign Out**.

137. Choose End Lab
138. Choose OK
139. (Optional):
    - Select the applicable number of stars
    - Type a comment
    - Choose **Submit**
    - 1 star = Very dissatisfied
    - 2 stars = Dissatisfied
    - 3 stars = Neutral
    - 4 stars = Satisfied
    - 5 stars = Very satisfied

    You may close the window if you don't want to provide feedback.

Optional Task Solution

140. Go to the tab with the AWS Management Console open.
141. On the Services menu, choose **EC2**.
142. In the left navigation pane, choose **Security Groups**.
143. Select Private SG.
144. Choose Actions then select **Edit inbound rules**.
145. On the **Edit inbound rules** page, in the *Inbound rules*.
    - Choose the Add rule button.
    - **Type:** *Custom ICMP - IPV4*
    - **Source:**
    - In the box to the right of Custom, type sg.
    - Select *Public SG* from the list that appears.

146. Choose the Save rules button.
147. Select the link here to go to **Optional Task** and re-run the steps. The *Public Instance* should now be able to successfully ping *Private Instance*.

Additional Resources

- VPC Introduction
- Subnets
- Internet Gateways
- Route Tables
- Security Groups for Your VPC
- NAT gateways
- Public IPv4 addresses and external DNS hostnames
- Understanding the basics of IPv6 networking on AWS

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*.