

Building Your First Amazon Virtual Private Cloud (VPC)

SPL-13 - Version 4.2.29

© 2024 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

In this lab, you create a basic Amazon Virtual Private Cloud (Amazon VPC) without using the VPC Wizard. 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 range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications.

The VPC that you build includes a web server and an Amazon RDS database. Once you have created both, you connect your address book application running on your web server to your Amazon RDS for MySQL instance. Once you successfully configure your address book application with your RDS instance, you are able to add and remove contacts from the address book.

TOPICS COVERED

In this lab you manually:

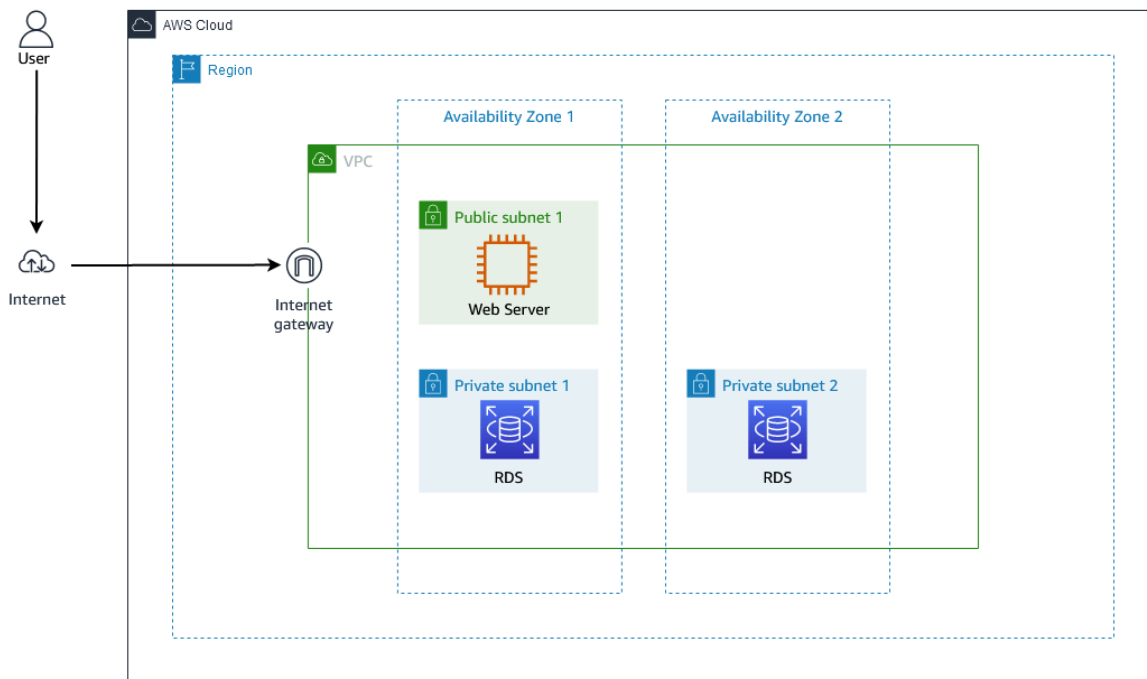
- Create an Amazon Virtual Private Cloud (VPC)
- Create a public and private subnets
- Create an Internet gateway
- Create a Route Table and add a route to the Internet
- Create a security group for your web server to only allow HTTP traffic to your web server
- Create a security group for your MySQL RDS instance to only allow MySQL traffic from your public subnet
- Deploy a web server and a MySQL RDS instance
- Configure your application to connect to your MySQL RDS instance

ARCHITECTURE OVERVIEW

You create a VPC with three subnets. A public and two private subnets. The web server is hosted in the public subnet, so that it can reach the public Internet. The MySQL RDS (*Database*) instance is hosted in the private subnets. In order to use a DB instance (*in this case the MySQL RDS*) in a VPC, your VPC must have at least two subnets. These subnets must be in two different Availability Zones in the AWS Region where you want to deploy your DB instance.

You act as the *user* in the image to test the application running in your new VPC. The diagram below shows a few general items you create. There are other VPC items not shown, for example, route tables and security groups.

Amazon Virtual Private 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 range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications.



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.

Task 1: Create a VPC

3. In this task, you create a base VPC.
4. Remember, a virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC. You can configure your VPC by modifying its IP address range, create subnets, and configure route tables, network gateways, and security settings.
5. In the **AWS Management Console** search field, type



6. Select **VPC** from the drop down menu

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

7. In the left navigation pane, choose **Your VPCs**.
8. Choose **Create VPC** then configure:
9. Choose **VPC Only**.

- **Name tag:**
- **IPv4 subnet CIDR block:**
- Choose **Create VPC**

Task 2: Create Your Public Subnet

In this task, you create a public subnet. Later, you launch your web server in the public subnet.

A subnet is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet. Use a public subnet for resources that must be connected to the internet, and a private subnet for resources that won't be connected to the internet.

CREATE YOUR PUBLIC SUBNET

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

11. Choose **Create subnet** then configure:

- **VPC ID:** *My VPC*
- **Subnet name:**
- **Availability Zone:** Select the *first* AZ in the list
- **IPv4 subnet CIDR block:**

12. Choose **Create subnet**

13. Select **Public 1**.

14. In the **Actions** menu, select **Edit subnet settings**, then configure:

- Select **Enable auto-assign public IPv4 address**
- Choose **Save**

Enable auto-assign public IPv4 address provides a public IPv4 address for all instances launched into the selected subnet.

Even though your subnet is labeled **Public 1**, it is not yet a public subnet. A public subnet must have an Internet Gateway, which you attach in the next task.

Task 3: Create an Internet Gateway

In this task, you create an Internet gateway so that traffic can access your web server.

An Internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the Internet. It therefore imposes no availability risks or bandwidth constraints on your network traffic.

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.

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

16. Choose **Create internet gateway** then configure:

- **Name tag:**

- Choose **Create internet gateway**

17. In the **Actions** menu, select **Attach to VPC**, then configure:

- **Available VPCs:** *My VPC*

- Choose **Attach internet gateway**

This attaches the Internet gateway to your VPC. Even though you created an Internet gateway and attached it to your VPC, you still have to tell instances within your public subnet how to get to the Internet.

Task 4: Create a Route Table, Add Routes, And Associate Public Subnets

In this task, you:

- Create a route table for internet-bound traffic
- Add a route to the route table to direct Internet-bound traffic to your Internet gateway
- Associate your public subnet with your route table

A route table contains a set of rules, called routes, that are used to determine where network traffic is directed. 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.

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; for example, the public IPv4 addresses of your company's public endpoints outside of AWS, or the Elastic IP addresses of other Amazon EC2 instances outside your VPC. If your subnet is associated with a route table that has a route to an Internet gateway, it's known as a public subnet.

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

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

19. Choose **Create route table**

20. Under **Route table settings** section then configure:

- **Name - optional:**

- **VPC:** *My VPC*

- Choose **Create route table**

21. Within 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. Add a new route to enable public traffic.

22. Choose **Edit routes**

23. Choose **Add route** then configure:

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

24. Choose the **Subnet associations** tab.

25. Under section **Edit subnet associations**, choose **Edit subnet associations**

26. Select **Public 1**.

27. Choose **Save associations**

The subnet is now *public* because it is connected to the Internet via the Internet Gateway.

Task 5: Create a Security Group for your Web Server

In this task, add a security group so that users can access your web server via HTTP.

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. When you launch an instance in a VPC, you can assign up to five security groups to the instance. Security groups act at the instance level, not the subnet level. Therefore, each instance in a subnet in your VPC could be assigned to a different set of security groups. If you do not specify a particular group at launch time, the instance is automatically assigned to the default security group for the VPC.

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

29. Choose **Create security group** then configure:

- **Security group name:**
- **Description:**
- **VPC:** *My VPC*

30. Under **Inbound rules**

- Choose **Add rule**
- **Type:** HTTP
- **Source:** *Anywhere-Ipv4*

31. At the bottom of the screen, choose **Create security group**

Task 6: Launch a Web Server in your Public Subnet

In this task, you launch a web server that runs an address book application. Later in the lab, you connect the address book application to a Amazon RDS for MySQL instance.

32. In the **AWS Management Console** search field, type

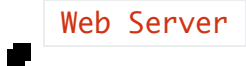

EC2

33. Select **EC2** from the drop down menu

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.

34. Choose **Launch instances** > **Launch instances**.

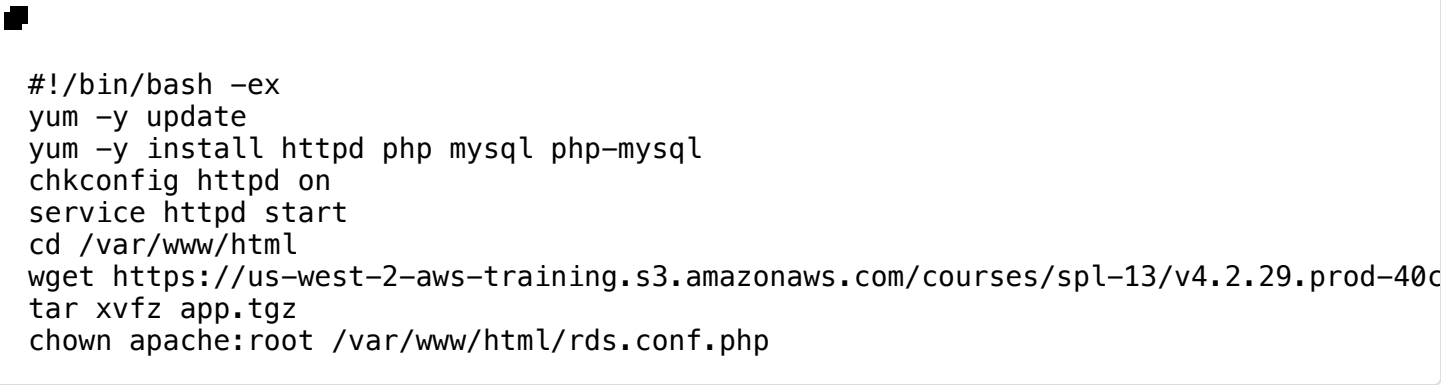
35. In the Launch an instance page, configure:

- **Name and tags** section, type:  **Web Server**
- **Application and OS Images (Amazon Machine Image)** Choose **Amazon Linux 2 AMI**
- **Instance type**, choose **t3.micro**
- **Key pair (login)** section, choose **Proceed without a key pair**
- **Network settings** section, choose  **Edit**

- **VPC**, choose  **My VPC**

Note -  **Public 1** populates under the subnet section

- Under **Firewall (security groups)**, choose **Select an existing security group**
- **Common security groups**, choose **Web server**
- Expand **Advanced Details** (at the bottom of the page)
- Copy and paste this script into the **User data** text box:



```
#!/bin/bash -ex
yum -y update
yum -y install httpd php mysql php-mysql
chkconfig httpd on
service httpd start
cd /var/www/html
wget https://us-west-2-aws-training.s3.amazonaws.com/courses/spl-13/v4.2.29.prod-40c
tar xvfz app.tgz
chown apache:root /var/www/html/rds.conf.php
```

This script is run the first time the instance is launched. It installs a web server on your EC2 instance, and runs an app that can be configured to point to your MySQL RDS instance. After you configure your RDS instance, it presents an address book that you can edit.

36. Choose **Launch instance**

37. Choose **View all instances**

This brings you to the **Instances** window where you can see your web server details.

38. Wait for your web server to fully launch. It should display the following:

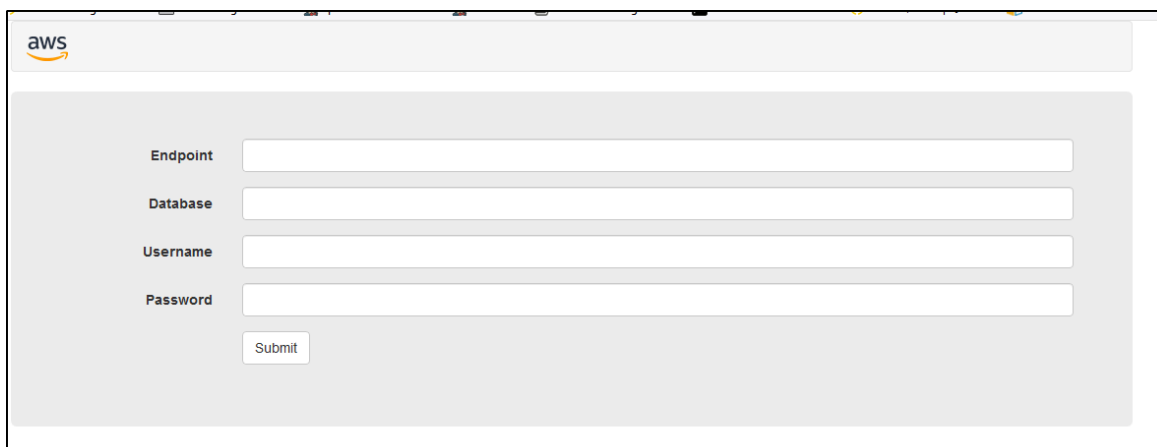
- **Instance State:** running
- **Status check:** 2/2 checks passed

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

39. Your instance should be selected if not, select it.
40. Copy the **Public IPv4 address** address of the instance to your clipboard.
41. Open a new web browser tab and paste the IP address into the browser.
42. Press **Enter** to go the web page.

If you receive an error, please wait 60 seconds and refresh the page to try again. It can take a couple of minutes for the EC2 instance to boot and run the script that installs software.

An application should appear:

A screenshot of a web application interface. At the top left is the AWS logo. Below it, there are four input fields labeled 'Endpoint', 'Database', 'Username', and 'Password'. Below these fields is a 'Submit' button. The interface is simple and clean, with a light gray background.

Congratulations! You should be able to see this page. Currently, you do not have a database. Once you create your RDS instance, you connect it to your web server.

Task 7: Create Private Subnets for your MySQL Server

To deploy your RDS database, your VPC must have at least two subnets. These subnets must be in two different Availability Zones in the AWS Region where you want to deploy your DB instance. In this task, you create two private subnets for your Amazon RDS instance.

CREATE YOUR FIRST PRIVATE SUBNET

43. In the **AWS Management Console** search field, type



44. Select **VPC** from the drop down menu

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

46. Choose **Create subnet** then configure:

- **VPC:** *My VPC*
- **Subnet name:**
- **Availability Zone:** Select the *first* AZ in the list
- **IPv4 subnet CIDR block:**
- Choose **Create subnet**

CREATE YOUR SECOND PRIVATE SUBNET

47. Choose **Create subnet** then configure:

- **VPC:** *My VPC*
- **Subnet name:**
- **Availability Zone:** Select the *second* AZ in the list
- **IPv4 subnet CIDR block:**
- Choose **Create subnet**

Task 8: Create a Security Group for your Database Server

Now that your private subnets are configured, you secure the types of traffic that can access your MySQL database. In this task, you create a security group to only allow MySQL traffic from your Web server.

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

49. Copy the **Security group ID** value of your *Web server* security group and paste it into your text editor.

50. Choose **Create security group** then configure:

- **Security group name:**
- **Description:**
- **VPC:** *My VPC*

51. Under **Inbound rules**

- Choose
- **Type:** MySQL/Aurora
Note: Select MySQL, *not* MSSQL,

- **Source:**
 - *Custom*
 - Paste the web server security group ID that you copied to your text editor

52. At the bottom of the screen, choose **Create security group**

This allows your web server to communicate with the database.

Task 9: Create a Database Subnet Group

Amazon RDS instances require a database subnet group. In this task, you create a database subnet group.

A DB subnet group is a collection of subnets (typically private) that you create in a VPC and that you then designate for your DB instances. Each DB subnet group should have subnets in at least two Availability Zones in a given region. When creating a DB instance in a VPC, you must select a DB subnet group.

53. In the **AWS Management Console** search field, type



54. Select **RDS** from the drop down menu

55. In the left navigation pane, choose **Subnet groups**.

56. Choose **Create DB Subnet Group** then configure:

- **Name:**
- **Description:**
- **VPC:** *My VPC*

ADD YOUR PRIVATE SUBNETS

57. In the **Add subnets** section, configure the following:

- **Availability zones:** Select the first and second Availability Zones in the list.

58. In the **Subnets** section, select:

- *10.0.2.0/24*
- *10.0.3.0/24*

59. At the bottom of the screen, choose **Create**

Task 10: Create an Amazon RDS Database

You are now ready to launch an Amazon RDS database running MySQL.

60. In the left navigation pane, choose **Databases**.

61. Choose **Create database** then configure:

- **Engine options:** *MySQL*
- **Version:** *MySQL 5.7.X*

It is very important to select latest version of 5.7.X. Select the version with highest number. This lab requires it for the application.

62. In the *Templates* section, select **Dev/Test**.

63. In the **Settings** section, configure:

- **DB instance identifier:**
- **Master username:**
- For **Credentials management**, choose **Self managed** option
- **Master password:**
- **Confirm password:**

64. In the **DB instance class** section, configure:

- **DB instance class:** *Burstable classes*
- Select **db.t3.micro**

65. In the **Storage** section, expand **Storage autoscaling** and then de-select **Enable storage autoscaling**

66. In the **Connectivity** section, configure:

- **Virtual Private Cloud (VPC):** *My VPC*
- **Public access:** *No*
- **Existing VPC security groups:**
 - Add the **Database** security group
 - Remove the **default** security group

67. In the **Monitoring** section, de-select **Enable Enhanced monitoring**

68. In the **Additional configuration** section (*located near the bottom*), choose **Additional configuration**, then configure:

- **Initial database name:**
- De-select **Enable automated backups** This turn off backups, which launches the database a little bit quicker for your lab.
- De-select **Enable auto minor version upgrade**

69. At the bottom of the screen, choose **Create database**

70. If you receive a pop up that says **Suggested add-ons for mydb**, choose **Close**

71. Choose refresh every 60 seconds until the instance has a status of **available**.

Congratulations! You have deployed a MySQL database.

Task 11: Connect Your Address Book Application to Your Database

In this task, you connect the address book application (in your Public subnet) to your database (in your Private subnet).

OBTAIN YOUR MYSQL DATABASE ENDPOINT

Before you can connect your address book application to your database, you need to know the *endpoint* of the RDS instance. This is the address of your RDS instance.

72. Choose your **mydb** instance.

73. In the **Connectivity & security** section, copy the **Endpoint** to your clipboard.

Your RDS endpoint should look similar to:

mydb.ciljcs3yv1rb.us-west-2.rds.amazonaws.com

CONNECT TO YOUR DATABASE

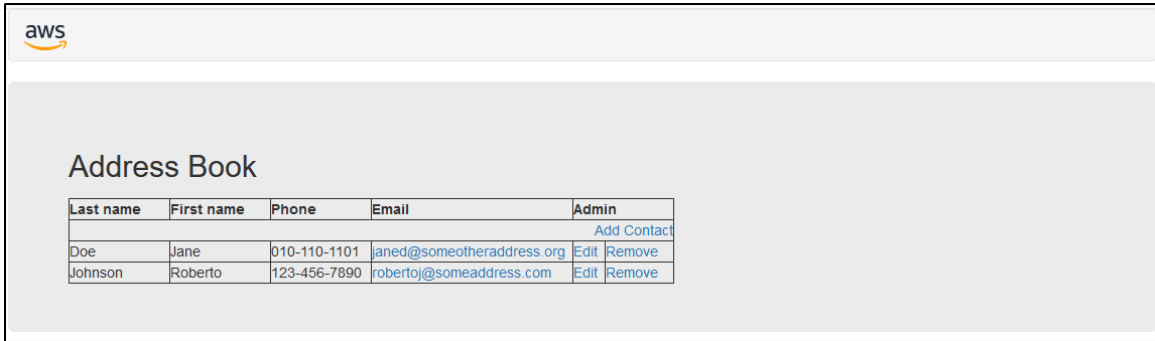
74. Return to the browser tab that is displaying your web server, then configure:

- **Endpoint:** Paste your MySQL endpoint
- **Database:**
- **Username:**
- **Password:**
- Choose

Once connected, you should see an address book with two entries.

Congratulations! You have successfully connected your address book application to your database.

75. Try adding and then removing a contact from the address book.



The address book information is saved in the Amazon RDS for MySQL database.

Conclusion

Congratulations! You have now successfully:

- Created an Amazon Virtual Private Cloud (VPC)
- Created a public and private subnets
- Created an Internet gateway
- Created a Route Table and added a route to the Internet
- Created a security group for your web server to only allow HTTP traffic to your web server
- Created a security group for your MySQL RDS instance to only allow MySQL traffic from your public subnet
- Deployed a web server and a MySQL RDS instance
- Configured your application to connect to your MySQL RDS instance

End lab

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

76. Return to the **AWS Management Console**.
77. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
78. Choose **End lab** and then confirm that you want to end your lab.

Additional Resources

- [VPC Introduction](#)
- [Route Tables](#)
- [Security Groups for Your VPC](#)
- [Internet Gateways](#)
- [Availability Zones](#)

- [Amazon RDS](#)