**Building Your First Amazon Virtual Private Cloud (VPC) | Qwiklabs**

**SPL-13 - Version 4.2.12**

© 2020 Amazon Web Services, Inc. and 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.

Errors, corrections or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Lab overview**

In this lab, you will create a basic virtual private cloud (VPC) without using the VPC Wizard. The VPC that you build will include a web server and an Amazon RDS database. Once you have created both, you will connect your address book application running on your web server to your Amazon RDS for MySQL instance. Once you have successfully configured your address book application with your RDS instance, you will be able to add and remove contacts from the address book.

**Topics covered**

In this lab you will manually:

* Create an Amazon Virtual Private Cloud (VPC)
* Create a public and private subnets
* Create an Internet gateway
* Create a Route Table and added 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 subnets
* Deploy a web server and a MySQL RDS instance
* Configure your application to connect to your MySQL RDS instance

**Amazon Virtual Private Cloud (VPC)**

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.

**Task 1: Create a VPC**

In this task, you will create a base VPC.

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.

1. In the **AWS Management Console**, on the Services menu, click **VPC**.

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.

1. In the left navigation pane, click **Your VPCs**.
2. Click Create VPC then configure:

* **Name tag:**
* **IPv4 CIDR block:**
* Click Create VPC

**Task 2: Create Your Public Subnets**

In this task, you will create two public subnets. Each subnet will reside in a separate availability group. Later in the lab, you will launch your web server into one of the public subnets.

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 First Public Subnet**

1. In the left navigation pane, click **Subnets**.
2. Click Create subnet then configure:

* **Name tag:**
* **VPC:** *My VPC*
* **Availability Zone:** Select the *first* AZ in the list
* **IPv4 CIDR block:**
* Click Create

1. Click Close
2. Select **Public 1**.
3. In the **Actions** menu, select **Modify auto-assign IP settings**, then configure:

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

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

**Create Your Second Public Subnet**

1. Click Create subnet then configure:

* **Name tag:**
* **VPC:** *My VPC*
* **Availability Zone:** Select the *second* AZ in the list
* **IPv4 CIDR block:**
* Click Create

1. Click Close
2. De-select **Public 1**.
3. Select **Public 2**.
4. In the **Actions** menu, select **Modify auto-assign IP settings**, then configure:

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

Even though your subnets are labeled **Public 1** and **Public 2**, they not yet public subnets. A public subnet must have an Internet Gateway, which you will attach in the next task.

**Task 3: Create an Internet Gateway**

In this task, you will 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.

1. In the left navigation pane, click **Internet Gateways**.
2. Click Create internet gateway then configure:

* **Name tag:**
* Click Create internet gateway

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

* **Available VPCs:** *My VPC*
* Click Attach internet gateway

This will attach 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 will:

* 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 subnets 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.

1. In the left navigation pane, click **Route Tables**.

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

1. Click Create route table then configure:

* **Name tag:**
* **VPC:** *My VPC*
* Click Create
* Click Close

1. Select **Public Route Table**.
2. Click 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 will now add a new route to enable public traffic.

1. Click Edit routes
2. Click Add route then configure:

* **Destination:**
* **Target:** Select **Internet Gateway** in the drop down and then select the displayed **Internet Gateway** id
* Click Save routes
* Click Close

1. Click the **Subnet Associations** tab.
2. Click Edit subnet associations
3. Select **Public 1** and **Public 2**.
4. Click Save

The two subnets are now *public* because they connect to the Internet via the Internet Gateway.

**Task 5: Create a Security Group for your Web Server**

In this task, you will 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.

1. In the left navigation pane, click **Security Groups**.
2. Click Create security group then configure:

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

1. Under **Inbound rules**

* Click Add rule
* **Type:** HTTP
* **Source:** *Anywhere*
* Click Create security group

In the next task you will launch your web server into one of your public subnets.

**Task 6: Launch a Web Server in your Public Subnet**

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

1. On the Services menu, click **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.

1. Click Launch instance > **Launch instance**.
2. On **Step 1**, click Select next to **Amazon Linux AMI**.

Do not select **Amazon Linux 2 AMI**.

You will launch a t2.micro instance. This instance type has 1 vCPU and 1 GiB of memory.

1. On **Step 2**, click Next: Configure Instance Details
2. On **Step 3**, configure:

* **Network:** *My VPC*
* 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://s3-us-west-2.amazonaws.com/us-west-2-aws-training/awsu-spl/spl-13/scripts/app.tgz

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 will present an address book that you can edit.

1. Click Next: Add Storage
2. On **Step 4**, click Next: Add Tags
3. On **Step 5**, click Add Tag then configure:

* **Key:**
* **Value:**

1. Click Next: Configure Security Group
2. On **6. Configure Security Group**, configure the following:

* Click **Select an existing security group**
* Select **Web server**
* Click Review and Launch

1. At the **Warning** screen, click Continue
2. On **Step 7**, review the settings, then click Launch
3. 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...**
* Click Launch Instances

1. Click View Instances

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

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

* **Instance State:** running

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

1. Your instance should be selected if not, select it.
2. Copy the **Public IPv4 address** address of the instance to your clipboard.
3. Open a new web browser tab and paste the IP address into the browser.
4. 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:



Congratulations! You should be able to see this page. Currently, you do not have a database. Once you create your RDS instance, you will be able to 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 one subnet in at least two Availability Zones in the region where you want to deploy your DB instance. In this task, you will create two private subnets for your Amazon RDS instance.

**Create Your First Private Subnet**

1. In the **AWS Management Console**, on the Services menu, click **VPC**.
2. In the left navigation pane, click **Subnets**.
3. Click Create subnet then configure:

* **Name tag:**
* **VPC:** *My VPC*
* **Availability Zone:** Select the *first* AZ in the list
* **IPv4 CIDR block:**
* Click Create

1. Click Close

**Create Your Second Private Subnet**

1. Click Create subnet then configure:

* **Name tag:**
* **VPC:** *My VPC*
* **Availability Zone:** Select the second AZ in the list
* **IPv4 CIDR block:**
* Click Create

1. Click Close

**Task 8: Create a Security Group for your Database Server**

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

1. In the left navigation pane, click **Security Groups**.
2. Copy the **Security group ID** value of your *Web server* security group and paste it into your text editor.

Next you will create a security group that will allow your *Web server* to communicate with your database.

1. Click Create security group then configure:

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

1. Under **Inbound rules**

* Click Add rule
* **Type:** MySQL/Aurora   
  Note: Select MySQL, *not* MS SQL,
* **Source:**
  + *Custom*
  + Paste the web server security group ID that you copied to your text editor
* Click Create security group

This will allow 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 will 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.

1. On the Services menu, click **RDS**.
2. In the left navigation pane, click **Subnet groups**.
3. Click Create DB Subnet Group then configure:

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

**Add Your First Private Subnet**

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

* **Availability zone:** Select the *first* AZ in the list
* **Subnet:** Select your first **private** subnet (10.0.3.0/24)

**Add Your Second Private Subnet**

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

* **Availability zone:** Select the *second* AZ in the list
* **Subnet:** Select your second **private** subnet (10.0.4.0/24)

1. At the bottom of the screen, click Create

**Task 10: Create an Amazon RDS Database**

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

1. In the left navigation pane, click **Databases**.
2. Click Create database then configure:

* **Engine options:** *MySQL*
* **Templates:** *Dev/Test*

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

* **DB instance identifier:**
* **Master username:**
* **Master password:**
* **Confirm password:**

1. In the **DB instance size** section, configure:

* **DB instance class:** *Burstable classes*
* *db.t2.micro*

1. In the **Storage** section, de-select **Enable storage autoscaling**
2. In the **Connectivity** section, configure:

* **Virtual Private Cloud (VPC)** *My VPC*
* Expand **Additional connectivity configuration**
* **Publicly access:** *No*
* **Existing VPC security groups:**
  + Add the **Database** security group
  + Remove the **default** security group

1. Expand **Additional configuration**, then configure:

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

1. Scroll to the bottom of the page, then click Create database
2. Click 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 will 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.

1. Click your **mydb** instance.
2. In the **Connectivity & security** section, copy the **Endpoint** to your clipboard.

You RDS endpoint should look similar to:

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

**Connect to Your Database**

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

* **Endpoint:** Paste your MySQL endpoint
* **Database:**
* **Username:**
* **Password:**
* Click Submit

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

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

1. 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 subnets
* 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, end your lab, and evaluate the experience.

1. Return to the AWS Management Console.
2. On the navigation bar, click **awsstudent@<AccountNumber>**, and then click **Sign Out**.
3. Click End Lab
4. Click OK
5. (Optional):

* Select the applicable number of stars
* Type a comment
* Click **Submit**
  + 1 star = Very dissatisfied
  + 2 stars = Dissatisfied
  + 3 stars = Neutral
  + 4 stars = Satisfied
  + 5 stars = Very satisfied

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

**Additional Resources**

* [VPC Introduction](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Introduction.html)
* [Route Tables](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Route_Tables.html)
* [Security Groups for Your VPC](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html)
* [Internet Gateways](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Internet_Gateway.html)

For more information about AWS Training and Certification, see [*http://aws.amazon.com/training/*](http://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*](https://support.aws.amazon.com/#/contacts/aws-training).