# Lab 1: Build Your VPC and Launch a Web Server

In this lab session, you use Amazon Virtual Private Cloud (VPC) to create your own VPC and add additional components to it to produce a customized network. You will create security groups for your EC2 instance. You configure and customize the EC2 instance to run a web server and launch it into the VPC.

**Amazon Virtual Private Cloud (Amazon VPC)** enables you to launch Amazon Web Services (AWS) resources into a virtual network that you defined. This virtual network closely resembles a traditional network that you would operate in your own data center, with the benefits of using the scalable infrastructure of AWS. You can create a VPC that spans multiple Availability Zones. A *security group* acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances.

An **Internet gateway (IGW)** is a VPC component that allows communication between instances in your VPC and the Internet. A *route table* contains a set of rules, called *routes*, that are used to determine where network traffic is directed. Each subnet in a VPC must be associated with a route table; the route table controls routing for the subnet.

After creating a VPC, you can add one or more subnets in each Availability Zone. Each subnet resides entirely within one Availability Zone and cannot span zones. If a subnet's traffic is routed to an Internet gateway, the subnet is known as a *public subnet*. If a subnet does not have a route to the Internet gateway, the subnet is known as a *private subnet*.

**Objectives**

After completing this lab, you can:

- Create a VPC.
- Create subnets.
- Configure a security group.
- Launch an EC2 instance into a VPC.

**Duration**

This lab takes approximately **45 minutes** to complete.

---

## Access the AWS Management Console

---

## Task 1: Create Your VPC

In this task, you create a VPC with two subnets in one Availability Zone.

1. [1] In the **AWS Management Console**, on the **Services** menu, click **VPC**.
2. Click **Start VPC Wizard**.

3. In the navigation pane, click **VPC with Public and Private Subnets**.
4. Click **Select**.
5. Configure the following settings (and ignore any settings that aren't listed):
    - **IPv4 CIDR block**: Type **10.0.0.0/16**
    - **VPC name**: type `My Lab VPC`
    - **Public subnet's IPv4 CIDR**: Type `10.0.1.0/24`
      You can safely ignore the error:
      *"Public and private subnet CIDR blocks overlap."*
      You will fix this when you change the value below.
    - **Availability Zone**: Click the *first* Availability Zone.
    - **Public subnet name**: type `Public Subnet 1`
    - **Private subnet's IPv4 CIDR**: Type `10.0.3.0/24`
    - **Availability Zone**: Click the *first* Availability Zone.
      The same as used for Public Subnet 1
    - **Private subnet name**: type `Private Subnet 1`
    - **Specify the details of your NAT gateway**: Click **Use a NAT instance instead**.
      On the far right of the screen - you may need to scroll.
    - **Key pair name**: Click the **Qwiklabs** key pair.
6. Click **Create VPC**.
7. In the success message, click **OK**.

---

## Task 2: Create Additional Subnets

In this task, you create two additional subnets in another Availability Zone and associate the subnets with existing route tables.

1. [6] In the navigation pane, click **Subnets**.
2. Click **Create Subnet**.
3. In the **Create Subnet** dialog box, configure the following settings (and ignore any settings that aren't listed):
    - **Name tag**: type `Public Subnet 2`
    - **VPC**: Click **My Lab VPC**.
    - **Availability Zone**: Click the *second* Availability Zone
    - **IPv4 CIDR block**: Type `10.0.2.0/24`
4. Click **Yes, Create**.
5. Click **Create Subnet**.
6. In the **Create Subnet** dialog box, configure the following settings (and ignore any settings that aren't listed):
    - **Name tag**: type `Private Subnet 2`
    - **VPC**: Click **My Lab VPC**.
    - **Availability Zone**: Select the *second* Availability Zone.
      The same as used for Public Subnet 2
    - **CIDR block**: Type `10.0.4.0/24`

7. Click **Yes, Create**.
8. In the navigation pane, click **Route Tables**.
9. Select the route table with the VPC **My Lab VPC** and **Yes** under **Main**.
10. Double-click the empty **Name** for this route table, type `Private Route Table`, and click the checkmark to save.
11. In the lower pane, click **Routes** and note that **Destination 0.0.0.0/0** is set to **Target eni-xxxxxxxx / i-xxxxxxxx**. This route table is used to route traffic from private subnets to the NAT instance, as identified by an Elastic Network Interface (ENI) and Instance ID.
12. Click **Subnet Associations**, and then click **Edit**.
13. Select **Private Subnet 1** and **Private Subnet 2**.
14. Click **Save**.
15. Select the route table with the VPC **My Lab VPC** and **No** under **Main**.
16. Double-click the empty **Name** for this route table, type `Public Route Table`, and click the checkmark to save.
17. In the lower pane, click **Routes** and note that **Destination 0.0.0.0/0** is set to **Target igw-xxxxxxxx**. This route table is used by public subnets for communication.
18. Click **Subnet Associations**, and then click **Edit**.
19. Select **Public Subnet 1** and **Public Subnet 2**.
20. Click **Save**.

---

### Task 3: Create a VPC Security Group

In this task, you create a VPC security group that permits access for web traffic.

1. [33] In the navigation pane, click **Security Groups**.
2. Click **Create Security Group**.
3. In the **Create Security Group** dialog box, configure the following settings (and ignore any settings that aren't listed):
    - **Name tag**: type `WebSecurityGroup`
      You can ignore the message:
      *"A security group description is required."*
    - **Group name**: Click **WebSecurityGroup**.
      This will be entered automatically
    - **Description**: type `Enable HTTP access`
    - **VPC**: Click **My Lab VPC**.
      This is the VPC you created in Task 1
4. Click **Yes, Create**.
5. Select **WebSecurityGroup**.
6. Click the **Inbound Rules** tab.
7. Click **Edit**.
8. For **Type**, click `HTTP (80)`.
9. Click in the **Source** box and type `0.0.0.0/0`
10. Click **Save**.

## Task 4: Launch Your First Web Server Instance

In this task, you launch an EC2 instance into the VPC you created and bootstrap the instance to act as a web server.

1. [43] On the **Services** menu, click **EC2**.
2. Click **Launch Instance**.
3. In the row for **Amazon Linux AMI**, click **Select**. If you receive a warning, click **Continue**.
4. On the **Step 2: Choose an Instance Type** page, confirm that **t2.micro** is selected and then click **Next: Configure Instance Details**.
5. On the **Step 3: Configure Instance Details** page, configure the following settings (and ignore any settings that aren't listed):
   - **Network**: Click **My Lab VPC**.
     This is the VPC you created in Task 1
   - **Subnet**: Click the **Public Subnet 2 (10.0.2.0/24)**.
     This is the subnet you created in Task 2
   - **Auto-assign Public IP**: Click **Enable**.
     You can safely ignore the message:
     *"You do not have permissions to list any IAM roles."*
6. Expand the **Advanced Details** section.
7. Click **Copy Code Block** below, and paste it into the **User data** box.

```
#!/bin/bash -ex
yum -y update
yum -y install httpd php mysql php-mysql
chkconfig httpd on
/etc/init.d/httpd start
if [ ! -f /var/www/html/lab2-app.tar.gz ]; then
cd /var/www/html
wget https://us-west-2-aws-training.s3.amazonaws.com/awsu-ilt/AWS-100-ESS/v4.2/lab-
tar xvfz lab2-app.tar.gz
chown apache:root /var/www/html/rds.conf.php
fi
```

   The user data transforms the Linux instance into a PHP web application.

8. Click **Next: Add Storage**.
9. Click **Next: Add Tags**.
10. Click **Add Tag**, and configure the following settings (and ignore any settings that aren't listed):
    - **Key**: type `Name`
    - **Value**: type `Web Server 1`
11. Click **Next: Configure Security Group**.
12. On the **Step 6: Configure Security Group** page, click **Select an existing security group** and then select the security group you created in Task 3 (**WebSecurityGroup**).

13. Click **Review and Launch**. When prompted with a *warning* that you will not be able to connect to the instance through port 22, click **Continue**.
14. Review the instance information and click **Launch**. Ignore any warning that appears regarding a security group being open to the world. This is expected behavior.
15. Click **Choose an existing key pair**, click the **Qwiklabs** key pair, select the acknowledgment check box, and then click **Launch Instances**.
16. Scroll down and click **View Instances**. You will see two instances – **Web Server 1** and the NAT instance launched by the VPC Wizard.
17. Wait until **Web Server 1** shows *2/2 checks passed* in the **Status Checks** column. This will take 3 to 5 minutes. Click the refresh icon in the upper right pane to check for updates.
18. Select Web Server 1 and copy the **Public DNS** value on the **Description** tab.
19. Paste the **Public DNS** value in a new web browser window or tab and press **ENTER**.
    You will see a web page displaying the AWS logo and instance meta-data values.

---

## Lab Complete

Congratulations! You have successfully created a VPC and launched an EC2 instance into it. To clean up your lab environment, do the following:

1. Cleanup the resources.