aws training and certification

Networking - Configuring and Deploying Amazon VPC and a Web Server

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 basic publicly accessible web server in your AWS Cloud environment. The VPC will be built with subnets, route tables so learners will understand all the components of a VPC.

Let's get a quick overview of what a VPC is.

Amazon Virtual Private Cloud (Amazon VPC) lets you 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 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.

For more information about VPC you can view the AWS Documentation using the following link: https://docs.aws.amazon.com/vpc/latest/userguide/how-it-works.html

Topics Covered

- Create an Amazon Virtual Private Cloud (VPC)
- Create a public subnet
- 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
- Deploy a web server instance

Technical Knowledge Prerequisites

To successfully complete this lab, you should be familiar with basic navigation of the AWS Management Console.

Task 1: Create a 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. You can use multiple layers of security, including security groups and network access control lists, to help control access to Amazon EC2 instances in each subnet.

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

1. In the AWS Management Console, choose **US-EAST-1 (N. Virginia Region)**.
2. Choose 

and select **VPC**. (You may also type VPC in the search bar and choose 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.

3. Choose **Your VPCs** on the left navigation menu.

 **Note:** You will see a default VPC (one is created whenever an AWS account is created). To learn more about default VPC, go to Default VPC and default subnets

4. Choose **Create VPC** on the right side of the console.
5. In the Create VPC section, enter the following:

   - **Name tag**: Enter VPC-1
   - **IPv4 CIDR block**: Choose a CIDR range in 10.0.0.0 network with mask in such a way that it provides 256 number of IP addresses

 **Hint:** If you want to learn about subnetting, go to the **Appendix** section at the end of this document.

 **Note:** This VPC will not have an IPv6 CIDR block, and we will leave it with default tenancy.

6. Choose **Create VPC**

A VPC with the specified CIDR block has been created. Now, let's create the subnets.

Task 2: Create Your 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. To protect the AWS resources in each subnet, you can use multiple layers of security, including security groups and network access control lists (ACL).

In this task, you will create one public subnet in the Lab VPC. The public subnets will be for internet-facing resources.

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

Note: You will see subnets for the default VPC. You can ignore them and go to the next step.

8. Choose **Create subnet** and configure it with the following details:

- **VPC ID**: Select *VPC-1*
- **Subnet name**: Enter VPC-1 PublicSubnet
- **Availability Zone**: Select the first Availability Zone in the list
- **IPv4 CIDR block**: Choose a CIDR range in 10.0.0.0 network with mask in such a way that it provides 16 number of IP addresses

Hint: If you want to learn about subnetting, go to the **Appendix** section at the end of this document.

9. Choose **Create subnet**

10. Select **VPC-1 PublicSubnet**.

11. In the **Actions** menu, select **Edit subnet setting**, 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

Note: When you create a VPC, you must specify an IPv4 CIDR block for the VPC. The CIDR block of a subnet can be the same as the CIDR block for the VPC (for a single subnet in the VPC)

or a subset of the CIDR block for the VPC (for multiple subnets). If you create more than one subnet in a VPC, the CIDR blocks of the subnets cannot overlap. Even though we named the subnet *VPC-1 PublicSubnet*, they are not yet public. A public subnet must have an internet gateway, which you will create and attach in the lab.

Task 3: Create an Internet Gateway

An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in a VPC and the internet. An internet gateway does not impose availability risks or bandwidth constraints on network traffic.

An internet gateway serves two purposes:

- Provide a target in route tables to connect to the internet
- Perform network address translation (NAT) for instances that have been assigned public IPv4 addresses

In this task, you will create an internet gateway so that internet traffic can access the public subnets.

12. In the left navigation pane, choose **Internet Gateways**.

Note: A default internet gateway was created with the default VPC. You can ignore this and proceed with the next step.

13. Choose **Create internet gateway** and configure:

- **Name tag**: Enter VPC-1 InternetGateway

14. Choose **Create internet gateway**

Once it's created, you need to attach the internet gateway to your Lab VPC.

15. Choose **Actions** > **Attach to VPC**.

16. For **VPC**, select *VPC-1*

17. Choose **Attach internet gateway**

The internet gateway is now attached to your Lab VPC. Even though you created an Internet gateway and attached it to your VPC, you still have to tell instance within your public subnet how to get to the Internet.

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

A route table contains a set of rules, called *routes*, used to determine where network traffic is directed. Each subnet in a VPC must be associated with a route table, which 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 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.

To use an internet gateway, a subnet's route table must contain a route that directs internet-bound traffic to the internet gateway. This subnet is called a *public subnet*.

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

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

Several route tables are displayed, but there is only one route table associated with the Lab VPC. This is the main route table.

19. Choose Create route table and configure:

- **Name tag**: Enter VPC-1 PublicRouteTable
- **VPC**: Select *VPC-1*

20. Choose **Create route table**

21. Select **Routes** tab for *VPC-1 PublicRouteTable*.

22. Choose **Edit routes**

Now, add a route to direct internet-bound traffic (0.0.0.0/0) to the internet gateway.

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

- **Destination**: Enter 0.0.0.0/0
- **Target**: Select *Internet Gateway* and *VPC-1 InternetGateway*.

24. Choose Save changes

The last step is to associate this new route table with the public subnets.

25. Choose the **Subnet Associations** tab.
26. Choose Edit subnet associations
27. Select the rows with *VPC-1 PublicSubnet*.
28. Choose Save associations

The *VPC-1 PublicSubnet* is now public because it has a route table entry that sends traffic to the internet via the internet gateway.

Task 5: Create a Security Group for your Web Server

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.

In this task, you will create a security group for the EC2 instances so that users can access your web server via HTTP.

29. In the left navigation pane, choose **Security Groups**.
30. Create another new security group with the following details:

- **Security group name**: Enter VPC-1 SG
- **Description**: Enter Allows HTTP access
- **VPC**: Select *VPC-1*

31. For **Inbound rules**, choose Add rule and configure it with the following details:

- **Type**: Select *HTTP*
- **Source**: Select *Anywhere-IPv4*

32. Create a new tag with the following details:

- **Key**: Enter Name
- **Value**: Enter VPC-1 SG

33. Choose Create security group

You have configured the inbound rules to permit HTTP ports 80 traffic to the EC2 instance.

Note: You can grant access to a specific CIDR range, or to another security group in your VPC or in a peer VPC (requires a VPC peering connection). 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.

Task 6: Launch an Instance in your Public Subnet

In this task, you provision the EC2 instance in the public subnet with user data so that you can load a web page in your web browser.

34. 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.

35. Click **Launch instance** > **Launch instance**.

36. On **Step 1**, click **Select** next to **Amazon Linux 2 AMI**.

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

37. On **Step 2**, click **Next: Configure Instance Details**

38. On **Step 3**, configure:

- **Network***: VPC-1*
- **Subnet:** *VPC-1 PublicSubnet*
- Expand **Advanced Details** (at the bottom of the page)

- Copy and paste this script into the **User data** text box:

```
#!/bin/bash
yum update -y
yum install httpd -y
service httpd start
chkconfig on
cd /var/www/html
echo "<html><body><h1>Hello World, This is a WebServer<h1></body></html>" > index.html
```

This script is run the first time the instance is launched. It installs a web server on your EC2 instance which you can load in your web browser using the public IP DNS name.

39. Click **Next: Add Storage**
40. On **Step 4**, click **Next: Add Tags**
41. On **Step 5**, click **Add Tag** then configure:
    - **Key:** Name
    - **Value:** VPC-1 Instance
42. Click **Next: Configure Security Group**
43. Configure the following:
    - Click ⊙ **Select an existing security group**
    - Select 🟦 **VPC-1 SG**
    - Click **Review and Launch**
44. At the **Warning** screen, click **Continue**
45. On **Step 7**, review the settings, then click **Launch**
46. On the **Select an existing key pair or create a new key pair** window, configure the following:
    - Select **Create a key pair**
    - Provide **Key pair name:** VirginiaKeyPair
    - Click on Download Key Pair and save the key pair on your local desktop
    - Click **Launch Instances**
47. Click **View Instances**

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

48. 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.

49. Your instance should be selected if not, select it.

50. Copy the **Public IPv4 address**, and paste it into a web browser tab to open it.

Note: If you choose the **open address** link, you must change the URL that opened in your web browser **from https:// to http://**. 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.

A Web Page should appear:

Hello World, This is a WebServer

Congratulations! If you are able to see this page then your Web Server has been deployed successfully.

Conclusion

Congratulations! You now have successfully:

- Create an Amazon Virtual Private Cloud (VPC)
- Create a public subnet
- 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
- Deploy a web server instance

End Lab

As you dive deeper in the course, we have one more lab for the leaners. If you plan to work on the upcoming lab, then please leave the resources running in your account.

However, if you do not plan to work on the next lab then go ahead and delete all the resources that we created in this lab. Follow the below instructions to delete the VPC and the resources that were launched inside your VPC.

Optional: Clean Up

In this task, you will delete all the resources that we created in this.

Terminate Instance

51. In the **us-east-1 (N. Virginia)** Region AWS Management Console, on the **Services** menu, choose **EC2**.
52. On the left navigation pane, choose **Instances**.
53. Select the instance named **VPC-1 Instance**
54. In the Instance State menu *(Located near the top section of the navigation panel)*, select **Terminate**.
55. Wait for your web server to terminate. It should display the following:

- **Instance State:** Terminated

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

You have now successfully deleted **VPC-1 Instance** that was launched in the VPC.

Delete VPC-1

In the previous step, you deleted the instance which was an underlying dependency in the VPC. Now in this task, you will finally delete the VPC which will also delete its components like Subnets, Route Tables, Security Groups, Network ACLs and Internet Gateways.

56. In the **us-east-1 (N. Virginia)** Region AWS Management Console, on the **Services** menu, choose **VPC**.
57. On the left navigation pane, choose **Your VPCs**.
58. Select the VPC named **VPC-1**.

59. In the Actions menu *(Located near the top section of the navigation panel)*, select

    Terminate

60. A dialogue box will pop-up. To confirm deletion, type *delete* in the field.

61. To confirm deletion, type *delete* in the field.

You can click the refresh icon to refresh your VPC console. You should no longer see **VPC-1.**

APPENDIX

This section will help you understand subnetting in Amazon VPC.

VPC and subnet sizing for IPv4

Subnetting allows you to create multiple logical networks that exist within a single Class A, B, or C network. If you do not subnet, you are only able to use one network from your Class A, B, or C network, which is unrealistic.

Each data link on a network must have a unique network ID, with every node on that link being a member of the same network. If you break a major network (Class A, B, or C) into smaller subnetworks, it allows you to create a network of interconnecting subnetworks. Each data link on this network would then have a unique network/subnetwork ID. Any device, or gateway, that connects n networks/subnetworks has n distinct IP addresses, one for each network / subnetwork that it interconnects.

When you create a VPC, you must specify an IPv4 CIDR block for the VPC. The allowed block size is between a /16 netmask (65,536 IP addresses) and /28 netmask (16 IP addresses).

When you create a VPC, we recommend that you specify a CIDR block from the private IPv4 address ranges as specified in RFC 1918:

RFC 1918 range	Example CIDR block
10.0.0.0 - 10.255.255.255 (10/8 prefix)	Your VPC must be /16 or smaller, for example, 10.0.0.0/16.
172.16.0.0 - 172.31.255.255 (172.16/12 prefix)	Your VPC must be /16 or smaller, for example, 172.31.0.0/16.
192.168.0.0 - 192.168.255.255 (192.168/16 prefix)	Your VPC can be smaller, for example 192.168.0.0/20.

The CIDR block of a subnet can be the same as the CIDR block for the VPC (for a single subnet in the VPC), or a subset of the CIDR block for the VPC (for multiple subnets). The allowed block

size is between a /28 netmask and /16 netmask. If you create more than one subnet in a VPC, the CIDR blocks of the subnets cannot overlap.

For example, if you create a VPC with CIDR block 172.16.0.0/24, it supports 256 IP addresses. You can break this CIDR block into multiple smaller subnets, each supporting 16 IP addresses. One subnet uses CIDR block 172.16.0.0/28 (for addresses 172.16.0.0 - 172.16.0.15) and the other uses CIDR block 172.16.0.16/28 (for addresses 172.16.0.16 - 172.16.0.31).

By extending the mask to be 255.255.255.240, you have taken four bits (indicated by "sub") from the original host portion of the address and used them to make subnets.

```
172.16.0.0 -          10101100.00010000.00000000.00000000
255.255.255.240 -     11111111.11111111.11111111.11110000
                      ----------------------------------------|sub|------
```

172.16.0.0	255.255.255.240	host address range 1 to 15
172.16.0.16	255.255.255.240	host address range 16 to 31
172.16.0.32	255.255.255.240	host address range 32 to 47
172.16.0.48	255.255.255.240	host address range 48 to 63

There are tools available on the internet to help you calculate and create IPv4 subnet CIDR blocks. You can find tools that suit your needs by searching for terms such as 'subnet calculator' or 'CIDR calculator'. Your network engineering group can also help you determine the CIDR blocks to specify for your subnets.

The first four IP addresses and the last IP address in each subnet CIDR block are not available for you to use, and cannot be assigned to an instance. For example, in a subnet with CIDR block 172.16.0.0/24, the following five IP addresses are reserved:

- 172.16.0.0: Network address.
- 172.16.0.1: Reserved by AWS for the VPC router.
- 172.16.0.2: Reserved by AWS. The IP address of the DNS server is the base of the VPC network range plus two. For VPCs with multiple CIDR blocks, the IP address of the DNS server is located in the primary CIDR. We also reserve the base of each subnet range plus two for all CIDR blocks in the VPC. For more information, see Amazon DNS server.
- 172.16.0.3: Reserved by AWS for future use.
- 172.16.0.255: Network broadcast address. We do not support broadcast in a VPC, therefore we reserve this address.

Additional Resources

- VPC Introduction
- Route Tables
- Security Groups for Your VPC
- Internet Gateways
- Amazon DNS server