Simulation: Creating a Virtual Private Cloud

Simulation overview

Traditional networking is difficult. It involves equipment, cabling, complex configurations, and specialist skills. Amazon Virtual Private Cloud (Amazon VPC) hides the complexity and simplifies the deployment of secure private networks.

This simulation shows you how to build your own virtual private cloud (VPC) and deploy resources into it.

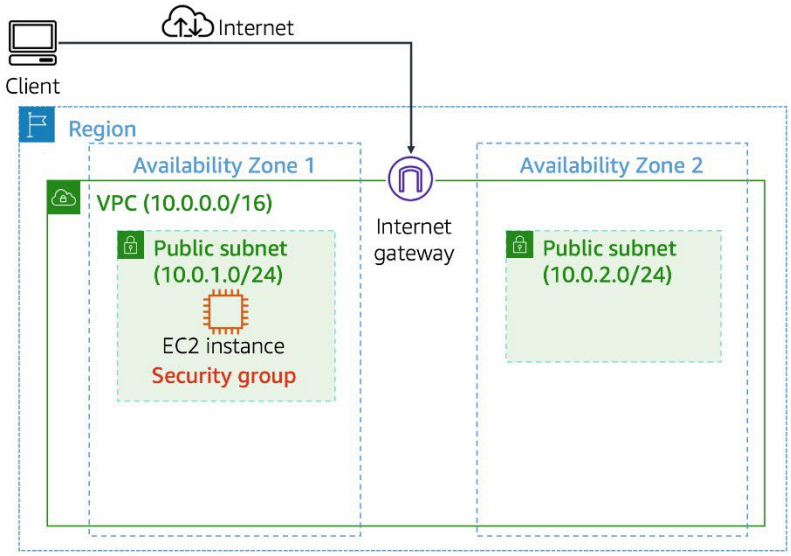
Additionally, this simulation is limited in its ability to be accessed by a screen reader. If you are using a screen reader, use the Simulation Instructions to understand how to perform the actions.

Objectives

After completing this simulation, you should be able to do the following:

- Explain the basic components of a VPC.
- Deploy a basic VPC with public subnets.
- Deploy an Amazon Elastic Compute Cloud (Amazon EC2) instance into a VPC.

At the end of this simulation, your architecture will look like the following example:



Public route table	
Destination	**Target**
10.0.0.0/16	Local
0.0.0.0/0	Internet gateway

In the preceding diagram, an EC2 instance is deployed into a VPC.

Duration

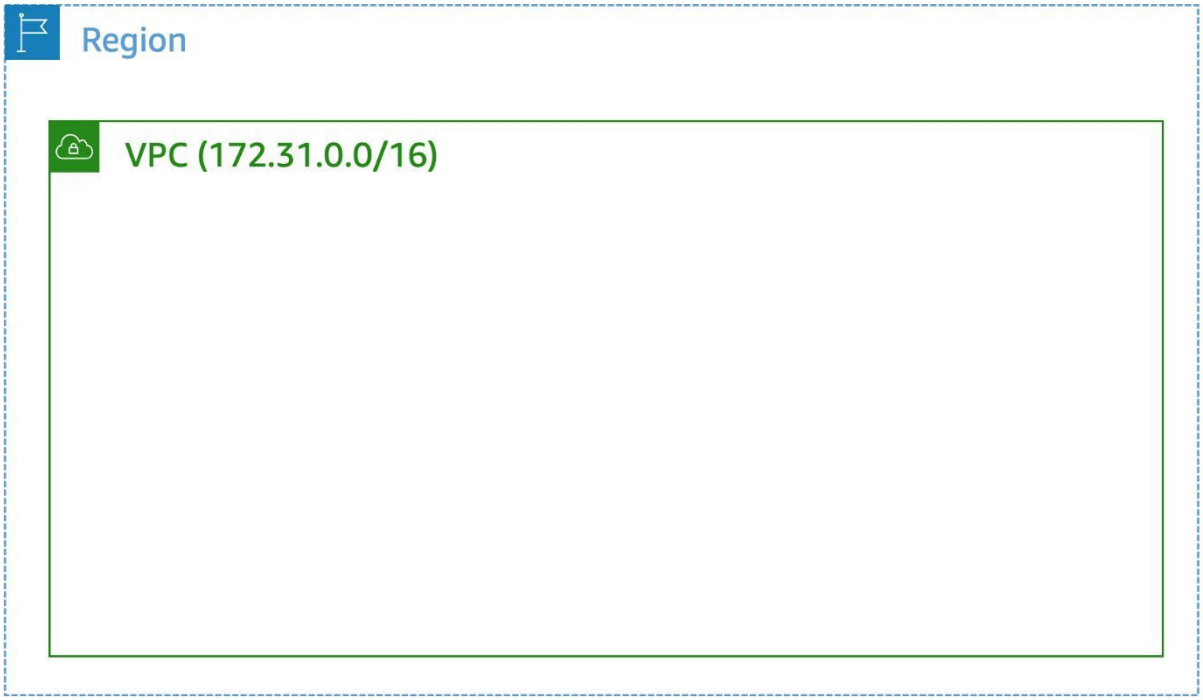
This simulation requires approximately **45 minutes** to complete.

Exploring the default VPC

Task 1: Explore the Example VPC configuration

In this simulation, you begin by exploring the Example VPC. The Example VPC is modeled after the default VPC that is automatically included with each Region within an Amazon Web Services (AWS) account.

A VPC is a virtual network that is dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch AWS resources, such as EC2 instances, into the VPC.



In the preceding diagram, a VPC is deployed into an AWS Region.

1. On the AWS Management Console on the **Services** menu, enter **VPC,** and press Enter on your keyboard.

2. From the search results, choose **VPC**.

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

You should see two VPCs: the default VPC and the **Example VPC**. Notice that the **Example VPC** is configured with the Classless Inter-Domain Routing (CIDR) range of **172.31.0.0/16**. This CIDR range includes all addresses from 172.31.0.0 through 172.31.255.255, a total of 65,536 addresses.

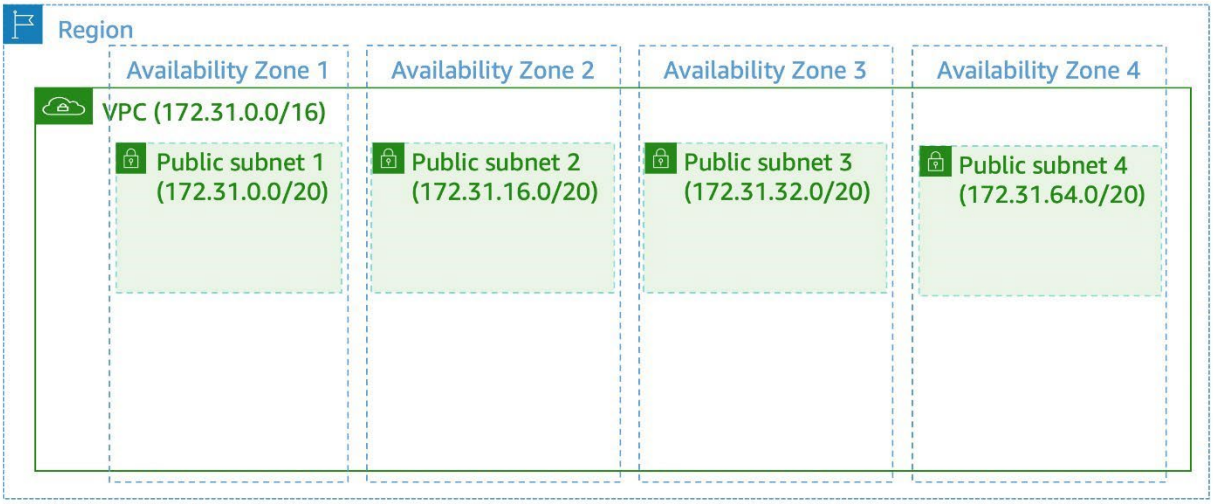
4. Make a note of the **VPC ID** for the **Example VPC** (the **VPC ID** ending in **882de**). You use this VPC ID later in the simulation.

5. Choose **Continue**.

Task 2: Explore a subnet

In this task, you explore a public subnet.

A subnet is a subrange of IP addresses in the VPC. AWS resources can be launched into a specified subnet. Use a *public subnet* for resources that must be accessible from the internet, and use a *private subnet* for resources that must remain inaccessible from the internet.



The preceding diagram includes the Example VPC and four subnets that reside inside it.

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

Notice that all of the subnets that begin with the name **PublicSubnet** are associated with the same VPC, the **Example VPC**. Also notice that each subnet has an IPv4 CIDR range. Each subnet CIDR range is a distinct subset of the addresses available in the VPC. When designing your subnets, you must ensure that the CIDR ranges do not overlap within the same VPC with address ranges used in other subnets.

7. From the list of subnets, select the check box for **PublicSubnet1**.

This subnet uses the IPv4 CIDR range 172.31.0.0/20.

The VPC has a CIDR block of 172.31.0.0/16, which includes all 172.31.x.x IP addresses. This subnet has a CIDR block of 172.31.0.0/20, which includes addresses 172.31.0.0–172.31.15.255. These CIDR ranges might look similar, but the subnet is smaller than the VPC because of the /20 in the CIDR range. This subnet uses the first 4,096 addresses available in the VPC. The console shows that only 4,091 addresses are available to use because AWS always reserves five addresses in each subnet for IP networking purposes.

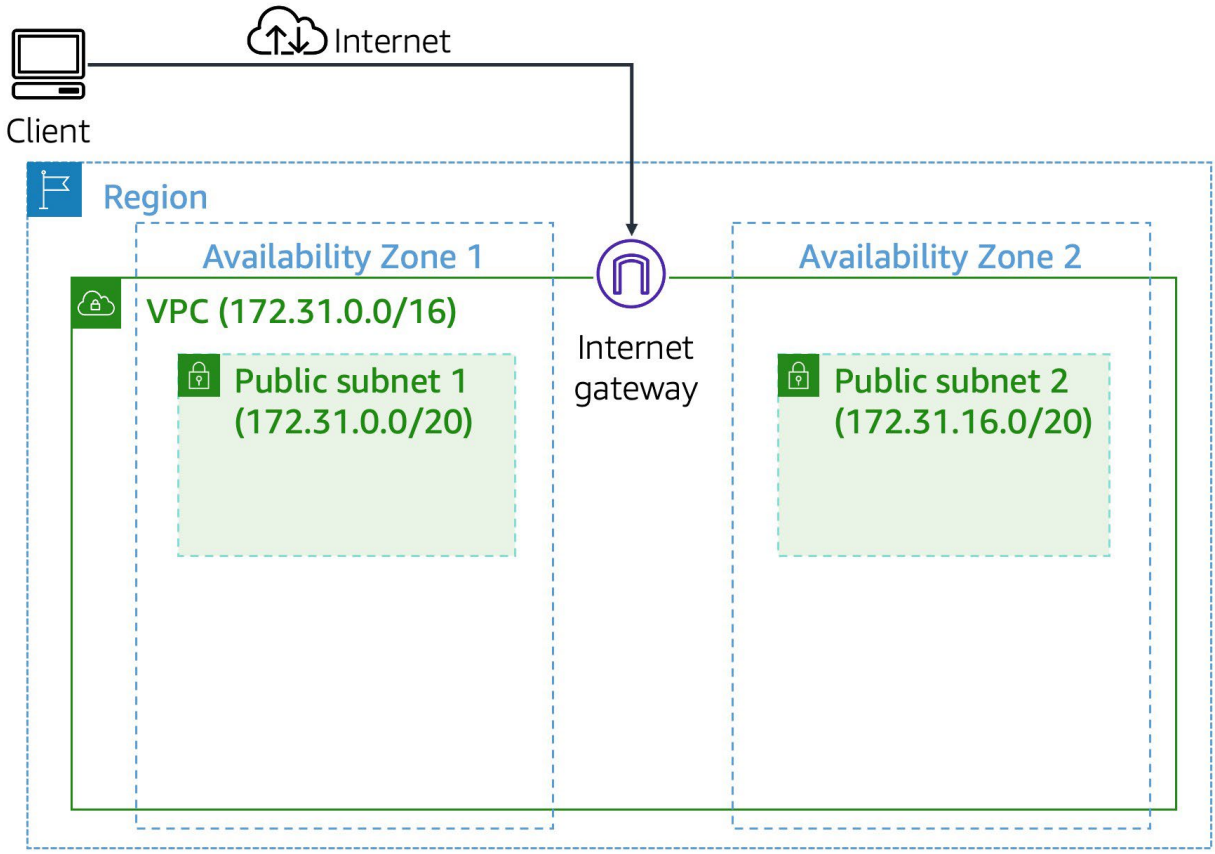
Notice that the value for **Auto-assign public IPv4 address** is **Yes**, which means that it is turned on. This means that the subnet automatically assigns a public IP address for all instances that are launched into it.

8. Choose **Continue.**

Task 3: Explore an internet gateway

In this task, you explore the VPC's internet gateway.

An *internet gateway* allows communication between the resources in a VPC and the internet. It is a horizontally scaled, redundant, and highly available VPC component. It imposes no availability risks or bandwidth constraints on network traffic.



In the preceding diagram, an internet gateway provides access to the internet to two subnets that reside in the VPC.

An internet gateway serves the following two purposes:

- To provide a target in route tables that connects to the internet
- To perform network address translation (NAT) for instances that were assigned public IPv4 addresses

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

Review the row containing the internet gateway named **Example Internet Gateway**. Notice that the **State** of the internet gateway is *Attached*. Also, notice that the **VPC ID** column shows that the internet gateway is attached the **VPC ID** for the **Example VPC**.

10. Choose **Continue**.

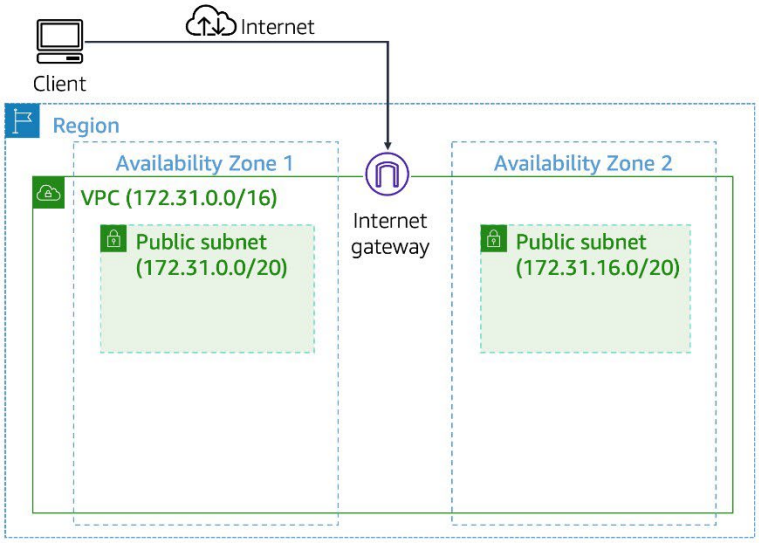
Task 4: Explore a route table

In this task, you explore the route table used by the Example VPC.

You verified that an internet gateway exists and that it is attached it to the Example VPC. Before the subnets can access the internet gateway, the route table associated with the subnets must be configured to use the internet gateway.

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 because the table controls the routing for the subnet. A subnet can be associated with only one route table at a time, but you can associate multiple subnets with the same route table.

To use an internet gateway, a subnet's route table must contain a route that directs internet-bound traffic to the internet gateway. If a subnet is associated with a route table that has a route to an internet gateway, it is known as a public subnet.



Public route table	
Destination	**Target**
172.31.0.0/16	Local
0.0.0.0/0	Internet gateway

In the preceding diagram, the route table directs traffic locally inside the VPC and sends public traffic to the internet gateway.

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

12. Locate the row containing the route table named **Public Route Table** and select it.

This route table is associated with the Example VPC.

The **Routes** tab is selected by default. Take a minute to review the routes.

There are two routes: a local route and a public route.

Route tables follow the longest prefix match when overlapping routes exist. Overlapping routes occur when two routes in a route table match on a specific destination IP. In this case, the 172.31.0.0/16 route overlaps with 0.0.0.0/0 route for all destination IPs within the 172.31.0.0/16 IP range.

Because the 172.31.0.0/16 prefix (16) is higher in value compared to the 0.0.0.0/0 prefix (0), the 172.31.0.0/16 route will be used first before the 0.0.0.0/0 route for all conflicting destination IPs.

As a result, all traffic that is destined for 172.31.0.0/16 (which is the range of the Example VPC) is routed locally. This route allows all subnets in a VPC to communicate with each other. All other traffic (0.0.0.0/0) is routed to the internet gateway.

13. Choose the **Subnet associations** tab.

In the **Explicit subnet associations** section, notice that the subnet with the **IPv4 CIDR** block of **172.31.0.0/20** is included in the list. This is the same subnet you reviewed earlier.

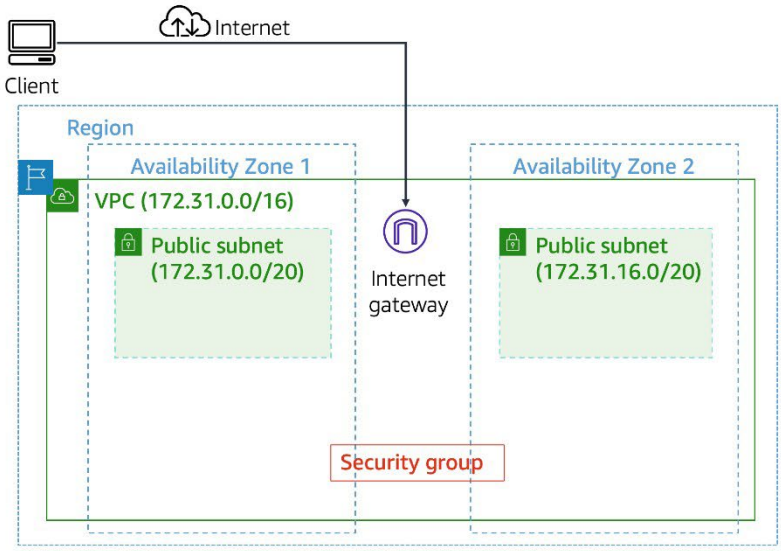
14. Choose **Continue**.

All of the subnets in this list are public subnets because they have a route table entry that sends traffic to the internet through the internet gateway.

Task 5: Explore a security group

In this task, you explore and update the security group used by the Example VPC subnets.

A *security group* acts as a virtual firewall for instances to control inbound and outbound traffic. Security groups operate at the level of the elastic network interface for the instance. Security groups do not operate at the subnet level. Thus, each instance can have its own firewall that controls traffic. If you do not specify a particular security group at launch time, the instance is automatically assigned to the default security group for the VPC.



Public route table	
Destination	**Target**
172.31.0.0/16	Local
0.0.0.0/0	Internet gateway

Security Group Inbound Rule		
Source	**Protocol**	**Port range**
Security Group	All	All

Security Group Outbound Rule		
Destination	**Protocol**	**Port range**
0.0.0.0/0	All	All

In the preceding diagram, the security group rules allow access to all ports for traffic that comes from the security group. The rules allow outbound access to the internet, both internal and external (0.0.0.0/0).

In this task, you review the default security group that is associated with the Example VPC. Then, you update a custom security group so that users can access resources by using HTTP.

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

16. Locate the row containing the **default VPC security group** for the Example VPC (VPC ID ending in **882de**) and select it.

17. In the lower half of the page, choose the **Outbound rules** tab.

    You see one rule. This rule allows **All** protocols and **All** port ranges to send traffic to any IP address (0.0.0.0/0).

18. Choose the **Inbound rules** tab.

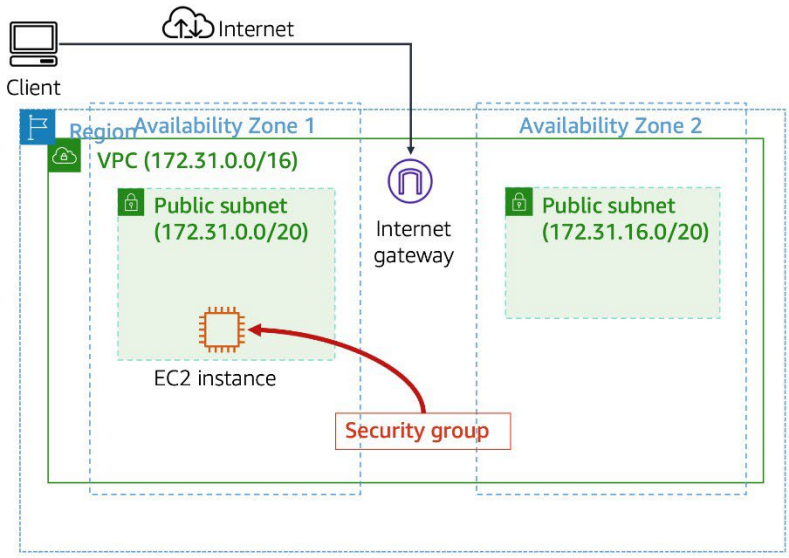
You find one rule for incoming traffic. This rule allows incoming traffic to **All** protocols and **All** port ranges from resources that use the default security group.

In a later step, you will test an EC2 instance that was deployed into the Example VPC when the simulation started. For incoming traffic from sources outside your VPC to access this website, you must add a new security group rule. Because you should not make changes to the default security group, you create a new one. Then, you add a rule to your new security group to permit HTTP (port 80) traffic that comes from anywhere on the internet (0.0.0.0/0).

19. Clear the **default** security group.
20. Locate the row containing the **Web-Server-SG** security group for the Example VPC and select it.
21. In the lower half of the page, choose the **Inbound rules** tab.
22. Choose **Edit inbound rules**.
23. Choose **Add rule**.
24. Configure the following settings:
    - For **Type**, choose **HTTP**.
    - From the **Source** dropdown list, choose **Anywhere-IPv4**.
    - For **Description**, enter **Allow web access**, and then press Enter.
25. Choose **Save rules**.

Task 6: Identify an EC2 instance's VPC and subnet and create a VPC

In this task, you find the VPC and subnet for an EC2 instance. You also test your Web-Server-SG security group configuration by confirming that you can access the EC2 instance from the internet.



Public route table	
Destination	**Target**
172.31.0.0/16	Local
0.0.0.0/0	Internet gateway

Security Group Inbound Rule		
Source	**Protocol**	**Port range**
0.0.0.0/0	TCP	80

Security Group Outbound Rule		
Destination	**Protocol**	**Port range**
0.0.0.0/0	All	All

In the preceding diagram, an EC2 instance is deployed into a public subnet in the default VPC. A security group is associated with the EC2 instance.

26. In the search box, enter **EC2**, and press Enter on your keyboard.

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

28. For **Resources**, choose **Instances (running).**
29. Select **Web-Server**.
30. In the bottom half of the screen, locate the value for **VPC ID** and the **Subnet ID.**

This EC2 instance was deployed into the Example VPC that you explored.

31. Review the following information, and then choose **Continue** in the player window.

This EC2 instance was deployed into Public Subnet 1, which is part of the Example VPC.

When you launch an EC2 instance, both the VPC and the subnet are defined for the instance. These configurations can't be changed. In a later task, you explore the steps for creating a new EC2 instance.

In a live environment, you could test that the website that runs on the EC2 instance is live. You would copy and paste the public IPv4 address into a web browser.

Being able to use the default VPC when you are first learning about and working with AWS cloud is very convenient. However, in the real world, you often need to create custom VPCs to meet a customer's requirements. For example, a customer might have already used the CIDR range of the default VPC in their on-premises network configuration. A customer might also want to vary how many addresses are included in each subnet. Because it is not possible to change the CIDR ranges assigned to the VPC or its subnets, you need to create a new VPC for your customer.

In this scenario, you create a new VPC. Your customer provided the following network requirements for the VPC's CIDR ranges:

Top-level VPC:

- VPC IPv4 CIDR block: 10.0.0.0/16

Availability Zones:

- They need to deploy their resources to two Availability Zones.

Two public subnets:

- Public Subnet 1: 10.0.0.0/24
- Public Subnet 2: 10.0.1.0/24

Two private subnets:

- Private Subnet 1: 10.0.2.0/24
- Private Subnet 2: 10.0.3.0/24

The Example VPC that you explored earlier did not have any private subnets. Remember that the difference between a public subnet and a private subnet is if they can be reached directly from the internet. The route table associated with a public subnet includes a route to an internet gateway, and the route table for a private subnet does not.

Task 7: Create a custom VPC

You can configure the VPC by defining its IP address range and creating subnets. You can also configure route tables, network gateways, and security settings.

The VPC console provides a wizard that can automatically create several VPC architectures. You use this wizard to create a new VPC.

If the configuration of a setting is not mentioned in these steps, leave the default value.

32. On the AWS Management Console on the **Services** menu, enter **VPC,** and press Enter on your keyboard.
33. From the search results, choose **VPC**.
34. In the left navigation pane, choose **Your VPCs**.

35. Choose **Create VPC.**
36. On the **Create VPC** page, configure the following settings:
    - For **Resources to create**, choose **VPC and more.**
    - For **Name tag auto-generation**, enter **Lab**, and then press Enter on your keyboard.
    - Ensure that **IPv4 CIDR block** is **10.0.0.0/16**.
    - For **Availability Zones (AZs)**, choose **2**.
    - For **Number of public subnets**, choose **2**.
    - For **Number of private subnets**, choose **2**.
    - Expand **Customize subnets CIDR blocks**.
    - Update the subnet CIDR block values with the following ranges. Press Enter after each input:

      Two public subnets:

      - **Public Subnet 1a:** 10.0.0.0/24
      - **Public Subnet 2b:** 10.0.1.0/24

      Two private subnets:

- **Private Subnet 1a:** 10.0.2.0/24
- **Private Subnet 2b:** 10.0.3.0/24

Take a moment to review the **Preview** diagram provided in the wizard.

37. Choose **Continue.**
38. Choose **Create VPC**.

The wizard immediately starts creating your VPC. After it finishes, you have a VPC that has all of the components that you explored earlier: subnets, route tables, an internet gateway, and a default security group. The VPC wizard also automatically configures the routes in the route tables for both the public subnets and the private subnets.

Like the default security group you explored earlier, the default security group created by the wizard blocks incoming traffic from the internet. To reach a web server in the new VPC, you need to add a rule to this default security group.

39. Choose **View VPC**.

Recall that a VPC's default security group does not allow traffic from outside the VPC. Because you should not change the default security group, you add a new security group to your custom VPC.

40. In the left navigation pane, choose **Security groups**.
41. Choose **Create security group**.
42. For **Security group name,** enter **Web-Server2-SG**, and then press Enter on your keyboard.
43. For **Description**, enter **Allows HTTP access**, and then press Enter on your keyboard.
44. For **VPC,** choose **Lab-vpc**.
45. In the **Inbound rules** section, choose **Add rule**, and then configure the following settings:
    - For **Type**, choose **HTTP**.
    - From the **Source** dropdown list, choose **Anywhere-IPv4**.
    - For **Description**, enter **Allow web access**, and then press Enter on your keyboard.
46. Choose **Create security group**.

Task 8: Explore the configuration settings for launching an EC2 instance into your custom VPC

In this task, you explore the **Launch an instance** page and enter the settings required to launch a new EC2 instance into your custom VPC.

47. In the **Launch instance** section, choose the **Launch instance** button.

48. On the **Launch an instance** page, configure the following options:

    - In the **Name and tags** pane, for **Name**, enter **Web-Server2,** and then press Enter on your keyboard.
    - In the **Application and OS Images (Amazon Machine Image)** section, **Amazon Linux** is the default. From the **Amazon Machine Image (AMI)** list, choose **Amazon Linux 2 AMI**.

Note: Do not choose **Amazon Linux 2023 AMI**.

    - The default instance type is **t2.micro**. Keep the default setting.
    - In the **Key pair (login)** section, from the **Key pair name - *required*** dropdown list, choose **Proceed without a key pair (Not recommended)**.
    - In the **Network settings** section, choose **Edit**, and configure the following settings:
        - For **VPC - *required***, choose **Lab-vpc**.
        - For **Subnet**, choose the subnet with **public1** in the name.
        - For **Auto-assign public IP**, choose **Enable**.
        - For **Firewall (security groups)**, choose **Select an existing security group**.
        - From the **Common security groups** dropdown list, choose the **WebServer2-SG** security group.

49. Choose **Launch instance**.

Well done! Now you know how to create a custom VPC and how to deploy a new EC2 instance into it.

Simulation complete

Congratulations! You have completed the simulation.

Your feedback is welcome and appreciated.

If you would like to share any suggestions or corrections, provide the details in the AWS Training and Certification Contact Form.