Amazon VPC

In this exercise, we will create a VPC with IPv4 CIDR block, a subnet with an IPv4 CIDR block and launch a public-facing instance into our subnet. Our instance will be able to communicate with the Internet and we will be able to access our instance from our local computer using SSH (if it's a Linux instance) or Remote Desktop (if it is a Windows instance). In real world environment, we can use this scenario to create a public-facing web server; for example, to host a blog.

This exercise is intended to help us set up your own nondefault VPC quickly.

To complete this exercise, we will do the following:

- Create a nondefault VPC with a single public subnet. Subnets enable us to group instances based on our security and operational needs. A public subnet is a subnet that has access to the Internet through an Internet gateway.
- Create a security group for our instance that allows traffic only through specific ports.
- Launch an Amazon EC2 instance into our subnet.
- Associate an Elastic IP address with our instance. This allows our instance to access the Internet.

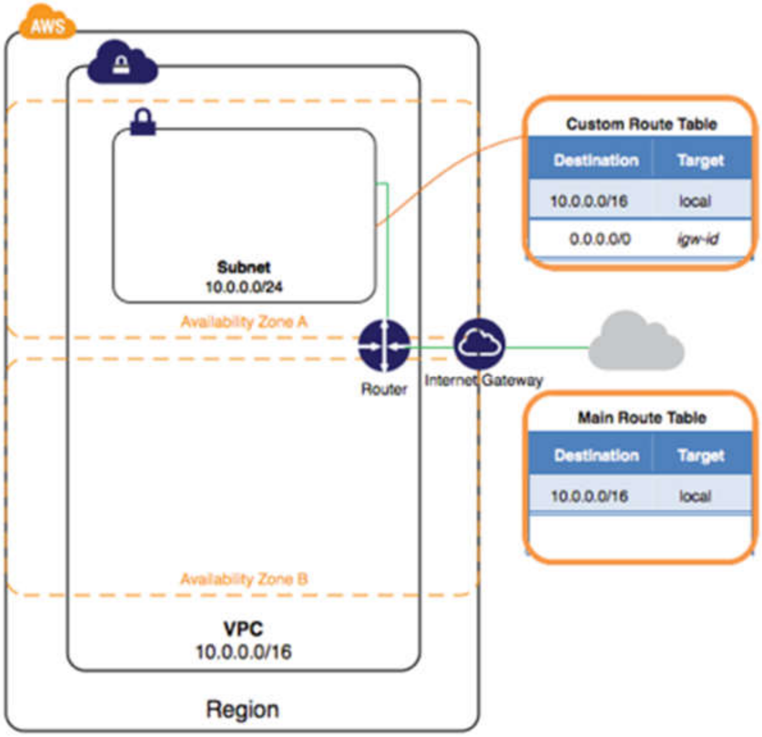
Contents

Step 1: Create the VPC

In this step, we will use the Amazon VPC wizard in the Amazon VPC console to create a VPC. The wizard performs the following steps for us:

- Creates a VPC with a /16 IPv4 CIDR block (a network with 65,536 private IP addresses).
- Attach an Internet gateway to the VPC.
- Creates a size /24 IPv4 subnet (a range of 256 private IP addresses) in the VPC.
- Creates a custom route table and associate it with our subnet so that traffic can flow between the subnet and the Internet gateway.

The following diagram represents the architecture of your VPC after we have completed this step.



Custom Route Table

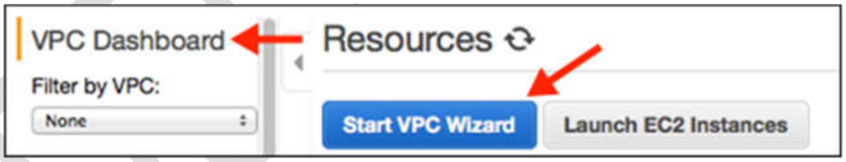
Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-id

Main Route Table

Destination	Target
10.0.0.0/16	local

To create a VPC using the Amazon VPC Wizard

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/
2. In the navigation bar, on the top-right, take note of the region in which we will be creating the VPC. Ensure that we continue working in the same region for the rest of this exercise, as we cannot launch an instance into our VPC from a different region.
3. In the navigation pane, choose **VPC dashboard**, and then choose **Start VPC Wizard**.



Note

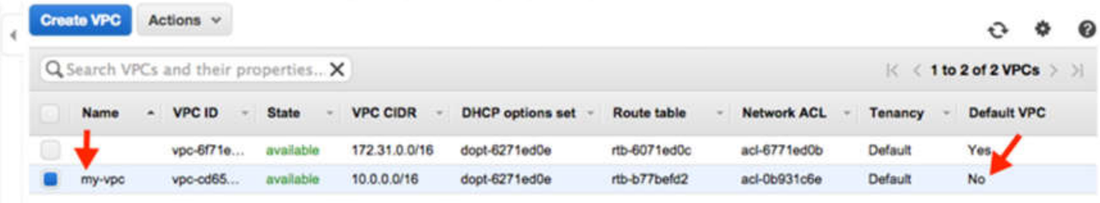
Do not choose **Your VPCs** in the navigation pane; you cannot access the VPC wizard from this page.

4. Choose the first option, **VPC with a Single Public Subnet**, and then choose **Select**.
5. On the configuration page, enter a name for our VPC in the **VPC name** field; for example, my-vpc, and enter a name for our subnet in the **Subnet name** field. This helps us to identify the VPC and subnet in the Amazon VPC console after we have created them. For this exercise, we can leave the rest of the configuration settings on the page, and choose **Create VPC**.

   (Optional) If you prefer, you can modify the configuration settings as follows, and then choose **Create VPC**.

- The **IPv4 CIDR block** displays the IPv4 address range that you'll use for your VPC (10.0.0.0/16), and the **Public subnet's IPv4 CIDR** field displays the IPv4 address range you'll use for the subnet (10.0.0.0/24). If you don't want to use the default CIDR ranges, you can specify your own.
- The **Availability Zone** list enables you to select the Availability Zone in which to create the subnet. You can leave **No Preference** to let AWS choose an Availability Zone for you.
- In the **Service endpoints** section, you can select a subnet in which to create a VPC endpoint to Amazon S3 in the same region.
- The **Enable DNS hostnames** option, when set to **Yes**, ensures that instances that are launched into your VPC receive a DNS hostname.
- The **Hardware tenancy** option enables you to select whether instances launched into your VPC are run on shared or dedicated hardware. Selecting a dedicated tenancy incurs additional costs.

6. A status window shows the work in progress. When the work completes, choose **OK** to close the status window.
7. The **Your VPCs** page displays your default VPC and the VPC that you just created. The VPC that we created is a nondefault VPC, therefore the **Default VPC** column displays **No**.



	Name	VPC ID	State	VPC CIDR	DHCP options set	Route table	Network ACL	Tenancy	Default VPC
□		vpc-6f71e...	available	172.31.0.0/16	dopt-6271ed0e	rtb-6071ed0c	acl-6771ed0b	Default	Yes
■	my-vpc	vpc-cd65...	available	10.0.0.0/16	dopt-6271ed0e	rtb-b77befd2	acl-0b931c6e	Default	No

Viewing Information About VPC

After you have created the VPC, you can view information about the subnet, the Internet gateway, and the route tables. The VPC that you created has two route tables — a main route table that all VPCs have by default, and a custom route table that was created by the wizard. The custom route table is associated with your subnet, which means that the routes in that table determine how the traffic for the subnet flows. If you add a new subnet to your VPC, it uses the main route table by default.

To view information about your VPC

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/
2. In the navigation pane, choose **Your VPCs**. Take note of the name and the ID of the VPC that you created (look in the **Name** and **VPC ID** columns). You will use this information to identify the components that are associated with your VPC.

3. In the navigation pane, choose **Subnets**. The console displays the subnet that was created when you created your VPC. You can identify the subnet by its name in **Name** column, or you can use the VPC information that you obtained in the previous step and look in the **VPC** column.
4. In the navigation pane, choose **Internet Gateways**. You can find the Internet gateway that's attached to your VPC by looking at the **VPC** column, which displays the ID and the name (if applicable) of the VPC.
5. In the navigation pane, choose **Route Tables**. There are two route tables associated with the VPC. Select the custom route table (the **Main** column displays **No**), and then choose the **Routes** tab to display the route information in the details pane:
    - The first row in the table is the local route, which enables instances within the VPC to communicate. This route is present in every route table by default, and you can't remove it.
    - The second row shows the route that the Amazon VPC wizard added to enable traffic destined for an IPv4 address outside the VPC (0.0.0.0/0) to flow from the subnet to the Internet gateway.
6. Select the main route table. The main route table has a local route, but no other routes.

Step 2: Create a Security Group

A security group acts as a virtual firewall to control the traffic for its associated instances. To use a security group, you add the inbound rules to control incoming traffic to the instance, and outbound rules to control the outgoing traffic from your instance. To associate a security group with an instance, you specify the security group when you launch the instance. If you add and remove rules from the security group, we apply those changes to the instances associated with the security group automatically.

Your VPC comes with a *default security group*. Any instance not associated with another security group during launch is associated with the default security group. In this exercise, you'll create a new security group, WebServerSG, and specify this security group when you launch an instance into your VPC.

Topics

- Rules for the WebServerSG Security Group
- Creating Your WebServerSG Security Group

Rules for the WebServerSG Security Group

The following table describes the inbound and outbound rules for the WebServerSG security group. You'll add the inbound rules yourself. The outbound rule is a default rule that allows all outbound communication to anywhere — you do not need to add this rule yourself.

Inbound

Source IP	Protocol	Port Range	Comments
0.0.0.0/0	TCP	80	Allows inbound HTTP access from any IPv4 address.
0.0.0.0/0	TCP	443	Allows inbound HTTPS access from any IPv4 address.
Public IPv4 address range of your home network	TCP	22	Allows inbound SSH access from your home network to a Linux/UNIX instance.
Public IPv4 address range of your home network	TCP	3389	Allows inbound RDP access from your home network to a Windows instance.
Outbound			
Destination IP	**Protocol**	**Port Range**	**Comments**
0.0.0.0/0	All	All	The default outbound rule that allows all outbound IPv4 communication.

Creating Your WebServerSG Security Group

You can create your security group using the Amazon VPC console.

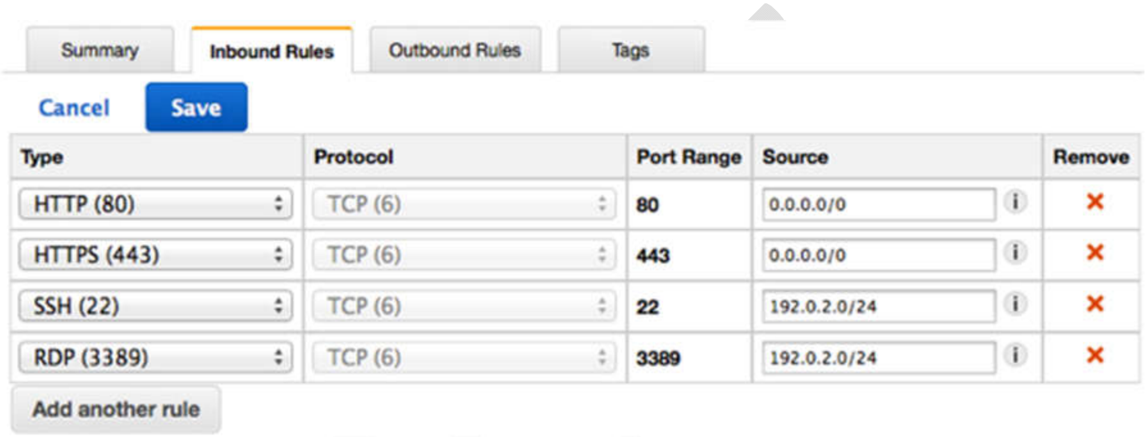
To create the WebServerSG security group and add rules

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Security Groups**.
3. Choose **Create Security Group**.
4. In the **Group name** field, enter WebServerSG as the name of the security group, and provide a description. You can optionally use the **Name tag** field to create a tag for the security group with a key of Name and a value that you specify.
5. Select the ID of your VPC from the **VPC** menu, and then choose **Yes, Create**.
6. Select the WebServerSG security group that you just created (you can view its name in the **Group Name** column).
7. On the **Inbound Rules** tab, choose **Edit** and add rules for inbound traffic as follows, and then choose **Save** when you're done:
    1. Select **HTTP** from the **Type** list, and enter 0.0.0.0/0 in the **Source** field.
    2. Choose **Add another rule**, then select **HTTPS** from the **Type** list, and enter 0.0.0.0/0 in the **Source** field.
    3. Choose **Add another rule**. If you're launching a Linux instance, select **SSH** from the **Type** list, or if you're launching a Windows instance, select **RDP** from the **Type** list. Enter your

network's public IP address range in the **Source** field. If you don't know this address range, you can use 0.0.0.0/0 for this exercise.

Important

If you use 0.0.0.0/0, you enable all IP addresses to access your instance using SSH or RDP. This is acceptable for the short exercise, but it's unsafe for production environments. In production, you'll authorize only a specific IP address or range of addresses to access your instance.

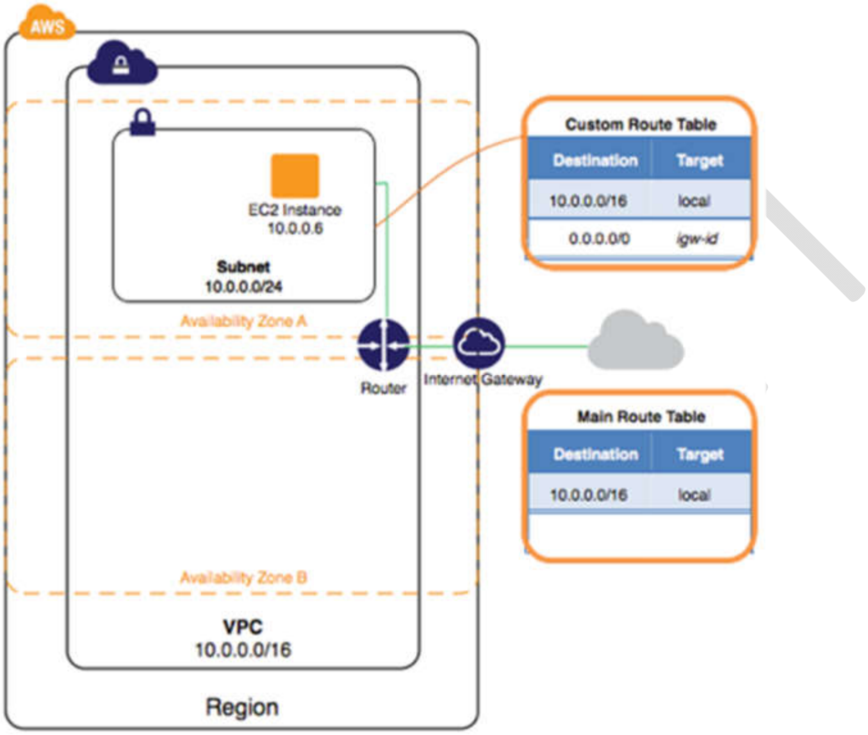


Type	Protocol	Port Range	Source	Remove
HTTP (80)	TCP (6)	80	0.0.0.0/0	✕
HTTPS (443)	TCP (6)	443	0.0.0.0/0	✕
SSH (22)	TCP (6)	22	192.0.2.0/24	✕
RDP (3389)	TCP (6)	3389	192.0.2.0/24	✕

Step 3: Launch an Instance into Your VPC

When you launch an EC2 instance into a VPC, you must specify the subnet in which to launch the instance. In this case, you'll launch an instance into the public subnet of the VPC you created. You'll use the Amazon EC2 launch wizard in the Amazon EC2 console to launch your instance.

The following diagram represents the architecture of your VPC after you've completed this step.



Custom Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-id

Main Route Table

Destination	Target
10.0.0.0/16	local

To launch an EC2 instance into a VPC

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation bar, on the top-right, ensure that you select the same region in which you created your VPC and security group.
3. From the dashboard, choose **Launch Instance**.
4. On the first page of the wizard, choose the AMI that you want to use. For this exercise, we recommend that you choose an Amazon Linux AMI or a Windows AMI.
5. On the **Choose an Instance Type** page, you can select the hardware configuration and size of the instance to launch. By default, the wizard selects the first available instance type based on the AMI you selected. You can leave the default selection, and then choose **Next: Configure Instance Details**.
6. On the **Configure Instance Details** page, select the VPC that you created from the **Network** list, and the subnet from the **Subnet** list. Leave the rest of the default settings, and go through the next pages of the wizard until you get to the **Add Tags** page.

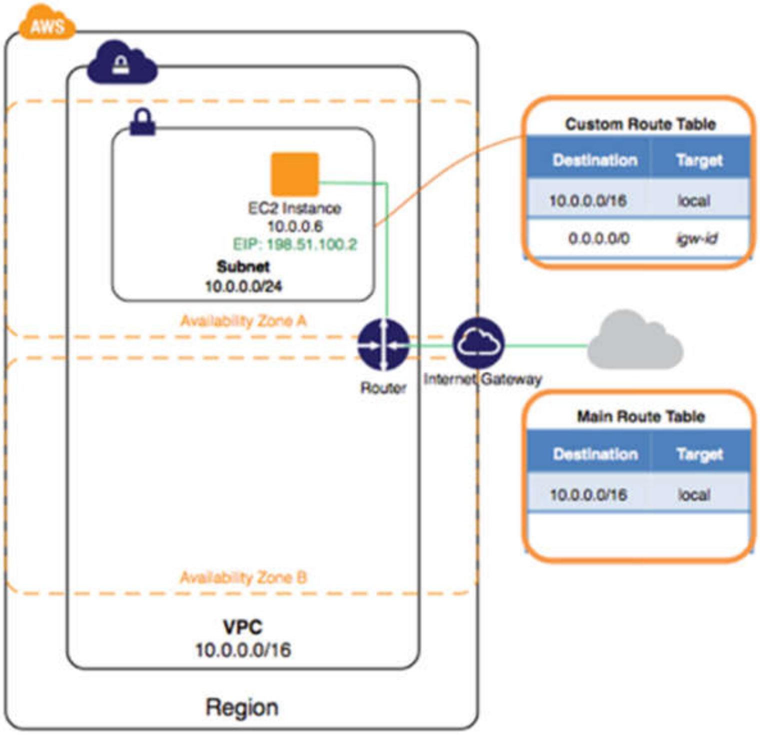
7. On the **Add Tags** page, you can tag your instance with a Name tag; for example Name=MyWebServer. This helps you to identify your instance in the Amazon EC2 console after you've launched it. Choose **Next: Configure Security Group** when you are done.
8. On the **Configure Security Group** page, the wizard automatically defines the launch-wizard-*x* security group to allow you to connect to your instance. Instead, choose the **Select an existing security group** option, select the **WebServerSG** group that you created previously, and then choose **Review and Launch**.
9. On the **Review Instance Launch** page, check the details of your instance, and then choose **Launch**.
10. In the **Select an existing key pair or create a new key pair** dialog box, you can choose an existing key pair, or create a new one. If you create a new key pair, ensure that you download the file and store it in a secure location. You'll need the contents of the private key to connect to your instance after it's launched.

    To launch your instance, select the acknowledgment check box, and then choose **Launch Instances**.
11. On the confirmation page, choose **View Instances** to view your instance on the **Instances** page. Select your instance, and view its details in the **Description** tab. The **Private IPs** field displays the private IP address that's assigned to your instance from the range of IP addresses in your subnet.

Step 4: Assign an Elastic IP Address to Your Instance

In the previous step, you launched your instance into a public subnet — a subnet that has a route to an Internet gateway. However, the instance in your subnet also needs a public IPv4 address to be able to communicate with the Internet. By default, an instance in a nondefault VPC is not assigned a public IPv4 address. In this step, you'll allocate an Elastic IP address to your account, and then associate it with your instance. For more information about Elastic IP addresses, see Elastic IP Addresses.

The following diagram represents the architecture of your VPC after you've completed this step.



Custom Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-id

Main Route Table

Destination	Target
10.0.0.0/16	local

To allocate and assign an Elastic IP address

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate New Address**, and then **Yes, Allocate**.

   Note

   If your account supports EC2-Classic, first select **EC2-VPC** from the **Network platform** list.
4. Select the Elastic IP address from the list, choose **Actions**, and then choose **Associate Address**.
5. In the dialog box, choose **Instance** from the **Associate with** list, and then select your instance from the **Instance** list. Choose **Yes, Associate** when you're done.

Your instance is now accessible from the Internet. You can connect to your instance through its Elastic IP address using SSH or Remote Desktop from your home network.

This completes the exercise; we can choose to continue using our instance in your VPC, or if we do not need the instance, we can terminate it and release its Elastic IP address to avoid incurring charges for them. We can also delete your VPC — note that you are not charged for the VPC and VPC components created in this exercise (such as the subnets and route tables).

Step 5: Clean Up

Before we can delete a VPC, you must terminate any instances that are running in the VPC. If we delete a VPC using the VPC console, it also deletes resources that are associated with the VPC, such as subnets, security groups, network ACLs, DHCP options sets, route tables, and Internet gateways.

To terminate our instance, release your Elastic IP address, and delete our VPC

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select your instance, choose **Actions**, then **Instance State**, and then select **Terminate**.
4. In the dialog box, expand the **Release attached Elastic IPs** section, and select the check box next to the Elastic IP address. Choose **Yes, Terminate**.
5. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
6. In the navigation pane, choose **Your VPCs**.
7. Select the VPC, choose **Actions**, and then choose **Delete VPC**.
8. When prompted for confirmation, choose **Yes, Delete**.