

---

# Amazon Virtual Private Cloud

## Getting Started Guide



## **Amazon Virtual Private Cloud: Getting Started Guide**

Copyright © 2017 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

## Table of Contents

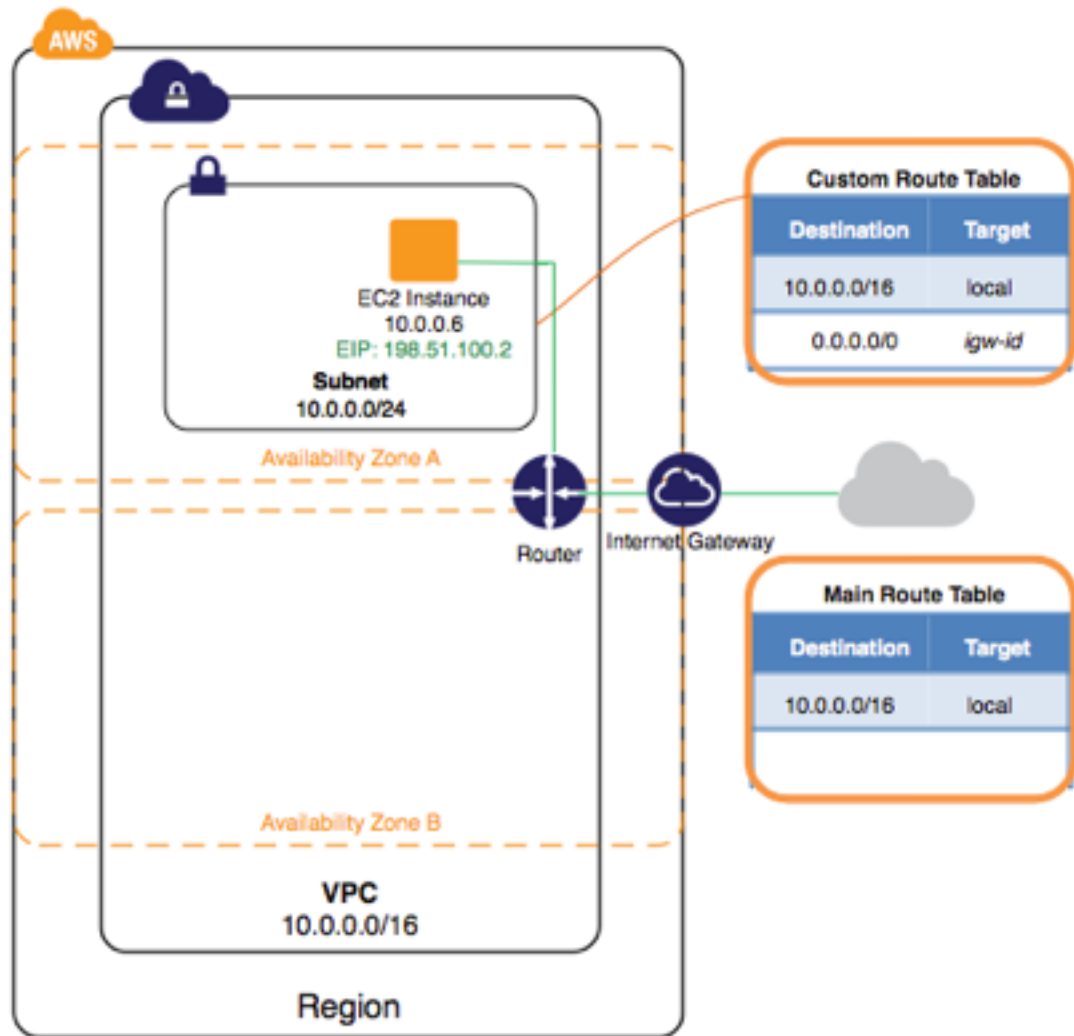
Overview .....	1
Getting Started With Amazon VPC .....	2
Step 1: Create the VPC .....	3
Step 2: Create a Security Group .....	6
Step 3: Launch an Instance into Your VPC .....	8
Step 4: Assign an Elastic IP Address to Your Instance .....	10
Step 5: Clean Up .....	12

# Overview

A virtual private cloud (VPC) is a virtual network that closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of Amazon Web Services (AWS). After you complete the tasks in this exercise, you'll have an Amazon EC2 instance running in a VPC that you can access from the Internet using SSH (for Linux instances) or Remote Desktop (for Windows instances).

For an overview of Amazon VPC, see [What is Amazon VPC?](#) in the *Amazon VPC User Guide*.

The following diagram shows the architecture that you'll create as you complete the exercise in this guide. The security group that you set up and associate with the instance allows traffic only through specific ports, locking down communication with the instance according to the rules that you specify. Using an Elastic IP address (EIP) enables an instance in a VPC, which is otherwise private, to be reached from the Internet through an Internet gateway (for example, it could act as a web server).



## Getting Started With Amazon VPC

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

### Note

This exercise is intended to help you set up your own nondefault VPC quickly. If you already have a default VPC and you want to get started launching instances into it (and not creating or configuring a new VPC), see [Launching an EC2 Instance into Your Default VPC](#). If you want to get started setting up a nondefault VPC that supports IPv6, see [Getting Started with IPv6 for Amazon VPC](#).

To complete this exercise, you'll do the following:

- Create a nondefault VPC with a single public subnet. Subnets enable you to group instances based on your 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 your instance that allows traffic only through specific ports.
- Launch an Amazon EC2 instance into your subnet.
- Associate an Elastic IP address with your instance. This allows your instance to access the Internet.

Before you can use Amazon VPC for the first time, you must sign up for Amazon Web Services (AWS). When you sign up, your AWS account is automatically signed up for all services in AWS, including Amazon VPC. If you haven't created an AWS account already, go to <http://aws.amazon.com>, and then choose **Create a Free Account**.

**Note**

This exercise assumes that your account supports the EC2-VPC platform only. If your account also supports the older EC2-Classic platform, you can still follow the steps in this exercise; however, you will not have a default VPC in your account to compare against your nondefault VPC. For more information, see [Supported Platforms](#).

Contents

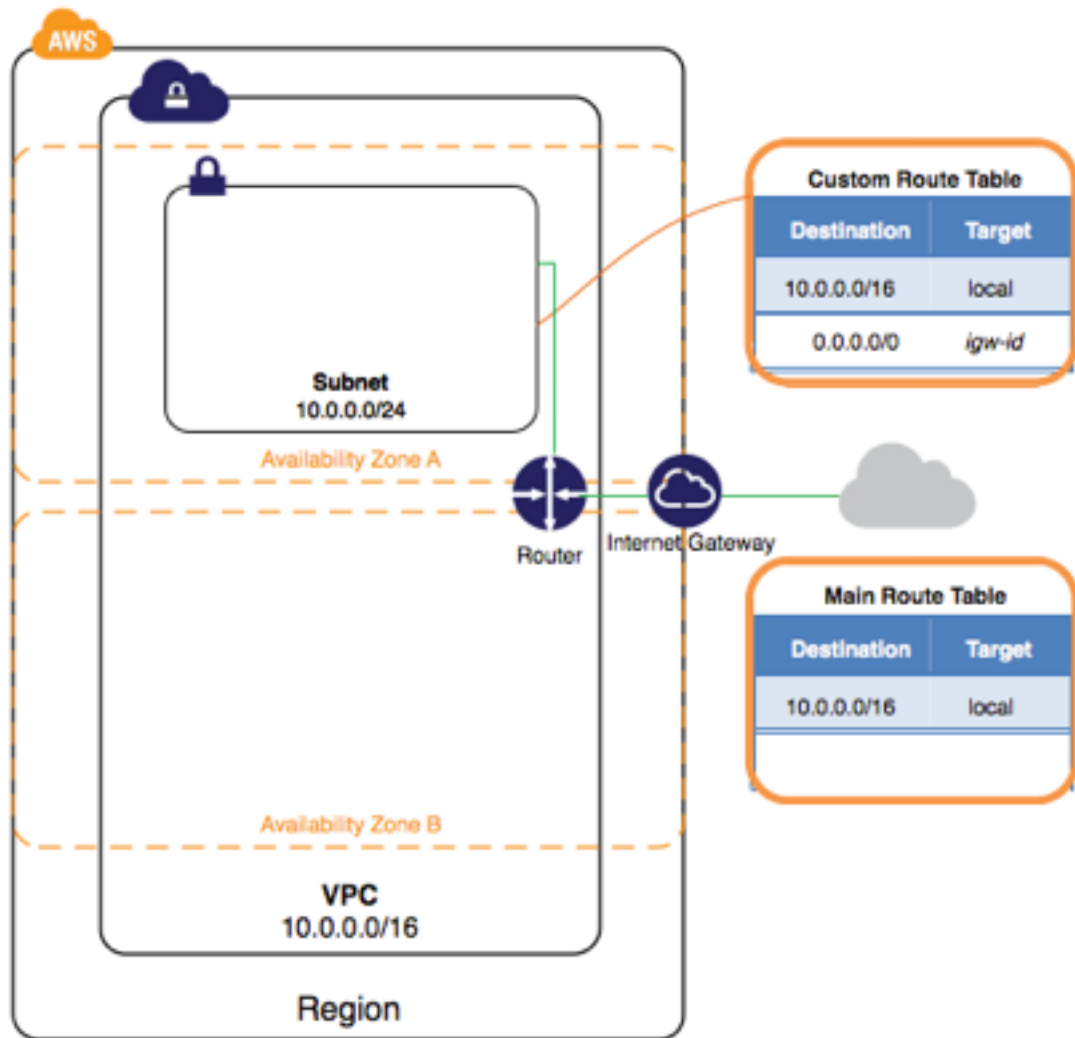
- [Step 1: Create the VPC \(p. 3\)](#)
- [Step 2: Create a Security Group \(p. 6\)](#)
- [Step 3: Launch an Instance into Your VPC \(p. 8\)](#)
- [Step 4: Assign an Elastic IP Address to Your Instance \(p. 10\)](#)
- [Step 5: Clean Up \(p. 12\)](#)

## Step 1: Create the VPC

In this step, you'll use the Amazon VPC wizard in the Amazon VPC console to create a VPC. The wizard performs the following steps for you:

- Creates a VPC with a /16 IPv4 CIDR block (a network with 65,536 private IP addresses). For more information about CIDR notation and the sizing of a VPC, see [Your VPC](#).
- Attaches an Internet gateway to the VPC. For more information about Internet gateways, see [Internet Gateways](#).
- Creates a size /24 IPv4 subnet (a range of 256 private IP addresses) in the VPC.
- Creates a custom route table, and associates it with your subnet, so that traffic can flow between the subnet and the Internet gateway. For more information about route tables, see [Route Tables](#).

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

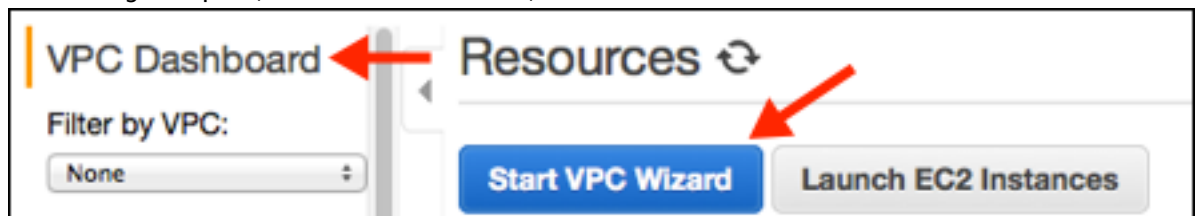


#### Note

This exercise covers the first scenario in the VPC wizard. For more information about the other scenarios, see [Scenarios for Amazon VPC](#).

#### 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 you'll be creating the VPC. Ensure that you continue working in the same region for the rest of this exercise, as you cannot launch an instance into your VPC from a different region. For more information about regions, see [Regions and Availability Zones](#).
3. In the navigation pane, choose **VPC dashboard**, and then choose **Start VPC Wizard**.



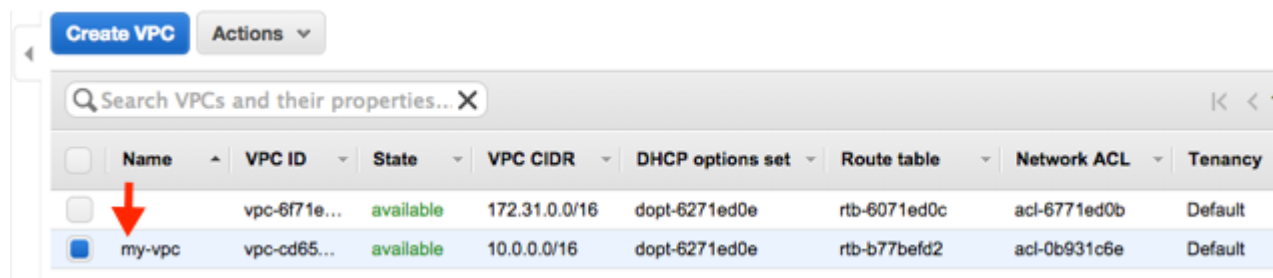
### Note

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 your VPC in the **VPC name** field; for example, `my-vpc`, and enter a name for your subnet in the **Subnet name** field. This helps you to identify the VPC and subnet in the Amazon VPC console after you've created them. For this exercise, you 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. For more information, see [VPC and Subnet Sizing](#).
  - 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. For more information, see [Regions and Availability Zones](#).
  - In the **Service endpoints** section, you can select a subnet in which to create a VPC endpoint to Amazon S3 in the same region. For more information, see [VPC Endpoints](#).
  - The **Enable DNS hostnames** option, when set to **Yes**, ensures that instances that are launched into your VPC receive a DNS hostname. For more information, see [Using DNS with Your VPC](#).
  - 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. For more information about hardware tenancy, see [Dedicated Instances](#) in the *Amazon EC2 User Guide for Linux Instances*.
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 you created is a nondefault VPC, therefore the **Default VPC** column displays **No**.



## Viewing Information About Your VPC

After you've 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 \(p. 6\)](#)
- [Creating Your WebServerSG Security Group \(p. 7\)](#)

## 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	TCP	22	Allows inbound SSH access from your home network to a Linux/UNIX instance.

of your home network			
Public IPv4 address range of your home network	TCP	3389	Allows inbound RDP access from your home network to a Windows instance.
<b>Outbound</b>			
<b>Destination IP</b>	<b>Protocol</b>	<b>Port Range</b>	<b>Comments</b>
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:
  - a. Select **HTTP** from the **Type** list, and enter `0.0.0.0/0` in the **Source** field.
  - b. Choose **Add another rule**, then select **HTTPS** from the **Type** list, and enter `0.0.0.0/0` in the **Source** field.
  - c. 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.

#### Warning

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.

Summary

Inbound Rules

Outbound Rules

Tags

Cancel

Save

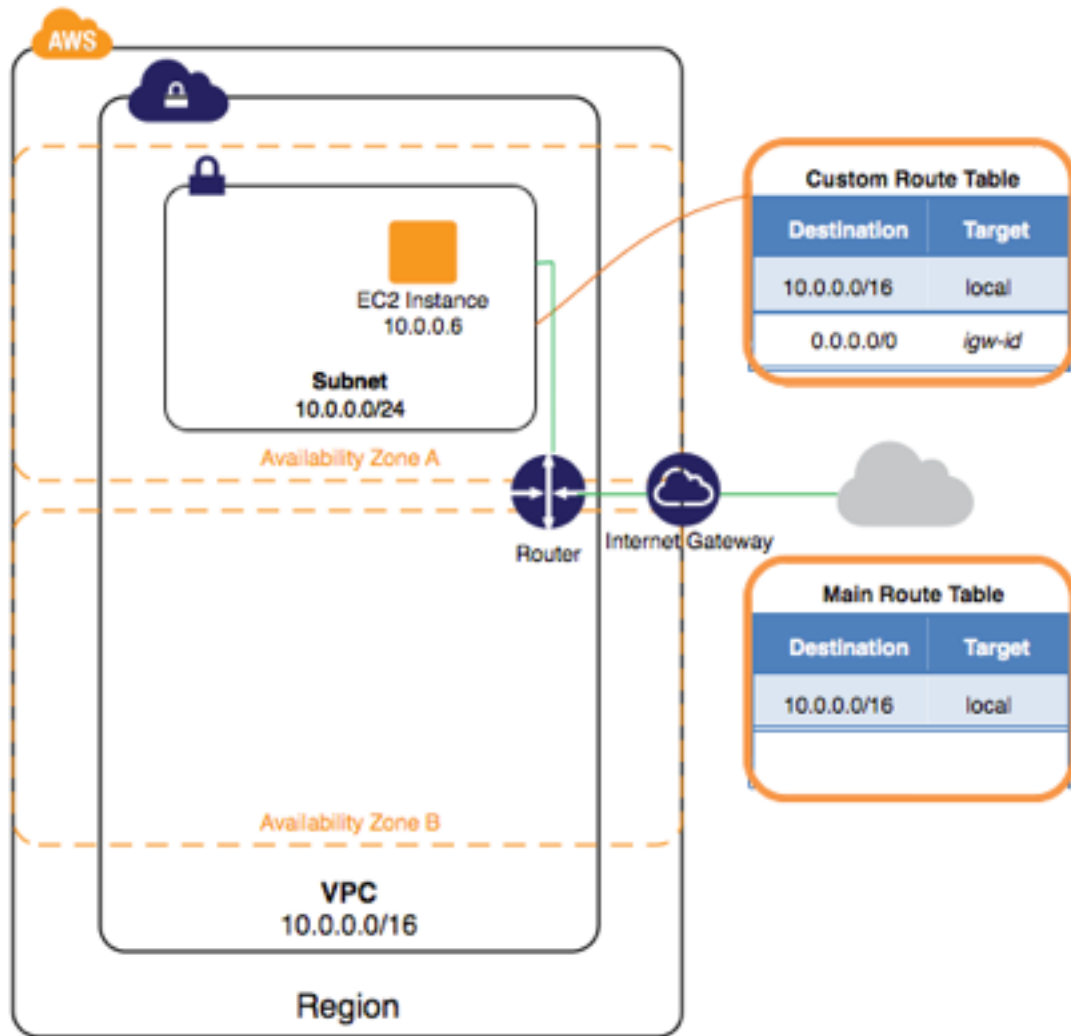
Type	Protocol	Port Range	Source	Remove
HTTP (80)	TCP (6)	80	0.0.0.0/0 ⓘ	✕
HTTPS (443)	TCP (6)	443	0.0.0.0/0 ⓘ	✕
SSH (22)	TCP (6)	22	192.0.2.0/24 ⓘ	✕
RDP (3389)	TCP (6)	3389	192.0.2.0/24 ⓘ	✕

Add another rule

## 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.



### 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**.

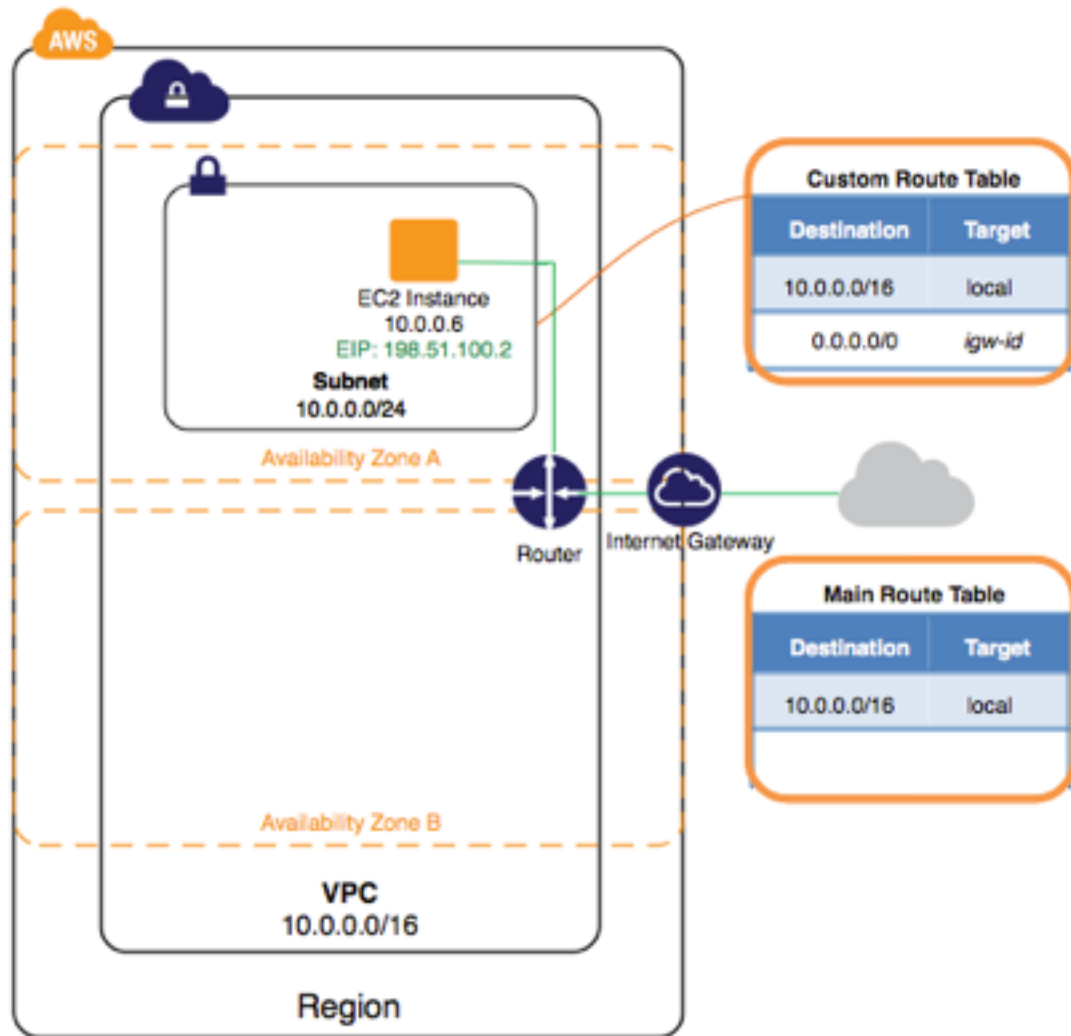
11. On the confirmation page, choose **View Instances** to view your instance on the **Instances** page. Select your instance, and view its details in the **Description** tab. The **Private IPs** field displays the private IP address that's assigned to your instance from the range of IP addresses in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

For more information about the options available in the Amazon EC2 launch wizard, see [Launching an Instance](#) in the *Amazon EC2 User Guide for Linux Instances*.

## 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.



### 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. For more information about how to connect to a Linux instance, see [Connecting to Your Linux Instance](#) in the *Amazon EC2 User Guide for Linux Instances*. For more information about how to connect to a Windows instance, see [Connect to Your Windows Instance Using RDP](#) in the *Amazon EC2 User Guide for Windows Instances*.

This completes the exercise; you can choose to continue using your instance in your VPC, or if you do not need the instance, you can terminate it and release its Elastic IP address to avoid incurring charges for them. You 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 you can delete a VPC, you must terminate any instances that are running in the VPC. If you 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 your instance, release your Elastic IP address, and delete your 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**.