Amazon Virtual Private Cloud

Getting Started Guide

API Version 2012-08-15

Amazon Web Services

Amazon Virtual Private Cloud: Getting Started Guide

Amazon Web Services

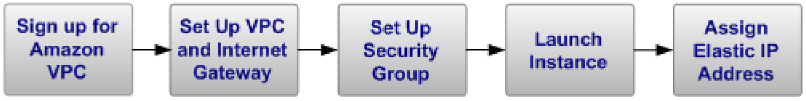
Get Started with Amazon VPC

Amazon Virtual Private Cloud enables you to create a virtual network topology—including subnets and routing—for your Amazon Elastic Compute Cloud (EC2) resources.

If you're familiar with Amazon EC2, you know that each instance you launch is randomly assigned a public IP address in the Amazon EC2 address space. Amazon VPC enables you to create an isolated portion of the Amazon Web Services (AWS) cloud—a *VPC*—and launch Amazon EC2 instances that have private (RFC 1918) addresses in the range of your choice (e.g., 10.0.0.0/16). You can define subnets within your VPC that enable you to group similar kinds of instances based on IP address range.

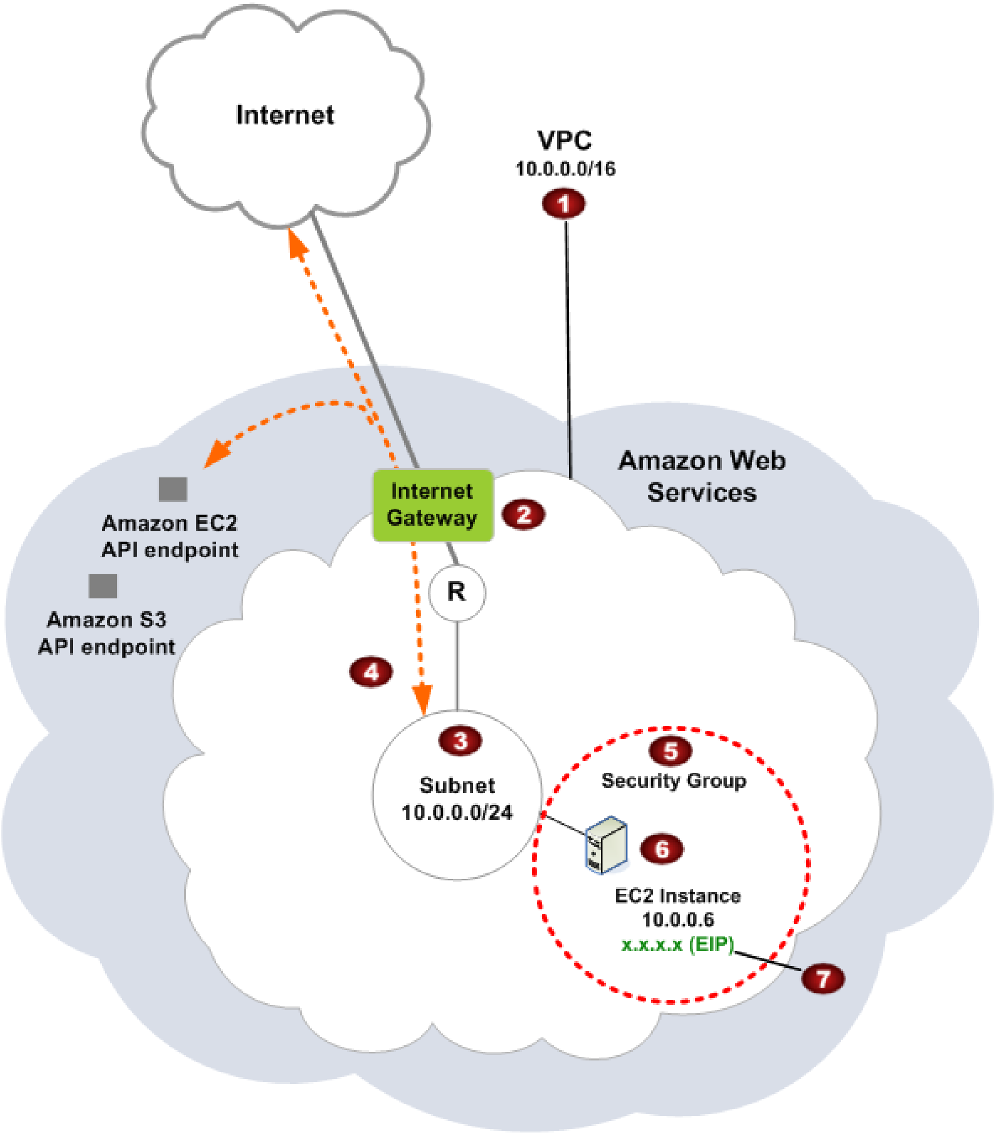
You can attach different types of gateways to your VPC to enable communication with the Internet or with your home network (over an IPsec VPN tunnel). You can set up routing and security to control the flow of traffic in and out of the instances and subnets.

This guide gives you a hands-on introduction to using Amazon VPC through the AWS Management Console. The exercise in this guide walks you through a simple scenario in which you set up a VPC with a single public subnet containing a running instance with an Elastic IP address. The following flow diagram shows the tasks you complete:



Overview of the Exercise

The following diagram and table summarize the tasks you perform in the exercise in this guide.



1	Create a *VPC*, which is an isolated portion of the AWS cloud.
2	Create and attach an Amazon VPC *Internet gateway*, which connects your VPC directly to the Internet and provides access to other AWS resources such as Amazon Simple Storage Service (Amazon S3).
3	Create an Amazon VPC *subnet*, which is a segment of a VPC's IP address range that you launch Amazon EC2 instances into. Subnets enable you to group instances based on your security and operational needs.

4	Set up routing in the VPC to enable traffic to flow between the subnet and the Internet.
5	Set up a security group to control the inbound and outbound traffic for the instances you launch.
6	Launch an instance in the subnet (either a Linux/UNIX instance or Windows instance depending on your preference). The instance has a private IP address from the subnet's range of addresses.
7	Assign an *Elastic IP address* to the instance. An Elastic IP address is a static, public address you can assign to any instance in your VPC. This assignment gives the instance a public IP address in addition to its private address. For an instance in your VPC to be reachable from the Internet, it must have an Elastic IP address.

After you complete the tasks in this exercise, you have a VPC with a running instance in it. You can connect to the instance from your home network using SSH (for a Linux/UNIX instance) or Remote Desktop (for a Windows instance). Because you've added an Elastic IP address to the otherwise private instance, the instance can be reached from the Internet (e.g., it could act as a web server). The security group that you've put the instance in opens only specific ports on the instance, effectively locking it down according to the rules you specify.

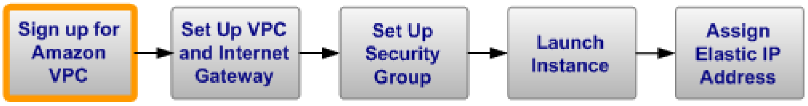
Important

Amazon VPC doesn't have a sandbox. When you do the exercise in this guide, you're charged the normal AWS rates for the Amazon EC2 instances you launch. (The charges are minimal—typically less than a few dollars.) For information about how you're charged for Amazon EC2 instances, go to the Amazon EC2 product page.

Tip

Two alternative versions of the scenario presented here include an IPsec VPN connection from your VPC to your data center, either instead of or in addition to the Internet gateway. To learn more about using a VPN connection with your VPC, go to Adding an IPsec Hardware Virtual Private Gateway to Your VPC in the *Amazon Virtual Private Cloud User Guide*.

Sign Up for Amazon VPC



The first task is to sign up for Amazon VPC.

> **Tip**
>
> If you're already an Amazon EC2 user, then you're already signed up for Amazon VPC automatically. You can skip directly to Set Up the VPC and Internet Gateway (p. 6).

To use Amazon VPC, you must:

- Sign up for an AWS account
- Sign up for Amazon EC2 (which automatically signs you up for Amazon VPC)

How to Sign Up for an AWS Account

AWS accounts are free. The login for the account is an email address you specify.

To sign up for an AWS account

1. Go to http://aws.amazon.com, and then click **Sign Up Now**.
2. Follow the on-screen instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

How to Sign Up for Amazon EC2 and Amazon VPC

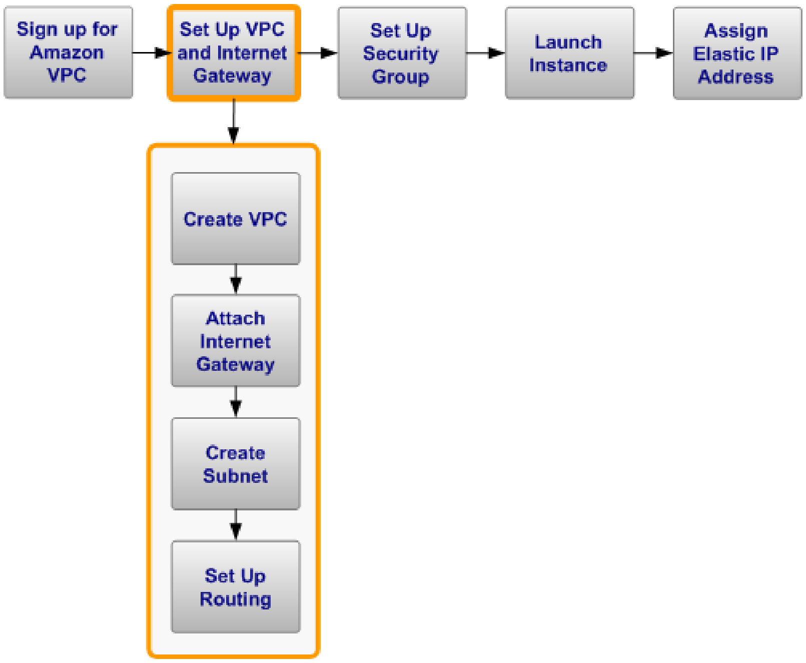
After you sign up for your AWS account, you need to sign up for Amazon EC2, which automatically signs you up for Amazon VPC.

To sign up for Amazon EC2 and Amazon VPC

1. Go to http://aws.amazon.com, click **Products**, and then select **Amazon Elastic Compute Cloud (Amazon EC2)**.
2. Click **Sign Up for Amazon EC2**.
3. Log in to your AWS account if prompted.
4. Follow the instructions.

AWS sends you a confirmation email.

Set Up the VPC and Internet Gateway



You're signed up for Amazon VPC, so now you're ready to start setting up your VPC.

For your convenience, you can use the wizard in the AWS Management Console to have Amazon VPC complete several of the setup tasks for you.

The wizard can perform the setup tasks that are indicated in the preceding flow diagram. You can use the wizard to automatically:

- Create a size /16 VPC. This means a network with 65,536 private (RFC 1918) IP addresses. For information about CIDR notation and what size "/16" means, go to the Wikipedia article about Classless Inter-Domain Routing.
- Attach an Internet gateway to the VPC.
- Add a size /24 subnet (which means 256 private IP addresses).
- Set up the routing in your VPC so that traffic can flow between the subnet and the Internet gateway. Here's more detail about that:
  - Your VPC has an implied router symbolized by the circle with the R in the layout diagram (see Get Started with Amazon VPC (p. 1)).
  - The wizard creates a route table, associates the subnet with the table, and adds a route that effectively says: "All traffic not intended for other instances in the VPC is routed to the Internet gateway."

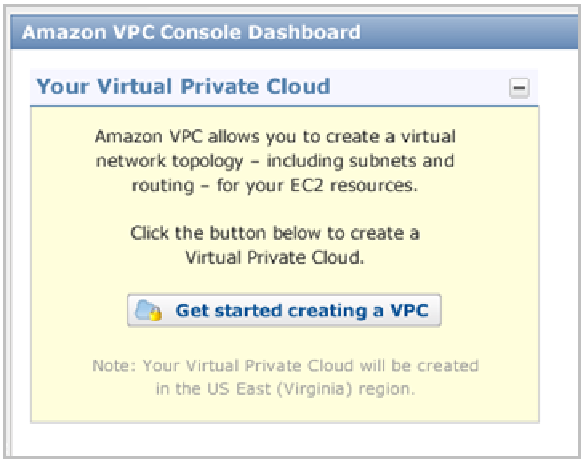
If you prefer, you can manually perform the preceding tasks in the console.

Important

AWS reserves the first four IP addresses and the last IP address in each subnet's CIDR block. They're not available for use.

To use the wizard

1. Sign in to the AWS Management Console and open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. On the **VPC Dashboard**, locate the **Your Virtual Private Cloud** area, and click **Get started creating a VPC**.



Amazon VPC Console Dashboard

Your Virtual Private Cloud

Amazon VPC allows you to create a virtual network topology – including subnets and routing – for your EC2 resources.

Click the button below to create a Virtual Private Cloud.

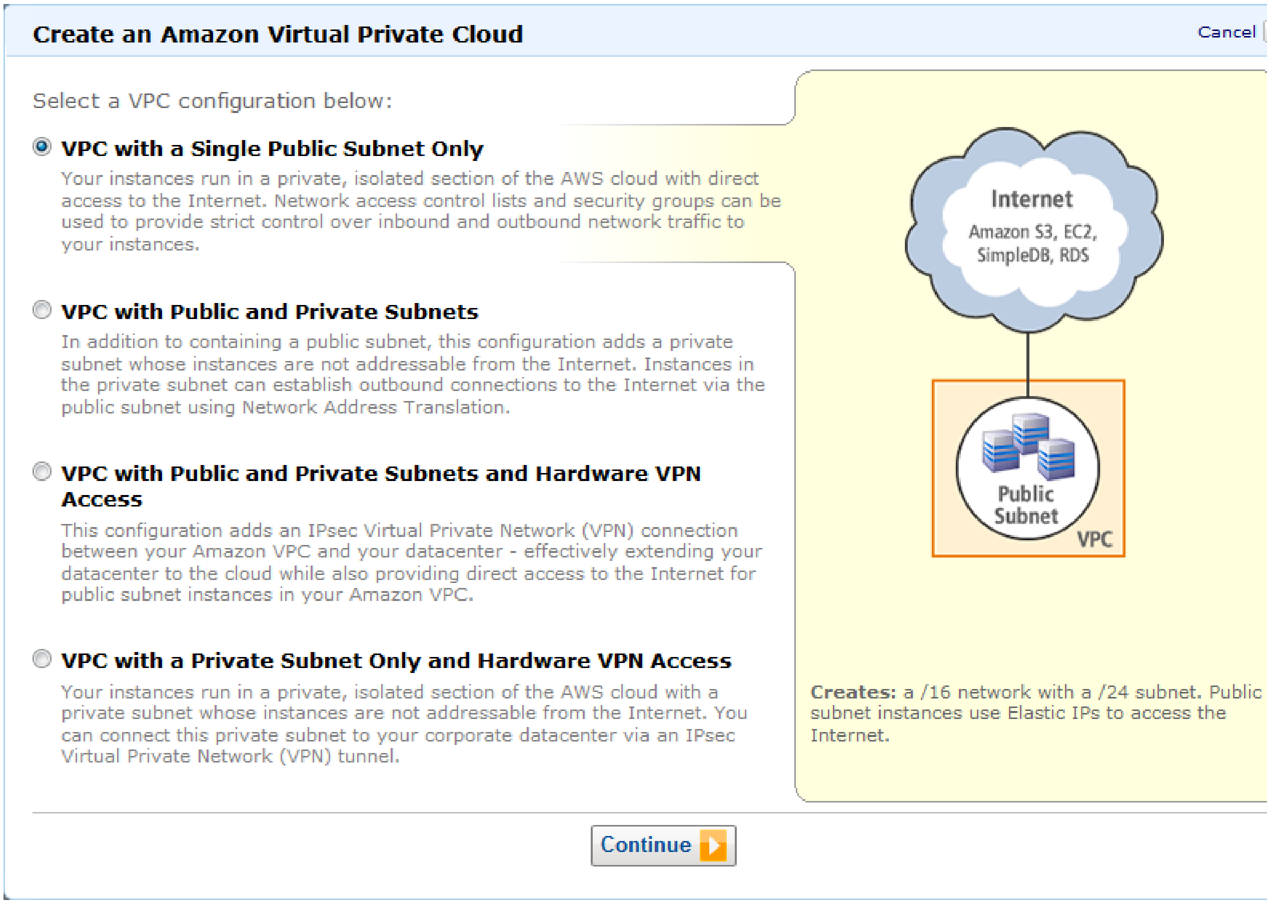
Get started creating a VPC

Note: Your Virtual Private Cloud will be created in the US East (Virginia) region.

The wizard starts and displays a page where you can select one of four options.

Tip

For this exercise, you'll use the first option; for information about how to use the other options, refer to the Amazon Virtual Private Cloud User Guide.



Create an Amazon Virtual Private Cloud

Select a VPC configuration below:

● **VPC with a Single Public Subnet Only**

Your instances run in a private, isolated section of the AWS cloud with direct access to the Internet. Network access control lists and security groups can be used to provide strict control over inbound and outbound network traffic to your instances.

○ **VPC with Public and Private Subnets**

In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation.

○ **VPC with Public and Private Subnets and Hardware VPN Access**

This configuration adds an IPsec Virtual Private Network (VPN) connection between your Amazon VPC and your datacenter - effectively extending your datacenter to the cloud while also providing direct access to the Internet for public subnet instances in your Amazon VPC.

○ **VPC with a Private Subnet Only and Hardware VPN Access**

Your instances run in a private, isolated section of the AWS cloud with a private subnet whose instances are not addressable from the Internet. You can connect this private subnet to your corporate datacenter via an IPsec Virtual Private Network (VPN) tunnel.

Creates: a /16 network with a /24 subnet. Public subnet instances use Elastic IPs to access the Internet.

3. Make sure the button is selected for the first option in the list (**VPC with a Single Public Subnet Only**), and click **Continue**.
   A confirmation page is displayed, showing the CIDR range that we'll use for your VPC and subnet (10.0.0.0/16 and 10.0.0.0/24, respectively). You can change any of these settings on this page.
4. Make any changes you want to the VPC, subnet CIDR ranges, and hardware tenancy settings. Then click **Create VPC**.

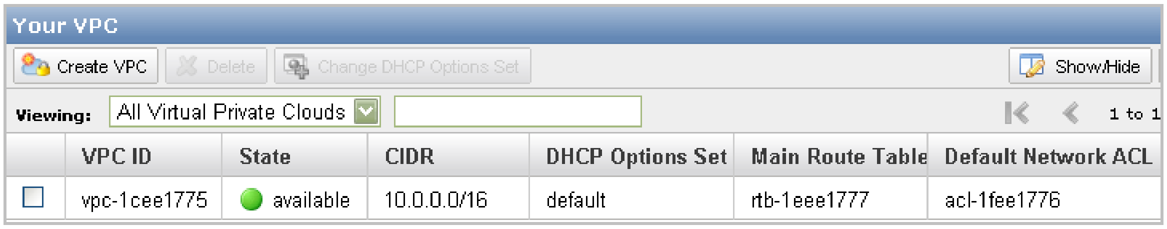
   ### Note

   For more information about hardware tenancy, see Using EC2 Dedicated Instances Within Your VPC in the *Amazon Virtual Private Cloud User Guide*.

   The wizard begins to create your VPC, Internet gateway, subnet, and route table. A status window shows the work in progress. When the wizard completes, a page is displayed confirming that your VPC has been created.
5. Click **Close**, which returns you to the VPC Dashboard.
6. Click **Your VPC** in the left navigation pane to display your VPC's information.

   ### Tip

   You might need to refresh the page for the VPC to appear.

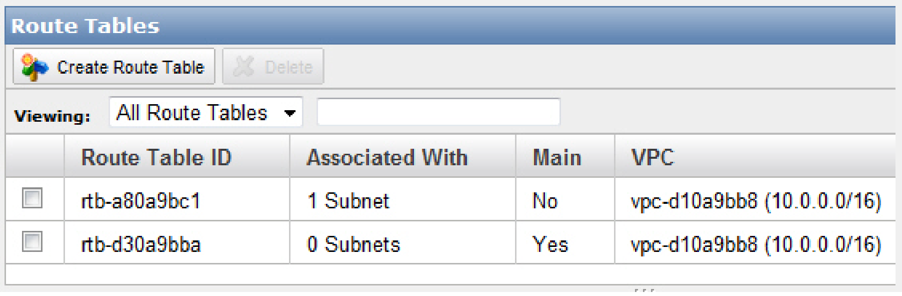


	VPC ID	State	CIDR	DHCP Options Set	Main Route Table	Default Network ACL
□	vpc-1cee1775	● available	10.0.0.0/16	default	rtb-1eee1777	acl-1fee1776

Your VPC has a default set of DHCP options, including a default DNS server we provide (AmazonProvidedDNS). The default settings are sufficient for this exercise, so you don't need to change these. For more information about DHCP options in your VPC, go to Using DHCP Options in Your VPC in the *Amazon Virtual Private Cloud User Guide*.

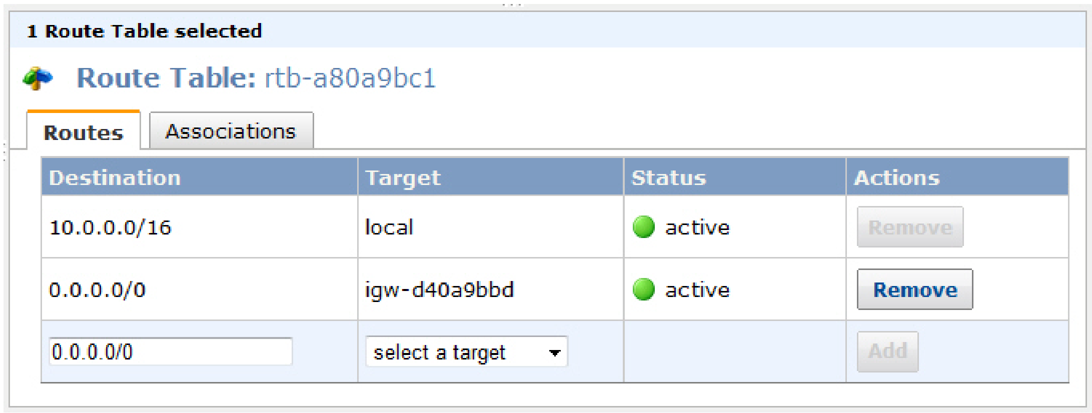
Your VPC also has a *main route table* and *default network ACL*, but they aren't important to this particular scenario.

7. Click **Internet Gateway** in the left navigation pane to display your Internet gateway's information. Notice that it has an ID (e.g., ig-10ee1779).
8. Click **Route Tables** in the left navigation pane.
   Your VPC has two route tables. One is the *main route table* that the VPC comes with by default, and the other is a custom route table the wizard created. Your subnet is associated with the custom route table, which means we use the routes in that table to determine how the subnet's traffic flows.



	Route Table ID	Associated With	Main	VPC
□	rtb-a80a9bc1	1 Subnet	No	vpc-d10a9bb8 (10.0.0.0/16)
□	rtb-d30a9bba	0 Subnets	Yes	vpc-d10a9bb8 (10.0.0.0/16)

9. Select the check box for the custom route table (the one with No in the *Main* column) and look at the route information displayed in the lower pane.



Destination	Target	Status	Actions
10.0.0.0/16	local	● active	Remove
0.0.0.0/0	igw-d40a9bbd	● active	Remove
0.0.0.0/0	select a target		Add

The first row in the table is the *local route*, which enables communication within the VPC. This route is present in the every route table by default, and you can't remove it.

The second row shows the route that the wizard added to the table to enable traffic destined for any IP address outside the VPC (i.e., 0.0.0.0/0) to flow from the subnet to the Internet gateway. We refer to this subnet as *public* because all traffic from the subnet goes to the Internet gateway.

Note

If you later decide to add new subnets to your VPC, by default they would use the main route table. The main route table has a local route, but no other routes. Therefore, any new subnet you create is initially not exposed to the Internet (i.e., it's *private*). If you decide to expose a new subnet as a public subnet, you could either change the routing in the main route table or associate the subnet with a custom route table.

You've completed the wizard portion of the implementation. Next you create a security group for your instances and add rules to it.

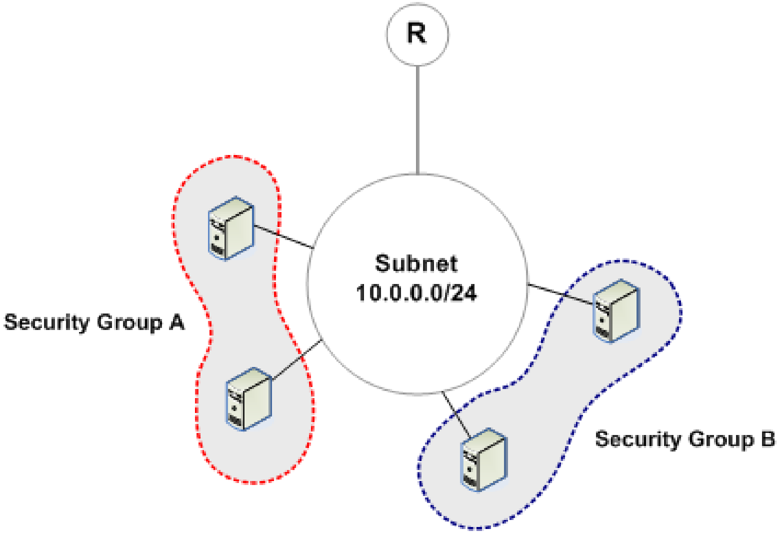
Set Up the Security Group

Sign up for Amazon VPC
Set Up VPC and Internet Gateway
Set Up Security Group
Launch Instance
Assign Elastic IP Address

You're now ready to create a security group for your VPC.

A security group is just a group of instances that shares a common set of rules that determine what traffic is allowed in and out of the instances. To use security groups, you create a group, add the inbound and outbound rules you want the group to use, and then launch instances into the group. You can add and remove rules from the group, and those changes automatically apply to the instances in the group.

The instances in a security group don't have to be in the same subnet in your VPC. Conversely, instances in the same subnet don't have to belong to the same security group. The following diagram illustrates how a subnet can have instances in more than one security group: two of the instances in the subnet are in group A, whereas the other two instances in that same subnet are in group B.



Important

Security groups are an existing concept used in Amazon EC2. However, the security groups you use in your VPC are different from the ones you use in EC2. If you're already an EC2 user, you can't use your existing security groups in your VPC. You must create new ones specifically for use in your VPC. However, the group names you use in your VPC can duplicate the group names you use in EC2 because each group has a unique AWS-assigned ID.

There are other differences between the two types of groups. However, for this exercise you just need to know that VPC security groups have both inbound and outbound rules, whereas EC2 security groups have only inbound rules. For more information about VPC security groups and how they differ from EC2 security groups in EC2, go to Security Groups in the *Amazon Virtual Private Cloud User Guide*.

Creating Your WebserverSG Group

For this exercise, you create a security group called *WebServerSG*, and launch an instance into the group.

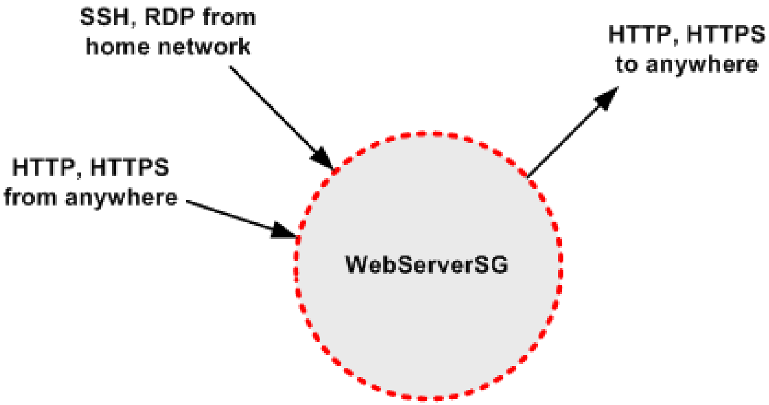
You also add rules to the WebServerSG group that enable:

- Inbound HTTP and HTTPS traffic from anywhere
- Inbound SSH and Remote Desktop (RDP) traffic from your home network
- Outbound HTTP and HTTPS traffic to anywhere

Note

This exercise adds SSH access for Linux/UNIX instances and RDP access for Windows instances. Your company might run only Linux or only Windows, so then you would need a rule for only one type of access.

The following figure shows the WebServerSG security group as a circle. The arrows pointing in and out of the security group circle represent the inbound and outbound rules you set for the group. Following the figure is a table that lists the inbound and outbound rules for the group and what they do.



Inbound			
Source IP	**Protocol**	**Port Range**	**Comments**

0.0.0.0/0	TCP	80	Allow inbound HTTP access from anywhere
0.0.0.0/0	TCP	443	Allow inbound HTTPS access from anywhere
Public IP address range of your home network	TCP	22	Allow inbound SSH access from your home network
Public IP address range of your home network	TCP	3389	Allow inbound RDP access from your home network
Outbound			
Dest IP	**Protocol**	**Port Range**	**Comments**
0.0.0.0/0	TCP	80	Allow outbound HTTP access to servers on the Internet (e.g., for software updates)
0.0.0.0/0	TCP	443	Allow outbound HTTPS access to servers on the Internet (e.g., for software updates)

Inbound rules regulate the traffic that is allowed to come into the instances in the group (i.e., the source of the traffic and the listening port on the instance). All return traffic is automatically allowed. For example, if a client on the Internet sends a request to a web server in your VPC inside the WebServerSG, the instance can respond, regardless of any outbound rules on the group. In this way, security groups are *stateful*.

Outbound rules control which destinations the instances in the group can send traffic to (i.e., the destination of the traffic and the destination port). All return traffic (i.e., a response from the host that received the traffic) is automatically allowed back into the instances, regardless of the inbound rules set on the security group.

> **Note**
>
> Your VPC comes with a *default security group*. Any instance not in another group automatically belongs to this group. Although we could use the default security group for this exercise, we've chosen to create the WebServerSG group instead.

To create the WebServerSG security group

1. Sign in to the AWS Management Console and open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the **Navigation** pane, click **Security Groups**.

   > **Note**
   >
   > This page shows all security groups that belong to your AWS account, including those you use in your VPC, and any EC2 groups you have. The VPC security groups have a value listed in the **VPC ID** column. You can filter the list to show only one type of security group if you want.

3. Click **Create Security Group**.

The **Create Security Group** dialog box opens.

4. Enter the name for your security group (WebServerSG), enter a description of the group, select your VPC's ID from the **VPC** menu, and click **Yes, Create**.
   The security group is created in your VPC and appears on the **Security Groups** page. Notice that it has an ID (e.g., sg-xxxxxxxx). You might have to turn on the **Group ID** column by clicking **Show/Hide** in the top right corner of the page.

You now have your WebServerSG security group. By default, every new security group starts with only an outbound rule that allows all traffic to leave the instances. You must add rules to enable any inbound traffic or to restrict the outbound traffic.

To add rules to the security group

1. In the list of security groups, select the check box for the group you just created.
   The lower pane displays the security group's details. There are also two tabs: one for working with the group's inbound rules and one for the outbound rules.
2. Add rules for inbound HTTP and HTTPS access to the group from anywhere:

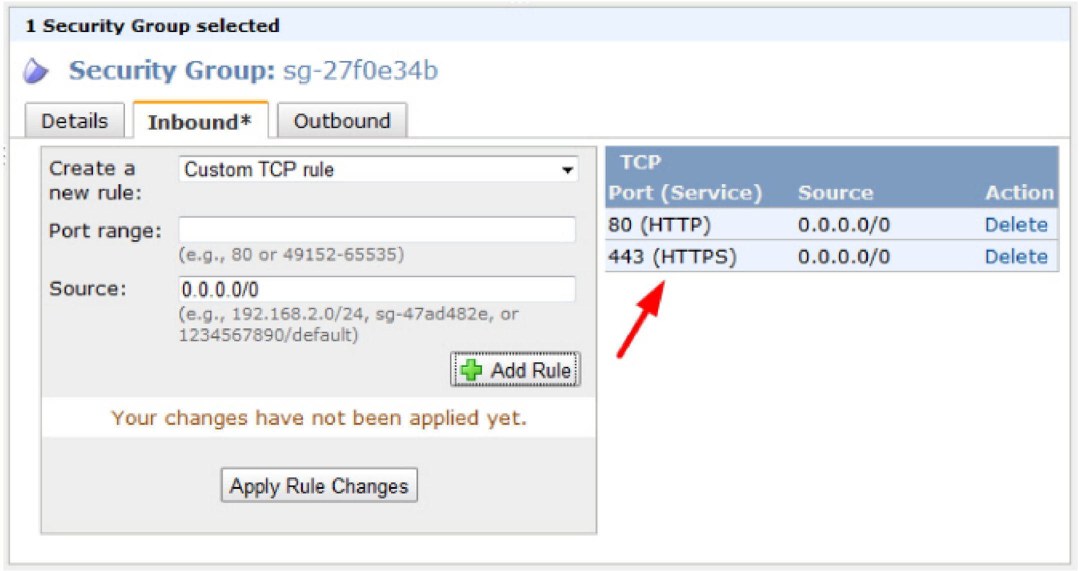
   a. On the **Inbound** tab, select `HTTP` from the **Create a new rule** drop-down list.

   b. Make sure the **Source** field's value is `0.0.0.0/0` and click **Add Rule**.

      The rule to allow HTTP access from anywhere (i.e., 0.0.0.0/0) is added to the **Inbound** tab, and an asterisk appears on the tab to indicate that you still need to click **Apply Rule Changes**. You'll apply rule changes after you've added all the inbound rules.

   c. Select `HTTPS` from the **Create a new rule** drop-down list and click **Add Rule**.

      The rule to allow HTTPS access from anywhere (i.e., 0.0.0.0/0) is added to the **Inbound** tab.



1 Security Group selected

Security Group: sg-27f0e34b

Create a new rule: Custom TCP rule

Port range:
(e.g., 80 or 49152-65535)

Source: 0.0.0.0/0
(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

Your changes have not been applied yet.

TCP Port (Service)	Source	Action
80 (HTTP)	0.0.0.0/0	Delete
443 (HTTPS)	0.0.0.0/0	Delete

3. Add rules for inbound SSH and Remote Desktop (RDP) access to the group from your home network's public IP address range:

   a. On the **Inbound** tab, select `SSH` from the **Create a new rule** drop-down list.

b. In the **Source** field, enter your home network's public IP address range (e.g., 192.0.2.0/24). If you don't know this address range, you can use 0.0.0.0/0 temporarily for this exercise.

c. Click **Add Rule**.
The rule is added to the **Inbound** tab.

d. Select `RDP` from the **Create a new rule** drop-down list.

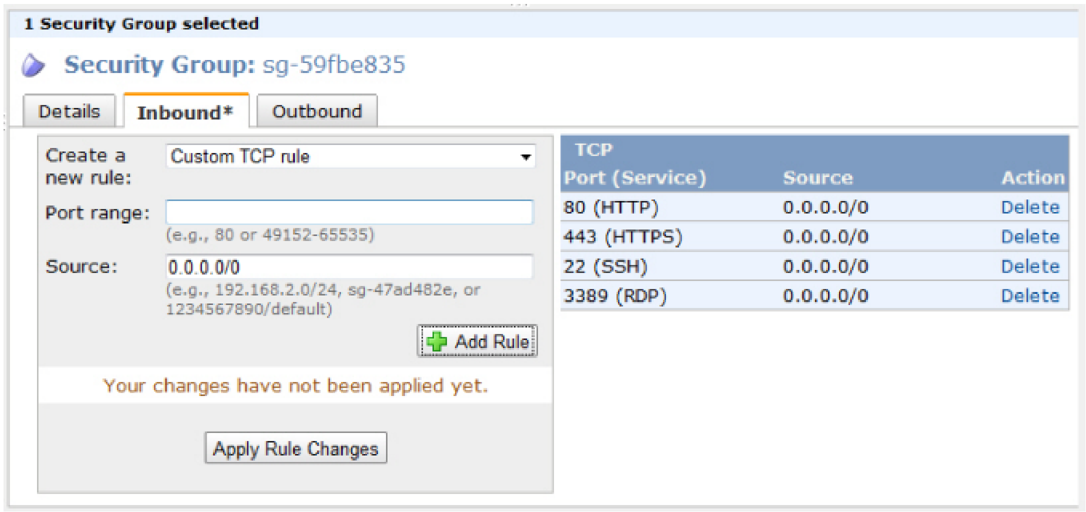
e. In the **Source** field, enter your home network's public IP address range. If you don't know this address range, you can use 0.0.0.0/0 temporarily for this exercise.

Caution

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.

f. Click **Add Rule**.

The rule is added to the **Inbound** tab.



1 Security Group selected

Security Group: sg-59fbe835

Create a new rule: Custom TCP rule

Port range: (e.g., 80 or 49152-65535)

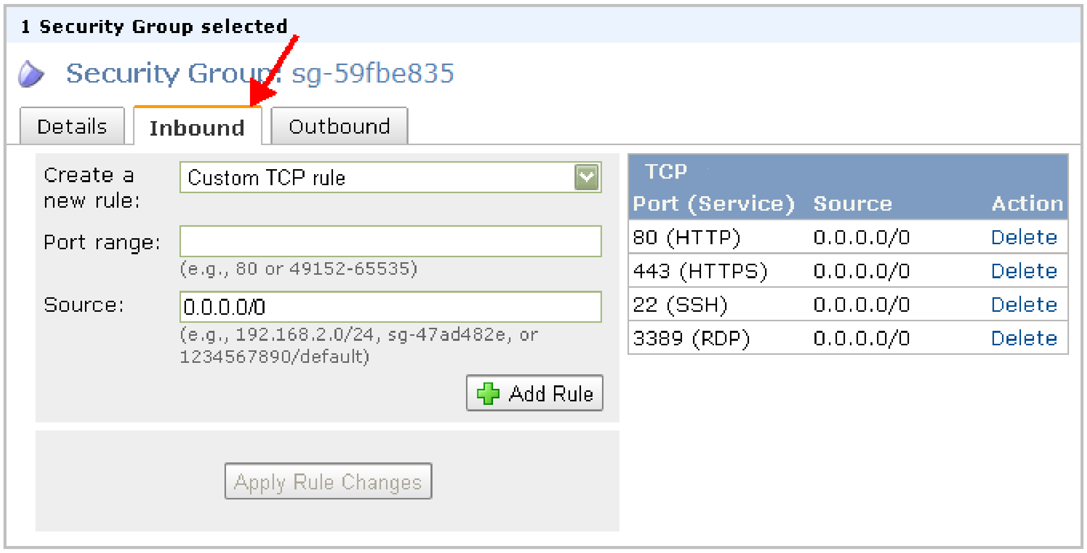
Source: 0.0.0.0/0 (e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

Your changes have not been applied yet.

TCP Port (Service)	Source	Action
80 (HTTP)	0.0.0.0/0	Delete
443 (HTTPS)	0.0.0.0/0	Delete
22 (SSH)	0.0.0.0/0	Delete
3389 (RDP)	0.0.0.0/0	Delete

4. Click **Apply Rule Changes**.

The new inbound rules now apply to the security group, and the asterisk no longer appears on the tab.



1 Security Group selected

Security Group: sg-59fbe835

Create a new rule: Custom TCP rule

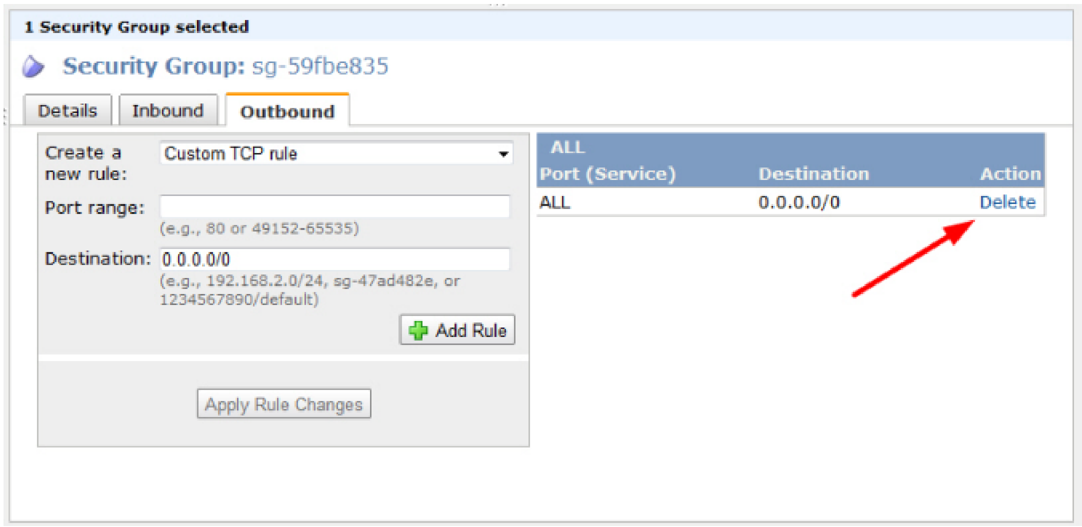
Port range:
(e.g., 80 or 49152-65535)

Source: 0.0.0.0/0
(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

TCP		
Port (Service)	Source	Action
80 (HTTP)	0.0.0.0/0	Delete
443 (HTTPS)	0.0.0.0/0	Delete
22 (SSH)	0.0.0.0/0	Delete
3389 (RDP)	0.0.0.0/0	Delete

5. Limit outbound access to only HTTP and HTTPS from the group to anywhere:

   a. On the **Outbound** tab, locate the default rule that enables all outbound traffic, and click **Delete**.



1 Security Group selected

Security Group: sg-59fbe835

Create a new rule: Custom TCP rule

Port range:
(e.g., 80 or 49152-65535)

Destination: 0.0.0.0/0
(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

ALL		
Port (Service)	Destination	Action
ALL	0.0.0.0/0	Delete

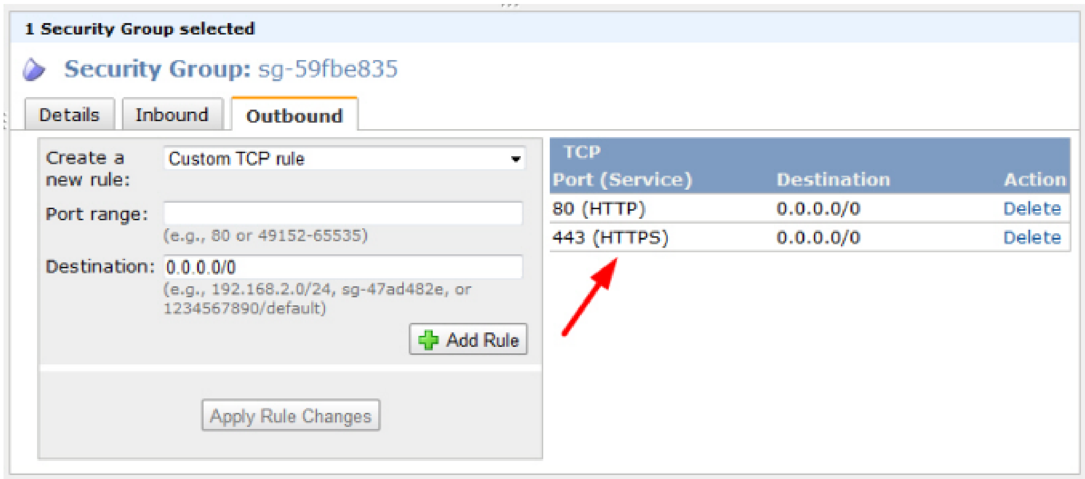
   The rule is marked for deletion, and an asterisk appears on the tab. The deletion will not take effect until you click **Apply Rule Changes**, which you'll do after adding new outbound rules to the group.

   b. Select HTTP from the **Create a new rule** drop-down list and click **Add Rule**.
   The rule allowing outbound HTTP access to anywhere (i.e., 0.0.0.0/0) is added to the **Outbound** tab. However, the rule will not be applied to the group until you click **Apply Rule Changes**, which you'll do after you've added all the outbound rules.

   c. Select HTTPS from the **Create a new rule** drop-down list and click **Add Rule**.
   The rule allowing outbound HTTPS access to anywhere (i.e., 0.0.0.0/0) is added to the **Outbound** tab.

6. Click **Apply Rule Changes**.

The default rule is deleted, and the new outbound rules now apply to the security group.



1 Security Group selected

Security Group: sg-59fbe835

Create a new rule: Custom TCP rule

Port range:
(e.g., 80 or 49152-65535)

Destination: 0.0.0.0/0
(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

TCP Port (Service)	Destination	Action
80 (HTTP)	0.0.0.0/0	Delete
443 (HTTPS)	0.0.0.0/0	Delete

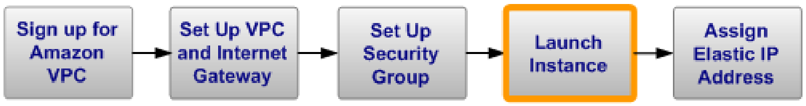
The VPC now includes a security group that you'll use when launching an instance in your next task. The group allows HTTP/HTTPS access in and out of the group to and from anywhere. The group also allows inbound SSH and RDP access from your home network's public IP address range. The group is not currently set up to enable instances inside the group to talk to each other. If you want that type of communication, you must add a rule to the security group to enable it. For more information about setting up security groups, go to Security Groups in the *Amazon Virtual Private Cloud User Guide*.

Tip

If you want another layer of security in addition to security groups, you can use *network ACLs*. Network ACLs control traffic at the subnet level. This exercise uses only security groups, which control traffic at the instance level. To learn more about network ACLs, go to Network ACLs in the *Amazon Virtual Private Cloud User Guide*.

The next task is to launch an instance into your subnet using the new security group you just set up.

Launch an Instance

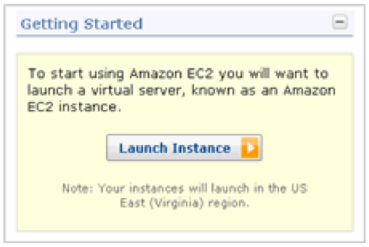


You're now ready to launch an instance into your subnet.

> **Tip**
>
> If you're already an EC2 user and familiar with launching instances, the only difference in this task is that you must specify the VPC and subnet you want to launch the instance into. Also, for this exercise, you should specify WebServerSG as the security group for the instance.

To launch an instance

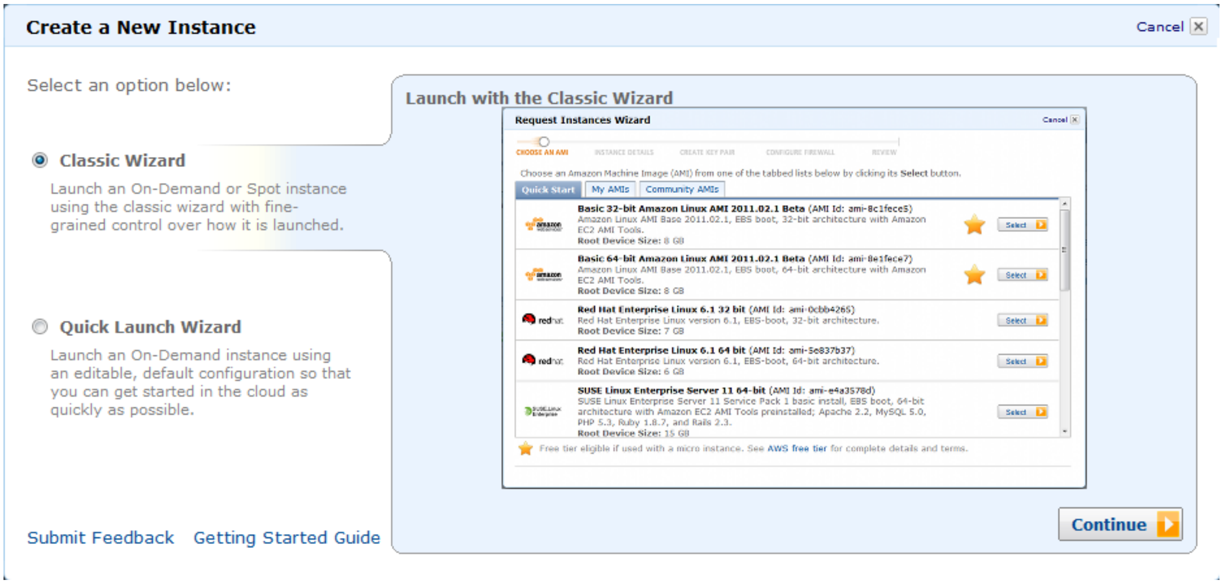
1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Click **Launch Instance**.


Getting Started

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US East (Virginia) region.

3. On the **Create a New Instance** screen, select **Classic Wizard**, and then click **Continue**.



Create a New Instance
Cancel ☒
Select an option below:
● Classic Wizard
Launch an On-Demand or Spot instance using the classic wizard with fine-grained control over how it is launched.
○ Quick Launch Wizard
Launch an On-Demand instance using an editable, default configuration so that you can get started in the cloud as quickly as possible.
Submit Feedback
Getting Started Guide
Launch with the Classic Wizard
Request Instances Wizard
Cancel ☒
CHOOSE AN AMI
INSTANCE DETAILS
CREATE KEY PAIR
CONFIGURE FIREWALL
REVIEW
Quick Start
My AMIs
Community AMIs
Select
Continue

Choose an Amazon Machine Image (AMI) from one of the tabbed lists below by clicking its **Select** button.

Basic 32-bit Amazon Linux AMI 2011.02.1 Beta (AMI Id: ami-8c1fece5)
Amazon Linux AMI Base 2011.02.1, EBS boot, 32-bit architecture with Amazon EC2 AMI Tools.
Root Device Size: 8 GB

Basic 64-bit Amazon Linux AMI 2011.02.1 Beta (AMI Id: ami-8e1fece7)
Amazon Linux AMI Base 2011.02.1, EBS boot, 64-bit architecture with Amazon EC2 AMI Tools.
Root Device Size: 8 GB

Red Hat Enterprise Linux 6.1 32 bit (AMI Id: ami-0cbb4265)
Red Hat Enterprise Linux version 6.1, EBS-boot, 32-bit architecture.
Root Device Size: 7 GB

Red Hat Enterprise Linux 6.1 64 bit (AMI Id: ami-5e837b37)
Red Hat Enterprise Linux version 6.1, EBS-boot, 64-bit architecture.
Root Device Size: 6 GB

SUSE Linux Enterprise Server 11 64-bit (AMI Id: ami-e4a3578d)
SUSE Linux Enterprise Server 11 Service Pack 1 basic install, EBS boot, 64-bit architecture with Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.0, PHP 5.3, Ruby 1.8.7, and Rails 2.3.
Root Device Size: 15 GB

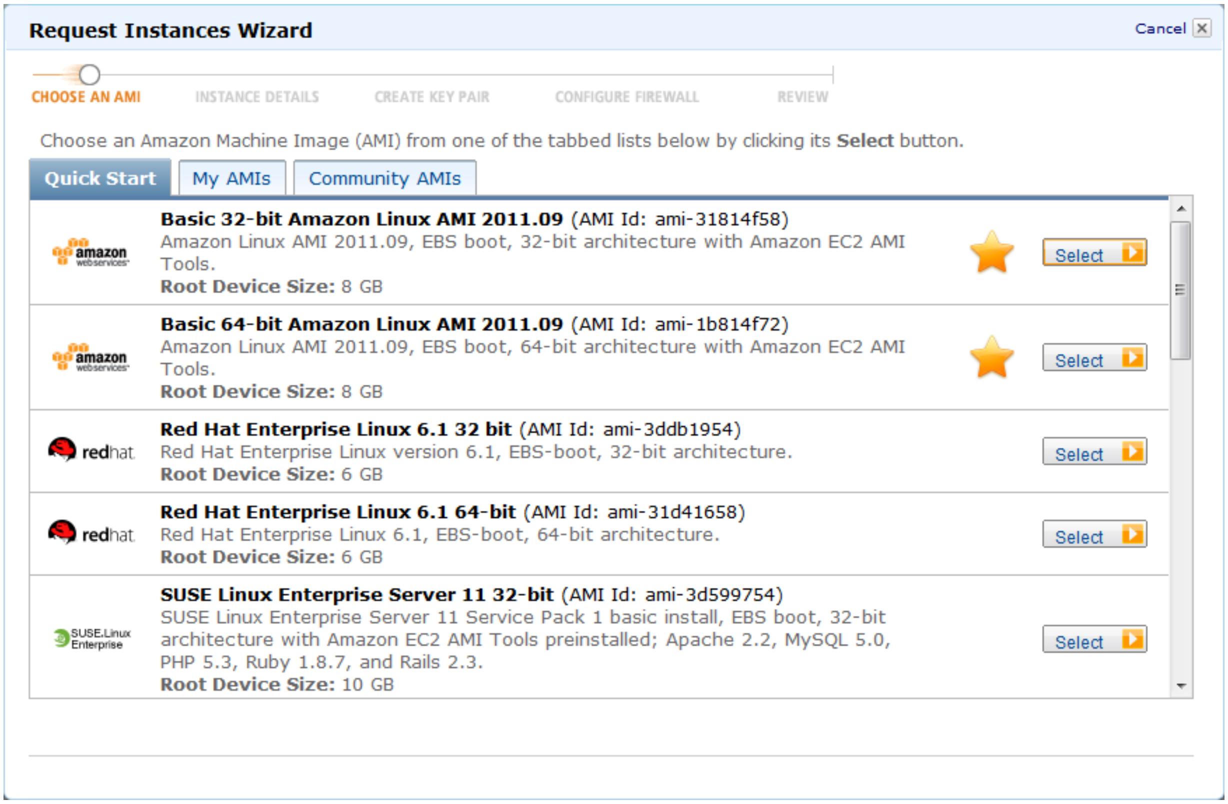
Free tier eligible if used with a micro instance. See AWS free tier for complete details and terms.

4. On the **Choose an AMI** page, on the **Quick Start** tab, select one of the Amazon Machine Images (AMIs).

   If you're not sure which to launch, select either the *Basic 32-bit Amazon Linux AMI*, or the *Getting Started on Microsoft Windows Server 2008* AMI.

   ### Note

   An AMI contains all the information needed to create a new instance of a server. For example, an AMI might contain all the software to act as a web server or all the software to act as a Windows database server (such as Windows Server and SQL Server).



Choose an Amazon Machine Image (AMI) from one of the tabbed lists below by clicking its **Select** button.

Basic 32-bit Amazon Linux AMI 2011.09 (AMI Id: ami-31814f58)
Amazon Linux AMI 2011.09, EBS boot, 32-bit architecture with Amazon EC2 AMI Tools.
Root Device Size: 8 GB

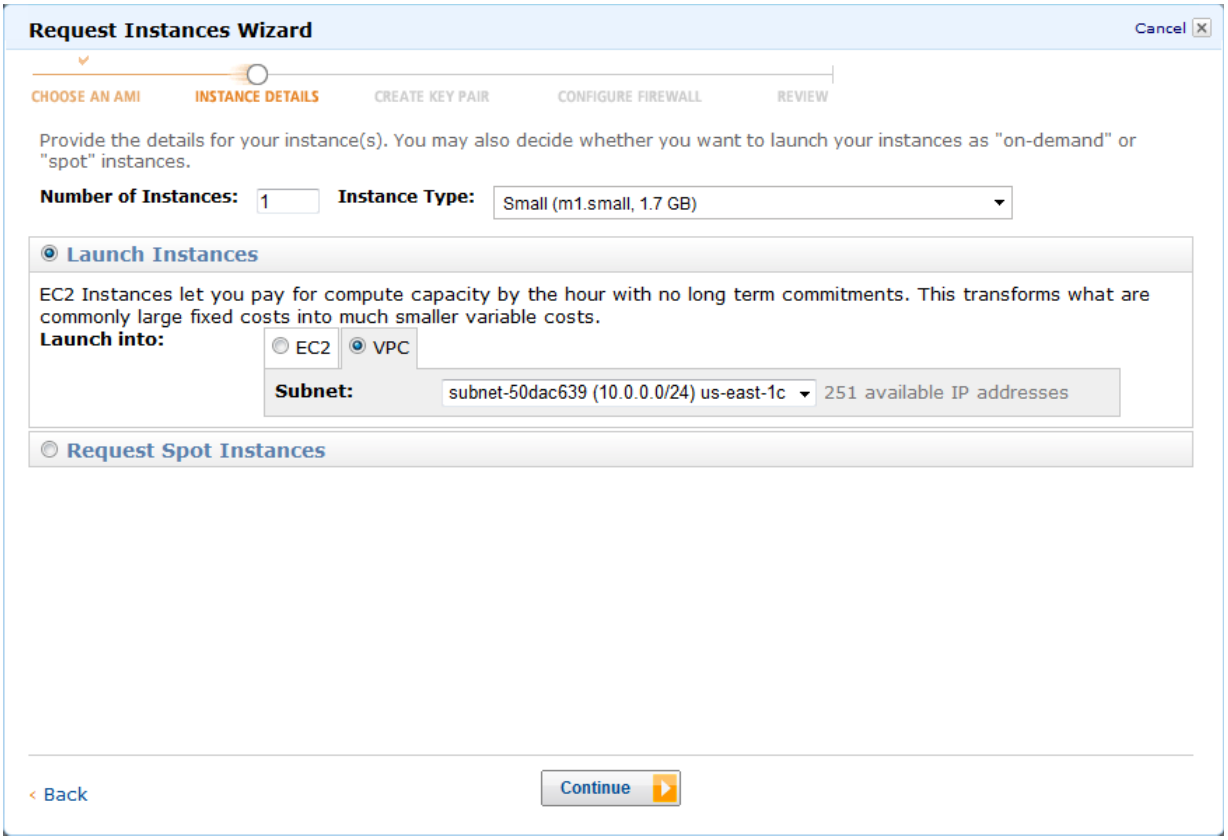
Basic 64-bit Amazon Linux AMI 2011.09 (AMI Id: ami-1b814f72)
Amazon Linux AMI 2011.09, EBS boot, 64-bit architecture with Amazon EC2 AMI Tools.
Root Device Size: 8 GB

Red Hat Enterprise Linux 6.1 32 bit (AMI Id: ami-3ddb1954)
Red Hat Enterprise Linux version 6.1, EBS-boot, 32-bit architecture.
Root Device Size: 6 GB

Red Hat Enterprise Linux 6.1 64-bit (AMI Id: ami-31d41658)
Red Hat Enterprise Linux 6.1, EBS-boot, 64-bit architecture.
Root Device Size: 6 GB

SUSE Linux Enterprise Server 11 32-bit (AMI Id: ami-3d599754)
SUSE Linux Enterprise Server 11 Service Pack 1 basic install, EBS boot, 32-bit architecture with Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.0, PHP 5.3, Ruby 1.8.7, and Rails 2.3.
Root Device Size: 10 GB

5. On the **Instance Details** page, in the **Instance Type** drop-down list box, select **Small (m1.small, 1.7 GB)** to launch a single small instance into your subnet.



Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

Number of Instances: 1 **Instance Type:** Small (m1.small, 1.7 GB)

● Launch Instances

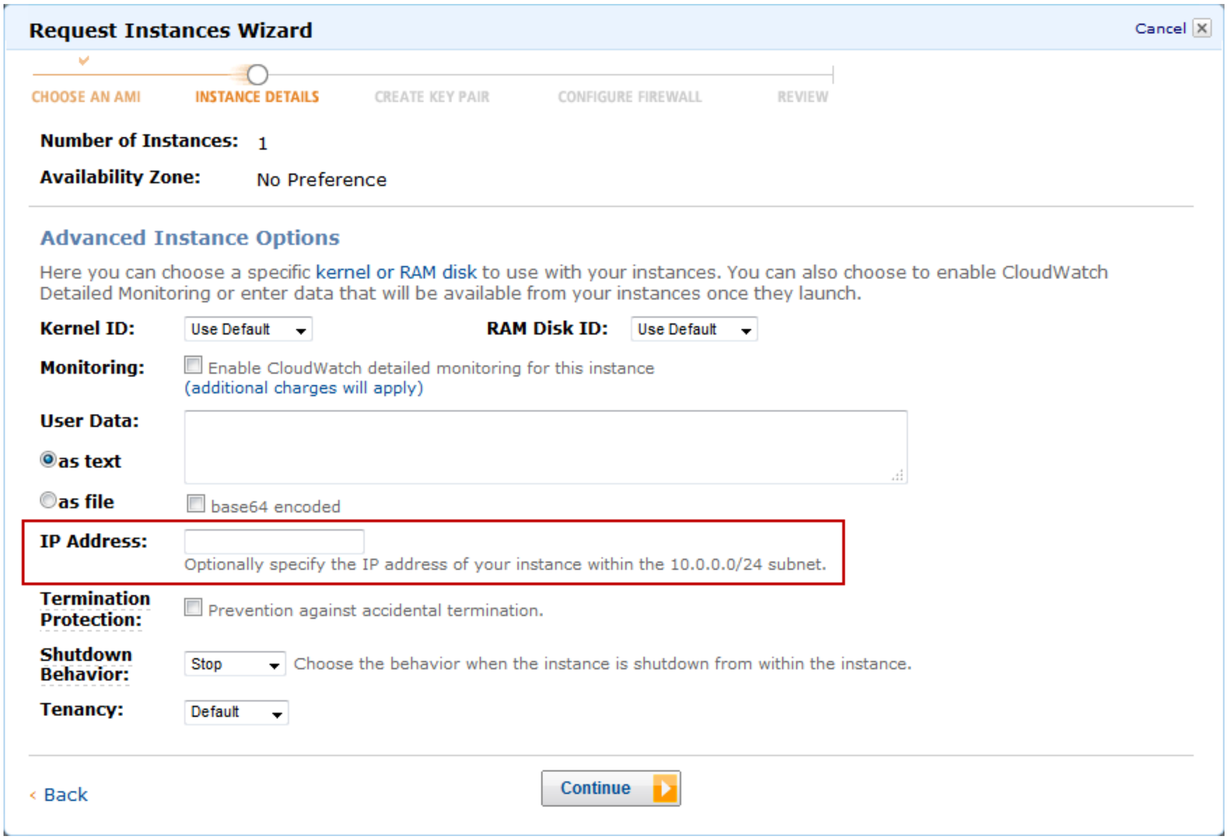
EC2 Instances let you pay for compute capacity by the hour with no long term commitments. This transforms what are commonly large fixed costs into much smaller variable costs.

Launch into: ○ EC2 ● VPC

Subnet: subnet-50dac639 (10.0.0.0/24) us-east-1c 251 available IP addresses

○ Request Spot Instances

6. Under **Launch Instances**, click the **VPC** tab, confirm that your subnet is selected in the **Subnet** drop-down list box, and then click **Continue**.
7. On the **Advanced Instance Options** page, you can specify the IP address to use for the instance. For this exercise, however, we'll leave the **IP Address** empty and click **Continue** to accept the default settings.



Request Instances Wizard

Number of Instances: 1

Availability Zone: No Preference

Advanced Instance Options

Here you can choose a specific kernel or RAM disk to use with your instances. You can also choose to enable CloudWatch Detailed Monitoring or enter data that will be available from your instances once they launch.

Kernel ID: Use Default

RAM Disk ID: Use Default

Monitoring: □ Enable CloudWatch detailed monitoring for this instance (additional charges will apply)

User Data:

● as text

○ as file

□ base64 encoded

IP Address:

Optionally specify the IP address of your instance within the 10.0.0.0/24 subnet.

Termination Protection: □ Prevention against accidental termination.

Shutdown Behavior: Stop — Choose the behavior when the instance is shutdown from within the instance.

Tenancy: Default

8. Give your instance a user friendly name or "tag," and then click **Continue**.
9. On the **Create Key Pair** page, you can select an existing key pair or create a new one.

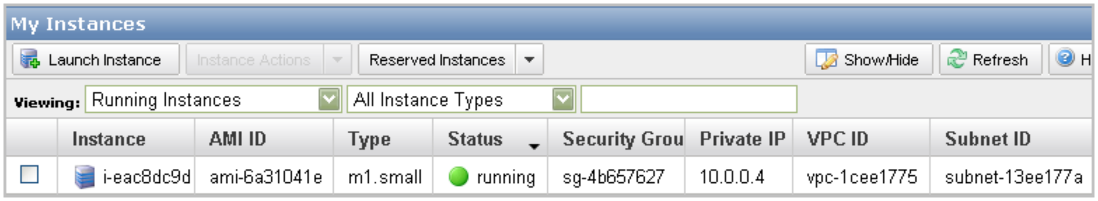
   A *key pair* is a security credential similar to a password, which you use to securely connect to your instance once it's running. If you're new to Amazon EC2 and haven't created any key pairs, when the wizard displays the **Create Key Pair** page, the **Create a new Key Pair** button is selected by default. You'll need to create a key pair.
10. Select **Create a new Key Pair**, enter a name for your key pair (e.g., `VPC_Keypair`), and then click **Create & Download your Key Pair**.

    Note

    EC2 uses this name to also name the private key file (with a `.pem` extension) associated with the pair.
11. When prompted, save the private key in a safe place on your system and click **Continue**.
12. On the **Configure Firewall** page, click **Choose one or more of your existing Security Groups**, select the WebServerSG group you created earlier, and then click **Continue**.
13. On the **Review** page, review your settings, and then click **Launch** to launch the instance. A confirmation page lets you know your instance is launching.
14. On the confirmation page, click **Close**, and then click **Instances** in the navigation pane to view your instance's status. It takes a short time for an instance to launch. The instance's status is *pending* while it's launching. After a short period, your instance's status switches to *running*. You can click **Refresh** to refresh the display.

Note

You might need to click **Show/Hide** in the top right corner to enable the display of the VPC ID, subnet ID, and private IP address of the instance in the list of instances.



Viewing: Running Instances | All Instance Types

	Instance	AMI ID	Type	Status	Security Grou	Private IP	VPC ID	Subnet ID
□	i-eac8dc9d	ami-6a31041e	m1.small	● running	sg-4b657627	10.0.0.4	vpc-1cee1775	subnet-13ee177a

At the bottom of the pane, you can view information about the Amazon EC2 instance.

You now have an instance running in your VPC. The next task is to assign it an Elastic IP address.

Assign an Elastic IP Address

Sign up for Amazon VPC
Set Up VPC and Internet Gateway
Set Up Security Group
Launch Instance
Assign Elastic IP Address

You're now ready to assign your running instance an Elastic IP address. This gives your normally private instance a public IP address so it can be reached from the Internet.

For this task, you first allocate an Elastic IP address to your VPC, and then you associate the address with the instance.

> **Important**
>
> Elastic IP addresses are an existing concept used in Amazon EC2. However, the addresses you use in your VPC are different from the ones you use in EC2. If you're already an EC2 user, you can't use your existing Elastic IP addresses in your VPC. You must allocate new ones specifically for use in your VPC. AWS has a limit on the number of EC2 addresses you can have, and there is a separate limit for the number of VPC addresses. To request to increase your limit, go to the Amazon VPC Limits Form.
>
> There are other differences between the two types of addresses. For more information, go to Elastic IP Addresses in the *Amazon Virtual Private Cloud User Guide*.

To allocate and assign a VPC Elastic IP address to an instance

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the **Navigation** pane, click **Elastic IPs**.
3. Click **Allocate New Address**.
   The **Allocate New Address** dialog box opens.
4. From the **EIP used in:** drop-down list, select `VPC` and click **Yes, Allocate**.
   The new address is allocated and is listed on the page.
5. Select the IP address in the list and click **Associate Address**.
   The **Associate Address** dialog box opens.
6. Select the instance you want to associate the address with and click **Yes, Associate**.
   The address is associated with the instance. Notice that the instance ID is displayed next to the IP address in the list.

Your instance now has an Elastic IP address associated with it. The instance is now accessible from the Internet. You can also access it using SSH or Remote Desktop from your home network. For instructions on using SSH to connect to a Linux/UNIX instance, go to Connect to Your Linux/UNIX Instance in the *Amazon Elastic Compute Cloud Getting Started Guide*. For instructions on using RDP to connect to a Windows instance, go to Connect to Your Windows Instance.

Important

When you use SSH or RDP to connect to the instance over the Internet, you must specify the instance's Elastic IP address as the address to connect to.

Summary of Your Setup

You now have a VPC with a public subnet, and a single instance running in the VPC. That instance has an Elastic IP address and therefore is reachable from the Internet. The VPC has a custom route table with a route that enables the instance to send traffic to the Internet. The instance is in a security group that enables HTTP and HTTPS traffic to flow in and out of the instance and enables SSH and RDP traffic to come in. This scenario is appropriate for a blog or other simple website that doesn't need a fleet of database servers running in a separate, private subnet. If you need a more complex setup, go to the next section, where you'll find a description of other scenarios for using Amazon VPC.

Where Do You Go From Here?

Topics

If you stepped through the preceding example, you've set up a VPC and launched an instance into it. Where do you go from here? This section describes other information you need to know about using Amazon VPC.

Optional Next Step: Deleting the VPC

If you completed the exercise in this guide, then you have a VPC with an Internet gateway and a running instance. If you'd like, you can keep the VPC and continue working with it. Or you might want to try another one of the options the Amazon VPC wizard offers.

If you'd like to delete the VPC, you must terminate the instance, and then delete the VPC and all of its related components (the Elastic IP address, the running instance, the security group, the subnet, and the Internet gateway).

To delete your VPC

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the **Navigation** pane, click **Instances**.
3. Right-click the instance running in the VPC and select **Terminate**.
   A confirmation dialog box appears.
4. Click **Yes, Terminate**.
5. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
6. In the **Navigation** pane, click **Your VPC**.
7. Select the VPC and click **Delete**.

A confirmation dialog box appears.

8. Click **Yes, Delete**.
   We begin deleting your VPC and its components. The dialog box displays the progress of the deletion.

you can manually create a new VPC or use one of the other options in the Amazon VPC wizard to create a new VPC. To get information about the other wizard options, see the next section.

Other Scenarios for Using Amazon VPC

This guide presents a simple scenario for using Amazon VPC. There might be others that fit your needs better. Following is a summary of the scenarios we cover in the Amazon VPC technical documentation and in the wizard. For a detailed discussion of the other scenarios, go to Amazon Virtual Private Cloud User Guide.

> **Note**
>
> The following scenarios are common ones we chose to present; you can configure subnets and other Amazon VPC components in other ways to suit your needs.

- Scenario 1: VPC with a Public Subnet Only
  We recommend this scenario if you want to run a single-tier, public-facing web application such as a blog or simple website.

  This is the scenario covered in the exercise in this guide. For a more detailed discussion, go to Scenario 1: VPC with a Single Public Subnet Only in the *Amazon Virtual Private Cloud User Guide*.
- Scenario 2: VPC with Public and Private Subnets
  We recommend this scenario if you want to run a public-facing web application while still maintaining non-publicly accessible backend servers in a second subnet.

  For more information, go to Scenario 2: VPC with Public and Private Subnets in the *Amazon Virtual Private Cloud User Guide*.
- Scenario 3: VPC with Public and Private Subnets and Hardware VPN Access
  We recommend this scenario if you want to extend your data center into the cloud and also directly access the Internet from your VPC. This scenario requires you to configure a gateway appliance in your home network to establish a VPN connection to the VPC.

  For more information, go to Scenario 3: VPC with Public and Private Subnets and VPN Access in the *Amazon Virtual Private Cloud User Guide*.
- Scenario 4: VPC with a Private Subnet Only and Hardware VPN Access
  We recommend this scenario if you want to extend your data center into the cloud and leverage Amazon's elasticity without exposing your network to the Internet. This scenario requires you to configure a gateway appliance in your home network to establish a VPN connection to the VPC.

  For more information, go to Scenario 4: VPC with a Private Subnet Only and VPN Access in the *Amazon Virtual Private Cloud User Guide*.

Current Service Limitations

With the current implementation of Amazon VPC:

- You can have up to five (5) VPCs per account per Region.
- You can have up to five (5) Amazon VPC Elastic IP Addresses per AWS account per Region.
- You can create up to twenty (20) subnets per Amazon VPC.

- Once you create a VPC or subnet, you can't change its IP address range.
- If you plan to have a VPN connection to your VPC, then you can have up to five virtual private gateways per AWS account per Region (one per VPC), with up to ten VPN connections per virtual private gateway.
- You can't use either broadcast or multicast within your VPC.
- CC1 and t1.micro instances do not work with a VPC.
- Amazon ElastiCache is not available for use in a VPC at this time.
- AWS Elastic Beanstalk is not available with your instances in a VPC.
- Amazon DevPay paid AMIs do not work with a VPC.

For more information about VPC limits, see Appendix B: Limits in the Amazon Virtual Private Cloud User Guide.

Other Ways to Access Amazon VPC

This guide has shown you how to use Amazon VPC through the AWS Management Console. You can continue using the service through the console, or you can try one of the other interfaces.

Continue Using the Console

To learn more about how to use Amazon VPC through the console, go to the Amazon Virtual Private Cloud User Guide. The console also has online Help to assist you (just click the **Help** button in the console).

Use the Command Line Interface

Amazon EC2 and Amazon VPC share a Java-based command line interface. These command line tools are a fast way to execute all the Amazon EC2 and Amazon VPC functions without coding to the API or using a library. For information about getting started with the command line tools, go to Getting Started with the Command Line Tools in the *Amazon Elastic Compute Cloud User Guide*. For a complete description of all the commands, go to the Amazon Elastic Compute Cloud Command Line Reference.

Use an Existing Library

Amazon EC2 and Amazon VPC share the same programmatic interface. If you prefer to use Amazon VPC through an API, there are libraries and resources available for the following languages:

- Java
- PHP
- Python
- Ruby
- Windows and .NET

For libraries and sample code in all languages, go to Amazon EC2 Sample Code & Libraries.

Code Directly to the Web Service API

If you want to write code directly to the Amazon EC2 and VPC API, go to Making API Requests in the *Amazon Elastic Compute Cloud User Guide*. The guide describes how to create and authenticate API requests, and how to use Amazon EC2 and Amazon VPC through the APIs. For a complete description of all the API actions, go to the Amazon Elastic Compute Cloud API Reference.

AWS Account and Security Credentials

This guide walked you through signing up for the service, getting an AWS account, and then completing a short exercise. Now that you've completed the exercise, we recommend you check with an administrator or co-worker in your organization to determine if they already have an AWS account and security credentials for you to use in future interactions with AWS.

If you're an account owner or administrator and would like to know more about AWS Identity and Access Management, go to http://aws.amazon.com/iam and to Using AWS Identity and Access Management.

Supporting Documentation

This Getting Started Guide covers the basic commands you can use with Amazon VPC. The following table summarizes the other available documentation for Amazon VPC.

Description	Documentation
Complete discussion of how to use Amazon VPC	Amazon Virtual Private Cloud User Guide
Information about configuring the customer gateway (if you decide to use a VPN connection with your VPC)	Amazon Virtual Private Cloud Network Administrator Guide
A hands-on introduction to Amazon EC2	Amazon Elastic Compute Cloud Getting Started Guide
Complete discussion of how to use Amazon EC2	Amazon Elastic Compute Cloud User Guide
Complete descriptions of all the Amazon VPC and Amazon EC2 commands	Amazon Elastic Compute Cloud Command Line Reference
Quick reference descriptions of the commonly used Amazon VPC commands	Amazon Virtual Private Cloud Quick Reference Card
Complete descriptions of the Amazon VPC and Amazon EC2 API operations, data types, and errors	Amazon Elastic Compute Cloud API Reference

Where to Get Additional Help

We recommend that you take advantage of the AWS Discussion Forums. These are community-based forums for users to discuss technical questions related to AWS services. For the Amazon VPC forum, go to https://forums.aws.amazon.com/forum.jspa?forumID=58.

You can also get help if you subscribe to AWS Premium Support, a one-on-one, fast-response support channel (for more information, go to http://aws.amazon.com/premiumsupport).

Please Provide Feedback

Your input is important to help make our documentation helpful and easy to use. Please tell us about your experience getting started with Amazon VPC by completing our Getting Started Survey.

Thank you.

Document History

This documentation is associated with the 2012-08-15 release of Amazon Virtual Private Cloud. This guide was last updated on 13 September 2012.

The following table describes the important changes since the last release of the Amazon VPC documentation set.

Change	Description	Release Date
VPC Everywhere	With this release, the getting started guide has been rewritten to reflect the new features available in the 2011-07-15 API version.	In this release

About This Guide

This is the *Amazon Virtual Private Cloud Getting Started Guide*. This guide was last updated on September 13, 2012.

Amazon Virtual Private Cloud is often referred to within this guide as "Amazon VPC." All copyrights and legal protections still apply.