Amazon Virtual Private Cloud

Getting Started Guide

API Version 2014-06-15

Amazon Virtual Private Cloud: Getting Started Guide

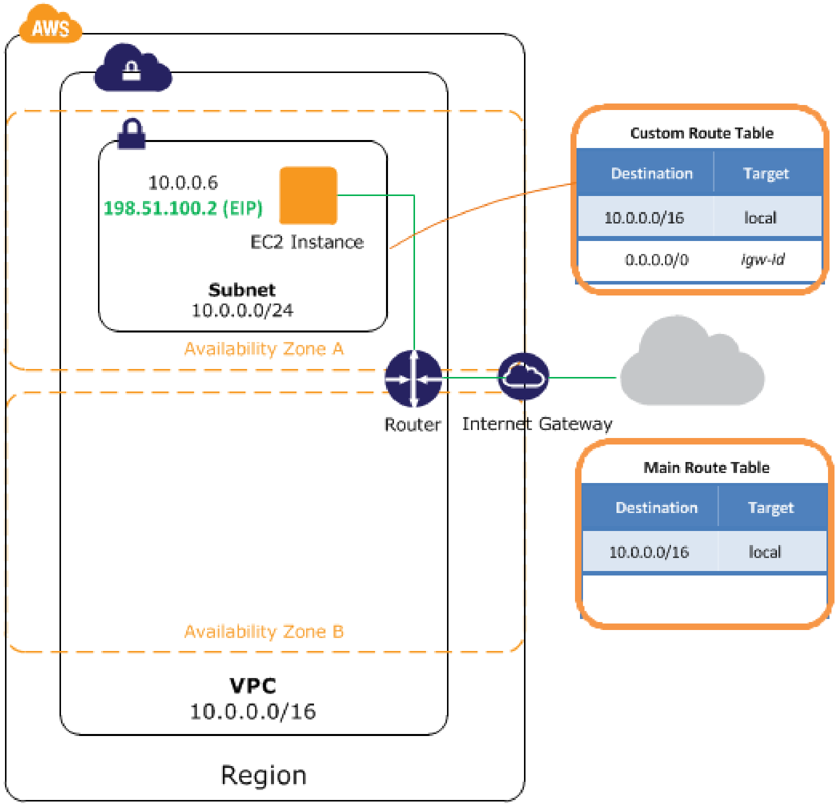
Table of Contents

Overview of the Exercise

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 Virtual Private Cloud 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).



The following table summarizes the tasks that you'll perform as you complete the exercise in this guide.

Step 1: Set Up the VPC and Internet Gateway (p. 4)	Use the VPC Wizard to do the following for you: 1. Create a VPC, which is an isolated portion of the AWS cloud. 2. Create and attach an Internet gateway, which connects your VPC directly to the Internet and provides access to other AWS products, 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 can 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. Your VPC has an implied router, as pictured in the diagram.
Step 2: Set Up a Security Group for Your VPC (p. 6)	Use the Amazon VPC console to set up a security group to control the inbound and outbound traffic for the instance.
Step 3: Launch an Instance into Your VPC (p. 9)	Use the Amazon EC2 console to launch an instance into the subnet. The instance gets a private IP address from the subnet's range of addresses.

Step 4: Assign an Elastic IP Address to Your Instance (p. 10)	Use the Amazon VPC console to allocate an Elastic IP address and assign it to the instance. An Elastic IP address provides the instance with a public IP address in addition to its private address, so that the instance is reachable from the Internet.
Optional Step: Delete the VPC (p. 11)	Use the Amazon EC2 console to terminate your instance and the Amazon VPC console to delete your VPC.

Getting Started with Amazon VPC

This guide provides 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 EC2 instance with an Elastic IP address.

> **Important**
> 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 click **Sign Up**.

Topics

For an overview of the exercise, see Overview of the Exercise (p. 1). For a basic overview of Amazon VPC, see What is Amazon VPC? in the *Amazon Virtual Private Cloud User Guide*.

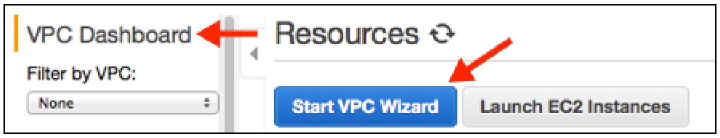
Step 1: Set Up the VPC and Internet Gateway

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

- Create a size /16 VPC (a network with 65,536 private IP addresses).
- Attach an Internet gateway to the VPC.
- Add a size /24 subnet (a range of 256 private IP addresses).
- Set up routing for your VPC so that traffic can flow between the subnet and the Internet gateway.

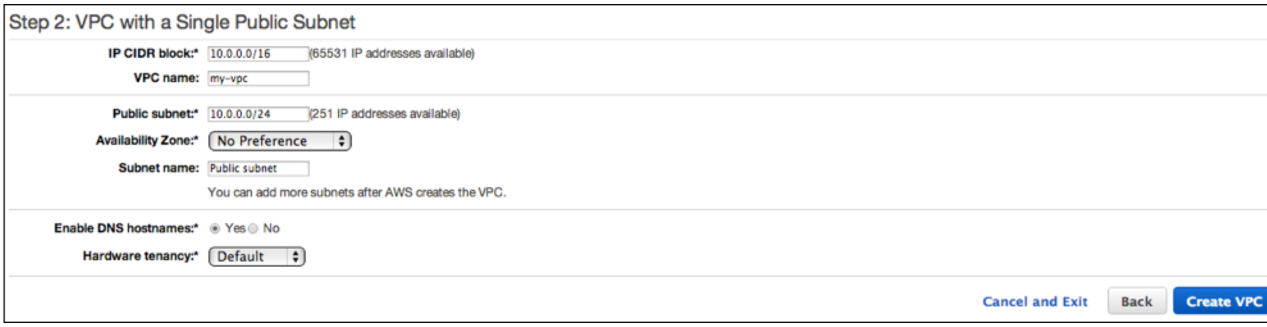
To create a VPC using the VPC Wizard in the AWS Management Console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. On the VPC dashboard, click **Start VPC Wizard**.



3. Select the first option, **VPC with a Single Public Subnet**, and click **Select**.
4. The confirmation page shows the CIDR ranges that we'll use for your VPC and subnet (10.0.0.0/16 and 10.0.0.0/24, respectively), and the hardware tenancy setting. The confirmation page also displays the subnet's Availability Zone. Make any changes to these settings that you need, and then click **Create VPC** to create your VPC, Internet gateway, subnet, and route table.

   Note
   For information about CIDR notation, see the Wikipedia article about Classless Inter-Domain Routing. For more information about hardware tenancy, see Using EC2 Dedicated Instances in the *Amazon Virtual Private Cloud User Guide*.



Step 2: VPC with a Single Public Subnet

IP CIDR block:* 10.0.0.0/16 (65531 IP addresses available)
VPC name: my-vpc

Public subnet:* 10.0.0.0/24 (251 IP addresses available)
Availability Zone:* No Preference
Subnet name: Public subnet
You can add more subnets after AWS creates the VPC.

Enable DNS hostnames:* ● Yes ○ No
Hardware tenancy:* Default

5. A status window shows the work in progress. When the work completes, a status window confirms that your VPC has been successfully created. Click **OK** to close the status window and return to the VPC dashboard.

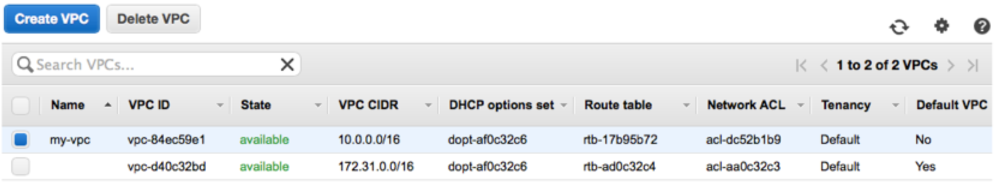
Verify Your VPC's Components

You can use the default settings for the components that the VPC Wizard created for you as you go through the exercise in this guide. This section describes how you can view these components and their settings using the VPC console.

In the navigation pane, click **Your VPCs**, and then select the VPC that you just created (**Default VPC** is `No`).

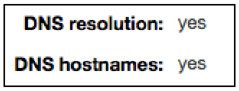
Tip
You might need to refresh the page for the VPC to appear.

The console displays your default VPC and the VPC that you just created. Each VPC has a set of DHCP options, a main route table, and default network ACL. For more information, see DHCP Options Sets, Route Tables, and Network ACLs in the *Amazon Virtual Private Cloud User Guide*.



□	Name	VPC ID	State	VPC CIDR	DHCP options set	Route table	Network ACL	Tenancy	Default VPC
■	my-vpc	vpc-84ec59e1	available	10.0.0.0/16	dopt-af0c32c6	rtb-17b95b72	acl-dc52b1b9	Default	No
□		vpc-d40c32bd	available	172.31.0.0/16	dopt-af0c32c6	rtb-ad0c32c4	acl-aa0c32c3	Default	Yes

The console displays the DNS settings for the VPC in the details pane. For more information about these DNS settings, see Using DNS with Your VPC in the *Amazon Virtual Private Cloud User Guide*.



DNS resolution: yes

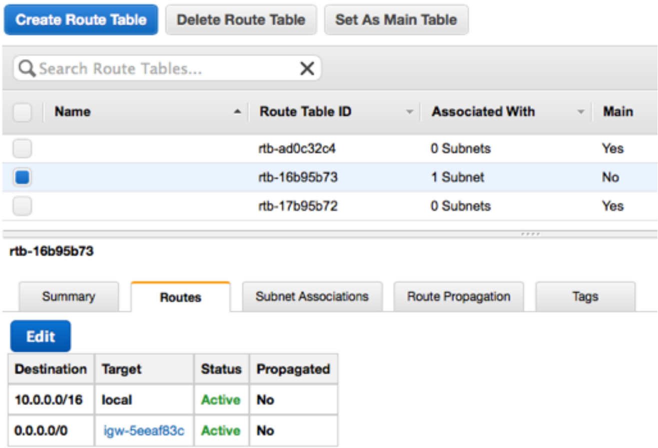
DNS hostnames: yes

To display information about your Internet gateways, click **Internet Gateways** in the navigation pane. You have one Internet gateway for your default VPC, and another for the VPC that you just created.

The VPC that you just created has two route tables. The VPC came with a main route table by default, and the VPC Wizard created a custom route table in addition. Your subnet is associated with the custom route table, which means that we use the routes in that table to 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 your route tables

1. In the navigation pane, click **Route Tables**.
2. Select the custom route table (the **Main** column has **No**), and then click the **Routes** tab to display the route information in the details pane.



	Name	Route Table ID	Associated With	Main
□		rtb-ad0c32c4	0 Subnets	Yes
■		rtb-16b95b73	1 Subnet	No
□		rtb-17b95b72	0 Subnets	Yes

rtb-16b95b73

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	igw-5eeaf83c	Active	No

3. The first row in the table is the local route, which enables instances within the VPC to communicate. 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 VPC wizard added to enable traffic destined for an IP address outside the VPC (0.0.0.0/0) to flow from the subnet to the Internet gateway. We refer to this subnet as a *public subnet* because all traffic from the subnet goes to the Internet gateway.
4. Select the main route table. The main route table has a local route, but no other routes. Therefore, any subnet you create is not exposed to the Internet initially, it's a *private subnet*. To expose a new subnet as a public subnet, you can either change the routing in the main route table, or associate the subnet with a custom route table.

Step 2: Set Up a Security Group for Your VPC

A *security group* acts as a virtual firewall to control the traffic allowed into its associated instances. To use security groups, you create a group, add the inbound and outbound rules that you want to use, and then associate your instances with the security group when you launch them. 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 is associated with the default security group. Although we could use the default security group for this exercise, we've chosen to create a security group, `WebServerSG`, instead. You'll specify this security group when you launch an instance into your VPC.

Topics

Rules for the WebServerSG Security Group

Inbound rules regulate the traffic that is allowed to reach the instances associated with the security group (the source of the traffic and the listening port on the instance). All return traffic is automatically allowed to reach the instances. For example, if a client on the Internet sends a request to a web server in your VPC associated with `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 associated with the security group can send traffic to (the destination of the traffic and the destination port). All return traffic (such as a response from the host that received the traffic) is automatically allowed to reach the instances, regardless of the inbound rules set on the security group.

The following table describes the inbound rules for the `WebServerSG` security group. Because the web server doesn't initiate outbound communication, we'll remove the default outbound rule.

> **Note**
> If your company uses only Linux or only Windows, you don't have to add access for both SSH and RDP.

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 (Linux/UNIX only)
Public IP address range of your home network	TCP	3389	Allow inbound RDP access from your home network (Windows only)

In this exercise, you won't add a rule to enable instances associated with the security group to talk to each other. To enable this type of communication, you must add a rule to the security group for this purpose. For more information, see Security Groups in the *Amazon Virtual Private Cloud User Guide*.

Creating Your WebServerSG Security Group

To create the WebServerSG security group

1. Open the Amazon VPC console.
2. Click **Security Groups** in the navigation pane.
3. Click the **Create Security Group** button.
4. Enter `WebServerSG` as the name of the security group, and provide a description. Select the ID of your VPC from the **VPC** menu, and then click **Yes, Create**.

By default, new security groups start 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.

Adding Rules to Your WebServerSG Security Group

To add rules to the WebServerSG security group

1. Click **Security Groups** in the navigation pane to display your security groups.
2. Select the `WebServerSG` security group that you just created. The details pane include a tab for information about the security group, plus tabs for working with its inbound rules and outbound rules.
3. On the **Inbound Rules** tab, click **Edit** and add rules for inbound traffic as follows:

   a. Select **HTTP** from the **Type** list, and enter `0.0.0.0/0` in the **Source** field.

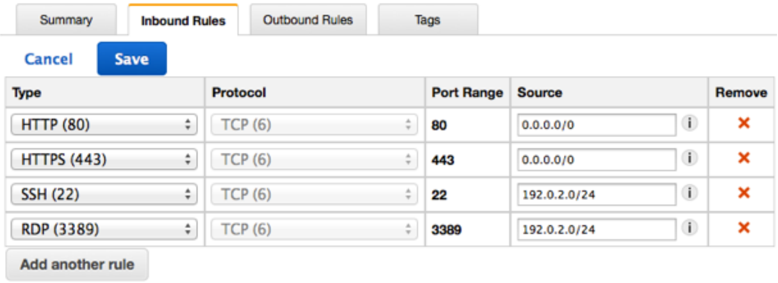
   b. Click **Add another rule**, then select **HTTPS** from the **Type** list, and enter `0.0.0.0/0` in the **Source** field.

   c. Click **Add another rule**, then select **SSH** 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.

   d. Click **Add another rule**, then 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.

   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.



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	✕

   e. Click **Save**.

4. Limit outbound access to responses by removing the default outbound rule:

   a. On the **Outbound Rules** tab, locate the default rule that enables all outbound traffic, and click **Remove**.
   b. Click **Save**.

Step 3: Launch an Instance into Your VPC

When you launch an EC2 instance into a VPC, you must specify the ID of a subnet in the VPC.

To launch an EC2 instance into a VPC

1. Open the Amazon EC2 console.
2. From the navigation bar, select the region for the instance. For this exercise, you can use the default region. For more information about regions and Availability Zones, see Regions and Availability Zones in the *Amazon Elastic Compute Cloud User Guide*.
3. From the dashboard, click the **Launch Instance** button.
4. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations called Amazon Machine Images (AMIs). Choose the AMI that you want to use and click its **Select** button.
5. On the **Choose an Instance Type** page, 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 stay within the free tier by selecting the **t2.micro** instance type. When you are ready, click **Next: Configure Instance Details**.
6. On the **Configure Instance Details** page, select your VPC from the **Network** list, and subnet from the **Subnet** list. You can also configure other options as follows, and then click **Next: Add Storage**:

   - **Public IP**: Select this check box to request that your instance receives a public IP address. For more information, see IP Addressing in Your VPC. If you don't assign a public IP address during launch, you can assign an Elastic IP address to your instance after launch, which is explained in the next step of this exercise.
   - **Network interfaces**: If you selected a nondefault subnet, you can specify up to two new or existing network interfaces in the wizard. You can also specify a private IP address to use for the instance, or leave the **Primary IP** field blank to let the wizard assign one for you.

7. On the **Add Storage** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume). Click **Next: Tag Instance** when done.
8. On the **Tag Instance** page, specify tags for the instance by providing key and value combinations. Click **Next: Configure Security Group** when you are done.
9. On the **Configure Security Group** page, the wizard automatically defines the launch-wizard-*x* security group to allow you to connect to your instance. Select the **Select an existing security group** option, and select the **WebServerSG** group that you created previously,. When you are done, click **Review and Launch**.
10. On the **Review Instance Launch** page, check the details of your instance, and make any necessary changes by clicking the appropriate **Edit** link.

    When you are ready, click **Launch**.
11. 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 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, then click **Launch Instances**.

Step 4: Assign an Elastic IP Address to Your Instance

By default, an instance in a nondefault VPC is not assigned a public IP address, and is private. You can make an instance in a nondefault VPC public by attaching an Internet gateway to the VPC and providing the instance with a public IP address. In this exercise, you created an Internet gateway for your VPC using the VPC wizard. You accepted the default settings in the launch wizard, so you did not receive a public IP address. Now, you'll create an Elastic IP address, which is a public IP address that belongs to your AWS account, and associate it with your instance to make it accessible from the Internet.

For more information about Elastic IP addresses, see Elastic IP Addresses in the *Amazon Virtual Private Cloud User Guide*.

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, click **Elastic IPs**.
3. Click the **Allocate New Address** button.
4. In the **Network platform** list, select **EC2-VPC**, and then click **Yes, Allocate**. Note that if your account supports EC2-VPC only, you will not have an option to select where your Elastic IP will be used.
5. Select the Elastic IP address from the list and click the **Associate Address** button.
6. In the **Associate Address** dialog box, do the following, and then click **Yes, Associate**:

   a. In the **Associate Address** dialog box, select **Instance** or **Network Interface** from the **Associate with** list, and then either the instance or network interface ID.

      Note
      A network interface can have several attributes, including an Elastic IP address. You can create a network interface and attach and detach it from instances in your VPC. The advantage of making the Elastic IP address an attribute of the network interface instead of associating it directly with the instance is that you can move all the attributes of the network interface from one instance to another in a single step. For more information, see Elastic Network Interfaces.

   b. Select the private IP address to associate the Elastic IP address with from the **Private IP address** list.

Your instance is now accessible from the Internet. You can also access the instance using SSH or Remote Desktop from your home network, specifying the Elastic IP address of the instance as the address to connect to. For instructions about how to connect to a Linux instance, see Connect to Your Linux Instance in the *Amazon Elastic Compute Cloud User Guide*. For instructions about how to connect to a Windows instance, see Connect to Your Windows Instance in the *Amazon Elastic Compute Cloud Microsoft Windows Guide*.

This completes the exercise; you've got your VPC set up, an instance running in your VPC, and the ability to connect to your instance from the Internet. You can continue to use your instance and your VPC, or you can terminate the instance and delete the VPC. For information about cleaning up, or finding additional VPC documentation, see Where Do I Go From Here? (p. 11).

Where Do I Go From Here?

If you stepped through the example in this guide, you've set up a VPC and launched an instance into it. If you'd like, you can keep using your VPC. Otherwise, you can delete the VPC. Whether you delete your VPC or not, you can also create additional VPCs.

Optional Step: Delete the VPC

Before you can delete a VPC, you must terminate any instances that are running in the VPC. Deleting a VPC 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 delete your VPC

1. Open the Amazon EC2 console.
2. Click **Instances** in the navigation pane.
3. Right-click the instance that's running in the VPC and select **Terminate**.
4. When prompted for confirmation, click **Yes, Terminate**.
5. Open the Amazon VPC console.
6. Click **Your VPCs** in the navigation pane.
7. Select the VPC and click the **Delete** button.
8. When prompted for confirmation, click **Yes, Delete**.

Additional Scenarios for Amazon VPC

The exercise in this guide covers the first scenario in the VPC wizard. For a detailed discussion of the other scenarios, see Scenarios for Amazon VPC in the *Amazon Virtual Private Cloud User Guide*.

Additional Documentation

This Getting Started Guide for Amazon VPC covers the basics of using Amazon VPC and Amazon EC2. The following table lists other documentation for Amazon VPC and Amazon EC2.

Description	Documentation
Information about how to use Amazon VPC	Amazon Virtual Private Cloud User Guide
Information about configuring your 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	Getting Started with Amazon EC2
Information about how to use Amazon EC2	Amazon Elastic Compute Cloud User Guide
Descriptions of the commands for Amazon EC2 and Amazon VPC	AWS Command Line Interface Reference (AWS CLI) Amazon Elastic Compute Cloud Command Line Reference (Amazon EC2 CLI)
Descriptions of the API actions, data types, and errors for Amazon EC2 and Amazon VPC.	Amazon Elastic Compute Cloud API Reference