What is cloud computing?

Cloud computing is the on-demand delivery of computing power, databases, storage, applications, and other IT resources through a cloud services platform via the Internet with pay-as-you-go pricing. Whether you're running applications that share photos with millions of mobile device users or supporting your company's core operations, a cloud services platform provides fast access to flexible, low-cost IT resources. With cloud computing, you don't need to make large upfront investments in hardware and spend a lot of time on the tedious task of managing that hardware. Instead, you can provision exactly the type and size of computing resources you need to power your brilliant new idea or manage your IT department. You can access as many resources as you need, almost instantly, and pay only for what you use.

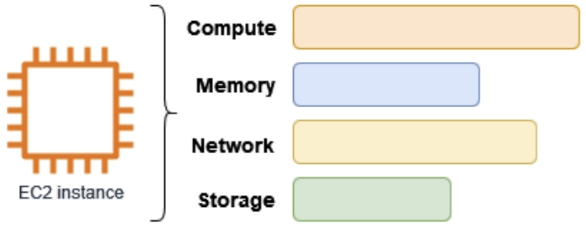
Cloud computing provides an easy way to access servers, storage, databases, and a wide range of application services over the Internet. A cloud services platform such as Amazon Web Services owns and maintains the network-connected hardware required for these application services, while you provision and use what you need through a web application.

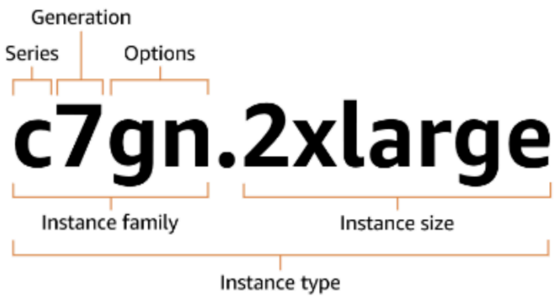
Elastic Compute Cloud

What is Amazon EC2?

Amazon Elastic Compute Cloud (Amazon EC2) provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 reduces hardware costs so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. You can add capacity (scale up) to handle compute-heavy tasks, such as monthly or yearly processes, or spikes in website traffic. When usage decreases, you can reduce capacity (scale down) again.

An EC2 instance is a virtual server in the AWS Cloud. When you launch an EC2 instance, the instance type that you specify determines the hardware available to your instance. Each instance type offers a different balance of compute, memory, network, and storage resources. For more information, see the Amazon EC2 Instance Types Guide.





1. What are the main features of Amazon EC2?
2. What services can be used with Amazon EC2?

Step 1: Launch an instance

You can launch an EC2 instance using the AWS Management Console as described in the following procedure. This tutorial is intended to help you quickly launch your first instance, so it doesn't cover all possible options.

To launch an instance

1. Open the Amazon EC2 console
2. In the navigation bar at the top of the screen, we display the current AWS Region — for example, Ohio. You can use the selected Region, or optionally select a Region that is closer to you.
3. From the EC2 console dashboard, in the Launch instance pane, choose Launch instance.
4. Under Name and tags, for Name, enter a descriptive name for your instance.
5. Under Application and OS Images (Amazon Machine Image), do the following: a. Choose Quick Start, and then choose the operating system (OS) for your instance. For your first Linux instance, we recommend that you choose Amazon Linux. b. From Amazon Machine Image (AMI), select an AMI that is marked Free Tier eligible.
6. Under Instance type, for Instance type, select an instance type that is marked Free Tier eligible.
7. Under Key pair (login), for Key pair name, choose an existing key pair or choose Create new key pair to create your first key pair.
8. Under Network settings, notice that we selected your default VPC, selected the option to use the default subnet in an Availability Zone that we choose for you, and configured a security group with a rule that allows connections to your instance from anywhere (0.0.0.0.0/0)

For your first instance, we recommend that you use the default settings. Otherwise, you can update your network settings as follows:

- (Optional) To use a specific default subnet, choose Edit and then choose a subnet.
- (Optional) To use a different VPC, choose Edit and then choose an existing VPC. If the VPC isn't configured for public internet access, you won't be able to connect to your instance.
- (Optional) To restrict inbound connection traffic to a specific network, choose Custom instead of Anywhere, and enter the CIDR block for your network.
- (Optional) To use a different security group, choose Select existing security group and choose an existing security group. If the security group does not have a rule that allows connection traffic from your network, you won't be able to connect to your instance. For a Linux instance, you must allow SSH traffic. For a Windows instance, you must allow RDP traffic.

9. Under Configure storage, notice that we configured a root volume but no data volumes. This is sufficient for test purposes.
10. Review a summary of your instance configuration in the Summary panel, and when you're ready, choose Launch instance.
11. If the launch is successful, choose the ID of the instance from the Success notification to open the Instances page and monitor the status of the launch.
12. Select the checkbox for the instance. The initial instance state is pending. After the instance starts, its state changes to running. Choose the Status and alarms tab. After your instance passes its status checks, it is ready to receive connection requests

Step 2: Connect to your instance

You can connect to your Linux instance using any SSH client. If you are running Windows on your computer, open a terminal and run the ssh command to verify that you have an SSH client installed. If the command is not found, install OpenSSH for Windows.

To connect to your instance using SSH

1. In the navigation pane, choose Instances.
2. Select the instance and then choose Connect.
3. On the Connect to instance page, choose the SSH client tab.
4. (Optional) If you created a key pair when you launched the instance and downloaded the private key (.pem file) to a computer running Linux or macOS, run the example chmod command to set the permissions for your private key.
5. Copy the example SSH command. The following is an example, where **key-pair-name.pem** is the name of your private key file, **ec2-user** is the username associated with the image, and the string after the @ symbol is the public DNS name of the instance.

```
ssh -i key-pair-name.pem ec2-user@ec2-198-51-100-1.us-east-2.compute.amazonaws.com
```

6. In a terminal window on your computer, run the ssh command that you saved in the previous step. If the private key file is not in the current directory, you must specify the fully-qualified path to the key file in this command.

The following is an example response:

```
The authenticity of host 'ec2-198-51-100-1.us-east-2.compute.amazonaws.com
(198-51-100-1)' can't be established. ECDSA key fingerprint is
l4UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY. Are you sure you want to continue
connecting (yes/no)?
```

7. (Optional) Verify that the fingerprint in the security alert matches the instance fingerprint contained in the console output when you first start an instance. To get the console output, choose Actions, Monitor and troubleshoot, Get system log. If the fingerprints don't match, someone might be attempting a man-in-the-middle attack. If they match, continue to the next step.
8. Enter yes.

Next steps

After you start your instance, you might want to explore the following next steps:

- What is the difference between stopping, terminating, and restarting an EC2 instance?
- What role does an AMI (Amazon Machine Image) play when launching an instance?
- In what cases would it be advisable to choose an AMI other than the default Amazon Linux AMI?
- How to add and attach volumes, distinguish between EBS and ephemeral storage?
- How to execute remote commands without SSH?
- What steps are required to attach an additional EBS volume to an existing Linux instance?
- What happens to the data on an EBS volume when the instance is stopped or terminated?
- Will the shell we configured in linux Slackware work in the instance we created? What do I need to change? Test its functionality.