# Introduction to instances in AWS

For many businesses and personal projects, maintaining a physical server is a costly endeavor with many security and recovery concerns. Instances are a cost-effective solution that is 100% virtual, secure, and remotely accessible. Amazon Web Services (AWS) is one of the best options for creating scalable instances based on your needs.

This guide will cover the basics of instances and why you should use AWS for instances. You will then learn how to use AWS to create, configure, and connect to a Windows instance. Only general knowledge of cloud computing and networking concepts like IP addresses, network traffic, and the command prompt is required to understand this guide.

## What is an instance?

An *instance* is a virtual computer that runs on the AWS Cloud. Instances look and act like any other computer; you can browse the internet, host other servers, run applications, etc. The only difference is that they’re entirely virtual; you can connect to an instance from your own computer, even if it uses a different operating system!

Instances offer a lot of benefits over physical servers and computers:

* **Cost-effective:** Instances eliminate the need to invest in hardware, staffing, or physical space to house a data center.
* **Scalable:** You can adjust capacity anytime, only paying for what you need. Physical data centers require guessing capacity, leading to spending too much money maintaining servers you don’t need, or not having enough servers.
* **Reliable:** In case of an outage, natural disaster, or other failures, instances are easily recoverable or replaceable as they're in the cloud. You can spend less time on maintenance and more on satisfying your customers.

## Why use AWS for instances?

AWS is a cloud computing service powered by Amazon. AWS uses the *Elastic Compute Cloud (EC2)* service to run and store instances. On top of instances, AWS offers a variety of free and paid cloud-computing products like VPCs, databases, APIs, and more.

Instances are the same as virtual machines offered by VMWare and the Google Cloud Platform (GCP). What makes AWS stand out is that it has more data centers around the globe, delivering greater accessibility and more storage and RAM. AWS is best for businesses looking to move away completely from physical data centers and legacy systems.

## Creating an instance

You can create and configure instances through the *EC2 Dashboard.* To create a Windows instance:

1. Type **EC2** into the search bar. Click the **EC2** result that should be the first to appear under **Services**. You are now in the EC2 Dashboard.
2. Under the **Instances** dropdown on the left side menu, click **Instances**. This is where you'd see each of your instances listed as a row in a table.
3. Click **Launch instances**. Give your instance a name.
4. In the **Application and OS Images** section, select an operating system for your instance. Each OS offers multiple free-tier and paid versions to choose from. Keep the default, free-tier version that is already selected for you.
5. In the **Key pair (login)** section, click **Create new key pair**. Enter a name for your key pair and check the **.pem** radio box for the file format. Click **Create key pair**. Your key pair file will download to your computer.
6. Keep the remaining settings as their defaults and select **Launch instance**.

Your instance will now appear on the Instances page and should say it's running in the **Status** column. If not running, right-click anywhere on your instance row and click **Start instance**.

## Configuring security settings

AWS provides a lot of customization for an instance’s security settings. You can modify most of these settings by adding and editing rules in your instance’s security group. A *security group* is your instance’s firewall that controls inbound and outbound traffic. You can create your own security groups that you can assign to one or multiple instances, or let AWS create and assign one for you.

### Finding your instance’s security group

When you launched your first instance, you allowed AWS to assign it a security group instead of creating your own. To find your instance’s security group:

1. On the Instances page of EC2, click on your instance. A panel will appear at the bottom showing your instance’s IP addresses, security, monitor checks, and other details.
2. Click the **Security** tab. Under **Security Groups** you will find your instance’s security group listed. It will probably be named something like **launch-wizard-1**.
3. Click your security group. This will redirect you to a page listing your security group’s details and allows you to edit its inbound and outbound rules.

### Adding an inbound rule

Let’s say you want your actual computer and your instance to be able to exchange data. Security groups block all inbound traffic by default, even if you could successfully connect to that instance. You can test this by pinging your instance’s public IP from your computer. Pinging tests the connection between two IP addresses.

To ping your instance from your computer, open your command prompt or terminal and type ping followed by the public IP address of your instance. The output will likely display Request timed out multiple times, meaning the instance IP is unreachable.

To fix this, you can add a rule in the security group’s inbound rules to allow pings from your computer/IP address. To add an inbound rule to your security group:

1. From the security group’s page, click the **Inbound rules** tab, then click **Edit inbound rules**. This will show a table listing your inbound rules.
2. Click **Add rule**. A new row/rule has been added.
3. Select the following options from the dropdowns in this rule:
   * **Type**: Custom ICMP – IPv4.
   * **Protocol**: Echo Request.
   * **Source**: My IP.
4. Click **Save rules**.

Ping your instance again. The output should now print Reply from followed by the instance’s public IP address and ping statistics and information. This is because the new inbound rule now allows traffic just from your IP.

**Note:** Editing outbound rules works just like editing inbound rules, but is instead done in the **Outbound rules** tab.

## Connecting to your instance

Now that your instance is running and security configured, you're now ready to connect to your instance. There are different ways to connect to a Windows instance, but the easiest way is to download the instance as a .rdp file and log into it using the password decrypted from the key pair file that you downloaded earlier. To connect to your instance:

1. Right-click your instance and select **Connect** (or click the **Connect** button at the top).
2. Select the **RDP client** tab, then click **Download remote desktop file**. A .rdp file with the same name as your instance will download to your computer.
3. Click **Get password**, and then click **Upload private key file**. Find and open the key-pair .pem file you downloaded in Launching an instance. The contents of your private key should appear in the textbox below.
4. Click **Decrypt password**, and your instance password now appears. Copy the password.
5. Open the .rdp file you downloaded earlier. The Remote Desktop Connection client may warn you that this remote connection can’t be identified. Click **Connect**.
6. Paste your instance password into the **Password** textbox and then click **OK**. Your instance will now open.

The instance will resemble a regular computer with the Windows operating system. Feel free to play around by browsing the internet, running an application, or hosting another server. When finished, click the **X** button at the top right to close the instance.

## Conclusion

Instances are a great option for server hosting because they eliminate hardware costs and provide advanced network, configuration, and auto-recovery features. You now know the basics of instances and how to navigate AWS and EC2 to create, configure, and connect to an instance.

# Audience Analysis

This guide covers the basics of AWS instances. Readers will learn what an instance is and how to use AWS to launch, configure, and connect to one. In the real world, this guide would likely be viewed as an online article or training course.

## Experience and Knowledge

The guide assumes that the audience is new to AWS but should have an account created beforehand. The audience should already have general knowledge of IP addresses, the command prompt, network traffic, and cloud computing. However, work experience or background with these concepts is not required.

On a related note, this guide doesn’t require you to have a particular role to understand it. If you have an AWS account and understand the concepts, you will fit in the target audience.

## Needs

The audience needs a general overview of the concepts behind instances and why they should use AWS to host instances. They also need to know the essential AWS instance functions like how to launch, configure, and connect to an instance. They need to know how to navigate to and around the EC2 dashboard to do all these things.

The audience does not need to know how to use AWS’s other services because they won’t use them in this guide. They also do not need any special kind of computer or operating system to use AWS or connect to an instance.

## Intentions and Attitude

The audience may have different intentions when approaching this guide. Some may be cloud computing professionals wanting to know why they should switch to AWS for creating instances. Others may be interested in learning about instances for personal projects.

To satisfy both groups, the guide explains why AWS is preferred over its competitors for creating instances and demonstrates how simple it is to use EC2 to create, configure, and connect to an instance.

Though the audience has never used AWS, some readers may transfer their preconceived notions and attitudes toward Amazon to judge AWS. The guide assumes that the audience has a positive opinion of Amazon based on experience. Anyone with negative experiences with Amazon will probably not touch the article at all due to recognizing AWS or will find out at the beginning and thus click off.