Introduction to Instances in AWS

Jennifer Pero

LAST REVISION: 4/19/2023

Table of contents

Introduction

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 Server instance. Only general knowledge of cloud computing and networking concepts like IP addresses, Remote Desktop Protocol (RDP), 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 servers, run applications, etc. They are even powered by virtual versions of physical hardware like CPUs and RAM.

The only difference between instances and physical computers is that instances are entirely virtual; you can connect to an instance from your own computer, even if it uses a different operating system!

Types of instances

Not all instances are built the same, from the operating system to the virtual hardware specs. In AWS you can select an instance type that best suits your computational needs, and each type offers different instance families with various sizes and features. AWS offers six instance types:

- **General Purpose:** Balances computing, networking, and memory resources. Best for web servers and small application servers.
- **Compute Optimized:** Offers high computation power. Best for gaming servers and data models requiring high performance.
- **Memory Optimized:** Processes large datasets quickly for fast performance. Best for databases and analytics.
- **Accelerated Computing:** Uses hardware accelerators to efficiently process data. Best for graphics applications.
- **Storage Optimized:** Stores large volumes of data. Best for data warehousing.
- **HPC Optimized:** Affordable alternative for high-performance computing.

In this guide, you will create a *t2.micro* instance, a general-purpose instance from the T2 family. When creating an instance, t2.micro is normally the default selection because it's free-tier eligible and has a baseline performance. For a detailed look at instance types and their different families, sizes, and specifications, see Amazon EC2 Instance Types.

> **Note:** Free-tier eligibility means that, under the AWS free tier, you can run a t2.micro instance for 750 hours per month for the first 12 months after creating your AWS account. Once your instance passes the 750-hour mark, you will be charged by the hour that your instance is running. For more pricing details, see [Amazon EC2 Pricing](#).

Benefits of instances

On top of flexibility, 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?

Amazon Web Services (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 a Windows Server Instance

You can create and configure your t2.micro instances through the *EC2 Dashboard.* You will create a t2.micro instance that uses the Windows Server operating system. *Windows Server* resembles the regular Windows OS but with additional programs and features specifically designed for server management. To create a Windows Server t2.micro instance:

1. Log in to the AWS Console.
2. 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.
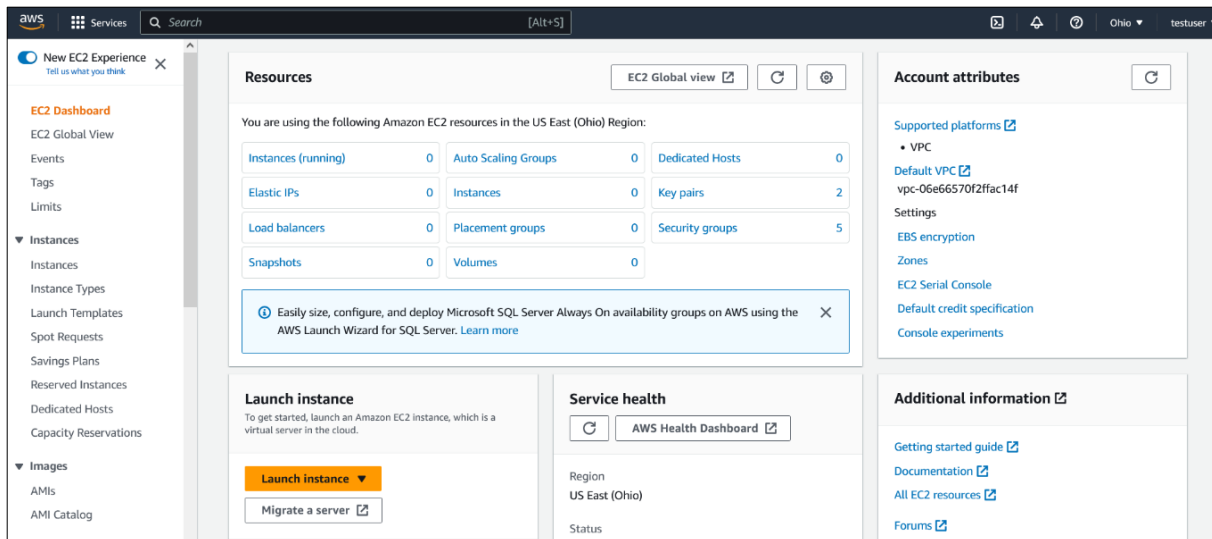
Figure 1: EC2 Dashboard

3. Click **Instances** under the ▼ **Instances** dropdown on the left side menu. This is where you'd see each of your instances listed as rows in a table. It should be empty.
4. Click **Launch instances**. Give your instance a name.
5. Select an operating system in the ▼ **Application and OS Images** section. Select **Windows** and keep the free-tier version that is already selected for you.
6. Select t2.micro under the ▼ **Instance type**. It may already be selected for you.
7. Click **Create new key** pair under ▼ **Key pair (login).**
8. Enter a name for your key pair and check the **.pem** radio box for the file format.
9. Click **Create key pair**. Your key pair file will be downloaded to your computer.
10. 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 **Instance state** column. If not running, right-click anywhere on your instance row and click **Start instance**.
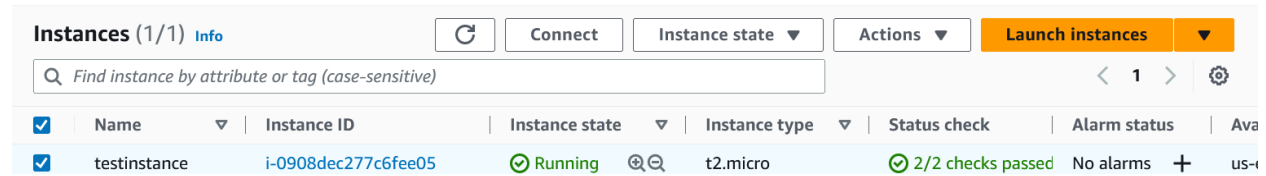

Figure 2: Instance page with a new launched instance.

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 to a security group instead of creating your own. To find your instance's security group:

1. Select your instance on the **Instances** page. A panel will appear at the bottom showing your instance's IP addresses, security, monitor checks, and other details.
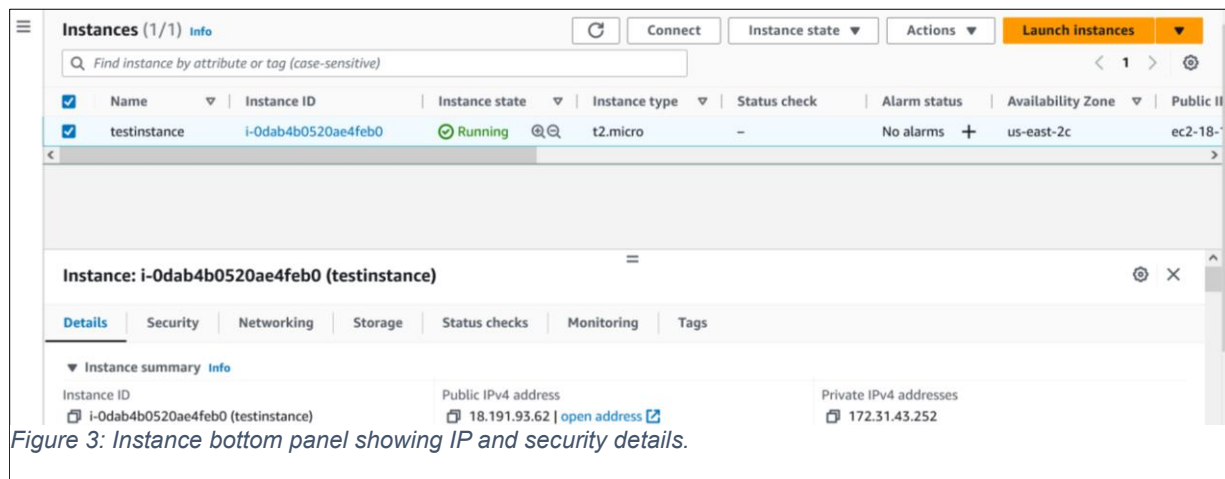

Figure 3: Instance bottom panel showing IP and security details.

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.

Figure 4: Security groups page for launch-wizard-1.

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 IPv4 address 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 IPv4 address of your instance which you can find in the **Details** tab of the bottom panel. The output will likely display `Request timed out` multiple times, meaning the instance IP is unreachable.

Figure 5: "Request timed out" shows when pinging your instance IP.

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. Click the **Inbound rules** tab from your security group's page.
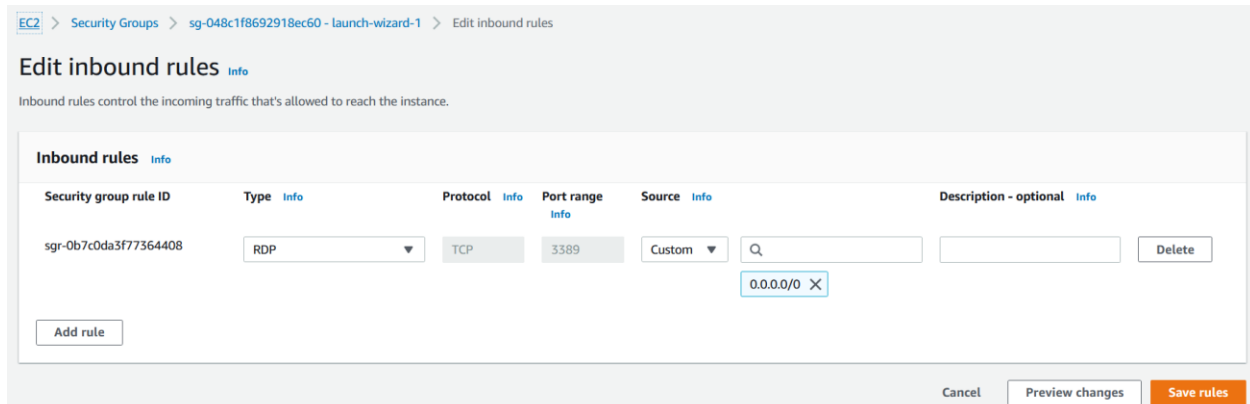

Figure 6: Inbound rules for launch-wizard-1, which is currently empty.

2. Click **Edit inbound rules**. This will show a page listing the inbound rules.
3. Click **Add rule**.
4. Select the following options from the dropdowns in this rule:
   - **Type**: Custom ICMP – IPv4.
   - **Protocol**: Echo Request.
   - **Source**: My IP.
5. 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 be downloaded to your computer.
3. Click **Get password**, and then click ⎗ **Upload private key file**.
4. Browse and open the key-pair .pem file you downloaded in the

6. **Creating** a Windows Server Instance section. The contents of your private key should appear in the textbox below.
7. Click **Decrypt password**, and your instance password now appears. Copy the password.
8. 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**.
9. Paste your instance password into the **Password** textbox and then click **OK**. Your instance will now open.

Your instance will resemble a Windows Server desktop with the wallpaper displaying your instance details, such as the IP address.
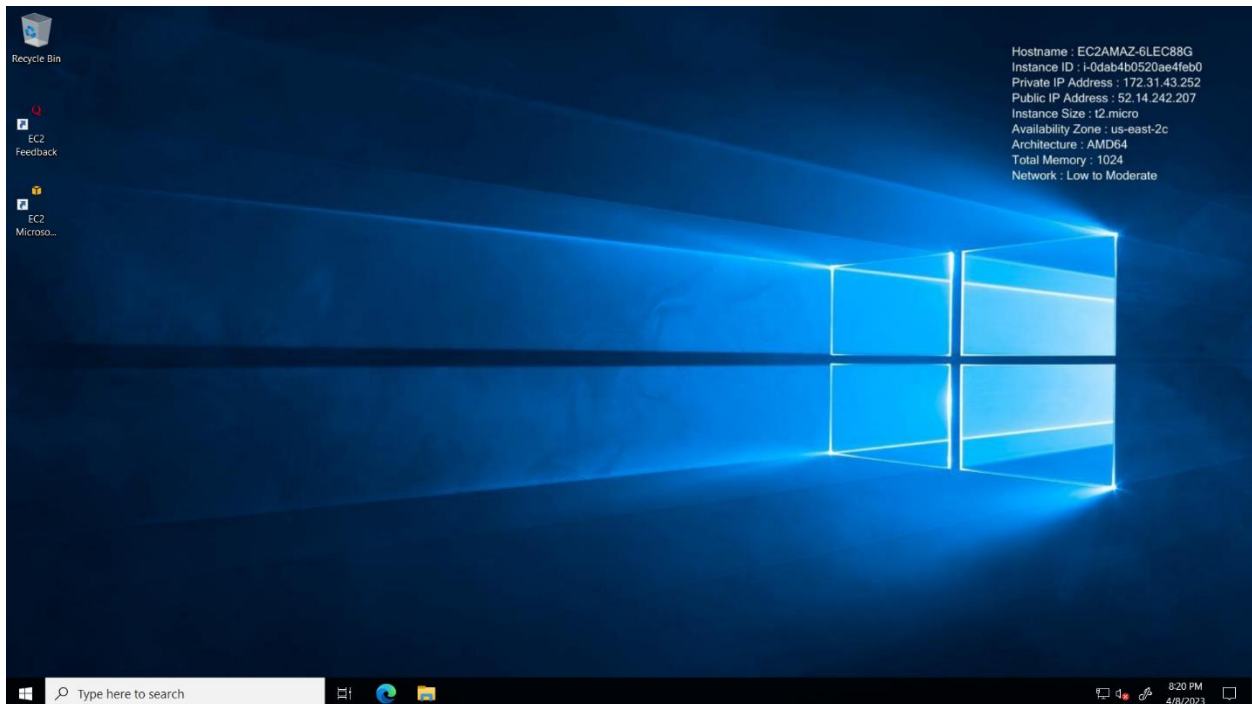

Figure 7: A connected Windows Server instance.

Feel free to play around by browsing the internet, running an application, or hosting a server. When finished, click the **X** button at the top to close the instance.

Caution: To avoid being charged after the 750-hour mark, you can stop your instance from running by right-clicking your instance on the Instances page and selecting **Stop Instance**. This will change the value in the **Instance state** column to **Stopped**. To delete your instance, right-click your instance and select **Terminate 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.

For your next steps, see the Amazon EC2 Windows Instance user guide to learn more about advanced instance topics, such as VPCs or Elastic Graphics. If you wish to use a different operating system, AWS also provides user guides for creating, configuring, and connecting to Mac and Linux instances. For further readings or assistance, see **Additional Resources**.

Additional Resources

- Official EC2 AWS Documentation
- AWS Video Tutorial on Amazon EC2
- AWS Support Services
- Video Introduction to AWS VPC and Subnets