AIGurukul Foundation

Complete Guide to Creating an AWS EC2 Instance

A Step-by-Step Tutorial for Beginners

By AIGurukul Foundation

Introduction

Amazon Elastic Compute Cloud (EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers by providing complete control of computing resources.

This tutorial will guide you through the process of creating and connecting to an EC2 instance, one of the fundamental services in Amazon Web Services (AWS).

Prerequisites

Before you begin, ensure you have:

- An active AWS account with appropriate permissions
- Basic understanding of cloud computing concepts
- A terminal application (Terminal for Linux/Mac or PuTTY/Command Prompt for Windows)
- Basic familiarity with command-line interfaces

Step 1: Access the EC2 Dashboard

Log into your AWS account and navigate to the EC2 service:

1. Sign in to the AWS Management Console at https://console.aws.amazon.com
2. In the search bar at the top, type **"EC2"** and click on the EC2 service
3. You will be directed to the EC2 Dashboard, which displays an overview of your EC2 resources

Step 2: Launch an EC2 Instance

2.1 Navigate to Instances

From the EC2 Dashboard, locate and click on **"Instances"** in the left sidebar. If no instances are running, you'll see an empty list.

2.2 Click Launch Instance

Click the **"Launch Instance"** button to begin the instance creation process.

Step 3: Configure Instance Settings

3.1 Name Your Instance

Provide a descriptive name for your instance:

- Example: **"my-web-server"** or **"rank-server"**
- This name helps you identify the instance later when managing multiple servers

3.2 Select Amazon Machine Image (AMI)

Choose your operating system:

- Select **Amazon Linux** or **Ubuntu Server**
- Choose **64-bit (x86)** architecture
- For this tutorial, we're using the Amazon Linux kernel with 64-bit architecture

3.3 Select Instance Type

Choose the instance size based on your needs:

- Select **t3.micro** instance type
- Specifications: **1 vCPU and 1 GB memory**
- Note: *t2.nano* is the cheapest option, while *t3.micro* is the second cheapest and offers better performance
- The t3.micro instance is eligible for the AWS Free Tier, making it cost-effective for learning and small projects

3.4 Configure Storage

Adjust the storage capacity for your instance:

- Default storage: **8 GB**
- You can increase this to **16 GB** or more depending on your requirements
- For this tutorial, we'll set it to 16 GB to ensure adequate space for applications

Step 4: Create a Key Pair

A key pair is essential for secure access to your EC2 instance. It consists of a public key (stored by AWS) and a private key (downloaded to your computer).

4.1 Create New Key Pair

4. Click on **"Create new key pair"**
5. Enter a name for your key pair (e.g., **"rank-server"**)
6. Select key pair type: **RSA**
7. Select file format: **.pem** (for OpenSSH/Linux/Mac) or **.ppk** (for PuTTY/Windows)
8. Click **"Create key pair"** — the .pem file will automatically download to your computer

4.2 Secure Your Key Pair

IMPORTANT: Store this file securely. You cannot download it again, and it's required to connect to your instance.

- Move the .pem file to a secure location (e.g., **~/.ssh/** on Linux/Mac)
- Set appropriate permissions (Linux/Mac): `chmod 400 your-key.pem`

Step 5: Configure Network Settings

Network settings control how your instance can be accessed from the internet.

5.1 Configure Security Group

Security groups act as virtual firewalls. Configure the following rules:

- **Allow SSH traffic:** Check **"Allow SSH traffic from anywhere"**

- This enables SSH connections on port 22 from any IP address (0.0.0.0/0)

- **Security Note:** In production environments, restrict SSH access to specific IP addresses

- **Allow HTTPS traffic:** Check **"Allow HTTPS traffic from the internet"**

- This enables secure web traffic on port 443

- Necessary if you plan to host a web application with SSL/TLS

- **HTTP traffic:** **NOT recommended** — HTTP is insecure; use HTTPS instead

Step 6: Launch the Instance

Review your configuration summary on the right side of the screen:

- Instance type: t3.micro
- Storage: 16 GB
- Key pair: Created
- Security group: Configured

Once everything is correct, click **"Launch Instance"**. AWS will begin provisioning your EC2 instance.

You'll see a success message with a link to view your instance. Click **"View Instances"** to see your newly created instance in the EC2 Dashboard.

Step 7: Connect to Your EC2 Instance

Once your instance state shows **"Running"** with a green indicator, you can connect to it.

7.1 Get Connection Information

9. Select your instance from the list
10. Click the **"Connect"** button at the top
11. Navigate to the **"SSH client"** tab

7.2 Connect from Linux/Mac

Follow these steps to connect using a Unix-based system:

12. **Open Terminal**
13. Navigate to the directory containing your .pem file:

```
cd ~/Downloads
```

14. Set the correct permissions on your key file:

```
chmod 400 rank-server.pem
```

15. Connect to your instance using SSH:

```
ssh -i "rank-server.pem" ec2-user@<your-instance-public-ip>
```

Replace `<your-instance-public-ip>` with the actual public IP address shown in the EC2 console.

16. When prompted *"Are you sure you want to continue connecting?"*, type **yes**

7.3 Connect from Windows

For Windows users, you have two main options:

Option 1: Using Windows PowerShell/Command Prompt (Windows 10+)

17. Open PowerShell or Command Prompt
18. Navigate to the directory with your .pem file
19. Use the same SSH command as Linux/Mac above

Option 2: Using PuTTY

20. Download and install PuTTY from www.putty.org
21. Convert your .pem file to .ppk format using PuTTYgen
22. Open PuTTY and enter your instance's public IP in the Host Name field
23. Navigate to Connection > SSH > Auth and browse to select your .ppk file
24. Click Open to connect

Step 8: Verify Your Connection

Once connected successfully, you'll see a command prompt indicating you're logged into your EC2 instance:

```
[ec2-user@ip-xxx-xx-xx-xx ~]$
```

Test basic commands to verify everything is working:

- **`pwd`** — Print working directory
- **`ls`** — List files and directories
- **`whoami`** — Display current username

Best Practices and Security Tips

Security Best Practices

- **Key Pair Management:** Never share your private key (.pem file) with anyone. Store it securely and back it up.
- **Restrict SSH Access:** In production environments, limit SSH access to specific IP addresses rather than allowing access from anywhere (0.0.0.0/0).
- **Regular Updates:** Keep your instance updated with the latest security patches using `sudo yum update -y` (Amazon Linux) or `sudo apt update && sudo apt upgrade -y` (Ubuntu).
- **Use IAM Roles:** Instead of storing AWS credentials on the instance, use IAM roles to grant permissions.

Cost Management

- **Stop When Not in Use:** Stop instances when not needed to avoid charges. You only pay for running instances.
- **Free Tier Awareness:** t2.micro and t3.micro instances qualify for the AWS Free Tier (750 hours/month for 12 months).
- **Monitor Usage:** Set up CloudWatch alarms and billing alerts to track your AWS spending.

Common EC2 Instance Types Comparison

Here's a comparison of commonly used EC2 instance types for beginners:

Instance Type	vCPUs	Memory	Best For
t2.nano	1	0.5 GB	Very light workloads, testing
t2.micro / t3.micro	1	1 GB	Small apps, learning, development
t2.small / t3.small	1	2 GB	Web servers, small databases
t2.medium / t3.medium	2	4 GB	Application servers, medium traffic

Note: t3 instances generally offer better performance per dollar than t2 instances and are newer generation.

Troubleshooting Common Issues

Connection Refused or Timeout

- Verify your security group allows SSH traffic (port 22) from your IP address
- Check that your instance is in the "Running" state
- Ensure you're using the correct public IP address or public DNS

Permission Denied (publickey)

- Verify the .pem file permissions are set to 400 (`chmod 400 keyfile.pem`)
- Ensure you're using the correct username (`ec2-user` for Amazon Linux, `ubuntu` for Ubuntu)
- Double-check you're using the correct key pair file that matches the instance

Instance Won't Start

- Check the instance status checks in the EC2 console
- Review the System Log (Actions > Monitor and troubleshoot > Get system log)
- Verify you haven't exceeded your EC2 instance limits

Next Steps

Now that you have a running EC2 instance, here are some recommended next steps:

25. **Install Software:** Install Git, Docker, Node.js, or other tools you need for development
26. **Deploy Applications:** Clone your GitHub repositories and deploy your applications
27. **Configure Elastic IP:** Assign a static IP address to your instance for consistent access
28. **Set Up Load Balancing:** For production applications, configure an Elastic Load Balancer
29. **Create AMI Snapshots:** Back up your configured instance by creating Amazon Machine Images
30. **Explore Other AWS Services:** Learn about RDS (databases), S3 (storage), Lambda (serverless), and more

Conclusion

Congratulations! You have successfully created and connected to your first AWS EC2 instance. This is a fundamental skill in cloud computing and opens up many possibilities for hosting applications, running experiments, and learning about cloud infrastructure.

Remember to stop or terminate your instance when you're done using it to avoid unnecessary charges. You can always launch new instances when needed, and with practice, this process will become quick and routine.

For more advanced topics and troubleshooting, refer to the official AWS documentation at docs.aws.amazon.com.