

What is EC2 instance?

Amazon EC2 (Elastic Compute Cloud) is a web service provided by AWS that allows you to rent virtual servers to run your applications. It offers scalable computing capacity in the cloud, enabling you to quickly launch and manage instances (virtual machines) with various operating systems and configurations. This flexibility helps you adjust your resources based on demand, making it easy to scale up or down as needed, while only paying for what you use. Whether you're hosting a website, running applications, or performing data analysis, EC2 provides the infrastructure to support your needs efficiently.

Here's a step-by-step lab guide for students to create an EC2 instance on AWS and SSH into it from their local computers.

Lab Objective

Students will learn how to launch an Amazon EC2 instance and connect to it using SSH.

Prerequisites

- An AWS account. If students do not have an account, they can sign up for a free tier.
- Basic knowledge of using the command line.
- SSH client installed on their local machine (for Windows, tools like PuTTY or Git Bash can be used).

Lab Steps

Step 1: Log in to AWS Management Console

1. Go to AWS Management Console.
2. Log in with your AWS account credentials.

Step 2: Navigate to EC2 Dashboard

1. In the AWS Management Console, search for **EC2** in the services search bar.
2. Click on **EC2** to go to the EC2 Dashboard.

Step 3: Launch an EC2 Instance

1. Click on the **Launch Instance** button.
2. **Choose an Amazon Machine Image (AMI):**
 - Select a **Linux** AMI (e.g., Amazon Linux 2).
3. **Choose an Instance Type:**

- Select **t2.micro** (eligible for the free tier).
- 4. Click on **Next: Configure Instance Details**.
- 5. Configure the instance as needed, then click **Next: Add Storage**.
- 6. You can leave the default storage settings and click **Next: Add Tags**.
- 7. (Optional) Add tags if needed, then click **Next: Configure Security Group**.
- 8. **Configure Security Group:**
 - Create a new security group:
 - **Name:** MyEC2SecurityGroup
 - **Description:** Allow SSH access
 - **Inbound Rules:**
 - Type: SSH
 - Protocol: TCP
 - Port Range: 22
 - Source: Anywhere (0.0.0.0/0) — *(for the lab, but recommend restricting this in production environments)*
- 9. Click **Review and Launch**.
- 10. Review the settings and click **Launch**.
- 11. When prompted to select an existing key pair or create a new key pair, choose:
 - **Create a new key pair.**
 - Name it (e.g., MyKeyPair).
 - **Download the key pair** (MyKeyPair.pem) and save it in a secure location.
- 12. Click **Launch Instances**.

Step 4: Locate the EC2 Instance

1. Once the instance is launched, click on **View Instances**.
2. Note the **Public DNS** or **Public IP** of your instance.

Step 5: Set Up SSH Access

For Linux/Mac Users

1. **Open the Terminal:**
 - You can find the Terminal application in your Applications folder or search for it.

2. Change Permissions of the Key File:

- Navigate to the directory where you saved your .pem file using the cd command. For example:

```
cd /path/to/your/key
```

Change the permissions of the key file to ensure it's secure:

```
chmod 400 MyKeyPair.pem
```

Connect to Your EC2 Instance:

- Use the SSH command to connect to the instance. Replace your-instance-public-dns with the actual Public DNS or Public IP you noted earlier:

```
ssh -i MyKeyPair.pem ec2-user@your-instance-public-dns
```

1.

- The default username for Amazon Linux 2 is ec2-user.

2. Accept the SSH Key Fingerprint:

- The first time you connect, you may be prompted to confirm the connection by accepting the SSH key fingerprint. Type yes and press Enter.

3. Verify Connection:

- You should now be logged into your EC2 instance. You can run commands like `uname -a` to check the system information.

For Windows Users (Using PuTTY)

1. Convert the PEM File to PPK Format:

- Open **PuTTYgen**:
 - Search for PuTTYgen in your Start menu and open it.
- Click on **Load**:
 - Set the file type to *All Files* in the dropdown to see your .pem file.
 - Select your .pem file and click **Open**.
- Click **Save private key** to save it as a .ppk file. You can ignore any warning about saving without a passphrase for this lab.

2. **Open PuTTY:**

- Search for PuTTY in your Start menu and open it.

3. **Configure the Connection:**

- **Host Name (or IP address):** Enter the Public DNS or Public IP of your EC2 instance.
- **Port:** Make sure this is set to 22.
- **Connection Type:** Ensure that SSH is selected.

4. **Load the PPK File:**

- In the left sidebar, expand **SSH** and select **Auth**.
- Click on **Browse** and select the .ppk file you saved earlier.

5. **Connect to Your EC2 Instance:**

- Go back to the **Session** category in PuTTY.
- Click **Open** to start the connection.
- A terminal window will appear, and you may see a security alert regarding the server's host key. Click **Yes** to continue.

6. **Login as ec2-user:**

- When prompted for a username, enter ec2-user and press Enter.

7. **Verify Connection:**

- You should now be logged into your EC2 instance. Run `uname -a` or other commands to verify.

Troubleshooting Tips

- **Connection Timed Out:** Ensure your security group allows inbound SSH traffic (port 22).
- **Permission Denied:** Make sure you're using the correct username (ec2-user for Amazon Linux) and that the key file permissions are set correctly.

Detail of Aws EC2 service

Amazon EC2 (Elastic Compute Cloud) is a core part of Amazon Web Services (AWS) that provides resizable compute capacity in the cloud. It allows users to run virtual servers, known as instances, which can be configured according to specific needs.

Key Features

1. Flexible Instance Types:

- EC2 offers a variety of instance types optimized for different use cases, such as compute-optimized, memory-optimized, storage-optimized, and GPU instances. This allows users to choose the best resources for their applications.

2. Scalability:

- With EC2, you can easily scale your resources up or down. This elasticity means you can launch as many or as few instances as needed, adjusting capacity based on demand without upfront investment.

3. Pay-as-You-Go Pricing:

- EC2 follows a pay-as-you-go pricing model, where you only pay for the compute capacity you use. This makes it cost-effective for various workloads.

4. Multiple Operating Systems:

- You can choose from various operating systems, including different versions of Linux and Windows, giving you the flexibility to run applications in the environment of your choice.

5. Storage Options:

- EC2 provides different storage solutions, such as Amazon Elastic Block Store (EBS) for persistent storage and Amazon S3 for object storage. This allows you to manage data effectively.

6. Security:

- EC2 integrates with AWS Identity and Access Management (IAM) to control access to instances. You can create security groups to define inbound and outbound traffic rules, enhancing your security posture.

7. Networking:

- EC2 instances can be launched in a Virtual Private Cloud (VPC), giving you control over your network configuration. You can set up public and private subnets, configure route tables, and utilize elastic IPs.

8. Load Balancing and Auto Scaling:

- EC2 works seamlessly with other AWS services like Elastic Load Balancing (ELB) and Auto Scaling, allowing you to distribute incoming traffic and automatically adjust the number of running instances based on traffic patterns.

9. **Monitoring and Management:**

- With Amazon CloudWatch, you can monitor the performance and health of your EC2 instances. You can set up alarms and automate responses to certain conditions.

10. **Instance Marketplace:**

- The AWS Marketplace provides a variety of pre-configured software applications that you can easily deploy on EC2 instances, allowing for rapid setup of popular applications and services.

Common Use Cases

- **Web Hosting:** Host websites and applications with dynamic scalability based on traffic.
- **Application Development and Testing:** Quickly spin up environments for developing and testing applications without long-term commitments.
- **Data Processing and Analysis:** Run big data frameworks like Hadoop or Spark on scalable instances.
- **Machine Learning:** Utilize GPU-optimized instances for training and inference in machine learning models.
- **Disaster Recovery:** Set up a cost-effective disaster recovery solution by spinning up instances in a different region as needed.

Conclusion

Amazon EC2 is a powerful and versatile service that provides the infrastructure needed to run a wide range of applications in the cloud. Its flexibility, scalability, and variety of features make it an essential tool for businesses and developers looking to leverage cloud computing effectively. Whether you need to run a simple website or complex data processing tasks, EC2 can adapt to your needs while offering robust security and management capabilities.