# Amazon Machine Image (AMI)

In Amazon Web Services (AWS), an Amazon Machine Image (AMI) is a pre-configured template that contains the information needed to launch an instance (a virtual server) in the cloud. It serves as a foundational building block for creating virtual servers within the AWS ecosystem. Here are key aspects of an AMI:

1. **Image Components:**

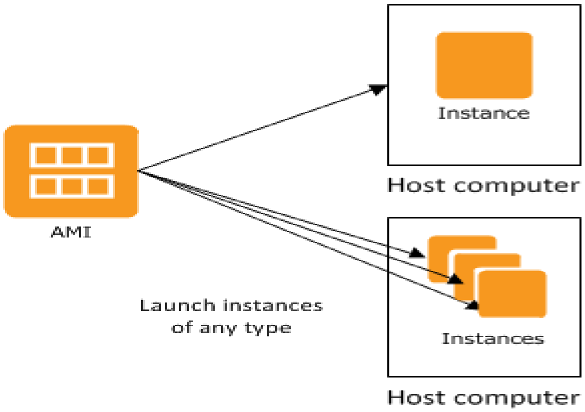
   - **Root Volume:** The root volume contains the necessary operating system, application server, and application software.
   - **Block Device Mapping:** Defines the block devices (Elastic Block Store volumes) to attach to the instance when it launches.

2. **AMI Types:**

   - **Public AMIs:** Created by AWS or the AWS community and are available for anyone to use. They are often used for common operating systems and applications.
   - **Private AMIs:** Created and managed by individual AWS account owners. These AMIs are restricted to the account that created them.

3. **AMI Creation:**

   - AMIs can be created from instances that are either running or stopped. This process involves taking a snapshot of the root volume and creating an image from that snapshot.
   - Custom AMIs can include specific configurations, installed software, and even data, providing a quick and consistent way to launch instances with a predefined setup.



4. **Usage:**
   - AMIs are used to launch instances, which are virtual servers in the AWS cloud.
   - Instances launched from an AMI inherit the configurations, software, and data defined in the AMI.
5. **Lifecycle:**
   - AMIs can be copied across AWS regions, enabling users to launch instances in different geographic locations.
   - They can be versioned to track changes and updates.
   - Users can share custom AMIs with other AWS accounts, either publicly or privately.
6. **Security Considerations:**
   - Users should be cautious about using public AMIs from unknown sources, as they may not follow best security practices.
   - Regularly updating and patching AMIs is crucial to maintaining a secure environment.
7. **Use Cases:**
   - **Reproducibility:** AMIs enable the creation of identical instances with a specific configuration, promoting consistency and reproducibility.
   - **Scalability:** AMIs facilitate the quick launch of multiple instances to handle varying workloads or to scale applications.

In summary, an Amazon Machine Image in AWS provides a convenient and efficient way to package, store, and replicate the configuration of a virtual server, streamlining the process of deploying and scaling applications in the cloud.

# Launching a instances and installing httpd package

## Launching an EC2 Instance:

1. **Sign in to the AWS Management Console:**
    - Go to AWS Console and sign in to your AWS account.
2. **Navigate to EC2:**
    - In the AWS Management Console, go to the "Services" dropdown and select "EC2" under the "Compute" section.
3. **Launch Instance:**
    - Click on the "Launch Instance" button to start the instance creation process.
4. **Choose an Amazon Machine Image (AMI):**
    - Select an AMI for your instance,you can choose an Amazon Linux or Ubuntu AMI.
5. **Choose an Instance Type:**
    - Select the instance type t2.micro instances.
6. **Select Key Pair:**
    - Choose an existing key pair or create a new one for secure SSH access.
7. **Configure Security Group:**
    - Configure the security group rules to allow incoming traffic. Ensure that you allow HTTP traffic (port 80) if you plan to install a web server.
8. **Configure Instance:**
    - Configure additional settings such as the number of instances, network settings, and IAM roles.
9. **Review and Launch:**
    - Review your instance configuration and click "Launch."
10. **Launch Instances:**
    - Click "Launch Instances" to launch your EC2 instance.

## Installing Apache HTTP Server (httpd):

1. **Connect to Your Instance:**
   - Once your instance is running, use SSH to connect to your instance.
2. **Update Package Information:**
   - Update the package information on the instance.

   **sudo yum update -y**
3. **Install  HTTP Server (httpd):**
   - Install the  HTTP Server package.

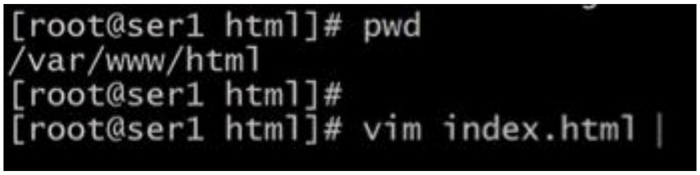
   **yum install httpd -y**
4. **Start the httpd Service:**
   - Start the httpd service.

   service httpd start  or  **systemctl start httpd**
5. **Check the status of httpd service:**

   service httpd status  or  **systemctl status httpd**
6. **Enable httpd to Start on Boot:**
   - Enable httpd to start on boot.

   **service httpd enable or  systemctl enable httpd**
7. **Navigate to the /var/www/html Directory:**
   - Use the cd command to change to the /var/www/html directory.
8. **Create index.html file**

   **Vim  index.html**



```
[root@ser1 html]# pwd
/var/www/html
[root@ser1 html]#
[root@ser1 html]# vim index.html |
```

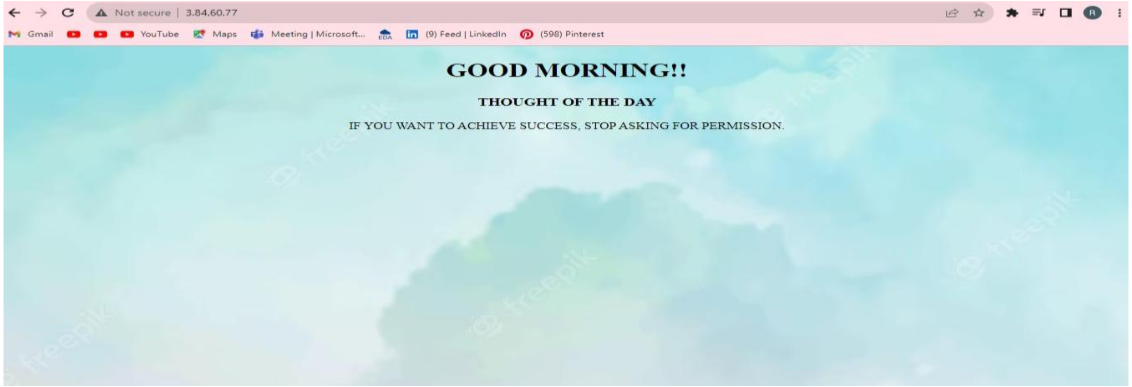
- Write your sample html code

```
 <!DOCTYPE html>
<html>
<body>
<h1>GOOD MORNING!!</h1>
<h2>THOUGHT OF THE DAY</h2>

<p>IF YOU WANT TO ACHIEVE SUCCESS, STOP ASKING
FOR PERMISSION</p>

</body>
</html>
```

- Save and exit from vi editor by command **:wq**

9. **Verify Installation:**

   - Open a web browser and enter your instance's public IP address (00.00.00:80)



GOOD MORNING!!

THOUGHT OF THE DAY

IF YOU WANT TO ACHIEVE SUCCESS, STOP ASKING FOR PERMISSION.

## Creating AMI

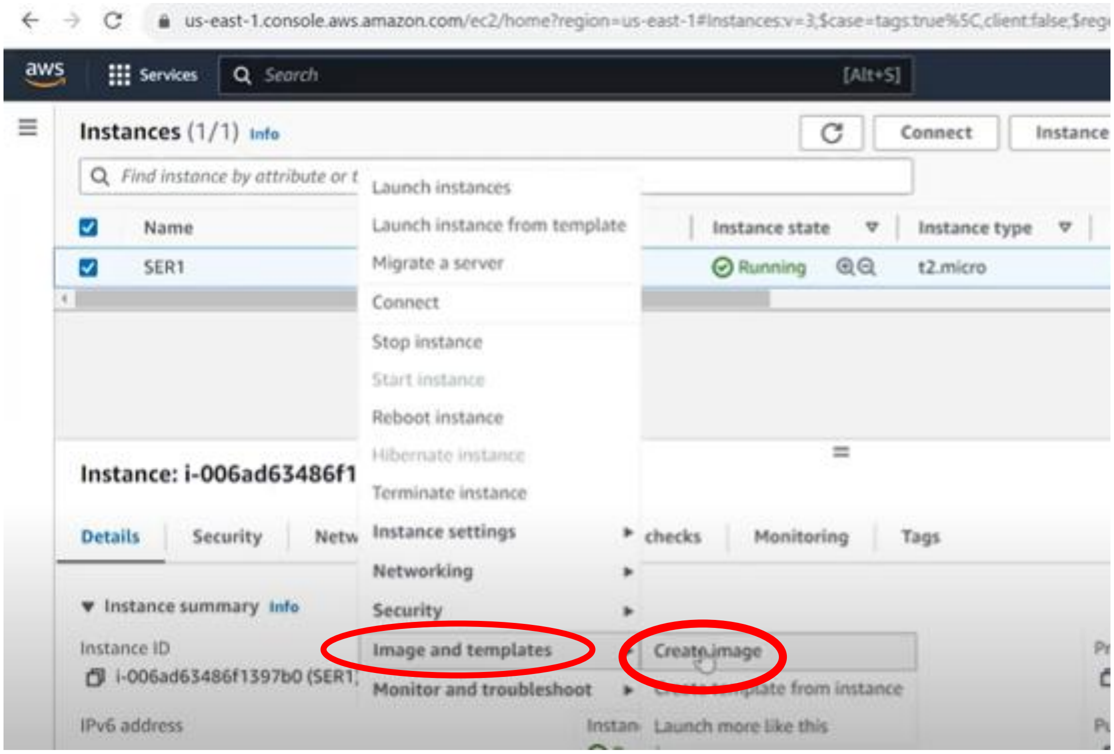
Creating an Amazon Machine Image (AMI) in AWS involves taking a snapshot of an existing EC2 instance's root volume. Here are the general steps to create an AMI using the AWS Management Console:

**Prerequisites:**

- You need an existing running or stopped EC2 instance.

Steps:

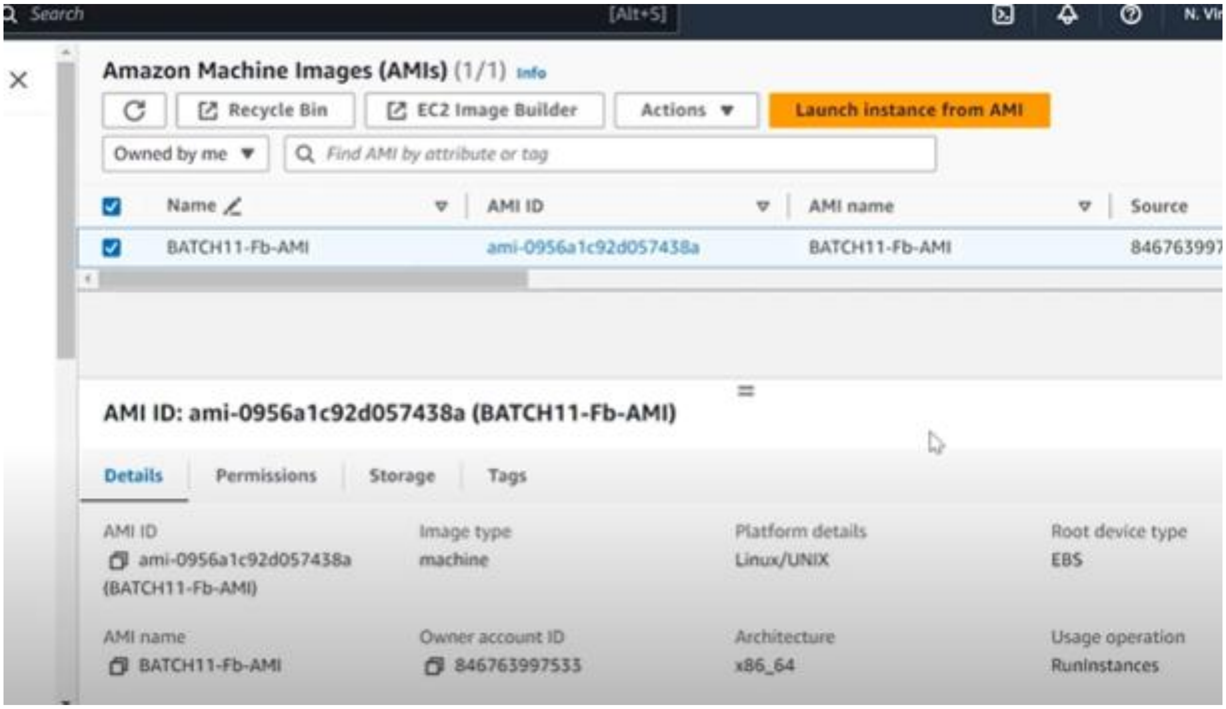
1. **Sign in to the AWS Management Console:**
   - Go to [AWS Console](), and sign in to your AWS account.
2. **Navigate to EC2:**
   - In the AWS Management Console, go to the "Services" dropdown and select "EC2" under the "Compute" section.
3. **Select Your Instance:**
   - In the EC2 Dashboard, click on "Instances" in the left navigation pane.
   - Select the EC2 instance for which you want to create an AMI.
4. **Create Image (AMI):**
   - With the instance selected, click on the "Actions" dropdown.
   - Under "Image and templates," click on "Create Image."



5. **Configure Image:**

- In the "Create Image" wizard, provide a unique and descriptive name for your AMI.
- You can also provide a description and add tags for better organization.

6. **Review and Create:**
    - Review the configuration settings, and click on "Create Image."
7. **Monitor Image Creation:**
    - The AMI creation process starts, and you can monitor its progress on the "AMIs" section in the EC2 Dashboard.
8. **Use the AMI:**
    - Once the AMI creation is complete, you can use the newly created AMI to launch new instances with the same configuration.



Amazon Machine Images (AMIs) (1/1)

| ☑ | Name | AMI ID | AMI name | Source |
|---|---|---|---|---|
| ☑ | BATCH11-Fb-AMI | ami-0956a1c92d057438a | BATCH11-Fb-AMI | 846763997 |

AMI ID: ami-0956a1c92d057438a (BATCH11-Fb-AMI)

| AMI ID | Image type | Platform details | Root device type |
|---|---|---|---|
| ami-0956a1c92d057438a (BATCH11-Fb-AMI) | machine | Linux/UNIX | EBS |
| **AMI name** | **Owner account ID** | **Architecture** | **Usage operation** |
| BATCH11-Fb-AMI | 846763997533 | x86_64 | RunInstances |

## Launching instances from AMI

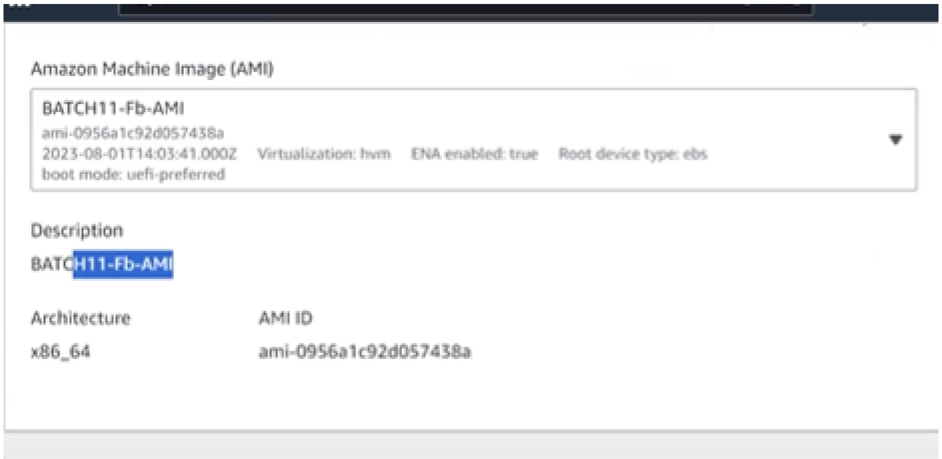
After creating an Amazon Machine Image (AMI) in AWS, you can launch instances from that AMI. Here are the steps to launch an instance from an AMI using the AWS Management Console:

**Prerequisites:**

- You should have already created an AMI.

**Steps:**

1. **Sign in to the AWS Management Console:**
    - Go to [AWS Console](https://console.aws.amazon.com/) and sign in to your AWS account.
2. **Navigate to EC2:**
    - In the AWS Management Console, go to the "Services" dropdown and select "EC2" under the "Compute" section.
3. **Launch Instance:**
    - In the EC2 Dashboard, click on "Launch Instance" to start the instance creation process.
4. **Choose an Amazon Machine Image (AMI):**
    - In the "Choose an Amazon Machine Image (AMI)" step, select the "My AMIs" tab.
    - Choose the AMI that you created earlier from the list.



Amazon Machine Image (AMI)

BATCH11-Fb-AMI
ami-0956a1c92d057438a
2023-08-01T14:03:41.000Z Virtualization: hvm ENA enabled: true Root device type: ebs
boot mode: uefi-preferred

Description

BATCH11-Fb-AMI

| Architecture | AMI ID |
|---|---|
| x86_64 | ami-0956a1c92d057438a |

5. **Choose an Instance Type:**

    - In the "Choose an Instance Type" step, select the instance type that best fits your requirements. Click "Next" to proceed.

6. **Configure Security Group:**

    - In the "Configure Security Group" step, you can specify the security group rules for your instance. Ensure that you have the necessary rules to allow incoming traffic based on your application's requirements. Click "Review and Launch" when ready.

7. **Select Key Pair:**

    - In the "Select an existing key pair or create a new key pair" step, choose an existing key pair or create a new one. This key pair is used for secure SSH access to your instance.

8. **Configure Instance:**

    - In the "Configure Instance" step, you can configure additional settings such as the number of instances, network settings, and IAM roles. Adjust the settings as needed and click "Next."

9. **Add Tags (Optional):**

    - In the "Add Tags" step, you can add tags to your instance for better organization. Tags are key-value pairs. Click "Next" when you're done.

10. **Review and Launch:**

    - Review your instance configuration in the "Review Instance Launch" step. Click "Launch" to proceed.

11. **View Instances:**

    - You'll see a confirmation screen. Click on the instance ID to view details about your new instance.

    launched an EC2 instance from the custom AMI you created. You can now connect to and use this instance as needed for your applications.

**12. Verify:**

- Now verify your instances that all data packages from server1 are integrated in present server.

## AWS Launch Template

An AWS Launch Template is a resource that contains the parameters to launch instances in an Amazon Elastic Compute Cloud (EC2) environment. It's a mechanism that simplifies and standardizes the process of launching instances, making it easier to maintain and manage configurations. Launch templates provide a way to configure instances, including the AMI, instance type, key pair, security groups, and other settings.

Key Features and Components:

1. **Template Versions:**
   - Launch templates support versioning, allowing you to create and manage different versions of a template.
2. **Specify Instance Details:**
   - You can specify various instance details in a launch template, such as the Amazon Machine Image (AMI), instance type, key pair, block device mappings, and network interfaces.

**Use Cases:**

1. **Standardization:**
   - Launch templates help in standardizing the configuration of instances, making it easier to maintain consistency across different deployments.
2. **Easy Maintenance:**
   - As configurations are defined in a template, it becomes easier to update and maintain settings without the need to modify individual launch configurations.
3. **Scalability:**
   - Launch templates are often used with services like EC2 Auto Scaling and EC2 Fleet to easily scale the number of instances based on demand.

**Example Scenario:**

Let's say you have a set of instances that require a specific configuration, including a particular AMI, instance type, and security group settings. Instead of manually specifying these details each time you launch an instance, you can create a launch template with the desired configurations. Then, whenever you need to launch instances with that configuration, you simply refer to the launch template.This approach simplifies the launch process, ensures consistency, and allows for easy updates to the configuration.