Day7 of learning AWS

08 February 2025 07:58 PM

Understanding AWS AMI: Boosting Performance with Golden Images

What is AWS AMI?

In the world of AWS (Amazon Web Services), AMI (Amazon Machine Image) plays a crucial role in efficiently managing virtual servers. Amazon Machine Image (AMI) is like a blueprint for creating virtual machines (EC2 instances) in AWS. It contains all the necessary information to launch instances, including the operating system, application servers, and any additional software.

How AWS Uses AMI:

AWS utilizes AMIs to streamline the process of deploying virtual machines. Here's a simplified breakdown:

1. **Creation of AMI:**
   - Users can create their custom AMIs or choose from the pre-built ones provided by AWS.
   - Customization may include installing specific software, configuring settings, and optimizing the system.
2. **Launching Instances:**
   - Users can launch EC2 instances from selected AMIs.
   - These instances inherit the configurations and software specified in the AMI.

How Can We Use AMI?

1. **Selecting an AMI:**
   - In the AWS Management Console, navigate to the EC2 service.
   - Choose the "Launch Instance" option.
   - Select an AMI based on your needs - either AWS-provided or your custom image.
2. **Customizing AMI:**
   - Customize the AMI during the launch process if needed.
   - Add or remove software, configure security settings, and define instance specifications.
3. **Launching Instances:**
   - Complete the instance launch wizard.

- The new instance will have the same configurations as specified in the chosen AMI.

What is a Golden Image?

A Golden Image is a term often used interchangeably with AMI. It refers to a fully configured and optimized virtual machine image that serves as the ideal starting point for creating instances. Think of it as a 'golden standard' configuration that can be replicated effortlessly.

How Golden Image Improves Performance:

1. **Consistency:**
   - Using a Golden Image ensures consistency across instances.
   - Every instance launched from the same AMI will have identical configurations.
2. **Time Efficiency:**
   - Golden Images save time in deployment.
   - Instead of manually configuring each instance, you can replicate a pre-configured and optimized image.
3. **Performance Optimization:**
   - Golden Images are pre-tuned for performance.
   - This eliminates the need for applying user data scripts during instance launch, reducing overhead.
4. **Scalability:**
   - Replicating instances from a Golden Image makes scaling up or down more straightforward.
   - This is crucial for dynamic workloads with varying demands.

In conclusion, AWS AMI, especially in the form of a Golden Image, is a powerful tool for ensuring consistent, efficient, and high-performance deployments. By understanding how to use and optimize AMIs, you can enhance your AWS experience and streamline your cloud infrastructure.

Demo

we'll walk you through the process of creating an EC2 instance on AWS, installing Nginx, customizing the index.html page, and finally, making your own AMI for future use.

Step 1: Create EC2 Instance with Preconfigured Ubuntu AMI

1. Go to the AWS Management Console.
2. Select "EC2" from the services menu.
3. Click "Launch Instance."
4. Choose an Ubuntu AMI preconfigured by AWS to simplify the setup.

Step 2: Install Nginx on Your EC2 Instance

5. Once the instance is running, connect to it using SSH.

6. Install Nginx using the command: `sudo apt-get update && sudo apt-get install nginx -y`.

Step 3: Customize Your Website's Welcome Message

7. Customize the index.html page with: `echo "Welcome to learning-ocean.com" > /var/www/html/index.html`.

Step 4: Create an AMI from Your EC2 Instance

8. In the AWS Management Console, go to "Instances" and select your running instance.
9. Click "Actions," go to "Image and templates," and choose "Create Image."
10. Fill in the required details and click "Create Image."

Step 5: Launch an EC2 Instance from Your New AMI

11. In the EC2 Dashboard, select "AMIs" from the sidebar.
12. Choose the AMI you just created and click "Launch Instance."
13. Follow the steps to configure your new instance, and you're all set!

Now you've successfully created an EC2 instance, installed Nginx, customized your website's welcome message, and made your own AMI. This simple guide ensures a smooth process for your demo purposes on AWS.

Here are the steps to **delete an AMI (Amazon Machine Image) in AWS**:

1. **Deregister the AMI**
   - Go to the **EC2 Dashboard** in AWS.
   - Click on **AMIs** under the "Images" section.
   - Select the AMI you want to delete.
   - Click **Actions** → **Deregister AMI** → Confirm deregistration.
2. **Delete the Associated Snapshots** (Optional)
   - Go to the **Snapshots** section in the EC2 Dashboard.
   - Find the snapshot(s) associated with the AMI.
   - Select the snapshot and click **Actions** → **Delete snapshot** → Confirm deletion.

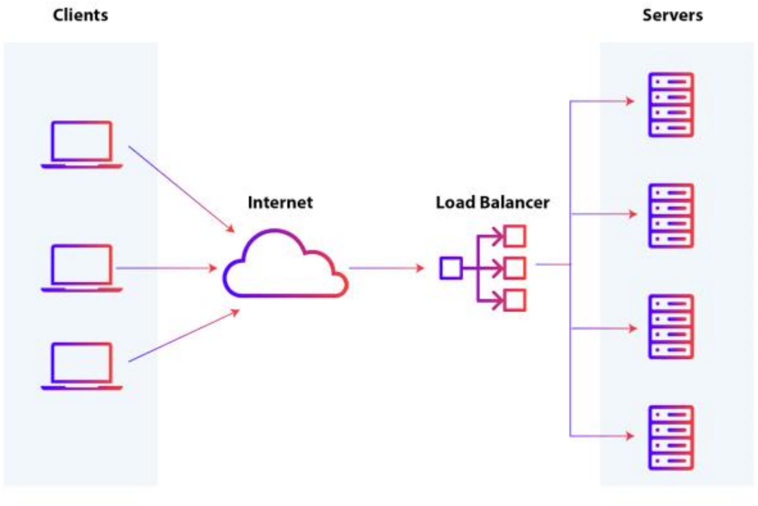
What is load balancer?

A **Load Balancer** is a system that distributes incoming network traffic across multiple servers to ensure no single server becomes overloaded. This improves **availability, reliability, and scalability** of applications.

How It Works

1. **Receives Requests** → Users send requests to a single endpoint (e.g., example.com).

2. **Distributes Load** → The load balancer determines which server should handle the request based on predefined rules.
3. **Routes Requests** → It forwards the request to the selected server.
4. **Monitors Health** → Continuously checks if servers are healthy and removes unhealthy ones from traffic distribution.


Clients
Internet
Load Balancer
Servers