aws

AWS AMI
Amazon Machine Image

AMI

An Amazon Machine Image (AMI) is a pre-configured template that provides the necessary information to launch an EC2 instance in AWS.

It includes:
- Operating system (e.g., Linux, Windows)
- Application server (e.g., Apache, Nginx)
- Pre-installed software and configurations

With an AMI, you can launch new EC2 instances with a consistent, predefined configuration.

You can also create custom AMIs to include specific software or settings, allowing for quick replication of environments.

Types of AWS AMIs

- **Public AMIs**: Available to all AWS users. Useful for basic use cases like popular operating systems (e.g., Ubuntu, CentOS).

- **Private AMIs**: Created by a user and only available within that account or shared with specific accounts.

- **Paid AMIs/Marketplace AMIs**: Provided by third parties through AWS Marketplace, offering software like databases, web servers, or pre-configured environments.

Common use cases for each type:

- **Public**: Testing or development environments.
- **Private**: Customized setups for production.
- **Marketplace**: Deploying pre-built solutions (e.g., enterprise software).

UseCase of Paid AMIs

Benefits:

- **Rapid Deployment**: The LAMP Stack is pre-configured, eliminating the need to manually install and configure Apache, MySQL, and PHP.
- **Scalability and Load Balancing**: Running on AWS enables quick scaling to match website traffic, while Elastic Load Balancer helps in distributing requests.
- **Cost Efficiency**: You only pay for the infrastructure and the software according to your usage.

Creating an AWS AMI

Run Cleanup script

```bash
#!/bin/bash

# Remove SSH keys
rm -rf ~/.ssh/authorized_keys

# Clear user credentials and history
rm -rf ~/.aws/credentials
rm -rf ~/.git-credentials
rm -rf ~/.bash_history

# Clean system logs and temporary files
rm -rf /var/log/*
rm -rf /tmp/*
rm -rf /var/tmp/*

# Remove user accounts
deluser tempuser --remove-home

# Lock root account
passwd -l root

# Reset configuration files (example for Nginx)
rm -rf /etc/nginx/nginx.conf
```

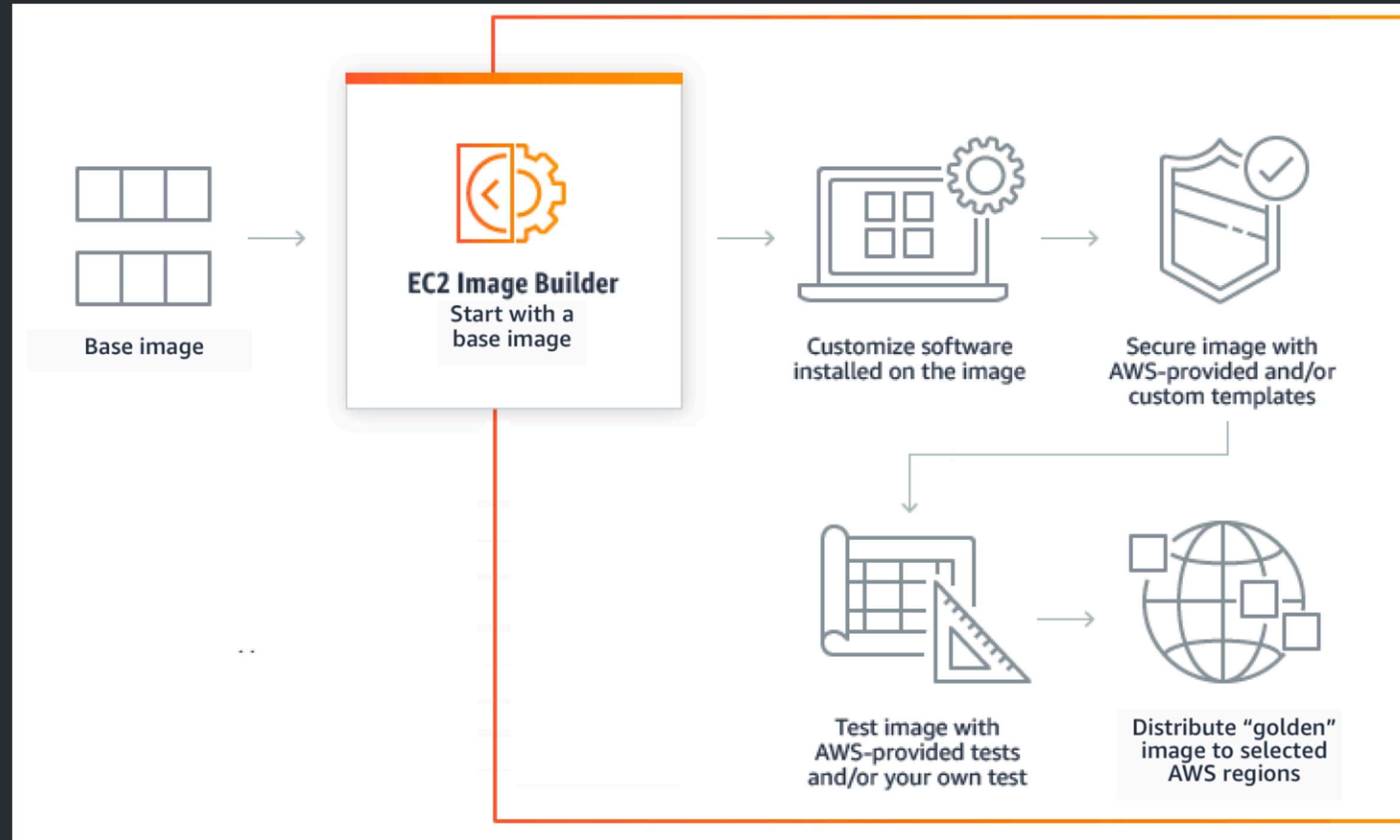# Create Launch Template
Vs
Create Image

Key Differences

Feature	Create Launch Template	Create Image (AMI)
Purpose	Create a reusable blueprint for launching instances	Create a custom AMI snapshot of an instance
Content	Contains configuration settings (e.g., AMI, instance type, security groups, etc.)	Captures OS, applications, configurations, and data
Reusability	Used repeatedly to launch new instances in a consistent way	Used to create new instances that replicate the captured image
Use Cases	Auto Scaling, Spot Instances, Standardizing instance settings	Backup, Replication, Migrating instances to another region
Versioning	Can have multiple versions for different configurations	Typically, an AMI is a point-in-time capture of an instance

Summary:

- Yes, all **installed applications, configuration settings, environment variables, network configurations, DNS settings, users, and firewall settings** will be included in the AMI.

- An AMI is essentially a **complete snapshot** of the instance at the point in time you created the image, allowing you to replicate the exact state of that server, including all software, configuration, and operating system-level changes.

- When you create an EC2 instance from this AMI, it will boot up as if it were an exact clone of the original, with all installed software and settings intact.

EC2 Image Builder

How it works

EC2 Image Builder
Start with a base image

Base image

Customize software installed on the image

Secure image with AWS-provided and/or custom templates

Test image with AWS-provided tests and/or your own test

Distribute "golden" image to selected AWS regions

- **Automate VM or Image Creation**
  - **creation, testing, and deployment of AMIs**

- **Can be configured to run at regular intervals (e.g., daily, weekly, or monthly)**

- **Free**