
# AMI Building Strategy



In this lesson, you will learn about strategies and recommendations on how to create an Amazon Machine Image (AMI).

# What you will learn

## At the core of the lesson

**You will learn how to:**

- Describe the value proposition of using an Amazon Machine Image (AMI)
- Create an AMI
- Explain the characteristics of an AMI
- Identify recommendations for creating a Microsoft Windows AMI

**Key terms:**

- Full AMI
- Hybrid AMI
- OS-only AMI

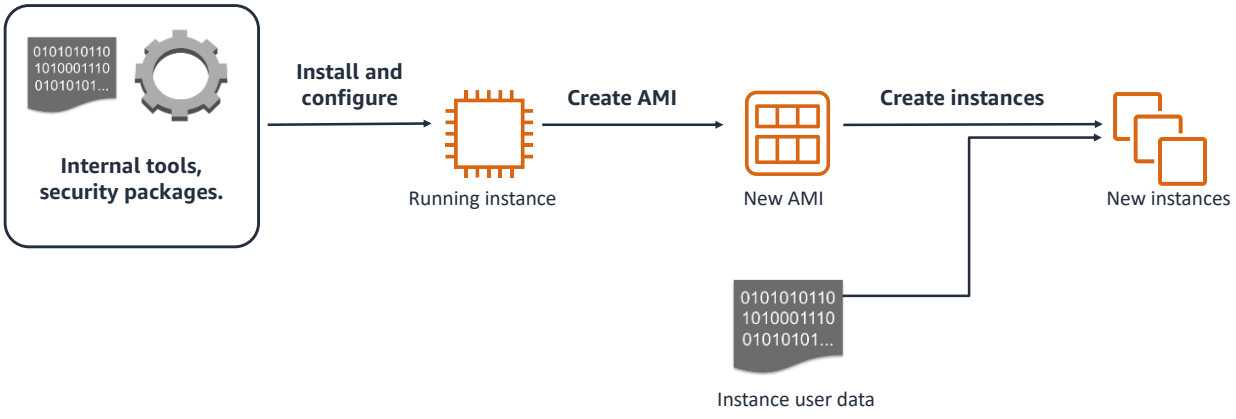




In this lesson, you will learn how to:

- Describe the value proposition of using an Amazon Machine Image (AMI)
- Create an AMI
- Explain the characteristics of an AMI
- Identify recommendations for creating a Microsoft Windows AMI

## Custom AMIs as a base configuration






aws re/start

Suppose that your organization has requested that all EC2 instances launched in the AWS Cloud should have a base set of software pre-installed on them. This might include utilities the organization develops, in-house tools for using AWS services, and advanced software for enterprise-scale activities, such as monitoring and intrusion detection.

Given such requirements, you might want to consider developing one or more custom AMIs as a base configuration. To accomplish this task, you follow these steps:

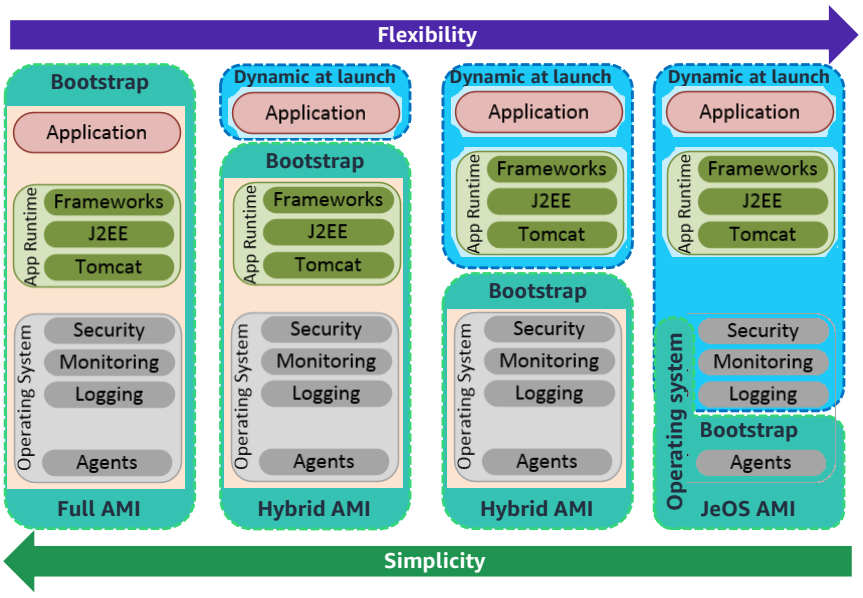
1. Launch an EC2 instance from a standard AMI.
2. Preconfigure all the software that your organization requires on an Amazon EC2 instance.
3. Create a custom AMI from that instance.

The new custom AMI then becomes the AMI that is used to create all new instances in the organization.

To enforce the policy that all new instances are launched only from the new base AMI, do the following:

- Create processes that scan the running Amazon EC2 instances in your account.
- Terminate any instances that are not using the standard AMIs.

## Configure instances at boot time






aws re/start

Another option is to configure instances at boot time. An example of configuring an instance at boot time is the use of the *user data* option to run a script when you launch an EC2 instance.

As shown in the graphic, this approach is compatible with creating custom AMIs. Many organizations settle on a hybrid approach, whereby some configurations are built into a custom base AMI, and other settings are configured dynamically at launch.

The ideal approach typically is determined by considering the tradeoffs between simplicity and flexibility.

Consider these factors:

- **Build times** – An AMI with pre-installed prerequisites and configurations lengthens the amount of time it takes to produce a build.

- **Boot times** – An AMI with an operating system (OS)-only configuration takes a long time to boot when a new instance is launched. Packaging prerequisites into a custom AMI shortens boot times.

- **Shelf life** – When you install more prerequisites on an AMI, you run a greater risk that your application will be vulnerable to a security risk. The risk exists if the underlying AMI is not frequently updated with security or application updates. Assess the risk that updates to your dependencies pose.

In summary, each approach creates tradeoffs:

- **Full AMI** – The applications and all dependencies are pre-installed, which shortens boot times but increases build times. Full AMIs typically have a shorter lifespan. Consider your rollback strategy.

- **Partially configured AMIs** – Only prerequisite software and utilities are pre-installed, which leads to a longer shelf life for the AMI. This approach provides a balance between boot speed and build time. Rollbacks become easier.

- **OS-only AMI** – This approach is fully configurable and upgradeable over time and shortens build times. However, it makes your EC2 instances slow to boot because all required installations and configurations must be run at boot time.

For more information about AMI design, refer to Best practices for building AMIs in the *AWS Marketplace Seller Guide.*

## Creating AMIs

Creation of AMIs results in the following:

- Resulting AMI is anchored to the current *AWS Region*.
- Instance is rebooted by default to ensure consistency.
- Amazon Elastic Block Store (Amazon EBS)-backed AMIs are created with all attached volumes.

```
aws ec2 create-image --instance-id i-1234567890abcdef0
--name "Our_Base_Image-2018-09-17"
```





To create an AMI, you can use any one of the following tools:

- AWS Management Console
- AWS Command Line Interface (AWS CLI)
- AWS application programming interface (API)

**AMI anchored to current Region**

The resulting AMI exists only in the current AWS Region. For example, if you create an AMI from an EC2 instance that is running in the eu-west-2 Region, the resulting AMI exists only in the eu-west-2 Region. You can launch a new instance from the AMI in the us-east-1 Region only if you copy the AMI to another Region first, as described later.

**Automatic reboot of Instance**

By default, when you create an AMI from an instance, the instance is rebooted to ensure consistency. You can override this default behavior. However, if you choose this option, AWS cannot facilitate the integrity of the file system of the created image.

**AMIs with attached EBS volumes**

Also, if you create an AMI from an EC2 instance with additional volumes attached to it, the attached volumes are captured as part of the process of creating the AMI.

The example command shown demonstrates how to create an AMI by using the AWS

CLI. When you use the `create-image` command, you must specify the instance ID. Also specify a name for the AMI that will be created.

Expected result:

```
{
    "ImageId”: "ami-1234567890abcdef0"
}
```

## Copying AMIs to different AWS Regions

Considerations for copying AMIs:

- Copy the AMI explicitly.
- Use one of the following copy methods:
  - On the AWS Management Console, choose **Copy AMI**.
  - From the command line, run `aws ec2 copy-image`.

```
aws ec2 copy-image --source-image-id ami-1234567890abcdef0
--source-region us-east-1 --region ap-northeast-1 --name "My server"
```



aws re/start

An AMI is anchored at the *Region* level. As a result, if you want to launch an EC2 instance from an AMI that you created in a different Region, you must first copy the AMI to the target Region. You can copy an AMI in or across an AWS Region by using one of the following methods:

- AWS Management Console
- AWS CLI
- Amazon EC2 API

In the AWS CLI example, the `source-image-id` and `source-region` are both specified. Also, the Region where the AMI is copied is identified by the `region` parameter. The name assigned to the new AMI is also specified.

Expected result:

```
{
    "ImageId": "ami-abcdef01234567890"
}
```

You can copy both Amazon EBS-backed AMIs and instance store-backed AMIs. You can copy encrypted AMIs and AMIs with encrypted snapshots.

**Encrypted snapshots**

You can encrypt snapshots with your default AWS Key Management Service (AWS

KMS) customer master key (CMK) or a custom key that you specify. In all cases, you must have permission to use the selected key. If you have an AMI with encrypted snapshots, you can choose to re-encrypt them with a different encryption key as part of the `CopyImage` action. `CopyImage` accepts only one key at a time and encrypts all of an image's snapshots (whether root or data) to that key. However, you can manually build an AMI with snapshots encrypted to multiple keys.

**Note:** A copy might fail if AWS cannot find a corresponding Amazon Kernel Image (AKI) in the target Region.

For more information, refer to the following pages in the *Amazon EC2 User Guide for Linux Instances:*

- Using encryption with EBS-backed AMIs
- Copying an AMI

## AMI creation details

- Costs are incurred for Amazon EBS snapshots of volumes stored in Amazon S3.
- Create Linux AMIs directly from an Amazon EC2 instance root volume snapshot using one of two tools:
  - AWS Management Console
  - AWS CLI command: `aws ec2 register-image`

```
aws ec2 register-image –root-device-name '/dev/sda1' --name
"MyImage"
```



aws re/start

When you create an AMI, Amazon EC2 creates snapshots of your instance's root volume and any other EBS volumes attached to your instance. You are charged for the snapshots until you deregister the AMI and delete the snapshots.

To create an Amazon EBS-backed Linux AMI, start from an instance that you launched from an existing Amazon EBS-backed Linux AMI. This AMI can be an AMI you obtained from the AWS Marketplace, an AMI you imported, or any other AMI you can access.

You can also create Linux AMIs directly from the EC2 instance root volume snapshot, for example, by running the AWS CLI command: `aws ec2 register-image`.

Expected result:

```
{
    "ImageId": "ami-0123456789"
}
```

## Creating Microsoft Windows AMIs

Best practices for creating Microsoft Windows AMIs:

- Run the `Sysprep` tool on an EC2 instance before you create an AMI from it.
- For Windows Server 2016 or later, run Sysprep with **EC2Launch**.
- For versions of Windows Server earlier that 2016, run Sysprep with **EC2Config**.





The Microsoft System Preparation (Sysprep) tool simplifies the process of duplicating a customized installation of Windows. We recommend that you use Sysprep to create a standardized AMI. You can then create new Amazon EC2 instances for Windows from this standardized image. We also recommend that you run Sysprep with EC2Launch (Windows Server 2016 and later) or the EC2Config service (before Windows Server 2016).

**EC2 Launch**
The *EC2Launch* service starts when the instance boots and performs tasks during startup. Examples of the tasks include setting the computer name, sending instance information to the EC2 console, setting a random password for the Windows Administrator user account, and more.

For more information about EC2Launch, refer to Configuring a Windows instance using EC2Launch in the *Amazon EC2 User Guide for Windows Instances.*

**EC2Config**
Microsoft Windows AMIs for Windows Server 2012 R2 and earlier include an optional service, the EC2Config service (EC2Config.exe). *EC2Config* starts when the instance boots. It performs tasks during startup and each time you stop or start the instance. EC2Config can also perform tasks on demand. Some of these tasks are automatically enabled, whereas others must be enabled manually. Although this service is optional,

it provides access to advanced features that are not otherwise available.

For more information, see “Create a standardized Amazon Machine Image (AMI) using Sysprep” in the *Amazon EC2 User Guide for Windows Instances* (https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/Creating_EBSbacked_WinAMI.html#ami-create-standard).

However, do not use Sysprep to create an EC2 instance backup. Sysprep removes system-specific information. Removing this information might have unintended consequences for an instance backup. Sysprep is used to prepare an ec2 image for AMI creation by removing the image-specific tools, configuration, and user identification within the Windows system.

## Sysprep phases

**Sysprep runs through the following phases:**






Sysprep runs through the following phases:

- **Generalize** – The tool removes image-specific information and configurations. For example, Sysprep removes the security identifier (SID), the computer name, the event logs, and specific drivers, to name a few. After this phase is completed, the operating system (OS) is ready to create an AMI.

- **Specialize** – Plug and Play scans the computer and installs drivers for any detected devices. The tool generates OS requirements, such as the computer name and SID. You can run commands in this phase.

- **Out-of-Box Experience (OOBE)** – The system runs an abbreviated version of Windows Setup. It asks the user to enter information such as a system language, the time zone, and a registered organization. When you run Sysprep with EC2Config, the answer file automates this phase.

## Checkpoint questions

1. What is an Amazon Machine Image (AMI)?
2. What is the purpose of the User Data option?
3. A new AMI is created in the us-east-1 Region and copied to the us-east-2 Region. A script that was used to launch EC2 instances in us-east-1 is then used to launch instances in us-east-2. When the script is run, it fails.

   What could be the problem?





**Answers**

1. An Amazon Machine Image (AMI) provides the information required to launch an instance. You must specify an AMI when you launch an instance. If you need multiple instances with the same configuration, you can launch multiple instances from a single AMI. If you need instances with different configurations, you can use different AMIs to launch instances.

2. Use user data to perform common automated configuration tasks and even run scripts after the instance starts.

3. The names of the AMIs are different between Regions. The script that is running in us-east-2 is probably using the AMI name from us-east-1.

## Key takeaways


- Given an organization's requirements, you can develop one or more custom AMIs as a base configuration.
- Instances can be configured at boot time.
- When you create an AMI, it creates a snapshot per volume. Storage and data retrieval costs are incurred for snapshots of Amazon EBS volumes stored in Amazon S3.





aws re/start

Key takeaways from this module include:

- Given an organization’s requirements, you can develop one or more custom AMIs as a base configuration.
- Instances can be configured at boot time.
- When you create an AMI, it creates a snapshot per volume. Storage and data retrieval costs are incurred for snapshots of Amazon EBS volumes stored in Amazon S3.