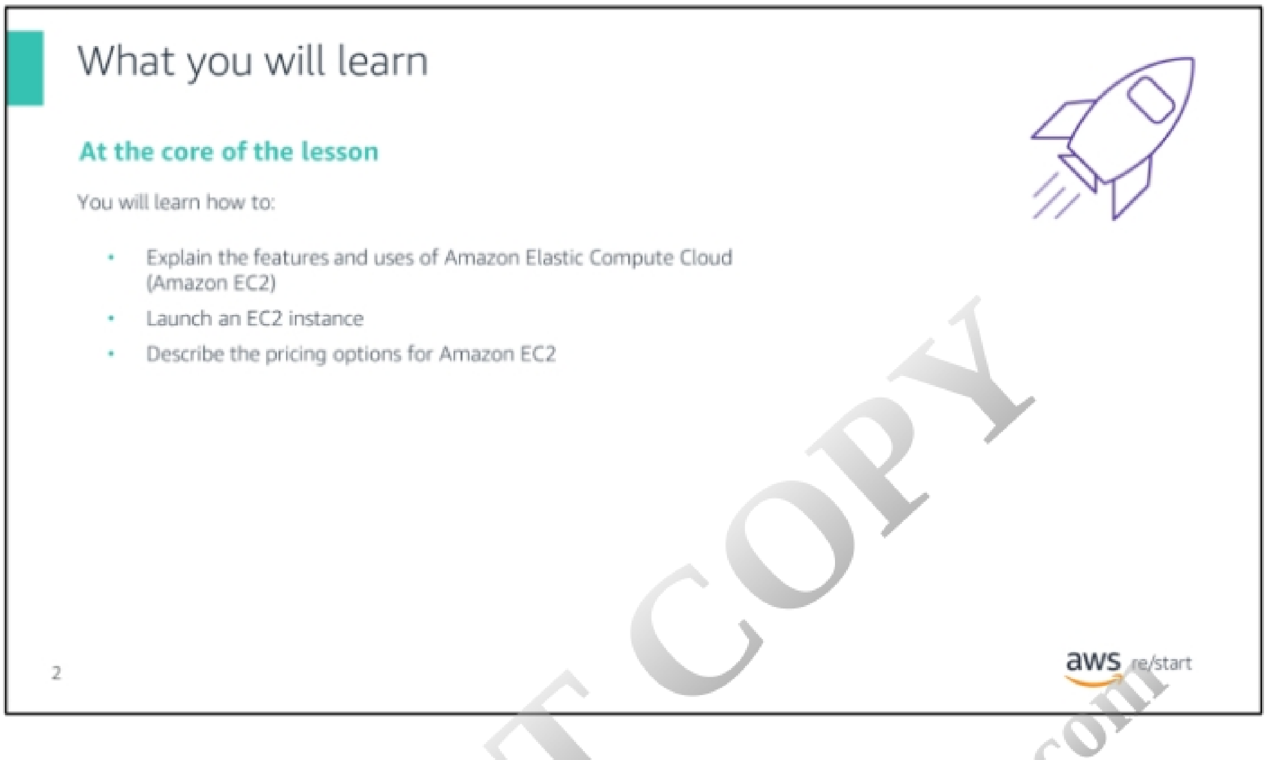
aws re/start

Amazon Elastic Compute Cloud



What you will learn

At the core of the lesson

You will learn how to:

- Explain the features and uses of Amazon Elastic Compute Cloud (Amazon EC2)
- Launch an EC2 instance
- Describe the pricing options for Amazon EC2

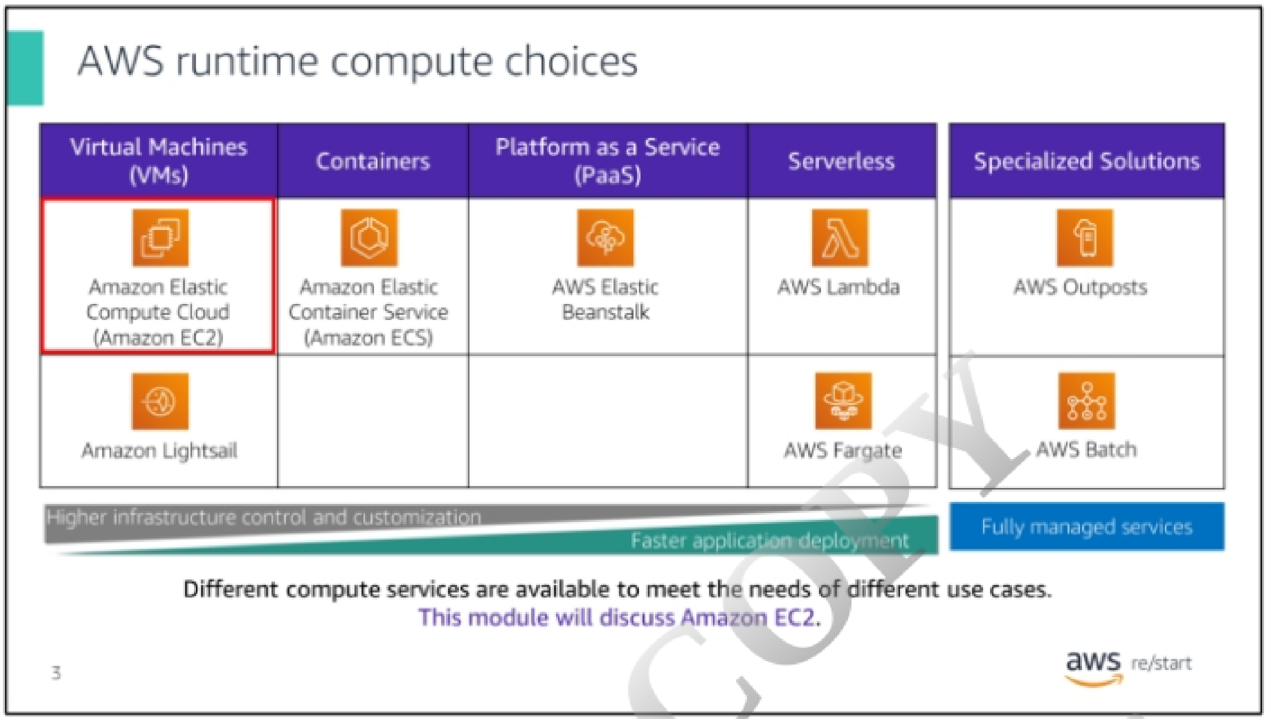
2

aws re/start

Amazon Web Services (AWS) provides multiple services to build a solution. Some of those services provide the foundation to all solutions, which are also known as the **core services**. This module provides insight into the offerings of each service category and looks at the first group of services, compute.

You will first get an overview of compute services. Then, you will learn about Amazon Elastic Compute Cloud (or Amazon EC2).

Whether you want to build mobile apps or run massive clusters to sequence the human genome, building and running your business starts with compute. AWS has a broad catalog of compute services. It offers everything from simple application services to flexible virtual servers, and even serverless computing.



AWS runtime compute choices

Virtual Machines (VMs)	Containers	Platform as a Service (PaaS)	Serverless	Specialized Solutions
Amazon Elastic Compute Cloud (Amazon EC2)	Amazon Elastic Container Service (Amazon ECS)	AWS Elastic Beanstalk	AWS Lambda	AWS Outposts
Amazon Lightsail			AWS Fargate	AWS Batch

Higher infrastructure control and customization
Faster application deployment
Fully managed services

Different compute services are available to meet the needs of different use cases.
This module will discuss Amazon EC2.

AWS offers several compute options to meet different needs. When you consider the service to use for a given type of workload, it's important that you understand the available compute options. As the diagram shows, the key runtime compute choices can be grouped into four categories of cloud-computing models: *virtual machines (VMs); containers; platform as a service*, which is also known as *PaaS*; and *serverless*. In addition, you can use *specialized solutions* to address specific compute use cases.

In the *virtual machines* category, AWS offers two core services. The first service is *Amazon Elastic Compute Cloud (Amazon EC2)*. It provides secure and resizable virtual servers in the cloud. The second service is Amazon Lightsail. It provides virtual private servers to run simple workloads in a cost-effective way.

In the containers category, AWS offers *Amazon Elastic Container Service (Amazon ECS)*. It enables you to run Docker container applications on AWS.

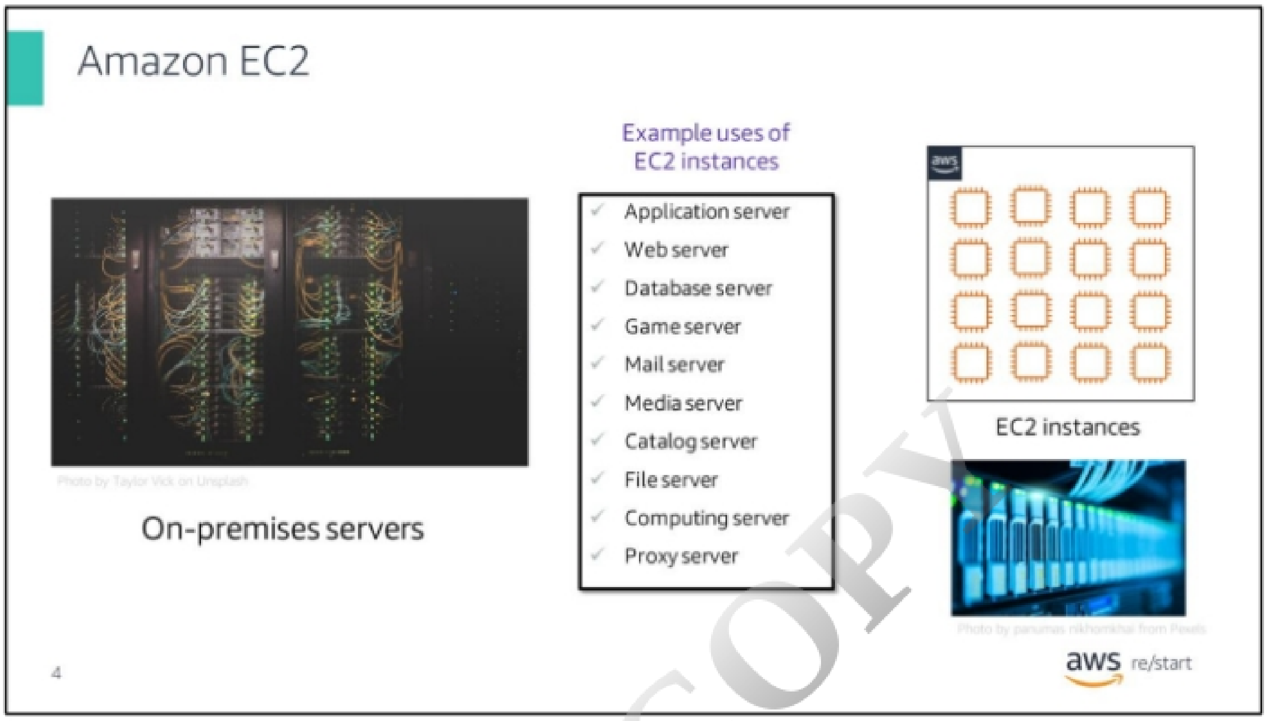
The *PaaS* category includes *AWS Elastic Beanstalk*. It's a solution that runs web applications and services that are developed in languages such as Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker.

The *serverless* category includes *AWS Lambda*, which is a serverless compute solution that runs Java, Go, PowerShell, Node.js, C#, Python, or Ruby code. This category also includes *AWS Fargate*, which provides a serverless compute platform for containers.

For specialized solutions, *AWS Outposts* provides a way to run AWS infrastructure and services on-premises, and *AWS Batch* is a service that runs batch jobs at any scale.

When you select an AWS compute runtime for your workload, consider that virtual machines and container-based services provide more control over your infrastructure and enable higher degrees of customization. PaaS and serverless services enable you to focus more on your application and less on infrastructure. They also enable quick deployment. The services in the specialized solutions category address specific types of workloads, or *hybrid cloud* and *batch*. These specialized services work well for these use cases because they are also fully managed by AWS.

Amazon EC2 is one of the core AWS services and is the focus of this module.



Amazon EC2

Photo by Taylor Vick on Unsplash

On-premises servers

Example uses of EC2 instances

- ✓ Application server
- ✓ Web server
- ✓ Database server
- ✓ Game server
- ✓ Mail server
- ✓ Media server
- ✓ Catalog server
- ✓ File server
- ✓ Computing server
- ✓ Proxy server

EC2 instances

Photo by panumas nikhomkhai from Pexels

4

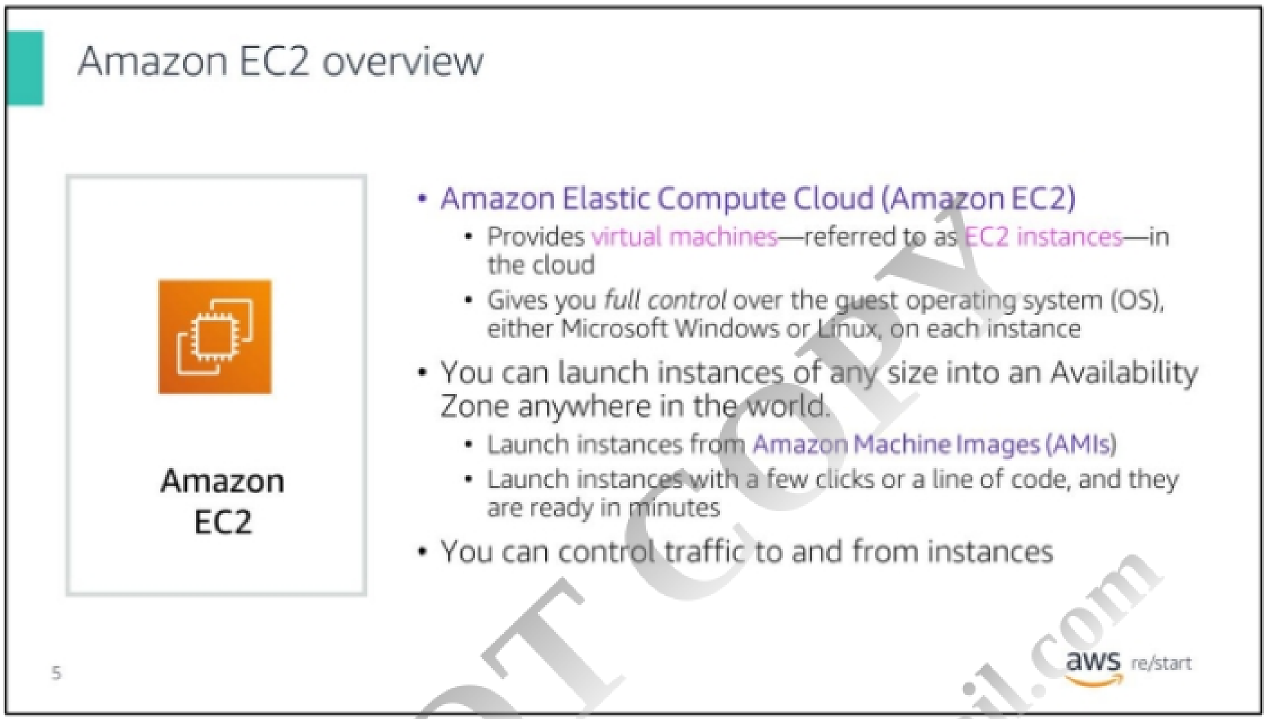
aws re/start

Running on-premises servers is an expensive undertaking. Hardware must be procured, and this procurement can be based on project plans instead of on the reality of how the servers are used. Data centers are expensive to build, staff, and maintain. Organizations also need to permanently provision a sufficient amount of hardware to handle traffic spikes and peak workloads. After traditional on-premises deployments are built, server capacity might be unused and idle for a significant portion of the time that the servers are running, which is wasteful.

Amazon EC2 provides virtual machines where you can host the same kinds of applications that you might run on a traditional on-premises server. It provides secure, resizable compute capacity in the cloud. EC2 instances can support a variety of workloads. Common uses for EC2 instances include:

- Application servers
- Web servers
- Database servers
- Game servers
- Mail servers
- Media servers
- Catalog servers
- File servers

- Computing servers
- Proxy servers



Amazon EC2 overview

- Amazon Elastic Compute Cloud (Amazon EC2)
  - Provides virtual machines—referred to as EC2 instances—in the cloud
  - Gives you *full control* over the guest operating system (OS), either Microsoft Windows or Linux, on each instance
- You can launch instances of any size into an Availability Zone anywhere in the world
  - Launch instances from Amazon Machine Images (AMIs)
  - Launch instances with a few clicks or a line of code, and they are ready in minutes
- You can control traffic to and from instances

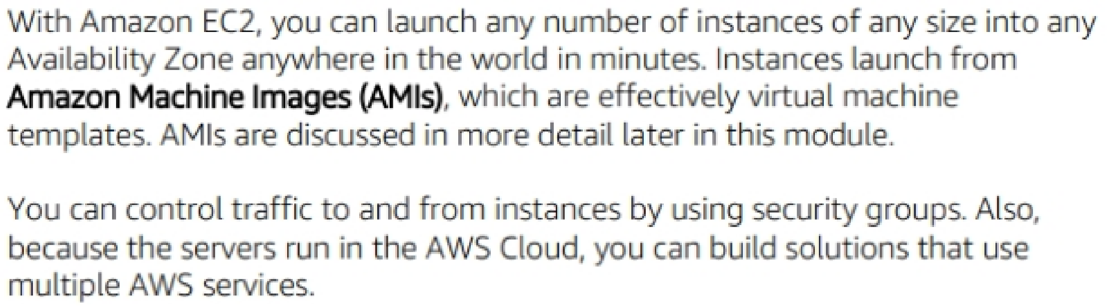
5

The **EC2** in Amazon EC2 stands for **Elastic Compute Cloud**:

- **Elastic** refers to the fact that you can easily increase or decrease the number of servers that you run to support an application automatically. You can also increase or decrease the size of existing servers.
- **Compute** refers to reason why most users run servers—to host running applications or process data. These actions require compute resources, including processing power (CPU) and memory (RAM).
- **Cloud** refers to the fact that the EC2 instances that you run are hosted in the cloud.

Amazon EC2 provides virtual machines in the cloud and gives you full administrative control over the Microsoft Windows or Linux operating system that runs on the instance. Most server operating systems are supported, including: Windows 2008, 2012, 2016, and 2019, Red Hat, SuSE, Ubuntu, and Amazon Linux.

An operating system (OS) that runs on a virtual machine is often called a *guest OS* to distinguish it from the host OS. The host OS is directly installed on any server hardware that hosts one or more virtual machines.



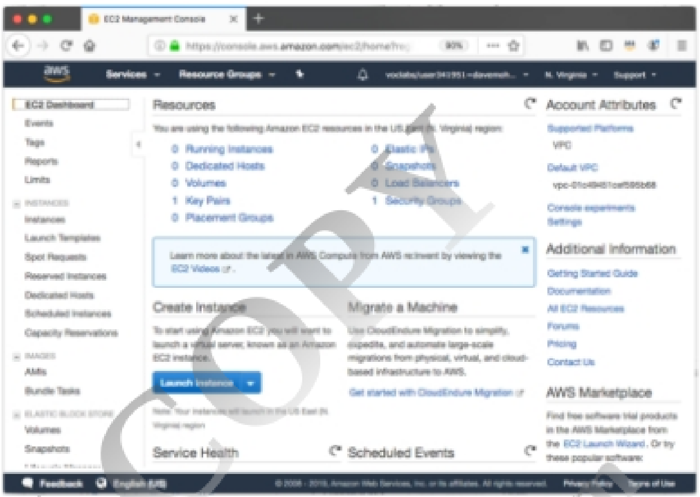
With Amazon EC2, you can launch any number of instances of any size into any Availability Zone anywhere in the world in minutes. Instances launch from **Amazon Machine Images (AMIs)**, which are effectively virtual machine templates. AMIs are discussed in more detail later in this module.

You can control traffic to and from instances by using security groups. Also, because the servers run in the AWS Cloud, you can build solutions that use multiple AWS services.

Launching an EC2 instance

This section of the module walks through nine key decisions to make when you create an EC2 instance by using the AWS Management Console Launch Instance Wizard.

- Along the way, essential Amazon EC2 concepts will be explored.



You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

0 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
0 Volumes	0 Load Balancers
1 Key Pairs	1 Security Groups
0 Placement Groups	

Learn more about the latest in AWS Compute from AWS re:Invent by viewing the EC2 Videos.

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US East (N. Virginia) region

Use CloudEndure Migration to simplify, expedite, and automate large-scale migrations from physical, virtual, and cloud-based infrastructure to AWS.

Get started with CloudEndure Migration

Find free software trial products in the AWS Marketplace from the EC2 Launch Wizard. Or try these popular software:

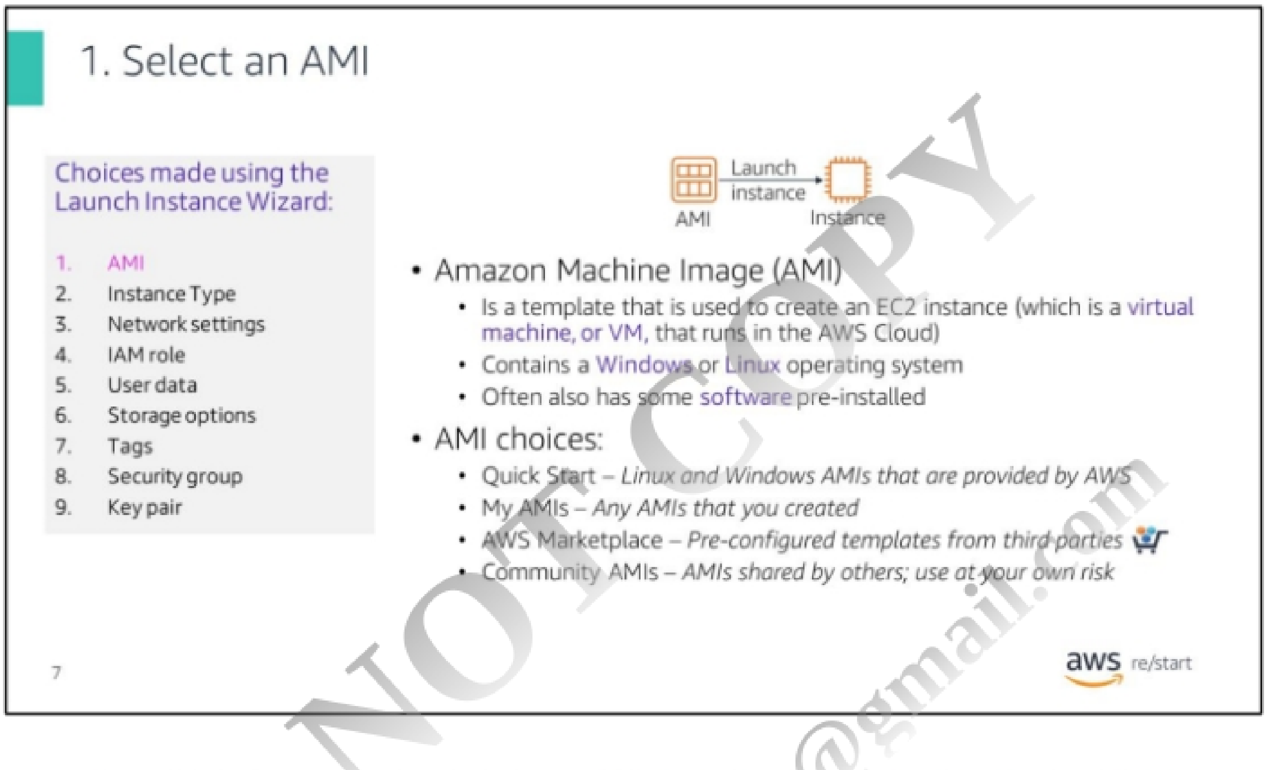
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The first time that you launch an EC2 instance, you will likely use the *AWS Management Console Launch Instance Wizard*. An example of this process is shown in the demonstration at the end of this section.

The **Launch Instance Wizard** makes it easy to launch an instance. For example, if you choose to accept all the default settings, you can skip most of the steps that are provided by the wizard and launch an EC2 instance in a few clicks.

However, for most deployments, you will want to modify the default settings so that the servers you launch are deployed in a way that matches your specific needs.

The next series of slides introduce you to the essential choices that you must make when you launch an instance. The slides cover essential concepts that are good to know when you make these choices. You learn about these concepts to help you understand the options that are available, and the effects of the decisions that you make.


AMI
Launch instance
Instance
aws re/start

1. Select an AMI

Choices made using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair

- Amazon Machine Image (AMI)
  - Is a template that is used to create an EC2 instance (which is a virtual machine, or VM, that runs in the AWS Cloud)
  - Contains a Windows or Linux operating system
  - Often also has some software pre-installed
- AMI choices:
  - Quick Start – *Linux and Windows AMIs that are provided by AWS*
  - My AMIs – *Any AMIs that you created*
  - AWS Marketplace – *Pre-configured templates from third parties*
  - Community AMIs – *AMIs shared by others; use at your own risk*

7

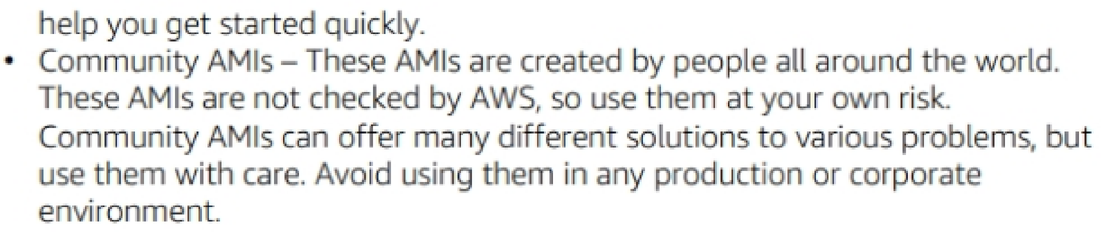
An **Amazon Machine Image (AMI)** provides information that is needed to launch an EC2 instance. You must specify a source AMI when you launch an instance. You can use different AMIs to launch different types of instances. For example, you can choose one AMI to launch an instance that will become a web server and another AMI to deploy an instance that will host an application server. You can also launch multiple instances from a single AMI.

An AMI includes the following components:

- A **template for the root volume** of the instance. A root volume typically contains an OS and everything that was installed in that OS (applications, libraries, and so on). Amazon EC2 copies the template to the root volume of a new EC2 instance, and then starts it.
- **Launch permissions** that control which AWS accounts can use the AMI.
- A **block device mapping** that specifies the volumes to attach to the instance (if any) when it is launched.

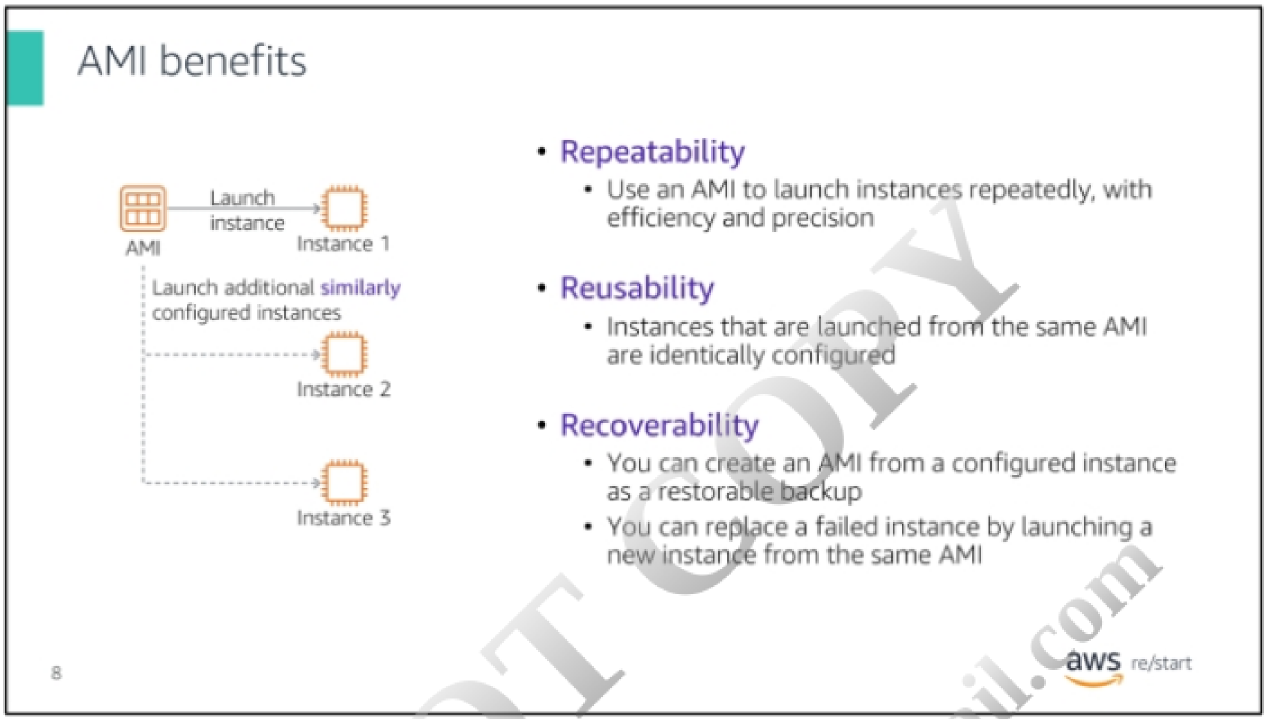
You can choose many AMIs:

- Quick Start – AWS offers a number of pre-built AMIs for launching your instances. These AMIs include many Linux and Windows options.
- My AMIs – These AMIs are AMIs that you created.
- AWS Marketplace – The AWS Marketplace offers a digital catalog that lists thousands of software solutions. These AMIs can offer specific use cases to



help you get started quickly.

- Community AMIs – These AMIs are created by people all around the world. These AMIs are not checked by AWS, so use them at your own risk. Community AMIs can offer many different solutions to various problems, but use them with care. Avoid using them in any production or corporate environment.



AMI benefits

AMI

Launch instance

Instance 1

Launch additional similarly configured instances

Instance 2

Instance 3

- Repeatability
  - Use an AMI to launch instances repeatedly, with efficiency and precision
- Reusability
  - Instances that are launched from the same AMI are identically configured
- Recoverability
  - You can create an AMI from a configured instance as a restorable backup
  - You can replace a failed instance by launching a new instance from the same AMI

8

aws re/start

An AMI provides the information that is needed to launch an instance. The benefits of using an AMI include *repeatability*, *reusability*, and *recoverability*.

AMIs enable *repeatability* because an AMI packages the full configuration and content of an EC2 instance. As such, it can be used repeatedly to launch multiple instances with efficiency and precision.

AMIs promote *reusability* because instances that are launched from the same AMI are exact replicas of each other. This design makes it easier to build clusters of similar instances or recreate compute environments.

AMIs also facilitate *recoverability*. If an instance fails, you can replace it by launching a new instance from the same AMI that you used to launch the original instance. In addition, AMIs provide a way to back up a complete EC2 instance configuration, which you can use to launch a replacement instance if there is a failure.