



# Adding a Compute Layer Using Amazon EC2

AWS Academy Cloud Architecting



# Introduction

Adding a Compute Layer Using Amazon EC2

# Module objectives

This module prepares you to do the following:

- Identify how to use Amazon Elastic Compute Cloud (Amazon EC2) in an architecture.
- Explain the value of using Amazon Machine Images (AMIs) to accelerate the creation and repeatability of infrastructure.
- Recommend EC2 instance types based on requirements.
- Recommend storage solutions for Amazon EC2.
- Recognize how to configure Amazon EC2 instances with user data.
- Describe EC2 pricing options and make recommendations based on cost.
- Launch an Amazon EC2 instance.
- Use the AWS Well-Architected Framework principles when designing a compute layer with Amazon EC2.

# Module overview

---

## Presentation sections

- Adding compute with Amazon EC2
- Choosing an AMI to launch an EC2 instance
- Selecting an EC2 instance type
- Adding storage to an Amazon EC2 instance
- Other EC2 configuration considerations
- Amazon EC2 pricing options
- Applying the AWS Well-Architected Framework principles to compute

## Demos

- Configuring an EC2 Instance with User Data
- Reviewing the Spot Instance History Page

## Activity

- Choosing Instance Types

## Knowledge checks

- 10-question knowledge check
- Sample exam question

# Hands-on labs in this module

## Guided lab

Introducing Amazon EFS

## Challenge (Café) lab

Creating a Dynamic Website for the Café



# As a cloud architect designing a computer layer using EC2:



- I need to analyze key characteristics of my workload so that I can choose the AMI, EC2 instance type, and attached storage options that optimize the performance and security of my EC2-based workloads.
- I need to choose an EC2 purchasing model that matches my compute use case so that I can optimize the costs of running my workloads.

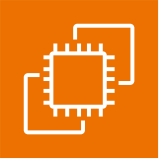








# **Adding compute with Amazon EC2**

Adding a Compute Layer Using Amazon EC2

# AWS runtime compute choices

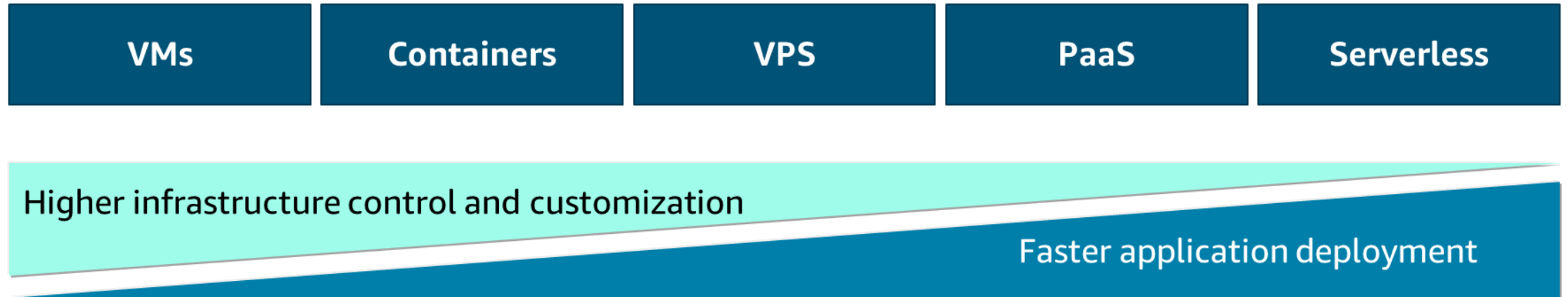
AWS offers different compute services to meet the needs of different use cases.

Virtual Machines (VMs)	Containers	Virtual Private Servers (VPS)	Platform as a Service (PaaS)	Serverless
 Amazon Elastic Compute Cloud (Amazon EC2)	 Amazon Elastic Container Service (Amazon ECS)   Amazon Elastic Kubernetes Service (Amazon EKS)	 Amazon Lightsail	 AWS Elastic Beanstalk	 AWS Lambda   AWS Fargate

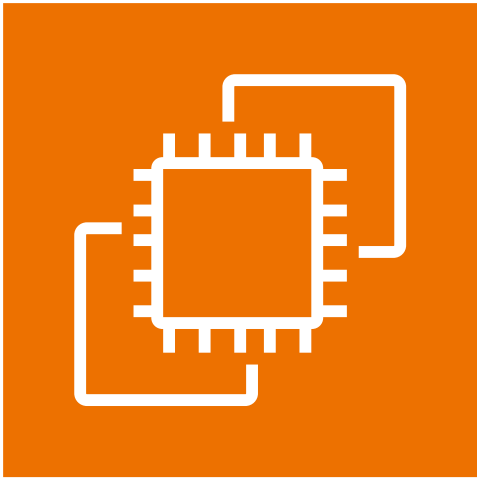


# Compute service category differentiators

Each service category offers different levels of infrastructure control and application deployment speed.



# Amazon EC2

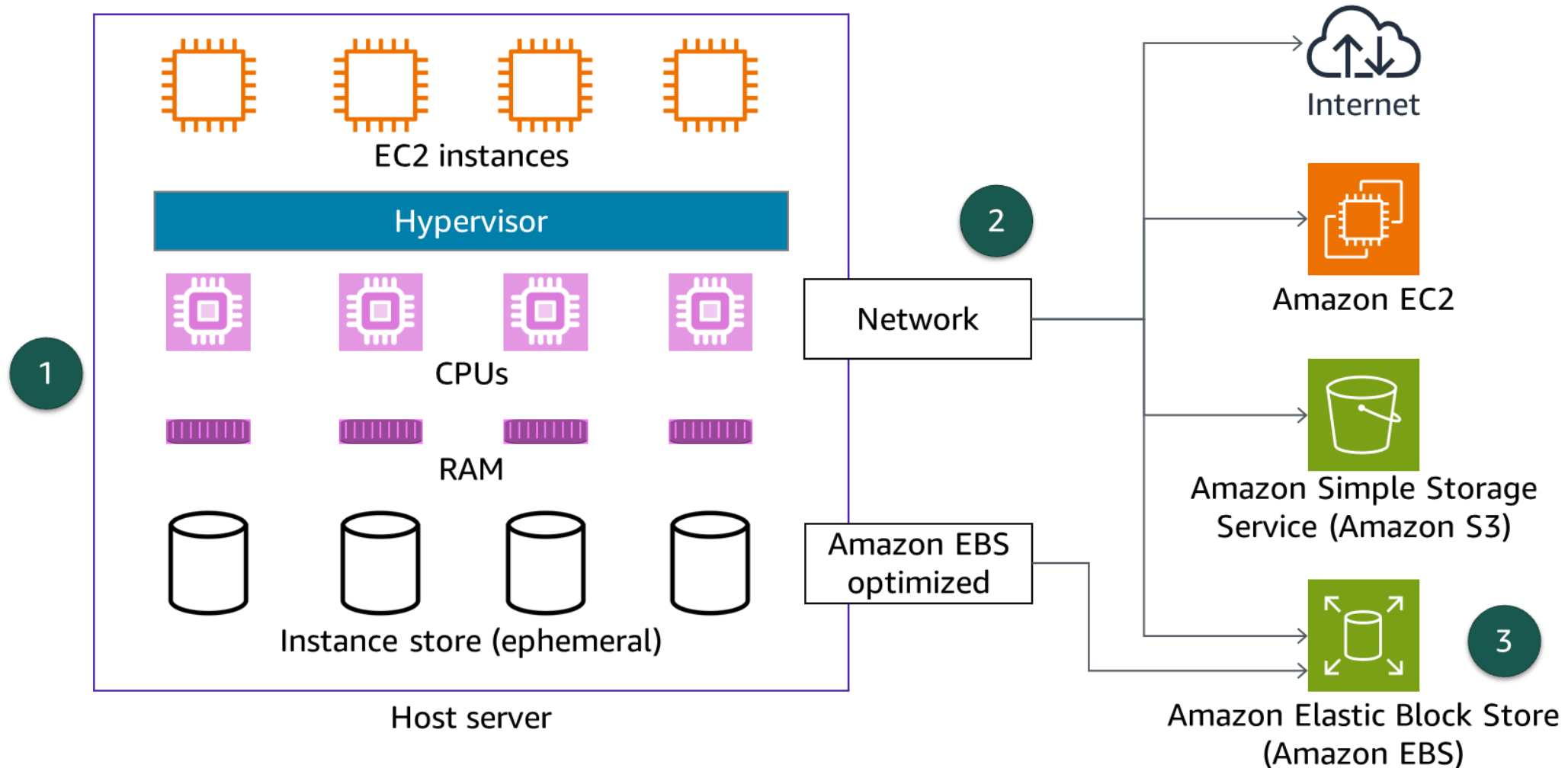


Amazon EC2

- Provides VMs (servers) in the cloud
- Provisions servers in minutes
- Can automatically scale capacity up or down as needed
- Enables you to pay only for the capacity that you use

# Amazon EC2 virtualization

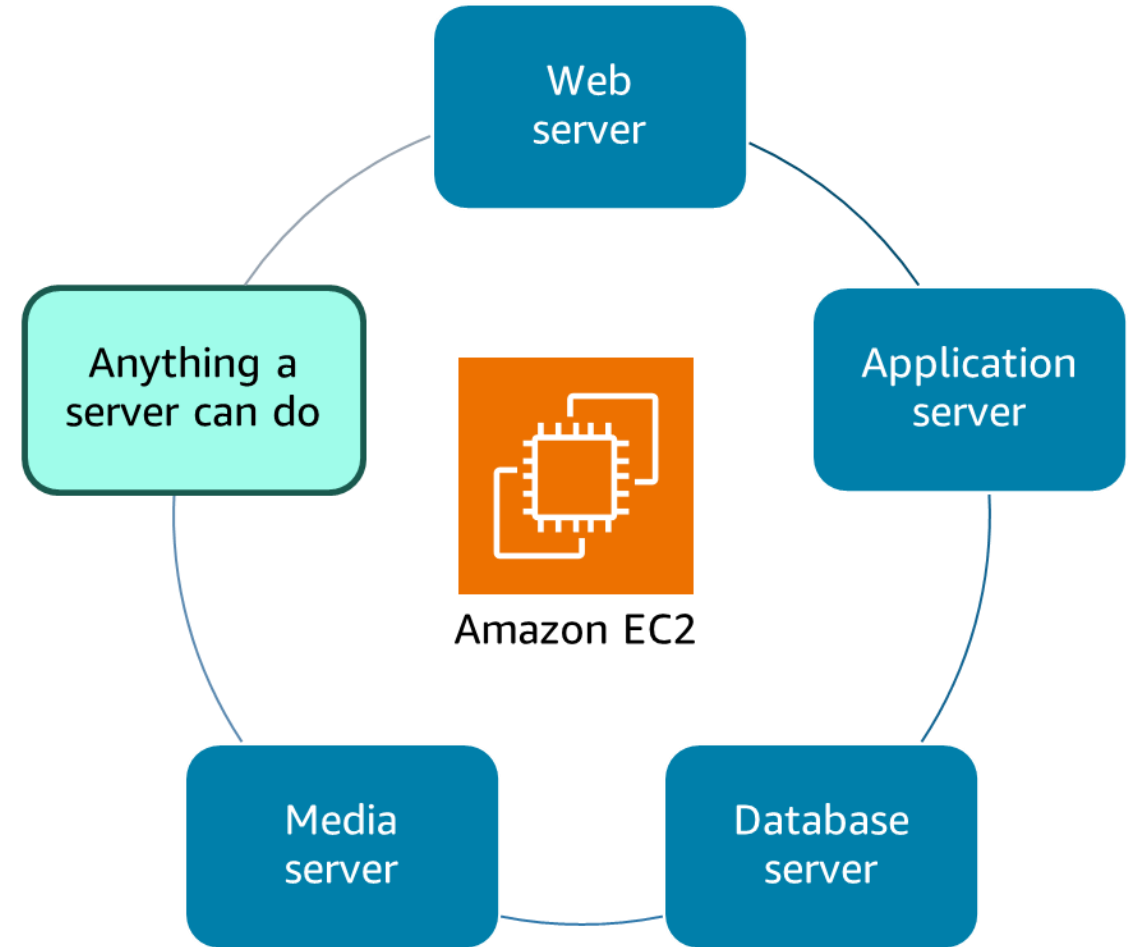
An EC2 instance is a VM that runs on a physical host.



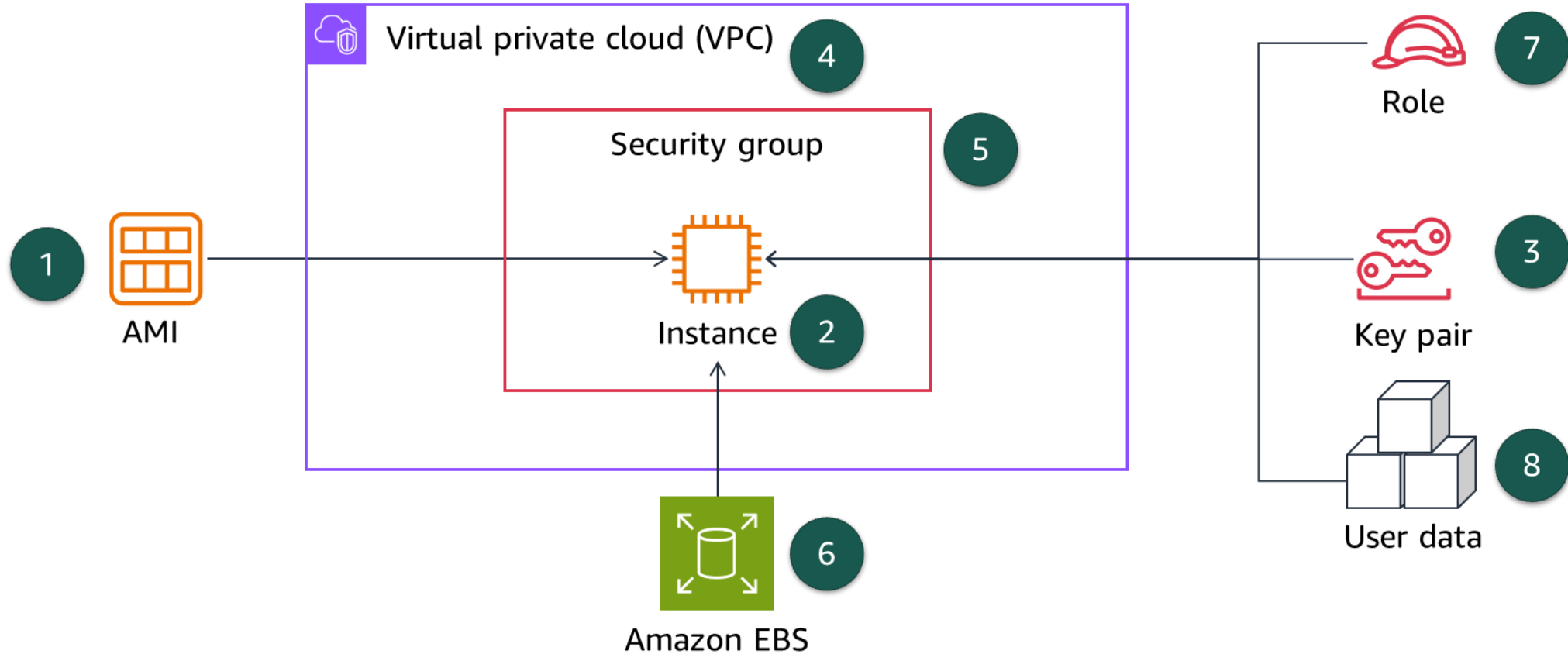
# Amazon EC2 use cases

Use Amazon EC2 when you need the following:

- Complete control of your computing resources, including operating system and processor type.
- Options for optimizing your compute costs
  - On-Demand Instances, Reserved Instances, and Spot Instances
  - Savings Plans
- Ability to run any type of workload
  - Simple websites
  - Enterprise applications
  - Generative artificial intelligence (generative AI) applications



# Steps for provisioning an EC2 instance



# Key takeaways: Adding compute with Amazon EC2



- Amazon EC2 enables you to run VMs in the cloud and easily scale capacity up or down as needed.
- You can use an EC2 instance when you need complete control of your computing resources and want to run any type of workload.
- When you launch an EC2 instance, you must choose an AMI and an instance type. Launching an instance involves specifying configuration parameters including network, security, storage, and user data settings.



# Choosing an AMI to launch an EC2 instance

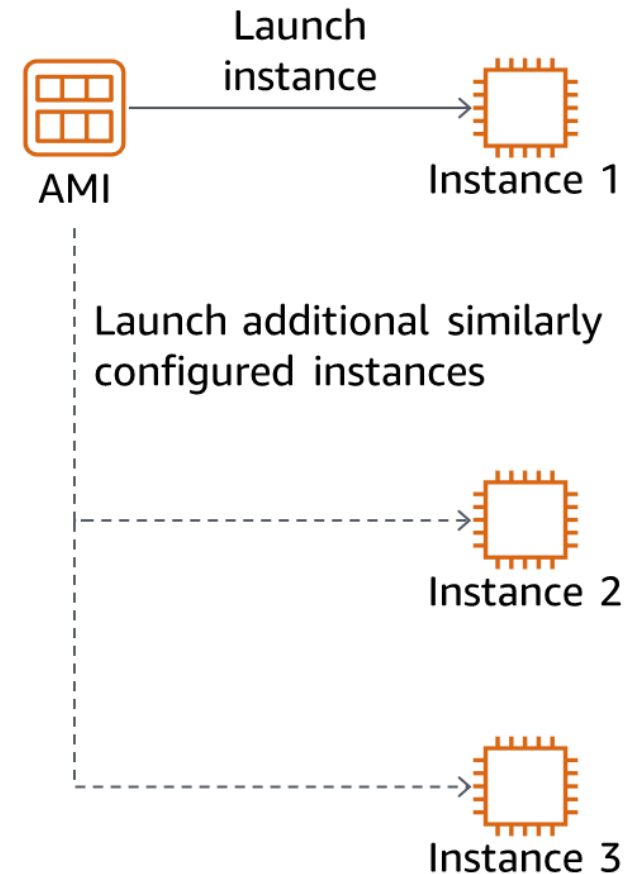
Adding a Compute Layer Using Amazon EC2

# Amazon Machine Image (AMI)

An AMI provides the information that is needed to launch an instance, including the following:

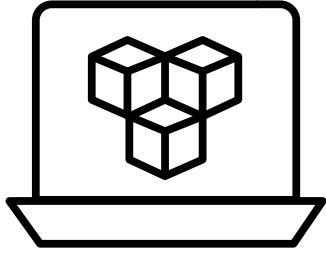
- A template for the root volume: Contains the guest operating system (OS) and perhaps other installed software
- Launch permissions: Controls who can access the AMI
- Block device mappings: Specifies any storage volumes to attach to the instance

Create multiple instances from the same AMI



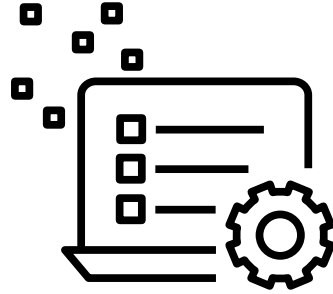


# AMI benefits



## Repeatability

An AMI can be used repeatedly to launch instances with efficiency and precision.



## Reusability

Instances 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.

# Choosing an AMI

## Choose an AMI based on the following:

- Region
- Operating system
- Storage type of the root device
- Architecture
- Virtualization type: For best performance, use an AMI with a Hardware Virtual Machine (HVM) virtualization type.

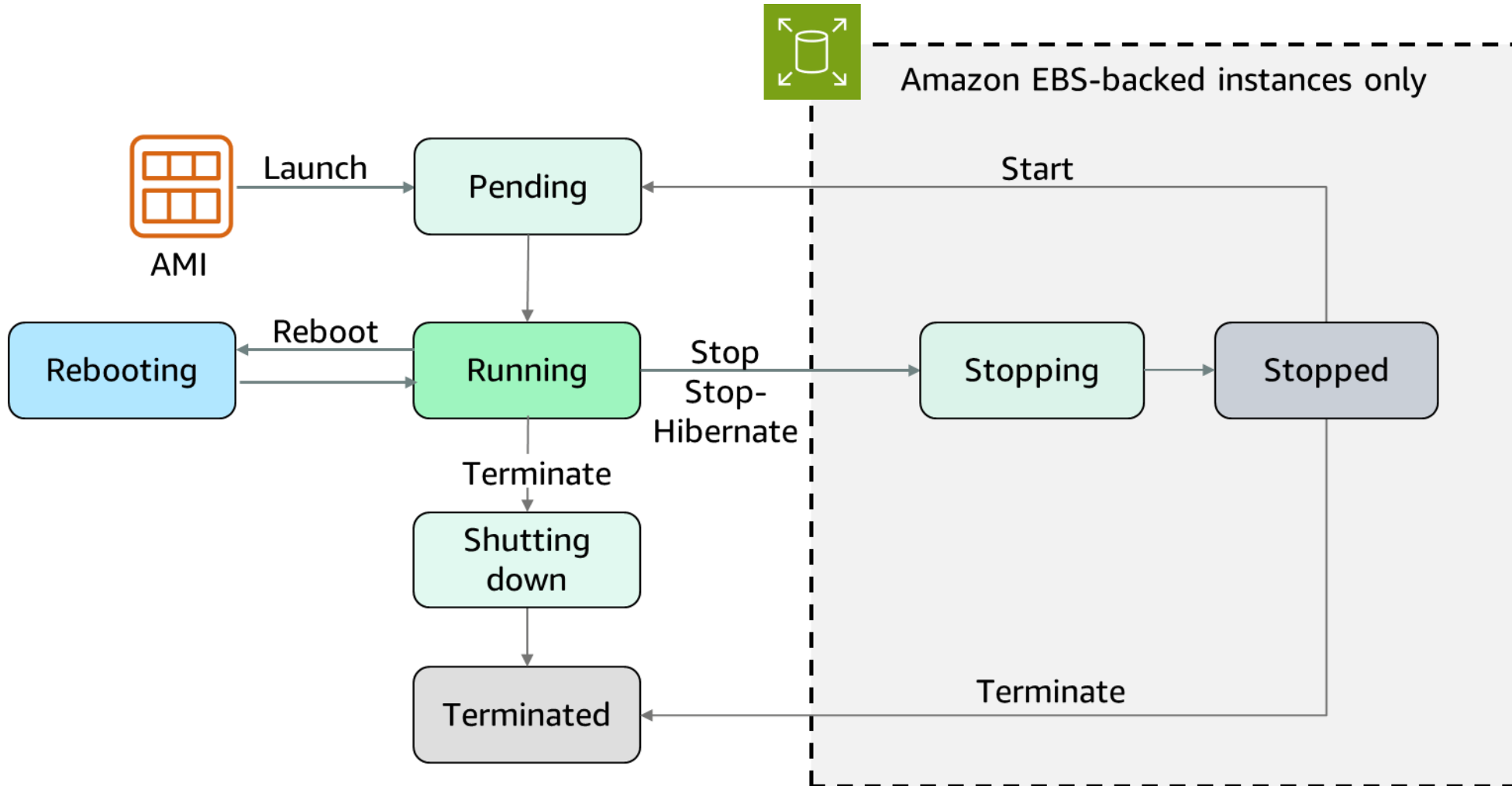
## AMI source:

- Quick Start: Linux and Microsoft Windows AMIs that are provided by AWS.
- My AMIs: Any AMIs that you create.
- AWS Marketplace: Pre-configured templates from third parties.
- Community AMIs: AMIs shared by others. Use at your own risk.

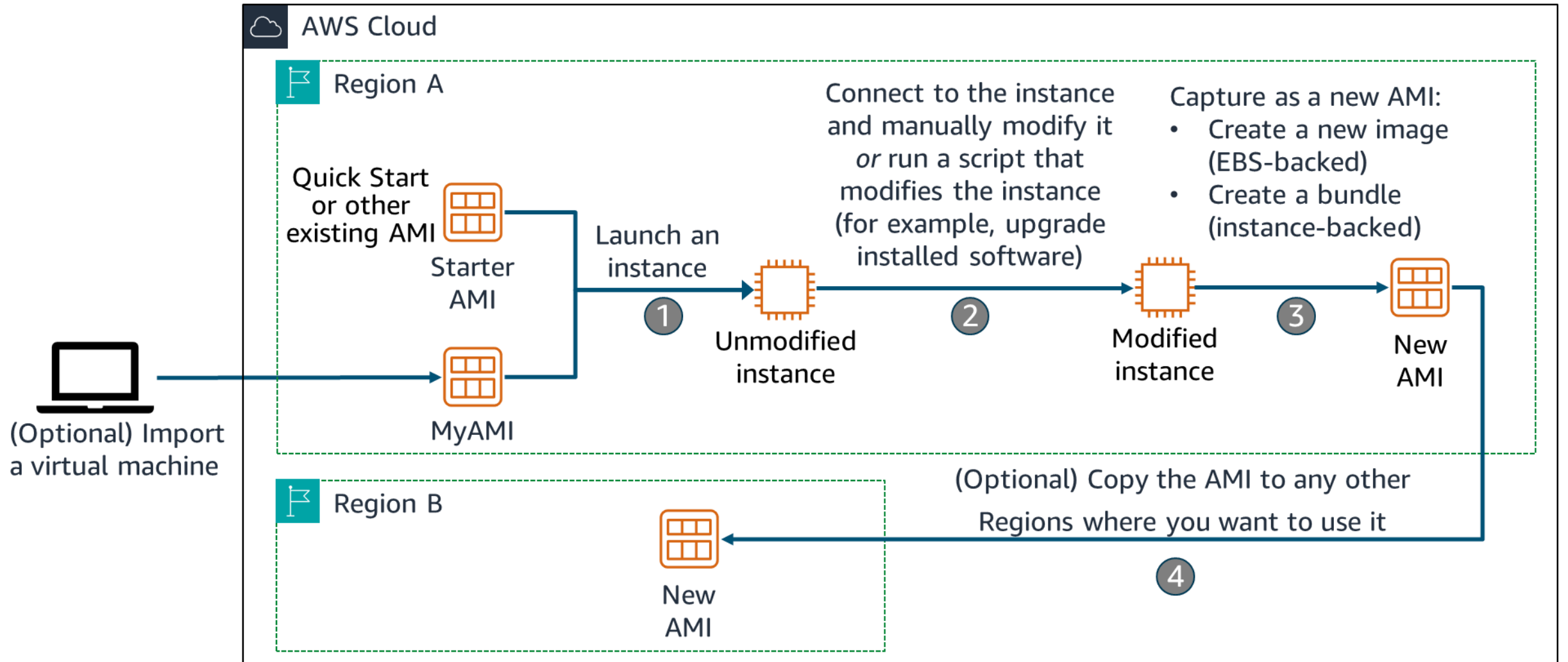
# Instance store-backed versus Amazon EBS-backed AMI

Characteristic	Amazon EBS-Backed Instance	Instance Store-Backed Instance
Boot time for the instance	Boots faster	Takes longer to boot
Maximum size of root device	16 TiB	10 GiB
Ability to stop the instance	Can stop the instance	Cannot be in a stopped state; instances are running or terminated
Ability to change the instance type	Can change the instance type by stopping instance	Can't change the instance type because the instance can't be stopped
Instance charges	You are charged for instance usage, EBS volume usage, and storing your AMI as an EBS snapshot	You are charged for instance usage and storing your AMI in Amazon S3
Use case	Persistent storage	Temporary storage

# Amazon EC2 instance lifecycle



# Creating a new AMI



# EC2 Image Builder



EC2 Image  
Builder

EC2 Image Builder automates the creation, management, and deployment of up-to-date and compliant golden VM images.

- Provides a graphical interface to create image-building pipelines
- Creates and maintains Amazon EC2 AMIs and on-premises VM images
- Produces secure, validated, and up-to-date images
- Enforces version control

# Key takeaways: Choosing an AMI to launch an EC2 instance



- An AMI provides the information that's needed to launch an EC2 instance.
- Benefits of AMIs include repeatability, reusability, and recoverability.
- For best performance, use an AMI with the HVM virtualization type.
- There are multiple sources to get AMIs including Quick Start, Community AMIs, AWS Marketplace, and My AMIs (that store AMIs that you create).



# Selecting an Amazon EC2 instance type

Adding a Compute Layer Using Amazon EC2



# EC2 instance type configuration

An EC2 instance type defines the configuration of CPU, memory, storage, and network performance.

Instance type	vCPU	Memory	Storage	Network performance
m5d.large	2	4 GiB	1 x 50 NVMe SSD	Up to 10 Gbps
m5d.xlarge	4	8 GiB	1 x 100 NVMe SSD	Up to 10 Gbps
m5d.8xlarge	32	128 GiB	2 x 600 NVMe SSD	10 Gbps

All current generation instance types support enhanced networking, except for T2 instances.

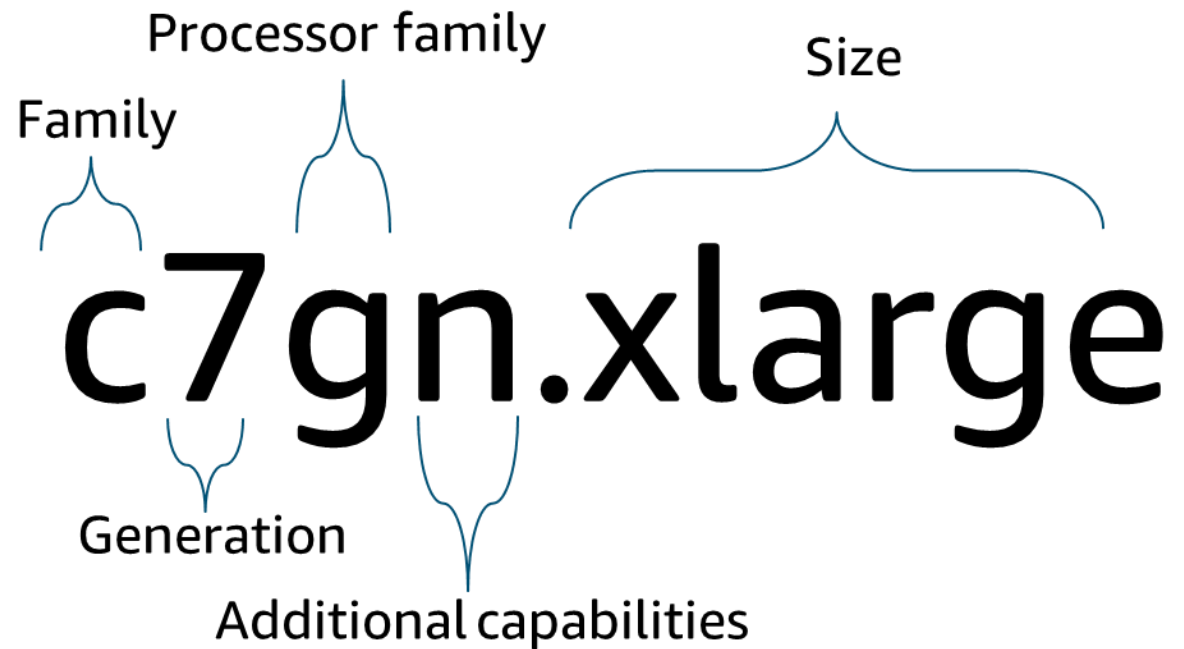


# EC2 instance type name

Instance types named components

- Family
- Generation
- Processor family
- Additional capabilities
- Size

Instance Type Naming:



The diagram illustrates the components of the EC2 instance type name **c7gn.xlarge** using curly braces and labels:

- Family**: Points to the **c** in **c7**.
- Processor family**: Points to the **gn** in **c7gn**.
- Generation**: Points to the **7** in **c7**.
- Additional capabilities**: Points to the **gn** in **c7gn**.
- Size**: Points to the **xlarge** part of the name.

# Suitability of instance types for workloads

Type	Workload examples	Instance type examples
General purpose instance types	<ul style="list-style-type: none"><li>• Web or application servers</li><li>• Enterprise applications</li><li>• Gaming servers</li><li>• Development or test environments</li></ul>	M7, Mac, M6, M5, M4, T4, T3, T2
Compute optimized instance types	<ul style="list-style-type: none"><li>• Batch processing</li><li>• Distributed analytics</li><li>• High performance computing (HPC)</li></ul>	C7, C6, C5, C4
Storage optimized instance types	<ul style="list-style-type: none"><li>• High-performance databases</li><li>• Real-time analytics</li><li>• Transactional workloads</li></ul>	I4, Im4, Is4, I3, D2, D3, H1
Memory optimized instance types	<ul style="list-style-type: none"><li>• In-memory caches</li><li>• High-performance databases</li><li>• Big data analytics</li></ul>	R7, R6, R5, R4, X2, X1, Z1
Accelerated computing instance types	<ul style="list-style-type: none"><li>• Machine learning, artificial intelligence (AI)</li><li>• HPC</li></ul>	P5, P4, P3, P2, DL1, Trn1, Inf2, Inf1, G5, G4, G3, F1, VT1
High performance computing (HPC) optimized instance types	<ul style="list-style-type: none"><li>• Deep learning workloads</li><li>• Compute-intensive HPC workloads</li></ul>	Hpc7, Hpc6

# Choosing an instance type

- With over 270 available instance types, how do you choose the right type?
  - Consider both performance requirements and cost requirements.
  - Use available resources to get recommended options.

Task	Solution
Creating a new instance	<ul style="list-style-type: none"><li>• In the EC2 console, use the Instance Types page to filter by characteristics that you choose.</li><li>• Recommendation: The latest generation in an instance family typically has a better price-to-performance ratio.</li></ul>
Optimizing an existing instance	<ul style="list-style-type: none"><li>• You can get recommendations for optimizing the instance type by using the AWS Compute Optimizer.</li><li>• You can evaluate recommendations and modify the instance accordingly.</li></ul>

# AWS Compute Optimizer



## AWS Compute Optimizer

- Recommends optimal instance type, instance size, and Auto Scaling group configuration
- Analyzes workload patterns and makes recommendations
- Classifies instance findings as Under-provisioned, Over-provisioned, Optimized, or None

AWS Compute Optimizer > Dashboard > Recommendations for EC2 instances

**Recommendations for EC2 instances (8)** [Info](#)

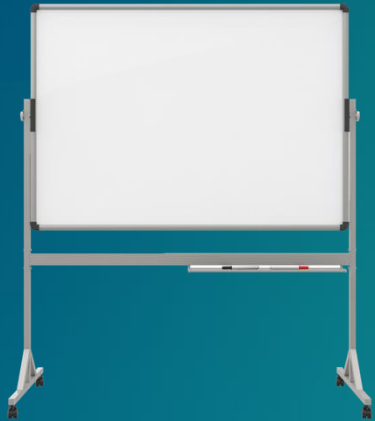
Recommendations for modifying current resources for better cost and performance.

Filter by one or more Regions: 090765505187 Over-provisioned < 1 > ⚙️

Region: US East (N. Virginia) X Clear filters

	Instance ID ▲	Instance name ▼	Finding ▼	Current instance type ▼	Current On-Demand price ▼	Recommended instance type ▼
<input type="radio"/>	i-0218a45abd8b53658	-	Over-provisioned	m5.xlarge	\$0.192 per hour	r5.large
<input type="radio"/>	i-069f6e837890db127	-	Over-provisioned	c5.xlarge	\$0.17 per hour	t3.large
<input type="radio"/>	i-07084b94d1bcf391b	-	Over-provisioned	c5.xlarge	\$0.17 per hour	t3.large
<input type="radio"/>	i-0af9322ff627d7e8f	-	Over-provisioned	m5.xlarge	\$0.192 per hour	r5.large
<input type="radio"/>	i-0ceb95ed248026d24	-	Over-provisioned	m5.xlarge	\$0.192 per hour	r5.large
<input type="radio"/>	i-0f277818dfef522e9	-	Over-provisioned	c5.xlarge	\$0.17 per hour	t3.large
<input type="radio"/>	i-0f4f4c06ad8afe81a	-	Over-provisioned	m5.2xlarge	\$0.384 per hour	r5.xlarge
<input type="radio"/>	i-0fb9323080785de1e	-	Over-provisioned	c5.xlarge	\$0.17 per hour	t3.large

# Activity: Choosing Instance Types



- Match example use cases to the EC2 instance type suited to each.
- Use the information at <https://aws.amazon.com/ec2/instance-types/>

# Activity: Choosing instance types

---

Workload
Transactional databases
Small development environments
Gaming servers
In-memory caches
Image and video generation
Machine learning
Batch processing

Instance families

C	M	I	P
Inf2	R	T	

# Activity: Choosing instance types (Answers)

---

Workload	Instance type family
Transactional databases	I
Small development environments	T
Gaming servers	M
In-memory caches	R
Image and video generation	Inf2
Machine learning	P
Batch processing	C



# Key takeaways: Selecting an EC2 instance type



- An EC2 instance type defines a configuration of CPU, memory, storage, and network performance characteristics.
- As a recommendation, choose new generation instance types in a family because they generally have better price-to-performance ratios.
- Use the Instance Types page in the Amazon EC2 console and AWS Compute Optimizer to find the right instance type for your workload.



# Adding storage to an Amazon EC2 instance

Adding a Compute Layer Using Amazon EC2

# Amazon EC2 storage overview

AWS EC2 storage resource	Root Volume	Data volumes for a single instance	Data volumes for data that is accessible from multiple Linux instances	Data volumes for data that is accessible from multiple Windows instances
Amazon EBS (SSD-backed only)	Yes	Yes	No	No
Instance store	Yes	Yes	No	No
Amazon Elastic File System (Amazon EFS) [Linux]	No	No	Yes	No
Amazon FSx for Windows File Server	No	No	No	Yes

An EC2 instance will *always* have a **root volume**, and can *optionally* have one or more **data volumes**.

# Instance store

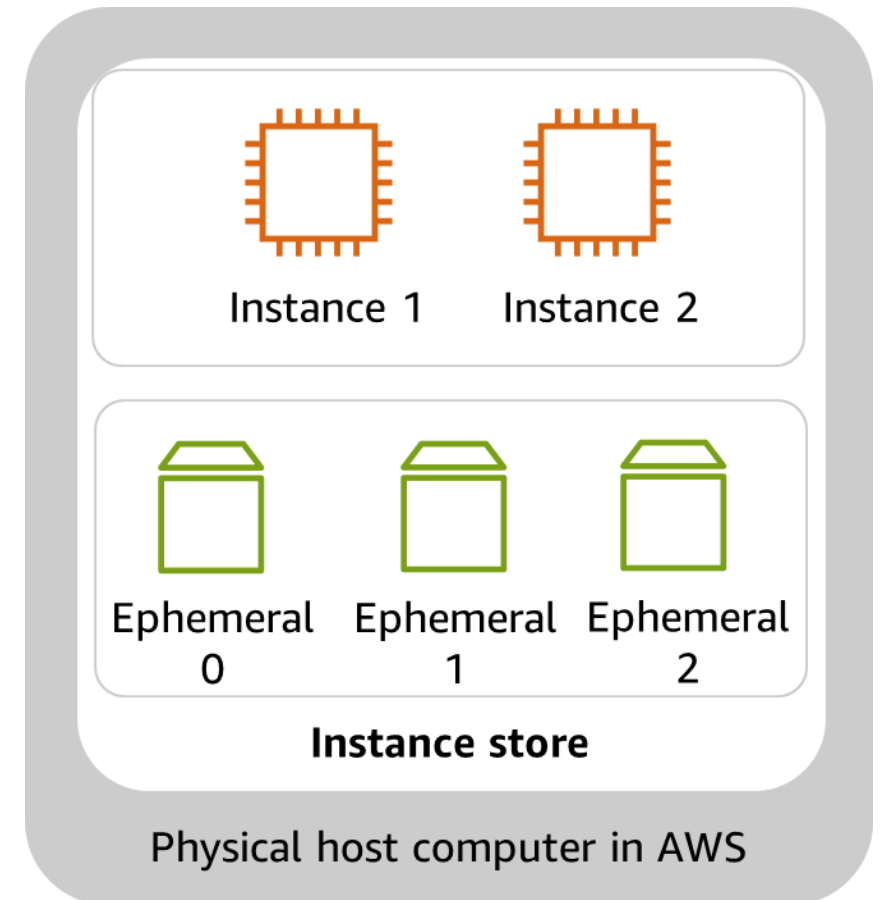
An instance store provides non-persistent storage to an instance. The data volume is stored on the same physical server where the instance runs.

## Characteristics

- Temporary block-level storage
- Uses HDD or SSD
- Instance store data is lost when the instance is stopped or terminated.

## Example use cases

- Buffers
- Cache
- Scratch data



# Amazon EBS

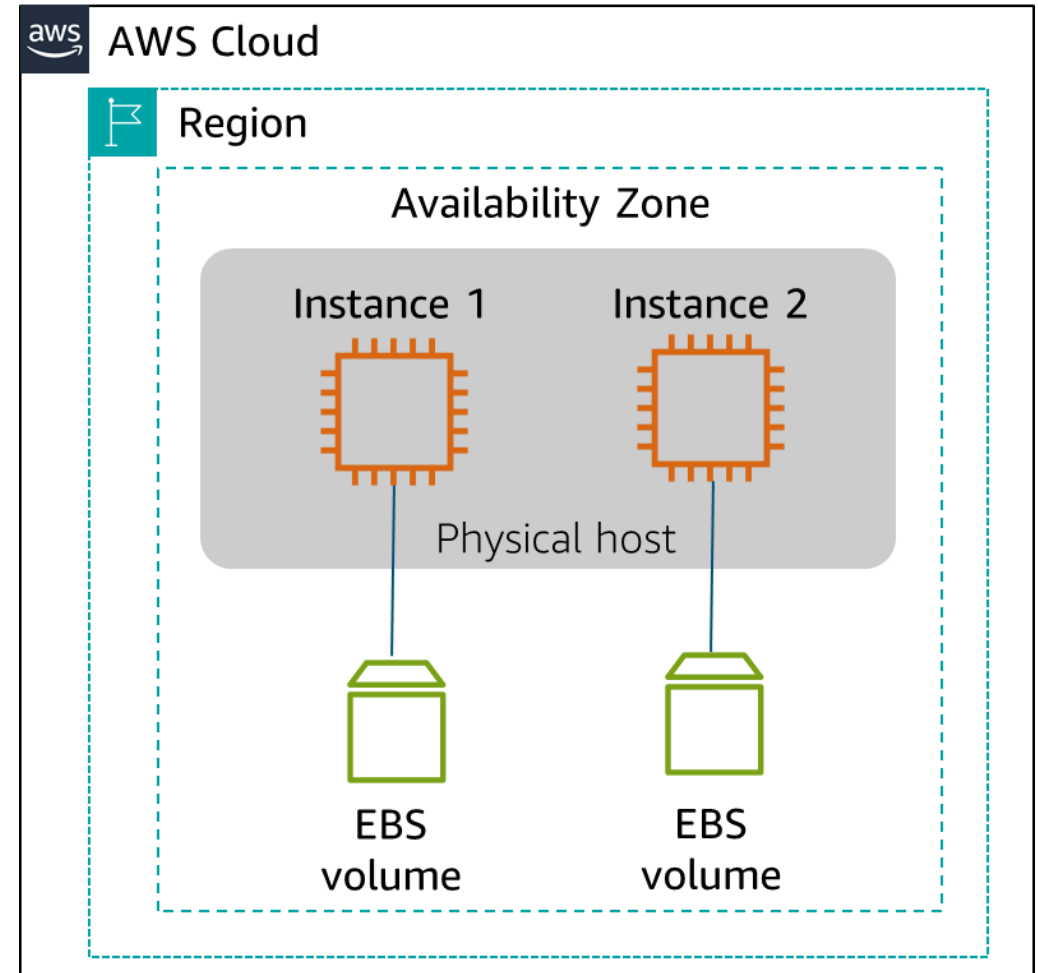
Amazon EBS volumes provide network-attached persistent storage to an EC2 instance.

## Characteristics

- Is persistent block-level storage
- Can attach to any instance in the same Availability Zone
- Uses HDD or SSD
- Can be encrypted
- Supports snapshots that are persisted to S3
- Data persists independently from the life of the instance

## Example use cases

- Stand-alone database
- General application data storage



# Amazon EBS SSD-backed volume types

Amazon EBS SSD-backed volumes are suited for use cases where the performance focus is on IOPS.

Volume type	Description	Use Cases
General Purpose SSD (gp2)	Balances price and performance for a wide variety of workloads	<ul style="list-style-type: none"><li>• Recommended for most workloads</li><li>• Can be a boot volume</li></ul>
Provisioned IOPS SSD (io1)	<ul style="list-style-type: none"><li>• Highest-performance SSD volume</li><li>• Good for mission-critical, low-latency, or high-throughput workloads</li></ul>	<ul style="list-style-type: none"><li>• Critical business applications that require sustained IOPS performance</li><li>• Large database workloads</li><li>• Transactional workloads</li><li>• Can be a boot volume</li></ul>

# Amazon EBS HDD-backed volume types

Amazon EBS HDD-backed volumes work well when the focus is on throughput.

Volume type	Description	Use Cases
Throughput Optimized HDD (st1) Description	<ul style="list-style-type: none"><li>• Low-cost volume type</li><li>• Designed for frequently accessed, throughput-intensive workloads</li></ul>	<ul style="list-style-type: none"><li>• Streaming workloads</li><li>• Big data</li><li>• Data warehouses</li><li>• Log processing</li><li>• Can't be a boot volume</li></ul>
Cold HDD (sc1)	<ul style="list-style-type: none"><li>• Lowest-cost HDD volume</li><li>• Designed for less frequently accessed workloads</li></ul>	<ul style="list-style-type: none"><li>• Throughput-oriented storage for large volumes of infrequently accessed data</li><li>• Use cases where the lowest storage cost is important</li><li>• Can't be a boot volume</li></ul>

# Amazon EBS-optimized instances

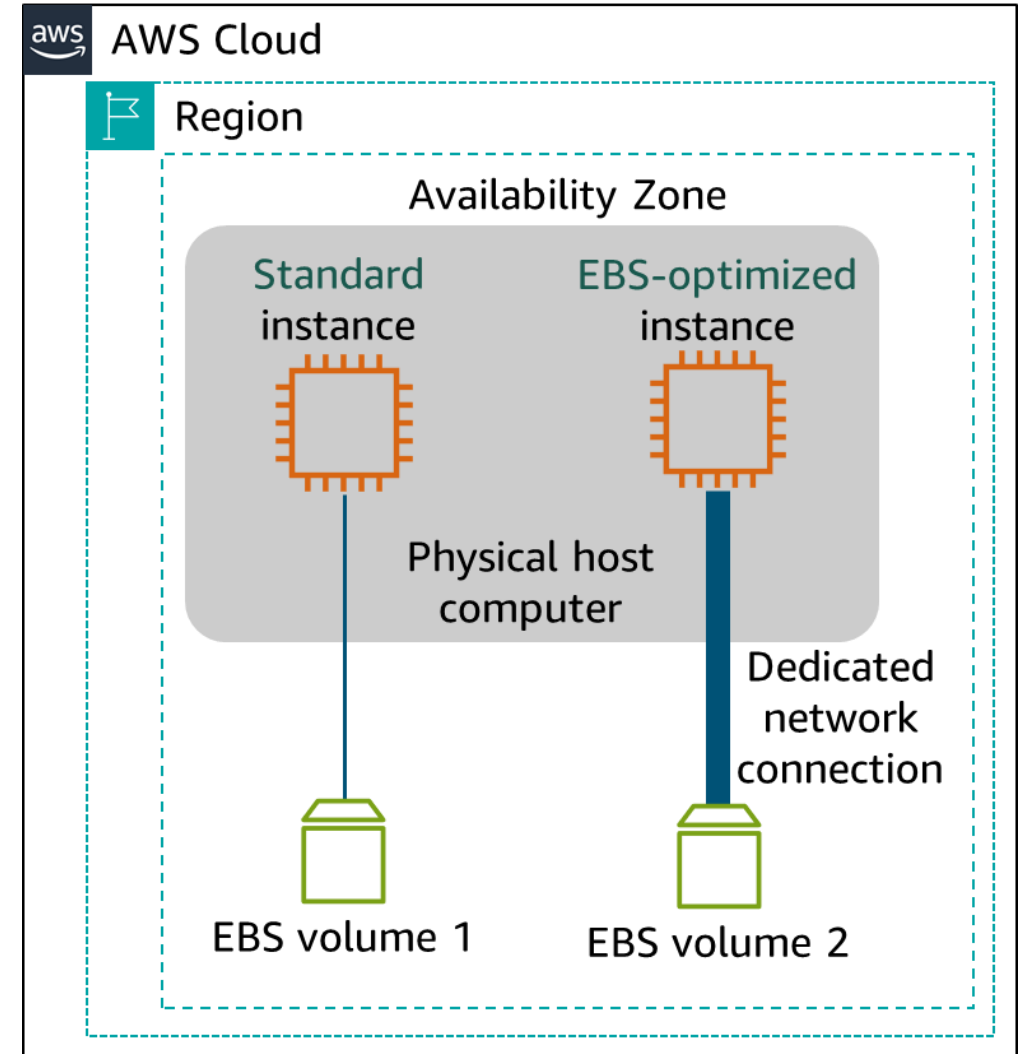
Certain EC2 instance types can be EBS-optimized

## Benefits

- It provides a dedicated network connection to attached EBS volumes.
- It increases I/O performance.
- Additional performance is achieved if using an Amazon EC2 Nitro System-based instance type.

## Usage

- For EBS-optimized instance types, optimization is enabled by default.
- For other instances types that support it, optimization must be manually enabled.





# Shared file systems for EC2 instances

What if you have multiple instances that must use the same storage?



Not an option



Amazon EBS

Attaches only to one instance



Not ideal option



Amazon S3

Amazon S3: Is an option, but is not ideal



Best option



Amazon EFS  
(Linux)



Amazon FSx for  
Windows File  
Server (Windows)

Amazon EFS *and* Amazon FSx for Windows File Server: Both satisfy the requirement

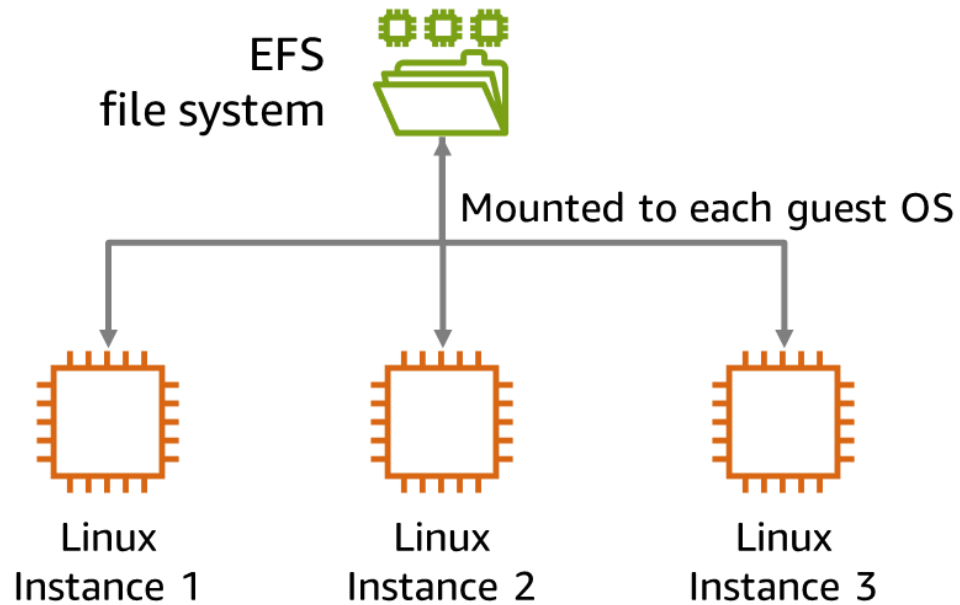
# Amazon Elastic File System (Amazon EFS)



Amazon EFS

- Provides file system storage for Linux-based workloads.
- Fully managed elastic file system
- Scales automatically up or down as files are added and removed
- Petabytes of capacity
- Supports Network File System (NFS) protocols
- Mounts the file system to the EC2 instance
- Compatible with all Linux-based AMIs for Amazon EC2

# Amazon EFS use cases



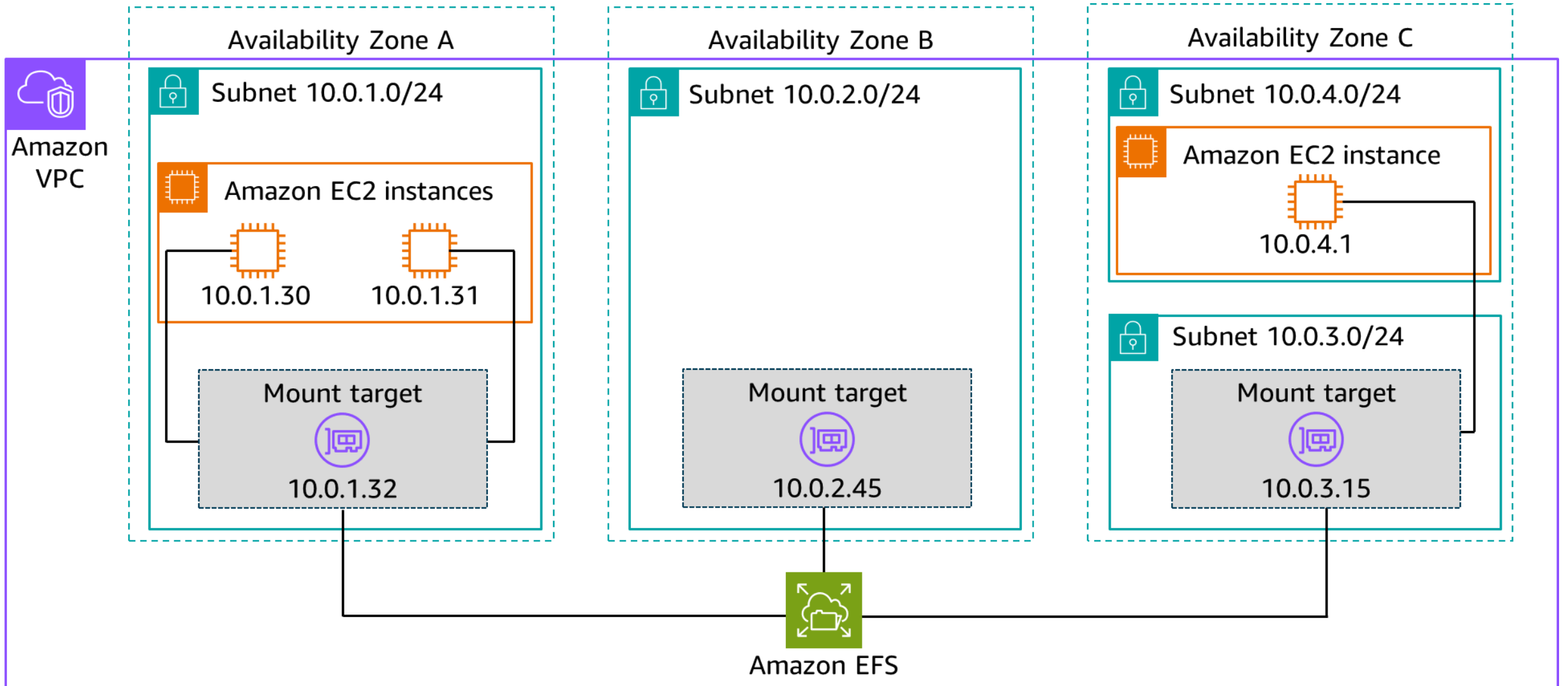
Example command to mount the file system to each guest OS:

```
$ sudo mount -t nfs4 mount-target-DNS:/ ~/efs-mount-point
```

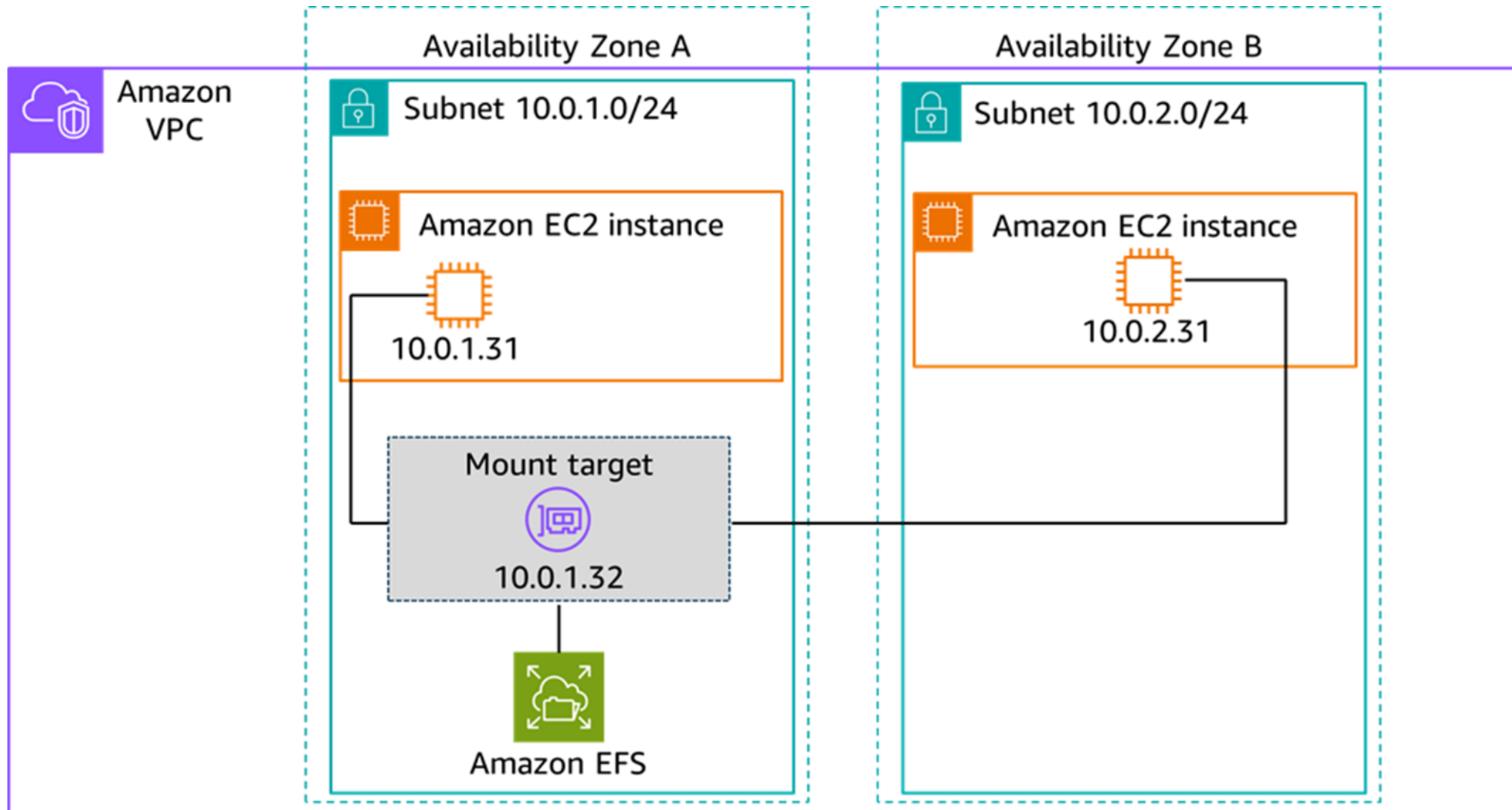
Common workloads and applications examples for Amazon EFS include the following:

- Home directories
- File system for enterprise applications
- Application testing and development
- Database backups
- Web serving and content management
- Media workflows
- Big data analytics

# Amazon EFS with Standard class storage



# Amazon EFS with One Zone class storage



# Amazon FSx for Windows File Server



Amazon FSx for  
Windows File  
Server

- Provides fully managed shared file system storage for Microsoft Windows EC2 instances.
- Native Microsoft Windows compatibility
- New Technology File System (NTFS)
- Uses Native Server Message Block (SMB) protocol version 2.0 to 3.1.1
- Distributed File System (DFS) Namespaces and DFS Replication
- Integrates with Microsoft Active Directory and supports Windows access control lists (ACLs)
- Backed by high-performance SSD storage

# Amazon FSx for Windows File Server use cases

---

- Home directories
- Lift-and-shift application workloads
- Media and entertainment workflows
- Data analytics
- Web serving and content management
- Software development environments

# Key takeaways: Adding storage to an Amazon EC2 instance



- Storage options for EC2 instances include instance store, Amazon EBS, Amazon EFS, and Amazon FSx for Windows File Server.
- For a root volume, use instance store or SSD-backed Amazon EBS.
- For a data volume that serves only one instance, use instance store or Amazon EBS storage.
- For a data volume that serves multiple Linux instances, use Amazon EFS.
- For a data volume that serves multiple Microsoft Windows instances, use Amazon FSx for Windows File Server.





# **Guided lab: Introducing Amazon EFS (Amazon EFS lab)**

# Amazon EFS lab tasks:



- In this lab, you perform the following main tasks:
  - Log in to the AWS Management Console
  - Create an Amazon EFS file system
  - Log in to an Amazon Elastic Compute Cloud (Amazon EC2) instance that runs Amazon Linux
  - Mount your file system to your EC2 instance
  - Examine and monitor the performance of your file system

# Debrief: Amazon EFS lab

---

- For instances that connect to an EFS file system, where do they need to be in relation to the EFS file system?
- Which port did you add to the inbound rules of the mount target's security group?

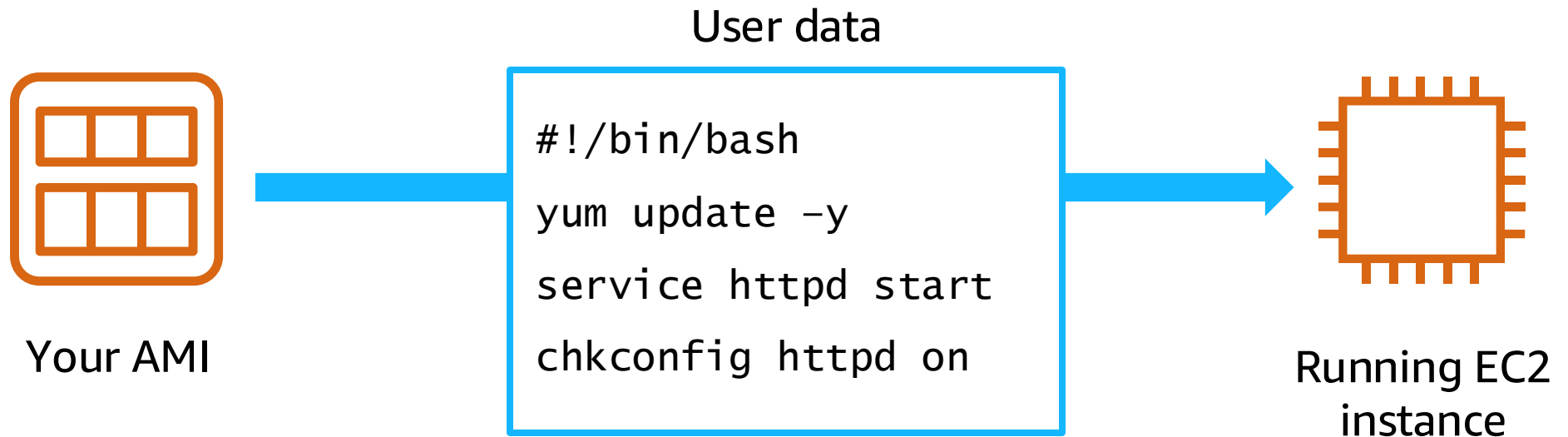


# Other EC2 configuration considerations

Adding a Compute Layer Using Amazon EC2

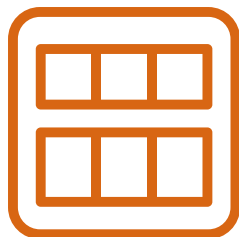
# EC2 instance user data

When you launch an EC2 instance, you can specify user data to run an initialization script (shell script or cloud-init directive).



# Retrieving instance metadata

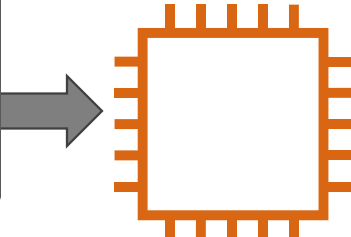
- Instance metadata is information about your instance.
- It's accessible from your instance at this URL: `http://169.254.169.254/latest/meta-data/`
- It can be retrieved from a user data script.



Your AMI

```
#!/bin/bash
yum update -y
hostname = $(curl -s http://169.254.169.254/latest/meta-data/public-hostname)
```

User data



Running EC2 instance

Metadata	Value
instance-id	i-1234567890abcdef0
mac	00-1B-63-84-45-E6
public-hostname	ec2-203-0-113-25.compute-1.amazonaws.com
public-ipv4	67.202.51.223
local-ipv4	10.251.50.12

# Working with user data on running instances

## Step 1

Stop the instance.



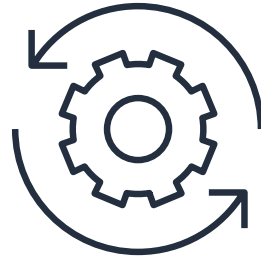
### Current user data:

Example user data script:

```
#!/bin/bash
yum update -y
```

## Step 2

Modify the user data script.



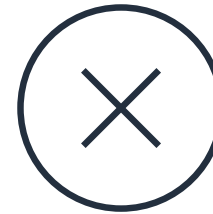
### Modified user data:

Example user data script:

```
#!/bin/bash
yum update -y
yum install httpd start
service httpd start
chkconfig httpd on
```

## Step 3

Remove the *config\_scripts\_user* file.



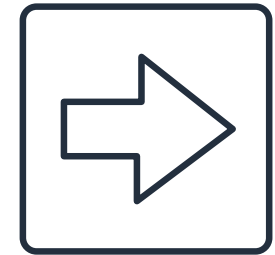
### Command to remove *config\_scripts\_user* file:

Example code to remove *config\_scripts\_user* file:

```
sudo rm /var/lib/cloud/Instances/*
sem/config_scripts_user
```

## Step 4

Re-run the user data script by restarting the instance or running a command.

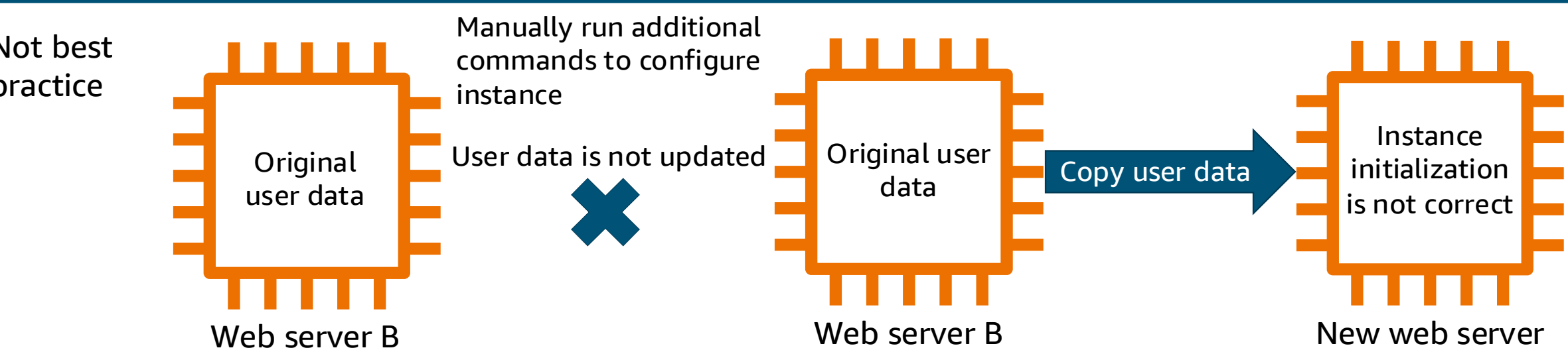
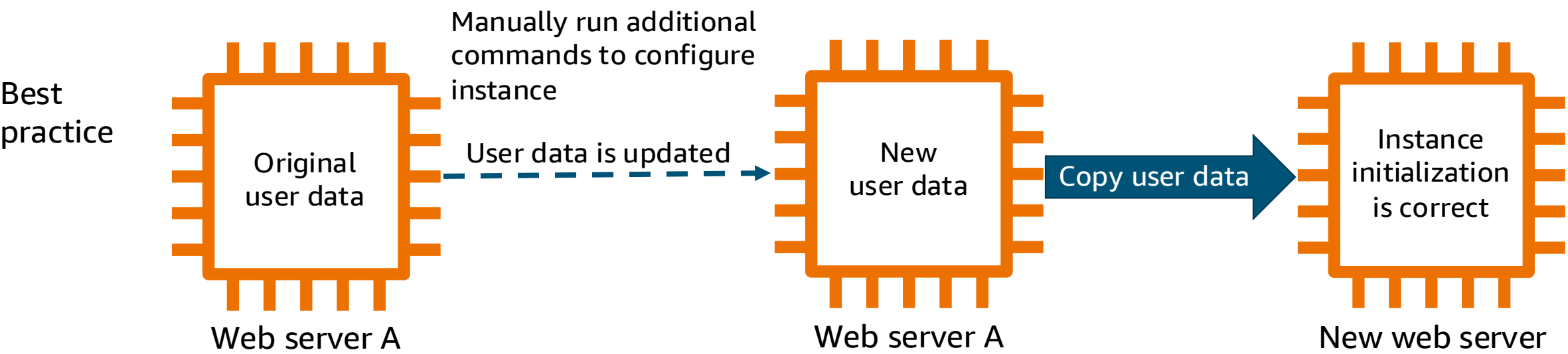


### Modified script runs:

Example command to run user data without restarting the instance:

```
/var/lib/cloud/instance/scripts/part-001
```

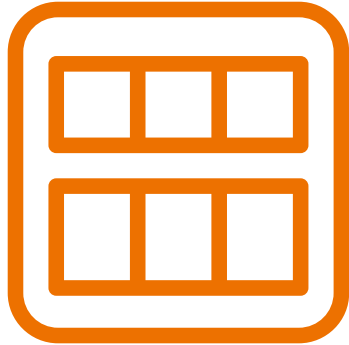
# Manually running commands best practices





# AMI deployment models

---



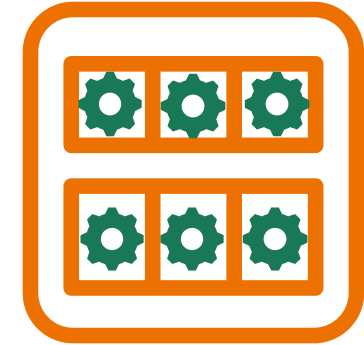
**Basic AMI**

- Base AMIs
- AMIs configured with OS-only
- Fully configurable and upgradeable
- Shorter build time
- Slower boot time



**Silver AMI**

- AWS Managed Services (AMS) provided mutable AMIs
- Configurations half baked into the AMI
- Some configurations need to be done manually or by user data scripts
- Provides a balance between boot speed and build time



**Golden AMI**

- Customized immutable AMIs
- Configurations fully baked into the AMI
- All instances using the same golden AMI behave the same
- Shorter boot times but increases build times
- Shorter lifespan of the AMI

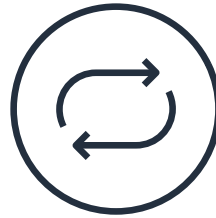
# Placement groups

Placement groups give you control of where a group of interdependent instances run in an Availability Zone.



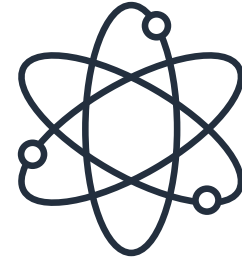
## Placement benefits

- Increase network performance between instances.
- Reduce correlated or simultaneous failure.



## Placement limitations

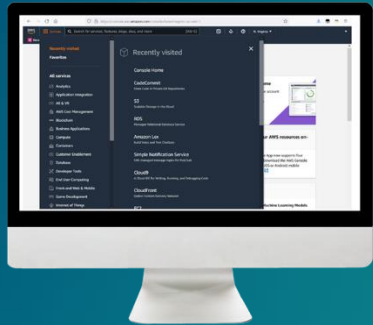
- An instance can be launched in only one placement group at a time.
- Instances with a tenancy of host can't be launched in a placement group.



## Placement strategies

- Cluster
- Partition
- Spread

# Demo: Configuring an EC2 Instance with User Data



- This demonstration uses Amazon EC2.
- In this demonstration, you will see how to do the following:
  - Create an EC2 instance
  - Configure user data
  - Test the configuration

## Key takeaways: Other EC2 configuration considerations



- User data lets you configure an EC2 instance when you launch it.
- Information about a running instance can be accessed in the instance through an instance metadata URL.
- There are three main AMI deployment models: Basic AMIs, Silver AMIs and Golden AMIs
- Placement groups give you control of where a group of interdependent instances run in an Availability Zone.

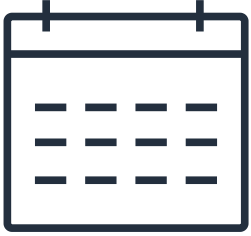


# Amazon EC2 pricing options

Adding a Compute Layer Using Amazon EC2

# AWS Free Tier: Amazon EC2

---

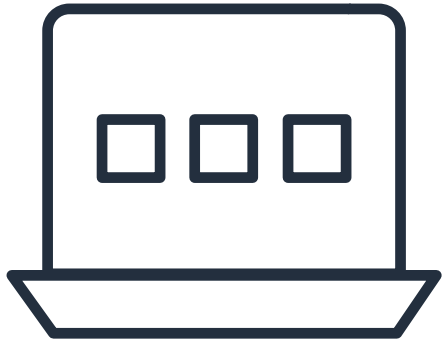


## 12 months free

- 750 hours per month of t4g.small instance dependent on region
- 750 hours per month of Linux, RHEL, or SLES t2.micro or t3.micro instance dependent on region
- 750 hours per month of Windows t2.micro or t3.micro instance dependent on region

# Amazon EC2 pricing models

Amazon EC2 provides the following purchasing strategies to help you optimize your costs based on your needs:



## Purchase models

Emphasis is on providing big saving through different use cases



## Capacity reserved models

Emphasis is on providing reserved instances to guarantee that you have them when you need them



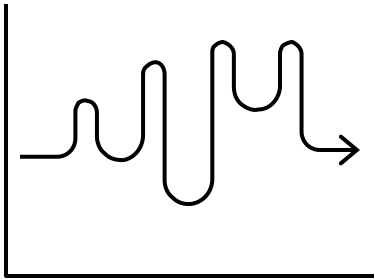
## Dedicated models

Emphasis is on providing dedicated hardware that will help you meet compliance and regulation requirements

# Amazon EC2 purchase models

## On-Demand

Pay for compute capacity by the second or by the hour with no long-term commitments.

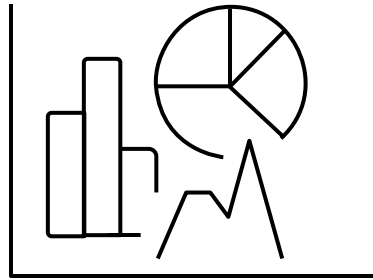


Recommended use cases:

- Spiky workloads
- Experimentation workloads

## Reserved

Make a 1-year or 3-year commitment and receive a significant discount off on-demand prices.

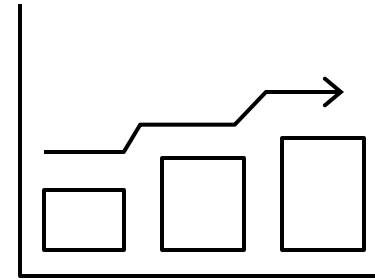


Recommended use cases:

- Committed workloads
- Steady-state workloads

## Savings Plans

Same discounts as Reserved Instances with more flexibility in exchange for a \$/hour commitment.

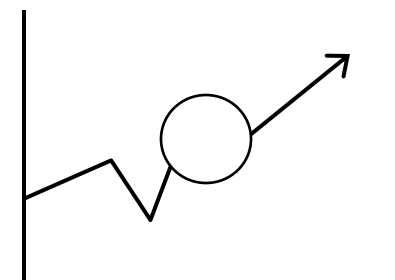


Recommended use cases:

- All Amazon EC2 workloads
- Amazon EC2 workloads that might need flexibility with committed usage

## Amazon EC2 Spot

Spare Amazon EC2 capacity at a substantial savings off the On-Demand Instance prices.

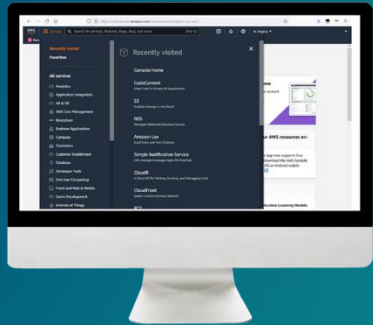


Recommended use cases:

- Fault-tolerant workloads
- Flexible workloads
- Stateless workloads



# Demo: Reviewing the Spot Instance History Page



- This demonstration uses Amazon EC2.
- In this demonstration, you will see how to do the following:
  - Review Pricing History
  - Customize Pricing History view

# Amazon EC2 Capacity Reservations

Capacity Reservations let you reserve compute capacity for Amazon EC2 instances in a specific Availability Zone.

## On-Demand Capacity Reservations

This guarantees that you always have access to EC2 capacity when you need it, for as long as you need it.

Recommended use cases:

- Workloads that need to meet regulatory requirements for high availability
- Workloads that require capacity assurance

## Amazon EC2 Capacity Blocks for ML

Reserve GPU instances for a future date to run any of your machine learning (ML) workloads.

Recommended use cases:

- Training and fine-tuning ML models
- Running experiments and building prototypes
- Planning for future surges in demand for ML applications

# Amazon EC2 dedicated options

Amazon EC2 dedicated options provide EC2 instance capacity on physical servers that are dedicated for your use (single-tenant hardware).

## Dedicated Instances

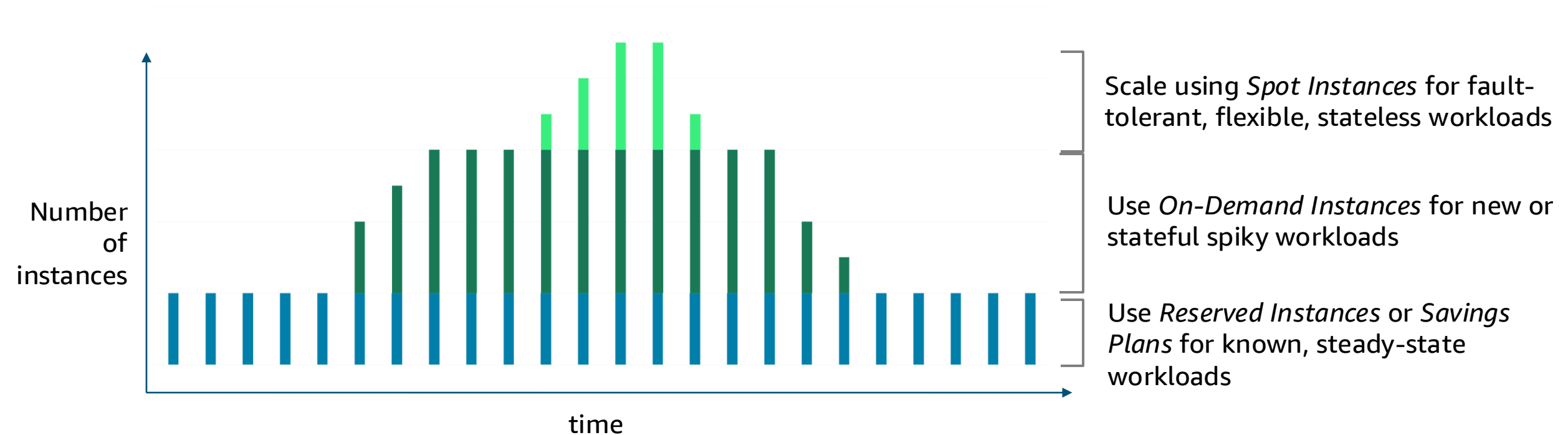
- Per-instance billing
- Automatic instance placement
- Isolates the hosts that run your instances

## Dedicated Hosts

- Per-host billing
- Visibility of sockets, cores, and host ID
- Affinity between a host and an instance
- Targeted instance placement
- Add capacity by using an allocation request
- Lets you to use your server-bound software licenses and address compliance requirements

# Amazon EC2 cost optimization guideline

To optimize the cost of Amazon EC2 instances, combine the available purchase options.



# Key takeaways: Amazon EC2 pricing options



- Amazon EC2 pricing models include On-Demand Instances, Reserved Instances, Savings Plans, Spot Instances, and Dedicated Hosts
- Per-second billing is available only for On-Demand Instances, Reserved Instances, and Spot Instances that run Amazon Linux or Ubuntu
- Use a combination of Reserved Instances, Savings Plans, On-Demand Instances, and Spot Instances to optimize Amazon EC2 compute costs



# **Café lab: Creating a Dynamic Website for the Café (Dynamic website lab)**

# The evolving café architecture: version 2

Architecture Version	Business reason for update	Technical requirements/ architecture update
V1	Static website for small business	Host the website on Amazon S3.
V2	Add online ordering	Deploy a web application and database on Amazon EC2.
V3	Reduce effort to maintain the database and secure its data	Separate web and database layers. Migrate database to Amazon RDS on a private subnet.
V4	Enhance the security of the web application	Use Amazon VPC features to configure and secure public and private subnets.
V5	Create separate access mechanisms based on role	Add IAM groups and attach resource policies to application resources. Add IAM users to groups based on role.
V6	Ensure the website can handle an expected increase in traffic	Add a load balancer, implement auto scaling on the EC2 instances and distribute compute and database instances across 2 availability zones.
V7	Module 11 info	
V8	Module 14 info	



# Dynamic website lab tasks



- In this lab, you will do the following:
  - Connect an IDE to an existing EC2 instance
  - Analyze the EC2 instance environment and confirm web server accessibility
  - Install a web application on an EC2 instance that also uses AWS Systems Manager Parameter Store
  - Test the web application
  - Create an AMI
  - Deploy a second copy of the web application to another AWS Region



# Debrief: Dynamic website lab

---

- In this lab, where did you find detailed information about your EC2 instance that is already launched and hosting your café website?
- In this lab, the café website is already running well on an existing EC2 instance. What did you do to duplicate the instance to the Oregon Region?



# **Applying the AWS Well-Architected Framework principles to compute**

## **Adding a Compute Layer Using Amazon EC2**

# AWS Well-Architected pillars



Security



Performance  
Efficiency



Cost  
Optimization



Sustainability

# Best practice approach: Infrastructure protection – Protecting compute



Security

## Best practice

Automate compute protection.

# Best practice approach: Infrastructure protection – Protecting networks



Security

## Best practice

Control traffic at all layers.

# Best practice approach: Compute and hardware



Performance  
Efficiency

## Best practices

Scale the best compute options for your workload.

Configure and right-size compute resources.

# Best practice approach: Cost effective resourcing



Cost  
Optimization

## Best practices

Select the correct resource type, size, and number.

Select the best pricing model.

# Best practice approach: Hardware and services



Sustainability

## Best practices

Use the minimum amount of hardware to meet your needs.

Use instance types with the least impact.

Use managed services.



# Key takeaways: Applying AWS Well-Architected Framework principles to compute



- Automate compute protection.
- Scale the best compute options for your workload.
- Configure and right-size compute resource.
- Select the correct resource type, size, and number.
- Select the best pricing model.
- Use the minimum amount of hardware to meet your needs.
- Use instance types with the least impact.



# Module wrap-up

Adding a Compute Layer Using Amazon EC2

# Module summary

---

This module prepared you to do the following:

- Identify how Amazon Elastic Compute Cloud (Amazon EC2) can be used in an architecture.
- Explain the value of using Amazon Machine Images (AMIs) to accelerate the creation and repeatability of infrastructure.
- Recommend EC2 instance types based on requirements.
- Recommend storage solutions for Amazon EC2.
- Recognize how to configure Amazon EC2 instances with user data.
- Describe EC2 pricing options and make recommendations based on cost.
- Launch an Amazon EC2 instance.
- Use the AWS Well-Architected Framework principles when designing a compute layer with Amazon EC2.

# Considerations for the cafe

---

Discuss how the café labs in this module answered the key questions and decisions that were presented at the start of this module for the café business.



# Module knowledge check



- The knowledge check is delivered online within your course.
- The knowledge check includes 10 questions based on material presented on the slides and in the slide notes.
- You can retake the knowledge check as many times as you like.

# Sample exam question

A solutions architect has a workload that will run for at least 1 year uninterrupted in the same Region. The workload will remain steady, except for occasional spikes during peak seasons. During these spikes, the size of the instance type might need to be increased to handle heavier workloads. However, the instance family will remain the same with any instance size increase. Which pricing option should be used to purchase the instance at the lowest cost?

Identify the key words and phrases before continuing.

The following are the key words and phrases:

- At least 1 year uninterrupted in the same Region
- Workload will remain steady
- Instance family will remain the same
- Lowest cost

# Sample exam question: Response choices

A solutions architect has a workload that will run for at least *1 year uninterrupted in the same Region*. The **workload** will remain *steady*, except for occasional spikes during peak seasons. During these spikes, the size of the instance type might need to be increased to handle heavier workloads. However, the instance family will remain the same with any instance size increase. Which pricing option should be used to purchase the instance at the *lowest cost*?

Choice	Response
A	Dedicated instance
B	Compute Saving Plans instance
C	EC2 Instance Savings Plans instance
D	On-Demand instance

# Sample exam question: Answer

The answer is C.

Choice	Response
--------	----------

C	EC2 Instance Savings Plans instance
---	-------------------------------------





# Thank you

Corrections, feedback, or other questions?

Contact us at <https://support.aws.amazon.com/#/contacts/aws-academy>.