# Compute Service

## Module objectives

At the end of this module, you should be able to:

- Identify how Amazon Elastic Compute Cloud (Amazon EC2) can be used.
- Explain the value of using Amazon Machine Images (AMIs) to accelerate the creation and repeatability of infrastructure
- Differentiate between the EC2 instance types
- Recognize storage solutions for Amazon EC2
- Describe EC2 pricing options
- Describe AWS Lambda service
- Describe Elastic Beanstalk service

# Adding compute with Amazon EC2

## 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.

# Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2)

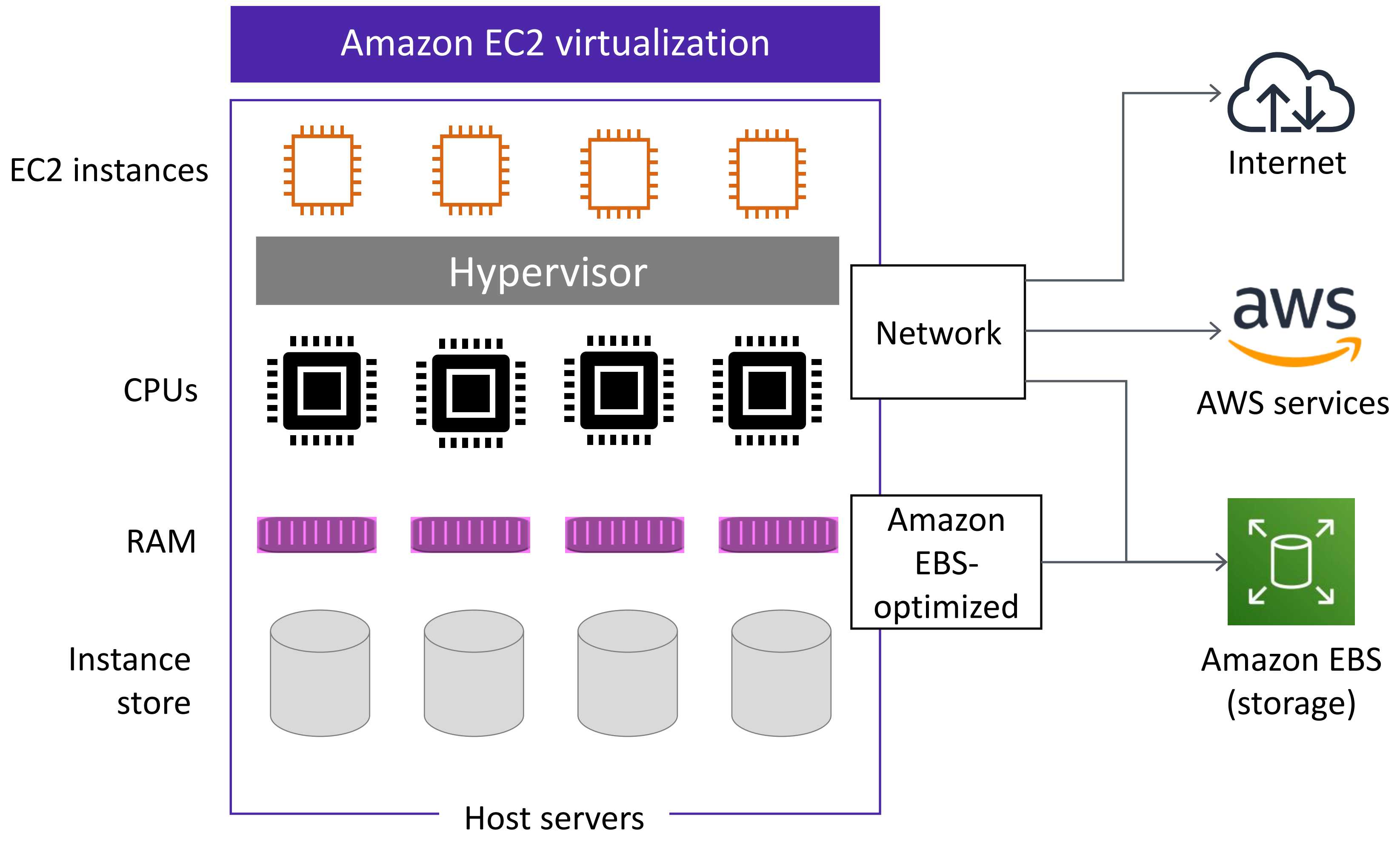
Amazon EC2 provides resizable compute capacity in the cloud.

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

# EC2 instances

An EC2 instance is a virtual machine that runs on a physical host.

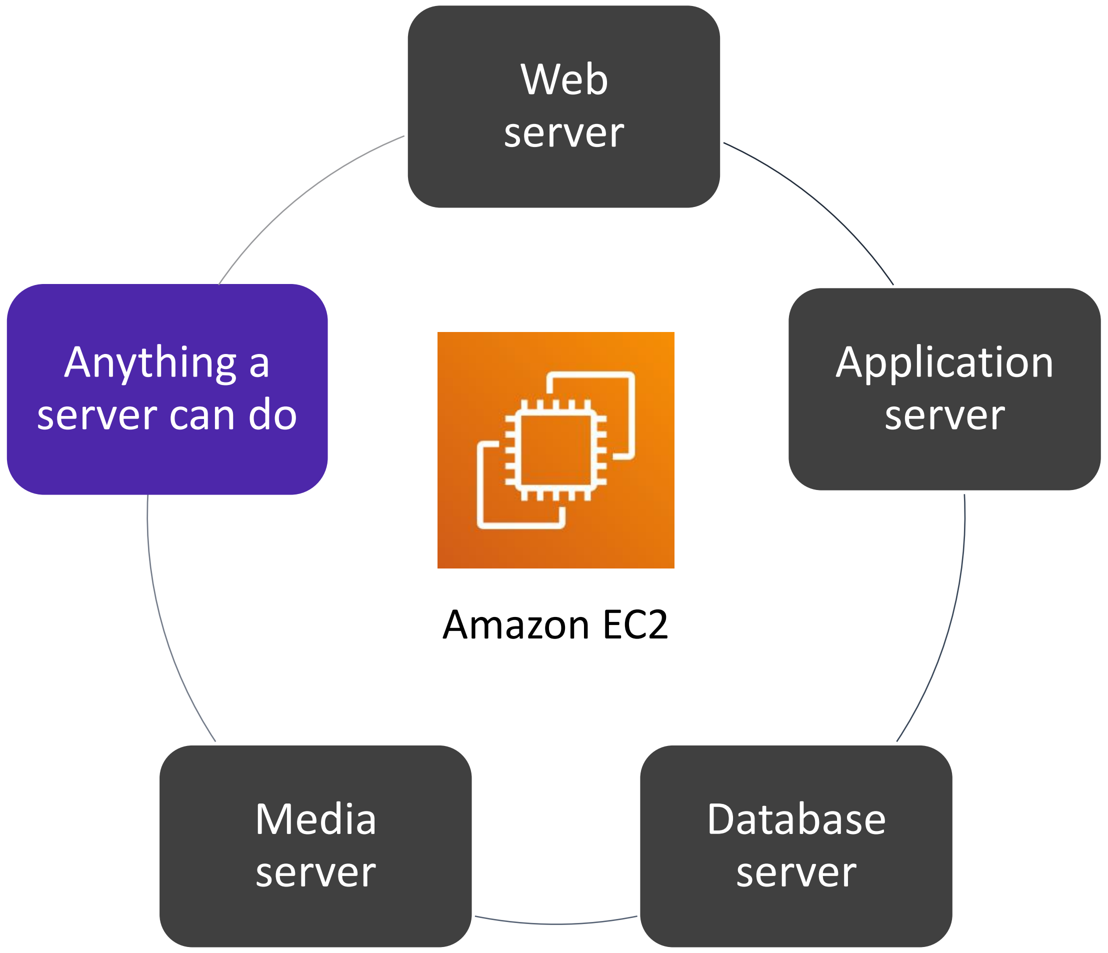
- You can choose different configurations of CPU and memory capacity
- Supports different storage options
  - Instance store
  - Amazon Elastic Block Store (Amazon EBS)
- Provides network connectivity



# Amazon EC2 use cases

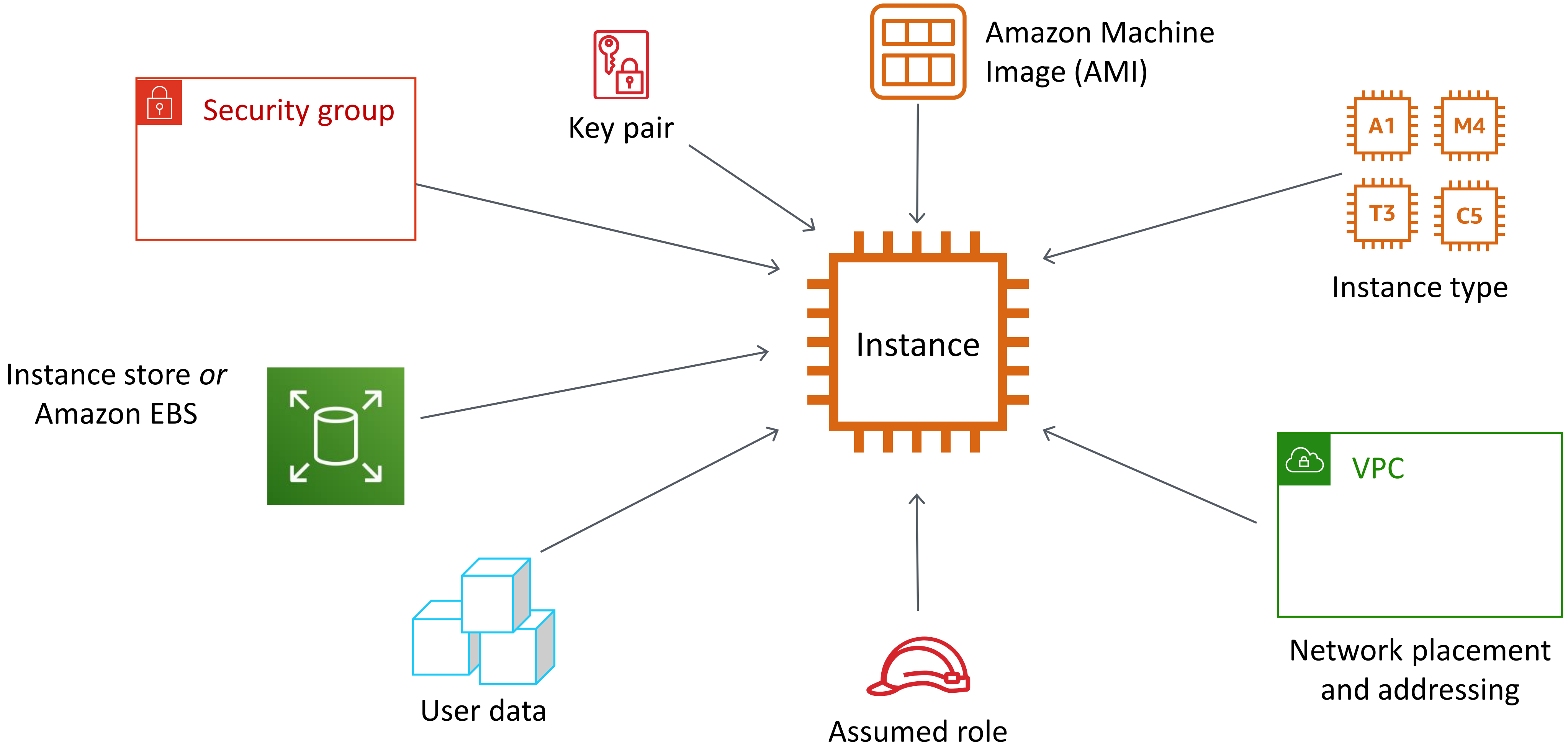
Use Amazon EC2 when you need:

- 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, for example –
  - Simple websites
  - Enterprise applications
  - High performance computing (HPC) applications



# Provisioning an EC2 instance

## Essential instance launch configuration parameters


Security group
Key pair
Amazon Machine Image (AMI)
A1
M4
T3
C5
Instance type
Instance
Instance store *or* Amazon EBS
VPC
Network placement and addressing
User data
Assumed role

## key takeaways

- Amazon EC2 enables you to run Microsoft Windows and Linux virtual machines in the cloud.
- 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.

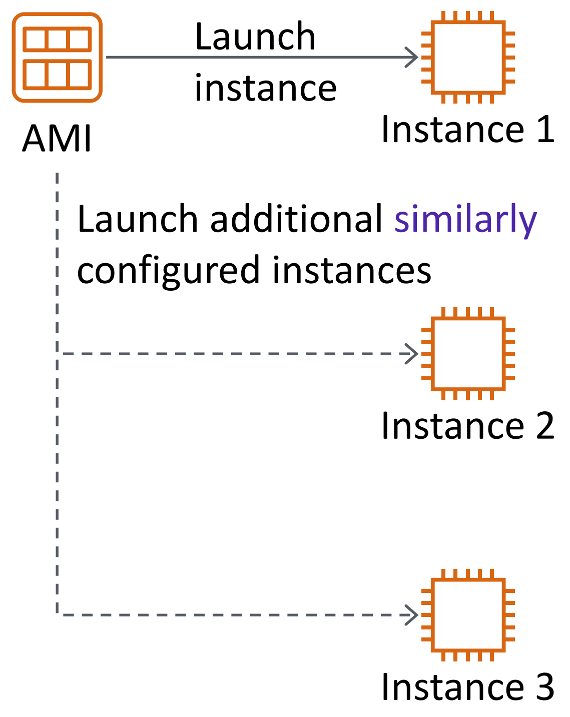
## Choosing an AMI to launch an EC2 instance

# Amazon Machine Image (AMI)

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

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

Create multiple instances from the same AMI



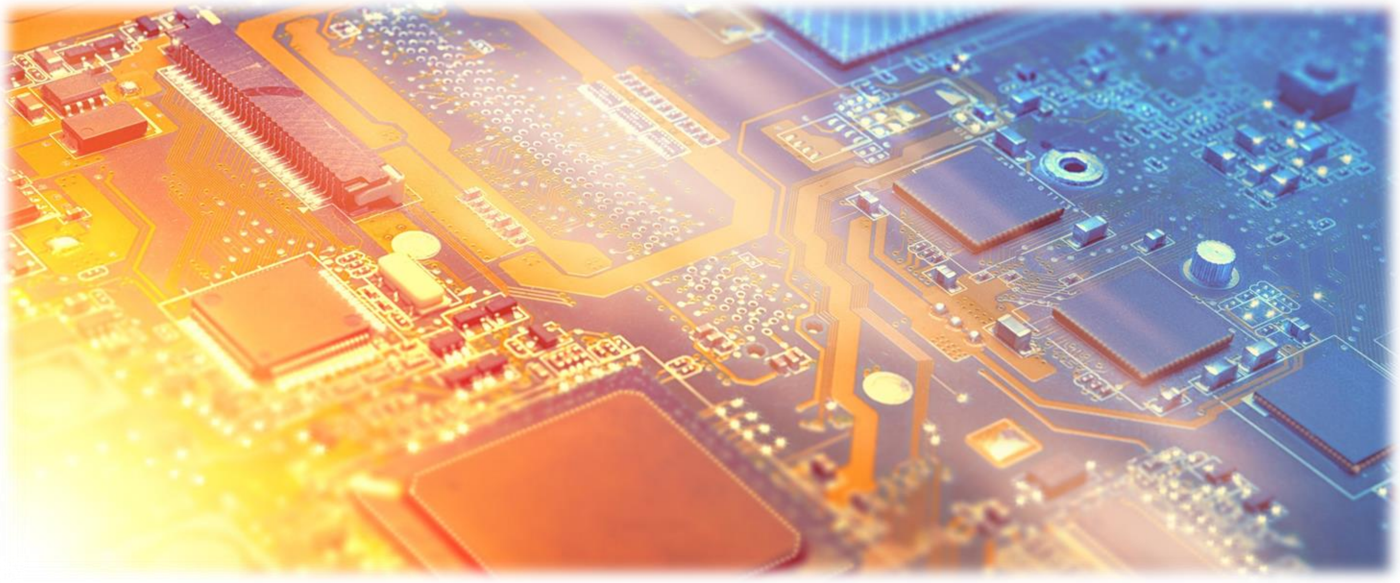
## AMI benefits

- 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:

- Region
- Operating system
  - Microsoft Windows or Linux
- Storage type of the root device
- Architecture
- Virtualization type



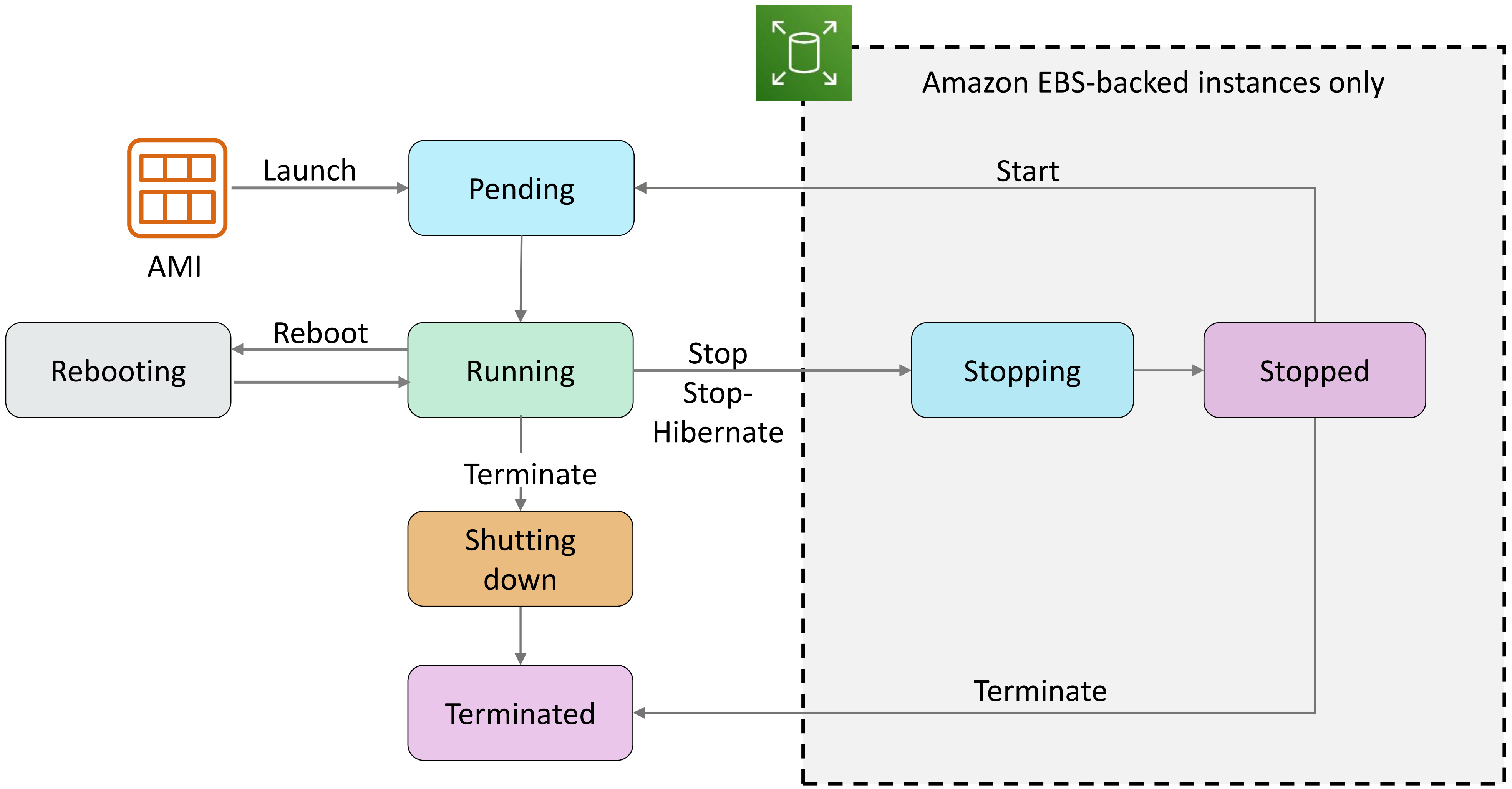
AMI sources:

- 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.*

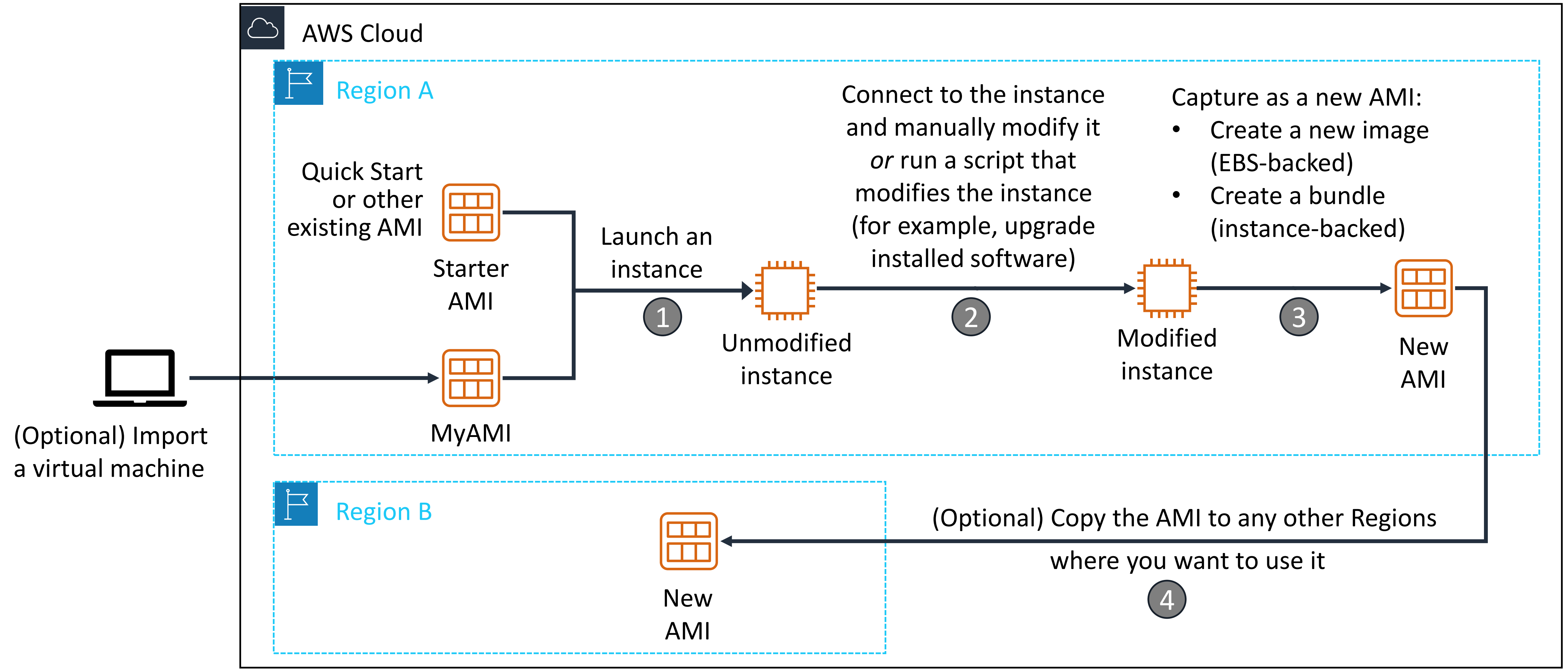
# Instance store-backed versus Amazon EBS-backed AMI

| 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 | Can't stop the instance, only reboot or terminate it |
| 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 |

# Amazon EC2 instance lifecycle



# Creating a new AMI


AWS Cloud
Region A
Quick Start or other existing AMI
Starter AMI
MyAMI
(Optional) Import a virtual machine
Launch an instance
1
Unmodified instance
Connect to the instance and manually modify it *or* run a script that modifies the instance (for example, upgrade installed software)
2
Modified instance
Capture as a new AMI:
• Create a new image (EBS-backed)
• Create a bundle (instance-backed)
3
New AMI
Region B
New AMI
(Optional) Copy the AMI to any other Regions where you want to use it
4

## EC2 Image Builder

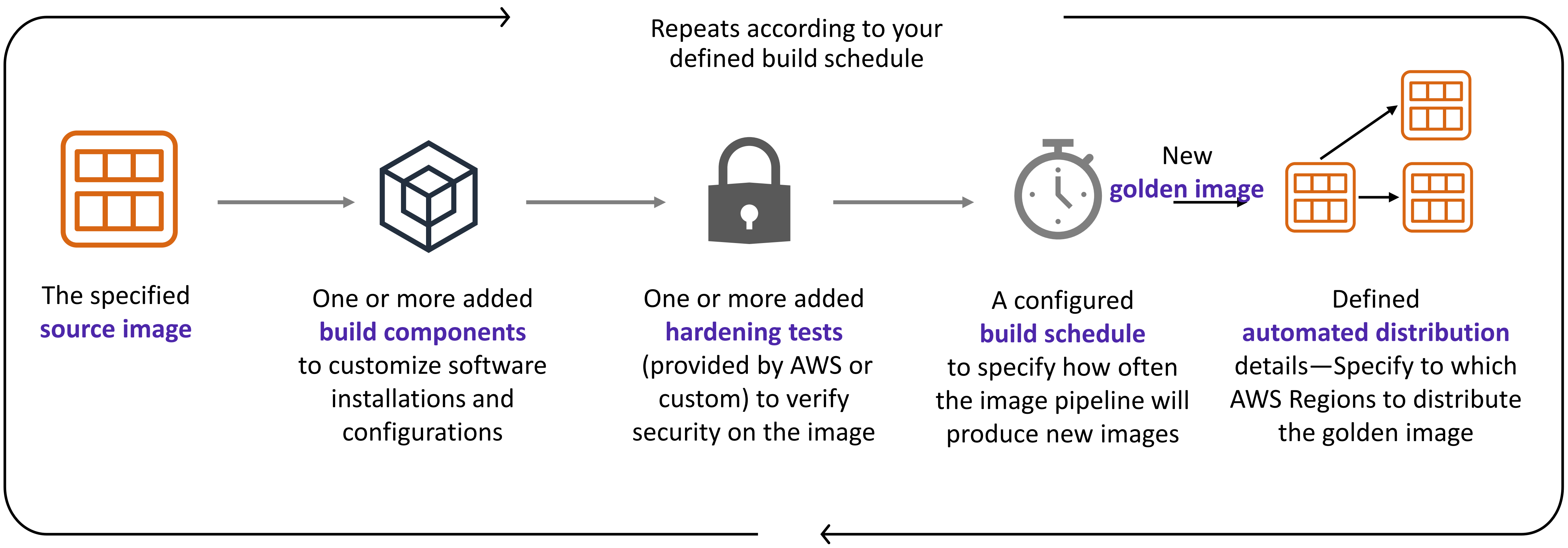
EC2 Image Builder

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

# How EC2 Image Builder works

## An EC2 Image Builder image pipeline



The specified **source image**

One or more added **build components** to customize software installations and configurations

One or more added **hardening tests** (provided by AWS or custom) to verify security on the image

A configured **build schedule** to specify how often the image pipeline will produce new images

Defined **automated distribution** details—Specify to which AWS Regions to distribute the golden image

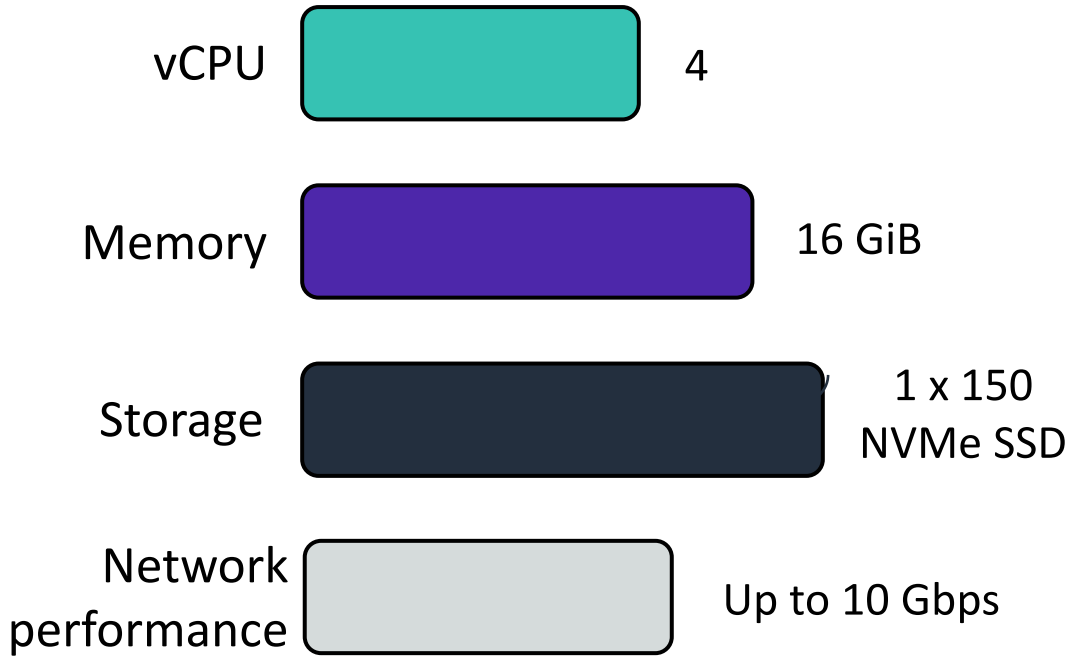
## key takeaways

- An AMI provides the information that is needed to launch an EC2 instance
- For best performance, use an AMI with HVM virtualization type
- Only an instance launched from an Amazon EBS-backed AMI can be stopped and started
- An AMI is available in a Region

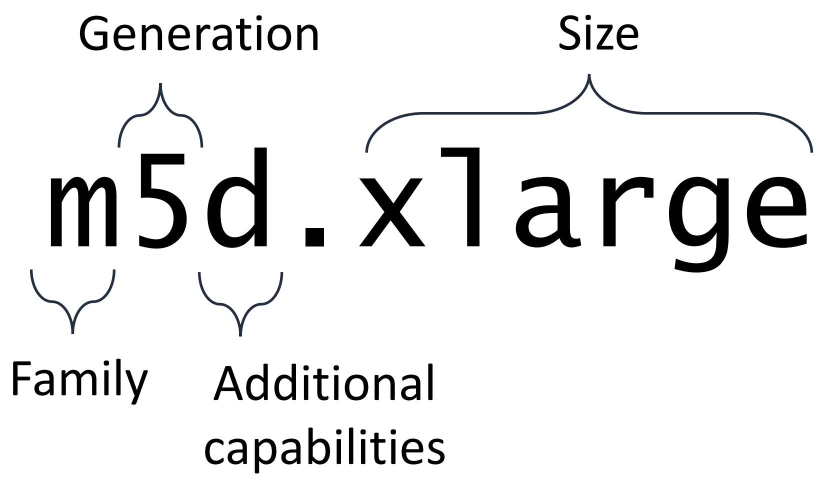
## Selecting an EC2 instance type

# EC2 instance type

An EC2 instance type defines a configuration of CPU, memory, storage, and network performance characteristics that provides a given level of compute performance.



Instance Type Naming



# Suitability of instance types for workloads (1 of 2)

## General purpose instance types

- Web or application servers
- Enterprise applications
- Gaming servers
- Analytics applications
- Development or test environments

Example instance types: M5 T3 A1

## Compute optimized instance types

- Batch processing
- Distributed analytics
- High performance computing (HPC)
- Ad server engines
- Multiplayer gaming
- Video encoding

Example instance types: C5 C5n

## Suitability of instance types for workloads (2 of 2)

### Memory optimized instance types

- In-memory caches
- High-performance databases
- Big data analytics

Example instance types: R5 X1 HMI

### Accelerated computing instance types

- Machine learning, artificial intelligence (AI)
- HPC
- Graphics

Example instance types: P3 G4 F1

### Storage optimized instance types

- High-performance databases[1]
- Real-time analytics[1]
- Transactional workloads[1]
- NoSQL databases[1]
- Big data[2]
- Data warehouse[2]
- Log processing[2]

[1]High I/O example instance type: I3

[2]Dense Storage example instance types: D2 H1

# Choosing an instance type

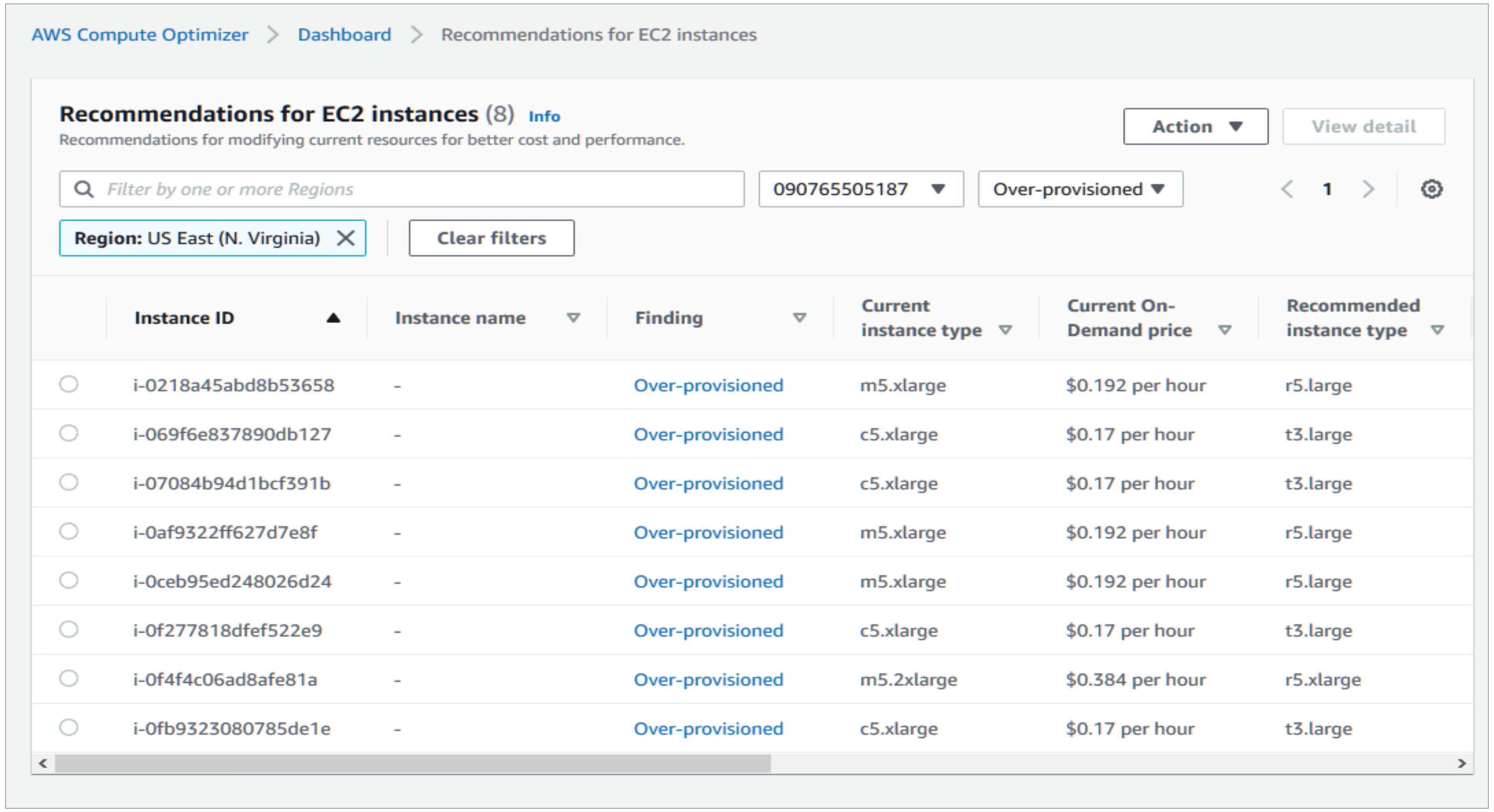
- Choose the instance type that meets –
  - The performance needs of your application
  - Your cost requirements
- When you create a *new* instance –
  - In the EC2 console, use the Instance Types page to filter by characteristics that you choose
  - Recommendation: The latest generation in an instance family typically has a better price-to-performance ratio
- If you have an *already existing* instance –
  - You can get recommendations for optimizing the instance type by using the AWS Compute Optimizer
  - You can evaluate recommendations and modify the instance accordingly

With over 270 available instances types, how do you choose the correct one?

# 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
Action
View detail
Filter by one or more Regions
Clear filters
Info

**Recommendations for EC2 instances** (8)

Recommendations for modifying current resources for better cost and performance.

090765505187 · Over-provisioned · Region: US East (N. Virginia)

| Instance ID | Instance name | Finding | Current instance type | Current On-Demand price | Recommended instance type |
|---|---|---|---|---|---|
| i-0218a45abd8b53658 | - | Over-provisioned | m5.xlarge | $0.192 per hour | r5.large |
| i-069f6e837890db127 | - | Over-provisioned | c5.xlarge | $0.17 per hour | t3.large |
| i-07084b94d1bcf391b | - | Over-provisioned | c5.xlarge | $0.17 per hour | t3.large |
| i-0af9322ff627d7e8f | - | Over-provisioned | m5.xlarge | $0.192 per hour | r5.large |
| i-0ceb95ed248026d24 | - | Over-provisioned | m5.xlarge | $0.192 per hour | r5.large |
| i-0f277818dfef522e9 | - | Over-provisioned | c5.xlarge | $0.17 per hour | t3.large |
| i-0f4f4c06ad8afe81a | - | Over-provisioned | m5.2xlarge | $0.384 per hour | r5.xlarge |
| i-0fb9323080785de1e | - | Over-provisioned | c5.xlarge | $0.17 per hour | t3.large |

## key takeaways

- 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

# Amazon EC2 storage overview

## Root volume

**This volume always contains the guest OS**

Instance store

Amazon EBS (SSD-backed only)

## Data volumes

**For data accessed by a single instance**

Instance store

Amazon EBS

**For data accessible from multiple instances**

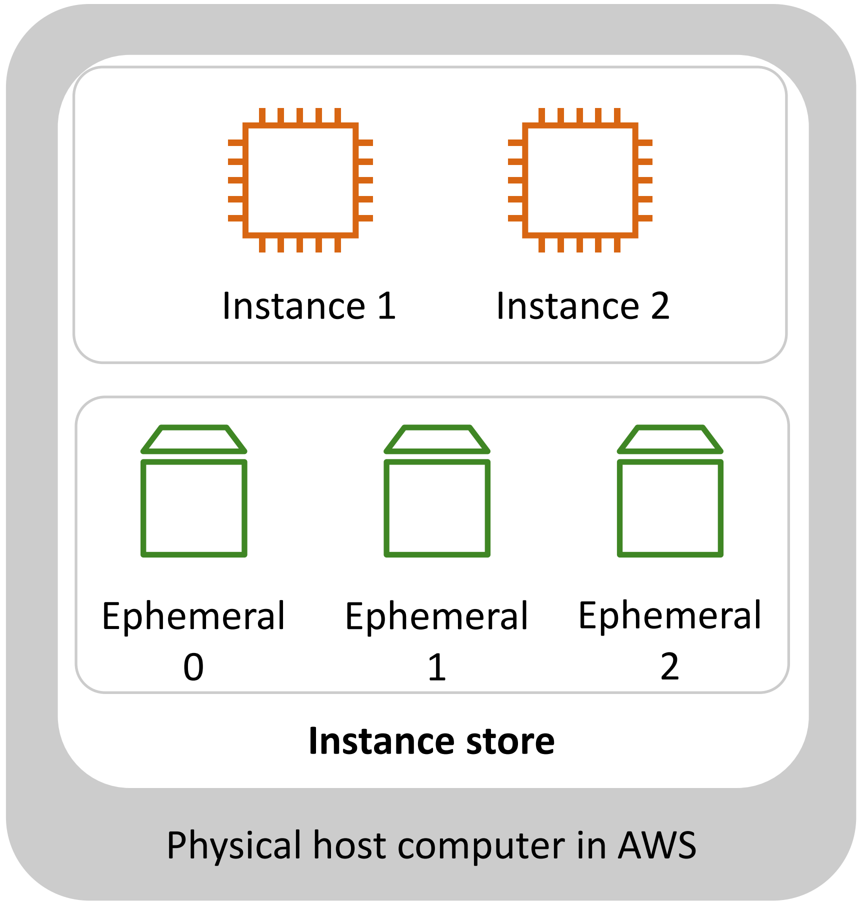
Amazon Elastic File System (Amazon EFS) [Linux]

Amazon FSx for Windows File Server

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 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

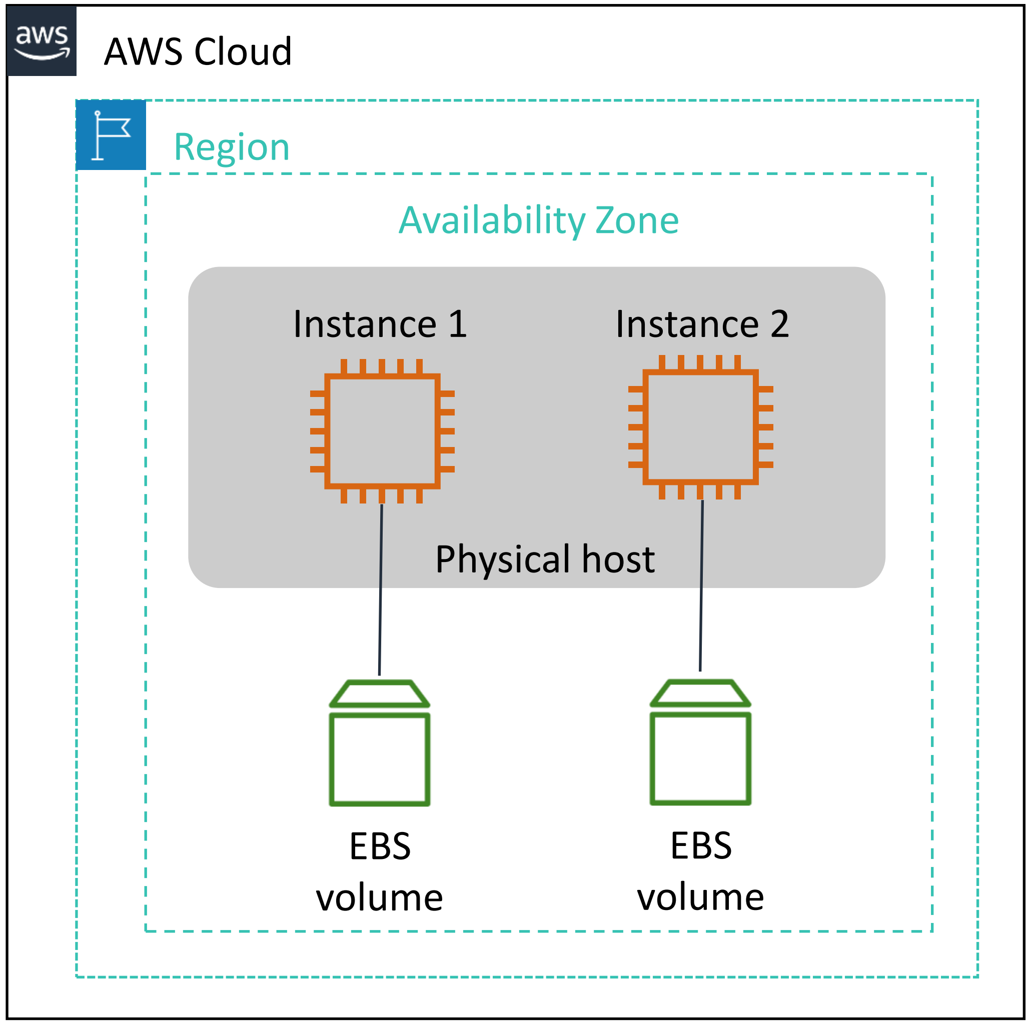


# Amazon EBS

- 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.

| | General Purpose SSD (gp2) | Provisioned IOPS SSD (io1) |
|---|---|---|
| Description | Balances price and performance for a wide variety of workloads | • Highest-performance SSD volume<br>• Good for mission-critical, low-latency, or high-throughput workloads |
| Use Cases | • Recommended for most workloads<br>• Can be a boot volume | • Critical business applications that require sustained IOPS performance<br>• Large database workloads<br>• Transactional workloads<br>• It can be a boot volume |

# Amazon EBS HDD-backed volume types

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

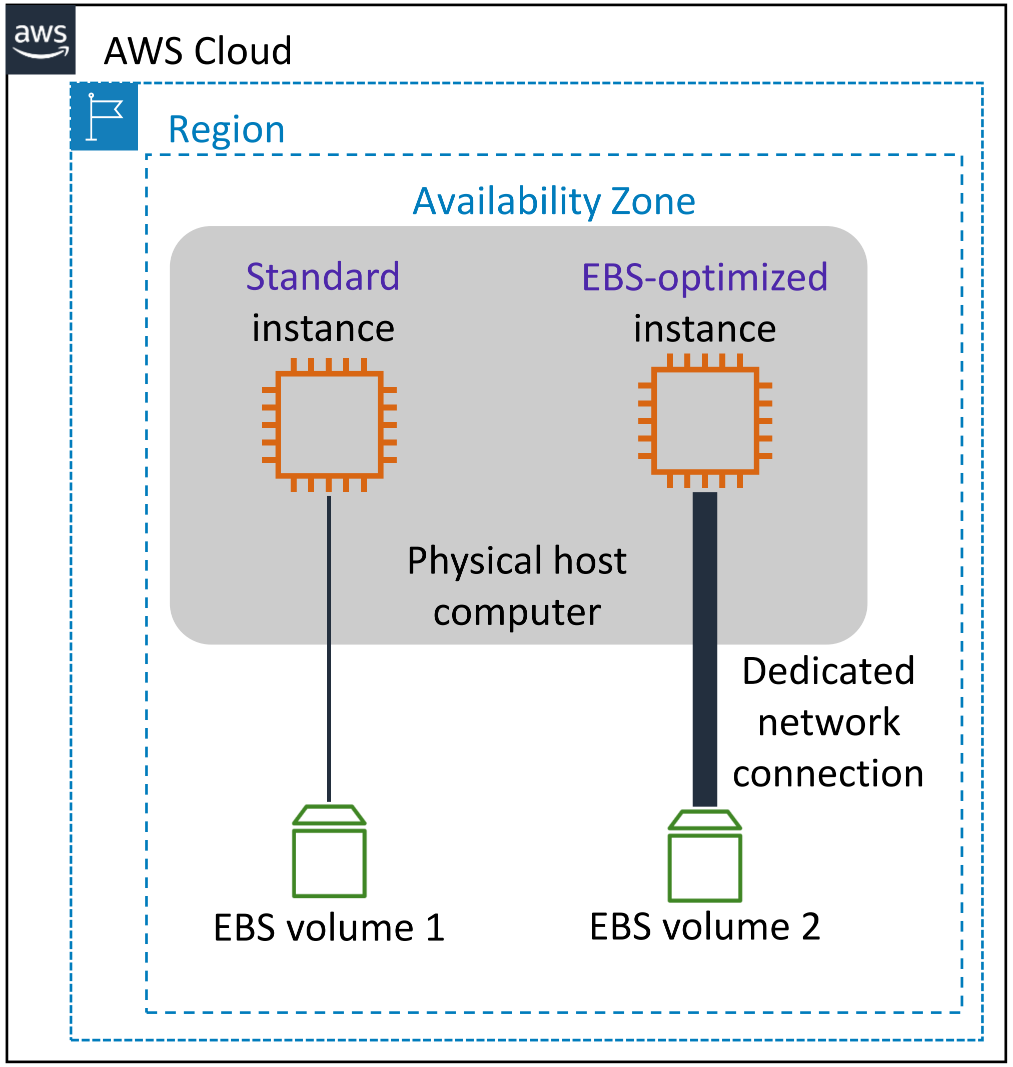
| | Throughput Optimized HDD (st1) | Cold HDD (sc1) |
|---|---|---|
| Description | • Low-cost volume type<br>• Designed for frequently accessed, throughput-intensive workloads | • Lowest-cost HDD volume<br>• Designed for less frequently accessed workloads |
| Use Cases | • Streaming workloads<br>• Big data<br>• Data warehouses<br>• Log processing<br>• It cannot be a boot volume | • Throughput-oriented storage for large volumes of infrequently accessed data<br>• Use cases where the lowest storage cost is important<br>• It cannot be a boot volume |

# Amazon EBS-optimized instances

- Certain EC2 instance types can be EBS-optimized

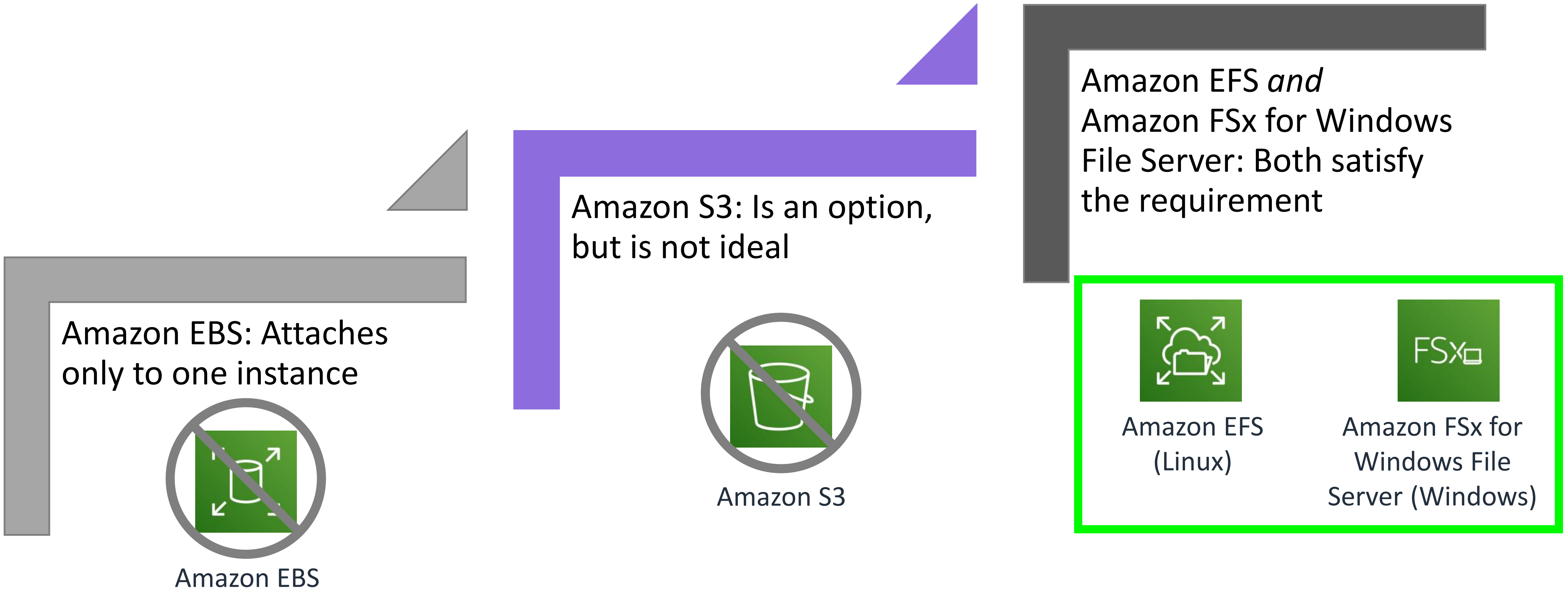
- Benefits –
  - Provides a dedicated network connection to attached EBS volumes
  - Increases I/O performance
  - Additional performance is achieved if using an Amazon EC2 Nitro System-based instance type

- 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?



Amazon EBS: Attaches only to one instance

Amazon S3: Is an option, but is not ideal

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

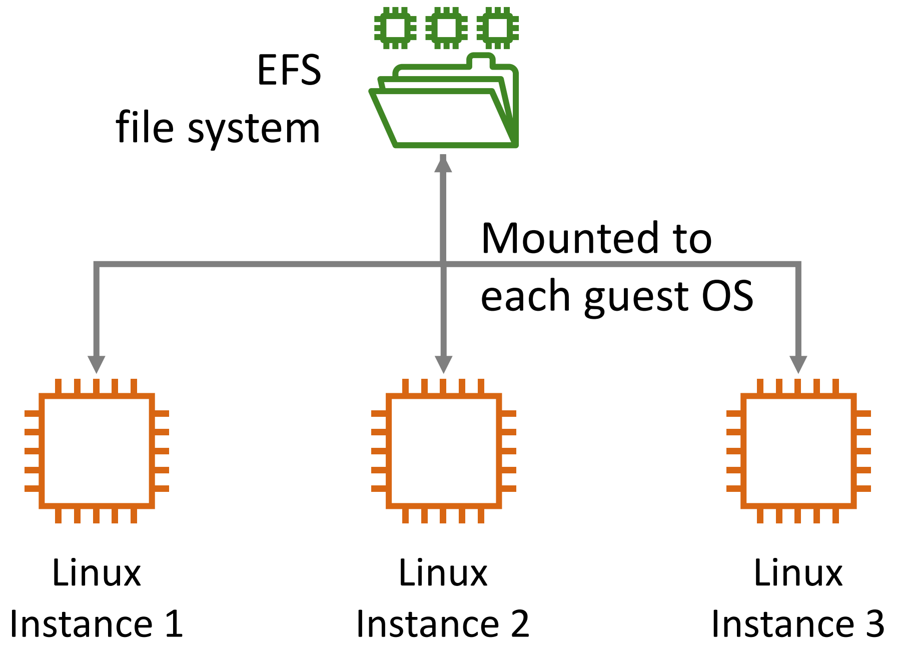
## Amazon EFS

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
  - Mount the file system to the EC2 instance
- Compatible with all Linux-based AMIs for Amazon EC2

# Amazon EFS use cases



Common workloads and applications:

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

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

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

# Amazon FSx for Windows File Server

Amazon FSx for Windows File Server

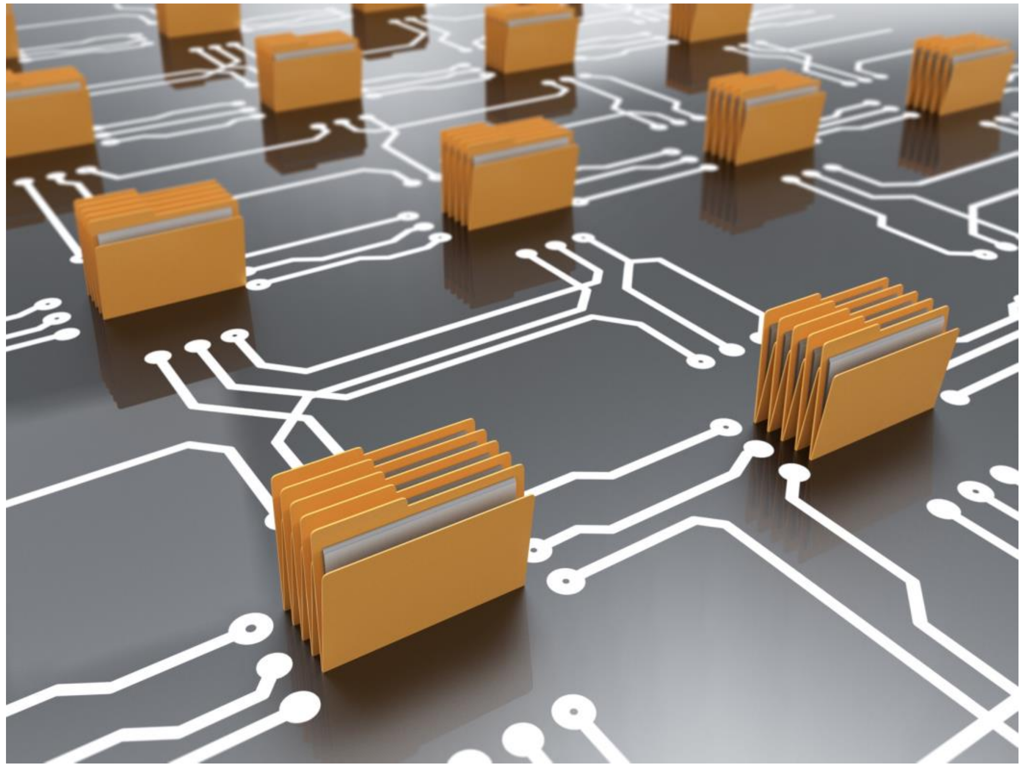
Provides fully managed shared file system storage for Microsoft Windows EC2 instances.

- Native Microsoft Windows compatibility
- New Technology File System (NTFS)
- 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

Amazon FSx for Windows File Server supports a broad set of Microsoft Windows workloads.

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



## Key takeaways

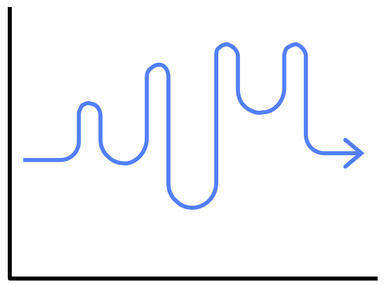
- Storage options for EC2 instances include
- Instance store, Amazon EBS, Amazon EFS, and Amazon FSx for Windows File Server
- For a root volume ---
- use instance store or SSD-backed Amazon EBS
- For a data volume that serves only one instance,
- Use instance store or Amazon EBS storage
- For a data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

## Amazon EC2 pricing options

# Amazon EC2 pricing options (1 of 2)

## On-Demand Instances

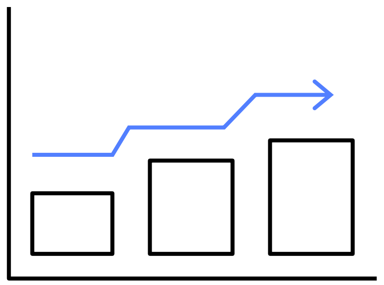
Pay for compute capacity by the second or by the hour with no long-term commitments.



Spiky workloads, workload experimentation

## Reserved Instances

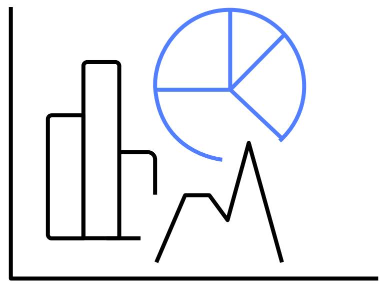
Make a 1-year or 3-year commitment and receive a significant discount off on-demand prices.



Committed and steady-state workloads

## Savings Plans

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

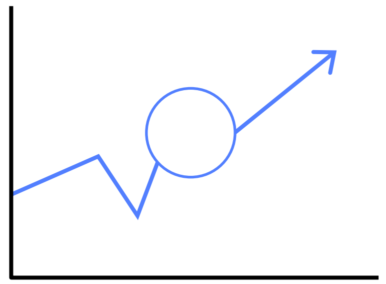


All Amazon EC2, AWS Fargate, and AWS Lambda workloads

# Amazon EC2 pricing options (2 of 2)

## Spot Instances

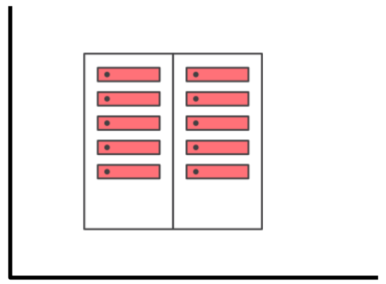
Spare Amazon EC2 capacity at substantial savings off On-Demand Instance prices.



Fault-tolerant, flexible, stateless workloads

## Dedicated Hosts

Physical server with Amazon EC2 instance capacity fully dedicated for your use.



Workloads that require the use of your own software licenses or single tenancy to meet compliance requirements

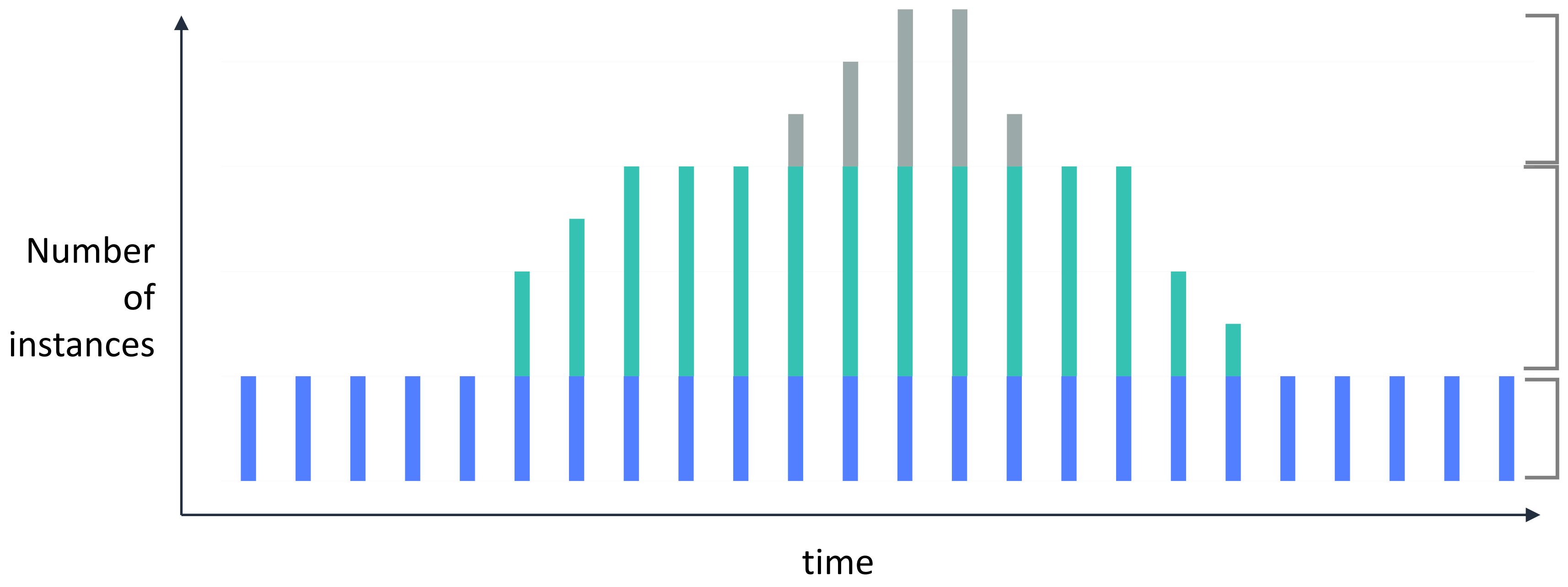
## Amazon EC2 dedicated options

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

| Dedicated Instances | Dedicated Hosts |
|---|---|
| • Per-instance billing<br>• Automatic instance placement<br>• Benefit – Isolates the hosts that run your instances | • Per-host billing<br>• Visibility of sockets, cores, and host ID<br>• Affinity between a host and an instance<br>• Targeted instance placement<br>• Add capacity by using an allocation request<br>• Benefit – Enables 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.



Scale using Spot Instances for fault-tolerant, flexible, stateless workloads

Use On-Demand Instances for new or stateful spiky workloads

Use Reserved Instances or Savings Plans for known, steady-state workloads

# Key takeaways

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

## Amazon EC2 cost optimization

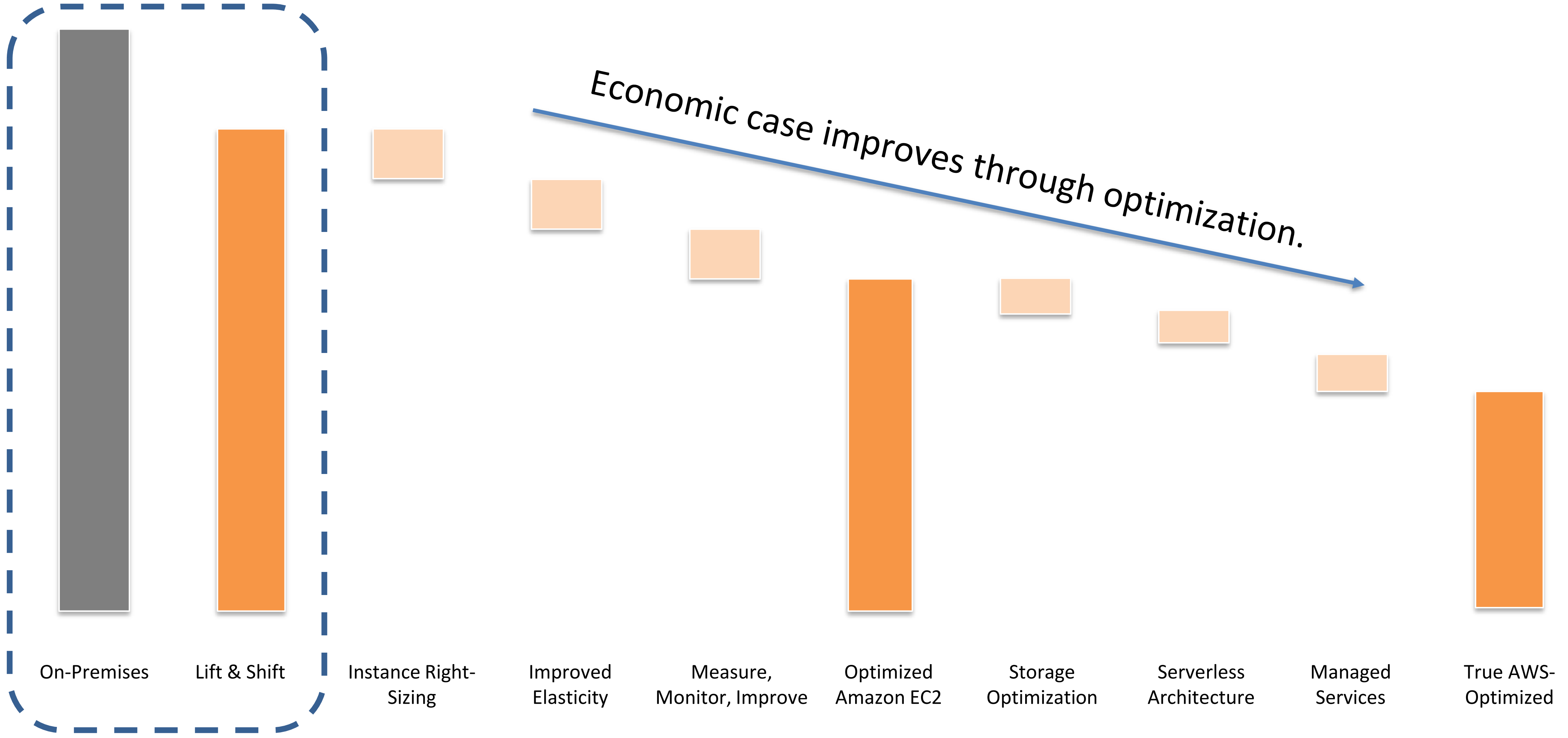
## What is Cost Optimization?

### Reduce Costs…

Pay only for ***what*** you need

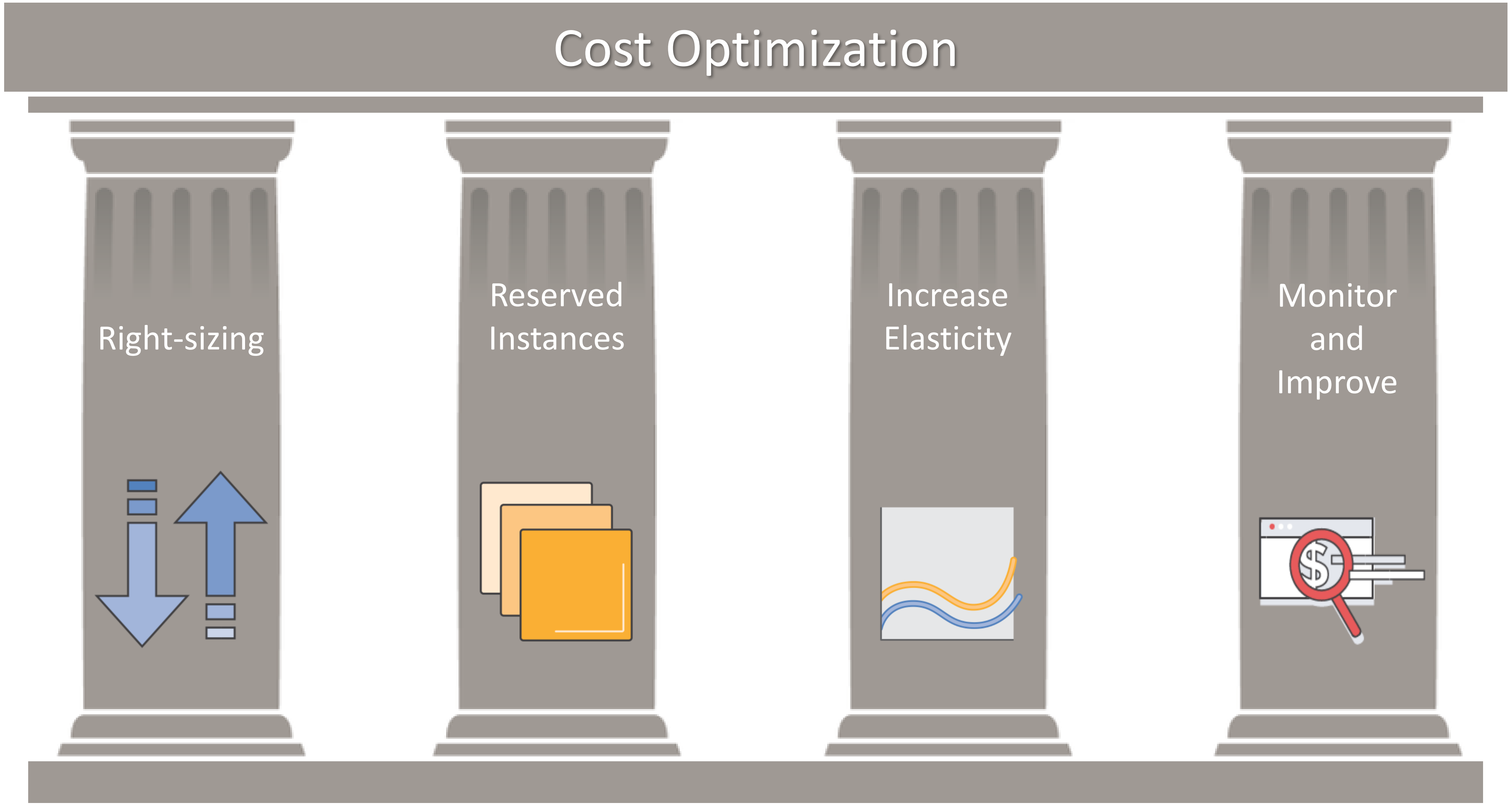
***when*** you need it.

## Lowering TCO Through Cost Optimization



Traditional TCO Comparisons

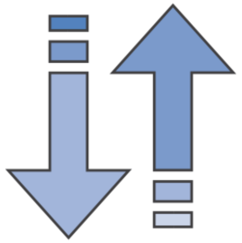
## The Four Pillars of Cost Optimization



# Driver 1: Right-Sizing

**Driver 1:**

Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve



- Select the appropriate instance types
- Downsize instances
- Leverage Amazon CloudWatch metrics

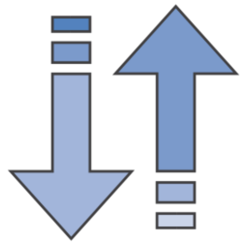
**Best practice:**

- Right size, then reserve

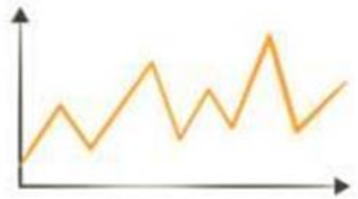
# Optimize and Combine Amazon EC2 Purchase Types

**Driver 1:**

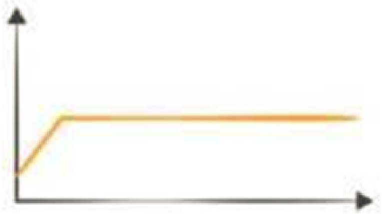
Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve



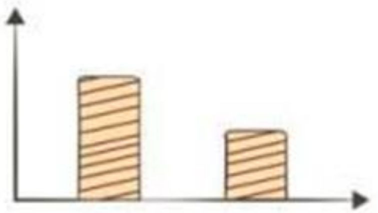
On-Demand



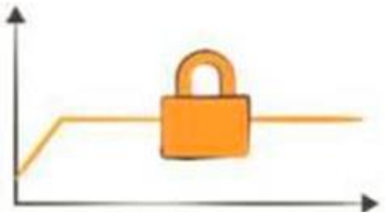
Reserved



Spot



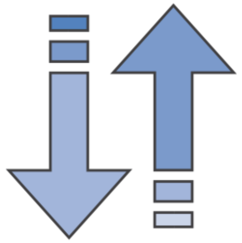
Dedicated



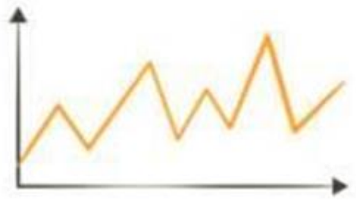
# Optimize and Combine Amazon EC2 Purchase Types

**Driver 1:**

Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve

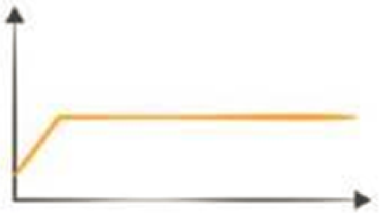


**On-Demand**

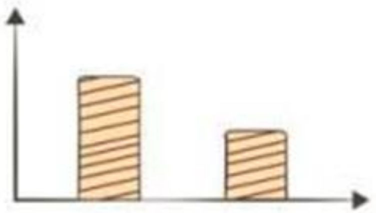


Spiky Workloads

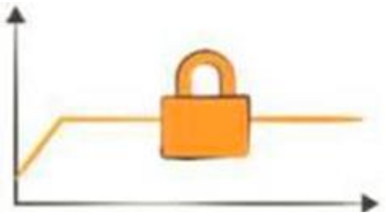
**Reserved**



**Spot**



**Dedicated**

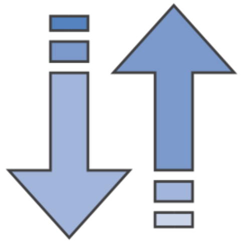


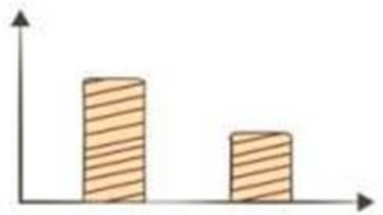
- Pay by the hour.
- No long-term commitments

# Optimize and Combine Amazon EC2 Purchase Types

**Driver 1:**

Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve



| On-Demand | Reserved | Spot | Dedicated |
|---|---|---|---|
| | |  | |
| Spiky Workloads | Steady-state Workloads | | |

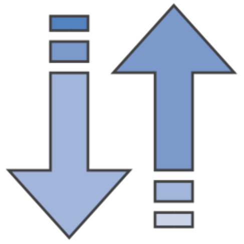
- Pay by the hour.
- No long-term commitments

- Pay upfront
- 50-75% lower hourly rate

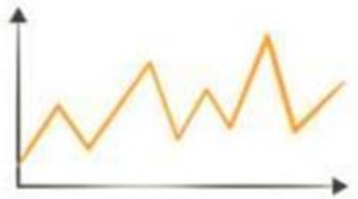
# Optimize and Combine Amazon EC2 Purchase Types

**Driver 1:**

Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve



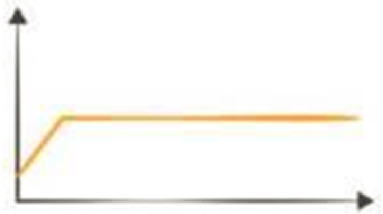
**On-Demand**



Spiky workloads

- Pay by the hour.
- No long-term commitments

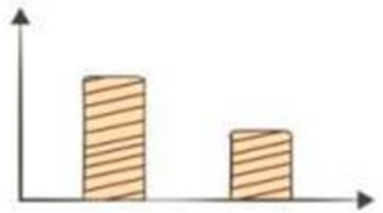
**Reserved**



Steady-state workloads

- Pay upfront
- 50-75% lower hourly rate

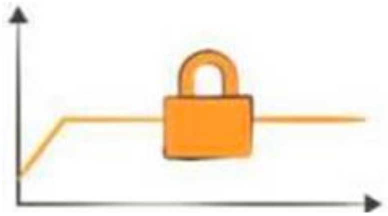
**Spot**



Time-insensitive workloads

- Bid for unused Amazon EC2 capacity

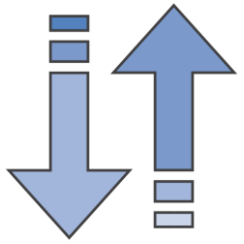
**Dedicated**



# Optimize and Combine Amazon EC2 Purchase Types

**Driver 1:**

Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve



**On-Demand**

**Spiky Workloads**

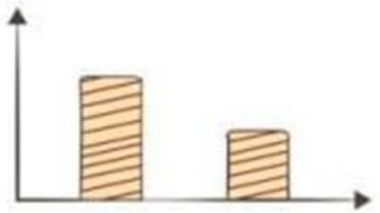
- Pay by the hour.
- No long-term commitments

**Reserved**

**Steady-state Workloads**

- Pay upfront
- 50-75% lower hourly rate

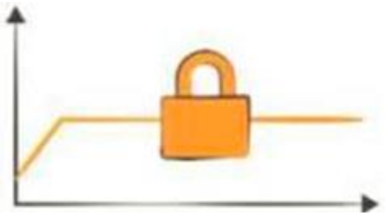
**Spot**



**Time-insensitive Workloads**

- Bid for unused Amazon EC2 capacity

**Dedicated**



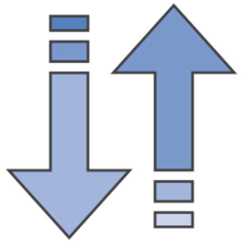
**Highly sensitive Workloads**

- In your VPC
- Isolated, steady-state workloads

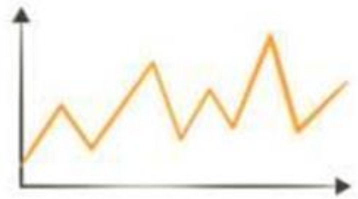
# Optimize and Combine Amazon EC2 Purchase Types

**Driver 1:**

Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve

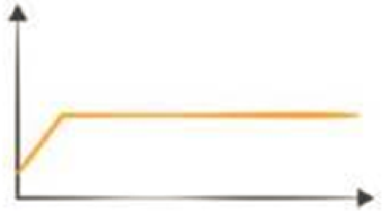


**On-Demand**



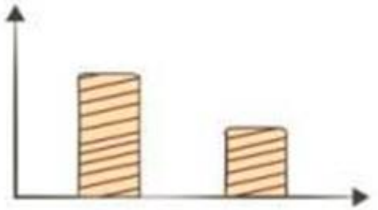
Spiky Workloads

**Reserved**



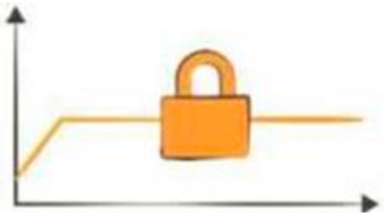
Steady-state Workloads

**Spot**



Time-insensitive Workloads

**Dedicated**



Highly sensitive Workloads

✓ Pay only for what you use
✓ On-demand, elastic provisioning
✓ Control and security

## Driver 2: Reserved Instance Capacity

**Driver 2:**

Right-Sizing
Reserved Instances (Ris)
Increase Elasticity
Monitor & Improve

### Reserved Instances (RIs)/Capacity

- Amazon Elastic Compute Cloud (EC2)
- Amazon Relational Database Service (RDS)
- Amazon DynamoDB
- Amazon Redshift
- Amazon ElastiCache

**Up to 75%+ savings**

### Commitment level

- 1 year
- 3 years

* Dependent on specific AWS service, size/type, and region

## Reserved Instances

**Driver 2:**

Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve

### Step 1: RI Coverage

- Cover always-on resources
- Target 70–80% always-on coverage

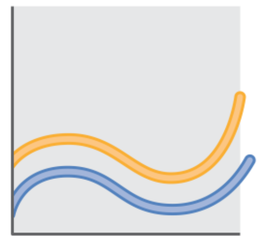
### Step 2: RI Utilization

- Leverage RI flexibility to increase utilization
- Merge and split RIs as needed
- Target 95% RI utilization rate

# Driver 3: Increase Elasticity

**Driver 3:**

Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve



- **Elasticity**
  Using an instance when you need, turning it off when you don’t

- **Turn off non-production instances**
  Example: Dev/test

- **Auto scale production**
  Use Auto Scaling to scale up and down based on demand and usage (e.g., spikes)

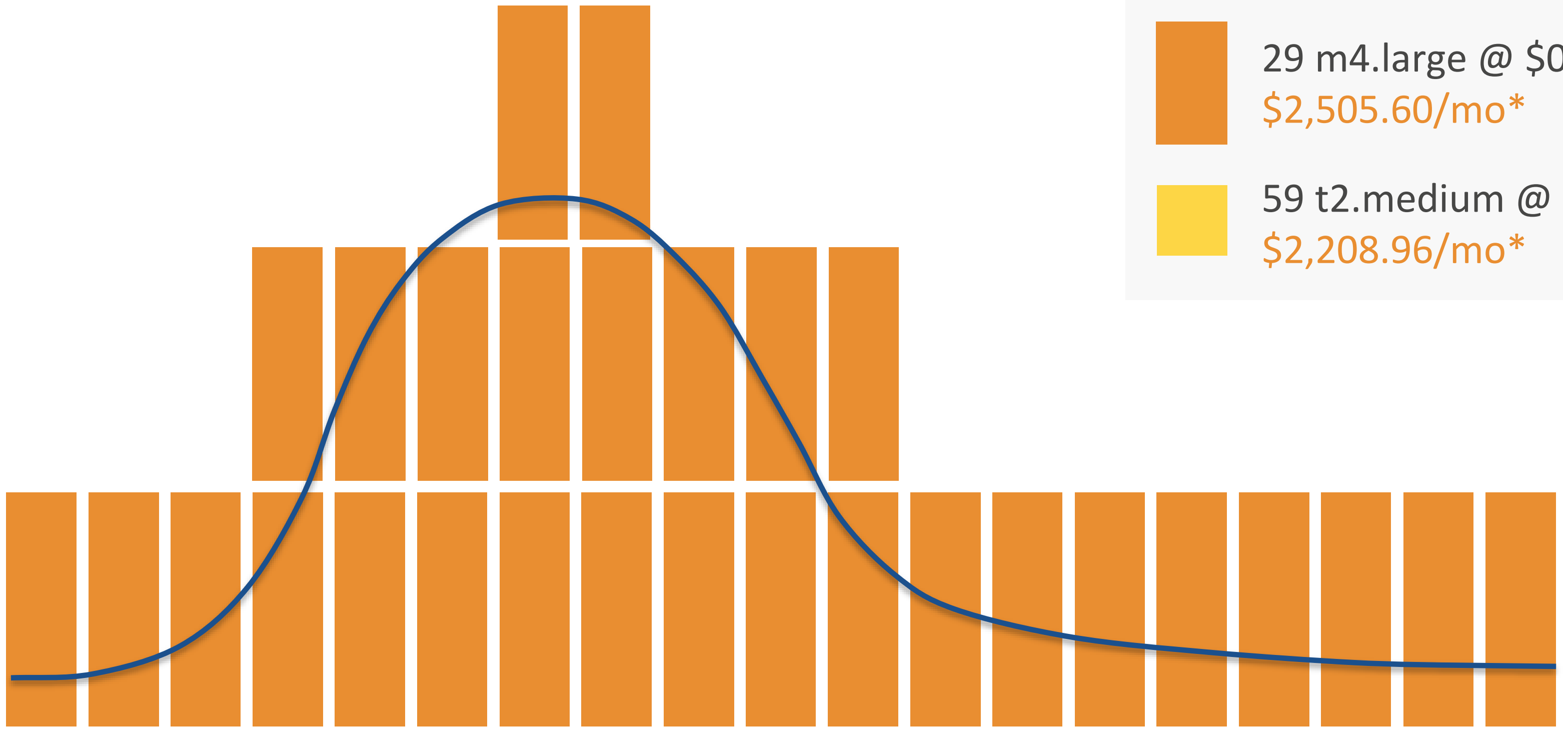
- **Target: 20-30% of Amazon EC2 instances**
  Run in On-demand or as Spot

## Using Right-sizing and Elasticity to Lower Cost

**Driver 3:**

Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve

More, smaller instances vs. fewer, larger instances



| | |
|---|---|
| 🟧 | 29 m4.large @ $0.120/hr<br>$2,505.60/mo* |
| 🟨 | 59 t2.medium @ $0.052/hr<br>$2,208.96/mo* |

*Assumes Linux instances in the US-East (N. Virginia) Region at 720 hours per month

## Driver 4: Measure, Monitor, and Improve

**Driver 4:**

Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve

**Cost Optimization Opportunities:**

1. Auto-tag resources
2. Identify always-on non production systems
3. Identify instances to downsize
4. Recommend Reserved Instance (RIs) to purchase
5. Dashboard your status
6. Consolidate your billing
7. Report on savings

## Measure, Monitor, and Improve

**Driver 4:**

Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve

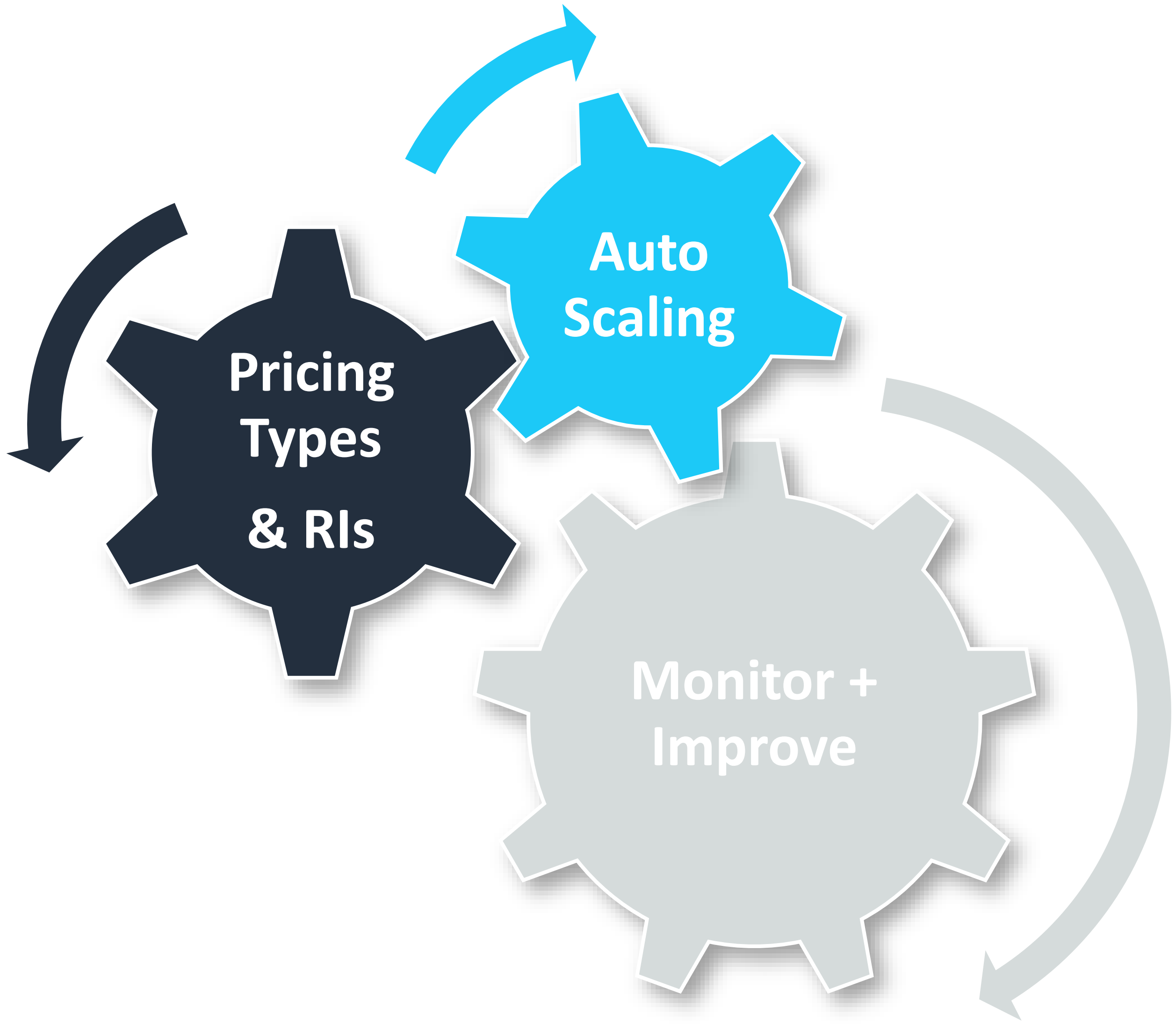
**AWS Trusted Advisor**

- Optimize your AWS environment
- Reduce cost, increase performance, and improve security

**Cost Explorer**

- View graphs of your costs: the last 13 months
- Forecast your likely costs: the next 3 months
- View time data by day or month

## Continual Process of Cost Optimization



## AWS Lambda service

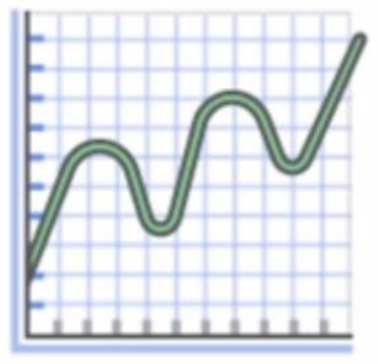
## What is AWS Lambda?

**AWS Lambda**

- Fully managed serverless compute
- Event-driven execution
- Sub-second metering
- Function execution limited to a maximum of 5 minutes
- Multiple languages supported

# Lambda Key Benefits

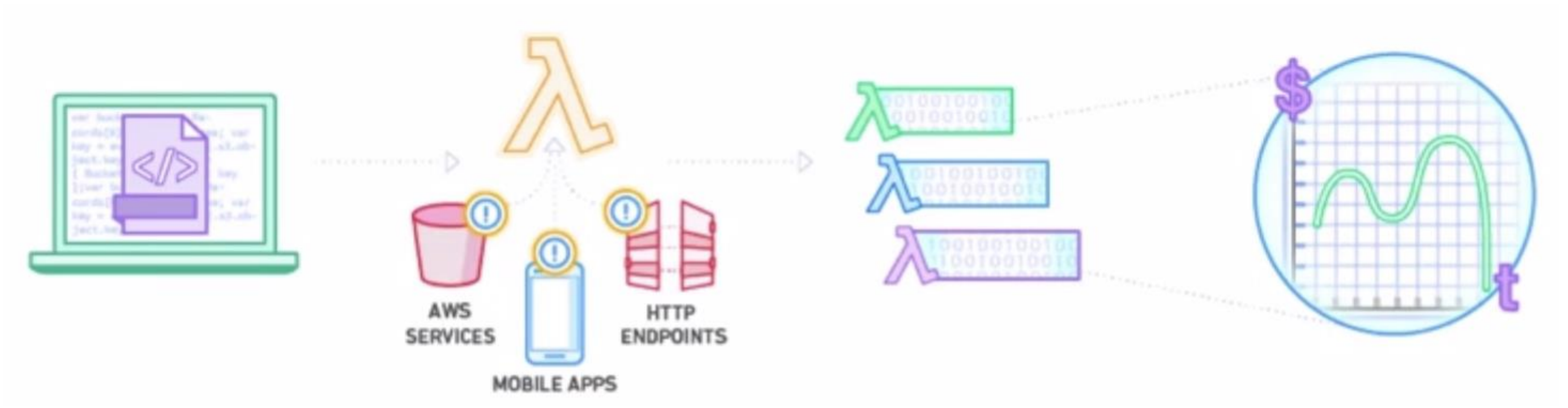
No Servers to Manage



Continuous Scaling

Sub-second Metering

## Getting Started with Lambda



Upload your code to AWS Lambda

Set up your code to trigger from other AWS services, HTTP endpoints, or in-app activity

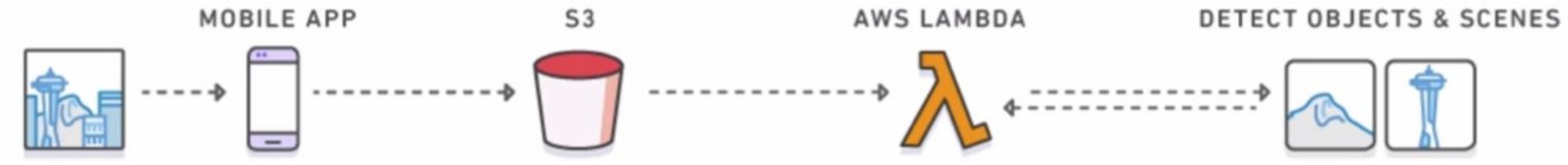
Lambda runs your code only when triggered using only the compute resources needed

Pay just for the compute time you use

## Lambda: Use Cases

- Run code in response to an events.
- For example:
    - Changes to an S3 bucket
    - Changes to an Amazon Dynamo DB table
    - Respond to HTTP request
    - Invoke code with API calls
- Build serverless applications triggered by Lambda functions.
- Deploy with AWS CodePipeline and AWS CodeDeploy.

# Lambda Example



1. User captures an image for their property listing.
2. The mobile app uploads the new image to Amazon S3.
3. A Lambda function is triggered and calls Amazon Rekognition.
4. Amazon Rekognition retrieves the image from Amazon S3 and returns labels for the detected property and amenities.

## In Review

- Fully managed serverless compute
- Event-driven execution
- Executes code only when needed and scales automatically
- Multiple languages supported

## AWS Elastic Beanstalk

## What is Elastic Beanstalk?

**AWS Elastic Beanstalk**

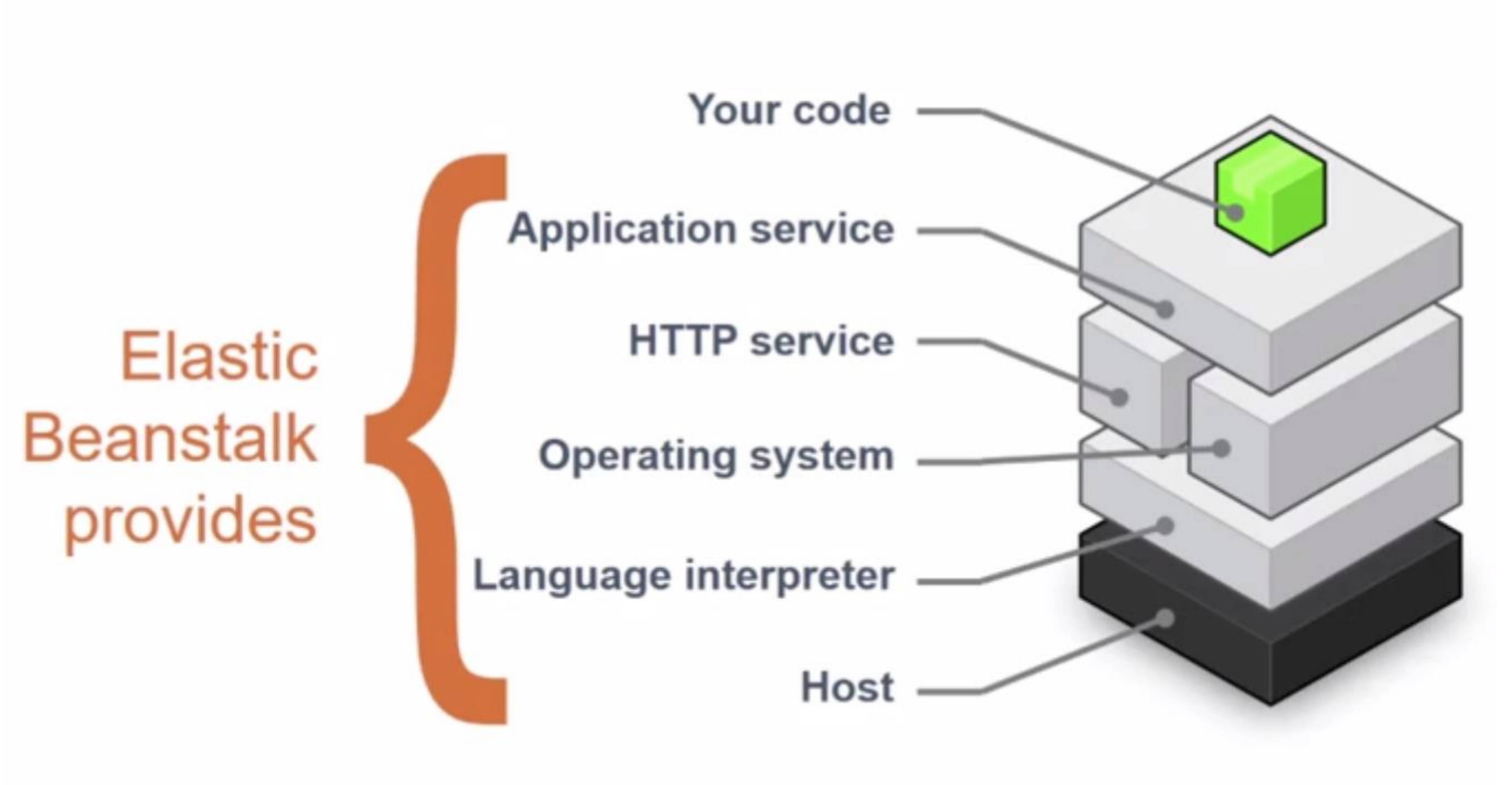
- Platform as a Service (PaaS)
- Quickly deploys, scales, and manages web apps
- Reduces management complexity
- Keeps control in your hands:
  - Choose your instance type
  - Choose your database
  - Set ant adjust Auto Scaling
  - Update your application
  - Access server log files
  - Enable HTTPS on load balancer

## What is Elastic Beanstalk?

**AWS Elastic Beanstalk**

- Supports a large range of platforms:
  - Packer Builder
  - Single Container, Multi-container, or Pre-configured Docker
  - Go
  - Java SE
  - Java with Tomcat
  - .NET on Windows Server with IIS
  - Node.js
  - PHP
  - Python
  - Ruby
- No charge for Elastic Beanstalk; pay only for the underlying services used.

## Elastic Beanstalk Components



## Elastic Beanstalk Key Benefits

Fast and simple to begin

Developer productivity

Impossible to outgrow

Complete resource control

## In Review

- Enhances developer productivity by simplifying the process of deploying your application.
- Reduces management complexity.
- There is no charge for Elastic Beanstalk. You pay only for the services you use.

# Sample exam question

A Solutions Architect wants to design a solution to save costs for EC2 instances that do not need to run during a 2-week company shutdown. The applications running on the instances store data in instance memory (RAM) that must be present when the instances resume operation.

Which approach should the Solutions Architect recommend to shut down and resume the instances?

A. Modify the application to store the data on instance store volumes. Reattach the volumes while restarting them.

B. Snapshot the instances before stopping them. Restore the snapshot after restarting the instances.

C. Run the applications on instances enabled for hibernation. Hibernate the instances before the shutdown.

D. Note the Availability Zone for each instance before stopping it. Restart the instances in the same Availability Zones after the shutdown.

# Additional resources

- Amazon EC2 User Guide for Linux Instances
- Amazon EC2 User Guide for Windows Instances
- Amazon EC2 FAQs
- EC2 Image Builder User Guide
- EC2 Image Builder FAQs
- AWS Compute Optimizer User Guide
- AWS Compute Optimizer FAQs
- How AWS Pricing Works