Cloud Compute with AWS

Welcome to the fifth lesson of the AWS Solutions Architect Associate level course — Cloud Compute with AWS

By the end of the lesson you will be able to:

- Explain EC2 and its uses
- Describe Amazon Machine Image
- List all the Instance Types
- Explain what EBS is and how EFS will be used
- Describe how EBS snapshots are taken and used
- Check the different instance pricing options
- Use placement groups
- Illustrate Elastic Load Balancers and Auto Scaling
- Use Elastic Beanstalk, Lambda, and ECS instead of EC2
- Explain the fundamentals of the AWS Command Line Interface
- Evaluate the AWS recommended best practices for EC2 and take a look at the costs associated with EC2

By the end of this lesson you'll be able to:

- Get an overview of EC2 and understand what it's used for
- Know what an Amazon Machine Image is
- List all the Instance Types
- Understand what EBS is and how EFS will be used
- Learn how EBS snapshots are taken and used
- Review the different instance pricing options
- Acquire an understanding of when to use placement groups
- Take a look at Elastic Load Balancers and Auto Scaling
- Learn how Elastic Beanstalk and Lambda and ECS can be used instead of EC2
- Learn the basics of the AWS Command Line Interface

- Review the AWS recommended best practices for EC2 and take a look at the costs associated with EC2

Amazon EC2 Overview

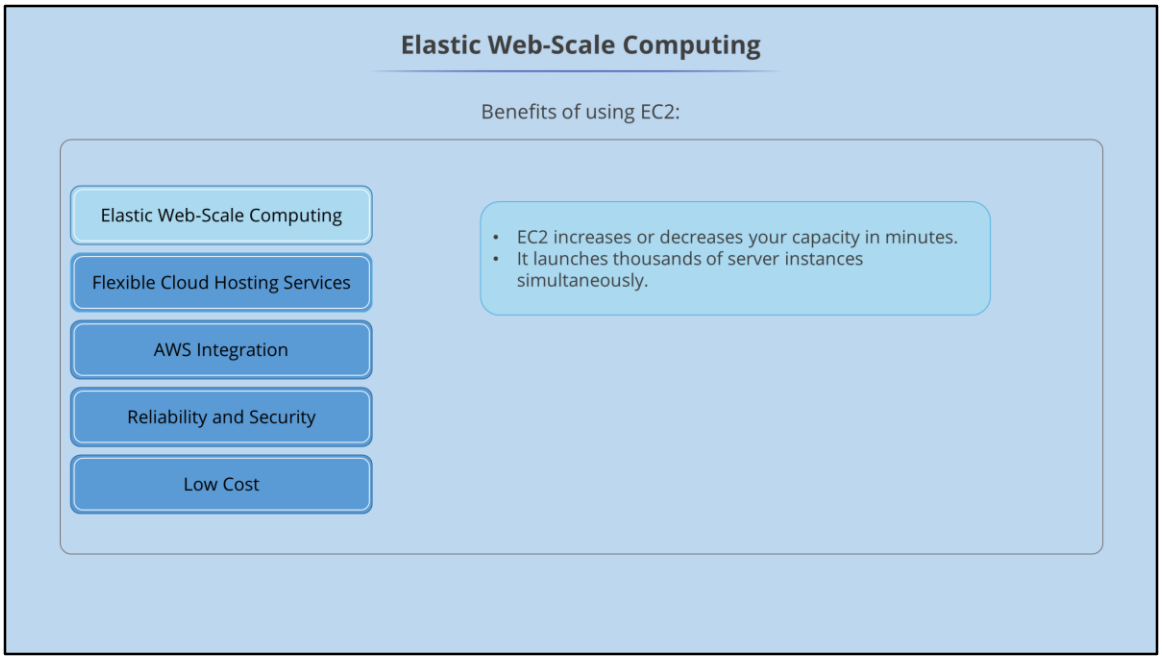
In this section you will learn about EC2 and its benefits.

Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud.

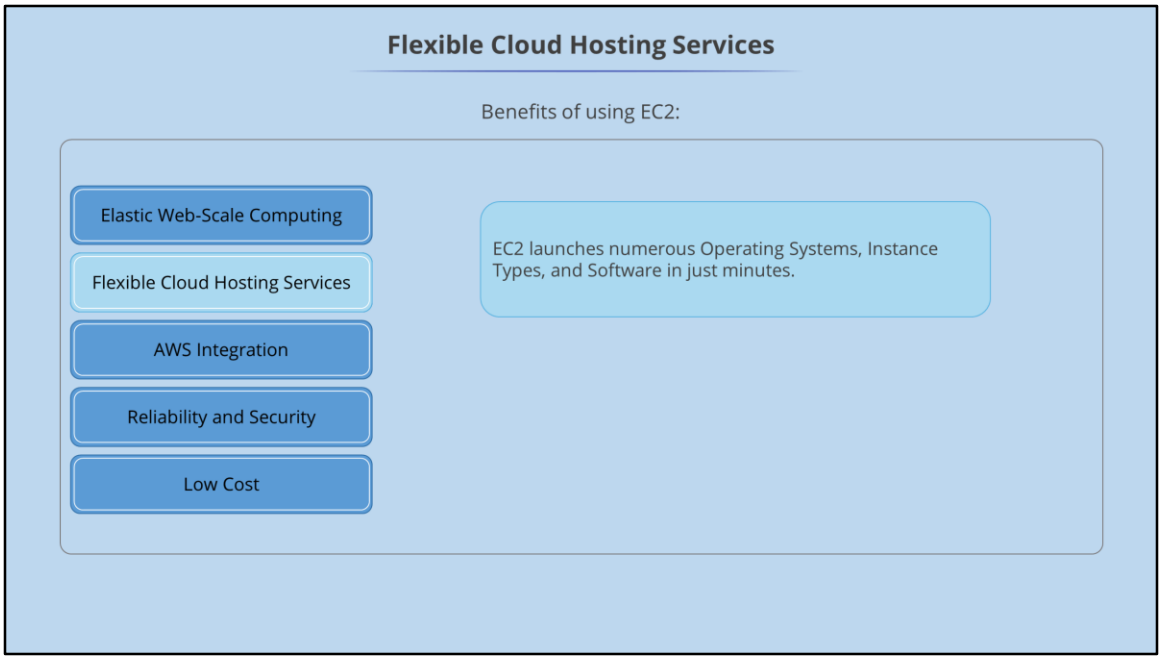
Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It provides a variety of different options to create your own cloud instances. EC2 instances use EBS volumes to store data.

You can launch your own virtual servers; Windows, Linux, etc., anywhere in the world on AWS infrastructure to run your applications, websites, databases, and more.

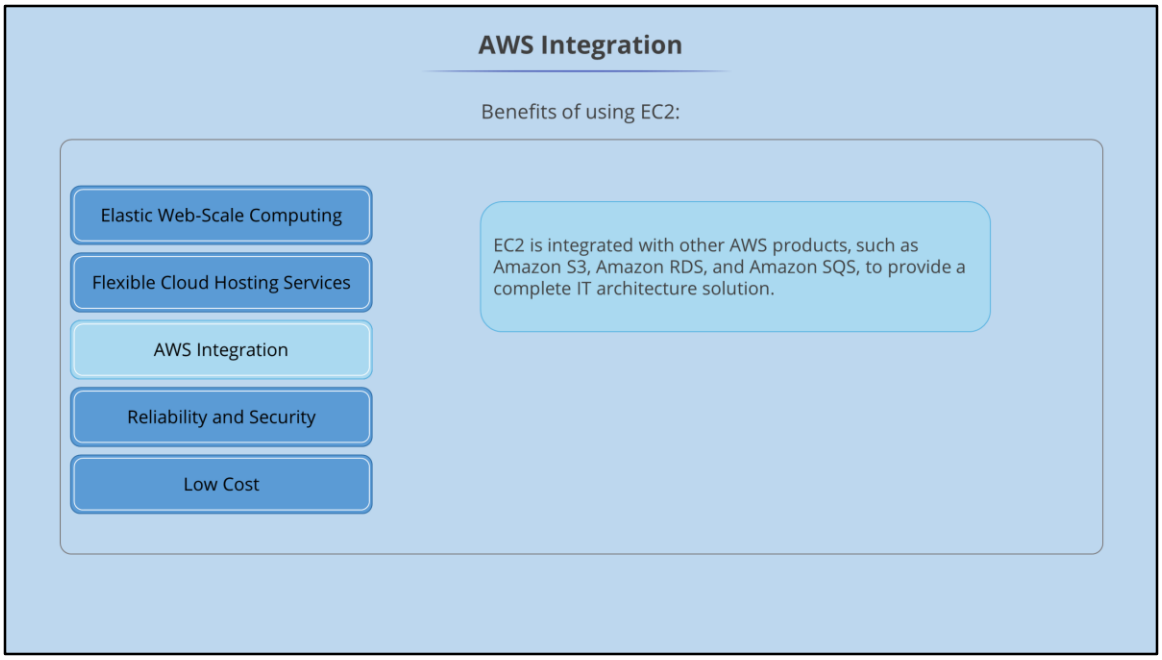


Some of the benefits of using EC2 are:

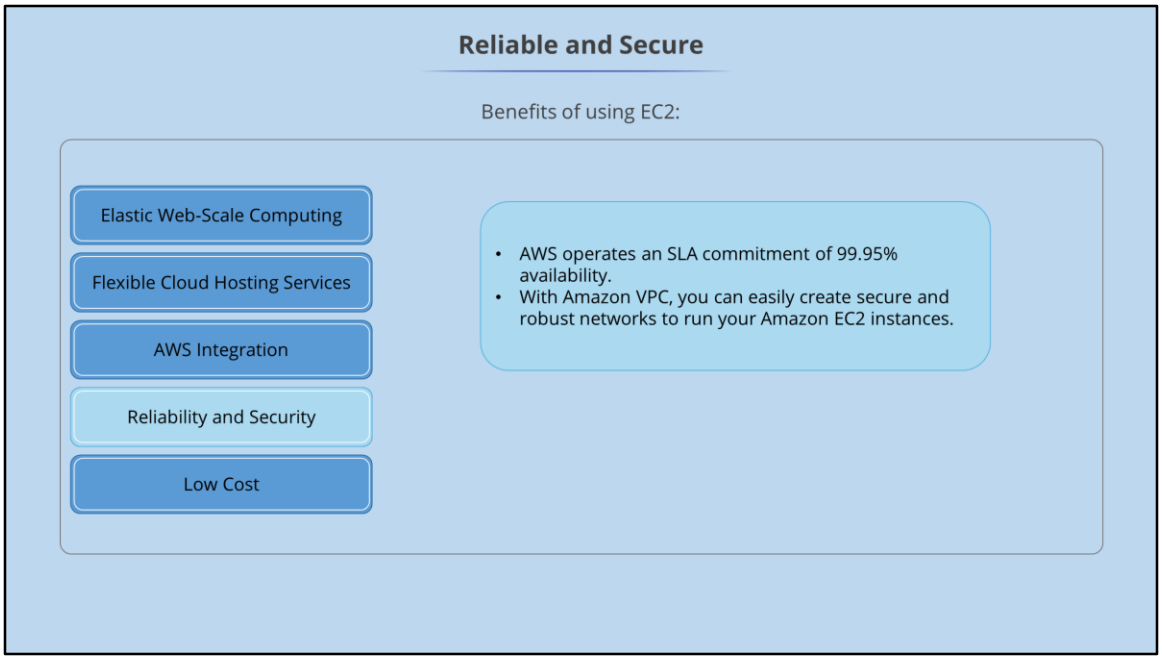
EC2 allows you to increase or decrease your capacity in minutes. You can launch hundreds or thousands of server instances simultaneously.



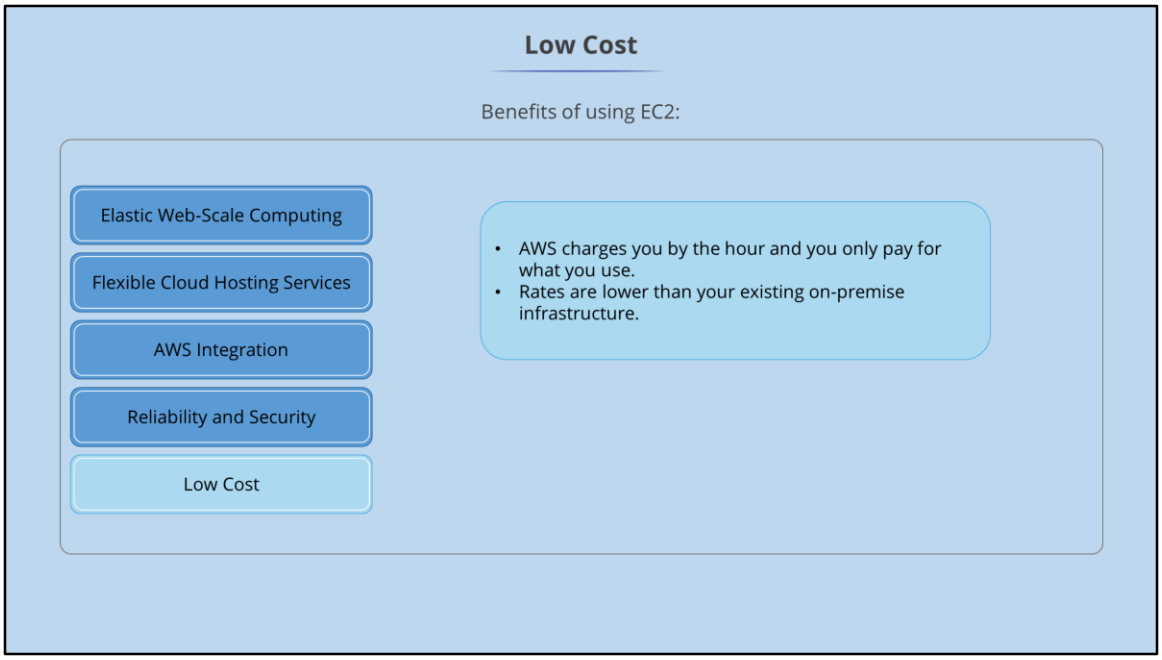
Numerous Operating Systems, Instance Types, and Software Packages are available to you to launch in just minutes.



Amazon EC2 is designed to be integrated fully with other AWS products, such as Amazon S3, Amazon RDS, and Amazon SQS to provide you with a complete IT architecture solution.



AWS operates an SLA commitment of 99.95% availability and using Amazon VPC you can easily create secure and robust networks to run your Amazon EC2 instances.



AWS charges you by the hour and you only pay for what you use. The rates are lower than your existing on-premise infrastructure.

Amazon Machine Images (AMI)

In this section you'll learn what Amazon Machine Images are and what they are used for. You'll learn about AMIs that AWS provides and AMIs in the AWS marketplace. Then you will learn to create AMIs.

Amazon Machine Image (AMI)

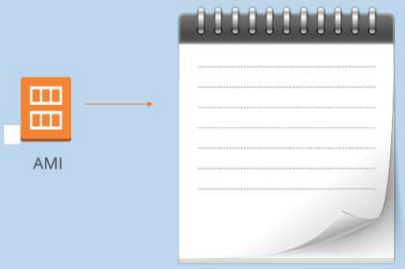
Amazon's definition of an AMI:
"An Amazon Machine Image (AMI) provides the information required to launch an instance. You specify an AMI when you launch an instance, and you can launch as many instances from the AMI as you need. You can also launch instances from as many different AMIs as you need."

An Amazon Machine Image (AMI) provides the information required to launch an instance. You specify an AMI when you launch an instance, and you can launch as many instances from the AMI as you need. You can also launch instances from as many different AMIs as you need.

Amazon Machine Image (AMI) (Contd.)

AMI is a virtual instance that includes:

- A template for the root volume for the instance
- Launch permissions to control AMI launch instances
- A block device mapping that specifies volumes to attach to the instance

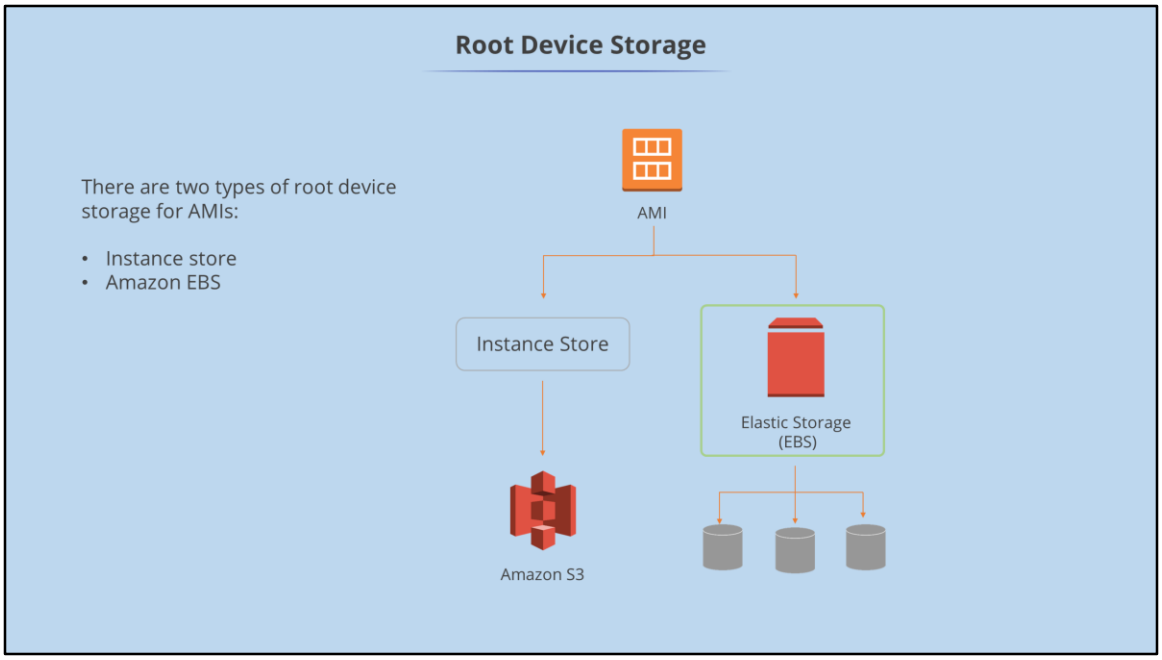


An AMI is a template of a virtual instance that includes the following:

A template for the root volume for the instance (for example, an Operating System, an application server, and applications).

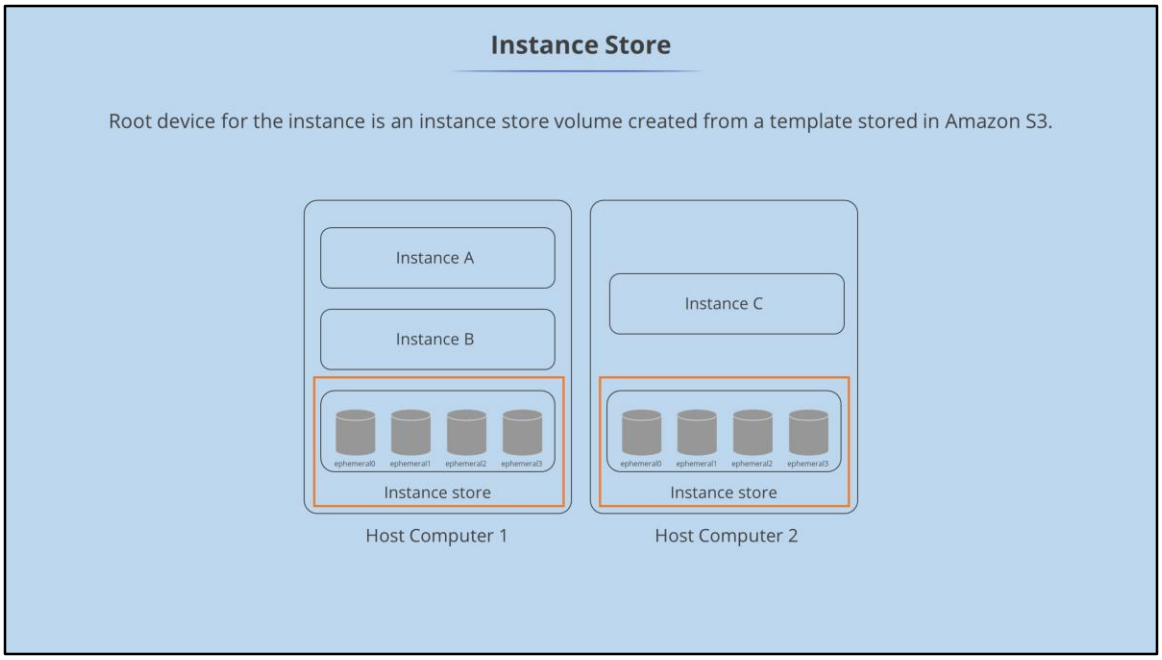
Launch permissions that control which AWS accounts can use the AMI to launch instances.

A block device mapping that specifies the volumes to attach to the instance when it is launched.

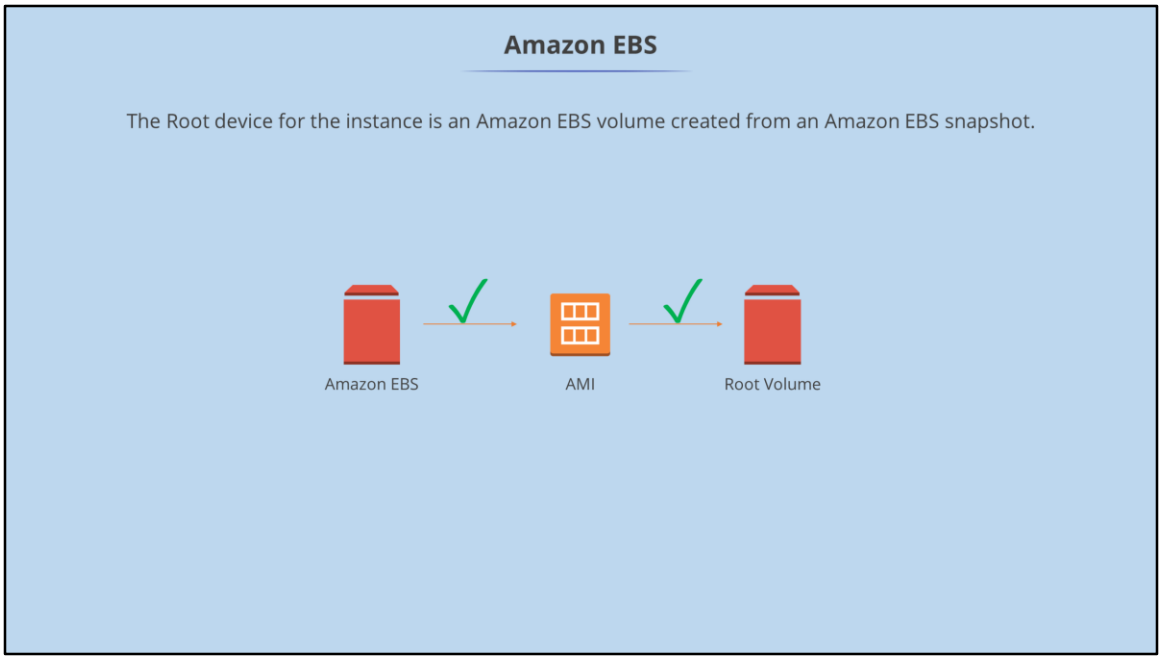


There are two types of root device storage for AMIs: Amazon EBS and Instance store.

When Amazon first launched EC2, all root volumes were instance store volumes and were hosted on Amazon S3. However, EBS was launched to provide an alternative, which is Amazon recommended root volume storage type.



Backed by instance store means that the root device for the instance is an instance store volume created from a template stored in Amazon S3. This was the original EC2 approach. When an instance terminates, as there is no concept of stopping and starting, all the data is lost. Therefore, Instance store volumes are known as ephemeral storage.



Backed by Amazon EBS means that the root device for the instance is an Amazon EBS volume created from an Amazon EBS snapshot. This is AWS's newer and recommended approach. Data on the root volume persists following an instance restart. By default, when you terminate an instance, the root volume is deleted. However, this can be changed. All data on other EBS volumes will remain following instance termination.

Amazon EBS vs. Instance Store

This table shows the different characteristics between the EBS-Backed and Amazon Instance Store-Backed volumes.

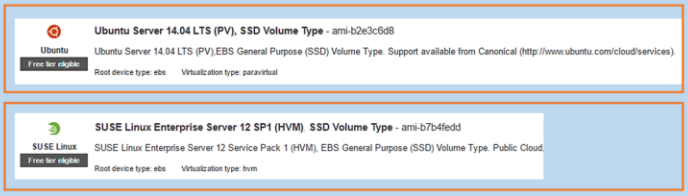
Characteristic	Amazon EBS-Backed	Amazon Instance Store-Backed
Boot time	Usually less than 1 minute	Usually less than 5 minutes
Size limit	16 TiB	10 GiB
Root device volume	Amazon EBS volume	Instance store volume
Data persistence	By default, the root volume is deleted when the instance terminates.* Data on any other Amazon EBS volumes persists after instance termination by default. Data on any instance store volumes persists only during the life of the instance.	Data on any instance store volumes persists only during the life of the instance. Data on any Amazon EBS volumes persists after instance termination by default.
Upgrading	The instance type, kernel, RAM disk, and user data can be changed while the instance is stopped.	Instance attributes are fixed for the life of an instance.
Charges	You're charged for instance usage, Amazon EBS volume usage, and storing your AMI as an Amazon EBS snapshot.	You're charged for instance usage and storing your AMI in Amazon S3.
AMI creation/bundling	Uses a single command/call	Requires installation and use of AMI tools
Stopped state	Can be placed in stopped state where instance is not running, but the root volume is persisted in Amazon EBS	Cannot be in stopped state; instances are running or terminated

This Table shows the different characteristics between the EBS-Backed and Amazon Instance Store-Backed volumes.

HVM vs. PV

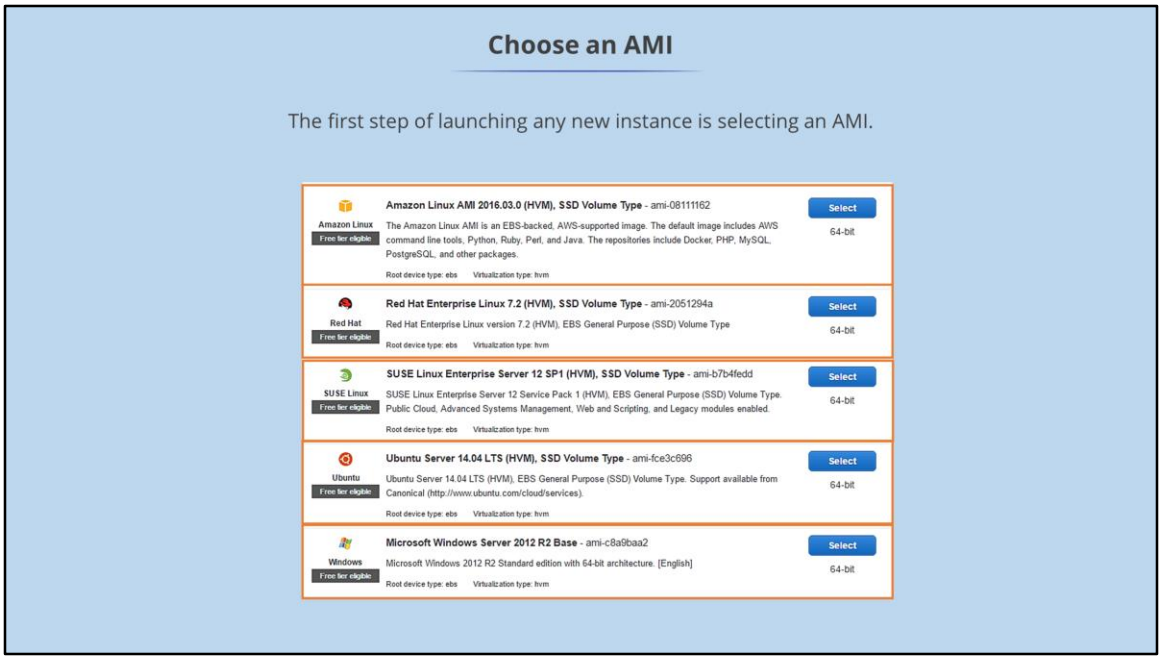
Linux Amazon Machine Images use one of the two types of virtualization:

- ParaVirtual (PV)
- Hardware Virtual Machine (HVM)

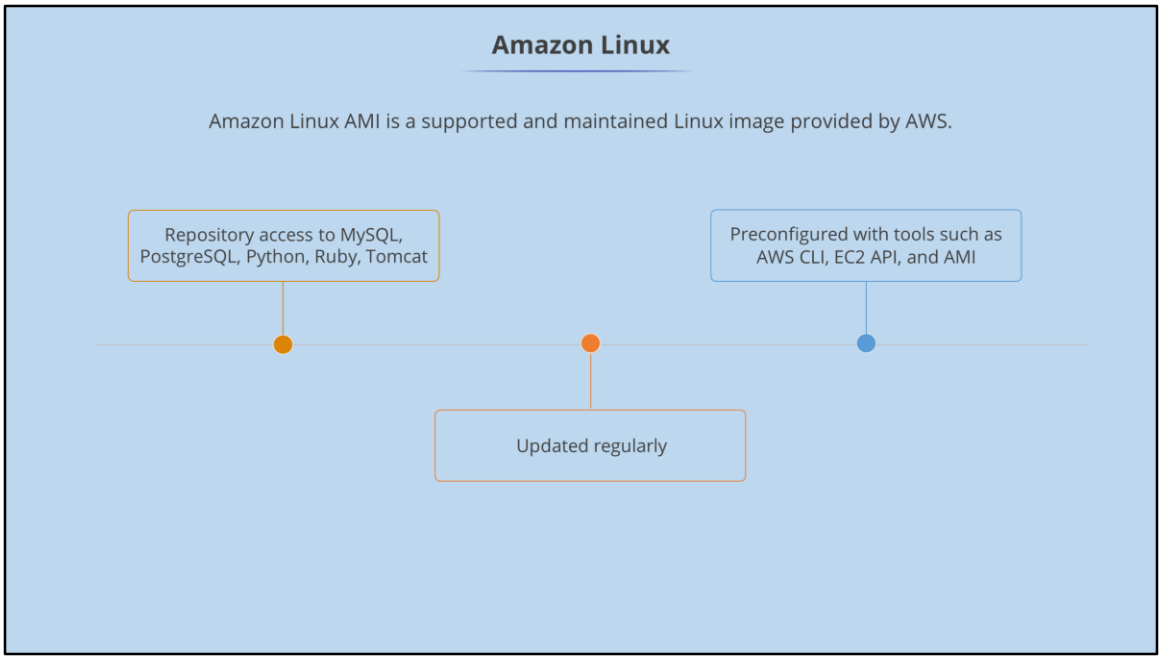


Linux Amazon Machine Images uses either one of the two types of virtualization: ParaVirtual (PV) or Hardware Virtual Machine (HVM). The main difference between PV and HVM AMIs is the way in which they boot and whether they are able to leverage special drivers for I/O that avoid the overhead of emulating network and disk hardware.

But now these drivers are available for HVM instances on the AWS platform, so the performance differences are negligible. AWS recommends using HVM in future.



First step of launching any new instance is selecting an AMI. Quick Start dashboard offers a number of AWS-configured AMIs, such as Windows, Red Hat Linux, Ubuntu, Amazon Linux in 32/64 bit versions (where applicable).



Amazon Linux AMI is a supported and maintained Linux image provided by AWS. It's worth taking a look at as AWS can include questions on it in the exam.

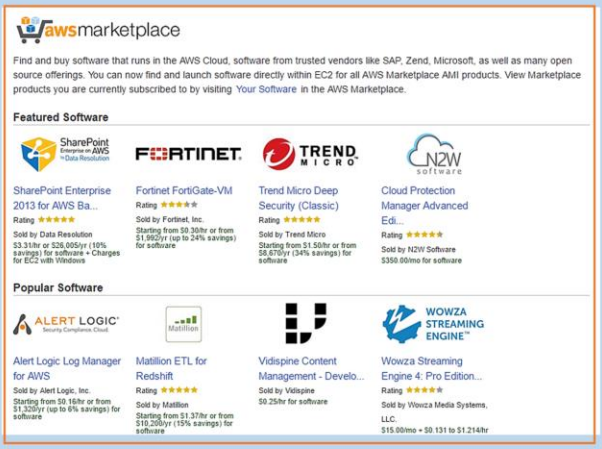
The Amazon Linux AMI provides repository access to multiple versions of MySQL, PostgreSQL, Python, Ruby, Tomcat, and other common packages.

Amazon Linux AMI is updated on a regular basis to include the latest components, and these updates are also made available in the yum repositories for installation on running instances.

It includes packages that enable easy integration with AWS services, such as the AWS CLI, Amazon EC2 API and AMI tools, the Boto library for Python, and the Elastic Load Balancing tools.

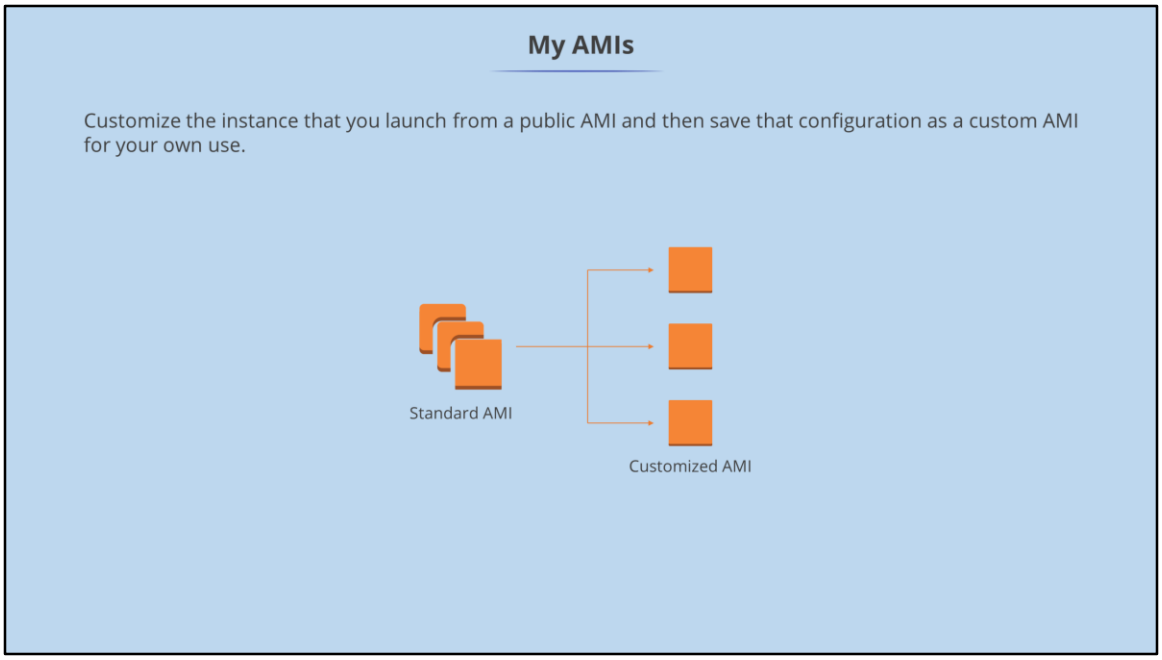
AWS Marketplace

The AWS Marketplace is an online store where you can buy software that runs on AWS from third-party vendors.

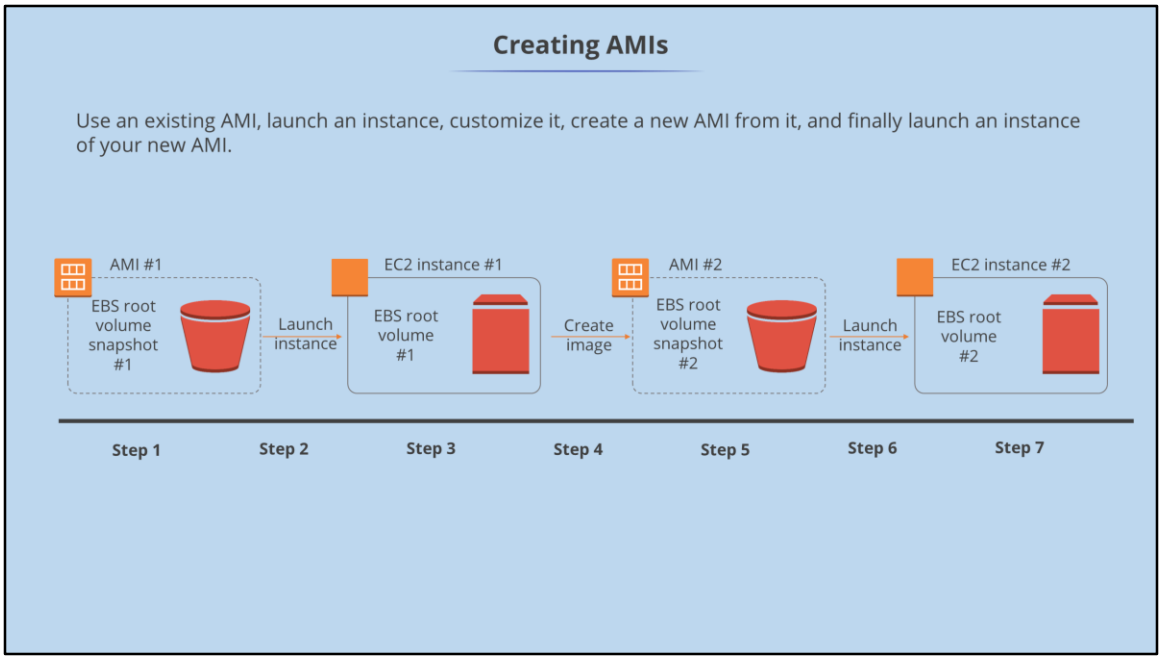


The AWS Marketplace is an online store where you can buy software that runs on AWS from third-party vendors such as Microsoft and SAP. This is called a paid AMI. The instance is charged according to the rates set by the owner of the AMI as well as the standard usage fees for the related web services.

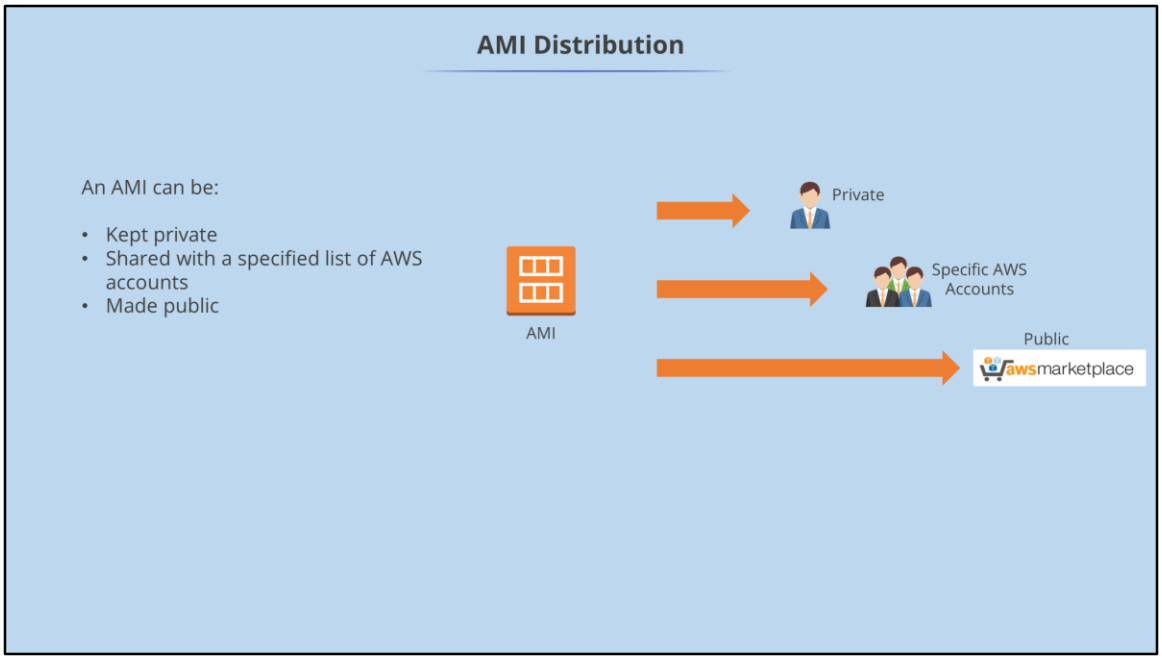
The AMI catalog is a community resource where people and development teams can list and exchange software or projects under development, without having to undergo extensive vetting. Listings in the community AMI catalog may or may not be from well-known vendors and generally have not undergone additional investigations.



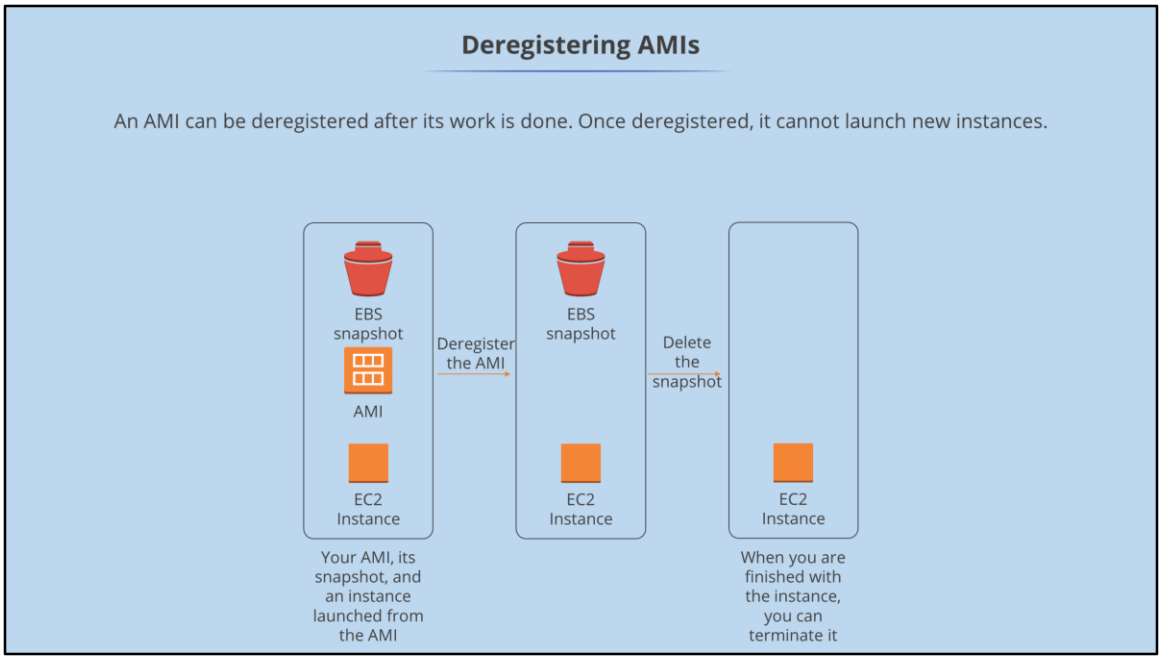
. You can customize the instance that you launch from a public AMI and then save that configuration as a custom AMI for your own use. Instances that you launch from your AMI use all the customizations that you've made.



Start with an existing AMI, launch an instance, customize it, create a new AMI from it, and finally launch an instance of your new AMI.



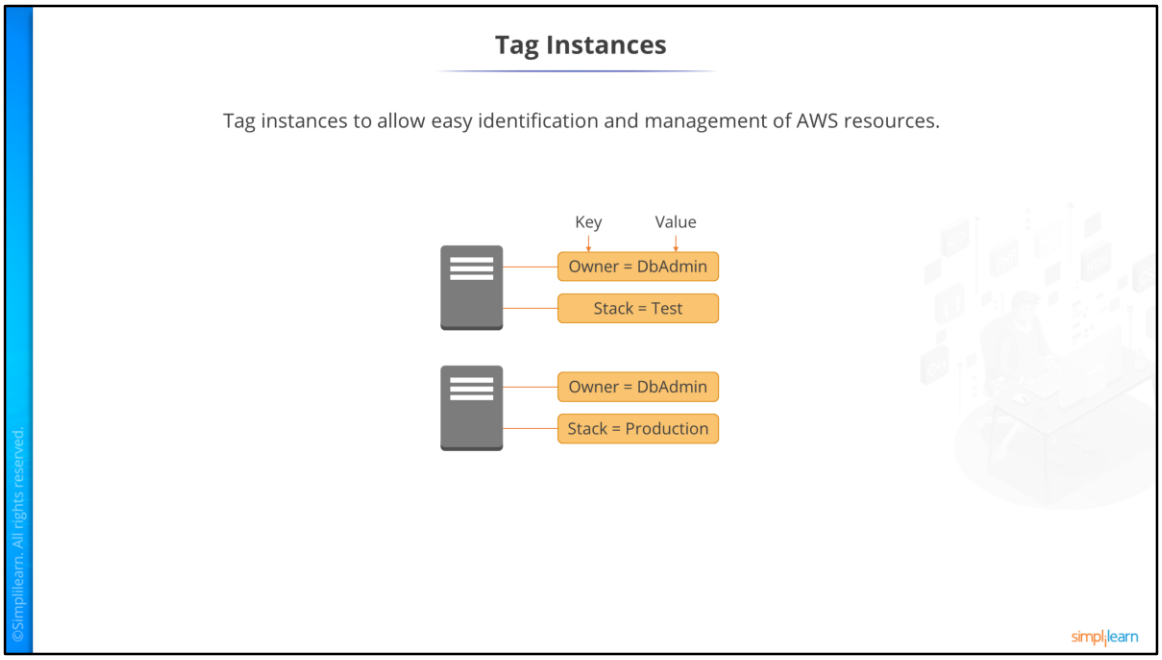
After you create an AMI, you can keep it private so that only you can use it, or you can share it with a specified list of AWS accounts. You can also make your custom AMI public so that the community can use it.



You can deregister an AMI when you have finished using it. After you deregister an AMI, you can't use it to launch new instances.

When you deregister an AMI, it doesn't affect any of the instances that you've already launched from the AMI so they will continue to run and incur costs.

When you deregister an Amazon EBS-backed AMI, the snapshot that was created for the root volume of the instance during the AMI creation process will still exist. To stop incurring storage costs for the snapshot, you need to delete it.



Always tag your instances to allow easy identification and management of AWS resources.

You can use Resource Groups and Tag Editor to view all your resources.

Demo: Launch and Connect to an EC2 Linux Instance

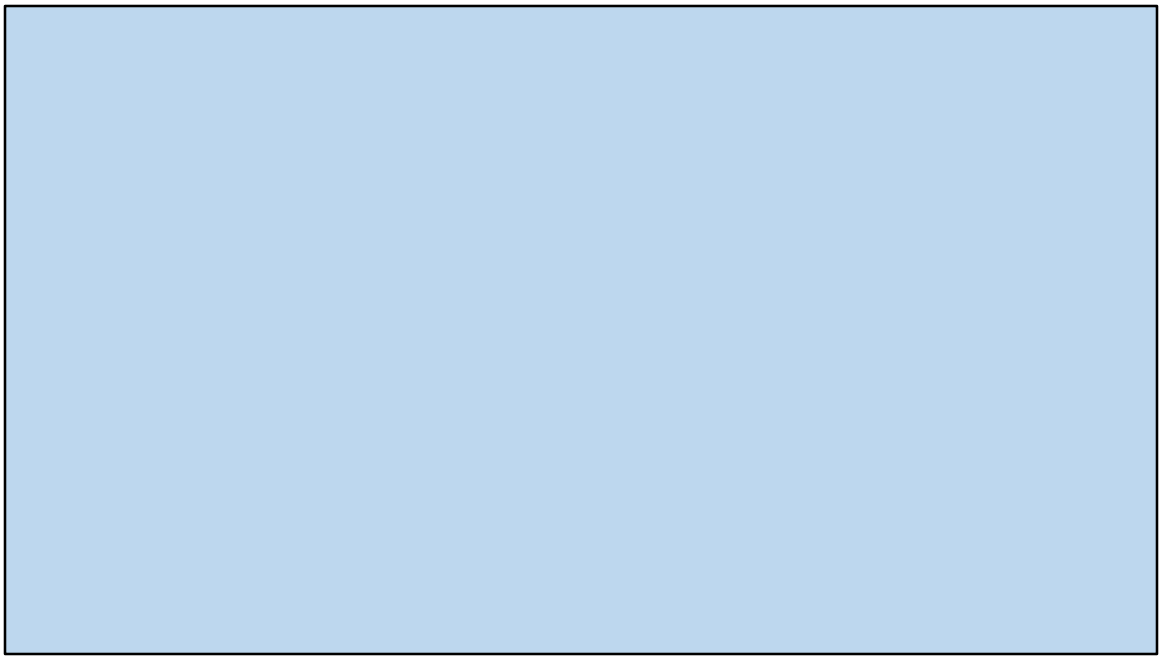
In this demonstration you will learn to launch and connect to an EC2 Linux instance.

Demo: Launch and Connect to an EC2 Windows Instance

In this demonstration you will learn to launch and connect to an EC2 Windows instance.

Demo: Create an AMI

In this demonstration you will learn to create an AMI and launch a new server from it.



(c) Avantus Traning Pte Ltd

1 **Which combination of root volume type and virtualization does AWS recommend?**

HVM and Instance Store

HVM and EBS Volumes

PV and Instance Store

PV and EBS Volumes

1 **Which combination of root volume type and virtualization does AWS recommend?**

HVM and Instance Store

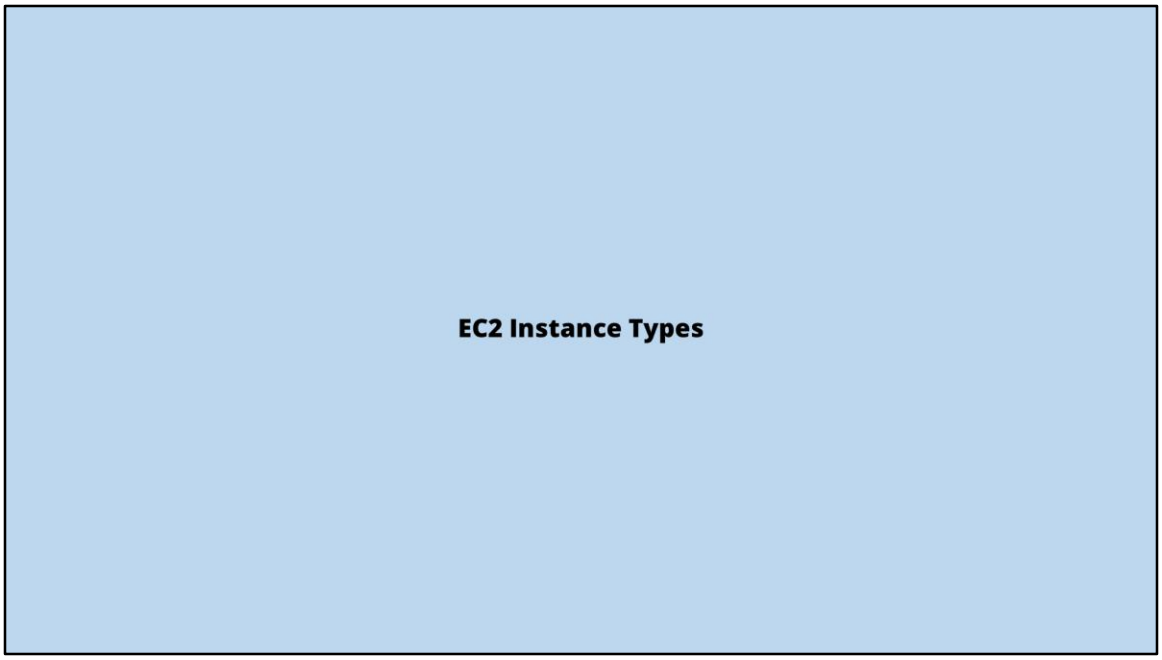
HVM and EBS Volumes

PV and Instance Store

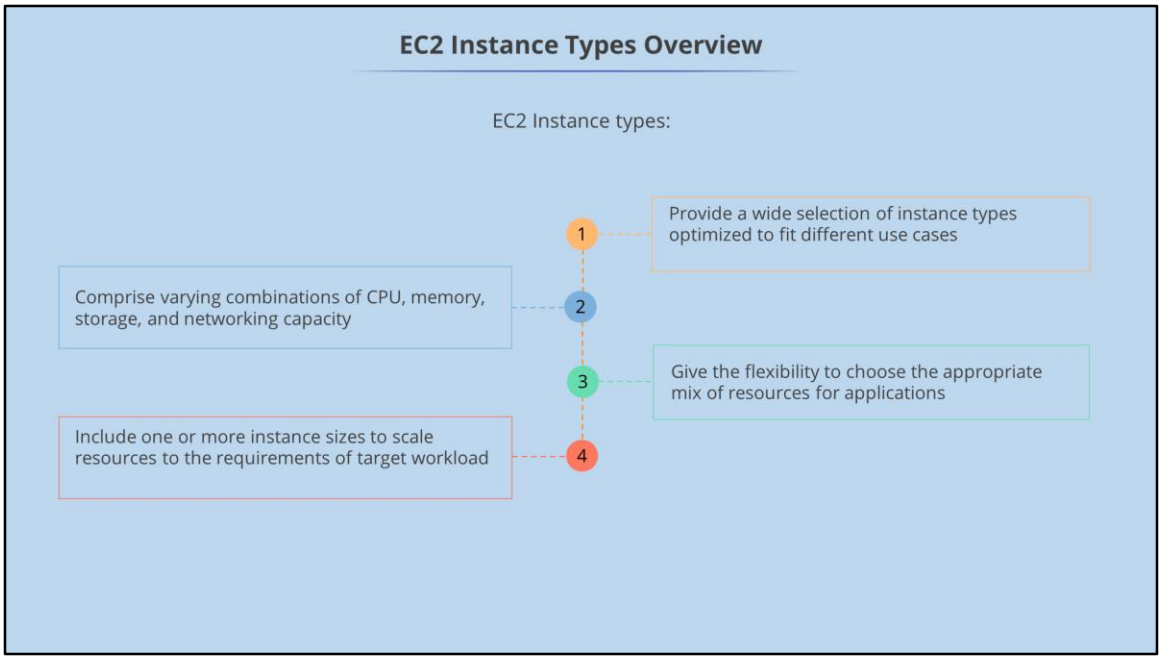
PV and EBS Volumes

b

AWS recommends HVM and EBS Volumes.



In this section, you'll learn about the different EC2 instances available, and when to use them.



EC2 Instance types:

- Provide a wide selection of instance types optimized to fit different use cases
- Comprise varying combinations of CPU, memory, storage, and networking capacity
- Give the flexibility to choose the appropriate mix of resources for applications
- Include one or more instance sizes to scale resources to the requirements of target workload

EC2 Instance Types

EC2 Instance types can be classified as:

- General Purpose
- Compute Optimized
- Memory Optimized
- Graphics Processing Unit (GPU)
- Storage Optimized
- EBS Optimized

FAMILY	Cost	Use
T2	Lowest	Small/medium databases and web servers
M4	Low-Medium	Application servers
M5	Low-Medium	Application servers

Suitable for small and mid-size databases, data processing tasks that require additional memory, caching fleets, and for running backend servers for SAP, Microsoft SharePoint, cluster computing, and other enterprise applications. M5 instances deliver up to 14% improvement in price/performance compared to M4 instances with the updated processor.

EC2 Instance Types (Contd.)

EC2 Instance types can be classified as:

- General Purpose
- Compute Optimized
- Memory Optimized
- Graphics Processing Unit (GPU)
- Storage Optimized
- EBS Optimized

FAMILY	Cost	Use
C4	Medium-High	CPU intensive databases, web servers, application servers, etc.
C5	Medium-High	CPU intensive databases, web servers, application servers, etc.

High performance front-end fleets, web-servers, batch processing, distributed analytics, high performance science and engineering applications, ad serving, MMO gaming, and video-encoding.

EC2 Instance Types (Contd.)

EC2 Instance types can be classified as:

- General Purpose
- Compute Optimized
- Memory Optimized
- Graphics Processing Unit (GPU)
- Storage Optimized
- EBS Optimized

FAMILY	Cost	Use
R3	Medium-High	Memory-intensive databases, web servers, application servers
R4	Medium-High	Offer better price per GB of RAM than R3
X1	High	X1 instances are optimized for large-scale, enterprise-class, in-memory applications

Memory-optimized instances are recommended for high performance databases, distributed memory caches, in-memory analytics, genome assembly and analysis, larger deployments of SAP, Microsoft SharePoint, and other enterprise applications.

EC2 Instance Types (Contd.)

EC2 Instance types can be classified as:

- General Purpose
- Compute Optimized
- Memory Optimized
- Graphics Processing Unit (GPU)
- Storage Optimized
- EBS Optimized

FAMILY	Cost	Use
G2	Medium-High	Video encoding, Graphics, Machine Learning, and so on
P2	Medium-High	General-purpose GPU compute applications
F1	In preview as of Jan 2017	Customizable hardware acceleration with field programmable arrays (FPGAs)

Graphics Processing Unit (GPU) instances are used for 3D application streaming, machine learning, video encoding, and other server-side graphics or GPU compute workloads.

EC2 Instance Types (Contd.)

EC2 Instance types can be classified as:

- General Purpose
- Compute Optimized
- Memory Optimized
- Graphics Processing Unit (GPU)
- Storage Optimized
- EBS Optimized

FAMILY	Cost	Use
I2	Medium-Very High	NoSQL databases, Hadoop, Data Warehousing
H1	Medium-Very High	MapReduce, high throughput application
D2	Medium-Very High	File systems, Data Warehousing, Hadoop

Storage optimized instances are designed for workloads that require high, sequential read and write access to very large data sets on local storage.

High I/O Instances: NoSQL databases like Cassandra and MongoDB, scale out transactional databases, data warehousing, Hadoop, and cluster file systems. very high random I/O performance, and provide high IOPS at a low cost.

Dense-storage Instances: Massively Parallel Processing (MPP) data warehousing, MapReduce and Hadoop distributed computing, distributed file systems, network file systems, log or data-processing applications. deliver high disk throughput

EC2 Instance Types (Contd.)

EC2 Instance types can be classified as:

- General Purpose
- Compute Optimized
- Memory Optimized
- Graphics Processing Unit (GPU)
- Storage Optimized
- EBS Optimized

FAMILY	Cost	Use
EBS Optimized	Low: additional cost to regular EC2 price	Applications that demand high IOPS and throughput

EBS-optimized instances allow associated EC2 instances to fully use the IOPS provisioned on an EBS volume.
A low hourly fee is charged in addition to the existing instance price. This is available only for select instances.

Demo: EC2 Instance Types

In this demonstration you will learn about the EC2 instance types available for selection.

Knowledge Check

(c) Avantus Traning Pte Ltd

What instance type would you choose if you required high disk throughput?

1

General Purpose

G2

D2

R3

What instance type would you choose if you required high disk throughput?

1

General Purpose

G2

D2

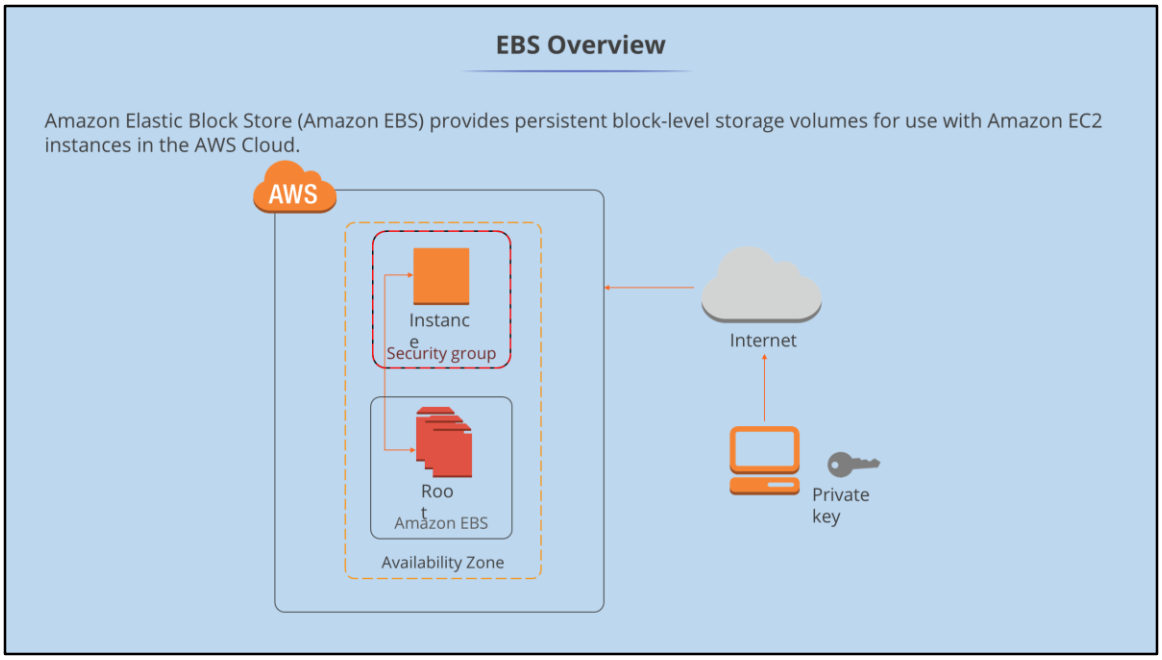
R3

c

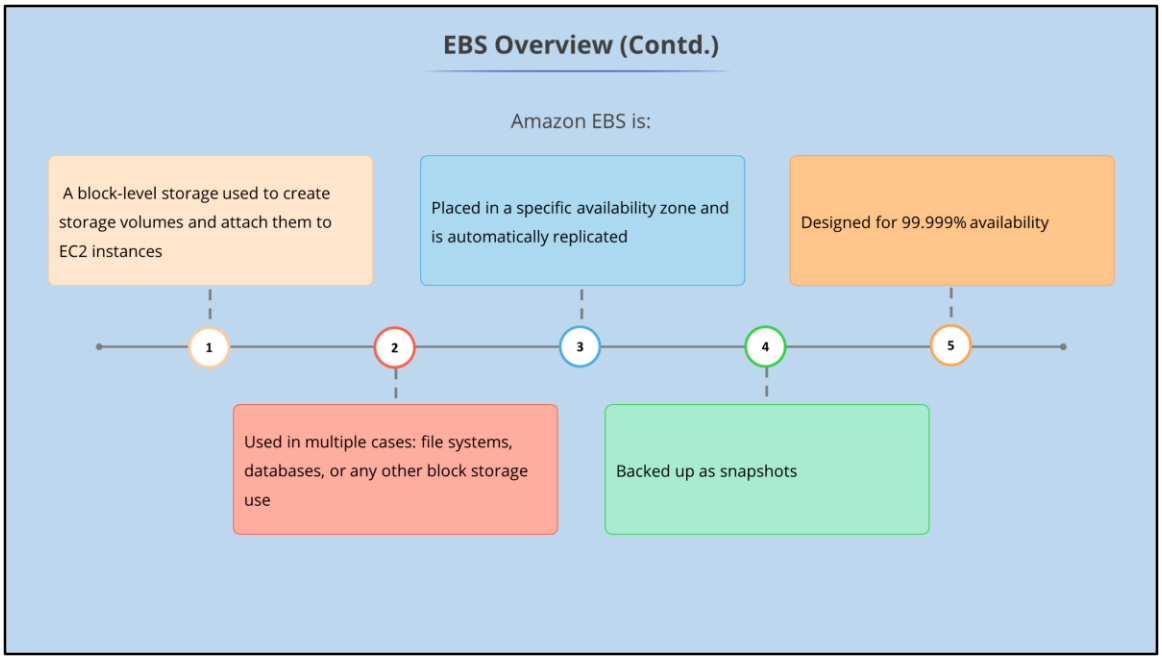
Dense Storage Instances are designed to deliver high disk throughput.

Amazon Elastic Block Store (EBS)

In this section, you'll learn what is an EBS, what are the different volume types available, and what are the differences between EBS and EFS.



Amazon Elastic Block Store (Amazon EBS) provides persistent block-level storage volumes for use with Amazon EC2 instances in the AWS Cloud. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability.



Amazon EBS is block storage that allows you to create storage volumes and attach them to Amazon EC2 instances. It can be used to create file systems, run databases, or you can use it in any other way you would use a block storage.
Amazon EBS volumes are placed in a specific Availability Zone and are automatically replicated.
All EBS volumes can have snapshots taken and are designed for 99.999% availability.

Storage Categories

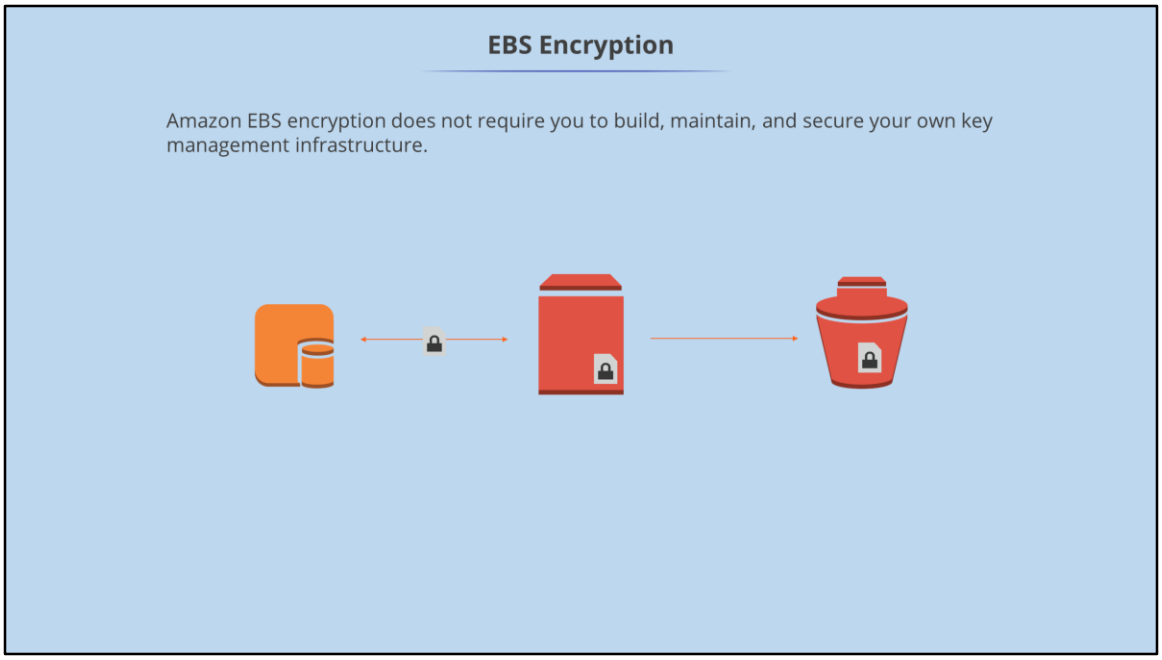
The two storage categories are:

	SSD-backed		HDD-backed		
Type	Provisioned IOPS SSD (io1)	General Purpose SSD (gp2)	Throughput Optimized HDD (st1)	Cold HDD (sc1)	Magnetic
Purpose	I/O intensive NoSQL and relational databases	Boot volumes, low-latency interactive applications, dev, test	Big data, data warehouses, log processing	Colder data requiring fewer scans per day	Infrequent access/ low performance requirements
Volume Size	4 GB – 16 TB	1 GB – 16 TB	500 GB – 16 TB	500 GB – 16 TB	1 GB - 1 TB
Max IOPS/Volume	20,000 (16 KB I/O size)	10,000 (16 KB I/O size)	500 (1 MB I/O size)	250 (1 MB I/O size)	40-200
Max Throughput/ Volume	320 MB/s	160 MB/s	500 MB/s	250 MB/s	90 MB/s
Price	Medium	Low-Medium	Low-Medium	Low	Lowest
Dominant Performance Attribute	IOPS	IOPS	MB/s	MB/s	N/A

Two storage categories are:

1. **SSD-backed storage** for transactional workloads such as databases and boot volumes (performance depends primarily on IOPS)
2. **New HDD-backed storage** for throughput intensive workloads such as MapReduce and log processing (performance depends primarily on MB/s)

EBS Magnetic volumes hard disk drives (HDDs), legacy storage, and are used for workloads with smaller datasets where data is accessed infrequently or when performance consistency is not of primary importance.



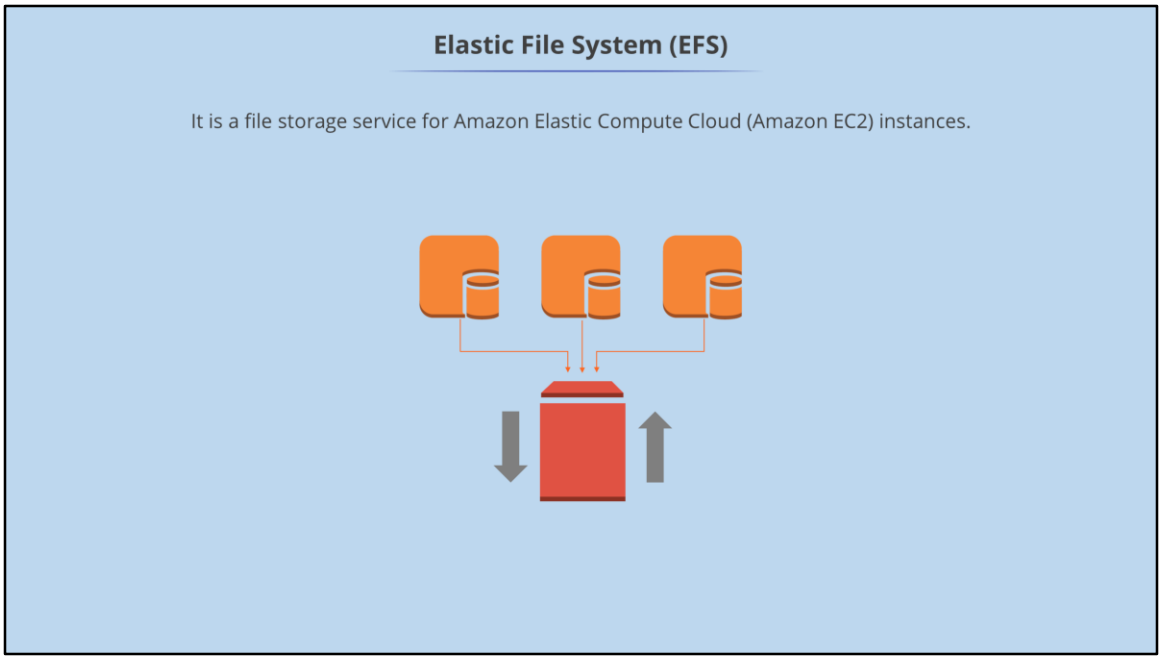
Amazon EBS encryption does not require you to build, maintain, and secure your own key management infrastructure. When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted:

- Data at rest inside the volume
- All data moving between the volume and the instance
- All snapshots created from the volume

You can expect the same IOPS performance on encrypted volumes as you would with unencrypted volumes.

The Amazon EBS encryption feature is also extended to snapshots of your

encrypted volumes. Snapshots taken from encrypted volumes are automatically encrypted.



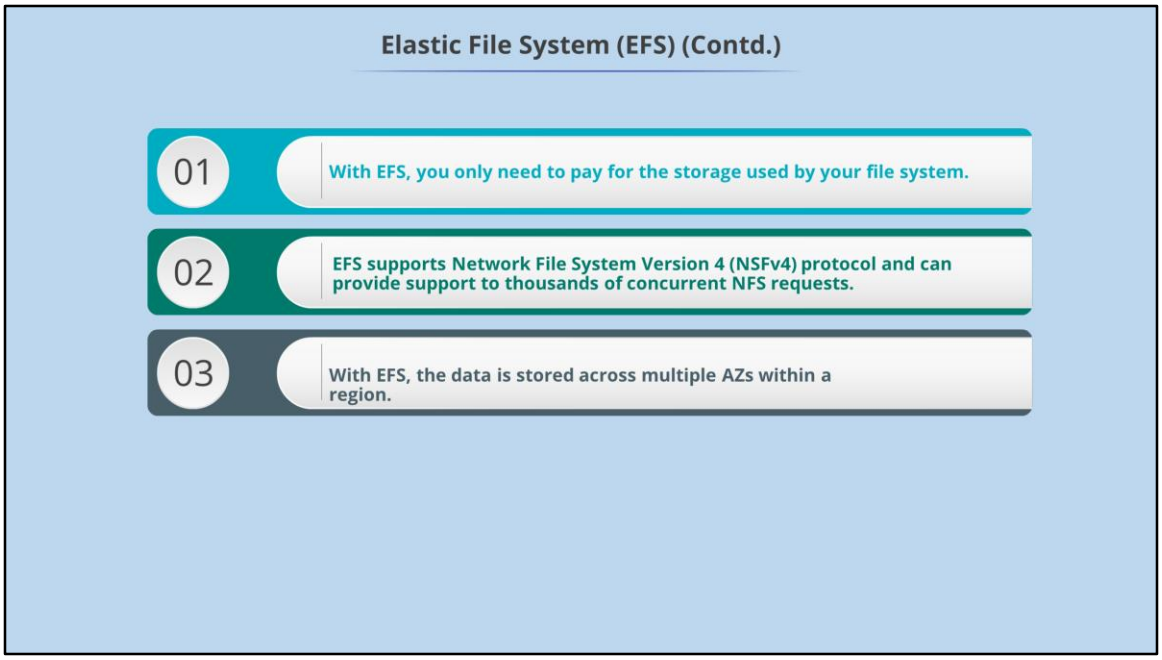
Amazon Elastic File System (Amazon EFS) is a file storage service for Amazon Elastic Compute Cloud (Amazon EC2) instances.

It is brand new, and currently, it is still in preview.

With Amazon EFS, storage capacity is elastic—growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it.

Multiple Amazon EC2 instances can access an Amazon EFS file system at the same time, providing a common data source for workloads and applications running on more than one instance.

With Amazon EFS, you pay only for the storage used by your file system.



Amazon Elastic File System (Amazon EFS) is a file storage service for Amazon Elastic Compute Cloud (Amazon EC2) instances.

It is brand new, and currently, it is still in preview.

With Amazon EFS, storage capacity is elastic—growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it.

Multiple Amazon EC2 instances can access an Amazon EFS file system at the same time, providing a common data source for workloads and applications running on more than one instance.

With Amazon EFS, you pay only for the storage used by your file system.

EFS vs EBS

Feature	EFS	EBS
Storage Size	No Limitations	Max 16TiB
File Size Limitation	Max file size 47.9TiB	No limitation
Data Throughput	Default throughput of 3GB	Variable depending on disk type
Data Access	Can be accessed concurrently	Limited to single EC2 instance
Operating System	Windows not supported	All operating systems
AZ Failure	Can survive one AZ failure	Cannot withstand AZ failure without snapshots

This table describes some of the key differences between EFS and EBS to help you identify when you would use each service.

EFS has no limitation on the size of a volume, whereas currently the largest volume you can create with EBS is 16 tebibytes.

Whereas EFS has a file size limitation of 47.9TiB but EBS does not.

The data throughput on EFS defaults to 3GB and EBS is variable depending on the type of disk type that you select.

The biggest different is that EFS is can be accessed concurrently by multiple users on different instances, but EBS can only be accessed from the instance it is attached to.

EFS is not currently supported on Windows and EFS.

EBS cannot withstand an AZ failure unless you have snapshots, whereas EFS can withstand one AZ failure.

Demo: Using EFS

In this demonstration you will learn to an EFS volume to EC2 instances.

Demo: Attaching EBS Volumes

In this demonstration you will learn to attach EBS volumes to EC2 instances.

Demo: EBS and RAID 0

In this demonstration you will learn to use EBS to create RAID 0 volumes.

Demo: EBS Volume resize

1

Select the correct EBS types available.

Fast SSD, Slow SSD, Fast HDD, Slow HDD, and Magnetic

Provisioned IOPS SSD, Reduced IOPS SSD, Throughput Optimized HDD, Throughput Reduced HDD, and Magnetic

Provisioned IOPS SSD, General Purpose SSD, Hot HDD, Cold HDD, and Magnetic

Provisioned IOPS SSD, General Purpose SSD, Throughput Optimized HDD, Cold HDD, and Magnetic

1

Select the correct EBS types available.

Fast SSD, Slow SSD, Fast HDD, Slow HDD, and Magnetic

Provisioned IOPS SSD, Reduced IOPS SSD, Throughput Optimized HDD, Throughput Reduced HDD, and Magnetic

Provisioned IOPS SSD, General Purpose SSD, Hot HDD, Cold HDD, and Magnetic

Provisioned IOPS SSD, General Purpose SSD, Throughput Optimized HDD, Cold HDD, and Magnetic

d

Provisioned IOPS SSD, General Purpose SSD, Throughput Optimized HDD, Cold HDD, and Magnetic are the types of storage available.

EBS Snapshots

In this section, you'll learn about EBS snapshots—how to take snapshots, and how to restore from snapshots.

Features of EBS Snapshots

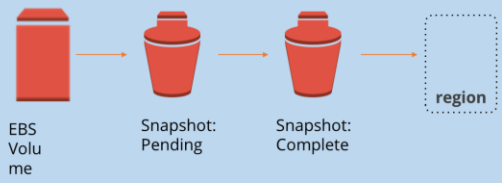
- Back up EBS volumes by taking point-in-time snapshots and storing them on Amazon S3.
- You can perform incremental backups, i.e., back up only the blocks on the device that have changed after the most recent snapshot are saved.
- When you delete a snapshot, only the data exclusive to that snapshot is removed.
- Active snapshots contain all of the information needed to restore your data (from the time the snapshot was taken) to a new EBS volume.
- If the EBS volume is encrypted, then the snapshot is also encrypted.
- Larger volumes can take up to 24 hours to copy to S3, but the snapshot is taken immediately.

Back up your EBS volumes by taking point-in-time snapshots and storing them on Amazon S3. Snapshots are incremental backups, which means only the blocks on the device that have changed after your most recent snapshot are saved.
When you delete a snapshot, only the data exclusive to that snapshot is removed.
Active snapshots contain all of the information needed to restore your data (from the time the snapshot was taken) to a new EBS volume.
If the EBS volume is encrypted, then the snapshot is also encrypted.
Larger volumes can take up to 24 hours to copy to S3, but the snapshot is taken immediately.

EBS Snapshot Storage

With Amazon EBS, you can:

- Create point-in-time snapshots of volumes that are stored for you in Amazon Simple Storage Service (Amazon S3)
- Copy a snapshot from one AWS region to another or within the same region.
- Encrypt your data with Amazon S3 server-side encryption (256-bit Advanced Encryption Standard)



With Amazon EBS, you can create point-in-time snapshots of volumes which are stored for you in Amazon Simple Storage Service (Amazon S3). After you've created a snapshot and it has finished copying to Amazon S3 (when the snapshot status is completed), you can copy it from one AWS region to another, or within the same region. Snapshots are copied with Amazon S3 server-side encryption (256-bit Advanced Encryption Standard) to encrypt your data and the snapshot copy receives a snapshot ID that's different from the ID of the original snapshot.

For larger volumes, it can take up to 24 hours to copy to S3, but the snapshot is taken immediately.

Demo: EBS Snapshots

In this demonstration you will learn to take an EBS snapshot.

1

Which of the following statements is NOT true?

Snapshots are full backups.

When you delete a snapshot, only the data exclusive to that snapshot is removed.

Active snapshots contain all of the information needed to restore your data.

You can copy snapshots between regions.

(c) Avantus Traning Pte Ltd

1

Which of the following statements is NOT true?

Snapshots are full backups.

When you delete a snapshot, only the data exclusive to that snapshot is removed.

Active snapshots contain all of the information needed to restore your data.

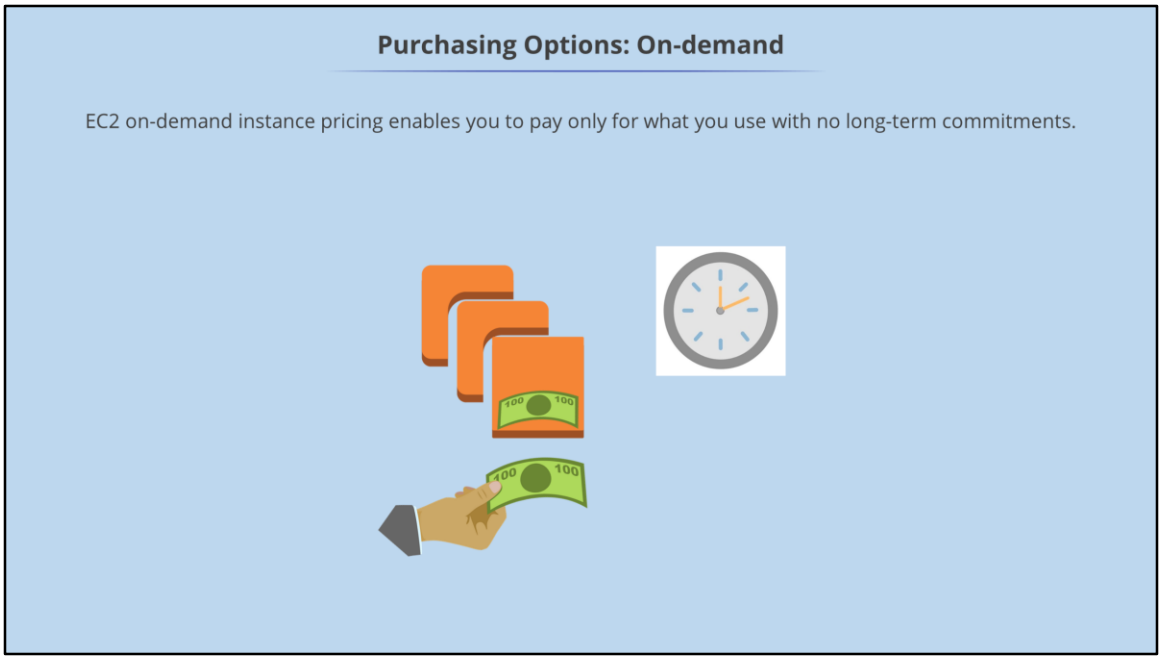
You can copy snapshots between regions.

a

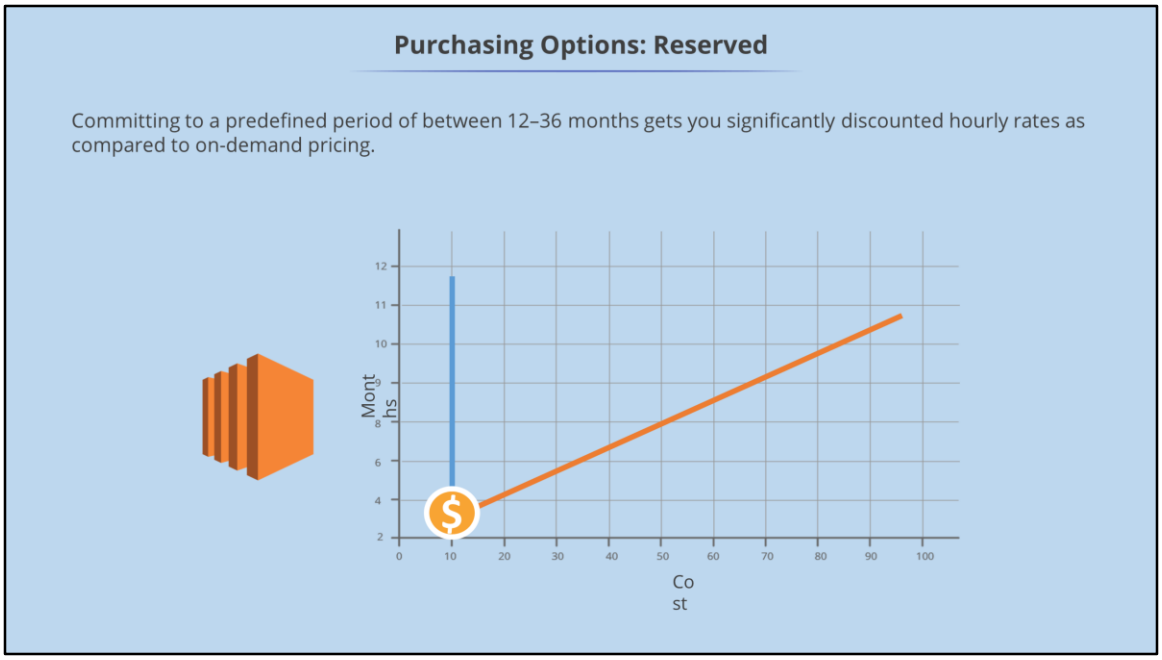
Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved.

EC2 Instance Pricing

In this section, you'll learn about the different EC2 pricing models: on-demand, spot, reserved, and dedicated.

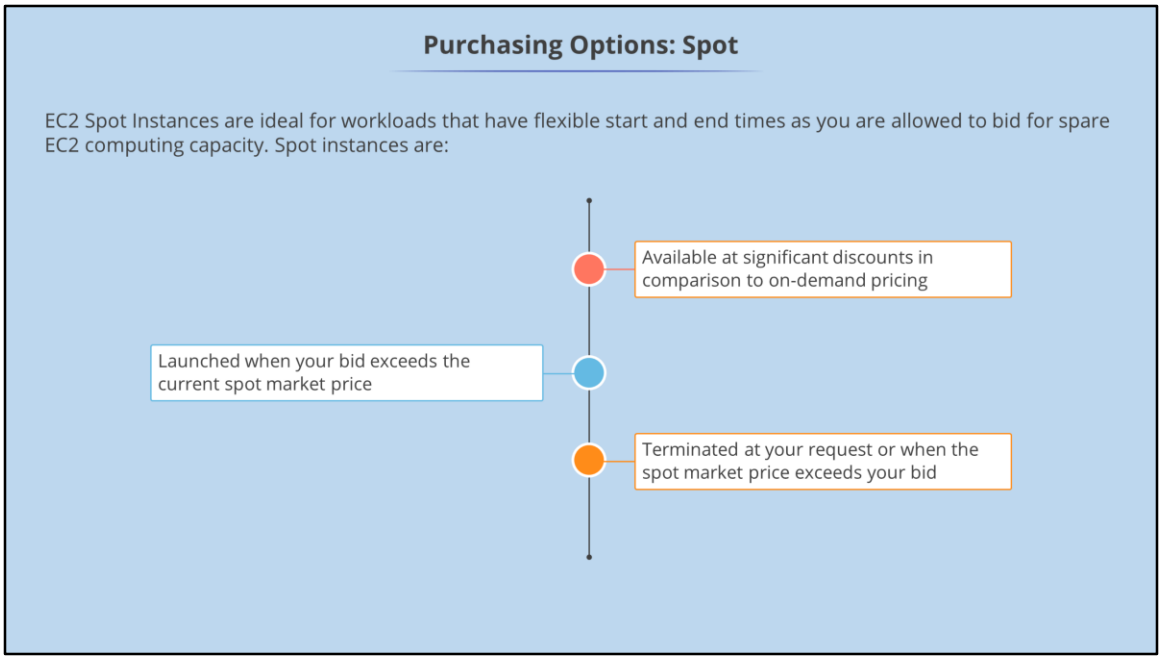


EC2 on-demand instance pricing enables you only pay for what you use with no long-term commitments.



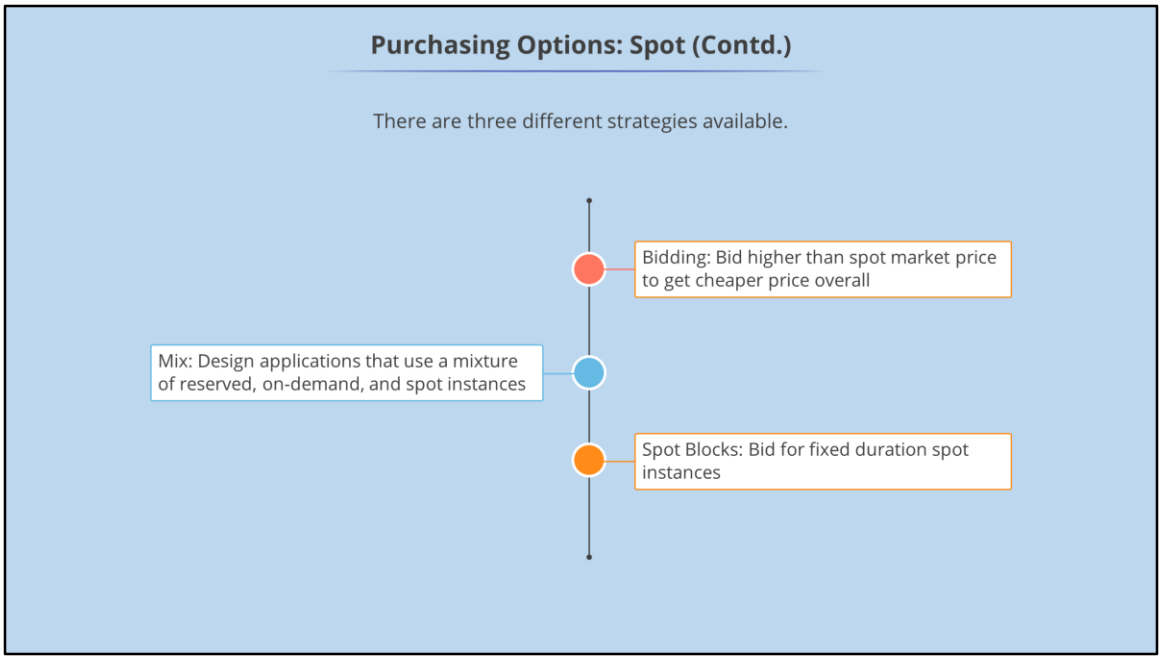
By committing to a predefined period of between 12–36 months, you can receive significantly discounted hourly rates as compared to on-demand pricing.

There is technically no difference between On-Demand EC2 instances and reserved instances. The only difference is the way you pay for it.



EC2 Spot Instances are ideal for workloads that have flexible start and end times as you are allowed to bid for spare EC2 computing capacity

Spot instances are often available at significant discounts in comparison to on-demand pricing. Your EC2 spot instance launch when your bid exceeds the current spot market price and will continue running until you terminate it, or the spot market price exceeds your bid. When the latter happens, your instance is terminated and you will not be charged for the partial hour that it ran.



Three different strategies are available:

Bidding–Bid higher than spot market price to get cheaper price overall

Mix—Design applications that use a mixture of reserved, on-demand, and spot instances

Spot Blocks—Bid for fixed duration spot instances

Purchasing Options: Dedicated

An Amazon EC2 Dedicated Host is a physical server with EC2 instance capacity fully dedicated to your use.

An Amazon EC2 Dedicated Host is a physical server with EC2 instance capacity fully dedicated to your use. Dedicated Hosts can help you address compliance requirements and reduce costs by allowing you to use your existing server-bound software licenses.

Demo: EC2 Spot Pricing

In this demonstration you will learn how to view EC2 Spot prices.

1

Which of the following are EC2 pricing models?

On-Request, Bid, Reserved, and Dedicated

On-Demand, Spot, Reserved, and Dedicated

On-Request, Auction, Reserved, and Constant

On-Demand, Bid, Reserved, and Constant

(c) Avantus Traning Pte Ltd

1

Which of the following are EC2 pricing models?

On-Request, Bid, Reserved, and Dedicated

On-Demand, Spot, Reserved, and Dedicated

On-Request, Auction, Reserved, and Constant

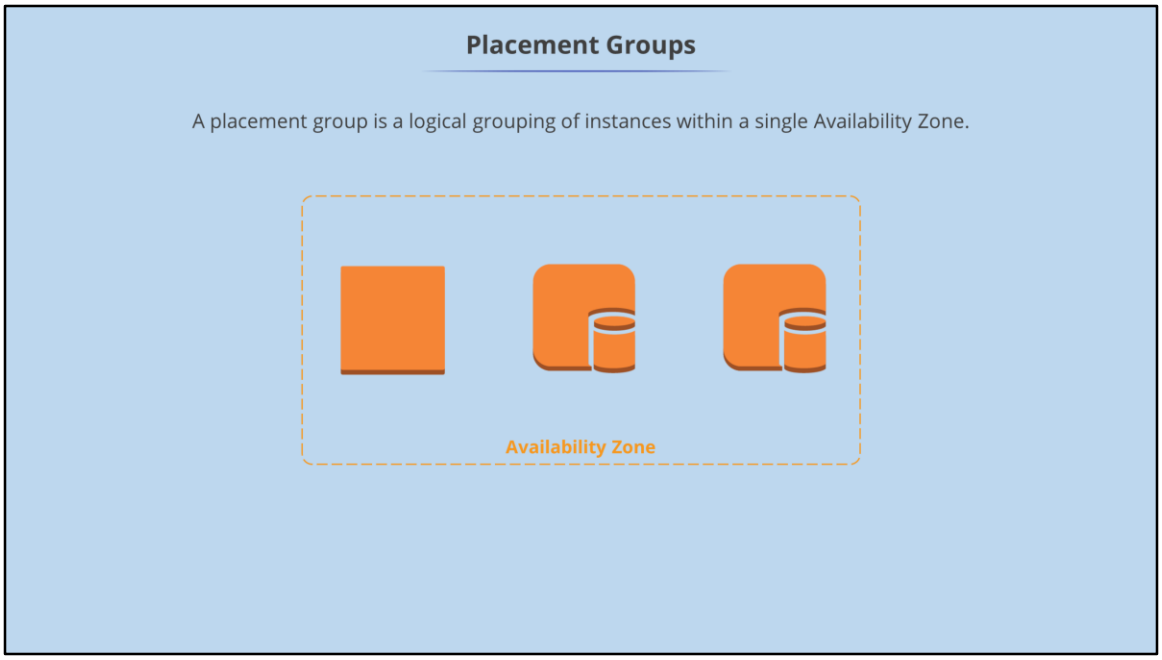
On-Demand, Bid, Reserved, and Constant

b

On-Demand, Spot, Reserved, and Dedicated are the EC2 pricing models.

Placement Groups

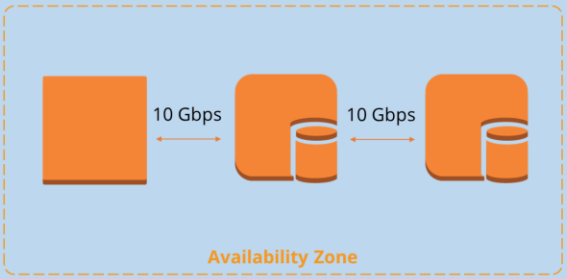
In this section, you'll learn about placement group, its uses, and the rules and restrictions of placement groups.



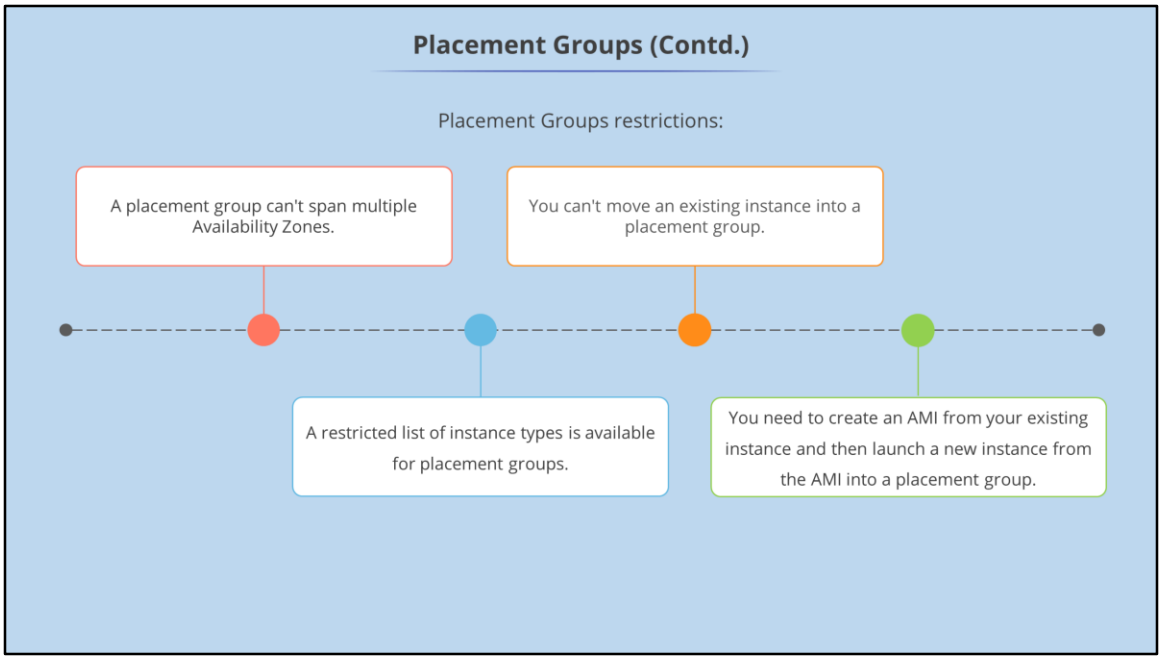
A *placement group* is a logical grouping of instances within a single Availability Zone.

Placement Groups (Contd.)

Placement groups are perfect for applications that require low network latency, high network throughput, or both. They allow 10 Gigabits per second (Gbps) networks.



Placement groups are perfect for applications that require low network latency, high network throughput, or both.
Placement groups allow 10 Gigabits per second (Gbps)-networks.



A placement group can't span multiple Availability Zones. A restricted list of instance types is available for placement groups. You can't move an existing instance into a placement group. You can create an AMI from your existing instance, and then launch a new instance from the AMI into a placement group.

1

Placement Groups are recommended for applications that benefit from:

low network latency, high network throughput, or both.

high network latency, high network throughput, or both.

low network latency, low network throughput, or both.

low network latency, zero network throughput, or both.

(c) Avantus Traning Pte Ltd

Placement Groups are recommended for applications that benefit from:

1

low network latency, high network throughput, or both.

high network latency, high network throughput, or both.

low network latency, low network throughput, or both.

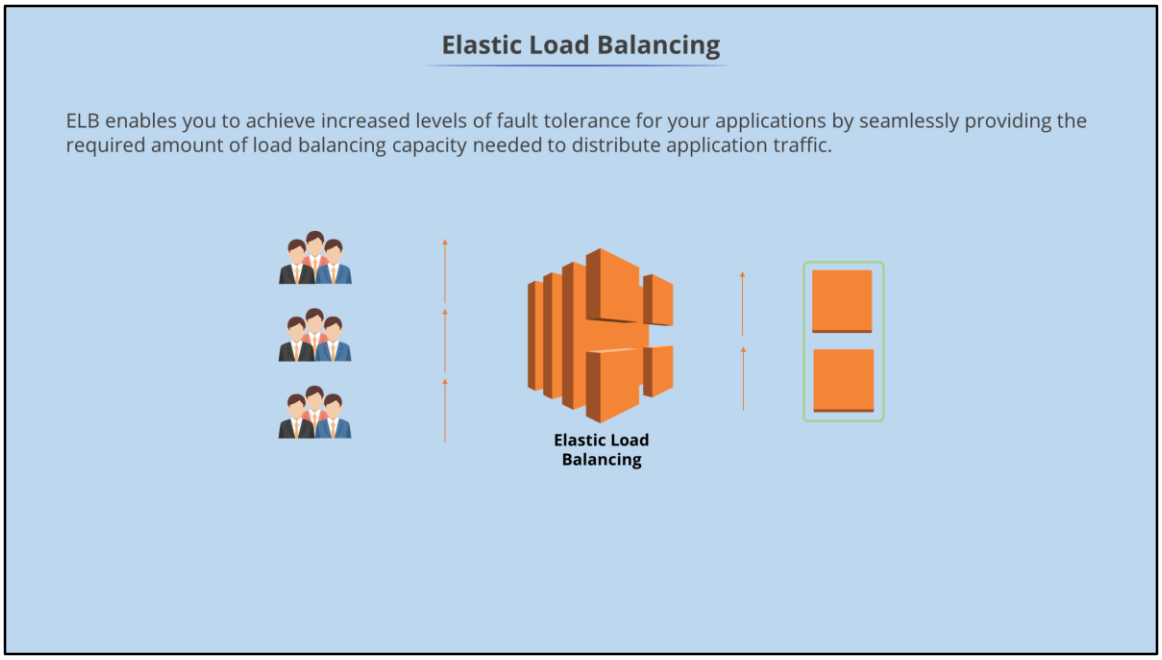
low network latency, zero network throughput, or both.

a

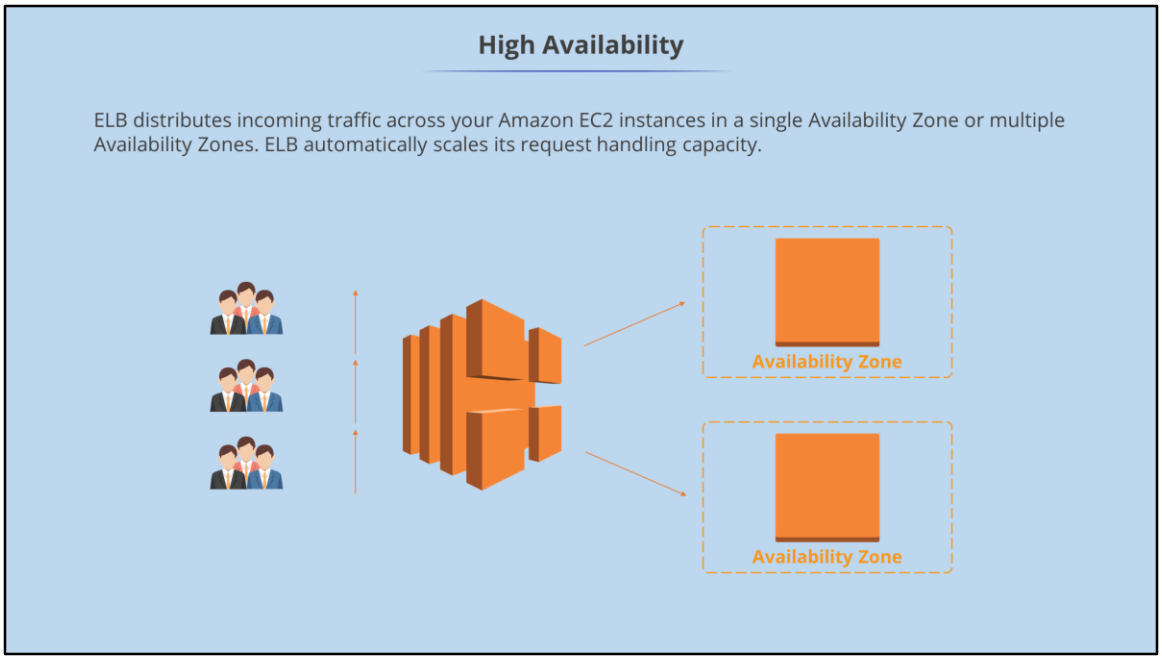
Placement Groups are recommended for applications that benefit from low network latency, high network throughput, or both.

Elastic Load Balancing (ELB)

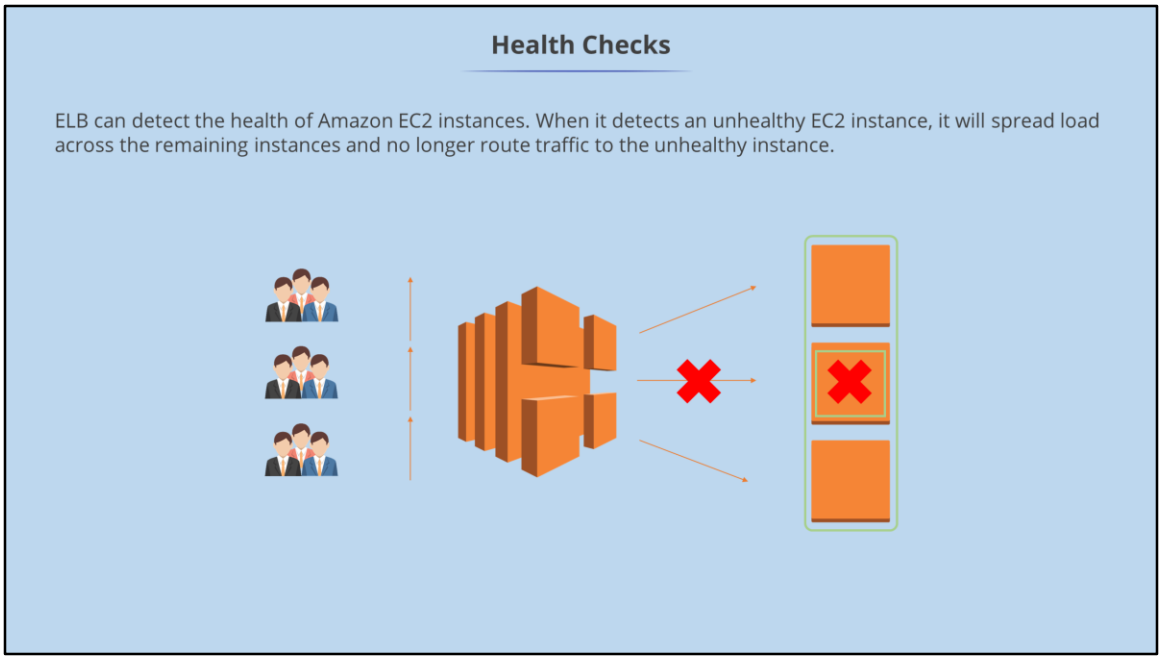
In this section, you'll learn about ELB and its uses.



Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances in the cloud. ELB enables you to achieve increased levels of fault tolerance for your applications by seamlessly providing the required amount of load balancing capacity needed to distribute application traffic.



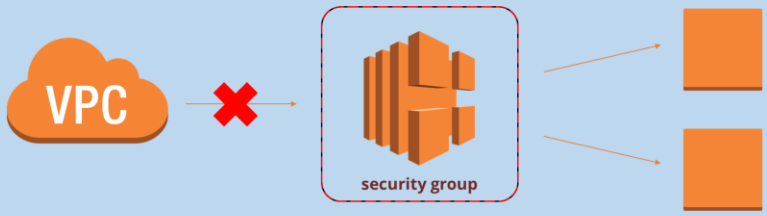
You can distribute incoming traffic across your Amazon EC2 instances in a single Availability Zone or multiple Availability Zones. Elastic Load Balancing automatically scales its request handling capacity in response to incoming application traffic.



Elastic Load Balancing can detect the health of Amazon EC2 instances. When it detects an unhealthy EC2 instance, it will spread load across the remaining instances and no longer route traffic to the unhealthy instance.

Security Features

ELB creates and manages security groups associated with it to provide additional networking and security options. You can also create a load balancer without public IP addresses to serve as an internal load balancer.



When using Virtual Private Cloud (VPC), you can create and manage security groups associated with Elastic Load Balancing to provide additional networking and security options. You can also create a load balancer without public IP addresses to serve as an internal (non-internet-facing) load balancer.

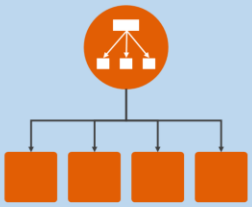
Sticky Sessions

ELB supports the ability to stick user sessions to specific EC2 instances using cookies. Traffic will be routed to the same instances as the user continues to access your application.

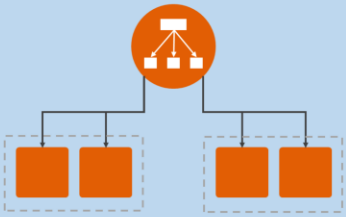
Elastic Load Balancing supports the ability to stick user sessions to specific EC2 instances using cookies. Traffic will be routed to the same instances as the user continues to access your application.

Types of Load Balancer

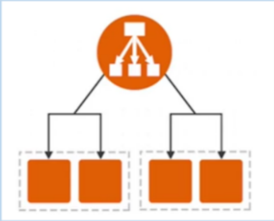
There are three types of load balancer:



Classic Load Balancer



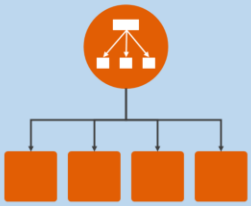
Application Load Balancer



Network Load Balancer

Classic Load Balancer

- The Classic load balancer is the original AWS load balancer.
- It operates at layer 4, which is the transport layer.

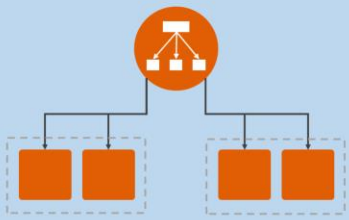


Classic Load Balancer

- The Classic load balancer is the original AWS load balancer.
- It operates at layer 4, which is the transport layer.

Application Load Balancer

- The Application load balancer is preferred for HTTP/HTTPS application.
- It operates at layer 7, which is the application layer.
- It can route requests to one or more ports on each EC2 container instance.

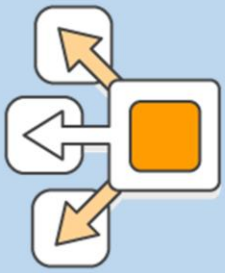


Application Load Balancer

- The Application load balancer is preferred for HTTP/HTTPS application.
- It operates at layer 7, which is the application layer.
- It can route requests to one or more ports on each EC2 container instances

Network Load Balancer

- Network load balancer is preferred for TCP.
- It can handle millions of requests per second and provide static/elastic IP addresses.



Preferred for TCP

Operates at Layer 3 – Network Layer

Choose a Network Load Balancer when you need ultra-high performance and static IP addresses for your application. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second while maintaining ultra-low latencies.

Network Load Balancer automatically provides a static IP per Availability Zone (subnet) that can be used by applications as the front-end IP of the load balancer. Can also provide Elastic IP for a fixed IP.

Choosing the Right Load Balancer

Feature	Application	Network	Classic
Protocol	HTTP, HTTPS	TCP	TCP, SSL, HTTP, HTTPS
Platform	VPC	VPC	EC2-Classic, VPC
Load Balancing to multiple ports on the same instance	Yes	Yes	No
IP addresses as targets	Yes	Yes	No
Path/Host-based routing	Yes	No	No
Static/Elastic IP address	No	Yes	No

This table shows the differences between the three load balancers that AWS offer. Going forward you should only consider the Classic Load balancer if you have an existing application running in the EC2-Classic network. If not then the decision is between the Application or Network load balancer and the difference is predominately TCP vs HTTP/S

Demo: Elastic Load Balancing

In this demonstration you will learn to configure Elastic Load Balancing.

Knowledge Check

(c) Avantus Traning Pte Ltd

1

Which of the following statements about ELB is NOT true?

You cannot distribute incoming traffic to instances to multiple Availability Zones.

Elastic Load Balancing automatically scales its request handling capacity in response to incoming application traffic.

Elastic Load Balancing can detect the health of Amazon EC2 instances.

Elastic Load Balancing supports the ability to stick user sessions to specific EC2 instances using cookies.

(c) Avantus Traning Pte Ltd

1

Which of the following statements about ELB is NOT true?

You cannot distribute incoming traffic to instances to multiple Availability Zones.

Elastic Load Balancing automatically scales its request handling capacity in response to incoming application traffic.

Elastic Load Balancing can detect the health of Amazon EC2 instances.

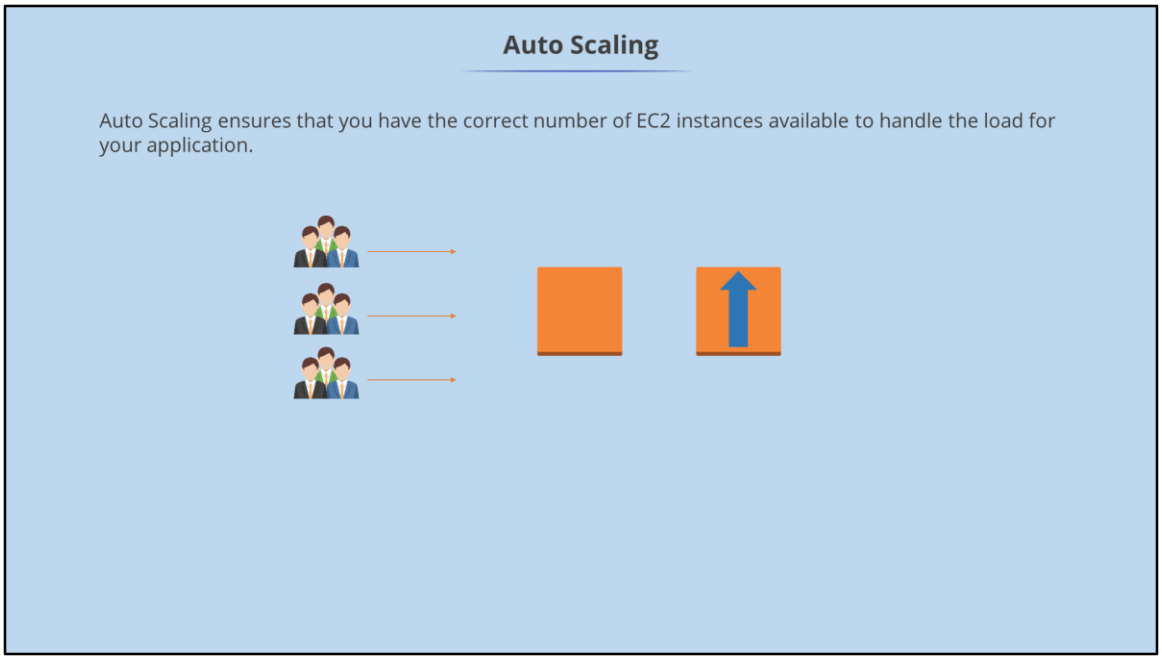
Elastic Load Balancing supports the ability to stick user sessions to specific EC2 instances using cookies.

a

You can distribute incoming traffic to instances to single Availability Zones and multiple Availability Zones.

Auto Scaling

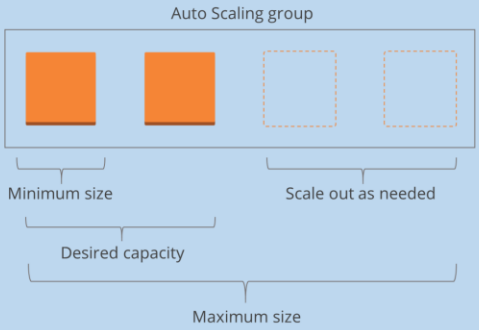
In this section, you'll learn Auto Scaling and its uses.



Auto Scaling helps you to ensure that you have the correct number of EC2 instances available to handle the load for your application.

Auto Scaling Groups

A collection of EC2 instances is called an Auto Scaling group. You can specify the minimum number of instances in each Auto Scaling group, and Auto Scaling ensures that your group never goes below this size.



You create collections of EC2 instances, called *Auto Scaling groups*. You can specify the minimum number of instances in each Auto Scaling group, and Auto Scaling ensures that your group never goes below this size. If you specify the desired capacity, either when you create the group or at any time later, Auto Scaling ensures that your group has these many instances.

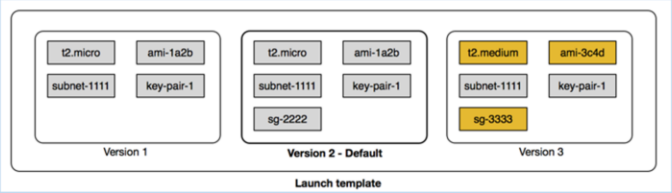
Auto Scaling Launch Configuration

A group uses a *launch configuration* as a template for its EC2 instances. In a launch configuration, specify information such as the AMI ID, instance type, key pair, security groups, and block device mapping for your instances.

Your group uses a *launch configuration* as a template for its EC2 instances. When you create a launch configuration, you can specify information such as the AMI ID, instance type, key pair, security groups, and block device mapping for your instances.

Launch Templates

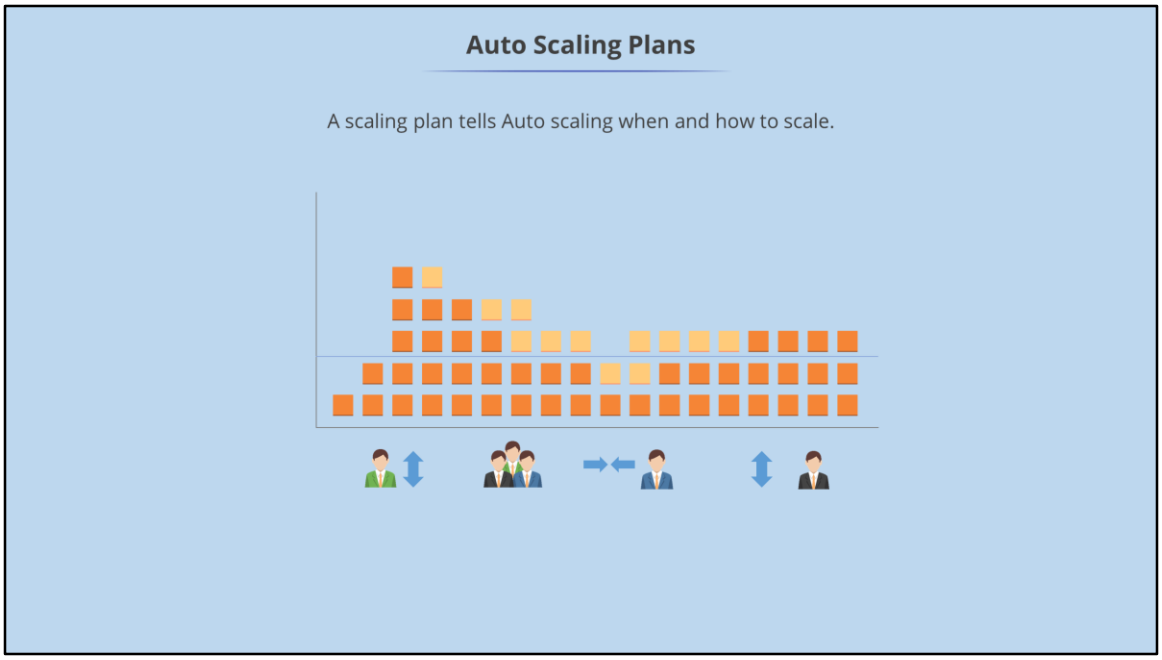
Launch Templates enables a new way to templatize your launch requests by reducing the number of steps required to create an instance.



Launch Templates enables a new way to templatize your launch requests rather than use the Launch instance wizard and can be used with auto scaling groups.

Launch Templates streamline and simplify the launch process for Auto Scaling, Spot Fleet, Spot, and On-Demand instances.
Launch Templates reduce the number of steps required to create an instance by capturing all launch parameters within one resource.

The following diagram shows a launch template with three versions. The first version specifies the instance type, AMI ID, subnet, and key pair to use to launch the instance. The second version is based on the first version and also specifies a security group for the instance. The third version uses different values for some of the parameters. Version 2 is set as the default version. If you launched an instance from this launch template, the launch parameters from version 2 would be used if no other version were specified.



A scaling plan tells Auto scaling when and how to scale. For example, you can base a scaling plan on the occurrence of specified conditions (dynamic scaling) or on a schedule.

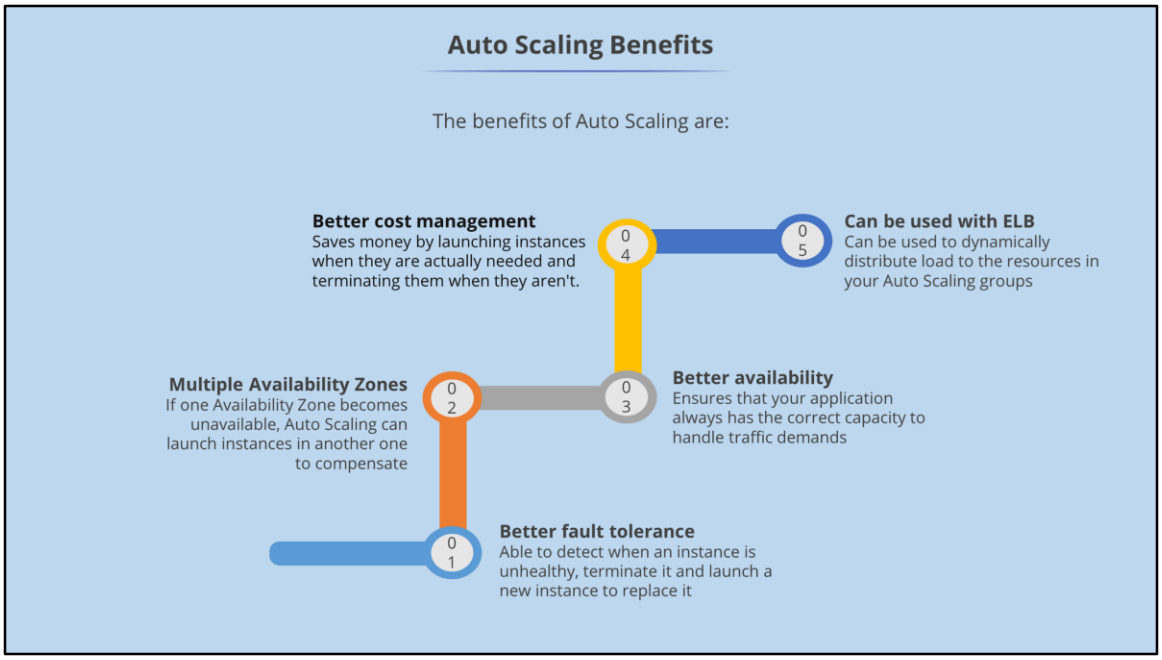
Auto Scaling Predictive Scaling

Enabled scaling based on data collected from your actual EC2 usage and further informed by billions of data points drawn from our AWS observations

- [x] **Enable predictive scaling**
  Support your scaling strategy by continually forecasting load and proactively scheduling capacity ahead of when you need it. Info
- [x] **Enable dynamic scaling**
  Support your scaling strategy by creating target tracking scaling policies to monitor your scaling metric and increase or decrease capacity as you need it. Info

Using data collected from your actual EC2 usage and further informed by billions of data points drawn from our own observations, we use well-trained Machine Learning models to predict your expected traffic (and EC2 usage) including daily and weekly patterns. The model needs at least one day's of historical data to start making predictions; it is re-evaluated every 24 hours to create a forecast for the next 48 hours.

Predictive scaling works by forecasting load and scheduling minimum capacity; dynamic scaling uses target tracking to adjust a designated CloudWatch metric to a specific target.



The benefits of Auto Scaling are the following:

- Better fault tolerance−Auto Scaling is able to detect when an instance is unhealthy. It will then terminate the instance and launch a new one to replace it. Auto Scaling can use multiple Availability Zones. If one Availability Zone becomes unavailable, Auto Scaling can launch instances in another one to compensate.
- Better availability−Auto Scaling will ensure that your application always has the correct capacity to handle traffic demands.
- Better cost management−Using auto scaling, you save money by launching instances when they are actually needed and terminating them when they aren't needed.
- ELB can be used to dynamically distribute load to the resources in

your Auto Scaling groups.

Demo: Auto Scaling

In this demonstration you will learn to configure Auto Scaling.

Knowledge Check

(c) Avantus Traning Pte Ltd

1

Why would you use Auto Scaling?

To improve your storage costs

To increase IOPS

To decrease network latency

To ensure that you have the correct number of EC2 instances available to handle the load for your application

(c) Avantus Traning Pte Ltd

Why would you use Auto Scaling?

1

To improve your storage costs

To increase IOPS

To decrease network latency

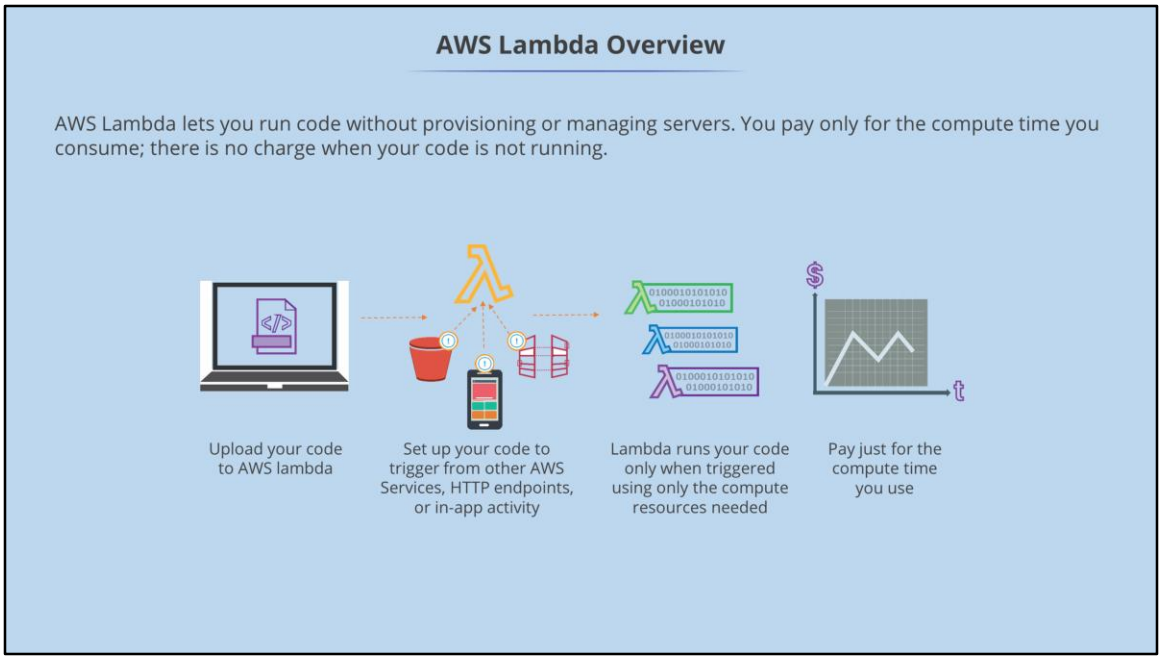
To ensure that you have the correct number of EC2 instances available to handle the load for your application

d

Auto Scaling ensures that you have the correct number of EC2 instances available to handle the load for your application using Auto Scaling groups, Launch Configuration, and Scaling Plans.

AWS Lambda

In this section, you'll learn about Lambda and its uses.



AWS Lambda lets you run code without provisioning or managing servers.
You pay only for the compute time you consume; there is no charge when your code is not running.
With Lambda, you can run code for virtually any type of application or backend service—all with zero administration.
Just upload your code and Lambda takes care of everything required to run and scale your code with high availability. You can set up your code to automatically trigger from other AWS services or call it directly from any web or mobile app.

Benefits

The benefits of AWS Lambda:

- It automatically runs your code without requiring you to provision or manage servers. Just write the code and upload it to Lambda.
- It automatically scales your application by running code in response to each trigger. Your code runs in parallel and processes each trigger individually, scaling precisely with the size of the workload.
- You are charged for every 100ms your code executes and the number of times your code is triggered. You don't pay anything when your code isn't running.

No Servers to Manage

Continuous Scaling

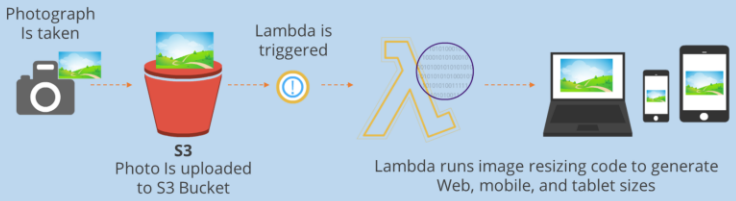
Sub-second Metering

AWS Lambda automatically runs your code, without requiring you to provision or manage servers. Just write the code and upload it to Lambda.
AWS Lambda automatically scales your application by running code in response to each trigger. Your code runs in parallel and processes each trigger individually, scaling precisely with the size of the workload.
With AWS Lambda, you are charged for every 100ms your code executes and the number of times your code is triggered. You don't pay anything when your code isn't running.

Case Study: The Seattle Times

The Seattle Times uses AWS Lambda to resize images for viewing on different devices such as desktop computers, tablets, and smartphones.

Example: Image Thumbnail Creation



The Seattle Times uses AWS Lambda to resize images for viewing on different devices such as desktop computers, tablets, and smartphones.

Triggering Lambda Functions

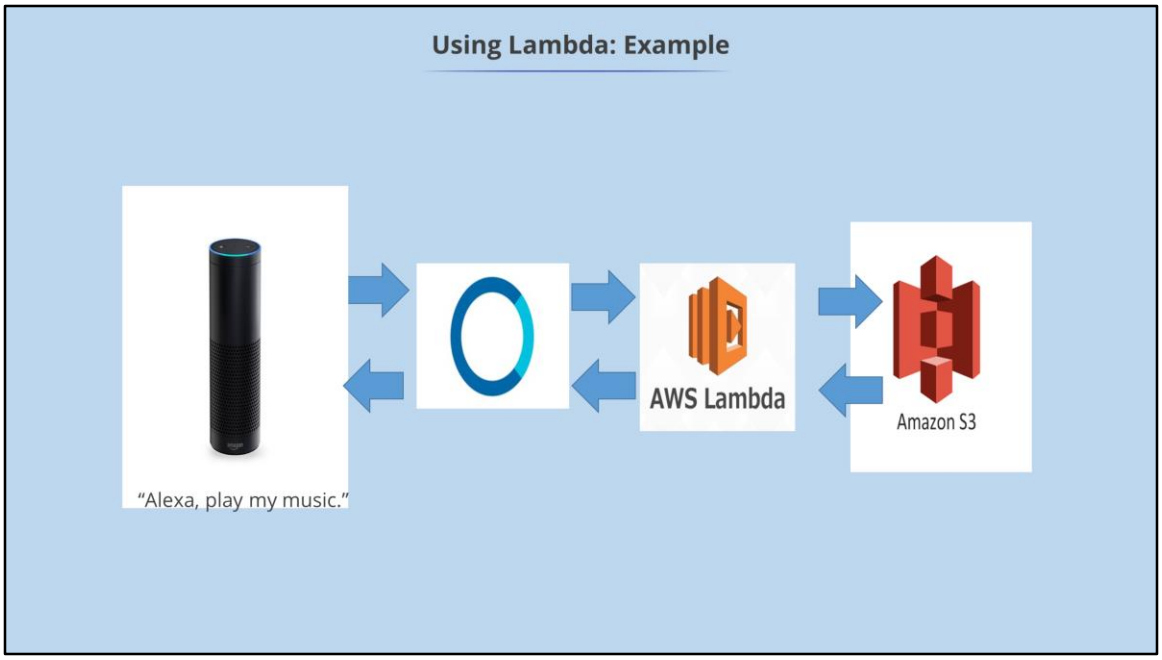
Lambda can be triggered by multiple AWS services such as:

- API Gateway
- AWS IoT
- Alexa Skills Kit
- CloudFront
- DynamoDB
- Lambda
- S3
- SNS

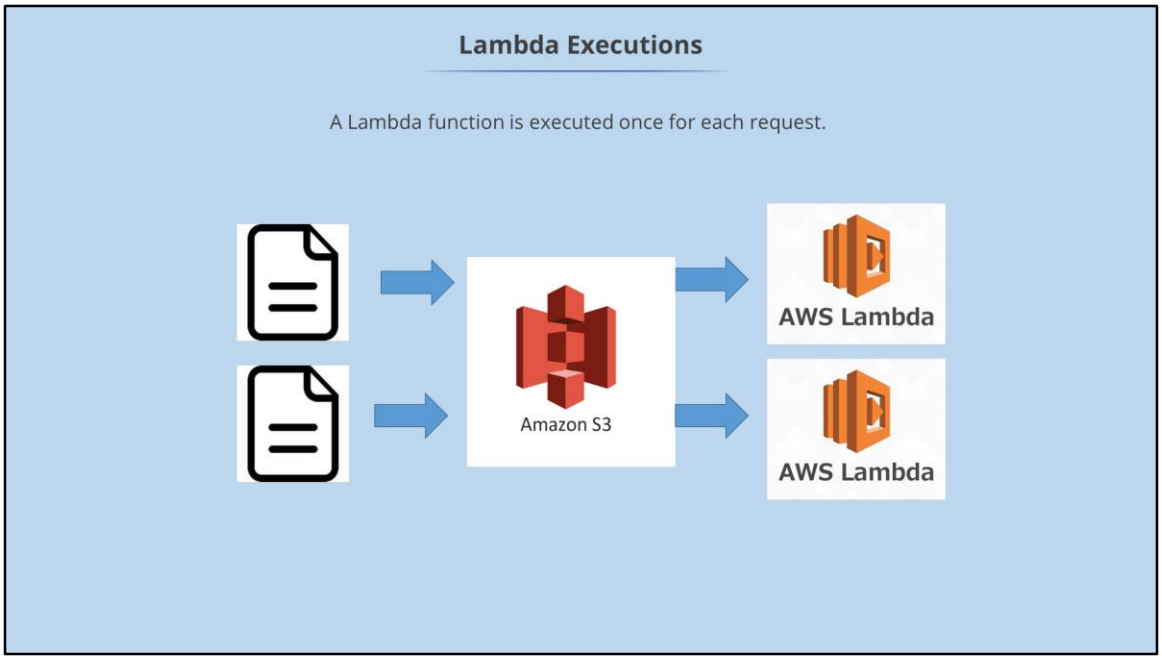
Lambda can be triggered by multiple AWS services, for example:

API Gateway
AWS IoT
CloudFront
DynamoDB
S3
SNS

Also, Lambda functions can trigger other lambda functions



An example of how to use Lambda is Alexa. When you ask Alexa something, it is triggering a Lambda function to go and do something to fulfil the request. If you asked Alexa to play your music, it could be designed to trigger a Lambda function via the Alexa Skills Kit, that grabs your MP3's off S3 storage and is then sent back to your Echo device.



A Lambda function is executed once per each request. If you have a function that fires whenever a new file is uploaded to S3, it will fire EACH time a file is uploaded. So if you upload 100 files, Lambda will fire 100 times as it automatic scales. Lambda can also be executed globally, for example to backup S3 buckets from one region to another.

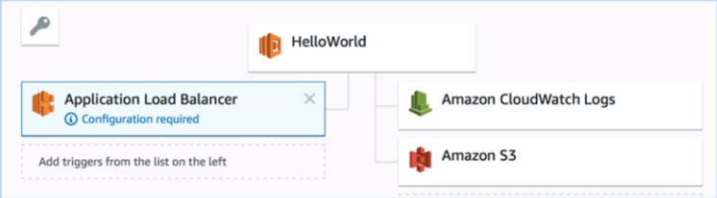
As of early 2019, Lambda supports the following languages:

Java, Go, PowerShell, **Node.js**, C#, **Python**, and **Ruby**

and provides a Runtime API which allows you to use any additional **programming languages** to author your functions.

ALB for Lambda

Application Load Balancers (ALBs) now support AWS Lambda functions as targets.



Application Load Balancers (ALBs) now support AWS Lambda functions as targets. Build websites and web applications as serverless code, using AWS Lambda to manage and run your functions, and then configure an ALB to provide a simple HTTP/S frontend for requests coming from web browsers and clients.

you can associate Lambda functions with ALBs on the Load Balancing section of the Amazon EC2 console or on the AWS Lambda console.

Demo: AWS Lambda

In this demonstration you will learn how to view EC2 Spot prices.

Knowledge Check

(c) Avantus Traning Pte Ltd

1

Which of the following is NOT a benefit of AWS Lambda?

No Servers to Manage

Continuous Scaling

Dedicated Hardware

Sub-second Metering

Which of the following is NOT a benefit of AWS Lambda?

1

No Servers to Manage

Continuous Scaling

Dedicated Hardware

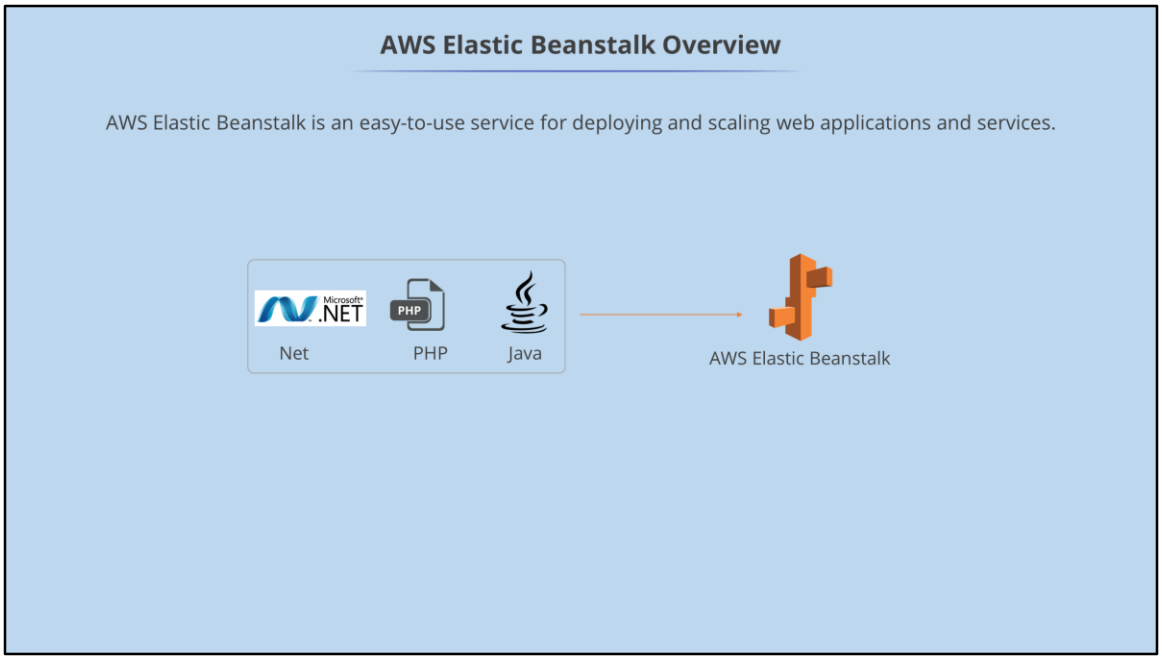
Sub-second Metering

c

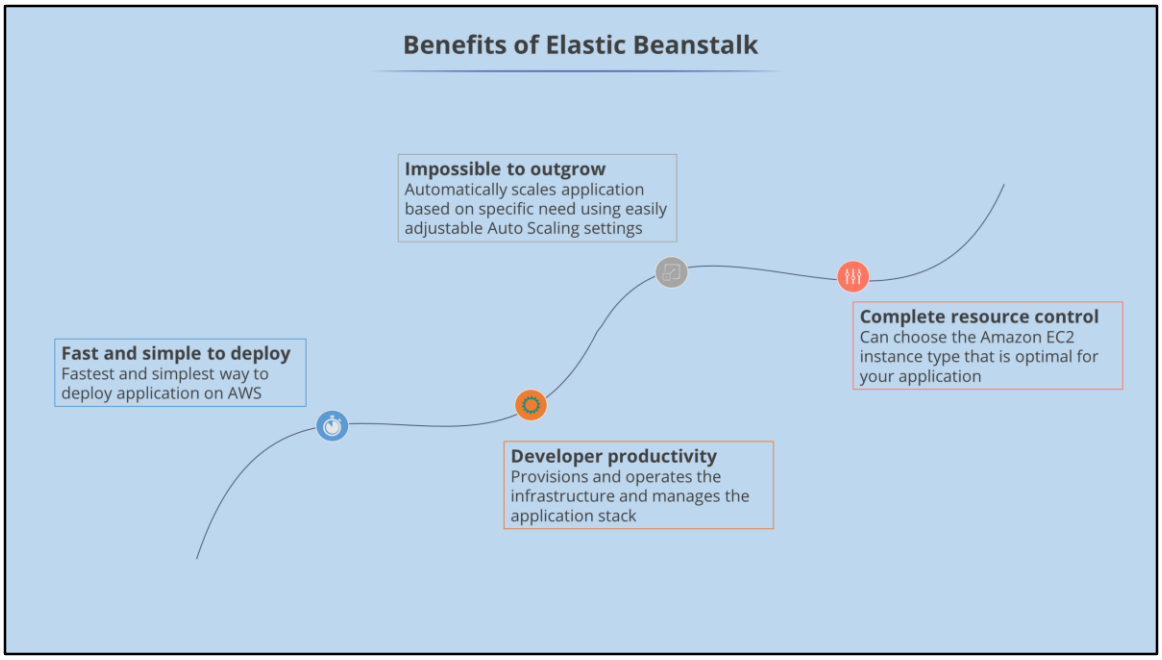
AWS Lambda automatically runs your code without requiring you to provision or manage servers. It automatically scales your application by running code in response to each trigger, and you are charged for every 100ms your code executes and the number of times your code is triggered.

AWS Elastic Beanstalk

In this section, you'll learn about Elastic Beanstalk and its uses.



AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services.
You can simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, and auto-scaling to application health monitoring.



Elastic Beanstalk is the fastest and simplest way to deploy your application on AWS. Within minutes, your application will be ready to use without any infrastructure or resource configuration work on your part.
One of the benefits of Elastic Beanstalk is Developer Productivity. It provisions and operates the infrastructure and manages the application stack for you, so you don't have to spend time on developing the expertise. It will also keep the underlying platform running your application up-to-date with the latest patches and updates.
Elastic Beanstalk automatically scales your application up and down based on your application's specific need using easily adjustable Auto Scaling settings.
Complete resource control is another advantage. You can choose the Amazon EC2 instance type that is optimal for your application.

Knowledge Check

1

What is the primary purpose of Elastic Beanstalk?

Load balancing of EC2 instances

Deploying and scaling web applications and services

Automatic storage management

Running autonomous EC2 instances

(c) Avantus Traning Pte Ltd

1

What is the primary purpose of Elastic Beanstalk?

Load balancing of EC2 instances

Deploying and scaling web applications and services

Automatic storage management

Running autonomous EC2 instances

b

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services. You just have to upload your code. Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, and auto-scaling to application health monitoring.

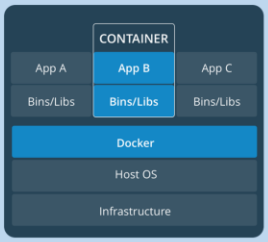
Amazon Elastic Container Service (ECS)

In this section, you'll learn Amazon Elastic Container Services also known as ECS

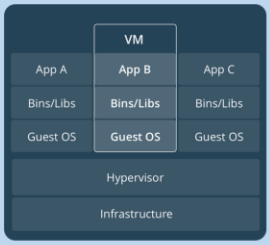
The main building block of docker is a container, which is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings.

So you could have a container that runs a website and your container would hold php, apache and mysql along with all the associated settings.

Containers vs. Virtual Machines



- Containers remove the OS requirements and instead packages just code and dependencies together.
- Multiple containers can run on the same machine and share the OS kernel with other containers, and all the resources running as a machine
- Extremely portable
- Can start almost instantly
- 10MBs+ in size



- VMs turn one server into many servers
- Each VM includes a full copy of an operating system, one or more applications, all necessary binaries, and libraries
- Can be slow to boot up
- 10GBs+ in size

- VMs turn one server into many servers
- Each VM includes a full copy of an operating system, one or more applications, all necessary binaries and libraries
- Can be slow to boot up
- 10GBs+ in size

- Containers remove the OS requirements and instead packages just code and dependencies together.
- Multiple containers can run on the same machine and share the OS kernel with other containers.
- with all resources running as a machine
- Extremely portable
- Can start almost instantly
- 10MBs+ in size

Docker Overview

Docker is an open-source container platform for automating the deployment of applications as portable, self-sufficient containers.

- Highly scalable
- Can run on Windows or Linux hosts
- Allows you to deploy into any environment knowing that your code will run
- Packages software into standardized units called containers, which include the application code, libraries, runtimes, config and dependencies

Docker is an open-source container platform for automating the deployment of applications as portable, self-sufficient containers.

- Highly scalable
- Can run on Windows or Linux hosts
- Allows you to deploy into any environment knowing that your code will run
- Packages software into standardized units called containers, which include the application code, libraries, runtimes, config and dependencies

Docker Components

Docker Image
- A template with instructions to create a docker container.

Docker Container
- Created from a docker image. Holds everything that an application needs to run.

Union File Systems
- File systems that operate by creating layers.

Docker File
- Used to build your own image. Stores instructions to run command, add a file, etc. Each instruction in a Docker File creates a layer in the image.

Docker Daemon/Engine
- Communicates with Docker client and manages Docker objects.

Docker Client
- Interface between you and Docker Engine.

Docker Registries
- Stores docker images. Can be public or private.

Docker Image
– A template with instructions to create a docker container.

Docker Container
– Created from a docker image. Holds everything that an application needs to run.

Union File Systems
– File systems that operate by creating layers.

DockerFile
– Used to build your own image. Stores instructions to run command, add a file, etc. Each instruction in a DockerFile creates a layer in the image.

Docker Daemon/Engine
– Communicates with Docker client and manages Docker objects.

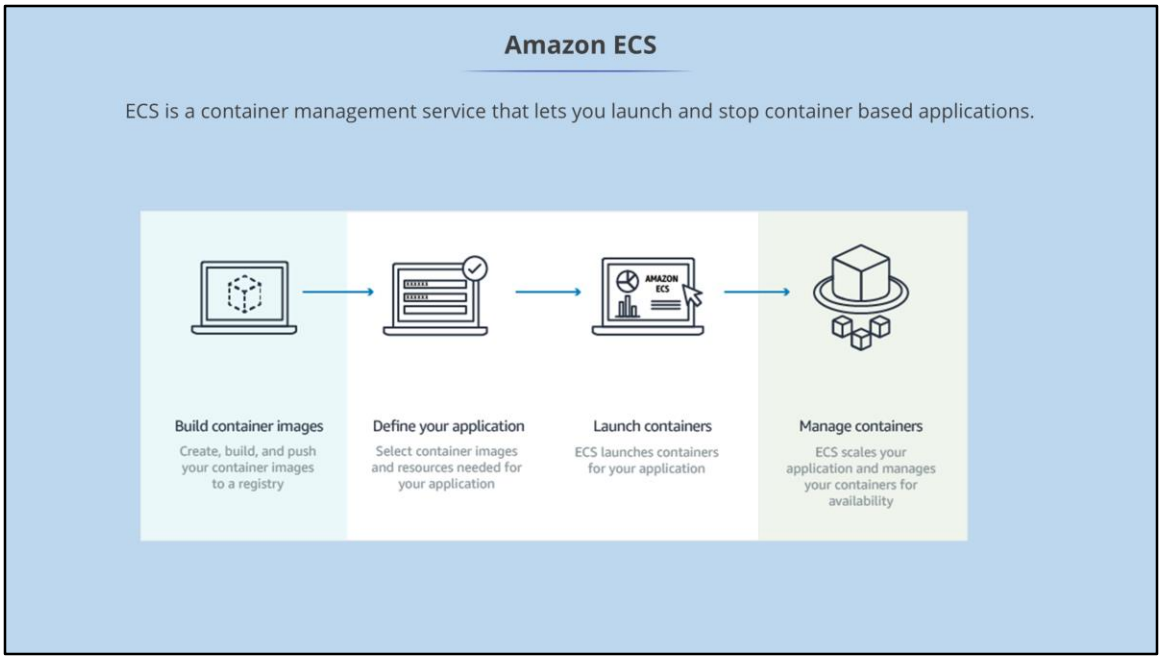
Docker Client
– Interface between you and Docker Engine.

Docker Registries
– Stores docker images. Can be public or private.

Docker image – a bit like an AMI, but only includes the files, no OS.

If the image is a recipe, the container is a cake. You can make as many cakes as you like with a given recipe.

Docker images are RO templates from which containers are launched. Rather than replace the entire image, you just update layers.



ECS is a container management service that lets you launch and stop container based applications.

Amazon's managed service of Docker.

No need to manage your own cluster and configuration management systems

ECS allows you to run and maintain a specified number/count of instances of a task definition. Auto Scaling for containers.

ECS takes the pain of managing infrastructure away from you, no need to provision a docker registry and docker images and then load docker onto EC2 instances. Provides all the scaling, scheduling, recovery, etc.

Amazon ECS Components

Amazon EC2 Container Registry (Amazon ECR)
- A repository for your containers. Amazon's version of DockerHub.

Task Definitions
- JSON text files that describe one or more containers that form your applications.

ECS clusters
- Logical grouping of container instances that you can place tasks on.

Amazon EC2 Container Registry (Amazon ECR)
- A repository for your containers. Amazon's version of DockerHub.

Task Definitions
– JSON text files that describe one or more containers that form your applications. Everything from which ports to use, commands to run, data volumes to access, how many CPUs, how much memory, etc

ECS clusters
– logical grouping of container instances that you can place tasks on. Can have PRD and DEV clusters. logical group of container instances, can be mixed types, one cluster at one time

Amazon EKS

- Kubernetes is a docker orchestration tool that can be used to create a more manageable, resilient application infrastructure.
- It is mostly used for large, complex environments.
- Amazon EKS is a managed version of Kubernetes that simplifies deployment onto the AWS cloud.

kubernetes

Amazon EKS

Kubernetes is a docker orchestration tool that can be used to create a more manageable, resilient application infrastructure. Can be complicated.

Especially useful for large, complex environments.

Amazon EKS is a managed version of Kubernetes that simplifies deployment on the AWS cloud.

AWS Fargate is a technology for Amazon ECS and EKS that allows you to run containers without having to manage servers or clusters.

Fargate takes all of that effort away from your and streamlines deployments.

Allows you to stop considering which instance types to use.

Removes need to determine the placement of resources, scheduling, scaling and patching.

Fargate launch type and EC2 launch type. With Fargate launch type, all you have to do is package your application in containers, specify the CPU and memory requirements, define networking and IAM policies, and launch the application.

No clusters to manage. Seamless scaling. Integrated with ECS and EKS.

Knowledge Check

(c) Avantus Traning Pte Ltd

1

What does not need to be included in a container?

Application code

Libraries

Operating System

Config

1

What does not need to be included in a container?

Application code

Libraries

Operating System

Config

c

A container does not have it's own kernel, which makes it much more lightweight and portable when compared to VMs.

AWS Command Line Interface (CLI)

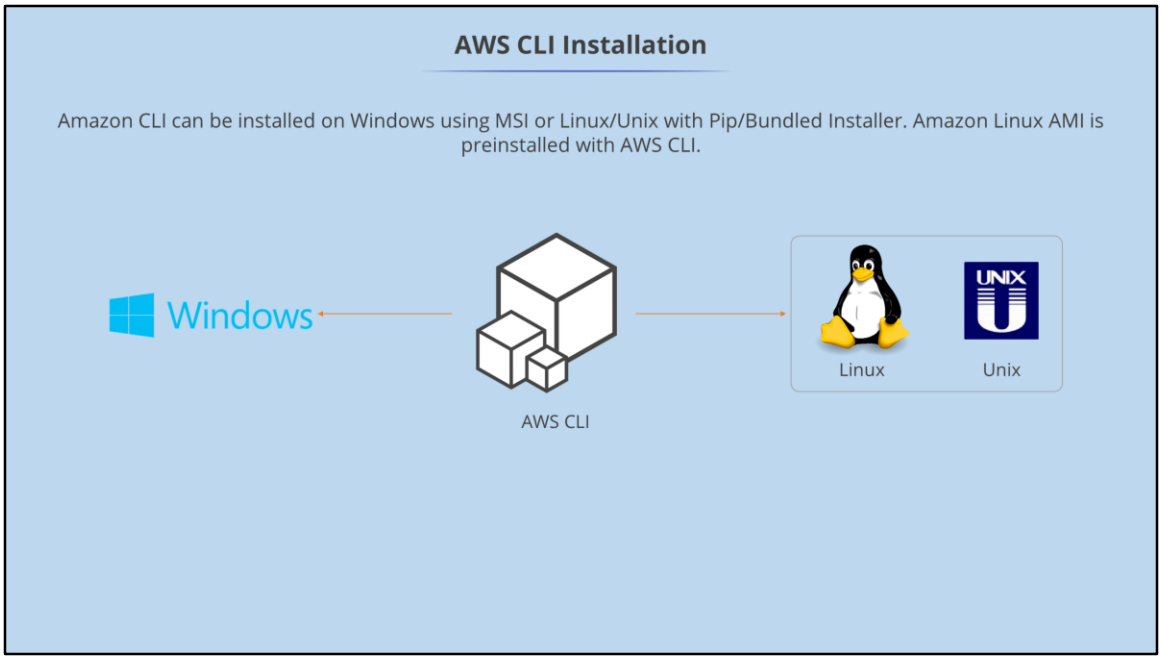
In this section, you'll learn about AWS CLI, its uses, and examples.

AWS CLI

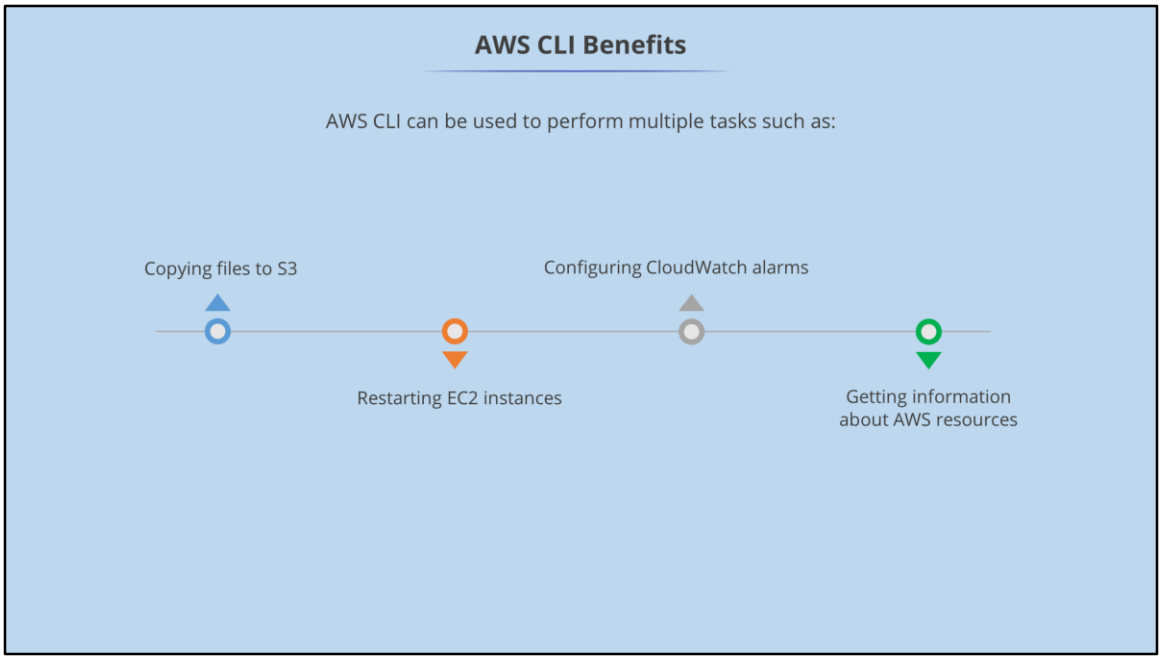
Amazon's definition of an AWS CLI:
"The AWS Command Line Interface (CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts."

AWS CLI

The AWS Command Line Interface (CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line, and automate them through scripts.

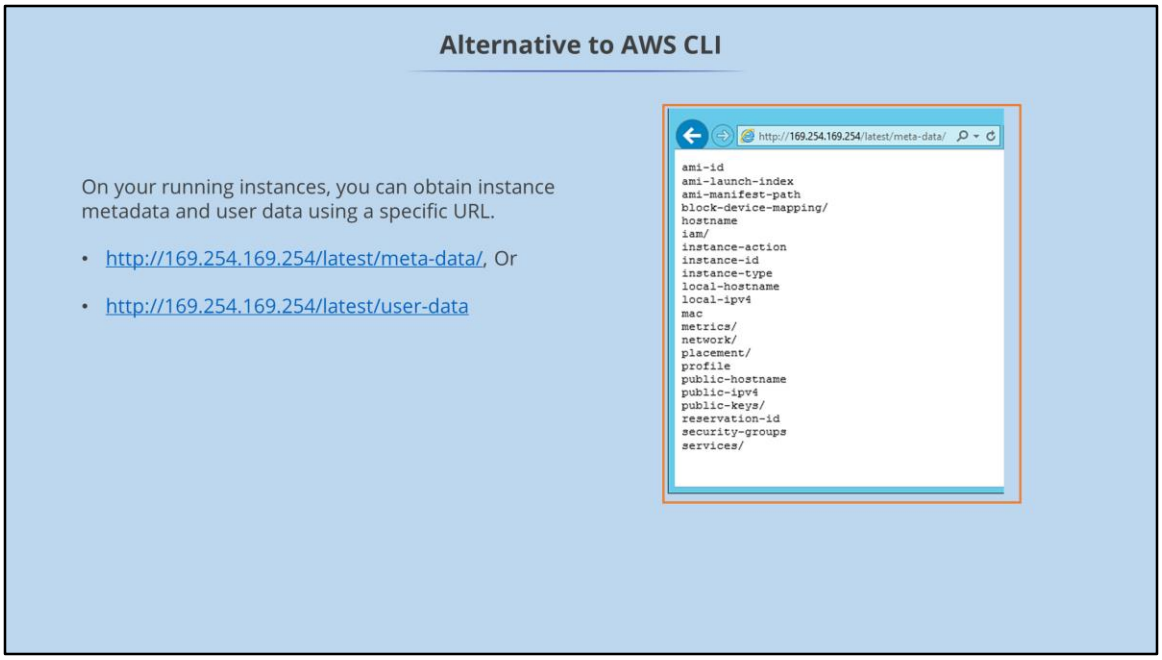


Amazon CLI can be installed on Windows using MSI, or Linux/Unix with Pip/Bundled Installer. Amazon Linux AMI is preinstalled with AWS CLI.



AWS CLI can be used to perform multiple tasks, such as:

- Copying files to S3
- Restarting EC2 instances
- Configuring CloudWatch alarms
- Getting information about AWS resources



Slide 117: Alternative to AWS CLI

On your running instances you can obtain instance metadata and user data using a specific URL.

http://169.254.169.254/latest/meta-data/

Or

http://169.254.169.254/latest/user-data

Demo: AWS CLI

Demonstrate how to use AWS CLI.

Knowledge Check

(c) Avantus Traning Pte Ltd

1

What is the key purpose of AWS CLI?

To manage multiple AWS resources from the command line

To completely replace the need to use the AWS Management Console

To lower your AWS resource costs

To improve your latency between AWS Regions

(c) Avantus Traning Pte Ltd

1

What is the key purpose of AWS CLI?

To manage multiple AWS resources from the command line

To completely replace the need to use the AWS Management Console

To lower your AWS resource costs

To improve your latency between AWS Regions

a

AWS CLI allows you to manage multiple AWS resources, such as EC2, S3, etc., from the command line.

2

Which URL do you use to get metadata or user data from a running instance?

http://0.0.0.0/

http://169.254.169.254/

http://254.169.254.169/

http://10.0.0.0/

(c) Avantus Traning Pte Ltd

2

Which URL do you use to get metadata or user data from a running instance?

http://0.0.0.0/

http://169.254.169.254/

http://254.169.254.169/

http://10.0.0.0/

b

You can use http://169.254.169.254/latest/meta-data/ or http://169.254.169.254/latest/user-data/.

AWS EC2 Best Practices

In this section, you'll learn the best practices on:

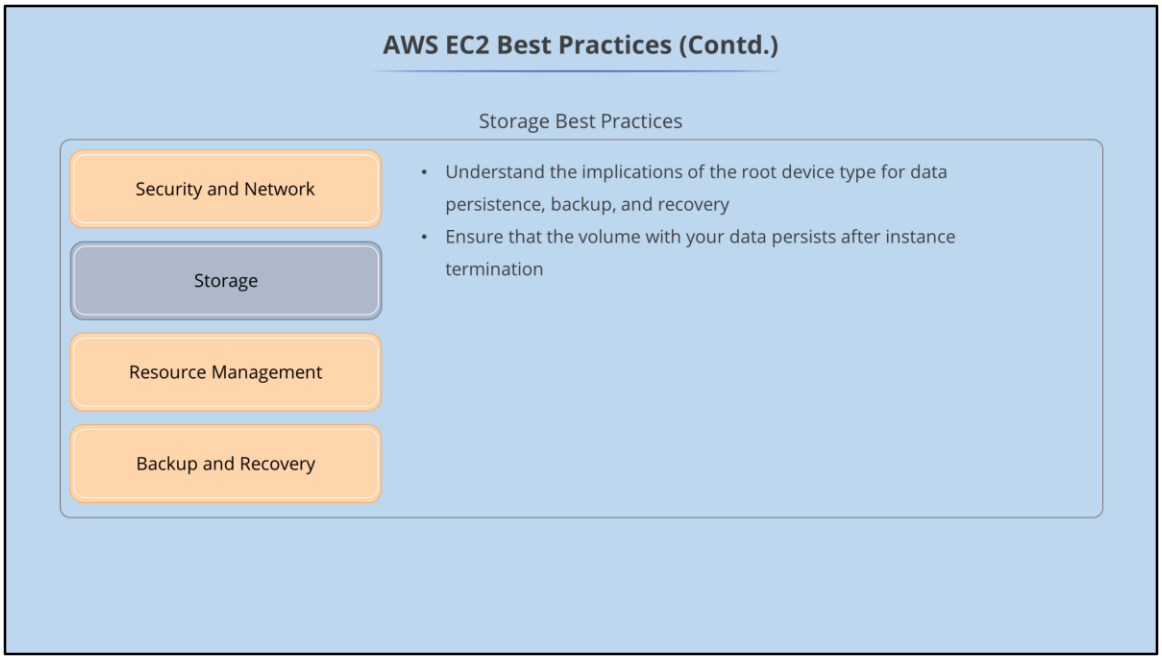
- Security and Network
- Storage
- Resource Management
- Backup and Recovery

Security and Network

Manage access to AWS resources and APIs using identity federation, IAM users, and IAM roles. Establish credential management policies and procedures for creating, distributing, rotating, and revoking AWS access credentials.

Implement the least permissive rules for your security group.

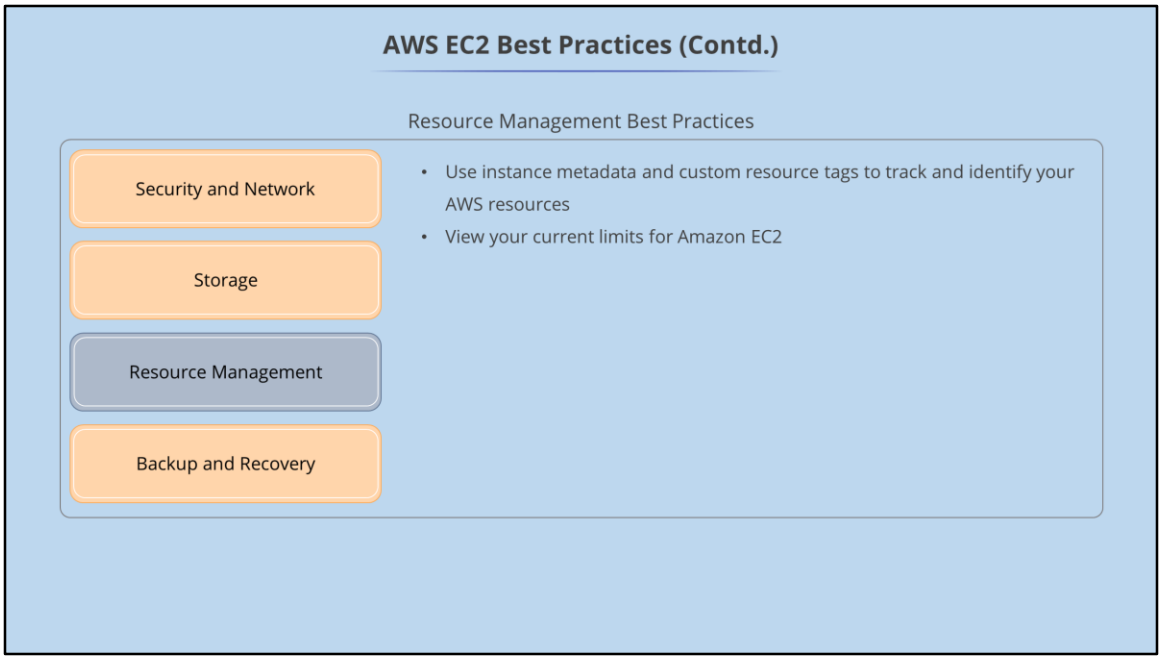
Regularly patch, update, and secure the operating system and applications on your instance.



Storage

Understand the implications of the root device type for data persistence, backup, and recovery. Use separate Amazon EBS volumes for the operating system versus your data.

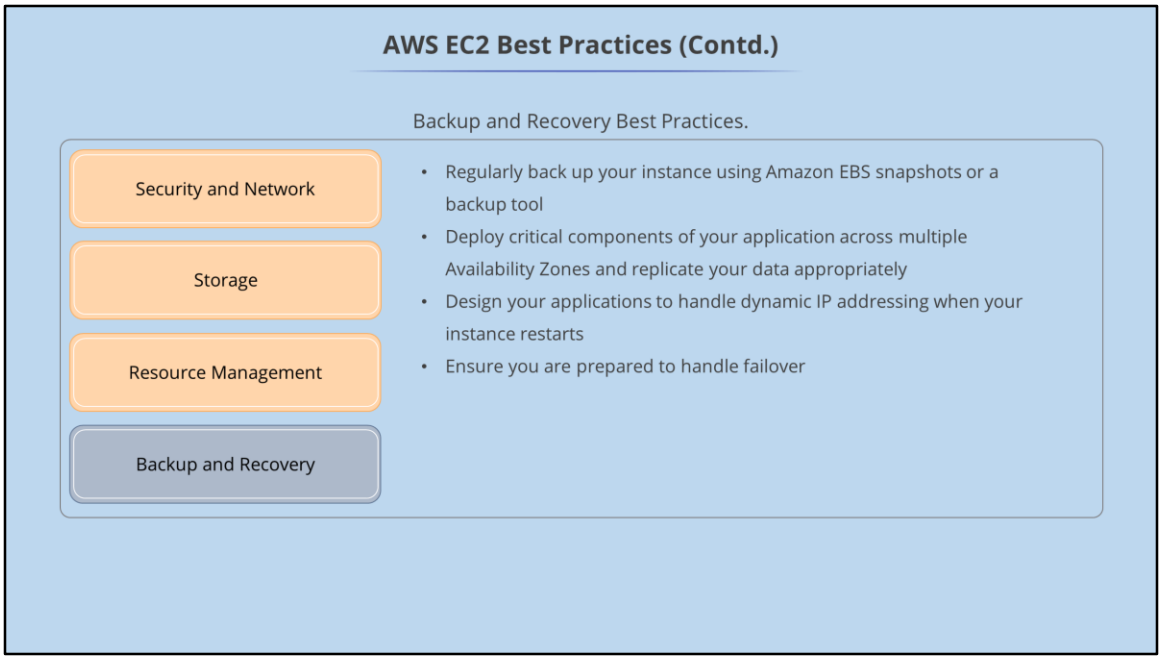
Ensure that the volume with your data persists after instance termination.



Resource Management

Use instance metadata and custom resource tags to track and identify your AWS resources.

View your current limits for Amazon EC2. Plan to request any limit increases in advance.



Backup and Recovery

Regularly back up your instance using Amazon EBS snapshots or a backup tool.

Deploy critical components of your application across multiple Availability Zones, and replicate your data appropriately.

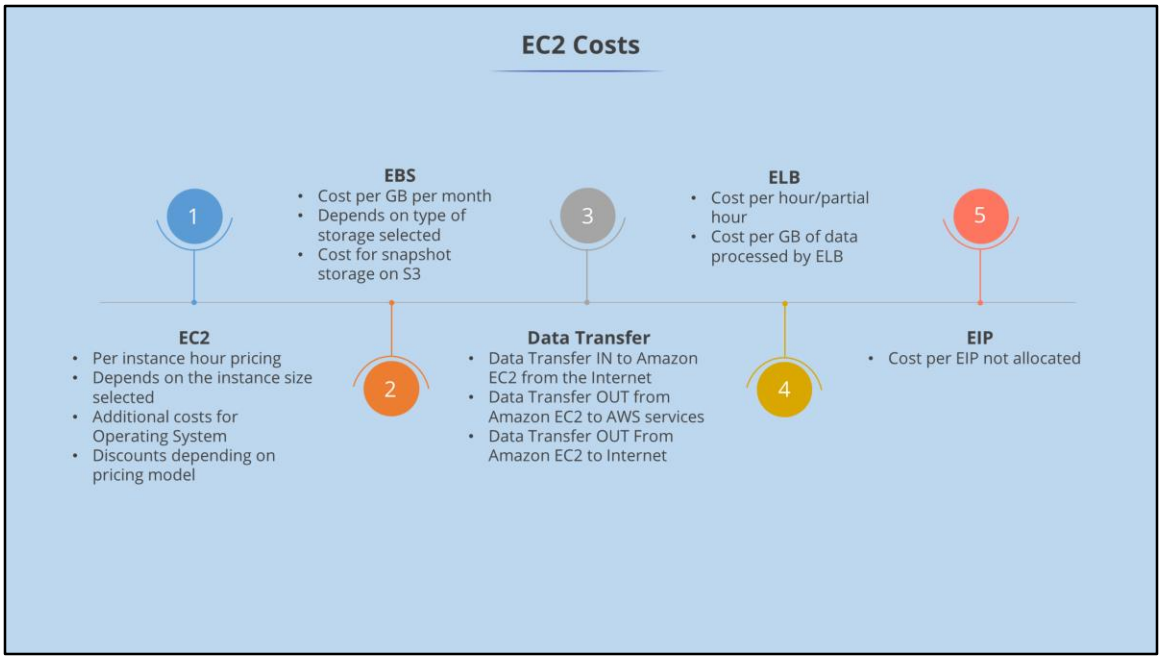
Design your applications to handle dynamic IP addressing when your instance restarts. For more information, see Amazon EC2 Instance IP Addressing.

Ensure that you are prepared to handle failover. For a basic solution, you can manually attach a network interface or Elastic IP address to a replacement instance. For more information, see Elastic Network

Interfaces (ENI). For an automated solution, you can use Auto Scaling. For more information, see the Auto Scaling Developer Guide.

AWS EC2 Costs

In this section, you'll learn about the costs associated with EC2 and other AWS compute resources.



EC2:

- Per instance hour pricing
- Depends on the instance size selected
- Additional costs for Operating System
- Discounts depends on pricing model

EBS:

- Cost per GB per month
- Depends on type of storage selected
- Cost for snapshot storage on S3

Data Transfer:

- Data transfer into Amazon EC2 from the Internet
- Data transfer out from Amazon EC2 to AWS services
- Data transfer out from Amazon EC2 to Internet

ELB:

- Cost per hour/partial hour
- Cost per GB of data processed by ELB

EIP:

- Cost per EIP not allocated

Practice Assignment: Configure ELB

Your client needs to set up a webserver that offers load balancing in the SIMPLILEARN_VPC.

ELB needs to launch into at least two subnets in different Availability Zones. So you will need to add a new public subnet to the SIMPLILEARN_VPC in a different Availability Zone to your existing public subnet. You will need to configure the custom route table so that the new public subnet has Internet access.

Then launch two Amazon Linux instances, one in each of your public subnets.

Set up a health_check.html file on each instance and configure ELB to serve both webservers.

BONUS: Rather than create two webservers, you can create one, configure it, create an AMI, and launch the second webserver from the newly created AMI. **ELB**

In this practice assignment you'll launch two webservers and configure ELB.

- Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud.
- An AMI is a template of a virtual instance that includes a template, launch permissions, and a block device mapping.
- Root devices can either be ephemeral (instance store) or EBS. AWS recommends EBS.
- There are a variety of instance types to suit all requirements.
- The storage categories are SSD-backed and HDD-backed.
- You can back up your EBS volumes by taking point-in-time snapshots and storing them on Amazon S3.
- EC2 on-demand instance pricing means you only pay for what you use with no long-term commitments.

- Placement groups are perfect for applications that require low network latency, high network throughput, or both.
- You can distribute incoming traffic across your Amazon EC2 instances in a single or multiple Availability Zones.
- Auto Scaling helps you ensure that you have the correct number of EC2 instances available to handle the load for your application.
- AWS Lambda lets you run code without provisioning or managing servers.
- AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services.
- The AWS Command Line Interface (CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

Knowledge Check

1

What happens when you deregister an AMI?

The AMI, Root Volume snapshot, and any instances launched from the AMI are deleted.

The AMI and any instances launched from the AMI are deleted.

The AMI and Root Volume snapshot are deleted.

Only the AMI is deleted.

1

What happens when you deregister an AMI?

The AMI, Root Volume snapshot, and any instances launched from the AMI are deleted.

The AMI and any instances launched from the AMI are deleted.

The AMI and Root Volume snapshot are deleted.

Only the AMI is deleted.

d

Only the AMI is deleted; the snapshot of the root volume persists as do any instances launched from the AMI.

2

Which EC2 Family is suitable for CPU-intensive applications?

T2

C4

R3

I2

Which EC2 Family is suitable for CPU-intensive applications?

2

T2

C4

R3

I2

b

C4 (and C3) are compute-optimized instances that are best suited for CPU-intensive databases, web servers, application servers, etc.

Which EC2 Family is suitable for Memory-intensive applications?

3

M4

EBS Optimized

R3

G2

Which EC2 Family is suitable for Memory-intensive applications?

3

M4

EBS Optimized

R3

G2

c

R3 instances are memory-optimized instances best suited for memory-intensive databases, web servers, application servers, etc.

4

EBS volumes are ___.

object-based storage

block-based storage

tape-based storage

USB-based storage

4

EBS volumes are ___.

object-based storage

block-based storage

tape-based storage

USB-based storage

b

EBS is block-based storage.

5 **Can you detach EBS volumes that are not the root volume without stopping the instance?**

Yes

No

Only on certain instance types

Only on EBS optimized instances

5 **Can you detach EBS volumes that are not the root volume without stopping the instance?**

Yes

No

Only on certain instance types

Only on EBS optimized instances

a

You can detach non-root volumes without stopping the instance, but it might take some time. Root volumes cannot be detached until an instance has been stopped.