

CSCI 5902 Adv. Cloud Architecting
Fall 2023
Instructor: Lu Yang

Modules 4 Adding a Compute Layer (Sections 4-6)
Sep 29, 2023

Housekeeping and feedback



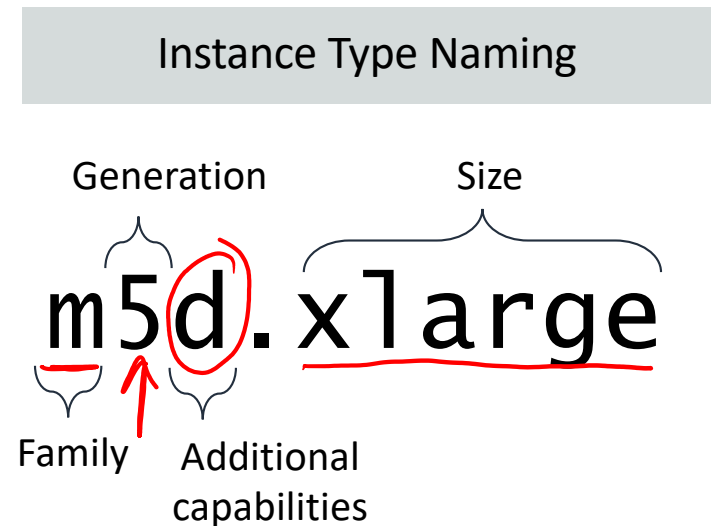
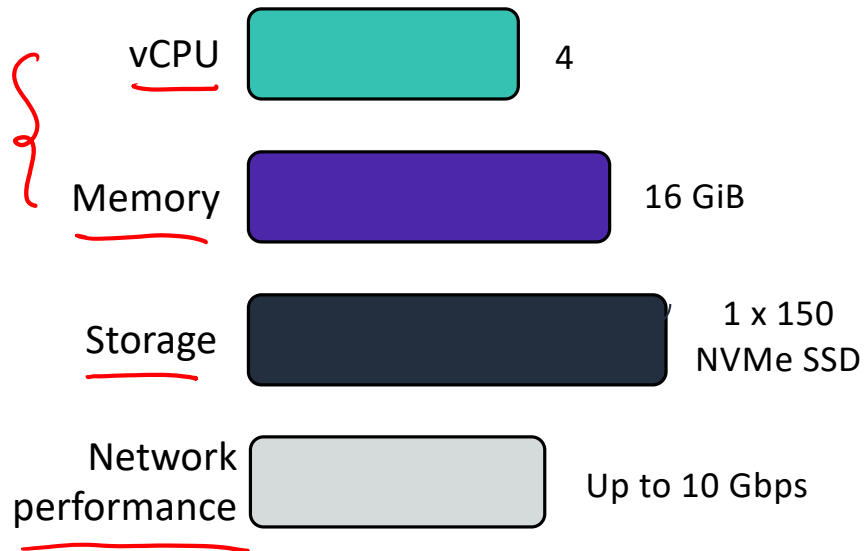
1. Final: tentatively 9:30-11:30am, Dec 11.
2. The first challenge lab due today.
3. No class on Monday, Oct 2.
4. PIER tours start 10:30-11:30am Tuesday, Oct 3 every two weeks. I will put up a sign-up spreadsheet Monday, Oct 2. Ten students each tour. Each student only has one chance to sign up for the tour. This is 100% optional.
5. Tech Career Fair
 - AWS vs. Azure 50:50
 - Certifications: Developer vs. Architect
 - Kubernetes
 - CRA to cloud
 - Career page of companies
 - Resume and first impression
 - High GPA vs. hands-on

Module 4: Adding a Compute Layer

Section 4: 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.



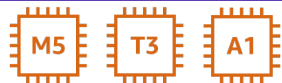
Suitability of instance types for workloads (1 of 2)



General purpose instance types

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

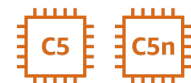
Example instance types:



Compute optimized instance types

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

Example instance types:

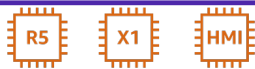


Suitability of instance types for workloads (2 of 2)

Memory optimized instance types

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

Example instance types:



Accelerated computing instance types

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

Example instance types:



Storage optimized instance types

- High-performance databases¹
- Real-time analytics¹
- Transactional workloads¹
- NoSQL databases¹
- Big data²
- Data warehouse²
- Log processing²

¹High I/O example instance type:



²Dense Storage example instance types:



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
- A good website summarizing EC2 instances: <https://instances.vantage.sh/>

With over 300 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

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

Recommendations for modifying current resources for better cost and performance.

Filter by one or more Regions 090765505187 Over-provisioned < 1 > [View detail](#)

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

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

Section 4 key takeaways



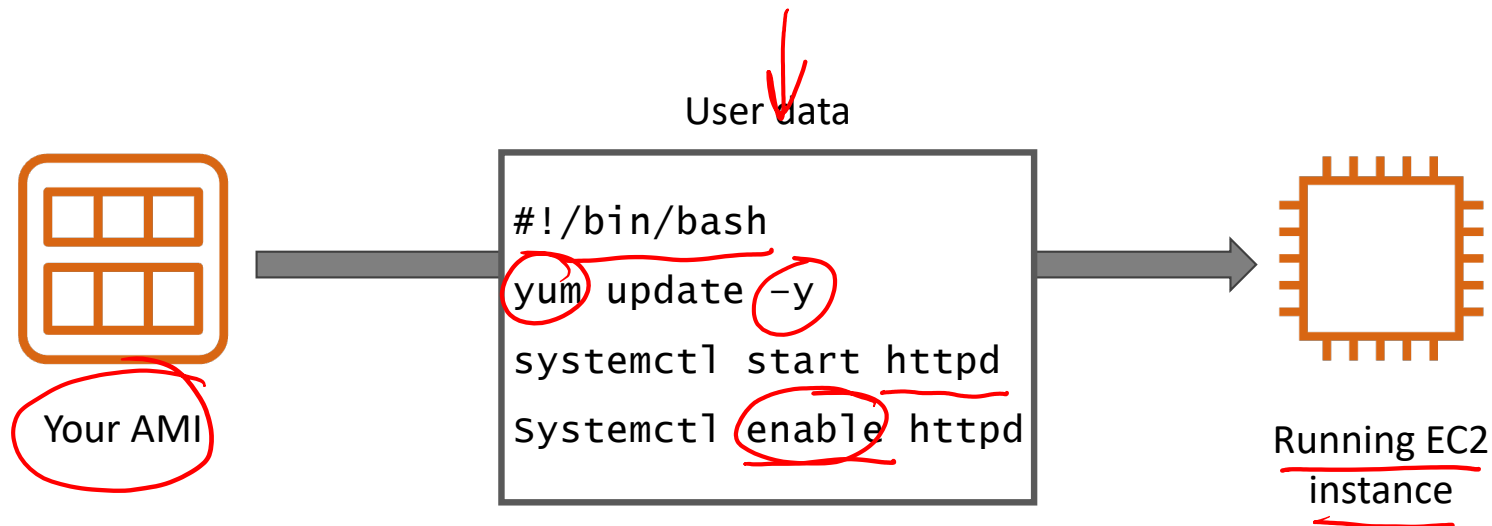
- An **EC2 instance type** defines a configuration of CPU, memory, storage, and network performance characteristics
- As a recommendation, choose **new generation instance types in a family** because they generally have better price-to-performance ratios
- Use the **Instance Types** page in the Amazon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

Module 4: Adding a Compute Layer

Section 5: Using user data to configure an EC2 instance

EC2 instance user data

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



Retrieving instance metadata

Instance metadata is information about your instance. *link-local*

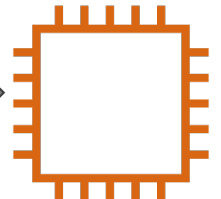
- Is accessible from your instance at URL: <http://169.254.169.254/latest/meta-data/>
- Can be retrieved from a user data script



Your AMI

User data

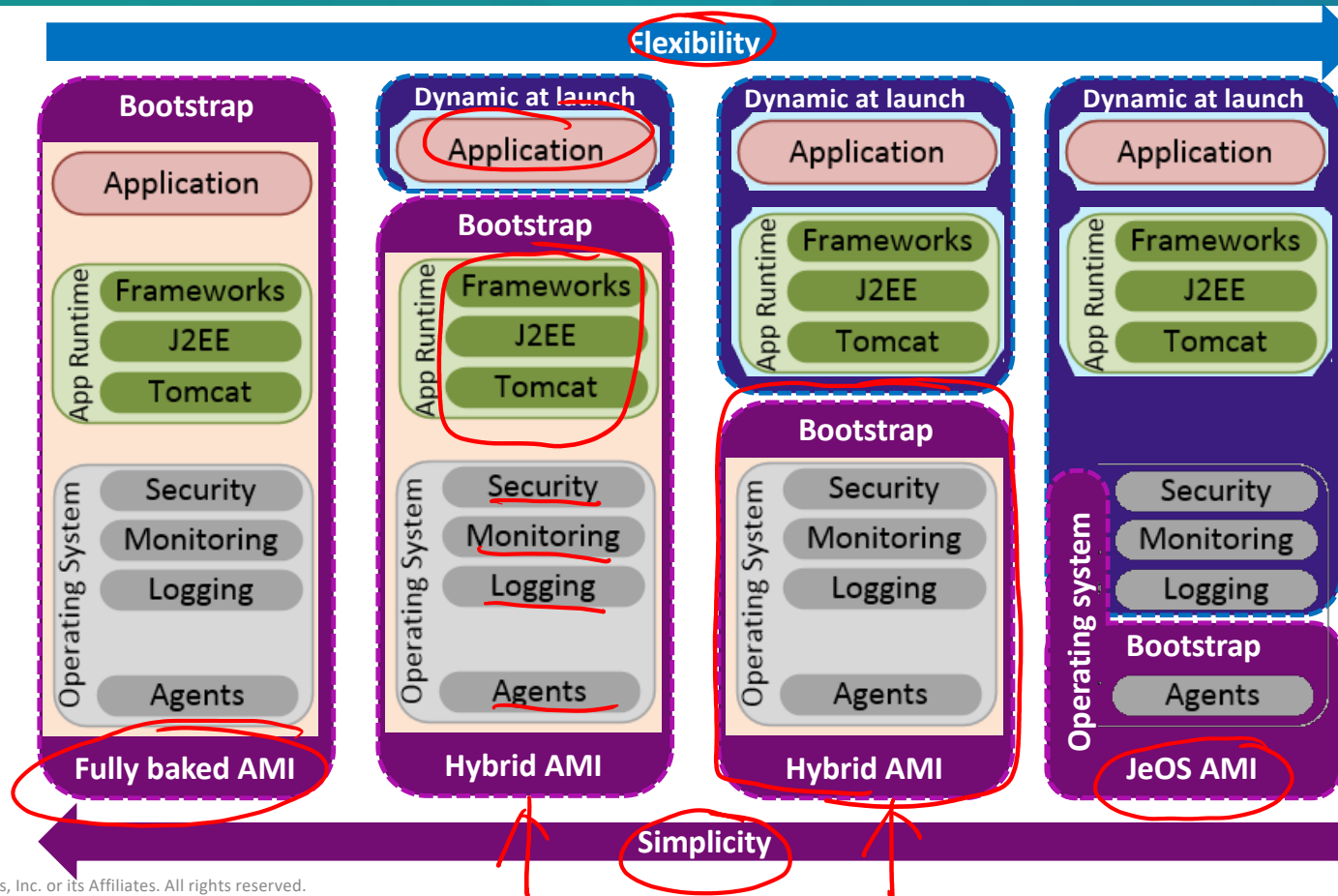
```
#!/bin/bash
yum update -y
hostname = $(curl -s http://169.254.169.254/latest/meta-data/public-hostname)
```



Running EC2 instance

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

Configuring an EC2 instance: AMI versus user data



Demonstration: Configuring an EC2 Instance with User Data

link



Section 5 key takeaways



- User data enables you to configure an EC2 instance when you launch it.
- Information about a running instance can be accessed in the instance through an instance metadata URL.
- Baking configurations into an AMI increases AMI build time, but decreases instance boot time. Configuring an instance by using user data decreases AMI build time, but increases instance boot time.

Module 4: Adding a Compute Layer

Section 6: 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)



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

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]

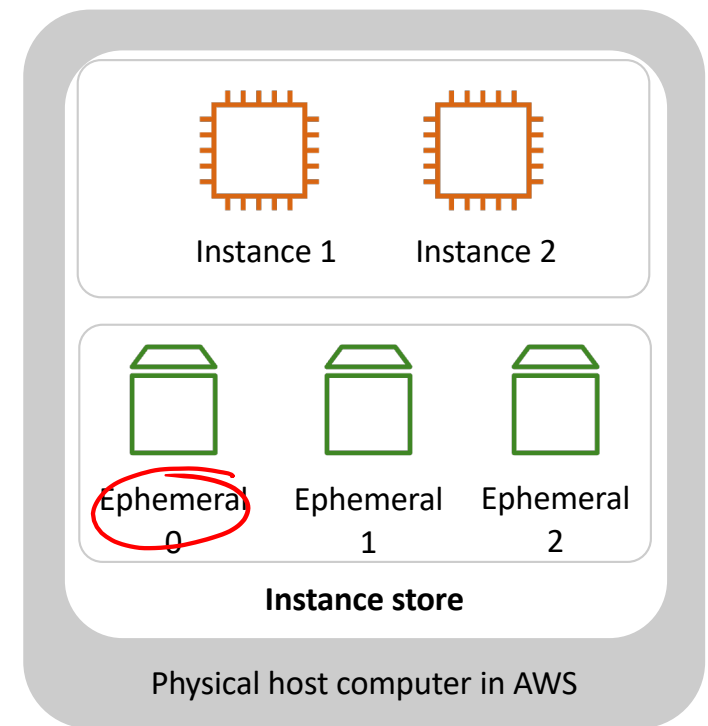


Amazon FSx for Windows
File Server

File systems

Instance store

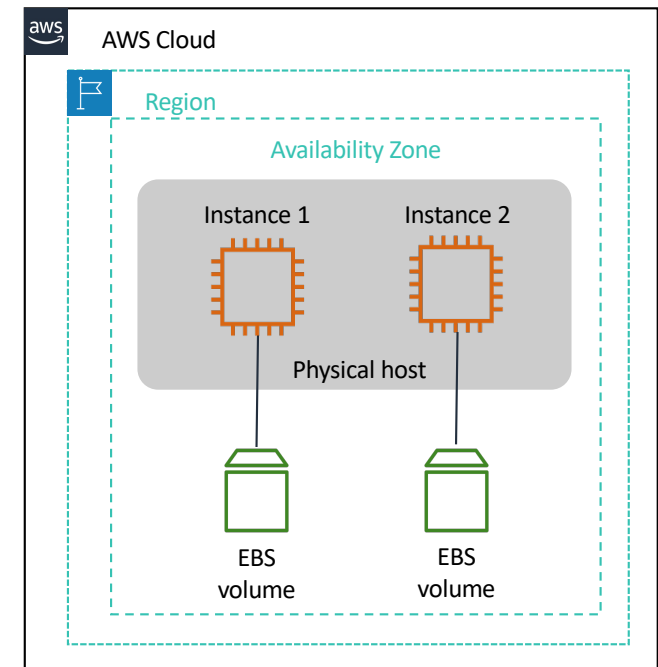
- An instance store provides non-persistent storage to an instance –
 - The data is stored on the same physical server where the instance runs
- Characteristics –
 - Temporary block-level storage
 - Uses HDD or SSD
 - Instance store data is lost when the instance is stopped or terminated
- Example use 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
 - Designed for mission-critical workloads
 - Automatically replicated within a single AZ
 - Dynamically increase capacity and change the volume type without downtime or performance to the life systems
- Example use cases –
 - Stand-alone database
 - Application
 - File system
 - Operation system
 - Data storage

Instance
store
encryption?



IOPS vs. Throughput

- IOPS and Throughput depend on block size of the disk.
- Block size is a defined chunk of storage where an amount of data can be written or read.
- Storage vendors define those block size to be 512 Bytes for HDD & 4K for SSD based.

IOPS

- The number of read and write operation per second
- The ability to action reads and writes very quickly
- Important metric for quick transactions, low-latency apps, transactional workloads
- Choose Provisioned IOPS SSD (io1 or io2)

VS

Throughput

- The number of bits read and written per second (MB/s)
- The ability to deal with large datasets
- Important metric for large datasets, large I/O sizes, complex queries
- Choose Throughput Optimized HDD (st1)

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 volumes		Provisioned IOPS SSD volumes		
Volume type	gp3	gp2	io2 Block Express ‡	io2	io1
Durability	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)		99.999% durability (0.001% annual failure rate)		99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)
Use cases	<ul style="list-style-type: none">Transactional workloadsVirtual desktopsMedium-sized, single-instance databasesLow-latency interactive applicationsBoot volumesDevelopment and test environments		Workloads that require: <ul style="list-style-type: none">Sub-millisecond latencySustained IOPS performanceMore than 64,000 IOPS or 1,000 MiB/s of throughput	<ul style="list-style-type: none">Workloads that require sustained IOPS performance or more than 16,000 IOPSI/O-intensive database workloads	
Volume size	1 GiB - 16 TiB		4 GiB - 64 TiB		4 GiB - 16 TiB
Max IOPS per volume (16 KiB I/O)	16,000		256,000		64,000 †
Max throughput per volume	1,000 MB/s	250 MiB/s *	4,000 MB/s		1,000 MB/s †
Amazon EBS Multi-attach	Not supported		Supported		
Boot volume	Supported				

Amazon EBS Multi-Attach enables you to attach a single Provisioned IOPS SSD (io1 or io2) volume to multiple instances that are in the same Availability Zone.

A lot of limitations.....

EFA

Micro System

500G GP2

Amazon EBS Multi-Attach enables you to attach a single Provisioned IOPS SSD (io1 or io2) volume to multiple instances that are in the same Availability Zone. A lot of limitations.....

Virtual System
EFA

500G GP2

Amazon EBS HDD-backed volume types



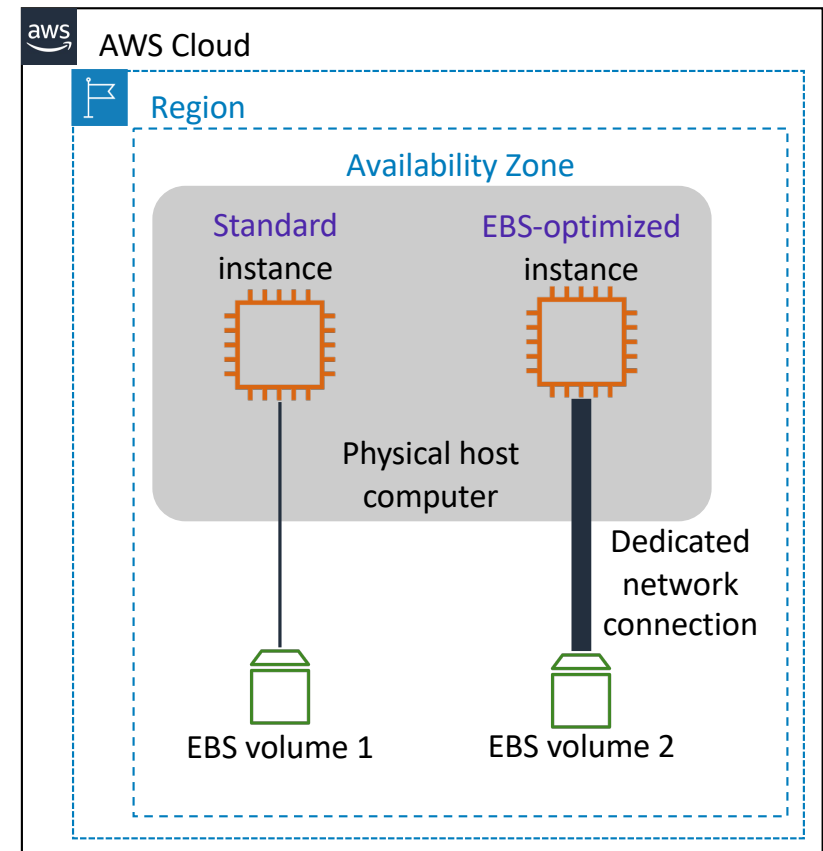
Amazon EBS HDD-backed volumes work well when the focus is on throughput.

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

↓
10%



Demonstration: Add an EBS to an EC2 instance

Attach the EBS
Format the EBS
Mount the EBS
Permanent mount the EBS

<https://www.youtube.com/watch?v=r4-lES7Zpk> (1:30-11:08)



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 EBS

Amazon S3: Is an option, but is not ideal



Amazon S3

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



Amazon EFS (Linux)



Amazon FSx for Windows File Server (Windows)

Amazon EFS (1/2)

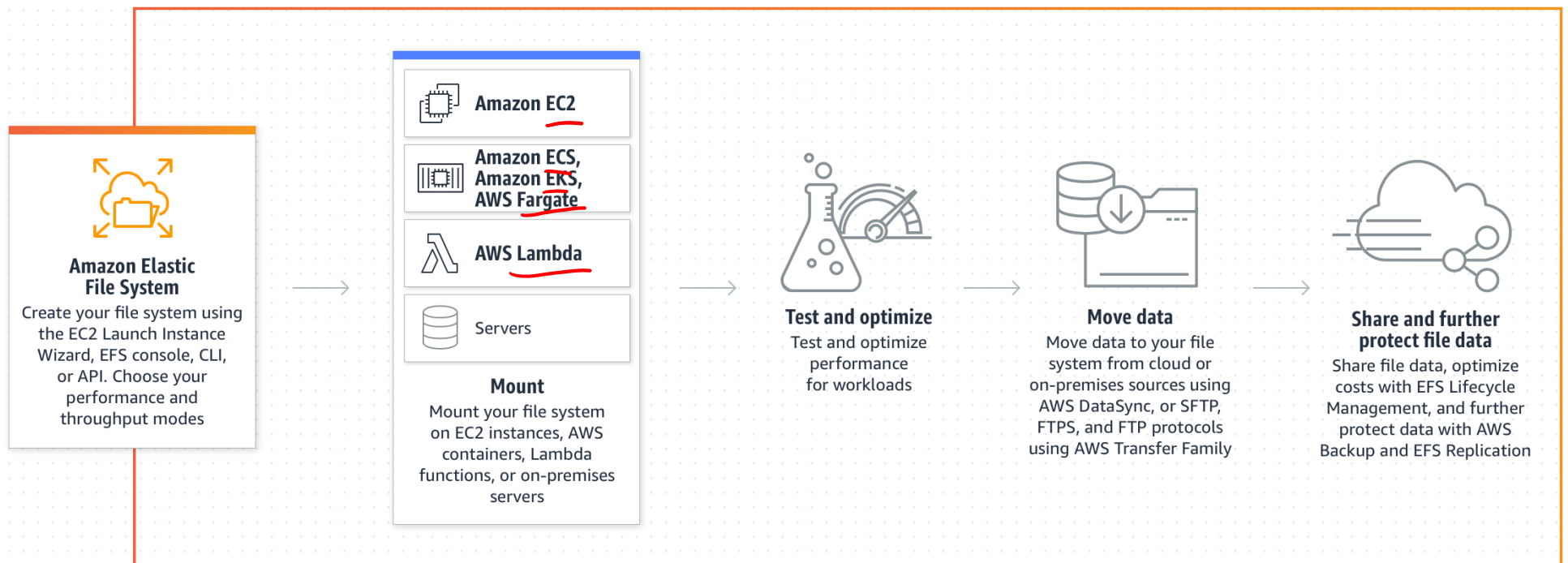


Amazon
Elastic File System
(Amazon EFS)

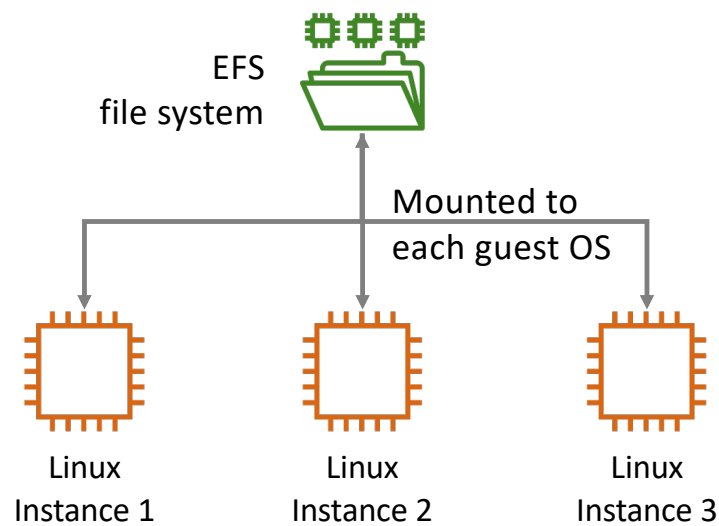
Amazon EFS provides file system storage for Linux-based workloads.

- 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 in multiple AZs
- Compatible with all Linux-based AMIs for Amazon EC2
- Can support thousands of concurrent NFS connections
- Read-after-write consistency
- Expensive

Amazon EFS (2/2)



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
```

EFS

Amazon FSx for Windows File Server



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



Amazon FSx for
Windows File
Server

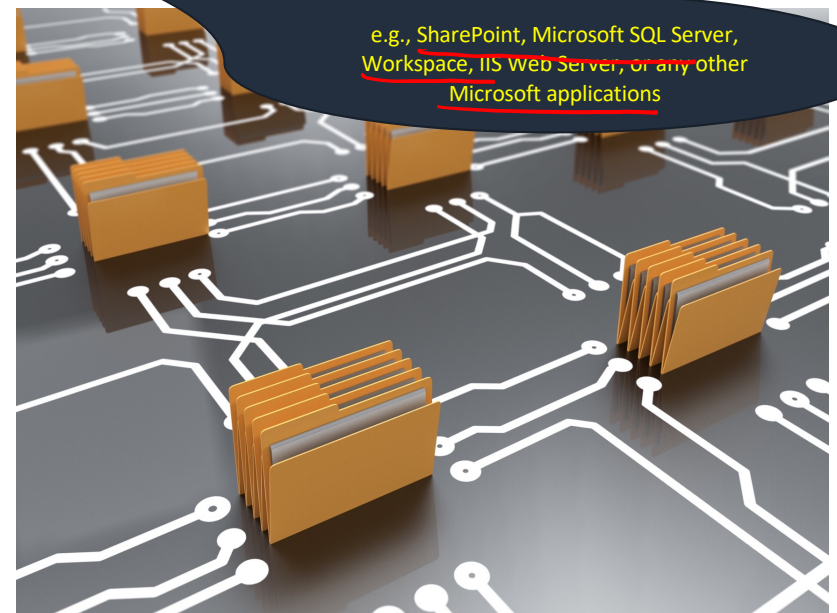
- Native Microsoft Windows compatibility
- New Technology File System (NTFS)
- Native Server Message Block (SMB) protocol version 2.0 to 3.1.1
- Distributed File System (DFS) Namespaces and DFS Replication
- Integrates with Microsoft Active Directory and supports Windows access control lists (ACLs)
- 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



Amazon FSx for Lustre vs. AWS EFS



Amazon FSx for Lustre?	AWS EFS
<p>FSx for Lustre is an entirely managed service that offers high-performance, scalable, cost-effective storage for compute workloads. A lot of workloads, including high-performance computing, financial simulation, video rendering, and machine learning, are reliant on compute instances having access to the same data via high-performance joint storage.</p>	<p>Amazon Elastic File System (EFS) is a set-and-forget, serverless, simple elastic file system which may be used with on-premises resources and AWS cloud services. It is created to scale on demand (petabytes) and doesn't disrupt applications. It grows and shrinks automatically as you remove or add files, doing away with the requirement to manage and provision capacity to allow for expansion.</p>
<p>AWS FSx for Lustre provides sub-millisecond latencies, millions of IOPS, and throughput of as much as <u>hundreds of gigabytes per second</u>. It offers several deployment types and storage options to maximize performance and cost for your workload needs.</p>	<p>The Amazon EFS web services interface lets you develop and configure file systems simply. The service deals with all the file storage infrastructure on your behalf, so you don't need to handle the deployment, maintenance and patching of intricate file system configurations.</p>

Section 6 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
- For a [data volume that servers extremely high-performance workloads](#), use Amazon FSx for Lustre

Thank you, Kahoot!

© 2020 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections or feedback on the course, please email us at: aws-course-feedback@amazon.com. For all other questions, contact us at: <https://aws.amazon.com/contact-us/aws-training/>. All trademarks are the property of their owners.

