What is Virtualization?

- Virtualization is a fundamental technology in cloud computing that enables the efficient utilization of physical hardware resources by creating multiple virtual instances of computing resources such as servers, storage, and networking.
- It allows for the efficient allocation and management of computing resources.
- It plays a crucial role in achieving the scalability, flexibility, and cost-effectiveness that cloud computing offers.

Benefits of Virtualization

- Resource Isolation: Virtualization provides isolation between VMs, ensuring that one VM's activities do not impact the performance or security of others.
- Resource Pooling: Virtualized resources can be pooled together and allocated dynamically based on demand, improving resource utilization.
- Elasticity: Cloud providers can quickly provision and deprovision VMs as needed, allowing users to scale their resources up or down easily.
- Hardware Independence: VMs are abstracted from the underlying physical hardware, making it easier to migrate workloads across different hardware platforms.

Architecture of Virtualization

The architecture of virtualization typically involves the following components:

1. Hypervisor or Virtual Machine Monitor (VMM): The hypervisor is a software layer that allows multiple operating systems to run on a single host. It allocates the CPU, memory, and other resources to each virtual machine (VM). There are two types of hypervisors: Type 1 (bare-metal) hypervisors, which run directly on the host's hardware, and Type 2 hypervisors, which run on a conventional operating system.
2. Host Machine: The physical server or hardware that hosts the hypervisor and runs multiple VMs is referred to as the host machine. It provides the necessary computing resources and hardware support for virtualization.
3. Virtual Machines (VMs): Virtual machines are isolated environments created by the hypervisor that mimic physical computers. Each VM has its own virtualized hardware resources, including CPU, memory, storage, and network interfaces.
4. Guest Operating Systems: These are the operating systems installed on each virtual machine. They can be different from the host operating system and can run various applications independently of each other.
5. Virtualization Management Software: This software allows administrators to manage and monitor the virtualized environment. It provides tools for creating, configuring, and

controlling virtual machines, as well as monitoring resource usage and optimizing performance.

6. Storage and Networking Infrastructure: Virtualization also requires a robust storage and networking infrastructure to support the virtualized environments effectively. This includes storage area networks (SANs), network-attached storage (NAS), and virtual networking components for communication between VMs and with the external network.

Types of Virtualization

- Server Virtualization: Server virtualization is the most common form of virtualization in cloud computing. It involves dividing a physical server into multiple virtual machines (VMs) using a hypervisor or virtual machine monitor (VMM). Each VM operates as an independent server with its own operating system (OS) and applications. This allows for the efficient sharing of physical server resources among multiple users or applications, leading to better resource utilization and cost savings.
- Storage Virtualization: In addition to server virtualization, cloud computing also employs storage virtualization. This involves abstracting and pooling physical storage resources to create virtual storage volumes that can be allocated to VMs as needed. Storage virtualization helps in managing data efficiently, ensuring data redundancy, and providing features like snapshots and backups.
- Network Virtualization: Network virtualization involves abstracting and virtualizing network resources such as switches, routers, and firewalls. It enables the creation of virtual networks that are isolated from each other and can be customized to suit the needs of different applications or tenants in a multi-tenant cloud environment.
- Desktop Virtualization: Desktop virtualization allows users to access virtual desktop environments hosted on remote servers. This is particularly useful for remote work scenarios and providing a consistent user experience across different devices.

Virtualization is a technology that enables the creation of virtual or simulated versions of computing resources, such as servers, storage devices, networks, and even operating systems. These virtualized resources can operate independently of the physical hardware on which they are hosted. The primary goal of virtualization is to make the most efficient use of available hardware resources, improve scalability, enhance resource isolation, and simplify management.

Here are some key aspects of virtualization:

1. **Server Virtualization**: In server virtualization, a hypervisor or virtual machine monitor (VMM) is used to create multiple virtual machines (VMs) on a single physical server. Each VM can run its own operating system and applications, and they are isolated from one another. This allows multiple workloads or applications to share the same physical hardware, improving resource utilization and flexibility.

2. **Desktop Virtualization**: Desktop virtualization involves creating virtual desktop instances that run on centralized servers and are accessed by end-users on their devices. It allows for centralized management, security, and easy access to desktop environments from various devices, making it useful for remote work and application compatibility.

3. **Storage Virtualization**: Storage virtualization abstracts physical storage resources and pools them into a single virtual storage resource. This enables more efficient management, allocation, and scalability of storage, as well as features like data migration, snapshots, and redundancy.

4. **Network Virtualization**: Network virtualization abstracts and virtualizes network resources, allowing multiple virtual networks to run on the same physical infrastructure. It enables isolation, segmentation, and customization of network services, making it easier to manage complex network environments.

5. **Application Virtualization**: Application virtualization separates applications from the underlying operating system, allowing them to run in isolated environments. This can simplify software deployment, enhance compatibility, and improve security by isolating applications from each other.

6. **Containerization**: Containerization is a lightweight form of virtualization that involves running applications and their dependencies in isolated containers. Containers share the host OS kernel but provide a consistent and isolated runtime environment for applications. Popular containerization technologies include Docker and Kubernetes.

7. **Operating System Virtualization**: This form of virtualization, often referred to as containerization or OS-level virtualization, allows multiple instances of an operating system to run on a single physical server. It's different from traditional VM-based virtualization, as it doesn't require a full OS for each instance, making it lightweight and efficient.

Benefits of virtualization include improved resource utilization, cost savings through consolidation, enhanced scalability, better resource isolation and security, simplified management, and increased flexibility for deploying and managing applications and services.

Virtualization is a foundational technology in cloud computing, data centers, and modern IT environments, as it enables the efficient use of hardware resources and the creation of flexible, scalable, and dynamic computing environments.

AWS EC2

- Amazon Elastic Compute Cloud (Amazon EC2) provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud.
- Using Amazon EC2 reduces hardware costs so you can develop and deploy applications faster.
- You can use Amazon EC2 to launch as many or as few virtual machines/servers as you need, configure security and networking, and manage storage.
- You can add capacity (scale up) to handle compute-heavy tasks, such as monthly or yearly processes, or spikes in website traffic.
- When usage decreases, you can reduce capacity (scale down) again.

Create EC2 Instance with User Data

To create an EC2 instance with user data, you can use the AWS Management Console, AWS CLI, or AWS SDKs. Here is an example of using the AWS Management Console to create an EC2 instance with user data:

1. Sign in to the **AWS Management Console** and Navigate to the **EC2 dashboard.**
2. **Launch Instance:** Click on the "Launch Instance" button.
3. **Choose an Amazon Machine Image (AMI):** Select an AMI based on your requirements.
4. **Choose an Instance Type:** Select the appropriate instance type based on your workload.
5. **Configure Instance Details:** Configure the instance details like the number of instances, network settings, and other options.

6. **Configure Security Group:** Configure the security group to control inbound and outbound traffic for your instance.
7. **Add User Data:** In the "Configure Instance Details" page, you can find the "Advanced Details" section. Enter your user data script in the "User data" field. User data can be used to perform common automated configuration tasks and runs when the instance starts.
8. **Add Storage:** Configure the storage options for your instance.
9. **Add Tags:** Optionally, add tags for better identification and organization of your resources.
10. **Review and Launch:** Review your configuration and launch the instance.

Amazon EC2 Instance Types

Amazon Elastic Compute Cloud (Amazon EC2) provides a wide range of instance types optimized to fit different use cases. These instance types vary in terms of computing power, memory, storage, and networking capacity.

1. **General Purpose Instances:** These instances provide a balance of compute, memory, and networking resources and are suitable for a variety of diverse workloads. Examples include the M6g, M5, and M5a instance types.

2. **Compute Optimized Instances:** These instances are ideal for compute-bound applications that benefit from high-performance processors. Examples include the C6g, C5, and C5a instance types.

3. **Memory Optimized Instances:** These instances are designed to deliver fast performance for workloads that process large data sets in memory. Examples include the R6g, R5, and R5a instance types.

4. **Accelerated Computing Instances:** These instances use hardware accelerators, such as Graphics Processing Units (GPUs), to perform functions that require additional processing power. Examples include the P4, P3, G4, and Inf1 instance types.

5. **Storage Optimized Instances:** These instances are designed for workloads that require high, sequential read and write access to very large data sets on local storage. Examples include the I3, I3en, and D2 instance types.

AWS EC2 Security Groups

- Amazon EC2 Security Groups act as virtual firewalls for your EC2 instances to control incoming and outgoing traffic.

- They control the network traffic to and from one or more EC2 instances.
- Every instance in your EC2 network should be associated with at least one security group.

Here are some key points to understand about EC2 Security Groups:

- ✓ **Traffic Control:** Security groups control inbound and outbound traffic for your EC2 instances. You can specify the protocols, ports, and source IP ranges that are allowed to access your instances.
- ✓ **Stateful:** Security groups are stateful, meaning if you allow inbound traffic, the return traffic is automatically allowed, regardless of any outbound rules. You don't need to create rules specifically for allowing response traffic.
- ✓ **Default Security Group:** Every EC2 instance is automatically associated with a default security group if you don't specify any security group during instance creation. This default security group allows all outbound traffic and allows inbound traffic only from other instances within the same security group.
- ✓ **Custom Security Groups:** You can create custom security groups and define rules based on your specific requirements. These rules control the inbound and outbound traffic for the instances associated with the security group.
- ✓ **Changes to Security Groups:** You can modify the rules of a security group at any time. Changes take effect immediately, affecting the traffic to the associated instances.

AWS EC2 Placement Groups

- Amazon EC2 Placement Groups are logical groupings used for the placement of instances in order to provide a higher level of control over the underlying hardware.

There are several types of placement groups:

1. **Cluster Placement Group:** This type of placement group is designed for applications that benefit from low network latency, high network throughput, or both. Instances in a cluster placement group are placed in a low-latency, full-bisection bandwidth network within a single Availability Zone (AZ).
2. **Spread Placement Group:** This type of placement group is recommended for applications that have a small number of critical instances that should be kept separate from each other. Spread placement groups place instances on distinct hardware, which reduces the risk of simultaneous failures.
3. **Partition Placement Group:** This type of placement group is recommended for large distributed and replicated workloads, such as Hadoop, Cassandra, and Kafka. Partition placement groups spread the instances across logical partitions, ensuring that each partition has its own set of racks, enabling you to isolate groups of instances from each other.

Public Vs. Private Vs. Elastic IP

Public Vs. Private IP	
Public IP	**Private IP**
Used over the public network such as internet.	Used within the private network such as LAN.
Recognized over the internet	Not recognized over the internet. Helps to recongnize inside the private network.
Public IPs are unique over the globe	Private IPs are unique within the private network or LAN
Public IPs are paid	Private IPs are free of cost
Assigned by network administrator	Assigned by service provider

Public Vs. Elastic IP	
Public IP	**Elastic IP**
Public IP is assigned to your launched instance	Elastic IP is assigned to your AWS account
Public IPs dynamic in nature	Elastic IPs are static in nature
When you stop, hibernate or terminate your instance, Public IP will release and when you restart the instance, new public IP will be assigned your instance	When you have assigned elastic IP to your instance, it will remain same across instance stop or hibernate.
When Public IP is released, you cannot reuse it back.	When Elastic IP is disassociated from your instance, you can reuse it back.
	AWS charge you small hourly charge for Elastic IP

Elastic Network Interface (ENI)

- In Amazon Web Services (AWS), Elastic Network Interface (ENI) is a virtual network interface that you can attach to an instance in a VPC.
- It enables an instance to have one or more network interfaces, each with its own private IP address, security groups, and Elastic IP addresses.

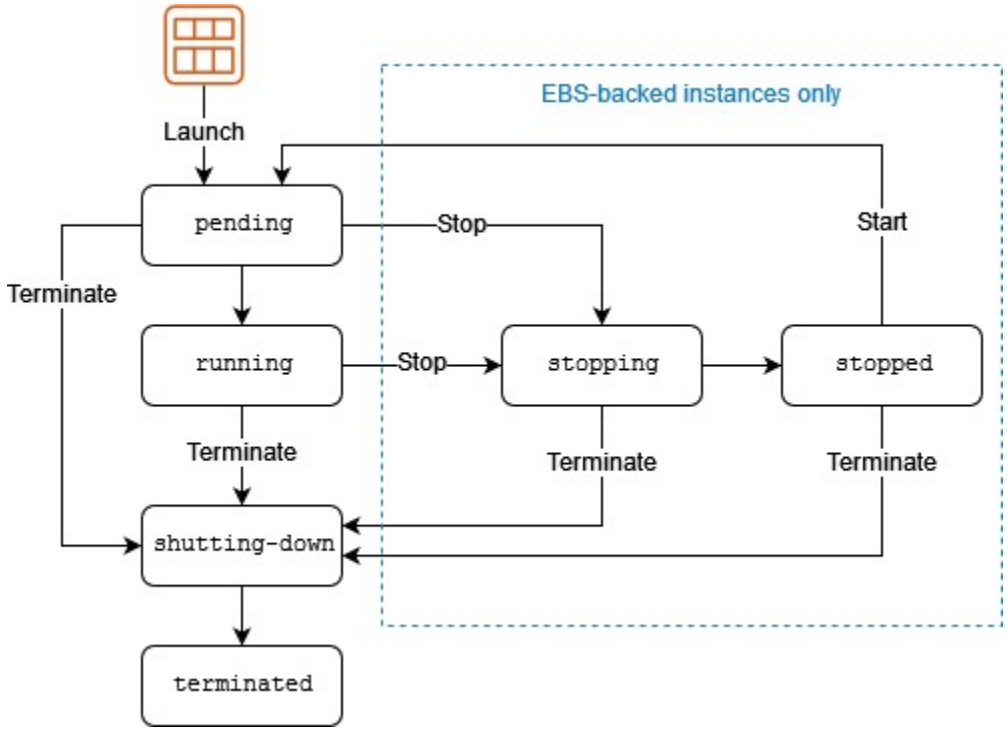
Here are some key points about EC2 ENIs:

✓ Attachment and Detachment: You can attach and detach ENIs to EC2 instances as needed.

- ✓ High Availability: ENIs provide high availability by allowing you to attach a new ENI to an instance if the primary ENI is lost due to hardware failure.
- ✓ Multiple IP Addresses: You can assign multiple private IP addresses to an ENI, enabling your instance to serve multiple purposes such as hosting multiple websites or applications.
- ✓ Elastic IP Addresses: You can also associate Elastic IP addresses with ENIs to ensure that your instance can be reached at the same IP address even if it's stopped and restarted.

Amazon EC2 Instance Lifecycle

An Amazon EC2 instance transitions through different states from the moment you launch it through to its termination.

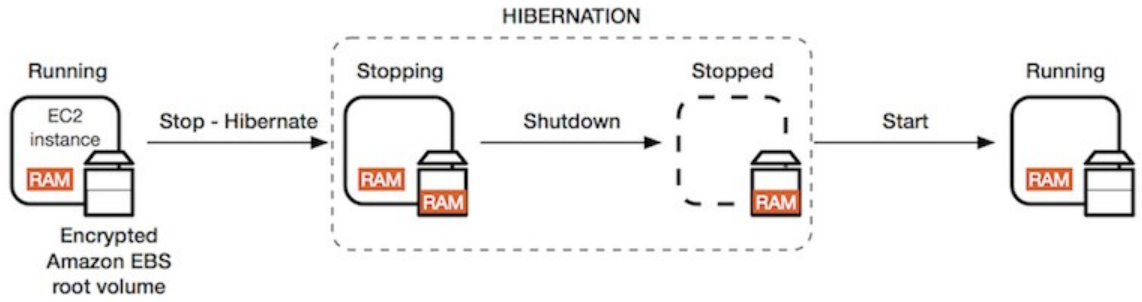


Instance state	Description
pending	The instance is preparing to enter the running state. An instance enters the pending state when it is launched or when it is started after being in the stopped state.
running	The instance is running and ready for use.

Instance state	Description
stopping	The instance is preparing to be stopped.
stopped	The instance is shut down and cannot be used. The instance can be started at any time.
shutting-down	The instance is preparing to be terminated.
terminated	The instance has been permanently deleted and cannot be started.

Amazon EC2 Instance Hibernate

When you hibernate an instance, we signal the operating system to perform hibernation (suspend-to-disk), which saves the contents from the instance memory (RAM) to your Amazon EBS root volume. We persist the instance's Amazon EBS root volume and any attached Amazon EBS data volumes. When you start your instance, the Amazon EBS root volume is restored to its previous state and the RAM contents are reloaded. Previously attached data volumes are reattached and the instance retains its instance ID.



When you hibernate a running instance, the following happens:

- When you initiate hibernation, the instance moves to the stopping state. Amazon EC2 signals the operating system to perform hibernation (suspend-to-disk). The hibernation freezes all of the processes, saves the contents of the RAM to the EBS root volume, and then performs a regular shutdown.
- After the shutdown is complete, the instance moves to the stopped state.

- Any EBS volumes remain attached to the instance, and their data persists, including the saved contents of the RAM.
- Any Amazon EC2 instance store volumes remain attached to the instance, but the data on the instance store volumes is lost.
- When you start the instance, the instance boots up and the operating system reads in the contents of the RAM from the EBS root volume, before unfreezing processes to resume its state.
- The instance retains its private IPv4 addresses and any IPv6 addresses. When you start the instance, the instance continues to retain its private IPv4 addresses and any IPv6 addresses.
- Amazon EC2 releases the public IPv4 address. When you start the instance, Amazon EC2 assigns a new public IPv4 address to the instance.
- The instance retains its associated Elastic IP addresses. You're charged for any Elastic IP addresses that are associated with a hibernated instance.

Amazon Machine Image (AMI)

- ✓ Amazon Machine Images (AMIs) are pre-configured templates that contain a software configuration (for example, an operating system, application server, and applications).
- ✓ You can use AMIs to launch EC2 instances with specific configurations, making it easier to set up new instances based on those templates.
- ✓ AMIs helps in quickly and efficiently deploying instances with predefined configurations, reducing the time and effort needed to set up new environments.

Here are some key points about EC2 AMIs:

- ✓ Types of AMIs: There are several types of AMIs, including AWS provided, community AMIs, and your own custom AMIs. AWS provides a wide variety of public AMIs that include different operating systems and software configurations. Community AMIs are created and shared by other AWS users. You can also create your own custom AMIs tailored to your specific requirements.
- ✓ Public and Private AMIs: Public AMIs are available to all AWS users, while private AMIs are only accessible to the AWS account that created them. You can control access to your private AMIs by sharing them with specific AWS accounts.

AWS EBS

- Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with EC2 instances.
- EBS volumes behave like raw, unformatted block devices.
- An Amazon EBS volume is a durable, block-level storage device that you can attach to your instances. After you attach a volume to an instance, you can use it as you would use a physical hard drive.
- EBS volumes are flexible. you can dynamically increase size, modify the provisioned IOPS capacity, and change volume type
- You can use EBS volumes as primary storage for data
- You create an EBS volume in a specific Availability Zone, and then attach it to an instance in that same Availability Zone. To make a volume available outside of the Availability Zone, you can create a snapshot and restore that snapshot to a new volume anywhere in that Region.
- You can attach multiple EBS volumes to a single instance. The volume and instance must be in the same Availability Zone.
- Amazon EBS provides the following volume types: General Purpose SSD (gp2 and gp3), Provisioned IOPS SSD (io1 and io2), Throughput Optimized HDD (st1), Cold HDD (sc1).
- You can create point-in-time snapshots of EBS volumes, The same snapshot can be used to create as many volumes as needed.
- You can create your EBS volumes as encrypted volumes, When you create an encrypted EBS volume and attach it to a EC2 instance, data stored on the volume, and snapshots created from the volume are all encrypted.

EC2 Instance Store

- Amazon EC2 instance store provides temporary block-level storage for EC2 instances.
- This storage is directly attached to the host server of the instance, which makes it ideal for temporary data that often requires very high I/O performance.
- The data on an instance store volume persists only during the life of the associated instance. If the instance is stopped or terminated, any data on instance store volumes is lost.
- Instance store volumes can be beneficial for applications that require temporary storage, like buffers, caches, and other temporary content.
- However, it's crucial to understand that it is not suitable for long-term data storage

Amazon EBS volume types:

1. **General Purpose (gp2):** General Purpose SSD (gp2) volumes are suitable for a wide range of workloads. They offer a balance of price and performance.
    - Performance: Good burst performance and consistent baseline performance.
    - Use Cases: Suitable for most workloads, including boot volumes and small to medium-sized databases.
2. **Provisioned IOPS (io1):** Provisioned IOPS SSD (io1) volumes are designed for applications that require high and consistent I/O performance.
    - Performance: Allows you to provision a specific amount of IOPS (Input/Output Operations Per Second) and throughput, ensuring predictable performance.
    - Use Cases: High-performance databases, large-scale applications, and applications with high I/O requirements.
3. **Throughput Optimized (st1):** Throughput Optimized HDD (st1) volumes are optimized for throughput and provide low-cost storage for frequently accessed, throughput-intensive workloads.
    - Performance: High throughput, suitable for large data transfers and data warehouses.
    - Use Cases**: Big data analytics, data warehousing, log processing, and streaming workloads.
4. **Cold HDD (sc1):** Cold HDD (sc1) volumes are designed for infrequently accessed data that can tolerate lower performance but offers cost savings.
    - Performance: Lower throughput and IOPS compared to other types but cost-effective.
    - Use Cases: Archiving, backups, and less frequently accessed data.

Amazon EBS Snapshots

- Amazon Elastic Block Store (EBS) Snapshots allow you to create point-in-time backups of your EBS volumes.
- EBS snapshots are essential for data backup, disaster recovery, and migrating data between AWS regions.
- EBS snapshots are a vital component of AWS data management and protection strategies. They ensure that your data remains safe, available, and recoverable in various scenarios, including data loss, hardware failures, and disaster recovery situations.

Here's an overview of EBS snapshots and how they work:

- Point-in-Time Backups**: EBS snapshots capture the data on an EBS volume at a specific point in time. This allows you to create backups of your volumes, protecting your data against accidental deletion, corruption, or hardware failures.
- Incremental Backups**: EBS snapshots are incremental. This means that when you create a snapshot, only the data that has changed since the last snapshot is copied to the new snapshot. This minimizes both the time it takes to create a snapshot and the storage costs associated with it.
- Snapshot Lifecycle: You can retain EBS snapshots for as long as needed. Over time, you can create a series of snapshots, and each snapshot represents the state of the volume at the moment it was created.
- Volumes from Snapshots**: You can create new EBS volumes from snapshots. These volumes are known as Amazon EBS volumes or Amazon Elastic Volumes. This feature is useful for creating clones of volumes, migrating data, and launching new instances with the same data.
- Data Encryption**: EBS snapshots can be encrypted, providing an extra layer of security for your backup data.
- Copying Snapshots**: You can copy snapshots across AWS regions or AWS accounts. This is helpful for disaster recovery, data replication, and sharing snapshots with other AWS accounts.
- Scheduled Snapshots**: AWS provides the ability to automate snapshot creation on a scheduled basis using AWS Lambda and Amazon CloudWatch Events.
- Snapshot Costs**: While creating EBS snapshots is relatively inexpensive, storing them incurs storage costs.

Amazon Elastic Block Store (EBS) Multi-Attach

- Amazon Elastic Block Store (EBS) Multi-Attach is a feature that allows you to attach a single EBS volume to multiple Amazon Elastic Compute Cloud (EC2) instances simultaneously.
- This capability is particularly useful for scenarios that require shared access to data across multiple EC2 instances, such as clustering, file sharing, and database replication.

Here are some key points about EBS Multi-Attach:

- Shared Data Access**: EBS Multi-Attach enables multiple EC2 instances within the same Availability Zone (AZ) to read and write to the same EBS volume simultaneously.
- Volume Types**: Multi-Attach is supported for certain types of EBS volumes, including Provisioned IOPS (io1 and io2) and General Purpose (gp2) volumes. It is not supported for Throughput Optimized (st1), Cold HDD (sc1), or Magnetic (standard) volumes.

- The performance of an EBS volume shared among multiple instances depends on the type of EBS volume and the workload.

Amazon EBS Encryption

- Amazon Elastic Block Store (EBS) provides block-level storage volumes for use with Amazon EC2 instances.
- EBS volumes can be encrypted to help protect data at rest.
- By encrypting EBS volumes, you can add an extra layer of security to your data, especially for sensitive data.
- When you create an encrypted EBS volume and attach it to an EC2 instance, the data on the volume is automatically encrypted before it is written to the underlying storage.

Here are some key points to know about EBS encryption:

- ✓ Encryption at Rest: EBS encryption ensures that data stored on the EBS volumes is encrypted at rest.
- ✓ Key Management: AWS Key Management Service (KMS) is used to manage the encryption keys.
- ✓ Enabling encryption for EBS volumes is straightforward and can be done during the volume creation process. You can also encrypt existing volumes by creating a snapshot of the unencrypted volume and then creating a new encrypted volume from that snapshot.
- ✓ Performance Impact: EBS encryption generally does not significantly impact the performance of EBS volumes. However, using encryption may add a small amount of latency and a minimal amount of additional cost.

AWS EFS

- Amazon Elastic File System (Amazon EFS) is a scalable, fully managed, cloud-based file storage service provided by Amazon Web Services (AWS).
- It provides simple, scalable file storage for use with Amazon EC2 instances in the AWS Cloud, as well as on-premises servers.

Here are some key features and benefits of Amazon EFS:

- Scalability: Amazon EFS automatically scales up or down as you add or remove files, providing a virtually unlimited storage capacity.
- Performance: It offers high throughput and low latency.
- Shared File Storage: Amazon EFS supports multiple EC2 instances, allowing them to access a shared file system concurrently.

- Fully Managed: Amazon EFS is a fully managed service, which means that AWS takes care of the underlying hardware provisioning, configuration, and maintenance tasks.
- Security: It provides robust security features, including encryption at rest and in transit.
- Integration with AWS Services: Amazon EFS integrates with other AWS services, allowing you to build scalable and reliable applications that require shared file storage.