Module – 2 -
AWS Compute & Storage Services

Introduction to AWS Compute

- AWS Compute refers to the suite of services provided by Amazon Web Services (AWS) that enable users to run applications and manage computing resources in the cloud.
- It is categorized under Infrastructure as a Service (IaaS), allowing businesses to provision and manage virtual machines, containers, and serverless computing resources.
- AWS provides many flexible computing services so as to meet the requirements of business organizations like
  - Amazon Elastic Compute Cloud (EC2),
  - Amazon Elastic Container Service (ECS),
  - Amazon Elastic Container Service for Kubernetes (EKS),
  - Amazon Lightsail,
  - AWS Lambda and many more.
- This infrastructure as a service can be considered as the processing power required by your applications, to host applications or run computation-intensive tasks.

Introduction to AWS Compute

- In AWS, with the use of these computing services, users can dynamically provision the number of resources they are using and then pay only for the computing resources they have used for.
- **This leads to the reduction of the upfront capital investment required.**
- These compute resources are closely related to regular server components like CPU and RAM.
- However, for regular server components, you need to manage and buy the infrastructure, provide for backups and emergency recovery, and ensure enough server capacity to handle traffic-intensive times.
- **With AWS compute all this headache is handed over to the AWS team.**

Amazon Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (EC2)

- Amazon Elastic Compute Cloud (EC2) is a web service that provides resizable compute capacity in the cloud.
- It allows users to run virtual servers, known as instances, on-demand.
- EC2 is designed to make web-scale cloud computing easier for developers by providing a reliable, scalable, and secure environment for hosting applications.
- Amazon EC2 reduces the time required to obtain and boot new server instances (called Amazon EC2 instances) to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.
- Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers and system administrators the tools to build failure resilient applications and isolate themselves from common failure scenarios

Key Components

- **Instances:** Virtual servers that run applications. Users can choose from various instance types optimized for different workloads.
- **Amazon Machine Images (AMIs):** Pre-configured templates that include the operating system, application server, and applications. Users can create custom AMIs to streamline deployment.
- **Elastic Block Store (EBS):** Provides persistent block storage for instances. EBS volumes can be attached to instances and are used for data storage.
- **Security Groups:** Virtual firewalls that control inbound and outbound traffic to instances. Users can define rules to allow or deny access based on IP addresses and ports.

Key Components

- **Key Pairs:** Used for secure login to instances. Users generate key pairs, consisting of a public key (stored in AWS) and a private key (downloaded by the user).
- **Elastic Load Balancing (ELB):** Distributes incoming traffic across multiple EC2 instances to ensure high availability and fault tolerance.
- **Auto Scaling**: Automatically adjusts the number of EC2 instances based on demand, ensuring that applications have the right amount of resources at all times.

Amazon EC2 instance types:

- Amazon EC2 instance types:
  - ✓ General Purpose Instances
  - ✓ Compute Optimized Instances
  - ✓ Memory Optimized Instances
  - ✓ Storage Optimized Instances
  - ✓ Accelerated Computing Instances
  - ✓ High-Performance Computing (HPC) Instances

Amazon EC2 instance types:

- **General Purpose Instances:** These provide a balance of compute, memory, and networking resources. They are suitable for a variety of applications that don't require optimization in any specific area, such as web servers, small databases, and development environments.
- **Compute Optimized Instances:** These are designed for applications that require high-performance processors, such as batch processing, high-performance computing, and machine learning workloads. They offer powerful CPUs for compute-intensive tasks.
- **Memory Optimized Instances:** These are optimized for workloads that process large datasets in memory, like high-performance databases, distributed web-scale in-memory caches, and real-time big data analytics. They provide high memory capacity.

Amazon EC2 instance types:

- **Storage Optimized Instances:** These are designed for workloads that require high, sequential read and write access to large datasets on local storage, such as distributed file systems, data warehousing, and NoSQL databases. They offer high storage performance.
- **Accelerated Computing Instances**: These utilize hardware accelerators like GPUs to perform certain functions more efficiently, such as machine learning, high-performance computing, and graphics-intensive applications.
- **High-Performance Computing (HPC) Instances:** These are specifically optimized for high-performance computing workloads, providing powerful CPUs and networking capabilities for complex simulations and scientific modeling

Quiz Time

- If you want to run Amazon.in or any e-commerce website, which instance type is best suitable?
  - Memory Optimized Instances

- If you want to run PrimeVideo or NetFlix or YouTube, which instance type is best suitable?
  - Compute Optimized Instances

- If you want to run PubG or FreeFire, which instance type is best suitable?
  - Accelerated Computing Instances.

AWS Pricing Models

AWS Pricing Models

- Amazon EC2 offers several pricing models to accommodate different usage patterns and budget considerations.
- EC2 pricing models
  - On-Demand Instances
  - Reserved Instances (RIs)
  - Spot Instances
  - Savings Plans
  - Dedicated Hosts

On-Demand Instances

- It allow users to pay for compute capacity by the hour or second, with no long-term commitments. This model provides the flexibility to launch instances as needed and terminate them when they are no longer required.
- Benefits:
  - Flexibility: Ideal for unpredictable workloads or applications that cannot be interrupted.
  - No Upfront Costs: Users pay only for what they use, making it easy to manage costs without upfront investment.
  - Scalability: Quickly scale resources up or down based on demand.
- Use Cases:
  - Development and testing environments.
  - Applications with variable workloads.
  - Short-term projects or applications with unpredictable usage.

Reserved Instances

- Reserved Instances (RIs) provide significant savings (up to 72% compared to On-Demand pricing) in exchange for a commitment to use a specific instance configuration for a one- or three-year term. RIs are not physical instances but rather a billing discount applied to the use of On-Demand Instances that match certain attributes.
- Benefits:
  - Cost Efficiency: Lower hourly rates for committed usage.
  - Capacity Reservation: Optionally reserve capacity in a specific Availability Zone, ensuring the availability of instances when needed.
  - Flexible Payment Options: Users can choose from All Upfront, Partial Upfront, or No Upfront payment options.
- Use Cases:
  - Steady-state applications with predictable usage patterns.
  - Long-term projects that require consistent compute capacity.
  - Organizations looking to optimize their cloud spending.

Spot Instances

- Spot Instances allow users to bid on unused EC2 capacity at potentially lower prices than On-Demand Instances. Spot pricing fluctuates based on supply and demand, and instances can be interrupted if the spot price exceeds the user's bid.
- Benefits:
  - Cost Savings: Users can save up to 90% compared to On-Demand pricing, making it an economical choice for flexible workloads.
  - Access to Additional Capacity: Spot Instances provide access to excess capacity that may not be available through On-Demand Instances.
- Use Cases:
  - Batch processing jobs.
  - Data analysis and big data workloads.
  - Fault-tolerant applications that can handle interruptions.

Savings Plan

- EC2 Savings Plans provide a flexible pricing model that offers low prices on EC2 usage, in exchange for a commitment to a consistent amount of usage (measured in $/hour) for a 1 or 3 year term.
  - Can reduce EC2 costs by up to 72% compared to On-Demand pricing
  - Automatically provide the discount based on your usage, regardless of instance family, size, AZ, tenancy or OS
  - Require a commitment to usage within an instance family in a region.
  - Allow you to automatically benefit from the discount when using instances from the specified family in that region
  - Provide the flexibility to change instance sizes, AZs, and even instance families within the same region
- Use Cases:
  - Workloads with consistent and steady-state usage
  - Customers who want to use different instance types across regions
  - Customers who can commit to a 1 or 3 year usage term to reduce total computing costs

Dedicated Hosts

- Dedicated Hosts are physical servers in Amazon EC2 that are reserved exclusively for your use.
  - **Cost Savings:** You can use existing software licenses (like Windows Server or SQL Server) on Dedicated Hosts, which can reduce costs.
  - **Compliance:** They help meet regulatory compliance requirementsby providing a dedicated environment.
  - **Billing**: You pay an hourly rate for the host, with no per-second billing.
  - **Capacity Reservation**: You can reserve capacity in specific Availability Zones to ensure resources are available when needed.
  - **Use Cases:** Best for applications that need a dedicated server, such as those with specific licensing or compliance requirements.

Feature	On-Demand Instances	Reserved Instances (RIs)	Spot Instances	Dedicated Hosts	Savings Plans
Description	Pay for compute capacity by the hour or second without commitment	Commit to a specific instance type for a term	Bid on unused capacity at discounted prices	Physical servers dedicated for your use	Commit to a consistent amount of usage for savings
Cost	Generally the highest cost per hour	Up to 75% savings compared to On-Demand	Up to 90% cheaper than On-Demand	On-Demand pricing for the host	Up to 72% savings compared to On-Demand
Availability	Always available, can be launched anytime	Guaranteed capacity in specific Availability Zones	Availability depends on spare capacity; can be interrupted	Dedicated capacity for your workloads	Availability based on commitment
Use Cases	Unpredictable workloads, short-term projects	Predictable workloads requiring reserved capacity	Flexible, fault-tolerant applications	Compliance and licensing requirements	Steady-state workloads across instance types
Commitment	No commitment required	Commitment to a specific instance type for 1 or 3 years	No commitment; instances can be terminated by AWS	Long-term commitment to a physical server	Commitment to usage for 1 or 3 years
Flexibility	Highly flexible; can change instance types easily	Less flexible; fixed instance type for the term	Limited flexibility; instances may be reclaimed by AWS	Limited flexibility; fixed hardware	Less flexible; tied to usage commitment
Billing	Billed based on actual usage	Billed at a discounted rate for the term	Billed based on the maximum price set by the user	Billed based on hourly usage of the host	Billed based on the committed usage

Free Tier

- AWS offers a Free Tier to allow users to try out over 100 different AWS products for free, with certain limitations.
  - Ideal For:
    - ✓ New users testing AWS services or developing proof-of-concept applications.
  - Pros:
    - ✓ Opportunity to explore AWS services without incurring costs.
  - Cons:
    - ✓ Limited usage and may not be suitable for production workloads.

Quiz Time

- What pricing model should you choose for your application if you anticipate unpredictable workloads that cannot be interrupted?
  - On-Demand Instances - This model is ideal for unpredictable workloads that cannot be interrupted, providing flexibility without long-term commitments.
- Which pricing model provides significant savings (up to 72%) for consistent and predictable workloads?
  - Savings Plans - Savings Plans offer significant discounts for consistent and predictable workloads, making them cost-effective for steady-state usage.
- If your application can tolerate interruptions and you want to maximize cost savings, which pricing model should you consider?
  - Spot Instances - Spot Instances provide the best cost savings for applications that can tolerate interruptions and have flexible start and end times.
- What is the primary benefit of using Reserved Instances for your workloads?
  - Capacity reservation in specific Availability Zones
- Which pricing model is best for organizations that have existing software licenses and want to reduce costs?
  - Dedicated Host

Create a report on the latest instance types

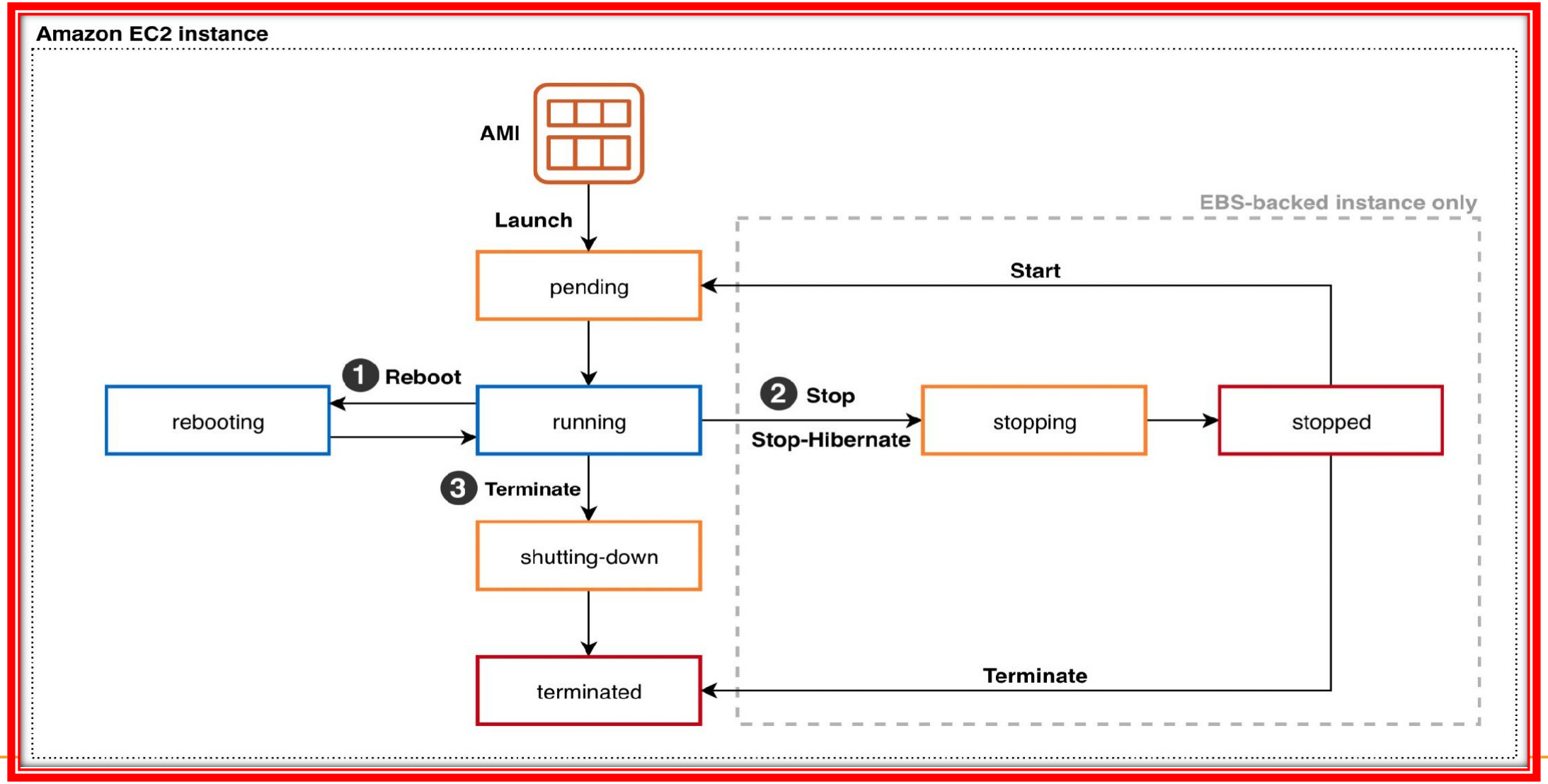
Create a report on the EC2 Pricing Models

EC2 Instance Lifecycle

EC2 Instance Lifecycle

- The lifecycle of an Amazon EC2 instance involves several distinct states that an instance transitions through, from launch to termination.
- Understanding these states is crucial for managing instances effectively and optimizing costs.
- Instance Lifecycle States
  - ✓ Pending
  - ✓ Running
  - ✓ Stopping
  - ✓ Stopped
  - ✓ Rebooting
  - ✓ Terminating
  - ✓ Retirement

EC2 Instance Lifecycle



EBS-Backed instance

- EBS-backed instances are a type of virtual server in Amazon Web Services (AWS) that use Elastic Block Store (EBS) volumes for their primary storage.
- The root volume (where the operating system is installed) is an EBS volume.
- Additional EBS volumes can be attached for extra storage.
  - You can start and stop these instances, just like turning a computer on and off.
  - When you stop it, you don't lose your data because it's stored on the EBS volume.
  - You can easily change the size of your "hard drive" if you need more space.
  - You can also detach this "hard drive" and attach it to a different virtual computer if needed.

EBS-Backed instance

- Key features of EBS-backed instances:
  - **Persistence**: Data on EBS volumes persists even when the instance is stopped. You can stop and start the instance without losing data.
  - **Flexibility**: You can change the instance type (e.g., upgrade from a small to a large instance) easily. EBS volumes can be detached from one instance and attached to another.
  - **Snapshots**: You can create point-in-time snapshots of EBS volumes for backup or replication. Snapshots are stored in Amazon S3 for durability.
  - **Encryption**: EBS volumes can be encrypted for added security.
  - **Performance**: EBS provides different volume types (e.g., General Purpose SSD, Provisioned IOPS SSD) to match performance needs.
  - **Separate lifecycle**: EBS volumes can exist independently of EC2 instances, allowing for more flexible management.

EC2 Instance Lifecycle States

- **Pending**
  - When an EC2 instance is launched, it first enters the pending state.
  - This state signifies that the instance is being prepared for use.
  - During this phase, the necessary resources are allocated, and the instance is initialized.
- **Running**
  - Once the instance is fully initialized, it transitions to the running state.
  - In this state, the instance is operational, and users can connect to it and utilize its resources.
  - Billing for the instance begins as soon as it enters the running state, with charges incurred per second, subject to a minimum of one minute, regardless of whether the instance is actively being used or is idle.

EC2 Instance Lifecycle States

- **Stopping**
  - If an EBS-backed instance needs to be temporarily halted, it can be stopped.
  - The instance transitions to the stopping state before entering the stopped state.
  - While in the stopped state, the instance does not incur usage charges, although storage costs for any attached EBS volumes still apply.
  - Users can modify certain attributes of the instance during this period, such as changing the instance type.
- **Stopped**
  - In the stopped state, the instance is effectively shut down and cannot be used until it is started again.
  - The data on the EBS volumes is preserved, but any data on instance store volumes will be lost.
  - The instance can be restarted at any time, which transitions it back to the pending state

EC2 Instance Lifecycle States

- **Rebooting**
  - Instances can be rebooted while in the running state.
  - This action keeps the instance on the same host and retains its public DNS name and private IP address.
  - Data on both EBS and instance store volumes is preserved during a reboot.
- **Terminating**
  - When an instance is no longer needed, it can be terminated.
  - This action transitions the instance through the terminating state to the terminated state.
  - In the terminated state, the instance is permanently deleted and cannot be restarted.
  - If the instance was EBS-backed, the root volume can be retained based on the termination settings, but data on instance store volumes will be lost.

EC2 Instance Lifecycle States

- **Retirement**
  - AWS may schedule an instance for retirement if it detects an irreparable failure in the underlying hardware.
  - In such cases, the instance may be stopped or terminated based on whether it is EBS-backed or instance store-backed.
  - EBS-backed instances can be restarted after being stopped, while instance store-backed instances are permanently terminated.

EC2 Instance Lifecycle States

- **Hibernation**
  - Hibernation in the EC2 instance lifecycle allows you to pause an instance and save its in-memory state to the EBS root volume.
  - This feature is particularly useful for applications that require a significant amount of time to initialize, as it enables you to resume operations without losing any progress.
  - **How Hibernation Works**
    - **State Transition:** When you hibernate an instance, it moves to the stopping state and then to the stopped state. The contents of the RAM are saved to the EBS root volume, and the instance can be resumed later.
    - **Restoration:** Upon restarting the hibernated instance, the EBS root volume is restored to its previous state, and the RAM contents are reloaded. This means that any processes that were running before hibernation can continue without needing to restart.

Quiz Time

- What state does an EC2 instance enter when it is being set up?
  - Pending
- When you temporarily shut down an EC2 instance, which state does it enter?
  - Stopping
- What happens to an EC2 instance in the terminating state?
  - It is permanently deleted
- If AWS detects a hardware issue, what state might your EC2 instance enter?
  - Retirement
- What is the billing status when an EC2 instance is in the running state?
  - Billed for usage
- What happens to data on EBS volumes when an instance is stopped?
  - Data is preserved

Summary of Instance States

Instance State	Description	Billing
Pending	Instance is being initialized	Not billed
Running	Instance is operational	Billed
Stopping	Preparing to stop the instance	Not billed
Stopped	Instance is shut down	Not billed (storage charges apply)
Rebooting	Instance is restarting	Billed (if running)
Terminating	Instance is being deleted	Not billed
Terminated	Instance is permanently deleted	Not billed

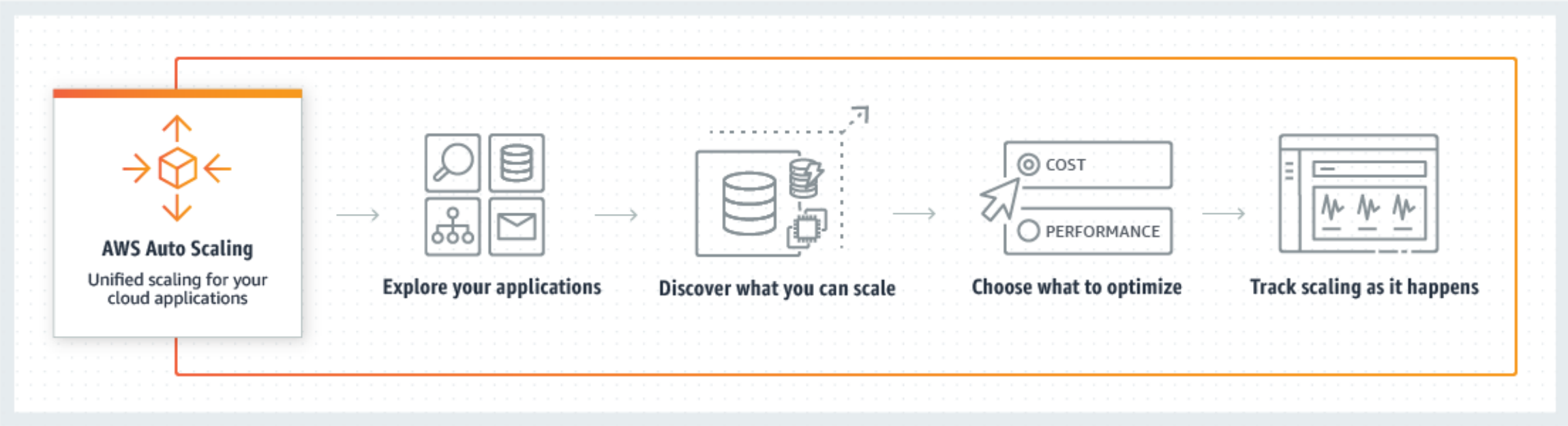
- What are the key states an EC2 instance goes through during its lifecycle?
- How does billing change as an instance transitions between different states?
- What are the implications of stopping an instance compared to terminating it?
- Why is it important to use the EC2 console for rebooting instances?
- What are the benefits of using Reserved Instances and Spot Instances?
- What steps would you take to prepare for the upcoming holiday season traffic surge?

AWS Auto Scaling

AWS Auto Scaling

- AWS Auto Scaling is a powerful service that automatically adjusts the capacity of your applications to maintain steady, predictable performance at the lowest possible cost.
- It monitors your applications and automatically adds or removes resources as needed, ensuring that your applications always have the right resources at the right time.

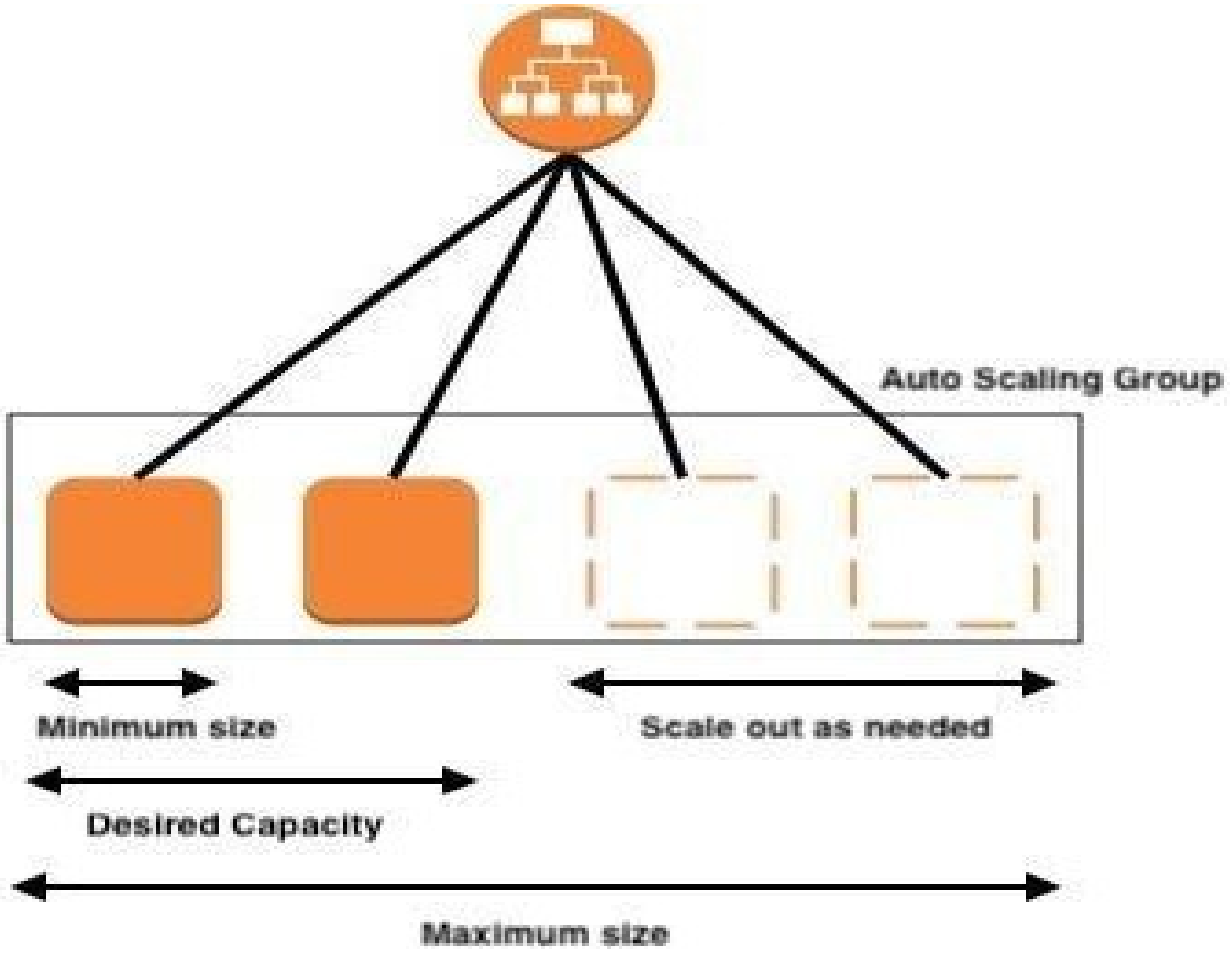
AWS Auto Scaling



Scaling Policies

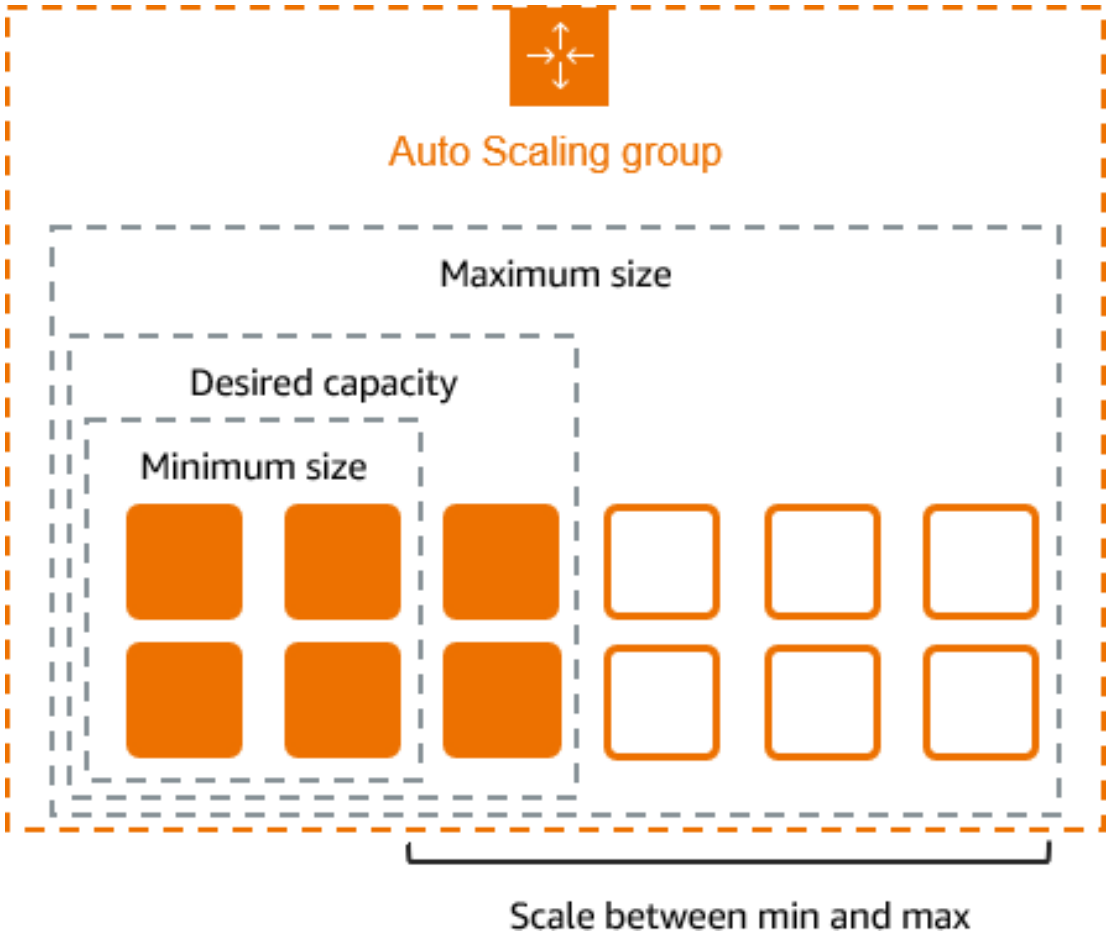
- Auto Scaling uses scaling policies to determine when and how to scale your resources.
- There are two main types of scaling policies:
  - **Dynamic Scaling:** This policy automatically scales your resources based on demand, using metrics such as CPU utilization, memory usage, or custom metrics. You can define scaling rules that specify the conditions for scaling up or down, such as scaling out when CPU utilization exceeds 80% or scaling in when CPU utilization drops below 40%.
  - **Scheduled Scaling:** This policy allows you to scale your resources based on a predetermined schedule, such as scaling out during business hours and scaling in during off-peak hours. You can define scaling actions for specific times or intervals, allowing you to proactively adjust capacity to meet expected traffic patterns.

AWS Auto Scaling



Auto Scaling Groups

- **EC2 Instances:**
  - ✓ These are the primary resources that are scaled up or down based on demand.
  - ✓ Auto Scaling groups manage a collection of EC2 instances, ensuring that the desired number of instances is running at all times.
- **Launch Configurations:**
  - ✓ These define the details of the EC2 instances to launch, such as the Amazon Machine Image (AMI), instance type, key pair, security groups, and block device mapping.



Benefits of AWS Auto Scaling

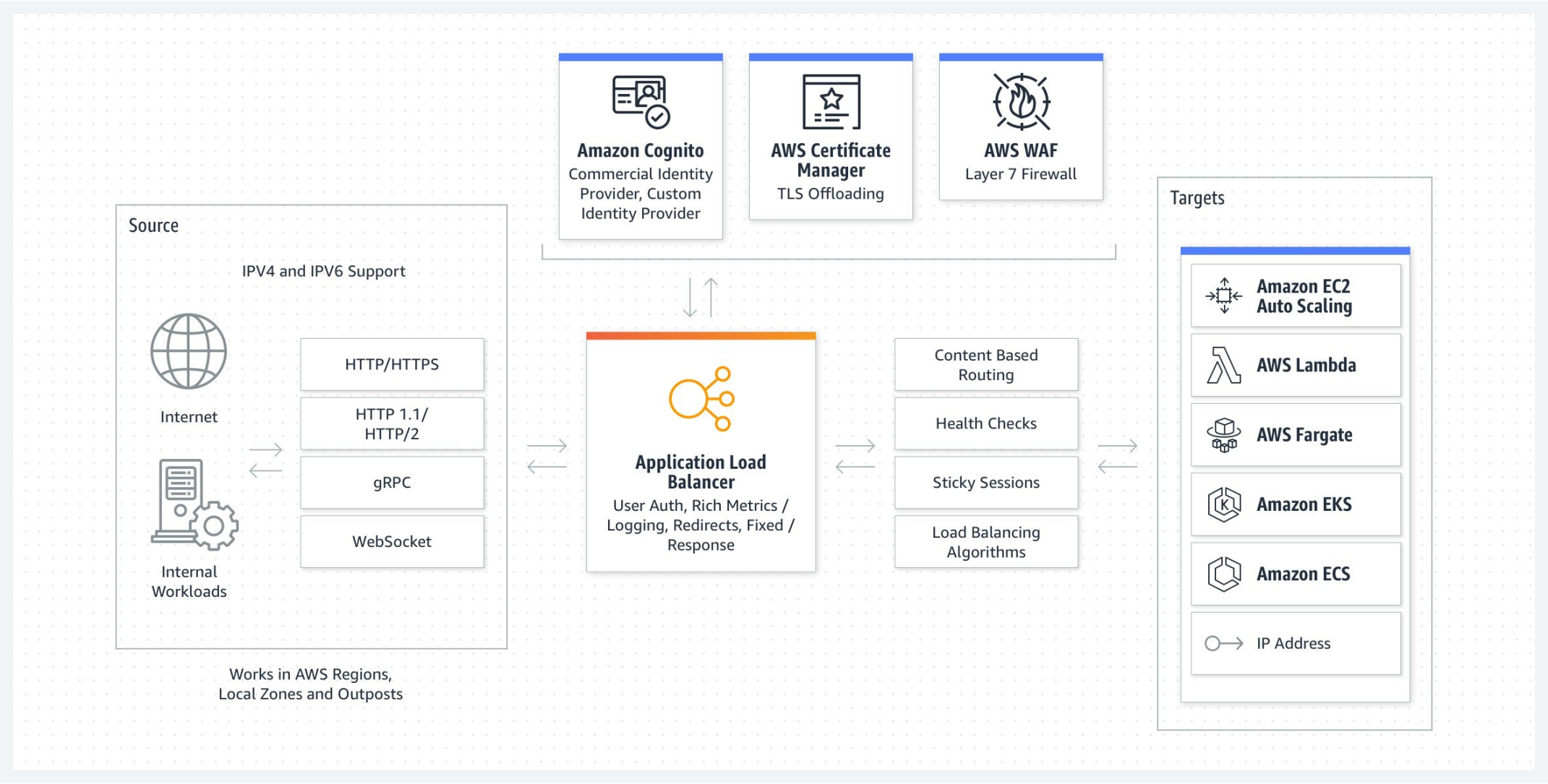
- **Improved availability:**
  - Auto Scaling helps maintain application availability by automatically adding or removing instances based on demand.
- **Cost savings:**
  - By scaling resources based on demand, you can optimize costs by running instances only when needed and scaling down during periods of low traffic.
- **Simplified management:**
  - Auto Scaling handles the complexity of managing instances, including launching, terminating, and replacing instances as needed.
- **Flexibility:**
  - Auto Scaling provides various scaling options, allowing you to choose the most appropriate scaling strategy for your application.
- **Provides smart scaling decisions:**
  - AWS Auto Scaling automatically creates scaling policies and sets targets based on your preferences, optimizing for availability, costs, or a balance of both.

Elastic Load Balancing (ELB)

Elastic Load Balancing (ELB)

- Elastic Load Balancing (ELB) is a key service in Amazon Web Services (AWS) that automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions.
- ELB ensures that your application can handle an increasing amount of traffic by automatically scaling resources up and down based on demand.
- It also checks the health of your registered targets and routes traffic only to the healthy ones.

https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/what-is-load-balancing.html



Key Features of Elastic Load Balancing

- **Automatic scaling:** ELB automatically scales your load balancer based on incoming traffic.
- **Health checks:** ELB regularly checks the health of your registered targets and routes traffic only to the healthy ones.
- **Security**: ELB supports SSL/TLS termination, allowing you to manage SSL/TLS certificates centrally.
- **Logging and monitoring:** ELB provides access logs that capture detailed information about requests sent to your load balancer and CloudWatch metrics for monitoring.
- **High availability:** ELB is designed to be highly available and fault-tolerant, with no single point of failure.

Types of Elastic Load Balancing

- **Application Load Balancer (ALB)**
  - Operates at the application layer (OSI layer 7)
  - Routes traffic based on advanced routing rules
  - Supports HTTP, HTTPS, and WebSocket protocols
  - Provides advanced features like path-based routing and host-based routing
- **Network Load Balancer (NLB)**
  - Operates at the transport layer (OSI layer 4)
  - Routes traffic based on IP protocol data
  - Supports TCP, TLS, and UDP protocols
  - Provides ultra-low latency and high throughput

Types of Elastic Load Balancing

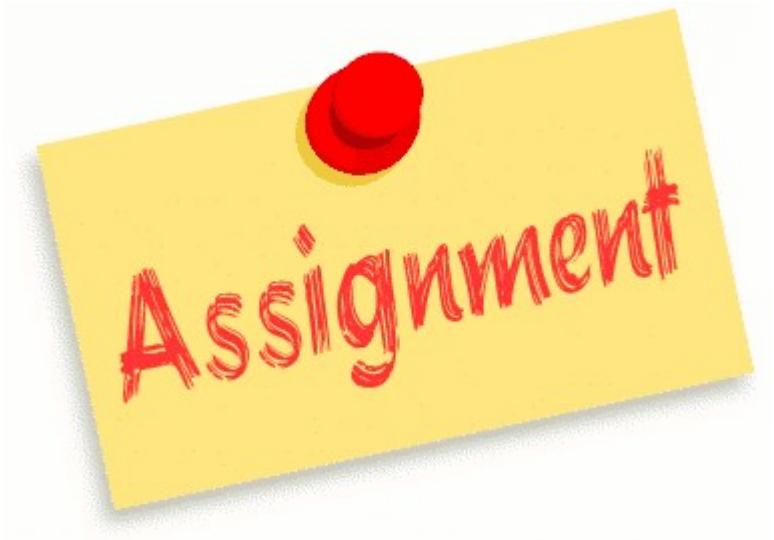
- **Classic Load Balancer (CLB)**
  - Operates at both the application layer (OSI layer 7) and transport layer (OSI layer 4)
  - Provides basic load balancing across multiple EC2 instances
  - Supports HTTP, HTTPS, TCP, and SSL protocols
  - An older type of load balancer still available for use, primarily for applications not yet migrated to the newer load balancer types
- **Gateway Load Balancer**
  - Used for deploying third-party virtual appliances, such as firewalls, intrusion detection systems, and other network appliances

Quiz Time

- Elastic Load Balancing (ELB) is used to distribute incoming traffic across multiple .
  - **Answer: targets**
- The Load Balancer operates at the application layer (OSI layer 7) and supports HTTP, HTTPS, and WebSocket protocols.
  - **Answer: Application**
- Elastic Load Balancing can be used to balance traffic across multiple .
  - **Answer: EC2 instances**
- Elastic Load Balancing can be used to the capacity of your application to handle increasing traffic.
  - **Answer: scale**
- The Load Balancer provides advanced features like path-based routing and host-based routing.
  - **Answer: Application**
- The Load Balancer is used for deploying third-party virtual appliances.
  - **Answer: Gateway**

Activity

- What are the three main types of Elastic Load Balancers in AWS?
- What layer of the OSI model does the Application Load Balancer operate on?
- Which load balancer provides ultra-low latency and high throughput?
- What is the purpose of the Gateway Load Balancer?
- Name two key features of Elastic Load Balancing.
- Describe the process of configuring Elastic Load Balancing.
- What are two best practices for using Elastic Load Balancing?



Configure an Application load balancer in AWS

AWS Storage Services

AWS Storage Services

- AWS comes up with different types of storage services for maintaining highly confidential data, frequently accessed data, and often accessed storage data.
- AWS offers various storage service types such as
  - **Object Storage as a Service(Amazon S3)**
  - Block Storage as a Service (Amazon EBS)
  - File Storage as a Service (Amazon EFS)
  - Hybrid Stroage
  - backups, and data migration options.

AWS Storage Services

- **Object Storage:** Primarily used for unstructured data, ideal for web applications, backups, and big data analytics. The main service is Amazon Simple Storage Service (S3).
- **Block Storage:** Suitable for applications requiring low-latency access, such as databases. This includes Amazon Elastic Block Store (EBS).
- **File Storage:** Designed for shared access across multiple instances, such as file systems. The primary service is Amazon Elastic File System (EFS).
- **Hybrid Storage:** Combines on-premises and cloud storage solutions, such as AWS Storage Gateway.
- **Backup and Archive:** Services like Amazon S3 Glacier for long-term data archiving.

Amazon Simple Storage Service (S3)

Amazon Simple Storage Service (S3)

- Amazon S3 is an object storage service that provides highly scalable, durable, and secure storage for any amount of data.
- It is designed to store and retrieve any type of data, making it suitable for a wide range of applications, from data lakes to backup and restore solutions.
- Each bucket can hold an unlimited number of objects, and the objects can range in size from **0 bytes to 5 Terabytes.**
- Buckets are globally unique, meaning that no two buckets can have the same name across all AWS accounts.

AWS S3: Key Features

- **Scalability:** S3 can handle virtually unlimited data storage.
- **Durability:** Offers 99.999999999% (11 nines) durability by automatically storing data across multiple devices and facilities.
- **Global Uniqueness:** Each bucket name must be unique across all AWS accounts.
- **Region Selection:** Choose an AWS region during bucket creation to affect latency, data transfer costs, and compliance.
- **Access Control:** Configure access with JSON-based bucket policies, fine-grained ACLs, and AWS IAM.
- **Versioning:** Preserve, retrieve, and restore every version of an object stored in a bucket.
- **Lifecycle Policies:** Automate object transition between storage classes based on age or criteria.
- **Logging and Monitoring:** Track requests with server access logging and monitor API calls with AWS CloudTrail.
- **Security:** Supports various security features, including encryption, access control lists (ACLs), and bucket policies.

AWS S3

- **Buckets:**
  - A bucket is a container for storing objects.
  - Each object is stored in a bucket, and buckets can be created in different regions to optimize latency and availability.
- **Objects:**
  - An object consists of the data itself (the file) and any metadatathat describes that data.
  - Metadata can include information such as the object's name, size, and creation date.

Structure of an S3 Bucket

- **Bucket Name:** Must be globally unique and follow specific naming conventions (e.g., must be between 3 and 63 characters, can contain lowercase letters, numbers, hyphens, and periods).
- **Objects:** Each object stored in a bucket consists of:
  - **Key**: A unique identifier for the object within the bucket.
  - **Value**: The actual data (the object itself).
  - **Metadata**: Information about the object, including size, content type, and custom metadata tags.
  - **Version ID (if versioning is enabled)**: A unique identifier for each version of an object.

S3 Buckets & Objects Limitations

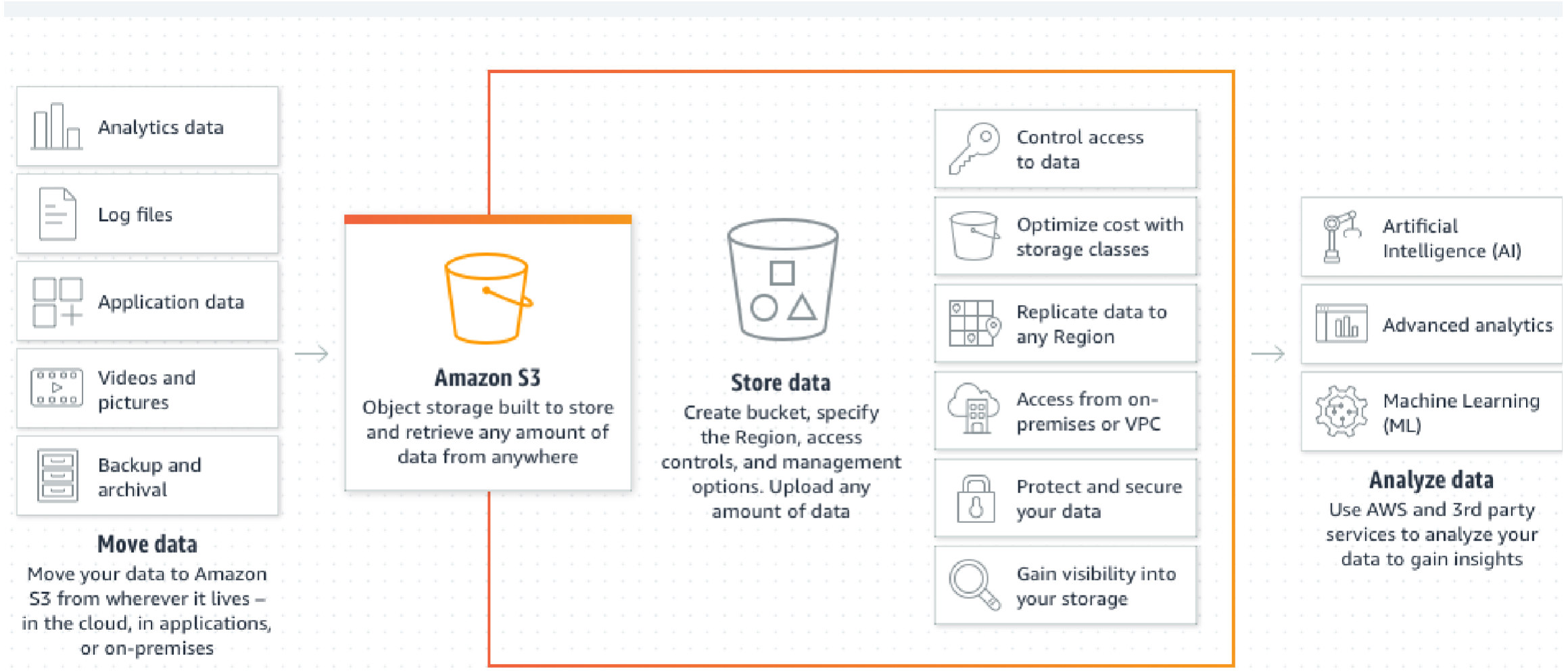
- **Bucket Limitations**
  - Each AWS account can create up to 100 buckets by default. However, you can request a service limit increase to create more buckets.
  - Bucket names must be unique across all existing bucket names in Amazon S3. Once a bucket is deleted, it may take some time before the name can be reused.
  - You cannot create a bucket inside another bucket.
- **Object Limitations**
  - Individual Amazon S3 objects can range in size from 0 bytes to a maximum of 5 TB.
  - The largest object that can be uploaded in a single PUT is 5 GB. For larger objects, you should use the multipart upload capability.

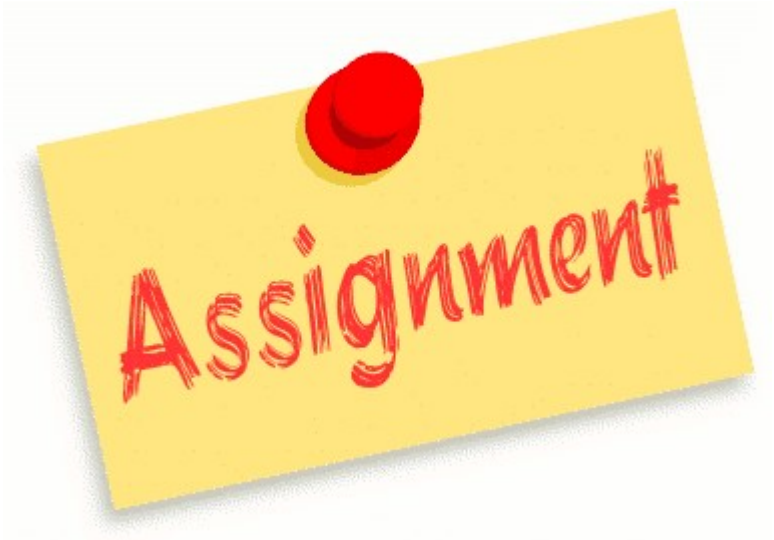
How S3 Works

- **Creating a Bucket:**
  - Before storing data, you must create a bucket.
  - You can specify the region where the bucket will be created, which can affect latency and availability.
- **Uploading Objects:**
  - Once a bucket is created, you can upload objects to it.
  - You can set permissions for each object, controlling who can access it.
- **Managing Objects:**
  - After uploading, you can manage your objects by modifying permissions, moving them between buckets, or deleting them when no longer needed.

S3 Use Cases

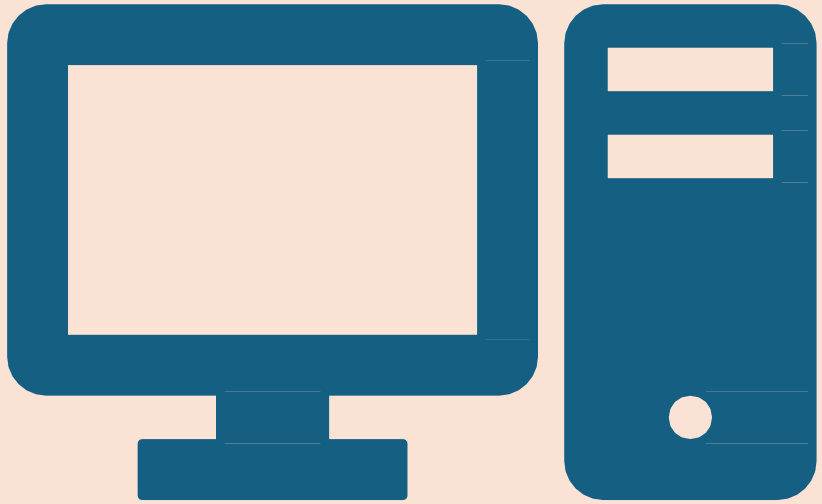
- **Data Storage:** Amazon s3 acts as the best option for scaling both small and large storage applications. It helps in storing and retrieving the data-intensitive applications as per needs in ideal time.
- **Backup and Recovery:** Many Organizations are using Amazon S3 to backup their critical data and maintain the data durability and availability for recovery needs.
- **Hosting Static Websites:** Amazon S3 facilitates in storing HTML, CSS and other web content from Users/developers allowing them for hosting Static Websites benefiting with low-latency access and cost-effectiveness.
- **Data Archiving:** Amazon S3 Glacier service integration helps as a cost-effective solution for long-term data storing which are less frequently accessed applications.
- **Big Data Analytics:** Amazon S3 is often considered as data lake because of its capacity to store large amounts of both structured and unstructured data offering seamless integration with other AWS Analytics and AWS Machine Learning Services.


Analytics data
Log files
Application data
Videos and pictures
Backup and archival
Move data
Move your data to Amazon S3 from wherever it lives – in the cloud, in applications, or on-premises
Amazon S3
Object storage built to store and retrieve any amount of data from anywhere
Store data
Create bucket, specify the Region, access controls, and management options. Upload any amount of data
Control access to data
Optimize cost with storage classes
Replicate data to any Region
Access from on-premises or VPC
Protect and secure your data
Gain visibility into your storage
Artificial Intelligence (AI)
Advanced analytics
Machine Learning (ML)
Analyze data
Use AWS and 3rd party services to analyze your data to gain insights



Create a S3 bucket for storing your app data and link it to the EC2

S3 Storage Classes



S3 Storage Classes

- AWS S3 provides multiple storage types that offer different performance and features and different cost structures.
- The list of AWS S3 storage classes includes:
  - ✓ S3 Standard
  - ✓ S3 Intelligent-Tiering
  - ✓ S3 Standard-IA (Infrequent Access)
  - ✓ S3 One Zone-IA
  - ✓ S3 Glacier
  - ✓ S3 Glacier Deep Archive
  - ✓ S3 Outposts

S3 Standard

- The S3 Standard storage class is designed for frequently accessed data that requires low latency and high throughput.
- It is the default storage class for Amazon S3 and is suitable for a wide range of use cases.
- **Durability:** S3 Standard provides 99.999999999% durability, meaning that your data is extremely safe and unlikely to be lost.
- **Availability:** It offers 99.99% availability, ensuring that your data is accessible when you need it.
- **Performance:** This class supports large-scale applications with high request rates, making it suitable for websites, mobile apps, and data analytics.
- **Use Cases:**
  - **Web Hosting:** Hosting static websites where quick access to files is essential.
  - **Big Data Analytics**: Storing data for analytics and machine learning applications that require frequent access to datasets.
  - **Content Distribution:** Delivering media files, images, and videos to users around the globe.

S3 Standard - Limitations

- **Higher Costs:**
  - S3 Standard is the most expensive storage class, making it less suitable for data that is infrequently accessed.
- **Potential Overuse:**
  - Organizations may inadvertently use S3 Standard for data that could be stored in a lower-cost class, leading to unnecessary expenses.

S3 Intelligent-Tiering

- S3 Intelligent-Tiering automatically moves data between two access tiers: frequent and infrequent access, based on changing access patterns.
- This class is particularly useful when you cannot predict how often your data will be accessed.
  - **Durability:** It maintains the same high durability of 99.999999999%.
  - **Availability:** Offers 99.9% availability, ensuring reliable access to your data.
  - **Cost Structure:** There are no retrieval fees for frequently accessed data, while infrequently accessed data incurs a small monthly monitoring and automation fee.
  - **Use Cases:**
    - ✓ **Data with Unpredictable Access Patterns:** Ideal for datasets that may have spikes in access or may become less frequently accessed over time.
    - ✓ **Data Lakes:** Storing diverse datasets where the access frequency is uncertain.

S3 Intelligent-Tiering - Limitations

- **Monitoring and Automation Fees:**
  - While S3 Intelligent-Tiering can optimize costs, it incurs a small monthly fee for monitoring and automatic tiering, which may not be cost-effective for small datasets.
- **Potential Delays:**
  - The automatic tiering process may introduce slight delays in accessing data that has been moved to the infrequent access tier.

S3 Standard-Infrequent Access (S3 Standard-IA)

- This class is designed for long-lived but infrequently accessed data.
- It offers lower storage costs compared to S3 Standard but incurs retrieval fees.
- **Durability:** Provides 99.999999999% durability, ensuring data safety.
- **Availability**: 99.9% availability, making it reliable for infrequent access.
- **Cost Structure**: Lower storage costs than S3 Standard, but retrieval costs apply when accessing data.
- **Use Cases:**
  - **Backups:** Storing backups of critical data that are not accessed regularly.
  - **Disaster Recovery**: Keeping copies of essential data that can be restored in case of data loss.
  - **Long-Term Storage**: Archiving data that must be retained for compliance but is not frequently accessed.

S3 Standard-IA - Limitations

- **Retrieval Fees:**
  - S3 Standard-IA incurs retrieval fees, which can add to the overall cost if data is accessed frequently. It's essential to carefully consider access patterns to avoid unexpected costs.
- **Higher Minimum Object Size:**
  - S3 Standard-IA has a minimum object size of 128 KB, making it less suitable for storing small objects.

S3 One Zone-Infrequent Access (S3 One Zone-IA)

- This class is similar to S3 Standard-IA but stores data in a single Availability Zone rather than multiple zones.
- This makes it a lower-cost option for infrequently accessed data.
- **Durability:** Maintains 99.999999999% durability.
- **Availability**: Offers 99.5% availability, which is lower than other classes due to its single Availability Zone architecture.
- **Cost Structure**: Significantly cheaper than S3 Standard-IA, but with less redundancy.
- **Use Cases:**
  - **Non-Critical Data**: Storing infrequently accessed data that can be recreated if lost.
  - **Temporary Data:** Data that is only needed for a limited time and can be easily regenerated.

S3 One Zone-IA - Limitations

- **Lower Durability:**
  - S3 One Zone-IA stores data in a single Availability Zone, which means it has lower durability compared to multi-zone storage classes. Data may be lost if the Availability Zone becomes unavailable.
- **Lower Availability:**
  - With 99.5% availability, S3 One Zone-IA has a higher likelihood of experiencing service interruptions compared to other storage classes.

S3 Glacier Instant Retrieval

- This storage class is designed for archival data that requires immediate access.
- It combines the low cost of archival storage with the ability to retrieve data instantly.
- **Durability**: Offers 99.999999999% durability.
- **Availability**: 99.9% availability, ensuring that data is accessible when required.
- **Cost Structure:** Lower storage costs compared to standard classes, with no retrieval fees.
- **Use Cases:**
  - **Immediate Access Archives:** Storing data that is rarely accessed but must be available instantly, such as compliance records.
  - **Media Archives:** Archiving media files that may need to be accessed quickly for production or legal reasons.

S3 Glacier Instant Retrieval - Limitations

- **Minimum Storage Duration:**
  - S3 Glacier Instant Retrieval has a minimum storage duration of 90 days.
  - If objects are deleted or transitioned to another storage class before this period, early deletion fees may apply.

S3 Glacier Flexible Retrieval

- This class is for long-term archival storage where retrieval times can range from minutes to hours.
- It offers a cost-effective way to store large amounts of data that are rarely accessed.
- **Durability**: Maintains the same high durability of 99.999999999%.
- **Cost Structure:** Very low storage costs, but retrieval fees apply based on the speed of retrieval (expedited, standard, or bulk).
- **Use Cases:**
  - **Long-Term Data Archiving:** Storing data that needs to be retained for years but is not expected to be accessed frequently.
  - **Regulatory Compliance:** Keeping records for compliance with regulations that require long-term data retention.

S3 Glacier Flexible Retrieval - Limitations

- **Retrieval Times:**
  - While S3 Glacier Flexible Retrieval offers different retrieval options, the standard retrieval time can take several hours, which may not be suitable for time-sensitive data.
- **Retrieval Fees:**
  - Retrieval fees apply based on the selected retrieval option, and expedited retrieval can be costly.

S3 Glacier Deep Archive

- This is the lowest-cost storage class for long-term archival data. It is designed for data that is rarely accessed and can tolerate retrieval times of 12 hours or more.
- **Cost Structure**: Extremely low storage costs, with retrieval fees based on the speed of access.
- **Use Cases:**
  - **Historical Data**: Storing data that is rarely accessed but must be kept for historical or legal reasons.
  - **Backup Archives**: Long-term backups of data that can be recreated if necessary
- **Limitations:**
  - **Retrieval Times:**
    - S3 Glacier Deep Archive has the longest retrieval time, up to 12 hours, making it unsuitable for data that requires immediate access.
  - **Retrieval Fees:**
    - Retrieval fees apply, and the cost can be significant if data is accessed frequently.

S3 Outposts

- S3 Outposts brings S3 storage to on-premises environments using AWS Outposts.
- This allows organizations to store data locally while still benefiting from S3's features.
- **Use Cases:**
  - **Local Data Residency:** Organizations that require data to remain on-premises for compliance or regulatory reasons.
  - **Hybrid Cloud Applications:** Applications that need to operate in both cloud and on-premises environments.
- **Limitations:**
  - **Limited Availability:** S3 Outposts is currently only available in certain AWS Regions and may not be accessible in all locations.
  - **On-Premises Infrastructure**: Implementing S3 Outposts requires on-premises infrastructure, which can be more complex and costly to manage compared to fully cloud-based solutions.

S3 Lifecycle Policies

S3 Lifecycle Policies

- Amazon S3 Lifecycle Policies are a powerful feature that allows you to automate the management of your objects stored in S3 buckets.
- By defining a set of rules, you can control how and when your data transitions between different storage classes or is deleted, optimizing both cost and data management practices.
- S3 Lifecycle Policies are a set of rules that define actions to be taken on objects in an S3 bucket over time.
- These actions can include:
  - **Transition Actions:** Automatically moving objects to different storage classes based on specified criteria. For example, you can set a rule to transition objects from S3 Standard to S3 Standard-IA after 30 days.
  - **Expiration Actions:** Automatically deleting objects after a specified period. For instance, you might want to delete temporary files after 90 days.

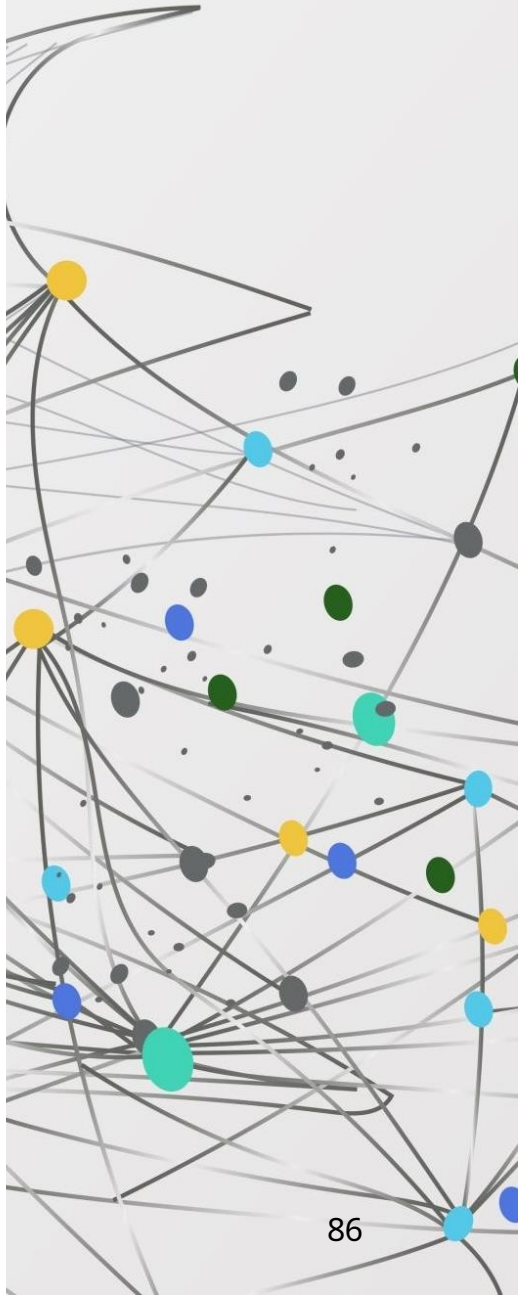
S3 Lifecycle Policies

- To create a Lifecycle Policy, you need to define one or more rules that specify the following:
  - **Filter**: Determines which objects the rule applies to, based on a prefix, tag, or a combination of both.
  - **Transition Actions:** Specifies the storage class to transition to and the number of days after object creation to perform the transition.
  - **Expiration Actions:** Defines the number of days after object creation to delete the object.

Benefits of Using Lifecycle Policies

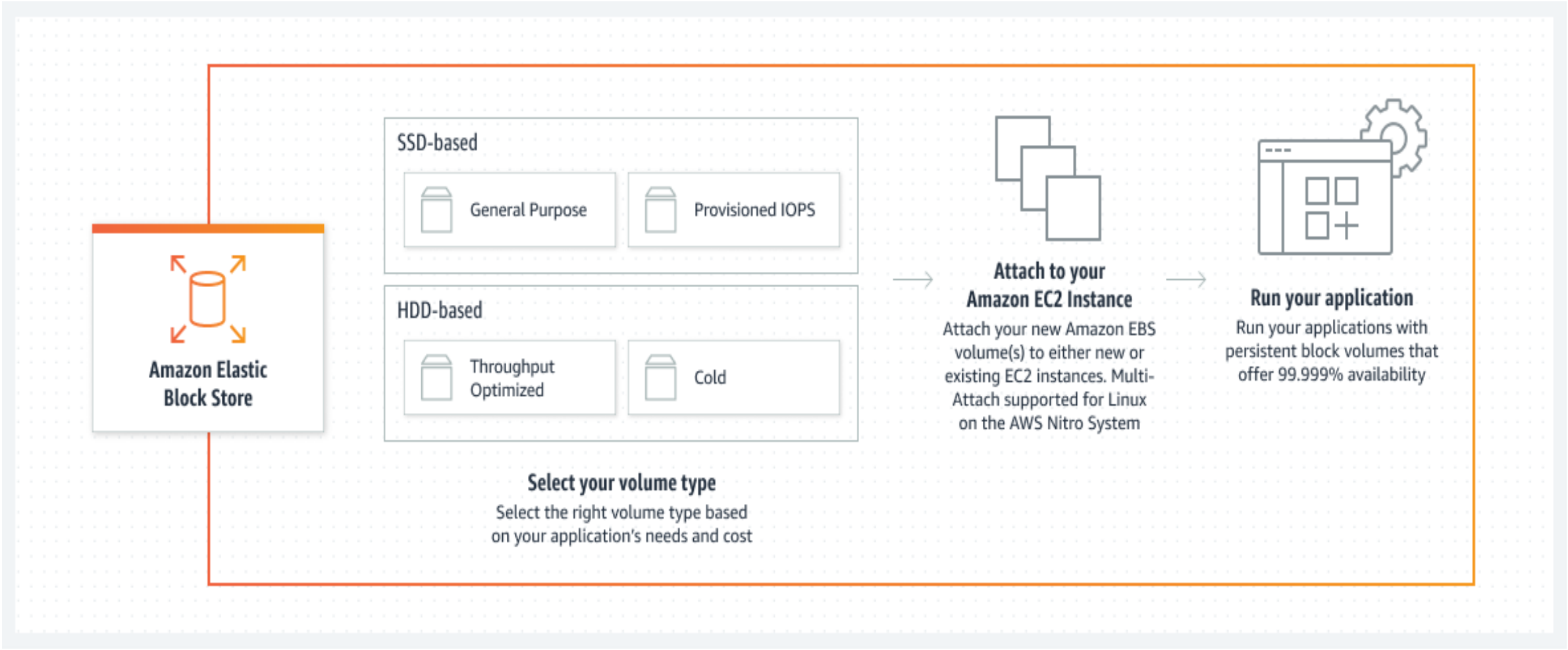
- **Cost Optimization:**
  - By automatically transitioning objects to less expensive storage classes or deleting them when they are no longer needed, organizations can significantly reduce their storage costs.
- **Data Management**:
  - Lifecycle policies help maintain a clean and organized storage environment by ensuring that only relevant data is retained, which is crucial for compliance and operational efficiency.
- **Automation**:
  - Once set up, lifecycle policies operate automatically without the need for manual intervention, saving time and reducing the risk of human error.

Amazon EBS
(Elastic Block Store)



Amazon Elastic Block Store (EBS)

- Amazon EBS is a block-level storage service that allows users to create storage volumes and attach them to EC2 instances.
- Unlike traditional file storage, EBS volumes behave like raw, unformatted block devices, which means they can be used to create file systems, run databases, or store application data.
- EBS volumes are designed to be highly available and durable, ensuring that data remains intact even when the associated EC2 instances are stopped or terminated.
- It provides scalable, durable, and reliable storage volumes that can be attached to EC2 instances and used like physical hard drives.
- EBS volumes are Network-attached storage, meaning they are not physically attached to the EC2 instances but rather accessed over the network. This allows for greater flexibility and scalability compared to local instance storage


Amazon Elastic Block Store
SSD-based
General Purpose
Provisioned IOPS
HDD-based
Throughput Optimized
Cold
Select your volume type
Select the right volume type based on your application's needs and cost
Attach to your Amazon EC2 Instance
Attach your new Amazon EBS volume(s) to either new or existing EC2 instances. Multi-Attach supported for Linux on the AWS Nitro System
Run your application
Run your applications with persistent block volumes that offer 99.999% availability

Amazon Elastic Block Store (EBS)

- **Scalability**: EBS volumes can be dynamically resized, and their performance can be tuned to meet the needs of your applications.
- **Durability**: EBS volumes are replicated within an Availability Zone to protect against component failure, with some volume types offering 99.999% durability.
- **Encryption**: EBS offers encryption at rest, with keys managed by the AWS Key Management Service (KMS), to protect sensitive data.
- **Snapshots**: EBS supports creating point-in-time backups called snapshots, which can be used to restore volumes or migrate data across AWS Regions and accounts.
- **Multiple volume types**: EBS provides a range of volume types optimized for different workloads, including SSD-backed volumes for transactional workloads and HDD-backed volumes for throughput-intensive workloads.

IOPS and Throughput

- **Input/Output Operations Per Second (IOPS)**
  - Measures the speed and efficiency of storage devices based on their ability to handle input/output operations.
  - **Performance Indicator:** A higher IOPS value indicates better performance and faster data access, making it essential for data-intensive applications.
- **Throughput**
  - Throughput measures the amount of data that can be processed by a storage system in a given amount of time, typically expressed in bytes per second (Bps) or megabytes per second (MB/s).
  - Throughput = (Read IOPS + Write IOPS) × Block Size
  - Throughput is influenced by both IOPS and the size of the data blocks being processed.

EBS Volume Types

- Amazon Elastic Block Store (EBS) provides several volume types optimized for different workloads and performance requirements.
- These volume types are broadly categorized into two main groups:
  - **Solid-State Drive (SSD)-backed volumes:**
    - Optimized for transactional workloads with frequent read/write operations and small I/O size.
    - The dominant performance attribute is IOPS.
  - **Hard Disk Drive (HDD)-backed volumes:**
    - Optimized for large streaming workloads where throughput (measured in MB/s) is more important than IOPS.

IOPS - Input/output Operations Per Second

EBS Volume Types - General Purpose SSD (gp2)

- The gp2 volume type is designed for a broad range of workloads, offering a balance of price and performance.
- It is suitable for applications that require moderate I/O performance and is widely used for boot volumes, small to medium-sized databases, and development environments
- Performance:
  - Provides 3 IOPS per GB of volume size.
  - Supports burst performance up to **3,000 IOPS** for volumes smaller than 1,000 GB, allowing for temporary spikes in performance.
- Capacity:
  - Available in sizes ranging from 1 GB to 16 TB.
- Use Cases:
  - Boot volumes for EC2 instances.
  - Development and test environments.
  - Small to medium-sized relational databases.

EBS Volume Types - General Purpose SSD (gp3)

- The gp3 volume type is the next generation of general-purpose SSD volumes, offering enhanced performance and cost efficiency compared to gp2.
- It allows users to independently configure IOPS and throughput, providing greater flexibility for various workloads.
- Performance:
  - Baseline performance of **3,000 IOPS**, which can be provisioned up to **16,000 IOPS.**
  - Throughput of 125 MiB/s, which can be increased to 1,000 MiB/s, independent of volume size.
- Capacity:
  - Supports volumes ranging from 1 GB to 64 TB.
- Use Cases:
  - Large databases requiring consistent performance.
  - Applications with variable workloads that need to scale IOPS and throughput dynamically.
  - Workloads that require higher performance without the need for larger volume sizes.

EBS Volume Types - **Provisioned IOPS SSD (io1)**

- Provisioned IOPS SSD volumes are designed for I/O-intensive applications that require consistent and high performance. These volumes are ideal for workloads such as large relational databases and mission-critical applications.
- Use Cases:
  - Best suited for applications that require high IOPS and low latency, such as Oracle, Microsoft SQL Server, and SAP HANA.
- Performance:
  - Can be provisioned up to **64,000 IOPS** and **1,000 MB/s** of throughput.
  - Offers a ratio of **30 IOPS per GB** of volume size, allowing users to tailor performance to their specific needs.
- Durability:
  - Designed for high availability and durability, making it suitable for production environments.

EBS Volume Types - Provisioned IOPS SSD (io2)

- Provisioned IOPS SSD volumes are designed for I/O-intensive applications that require consistent and high performance. These volumes are ideal for workloads such as large relational databases and mission-critical applications.
- **Use Cases:** Ideal for the most demanding applications that require sustained performance and high durability.
- **Performance:**
  - Supports up to **256,000 IOPS and 4,000 MB/s** throughput, significantly enhancing performance capabilities.
  - Offers a higher durability level with a 99.999% durability rating.
  - Allows provisioning of up to **500 IOPS per GB**, giving users more flexibility in performance management.
- **Block Express:**
  - io2 Block Express is the latest generation of io2, providing enhanced performance and scalability for enterprise applications.

EBS Volume Types - Throughput Optimized HDD (st1)

- Throughput Optimized HDD volumes are designed for workloads that require high throughput rather than high IOPS. These volumes are a cost-effective solution for applications that involve large, sequential I/O operations.
- Use Cases:
  - Suitable for big data analytics, data warehouses, and log processing applications where high throughput is essential.
- Performance:
  - Offers a maximum throughput of **500 MB/s.**
  - Supports up to **500 IOPS**, making it efficient for workloads that do not require rapid random access but benefit from sustained throughput.
- Cost Efficiency:
  - Generally more affordable than SSD options, making it a popular choice for less critical workloads that still require solid performance.

EBS Volume Types - Cold HDD (sc1)

- Cold HDD volumes are optimized for infrequently accessed data and are the most cost-effective EBS volume type. They are designed for use cases where performance is less critical.
- Use Cases:
  - Ideal for data archiving, backup solutions, and scenarios where data is rarely accessed.
- Performance:
  - Provides a maximum throughput of **250 MB/s.**
  - Supports up to **250 IOPS**, making it suitable for workloads that can tolerate lower performance levels.
- Cost Efficiency:
  - Offers the lowest cost per gigabyte of all EBS volume types, making it an excellent choice for long-term storage of data that does not require frequent access.

EBS Volume

Volume Type	Description	Use Cases
Provisioned IOPS SSD (io1/io2)	Highest performance SSD volumes for mission-critical applications	Latency-sensitive transactional databases, enterprise applications
General Purpose SSD (gp2/gp3)	Balanced performance and cost for a wide range of workloads	Boot volumes, low-latency interactive apps, development and test environments
Throughput Optimized HDD (st1)	Low-cost HDD volumes designed for frequently accessed, throughput-intensive workloads	Big data, data warehouses, log processing
Cold HDD (sc1)	Lowest cost HDD volumes for infrequently accessed workloads	Scenarios where the lowest storage cost is important

EBS Snapshots

- EBS snapshots are point-in-time backups of EBS volumes that are stored in Amazon S3.
- Snapshots are incremental, meaning only the blocks of data that have changed since the last snapshot are stored, saving on storage costs.
- You can create snapshots manually or automate the process using the Amazon Data Lifecycle Manager.
- They provide a way to protect your data and enable various data management tasks:
  - **Backup and restore:** Snapshots can be used to restore volumes or create new volumes, protecting against data loss.
  - **Migration**: Snapshots can be copied across AWS Regions and accounts, allowing you to migrate data.
  - **Disaster recovery**: Snapshots stored in multiple Availability Zones provide a reliable way to recover from failures.
  - **Compliance**: Snapshots can be retained for long periods to meet regulatory and compliance requirements.

EBS Pricing

- **Provisioned capacity:**
  - You pay for the size of the EBS volume, regardless of actual usage.
- **IOPS:**
  - For io1/io2 volumes, you pay for the provisioned IOPS.
- **Snapshots**:
  - You pay for the amount of data stored in snapshots, based on the Amazon S3 pricing.

Amazon Glacier

Amazon Glacier

- Amazon Glacier, is a low-cost cloud storage service specifically designed for long-term data archiving and backup.
- It is ideal for storing "cold data," which refers to data that is infrequently accessed but needs to be retained for future use.
- This makes it ideal for storing data that you don't need to access often but want to keep for compliance, historical analysis, or disaster recovery purposes.
- S3 Glacier offers three retrieval options:
  - **Expedited**: Fastest option, retrieving data in 1-5 minutes, but most expensive.
  - **Standard**: Default option, retrieving data in 3-5 hours.
  - **Bulk**: Least expensive option, retrieving large amounts of data in 5-12 hours.

Amazon Glacier:

- Data in S3 Glacier is organized into ”**Archives**" and ”**Vaults**."
  - An **archive** is a single file or a collection of files that you store as a single unit.
  - A **vault** is a container for storing archives, similar to an S3 bucket. You can set access policies on vaults to control who can access the data and what actions they can perform.
- S3 Glacier integrates seamlessly with other AWS services.
- For instance, you can set up lifecycle policies in S3 to automatically move data to Glacier after a certain period, helping you optimize storage costs.
- It also supports server-side encryption for data at rest and in transit, ensuring your archived data remains secure.

Amazon Glacier:

- When using S3 Glacier, it's crucial to understand its pricing model.
- While storage is very cheap, retrieval can be costly if not managed properly.
- You're charged for storage, data transfer out, and retrieval requests.
- There's also a minimum storage duration charge of 90 days, meaning if you delete data before 90 days, you'll still be charged for the full 90 days.

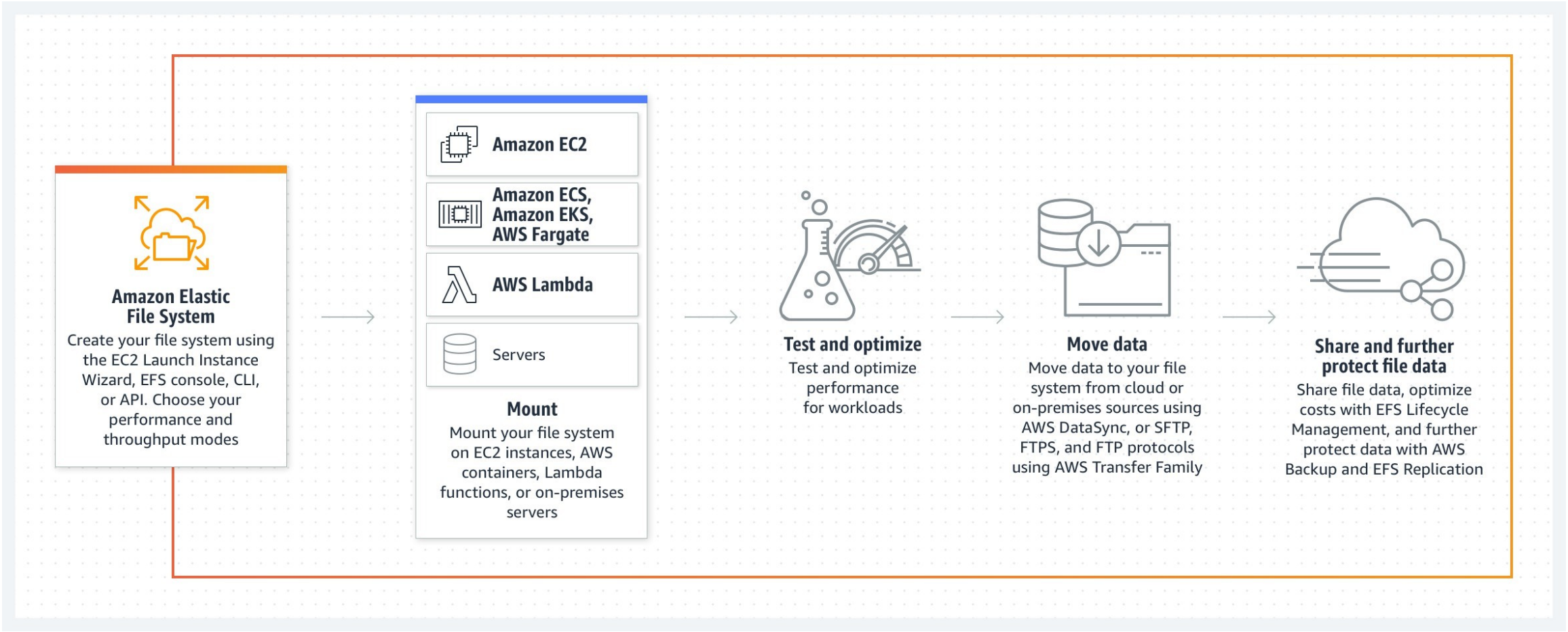
Enlist the scenarios where S3 Glacier can be used

Amazon EFS (Elastic File System)

Amazon EFS (Elastic File System)

- Amazon EFS is a fully managed, scalable, and highly available file storage service designed for use with Amazon EC2 instances and other AWS services.
- EFS is particularly beneficial for applications that require shared access to file systems, allowing multiple Amazon Elastic Compute Cloud (EC2) instances to access the same data simultaneously.
- It provides a simple, serverless, and elastic file system that automatically grows and shrinks as you add and remove files, eliminating the need to provision and manage capacity.
- This capability makes EFS ideal for a variety of workloads, including content management systems, big data analytics, and web serving.

Amazon EFS (Elastic File System)



Amazon EFS (Elastic File System)

- EFS is built to scale on demand to petabytes without disrupting applications.
- This elasticity is one of its key features, allowing it to accommodate growing datasets without the need for manual intervention. As your storage needs increase, EFS automatically allocates more space, and when files are deleted, it automatically reduces the allocated space.
- One of the most significant advantages of EFS is its support for concurrent access by thousands of EC2 instances.
- This makes it ideal for big data and analytics, media processing workflows, content management, web serving, and home directories.
- Multiple EC2 instances can access an EFS file system at the same time, allowing for easy sharing of files across a distributed application.
- EFS uses the Network File System version 4 (NFSv4) protocol, which is widely supported and allows for easy integration with existing applications and workflows.
- This compatibility means that you can mount EFS file systems on EC2 instances running Linux, making it feel like a local file system to your applications.

How Amazon EFS Works

- EFS uses the Network File System (NFS) protocol, allowing files to be accessed and shared across multiple EC2 instances and on-premises servers.
- Users can create and manage EFS file systems through the AWS Management Console, AWS CLI, or AWS SDKs.
  - **Mount Targets:** To access EFS, you need to create mount targets in your VPC (Virtual Private Cloud). These targets allow EC2 instances to connect to the EFS file system.
  - **Access Control:** Access to EFS is managed through AWS Identity and Access Management (IAM) and can be restricted using VPC security groups, ensuring that only authorized users and applications can access the file system

Amazon EFS (Elastic File System)

- EFS provides **two performance** modes:
  1. **General Purpose:** Suitable for most applications, offering low latency and high IOPS.
  2. **Max I/O:** Designed for highly parallelized workloads that require higher throughput and IOPS.
- EFS also offers **two throughput** modes:
  1. **Bursting Throughput:** The default mode, which scales as your file system grows.
  2. **Provisioned Throughput:** Allows you to specify the throughput of your file system independent of its size.

IOPS - Input/output Operations Per Second

Advantages and Limitations

- **Advantages:**
  - **Elastic and Scalable:** EFS automatically adjusts storage capacity based on usage, eliminating the need for manual provisioning.
  - **Shared Access:** Multiple EC2 instances can access the same file system concurrently, facilitating collaboration.
  - **Durability and Security:** Data is stored across multiple AZs, ensuring high durability and availability.
- **Limitations:**
  - **No Windows Support:** EFS is not compatible with Windows-based EC2 instances, as it only supports Linux instances that can use NFS.
  - **File Size and Throughput Limits**: EFS has a maximum file size of 47.9 TB and a maximum throughput of 1,000 MB/s per file system.

When To Choose Amazon EFS?

- **Shared Access:** Use EFS when multiple Amazon EC2 instances or on-premises servers need to access the same files at the same time. This is ideal for applications like web hosting and content management systems.
- **Scalability**: EFS automatically grows and shrinks storage capacity based on your needs. If your data usage varies or spikes, EFS can handle it without requiring manual adjustments.
- **Performance Needs:** EFS offers high throughput and low-latency access, making it suitable for applications that need fast file operations, such as big data analytics and media processing.
- **POSIX Compliance:** If your application requires traditional file system features like file locking and strong consistency, EFS supports POSIX file system semantics.
- **Integration with AWS Services:** EFS works well with other AWS services, such as AWS Lambda and Amazon ECS, making it easy to share data across different applications.
- **Cost Management:** With EFS, you only pay for the storage you use. You can also automatically move less frequently accessed files to a cheaper storage class to save money.

Use Cases for Amazon EFS

- **Web Hosting:**
  - Ideal for websites where multiple servers need to access the same files, like images and scripts.
- **Big Data and Analytics**:
  - Suitable for applications that require quick access to large datasets for processing.
- **Media Processing:**
  - Great for workflows that involve handling large media files efficiently.
- **Development and Testing:**
  - Provides a shared space for development teams to collaborate and share code easily.

Feature	Amazon EFS	Amazon EBS	Amazon S3
Type	File storage	Block storage	Object storage
Access	Multiple EC2 instances across AZs	Single EC2 instance in same AZ	Any device with internet access
Use Cases	Shared file systems, content management, web serving	Boot volumes, databases, apps needing high I/O	Static website hosting, data lakes, backup/archiving
Scalability	Automatically scales to petabytes	Fixed size, manual scaling	Virtually unlimited
Performance	Varies based on total size	Consistent, predictable	High throughput, variable latency
Durability	99.999999999% (11 9's)	99.999%	99.999999999% (11 9's)
Availability	Multi-AZ by default	Single AZ (unless using Multi-Attach)	99.99%
Protocol	NFS	Block-level	RESTful API
Lifecycle Management	Yes (for Infrequent Access)	No	Yes
Encryption	At-rest and in-transit	At-rest	At-rest and in-transit
Snapshots	No native snapshots	Yes	Versioning
Cost Model	Pay for used storage	Pay for provisioned storage	Pay for storage used and requests
Min Capacity	No minimum	1 GB	No minimum
Max File/Object Size	47.9 TB	16 TB	5 TB
Latency	Low	Lowest	Variable

Enlist the use cases of EFS

Thank You !😎