**­­Notes :** A hypervisor is a software that you can use to run multiple virtual machines on a single physical machine. Every virtual machine has its own operating system and applications. The hypervisor allocates the underlying physical computing resources such as CPU and memory to individual virtual machines as required. Thus, it supports the optimal use of physical IT infrastructure.

Page 1 to 16 -🡪 1 week before interview

**Hypervisor:**

- Software for multiple VMs on one physical machine.

- Allocates CPU, memory for VMs.

- Enables optimal hardware use.

**Importance:**

- Basis for virtualization.

- Creates multiple VMs on a single host.

- Enables efficient cloud scaling.

**Benefits: [HIPES]**

1. Hardware Independence:

- Abstracts hardware.

2. Efficiency:

- Simplifies server OS setup.

3. scalability:

- Maximizes resource use.

4. Portability:

- Shifts resources easily.

**Use Cases:[ DV RO FR LSC]**

1. Desktop Virtualization:

- Remote workstation access.

2. Resource Optimization:

- Consolidates functions.

3. Failure Recovery:

- Snapshots for restoration.

4. Legacy System Continuity:

- Virtualizes hardware.

**How it Works:**

- Installed on servers.

- Intermediary for resource allocation.

**Types of Hypervisors:[BM-H]**

1. Type 1 (Bare-Metal):

- Efficient, direct resource management.

- Secure and stable.

2. Type 2 (Hosted):

- Installed on host OS.

- Introduces latency, OS-dependent.

**Cloud Hypervisor:**

- Cloud data center virtualization.

- Allows multi-tenancy, logical independence.

**Hypervisors vs. Containers:**

- Hypervisor: Hardware abstraction.

- Containers: OS abstraction.

**Security Considerations:**

- Hypervisor critical for VM security.

- Built-in safeguards essential.

**AWS Nitro System:**

- AWS lightweight hypervisor.

- Delegates tasks to specific hardware.

- Monitors resources, enhances performance.

- Compatible with existing workloads.

**# Instance Purchasing Options**

**1.On-Demand Instances:**

- Pay by the second for launched instances.

- **Savings Plans:**

- Commit to consistent usage for 1 or 3 years.

- Reduce costs with a USD per hour commitment

- **Reserved Instances:**

- Commit to consistent instance configuration.

- Term: 1 or 3 years.

- **Spot Instances:**

- Request unused instances.

- Significant cost reduction.

- **Dedicated Hosts:**

- Pay for a dedicated physical host.

- Use existing software licenses for cost reduction.

**- Dedicated Instances:**

- Pay by the hour for single-tenant hardware.

**1.1 On-Demand Instances: Working Methods**

**1. Launch:**

- Create and start an On-Demand Instance.

**2. Connect:**

- Access your Linux instance.

**3. Stop and Start:**

- Temporarily halt and resume your instance.

**4. Hibernate:**

- Preserve state for On-Demand or Spot Instances.

**5. Reboot:**

- Restart your instance.

**6. Instance Retirement:**

- Handle instances reaching retirement.

**7. Terminate:**

- Permanently end an instance.

**8. Recover:**

- Restore or retrieve a terminated instance.

**9. Configure:**

- Customize settings for Amazon Linux instances.

10. Identify:

- Recognize and manage EC2 Linux instances.

**2. Reserved Instances (RIs): -** AWS Reserved Instances (RIs): Up to 75% discount on On-Demand Instances with advance purchase for one or three years.

* Three Payment Options:
* All Up-front Reserved Instances (AURI)
* Partial Up-front Reserved Instances (PURI)
* No Up-front Reserved Instances (NURI)
* Booking Computing Power: organizations "book" computing power in advance, ensuring cost predictability.
* Duration Options: RIs available for one or three-year terms.
* Savings: Significant cost savings compared to On-Demand Instances.
* Payment Flexibility: Different payment options (AURI, PURI, NURI) for managing upfront costs.
* Advance Purchase: RIs require upfront payments for reserved capacity.

**- Cost Savings:**

- Significantly reduce Amazon EC2 costs compared to On-Demand pricing.

**- Nature:**

- Not physical instances.

- Billing discount applied to On-Demand Instances.

- **Attributes for Discount:**

- On-Demand Instances must match specific attributes.

- Example: Instance type, Region.

**2.1 Reserved Instances Scope:**

**- Regional:**

- Purchased for a Region.

- Referred to as regional Reserved Instances.

**- Zonal:**

- Purchased for a specific Availability Zone.

- Referred to as zonal Reserved Instances

**2.2 Types of Reserved Instances:**

- **Offering Classes:**

- Standard or Convertible.

- **Standard Reserved Instance:**

- Higher discount.

- Non-exchangeable.

- Modifiable.

- **Convertible Reserved Instance:**

- Lower discount.

- Exchangeable.

- Modifiable.

**2.3 Application of Reserved Instances:**

**- Nature:**

- Billing discount, not physical instances.

**- Applied To:**

- Running On-Demand Instances.

- **Matching Specifications:**

- On-Demand Instances must match Reserved Instance specs.

- **Automatic Application:**

- Billing discount applies immediately.

- No need to restart instances.

- **Procedure**:

- If no eligible running On-Demand Instance, launch one with matching specs.

**Differences between Standard and Convertible Reserved Instances**

**Standard Reserved Instance**:

* Modify Reserved Instances: Some attributes can be modified.
* Exchange Reserved Instances: Can't be exchanged.
* Sell in the Reserved Instance Marketplace: Can be sold in the Reserved Instance Marketplace.

**Convertible Reserved Instance:**

* Modify Reserved Instances: Some attributes can be modified.
* Exchange Reserved Instances: Can be exchanged during the term for another Convertible Reserved Instance with new attributes, including instance family, instance type, platform, scope, or tenancy.
* Sell in the Reserved Instance Marketplace: Can't be sold in the Reserved Instance Marketplace.

**3. Spot Instances:**

**- Definition:**

- Uses spare EC2 capacity.

- Available for less than On-Demand price.

**- Cost Benefits:**

- Steep discounts on EC2 costs.

- Hourly price known as Spot price.

**- Pricing Mechanism:**

- Set by Amazon EC2.

- Adjusted based on supply and demand.

**- Availability:**

- Runs whenever capacity is available.

**- Suitability:**

- Cost-effective for flexible, interruptible applications.

**- Use Cases:**

- Data analysis, batch jobs, background processing, optional tasks.

**3.1 Spot Instance Interruptions:**

**- Launch on Spare EC2 Capacity:**

- Offers steep discounts but may be interrupted.

**- Spot Instance Interruption:**

- Event when EC2 reclaims Spot Instance.

**- Interrupt Action:**

- Termination, stopping, or hibernation based on user's specification.

**- Demand and Availability:**

- Varies significantly, subject to EC2 capacity.

**- Possibility of Interruption:**

- Spot Instances may be interrupted.

**- Fleet Instances:**

- On-Demand Instances in EC2 Fleet or Spot Fleet cannot be interrupted.

**3.2 - Reasons for Spot Instance Interruption (CPC)☹**

**1. Capacity:**

- EC2 reclaims Spot Instances for repurposing capacity.

- Host maintenance or hardware decommission.

**2. Price:**

- Spot price exceeds your specified maximum price.

- Specifying a maximum price may increase interruptions.

**3. Constraints:**

- Termination as a group if constraints (launch group, AZ group) can't be met.

**3.3 - Specifying Interruption Behavior:**

**- Default Behavior:**

- EC2 terminates Spot Instances by default.

**- Launch Instance Wizard:**

- Advanced details > Request Spot Instances > Customize > Interruption behavior.

- Option for hibernation (Stop - Hibernate behavior).

**- run-instances CLI:**

- (--instance-market-options) > Instance Interruption Behavior.

- Hibernation: --hibernation-options Configured=true.

**- Launch Template:**

- Advanced details > Request Spot Instances > Customize > Interruption behavior.

**- Spot Console:**

- Maintain target capacity > Interruption behavior.

**- create-fleet CLI:**

- Instance Interruption Behavior in request configuration.

- **request-spot-fleet CLI:**

- Instance Interruption Behavior in request configuration.

**- request-spot-instances CLI:**

- --instance-interruption-behavior for specifying behavior.

**3.4 - Hibernate for Interrupted Spot Instances (MOST IMPORTANT)**

**- Specification:**

- Option to hibernate Spot Instances during interruption.

**- Hibernation Experience:**

- Similar to On-Demand Instances.

**- Enhancements:**

- More supported AMIs.

- Expanded support for instance families.

- User-initiated hibernation now supported.

**4. - Burstable Performance Instances (T Types)☹MostImportant**

**- Nature:**

- T instance types are burstable performance instances.

**- Launch Mode Recommendation:**

- For immediate, short-duration usage, launch in Standard mode to avoid higher costs.

- **Unlimited Mode Consideration:**

- Suitable if instance runs long enough to accrue CPU credits for bursting.

- Otherwise, more expensive than other instances due to surplus credit charges.

**- Launch Credits:**

- T2 instances in Standard mode receive launch credits for initial launch experience.

- Repeated launches for new credits are not allowed.

**- Sustained CPU Usage:**

- For sustained CPU usage, use Unlimited mode for T2 Spot Instances or dedicated CPU instance types.

**4.1 - Dedicated Hosts:**

**- Definition:**

- Physical server fully dedicated to your EC2 instance capacity.

- **Purpose:**

- Allows use of existing per-socket, per-core, or per-VM software licenses.

**- License Compatibility:**

- Supports licenses for Windows Server, Microsoft SQL Server, SUSE, and Linux Enterprise Server.

**4.2 - Burstable T3 Instances on Dedicated Hosts: [ Important questions will be raised from here]**

**- Support:**

- Dedicated Hosts support T3 instances.

**- Cost Efficiency:**

- Cost-efficient BYOL license use on dedicated hardware.

**- vCPU Footprint:**

- Smaller vCPU footprint for workload consolidation.

**- Suitable For:**

- Low to moderate CPU utilization workloads.

- Small to medium databases, virtual desktops, development and test environments, code repositories, and product prototypes.

**- Not Recommended For:**

- Sustained high CPU utilization or correlated CPU bursts.

**- Credit Model:**

- Standard mode, no unlimited credit mode support.

**- Performance Characteristics:**

- Baseline CPU with burst capability using accrued credits.

- Credits spent during bursts; lowered CPU when credits are depleted.

**- T3 Dedicated Host Features:**

- Supports EC2 Dedicated Host features (multiple sizes, Host resource groups, BYOL).

**- Supported Sizes/Configurations:**

- General-purpose burstable T3 instances on Dedicated Hosts.

- 48 cores per T3 Dedicated Host, supporting up to 192 instances.

- Automatic calculation of supported instances and sizes.

- Multiple T3 instance types supported on the same host, up to CPU limit.

**4.3 - Shared Dedicated Hosts**

**- Sharing Capability:**

- Allows Dedicated Host owners to share with other AWS accounts or within an AWS organization.

**- Centralized Management:**

- Enables central creation and management of Dedicated Hosts.

**- Model Overview:**

- Owner (AWS account) shares with consumers (other AWS accounts).

- Consumers can launch instances on shared Dedicated Hosts as if they were in their own account.

**- Responsibilities:**

- Owner manages Dedicated Host and instances launched on it.

- Consumers manage instances launched on Dedicated Hosts shared with them.

**- Owner Actions:**

- Can't modify instances launched by consumers on shared Dedicated Hosts.

**- Consumer Actions:**

- Can't modify Dedicated Hosts shared with them or instances owned by others.

**- Sharing Options:**

- Share with specific AWS accounts (inside/outside organization).

- Share with organizational units inside the AWS organization.

- Share with the entire AWS organization.

**4.4 - Host Recovery:**

**- Auto Recovery:**

- Restarts instances on a new host when specific issues detected.

**- Purpose:**

- Reduces manual intervention and operational burden during unexpected host failures.

**- Conditions for Recovery:**

- Detected issues related to system power or network connectivity.

**- Manual Intervention:**

- Required for other Dedicated Host issues.

**4.5 - Host Maintenance Basics:**

**- Degradation Detection:**

- Triggered by hardware issues or specific problematic conditions.

**- Recovery Process:**

- New Dedicated Host allocated; instances on degraded host automatically rebooted.

**- Attributes Retained:**

- Auto placement settings, Availability Zone, Reservation, Host affinity, Host maintenance settings, Host recovery settings, Instance type, Tags.

**- Release of Degraded Host:**

- Occurs after instances are rebooted or stopped on the new Dedicated Host.

**- Access to Degraded Host:**

- Instances accessible before scheduled maintenance but no new launches supported.

**- Use of New Host:**

- Can launch new instances on the new Dedicated Host before maintenance.

- **Capacity Reservation:**

- Some capacity reserved for rebooting instances from the degraded host.

**- Availability:**

- Host maintenance available in all AWS Regions for supported Dedicated Hosts.

**5. Dedicated Instances:**

**- Default Tenancy:**

- EC2 instances run on shared hardware by default.

**- Dedicated Instances:**

- Run on hardware dedicated to a single customer.

- Physically isolated for different AWS accounts, even if linked to a common payer account.

- Might share hardware with non-Dedicated Instances from the same AWS account.

**- Dedicated Host:**

- Physical server dedicated for customer use.

- Provides visibility and control over instance placement on the server.

**5.1 - Dedicated Instances vs. Dedicated Hosts:**

**- Common Features:**

- Used for launching EC2 instances on dedicated physical servers.

- No performance, security, or physical differences.

**- Differences:**

**- Billing:**

- Dedicated Host: Per-host billing.

- Dedicated Instance: Per-instance billing.

**- Visibility:**

- Dedicated Host: Provides visibility of sockets, cores, and host ID.

- Dedicated Instance: No visibility.

**- Host and Instance Affinity:**

- Dedicated Host: Allows consistent deployment to the same physical server over time.

- Dedicated Instance: Not supported.

**- Targeted Instance Placement:**

- Dedicated Host: Provides additional visibility and control over instance placement.

- Dedicated Instance: Not supported.

**- Automatic Instance Recovery:**

- Dedicated Host: Supported (Host recovery).

- Dedicated Instance: Supported.

**- Bring Your Own License (BYOL):**

- Dedicated Host: Supported.

- Dedicated Instance: Partial support.

**- Capacity Reservations:**

- Dedicated Host: Not supported.

- Dedicated Instance: Supported.

**6. - Capacity Reservations Overview:**

**- Definition:**

- Reserve compute capacity for EC2 instances in a specific Availability Zone.

**- Types:**

**- 1. On-Demand Capacity Reservations:**

- Use Cases:

- Scaling events.

- Regulatory requirements and disaster recovery.

**- When to Use:**

- Strict capacity requirements.

- Business-critical workloads.

**- 2. Capacity Blocks for ML:**

**- Use Cases:**

- ML model training and fine-tuning.

- ML experiments and prototypes.

**- When to Use:**

- Ensure uninterrupted access to GPU instances for a defined period.

- Ideal for ML workloads, experiments, and short-term surges in inference demand.

**6.1 - On-Demand Capacity Reservations:**

**- Purpose:**

- Reserve compute capacity for EC2 instances in a specific Availability Zone.

- Mitigate the risk of being unable to get On-Demand capacity during constraints.

**- Recommendation:**

- Suitable for strict capacity requirements and business-critical workloads.

**- Flexibility:**

- Reserve capacity for any duration without a long-term commitment.

- Start and stop capacity assurance as needed.

**- Billing and Discounts:**

- Billing starts as soon as the Reservation is provisioned.

- Utilize billing discounts from Savings Plans and Regional Reserved Instances.

**- Reservation Configuration:**

- Specify:

- Availability Zone.

- Number of reserved instances.

- Instance attributes (type, tenancy, platform/OS).

**- Usage and Matching:**

- Automatically used by running instances with matching attributes.

- Unused if no running instances match the specified attributes.

- Attributes include instance type, tenancy, and platform/OS.

**6.2 - Pricing and Billing:**

**- Payment Options:**

- On-Demand Dedicated Hosts

- Dedicated Host Reservations

- Savings Plans

**- On-Demand Dedicated Hosts:**

- Billing activated upon host allocation.

- Price varies by family and Region.

- Pay per second, minimum 60 seconds.

- Release anytime to stop charges.

**- Dedicated Host Reservations:**

- Provide billing discount compared to On-Demand.

- Three payment options: No Upfront, Partial Upfront, All Upfront.

- Cover one or more hosts in a single Availability Zone.

- Reservation associated hosts can't be released until term ends.

**- Savings Plans:**

- Flexible pricing model for significant savings.

- Commitment to consistent usage, USD per hour, for one or three years.

- Offers flexibility to use Dedicated Hosts as needed.

- Not supported with specific Dedicated Host types.

**- Pricing for Windows Server on Dedicated Hosts:**

- Bring existing licenses with no additional charge.

- Use Windows Server AMIs for latest versions on Dedicated Hosts.

- Supported on current generation instance types.

**6.3 -Capacity Reservation Pricing and Billing:**

**- Pricing:**

- Charged at On-Demand rate, whether used or not.

- Unused reservations appear on the bill.

- No upfront or additional charges.

- Billing discounts for Savings Plans and Regional Reserved Instances apply.

**- Billing:**

- Billing starts upon reservation provisioning.

- Billed at per-second granularity.

- Charged for partial hours.

- Example: Reservation for m4.large Linux instance, billed based on usage hours.

- Billing discounts apply to matching attributes.

- Discounts applied preferentially to instance usage before covering unused reservations.

**- Billing Discounts:**

- Savings Plans and Regional Reserved Instances discounts apply.

- Applied to Capacity Reservations with matching attributes.

- Automatically applied to used instances.

- Discounts preferentially applied to instance usage.

- Zonal Reserved Instances discounts do not apply to Capacity Reservations.

**# SAVING PLANS**

- Flexible pricing model

- EC2, Fargate, Lambda usage

- Low prices

- Commitment to consistent usage

- 1 or 3-year term

- $/hour measurement

- Flexibility in compute options

- Automatic cost savings

- No need for exchanges or modifications

- Discounted Savings Plans price

- Easily reduce your bill

- Commitment to compute usage

- Compute Savings Plans

**1. Compute Savings Plans**

* Flexible
* Reduce costs by up to 66%
* Automatically apply to:
* EC2 instance usage (regardless of instance characteristics)
* Fargate usage
* Lambda usage (Duration, Provisioned Concurrency, Provisioned Duration)

**2. EC2 Instance Savings Plans**

* Significant Discounts
* Lowest prices
* Savings up to 72%
* Commitment to usage of individual instance families in a region
* e.g., M5 usage in N. Virginia
* Automatic cost reduction on selected instance family
* Regardless of AZ, size, OS, or tenancy
* Flexibility to change usage between instances within a family in that region

**Rest notes of AMIs in Note book ….**

**Notes**: By default, EBS root volumes have the DeleteOnTermination flag set to true. meaning the volume will automatically be deleted when you destroy the instance

**# Amazon S3 [ See the pdf provided by sir because it is easy to understand]**

**Amazon S3 Overview:**

* Object storage service by Amazon Web Services (AWS).
* Scalable, durable, and versatile for various use cases.

**Key Features:**

* Stores data as objects in buckets.
* Highly scalable and supports virtually unlimited data.
* High data availability and durability across geographically dispersed data centers.

**Security:**

* Access control lists (ACLs) and bucket policies for data access control.
* Supports server-side encryption for enhanced security.

**Performance:**

* Low-latency and high-throughput performance.
* Multipart uploads for efficient handling of large objects.

**Use Cases:**

* Data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics.

**Management Features:**

* Versioning, lifecycle policies, logging, and event notifications.

**Pay-as-You-Go Model:**

* + Billing based on actual usage of storage and data transfer.

**Conclusion:**

* + Reliable and scalable storage solution for diverse storage needs in cloud computing.

1. **Amazon S3 Features**

* **Storage Classes:**
  + Different classes for varied use cases (e.g., S3 Standard, S3 Glacier).
  + S3 Express One Zone for high-performance, low-latency storage.
* **Storage Management:**
  + S3 Lifecycle for cost-effective object management.
  + S3 Object Lock for data protection.
  + S3 Replication for object duplication.
  + S3 Batch Operations for massive-scale object management.
* **Access Management and Security:**
  + S3 Block Public Access for control.
  + IAM for access control.
  + Bucket policies for permissions.
  + S3 Object Ownership for simplified access.
* **Data Processing:**
  + S3 Object Lambda for custom code execution.
  + Event notifications for workflow triggers.
* **Storage Logging and Monitoring:**
  + CloudWatch metrics for health.
  + CloudTrail for detailed API tracking.
  + Server access logging for request records.
  + Trusted Advisor for best practices.
* **Analytics and Insights:**
  + S3 Storage Lens for storage analysis.
  + Storage Class Analysis for access pattern analysis.
  + S3 Inventory for auditing and reporting.
* **Strong Consistency:**
  + Read-after-write consistency for PUT and DELETE.
  + Consistency for S3 Select, ACLs, Object Tags, and metadata.

**2.** **How Amazon S3 Works:**

* Object Storage Model:
  + Amazon S3 stores data as objects within buckets.
  + An object includes the file and associated metadata.
  + A bucket serves as a container for objects.
* Data Upload Process:
  + To store data, create a bucket with a unique name and specify an AWS Region.
  + Upload data as objects to the created bucket.
  + Each object has a unique key within the bucket.
* Object Configuration:
  + Configure features based on use case needs.
  + S3 Versioning maintains multiple versions of an object for recovery.
* Access Control:
  + Buckets and objects are private by default.
  + Access is granted explicitly through permissions.
  + Use bucket policies, IAM policies, ACLs, and S3 Access Points for access control.
* Additional Topics:
  + Buckets: Containers for objects.
  + Objects: Files and their associated metadata.
  + Keys: Unique identifiers for objects within buckets.
  + S3 Versioning: Keeps track of multiple versions of objects.
  + Version ID: Unique identifier for each version of an object.
  + Bucket Policy: Defines access rules for a bucket.
  + S3 Access Points: Named network endpoints for data access.
  + ACLs: Access control lists for granular permissions.
  + Regions: Specify AWS Region for bucket location.

**3. Amazon S3 Components Key Points:**

* **Buckets:**
  + Containers for objects in S3.
  + Up to 100 buckets/account.
  + Manage access with policies, ACLs, and Access Points.
* **Objects:**
  + Fundamental entities with data and metadata.
  + Identified by a unique key within a bucket.
* **Keys:**
  + Unique identifier for an object in a bucket.
* **S3 Versioning:**
  + Preserves, retrieves, and restores object versions.
  + Helps recover from unintended actions.
* **Version ID:**
  + Generated for each object with versioning.
  + Unique identifier for object versions.
* **Bucket Policy:**
  + IAM policy for access permissions.
  + Limited to 20 KB in size.
  + Uses JSON-based language.
* **S3 Access Points:**
  + Named endpoints for access policies.
  + Simplifies data access management.
  + Each has its access point policy.
* **ACLs (Access Control Lists):**
  + Grants read/write permissions to users.
  + Attached to buckets and objects.
* **Regions:**
  + Geographical location for bucket storage.
  + Choice based on latency, costs, or regulations.
  + Objects stay within the region unless explicitly moved.

**# STORAGE [EBS & INSTANCE STORE]**

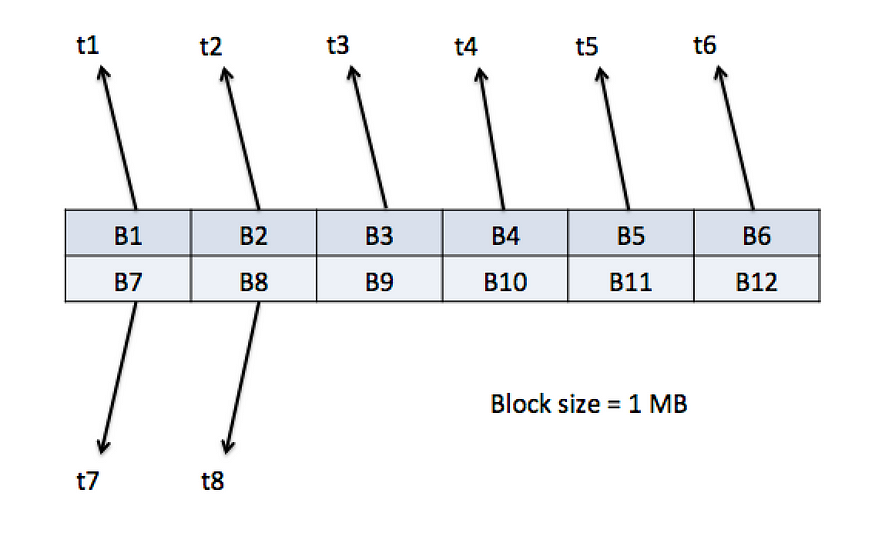
**NOTES:**

**What is an IO operation?**

IO operation stands for Input Output operation. It is a common unit of measurement for storage and system performance. Here, Input is a write operation and Output is a read operation that the disk can handle. Read or write of a single block is considered as an IO operation. Based on the above configuration, if an application wants to read say 5 records from the disk and if those 5 records span in 3 blocks, then the number of IO operations required to read 5 records is 3. Similarly, if each of the 5 records are in 5 different blocks, then the number of IO operations required are 5.

# Throughput

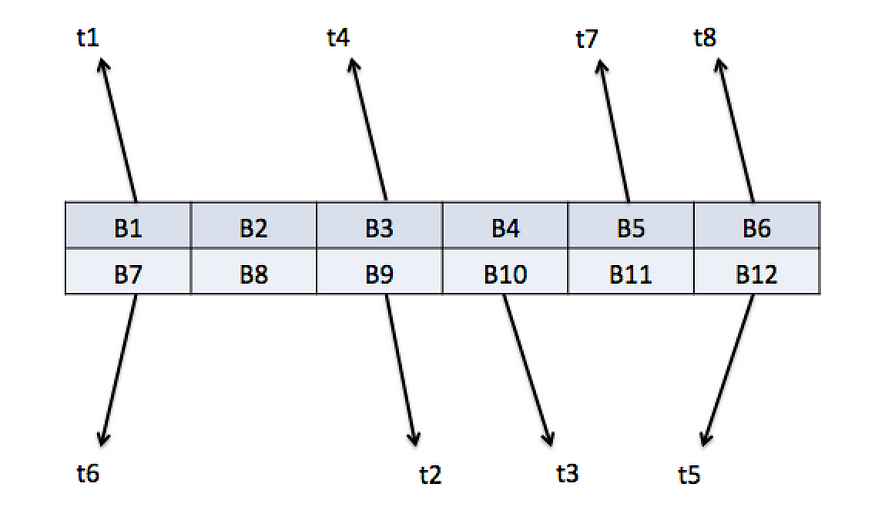
Throughput is the sequential transfer rate which the disk maintains continuously. Throughput deals with only sequential reads or writes. It is simply a measure of IO operations performed on sequential blocks one after the other on the disk. It is represented as total blocks size read/written in a second.



Above is a internal representation of a disk with each small box as a block numbered from B1 to B12. When the application reads 8 blocks in sequence one after the other say block1, block2, block3 …….. block8 in a second (each block is read at sub second interval t<n>) then throughput is measured as 8 MB/S (each block being 1 MB of size)HDDs are designed for sequential IO operations. They give better throughput compared to SSDs.

# IOPS

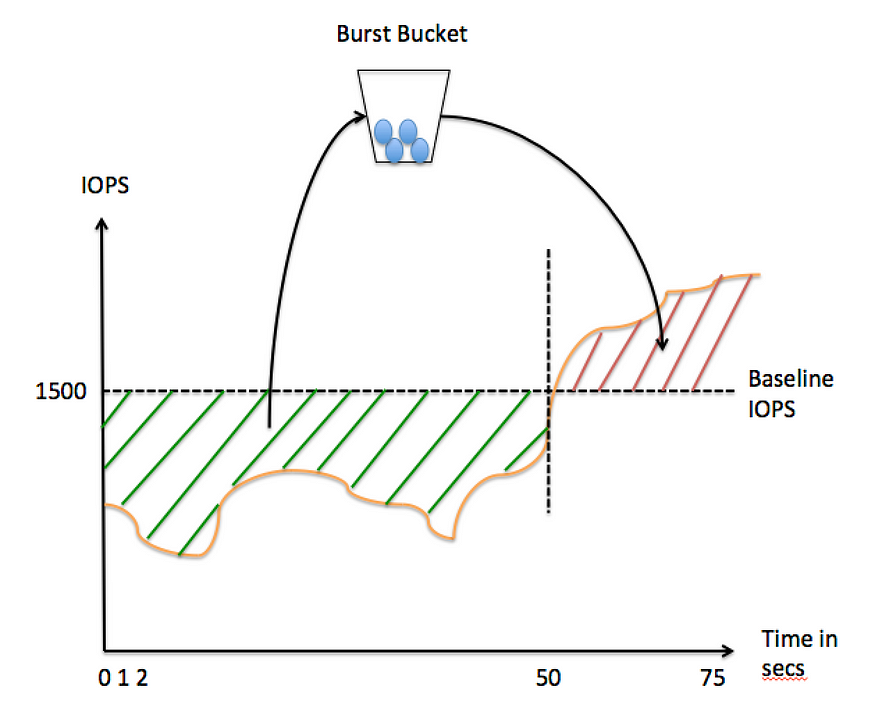
IOPS stands for Input Output operations Per Second. It is a integer value measure of number of IO operations that a disk can handle per second with each block being read from or written to a random location on the disk.



In the above representation, 8 blocks are read from random locations on the disk in a second making IOPS as 8.SSDs are better suited for random IO operations.

# Burst IOPS

In AWS EBS volume types, we often encounter something like **Burst performance IOPS**. AWS volumes provide baseline IOPS for each volume type and when the application do not consume all of these baseline performance IOPS, then the unused IOPS are stored in a burst bucket. These unused IOPS can be drained by the application when it needs more performance.



* **Amazon EBS Overview:**
  + Block level storage for EC2 instances.
* **Volume Behavior:**
  + Behaves like raw, unformatted block devices.
* **Mounting Volumes:**
  + Volumes can be mounted as devices on EC2 instances.
* **Persistence:**
  + Volumes persist independently of the instance's lifecycle.
* **File System and Usage:**
  + Supports file system creation on volumes.
  + Can be used similarly to a block device.
* **Configuration Changes:**
  + Configuration of attached volumes can be dynamically changed.
* **Recommended Use:**
  + Recommended for quickly accessible and persistently stored data.
* **Suitable Applications:**
  + Well-suited for primary storage in file systems, databases, and applications needing fine granular updates.
* **Access to Raw Storage:**
  + Useful for applications requiring access to raw, unformatted, block-level storage.
* **Versatility:**
  + Suited for both database-style applications with random reads and writes, and throughput-intensive applications with long, continuous reads and writes.

**Features of EBS**

* **Volume Creation:**
* Create EBS volumes in a specific Availability Zone and attach to an instance.
* **Snapshot and Restore:**
  + Use snapshots for cross-Availability Zone and cross-Region availability.
* **Volume Types:**
  + General Purpose SSD (gp2 and gp3)
  + Provisioned IOPS SSD (io1 and io2)
  + Throughput Optimized HDD (st1)
  + Cold HDD (sc1)
* **Performance and Use Cases:**
  + General Purpose SSD: Balanced for various workloads.
  + Provisioned IOPS SSD: Consistent IOPS for intensive workloads.
  + Throughput Optimized HDD: Ideal for large, sequential workloads.
  + Cold HDD: Cost-effective for sequential, cold-data workloads.
* **Encryption:**
  + Create encrypted volumes for data-at-rest security.
* **Snapshot Features:**
  + Point-in-time snapshots persisted to Amazon S3.
  + Serve as starting points for new volumes.
* **Cross-Region Snapshot Copy:**
  + Snapshots can be copied across AWS Regions.
* **Performance Monitoring:**
  + Use CloudWatch for metrics (bandwidth, throughput, latency) to optimize resource usage.

**Benefits of Using EBS Volume**

* **Data Availability:**
  + EBS volumes are replicated in the same Availability Zone to prevent data loss.
  + Attached volumes appear as native block devices.
* **Data Persistence:**
  + Volumes persist independently of the instance's lifecycle.
  + Can be detached with data intact for reattachment to new instances.
* **Data Encryption:**
  + Encrypted EBS volumes offer simplified data security.
  + Utilizes 256-bit AES encryption and AWS Key Management Service (AWS KMS).
* **Data Security:**
  + EBS volumes are presented as raw, unformatted block devices.
  + Supports secure data erasure procedures.
* **Snapshots:**
  + Snapshots are backups stored redundantly in Amazon S3.
  + Used for creating new volumes, moving volumes across Availability Zones, and incremental backups.
* **Flexibility:**
  + Supports live configuration changes in production.
  + Allows modification of volume type, size, and IOPS capacity without service interruptions.

**EBS Volume Types**

**Volume types**

* [Solid state drive (SSD) volumes](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html#vol-type-ssd)
* [Hard disk drive (HDD) volumes](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html#vol-type-hdd)
* [Previous generation volumes](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html#vol-type-prev) [magnetic]

**1.Previous-Gen EBS Volume (Magnetic):**

* Type: Standard
* Use Cases: Infrequently accessed data
* Size: 1 GiB - 1 TiB
* Max IOPS: 40 - 200
* Max Throughput: 40 - 90 MiB/s
* Boot Volume: Supported

1. **General Purpose SSD (gp2 and gp3) Volumes:**
   * **Durability:**
   * gp2: 99.8% - 99.9% (0.1% - 0.2% annual failure rate)
   * gp3: 99.8% - 99.9% (0.1% - 0.2% annual failure rate)

* **Use Cases:**
  + Transactional workloads, virtual desktops, medium-sized databases, low-latency apps, boot volumes, development, and test environments
  + **Volume Size:**
  + gp2: 4 GiB - 16 TiB
  + gp3: 1 GiB - 16 TiB
  + **Max IOPS per volume:**
  + gp2: 16,000 (16 KiB I/O)
  + gp3: 16,000 (64 KiB I/O)
  + **Max Throughput per volume:**
  + gp2: 250 MiB/s
  + gp3: 1,000 MiB/s
  + **Amazon EBS Multi-attach:**
  + Not supported for gp2 and gp3.
  + **NVMe Reservations:**
  + Supported for gp2, not supported for gp3.
  + NVMe (nonvolatile memory express) is a new storage access and transport protocol for flash and next-generation solid-state drives (SSDs) that delivers the highest throughput and fastest response times yet for all types of enterprise workloads.
  + **Boot Volume:**
  + Supported for both gp2 and gp3.

**3.Provisioned IOPS SSD (io1 and io2) Volumes:**

* + **Durability:**
  + io1: 99.8% - 99.9% (0.1% - 0.2% annual failure rate)
  + io2: 99.999% (0.001% annual failure rate)
  + **Use Cases:**
  + io1: Workloads requiring sustained IOPS performance or more than 16,000 IOPS, I/O-intensive database workloads.
  + io2: Workloads requiring sustained IOPS performance or more than 64,000 IOPS, I/O-intensive database workloads.
  + **Volume Size:**
  + io1: 4 GiB - 16 TiB
  + io2: 4 GiB - 64 TiB
  + **Max IOPS per volume:**
  + io1: 64,000 (16 KiB I/O)
  + io2: 256,000 (16 KiB I/O)
  + **Max Throughput per volume:**
  + io1: 1,000 MiB/s
  + io2: 4,000 MiB/s
  + **Amazon EBS Multi-attach:**
  + Supported for io1, not supported for io2.
  + **NVMe Reservations:**
  + Supported for io1, not supported for io2.
  + **Boot Volume:**
  + Not supported for both io1 and io2.

**NOTES:**

* + gp2 Volume Throughput:
  + Range: 128 MiB/s to 250 MiB/s (depends on volume size).
  + Pre-Dec 3, 2018: Older volumes may need modification for full performance.
* Max Throughput for Nitro Instances:
  + Achieve 1,000 MiB/s with:
  + 64,000 IOPS.
  + Attachment to Nitro System-built instances.
  + Pre-Dec 6, 2017: Older volumes may need modification for full performance.
* io2 Volume Evolution:
  + All io2 volumes after Nov 21, 2023, are io2 Block Express.
  + Conversion: Modify IOPS or size for io2 volumes before Nov 21, 2023.
* Volume Size Limits:
  + Nitro System Instances: Attach up to 64 TiB volumes.
  + Other Instances: Attach up to 16 TiB volumes.
* IOPS Limits:
  + Nitro System Instances: Attach up to 256,000 IOPS.
  + Other Instances: Attach up to 64,000 IOPS, achieving max 32,000 IOPS.

**4.Hard Disk Drive (HDD) Volumes:**

* **Volume Types:**
  + Throughput Optimized HDD (st1)
  + Cold HDD (sc1)
* **Durability:**
  + Both types have 99.8% - 99.9% durability with a 0.1% - 0.2% annual failure rate.
* **Use Cases:**
  + Throughput Optimized HDD (st1):
  + Big data, data warehouses, log processing.
  + Throughput-oriented storage for infrequently accessed data.
* **Cold HDD (sc1):**
  + Scenarios prioritizing low storage cost.
* **Volume Size:**
  + Both types support volumes ranging from 125 GiB to 16 TiB
  + Max IOPS per Volume (1 MiB I/O):
  + Throughput Optimized HDD (st1):500 IOPS
  + Cold HDD (sc1): 250 IOPS
* **Max Throughput per Volume:**
  + Throughput Optimized HDD (st1): 500 MiB/s
  + Cold HDD (sc1): 250 MiB/s
* **Amazon EBS Multi-attach:**
  + Not supported for both types.
* **Boot Volume:**
  + Not supported for both types.

# 5. General Purpose SSD volumes

# Volume Types:

# General Purpose SSD (gp3) volumes

# General Purpose SSD (gp2) volumes

# Backed by SSDs:

# General Purpose SSD (gp2 and gp3) volumes use solid-state drives.

# Balanced Price and Performance:

# Optimized for a variety of transactional workloads.

# Recommended Workloads:

# Virtual desktops, medium-sized single-instance databases, latency-sensitive interactive applications, development and test environments, and boot volumes.

# Amazon EBS Recommendation:

# Recommended for most workloads.

# 6. gp3 Volume Performance:

# - Burst Performance:

# Not applicable; gp3 volumes do not use burst performance.

# Sustain full provisioned IOPS and throughput indefinitely.

# - IOPS Performance:

# Consistent baseline of 3,000 IOPS included with storage cost.

# Additional IOPS can be provisioned, up to a max of 16,000, at a ratio of 500 IOPS per GiB of volume size.

# Maximum IOPS for volumes 32 GiB or larger.

# - Throughput Performance:

# Consistent baseline throughput of 125 MiB/s included with storage cost.

# Additional throughput can be provisioned, up to a max of 1,000 MiB/s, at a ratio of 0.25 MiB/s per provisioned IOPS.

# Maximum throughput for volumes 8 GiB or larger, provisioned at 4,000 IOPS or higher.

# - gp3 Volume Size:

# Range: 1 GiB to 16 TiB.

# 7. Migrate to gp3 from gp2:

# -Method: Use Amazon EBS Elastic Volumes for migration.

# -Operations: Modify type, IOPS, and throughput without instance interruption.

# Console Defaults:

# When creating volume or AMI from snapshot:

# Default is General Purpose SSD gp3.

# Otherwise:

# Default is gp2.

# Selection Option: Manually select gp3 when gp2 is default.

# 8. General Purpose SSD (gp2) Volumes:

# -Characteristics:

# Cost-effective storage for various transactional workloads.

# Performance scales with volume size.

# Tip:

# gp3 volumes are the latest generation with more predictable scaling and up to 20% lower prices than gp2.

# Estimate savings with the Amazon EBS gp2 to gp3 migration cost savings calculator.

# Performance Metrics:

# Single-digit millisecond latency.

# Volume durability: 99.8% to 99.9% with an AFR of no higher than 0.2%.

# Designed to deliver provisioned performance 99% of the time.

# 9. gp2 Volume Performance:

# IOPS Performance:

# Baseline IOPS scales linearly: 100 to 16,000 (3 IOPS per GiB).

# Provisioning:

# Volumes ≤ 33.33 GiB: 100 IOPS.

# - Volumes > 33.33 GiB and ≤ 5,334 GiB: 3 IOPS per GiB (up to 16,000 IOPS).

# Volumes > 5,334 GiB: 16,000 IOPS.

# Burst Capability (for volumes < 1 TiB and < 3,000 IOPS):

# Bursts to 3,000 IOPS, governed by I/O credits.

# Burst Duration Formula:

* + - * + (I/O credit balance)
    - Burst duration = -----------------------------
      * + (Burst IOPS) - (Baseline IOPS)

# I/O credit accrual at 3 credits per GiB per second when demand drops.

# I/O credit accrual limit: 5.4 million, sustaining max burst for at least 30 minutes.

**10. gp2 Volume Characteristics:**

* **-Burst Capability:**
* Burst duration and refill time based on volume sizes, baseline performance, and burst duration at 3,000 IOPS.
* Example sizes: 1 to 33.33 GiB, 334 GiB, 750 GiB, 1,000 GiB, 5,334 GiB, and larger.
* Monitor I/O credit balance with "Amazon EBS BurstBalance" metric in CloudWatch.
* **Throughput Performance:**
* Varies between 128 MiB/s and 250 MiB/s based on volume size.
* Up to 128 MiB/s for volumes ≤ 170 GiB.
* Up to 250 MiB/s for volumes > 170 GiB but < 334 GiB.
* 250 MiB/s for volumes ≥ 334 GiB.
* **Throughput Calculation Formula:**
* Throughput in MiB/s = IOPS performance × I/O size in KiB / 1,024
* **-Volume Size:** Ranges from 1 GiB to 16 TiB, with linear performance scaling with volume size.

**11. Provisioned IOPS SSD Volumes:**

**Types:**

* io2 Block Express volumes
* io1 volumes

**io2 Block Express Volumes:**

* Designed for demanding I/O applications.
* Uses Block Express for durability, low latency, high performance.
* Communicates with Nitro System via SRD protocol.
* 99.999% durability, sub-ms latency, high IOPS, throughput.
* Supports Nitro instances, specific regions, conversion from io2.
* Up to 64 TiB, 256,000 IOPS, 4,000 MiB/s throughput.

**Performance:**

* Sub-ms latency, 64 TiB storage.
* Up to 256,000 IOPS, 4,000 MiB/s throughput.
* Achievable on Nitro; up to 32,000 IOPS on other instances.
* Throughput scales with 0.256 MiB/s per IOPS.
* Max throughput at 16,000 IOPS or higher.

**Considerations:**

* Convert io2 volumes to io2 Block Express.
* Attachment limits based on instance type.
* Creation of encrypted volumes >16 TiB, IOPS >64,000 requires specific steps.

**NOTES:**

* The Nitro System provides enhanced security that continuously monitors, protects, and verifies the instance hardware and firmware. Virtualization resources are offloaded to dedicated hardware and software minimizing the attack surface.
* A Xen hypervisor allows multiple instances to share a single hardware platform. A hardware on which a hypervisor is running is called a Host machine and the virtual machine is called a Guest machine.

**Differences between MBR (Master Boot Record) and GPT (GUID Partition Table) partitioning schemes.**

**1. Definition:**

MBR: Master Boot Record is a partitioning scheme used to set up a hard disk. It is the traditional and older method.

GPT: GUID Partition Table is a modern partitioning scheme crucial for understanding whether to create a new partition or convert an old one.

**2. Compatibility:**

MBR: Compatible with BIOS systems, which were predominant in older computers.

GPT: Designed to work with UEFI systems, which are more common in modern computers.

**3. Limitation:**

MBR: Limited to four primary partitions or three primary partitions plus one extended partition.

GPT: Can accommodate an unlimited number of partitions.

**4. System Types:**

MBR: Used with BIOS systems.

GPT: Can be used with either MBR or GPT on UEFI systems.

**5. Complexity:**

MBR: Less complex than GPT.

GPT: More complex than MBR but works seamlessly with all operating systems.

It's important to note that as technology has advanced, GPT has become the more preferred partitioning scheme due to its advantages in terms of scalability, compatibility with larger storage capacities, and support for modern UEFI systems. However, the choice between MBR and GPT depends on factors such as the system's architecture, the intended use of the storage, and compatibility requirements.

# Amazon EBS snapshots

# You can back up the data on your Amazon EBS volumes by making point-in-time copies, known as Amazon EBS snapshots. A snapshot is an incremental backup, which means that we save only the blocks on the device that have changed since your most recent snapshot. This minimizes the time required to create the snapshot and saves on storage costs by not duplicating data

**EBS Snapshots Storage**: Stored in Amazon S3

* Location: S3 buckets not directly accessible

**Management Tools:**

* Creation and Management: Amazon EC2 console or Amazon EC2 API
* Inaccessibility: Not accessible via Amazon S3 console or Amazon S3 API

**Snapshot Contents:**

* Data Restoration: Snapshot contains all data needed for restoration to a new EBS volume

**Volume Creation from Snapshot:**

* Replication: New volume is an exact replica of the source volume
* Background Loading: Data loaded in the background for immediate use
* On-Demand Loading: Accessing unloaded data triggers download from Amazon S3

**Snapshot Deletion:**

* Data Removal: Only unique data from the specific snapshot is deleted

**1. EventBridge for Amazon EBS:**

**Integration:** Amazon EBS sends events to Amazon EventBridge for volumes and snapshots

**Rules:** Create rules in EventBridge for programmatic responses to events

**Example Rule:** Notify via email when a snapshot enables fast restore

**Event Representation:**

**Format:** Events as JSON objects

**Sections:**

* "detail" Section: Contains unique event fields
* "event" Field: Specifies event name
* "result" Field: Indicates completion status of triggering action

**Event Types:**

* -Volume Events: createVolume, deleteVolume, attachVolume
* Snapshot Events: EBS snapshot events, EBS Snapshots Archive events, EBS fast snapshot restore events

**Handling with AWS Lambda:**

**Integration:** AWS Lambda manages EventBridge events

**Specific Volume Events:**

**createVolume Event:**

* Sent when volume creation completes

**deleteVolume Event:**

* Sent when volume deletion completes

**attachVolume Event:**

* Sent when volume is attached to an instance

**2. Amazon EBS Volume Events for EventBridge:**

- Events:

* createSnapshot
* createSnapshots
* copySnapshot
* shareSnapshot

**3. Multi-volume Snapshots:**

**Purpose**: Backup for critical workloads like large databases or file systems spanning multiple EBS volumes

**Advantages:**

* Point-in-Time: Exact point-in-time snapshots
* Data Coordinated: Coordinated data across multiple EBS volumes
* Crash-Consistent: Snapshots are crash-consistent

**Process:**

* No need to stop the instance or coordinate between volumes
* Automatic snapshot creation across multiple EBS volumes

**Reference:** Detailed steps for creating multi-volume EBS snapshots in the "Create Amazon EBS snapshots" guide.

**4. Snapshot Pricing:**

* Basis for Charges: Snapshot charges based on stored data amount
* Incremental Nature: Deleting a snapshot may not reduce data storage costs
* Data Removal: Data referenced only by a deleted snapshot is removed
* Preservation: Data referenced by other snapshots is preserved

**5. Multi-volume Snapshots:**

**- Purpose:** Backup for critical workloads like large databases or file systems spanning multiple EBS volumes

**- Advantages:**

* Point-in-Time: Exact point-in-time snapshots
* Data Coordinated: Coordinated data across multiple EBS volumes
* Crash-Consistent: Snapshots are crash-consistent

**- Process:**

* No need to stop the instance or coordinate between volumes
* Automatic snapshot creation across multiple EBS volumes

- **Reference:** Detailed steps for creating multi-volume EBS snapshots in the "Create Amazon EBS snapshots" guide.

- **Snapshot Pricing:**

* Basis for Charges: Snapshot charges based on stored data amount
* Incremental Nature: Deleting a snapshot may not reduce data storage costs
* Data Removal: Data referenced only by a deleted snapshot is removed
* Preservation: Data referenced by other snapshots is preserved

**6. Snapshot Types:**

* First Snapshot: Always a full snapshot

Includes all data blocks from the volume at snapshot creation

* Subsequent Snapshots: Incremental snapshots

Include only changed and new data blocks since the last snapshot

**6.1 Size Determination:**

6.1.1 Full Snapshot:

* Determined by the size of data being backed up
* Not influenced by the size of the source volume

6.1.2 Incremental Snapshot:

* Determined by the size of data written since the previous snapshot

**6.2 Storage Costs:**

6.2.1 Full Snapshot:

* Costs based on the size of the full snapshot, not the source volume size
* Example: First snapshot of a 200 GiB volume with 50 GiB data results in a 50 GiB full snapshot, billed for 50 GiB storage

6.2.2 Incremental Snapshot:

* Costs based on the size of added/changed data since the last snapshot
* Example: Second snapshot after changing 20 GiB and adding 10 GiB results in a 30 GiB incremental snapshot, billed for the additional 30 GiB storage.

**7. EBS Snapshot State Capture:**

* **Objective:** Captures volume state at a specific point in time
* **History Creation**: Subsequent snapshots of a changing volume create a historical record of changes

- **Multiple Snapshots of the Same Volume:**

* Diagram Overview:


        Snapshots capturing an initial volume state and two subsequent states after data has
          been changed.
      

- Volume 1: 15 GiB in size

- Snapshots at three points in time: State 1, State 2, State 3

- **Snapshot Details:**

* State 1:

- Volume has 10 GiB of data

- Snap A: First snapshot, full snapshot, backs up entire 10 GiB

* State 2:

- Volume still has 10 GiB, 4 GiB changed after Snap A

- Snap B: Incremental snapshot, backs up only the 4 GiB of changes

- References the 6 GiB of unchanged data in Snap A (dashed arrow)

* State 3:

- 2 GiB added, total 12 GiB, after Snap B

- Snap C: Incremental snapshot, backs up only the 2 GiB added

- References 4 GiB in Snap B and 6 GiB in Snap A (dashed arrows)

**- Total Storage Required:**

- Snap A: 10 GiB

- Snap B: 4 GiB

- Snap C: 2 GiB

- Total:16 GiB (10 GiB + 4 GiB + 2 GiB)

**8. Incremental Snapshots of Different Volumes:**

**8.1 Diagram Overview:**


        Snapshots capturing an initial volume state and two subsequent states after data has
          been changed.
      

* Vol 1: 14 GiB in size, 10 GiB data
* Snap A: First snapshot, full snapshot, backs up 10 GiB data
* Vol 2: Created from Snap A, exact replica of Vol 1
* Over time, 4 GiB data added to Vol 2, total size 14 GiB
* Snap B: Incremental snapshot of Vol 2, backs up only the 4 GiB added data

**8.2** **Snapshot Details:**

* Vol 1 and Snap A:

- Vol 1: 14 GiB, 10 GiB data

- Snap A: Full snapshot, backs up entire 10 GiB data

* Vol 2 Creation:

- Vol 2: Created from Snap A, replica of Vol 1 at snapshot time

* Vol 2 Changes Over Time:

- 4 GiB data added, total 14 GiB

* Snap B:

- Incremental snapshot of Vol 2

- Backs up only the 4 GiB added data

- References the 10 GiB unchanged data in Snap A

* Snapshot Type Clarification:

- Snap B: Incremental snapshot of Snap A

- Note: Created from a different volume (Vol 2), but still considered an incremental snapshot of the original snapshot (Snap A).

**9. Copy and Share Snapshots:**

9.1 Sharing Snapshots:

* Modify access permissions to share snapshots across AWS accounts
* Copies of own snapshots and shared snapshots can be made
* Refer to "Share an Amazon EBS snapshot" for detailed information

**9.2 Snapshot Constraints:**

-Region Specific:

* Snapshots are constrained to the AWS Region of creation
* Can create new volumes in the same Region using the snapshot
* Refer to "Create a volume from a snapshot" for more details

**9.3 Cross-Region Copy:**

* Snapshots can be copied across Regions
* Enables geographical expansion, data center migration, and disaster recovery
* Refer to "Copy an Amazon EBS snapshot" for detailed information

**9.4 Encryption Support:**

* EBS Snapshots Encryption:
* Fully supports EBS encryption
* Snapshots of encrypted volumes are automatically encrypted
* Volumes created from encrypted snapshots are automatically encrypted
* Unencrypted snapshots can be used to create encrypted volumes on-the-fly
* Refer to "Create Amazon EBS snapshots" and "Copy an Amazon EBS snapshot" for comprehensive documentation on encryption scenarios

**9.5 First Snapshots and Encryption:**

* First Snapshot of Encrypted Volume:

- Always a full snapshot

* First Snapshot of Re-encrypted Volume:

- Different CMK compared to the source snapshot

- Always a full snapshot

**9.6 Documentation:** Comprehensive documentation on possible snapshot encryption scenarios provided in "Create Amazon EBS snapshots" and "Copy an Amazon EBS snapshot."

**#NOTES**

# 1. Archive Amazon EBS snapshots

1. Amazon EBS Snapshots Archive is a new storage tier for low-cost, long-term storage of rarely-accessed snapshots.

2. Snapshots created are stored in the default Amazon EBS Snapshot Standard tier (standard tier), and they are incremental, saving only changed blocks.

3. Archiving a snapshot converts it to a full snapshot and moves it to the Amazon EBS Snapshots Archive tier (archive tier).

4. Full snapshots in the archive tier include all blocks written to the volume at the time of snapshot creation.

5. Archived snapshots can be restored from the archive tier to the standard tier when needed for use.

6. Amazon EBS Snapshots Archive offers up to 75 percent lower storage costs for snapshots stored for 90 days or longer with rare access.

7. Use cases for archiving include end-of-project snapshots, compliance-related full, point-in-time incremental snapshots, and archiving monthly, quarterly, or yearly incremental snapshots.

**1.1 Considerations:**

1. Minimum archive period: 90 days; early deletion bills remaining days.

2. Restoration: Up to 72 hours, depends on snapshot size.

3. Archived snapshots: Always full, may be larger.

4. Recommended archiving: Monthly, quarterly, yearly snapshots.

5. Snapshot lineage: Retains referenced data on standard tier.

6. Archived snapshots matching Recycle Bin rules are retained.

**1.2 Limitations:**

1. Archived snapshots: Can't be used in block mapping or creating volumes.

2. Archive: Completed state snapshots only; owned snapshots can be archived, shared ones need copying.

3. Enabling: Requires permanent restoration of associated snapshots.

4. Archive/restore: Non-cancelable processes.

5. Sharing/Copying: Not possible; restoration needed for copying.

6. Fast snapshot restore: Automatically disabled; manual re-enabling post-restoration.

**1.3 Pricing & Billing:**

1. Archived snapshots: $0.0125 per GB-month; restoring: $0.03 per GB.

2. After restoration: Standard rate $0.05 per GB-month.

3. Minimum archive period: Pro-rated charges for early deletion.

4. Restores: Temporary dual billing; permanent restores delete from archive tier.

5. Deletion: Incurs charges for data moved to archive tier.

6. Recycle Bin: Billed for archived snapshots; deletion before 90 days incurs remaining days' charges.

7. Archived snapshots in Cost and Usage Report with associated tracking types.

Temporary Restores:

- Restoring a snapshot from archive to standard tier results in dual billing.

- Once removed from standard tier, only archive snapshot incurs billing.

**1.4 Permanent Restores:**

- Moving a snapshot from archive to standard tier deletes it from the archive tier.

- Billing is for the standard tier snapshot only.

**1.5 Deleting Snapshots:**

- Deleting an archived snapshot incurs billing for the moved data.

- Charges follow the 90-day minimum archive period.

- If deleted during restoration, you're billed for the full snapshot size.

**1.6 Recycle Bin:**

- Archived snapshots in the Recycle Bin are billed at archive rates.

- Subject to the 90-day minimum archive period; charges apply if deleted early.

- Deleted snapshot matching a retention rule during archiving is kept in the Recycle Bin, billed at archive rates.

- If deleted during restoration, the restored snapshot is in the Recycle Bin, billed at standard snapshot rates.

**1.7 Cost Tracking:**

- Archived snapshots in AWS Cost and Usage Report with the same resource ID and ARN.

- Usage types for identifying costs: SnapshotArchiveStorage (monthly storage fee), SnapshotArchiveRetrieval (one-time restore fee), SnapshotArchiveEarlyDelete (fee for deleting/restoring before 90 days).

**2. Recover from Recycle Bin:**

- Recycle Bin is a data recovery feature for accidentally deleted EBS snapshots and EBS-backed AMIs.

- Deleted resources stay in the Recycle Bin for a specified time before permanent deletion.

- Restore a resource before the retention period ends; after restoration, it's removed from the Recycle Bin.

- Restored resource can be used like any other of the same type.

- If not restored within the retention period, the resource is permanently deleted from the Recycle Bin.

- Snapshots in the Recycle Bin are billed at the regular snapshot rate; no extra charges for using Recycle Bin and retention rules.

**3. Restoring EBS Volumes and EC2 Instances:**

1. For a single volume restoration, restore and replace the volume on an EC2 instance.

2. To restore an entire EC2 instance, use an AMI backup.

**Best Practices for Restoration:**

1. Minimize recovery time and impact by considering the resource being replaced.

2. Regularly test restore processes in lower environments for RPO and RTO validation.

3. Automate and test restores to reduce the risk of failures or inconsistencies.

**Additional Considerations:**

1. For Elastic Load Balancing users, take out a failed instance and restore a new one without traffic disruption.

2. Specific restore processes are outlined for instances not using Elastic Load Balancing.

**Restore Processes for Instances without Elastic Load Balancing:**

1. Restore individual files and directories from EBS snapshots.

2. Restore an EBS volume from an Amazon EBS snapshot.

3. Create or restore an EC2 instance from an EBS snapshot.

4. Restore a running instance from an AMI.

**#Placement Groups**

1. Placement groups allow the grouping of interdependent EC2 instances to influence their placement.

2. Placement groups support different strategies: Cluster, Partition, and Spread.

3. **Cluster** strategy packs instances closely within an Availability Zone for low-latency network performance, suitable for high-performance computing applications.

4. **Partition** strategy spreads instances across logical partitions, preventing sharing of underlying hardware between different partitions, ideal for large distributed workloads like Hadoop, Cassandra, and Kafka.

5. **Spread** strategy strictly places a small group of instances across distinct underlying hardware to reduce correlated failures.

6. Placement groups are optional, and if not used, EC2 tries to spread instances across underlying hardware to minimize correlated failures.

7. No charge is incurred for creating a placement group.

###### **Placement strategies:**

* [Cluster placement groups](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html#placement-groups-cluster)

1. Cluster placement groups are a logical grouping of instances within a single Availability Zone.

2. Cluster placement groups can extend across peered VPCs in the same Region.

3. Instances within a cluster placement group benefit from a higher per-flow throughput limit for TCP/IP traffic.

4. Instances in a cluster placement group are positioned in the same high-bisection bandwidth segment of the network.

**The following image shows instances that are placed into a cluster placement group.**


                    A partition placement group with three partitions.
                

1. Cluster placement groups are recommended for applications requiring low network latency or high network throughput.

2. Enhanced networking is advised for instances in a cluster placement group to achieve the lowest latency and highest packet-per-second network performance.

3. It is recommended to launch instances in a cluster placement group using a single launch request and with the same instance type for all instances.

4. Adding more instances later or using different instance types in the placement group increases the risk of insufficient capacity errors.

5. If instances in a placement group are stopped and restarted, they continue to run in the same placement group, but starting may fail if there isn't enough capacity.

6. If a capacity error occurs when launching an instance in a placement group with running instances, stopping and starting all instances in the group may resolve the issue by migrating them to hardware with sufficient capacity.

* [Partition placement groups](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html#placement-groups-partition)

1. Partition placement groups are designed to minimize the risk of correlated hardware failures for applications.

2. In partition placement groups, EC2 divides the group into logical segments called partitions.

3. Each partition in a placement group has its own set of racks, each with its network and power source.

4. Instances in different partitions of a placement group do not share racks, isolating the impact of hardware failures to specific partitions.

5. The image illustrates a partition placement group with three partitions (Partition 1, Partition 2, and Partition 3), where instances within a partition do not share racks with instances in other partitions.


                    A partition placement group with three partitions.
                

* + Partition placement groups are suitable for deploying large distributed and replicated workloads like HDFS, HBase, and Cassandra across distinct racks.
* When launching instances into a partition placement group, EC2 attempts to evenly distribute them across the specified number of partitions.
* Instances can be launched into a specific partition for more precise control over their placement.
* A partition placement group can span multiple Availability Zones in the same Region, with a maximum of seven partitions per Availability Zone.
* There is no specific limit on the number of instances that can be launched into a partition placement group, subject to the account's overall limits.
* Partition placement groups provide visibility into partitions, allowing awareness of which instances are in each partition.
* This information can be shared with topology-aware applications like HDFS, HBase, and Cassandra, enabling intelligent data replication decisions.
* If there is insufficient unique hardware to fulfill a request when starting or launching an instance in a partition placement group, the request fails. However, more distinct hardware becomes available over time, allowing a retry of the request later.
* [Spread placement groups](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html#placement-groups-spread)

1. Spread placement groups involve instances on separate hardware, recommended for critical applications needing isolation.

2. They reduce simultaneous failures by avoiding shared equipment, suitable for scenarios with a small number of crucial instances.

3. Spread groups allow mixing instance types or launching instances gradually, providing access to distinct hardware.

4. If there's insufficient unique hardware for an instance request, the launch fails initially, but EC2 gradually makes more available.

5. Placement groups can spread instances across racks or hosts, with rack-level spread in AWS Regions and on AWS Outposts.

6. Host-level spread groups are exclusive to AWS Outposts.

###### **Rack level spread placement groups**

The following image shows seven instances in a single Availability Zone that are placed into a spread placement group. The seven instances are placed on seven different racks, each rack has its own network and power source.


                    A spread placement group.
                

1. Rack level spread placement groups can extend across multiple Availability Zones in the same Region.

2. In a Region, a rack level spread placement group can accommodate a maximum of seven running instances per Availability Zone.

3. With Outposts, a rack level spread placement group can host as many instances as there are racks in the Outpost deployment.

###### **Host level spread placement groups**

Host level spread placement groups are only available with AWS Outposts. A host spread level placement group can hold as many instances as you have hosts in your Outpost deployment.

### **General rules and limitations**

1. Maximum of 500 placement groups per account in each Region.

2. Placement group names must be unique within an AWS account for a specific Region.

3. Placement groups cannot be merged.

4. An instance can be in only one placement group at a time; it cannot span multiple groups.

5. On-Demand Capacity Reservation and zonal Reserved Instances provide capacity reservations for instances in a specific Availability Zone and can be used by instances in a placement group.

6. For a cluster placement group with capacity reservation, it's recommended to reserve capacity within the group.

7. Zonal Reserved Instances provide capacity reservations for instances in an Availability Zone but cannot explicitly reserve capacity within a placement group.

8. Dedicated Hosts cannot be launched in placement groups.

9. Spot Instances configured to stop or hibernate on interruption cannot be launched in a placement group.

### **Cluster placement group rules and limitations**

1. Cluster placement groups support current generation instances (excluding T2, Mac1, M7i-flex) and certain previous generation instances (A1, C3, C4, G2, I2, M4, R3, R4).

2. Cluster placement groups cannot span multiple Availability Zones.

3. Maximum network throughput between two instances is limited by the slower instance in a cluster placement group.

4. Enhanced networking-enabled instances in a cluster placement group can use up to 10 Gbps for single-flow traffic; outside, it's up to 5 Gbps.

5. Traffic to/from Amazon S3 buckets within the same Region has no bandwidth restrictions.

6. Launching multiple instance types in a cluster group reduces the likelihood of successful capacity availability; using the same type is recommended.

7. Network traffic to the internet and over AWS Direct Connect to on-premises resources is limited to 5 Gbps.

### **Partition placement group rules and limitations**

1. Partition placement groups support a maximum of seven partitions per Availability Zone.

2. The number of instances that can be launched in a partition placement group is limited by account limits.

3. EC2 attempts to evenly distribute instances across all partitions when launched, but an even distribution is not guaranteed.

4. A partition placement group with Dedicated Instances can have a maximum of two partitions.

5. Capacity Reservations do not reserve capacity in a partition placement group.

### **Spread placement group rules and limitations**

1. Rack spread placement groups support a maximum of seven running instances per Availability Zone.

2. In a Region with three Availability Zones, you can run a total of 21 instances in the group (seven instances per Availability Zone).

3. Trying to start an eighth instance in the same Availability Zone and spread placement group will fail.

4. For more than seven instances in an Availability Zone, it's recommended to use multiple spread placement groups, though it doesn't guarantee spread between groups.

5. Spread placement groups are not supported for Dedicated Instances.

6. Host level spread placement groups are only supported on AWS Outposts, holding as many instances as hosts in the Outpost deployment.

7. Rack level spread placement groups in a Region allow a maximum of seven running instances per Availability Zone per group; with AWS Outposts, it can hold instances equal to the number of racks in the Outpost deployment.

8. Capacity Reservations do not reserve capacity in a spread placement group.

# #Elastic IP addresses (STATIC/CONSTANT PUBLIC IP)

# 1. Elastic IP is a static IPv4 address for dynamic cloud computing, allocated until released.

# 2. It helps mask instance or software failures by quickly remapping the address to another instance.

# 3. Elastic IP can be used for DNS records in domain mapping or dynamic DNS on Amazon Linux.

# 4. It is a public IPv4 address reachable from the internet.

# 5. Useful for instances without a public IPv4 address, enabling communication with the internet for connections from local computers.

## **Elastic IP address pricing**

1. Hourly charge for an Elastic IP address if not associated with a running instance or associated with a stopped instance or unattached network interface.

2. No charge for one Elastic IP associated with a running instance, but additional Elastic IPs are billed.

3. AWS charges for all public IPv4 addresses, including those associated with running instances and Elastic IP addresses.

**The following are the basic characteristics of an Elastic IP address:**

1. Elastic IP address is static and doesn't change over time.

2. Tied to a specific AWS Region, cannot be moved to a different Region.

3. Comes from Amazon's IPv4 pool or a custom IPv4 address pool in your AWS account.

4. Allocation and association with instances or network interfaces are necessary for use.

5. Associating with an instance releases the instance's public IPv4 address back to Amazon's pool.

6. Disassociating an Elastic IP allows association with a different resource after closing active connections.

7. Disassociated Elastic IP remains allocated until explicitly released, incurs a small hourly charge.

8. Association with an instance changes the instance's public DNS hostname to match the Elastic IP.

9. Public DNS resolution varies outside and inside the instance's network.

10. Allocated Elastic IPs from a custom pool don't count toward limits; network border group association is possible.

11. Network border group specifies the location for advertising CIDR block; defaults to all Availability Zones in the Region.

12. Elastic IP is intended for use in a specific network border group only.

**# NAT GATEWAY**

A NAT gateway is a Network Address Translation (NAT) service. You can use a NAT gateway so that instances in a private subnet can connect to services outside your VPC but external services cannot initiate a connection with those instances.

**1. Public NAT Gateway:**

- Instances in private subnets can connect to the internet through a public NAT gateway.

- They cannot receive unsolicited inbound connections from the internet.

- Created in a public subnet and requires an associated elastic IP address.

- Routes traffic from the NAT gateway to the internet gateway for the VPC.

- Can be used to connect to other VPCs or on-premises networks via a transit gateway or virtual private gateway.

**2. Private NAT Gateway:**

- Instances in private subnets can connect to other VPCs or on-premises networks through a private NAT gateway.

- Cannot have an associated elastic IP address.

- Routes traffic through a transit gateway or a virtual private gateway.

- Attaching an internet gateway to a VPC with a private NAT gateway is allowed, but routing traffic to the internet gateway results in dropped traffic.

**3. Address Mapping:**

- Both private and public NAT gateways map the source private IPv4 address of instances to the NAT gateway's private IPv4 address.

- In the case of a public NAT gateway, the internet gateway maps the private IPv4 address of the NAT gateway to the associated Elastic IP address.

- When sending response traffic, the NAT gateway translates the address back to the original source IP address.

**4. Routing Traffic:**

- Both public and private NAT gateways can route traffic to transit gateways and virtual private gateways.

- Using a private NAT gateway results in traffic coming from the private IP address of the NAT gateway.

- Using a public NAT gateway results in traffic coming from the private IP address of the public NAT gateway unless an internet gateway is used, in which case the public NAT gateway uses its Elastic IP address as the source.

**Important Note:**

- It is emphasized that you can use either a public or private NAT gateway to route traffic to transit gateways and virtual private gateways.

# Compare NAT gateways and NAT instances

**NAT Gateways:**

1. Highly available with redundancy in each Availability Zone.

2. Offers up to 100 Gbps bandwidth.

3. Managed by AWS, no user maintenance required.

4. Optimized for handling NAT traffic.

5. Uniform offering; no need to choose type or size.

6. Automatically selects private IP addresses.

7. No support for port forwarding or bastion servers.

8. Specific timeout behavior (returns RST packet).

9. Supports forwarding of IP fragmented packets for UDP, not for TCP and ICMP.

**NAT Instances:**

1. Requires user-managed failover between instances.

2. Bandwidth depends on instance type.

3. User-managed maintenance, including software updates.

4. Uses generic AMI for NAT functionality.

5. Cost depends on instances, usage duration, and type/size.

6. Users choose suitable instance type and size.

7. Assigns specific private IP addresses upon instance launch.

8. Supports security groups for instance control.

9. Manual configuration needed for port forwarding.

10. Can be used as bastion servers.

11. Specific timeout behavior (sends FIN packet).

12. Supports forwarding of IP fragmented packets for UDP, TCP, and ICMP.

# #security groups

# Control Traffic with Security Groups:

# 1. Functionality:

# - Security groups manage inbound and outbound traffic for associated AWS resources.

# - For instance, when linked to an EC2 instance, a security group regulates its traffic flow.

# 2. Default Security Group:

# - VPCs are equipped with a default security group.

# - Additional security groups can be created for a VPC, each with its own set of rules.

# 3. Rule Customization:

# - Inbound rules can be specified with source, port range, and protocol details.

# - Outbound rules can be configured with destination, port range, and protocol specifications.

# In essence, security groups offer a fine-grained control mechanism for regulating the flow of traffic to and from AWS resources within a VPC.

# # Network Access Control Lists (ACLs)

# 1. Functionality:

# - Network ACLs control specific inbound or outbound traffic at the subnet level within a VPC.

# - They operate as an additional layer of security, alongside security groups.

# 2. Usage Options:

# - You can utilize the default network ACL provided with your VPC.

# - Alternatively, create a custom network ACL with rules similar to those in security groups.

# 3. Cost:

# - No additional charges apply for using network ACLs.

# 4. Configuration:

# - Network ACL rules define allowed or denied traffic based on source, destination, port, and protocol.

# - Similar to security groups, network ACLs contribute to securing the VPC environment.

# 5. Diagram Illustration:

# - The diagram depicts a VPC with two subnets, each having its own network ACL.

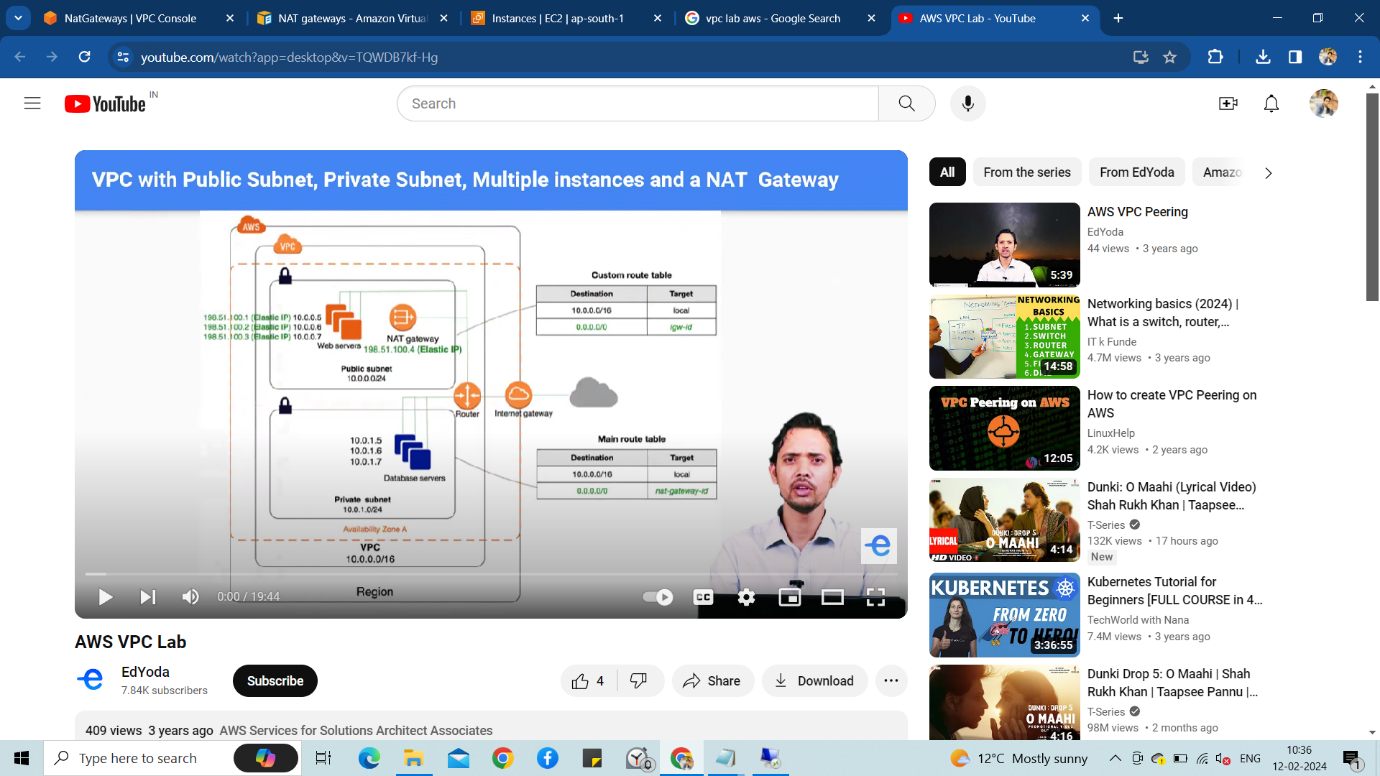
# - Inbound and outbound traffic is governed by respective network ACLs, specifying permissions for traffic entering or leaving subnets.

# A VPC with a security group. The EC2 instance in the subnet is associated with the security group.

# In summary, Network Access Control Lists offer subnet-level traffic control, allowing or denying specific data flows within a VPC. They add an extra layer of security without incurring additional costs.

# # Important figure to be understand

# AWS Default VPC | Understanding Virtual Private Cloud



# #What is VPC peering?

A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch AWS resources, such as Amazon EC2 instances, into your VPC.

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account. The VPCs can be in different Regions (also known as an inter-Region VPC peering connection).


            A VPC peering connection
        

* VPC Peering Infrastructure: AWS uses existing VPC infrastructure for connections; no need for gateways, VPNs, or extra hardware.
* No Single Point of Failure: VPC peering eliminates failure points and avoids bandwidth bottlenecks.
* Data Transfer: Facilitates data transfer, useful for cross-account scenarios like file sharing.
* Cross-Account VPC Peering: Connects VPCs from different AWS accounts, enabling resource access.
* Inter-Region VPC Peering: Allows secure communication between VPCs in different regions using private IP addresses.
* Security Features: Ensures encrypted traffic, avoiding public internet traversal, reducing threats.

**- Benefits of Inter-Region VPC Peering:**

* Simple, cost-effective resource sharing.
* Geographic redundancy through data replication.
* Enhanced security with encrypted traffic.
* No single failure point or bandwidth issues.

**- VPC Peering Pricing:**

* Creation: No charge for creating a VPC peering connection.
* Data Transfer within Availability Zone (AZ): Free of charge.
* Data Transfer across AZs and Regions: Charges apply for data transfer in these scenarios.