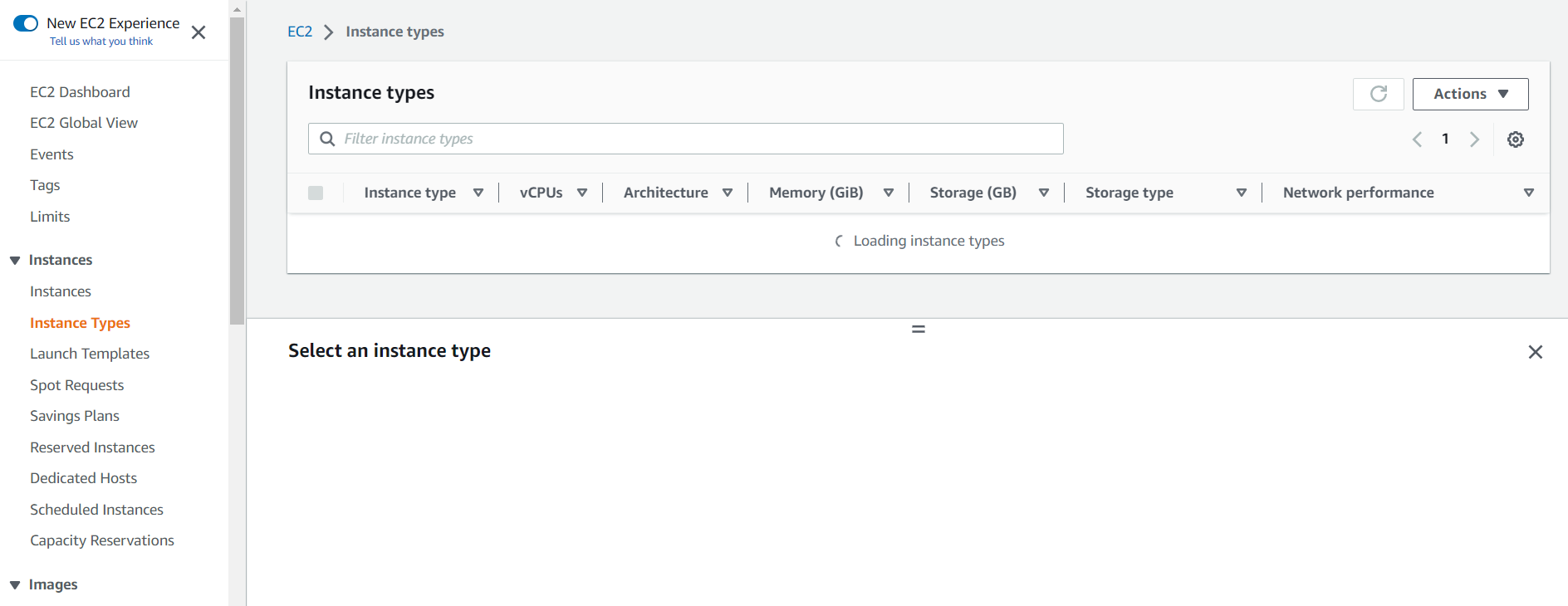
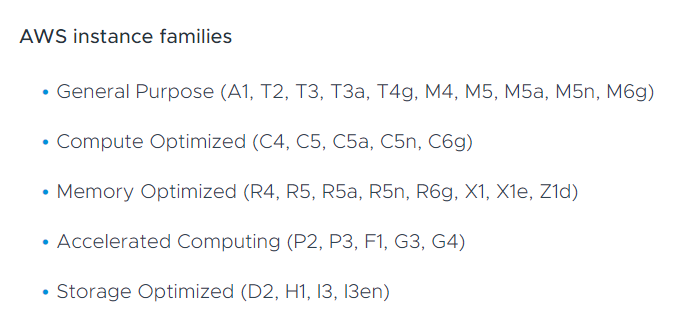
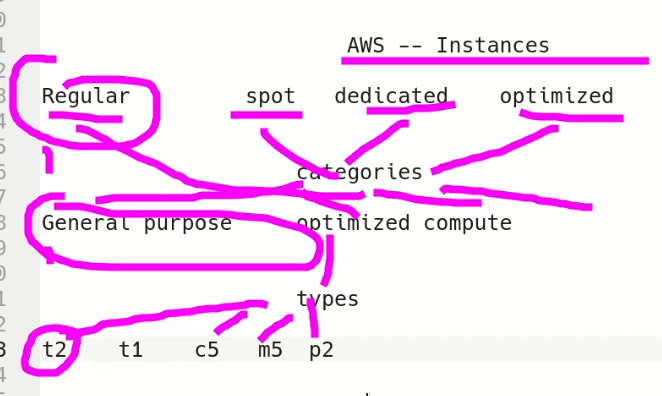
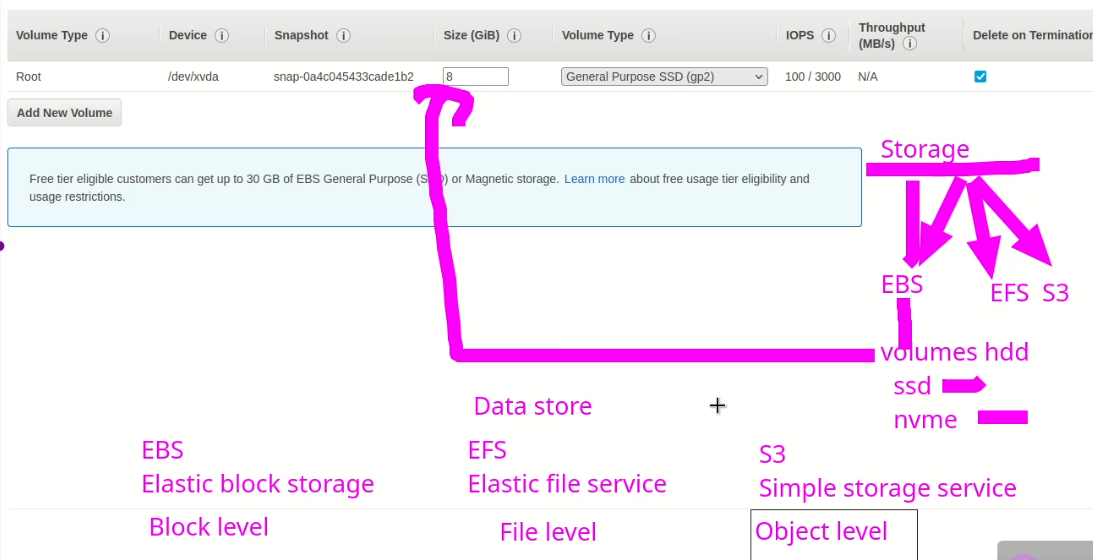
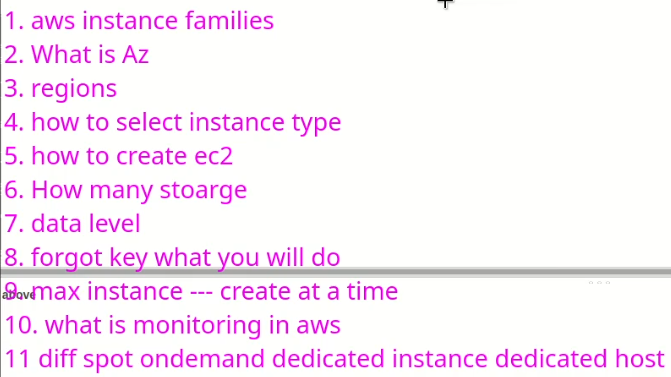
Lecture 5

**AWS-Instace Types-Families-EBS-Volumes**

* Instance types/Family
* 
* <https://blogs.vmware.com/cloudhealth/aws-instance-types-and-comparison/>
* 
* Knowing instances families is an interview question.
* 
* 
* Further these Instances are classified as
* Table 1: AWS instance types and comparison of use cases

| **Instance Family** | **Instance Types** | **Use Cases** |
| --- | --- | --- |
| General Purpose | A Instance Family | The A instance family consists of instances suitable for suited for scale-out and Advanced RISC Machine (ARM)-based workloads that are supported by the extensive Arm ecosystem. |
| General Purpose | T Instance Family | The T instance family consists of burstable instances suitable for websites, web applications, development environments, microservices, and line of business applications. |
| General Purpose | M Instance Family | The M instance family is suitable for small and mid-size databases, data processing tasks that require additional memory, cluster computing, and other enterprise applications. |
| Compute Optimized | C Instance Family | The compute-optimized C instance family is a better match for high-performance web servers and processes such as distributed analytics and high-performance computing. |
| Memory Optimized | R Instance Family | Memory intensive tasks such as high-performance database analytics and data mining processes are best suited by the R instance family along with mid-size in-memory databases and similar applications. |
| Memory Optimized | X and Z Instance Families | The X and Z instance families are designed for high performance, large enterprise databases with in-memory databases (e.g. SAP HANA), or big data processing engines (e.g. Apache Spark or Presto). |
| Accelerated Computing | P Instance Family | The P instance family is the one to use for tasks such as machine learning and those with high-performance requirements such as computational finance, seismic analysis, and speech recognition. |
| Accelerated Computing | G Instance Family | If a task involves 3D visualizations, graphics-intensive remote workstations, 3D rendering, application streaming, video encoding, or other graphics workloads, the G instance family is best. |
| Accelerated Computing | F Instance Family | The F instance family offers customizable hardware acceleration with field-programmable gate arrays (FPGAs for tasks such as genomics research, financial analytics, and real-time video processing. |
| Storage Optimized | H Instance Family | The H instance family is suitable for MapReduce-based workloads, distributed file systems, network file systems, log or data processing applications such as Apache Kafka, and big data workloads. |
| Storage Optimized | I Instance Family | Tasks suitable for the I instance family include NoSQL databases, in-memory databases (e.g. Aerospike), scale-out transactional databases, data warehousing, and analytics workloads. |
| Storage Optimized | D Instance Family (D2) | Finally, the D Instance family is best suited to tasks such as Massively Parallel Processing (MPP) data warehousing, MapReduce and Hadoop distributed computing, and log or data-processing applications. |

* Add volume (storage)
* Root volume by default 🡪 8 Gb (free tire)
* Storage is categories are EBS (Elastic Block Store), EFS (Fast and Efficient Storage), S3 (Simple Storage Service)
* Hierarchy
* 
* Interview questions.
* 
* 1.
* General Purpose Instances: These instances are designed for a wide range of workloads and provide a balance of compute, memory, and networking resources. Examples include the M5 and T3 instance families.
* Compute Optimized Instances: These instances are optimized for compute-intensive workloads, such as high-performance computing and machine learning. Examples include the C5 and P3 instance families.
* Memory Optimized Instances: These instances are optimized for memory-intensive workloads, such as large-scale databases and in-memory analytics. Examples include the R5 and X1 instance families.
* Storage Optimized Instances: These instances are optimized for storage-intensive workloads, such as big data processing and data warehousing. Examples include the I3 and D2 instance families.
* GPU Instances: These instances are designed for workloads that require GPU acceleration, such as graphics rendering, machine learning, and scientific computing. Examples include the G3 and G4 instance families.
* FPGA Instances: These instances are designed for workloads that require field-programmable gate arrays (FPGAs), which are programmable logic devices that can be configured to perform specific tasks. Examples include the F1 instance family.
* 2.
* An Availability Zone (AZ) is an isolated data center within a geographic region that is designed to be highly available and fault-tolerant. In other words, it is a physically separate data center with its own power, cooling, and networking infrastructure that is located within a single geographic region.
* AWS provides multiple Availability Zones within each of its regions to help customers build highly available and fault-tolerant applications. Each Availability Zone is designed to be independent of the others, meaning that if one Availability Zone goes down due to a failure or outage, other Availability Zones in the same region are unaffected and continue to operate normally.
* By deploying resources across multiple Availability Zones within a region, customers can ensure that their applications are highly available and resilient to failure. AWS also provides services such as Amazon Elastic Load Balancing and Amazon Route 53 that help distribute traffic and handle failover between multiple Availability Zones within a region.
* It's important to note that while Availability Zones are designed to be independent and resilient, they are not a substitute for backup and disaster recovery solutions. Customers should always have backup and recovery plans in place to protect against data loss or application downtime in the event of a catastrophic failure or outage.
* 3.

In Amazon Web Services (AWS), a region is a geographic area that contains multiple Availability Zones (AZs). Each region is isolated from other regions, and each AZ within a region is designed to be independent and fault-tolerant. AWS currently has over 25 regions around the world, with plans to add more in the future.

Each AWS region is identified by a unique name, such as us-west-2 (Oregon) or eu-central-1 (Frankfurt). Within each region, customers can deploy resources such as EC2 instances, S3 buckets, and RDS databases. Customers can also use AWS services such as Amazon SNS, Amazon SQS, and Amazon Lambda across regions.

Deploying resources across multiple regions can provide several benefits, such as:

1. Low-latency access: By deploying resources closer to end-users or customers, applications can provide low-latency access and improved performance.
2. Data residency and compliance: By deploying resources in specific regions, customers can ensure that their data is stored in compliance with data residency and privacy regulations.
3. Disaster recovery and business continuity: By deploying resources in multiple regions, customers can ensure that their applications are highly available and resilient to catastrophic failure or outages.

It's important to note that not all AWS services are available in all regions. Customers should check the availability of each service in each region before deploying resources.

* 4.

Selecting the right instance type in Amazon Web Services (AWS) depends on several factors, including the requirements of your application, the workload type, and the budget. Here are some steps to help you select the right instance type:

1. Identify the requirements of your application: Consider the CPU, memory, storage, and networking requirements of your application. This will help you narrow down the list of instance types that are suitable for your workload.
2. Determine the workload type: Consider the workload type, such as compute-intensive, memory-intensive, or storage-intensive. This will help you choose an instance family that is optimized for your workload type.
3. Choose the right instance size: Within each instance family, there are multiple instance sizes that offer different amounts of CPU, memory, and storage. Choose the instance size that best fits your workload requirements and budget.
4. Consider pricing options: AWS offers various pricing options, including on-demand, reserved, and spot instances. Consider the pricing option that best fits your budget and usage patterns.
5. Test and optimize: Once you have selected an instance type, test your application and optimize it for the selected instance. You can use AWS tools such as Amazon CloudWatch and AWS Trusted Advisor to monitor and optimize your application performance.

It's important to note that selecting the right instance type is not a one-time decision. As your application evolves and grows, you may need to reevaluate and adjust your instance type selection to ensure optimal performance and cost-effectiveness.

* 5.

To create an EC2 (Elastic Compute Cloud) instance in Amazon Web Services (AWS), follow these steps:

1. Log in to your AWS account and navigate to the EC2 console.
2. Click the "Launch Instance" button to begin the EC2 instance creation wizard.
3. Choose an Amazon Machine Image (AMI) for your instance. The AMI contains the operating system and other software that will run on the instance.
4. Choose an instance type based on your application requirements and workload type.
5. Configure the instance details, such as the number of instances to launch, the network settings, and the storage options.
6. Choose a security group to control inbound and outbound traffic to your instance.
7. Review your instance configuration and click the "Launch" button.
8. Create or select an existing key pair to securely connect to your instance.
9. Launch the instance and wait for it to become available.

Once the instance is launched, you can connect to it using an SSH client or other remote access tools. You can also configure the instance further by installing additional software, configuring security settings, and connecting it to other AWS services.

It's important to note that you will be charged for EC2 instances based on the instance type, usage, and other factors. Be sure to monitor your EC2 usage and cost regularly to avoid unexpected charges.

* 6.

In Amazon Web Services (AWS), there are several types of storage services available to customers, including:

1. Amazon S3 (Simple Storage Service): A highly scalable and durable object storage service for storing and retrieving any type of data.
2. Amazon EBS (Elastic Block Store): A block-level storage service for EC2 instances that provides persistent block-level storage volumes.
3. Amazon EFS (Elastic File System): A scalable and fully managed file storage service that provides a common file system for multiple EC2 instances.
4. Amazon FSx (File System): A fully managed file storage service for Windows and Linux workloads that provides file shares accessible via SMB or NFS protocols.
5. Amazon Glacier: A low-cost and secure data archiving service for long-term data retention.
6. AWS Storage Gateway: A hybrid storage service that enables customers to seamlessly integrate their on-premises storage environments with AWS cloud storage.
7. Amazon DynamoDB: A fully managed NoSQL database service that provides low-latency and high-performance storage for unstructured data.
8. Amazon RDS (Relational Database Service): A fully managed relational database service that provides scalable and highly available database storage for MySQL, PostgreSQL, Oracle, and other database engines.

Each storage service has different features and pricing models, and is optimized for different use cases and workloads. Customers can choose the storage service that best fits their requirements and budget

* 7.
* Block-level data refers to the data that is stored at the lowest level of a storage device. In this case, data is organized into fixed-size blocks, usually ranging from a few kilobytes to several megabytes. These blocks are typically accessed and managed by the operating system, which can read or write data to them.
* File-level data refers to data that is organized into files and directories. File systems manage file-level data and provide a way to organize data into directories and files. This type of data is typically accessed by applications and users, who create, modify, and delete files as needed.
* Object-level data refers to data that is stored as discrete units, known as objects, which are self-contained and independent of any specific application or file system. Object storage systems use a unique identifier to manage objects, making them easily retrievable and accessible from any application or system that has access to the storage device. This type of data is typically used for storing unstructured data such as images, videos, and other multimedia files.
* 8.

If you have lost or forgotten the private key pair for your Amazon EC2 instance, you will not be able to log in to the instance using SSH. However, there are a few different methods to regain access to the instance, depending on the situation:

1. If you have created an Amazon Machine Image (AMI) of the instance, you can launch a new instance from the AMI with a new key pair. This will create a new instance with the same configuration and data as the original instance, but with a new private key pair.
2. If you have attached an Elastic Block Store (EBS) volume to the instance, you can detach the volume and attach it to another instance that you have access to. From there, you can mount the volume and modify the contents of the original instance's file system, including the authorized\_keys file.
3. If you have enabled the EC2 Instance Connect service for your instance, you can use the EC2 Instance Connect CLI or web-based console to connect to the instance without requiring the private key pair.
4. If none of the above options are available or feasible, you may need to create a new instance and manually migrate your data and applications to the new instance. This may be more time-consuming and complex, but it is a guaranteed way to regain access to your system.

It is important to keep your private key pair in a safe and secure location, as losing access to it can be a serious security risk. Be sure to follow best practices for key management, such as creating backups and limiting access to authorized users only.

* 9.

In Amazon Web Services (AWS), there is no hard limit on the number of instances that can be created at a time, as this can depend on a variety of factors, including the available resources in the selected region and account limits.

However, there are several best practices and recommended limits for launching and managing instances in AWS, including:

1. Launch instances in batches: Rather than launching a large number of instances at once, it is generally recommended to launch instances in smaller batches, allowing you to monitor and manage them more effectively.
2. Use Auto Scaling groups: Auto Scaling groups can be used to automatically launch and manage multiple instances based on predefined conditions, such as changes in demand or resource utilization.
3. Monitor resource utilization: Monitor the resource utilization of your instances and adjust the number of instances as needed to ensure optimal performance and cost efficiency.
4. Consider instance limits: AWS accounts have default limits on the number of instances that can be launched, and these limits can be increased by submitting a request to AWS support.

In general, it is recommended to follow AWS best practices and monitor your resources carefully to ensure that you are optimizing your instance usage while maintaining cost-effectiveness and performance.

* .
* 10

Monitoring in AWS refers to the process of collecting, processing, and analyzing data about the performance, health, and resource utilization of AWS resources and services.

AWS provides several monitoring tools and services that allow customers to monitor their AWS resources and services in real-time, including:

1. Amazon CloudWatch: A monitoring and management service that provides metrics, alarms, and logs for AWS resources and applications.
2. AWS CloudTrail: A service that records API calls made to AWS services and resources and provides detailed audit trails for compliance and security.
3. AWS Config: A service that provides resource inventory, configuration history, and change notifications for AWS resources and services.
4. Amazon X-Ray: A service that allows customers to trace and analyze requests made to distributed applications and microservices.
5. AWS Personal Health Dashboard: A personalized view of the health of AWS services and resources that are associated with a customer's AWS account.

By using these monitoring tools and services, customers can gain visibility into their AWS resources and services, identify performance bottlenecks, troubleshoot issues, and optimize resource utilization. Additionally, customers can set up automated alerts and notifications to proactively detect and respond to issues before they impact their applications and users.

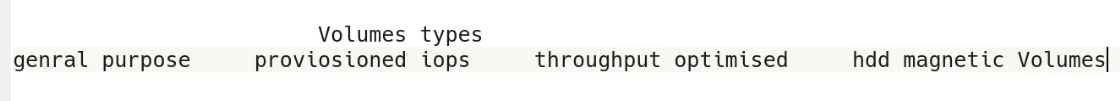
11.

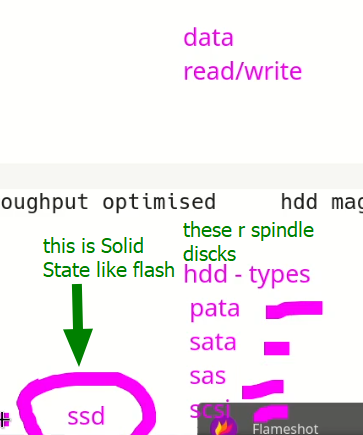
The following are the differences between the different instance types in Amazon Web Services (AWS):

1. On-Demand Instances: These instances are the most commonly used instances in AWS. They are pay-as-you-go, meaning that you only pay for the instances that you use, by the hour or by the second. You can launch and terminate instances as per your requirement, and there are no long-term commitments.
2. Spot Instances: These instances are also pay-as-you-go, but they can be much cheaper than On-Demand instances. The price of Spot instances fluctuates based on supply and demand. You can bid for an instance and use it as long as your bid price is higher than the current Spot price. However, your instance may be terminated if the Spot price goes higher than your bid price.
3. Dedicated Instances: These instances are similar to On-Demand instances, but they run on hardware that is dedicated to a single customer. This means that the instances are not shared with other customers, which can improve security and compliance. Dedicated instances can be launched as needed, and you pay an hourly fee for the instance.
4. Dedicated Hosts: These are physical servers that are dedicated to a single customer. With Dedicated Hosts, you have complete control over the placement of your instances, which can help you meet regulatory and compliance requirements. Dedicated Hosts are billed on a per-hour basis, and you can launch instances on them as per your requirement.

In summary, the main differences between these instance types are related to their pricing, provisioning, and allocation models. On-Demand and Dedicated Instances provide flexibility and ease of use, while Spot Instances and Dedicated Hosts provide cost savings and control over instance placement.

* .



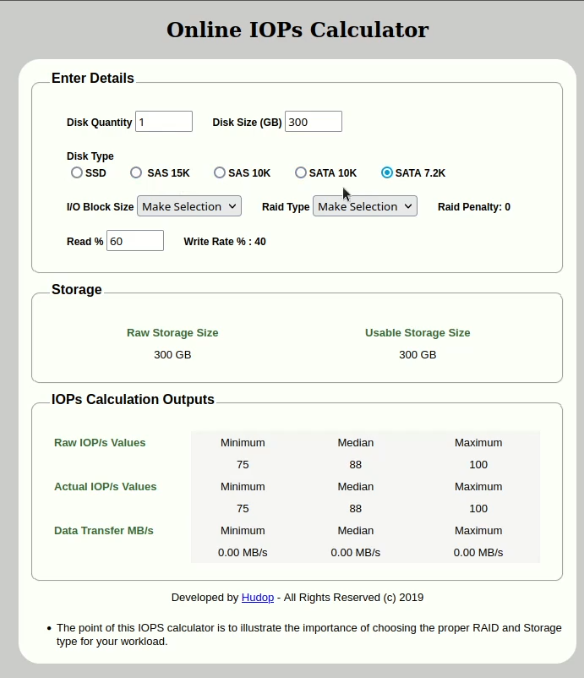
* 
* SSD is fastest HD
* Types of Hard Drives

There are two main types of hard drives:

1. Hard Disk Drive (HDD): This type of hard drive uses spinning disks to store data. It is a mechanical device with read/write heads that move over the spinning disks to read and write data. HDDs have been the traditional form of storage in computers for many years.
2. Solid State Drive (SSD): This type of hard drive uses flash memory to store data. It has no moving parts and is faster and more reliable than an HDD. SSDs are becoming increasingly popular in computers and other electronic devices due to their speed and durability.

PATA, SATA, SAS, and SCSI are all types of interfaces used for connecting hard drives to a computer or other device.

1. PATA (Parallel ATA): This is an older type of interface that uses parallel cables to connect hard drives to a motherboard. It is also known as IDE (Integrated Drive Electronics). PATA interfaces are slower than newer interfaces like SATA and are becoming less common.
2. SATA (Serial ATA): This is a newer type of interface that uses serial cables to connect hard drives to a motherboard. SATA interfaces are faster than PATA and are now the most commonly used interface for hard drives in desktop and laptop computers.
3. SAS (Serial Attached SCSI): This is a high-performance interface used for connecting hard drives in servers and other high-end systems. SAS interfaces are faster and more reliable than SATA interfaces, and are typically used in enterprise environments.
4. SCSI (Small Computer System Interface): This is an older interface that was commonly used in high-performance systems such as servers and workstations. SCSI interfaces are faster than PATA and SATA interfaces, but are also more expensive and less commonly used today.

* **IOPS** stands for Input/Output (kbs) Operations Per Second. It is a measure of the input/output performance of a storage device, such as a hard drive or solid-state drive (SSD). SSD is fastest.
* In the context of storage devices, **transfer rate** is often used to describe the rate at which data can be read from or written to a storage device, such as a hard drive or solid-state drive (SSD). The transfer rate of a storage device is determined by various factors, including the storage interface (e.g. SATA, SAS, NVMe), the rotational speed of the disk (in the case of hard drives), and the amount of data being transferred.
* 
* Graphical user interface, text, application

  Description automatically generated

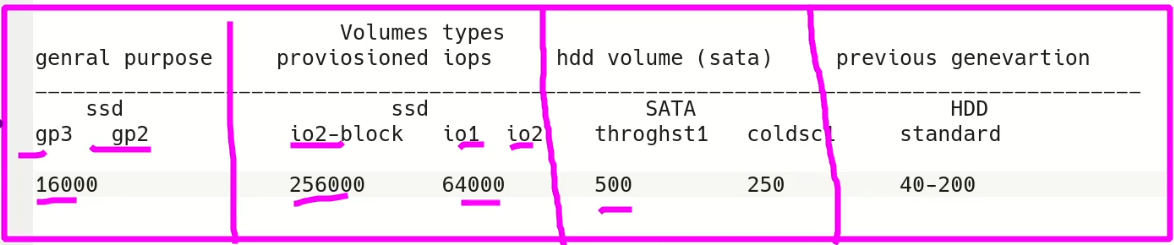
**AWS EBS Storage Types**

AWS EBS (Elastic Block Store) is a cloud-based storage service provided by Amazon Web Services (AWS). EBS provides block-level storage volumes that can be attached to EC2 instances, allowing data to be stored separately from the instance itself. There are several types of EBS storage volumes available, each designed for different use cases and workloads:

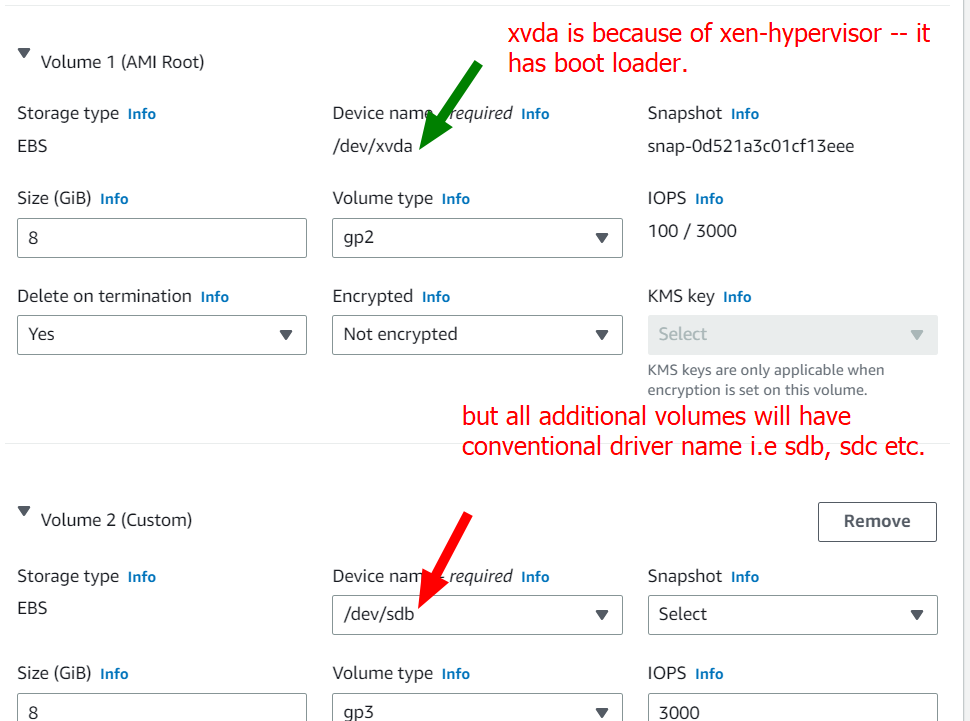
1. General Purpose SSD (gp2): This is a default EBS storage type that provides a balance of price and performance. It is ideal for workloads that require low latency and high throughput, such as boot volumes, small to medium-sized databases, and development and test environments.
2. Provisioned IOPS SSD (io1): This EBS storage type is designed for workloads that require consistent and high performance, such as databases, large transactional workloads, and applications that require low latency. It allows you to provision a specific number of IOPS (Input/Output Operations Per Second) based on your performance requirements.
3. Throughput Optimized HDD (st1): This EBS storage type is optimized for workloads that require large amounts of sequential data throughput, such as big data and data warehouse workloads. It provides low cost storage with high throughput.
4. Cold HDD (sc1): This EBS storage type is designed for infrequent access to data, such as long-term storage and backup. It provides low-cost storage with low performance and high throughput.
5. Magnetic (standard): This is the original EBS storage type and is now considered a legacy option. It provides low-cost storage with low performance and is suitable for infrequent access to data.

Each of these EBS storage types provides different performance and pricing characteristics, allowing you to choose the best option for your specific workload and budget.

EBS Volume Types



* /dev/xvda 🡪 sue to Xen hypervisor
* Xen hypervisor
* Xen is an open-source type-1 hypervisor that allows multiple operating systems to run on a single host machine simultaneously, each isolated in its own virtual machine. Xen was first developed at the University of Cambridge, and it has been widely adopted in various virtualization solutions, including Amazon Web Services (AWS), IBM Power Systems, Oracle VM, and Citrix XenServer.
* The Xen hypervisor provides a lightweight and efficient virtualization layer between the hardware and the guest operating systems, enabling each guest OS to run independently of the others with minimal overhead. This allows for efficient use of hardware resources, as well as improved security, scalability, and flexibility.
* Xen operates at the hardware level, which makes it a type-1 hypervisor. This means that it runs directly on the host machine's hardware, rather than on top of a host operating system like a type-2 hypervisor would. As a result, Xen provides a more efficient and secure virtualization environment.
* Xen supports multiple guest operating systems, including Linux, Windows, and various BSD variants, and it provides advanced features such as live migration, memory overcommitment, and hardware-assisted virtualization. These features make Xen a popular choice for cloud computing and virtualization solutions, including Amazon Elastic Compute Cloud (EC2), which is based on Xen virtualization.



GP2 and GP3 are two types of Amazon Elastic Block Store (EBS) volumes offered by Amazon Web Services (AWS) that are designed for use with EC2 instances.

**GP2** (General Purpose SSD) volumes are the most commonly used EBS volume type. They provide a balance of price and performance, offering low-latency and high-throughput performance for a wide range of workloads, including boot volumes and small to medium-sized databases. GP2 volumes offer a baseline performance of 3 IOPS per GB with a maximum burst performance of up to 16,000 IOPS and a maximum throughput of 250 MB/s.

**GP3** (General Purpose SSD) volumes are a newer, higher-performance EBS volume type that is designed for workloads that require more consistent and higher baseline performance, such as large databases, data warehousing, and log processing. GP3 volumes provide a baseline performance of 3 IOPS per GB, with a minimum baseline of 1,000 IOPS and a maximum burst performance of up to 16,000 IOPS, and a maximum throughput of 1,000 MB/s. GP3 volumes also support elastic volumes, which allow you to dynamically adjust the volume size and performance without downtime.

Both GP2 and GP3 volumes are backed by solid-state drive (SSD) technology, which provides faster and more reliable storage performance than traditional hard disk drives (HDDs). The main difference between the two volume types is that GP3 volumes offer higher baseline performance and the ability to scale performance independent of volume size, while GP2 volumes offer a lower cost-per-GB and high burst performance.

* TASK
* 
* I completed this task on my own Alhamdulillah.
  + I attached an 8 Gb Volume in AWS console
  + The point to remember is that volume and EC2 instance should be in same region and AZ.
  + Rest of the process requires attaching that new volume with the specific Ec2 instance.
  + Then $ fdisk command to further device the volume into partitions or create only 1 partition.
  + Then $ partprobe command
  + Then $ mkfs.ext4 command
  + Then mounting the partition to “/jenkins” mount point
  + Then I made it persistent in /etc/fstab file
  + $ blkid 🡪 command to note UUID