**AWS Cloud Notes**

**What is cloud computing?**

Cloud computing is the **on-demand** delivery of IT resources **over the Internet** with **pay-as-you-go pricing**. Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services, such as computing power (CPU, RAM, GPU), storage, and databases, on an as-needed basis from a cloud provider like Amazon Web Services (AWS).

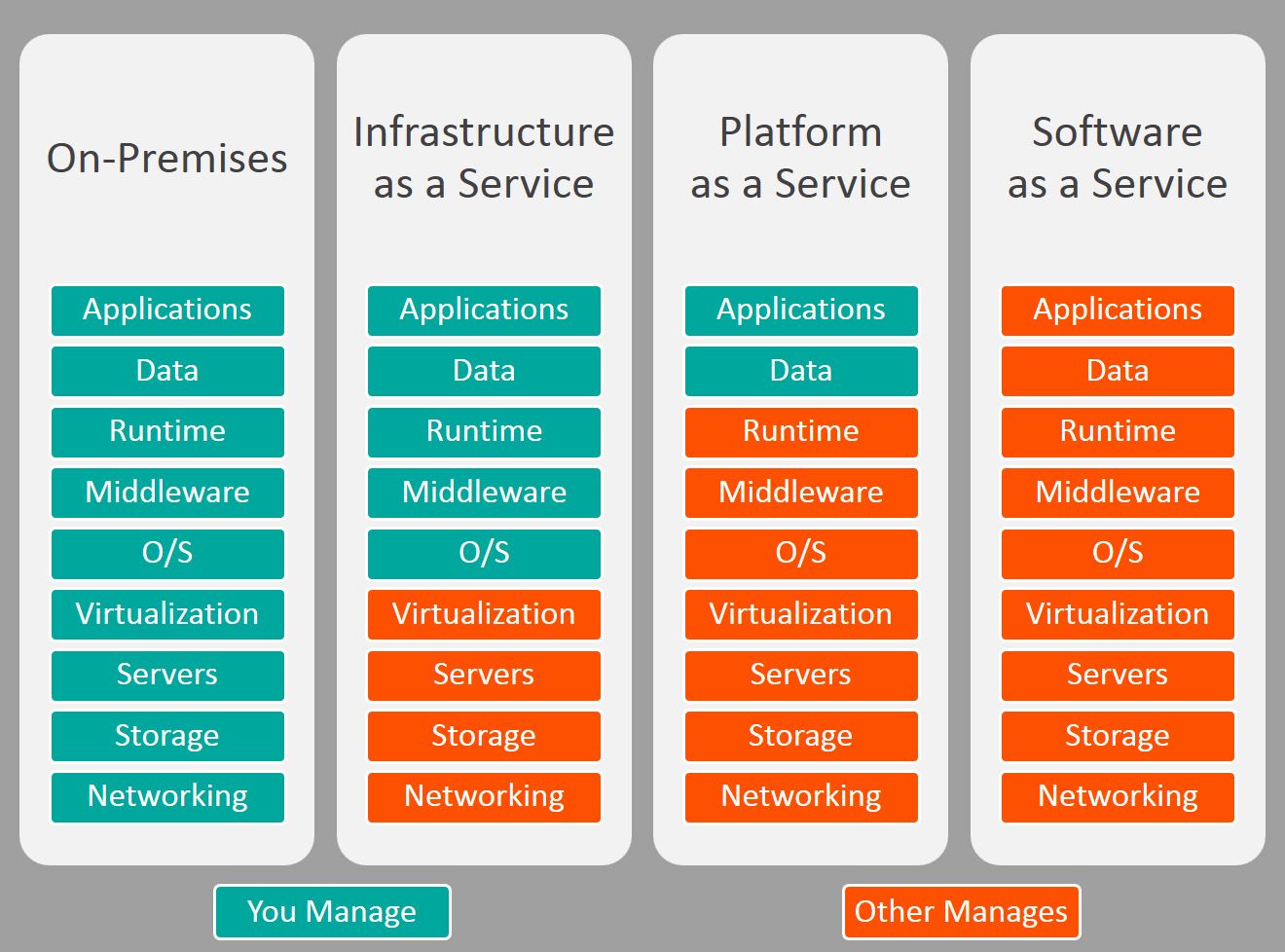
Pay-as-you-go means, based on the resources used and for the duration those resources are used only that will be billed.

**Cloud Service Providers**

* **AWS -** Amazon Web Services
* **Azure -** Microsoft
* **GCP -** Google Cloud Platform

|  |  |
| --- | --- |
| **On-Premises Data Center** | **Cloud Computing** |
| * Capital expenditure (initial investment) * Present on-premises * 1-2 weeks for purchasing the server * Labors * Rack and stack * Limited Servers, difficult to expand * Technical staff * Cooling facility * Electricity and power backup * Physical space * Maintenance and patches | * No capital expenditure * Present in remote location * On demand (use immediately) * No need of any labor * AWS will manage rack and stack * There is no limit of servers, easy to expand * AWS will manage technical staffs * AWS will take care of cooling facility, electricity and power backup, physical space, Maintenance and patches |

**Cloud Service Models**

****

**Infrastructure as a Service (Iaas)**

Infrastructure as a service (IaaS) is a service model that delivers computer infrastructure on an outsourced basis to support various operations. Typically IaaS is a service where infrastructure is provided as outsourcing to enterprises such as networking equipment, devices, database, and web servers.

It simply provides the underlying security, networking, and servers for developing such applications, and services, and deploying development tools, databases, etc.

Advantages of IaaS:

* Cost-Effective: Eliminates capital expense and reduces ongoing cost and IaaS customers pay on a per-user basis, typically by the hour, week, or month.
* Website hosting: Running websites using IaaS can be less expensive than traditional web hosting.
* Security: The IaaS Cloud Provider may provide better security than your existing software.
* Maintenance: There is no need to manage the underlying data center or the introduction of new releases of the development or underlying software. This is all handled by the IaaS Cloud Provider.

The various companies providing Infrastructure as a service are Amazon web services, Bluestack, IBM, Openstack, Rackspace, and Vmware.

Disadvantages of laaS :

* Limited control over infrastructure: IaaS providers typically manage the underlying infrastructure and take care of maintenance and updates, but this can also mean that users have less control over the environment and may not be able to make certain customizations.
* Security concerns: Users are responsible for securing their own data and applications, which can be a significant undertaking.
* Limited access: Cloud computing may not be accessible in certain regions and countries due to legal policies.

**Platform as a Service (PaaS)**

PaaS is a category of cloud computing that provides a platform and environment to allow developers to build applications and services over the internet. PaaS services are hosted in the cloud and accessed by users simply via their web browser.

A PaaS provider hosts the hardware and software on its own infrastructure. As a result, PaaS frees users from having to install in-house hardware and software to develop or run a new application. Thus, the development and deployment of the application take place independent of the hardware.

Advantages of PaaS:

* Simple and convenient for users: It provides much of the infrastructure and other IT services, which users can access anywhere via a web browser.
* Cost-Effective: It charges for the services provided on a per-use basis thus eliminating the expenses one may have for on-premises hardware and software.
* Efficiently managing the lifecycle: It is designed to support the complete web application lifecycle: building, testing, deploying, managing, and updating.
* Efficiency: It allows for higher-level programming with reduced complexity thus, the overall development of the application can be more effective.

The various companies providing Platform as a service are Amazon Web services Elastic Beanstalk, Salesforce, Windows Azure, Google App Engine, cloud Bees and IBM smart cloud.

Disadvantages of Paas:

* Limited control over infrastructure: PaaS providers typically manage the underlying infrastructure and take care of maintenance and updates, but this can also mean that users have less control over the environment and may not be able to make certain customizations.
* Dependence on the provider: Users are dependent on the PaaS provider for the availability, scalability, and reliability of the platform, which can be a risk if the provider experiences outages or other issues.
* Limited flexibility: PaaS solutions may not be able to accommodate certain types of workloads or applications, which can limit the value of the solution for certain organizations.

**Software as a Service (SaaS)**

Software-as-a-Service (SaaS) is a way of delivering services and applications over the Internet. Instead of installing and maintaining software, we simply access it via the Internet, freeing ourselves from the complex software and hardware management. It removes the need to install and run applications on our own computers or in the data centers eliminating the expenses of hardware as well as software maintenance.SaaS applications are sometimes called Web-based software, on-demand software, or hosted software.

Advantages of SaaS:

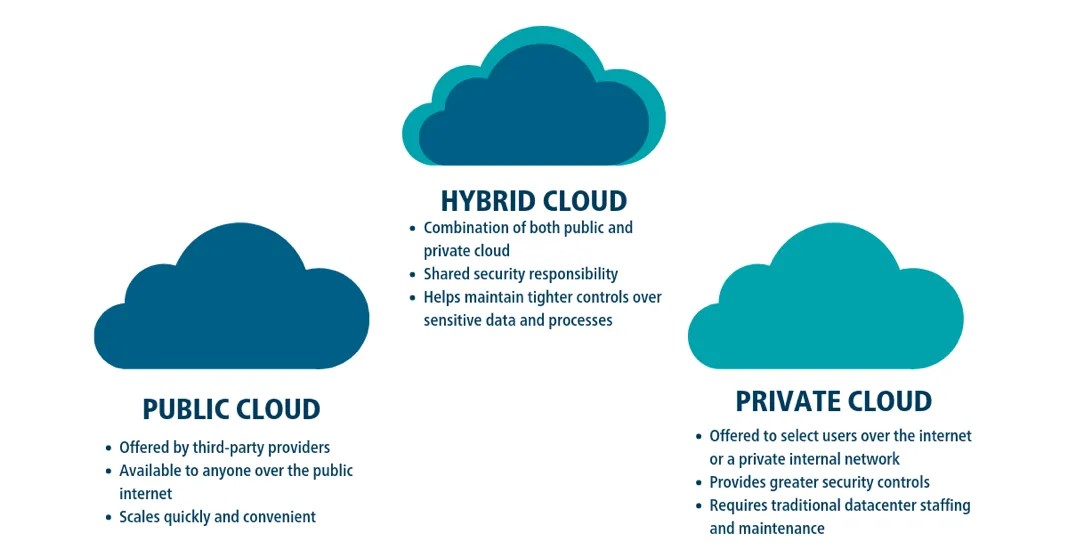
* Cost-Effective: Pay only for what you use.
* Reduced time: Users can run most SaaS apps directly from their web browser without needing to download and install any software. This reduces the time spent in installation and configuration and can reduce the issues that can get in the way of the software deployment.
* Accessibility: We can Access app data from anywhere.
* Automatic updates: Rather than purchasing new software, customers rely on a SaaS provider to automatically perform the updates.
* Scalability: It allows the users to access the services and features on-demand.

The various companies providing Software as a service are Cloud9 Analytics, Salesforce.com, Cloud Switch, Microsoft Office 365, Big Commerce, Eloqua, dropBox, and Cloud Tran.

Disadvantages of Saas :

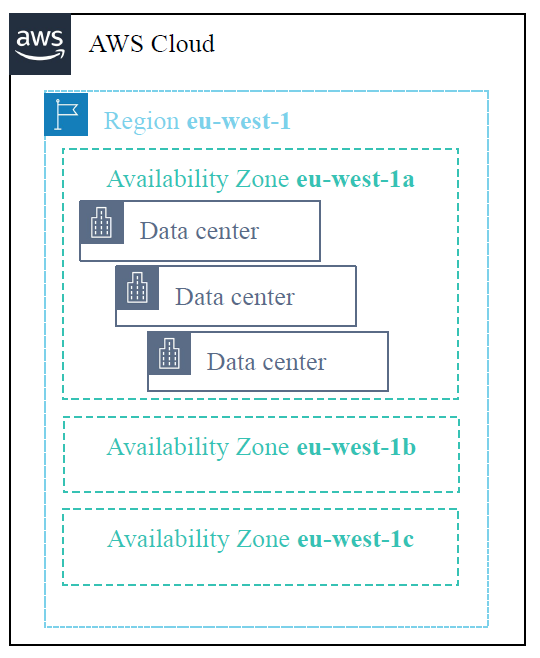
* Limited customization: SaaS solutions are typically not as customizable as on-premises software, meaning that users may have to work within the constraints of the SaaS provider’s platform and may not be able to tailor the software to their specific needs.
* Dependence on internet connectivity: SaaS solutions are typically cloud-based, which means that they require a stable internet connection to function properly. This can be problematic for users in areas with poor connectivity or for those who need to access the software in offline environments.
* Security concerns: SaaS providers are responsible for maintaining the security of the data stored on their servers, but there is still a risk of data breaches or other security incidents.
* Limited control over data: SaaS providers may have access to a user’s data, which can be a concern for organizations that need to maintain strict control over their data for regulatory or other reasons.

**Cloud Deployment Models**



**AWS Regions**

AWS Region is a physical geographical location where data centers are present. Each region can have two or more availability zones (AZs), each availability zone separated by a distance of approximately 150 - 200 kms. Each availability zone has multiple data centers. Data centers have approximately 50,000 to 80,000 servers.



**What basis regions are selected?**

* Data governance and legal requirements
* Proximity to customers (latency)
* Services available within the region
* Costs (vary by region)
* **AWS Identity and Access Management (IAM)**

AWS Identity and Access Management (IAM) allows you to control access to compute, storage, database, and application services in the AWS Cloud. IAM can be used to handle authentication, and to specify and enforce authorization policies so that you can specify which users can access which services.

With IAM, you can manage:

* Who can access the resources
* Which resources can be accessed
* How these resources can be accessed

**Lab: Create an IAM user, give access to S3 bucket. Create an S3 bucket and upload some files to it.**

**Step 1 - Creating IAM user**

* Login to AWS Console
* Search for **IAM**
* Select **Users** from left pane
* Click on **Create user**
* Provide a username
* Check this option **Provide user access to the AWS Management Console**
* Select **I want to create an IAM User**
* Uncheck **Users must create a new password at next sign-in**
* Click on **Next**
* Select **attach policies directly**
* Under permissions policies, search for **AmazonS3FullAccess** and check that option
* Clickon **Next**
* Click on **Create User**
* Click on **Download .csv file**
* Now return to users tab
* Open the new user created by clicking the **user name**
* Under **security credentials**, copy the **console sign-in link** and paste it in browser’s incognito mode.
* Get the IAM username and password from the downloaded .csv file
* Paste the respective credentials and click on **sign in**

Note: Now we are logged in as the newly created IAM user

**Step 2 - Creating S3 bucket and uploading files**

* Search and open **S3**
* Click on **Create bucket**
* Enter bucket name (follow the rules of naming a bucket)
* Click on **Create bucket**
* Open the newly created bucket by clicking the **bucket name**
* Click on **Upload**
* Click on **Add files** or **Add folder** based on the requirement
* Select the required file or folder from the device and click on **Open**
* Scroll down and click on **Upload**
* Now we can find the uploaded file in the bucket.

Note: We can delete the files in the bucket by selecting the files and folders and click on **delete.** We can delete S3 bucket, but before that the bucket should be empty, select the bucket and click on **empty** and provide the empty confirmation, now delete the bucket by selecting the bucket and click on **delete**.

**Lab: Create an IAM Users Group, create 2 users say user1 and user2. Give S3 access to user1 and EC2 access to user2. Create new Users Group add EC2FullAccess to the created group, add the two users user1 and user2 to the created group**

**Step 1 - Create user1**

* Login to AWS Console
* Search for **IAM**
* Select **Users** from left pane
* Click on **Create user**
* Provide a username (user1)
* Check this option **Provide user access to the AWS Management Console**
* Select **I want to create an IAM User**
* Uncheck **Users must create a new password at next sign-in**
* Click on **Next**
* Select **attach policies directly**
* Under permissions policies, search for **AmazonS3FullAccess** and check that option
* Clickon **Next**
* Click on **Create User**
* Click on **Download .csv file**
* Now return to users tab

**Step 2 - Create user2**

* Click on **Create user**
* Provide a username (user2)
* Check this option **Provide user access to the AWS Management Console**
* Select **I want to create an IAM User**
* Uncheck **Users must create a new password at next sign-in**
* Click on **Next**
* Select **attach policies directly**
* Under permissions policies, search for **AmazonEC2FullAccess** and check that option
* Clickon **Next**
* Click on **Create User**
* Click on **Download .csv file**
* Now return to users tab

**Step 3 - Create User group**

* Select **User groups** from left pane
* Click on **Create group**
* Enter a group name
* Add group users now by selecting the users (user1 and user2)
* Under **Attach permissions policies** search for **AmazonEC2FullAccess** and select it
* Click on **Create user group**
* The group will be created, we can see it in the User groups

Note: Under the user groups find the newly created group and open it.

Under the users in this group find **user1,** open user1 we can find two policies attached to the user, one is S3FullAccess which we gave while creating the user and the other is EC2FullAccess which the user got by being the member of the group.

**Lab: Create Images of EC2 instance**

**Scenario:** Suppose there is an application running in production, suddenly there is a spike in traffic. The currently running EC2 instance cannot handle that load, we need another EC2 instance with the same data and configuration to handle the traffic. Instead of manually creating the EC2 instance with the same configuration and replicating data which requires hours of time, we can create an image which clones the currently running EC2 instance with all the configuration and data.

**Step 1 - Create an EC2 instance**

* Search and select **EC2**
* Open **instances** section
* Click on **Launch instances**
* Enter instance name
* Select OS
* Select existing key pair or create new key pair
* Allow **SSH, HTTPS** and **HTTP** traffic from the internet
* Click on **Launch Instance** at the bottom
* Open the instance in command prompt
* Create some files and install a web server
* Start the web server

**Step 2 - Create an image**

* Select the EC2 instance
* Click on **Actions**
* Click on **Image and Templates**
* Click on **Create Image**
* Enter an image name
* Click on **Create image**
* New image will be created, this can be found in **AMIs** section under **Images**

**Note:** When an image of an EC2 instance is created all the configurations of the instance and data of the VM is cloned. Data will be backed up and we can see the backed up data file under the Snapshots section.

To delete the image we need to first delete the EC2 instance which was launched using the image, then delete the image and then delete the backup in the **Snapshots** section.

**Lab: How to create a template and launch instance from it**

**Step 1 - Create an EC2 instance**

* Search and select **EC2**
* Open **instances** section
* Click on **Launch instances**
* Enter instance name
* Select OS
* Select existing key pair or create new key pair
* Allow **SSH, HTTPS** and **HTTP** traffic from the internet
* Click on **Launch Instance** at the bottom

**Step 2 - Create a template using EC2 instance**

* Select the EC2 instance
* Click on **Actions**
* Click on **Image and Templates**
* Click on **Create template from instance**
* Enter template name
* Make any changes to the configuration (only if required)
* Click on **Create launch template**

**Step 3 - Launch an instance using template**

* Find the newly created template under **Launch Templates** section
* Select the template
* Click on **Actions**
* Click on **Launch instance from template**
* Enter the name at the bottom in **Resource tags** section
* Click on **Launch instance**

Note: A new instance will be launched and running, connect to the instance either using command prompt or putty.

**S3 - Simple Storage Service**

It is an object storage service

**Features of S3**

* It can have images, videos, files and documents (these are considered as objects)
* This offers unlimited storage
* It is designed for high availability (minimum down time) - 99.99%
* Scalability
* High durability - 99.99999999999 (11 9s of durability) The chance of losing one object the data once in 10 thousand years
* Supports versioning (A file can be uploaded multiple times with the same name after making necessary changes, each version of the file will be stored as versions)
* Launch static website in S3 bucket

**S3 Storage Classes**

Amazon S3 provides a range of storage classes that we can choose from based on the data access, resiliency, and cost requirements of our workloads. S3 storage classes are built in such an efficient manner that they can address almost any of our use cases, be it performance needs or data residency requirements or variable/unknown/infrequent access patterns or archival storage or low cost and so on.

1. **S3 Standard**

★ S3 Standard is the default storage class provided by AWS.

★ If we don’t specify the storage class while uploading an object in S3, Amazon S3 automatically assigns the object to S3 Standard storage class.

★ It is suitable for frequently accessed data as it delivers low latency and high throughput.

★ It is also the most expensive storage class among the available storage classes.

1. **S3 Intelligent Tiering**

★ S3 Intelligent Tiering is designed as an intelligent storage class.

★ S3 intelligent tiering is designed in such a way that it analyzes the access frequency of the stored data and automatically moves the data to the most cost-effective access tier, without performance impact, retrieval charges or operational overhead.

★ It offers three types of Access Tiers, namely, Frequent Access Tier, Infrequent Access Tier, and Archive instant Access Tier.

★ Similar to S3 Standard Storage Class, it delivers milliseconds latency and high throughput. But, unlike S3 Standard which is suitable for frequently accessed data, S3 Intelligent Standard Tiering is suitable for frequently, infrequently as well as rarely accessed data because of its possession of three types of access tiers.

★ It is suitable for storing long-lived data where access patterns are unknown.

AWS decides which is the suitable class for the data (objects) to be stored based on our access pattern

1. **S3 Standard IA**

★ S3 Standard-IA (S3 Standard — Infrequent Access) is suitable for data that is accessed less frequently, but requires rapid access when needed.

★ Similar to S3 Standard, it offers a high throughput and low latency, but with a low per GB storage price and per GB retrieval charge.

★ Due to the low cost and high-performance storage, S3 Standard-IA can be used for long-term storage, backups, and as a data store for disaster recovery files.

1. **S3 One Zone IA**

★ S3 One Zone — IA (S3 One Zone — Infrequent Access) is similar to S3 Standard-IA in terms of use case i.e. it is also suitable for data that is accessed less frequently and which requires rapid access when needed.

★ Similar to S3 Standard-IA, it also offers a high throughput and low latency.

★ S3 One Zone-IA stores the object data in only one Availability Zone, unlike the rest of the storage classes which store object data redundantly across multiple geographically availability zones.

★ Since the data is stored in a single availability zone, the cost to store the data is somewhat less in comparison to other storage classes.

★ But, storage of data in a single availability zone means that the stored data is not resilient to the physical loss of the availability zone that may occur due to natural calamities such as flood, earthquake and so on.

★ Also, storage of data in a single availability zone results in less availability (99.5%) in comparison to other storage classes (99.999999999%).



**Lab: Create an S3 bucket, upload a file once, make some changes to the same file and upload the file with the same filename to the same bucket. Now there will be only one file stored in the bucket, but S3 bucket will also have previous versions of the file.**

**Step 1 - Creating an S3 Bucket**

* Search and open **S3**
* Click on **Create bucket**
* Enter bucket name (follow the rules of naming a bucket)
* **Enable** bucket versioning
* Click on **Create bucket**
* Open the newly created bucket by clicking the **bucket name**

**Step 2 - Upload a file**

* Open notepad
* Type in “Version 1”
* Save and Exit
* Open the newly created bucket by clicking the **bucket name**
* Click on **Upload**
* Click on **Add files** or **Add folder** based on the requirement
* Select the notepad file from the device and click on **Open**
* Scroll down and click on **Upload**
* Now we can find the uploaded file in the bucket.

**Step 3 - Make changes to the file**

* Open the file on the device using notepad, make some changes and save the file.
* Upload the same file to the existing S3 bucket.
* Now there will be only one file in the bucket.

We can access the previous version of the file by opening the file and access previous versions under the versions tab.

**Elastic Block Storage**

**Features**

* Can be used to host dynamic website
* It offers block level storage
* Volumes are automatically replicated in availability zones
* Data is arranged in the form of a block
* High durability
* High availability
* High performance
* It contains file system

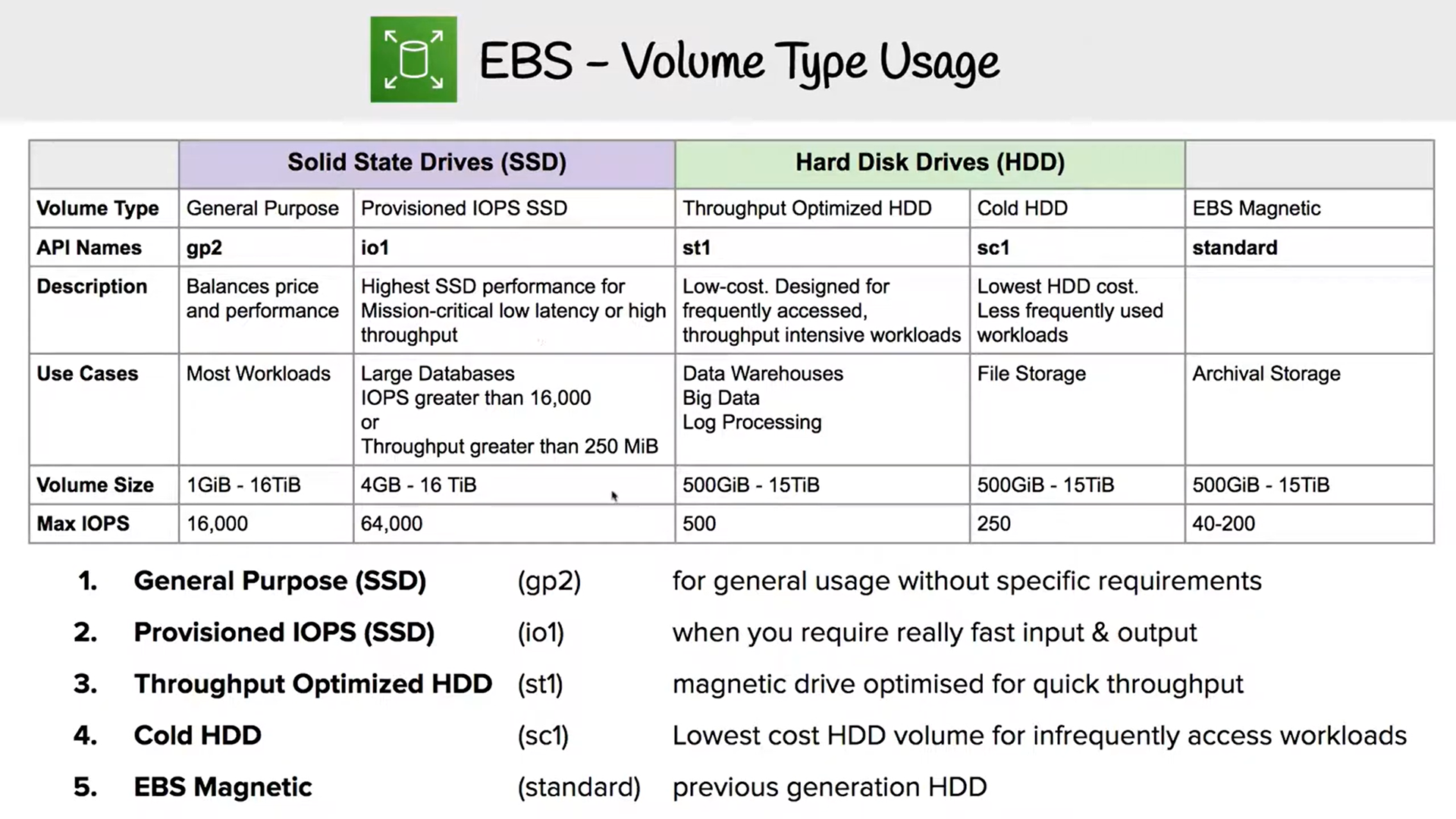
If we need to make some changes in a file of 1GB in an object storage (S3) we need to update the whole file, in block storage we can modify only the required content in the file.

S3 bucket can’t run applications because it doesn't have a file system, EBS can run applications because it has a filesystem.

Boot volume by default is EBS (where OS is present)

Enterprise applications uses EBS volume to store the data

**EBS Volume Types**



**Features**

Snapshots

* Point-in-time snapshots (incremental changes)
* Recreate a new volume at any time

Encryption

* Encrypted Amazon EBS volumes
* No additional cost

Elasticity

* Increase capacity
* Change to different types (SSD or HDD)

**Amazon EBS: Volumes, IOPS and Pricing**

**Volumes**

* Amazon EBS volumes persist independently from the instance.
* All volume types are charged by the amount that is provisioned per month

**IOPS**

* General Purpose SSD: Charged by the amount that you provision in GB per month until storage is released.
* Magnetic: Charged by the number of requests to the volume
* Provisioned IOPS SSD: Charged by the amount that you provision in IOPS (multiplied by the percentage of days that you provision for the month),

**Snapshots**

* Added cost of Amazon EBS snapshots to Amazon S3 is per GB-month of data stored.

**Data Transfer**

* Inbound data transfer is free
* Outbound data transfer across regions incurs charges

Note: Taking multiple snapshots of the same volume keeps incremental backup of the file (For example even though it shows multiple backups of 8GB files only incremental updates will be saved. Thus resulting in saving space)

**Lab: This lab is designed to show you how to create an Amazon EBS volume. After you create the volume, you will attach the volume to an Amazon EC2 instance, configure the instance to use a virtual disk, create a snapshot and then restore from the snapshot.**

**Step 1 - Create an EC2 instance**

* Search and select **EC2**
* Open **instances** section
* Click on **Launch instances**
* Enter instance name, say **instance-1**
* Select OS
* Select existing key pair or create new key pair
* Allow **SSH, HTTPS** and **HTTP** traffic from the internet
* Click on **Launch Instance** at the bottom

**Step 2 - Attach an EBS volume**

* Click on **Volumes** under **Elastic Block Store**
* Click on **Create volume**
* Enter size as **10**
* Make sure the availability zone is same as the instance
* Click on **Create volume** at the bottom
* Select the newly created volume
* Click on **Actions** and click on **Attach volume**
* Select the instance to which the volume should be attached
* Select a device name
* Click on **Attach volume**

**Step 3 - Configure the newly attached volume and create files**

* Connect to the instance using command prompt or putty
* In the linux system, change to super user (**sudo su**)
* See the list of the disks (**fdisk -l**), find the newly attached disk (Eg: /dev/xvdb)
* Create a partition (**fdisk /dev/xvdb**)
* Enter **n**, **p**, **1**, **enter**, **+5G** and **w**
* Create a file system (**mkfs.ext4 /dev/xvdb1**)
* Mount the file system, create a directory first (**mkdir /mnt/data**)
* Mount the file system to /mnt/data directory (**mount /dev/xvdb1 /mnt/data**)
* Now change the directory to the newly attached volume (**cd /mnt/data**)
* Create some files (**touch f{1..10}**)

**Step 4 - Create a second EC2 instance**

* Search and select **EC2**
* Open **instances** section
* Click on **Launch instances**
* Enter instance name, say **instance-2**
* Select OS
* Select existing key pair or create new key pair
* Allow **SSH, HTTPS** and **HTTP** traffic from the internet
* Click on **Launch Instance** at the bottom

**Step 5 - Create a snapshot of the attached volume of instance-1**

* Open **Snapshots** under Elastic Block Store
* Click on Create snapshot
* Select the volume id of the attached volume from the list
* Click on **Create snapshot at the bottom**

Note: By creating a snapshot we take the backup of the entire volume, which means all the data in the storage will be backed up.

**Step 6 - Create a volume from the snapshot and attach it to instance-2**

* Select the snapshot
* Click on **Actions** and click on **Create volume from snapshot**
* Click on **Create volume**
* A new volume will be created
* Attach the newly created volume to **instance-2,** select the newly created volume
* Click on **Actions** and click on **Attach volume**
* Select the instance-2 id
* Select a device name
* Click on **Attach volume** at the bottom

**Step 7 - Connect to instance-2 and mount the partition of added volume**

* **sudo su**
* **fdisk -l** (here we can see the partition ‘/dev/xvdb1’ from the snapshot of another volume, the partition will have the file system as we have already created it earlier. Now we have to mount the filesystem to locate it and use it)
* Create a directory in /mnt (**mkdir /mnt/data**)
* Mount the partition, we need to mount the partition so that we can locate the disk on our device (**mount /dev/xvdb1 /mnt/data**)
* Now change the current directory (**cd /mnt/data**)
* List all the files (**ls**)

Note:

* Now we can see the files **f1, f2, f3, f4, f5, f6, f7, f8, f9 and f10**
* The data from the instance-1 volume was taken as a snapshot and a new volume was created from that snapshot.
* The new volume consists of all the data. So, the instance-2 has all the data of the snapshot of the attached volume of instance-1.

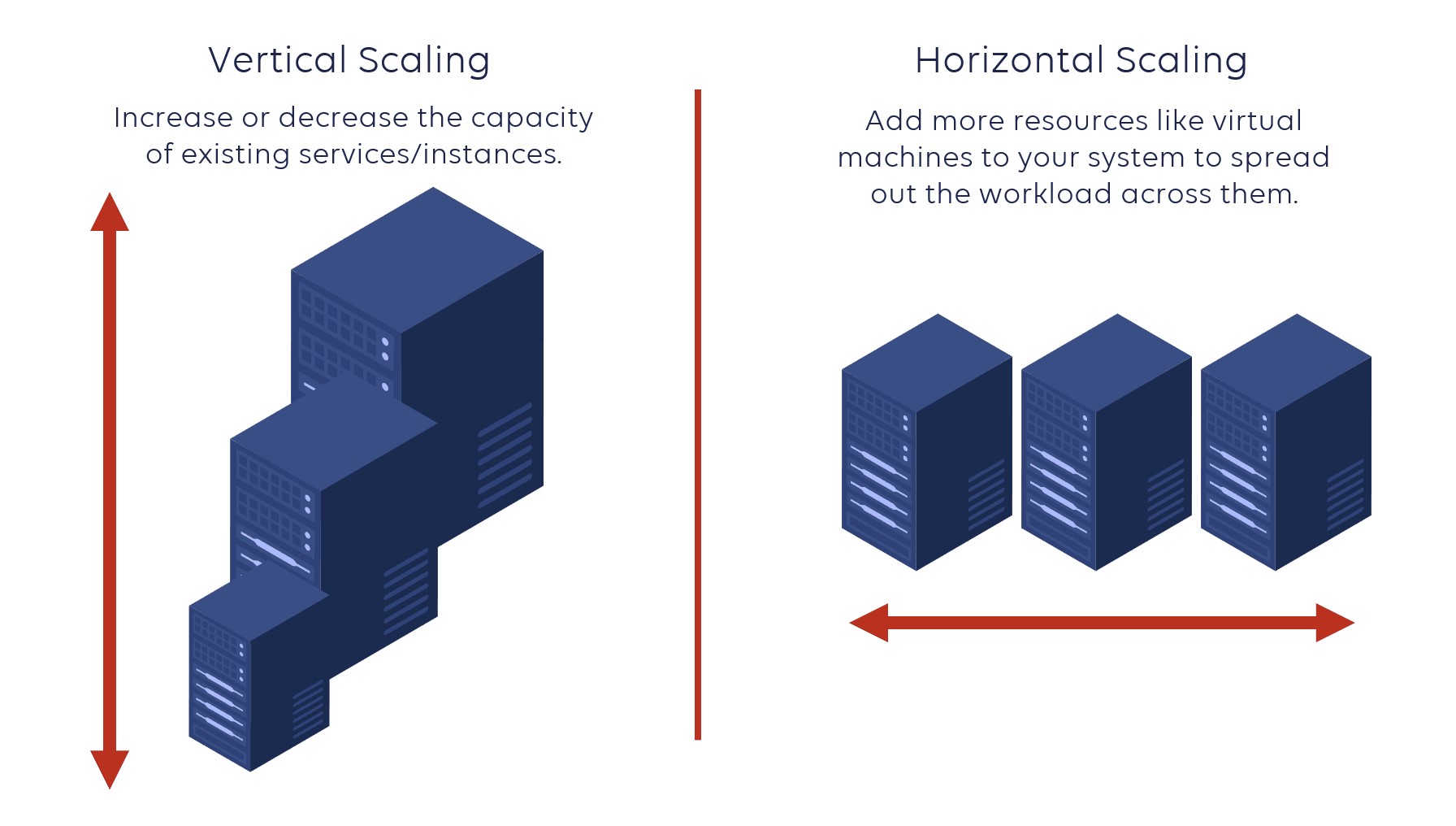
**Autoscaling**

Scaleup - scaleout (Vertical scaling - horizontal scaling)

Scaledown - scalein (Vertical scaling - horizontal scaling)

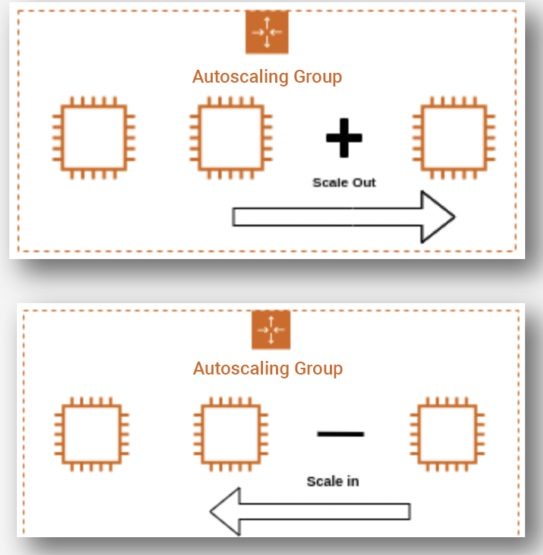
* Vertical scaling -> Increasing or decreasing the resources of the existing virtual machines according to the demand is called vertical scaling.

Example: There is an EC2 instance (which has 1 vCPU, 1GB RAM) if the resources of the of the existing EC2 instance is increased (for example to increase 1 vCPU to 3 vCPU and 1GB RAM to 2 GB RAM) this called **scaleup.** If the existing resources of an EC2 instance is decreased it is called **scaledown**

****

* Horizontal scaling -> Adding or removing new virtual machines to the existing virtual machines one after the other according to the demand is called horizontal scaling.

Example: There is an EC2 instance (which has 1 vCPU, 1GB RAM) if we add some more instances (of same configuration) to the existing instance it is called **scaleout** (horizontal scaling)**,** if we remove some EC2 instances (of same configuration) it is called **scalein** (horizontal scaling)



**Lab: Auto Scaling EC2 Instances**

**Step 1 - Create two security groups**

1. **Security group for application load balancer**

* Search and open **EC2**
* Under **Network & Security**, open **Security Groups**
* Click on **Create security group**
* Enter a security group name (Ex: **Auto Load Balancer**)
* Give a description
* In **Inbound rules**, select **add rule**
* In type, select **HTTP** and source **Anywhere-IPv4**
* Click on **Create security group**

1. **Create security group for auto scaling group**

* Search and open **EC2**
* Under **Network & Security**, open **Security Groups**
* Click on **Create security group**
* Enter a security group name (Ex: **Auto Scaling**)
* Give a description
* In **Inbound rules**, select **add rule**
* In type, select **Custom TCP** and source search and select **Auto Load Balancer**
* Again click on **add rule**
* In type, select **SSH** and for source, select **Anywhere-IPv4**
* Click on **Create security group**

**Note:** No need to make any changes to outbound rules for both the security groups

**Step 2 - Create launch template**

* Under **Instances**, click on **Launch Templates**
* Click on **Create launch template**
* Enter launch template name (Ex: **AutoScaleTemplate)**
* **Check** Auto Scaling guidance
* Select the required OS (Ex: **Ubuntu - Free Tier**)
* Select instance type (Ex: **t2.micro**)
* Select an existing key pair or create a new key pair
* In network settings, select auto scaling security group (Ex: **Auto Scaling)** under **security groups**
* In **Advance details,** scroll down to find **User data** section
* Paste the following code in the available text box.

#! /bin/bash

sudo apt update -y

sudo apt install apache2 -y

sudo systemctl start apache2

sudo systemctl enable apache2

* The above code is to install a web server to the EC2 instance which gets created
* Click on **Create launch template**

**Step 3 - Create Auto Scaling Group**

* Under **Auto Scaling** section, click on **Auto Scaling Groups**
* Enter auto scaling group name (Ex: DemoAutoScaling)
* Select launch template, **AutoScaleTemplate** and click on **Next**
* Under **Network**, select the required availability zones and click on **Next**
* Under **Load balancing,** select **Attach to a new load balancer**
* Under **Attach to a new load balancer**, choose the load balancer type to be **Application Load Balancer**
* Choose load balancer scheme to be **Internet-facing**
* Select default routing to **Create a target group,** new target name will be created by default
* Under **Health checks**, check **Turn on Elastic Load Balancing health checks** and click on **Next**
* Enter Desired capacity (Ex: **2**)
* Set minimum desired capacity and maximum desired capacity (**1** and **3**)
* Click on **Next**
* Click on **Next**
* Click on **Next**
* Click on **Create Auto Scaling group**

**Note:**

* Now two EC2 instances will be running
* After creating a load balancer, the security group of the auto scaling group will be selected. We have to change it to load balancer security group which we created first
* In **EC2**, under **Load Balancing**, select **Load Balancers**
* Open the load balancer, same name (Ex: **DemoAutoScaling-1**) we gave while creating a load balancer in auto scaling.
* Scroll down, under Security section, click on Edit
* Delete the existing security group
* Search and select load balancer security group (Ex: **Auto Load Balancer**)
* Click on **Save changes**

**Step 4 - Create scaling policy**

* Under Auto Scaling groups, open the auto scaling group (Ex: **DemoAutoScaling**)
* Click on **Automatic scaling**
* Click on **Create dynamic scaling policy**
* Set target value to be 45 and click on **Create** (whenever the CPU utilization crosses 45% a new EC2 instance will be created and after 300 seconds [instance warmup] the load balancer will send the traffic to the newly created EC2 instance)

**Note:**

* To monitor the policies, we can search and open **cloud watch,** under the **Alarms** section, open **All alarms.**
* We can see two alarms, one is because of the policy we created, the other is created by the AWS.
* If the state is **OK**, no problem.
* If the state is **In alarm,** EC2 instances will be scaled-out or scaled-in based on the policy.

**Step 5 - Create fake stress**

* Go to **Auto Scaling groups** and click on the created group
* Under **Instance management**, click on any instance id and connect to it
* Type the following commands,

1. sudo su
2. apt update -y
3. apt install stress -y
4. stress --cpu 12 --timeout 240

* In the above command, 12 represents the number of CPU threads and 240 represents the number of seconds the threads should run.

**Note:**

* We can see the CPU utilization under **cloud watch** and see if any alarms are triggered (when CPU utilization crosses 45%)
* When CPU utilization crosses 45% alarm is triggered and a new EC2 instance is created. We can check the newly created instance under **Instances** section
* Just terminating the EC2 instances won't stop the experiment, because the auto scaling group keeps on creating new EC2 instances until the desired number is matched.
* We have to delete the auto scaling group we created to stop this.
* Deleting the auto scaling group also deletes the EC2 instances associated with it.
* Before deleting the auto scaling group take the images of EC2 instances (if data and configuration of that EC2 instance are required)
* Load balancer can be deleted from **Load Balancing** section