**AWS Self-Notes**

**1) AWS:-** AWS stands for Amazon Web Services. AWS is a platform that provides on-demand resources for hosting web services, storage, networking, databases and other resources over the internet with a pay-as-you-go pricing.

**2) Data Centre:-**

* Data centre is a facility that houses computer systems and associated components.
* Such as storage systems and networking equipment, to efficiently store, process, manage, and disseminate large amounts of data.

**\*Traditional Data Centres (on premises Data Centre):-**

* These data canters are typically owned, operated, and maintained by the organization itself, rather than being outsourced to a third-party provider.
* It includes Servers, networking equipment (router, switches, firewalls….), storage system, power and internet supply, cooling, manpower.

**\*Drawback of on premises data centre:-**

* Capital Expenditure
* Operational Expenditure
* Resource Management
* Expertise required
* Environment impact
* Scalability
* Backup is very complicated task (and many more)

**\*Cloud Service provider in market:-**

* Amazon Web Services
* Microsoft Azure
* Google Cloud Platform
* IBM Cloud
* Digital Occian.

**\*Why AWS why not other cloud provider?**

* Huge Market
* Scalability
* Availability
* Wide Service Offering
* Cost Savings
* Huge Community and Support.

**3) Cloud Computing:-**

* It is involves delivering and consuming computing services over the internet.
* Rather than owning and managing physical hardware or on-premises servers, users can access a variety of computing resources, such as servers, storage, databases, networking, software, analytics, and intelligence, through a third-party provider.
* These services are hosted in data centres that are maintained and managed by the cloud service provider.

**\*Types of cloud computing:-**

* **IAAS:-** IAAS is a cloud service that runs services on “pay-for-what-you-use” basis IaaS workers include Amazon Web Services, Microsoft Azure and Google Compute Engine.

**Users:-** IT Administrators.

* **PAAS:-** PAAS runs cloud platforms and runtime environments to develop, test and manage software.

**Users:-** Software Developers.

* **SAAS:-** In SAAS, cloud workers host and manage the software application on a pay-as-you-go pricing model.

**User:-** End Customer.

**4)** [**AWS Benefits and Advantages**](https://intellipaat.com/blog/aws-benefits-and-drawbacks/#no1)**:-**

* + [User-friendly](https://intellipaat.com/blog/aws-benefits-and-drawbacks/#no2)
  + [Flexible](https://intellipaat.com/blog/aws-benefits-and-drawbacks/#no3)
  + [Secure](https://intellipaat.com/blog/aws-benefits-and-drawbacks/#no4)
  + [Cost-effective](https://intellipaat.com/blog/aws-benefits-and-drawbacks/#no5)
  + [Reliable](https://intellipaat.com/blog/aws-benefits-and-drawbacks/#no6)
  + [Scalable and Elastic](https://intellipaat.com/blog/aws-benefits-and-drawbacks/#no7)
  + [Highly Performant](https://intellipaat.com/blog/aws-benefits-and-drawbacks/#no8)

**5)** [**Disadvantages of AWS**](https://intellipaat.com/blog/aws-benefits-and-drawbacks/#no9)**:-**

* + [Limitations](https://intellipaat.com/blog/aws-benefits-and-drawbacks/#no10)
  + [Lack of Experts](https://intellipaat.com/blog/aws-benefits-and-drawbacks/#no11)
  + [Price Variations](https://intellipaat.com/blog/aws-benefits-and-drawbacks/#no12)
  + [General Issues](https://intellipaat.com/blog/aws-benefits-and-drawbacks/#no13)

**6) Services provided by Amazon Web Services are:-**

* [Amazon Elastic Cloud Compute (EC2)](https://intellipaat.com/blog/what-is-amazon-ec2-in-aws/)
* [Amazon Simple Storage Service (S3)](https://intellipaat.com/blog/what-is-amazon-s3/)
* [Amazon Virtual Private Cloud (VPC)](https://intellipaat.com/blog/aws-vpc-peering-tutorial/)
* [Amazon Relational Database Services (RDS)](https://intellipaat.com/blog/what-is-amazon-rds-in-aws/)

**7**) **Storage in AWS:-**

1. Object storage **(S3)**

2. File Storage **(EFS, Fsx)**

3. Block storage **(EBS, Instance storage)**

**8) User-Data in AWS:-** AWS user-data is the set of commands/data you can provide to a instance at launch time.

**9) EC2:-** Amazon Elastic Compute Cloud (Amazon EC2) (IAAS platform) **provides scalable computing capacity in the Amazon Web Services (AWS) Cloud.**

1)Renting virtual machines (ec2)

2) Scalling the services using ASG.

3) Distribute load across machines. (ELB)

4) Storing data on virtual driver. (EBS)

**10) Instance:-** An instance is **a virtual server in the AWS Cloud.**

**\*Types of instances:-**

**1) General Purpose:-** It is used for email responding systems. **(t2,m3,4)**

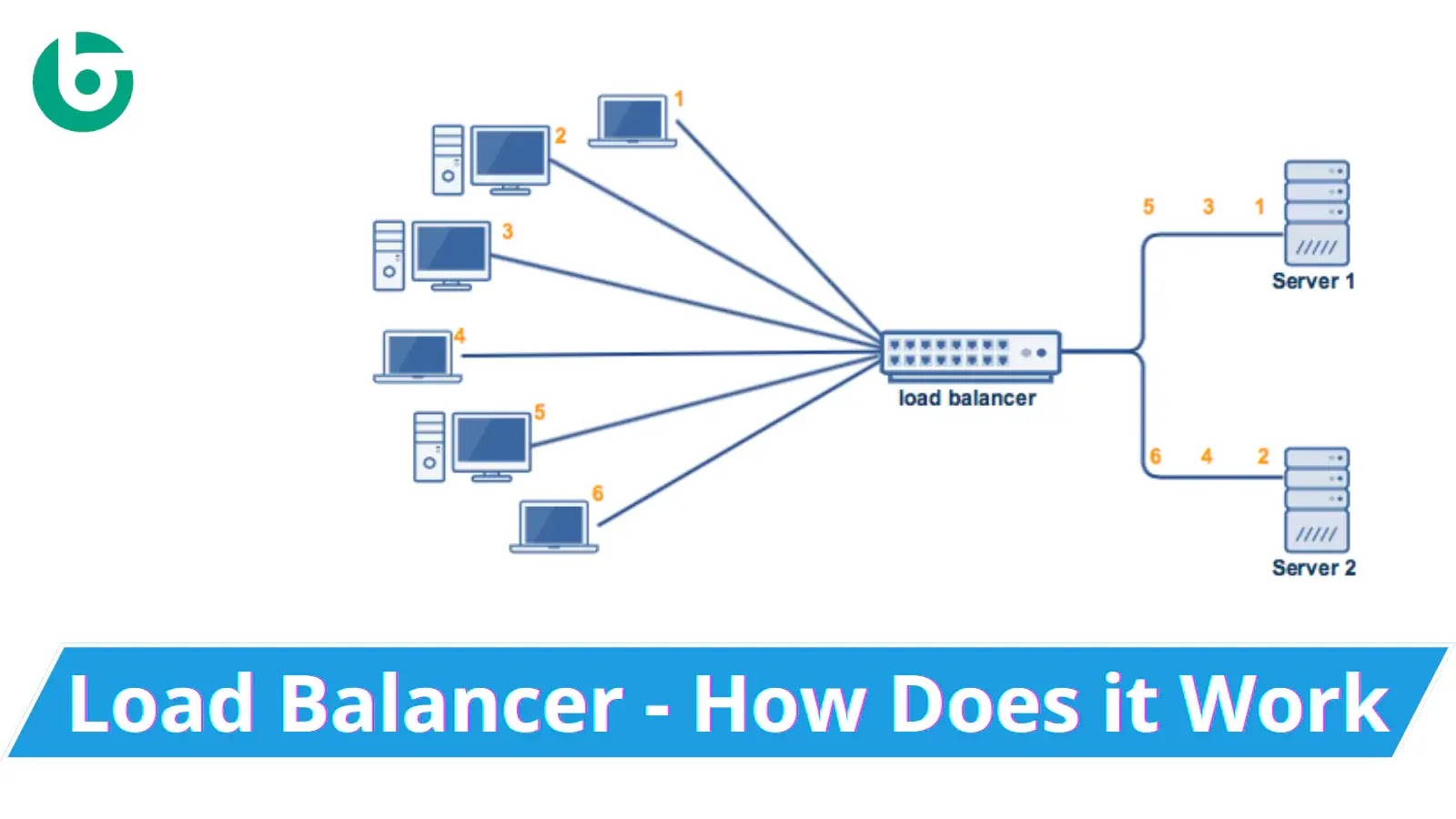
**2) Computer optimized:-** For applications that require a lot of processing from the CPU. **(c4 , c3)**

**3) Memory optimized:-** It is Used for multitasking. **(r3 , x1)**

**4) Storage optimized:-** When your application is of huge size. **(i2 , d2)**

**11) Load Balancer:-**

* It is used to distribute a load or traffic. It is a single point of contact for client.
* It is automatically distributes incoming application traffic across to targets, such as EC2 instances, ensuring high availability and fault tolerance.

****

**\*Elastic load Balancer:-** It is used to scale your load balancer capacity automatically in response to change incoming traffic.

**\*Types of Load Balancer:-**

**1) Application Load Balancer:-**

* It is used to direct user traffic to the public AWS cloud.
* Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic.

1) Best for routing HTTP/HTTPS traffic.

2) Understands content and can route requests based on it.

3) Suitable for modern web applications.

**2) Network Load Balancer:-**

* It is operated at the Layer 4 of the OSI model. It is best suited for load balancing the TCP traffic.
* When you need very high performance then we use NLB Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies ex; video streaming, gaming etc.
* Best for Handling Transmission Control Protocol (TCP) /User Datagram Protocol (UDP) traffic.
* High-performance and handles millions of requests per second.
* Ideal for applications with extreme demand

**3) Classic Load Balancer:-**

* It is operated at Layer 4 of the OSI model. It routes the traffic between clients and backend servers based on IP address.

**4) Gateway Load Balancer:-**

* Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE (Generic Network Virtualization Encapsulation).
* These appliances enable you to improve security, compliance, and policy controls.

**\*Round Robin Load-Balancing:-**

* Round Robin Load-Balancing distributes traffic to a list of servers in rotation using DNS.
* It is one of the simplest methods for distributing client requests across a group of servers. Going down the list of servers in the group, the round-robin load balancer forwards a client request to each server in turn.

**\*Listener:-**

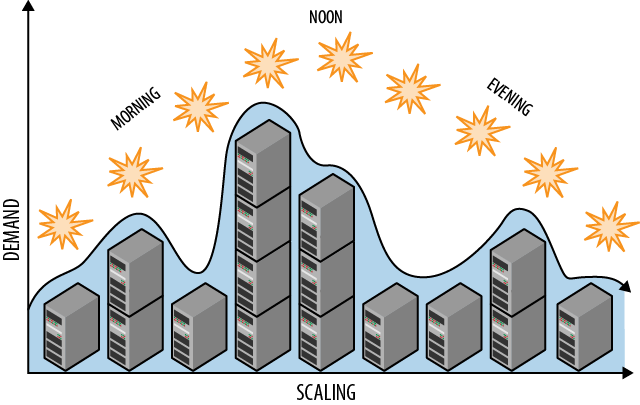
* It is a point/zone in your LB where Incoming traffic will enter.
* A listener is a process that checks for connection requests, using the protocol and port that you configured.
* Your load balancer will automatically perform health checks on your EC2. Instances and only route traffic to instances that pass the health check.
* If an instance fails the health check, it is automatically removed from the load balancer.
* Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only.
* You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

**12) Target Group:-**

* A Target Group tells load balancer where direct to traffic to ec2 instance and fixed ip addr.

**13) Auto-Scalling:-**

* AWS provides the Auto Scaling service to help you achieve this, ensuring that your application can dynamically scale in or out to handle varying workloads.
* It is easy to setup application scalling for multiple resources across multiple service in minute.



**\*Benefits of Auto-scalling:-**

* Improved Fault Tolerance.
* Improved Availability.
* Improved Cost Management.

**\*Dynamic Auto-scalling:-**

* It is the functionality that automatically increases or decreases capacity based on load or other metrics.
* Condition is required.

**\*Fleet Management Scalling:-**

* Runs with fixed no of instances.
* Condition is not required.

**\*Auto-Scalling-Group(ASG):-**

* It used to declare Min, Max & Desired capacity & condition based on metrics to decide number of instances.
* Metrics used in scalling policy: Target Tracking Scalling Policy
* Metrics to use: 1) Avg CPU Utilization.

2) Network in

3) Network out

4) No of hits target instances of load balancer.

* Scalling is the process adding/removing capacity/resources as needed.

**\*Scale out is adding the capacity /resources.**

**\*Scale in is removing the capacity /resources.**

**\*Formula:-**

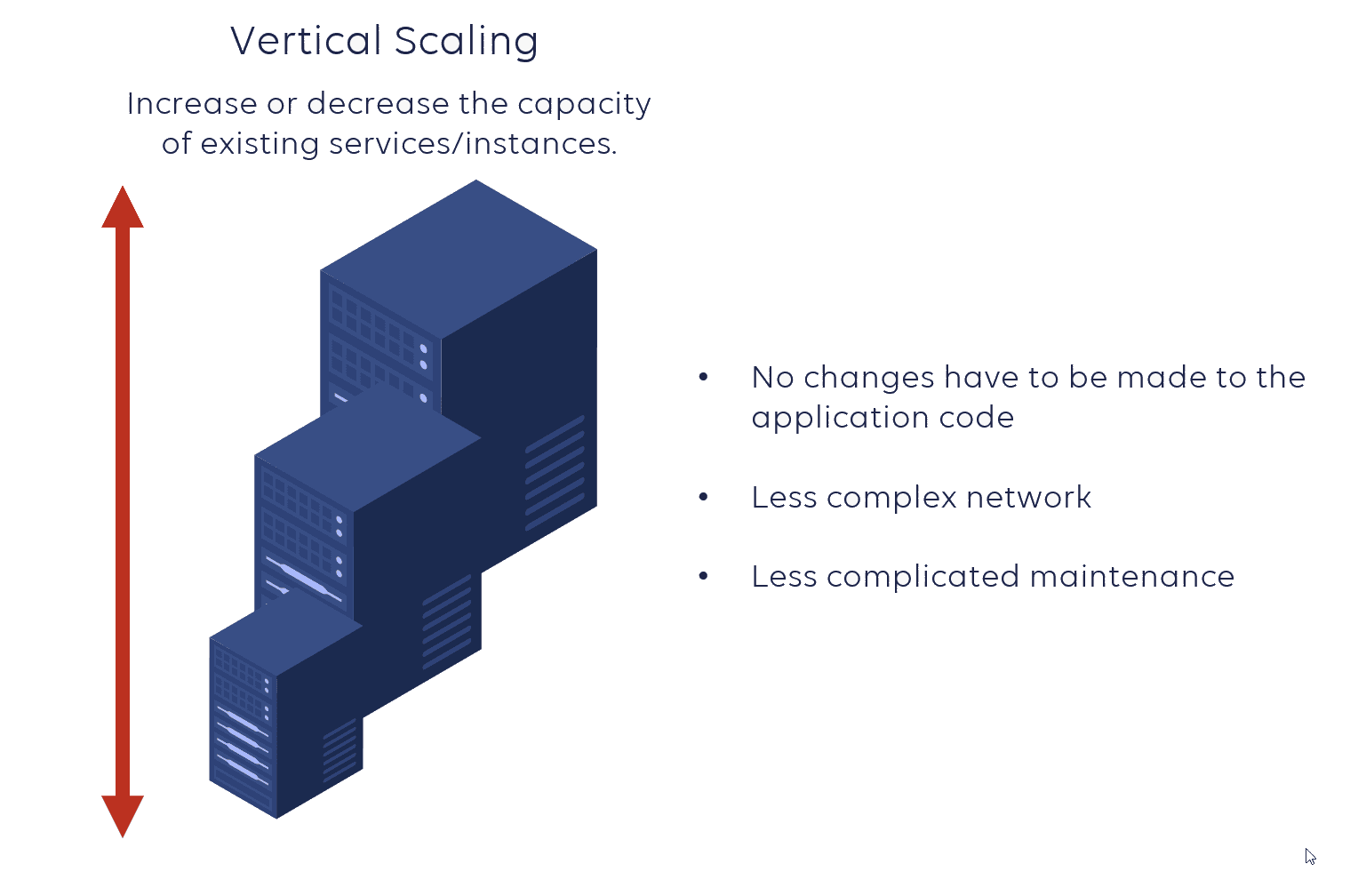
**No of instances will join the grp= Current utilization of CPU =99.0/Expected Avg. CPU Util >35%**

**\* If Avg. CPU utilization of an instance > 35% then additional instance will join the ASG.**

**\*Types of Scalling:-**

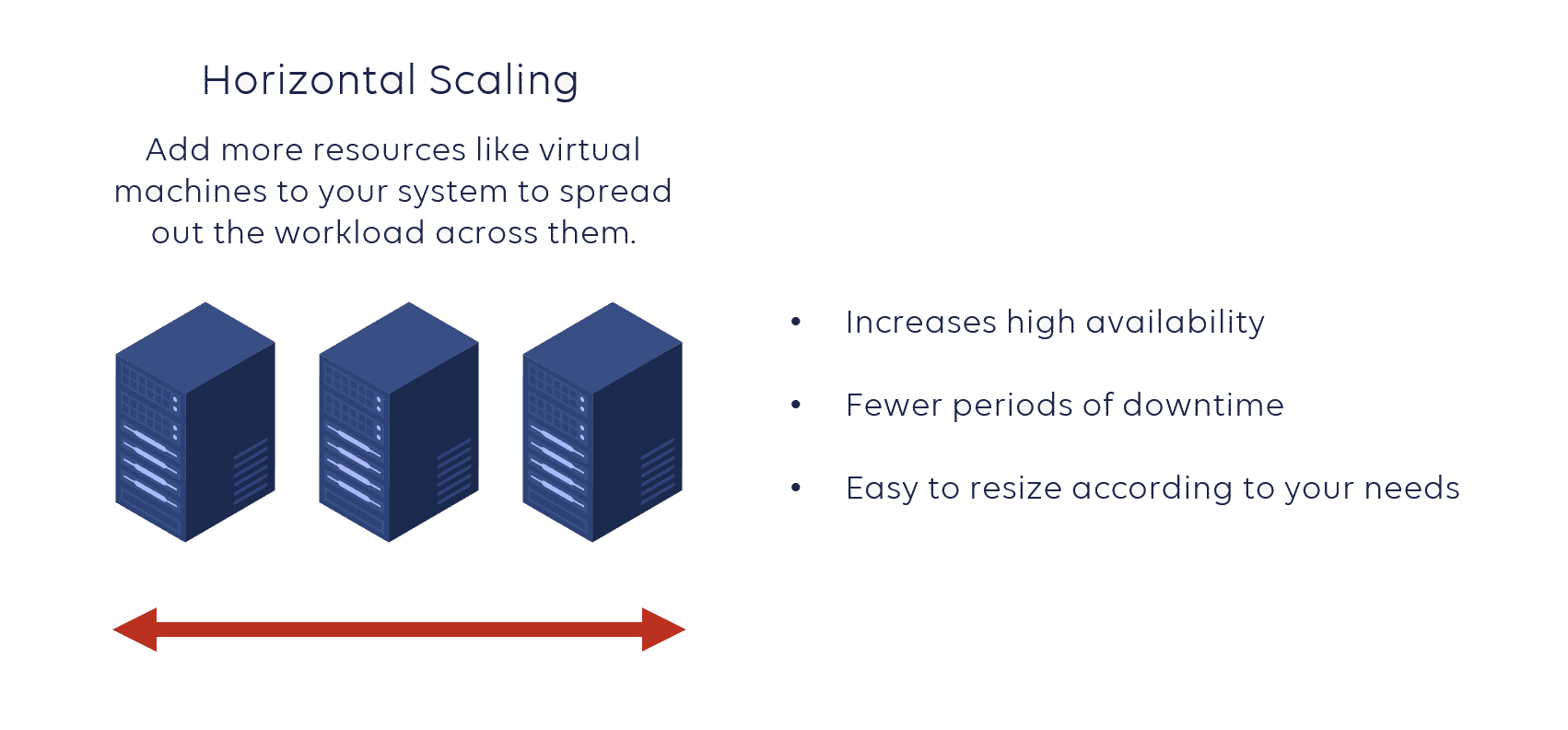
**1) Vertical Scalling:-**

* Vertical scaling, or scaling up, involves increasing the resources of a single instance to handle a higher load.
* This typically involves upgrading the existing hardware or moving to a more powerful machine. ex: CPU,RAM,DATABASE.



**2) Horizontal Scalling:-**

* Horizontal scaling, also known as scaling out, involves adding more instances or nodes to a system to distribute the load.
* Each instance operates independently, and the workload is balanced across multiple instances.



**14) Launch Configuration:-**

* Launch configuration is an instance configuration template that an autoscalling group uses to launch ec2 instance.
* Auto scalling uses a launch configuration to launch instance AMI ID, Instance Type, Key pair, Security groups, & EBS volume configuration.

**15) VPC (Virtual Private Cloud):-**

* It is a virtual private cloud. VPC is a virtual network inside in aws for our client. We can create max 5 vpc and 200 subnet in one region.
* Its process of creating private, isolated cloud environments with custom network settings, enabling secure deployment and management of AWS resources.
* Vpc provide customised and additional security as per customer requirement.

**a) Types of VPC:-**

**1) Default VPC:-**Preconfigured by AWS in all Regions to work on it.

**2) Custom VPC:-**Created By User.

**b) Subnet:-**

* Subnet is a range of i/p address in your pc. Subnet is specific for availability zone.
* Subnets are like breaking a large network into sub-networks.
* After creating a VPC, you can add one or more subnets in each Availability Zone.
* Subnet's CIDR Address must be within CIDR Address of VPC.

**10.0.0.0 To 10.255.255.255 ( Any IP starting from 10 is a private IP )**

* Subnets can’t be nested. One Subnet can’t be created within another Subnet.
* A Subnet can never be bigger than VPC & maximum it can be as big as the VPC, then in that case no other Subnet's can be created.
* The number of Subnet's which can be created in VPC will be decided by the number of available IPs in the VPC.
* Subnet's CIDR Address must be within CIDR Address of VPC.

**\*IMP NOTES:-**

A VPC is a regional service i.e. it will remain in a region. a single VPC can’t span regions.

1) A Subnet will be locked in an AZ. one Subnet can’t span multiple AZs.

2) An Internet gateway can only be attached to one VPC at a time

3) A VPC can only have one Internet gateway. There is strict one-to-one relationship i.e. one VPC <--> One IGW.

**\*Ipv4 private IP addresses:-**

10.0.0.0 to 10.255.255.255

172.16.0.0 to 172.31.255.255

192.168.0.0 to 192.168.255.255

**EXAMPLE:-**

192.168.0.0

192.168.0.1

192.168.0.2

.

‘

‘

192.168.0.255

‘192.168.1.0

192.168.1.1

192.168.1.2

.

192.168.1.255

SOO ON…………….

Let’s take

192.168.0.0/16 (prefix)

The main CIDR for VPC its huge network it have IP 65536

(to calculate the no of IP can user formula

Number of IP addresses=2^32−CIDR notation

**For example, if you have a CIDR notation of "/16":**

Number of IP addresses=232−16=216=65536

Number of IP addresses=2

32−16

=2^16 = 65536 (ip)

192.168.0.0/16 (prefix)

We need to make 2 subnet from above VPC IP range

Network 1: 192.168.1.0/24

Here are 256 IP addresses ranging from 192.168.1.0 to 192.168.1.255.

Network 2: 192.168.2.0/24

256 IP addresses ranging from 192.168.2.0 to 192.168.2.255**.**

**c) Route Tables:-**

* It is set of rules which determine the n/w traffic from your subnet is directed.
* The route table contains existing routes to CIDR blocks outside of the ranges in your VPC.
* When we work on vpc level we used route tables.

**d) Internet Gateway (IG):-**

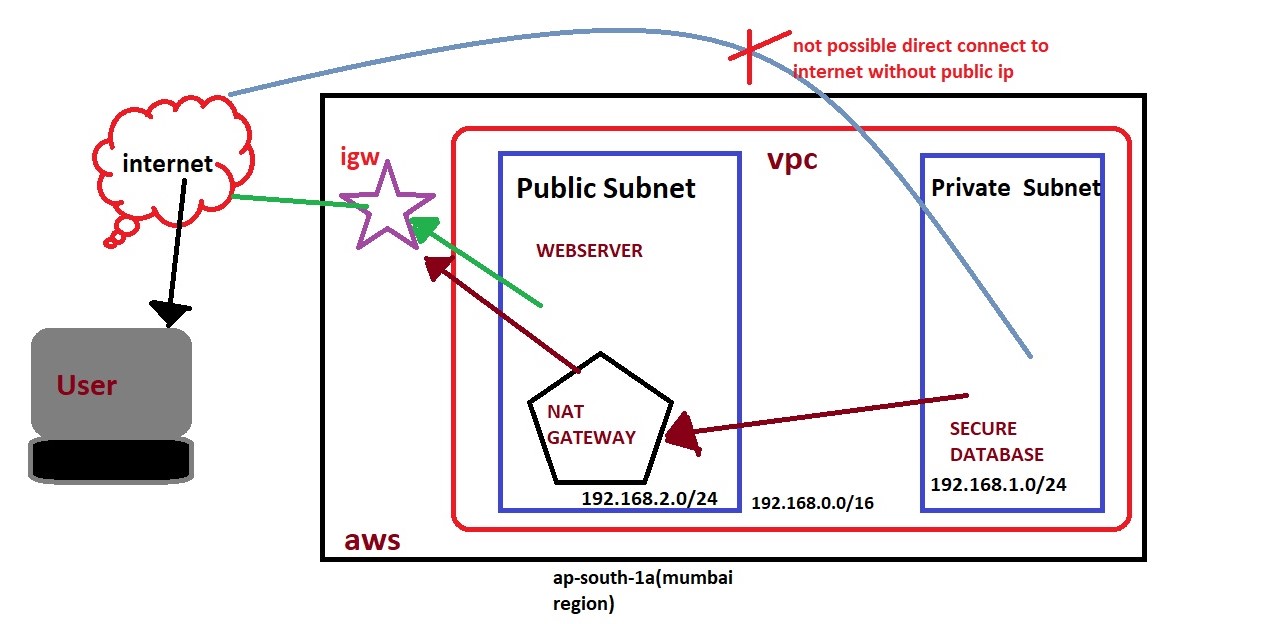
* VPC to connect to the internet. Enabling communication between VPC resources and external networks.
* It convert info, data or other communication from one protocol to another.
* A computer that sits between different networks or applications.
* It make the subnet public and responsible to connect to the internet.

**e) Engress (Only-Internet Gateway):-**It’s highly available vpc component which allows outbound in your pc.

**f) Ingress (Only-Internet Gateway):-** From internet to resource communication is called ingress (Only-Internet Gateway).

**g) NAT Instance/Gateway:-**

* NAT instance is an EC2 instance that lives in your public subnet.
* NAT gateways enable devices in a private network to access the internet using a shared public IP (elastic ip) address.
* They perform IP address translation, converting private IP addresses to a single public IP for outbound traffic.
* Typically doesn't allow direct inbound traffic from the internet to internal devices without additional configurations.
* Provides a level of security by hiding internal IP addresses from external networks.
* Widely used in cloud environments to enable private instances to access the internet.



**\*NAT gateway's pricing will depend**

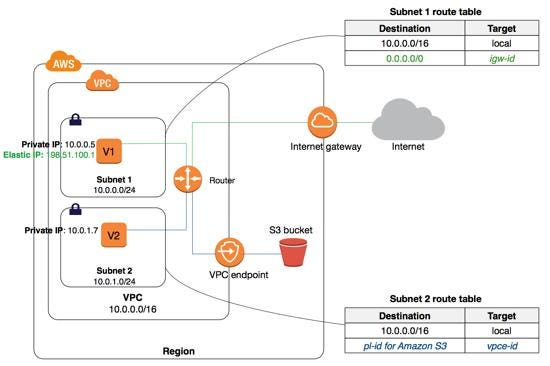
1. Duration or no. of hours for which it will run

2. Amount of data transferred thru NAT gateway

**h) NACL:-** It is network access control list. NACL is used for who as access your bucket and object.

**i) VPC Endpoint:-**

* It provides a way to connect your VPC with other AWS services without requiring internet traffic.
* Instead of routing traffic over the public internet, a VPC Endpoint allows for a more direct and secure connection between your VPC and the AWS service like S3 and dynamo-db.
* A VPC endpoint enables customers to privately connect to supported AWS services.
* Amazon VPC instances do not require public IP addresses to communicate with resources of the service.
* Traffic between an Amazon VPC and a service does not leave the Amazon network.



**j) VPC Peering:-**

* VPC peering is like creating a direct, secure tunnel between these two spaces.
* It 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.

1. Two VPCs which are to be peered must have non-overlapping CIDR block. A peering connection cannot be created between 2 VPCs that have overlapping CIDRs. Please select 2 VPCs which have distinct CIDRs.

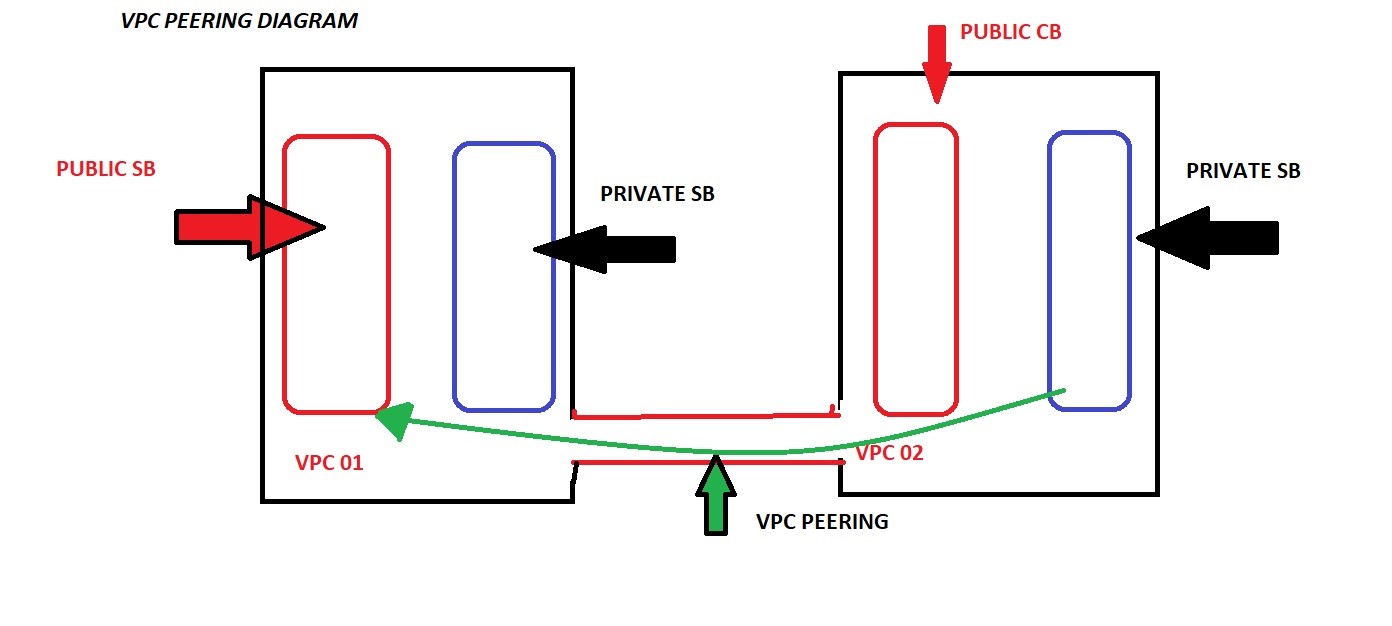
2. VPC Peering supports Global peering (both VPC in diff region) and Regional peering ( both VPCs in same region)

3. VPC peering supports intra-account (both VPCs in same AWS account) peering and also inter-account (both VPCs in diff AWS account) peering.

4. There can be only one peering connection between any two VPCs

5. VPC Peering does not supports Transitive Peering.

\*For peering need to add route in routes tables.



**16) Security Group:-** It is used for control the inbound and outbound traffic. When we work on resource level we use security group.

**A) Inbound Rule:-** control the incoming traffic to your instance.

**B) Outbound Rule:-** control the outgoing traffic from your instance.

**\*Difference between NACL and Security Group:-**

|  |  |
| --- | --- |
| **Security Group** | **NACL** |
| 1) Security group attach to the ec2 instance. | 1) NACL attach to the VPC level. |
| 2) It can add allow rule only. | 2) It can add both rule allow and deny. |
| 3) Can assign unlimited number of security groups | 3) Can assign up to 5 security groups |
| 4) State-full filtering | 4) Stateless filtering |

**17) IAM (Identity Access Management):-** It is a web service that helps you securely control access to AWS resource.

**18) User:-** When we have to access a service we used user.

**\*Two types of user:-** 1) AWS Management Console Access Users 🡪 UI (username & password)

2) Programmatic Access Users 🡪 Username Access key & secrete access key

Recommended for AWS CLI , APIs , SDKs , & other dev. Tools.

**19) Role:-** When we have to access a service to service we used role.

**20) User-group:-** An IAM user group is **a collection of IAM users**. User groups specify permissions for multiple users.

**21) Policy:-** It is used for giving access and permission.

**22) Identity Provider:-** It is used to manage access to multiple account provide user single sigh-in.

**23) S3 (Simple Storage Service):-** S3 is used to store and retrieve any amount of data at any time, from anywhere.

**\*S3 Data Consistency Model:-** 1)Data uploaded in a particular region never leaves it.

2) Read after write consistency.

**\*Storage Hierachy:-** 1) S3 follows a storage hierarchy while keeping data.(doc, image, videos, file)

2) Management console or S3 API can be used to manage buckets & objects.

3) By default, the max no of buckets that can be created per acc in 100.

4) Bucket is region specific , but globally accessible.

5) For additional buckets , one can submit service limit increase.

**\*Objects in S3:-** 1) Object are videos , images ,documents etc. which are stored inside bucket.

2) While creating a bucket a name is given and the name is the object key.

**\*Object Metadata:-** 1)For each object , S3 maintain a set of system metadata.

2) **Date:-** a)Current date & time.

3) **Content-Length:-** Object size in bytes.

4) **Last-Modified:-** Object creation & last modified date.

5) **x-amz-version-id:-** Object version.

**\*Storage Class:-** Each object has storage class associated with it.

**\*Replication in S3:-** S3 Replication gives you the ability to replicate data from one source bucket to multiple destination buckets in the same, or different, AWS Regions.

**\*Prerequisite for Replication:-** 1)Both buckets must be different regions.

2)Versioning on the both buckets must be allowed.

**\*why use Replication?**

1) DR

2) During the transit you can store data from one storage class to another.

**\*Versioning:-** 1) To maintain multiple variants on an object is called versioning.

2) Versioning is disabled by default on buckets.

3) Once you enable it , can’t be disabled , It can only be suspended.

4) If the object is deleted accidently , can be restored because of versioning.

**\*Difference between Folder & Objects**

|  |  |
| --- | --- |
| **Folder** | **Objects** |
| Folder can be created , deleted , & made public , but they can’t renamed. | Objects can be copied from one folder to another. |

**24) Two types of Block store devices are available for EC2:-**

**1) EBS (Elastic Block Storage):-**

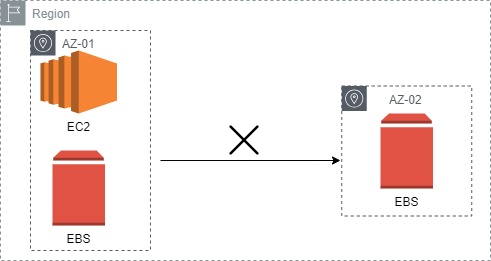
* It is a easy way to use scalable , high performance , block storage service designed by aws. Amazon EBS Provides persistent block level storage volumes for use with Amazon EC2 instances.
* Persistent volume.
* Network attached virtual drive.

**2) Instance Storage:-**

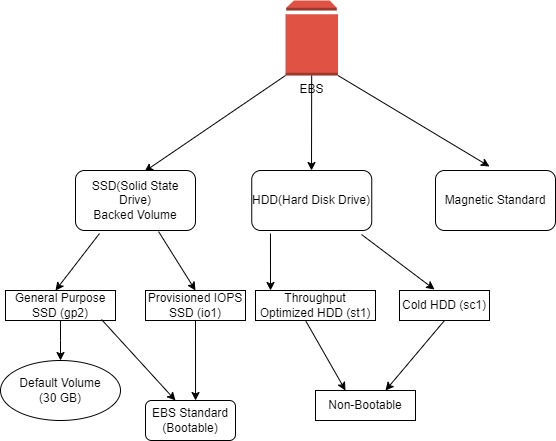
* Instance store Backend(volume store in instance) EC2 basically the virtual hard drive on the host allocated to this EC2 instance.
* Limited to 10 GB per device.
* Non-persistent volume.
* The ec2 instance can’t be stopped, can only be rebooted or terminated then terminate will delete data.
* When we click on Delete-On-Termination & tick it. It will delete the storage volume.

**3) EBS volume:-**

* EBS volumes behave like raw, unformatted, external block storage device that you can attach to your ec2 instance.
* EBS volumes are block storage devices suitable for database style data that requires frequent reads & writes.
* EBS volumes are attached to your ec2 instance through AWS network, like virtual hard driver.
* An EBS volume can attach to a single ec2 instance only at a time.
* Each EBS volume & EC2 instance must be in same availability zone.
* An EBS volume data is replicated by AWS across multiple services in the same AZ to prevent data loss resulting from any single AWS component failure.



**4) EBS volume types:-**



**1) General Purpose SSD (gp2):-**

* GP2 is the default EBS volume type for the amazon EC2 instance.
* GP2 volume are backed by SSDs.
* GP2 balances both price & performance. 50 GB 🡪 50 IOPS (Input Output per sec).
* Ratio of 3 IOPS /GB with upto 10000 IOPS.
* Boot volume having low latency. (boot volume🡪you can install OS)
* Volume size – 1 GB 🡪 16 TB.
* Price - $0 10 GB/month.

**2) Provisioned IOPS SDD (io1):-**

* These volumes are ideal for both IOPS intensive & throughput intensive workloads that require extremely low latency or for mission critical application.
* Designed for i/o intensive applications such as large relational or NOSQL databases.
* Use if you need more than 10,000 IOPS.
* Can provision upto 32,000 IOPS per volume. Special case we use 64,000 IOPS.
* Volume size – 4 GB 🡪 16 TB
* Price - $0 125/GB/month

**3) Throughput Optimized HDD (st1):-**

* ST1 is backed by hard disk drives & is ideal for frequently accessed, throughput intensive workloads with large datasets.
* ST1 volumes deliver performance in term of throughput, measured in MB/s
* Big data, Data warehouse, log processing.
* It can’t be a boot volume.
* Can provisioned upto 500 IOPS per volume.
* Volume size – 500 GB 🡪16 TB.
* Price - $0 045 GB/month.

**4) Cold HDD (sc1):-**

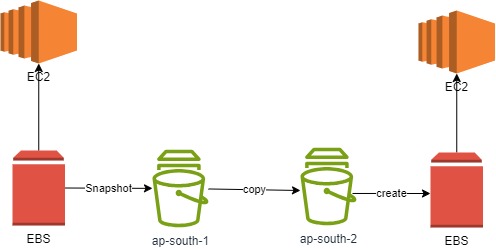
* SC1 is also backed by HDD & provides the lowest cost per GB of all EBS volume types.
* Lowest cost storage for infrequent access workloads.
* Used in file server.
* Can’t be a boot volume.
* Can provisioned upto 250 IOPS per volume.
* Volume size – 500 GB 🡪 16 TB.
* Price - $0 025 GB/month.

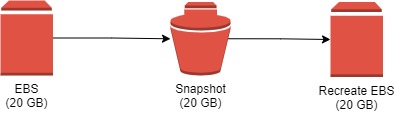
**5) Magnetic Standard:-**

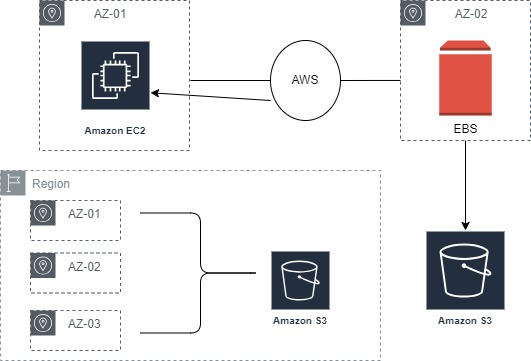
* Lowest cost per GB of all EBS volume type that is bootable.
* Magnetic volumes are ideal for workloads where data is accessed infrequently & application where the lowest storage cost is important.
* Volume size – 1 GB🡪1 TB.
* Max IOPS/volume – 40 – 200.
* Price - $0 05 per GB/month.

**\*EBS Snapshot:-**

* EBS snapshot are point in time images/copies of your EBS volume.
* Snapshots are used to backup data on EBS Volume.
* Any data written to the volume after the snapshot process is initiated, will not be included in the resulting snapshot (But will be included in future, incremental update.)
* Per AWS account, upto 5,000 EBS volume can be created.
* Per account, upto 10,000 EBS snapshots created.
* EBS snapshots are copies/stored on S3, however you cannot access them directly you can only access them through EC2 API’s.
* While EBS volumes are AZ specific, Snapshots are region specific.
* Any AZ in region can use snapshots to create EBS volume.
* To migrate an EBS from one AZ to another, create a snapshot (region specific) & create an EBS volume from the snapshot in the intended AZ.
* You can create a snapshot to an EBS volume of the same or larger size than the original volume size, from which the snapshot was initially created.







**\*Non-boot/root Snapshot:-**

* You can take a snapshot of non-root/boot EBS volume while the volume is in use on a running ec2 instance.
* This you can still access it while the snapshot is being processed.
* However the snapshot will only include data that is already written to your volume.
* The snapshot is created immediately but it may stay in pending status until full snapshot is complete, specially for the first time snapshot of volume.
* During the period , when the snapshot status is pending, you can still access the volume (non-root/boot), but I/O might be slower because of the snapshot acivity.
* While in pending state, on in-progress snapshot will not include data from ongoing reads & writes the volume.
* To take complete snapshot of your Non-root/boot EBS volume🡪 Stop or Unmount volume.
* To create snapshot for root EBS volume, you must stop the instance first then take the snapshot.

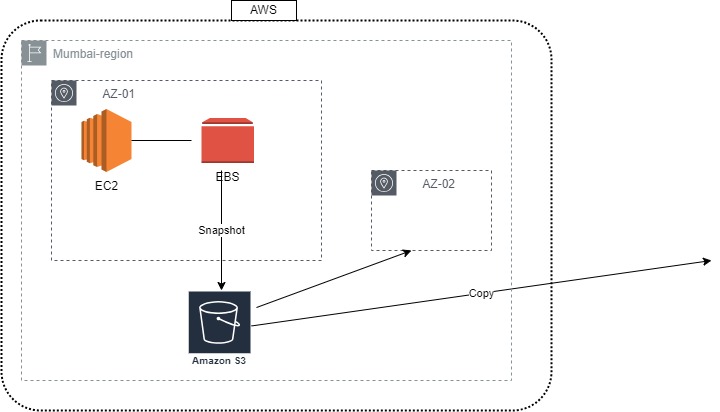
**\*Incremental Snapshot:-**

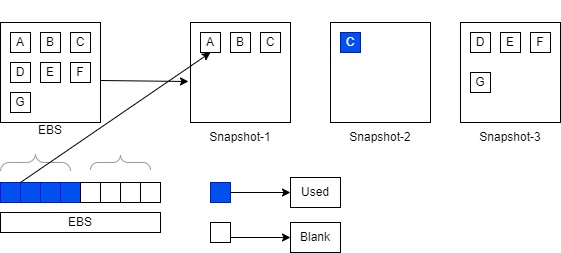
* EBS snapshot are stored incrementally.
* For low cost storage on S3 & guarantee to be able to restore data from the snapshots.
* What you need is a single snapshot, then further snapshots will only carry the changed blocks (Incremental updates).
* Therefore you don’t need to have multiple full/complete copies of the snapshot.
* You are changed for 🡪

Data transferred to S3 from your EBS volume you are taking snapshot

Snapshot stored in S3.

* Deleting snapshot will only remove data exclusive to that snapshot.





**\*EBS Encryption:-**

* EBS encryption is supported on all EBS volume types & all EC2 instance families.
* Snapshot of encrypted volume are also encrypted.
* Creating an EBS volume from an encrypted snapshot will result it an encrypted volume.
* Data encryption at rest means encrypting data while it is stored on the data storage device.
* There are many ways you can encrypt data on an EBS volume at Rest, while the volume is attached to an EC2 instance.

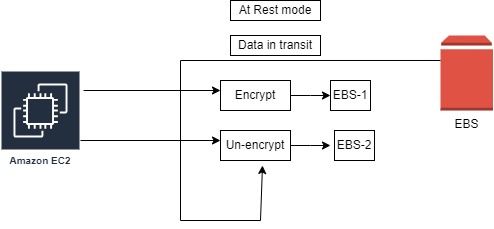
Use 3rd party EBS volume.

Encryption tools.

Use encrypted EBS volumes.

Use encryption at the OS level.

* Encrypt data at the application level before storing it to the volume.
* Use encrypted file system on the top of the EBS volume.
* Encrypted volume are accessed exactly like unencrypted ones, basically encryption is handled transparenty.
* You can attach an encrypted & unencrypted volumes the same EC2 instance.
* Remember that the EBS volumes are not physically attached to the ec2 instance, rather they are virtually attached through the EBS infrastructure.
* This means when you encrypt data on an EBS volume, data is actually encrypted on the ec2 instance then transferred, encrypted to be stored on the on the EBS volume.
* This means data in transit between EC2 & encrypted EBS volume is also encrypted.
* There is no direct way to change the encryption state of the volume.



**\*To change the state (indirectly) you need to follow either of the following two ways:-**

1) Attach a new, encrypted, EBS volume to the ec2 instance that has the data to be encrypted then-

* Mount the new volume to the ec2 instance.
* Copy the data from the un-encrypted volume to the new volume.
* Both volumes must be on the same ec2 instance.

2) Second method

* Create a snapshot of the un-encrypted volume.
* Copy the snapshot & choose encryption for the new copy, this will create an encrypted copy of the snapshot.
* Use this new copy to create an EBS volume, which will be encrypted too.
* Attach the new, encrypted EBS volume to the ec2 instance.

**\*Root EBS volume Encryption:-**

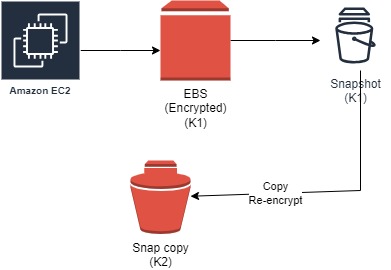
* There is no direct way to change the encryption state of a volume.
* There is an indirect workaround to this-
* Launch the instance with the EBS volume required.
* Do whatever install applications.
* Create an AMI from the ec2 instance.
* Copy the AMI & choose encryption while copying.
* This result it an encrypted AMI that is private. (yours only)
* Use the encrypted AMI to launch new ec2 instance which will have these EBS root volume encrypted.

**\*EBS Encryption Key:-**

* To encrypt a volume or a snapshot you need an encryption key, these keys are called Customer Master Key (CMK) & are managed by AWS Key Management Service (KMS).
* When the encrypting the first EBS volume, AWS KMS creates CMK key.
* This key is used for your first volume encryption, Encryption of snapshots created from this volume & subsequent volumes created from these snapshots.
* After that each newly encrypted volume is encrypted with a unique/separate AES-256 bit encryption key. This key is used to encrypt the volume it snapshots & any volumes created of it snapshots.

**\*Changing Encryption Key:-**

* You cannot change the encryption (CMK) key used to encrypt on existing encrypted snapshot or encrypted EBS volume.
* If you want to change the key, create copy of the snapshot & specify during the copy process, that you want to re-encrypt the copy with a different key.
* This comes in handy when you have to change the key in order to be able to share the snapshot with other account.

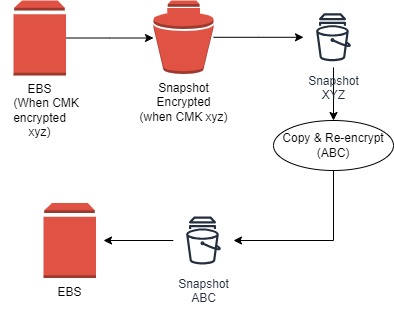


**\*Sharing EBS snapshot:-**

* By default, only the account owner can create volumes from the account snapshots.
* You can share your un-encrypted snapshots with the AWS community by making them public.
* Also you can share your un-encrypted snapshot with a selected AWS account by making them private then selecting the AWS accounts to share with.
* You can’t make your encrypted snapshot public.
* You can’t make a snapshot of an encrypted EBS volume on AWS.

**\*You can share your encrypted snapshot with specific AWS account as follows:-**

* Make sure that you use a non-default/custom CMK key to encrypt the snapshot, not the default CMK key (AWS will not allow the sharing if default CMK is used).
* Configure cross account permission in order to give accounts with which you want to share the snapshot, access to the custom CMK key used to encrypt the snapshot.
* Without this, the other accounts will not be able to copy the snapshot, not will be able to create volumes of the snapshot.
* AWS will not allow you to share snapshots encrypted using your default CMK key.



**\*For the AWS accounts with whom an encrypted snapshot is stored-**

* They must first create their own copies of the snapshot.
* Then they use that copy to restore/create EBS volume.
* You can only make a copy of the snapshot when it has been fully saved to S3 (Its status show as complete) & not during the snapshot’s pending status (when data blocks are being moved to S3).
* Amazon S3 Server Side Encryption (SSE) protect the snapshot data in transit while copying.
* You can have upto 5 snapshot copy request running in a single destination per account.

**Q. Can I create an encrypted data volume at the time of instance launch?**

**🡪** Yes, using customer master keys (CMKs) that are either AWS-managed or customer-managed. You can specify the volume details

**Q. Can I create additional encrypted data volumes at the time of instance launch that are not part of the AMI?**

**🡪** Yes, you can create encrypted data volume with either default or custom CMK encryption at the time of instances launch.

**Q. Can I launch an encrypted EBS instance from an unencrypted AMI?**

**🡪** Yes.

**Q. Can I share encrypted snapshots and AMIs with other accounts?**

**🡪** Yes. You can share encrypted snapshots and AMIs using a customer-managed customer master key (CMK) with other AWS accounts.

**Q. Can I ensure that all new volumes created are always encrypted?**

**🡪** Yes, you can enable EBS encryption by default with a single setting per region. This ensures that all new volumes are always encrypted.

**Q. Can I boot an EC2 instance using a Multi-Attach enabled volume?**

🡪 No

**25) EFS:-** EFS(Elastic File System) is Managed File Storage for EC2 & used to share common storage for multiple instances which are only in same region.

**1) You will pay only for used. 2) It can grow upto petabyte.**

**3) Easy to manage. 4) It is highly scalable.**

**26) Fsx:-** Amazon FSx is built on the latest AWS compute, networking, and disk technologies to provide high performance and lower TCO.

**27) AWS Cloud-watch:-**

* Cloud-watch is a monitoring tool for AWS resources and application.
* Send notification if anything goes wrong.

**\*Two Type Of monitoring AWS cloud-watch:-**

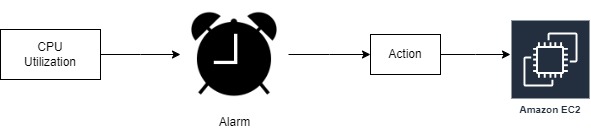
* **Basic --** is free of cost -- will fetch data points every 5 min duration
* **Detailed --** has additional cost -- will fetch data points every 1 min duration

**\*Two fundamental component of monitoring:-**

* **Metrics:-** Cloud-Watch collects and stores data in the form of metrics. These metrics represent different aspects of your AWS resources, such as CPU usage, network activity, and more.
* **Logs:-** Cloud-Watch Logs allow you to collect, monitor, and store log files from your applications, operating systems, and AWS resources. This is particularly useful for troubleshooting and auditing

**\*Cloud-watch Alarm:-**

* Alarms watch over metrics & metrics only.
* Alarms can be set to take action based on metrics data.
* Cloud-Watch Alarms can be set to notify you when certain thresholds are hit so it protect the system from major accident.

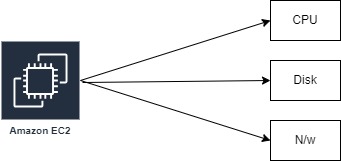


**\*Three stages of Alarm:-**

* **Insufficient data** -- Data gathering in progress or metric not available/missing data.
* **OK** -- The current value is well within the specified limit or within threshold.
* **ALARM** -- Crossed threshold.

**\*Metrics & Namespaces:-**

* Metrics are fundaments to cloud-watch monitoring.
* Individual data points which are monitored , all actions are based on metrics e.g. CPU utilization procedure.
* All AWS services send metrics to cloud-watch by defaults.



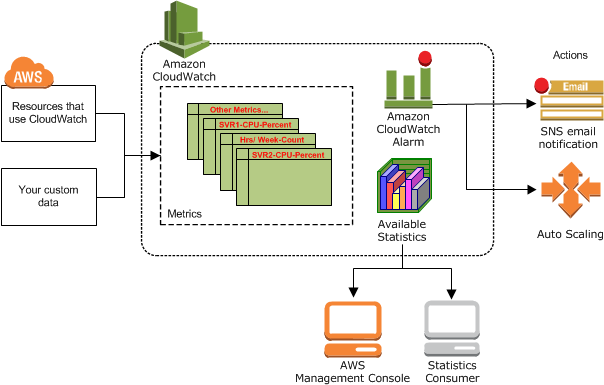
**\*2/2 check-up after launch instance:-**

**1. System Status Checks:-**

* It check infrastructure part that support to ec2 instance it including hardware, networking and other infrastructure component.

**2. Instance Status Checks:-**

* Focus on the health of the individual EC2 instances. These checks look at the software and configuration of the instance itself.



**28) Cloud-trail:-** Cloud-trail is a web service that records API activity in your AWS acc.

**29) Route53:-** It is a DNS web service in AWS & it is server-less. It connect user request to internet application on AWS. You can manage multiple hosted zone for every hosted zone you need to pay $0.5 per domain per month per hosted zone.

1) Register domain names.

2) To route internet traffic to the resources for domain.

3) Check the health of the resources.

4) Hosted Zone Configuration.

**\*Hosted Zone:-** It is a set of records for a particular domain name.

**\*DNS is containing Record of Resources.**

**\*Number of Resources Records (RR):-**

**1) A🡪**Hostname/Domain🡪 IPv4 address of the websites.

**2) AAAA🡪**Hostname/Domain🡪 IPv4 address of the websites

**3) CNAME🡪**Canonical name (Alias Name)🡪Another name of the domain.

**4) NS🡪**Name Server🡪It is connected with domain name provider name server.

**5) SOA🡪**Start of Authority🡪Providing some important parameter such as admin email id,TTL,DNS

ver name etc.

**6) MX🡪**Mail Exchange Record🡪It is for mail server , It routes your traffic to the mail server in your infrastructure

.

**30) Secret key and access key:-** Access keys consist of an access key ID and secret access key, which are used to sign programmatic requests that you make to AWS.

**31) AMI:-** AMI stands for Amazon Machine Image. AMI is a template that contains the software configurations, launch permission and a block device mapping that specifies the volume to attach to the instance when it is launched.

**32) AWS CLI:-** AWS CLI is a unified tool to manage your AWS services.

**33) Difference between private and public subnet:-** The instances in the public subnet can send outbound traffic directly to the internet, whereas the instances in the private subnet can't. private subnet can access the internet by using a network address translation (NAT) gateway that resides in the public subnet.

**34) S3 Glacier:- Amazon S3 Glacier (S3 Glacier) is a secure and durable service for low-cost data archiving and long-term backup.**

**35) Snowball:-** Snowball is use to create jobs, track data, and track the status of your jobs through to completion.

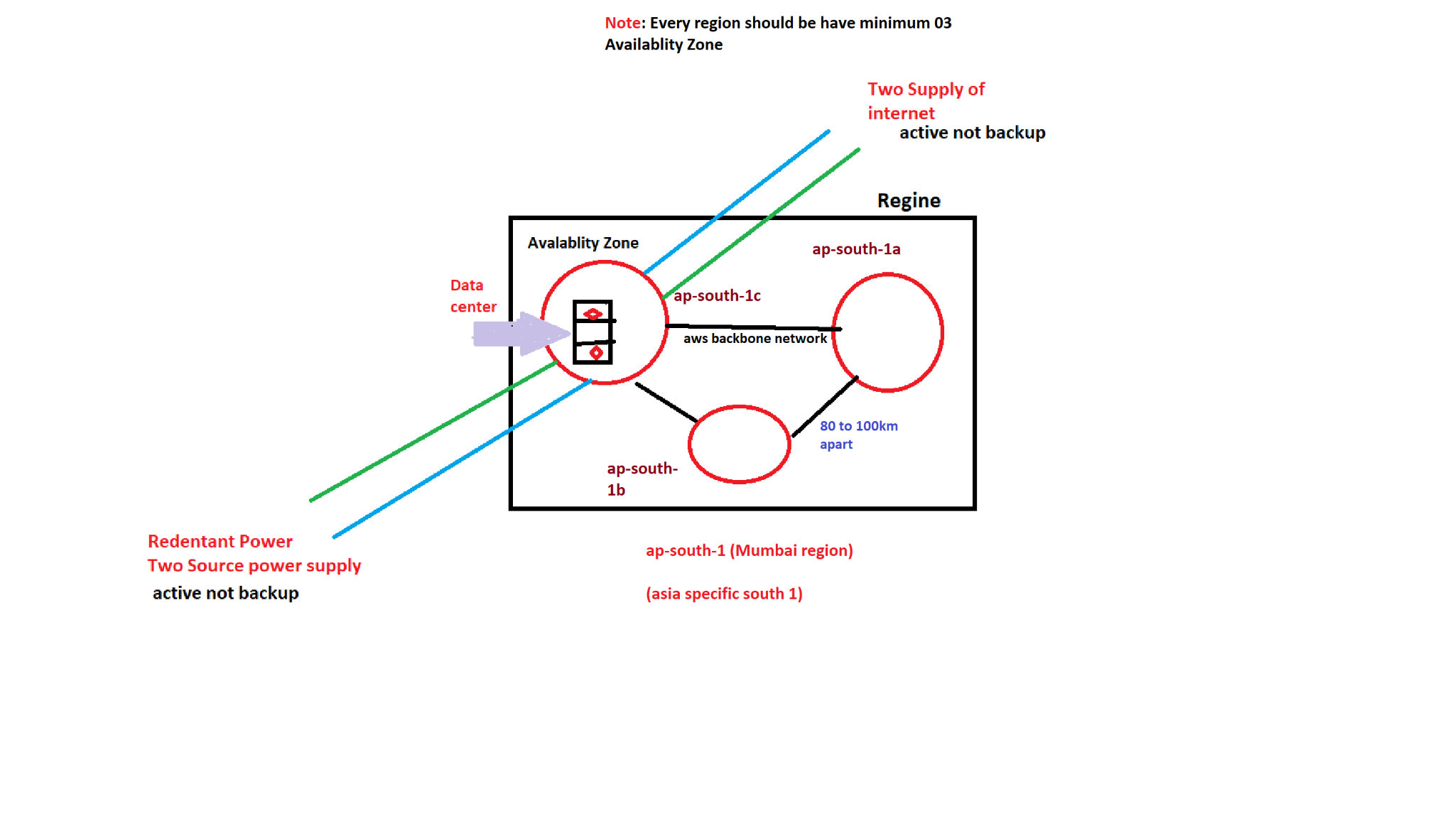
**36) Static or Elastic IP:-** An Elastic IP address is **a static IP address associated with your AWS account in a specific Region.**

**37) AWS Region:-**

* AWS Region is **a separate geographic area.**
* Every region should be have minimum 3 availability zone.

**38) AWS Availability Zone:-**

* In AWS Region has multiple, isolated locations known as Availability Zones.



**39) Types of storage in S3:-** i) S3 Standard

ii) S3 Intelligent-Tiering

iii) S3 Standard-IA

iv) S3 One Zone-IA

**40) Trusted Advisor:-** AWS Trusted Advisor is an online resource to help you reduce cost, increase performance, and improve security by optimizing your AWS environment. AWS Trusted Advisor provides real-time guidance to help you provision your resources.

**41) API-(Application Programming Interface):-** An API is a list of specification that describe how information is exchanged between programs.

Software that wants to access another will call the API published by the other program.

Java , NET and Python 🡪 Developer

**42) RDS (Relational Database System):-** RDS is a collection of managed services that makes it simple to set up, operate, and scale databases in the cloud.

**\*Feature:-**

1) Cost efficient & resizable.

2) No shell access.

3) Flexibility feature.

4) Server configuration is easy to manage.

5) All popular DB engines supports in AWS RDS.

**\*Database:-**

**43) No-SQL Database:-** No-SQL databases are purpose build for specific data models & have flexible schemas for building modern application.

**44) Data Warehouse:-** A data warehouse is a central repo of information that can be analysed to make better informed decision.

**45) AWS Lambda:-** AWS lambda is compute service that’s lets you run code without provisioning any instance. It is also called server-less service to run code.

* With AWS lambda, you can run code for virtually any type of application or backend service –all with zero administration**.**
* AWS lambda runs your code on a high-availabilty compute infrastructure.
* AWS lambda executes your code only when needed and scales automatically from a few request per day to thousand per second.
* You pay only for the compute time & your consume – No charge when your code is not running.
* If you don’t want to manage your own compute , you can use EC2 or elastic beanstalk.

**\*How Lambda work?**

🡪1) First you upload your code to lambda in one or more lambda function.

2) AWS lambda will then execute the code in your behalf.

3) After the code is invoked , lambda automatically take care of provisioning & managing the

required service.

**\*AWS Lambda manages all the administration it manages:-**

1) Server and OS maintain.

2) High availability & Automatic scalling.

3) Monitoring fleet health.

4) Applying security patches.

5) Deploying your code.

6) Monitoring & Logging your lambda function.

7) AWS lambda runs your code on a high availability compute infrastructure.

**\*Where we use AWS Lambda?**

🡪1) File processing

2) Data & Analytics

3) Websites

4) Mobile Application.

**\*Advantages of using AWS Lambda:-**

1) No server management

2) Flexible scalling

3) No idle capacity

4) High Availability

**\*Important Points of AWS Lambda:-**

**1) Function:-** A function is a resource that you can invoke to run your code in AWS lambda. A function has code to process the events and a runtime that passes request and response to between lambda & function code.

**2) Runtime:-** Lambda runtime allows function in diff languages to run in the same base execution. The runtime seats between the lambda service and function code. In runtime , AWS lambda supports only limited languages like Node js , Python , Java , Ruby , C# , Power-shell & Go.

**3) Event source/Trigger:-** An AWS service such as Amazon SNS or custom service that triggers your function and execute its logic.

**4) Layers:-**  A Lambda layer is a .zip file archive that can contain additional code or other content. A layer can contain libraries, a custom runtime, data, or configuration files.

**5) Concurrency:-** No of request that your function is servicing in any given time.

**6) Event:-** It is JSON formatted document that contains data for a function to process.

**7) Downstream Resource:-** An AWS service Dynamo-DB table or S3 buckets that your lambda function calls once it is triggered.

**\*Some AWS services we can use for lambda trigger:-**

1) Amazon Simple Storage Service. (S3)

2) Amazon Simple Notification Service. (SNS)

3) Amazon Simple Email Service. (SES)

4) AWS Cloud Formation.

5) Amazon Cloud Watch Logs.

6) Amazon Cloud Watch Events.

7) AWS Code Commit.

**\*Difference between AWS lambda & EC2?**

|  |  |
| --- | --- |
| **AWS Lambda** | **EC2** |
| 1) AWS lambda is platform-as-a service.(Paas). | 1)AWS EC2 is an Infrastucture as a service |
| 2) It supports only limited languages like Node js,Python,Java,C#,Ruby,Go & Power shell. | 2) No Environment Restricton you can run only code or language. |
| 3) Write your code and push the code into AWS lambda. | 3) For the first time in ec2,you have to choose the OS install all the software required and then push your code in ec2. |
| 4) You cannot log into compute instance , choose customized OS or language platform. | 4)You can select variety of OS , instance types ,network & security patches, RAM & CPU etc. |

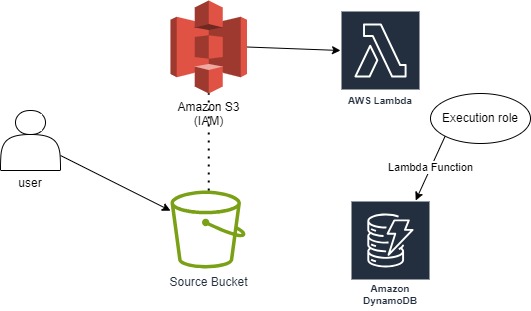
\* **When lambda triggered?**

🡪 You can use AWS lambda to run your code in response to –

* Events such as changes to data in an amazon S3 bucket or amazon dynamoDB table .
* To run your code in response to HTTP request using amazon API gateway.
* With these capabilities , you can use lambda to easily build data processing triggers for AWS services like S3 , DynamoDB , process streaming data stored in kinesis or create your own backend that operates at AWS scale performance & security.

**\*Example of S3**

* The user create an object in a bucket.
* Amazon S3 detects the object created event.
* Amazon S3 invokes your lambda function using the permission provided by the execution role.
* Amazon S3 knows which lambda function to invoke based on the event source mapping that is stored in the bucket notification configuration.



**\*AWS lambda function configuration**

* A lambda function consist of code & any associated dependencies.
* In addition , a lambda function also has configuration information associated with it.
* Initially , You specify the configuration information when you create lamda function.
* Lambda provides an API for you to update some of the configuration data.

**\*Lambda function configuration information include the following key element**

* Compute Resource that you need . You only specify the amount of memory you want to allocate from your lambda function.
* AWS lambda allocates CPU power proportional to the memory by using the same ratio as a general purpose amazon ec2 instance type , such as an M3 type.
* You can update the configuration and request additional memory in 64 MB increment from 128 MB to 3008 MB.
* Function larger than 1536 MB are allocated multiple CPU threads.

**\*Maximum execution timeout**

* You pay for the AWS resources that are used to run your lambda function.
* To prevent your lambda function from running indefinitely , you specify a timeout.
* When specified timeout is reached , AWS lambda terminates your lambda function.
* Default is 3 seconds and maximum is 900 seconds (15 minutes)

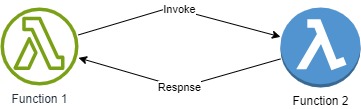
**\*AWS Lambda function can access**

* AWS services or Non-AWS services.
* AWS services running in AWS VPC (eg.Redshift , Elasticache , RDS instnce)
* Non-AWS services running on EC2 instance in an AWS VPC.
* AWS lambda run your function code securely within a VPC by default.

**\*Different way to invoke lambda function**

**1) Synchronous Invoke (Push):-**

* Synchronous Invoke are the most straight forward way to invoke your lambda function .
* In this model your function executes immediately when you perform the lambda invoke API call.
* Invocation flag specifies a value of ‘Request Response’.
* You wait for the function to process the event and return q response.

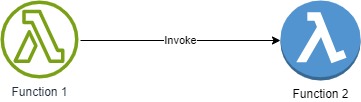


**\*List of services that inoke lambda function synchronously**

* Elastic Load Balancer
* Amazon cognito
* CloudFront
* API Gateway
* Amazon Lex
* Kinesis Data firehouse

**2) Asynchronous Invoke (Event):-**

* For synchronous invocation , lambda places the event in a queue return a success response without additional information.
* Lambda queues the event for processing and returns response immediately.
* You can configure lambda to send an invocation record to another services like SQS , SNS , Lambda & Eventbridge.
* Does not wait for function 2 to finish or response from function 2

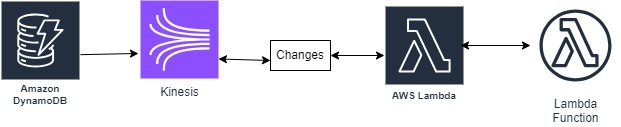


**\*List of services that inoke lambda function asynchronously**

* Amazon S3
* Amazon SNS
* SES
* Cloudformation
* Cloudwatch logs
* Cloudwatch events
* AWS codecommit
* AWS config

**3) Poll-based Invoke (Pull based):-**

* The invocation model is designed to allow you to integrate with AWS stream & Queue based service with no code or server management lambda will poll the following services on your behalf , retrieve records and invoke your function.



**The following are supported service:-**

* Amazon Kinesis
* Amazon SQS
* Amazon Dynamo-DB Streams

**46) Database Types:-**

**1) Unstructured Data:-**

* Unstructured Data is information that either does not have a pre-defined data model or is not organized in a pre-defined manner.
* Example of Unstructured Database:- 1) Weather Data

2) Employees Emails

3) Stock Market Data

4) Gaming Data

**2) Semi-Structured Data:-**

* Semi-Structured Data is information that does not reside in a relational database but that does have some organizational properties that make it easy to analyse. Eg. XML, JSON.

**3) Structured Data:-**

* Structured Data refers to information with a high degree of organization such that inclusion in a relational database is seamless & readly.
* All data which can be stored in database SQL in table with rows & columns. They have relational key & can be easily mapped into pre-defined fields.

**47) Dynamo-DB:-**

* Dynamo-DB is fully managed, multi-region, multi-master, durable-database with built in security.
* It is unstructured & semi-structured database.
* No SQL database use alternate model for data management such as key-value pairs or document storage.
* It can handle 10 trillion per day & 20 million requests per second.
* Dynamo-DB is a server-less service & you can store any amount of data in dynamo-DB tables.
* Dynamo-DB allows low latency read/write access to items ranging from 1 byte to 400 KB.
* Dynamo-DB can be used to store pointer to S3 stored object or items of sizes larger than 400 KB to if needed.

**\*Dynamo-DB stores data indexed by a primary key:-**

* You specify the primary key when you create the table.
* Each item in the table has a unique identifier or primary key, that distinguishes item from all of the others in the table.
* The primary key is the only required attributes for items in table.
* Each attribute can have its own distinct attributes.

**\*Dynamo-DB Table:-**

* A table is a collection of data items.
* Like all other DB, dynamo-DB store data in tables.

**\*Items:-**

* Each table contains multiple items.
* An Item is a group of attributes that is uniquely identifiable among all of the other items.
* An Item consist of a primary or composite key & a flexible number of attribute.
* Items in dynamo-DB are similar into rows records in other db.

**\*Attributes:-**

* Each item is composed of one or more attribute.
* An attribute consist of the attribute name & value of a set of values.
* An attribute is a fundamental data element, something does not need to be broken down any further.
* Aggregate size of an item cannot exceed 400 KB include key & all attribute.

**\*Dynamo-DB – Read Capacity Unit (RCU):-**

* One read capacity unit represents one strongly consistent read per second or two eventually consistent reads per second for an item 4 KB in size.
* If you need to read an item that is larger than 4 KB, Dynamo-DB will need to consume additional read capacity units.
* The total number of read capacity units required depends on the item size, & whether you want an eventually consistent or strongly consistent read.
* Eventually consistent (Fast & cheaper) (2 RCU=8 KB/sec).
* Strongly consistent (Little slow but expensive) (1 RCU=4 KB/sec).

**\*Dynamo-DB – Write Capacity Unit (RCU):-**

* One write capacity unit represents one write per second for an item up-to 1 KB in size.
* If you need to write an item that is larger than 1 KB, Dynamo-DB will need to consume additional write capacity units.
* The total number of write capacity units required depends on the item size.
* Write capacity unit (WCU) (1 WCU=1 KB/sec)

**\*Dynamo-DB – Pricing:-**

* Reads are cheaper than writes when using Dynamo-db.
* You pay for – Each tables provisioned read/write throughput (Hourly Rate).
* You are charged for provisioned throughput regardless whether you use it or not.
* Indexed data storage.
* Internet data transfer (if crosses region)
* Free tier per account (across all tables) of 25 read capacity units & 25 write capacity units per month.
* Dynamo-DB can do 10000 write capacity units/sec, or 10000 read capacity units per seconds per table.

**\*Dynamo-DB Limits:-**

* 256 tables per account per region.
* No limits on the size of any table.
* 25 GB free of cost for AWS DynamoDB.

**\*Read/Write Capacity Mode:-**

* **On-Demand:-** 1) It is good if you create new tables with unknown workloads.

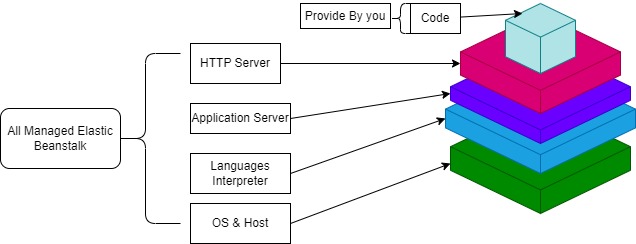
2) It is good if you have unpredictable application traffic.

3) It is good if you prefer the ease of paying for only what you use.

* **Provisioned (Default , free tier eligible):-**
  + - It is good if you have predictable traffic.
    - It is good if you run application whose traffic is consistent & ramps gradually
    - It is good if you can forecast capacity requirement to control costs.

**48) AWS Elastic Beanstalk:-**

* It is platform-as-a service (PAAS) & it is used for Application Deployment.
* It is also used for auto deployment of application.
* Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.
* Prerequisite:- You must have an application to deploy.



**49) AWS Redshift**:-

* Redshift is a fast and powerful, fully managed, petabyte-scale data warehouse service in the cloud.
* Customers can use the Redshift for just $0.25 per hour with no commitments or upfront costs and scale to a petabyte or more for $1,000 per terabyte per year.
* Amazon Redshift uses SQL to analyze structured and semi-structured data across data warehouses, operational databases, and data lakes.
* Amazon Redshift makes it easier for you to run and scale analytics without having to manage your data warehouse.

**50) Amazon Redshift Server-less:-** Amazon Redshift Server-less is a server-less option of Amazon Redshift that makes it more efficient to run and scale analytics in seconds without the need to set up and manage data warehouse infrastructure. **Max storage for Redshift Server-less is 8 PB of RMS for 512 RPUs.**

**\*Redshift configuration:-**

* **Single Node:-** A single node stores up to 160 GB.
* **Multi-Node:-** Multi-node is a node that consists of more than one node.

**\*Redshift Features:-**

* Faster performance.
* Easy to set-up, deploy & manage.
* Query your data lake.
* Secure.

**51) Amazon API:-**

* Amazon API Gateway is an AWS service for creating, publishing, maintaining, monitoring, and securing REST, HTTP, and Web Socket APIs at any scale.
* API developers can create APIs that access AWS or other web services, as well as data stored in the AWS cloud.
* API Gateway API developer, you can create APIs for use in your own client applications.
* You can make your APIs available to third-party app developers.

**\*Features of API gateway:-**

* API Security🡪Access control is a major driver for the adoption of API gateway technology.
* Rate-Limiting🡪An API gateway reduces load on back end APIs and helps prevent abuse.
* API Monitoring and Logging.
* API Transformation.
* API Scalability.

**\*Types of API Gateway:-**

* **HTTP API:-** It is a build low-latency and cost-effective REST APIs with built-in features such as OIDC and OAuth2, and native CORS support. It is work with Lambda, HTTP backend.
* **Web Socket API:-** It is a build a Web Socket API using persistent connections for real-time use cases such as chat applications or dashboards. It is work with Lambda, HTTP, AWS Services.
* **REST API:-** It is a develop a REST API where you gain complete control over the request and response along with API management capabilities. It is work with Lambda, HTTP, AWS Services.
* **REST API Private:-** It is a create a REST API that is only accessible from within a VPC. It is work with Lambda, HTTP, AWS Services.

**\*** **What is a resource?**

**🡪** A resource is a typed object that is part of your API’s domain. Each resource may have associated a data model, relationships to other resources, and can respond to different methods.You can also define resources as variables to intercept requests to multiple child resources.

**\* What is a method?**

**🡪** Each resource within a REST API can support one or more of the standard HTTP methods. You define which verbs should be supported for each resource (GET, POST, PUT, PATCH, DELETE, HEAD, OPTIONS ) and their implementation.

**52) AWS Light-Sail:-**

* Light-sail is designed to be the easy way to launch & manage a virtual private server with AWS.
* Light-sail provides developers compute, storage, and networking capacity and capabilities to deploy and manage websites and web applications in the cloud.
* Light-sail includes everything you need to launch your project quickly – virtual machines, containers, databases, CDN, load balancers, DNS management & static IP etc. – for a low, predictable monthly price.

**\*Features of AWS Light-sail:-**

* Fast & easy as compared with ec2 Instance.
* Low, Predictable price.
* Simplified Load Balancing
* Managed Databases
* Support various OS like Linux/Unix windows.
* DNS Management

**\*Difference between AWS Light-sail & EC2 instance**

|  |  |
| --- | --- |
| **AWS Light-sail** | **EC2 Instance** |
| 1) Light-sail is a cloud platform that’s cost-effective, fast, & reliable with an easy to use interface. | 1) EC2 is a compute web service that offers secure, resizable compute capacity in the cloud. |
| 2) Can use for simple web applications. | 2) Can use for simple enterprise applications. |
| 3) Websites, including custom code, Word-press, & ecommerce. | 3) HPC, Big Data & Analytics workloads. |
| 4) Single-server business software. | 4) Migration from on-premises environment. |
| 5) Dev/Test environment. | 5) Application modernization. |

**53) Virtual Private Server:-**

* A virtual private server, also known as an "instance", allows users to run websites and web applications in a highly secure and available environment, while being cost effective.

**\*Benefits of a VPS:-**

* There are many benefits to using a virtual private server, including affordability, scalability, security, and customizable resources.

**Q. How large can I make my attached disk?**

🡪 Each attached disk can be up to 16 TB.

**Q. How many disks can I attach per Light-sail instance?**

🡪 You can attach up to 15 disks per Light-sail instance.

**Q. Can I attach a disk to more than 1 instance?**

🡪 No, disks can only be attached to one instance at a time.

**Q. Does my disk need to be attached to an instance?**

🡪 No, you can choose not to attach a disk to an instance. The disk will remain in your account in an unattached state. There is no difference in price if your disk is not attached to an instance.

**Q. Can I increase the size of my attached disk?**

🡪 Yes, you can increase the size of a disk by taking a disk snapshot and then creating a new, larger disk from snapshot.

**Q. Can I assign one instance to multiple load balancers?**

**🡪** Yes, Light-sail supports adding instances as target instances for more than one load balancer, if desired.

**Q. What kind of connections do Light-sail load balancers support?**

**🡪** Light-sail load balancers support HTTP and HTTPS connections.

**Q. Can I pull my container images from a private container registry?**

**🡪** Currently, only public container registries are supported by Light-sail container services. Alternately, you can push your custom container images from your local machine to Light-sail to keep them private.

**Q. Can Light-sail container service run Docker containers?**

**🡪** Yes. Light-sail supports Linux-based Docker containers. Windows containers are currently not supported.

**Q. What are snapshots?**

**🡪** Snapshots are point-in-time backups of instances, databases, or block storage disks. You can create a snapshot of your resources at any time.

**Q. Can I manage firewall settings for my instance?**

**🡪** Yes. You can control the data traffic for your instances by using the Light-sail firewall. From the Light-sail console, you can set rules about which ports of your instance are publicly accessible for different types of traffic.

**54) Amazon Inspector:-**

* Amazon inspector is an automated vulnerability management service that continually scans AWS workloads for software vulnerabilities unintended network exposure.
* Amazon inspector now supports windows operating system (OS) for continual software vulnerability scanning of EC2 workloads.

**\*Benefits:-**

* Automate the security assessment.
* Assess the compliance with best practices & policies.
* Discovers automate the security assessment workloads.
* Such as amazon EC2 instances, Containers & lambda functions.

**\*Managing multiple accounts in amazon inspector with AWS organizations:-**

* You can manage multiple accounts that are associated through AWS Organization. To manage multiple amazon inspector accounts.

**\*How?**

* Designate an account within the organization as the delegated administrator accounts for amazon inspector.

**\*What?**

* Activating scans (Member account)
* Deactivating scans (Member account)
* Viewing aggregated finding (ORG)

**\*AWS Organization:-**

🡪AWS organization is an account management service that enables you to consolidate multiple AWS account into an organization that you create & centrally manage.

**\*Member accounts within an organization can also perform the following tasks in amazon inspector:-**

* Activate amazon inspector for their own account.
* View resource coverage for their own account.
* View finding details for their own account.
* View the ECR container image automated re-scan duration setting for their own account.

**\*Amazon inspector delegated administrator account can perform the following tasks:-**

* View & manage the status of amazon inspector – including activating & deactivating amazon inspector.
* Activate or deactivate scans for all member account in the organization .
* View aggregated finding data across the organization.
* Finding details for all member accounts within the organization.
* Create & manage suppression rules that apply to findings for all accounts in the organization.
* View resource coverage for the entire organization.

**Q. How is a finding severity determined?**

**🡪** Amazon Inspector Score Severity

0 Informational

0.2–3.9 Low

4.0–6.9 Medium

7.0–8.9 High

9.0–10.0 Critical

**Q.Is it possible for member accounts in a multi-account setup to modify the scan mode for EC2 scanning for their respective accounts?**

🡪 No, in a multi-account setup, only delegated admins can set up scan mode configuration for the complete organization.

**Q. Do I need to enable Amazon Inspector to use Amazon Inspector CI/CD integration for container image scanning?**

🡪 No, you don’t need to enable Amazon Inspector to use this feature provided you have an active AWS account.

**Q. Can I scan my private Amazon EC2 instances by setting up Amazon Inspector as a VPC endpoint?**

**🡪** Yes. Amazon Inspector uses SSM Agent to collect application inventory, which can be set up as Amazon Virtual Private Cloud (VPC) endpoints to avoid sending information over the internet.

**Q. Which programming language packages does Amazon Inspector support for container image scanning?**

**🡪** 1)Java

2) JavaScript

3) Python

**Q. Will Amazon Inspector work with instances that use Network Address Translation (NAT)?**

**🡪** Yes. Instances that use NAT are automatically supported by Amazon Inspector.

**Q. I use a proxy for my instances. Will Amazon Inspector work with these instances?**

**🡪** Yes

**Q. Does Amazon Inspector work with AWS Partner solutions?**

**🡪** Yes.

**Q. Can I deactivate Amazon Inspector?**

**🡪** Yes. You can deactivate all scanning types (Amazon EC2 scanning, Amazon ECR container image scanning, and Lambda function scanning) by deactivating the Amazon Inspector service, or you can deactivate each scanning type individually for an account.

**Q. Can I suspend Amazon Inspector?**

🡪 No. Amazon Inspector does not support a suspended state.

**55) Amazon EMR:-**

* Amazon EMR is a managed cluster platform that simplifies running big data frameworks, such as Apache Hadoop & Apache spark, on AWS to process & analyse vast amount of data.
* The central component of amazon EMR is the cluster.
* A cluster is a collection of EC2 instances. Each instance in the cluster is called node.
* Each node has a role within the cluster.

\*In AWS EMR, when submitting a pyspark job, you can choose between two nodes cluster mode & client mode.

**1) Cluster Mode:-** In cluster mode, the driver program runs on one of the nodes in the EMR cluster.

The cluster manages the resources & handles the execution of the driver program.

**2) Client Mode:-** In client mode, the driver program runs on the machine from the job is submitted rather than on the EMR cluster itself.

**\*Difference between Hadoop & Spark.**

|  |  |
| --- | --- |
| **Hadoop** | **Spark** |
| 1) Hadoop stores and processes data on external storage. | 1) Spark stores and process data on internal memory. |
| 2) Hadoop processes data in batches. | 2) Spark processes data in real time. |
| 3) Hadoop is affordable. | 3) Spark is comparatively more expensive. |
| 4) Hadoop has strong security & Hadoop is easily scalable by adding more nodes | 4) Spark has basic security & Spark is comparatively more challenging. |

**\*Spark Architecture:-**

**1) Driver:-** The driver is the central component of the spark application. It runs the main function & defines the high-level control flow of the application.

**2) Executors:-** Executors are distributed worker nodes that executes tasks on behalf of the driver.

**3) Cluster Manager:-** The cluster manager is responsible for managing the resources of the cluster & allocating them to spark application.

**4) Distributed Storage:-** Spark support various distributed storage systems, such as Hadoop Distributed File System (HDFS), S3.

\*Node type in EMR:-

**1) Primary Node/Master Node:-** The primary node manages the cluster and typically runs primary components of distributed applications. The primary node tracks the status of tasks & monitor the health of the cluster.

**2) Core Node:-** Core nodes are responsible for storing & processing data in an EMR cluster. They run the data processing frameworks & distribute the workload across the cluster.

**3) Task Node:-** Task nodes are additional worker nodes in the EMR cluster that can be added or removed dynamically based on the workload. They are primary used for parallel data processing tasks & don’t store data persistenty.

**\*Architecture of EMR:-** EMR provides a wide range of pre-installed big data tools & frameworks, including sparks, Hadoop Map-Reduce, Hive, Pig, Presto.

**1) Managed Big Data processing:-** EMR is fully managed service that enables processing & analysis of large amount of data using popular distributed frameworks like Apache Spark, Apache Hadoop, Map-Reduce, Hive & more.

**2) Data storage with AWS S3:-** EMR leverages AWS S3 as the primary data storage layer, data is stored in S3.

**3) Master & Core nodes:-** It is a collection of master node & core node. We can create 1 master node & multiple core node.

**4) YARN:-** YARN allow EMR to efficiently allocate resources. YARN is used for cluster resource allocation & management.

**\*Advantages of EMR:-**

1) Low cost

2) Scalability

3) Integration with AWS services

4) Security.

**Q. How can I deploy and manage Amazon EMR?**

**🡪** You can deploy your workloads to EMR using Amazon EC2, Amazon Elastic Kubernetes Service (EKS), or on-premises AWS Outposts.

**Q. How do I develop a data processing application?**

**🡪** You can develop, visualize and debug data science and data engineering applications written in R, Python, Scala, and PySpark in Amazon EMR Studio. You can also develop a data processing job on your desktop.

**Q. Can I add steps to a cluster that is already running?**

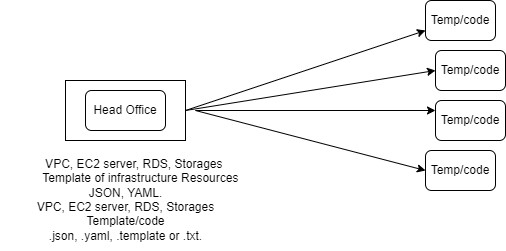
**🡪** Yes. Once the job is running, you can optionally add more steps to it via the AddJobFlowSteps API.

**Q. Is there a limit to the amount of data that can be analyzed with Amazon EMR? If yes, then what is it?**

**🡪** There is no limit to the amount of data that can be analyzed with Amazon EMR.

**56) AWS Cloud-formation:-**

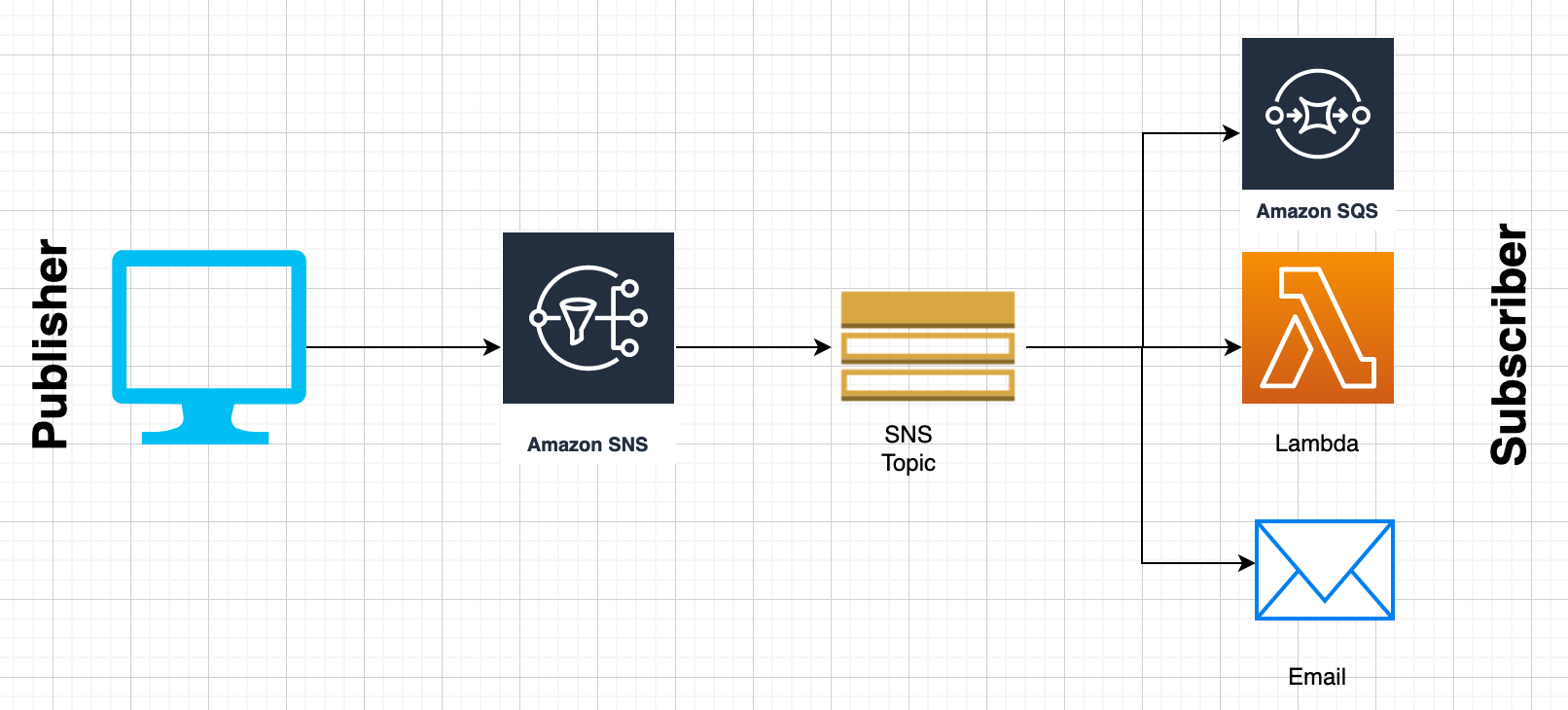
* AWS Cloud-Formation is an infrastructure as code (IAC) service that allows you to easily model, provision, and manage AWS and third-party resources.
* It is used to own your template or sample temp to quick get started.
* You create a template that describes all the AWS resources that you want (like Amazon EC2 instances or Amazon RDS DB instances).
* Cloud-Formation takes care of provisioning and configuring those resources for you.
* Cloud-Formation is designer tool🡪Write a template for infrastructure.



**\*Stack:-** It is a group of resources.

**57) SNS (Simple Notification Service):-**

* SNS is a fast, flexible, fully managed push notification service.
* It is push based system & It is easy way to setup & operate to send notification from cloud.
* SNS is used to publish msg & we can send msg to emails/Mobiles/SMS/Lambda. Etc.
* It is delivered msg immediately to subscriber or two subscribed application.
* It can send msg to all popular devices like android, google & apple.
* It is a way of sending messages. When you are using auto-scalling , it triggers an SNS service which will email, you that ‘your ec2 instance is growing’.



**\*Publisher:-** Publisher are also known as producer that produce & send the message to the SNS which is a logical access point.

**\*Subscriber:-** Subscriber such as webservers, e-mail, Amazon SQS Queues, AWS Lambda, receive the message or notification from the SNS over one of the supported protocols (AWS SQS, E-mail, Lambda, HTTPS, SMS).

**\*SNS Topic:-**

* It is a logical access point & communication channel.
* Each topic has unique name.
* A topic name is limited to 256 alphanumeric character.
* The topic name must be unique within the AWS acc.
* Each topic is assigned an AWS ARN once it gets created.
* A topic can support subscriber & notification deliveries over multiple protocols.
* Messages/request published to a single topic can be delivered over multiple protocols as configured when creating each subscriber.
* Delivery formats /transport protocols (endpoints)

SMS

E-mail

E-mail-JSON 🡪for application

SQS

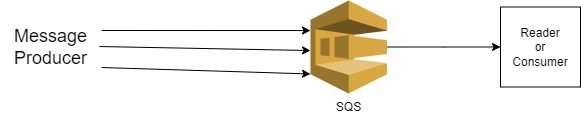
AWS Lambda

**\*AWS SNS Pricing:-**

* **Publish Action🡪** Each 64 KB of request payload count as one request. so, 256 KB payload will charged as four request.
* **Mobile Push Notification🡪** For Ex. $0.50 /million request.
* **SMS 🡪** Price depends on country.
* **E-mail🡪** $2/100000
* **SQS & lambda** calls are free. These are charged at SQS & Lambda roles.
* **Data Transfer**

**58) SQS (Simple Queue Service):-**

* SQS is a pull based service &fast, reliable, fully managed message queue service.
* It is a web-service that gives you access to messages waiting to be processed.
* It offers a reliable, highly scalable, hosted queue for storing messages between server.
* Using SQS, you no longer need a highly available message cluster or the burden of running it
* It can contain up-to 256 KB in any format JSON, XML, etc.
* You can delete all the messages in an SQS queue without deleting the SQS queue itself.
* You can use applications on EC2 instances to read & process the SQS queue messages.



**\*Benefits of SQS:-**

* Highly scalable Standard and FIFO queues.
* Durability and availability
* Security
* Batching

**\*AWS SQS Types:-**

**A) FIFO (First In First Out) Queue:-**

* Limited throughput (300 TPS).
* Exactly one processing.
* Duplicacy not possible.
* Strict ordering 🡪 first-in-first-out.
* FIFO queues are limited to 300 transactions per second (TPS), but have all the capabilities of standard queue.

**B) Standard Queue (D):-**

* High unlimited throughputs.
* At least one region.
* Standard Queue 🡪 $0.40/million request
* FIFO Queue 🡪 $0.50/million request.

**\*SQS Pricing:-**

* The first 1 million monthly requests are free, after that pricing is according to region for ex. Mumbai region.
* Standard Queue🡪 $0.40 /million request.
* FIFO Queue🡪 $0.50 /million request.

**\*How Amazon SQS Charges:-**

* **API Action**🡪 Every Amazon SQS action count as a request.
* **FIFO Request**🡪 API action for sending, Receiving, Deleting & changing visibility of messages from FIFO queue are changed at FIFO rates.
* **Content of Request**🡪 A single request can have from 1 to 10 messages, up-to a maximum total payload of 256 KB.
* **Size of Payload**🡪 Each 64 KB chunk of payload is billed as 1 request (for ex. API action with a 256 KB payload is billed as 4 request).
* Interaction with Amazon S3.
* Interaction with AWS KMS.

**\*Short Polling:-**

A request is returned immediately even if the queue is empty.

* It does not wait for messages to appear in the queue.
* It queries only a subset of the available server for messages (based on weighted random distribution).
* Default by SQS.
* Receive message wait time is 0.
* More request are used, which implies higher cost.

**\*SQS-Retention period:-**

* SQS messages can remain in the queue for up-to 14 days (SQS retention period).
* Range is 1 min to 14 days (default is 4 days).
* Once the maximum retention period of a message is reached, it will be deleted automatically from the queue.
* Messages can be sent to the queue & read from the queue simultaneously.
* SQS can be used with Dynamo-DB, EC2, ECS, Redshift, RDS, lambda, s3 to make distributed/decoupled applications.
* You can have multiple queues with different priorities.

**\*SQS Visibility Timeout:-**

* Is the duration of time a messages is locked for read by other services.
* Max is 12 hours & default is 30 sec.
* A server that read a message to process it, can change the message visibility timeout if it needs more time to process the message.

**After a message is read there are the following possibilities:-**

* An ACK is received that a message is processed, so it must be deleted from the queue to avoid duplicates.
* If a fail is received or the visibility timeout expires, the messages will then be unlocked for read, such that can be read & processed by another service.

**\*Delivery Delay:-**

* AWS SQS provides delivery delay option to postpone the delivery of new message to a queue
* If delivery delay is defined for a queue, any new message will not be visible to the server for the duration of delay.
* The default (min.) delay for a queue is 0 sec & max. is 15 min.

**\*Receive Message Wait Time:-**

* The receive message wait time is the maximum amount of time that polling will wait for messages to become available to receive.
* The minimum value is 0 seconds and the maximum value is 20 seconds.

**\*Dead Letter Queue:-**

* If a message can't be consumed successfully, you can send it to a dead-letter queue (DLQ).
* Dead-letter queues let you isolate problematic messages to determine why they are failing.
* Don’t use a dead letter queue with a FIFO Queue, if you don’t want to break the exact order of messages.
* DLQ must be of the same type as the source queue. (Standard or FIFO)

**59) AWS Athena:-**

* AWS Athena is a query service that allows you to easily analyse data in Amazon S3 using standard SQL.
* Because Athena is server-less, there is no infrastructure to manage & you only pay for the queries you run.

**\*Benefits:-**

* Begin querying immediately. It is server-less, No ETL.
* It is standard, open & power-full.
* It is presto-based application that runs standard SQL.
* It is really fast.
* Even for large datasets, interactive performance is excellent.
* Pay per search.
* Only pay for data that has been scanned.

**\*Pricing:-**

* Amazon Athena charges you for the no. of bytes scanned, rounded upto the nearest megabyte with a 10 MB minimum per query.
* There are no fees for data definition language (DDL) statement such as CREATE/ALERT/DROP TABLE, partition management.
* Queries that are cancelled are charged based on the amount od data scanned.

**60) AWS Lake Formation:-**

* A Data lake is a central location that holds a large amount of data in its native, raw format.
* Compared to hierarchical data warehouse, which store data in files or folders.
* A Data lake uses a flat architecture & object storage to store the data.
* A centralised repo that allows you to store all your structured data & unstructured data at any scale.
* Collect & catalog data from databases & object storage.
* Secure access to the data.
* Build on the capabilities available in AWS Glue.
* Managed data centrally in AWS S3 based data lake.