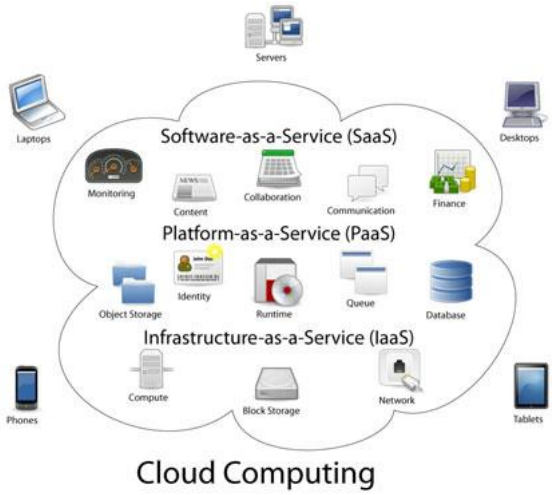
**What is Cloud Computing?**

- **It is the use of remote servers on the internet to store, manage and process data rather than a local server or personal computer.**

**There are basically 3 categories in cloud computing:**





## What is AWS?

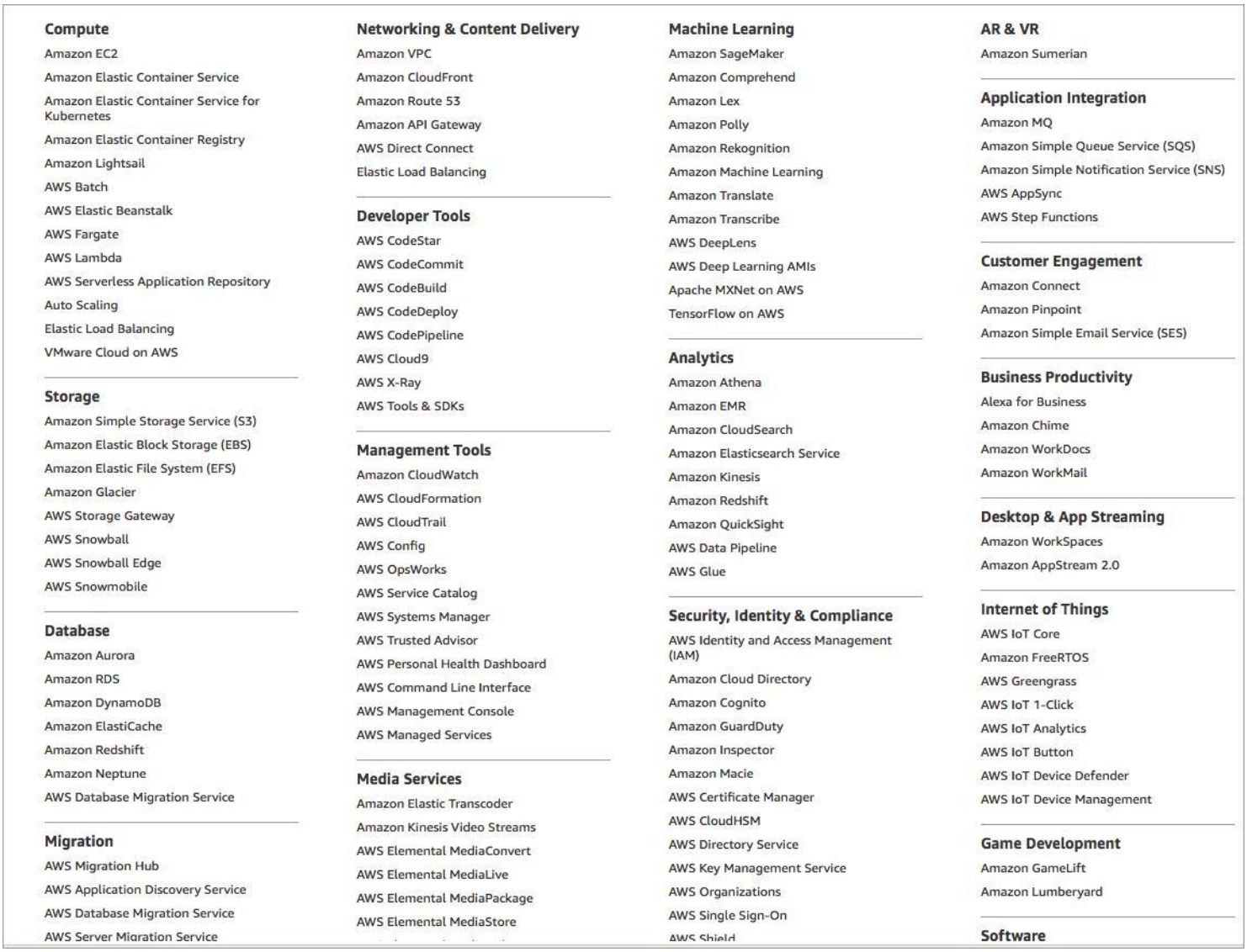
- AWS stands for **Amazon Web Services**.
- The AWS service is provided by the Amazon that uses distributed IT infrastructure to provide different IT resources available on demand. It provides different services such as infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS).

- Amazon launched AWS, a cloud computing platform to allow the different organizations to take advantage of reliable IT infrastructure.

## Uses of AWS

- A small manufacturing organization uses their expertise to expand their business by leaving their IT management to the AWS.
- A large enterprise spread across the globe can utilize the AWS to deliver the training to the distributed workforce.
- An architecture consulting company can use AWS to get the high-compute rendering of construction prototype.
- A media company can use the AWS to provide different types of content such as ebox or audio files to the worldwide files.

## AWS (Amazon Web Services)




different domains in which AWS offer services:

- **Compute**
  It is used to process data on the cloud by making use of powerful processors which serve multiple instances at a time.
- **Storage and Content Delivery**
  The storage as the name suggests, is used to store data in the cloud, this data can be stored anywhere but content delivery on the other hand is used to cache data nearer to the user so as to provide low latency.
- **Database**
  The database domain is used to provide reliable relational and non-relational database instances managed by AWS.
- **Networking**
  It includes services which provide a variety of networking features such as security, faster access etc.
- **Management Tools**
  It includes services which can be used to manage and monitor your AWS instances.
- **Security and Identity**
  It includes services for user authentication or limiting access to a certain set of audience on your AWS resources.
- **Application Services**
  It includes simple services like notifications, emailing and queuing.

To include every customer need under the sun, amazon has further categorized services under each domain. Let's discuss each one of them.

**Compute Services**

- Amazon EC2
- Amazon EC2 Auto Scaling
- Amazon Elastic Container Registry
- Amazon Elastic Container Service
- Amazon Elastic Kubernetes Service
- Amazon Lightsail
- AWS Batch
- AWS Elastic Beanstalk
- AWS Fargate
- AWS Lambda
- AWS Serverless Application Repository
- AWS Outposts
- VMware Cloud on AWS

- **AWS EC2**

- EC2 stands for Amazon Elastic Compute Cloud.
- It is a web service which provides re-sizable compute capacity in the cloud.
- It is designed to make the web scale computing easier for developers

**Therefore, AWS EC2 offers 5 types of instances which are as follows:**

- **General Instances**
  - For applications that require a balance of performance and cost.
    - E.g email responding systems, where you need a prompt response as well as the it should be cost effective, since it doesn't require much processing.

- **Compute Instances**
  - For applications that require a lot of processing from the CPU.
    - E.g analysis of data from a stream of data, like Twitter stream

- **Memory Instances**
  - For applications that are heavy in nature, therefore, require a lot of RAM.
    - E.g when your system needs a lot of applications running in the background i.e multitasking.

- **Storage Instances**
  - For applications that are huge in size or have a data set that occupies a lot of space.
    - E.g When your application is of huge size.

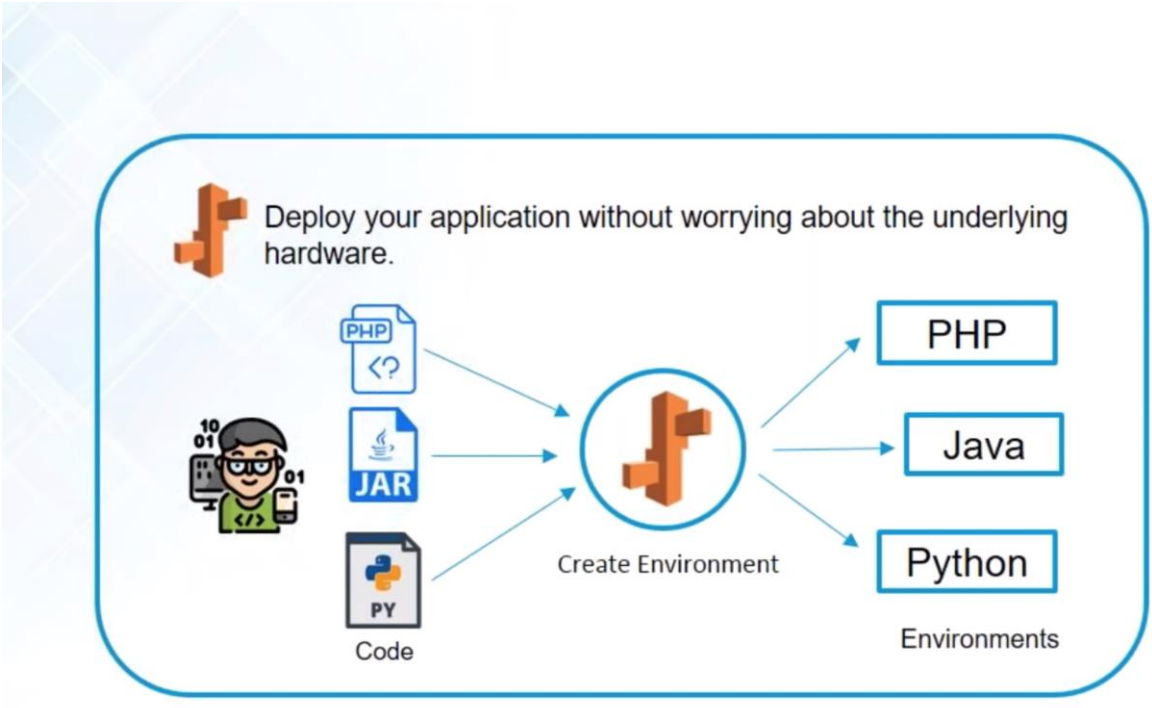
- **GPU Instances**
  - For applications that require some heavy graphics rendering.
    - E.g 3D modelling etc.

**Now, every instance type has a set of instances which are optimized for different workloads:**

- General Instances
  - t2
  - m4
  - m3
- Compute Instances
  - c4
  - c3
- Memory Instances
  - r3
  - x1
- Storage Instances
  - i2
  - d2
- GPU Instances
  - g2

**AWS Elastic Beanstalk**

Elastic Beanstalk quickly deploy and manage applications in AWS without worrying about the underlying infrastructure.

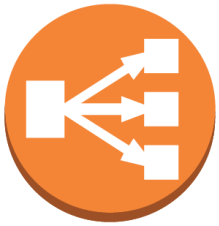



- Elastic Beanstalk is a service provided by AWS which is used for deploying infrastructure which consists of many AWS services.
- These services include **AWS S3**, EC2, **auto-scaling**, cloud watch, Elastic load balancer, and simple notification service.
- It is easy to start with Elastic Bean Stalk as you can see **AWS Management Console**, the command line interface or the API.
- All you have to do is choose your platforms such as Node.js or Ruby and Amazon EC2 instance type.
- After the code is uploaded the AWS Elastic Beanstalk will handle the rest of the activities such as provisioning, load balancing, auto-scaling, and other activities.
- AWS does not implement any extra charges for Elastic Beanstalk as you have pay only for the AWS resources needed to run your applications without any hidden or upfront cost.

| EC2 | Beanstalk |
|---|---|
| EC2 is Amazon's service that allows you to create a server (AWS calls these instances) | Elastic Beanstalk is one layer of abstraction away from the EC2 layer. Elastic Beanstalk |

| | |
|---|---|
| in the AWS cloud. You pay by the hour and only what you use. You can do whatever you want with this instance as well as launch n number of instances. | will setup an "environment" for you that can contain a number of EC2 instances, an optional database, as well as a few other AWS components such as a Elastic Load Balancer, Auto-Scaling Group, Security Group. Then Elastic Beanstalk will manage these items for you whenever you want to update your software running in AWS. Elastic Beanstalk doesn't add any cost on top of these resources that it creates for you. If you have 10 hours of EC2 usage, then all you pay is 10 compute hours. |
| We can't run our apps on plain EC2 | We can run our apps on EB |
| with an EC2 instance, you can turn it off and on at any time and save money. You can have everything on one and save money too. For large operations, this won't matter, but for a bootstrapped start-up, this makes a difference. | Beanstalk is a good product and really a good fit if you know your service is going to grow. You get load balancers and auto scaling configured automatically, which is way out of my domain |

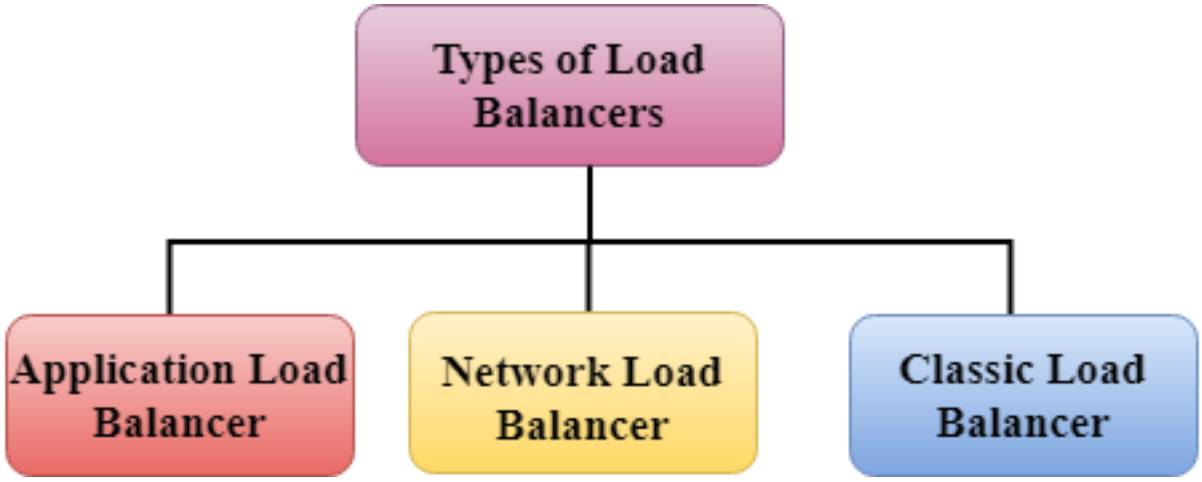
**AWS Elastic Load Balancing**



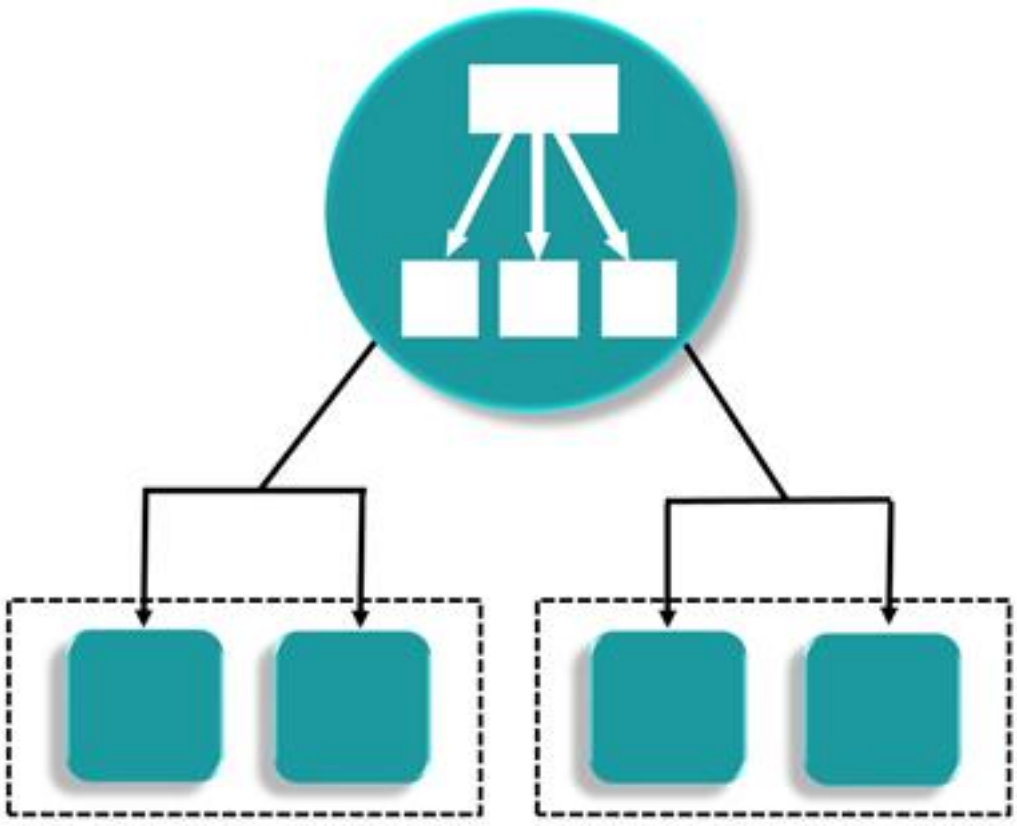
ELB automatically manages the workload on your instances and distributes them to other instances in case of an instance failure.

## What is Load Balancer?

Load Balancer is a virtual machine or appliance that balances your web application load that could be Http or Https traffic that you are getting in. It balances a load of multiple web servers so that no web server gets overwhelmed.



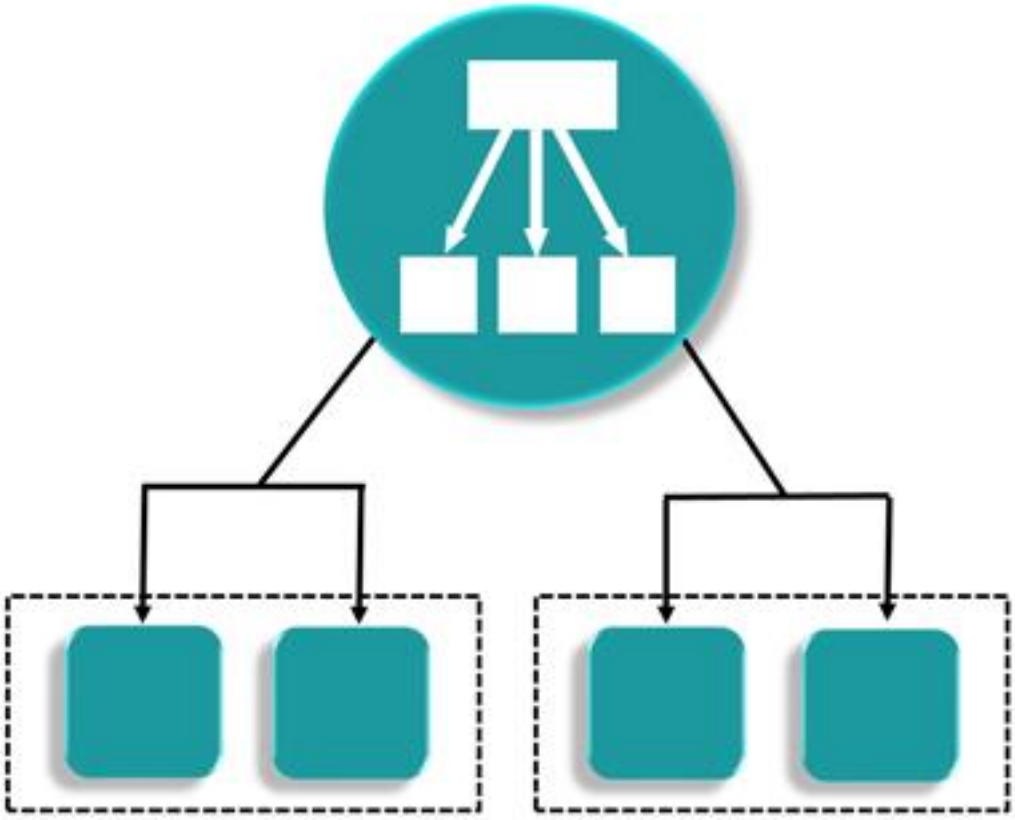

**Application Load Balancer**



- An Amazon Web Services (AWS) launched a new load balancer known as an Application load balancer (ALB) on August 11, 2016.
- It is used to direct user traffic to the public AWS cloud.
- It identifies the incoming traffic and forwards it to the right resources. For example, if a URL has /**API** extensions, then it is routed to the appropriate application resources.
- It is operated at Layer 7 of the OSI Model.
- It is best suited for load balancing of HTTP and HTTPs traffic.
- Application load balancers are intelligent, sending specific requests to specific web servers.

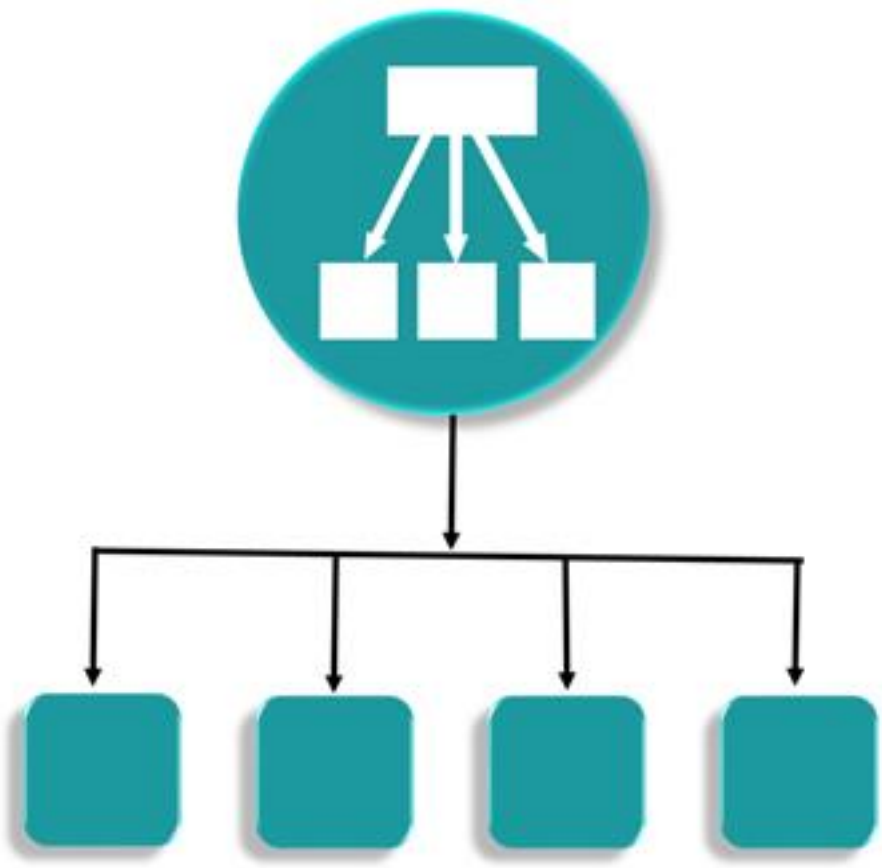
- If we take an example of TESLA. We have three models of TESLA, i.e., TESLA Model X, TESLA Model S, and TESLA Model 3 and TESLAs have onboard computing facility. You will have a group of web servers that serve the Model X, a group of web servers that serve the Model S, and similarly for Model 3. We have one Load balance that checks whether the incoming traffic comes from either Model X, Model S or Model 3, and then sends it to the intended froup of servers.

### Network Load Balancer



- It is operated at the Layer 4 of the OSI model.
- It makes routing decisions at the transport layer (TCP/SSL), and it can handle millions of requests per second.
- When a load balancer receives a connection, it then selects a target from the target group by using a flow hash routing algorithm. It opens the TCP connection to the selected target of the port and forwards the request without modifying the headers.
- It is best suited for load balancing the TCP traffic when high performance is required.

## Classic Load Balancer

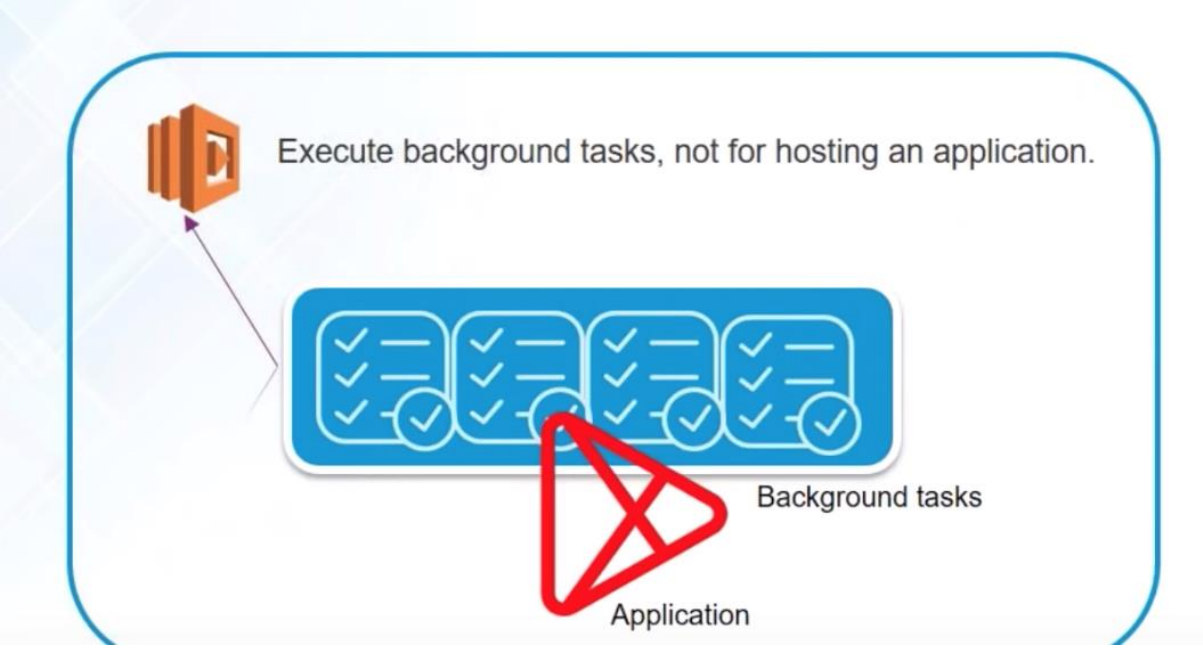


- It is operated at Layer 4 of the OSI model.
- It routes the traffic between clients and backend servers based on IP address.
- For example, an Elastic Load balancer receives a request from a client on TCP port 80, it will then routes the request to a specified port of backend servers. The port on which the Load Balancer routes to the target server will be having port number 80. The backend server will then send the requested data back to the ELB, which will then forward the Backend server reply to the client. According to the client's perspective, the request has been fulfilled by the ELB, not by the backend server.
- Classic Load balancers are legacy Elastic load balancers.
- It can also be used for load balancing the HTTP or HTTPs traffic and use layer 7-specific features, such as X-forwarded and sticky sessions.
- You can also use the Layer 4 load balancing for applications that rely purely on the TCP protocol.

**AWS Lambda**

AWS Lambda is used to execute backend code without worrying about the underlying architecture, you just upload the code and it runs, it's that simple!

AWS Lambda is a compute service offered by Amazon.




What is AWS Lambda?

- Amazon explains, AWS Lambda (λ) as a 'serverless' compute service, meaning the developers, don't have to worry about which AWS resources to launch, or how will they manage them, they just put the code on lambda and it runs, it's that simple! It helps you to focus on core-competency i.e. App Building or the code.
- Lambda is used to encapsulate Data centres, Hardware, Assembly code/Protocols, high-level languages, operating systems, AWS APIs.
- Lambda is a compute service where you can upload your code and create the Lambda function.
- Lambda takes care of provisioning and managing the servers used to run the code.
- While using Lambda, you don't have to worry about scaling, patching, operating systems, etc.
- **Then why not EC2?**

| EC2 | Lambda |
| --- | --- |

| | |
|---|---|
| If you were to use EC2, you would have to architect everything i.e. load balancer, EBS volumes, software stacks etc<br><br>For example, in EC2 you would be installing the software packages on your virtual machine which would support your code | In lambda you don't have to worry about anything, just insert your code, and AWS will manage the rest!<br><br>in Lambda you don't have to worry about any VM, just insert plain code and Lambda will execute it for you. |

- But, if your code will be running for hours, and you expect a continuous stream of requests, you should probably go with EC2, because the architecture of Lambda is for a sporadic kind of workload, wherein there will be some quiet hours and some spikes in the no. of requests as well.
- For example, logging the email activity for say a small company, would see more activity during the day than in the night, also there could be days when there are less emails to be processed, and sometimes the whole world could start emailing you! In both the cases, Lambda is at your service.
- Considering this use case for a big social networking company, where the emails are never ending because it has a huge user base, Lambda may not be the apt choice.

**AWS Autoscaling**

- The Autoscaling feature is used to scale up and down automatically as and when required.
- The application available at AWS requires space and load and the Auto Scaling helps us by providing surety that there is a sufficient number of Amazon EC2 instances available to handle that load.
- **You can set a limit on EC2 instances such that the number doesn't go below this**.
- **The maximum numbers of EC2 instances can be set to be on a safer side.**
- **AWS Autoscaling ensures that your group has a sufficient amount of servers.**
- Auto-scaling automatically modifies the EC2 instance as per your demand changes.
- One can access Auto Scaling by signing into the **AWS Management Console**.
- AWS Auto-scaling helps you if you are using language-specific APIs rather than submitting requests over HTTP or HTTPS Auto Scaling provides a benefit of libraries, Sample code, tutorial, and other resources for the development of the software.
- It also helps us with some functions such as retrying requests, and handling error responses, making it is easier for the applicant to get started.

Amazon Elastic Container Service

## AMAZON EC2 CONTAINER SERVICE (ECS)

- Amazon EC2 Container Service (Amazon ECS) is a highly scalable, fast, container management service that makes it easy to run, stop, and manage Docker containers on a cluster of Amazon EC2 instances.

- Amazon ECS uses **Docker images** in task definitions to launch containers on EC2 instances in our clusters.

- Docker is a technology that allows us to build, run, test, and deploy distributed applications that are based on Linux containers.

ECS is basically a set of APIs that turn EC2 instances into compute cluster for container management:

1. EC2 instances must call RegisterContainerInstance API to signal that they are ready to run containers.
2. Need to call RegisterTaskDefinition API to define the tasks (setting an image, command and memory for docker run etc.)
3. We use RunTask API to start a new task.
4. Lastly, we make a CreateService API call to run a long-running container.

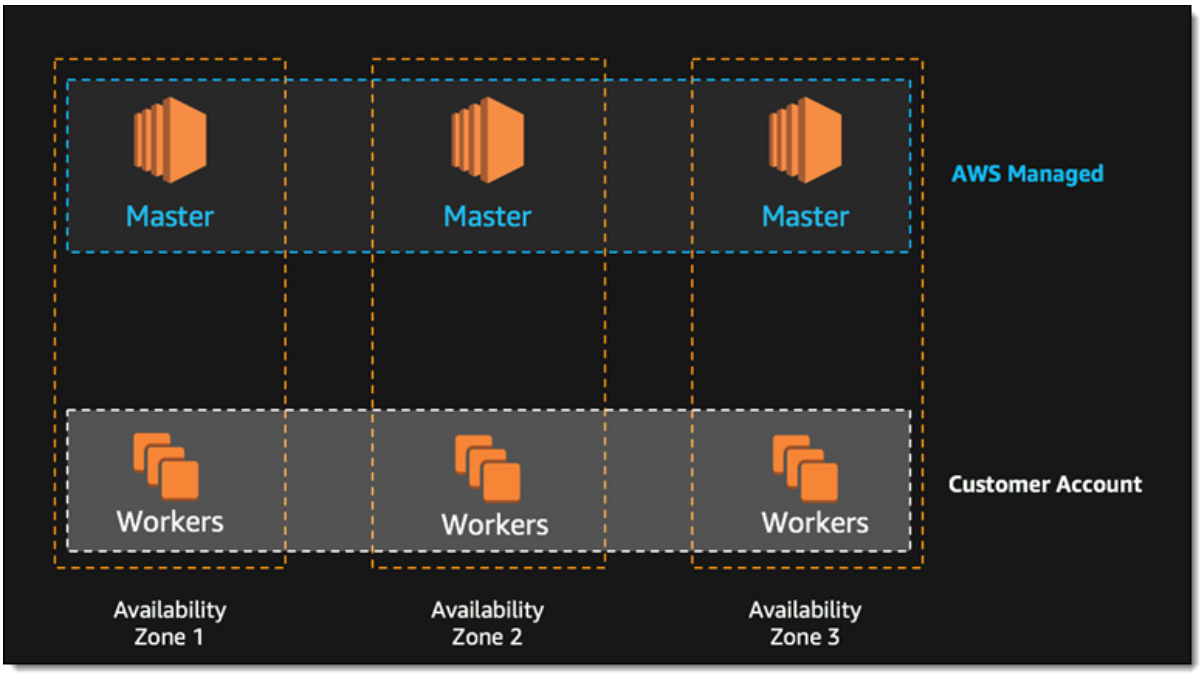
- We can start using Amazon EC2 Container Service (Amazon ECS) by creating a task definition, scheduling tasks, and configuring a cluster in the Amazon ECS console. **Note that we do not need any orchestration tools such as Mesos, Kubernetes or Docker Swarm.**

Amazon Elastic Kubernetes Service

## AWS : EKS (ELASTIC CONTAINER SERVICE FOR KUBERNETES)

- Amazon Elastic Container Service for Kubernetes (Amazon EKS) is a fully managed service that makes it easy for you to use Kubernetes on AWS without having to be an expert in managing Kubernetes clusters.
- There are few things that we think developers will really like about this service.
- First, Amazon EKS runs the upstream version of the open-source Kubernetes software, so you can use all the existing plugins and tooling from the Kubernetes community.
- Applications running on Amazon EKS are fully compatible with applications running on any standard Kubernetes environment, whether running in on-premises datacentre's or public clouds. This means that you can easily migrate your Kubernetes application to Amazon EKS with zero code changes. Second, Amazon EKS automatically runs K8s with three masters across three AZs to protect against a single point of failure.

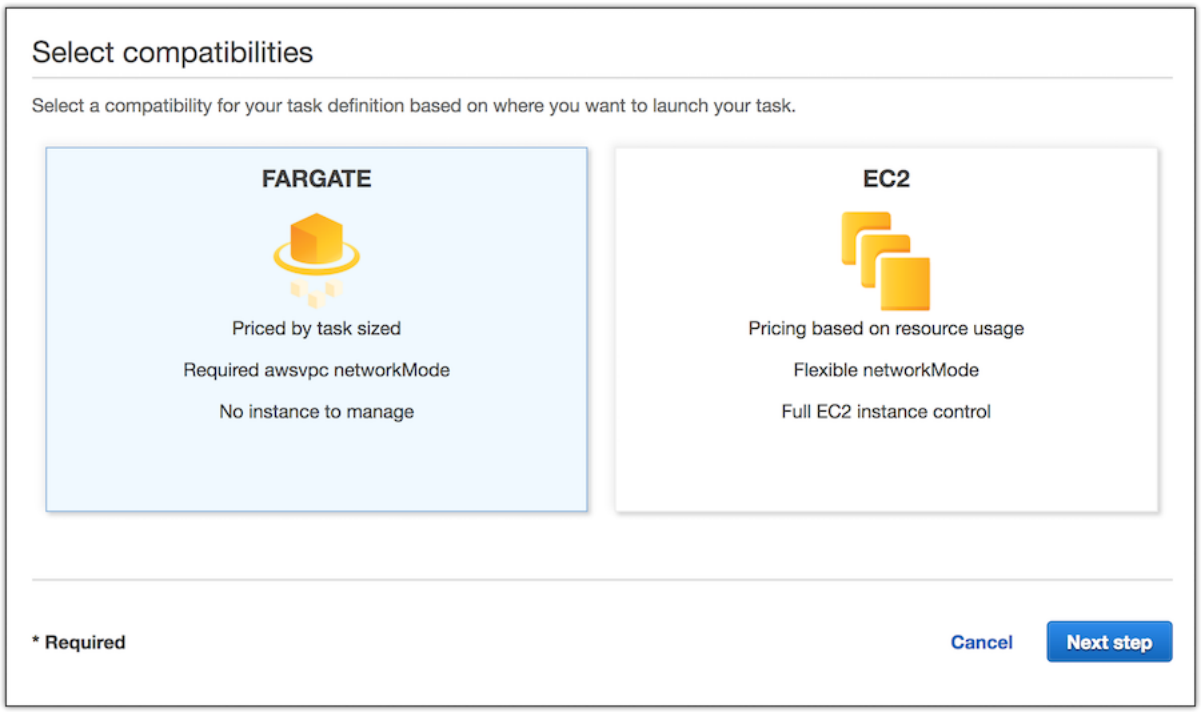
- This multi-AZ architecture delivers resiliency against the loss of an AWS Availability Zone.




- Third, Amazon EKS also automatically detects and replaces unhealthy masters, and it provides automated version upgrades and patching for the masters.
- Last, Amazon EKS is integrated with a number of key AWS features such as Elastic Load Balancing for load distribution, IAM for authentication, Amazon VPC for isolation, AWS Private Link for private network access, and AWS CloudTrail for logging.

**AWS Fargate – A Compute Engine For ECS**

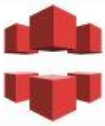
- AWS Fargate is a compute engine for Amazon Elastic Container Service(ECS) that allows you to run containers without having to provision, configure & scale clusters of VMs that host container applications.
- AWS Fargate eliminates the need for users to manage the EC2 instances on their own.
- In fact, users don't need to use EC2 instances at all.
- Fargate itself will act as compute engine.
- It lets you focus on elements like designing and constructing your application instead of managing the infrastructure that runs them.
- With **Fargate launch type**, all you need to do is package your application in containers, specify the memory and CPU requirements, define IAM policies & launch your application.
- AWS Fargate also makes it easy to scale your applications.
- Once you define all your application requirements, AWS Fargate manages all the scaling and infrastructure needed to run your containers in a highly-available manner.
- It seamlessly integrates with Amazon ECS & EKS, launches and manages your containers for you.

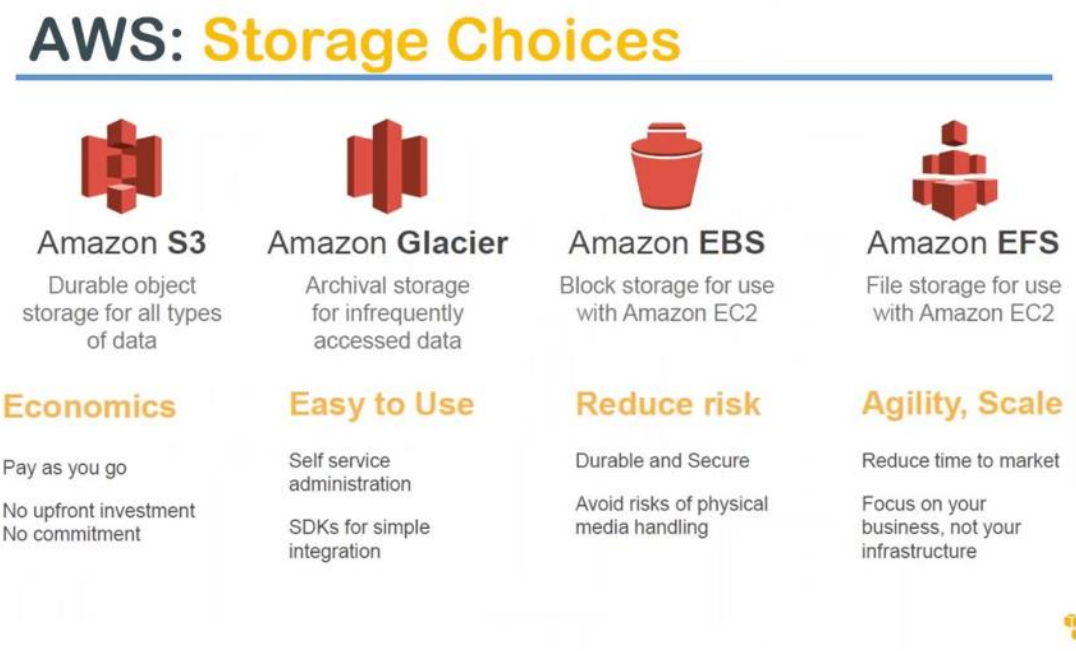



## Storage and Content Delivery

### Storage

- Amazon S3
- Amazon Elastic Block Store
- Amazon Elastic File System
- Amazon FSx for Lustre
- Amazon FSx for Windows File Server
- Amazon S3 Glacier
- AWS Storage Gateway

| | | |
|---|---|---|
|  | Amazon Simple Storage Service (Amazon S3) | A service that provides scalable and highly durable object storage in the cloud. |
|  | Amazon Glacier | A service that provides low-cost highly durable archive storage in the cloud. |
|  | Amazon Elastic File System (Amazon EFS) | A service that provides scalable network file storage for Amazon EC2 instances. |
|  | Amazon Elastic Block Store (Amazon EBS) | A service that provides block storage volumes for Amazon EC2 instances. |
|  | Amazon EC2 Instance Storage | Temporary block storage volumes for Amazon EC2 instances. |
|  | AWS Storage Gateway | An on-premises storage appliance that integrates with cloud storage. |
|  | AWS Snowball | A service that transports large amounts of data to and from the cloud. |
|  | Amazon CloudFront | A service that provides a global content delivery network (CDN). |



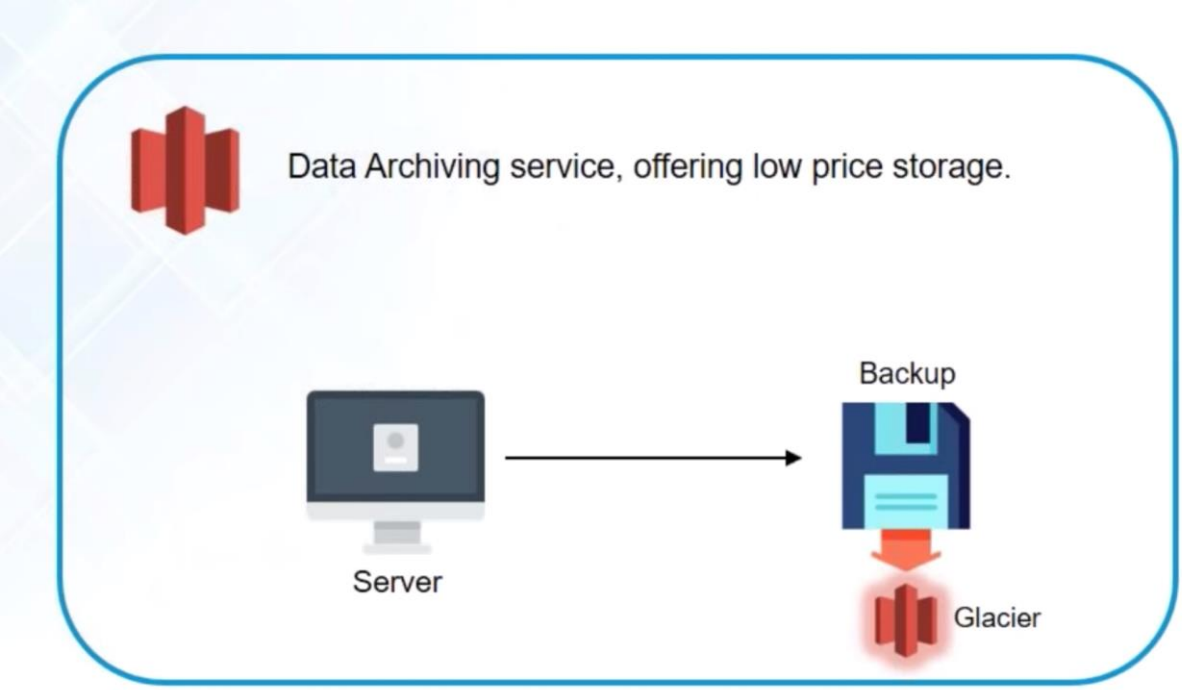
- **S3 AWS**

S3 stands for simple storage service, it is used for storing data in the form of objects in the AWS Cloud.

- Amazon Simple Storage Service (S3) is a storage for the internet.
- It is designed for large-capacity, low-cost storage provision across multiple geographical regions.
- Amazon S3 provides developers and IT teams with **Secure**, **Durable** and **Highly Scalable** object storage.
- S3 is a safe place to store the files.
- It is Object-based storage, i.e., you can store the images, word files, pdf files, etc.
- The files which are stored in S3 can be from 0 Bytes to 5 TB.
- It has unlimited storage means that you can store the data as much you want.
- Files are stored in Bucket. A bucket is like a folder available in S3 that stores the files.
- S3 is a universal namespace, i.e., the names must be unique globally. Bucket contains a DNS address. Therefore, the bucket must contain a unique name to generate a unique DNS address.

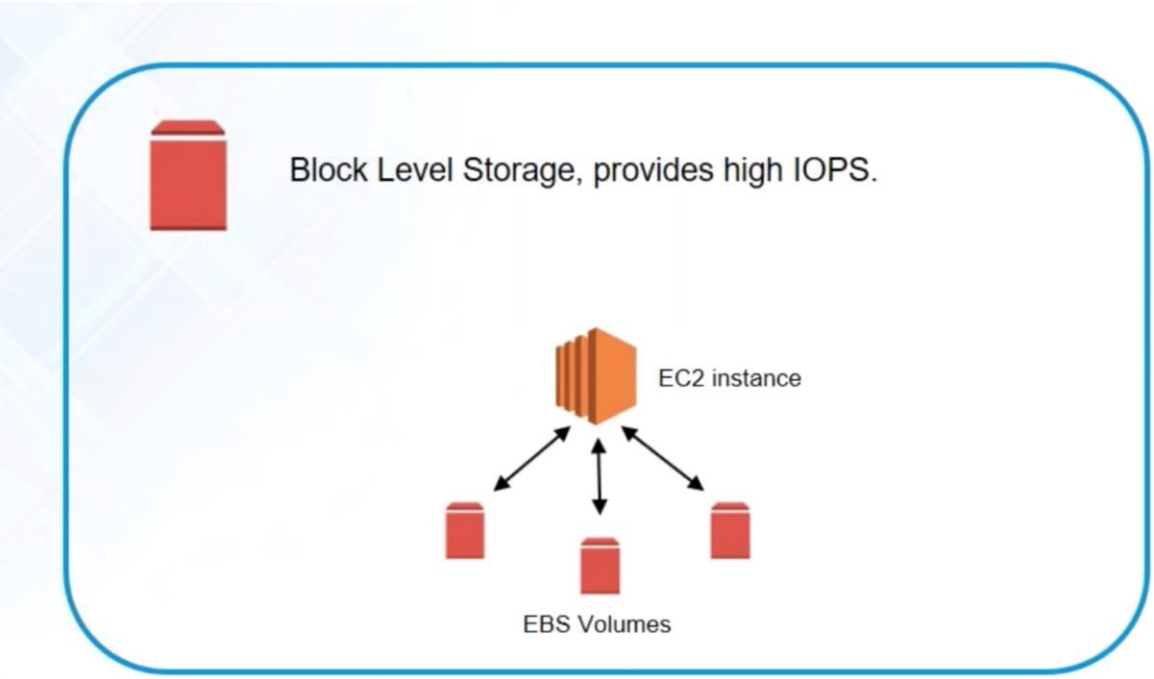
**Amazon Glacier**

- Glacier is an archiving service offered by Amazon, which offers low cost data archiving.
- **Amazon Glacier is extremely low cost, secure, and durable storage service for data archiving and backup**.
- It is designed to keep the cost low and optimized for the cold data where the retrieval time is 3 to 4 hours. Within Glacier, the user can reliably store the small and large amount of data.
- In AWS Glacier, there is no limit for the data user stores. Moreover, the data is secure and can access easily.
- Amazon Glacier helps to protect the data by redundantly storing it on multiple devices using multiple facilities.
- AWS Glacier has a Data Integrity Check which regularly monitors the data in the Glacier.
- It also provides security and fine-grained access to the data of the user with AWS Access Management policies.




**Amazon EBS**

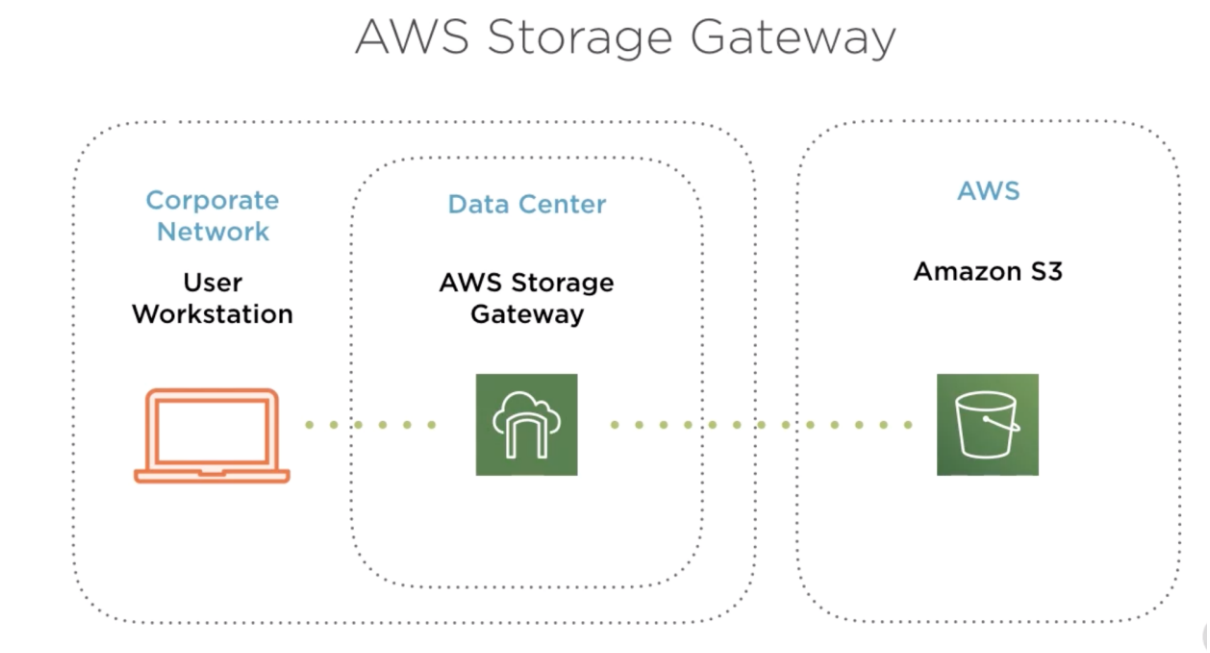
Amazon Elastic Block Storage is a storage service wherein each block of storage acts like a separate hard drive.

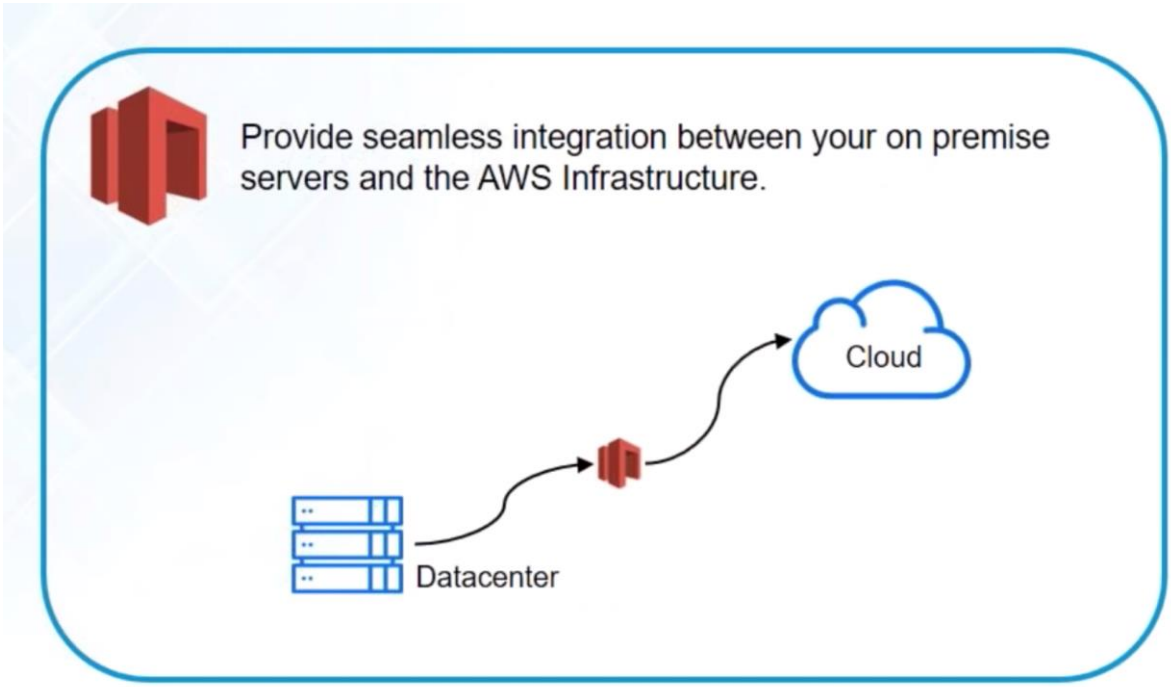



- **Amazon Elastic Block Store** (EBS) is a block storage system used to store persistent data.
- Amazon EBS is suitable for EC2 instances by providing highly available block level storage volumes.
- It has three types of volume
    1. General Purpose (SSD)
    2. Provisioned IOPS (SSD)
    3. Magnetic.

**Amazon EBS Benefits**

- **Reliable and secure storage** − Each of the EBS volume will automatically respond to its Availability Zone to protect from component failure.
- **Secure** − Amazon's flexible access control policies allows to specify who can access which EBS volumes. Access control plus encryption offers a strong defense-in-depth security strategy for data.
- **Higher performance** − Amazon EBS uses SSD technology to deliver data results with consistent I/O performance of application.
- **Easy data backup** − Data backup can be saved by taking point-in-time snapshots of Amazon EBS volumes.



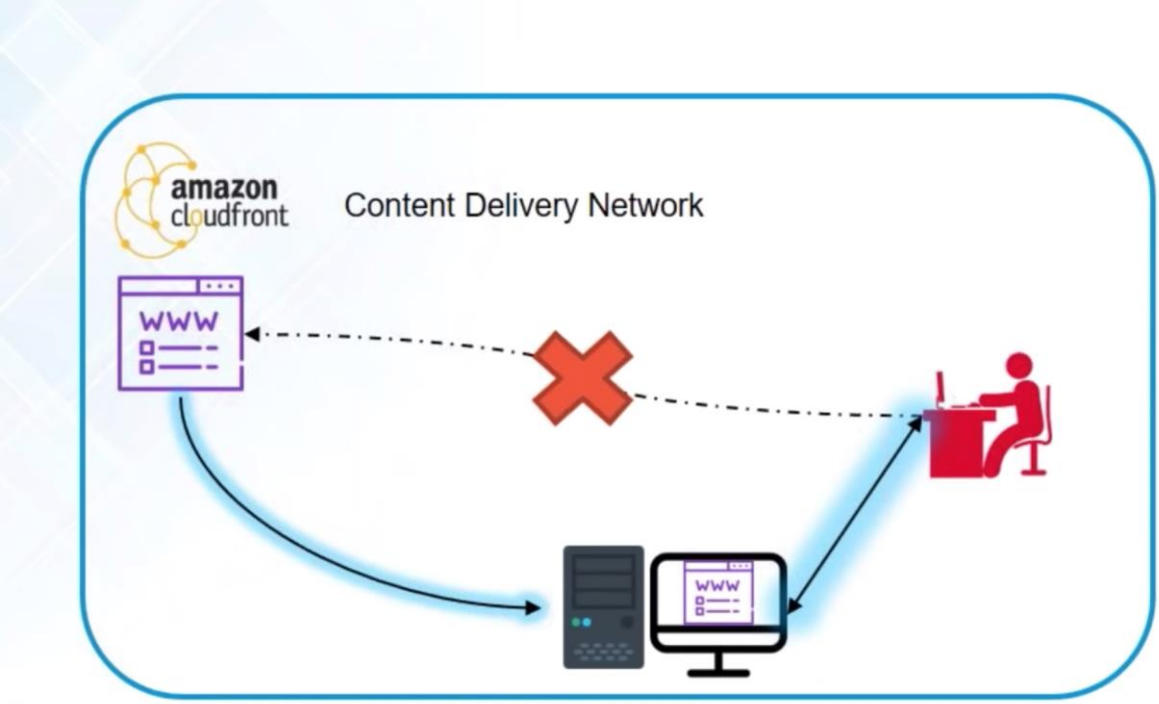




**What is AWS Storage Gateway?**

- **Amazon Storage Gateway is a modified storage service which enables the applications to use the AWS Cloud for storage purpose**.
- **Amazon SG can help for backup and archiving, cloud processing, disaster recovery, and migration**.
- Standard storage protocol such as NFS, SMB, and Amazon EBS connects the applications to a gateway appliance using **standard storage protocol**.

- The gateways get connected to the storage services such as **Amazon S3**, Amazon Glacier, and **Amazon EBS**.
- This service benefits the user in many ways such as It includes highly-optimized data transfer mechanism.
- Low-latency data along with the on-premise local cache provides access to the data.

## CloudFront CDN




- CloudFront CDN (Computer Delivery Network) is a system of distributed servers that deliver web pages and other web content to a user based on the geographic locations of the user, the origin of the webpage and a content delivery server.
- Suppose I am running the website outside the UK and I am serving the website all around the world.
- When the user wants to access my website, then they request to the web server, and users from different countries will have different latency.
- For example, People who live in Australia will have more latency than those who stay in India.
- South Africa has a terrible latency, but they would run internet backbone that makes quicker to connect to the UK.
- This is how it works with CloudFront CDN in which people spread all around the world, and they can turn on access to the web page, audio files, etc. in the UK.

**Snowball**

- The **Snowball** is a way of transferring your data physically. In this Amazon sends an equipment to your premises, on which you can load the data. It has a kindle attached to it which has your shipping address when it is shipped from Amazon. When data transfer is complete on the Snowball,

- kindle changes the shipping address back to the AWS headquarters where the Snowball has to be sent.
- The Snowball is ideal for customers who have large batches of data move. The average turnaround time for Snowball is 5-7 days, in the same time Transfer Acceleration can transfer up to 75 TB of data on a dedicated 1Gbps line. So depending on the use case, a customer can decide.

**Database**

**AWS database service includes the following services:**

- **Amazon Relational Database Service:** It supports six (Amazon aurora ,MySQL ,PostgreSQL's Server ,Oracle, MariaDB)commonly used database engines.
- **Amazon Aurora:** It is a MySQL-Compatible relational database with five times performance.
- **Amazon DynamoDB:** It is a fast and flexible NoSQL database service.
- **Amazon Redshift:** It is a petabyte-scale data warehouse service.
- **Amazon Elasticache:** It is an in-memory cache service with support for Memcached and Redis.
- **AWS Database Migration Service:** It is a service that provides easy and inexpensive to migrate your databases to AWS cloud.

**The Amazon Relational Database Service (RDS AWS)** is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, re-sizable capacity in an industry-standard relational database and manages common database administration tasks.

So people often develop a misconception, when they confuse **RDS with a database.**

**RDS is** not **a database, it's a service that manages databases**, having said that, let's discuss the databases that RDS can manage as of now:

- Amazon aurora
- Mysql
- PostgreSQL
- SQL Server
- Oracle
- MariaDB

- **Amazon Aurora**


It is a relational database engine that combines the speed and reliability of high-end commercial databases and the cost effectiveness and simplicity of open-source databases.

- It is a relational database, and closed source database engine.
- It is compatible with MySQL and delivers five times throughput of MySQL on the same hardware.
- It is also compatible with PostgreSQL and delivers three times throughput of PostgreSQL on the same hardware.
- Amazon RDS with Aurora manages the time-consuming administrative tasks such as software installation, patching, and backups.
- The main features of Aurora are fault-tolerant, distributed, a self-healing storage system that auto-scales upto 64 TB per database instance.
- It provides high-performance, availability, point-in-time recovery, continuous backed up to S3, and replication across three availability zones.

- **Amazon RDS**

Amazon RDS is a managed relational database service which does routine database tasks in 6 familiar databases like Amazon Aurora, MySQL, MariaDB, Oracle, Microsoft SQL Server, and PostgreSQL.

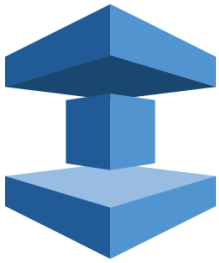
- **Amazon DynamoDB**

It is a fully managed No-SQL database service. It is known for extremely low latencies and scalability.

**Amazon DynamoDB**


DynamoDB is a fully managed NoSQL database service provided by Amazon. These days, databases have become the backbone for any company irrespective of how big they are. Traditional database systems which were initially used, are not the go-to solution today because of the dynamic change in requirements and type of data procured. In this Amazon DynamoDB tutorial, I will be discussing the new and fast way of storing and retrieving data using DynamoDB.

- **Amazon ElastiCache**



- It is a web service that makes it easy to set up, manage and scale a distributed cache-in environment in the cloud.

- **Amazon Redshift**

Amazon Redshift is a fully managed petabyte-scale data warehouse service in the cloud.

**Networking**

Networking and Content Delivery

- Amazon VPC
- Amazon CloudFront
- Amazon Route 53
- AWS Private Link
- AWS Direct Connect
- AWS Global Accelerator
- Amazon API Gateway
- AWS Transit Gateway
- AWS App Mesh
- AWS Cloud Map
- Elastic Load Balancing

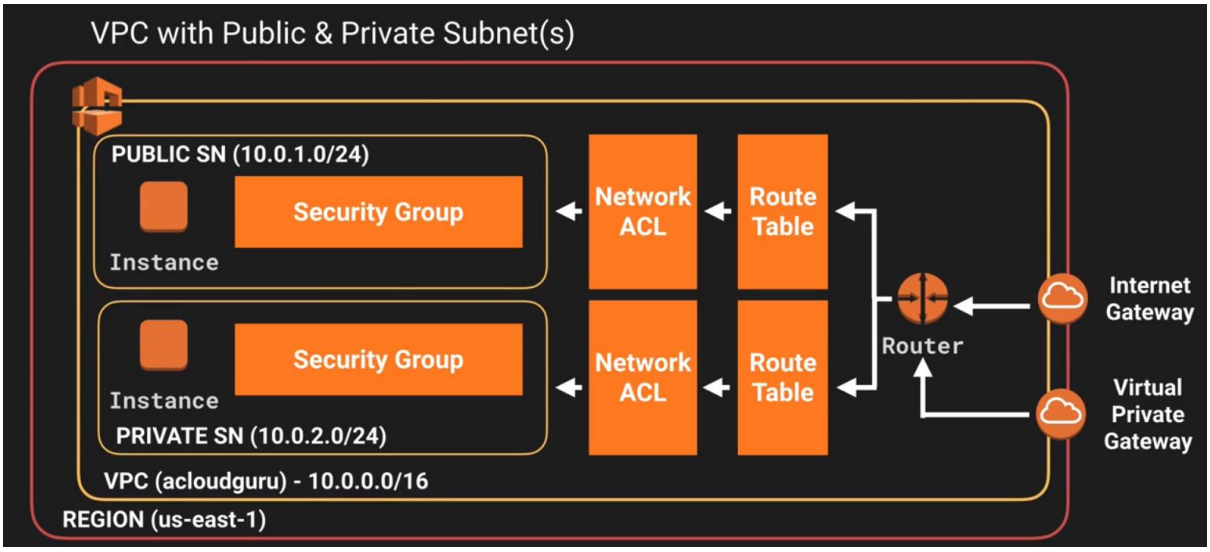
**VPC AWS**

Amazon VPC lets you launch AWS resources in a virtual network that you define. It closely resembles a traditional network that you'd operate in your data centre.

- Amazon Virtual Private Cloud (VPC) helps a firm or a user by providing virtual cloud space for integrating the business.
- With AWS VPC one can completely monitor virtual networking environment, including the selection of your own IP address range, the creation of subnets, and configuration of route tables and network gateways these features helps a lot to integrate businesses.
- Amazon VPC allows you to logically analyse the section of Amazon Cloud where one can launch AWS Resources in the virtual network.

- To provide secure and easy access fourth and sixth revision to the Internet Protocol can be used.

- VPC in AWS as a logical container that separates resources you create from other customers within the Amazon Cloud. It is you defining a network of your own within Amazon.




Subnet and Its Utility
**Subnets are like breaking a large network into sub-networks**. Maintaining a smaller network is easy as compared to maintaining a large network.

What Is Route Table?
Route table can be understood as a **table that contains rules for routing traffic within and outside a subnet**. The route table is also used to add Internet Gateway to the subnet. There can be multiple route tables in a VPC.

What Is Internet Gateway?
**Internet Gateway allows instance to connect to the internet**. It allows the user to make the subnet pubic by providing a route to the internet. With the help of Internet Gateway, an instance can access the internet and the resources outside instance can access the instance.

NAT - Network Address Translation.

**What is NAT?**

NAT is designed for IP address conservation. It enables private IP networks that use unregistered IP addresses to connect to the Internet.

**How does NAT work?**

NAT allows a single device, such as a router, to act as an agent between the Internet and a local network, which means a single unique IP address is required to represent an entire group of computers to public network i.e Outside of their Network.

**What is NAT Instance?**

NAT instance enable instances in the private subnet to initiate outbound traffic to the Internet but prevent the instances from receiving inbound traffic initiated by someone on the Internet.

Note: NAT Instance is a legacy, you can use NAT Gateway

**What is NAT Gateway?**

**NAT Gateway is a managed NAT service that provides better availability, higher bandwidth, and requires less administrative effort.**

- **AWS Direct Connect**

It helps you establish a private connection between your premises and AWS, therefore giving better network performance and throughput than an Internet based connection.

- **Amazon Route 53**

- Route 53 is a highly scalable and highly available Domain Name System by Amazon AWS. The name is in reference to the TCP and UDP's port 53 where DNS requests are addressed.
- AWS Route 53 is a domain name system. Domain name system translates human-readable domain name such as **www.amazon.com** to machine-readable IP address such as 192.0.2.44. Amazon Route 53 connects the request of users to the system running in AWS. This system includes **Amazon EC2** instances, **Elastic Load Balancing** load balancers, or **Amazon S3** buckets. Moreover, it can connect the user infrastructure outside of AWS. Amazon Route 53 is totally compatible with IPv6. It is designed to boost business in a reliable and cost-effective way. AWS Route 53 answers all the queries with the help of the global network of DNS servers.

  Queries of the domain are sent to the nearest DNS Server and thus it answers with the best possible performance. With the help of **AWS management console** or easy-to-use API, one can create and manage the public DNS. AWS Route 53 also helps us to register an available domain name. It helps in a way such that the person has to pay only for the management of domains, and the registered domains in AWS.

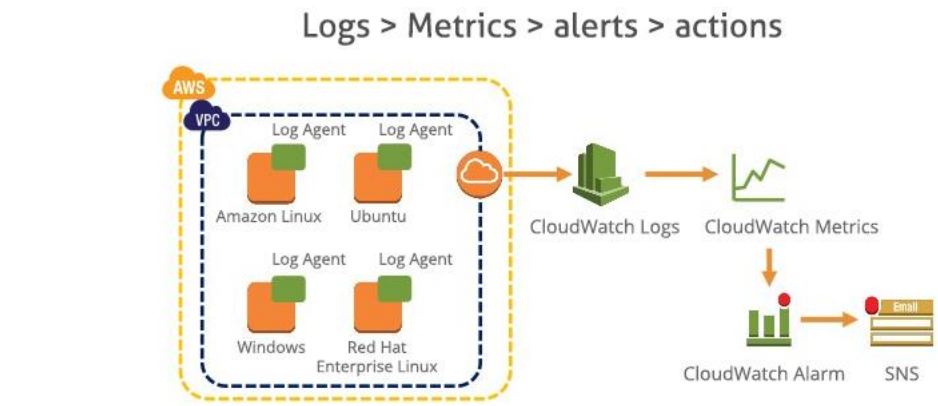
- **Management Tools**

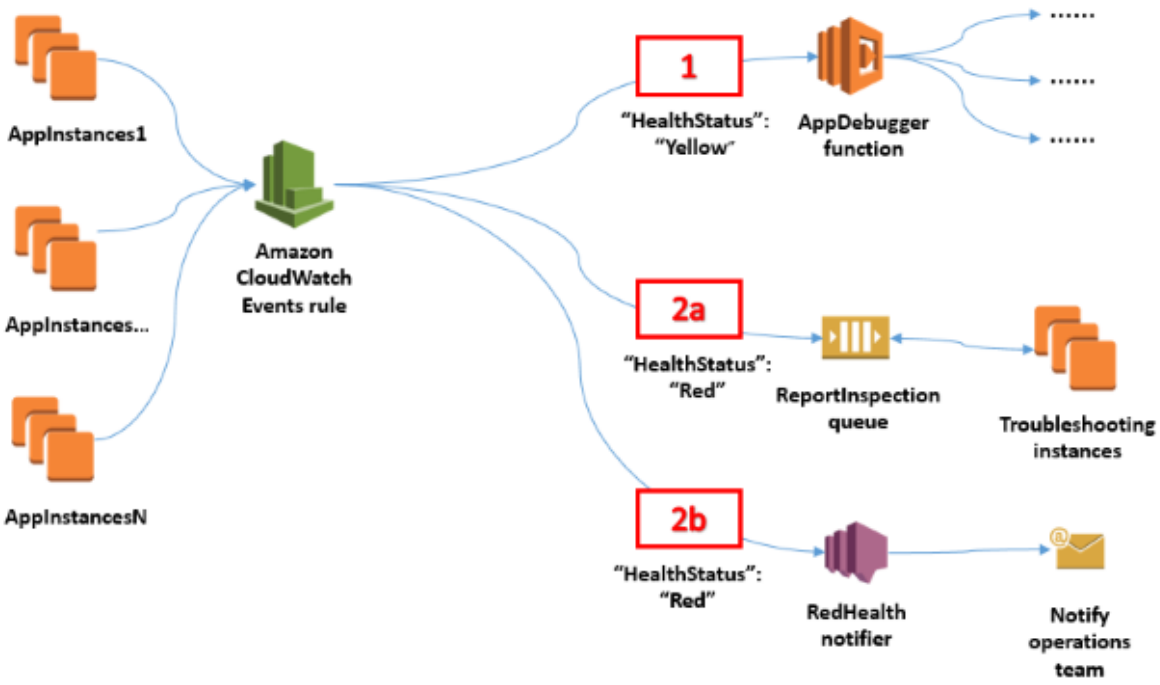
- **Amazon CloudWatch**

It is a monitoring tool by AWS which is used to keep a track on the AWS resources and the applications you run on Amazon AWS.

**Amazon CloudWatch**
Amazon CloudWatch is a monitoring and management service built for developers, system operators, site reliability engineers (SRE), and IT managers. CloudWatch provides you with data and actionable insights to monitor your applications, understand and respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health. CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications and services that run on AWS, and on-premises servers. You can use CloudWatch to set high resolution alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to optimize your applications, and ensure they are running smoothly.

With Amazon CloudWatch, it is easy to get started. There is no up-front commitment or minimum fee; you simply pay for what you use. You will be charged at the end of the month for what you use.

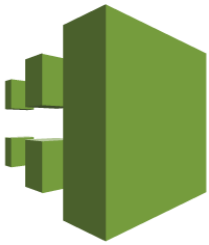



- **AWS CloudFormation**

- It is a service which helps you setup and model your Amazon AWS resources so that you can spend less time managing these resources and more time focusing on the development.

- **AWS CloudTrail**



AWS CloudTrail is a logging service which records the API calls to your Amazon AWS account and delivers them to you.

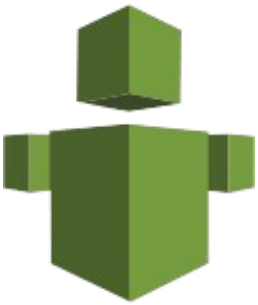
- **AWS Command Line Tool**


- It is an all in one tool to manage all your AWS services, by downloading and configuring only one tool you can manage all the AWS services through the command line.

- **AWS OpsWorks**

It is a configuration management tool that helps configure and operate applications of all size and shapes using Chef.

- **Trusted Advisor**

- 
  Trusted Advisor is a customized cloud monitoring tool, that analyzes your AWS environment and gives insights on the expense, performance improvement, security gaps and reliability.

- **Security and Identity**

**AWS security services**

The services covered within this learning path are as follows:

- AWS Identity & Access Management (IAM)
- AWS Key Management Service (KMS)
- AWS CloudHSM
- AWS WAF
- AWS CloudTrail
- Amazon Inspector
- AWS Config
-

  - **AWS Identity and Access Management(IAM)**

It is an AWS service that helps you control access to your AWS resources for your users.

**Identity and Access Management in(IAM)**


The AWS IAM enables the user to securely control access to AWS services and resources for the users. IAM enables user to create and manage users in AWS, and it also enables the user to grant access to AWS resources for users managed outside the AWS in the corporate directory. IAM enables identity federation between the user's corporate directory and AWS services. This enables the user to use existing corporate identities to grant secure and direct access to AWS resources, such as S3 buckets, without creating a new AWS identity for those users.

The biggest advantage of IAM is that it is free. But if users launch EC2 it will be charged.

**AWS Key Management Service**

- 
  It is a managed service that helps you create and control encryption keys which is used to encrypt your data, and uses Hardware Security Modules to protect the security of your keys.

**Application Services**

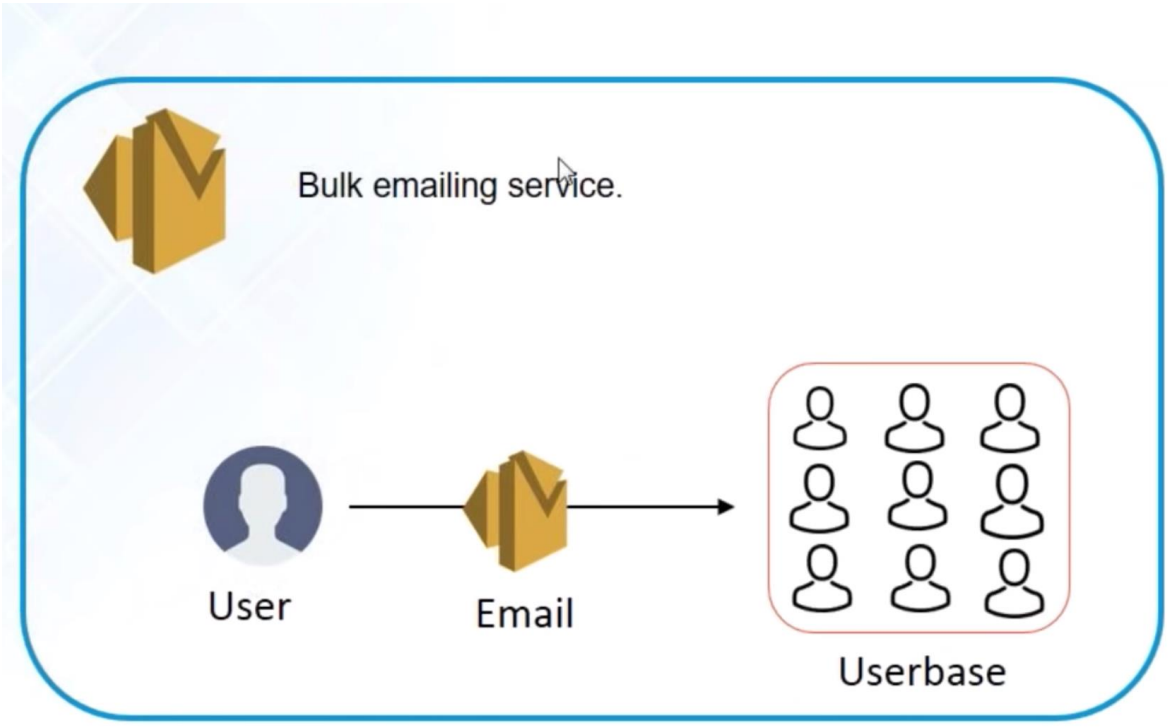
**# Application Integration**

- Step Functions
- Amazon MQ
- Simple Notification Service
- Simple Queue Service
- SWF

- **Amazon SES**

It is a cost effective emailing service which is built on the scalable and reliable infrastructure of Amazon.com

**Amazon SES**



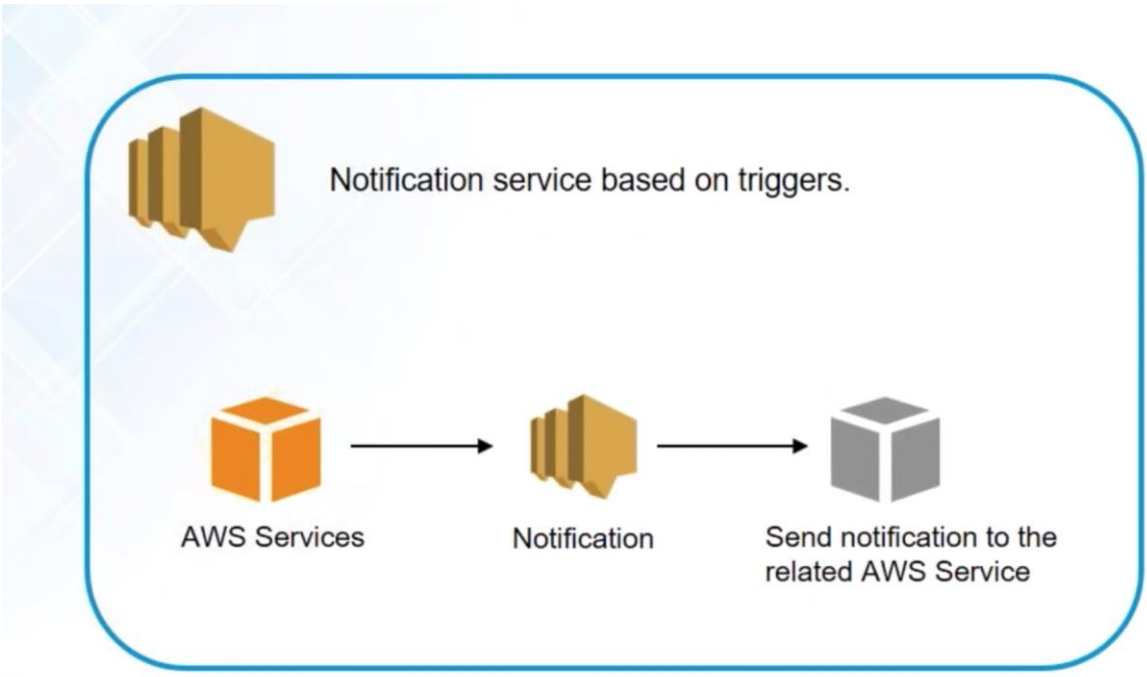

**What is Amazon SES?**

Amazon SES (Simple Email Service) is a service which sends an email regarding marketing, transaction, and notifications. It is suitable for small as well as large industries as the cost is less and it is reliable. Amazon SES can be directly integrated to the existing application with the help of SMTP Interface and Amazon SDK. Email sending capabilities can also be introduced in Amazon SES such as ticketing system and email clients.
Building a large-scale email answer is a complex and expensive challenge for a business: you've got to make your infrastructure, assemble your network, warm up your IP addresses and shield your sender name. Several third-party email solutions need contract negotiations and important up-front prices.

- **Amazon SNS**

- It is a web service offered by AWS that manages the delivery of messages to subscribed endpoints or clients.

**Simple Notification Service**

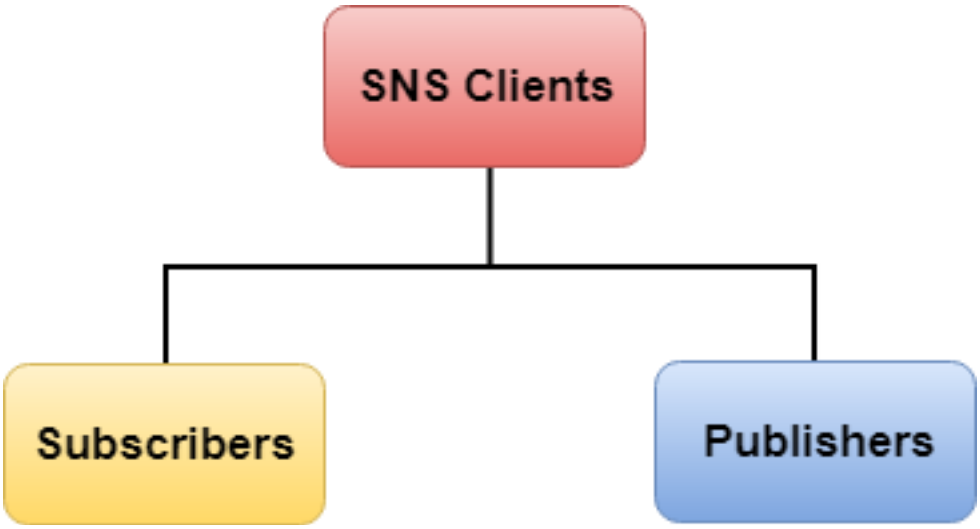



## What is SNS?

- SNS stands for Simple Notification Service.
- It is a web service which makes it easy to set up, operate, and send a notification from the cloud.
- It provides developers with the highly scalable, cost-effective, and flexible capability to publish messages from an application and sends them to other applications.
- It is a way of sending messages. When you are using AutoScaling, it triggers an SNS service which will email you that "your EC2 instance is growing".
- SNS can also send the messages to devices by sending push notifications to Apple, Google, Fire OS, and Windows devices, as well as Android devices in China with Baidu Cloud Push.
- Besides sending the push notifications to the mobile devices, Amazon SNS sends the notifications through SMS or email to an Amazon Simple Queue Service (SQS), or to an HTTP endpoint.

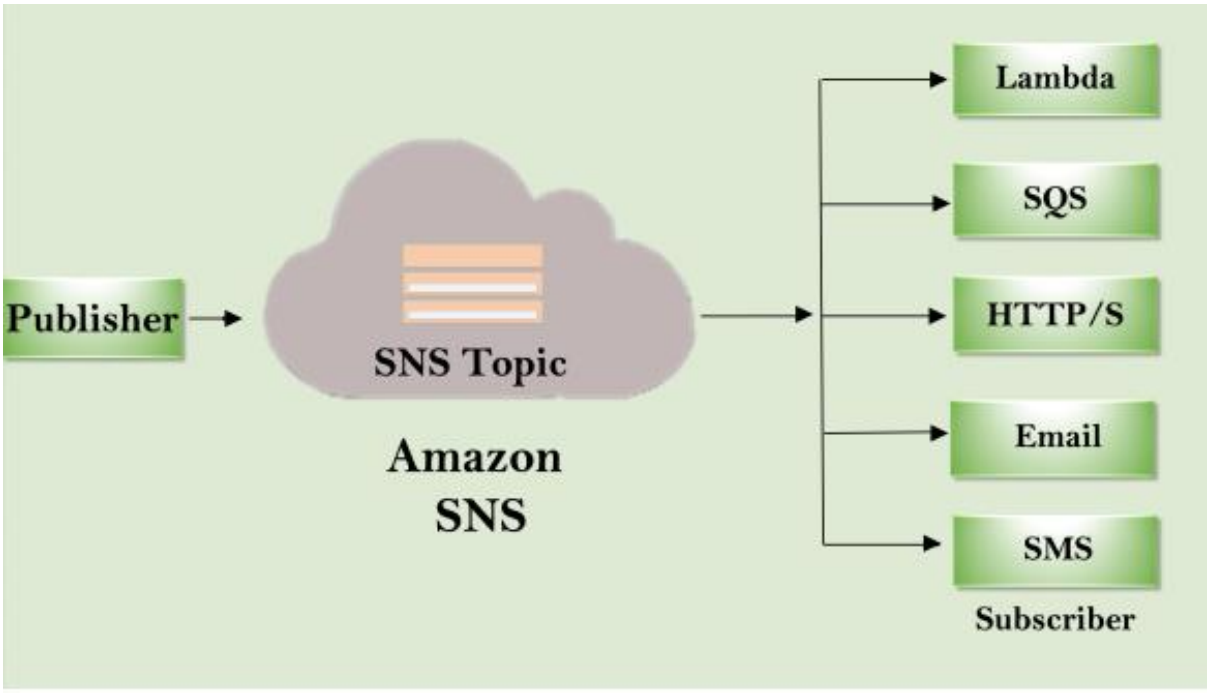
- SNS notifications can also trigger the Lambda function. When a message is published to an SNS topic that has a Lambda function associated with it, Lambda function is invoked with the payload of the message. Therefore, we can say that the Lambda function is invoked with a message payload as an input parameter and manipulate the information in the message and then sends the message to other SNS topics or other AWS services.
- Amazon SNS allows you to group multiple recipients using topics where the topic is a logical access point that sends the identical copies of the same message to the subscribe recipients.
- Amazon SNS supports multiple endpoint types. For example, you can group together IOS, Android and SMS recipients. Once you publish the message to the topic, SNS delivers the formatted copies of your message to the subscribers.
- To prevent the loss of data, all messages published to SNS are stored redundantly across multiple availability zones.

### SNS Publishers and Subscribers




Amazon SNS is a web service that manages sending messages to the subscribing endpoint. There are two clients of SNS:

- Subscribers
- Publishers




**Publishers**

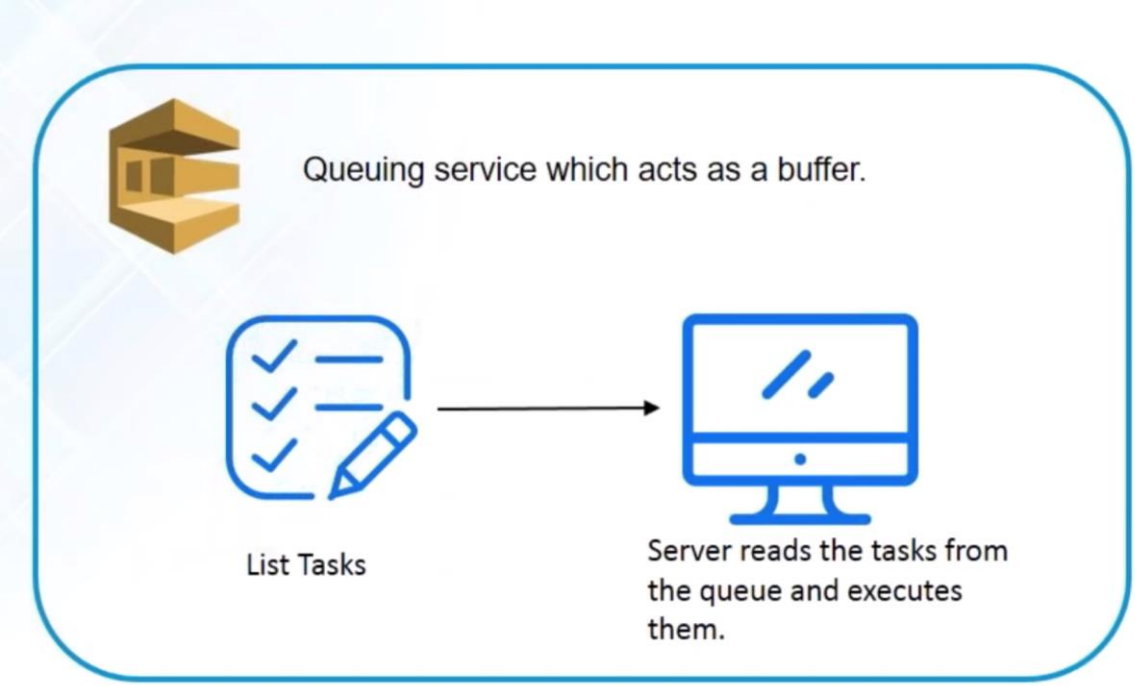
Publishers are also known as producers that produce and send the message to the SNS which is a logical access point.

**Subscribers**

Subscribers such as web servers, email addresses, Amazon SQS queues, AWS Lambda functions receive the message or notification from the SNS over one of the supported protocols (Amazon SQS, email, Lambda, HTTP, SMS).

- **Amazon SQS**

1. What is Amazon SQS (Simple Queue Service)?




1. AWS SQS (Amazon Simple Queue Service) is a service which helps to align the message. Moreover, it also helps to enable the user to separate and scale microservices, distributed system, and serverless applications.
2. Amazon SQS makes it easy to manage the operating message-oriented middleware and enhances the developers to focus on their work.
3. Amazon Simple Queue Service works at any volume without losing the message or requiring other services to be available.
4. It helps to send, store, and receive messages between software components. AWS SQS can start with the help of the tools such as Amazon Console, command line interface, and SDK.

In Amazon SQS there are two types of queues which are

- Standard Queue : Standard queue offers at least one delivery and maximum throughput

Standard Queue

- It has a benefit of supporting an ample amount of transactions per second per API action.
- As the message is delivered on at a time but at the same time, it delivers more than one copy of a message.
- It may happen that the message delivered is in the different order from the source in which they were sent.