Cloud Computing

It is an internet-based computing service in which various remote servers are networked to allow centralized data storage and online access to computer services and resources.

Types of Cloud

There are three cloud types:

- **Public cloud:** Here, the resources and services provided by third-party service providers are available to customers via the Internet.
- **Private cloud:** In a private cloud, the resources and services are managed in-house or by third parties, exclusively for a particular organization.
- **Hybrid cloud:** It is a combination of both public and private cloud types. The decision whether to run the services on public or private depends on some parameters such as the sensitivity of the data and applications, industry certifications and required standards, etc.

AWS

Amazon Web Services (AWS) is a collection of various Cloud Computing services and applications that offers flexible, reliable, easy-to-use, and cost-effective solutions.

Instance: An instance is a virtual server for running applications on AWS.

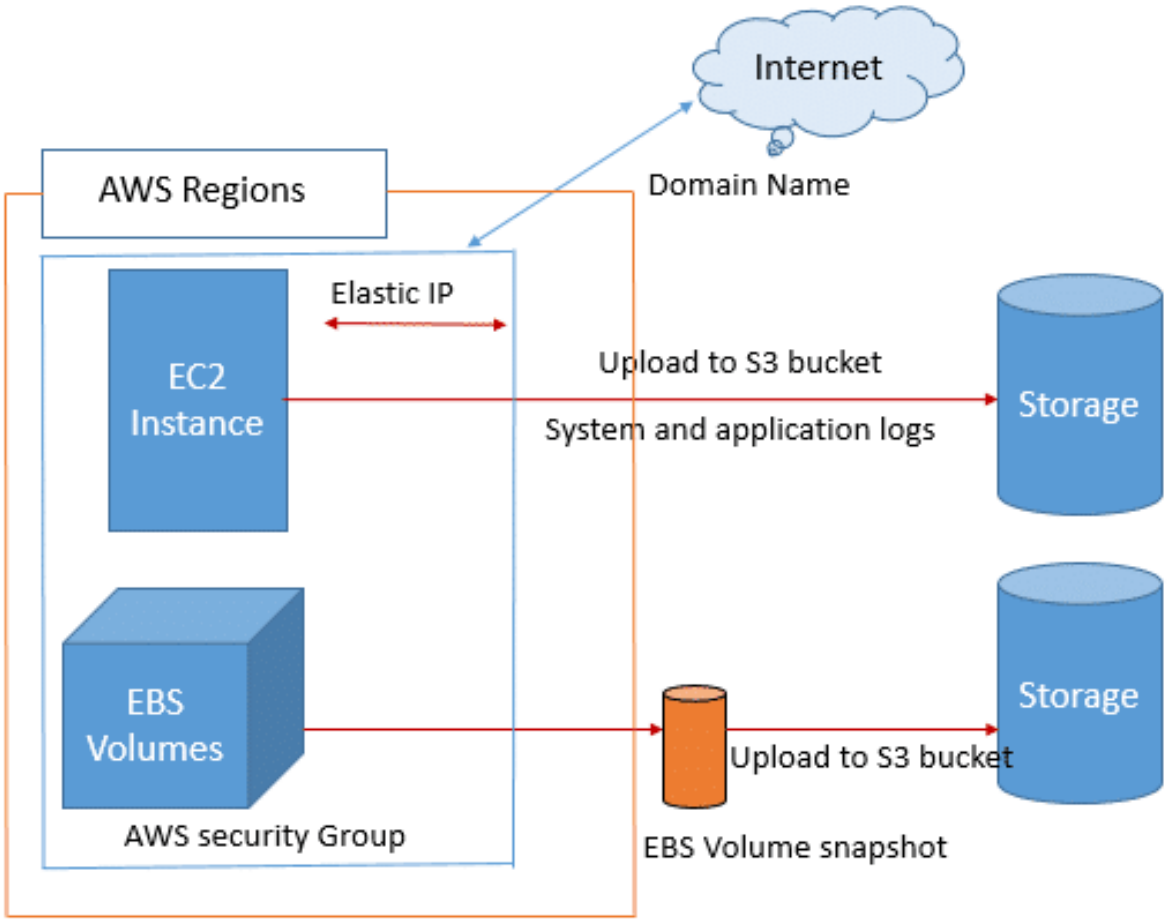
Types of AWS Services

Auto-scaling: AWS Auto Scaling is a service designed by EC2, which is used to launch or terminate EC2 instances based on user-defined policies.

Why do we need autoscaling in the first place? A limited number of servers to cope with the application load can result in various failures and latency issues when the number of requests rises. Increase in the number of requests is inevitable within a growing business and with the increase in the number of requests, the load on servers also increases.

Amazon Web Services offer an AWS **auto scaling service** to deal with these issues. Autoscaling makes it possible to make resources highly available which in turn eliminates various failure issues related to the increased load on limited servers.

Elastic Load Balancing: It automatically distributes the incoming application traffic across multiple instances at multiple availability zones.



Networking Services

- **VPC:** Amazon Virtual Private Cloud (VPC) is a virtual data center in AWS consisting of a set of isolated resources.
- **Direct Connect:** It is used to establish a dedicated network connection from the host network to AWS without an Internet connection.
- **Route 53:** It is a scalable and highly available Domain Name System (DNS) and domain name registration service, and 53 is the port on which this service runs.

Computing Services

- **EC2:** It is a virtual server that provides resizable compute capacity on the cloud.
- **Elastic Beanstalk:** It is an application container used for deploying and managing containers. It creates an environment for working with web applications.
- **Lambda:** It is a computing service that runs the code in response to events and automatically manages the computing resources.
- **EC2 Container Service:** It allows us to easily run and manage Docker containers across a cluster of EC2 instances.

Storage Services

- **S3:** It refers to Simple Storage Service and allows the storage of data objects of any sort and flat files in the cloud. It is secure, scalable, and durable.
- **CloudFront:** **CloudFront** defines a Content Delivery Network. It provides a way to distribute content to end-users with low latency and high data-transfer speeds.
- **Glacier:** It is a low-cost storage service that provides secure and durable storage for long-term data archiving and backup.

- **EFS (Elastic File Storage):** It is a file storage service used in EC2 instances and connects to multiple EC2 instances.
- **Snowball:** It is used for moving large amounts of data into/out of AWS using secure appliances, i.e., it provides the data archiving functionality for the data that no longer needs to be accessed actively.
- **Storage Gateway:** AWS Storage Gateway is used for securely integrating on-premises IT environments with cloud storage for backup and disaster recovery.
- **RDS (Relational Database Service)**: It allows the storage of data objects as part of the relational database. It makes it easy to set up, operate, and scale familiar relational databases in the cloud.
- **DynamoDB:** It is a scalable NoSQL data store that is used to manage distributed replicas of data for high availability.
- **ElastiCache:** It improves application performance by allowing us to retrieve information from an in-memory caching system. It is a way of caching databases in the cloud.
- **Redshift:** It is a fast, fully managed data warehousing service, which makes it cost-effective to analyze all data using the existing Business Intelligence tools.
- **DMS (Data Migration Service)**: It helps in migrating databases to the cloud easily and securely. It can also be used for converting databases.

Analytics

- **Amazon EMR:** Amazon Elastic MapReduce helps in performing big data tasks such as web indexing, data mining, and log file analysis.
- **Data Pipeline:** It helps in moving data from one service to another. It is a service used for periodic, data-driven workflows.
- **AWS Elasticsearch:** It is a managed service that helps in deploying, operating, and scaling Elasticsearch.
- **Kinesis:** It makes it easy to work with real-time streaming data in the AWS cloud.
- **AWS Machine Learning:** It is a service that enables us to easily build smart applications.

- **QuickSight:** AWS QuickSight is a cloud-assisted Business Intelligence service that helps in deriving insights from data easily.

Security and Identity

- **IAM:** AWS IAM helps in configuring security for all the services. It is used to ensure that our other services remain safe and inaccessible to others.
- **Directory Service:** It is used to provide a managed directory in the cloud.
- **Inspector:** It enables us to analyze the behaviour of the applications we run on AWS and helps in identifying potential security issues.
- **AWS WAF (Web Application Firewall):** It protects our web application from attacks by providing web traffic filters.
- **Cloud HSM:** It is a Hardware Security Module.
- **KMS:** It is a Key Management Service.

Management Tools

- **CloudWatch:** It is used to create different metrics. It provides monitoring for resources and applications.
- **CloudFormation:** It helps in creating and updating a collection of related AWS resources.
- **CloudTrial:** It provides increased visibility into user activity by recording API calls made on an account.
- **AWS OpsWorks:** It is a DevOps platform for managing applications of any size or complexity on the AWS cloud.
- **Config:** It gives an inventory of AWS resources, lets us audit the AWS resource configuration history, and notifies the changes.
- **Service Catalog:** It allows organizations to manage approved catalogs of IT resources.
- **Trusted Advisor:** It inspects the AWS environment and finds opportunities to save money and improve system performance.

Application Services

- **API Gateway:** AWS API Gateway is used to create, maintain, monitor, and secure APIs.
- **AppStream:** It is used to stream resource-intensive applications and games from the cloud to multiple users.
- **CloudSearch:** It is a completely managed search service for websites and apps.
- **Elastic Transcoder:** It is used to convert media files in the cloud easily at a lower cost.
- **SES (Simple Email Service):** It is used to send and receive emails.
- **SQS (Simple Queue Service):** It is a reliable, hosted queue for storing messages.
- **SWF (Simple Workflow Service):** It is used to coordinate all the processing steps with an application.

Developer Tools

- **AWS Code commit:** It is a managed source-control service that hosts private Git repositories.
- **AWS Code deploy:** It is used to automate the code deployment.
- **AWS Code pipeline:** It is a continuous delivery service that enables us to visualize and automate the steps required to release software.
- **AWS Amplify:** The AWS Amplify tool includes UI components, a command line interface, and a set of libraries to integrate your backend in any mobile or web app.

Mobile Services

- **Mobile Hub:** It helps in building, testing, and monitoring the usage of mobile apps.

- **Cognito:** AWS Cognito is a simple user-identity and data-synchronization service that helps in securely managing and synchronizing the app data for users across their mobile devices.
- **Device Farm:** It helps in improving the quality of Android, Fire OS, and iOS apps by testing them against real phones and tablets on the cloud.
- **Mobile Analytics:** It is a service that is used to easily collect, visualize, and understand app usage.
- **SNS (Simple Notification Service):** It helps in publishing messages to subscribers or to other applications.

Enterprise Apps

- **WorkSpace:** It is a fully managed desktop computing service on the cloud.
- **WorkDocs:** It is a storage and sharing service with strong administrative controls and feedback capabilities that improve user productivity.
- **WorkMail**: It is an email and calendaring service that offers strong security controls and support for the existing desktop and mobile clients.

Types of EC2 Computing Instances

Following are the AWS ec2 instance types:

- **General Instances:** Used for applications that require a balance of performance and cost
- **Compute Instances:** Used for applications that require a lot of processing from the CPU
- **Memory Instances:** Used for applications that need a lot of RAM
- **Storage Instances:** Used for applications with datasets that occupy a lot of space
- **GPU Instances:** Used for applications requiring heavy graphics rendering

Basic CLI Commands

- **cat /proc/mounts**: To display a list of mounted drives
- **rm <*filename*>**: To remove the specified file from the current directory
- **rpm – ql‘<*package name*>‘**: To obtain a list of utilities contained within a package
- **sudo chmod <*options*>**: To change the access mode for the current directory
- **sudo mkdir <*directory name*>**: To create a new directory to hold files
- **sudo reboot**: To reboot the removed AWS system so that we can see the results of any changes we make
- **sudo rmdir <*directory name*>**: to remove the specified directory
- **sudo yum groupinstall ‘‘<*group package name*> ‘‘**: To install the specified group of packages
- **sudo yum search ‘<*package name*> ‘**: To search for a package
- **sudo yum update**: To perform the required AWS updates
- **sudo yum -y install <*service or feature*>**: To install a required support service or feature onto the AWS system

What is VPC in AWS?

Among all services that AWS offers, Amazon VPC is one that provides an additional layer of security for all AWS services that you use. AWS defines VPC as ‘a service that enables users to launch AWS resources, such as instances, into a virtual network that users define.’ This basically means that this service lets you use any of the services by AWS according to your needs in a logically isolated space in the AWS Cloud that you define. It also gives you full control over routing traffic to and from your instances.

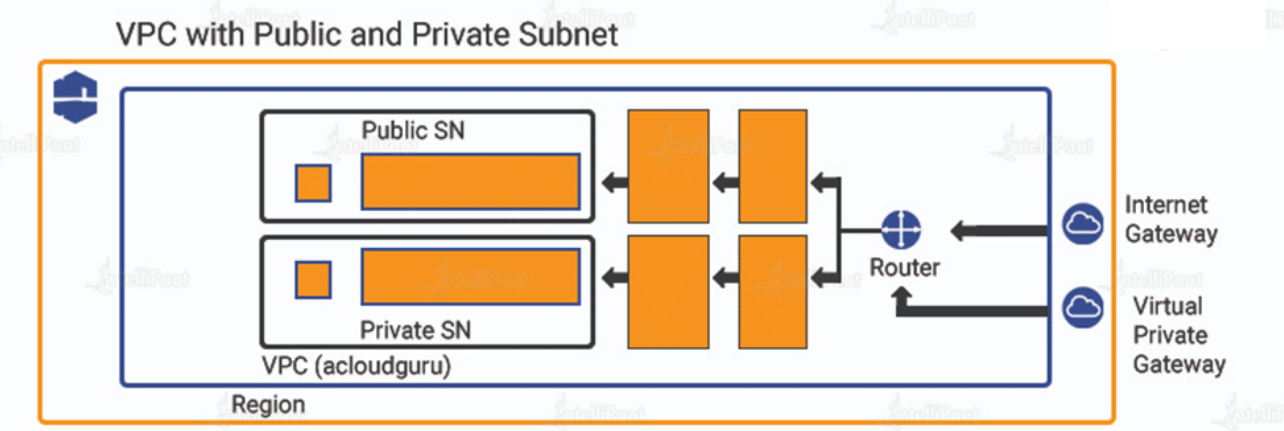
There are two types of VPCs in AWS, namely:

- **Default VPC:** Every account created after 2013 supports VPCs and all these accounts come up with a default VPC in every region.
- **User-defined VPC:** User-defined VPCs, as suggested by the name, are created by users as per their requirements.

AWS VPC Peering

AWS VPC Peering is a functionality that enables two private networks to communicate with each other by building fast and reliable connections. AWS VPC peering connections can be used to route traffic from one VPC to another VPC network or to provide access to resources of one network to another. Each and every AWS account comes with a default VPC in every region that it supports. Peering actually allows traffic between two VPCs based on a specific resource's network address. However, transitive peering is not supported in AWS VPC Peering. Transitive peering simply means that VPC-A can communicate with resources in VPC-C via VPC-B, just because VPC-B is connected to VPC-C. This type of networking and communication arrangement is not supported by AWS VPC Peering. Now, having talked about the meaning of AWS VPC, let us dig further into the benefits of the same.

AWS VPC Architecture



1- Subnets

A subnet is a subdivision of a network. When a network is broken down into smaller sub networks or subnets, that process is called subnetting. Now, we will talk about public and private subnets.

Public subnets: They are typically used in cases where the resources must be connected to the internet, for example, web servers. The main route table sends the subnet traffic to the internet gateway where the traffic is meant for the internet. Hence, this type of subnet is referred to as a public subnet.

Private subnets: On the contrary, private subnets are used for resources that do not need an internet connection.

Subnet sizing: Usually, it is found that private subnets have double the number of instances as compared to public subnets. Now, the sizing of CIDR blocks used in subnets is based on this typical deployment. However, the subnet resizing can be done during deployment by using CIDR block parameters as per architectural requirements.

2- Route Table

As mentioned earlier, VPC in AWS provides full control over the traffic. To do that, you have route table. A route table consists of rules that are used to determine how and to where the traffic will be directed in a network.

Every subnet in Amazon Virtual Private Cloud should be associated with a route table that will control the routing for their respective subnet. A route table can be associated with multiple subnets in a network.

3- Internet Gateway

An Internet gateway is what allows your instance, launched in a subnet in your VPC, to connect to the internet. It lets the instance access the internet, and the internet and other resources, outside of the VPC, access the instance. Internet gateway is one of the most important components of VPC.

4- VPC Endpoints

VPC endpoints are used when you need to create a private network between your VPC and another AWS Service outside your VPC without relying on the internet, VPN, or NAT devices. Once an endpoint is created, it cannot be transferred from one VPC to another or any other service.

Endpoints are also only supported within the same region. If they are not in the same region, endpoints cannot be used to connect service and VPC.

AWS VPC peering benefits

There are multiple benefits of AWS VPC peering. It can either be easy deployment of cloud resources or ease of transferring data across resources. However, the most important benefits are security of the private networks, easy set-up and application performance. These are explained below:

Security

The first and foremost benefit of VPC is security. VPC in AWS provides advanced security at the instance level and at the subnet level. With VPC, you can specify the users who are allowed to access cloud resources and who are not.

Easy to Set-up and Use

AWS VPC is as easy to set-up as any other services offered by AWS. Using the AWS Management Console, you can easily set-up Amazon VPC. As for the default VPC for your account, it is pre-configured, which lets you focus on building and deploying apps.

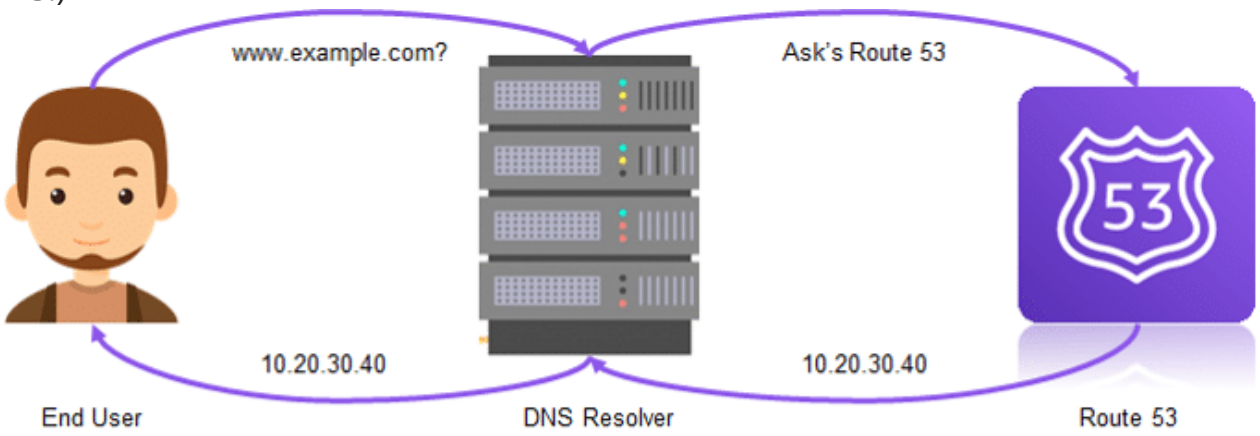
Application Performance

Application performance is largely affected by the congestion in the Internet. It can slow down the application traffic or even make the application slow. With VPC, the

probability of application performance going down decreases as it helps in delivering the traffic with high priority, accordingly.

What Is Amazon Route 53?

Route 53 is a web service that is a highly available and scalable Domain Name System (DNS.)



Let's understand what is Amazon Route 53 in technical terms. AWS Route 53 lets developers and organizations route end users to their web applications in a very reliable and cost-effective manner. It is a Domain Name System (DNS) that translates domain names into IP addresses to direct traffic to your website. In simple terms, it converts World Wide Web addresses like www.example.com to IP addresses like 10.20.30.40.

Basically, domain queries are automatically routed to the nearest DNS server to provide the quickest response possible. If you use a web hosting company like GoDaddy, it takes 30 minutes to 24 hours to remap a domain to a different IP, but by using Route 53 in AWS it takes only a few minutes.

How Amazon Route 53 works?

AWS Route 53 connects requests to the infrastructure running in AWS. These requests include AWS ELB, Amazon EC2 instances, or Amazon S3 buckets. In addition to this, AWS Route 53 is also used to route users to infrastructure outside of AWS.

AWS Route 53 can be easily used to configure DNS health checks, continuously monitor your applications' ability to recover from failures, and control application recovery with Route 53 Application Recovery Controller. Further, AWS Route 53 traffic flow helps to manage traffic globally via a wide variety of routing types including latency-based routing, geo DNS, weighted round-robin, and geo proximity. All these routing types can be easily combined with DNS Failover in order to enable a variety of low-latency, fault-tolerant architectures.

Let us understand, step by step, how does AWS Route 53 work:

- A user accesses www.example.com, an address managed by Route 53, which leads to a machine on AWS.
- The request for www.example.com is routed to the user's DNS resolver, typically managed by the ISP or local network, and is forwarded to a DNS root server.
- The DNS resolver forwards the request to the TLD name servers for ".com" domains.
- The resolver obtains the authoritative name server for the domain—these will be four Amazon Route 53 name servers that host the domain's DNS zone.
- The DNS resolver chooses one of the four Route 53 servers and requests details for the hostname.
- The Route 53 name server looks in the DNS zone for www.example.com, gets the IP address and other relevant information, and returns it to the DNS resolver.
- The DNS resolver returns the IP address to the user's web browser. The DNS resolver also caches the IP address locally as specified by the Time to Live (TTL) parameter.
- The browser contacts the webserver or other Amazon-hosted services by using the IP address provided by the resolver.

- The website is displayed on the user's web browser.

Now, take a look at the benefits provided by Route 53.

Amazon Route 53 Benefits

Route 53 provides the user with several benefits.

They are:

- Highly Available and Reliable
- Flexible
- Simple
- Fast
- Cost-effective
- Designed to Integrate with Other AWS Services
- Secure
- Scalable

Highly Available and Reliable

- AWS Route 53 is built using AWS's highly available and reliable infrastructure. DNS servers are distributed across many availability zones, which helps in routing end users to your website consistently.
- Amazon Route 53 Traffic Flow service helps improve reliability with easy re-route configuration when the system fails.

Flexible

- Route 53 Traffic Flow provides users flexibility in choosing traffic policies based on multiple criteria, such as endpoint health, geographic location, and latency.

Simple

- Your DNS queries are answered by Route 53 in AWS within minutes of your setup, and it is a self-service sign-up.
- Also, you can use the simple AWS Route 53 API and embed it in your web application too.

Fast

- Distributed Route 53 DNS servers around the world make a low-latency service. Because they route users to the nearest DNS server available.

Cost-effective

- You only pay for what you use, for example, the hosted zones managing your domains, the number of queries that are answered per domain, etc.
- Also, optional features like traffic policies and health checks are available at a very low cost.

Designed to Integrate with Other AWS Services

- Route 53 works very well with other services like **Amazon EC2** and **Amazon S3**.
- For example, you can use Route 53 to map your domain names or IP addresses to your EC2 instances and Amazon S3 buckets.

Secure

- You can create and grant unique credentials and permissions to each and every user with your AWS account, while you have to mention who has access to which parts of the service.

Scalable

- Amazon Route 53 is designed to automatically scale up or down when the query volume size varies.

These are the benefits that Amazon Route 53 provides, moving on with this what is Amazon Route 53 tutorial, let's discuss the AWS routing policies.

Amazon Route 53 Limitations

Amazon Route 53 is a robust DNS service with advanced features, but it has several limitations as well. Some of them are discussed below:

- **No DNSSEC support:** DNSSEC stands for Domain Name System Security Extensions. It is a suite of extensions specifications by the Internet Engineering Task Force. It is used to secure the data exchanged in DNS in Internet Protocol networks. It is not supported by AWS Route 53.
- **Forwarding options:** Route 53 does not provide forwarding or conditional forwarding options for domains used on an on-premise network.
- **Single point of failure:** Used in conjunction with other AWS services, Route 53 may become a single point of failure. This becomes a major problem for AWS route 53 disaster recovery and other relevant issues.
- **Limited Route 53 DNS load balancing:** The features of AWS Route 53 load balancer lack advanced policy support and enterprise-class features and provide only basic load balancing capabilities.
- **Route 53 Cost:** For businesses using Route 53 with non-AWS endpoints or services, the service is expensive. In particular, the visual editor is costly including the cost of each query.
- **No support for private zone transfers:** AWS Route 53 DNS cannot be appointed as the authoritative source for cloud websites.com, even after having the root-level domain registered.
- **Latency:** All AWS Route 53 queries must be forwarded to external servers after contacting Amazon infrastructure.

AWS Route 53 Alternatives

When buying a solution, buyers often compare and evaluate similar products by different market players based on certain parameters such as specific product

capabilities, integration, contracting, ease of deployment, and offered support and services. Based on the mentioned parameters and a few more, we have listed some potential AWS Route 53 alternatives below:

- **Azure DNS:** It allows you to host your DNS domain in Azure. This helps to manage DNS records by using the same credentials, billing, and support contract just as other Azure services.
- **Cloudflare DNS:** As a potential alternative to AWS Route 53, Cloudflare DNS is described as the fastest, privacy-first consumer DNS service. It is a free-of-charge service for ordinary people; however, professionals and enterprises have to take up a monthly subscription.
- **Google Cloud DNS:** Google Cloud DNS is a scalable, reliable, and managed authoritative DNS service that runs on the same infrastructure as Google.
- **DNSMadeEasy:** It offers affordable DNS management services that are easy to manage. It also has the highest uptime and amazing ROI.
- **DNSimple:** With DNSimple, you can register a domain quickly with no upselling and hassles.

Does Avi Offer Route 53 Monitoring Capabilities?

Avi Vantage is a next-generation, full-featured elastic application services fabric that offers a range of application services such as security, monitoring and analytics, load balancing, and multi-cloud traffic management for workloads. All workloads are deployed in bare metal, virtualized, or container environments in a data center of a public cloud such as AWS. Avi Vantage delivers full-featured load balancing capabilities in an as-a-service experience and easily integrated Web Application Firewall (WAF) capabilities.

Enterprises often leverage the power of AWS in order to maximize and modernize infrastructure utilization. The next phase of this modernization is represented by extending app-centricity to the networking stack.

Avi Networks integrates with AWS Route 53 and delivers elastic application services that extend beyond load balancing to deliver real-time app and security insights, simplify troubleshooting, enable developer self-service, and automation.

Amazon Route 53 Resolver for Hybrid Cloud

The user merges a private center with one of their Amazon VPCs using a managed VPN or AWS Direct Connect in a typical hybrid cloud environment. As the private cloud and the user's VPC is a pre-established connection to AWS, whenever a lookup is performed across this connection, it often fails. As a result, some users reroute requests using on-premises DNS servers to another Amazon VPC server. It can perform outbound communication from VPC to the data center and inbound communication from an on-premises source to VPC.

Some of the advantages of AWS Route 53 resolver are as follows:

Security: AWS benefits from the added security of **Identity Access Management (IAM)**. AWS IAM allows secure user control access to all web resources and services. It can also assign specific permissions to allow or deny access to AWS resources and the creation and management of AWS users or groups.

Cost: AWS Route 53 proves to be really cost-effective as it redirects website requests without extra hardware and does not charge for queries to **CloudFront** distributions, ELBs, S3 buckets, VPC endpoints, and other AWS resources.

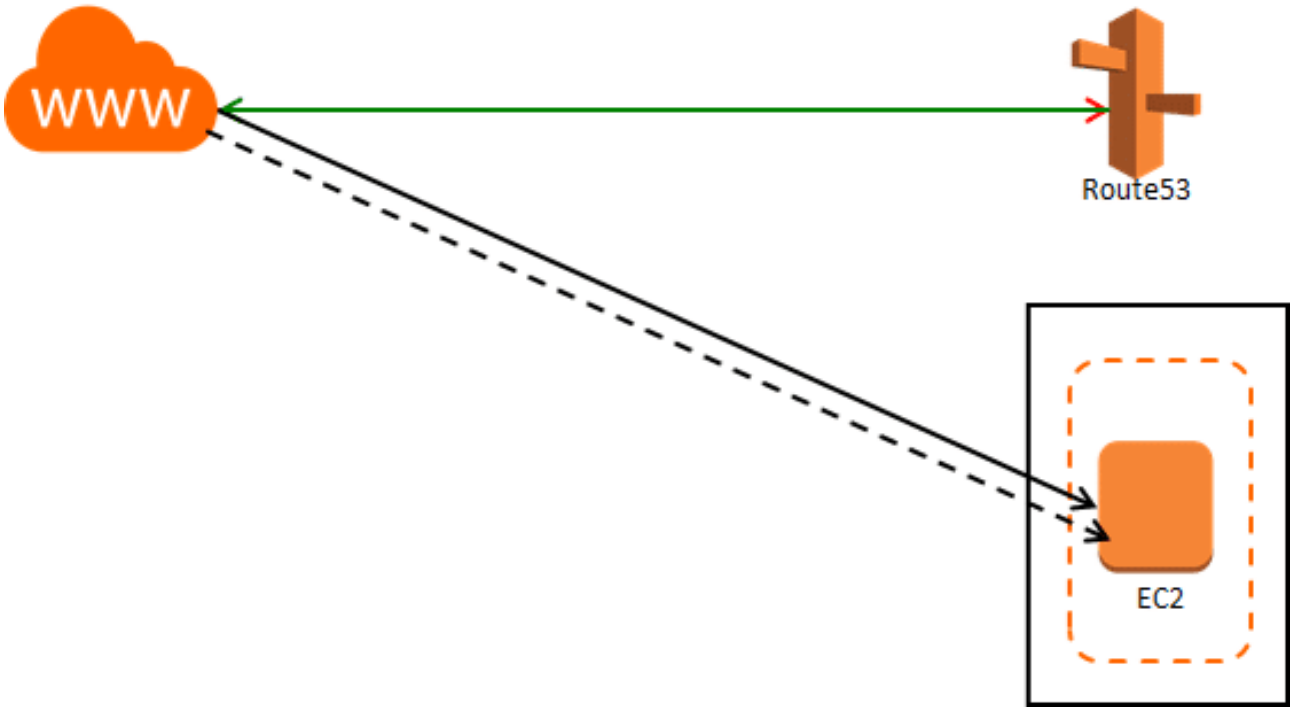
Reliability: All features of Route 53, such as geographically-based and latency-based policies, are designed to be highly reliable and cost-effective. In addition to this, Amazon Route 53 is designed to help the system stay running in a coordinated way with all the other AWS services.

AWS Routing Policies

There are several types of routing policies. The below list provides the routing policies which are used by AWS Route 53.

- Simple Routing
- Latency-based Routing
- Geolocation Routing

Simple Routing

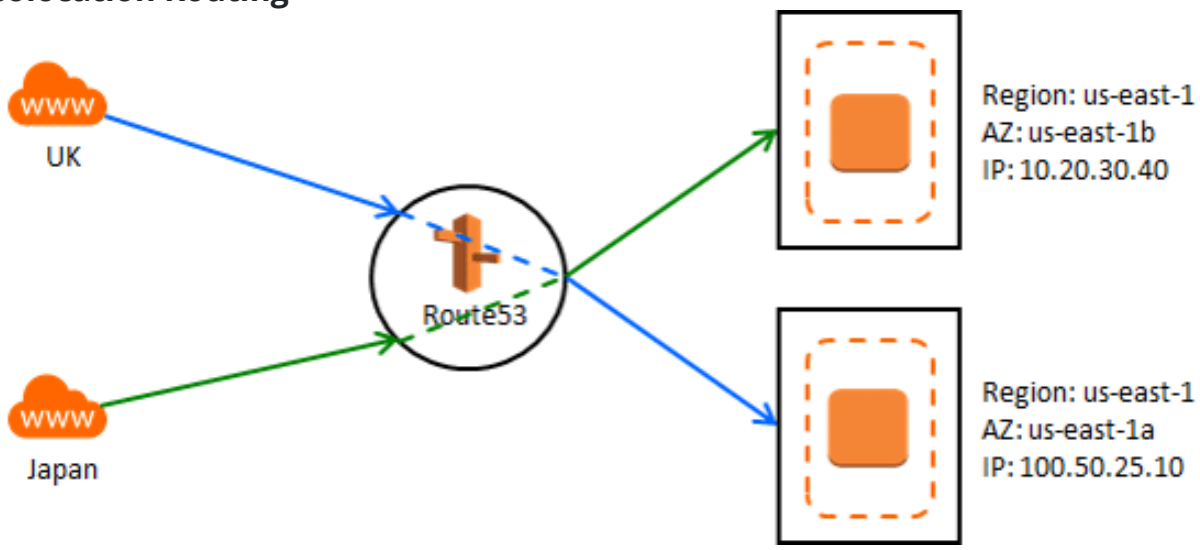


Simple routing responds to DNS queries based only on the values in AWS route table. Use the simple routing policy when you have a single resource that performs a given function for your domain.

Latency-based Routing

If an application is hosted on EC2 instances in multiple regions, user latency can be reduced by serving requests from the region where network latency is the lowest. Create a latency resource record set for the Amazon EC2 resource in each region that hosts the application. Latency will sometimes change when there are changes in the routes.

Geolocation Routing



Geolocation routing can be used to send traffic to resources based on the geographical location of users, e.g., all queries from Europe can be routed to the IP address 10.20.30.40. Geolocation works by mapping IP addresses, irrespective of regions, to locations.

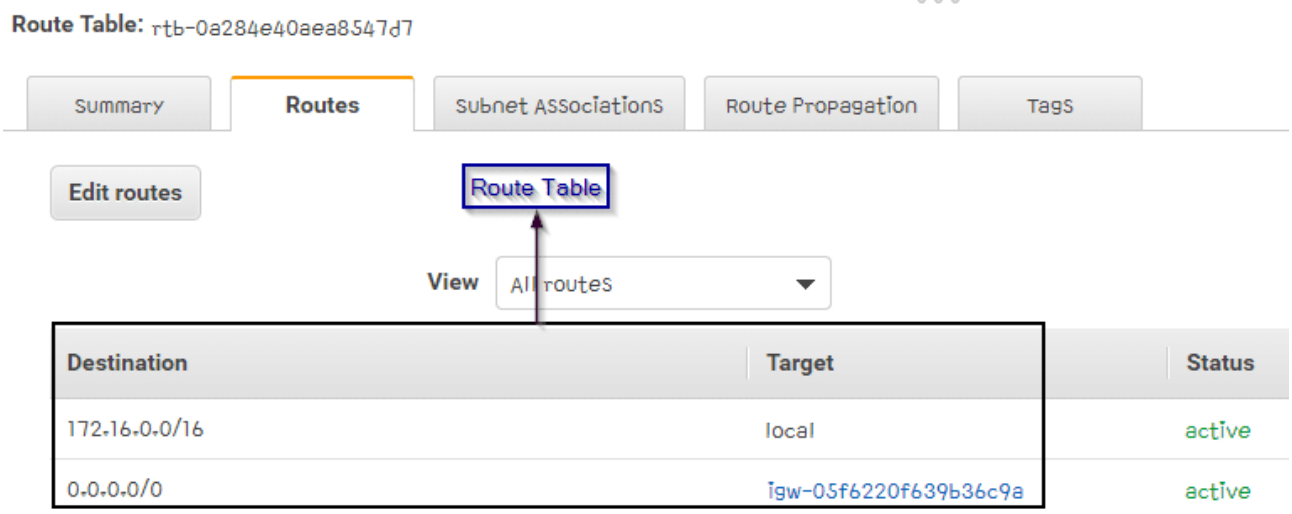
Now, you understood that Route 53 in AWS maps the end user to an IP address or a domain name. But, where are the routes stored?

AWS Route Tables

An AWS route table contains a set of rules or routes, which is used to determine where the network traffic is directed to.

All subnets in your **VPC** have to be attached to an AWS route table, and the table will take control of routing for those particular subnets. A subnet cannot be associated with multiple route tables at the same time, but multiple subnets can be connected

with a single AWS route table. An AWS route table consists of the destination IP address and the target.



These are the benefits provided by Route 53. What key features make Route 53 special?

AWS Route 53 Key Features

- **Traffic Flow**

You can route end users to the best endpoint possible according to your application's geo proximity, latency, health, and other considerations.

- **Latency-based Routing**

You can route end users to the AWS region with the lowest possible latency.

- **Geo DNS**

You can route your end users to the endpoint which is present in their specific region or the nearest geographic location.

- **DNS Failover**

You can route your end users to an alternate location to avoid website crashes or outages.

- **Health Checks and Monitoring**

The Health and performance of your website or application is monitored by Amazon Route 53. Your servers can be monitored as well.

- **Domain Registration**

You can search for and register available domain names using Amazon Route 53. A full list of currently available Top-level Domains (TLDs) are provided with the current pricing.

What is Amazon AWS Elastic Load Balancer?

Load balancer is a service which uniformly distributes network traffic and workloads across multiple servers or cluster of servers. Load balancer in AWS increases the availability and fault tolerance of an application. AWS Elastic Load Balancer is the single point of contact to all the clients, they can be sent to the nearest geographic instance or the instance with the lowest latency.

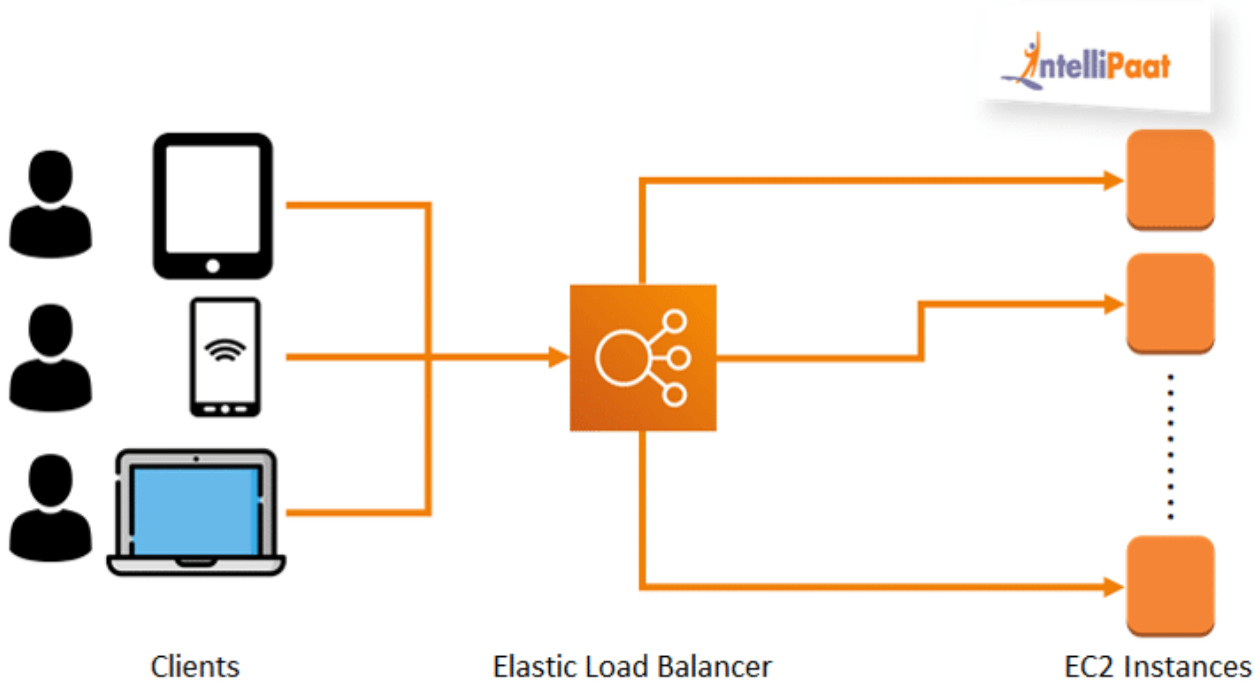
AWS Load balancer will distribute your workloads across multiple compute resources, such as a Virtual Machine or Virtual Server. The applications availability and fail over will decrease due to this. You can also let your load balancer take care of your encryption and decryption and lets you compute services do their main work.

But how to access a Amazon ELB?

There are multiple ways to that,

- **AWS Management Console** – Using the AWS web interface you can create load balancers
- **AWS Command Line Interface –** AWS provides a command line interface which is compatible in Mac, Windows, and Linux
- **AWS SDKs –** Language specific APIs are provided and can be used for any function using the load balancer or other services.
- **Query API –** This is the most direct way to call load balancers in AWS, but you must only use low-level API actions like sending HTTPs requests.

How does an AWS ELB Work?



The working of the AWS Elastic Load Balancer is very simple. The illustration above sums it all up.

The basic working principle is the Elastic Load Balancer accepts incoming traffic from its clients and then routes requests to the targets which the client want. If the load balancer finds an unhealthy target, then it will stop redirecting it's users there and will move with the other healthy targets until that target is declared healthy.

To make an AWS ELB accept incoming traffic you have to configure them by specifying one or more listeners. A listener is a process which will check for connection requests.

Availability zones – You can enable Availability Zones for your Amazon load balancer, then a load balancer node will be created in that Availability Zone. Enabling multiple Availability Zones and also make sure they at least have one registered target. Having at least one registered target will allow the load balancer to route traffic there. The advantage of having multiple AZs and targets will allow AWS load balancer to route traffic to other targets when few targets fail.

Now, let us have a look at different load balancers in AWS and their functionalities.

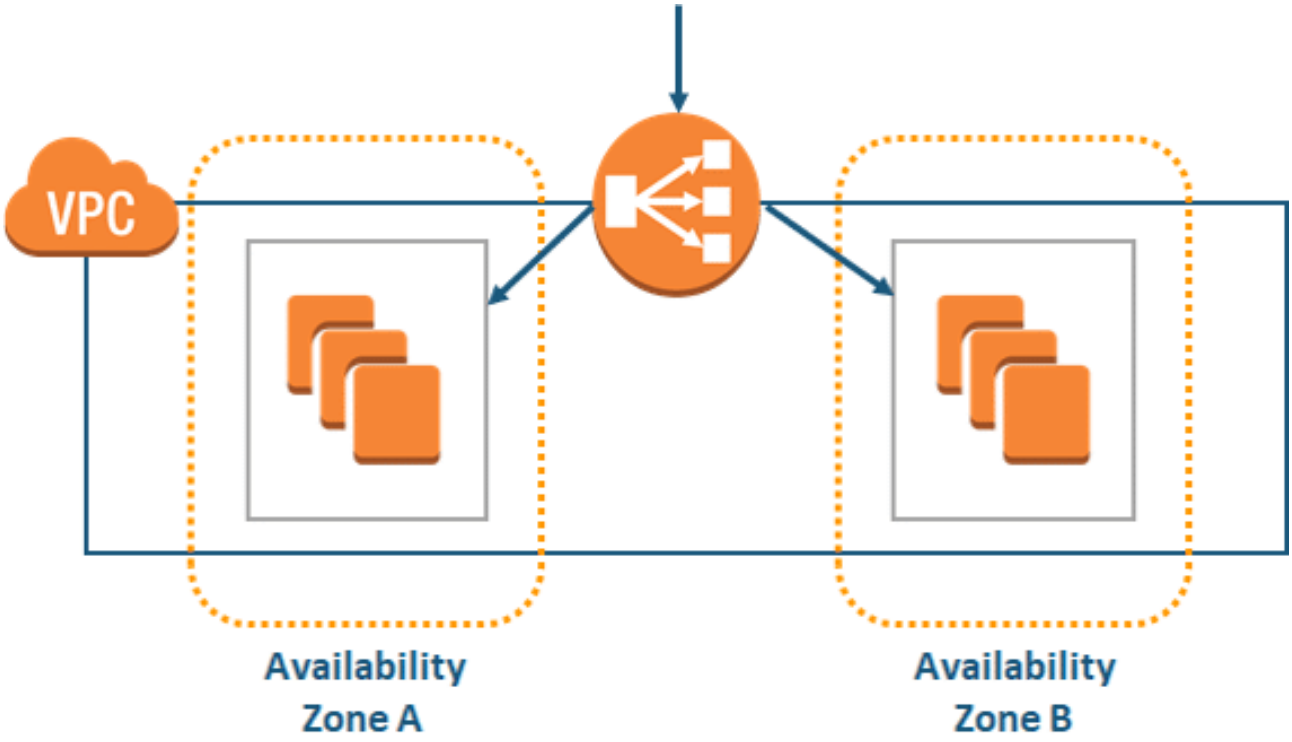
Types of AWS Elastic Load Balancers

There are mainly three types of Amazon load balancers:

- Classic Load Balancer
- Network Load Balancer
- Application Load Balancer

Classic Load Balancer

Classic Load balancer in AWS is used on EC2-classic instances. This is the previous generation's load balancer and also it doesn't allow host-based or path based routing.



The Classic Load balancer will route traffic to all registered targets in the Availability Zones, it doesn't check what is in the servers in those targets. It routes to every single target. Mostly it is used to route traffic to one single URL.

Routing decisions can be taken in transport layer (TCP/SSL) or the application layer (HTTP/HTTPS). Currently, the Classic Load Balancers require a fixed connection between the load balancer port and container instance port.

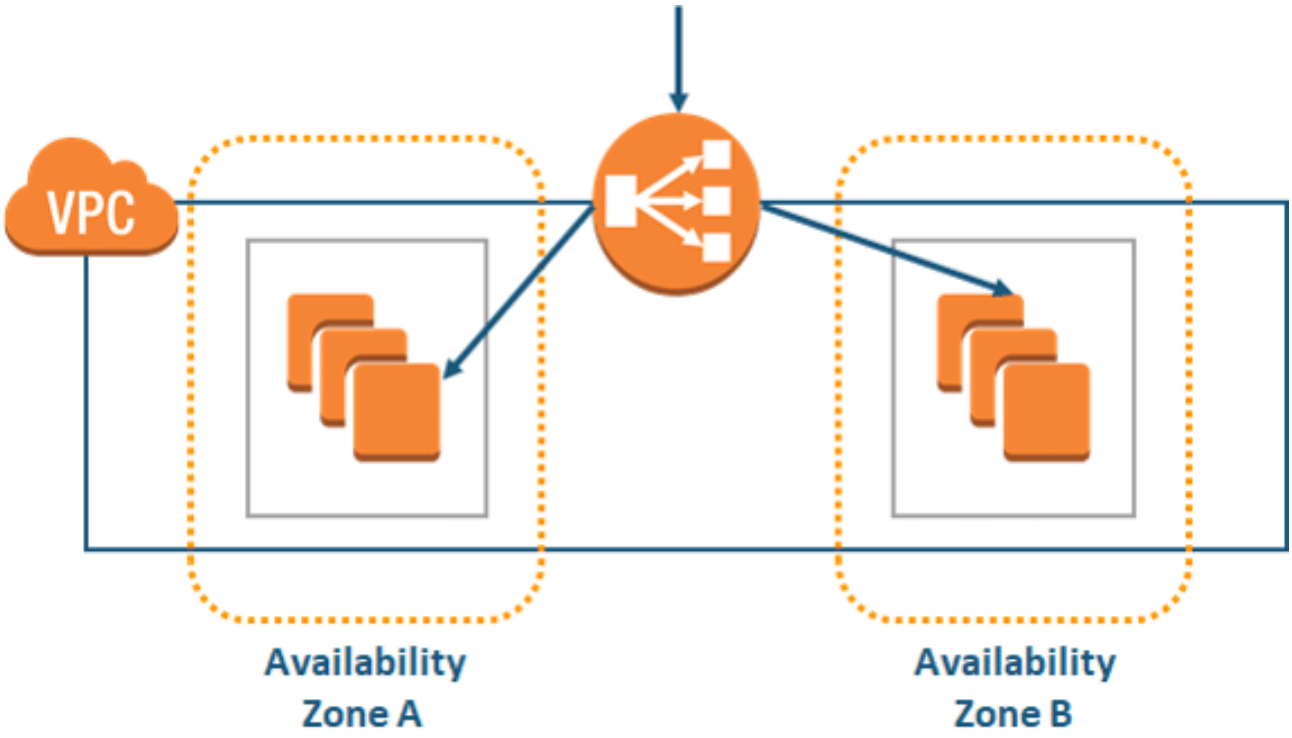
Network Load Balancer

Network Load Balancer in AWS takes routing decisions in the Transport layer (TCP/SSL) of the OSI model, it can handle millions of requests per second. Widely used to load balancing the TCP traffic and it will also support elastic or static IP.

Let us see a simple example, you own a video sharing website which has decent traffic every day. One day, after a video on your website, went viral the website's traffic is

very high and you need an immediate solution to maintain it. AWS Network Load Balancer to the rescue!

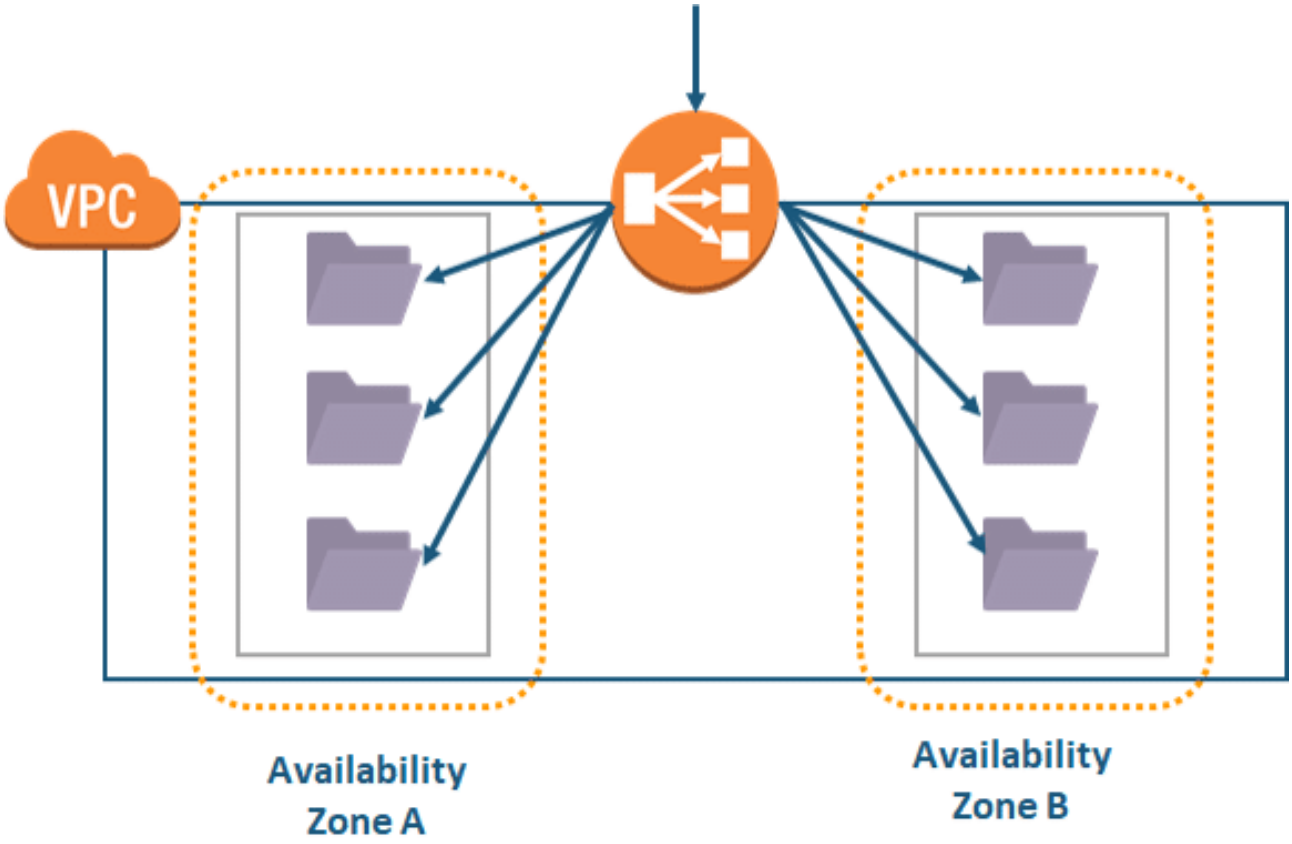
AWS Network Load Balancer can be trusted in these types of situations. It can handle millions of requests and a sudden spike of traffic because it works at the connection level.



Application Load Balancer

An Application Load Balancer in AWS makes routing decisions at the application layer (HTTP/HTTPs) of the OSI model, thus the name Application Load Balancer. ALB supports path-based and host-based routing, we will look at them after learning how the ALB works.

The Application Load Balancer receives the route requests, then it inspects the received packets. Then it chooses the best target possible for the type of load and sends to the target with the highest efficiency.



Host-based Routing using ALB

If you have two websites, intellipaat.com and dashboard.intellipaat.com. Both of the websites are hosted in different EC2 instance and you want to distribute incoming traffic between them to make them highly available.

Normally, we would create two AWS load balancers using CLB, but using ALB it is possible with one and also your money is saved. Instead of paying for 2 ELBs, only pay for a single ELB.

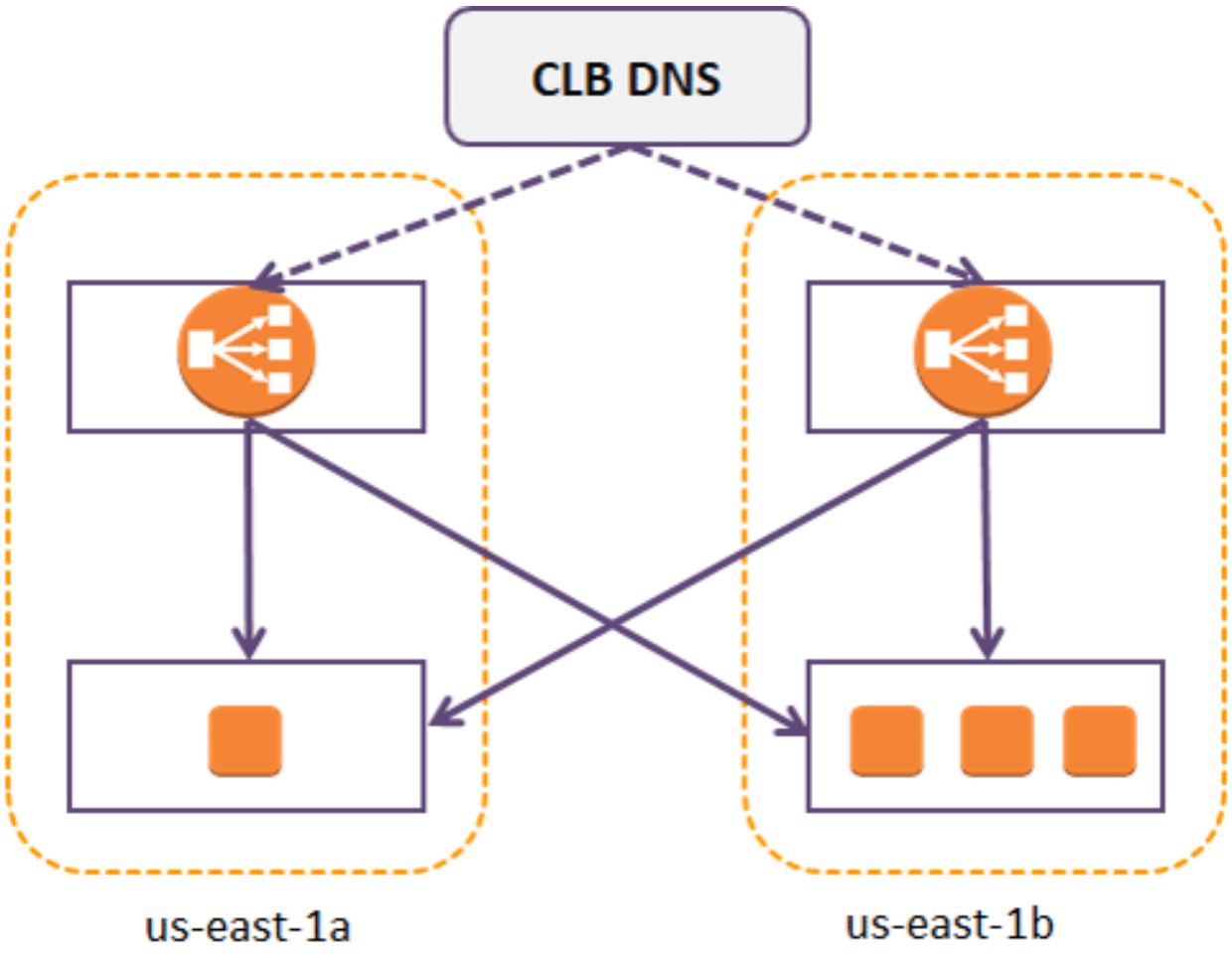
Path-based Routing using ALB

In this type of routing, the websites URL path will be hosted on different EC2 instances. For example, consider intellipaat.com and intellipaat.com/tutorial and these URL paths are hosted on different EC2 instances. Now, if you want to route traffic between these two URLs then you can use a path-based routing method. ALB can be

used to solve this problem too, you can use traffic routing according to the path feature by using just one ALB.

Cross-Zone Load Balancing

By default CLB nodes distributes traffic to instances in its availability zone only. Enable cross-zone load balancing to route evenly across EC2 instances. If cross-zone load balancing is disabled, the load balancer will only distribute traffic to instances in its Availability Zone. Enabling it will evenly distribute traffic across all AZs where registered targets are available.



Advantages of AWS ELB

- **Highly Available**

        - ELB distributes traffic evenly among all the targets, for example multiple EC2 instances.
        - ELB has an SLA of 99.99%

- **Flexible**
    - ELB lets you route traffic with the application's IP address, this allows you launch multiple applications in a single instance.

- **Highly secure**
        - You can implement robust security features using Amazon VPC with Amazon ELB

- **Elastically scalable**
        - ELB can handle sudden spikes in traffic and can handle millions of requests per second. Whenever there is a traffic increase, AWS auto scaling feature will be enabled and also load balancing rules will be used to provide the website users an seamless performance

- **Hybrid load balancing**
        - You can use the same Amazon load balancer to balance across applications on your on-premises set up and you AWS infrastructure. Now, it will be very easy to migrate your application from on-premise to AWS cloud.

- **Robust monitoring and auditing**
    - Applications and their performance can be monitored and maintained. You can also use CloudWatch metrics and logs to analyze our applications data, traffic, and working.

What is EC2 in AWS?

The full form of Amazon EC2 is Amazon Elastic Compute Cloud. Amazon EC2 is one of the most used and most basic services on Amazon so it makes sense to start with EC2 when you are new to AWS.

Well, to be very simple, EC2 is a machine with an operating system and hardware components of your choice. But the difference is that it is totally virtualized. You can run multiple virtual computers in a single physical hardware.

Elastic Compute Cloud (EC2) is one of the integral parts of the AWS ecosystem. EC2 enables on-demand, scalable computing capacity in the AWS cloud.

Amazon EC2 instances eliminate the up-front investment for hardware, and there is no need to maintain any rented hardware. It enables you to build and run applications faster. You can use EC2 in AWS to launch as many virtual servers as you need. Also, you can scale up or down when there is an increase or decrease in website traffic.

The word 'elastic' in Elastic Compute Cloud talks about the system's capability of adapting to varying workloads and provisioning or de-provisioning resources according to the demand.

Why Amazon EC2?

Now that we know the EC2 overview, let's now move forward and understand Why exactly we need Amazon EC2. AWS Elastic Compute Cloud provides a lot of benefits, let me give you an overview of what I am going to discuss.

- Auto-scaling
- Pay-as-you-go
- Increased Reliability
- Elasticity

Auto-scaling:

This is the benefit that makes most businesses opt for AWS EC2. It is already explained earlier how Netflix uses Amazon EC2 auto-scaling to its advantage and provides a crash-free experience.

Auto-scaling is basically providing resources according to the demand. They either scale up or scale down corresponding to the increase or decrease in demand.

Pay-as-you-go:

You will be charged by the hour, and you have to pay only for what you have used. A company, XYZ might be using 100 servers normally, and on Mondays, it scales down to 50 servers. So, it only has to pay for 50 servers those days, not the usual fee for the usage of 100 servers.

Even when you use your Amazon EC2 instances services for a few hours, you only need to pay for that time period and nothing more.

Increased Reliability:

AWS is spread across 20 worldwide regions with 61 availability zones (AZs) which helps your business when it is expanding. Also, this will increase the load speed of your application around the world.

You can always store multiple copies of your application in multiple AZs so that when one data center fails or loses data, the application will not fail completely.

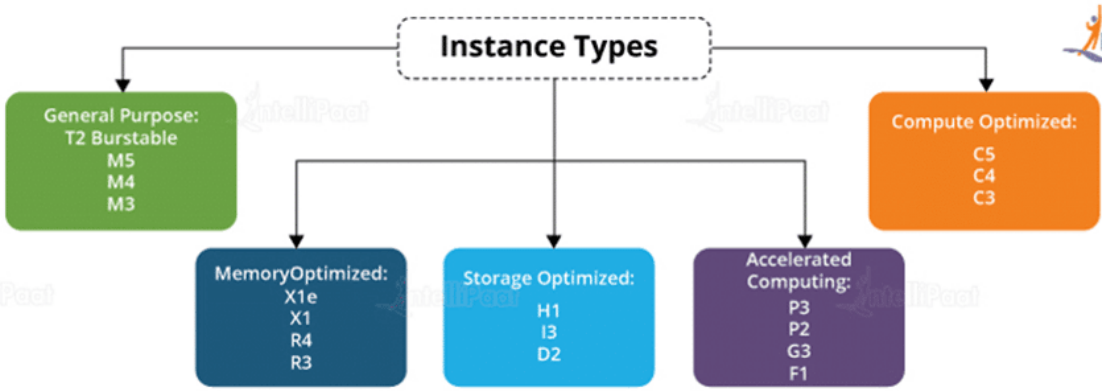
Elasticity:

Instead of 10 low-configuration machines, you could rent a single high-configuration machine with an OS of your preferred choice for your application. Elasticity is the feature from which Elastic Compute Cloud got its name.

Moving on, let’s now see different types of Amazon EC2 Instance Types.

AWS EC2 Instance Types

AWS EC2 instance types determine the underlying hardware of the instances which are launched.



There are several types of AWS instances with different configurations and benefits.

- General purpose
- Compute optimized
- Memory-optimized
- Accelerated Computing
- Storage optimized

General-purpose Instances

General-purpose instances provide a balance among compute, memory, and networking resources, and they can be used for a variety of workloads.

A1 Instances

A1 instances are used in applications that work in synchrony with the Arm ecosystem and are suitable for scale-out workloads.

M5, M5a, and M5d Instances

These instances offer a balance among compute, memory, and networking resources providing an ideal cloud design. It could be used for a wide range of applications.

T2 and T3 Instances

These instances provide clock up or down CPU performance.

Compute-optimized Instances

These instances are useful for compute-dependent applications that need high-performance processors. They are well suited for the following applications:

- Batch processing workloads
- High-performance web servers
- High-performance computing (HPC)
- Scientific modeling
- Dedicated gaming servers and ad serving engines

C5, C4, and C5n are the instances under this category.

Memory-optimized Instances

These instances are for delivering fast performance for processing large data sets in memory.

R4, R5, R5a, and R5d instances are memory optimized.

Accelerated Computing Instances

These instances are the latest gen general-purpose instances, and they provide an accelerated performance when the CPU clock rate increases.

P3, P2, G3, and F1 are instances for accelerated computing.

P3 and P2 are general-purpose instances.

G3 is for graphic-intensive applications.

Storage-optimized Instances

Storage-optimized instances are designed for workloads that contain very large data sets which has to be written in memory and require high, sequential read and write access.

EC2 vs. S3

Both Amazon EC2 and Amazon S3 are important services that allow developers to maximize use of the AWS cloud. The main difference between Amazon EC2 and S3 is that EC2 is a computing service that allows companies to run servers in the cloud. While S3 is an object storage service used to store and retrieve data from AWS through the Internet. S3 is like a giant hard drive in the cloud, while EC2 offers CPU and RAM in addition to storage. Many developers use both services for their cloud computing needs.

Amazon EC2 Instance Features

Many of the Amazon instances features are customizable such as storage, virtual processor, memory available to instances, etc. Below are some of the features of the Amazon EC2 instance:

Elastic IP addresses:

It can be moved from instance to instance without requiring a network administrator's help. This helps in the case of failover clusters, for load balancing, or for other purposes where multiple servers run at the same time.

Operating system:

EC2 instance supports several OSes such as Linux, Microsoft Windows Server, CentOS and Debian.

Amazon CloudWatch:

It is used to collect, store, and analyze historical and real-time performance data. The CloudWatch service allows for the monitoring of AWS cloud services and applications deployed on AWS. It also actively monitors applications, improves resource use, optimizes costs, and scales up or down.

Automated scaling:

Automated scaling helps in adding or removing capacity from Amazon EC2 virtual servers in response to application demand. Auto Scaling provides more capacity to handle temporary increases in traffic during a product launch.

Bare-metal instances:

The virtual server instances consist of the hardware resources, such as processor, storage, and network. They are not virtualized and do not run an OS.

Amazon EC2 Fleet:

As a single virtual server, Amazon EC2 Fleet enables the deployment and management of instances. It also provides programmatic access to fleet operations using an API. Further, Fleet management can be integrated into existing management tools.

Pause and resume instances:

EC2 instances can be easily paused and resumed from the same state later. In case, there are too many instances running at the same time, they can be paused without incurring charges.

Persistent storage:

Amazon EBS service enables block-level storage volumes to be attached to EC2 instances and be used as hard drives. With the help of EBS, it is possible to increase or

decrease the amount of storage available to an EC2 instance and attach EBS volumes to multiple instances.

AWS EC2 Pricing

Generally, Free Tier 750 hours of free usage for up to one year is provided by AWS. Only the t2.micro instance can be used on Linux and Windows AMIs.

On-demand Price:

m5.large	$0.096/hour
c5.large	$0.085/hour
r4.large	$0.133/hour

Data Transfer IN:

FREE from any region in the world

Data Transfer OUT:

From EC2 to:

S3, Glacier, DynamoDB, SES, and SQS in the same region	FREE
S3, Glacier, DynamoDB, SES, and SQS in the same region	$0.020/GB
EC2, RDS, Redshift, ElastiCache, ELB, and ENI in the same AZ with private IP	FREE
EC2, RDS, Redshift, ElastiCache, ELB, and ENI in the same AZ with public IP	$0.010/GB
EC2, RDS, Redshift, ElastiCache, ELB, and ENI in different AZs	$0.010/GB

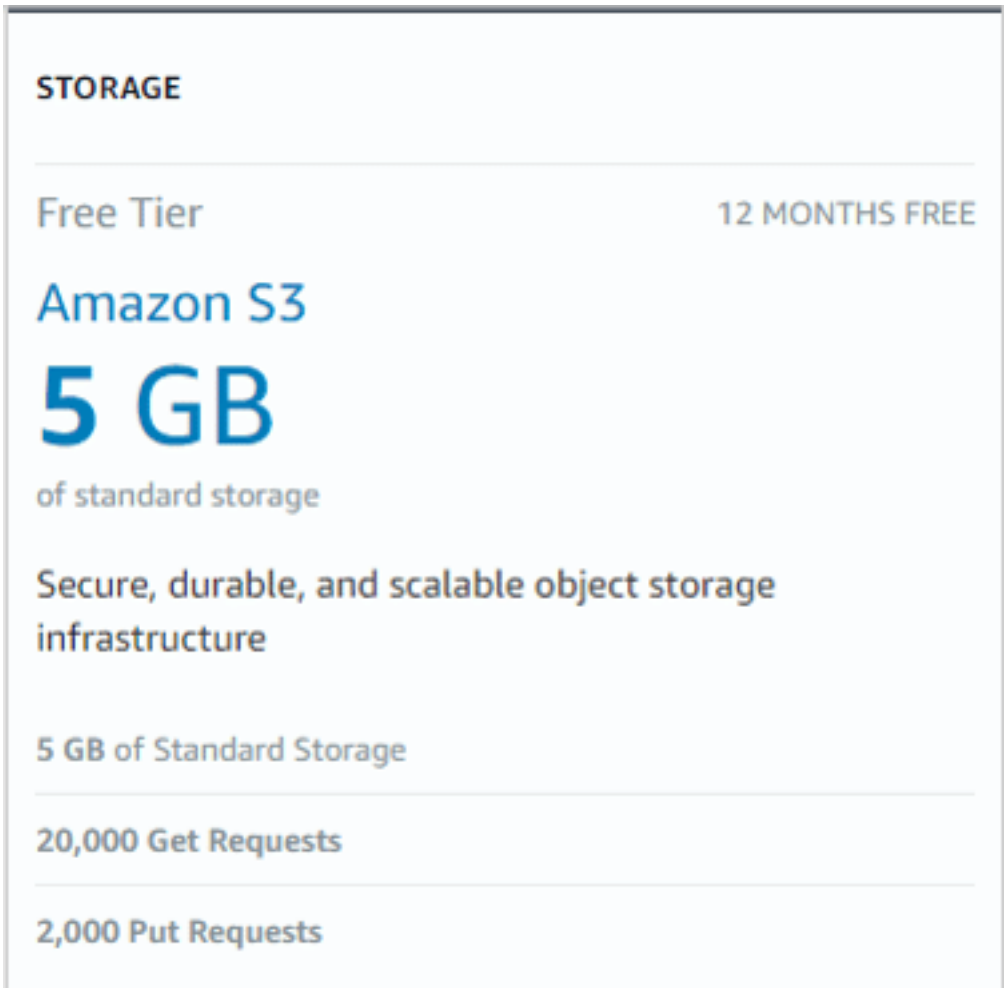
Knowing about EC2 in AWS is one of the first things one must do when they are starting with AWS but it won't do any good if you don't know how to create an EC2 instance.

What Is Amazon S3?

Amazon Simple Storage Service (S3) is a storage that can be maintained and accessed over the Internet. Amazon S3 provides a web service that can be used to store and

retrieve an unlimited amount of data. The same can be done programmatically using Amazon-provided APIs.

Also, S3 in Amazon is provided for free in the free-tier category for 12 months.

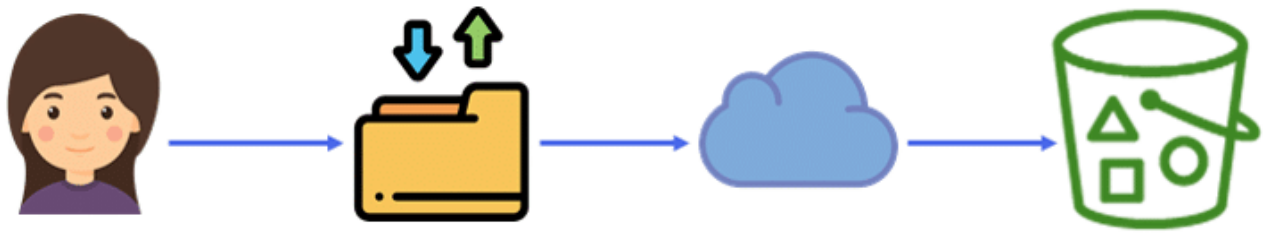


What Is an Amazon S3 Bucket?

S3 in Amazon has two primary entities called buckets and objects. Objects are stored inside buckets. Also, it does have a flat hierarchy, not like the one you would find in a file system. But in an organization, a file system is needed in an ordered fashion, and that's why AWS S3 introduced a file system which seems like a traditional one.

Basically, it works like, if you upload images and you want to differentiate it from other files, you can create a file for it and store it so that the logical address of the file would have the prefix 'pictures.' For example, pictures/hello.jpg.

By default, the maximum number of buckets that can be created per account is 100. For additional buckets, one can submit a request for a service limit increase.



Bucket names have to be globally unique irrespective of which region they are created in. As buckets can be accessed using URLs, it is recommended that bucket names follow DNS naming conventions: all letters should be in lowercase.

We'll also learn how to create an AWS S3 bucket from scratch in this Amazon S3 tutorial, but first, we need to know to have some basic understanding of Amazon S3 concepts.

Amazon S3 Concepts

Now that you know what is Amazon S3 and what is Amazon S3 bucket, let's move on and discuss some basic Amazon S3 concepts starting from Data Consistency Models.

Data Consistency Models

S3 in Amazon provides amazing highly available and durable storage solutions by replicating the data under one bucket in multiple data centers in the region. Also, the data uploaded to an Amazon S3 bucket never leave it until you delete it.

S3 has 2 types in the consistency models:

- read-after-write consistency
- Eventual consistency

Read-after-write consistency for PUTS of new objects in your S3 bucket in all the regions with one limitation. The limitation is that if you perform a GET or HEAD for an object which does not exist then first eventual consistency will be provided.

Eventual consistency is provided by AWS S3 when there are PUT and DELETES in all regions.

Storage Classes

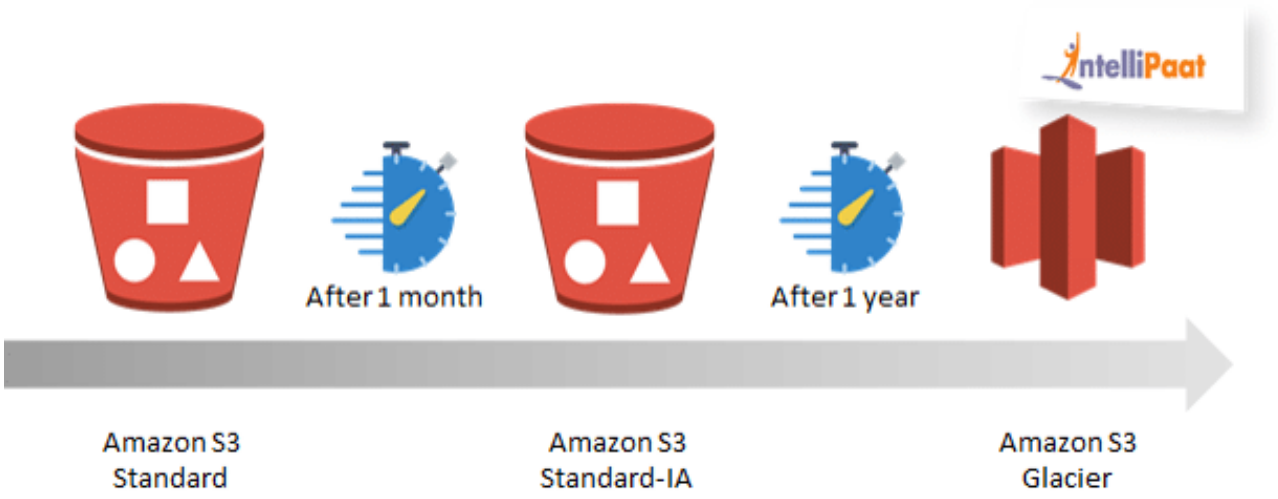
Storage classes are used to distinguish between the use cases of a particular object stored in a bucket. There are various types of storage classes and they are listed below.

Storage Class	***Description***
Standard	Frequently used objects
Standard-IA	Infrequently used objects
Intelligent-tiering	Designed to optimize storage costs by choosing Standard and Standard-IA for objects
One Zone-IA	Infrequently used objects and non-mission-critical data
Glacier	Long-term data archiving with retrieval times ranging from minutes to hours
Deep Archive	Archiving and rarely accessing with retrieval times averaging 12 hours

Object Lifecycle Management

Configuring your object's lifecycle could ensure they are stored cost effectively for their entire lifecycle. A lifecycle configuration is a set of rules that define actions that AWS S3 applies to a group of objects. There are two types of actions:

- **Transaction Actions**
    - This action defines objects' transition from one storage class to another.
    - You might have provided your object with a Standard class at first, then after 2 months you want to make it Standard-IA, and after a year you want to keep it in Amazon Glacier.
- **Expiration Actions**
    - This action deletes objects in the Amazon S3 bucket.
    - Also, S3 deletes the expired objects on your behalf.



But you might think, why do you need a lifecycle for the objects? Check out these examples.

- You wanted to store log data for the month of April for services; you might need it for a week or a month, and then you might want to delete it.
- You have objects which are frequently used for a week and then they become infrequent. After that, once the usage for it has reduced to the rock bottom, you can archive it and then delete it.

- You also might store data for long-term archiving

Object Versioning

Object versioning provides the flexibility of storing objects with the same name by giving objects version numbers. This prevents unintended overwriting or deletion of the objects.

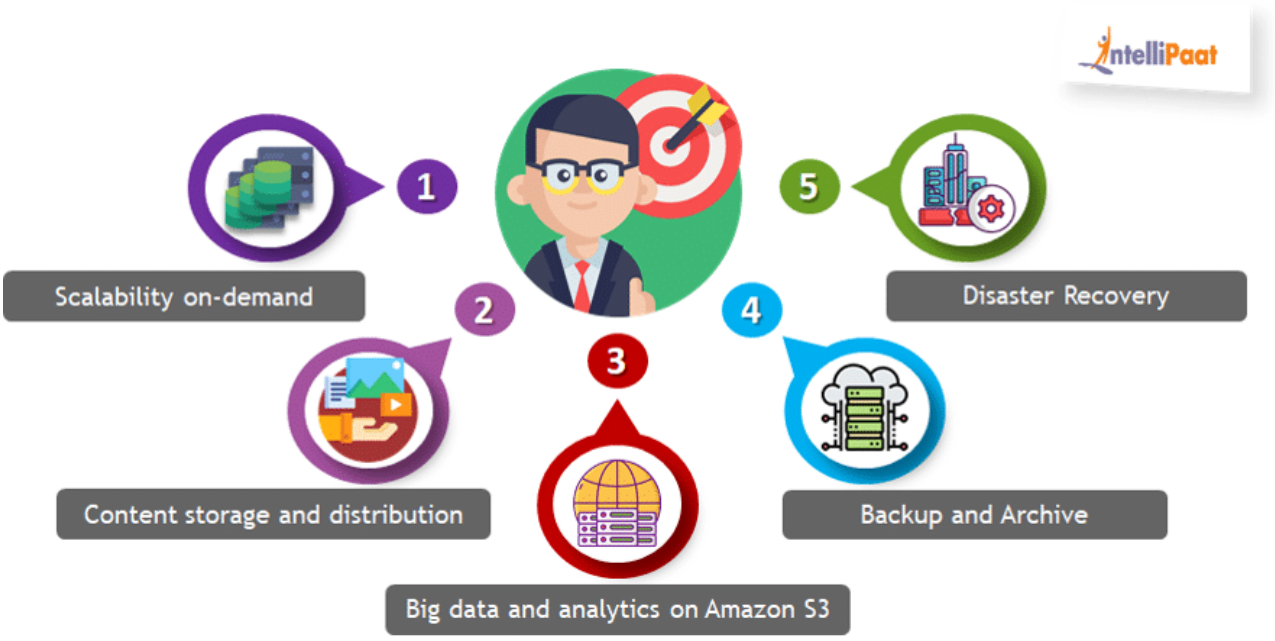
For example, you have a file called image.jpg with the versioning of 111, and now you are uploading another file named image.jpg, again. So, the newly uploaded image.jpg file will have a versioning of 222. So, they are image.jpg (version 111) and image.jpg (version 222).

Versioning can also be used so that you can compare the older versions of the file.

Encryption

Using S3 default encryption you can set the default encrypted behavior for an Amazon S3 bucket. You can also create a default encryption where every object gets encrypted when stored in a bucket. The objects are encrypted using server-side encryption with either Amazon S3-managed keys (SSE-S3) or AWS KMS-managed keys (SSE-KMS).

Advantages of AWS S3 Service



- **Scalability on Demand**
  - If you want your application's scalability varying according to the change in traffic, then AWS S3 is a very good option.
  - Scaling up or down is just mouse-clicks away when you use other attractive features of AWS.
- **Content Storage and Distribution**
  - S3 in Amazon could be used as the foundation for a Content Delivery Network. Because Amazon S3 is developed for content storage and distribution.
- **Big Data and Analytics on Amazon S3**
  - Amazon QuickSight UI can be connected with Amazon S3, and then large amounts of data can be analyzed with it.
- **Backup and Archive**
  - Whether you need timely backups of your website, or store static files for once, or store versions of files you are currently working on. S3 in Amazon has got you covered.
- **Disaster Recovery**
  - Storing data in multiple availability zones in a region gives the user the flexibility to recover files, which are lost, as soon as possible. Also, the

cross-region replication technology can be used to store in any number of Amazon's worldwide data centers.

Understanding AWS Public Vs Private Services

First of all on the basis of networking, we can categorise AWS services as...

1. Public Services
2. Private Services

Public Services

👉 Public services are something that has public endpoints
👉 And because of these endpoints you can access it from a public internet
👉 Example: Amazon S3
👉 But even though it is a public service, by default the access is denied, you have to give permission

Private Services

👉 A private service is the one that runs within VPC
👉 So only other components within VPC has access to this service
👉 Obviously these services can not be accessed on the public internet

AWS Public Zone

👉 AWS public services don't operate on the public internet
👉 They operate in AWS's public zone
👉 This AWS's public zone is connected with public internet and AWS private zone