Abstract

*This document is aimed at looking for the overview of Amazon Web Services (***AWS***) is the market leader in* ***IaaS*** *(Infrastructure-as-a-Service) and* ***PaaS*** *(Platform-as-a-Service) for cloud ecosystems, which can be combined to create a scalable cloud application without worrying about delays related to infrastructure provisioning (***C***ompute,* ***S***torage, and* ***N***etwork) and management. Amazon offers many services for application development and analytics. Here are some key building blocks in the AWS environment and a brief description of how they are leveraged against business needs.*

Architecting for the Cloud &

Overview of IAM, EC2 &VPC in AWS

Prepared by: **Debi Mishra**

Employee ID: **295874**

Version: 0.6

Last Modified Date: **04-Jan-20**

Révision History

Date	Version	Author	Reviewer	Change Description
28-Sep-19	0.1	Debi Mishra		Prepared the draft version
16-Oct-19	0.2	Debi Mishra		Added the MFA contents in AWS IAM Service
10-Nov-19	0.3	Debi Mishra		Modified the contents on Design Principles
27-Nov-19	0.4	Debi Mishra		Added EC2 Instance Life Cycle diagrammatic representation
19-Dec-19	0.5	Debi Mishra		Modified the Security Groups & NACL Rules in VPC section
04-Jan-20	0.6	Debi Mishra		Modified the contents of VPC Peering in VPC section

Glossary of Terms

Terms/ Acronym	Definition
AWS	Amazon Web Service
EC2	Elastic Compute Cloud
S3	Simple Storage Services
IAAS	Infrastructure as a Service
PAAS	Platform as a Service
SAAS	Software as a Service
EBS	Elastic Block Storage
IS	Instance Store
VPC	Virtual Private Compute
AZ	Availability Zone
AMI	Amazon Machine Image
RDS	Relational Data Storage
SQS	Simple Queue Service
SNS	Simple Notification Service
CDN	Content Delivery Network
CFT	Cloud Front Technology
RDS	Relational Data base Services
MPP	Massively Parallel Processing
DNS	Domain Name System
MFA	Multi Factor Authentication
IAM	Identity and Access Management
CIDR	Classless Inter Domain Routing
DHCP	Dynamic Host Control Protocol
ENI	Elastic Network Interface
MAC	Medium Access Control
NAT	Network Address Translation

Table of Contents

Introduction

Amazon **W**eb **S**ervices (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered **pay-as-you-go** basis. In aggregate, these cloud computing web services provide a set of primitive abstract technical infrastructure and distributed computing building blocks and tools. One of these services is Amazon Elastic Compute Cloud, which allows users to have at their disposal a virtual cluster of computers, available all the time, through the Internet. AWS's version of virtual computers emulate most of the attributes of a real computer, including hardware central processing units (CPUs) and graphics processing units (GPUs) for processing; local/RAM memory; hard-disk/SSD storage; a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, and customer relationship management (CRM).

The AWS technology is implemented at server farms throughout the world, and maintained by the Amazon subsidiary. Fees are based on a combination of usage (known as a "Pay-as-you go" model), the hardware/OS/software/networking features chosen by the subscriber, required availability, redundancy, security, and service options. Subscribers can pay for a single virtual AWS computer, a dedicated physical computer, or clusters of either. As part of the subscription agreement Amazon provides security for subscribers' system.

In 2017, AWS comprised more than 90 (165 as of 2019) services spanning a wide range including computing, storage, networking, database, analytics, application services, deployment, management, mobile, developer tools, and tools for the Internet of Things. The most popular include Amazon Elastic Compute Cloud (EC2) and Amazon Simple Storage Service (Amazon S3). Most services are not exposed directly to end users, but instead offer functionality through APIs for developers to use in their applications. Amazon Web Services' offerings are accessed over HTTP, using the REST architectural style and SOAP protocol.

Amazon markets AWS to subscribers as a way of obtaining large scale computing capacity more quickly and cheaply than building an actual physical server farm. All services are billed based on usage, but each service measures usage in varying ways. As of 2017, AWS owns a dominant 34% of all cloud (IaaS, PaaS) while the next three competitors Microsoft, Google, and IBM have 11%, 8%, 6% respectively according to Synergy Group

What is Cloud Computing

Cloud computing is an internet-based computing service in which large groups of remote servers are networked to allow centralized data storage, and online access to computer services or resources. Using cloud computing, organizations can use shared computing and storage resources rather than building, operating, and improving infrastructure on their own. Cloud computing is a model that enables the following features.

- Users can provision and release resources on-demand.
- Resources can be scaled up or down automatically, depending on the load.
- Resources are accessible over a network with proper security.

- Cloud service providers can enable a pay-as-you-go model, where customers are charged based on the type of resources and per usage.

Types of Clouds

There are three types of clouds − Public, Private, and Hybrid cloud.

- **Public Cloud -** In public cloud, the third-party service providers make resources and services available to their customers via Internet. Customer's data and related security is with the service providers' owned infrastructure.
- **Private Cloud -** A private cloud also provides almost similar features as public cloud, but the data and services are managed by the organization or by the third party only for the customer's organization. In this type of cloud, major control is over the infrastructure so security related issues are minimized.
- **Hybrid Cloud -** A hybrid cloud is the combination of both private and public cloud. The decision to run on private or public cloud usually depends on various parameters like sensitivity of data and applications, industry certifications and required standards, regulations, etc.

Cloud Service Models

There are three types of service models in cloud − IaaS, PaaS, and SaaS.

- **IaaS -** IaaS stands for Infrastructure as a Service. It provides users with the capability to provision processing, storage, and network connectivity on demand. Using this service model, the customers can develop their own applications on these resources.
- **PaaS -** PaaS stands for Platform as a Service. Here, the service provider provides various services like databases, queues, workflow engines, e-mails, etc. to their customers. The customer can then use these components for building their own applications. The services, availability of resources and data backup are handled by the service provider that helps the customers to focus more on their application's functionality.
- **SaaS -** SaaS stands for Software as a Service. As the name suggests, here the third-party providers provide end-user applications to their customers with some administrative capability at the application level, such as the ability to create and manage their users. Also some level of customizability is possible such as the customers can use their own corporate logos, colors, etc.

Cloud Benefits over On-premises Architecture

Business Benefits

- Almost zero upfront infrastructure investment: If you have to build a large-scale system it may cost a fortune to invest in real estate, physical security, hardware (racks, servers, routers, backup power supplies), hardware management (power management, cooling), and operations personnel. Because of the high upfront costs, the project would typically require several rounds of management approvals before the project could even get started.

- Just-in-time Infrastructure: In the past, if your application became popular and your systems or your infrastructure did not scale you became a victim of your own success. Conversely, if you invested heavily and did not get popular, you became a victim of your failure. By deploying applications in-the-cloud with just-in-time self-provisioning, you do not have to worry about pre-procuring capacity for large-scale systems.
- More efficient resource utilization: System administrators usually worry about procuring hardware (when they run out of capacity) and higher infrastructure utilization (when they have excess and idle capacity). With the cloud, they can manage resources more effectively and efficiently by having the applications request and relinquish resources on-demand.
- Usage-based costing: With utility-style pricing, you are billed only for the infrastructure that has been used. You are not paying for allocated but unused infrastructure. This adds a new dimension to cost savings. You can see immediate cost savings (sometimes as early as your next month's bill) when you deploy an optimization patch to update your cloud application.
- Reduced time to market: Parallelization is the one of the great ways to speed up processing. If one compute-intensive or data-intensive job that can be run in parallel takes 500 hours to process on one machine, with cloud architectures, it would be possible to spawn and launch 500 instances and process the same job in 1 hour. Having available an elastic infrastructure provides the application with the ability to exploit parallelization in a cost-effective manner reducing time to market.

IT/ Technical Benefits

- Automation – "Scriptable infrastructure": You can create repeatable build and deployment systems by leveraging programmable (API-driven) infrastructure.
- Auto-scaling: You can scale your applications up and down to match your unexpected demand without any human intervention. Auto-scaling encourages automation and drives more efficiency.
- Proactive Scaling: Scale your application up and down to meet your anticipated demand with proper planning understanding of your traffic patterns so that you keep your costs low while scaling.
- More Efficient Development lifecycle: Production systems may be easily cloned for use as development and test environments. Staging environments may be easily promoted to production.
- Improved Testability: Never run out of hardware for testing. Inject and automate testing at every stage during the development process. You can spawn up an "instant test lab" with pre-configured environments only for the duration of testing phase.
- Disaster Recovery and Business Continuity: The cloud provides a lower cost option for maintaining a fleet of DR servers and data storage. With the cloud, you can take advantage of geo-distribution and replicate the environment in other location within minutes.
- "Overflow" the traffic to the cloud: With a few clicks and effective load balancing tactics, you can create a complete overflow-proof application by routing excess traffic to the cloud.

Design Principles

Scalability

- Systems that are expected to grow over time need to be built on top of a scalable architecture. Such an architecture can support growth in users, traffic, or data size with no drop-in performance. It should provide that scale in a linear manner where adding extra resources results in at least a proportional increase in ability to serve additional load.
- **Scaling vertically** takes place through an increase in the specifications of an individual resource, such as upgrading a server with a larger hard drive or a faster CPU. With Amazon EC2, you can stop an instance and resize it to an instance type that has more RAM, CPU, I/O, or networking capabilities.
- **Scaling horizontally** takes place through an increase in the number of resources, such as adding more hard drives to a storage array or adding more servers to support an application. This is a great way to build internet-scale applications that leverage the elasticity of cloud computing.

Distribute Load to Multiple Nodes

- To distribute the workload to multiple nodes in your environment, you can choose either a push or a pull model.
- With a push model, you can use Elastic Load Balancing (ELB) to distribute a workload. ELB routes incoming application requests across multiple EC2 instances. When routing traffic, a Network Load Balancer operates at layer 4 of the Open Systems Interconnection (OSI) model to handle millions of requests per second. An Application Load Balancer provides Layer 7 of the OSI model and supports content-based routing of requests based on application traffic. Alternatively, you can use Amazon Route 53 to implement a DNS round robin. In this case, DNS responses return an IP address from a list of valid hosts in a round-robin fashion.
- Instead of a load balancing solution, you can implement a pull model for asynchronous, event-driven workloads. In a pull model, tasks that need to be performed or data that needs to be processed can be stored as messages in a queue using Amazon Simple Queue Service (Amazon SQS) or as a streaming data solution such as Amazon Kinesis. Multiple compute resources can then pull and consume those messages, processing them in a distributed fashion.

Distributed Processing

- Use cases that involve the processing of very large amounts of data—anything that can’t be handled by a single compute resource in a timely manner—require a distributed processing approach. By dividing a task and its data into many small fragments of work, you can execute them in parallel across a set of compute resources.
- Offline batch jobs can be horizontally scaled by using distributed data processing engines such as AWS Batch, AWS Glue, Apache Hadoop and Amazon EMR to run Hadoop workloads in AWS.
- For real-time processing of streaming data, Amazon Kinesis partitions data in multiple shards that can then be consumed by multiple Amazon EC2 or AWS Lambda resources to achieve scalability.

Automation

- In a traditional IT infrastructure, you often have to manually react to a variety of events. When deploying on AWS, there is an opportunity for automation, so that you improve both your system's stability and the efficiency of your organization.

Loose Coupling

- As application complexity increases, a desirable attribute of an IT system is that it can be broken into smaller, loosely coupled components. This means that IT systems should be designed in a way that reduces interdependencies—a change or a failure in one component should not cascade to other components.

Asynchronous Integration

- It is another form of loose coupling between services. This model is suitable for any interaction that does not need an immediate response and where an acknowledgement that a request has been registered will suffice.
- It involves one component that generates events and another that consumes them. The two components do not integrate through direct point-to-point interaction but usually through an intermediate durable storage layer, such as an SQS queue or a streaming data platform such as Amazon Kinesis, cascading Lambda events, AWS Step Functions, or Amazon Simple Workflow Service.

Services, Not Servers

- AWS managed services provide building blocks that developers can consume to power their applications. These managed services include databases, machine learning, analytics, queuing, search, email, notifications, and more.
- For example, with Amazon SQS you can offload the administrative burden of operating and scaling a highly available messaging cluster, while paying a low price for only what you use. Amazon SQS is also inherently scalable and reliable.
- The same applies to Amazon S3, which enables you to store as much data as you want and access it when you need it, without having to think about capacity, hard disk configurations, replication, and other related issues.

Other examples of managed services that power your applications include:

- Amazon Cloud Front for content delivery
- ELB for load balancing
- Amazon Dynamo DB for NoSQL databases
- Amazon Cloud Search for search workloads
- Amazon Elastic Transcoder for video encoding
- Amazon Simple Email Service (Amazon SES) for sending and receiving emails

Databases

- Amazon RDS makes it easy to set up, operate, and scale a relational database in the cloud with support for many familiar database engines.
- Relational databases can scale vertically by upgrading to a larger Amazon RDS DB instance or adding more and faster storage. In addition, consider using Amazon Aurora, which is a database engine designed to deliver much higher throughput compared to standard MySQL running on the same hardware. For read-heavy applications, you can also horizontally scale beyond the capacity constraints of a single DB instance by creating one or more read replicas.
- NoSQL databases trade some of the query and transaction capabilities of relational databases for a more flexible data model that seamlessly scales horizontally. NoSQL databases use a variety of data models, including graphs, key-value pairs, and JSON documents, and are widely recognized for ease of development, scalable performance, high availability, and resilience. Amazon Dynamo DB is a fast and flexible NoSQL database service for applications that need consistent, single-digit, millisecond latency at any scale. It is a fully managed cloud database and supports both document and key-value store models.
- NoSQL database engines will typically perform data partitioning and replication to scale both the reads and the writes in a horizontal fashion. They do this transparently, and don’t need the data partitioning logic implemented in the data access layer of your application. Amazon Dynamo DB in particular manages table partitioning automatically, adding new partitions as your table grows in size or as read-provisioned and write-provisioned capacity changes. Amazon Dynamo DB Accelerator (DAX) is a managed, highly available, in-memory cache for Dynamo DB to leverage significant performance improvements.

Data warehouse

- On AWS, you can leverage Amazon Redshift, a managed data warehouse service that is designed to operate at less than a tenth the cost of traditional solutions. Amazon Redshift achieves efficient storage and optimum query performance through a combination of massively parallel processing (MPP), columnar data storage, and targeted data compression encoding schemes. It is particularly suited to analytic and reporting workloads against very large data sets.
- The Amazon Redshift MPP architecture enables you to increase performance by increasing the number of nodes in your data warehouse cluster. Amazon Redshift Spectrum enables Amazon Redshift SQL queries against exabytes of data in Amazon S3, which extends the analytic capabilities of Amazon Redshift beyond data stored on local disks in the data warehouse to unstructured data, without the need to load or transform data.

Caching

- Caching is a technique that stores previously calculated data for future use. This technique is used to improve application performance and increase the cost efficiency of an implementation.

- Amazon Elasti Cache is a web service that makes it easy to deploy, operate, and scale an in-memory cache in the cloud. Copies of static content (images, CSS files, or streaming pre-recorded video) and dynamic content (responsive HTML, live video) can be cached at an Amazon Cloud Front edge location, which is a CDN with multiple points of presence around the world.
- Edge caching allows content to be served by infrastructure that is closer to viewers, which lowers latency and gives you the high, sustained data transfer rates necessary to deliver large popular objects to end users at scale.

AWS Services Highlights

The Amazon Web Services (AWS) cloud provides a highly reliable and scalable infrastructure for deploying web-scale solutions, with minimal support and administration costs, and more flexibility than you've come to expect from your own infrastructure, either on-premise or at a data center facility.

Understanding the services

- **Amazon Elastic Compute Cloud** (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. You can bundle the operating system, application software and associated configuration settings into an Amazon Machine Image (AMI). You can then use these AMIs to provision multiple virtualized instances as well as decommission them using simple web service calls to scale capacity up and down quickly, as your capacity requirement changes. Instances can be launched in one or more geographical regions. Each region has multiple Availability Zones. Availability Zones are distinct locations that are engineered to be insulated from failures in other Availability Zones and provide inexpensive, low latency network connectivity to other Availability Zones in the same Region.
- **Elastic IP** addresses allow you to allocate a static IP address and programmatically assign it to an instance. You can enable monitoring on an Amazon EC2 instance using Amazon CloudWatch in order to gain visibility into resource utilization, operational performance, and overall demand patterns (including metrics such as CPU utilization, disk reads and writes, and network traffic).
- **Amazon S3** is highly durable and distributed data store. With a simple web services interface, you can store and retrieve large amounts of data as objects in buckets (containers) at any time, from anywhere on the web using standard HTTP verbs. Copies of objects can be distributed and cached at 14 edge locations around the world by creating a distribution using Amazon CloudFront service – a web service for content delivery (static or streaming content).
- **Amazon Relational Database Service** (Amazon RDS) provides an easy way to setup, operate and scale a relational database in the cloud. You can launch a DB Instance and get access to a full-featured AWS provided databases and not worry about common database administration tasks like backups, patch management etc.
- **Amazon Simple Queue Service** (Amazon SQS) is a reliable, highly scalable, hosted distributed queue for storing messages as they travel between computers and application components.

- **Amazon Simple Notifications Service** (Amazon SNS) 11provides a simple way to notify applications or people from the cloud by creating Topics and using a publish-subscribe protocol.
- **Amazon Elastic MapReduce** provides a hosted Hadoop framework running on the web-scale infrastructure of Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Service (Amazon S3) and allows you to create customized JobFlows. JobFlow is a sequence of MapReduce steps.
- **Amazon Virtual Private Cloud** (Amazon VPC) allows you to extend your corporate network into a private cloud contained within AWS. Amazon VPC that enables you to create a secure connection between a gateway in your data center and a gateway in AWS.
- **Amazon Route53** is a highly scalable DNS service that allows you manage your DNS records by creating a HostedZone for every domain you would like to manage.
- **AWS Identity and Access Management** (IAM) enable you to create multiple Users with unique security credentials and manage the permissions for each of these Users within your AWS Account.

AWS IAM Services

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources for your users. IAM is used to control: **Identity** – who can use your AWS resources (authentication) & **Access** – what resources they can use and in what ways (authorization). IAM can also keep your account credentials private. With IAM, multiple IAM users can be created under the umbrella of the AWS account or temporary access can be enabled through identity federation with corporate directory or third party providers IAM also enables access to resources across AWS accounts.

AWS IAM Features

- **Shared access to your AWS account** - Grant other people permission to administer and use resources in your AWS account without having to share your password or access key.
- **Granular permissions -** Each user can be granted with different set granular permissions as required to perform their job.
- **Secure access to AWS resources** for applications that run on EC2 - Provide applications running on EC2 instance temporary credentials that they need in order to access other AWS resources.
- **Identity Federation -** IAM allows users to access AWS resources, without requiring the user to have accounts with AWS, by providing temporary credentials for e.g. through corporate network or Google or Amazon authentication.
- **Identity information for assurance** – Cloud Trail can be used to receive log records that include information about those who made requests for resources in the account.
- **PCI DSS Compliance -** IAM supports the processing, storage, and transmission of credit card data by a merchant or service provider, and has been validated as being Payment Card Industry Data Security Standard (PCI DSS) compliant.
- **Integrated** with many AWS services - IAM integrates with almost all the AWS services.

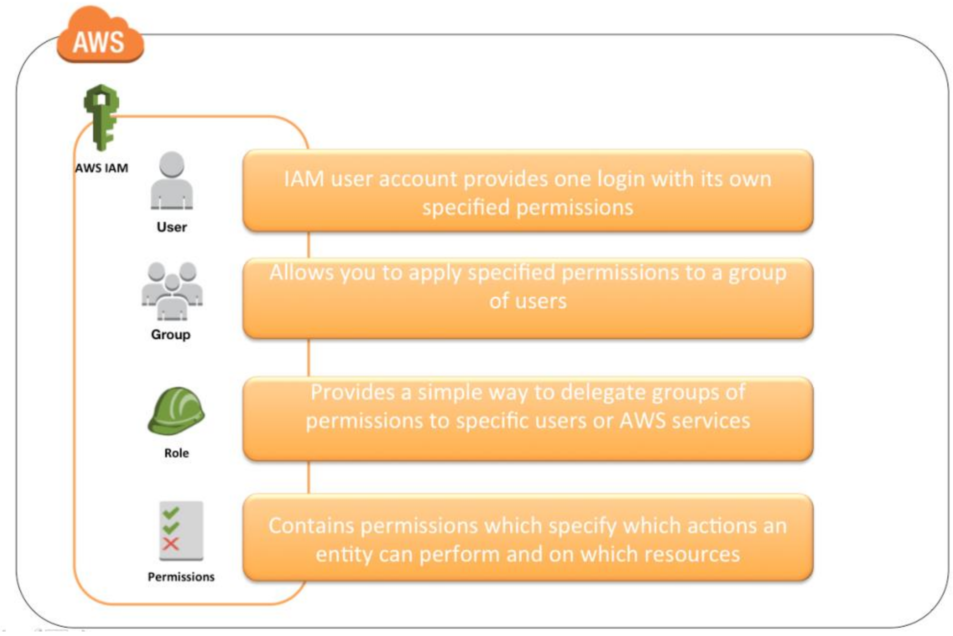
- **Eventually Consistent -** IAM is eventually consistent and achieves high availability by replicating data across multiple servers within Amazon’s data centers around the world. Changes made to IAM would be eventually consistent and hence would take some time to reflect
- **Free to use -** IAM is offered at no additional charge and charges are applied only for use of other AWS products by your IAM users.
- **AWS Security Token Service -** IAM provide STS which is an included feature of the AWS account offered at no additional charge. AWS charges only for the use of other AWS services accessed by the AWS STS temporary security credentials.

AWS Identities

IAM identities determine who can access and help to provide authentication for people and processes in your AWS account.

- Root Account Credentials are the email address and password with which you sign-in into the AWS account.
- Root Credentials has full unrestricted access to AWS account including the account security credentials which include sensitive information
- IAM Best Practice – Do not use or share the Root account once the AWS account is created, instead create a separate user with admin privilege
- An Administrator account can be created for all the activities which too has full access to the AWS account except the accounts security credentials, billing information and ability to change password.

AWS IAM Identities



IAM user account provides one login with its own specified permissions

Allows you to apply specified permissions to a group of users

Provides a simple way to delegate groups of permissions to specific users or AWS services

Contains permissions which specify which actions an entity can perform and on which resources

AWS IAM User

- IAM user represents the person or service who uses the access to interact with AWS. IAM Best Practice is create Individual Users.

- User credentials can consist of - Password to access AWS services through AWS Management Console. Access Key/Secret Access Key to access AWS services through API, CLI or SDK.
- IAM user starts with no permissions and is not authorized to perform any AWS actions on any AWS resources and should be granted permissions as per the job function requirement.
- IAM Best Practice – Grant least Privilege. Each IAM user is associated with one and only one AWS account. IAM User cannot be renamed from AWS management console and has to be done from CLI or SDK tools.

AWS IAM Group

- IAM group is a collection of IAM users. It can be used to specify permissions for a collection of users sharing the same job function making it easier to manage.
- IAM Best Practice – Use groups to assign permissions to IAM Users.
- A group is not truly an identity because it cannot be identified as a Principal in an access policy. It is only a way to attach policies to multiple users at one time.
- A group can have multiple users, while a user can belong to multiple groups (10 max). Groups cannot be nested and can only have users within it.
- Deletion of the groups requires you to detach users and managed policies and delete any inline policies before deleting the group. With AWS management console, the deletion and detachment is taken care of.

AWS IAM Role & Policy

- Roles can be created to act as a proxy to allow users or services to access resources. Roles supports trust policy which helps determine who can access the resources and permission policy which helps to determine what they can access.
- User who assumes a role temporarily gives up his or her own permissions and instead takes on the permissions of the role. When the user exits, or stops using the role, the original user permissions are restored. Roles can be used to provision access to almost all the AWS resources.
- Resource based policy allows you to attach a policy directly to the resource that you want to share, instead of using a role as a proxy.
- With Cross-account access with a resource-based policy, User still works in the trusted account and does not have to give up her user permissions in place of the role permissions.
- User can work on the resources from both the accounts at the same time and this can be useful for scenarios for e.g. copying of objects from one bucket to the other.
- Resource that you want to share are limited to resources which support resource-based policies,
  - Amazon S3 allows you to define Bucket policy to grant access to the bucket and the objects
  - Amazon Simple Notification Service (SNS)
  - Amazon Simple Notification Service (SNS)
  - Amazon Simple Queue Service (SQS)

- Amazon Glacier Vaults
- AWS OpsWorks stacks
- AWS Lambda functions

- Resource based policies need the trusted account to create users with permissions to be able to access the resources from the trusting account.

AWS IAM Multi Factor Authentication

- For increased security and to help protect the AWS resources, Multi-Factor authentication can be configured. IAM Best Practice – Enable MFA on Root accounts and privilege users.
- Multi-Factor Authentication can be configured using below ways as mentioned.
  - **Security token-based**
- AWS Root user or IAM user can be assigned a hardware/virtual MFA device.
- Device generates a six digit numeric code based upon a time-synchronized one-time password algorithm which needs to be provided during authentication.
  - **SMS text message-based** (Preview Mode)
- IAM user can be configured with the phone number of the user's SMS-compatible mobile device which would receive a 6 digit code from AWS.
- SMS-based MFA is available only for IAM users and does not work for AWS root account.

AWS Virtual Private Cloud

A virtual private cloud (VPC) is a virtual network dedicated to the AWS account. It is logically isolated from other virtual networks in the AWS cloud. VPC allows the user to select IP address range, create subnets, and configure route tables, network gateways, and security settings.

VPC Sizing

- VPC needs a set of IP addresses in the form of a Classless Inter-Domain Routing (CIDR) block for e.g, 10.0.0.0/16, which allows 2^16 (65536) IP address to be available.
- Allowed CIDR block size is between

  /28 netmask (minimum with 2^4 – 16 available IP address) and

  /16 netmask (maximum with 2^16 – 65536 IP address)
- It's possible to specify a range of publicly routable IP addresses; however, direct access to the Internet is not currently supported from publicly routable CIDR blocks in a VPC.
- Each VPC is separate from any other VPC created with the same CIDR block even if it resides within the same AWS account.
- VPC allows VPC Peering connections with other VPC within the same or different AWS accounts.

IP Addresses

Instances launched in the VPC can have Private, Public and Elastic IP address assigned to it and are properties of ENI (Network Interfaces).

- **Private IP Addresses -** Are not reachable over the Internet, and can be used for communication only between the instances within the VPC. All instances are assigned a private IP address, within the IP address range of the subnet.
- Primary IP address is associated with the network interface for its lifetime, even when the instance is stopped and restarted and is released only when the instance is terminated.
- **Public IP address** – Are reachable over the Internet, and can be used for communication between instances and the Internet, or with other AWS services that have public endpoints.
- Public IP address assignment to the Instance depends if the Public IP Addressing is enabled for the Subnet. Public IP address is assigned from AWS pool of IP addresses and it is not associated with the AWS account.
- **Elastic IP address -** Elastic IP addresses are static, persistent public IP addresses which can be associated and disassociated with the instance, as required.
- Elastic IP address is allocated at VPC and owned by the account unless released.
- A Network Interface can be assigned either a Public IP or an Elastic IP. If you assign an instance, already having Public IP, an Elastic IP, the public IP is released.
- Elastic IP addresses can be moved from one instance to another, which can be within the same or different VPC within the same account.

Bastion Host

- Bastion means a structure for Fortification to protect things behind it.
- In AWS, a Bastion host (also referred to as a Jump server) can be used to securely access instances in the private subnets.
- Bastion host launched in the Public subnets would act as a primary access point from the Internet and acts as a proxy to other instances.
- Bastion host is deployed in the Public subnet and acts as a proxy or a gateway between you and your instances.
- Bastion host is a security measure that helps to reduce attack on your infrastructure and you have to concentrate to hardening a single layer.
- Bastion host allows you to login to instances in the Private subnet securely without having to store the private keys on the Bastion host (using ssh-agent forwarding or RDP gateways)
- Bastion host security can be further tightened to allow SSH/RDP access from specific trusted IPs or corporate IP ranges.
- Security for all the Instances in the private subnet should be hardened to accept SSH/RDP connections only from the Bastion host

Elastic Network Interface

- Each Instance is attached with default elastic network interface (Primary Network Interface eth0) and cannot be detached from the instance.

- ENI can include the following attributes
  - Primary private IP address
  - One or more secondary private IP addresses
  - One Elastic IP address per private IP address
  - One public IP address, which can be auto-assigned to the network interface for eth0 when you launch an instance, but only when you create a network interface for eth0 instead of using an existing ENI
  - One or more security groups
  - A MAC address
  - A source/destination check flag
  - A description

Route Tables

- Route table defines rules, termed as routes, which determine where network traffic from the subnet would be routed. Each VPC has a implicit router to route network traffic.
- Each VPC has a Main Route table, and can have multiple custom route tables created
- Each Subnet within a VPC must be associated with a single route table at a time, while a route table can have multiple subnets associated with it.
- Subnet, if not explicitly associated to a route table, is implicitly associated with the main route table.
- Every route table contains a local route that enables communication within a VPC which cannot be modified or deleted.
- Route tables needs to be updated to defined routes for Internet gateways, Virtual Private gateways, VPC Peering, VPC Endpoints, NAT Device etc.

Internet Gateways – IGW

- An Internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in the VPC and the Internet.
- IGW imposes no availability risks or bandwidth constraints on the network traffic.
- An Internet gateway serves two purposes:
  - To provide a target in the VPC route tables for Internet-routable traffic,
  - To perform network address translation (NAT) for instances that have been NOT been assigned public IP addresses.
- Enabling Internet access to an Instance requires
  - Attaching Internet gateway to the VPC.
  - Subnet should have route tables associated with the route pointing to the Internet gateway.
  - Instances should have a Public IP or Elastic IP address assigned.
  - Security groups and NACLs associated with the Instance should allow relevant traffic.

AWS NAT

- NAT device enables instances in a private subnet to connect to the Internet or other AWS services, but prevents the Internet from initiating connections with the instances.
- NAT devices do not support IPv6 traffic, use an egress-only Internet gateway instead.

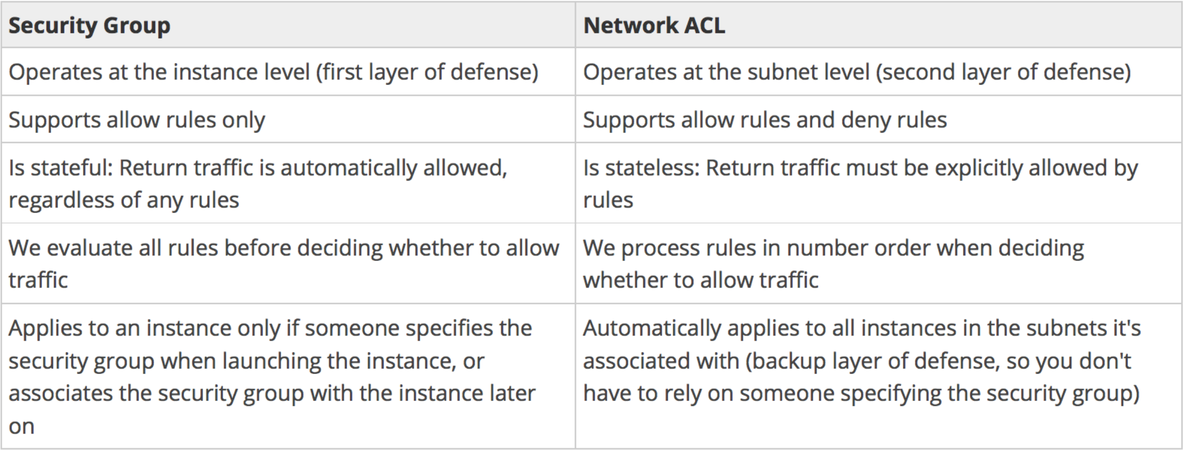
AWS Egress Only Internet Gateway

- Egress-only Internet gateway works as a NAT gateway, but for IPv6 traffic.
- Egress-only Internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows outbound communication over IPv6 from instances in the VPC to the Internet, and prevents the Internet from initiating an IPv6 connection with the instances.
- An egress-only Internet gateway is for use with IPv6 traffic only. To enable outbound-only Internet communication over IPv4, use a NAT gateway instead.

VPC Security Groups & NACL

In a VPC, both Security Groups and Network ACLs (NACLS) together help to build a layered network defence. **Security groups** – act as a firewall for associated Amazon instances, controlling both inbound and outbound traffic **at the instance level**. **Network access control lists** (NACLs) – act as a firewall for associated subnets, controlling both inbound and outbound traffic **at the subnet level**.

- Security group acts at an Instance level and not at the subnet level.
- Each instance within a subnet can be assigned a different set of Security groups.
- An instance can be assigned 5 security groups with each security group having 50 rules.
- Security groups allows you to add or remove rules (authorizing or revoking access) for both Inbound (ingress) and Outbound (egress) traffic to the instance.
  - Default Security group allows no external inbound traffic but allows inbound traffic from instances with the same security group.
  - Default Security group allows all outbound traffic.
  - New Security groups start with only an outbound rule that allows all traffic to leave the instances.
- A Network ACLs (NACLs) is an optional layer of security for the VPC that acts as a firewall for controlling traffic in and out of one or more subnets.
- NACLs are not for granular control and are assigned at a Subnet level and is applicable to all the instances in that Subnet.
- Network ACL has separate inbound and outbound rules, and each rule can either allow or deny traffic
  - Default ACL allows all inbound and outbound traffic.
  - Newly created ACL denies all inbound and outbound traffic.
- A Subnet can be assigned only 1 NACLs and if not associated explicitly would be associated implicitly with the default NACL.



Security Group	Network ACL
Operates at the instance level (first layer of defense)	Operates at the subnet level (second layer of defense)
Supports allow rules only	Supports allow rules and deny rules
Is stateful: Return traffic is automatically allowed, regardless of any rules	Is stateless: Return traffic must be explicitly allowed by rules
We evaluate all rules before deciding whether to allow traffic	We process rules in number order when deciding whether to allow traffic
Applies to an instance only if someone specifies the security group when launching the instance, or associates the security group with the instance later on	Automatically applies to all instances in the subnets it's associated with (backup layer of defense, so you don't have to rely on someone specifying the security group)

VPC Flow Logs

- VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in the VPC and can help in monitoring the traffic or troubleshooting any connectivity issues.
- Flow log data is stored using Amazon Cloud Watch Logs.
- Flow log can be created for the entire VPC, subnets or each network interface. If enabled, for entire VPC or subnet all the network interfaces are monitored.
- Flow logs do not capture real-time log streams for network interfaces.
- Flow logs can be created for network interfaces that are created by other AWS services; for example, Elastic Load Balancing, RDS, Elastic Cache, Redshift, and Work Spaces.

VPC Subnets

- Subnet spans a single Availability Zone, distinct locations engineered to be isolated from failures in other AZs, and cannot span across AZs.
- Subnet can be configured with an Internet gateway to enable communication over the Internet, or virtual private gateway (VPN) connection to enable communication with your corporate network.
- Subnet can be Public or Private and it depends on whether it has Internet connectivity i.e. is able to route traffic to the Internet through the IGW.
- Instances within the Public Subnet should be assigned a Public IP or Elastic IP address to be able to communicate with the Internet.
- For Subnets not connected to the Internet, but has traffic routed through Virtual Private Gateway only is termed as VPN-only subnet.

Shared VPCs

- VPC sharing allows multiple AWS accounts to create their application resources, such as EC2 instances, RDS databases, Redshift clusters, and AWS Lambda functions, into shared, centrally-managed VPCs.
- In this model, the account that owns the VPC (owner) shares one or more subnets with other accounts (participants) that belong to the same organization from AWS Organizations.

- After a subnet is shared, the participants can view, create, modify, and delete their application resources in the subnets shared with them. Participants cannot view, modify, or delete resources that belong to other participants or the VPC owner.

VPC Endpoints

- VPC endpoint enables creation of a private connection between VPC to supported AWS services and VPC endpoint services powered by Private Link using its private IP address.
- VPC Endpoint does not require a public IP address, access over the Internet, NAT device, a VPN connection or AWS Direct Connect.
- Endpoints are virtual devices, that are horizontally scaled, redundant, and highly available VPC components that allow communication between instances in the VPC and AWS services without imposing availability risks or bandwidth constraints on your network traffic.
- Endpoints currently do not support cross-region requests, ensure that the endpoint is created in the same region as your bucket. AWS currently supports two types of Endpoints
    - VPC Interface Endpoints
    - VPC Gateway Endpoints

VPC Peering

- A VPC peering connection is a networking connection between two VPCs that enables routing of traffic between them using private IP addresses.
- Instances in either VPC can communicate with each other as if they are within the same network.
- VPC peering connection can be established between your own VPCs, or with a VPC in another AWS account in a single different region.
- AWS uses the existing infrastructure of a VPC to create a VPC peering connection; it is neither a gateway nor a VPN connection, and does not rely on a separate piece of physical hardware.
- VPC peering connection cannot be created between VPCs that have matching or overlapping CIDR blocks.VPC peering connection are limited on the number active and pending VPC peering connections that you can have per VPC.
- Only one VPC peering connection can be established between the same two VPCs at the same time.

AWS Elastic Compute Cloud

EC2 overview & it’s features

Elastic Compute Cloud (EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. EC2 eliminates the need to invest in hardware up front, so applications can be developed and deployed faster. EC2 can be used to launch as many or as few virtual servers as you need, configure security and networking, and manage storage.

- Virtual computing environments, known as Ec2 instances.
- Preconfigured templates for your instances, known as Amazon Machine Images (AMIs), that package the bits you need for your server (including the operating system and additional software).

- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as Instance types.
- Secure login information for your instances using key pairs (AWS stores the public key, and you store the private key in a secure place).
- Storage volumes for temporary data that's deleted when you stop or terminate your instance, known as Instance store volumes.
- Persistent storage volumes for your data using Amazon EBS known as Amazon EBS volumes.
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as Regions and Availability Zones.
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using security groups.
- Static IP addresses for dynamic cloud computing, known as Elastic IP addresses.
- Metadata, known as tags, can be created and assigned to EC2 resources.
- Virtual networks you can create that are logically isolated from the rest of the AWS cloud, and that you can optionally connect to your own network, known as Virtual private clouds (VPCs).

EC2 Amazon Machine Image

- An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud is basically a template and can be used to launch as many instances as needed.
- Within a VPC, instances can be launched from as many different AMIs.
- An AMI includes the following:

  A template for the root volume for the instance for e.g. an operating system, an application server, and applications.

  Launch permissions that control which AWS accounts can use the AMI to launch instances for e.g. AWS account ids with whom the AMI is shared.

  A block device mapping that specifies the volumes to attach to the instance when it's launched.
- AMIs can be either

  AWS managed, provided and published AMIs.

  Third party or Community provided public custom AMIs.

  Private AMIs created by other AWS accounts and shared with you.

  Private and Custom AMIs created by you.
- AMI lifecycle

  After you create and register an AMI.

  You can use it to launch new instances. (You can also launch instances from an AMI if the AMI owner grants you launch permissions.)

You can copy an AMI to the same region or to different regions.

When you are finished launching instance from an AMI, you can deregister the AMI.

- Shared AMI is an AMI that can be created and shared with others for use.
- Shared AMI with all the components needed can be used to get started and then add custom components as and when needed.

EC2 AMI Type

- AMIs are specific to a region and if needed in other region must be copied over.
- AMIs can have EBS or Instance store as the root device storage.
- EBS volume are independent of the EC2 instance lifecycle and can persist independently.
- EBS backed instances can be stopped without losing the volumes.
- EBS instance can also be persist without losing the volumes on instance termination, if the Delete On Termination flag is disabled.
- EBS backed instances boot up much faster than the Instance store.

EC2 Instance Type

- EC2 Instance types determines the hardware of the host computer used for the instance.
- Each instance type offers different compute, memory, and storage capabilities and are grouped in instance families based on these capabilities.
- EC2 provides each instance with a consistent and predictable amount of CPU capacity, regardless of its underlying hardware.
- **EBS-optimized** instance uses an optimized configuration stack and provides additional, dedicated capacity for EBS I/O.
- EBS-optimized instances enable you to get consistently high performance for the EBS volumes by eliminating contention between EBS I/O and other network traffic from the instance
- **T2 instances (General Purpose)** are designed to provide moderate baseline performance and the capability to burst to significantly higher performance as required by your workload.
- Mainly intended for workloads that don’t use the full CPU often or consistently.
- T2 instances are well suited for - general purpose workloads, such as web servers, developer environments, remote desktops and small databases.
- **C4 instances** (Compute Intensive) are ideal for compute-bound applications that benefit from high performance processors.
- Well suited for

  Batch processing workloads, Media transcoding,

  High-traffic web servers, massively multiplayer online (MMO) gaming servers, and ad serving engines,

  High-traffic web servers, massively multiplayer online (MMO) gaming servers, and ad serving engines

- **I2 instances** (I/O Intensive) are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications.
- Well suited for applications

  NoSQL databases (for example, Cassandra and MongoDB)

  Clustered databases

  Online transaction processing (OLTP) systems

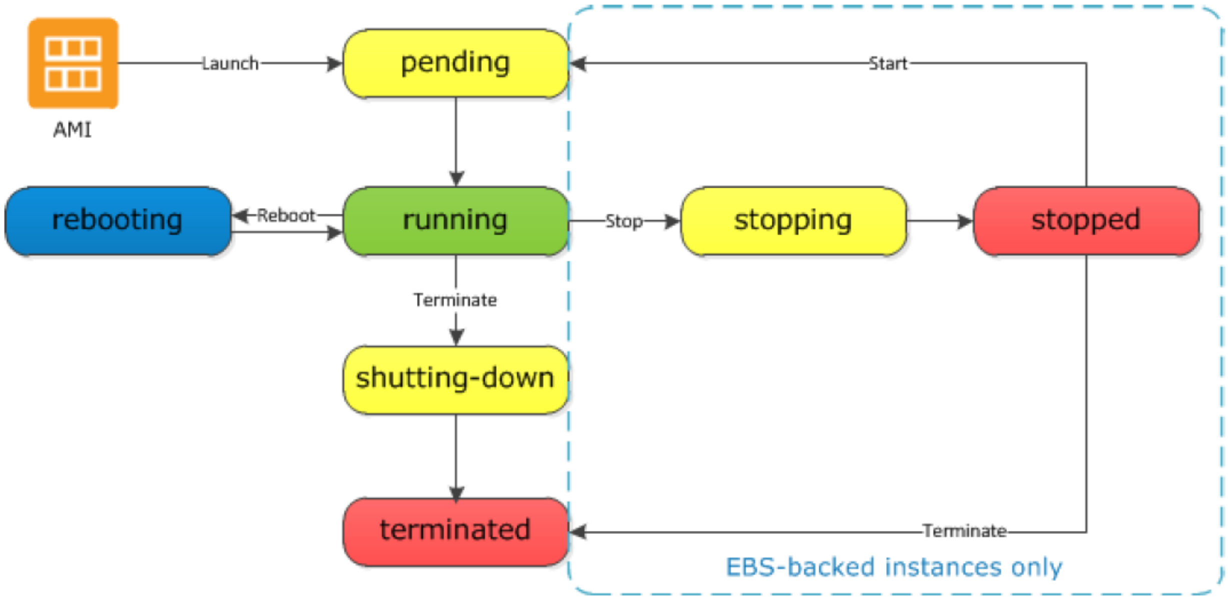
EC2 Instance Purchase Options

Amazon provides different ways to pay for the EC2 instances

- **Dedicated Instances** - Dedicated Instances are EC2 instances that run in a VPC on hardware that's dedicated to a single customer.
- Dedicated Instances are physically isolated at the host hardware level from the instances that aren't dedicated Instances and from instances that belong to other AWS accounts.
- **On-Demand Instances** - Pay for the instances and the compute capacity that you use by the hour, with no long-term commitments or up-front payments. Instances can be scaled accordingly as per the demand.
- **Reserved Instances** - Reserved Instances provides lower hourly running costs by providing a billing discount as well as capacity reservation that is applied to instances and there would never be a case of insufficient capacity from AWS.
- Discounted usage price is fixed for as long as you own the Reserved Instance, allowing compute costs prediction over the term of the reservation.
- **Spot Instances** - Spot instances enables bidding on unused EC2 instances, and are launched whenever the bid price exceeds the current market spot price.
- EC2 sets up the hourly price which fluctuates depending upon the demand and supply of spot instances. Spot instances are a cost-effective choice and can bring the EC2 costs down significantly.

EC2 Instance Lifecycle

An instance is a virtual server in the AWS cloud. You launch an instance from an Amazon Machine Image (AMI). AMI provides the operating system, application server, and applications for your instance. When an instance is launched, it receives a public DNS name that can be used to contact the instance from the Internet. Instance also receives a private DNS name that other instances within the same Amazon EC2 network can use to contact the instance.



EC2 Storage

Amazon EC2 provides flexible, cost effective and easy-to-use EC2 storage options with a unique combination of performance and durability.

- Amazon Elastic Block Store (EBS)
- Amazon EC2 Instance Store
- Amazon Simple Storage Service (S3)

While EBS and Instance store are **Block level**, Amazon S3 is an **Object level storage**.

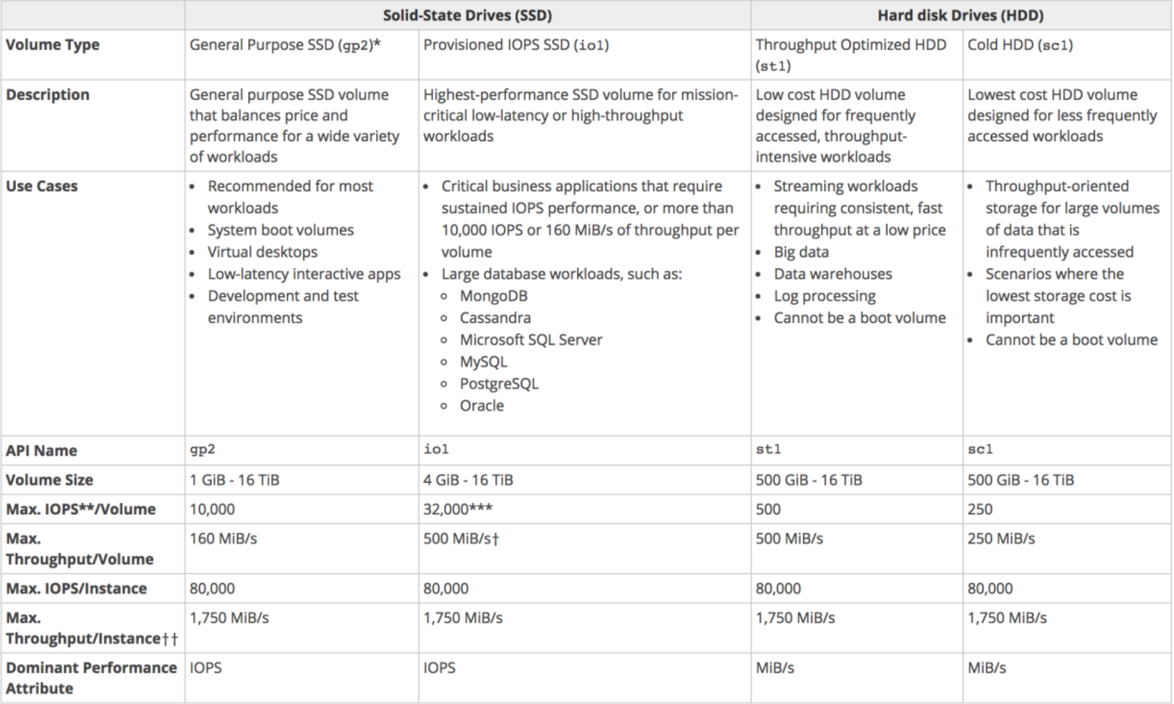
EC2 Elastic Block Storage

- Amazon EBS provides highly available, reliable, durable, block-level storage volumes that can be attached to a running instance.
- EBS as a primary storage device is recommended for data that requires frequent and granular updates for e.g. running a database or filesystems.
- An EBS volume behaves like a raw, unformatted, external block device that can be attached to a single EC2 instance at a time.
- EBS volume persists independently from the running life of an instance.
- An EBS volume can be attached to any instance within the same Availability Zone, and can be used like any other physical hard drive.
- EBS volumes allows encryption using the Amazon EBS encryption feature.
- EBS volumes are created in a specific Availability Zone, and can then be attached to any instances in that **same Availability Zone**. To make a volume available outside of the Availability Zone, create a **snapshot** and restore that snapshot to a new volume anywhere in that region.

EC2 EBS Benefits

- **Data Availability -** EBS volume is automatically replicated in an Availability Zone to prevent data loss due to failure of any single hardware component.
- **Data Persistence -** EBS volume persists independently of the running life of an EC2 instance
- EBS volume persists when an instance is stopped and started or rebooted
- **Data Encryption -** EBS volumes can be encrypted by EBS encryption feature
- Snapshots of encrypted EBS volumes are automatically encrypted
- **Snapshots -** EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to Amazon S3, where it is stored redundantly in multiple Availability Zones. Snapshots can be used to create new volumes, increase the size of the volumes or replicate data across Availability Zones.

EC2 EBS Volume Types



	Solid-State Drives (SSD)		Hard disk Drives (HDD)	
Volume Type	General Purpose SSD (gp2)*	Provisioned IOPS SSD (io1)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads	Low cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use Cases	• Recommended for most workloads • System boot volumes • Virtual desktops • Low-latency interactive apps • Development and test environments	• Critical business applications that require sustained IOPS performance, or more than 10,000 IOPS or 160 MiB/s of throughput per volume • Large database workloads, such as: ○ MongoDB ○ Cassandra ○ Microsoft SQL Server ○ MySQL ○ PostgreSQL ○ Oracle	• Streaming workloads requiring consistent, fast throughput at a low price • Big data • Data warehouses • Log processing • Cannot be a boot volume	• Throughput-oriented storage for large volumes of data that is infrequently accessed • Scenarios where the lowest storage cost is important • Cannot be a boot volume
API Name	gp2	io1	st1	sc1
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB
Max. IOPS/Volume**	10,000	32,000***	500	250
Max. Throughput/Volume	160 MiB/s	500 MiB/s†	500 MiB/s	250 MiB/s
Max. IOPS/Instance	80,000	80,000	80,000	80,000
Max. Throughput/Instance††	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s
Dominant Performance Attribute	IOPS	IOPS	MiB/s	MiB/s

EC2 EBS Snapshot

- EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to Amazon S3, where it is stored redundantly in multiple Availability Zones.
- Snapshots can be used to create new volumes, increase the size of the volumes or replicate data across Availability Zones.
- Snapshots are incremental backups and store only the data that was changed from the time the last snapshot was taken.
- Snapshots size can probably be smaller than the volume size as the data is compressed before being saved to S3.

- Snapshots can be created from EBS volumes periodically and are point-in-time snapshots.
- Snapshots are constrained to the region in which they are created and can be used to launch EBS volumes within the same region only.

References

- ✓ https://en.wikipedia.org/wiki/Amazon_Web_Services
- ✓ https://www.sumologic.com/insight/aws/
- ✓ https://www.tutorialspoint.com/amazon_web_services/index.htm
- ✓ http://jayendrapatil.com/
- ✓ https://d1.awsstatic.com/whitepapers/AWS_Cloud_Best_Practices.pdf
- ✓ http://aws.amazon.com/ec2
- ✓ https://aws.amazon.com/iam/
- ✓ https://aws.amazon.com/vpc/