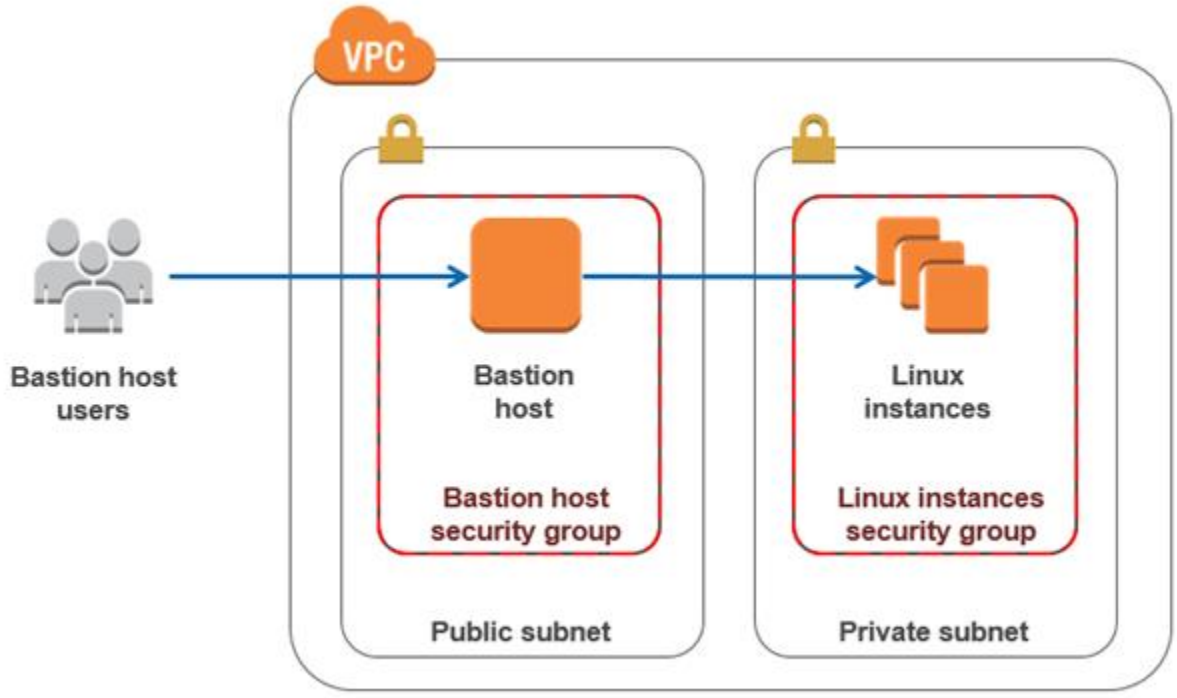
Bastion Host

What is a Bastion Host?

A Bastion Host is a special purpose computer on a host designed and configured to withstand attacks.

The computer hosts a single application, for example, a proxy server and all the other services are removed to reduce the threat to the computer.

A Bastion host is hardened due to its location and purpose, which is either on the outside of a firewall or demilitarized zone, i.e., public subnet and it usually accesses from untrusted networks or computers.



AWS PrivateLink and VPC endpoints

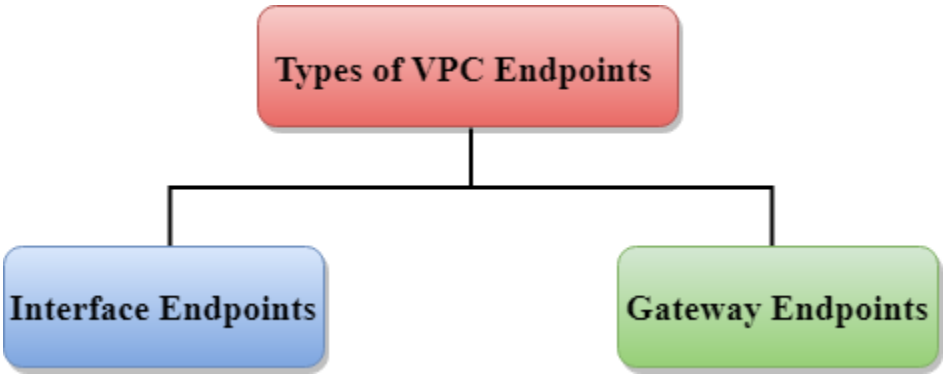
VPC Endpoint

- A VPC endpoint allows you to privately connect your VPC to supported AWS services and VPC endpoint services powered by PrivateLink without requiring an internet gateway, NAT device, VPN Connection, or AWS Direct Connect connection.
- Instances in your VPC do not require public addresses to communicate with the resources in the service. Traffic between your VPC and the other service does not leave the Amazon network.
- VPC endpoints are virtual devices.
- VPC Endpoints are horizontally scaled, redundant and highly available VPC components that allow communication between instances in your VPC and services without imposing availability risks or bandwidth constraints on your network traffic.

Types of VPC Endpoints

Interface Endpoints

Gateway Endpoints



VPC FlowLog

What is a VPC FlowLog?

- VPC FlowLog is a feature of aws that captures the information about the IP traffic going to or from the network interfaces in a VPC.
- Amazon FlowLog data can be either stored either by using the Amazon CloudWatchLogs or Amazon S3 bucket.
- After you have created a FlowLog, you can view and retrieve the data from the Amazon CloudWatch Logs.
- In short, we can say that VPC FlowLog is a way of storing the traffic going in a VPC.
- FlowLogs serve a number of purposes:
    - Troubleshoot the problem "why specific traffic is not reaching an instance".
    - VPC FlowLog can also be used as a security tool to monitor the traffic which is reaching your instance.

Limitations of VPC FlowLog:

- You cannot enable the flowlog of VPC that are peered with your VPC unless it has peered with the VPC in the same account.
- While creating a flowlog, you cannot tag a flowlog.
- Once you have created the flowlog, you cannot change its configuration. For example, if you associate an IAM role to the flowlog then you cannot change the IAM role. In such cases, you need to delete the flowlog and create the new flowlog with the desired configuration.

VPC FlowLogs can be created at three levels:

- VPC
- Subnet
- Network Interface Level

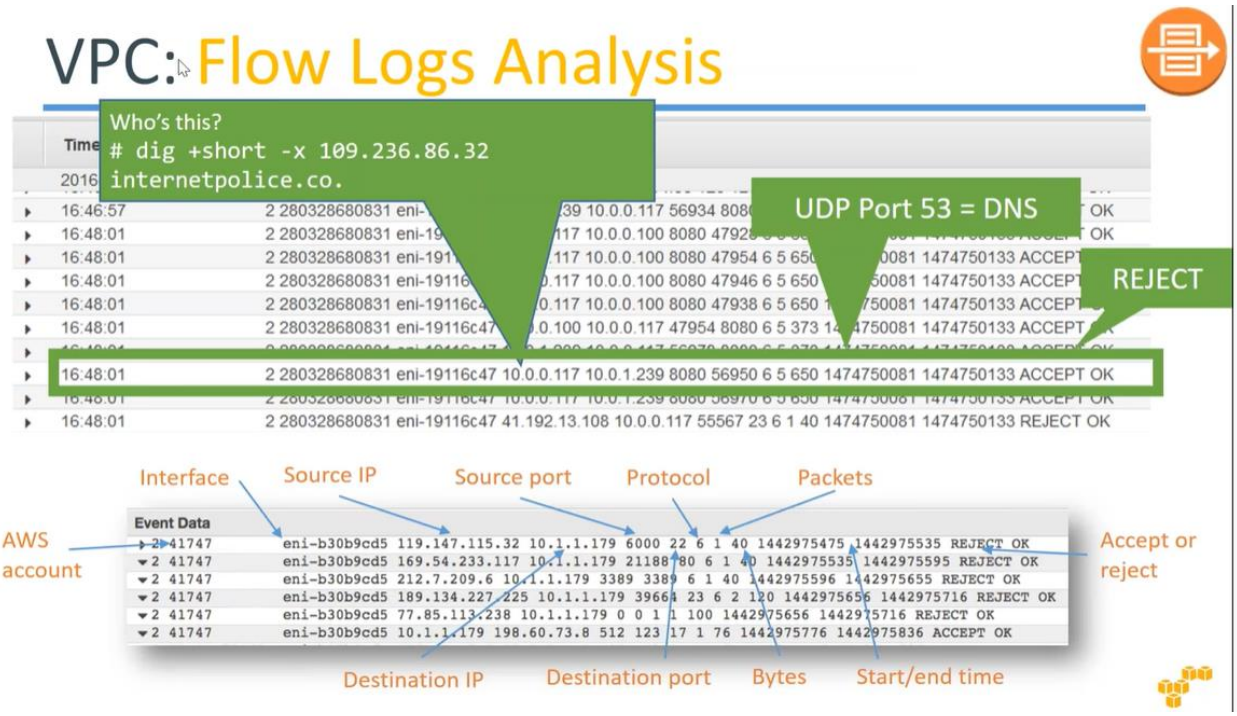
Hands on :

Create a cloud watch log group

Create IAM role for mapping cloudwatch with VPC flow log

Create VPC flow log

VPC: Flow Logs Analysis

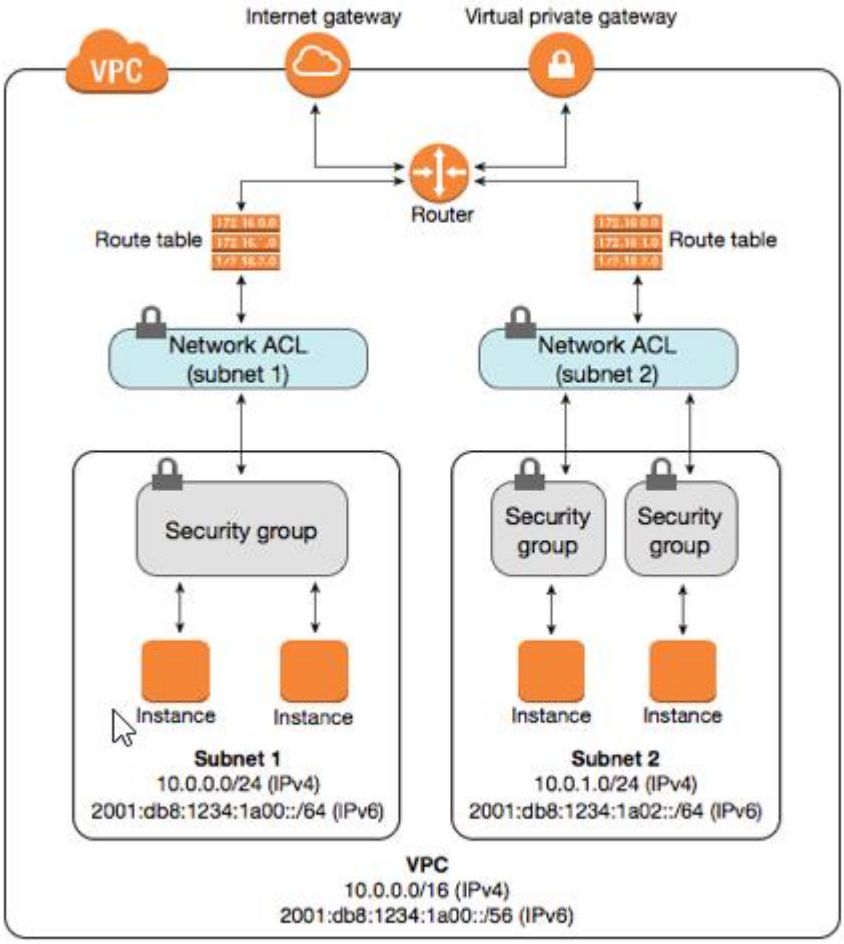


NACL

NACL refers to Network Access Control List, which helps provide a layer of security to the Amazon Web Services stack.

NACL helps in providing a firewall thereby helping secure the VPCs and subnets. It helps provide a security layer which controls and efficiently manages the traffic that moves around in the subnets. It is an optional layer for VPC, which adds another security layer to the Amazon service.

VPC refers to Virtual private Cloud, which can be visualized as a container that stores subnets. Subnets can be considered as a container, which helps store data.



Security Groups and Network ACLs

Components of NACL

Following are the components of Network Access Control List (NACL):

- Rule number: Every rule is assigned a unique number. The rule's priority is also based on the number it is assigned. When it matches to a specific request or traffic, this rule is applied to the request, irrespective of whether another high-numbered rule contradicts it or not.

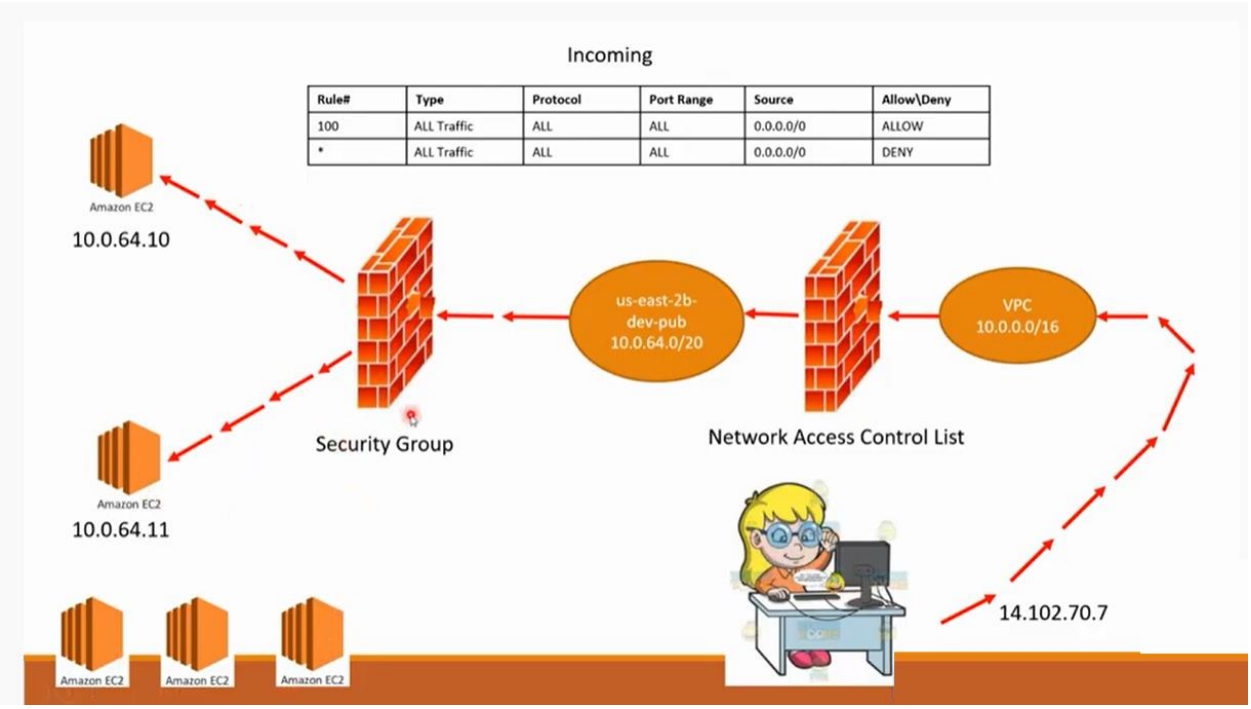
Rules are created with specific increments, like the difference between 2 rules is either 1,10, 100 and all the rules created have this same difference.

- Type: This tells about the type of traffic, like SSH, HTTP, HTTPS.
- Protocol: Protocol is a set of rules, that is applied to every request, ex: http, https, ICMP, SSH.
- Portrange: The listening port, which takes in the request from the user, such as HTTP is associated with port 80.
- Inboundrules: Also known as source. These rules talk about the source from where the request or traffic is coming from, and about the destination port/ the port through which the response is sent.
- Outboundrules: Also known as destination. These rules talk about where the response should be sent and about the destination port.
- Allow/Deny: Whether the specific traffic has to be allowed or denied.

Rule#	Type	Protocol	Port Range	Source	Allow\Deny
10	Custom TCP Rule	RDP	3389	172.16.2.10	Deny
11	Custom TCP Rule	RDP	3389	172.16.2.10	Allow
12	Custom TCP Rule	HTTP	80	172.16.2.10	Allow
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

There are two types of NaCl:

1. Customized NACL: It can also be understood as a user-defined NACL, and its inherent characteristic is to deny any incoming and outgoing traffic until a rule is added to handle the traffic.
2. Default NACL: This is the opposite of customized NACL, which allows all the traffic to flow in and out of the network. It also comes with a specific rule which is associated with a rule number, and it can't be modified or deleted. When the request doesn't match with its associated rule, the access to it is denied. When a rule is added or removed, changes are automatically applied to the subnets which are associated with it.



Security Groups & NACL Comparison

Security Groups	Network Access Control List
Operates at the instance level.	Operates at the subnet level
Supports only allow rule	It supports both allow and deny rule.
Is stateful: Return traffic is automatically allowed, regardless of any rules	Is stateless: Return traffic must be explicitly allowed by rules
It evaluate all rules before deciding whether to allow traffic.	It process rules in number order when deciding whether to allow traffic.
It apply on instance level.	It apply on subnet and all instances under same subnet.

Amazon S3

S3 is one of the first services that has been produced by aws.

S3 stands for Simple Storage Service.

S3 provides developers and IT teams with secure, durable, highly scalable object storage.

It is easy to use with a simple web services interface to store and retrieve any amount of data from anywhere on the web.

What is S3?

S3 is a safe place to store the files.

It is Object-based storage, i.e., you can store the images, word files, pdf files, etc.

The files which are stored in S3 can be from 0 Bytes to 5 TB.

It has unlimited storage means that you can store the data as much you want.

Files are stored in Bucket. A bucket is like a folder available in S3 that stores the files.

S3 is a universal namespace, i.e., the names must be unique globally. Bucket contains a DNS address. Therefore, the bucket must contain a unique name to generate a unique DNS address.

If you upload a file to S3 bucket, then you will receive an HTTP 200 code means that the uploading of a file is successful.

Advantages of Amazon S3

Create Buckets: Firstly, we create a bucket and provide a name to the bucket. Buckets are the containers in S3 that stores the data. Buckets must have a unique name to generate a unique DNS address.

Storing data in buckets: Bucket can be used to store an infinite amount of data. You can upload the files as much you want into an Amazon S3 bucket, i.e., there is no maximum limit to store the files. Each object can contain upto 5 TB of data. Each object can be stored and retrieved by using a unique developer assigned-key.

Download data: You can also download your data from a bucket and can also give permission to others to download the same data. You can download the data at any time whenever you want.

Permissions: You can also grant or deny access to others who want to download or upload the data from your Amazon S3 bucket. Authentication mechanism keeps the data secure from unauthorized access.

Standard interfaces: S3 is used with the standard interfaces REST and SOAP interfaces which are designed in such a way that they can work with any development toolkit.

Security: Amazon S3 offers security features by protecting unauthorized users from accessing your data.

S3 is a simple key-value store

S3 is object-based. Objects consist of the following:

Key: It is simply the name of the object. For example, hello.txt, spreadsheet.xlsx, etc. You can use the key to retrieve the object.

Value: It is simply the data which is made up of a sequence of bytes. It is actually a data inside the file.

Version ID: Version ID uniquely identifies the object. It is a string generated by S3 when you add an object to the S3 bucket.

Metadata: It is the data about data that you are storing. A set of a name-value pair with which you can store the information regarding an object. Metadata can be assigned to the objects in Amazon S3 bucket.

Sub resources: Sub resource mechanism is used to store object-specific information.

Access control information: You can put the permissions individually on your files.

Buckets

A bucket is a container used for storing the objects.

Every object is incorporated in a bucket.

For example, if the object named photos/tree.jpg is stored in the treeimage bucket, then it can be addressed by using the URL http://treeimage.s3.amazonaws.com/photos/tree.jpg.

A bucket has no limit to the amount of objects that it can store. No bucket can exist inside of other buckets.

S3 performance remains the same regardless of how many buckets have been created.

The AWS user that creates a bucket owns it, and no other AWS user cannot own it. Therefore, we can say that the ownership of a bucket is not transferrable.

The AWS account that creates a bucket can delete a bucket, but no other AWS user can delete the bucket.

Objects

Objects are the entities which are stored in an S3 bucket.

An object consists of object data and metadata where metadata is a set of name-value pair that describes the data.

An object consists of some default metadata such as date last modified, and standard HTTP metadata, such as Content type. Custom metadata can also be specified at the time of storing an object.

It is uniquely identified within a bucket by key and version ID.

Key

A key is a unique identifier for an object.

Every object in a bucket is associated with one key.

An object can be uniquely identified by using a combination of bucket name, the key, and optionally version ID.

For example, in the URL http://jtp.s3.amazonaws.com/2019-01-31/Amazons3.wsdl where "jtp" is the bucket name, and key is "2019-01-31/Amazons3.wsdl"

Regions

You can choose a geographical region in which you want to store the buckets that you have created.

A region is chosen in such a way that it optimizes the latency, minimize costs or address regulatory requirements.

Objects will not leave the region unless you explicitly transfer the objects to another region.

Data Consistency Model

Amazon S3 replicates the data to multiple servers to achieve high availability.

Two types of model:

Read-after-write consistency for PUTS of new objects.

For a PUT request, S3 stores the data across multiple servers to achieve high availability.

A process stores an object to S3 and will be immediately available to read the object.

A process stores a new object to S3, it will immediately list the keys within the bucket.

It does not take time for propagation, the changes are reflected immediately.

Eventual consistency for overwrite PUTS and DELETES

For PUTS and DELETES to objects, the changes are reflected eventually, and they are not available immediately.

If the process replaces an existing object with the new object, you try to read it immediately. Until the change is fully propagated, the S3 might return prior data.

If the process deletes an existing object, immediately try to read it. Until the change is fully propagated, the S3 might return the deleted data.

If the process deletes an existing object, immediately list all the keys within the bucket. Until the change is fully propagated, the S3 might return the list of the deleted key.

Creating an S3 Bucket

Important points to remember

Buckets are a universal namespace, i.e., the bucket names must be unique.

If uploading of an object to S3 bucket is successful, we receive a HTTP 200 code.

S3, S3-IA, S3 Reduced Redundancy Storage are the storage classes.

Encryption is of two types, i.e., Client Side Encryption and Server Side Encryption

Access to the buckets can be controlled by using either ACL (Access Control List) or bucket policies.

By default buckets are private and all the objects stored in a bucket are also private.

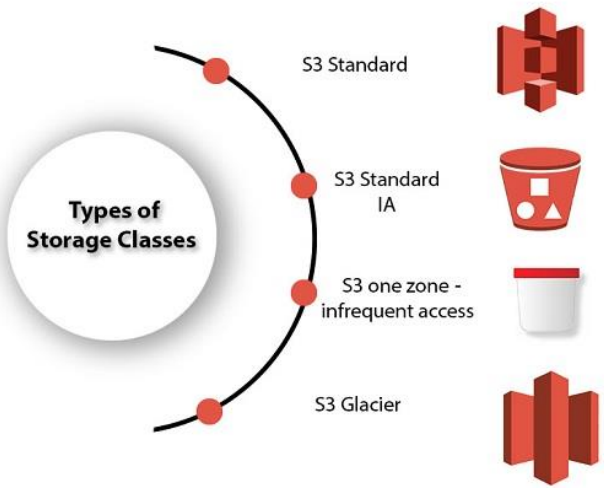
AWS Storage Classes

S3 storage classes are used to assist the concurrent loss of data in one or two facilities.

S3 storage classes maintain the integrity of the data using checksums.

S3 provides lifecycle management for the automatic migration of objects for cost savings.

S3 contains four types of storage classes:



S3 Standard

Standard storage class stores the data redundantly across multiple devices in multiple facilities.

It is designed to sustain the loss of 2 facilities concurrently.

Standard is a default storage class if none of the storage class is specified during upload.

It provides low latency and high throughput performance.

It designed for 99.99% availability and 99.999999999% durability

S3 Standard IA

IA stands for infrequently accessed.

Standard IA storage class is used when data is accessed less frequently but requires rapid access when needed.

It has a lower fee than S3, but you will be charged for a retrieval fee.

It is designed to sustain the loss of 2 facilities concurrently.

It is mainly used for larger objects greater than 128 KB kept for atleast 30 days.

It provides low latency and high throughput performance.

It designed for 99.99% availability and 99.999999999% durability

S3 one zone-infrequent access storage class is used when data is accessed less frequently but requires rapid access when needed.

It stores the data in a single availability zone while other storage classes store the data in a minimum of three availability zones. Due to this reason, its cost is 20% less than Standard IA storage class.

It is an optimal choice for the less frequently accessed data but does not require the availability of Standard or Standard IA storage class.

It is a good choice for storing the backup data.

It is cost-effective storage which is replicated from other AWS region using S3 Cross Region replication.

It has the same durability, high performance, and low latency, with a low storage price and low retrieval fee.

It designed for 99.5% availability and 99.999999999% durability of objects in a single availability zone.

It provides lifecycle management for the automatic migration of objects to other S3 storage classes.

The data can be lost at the time of the destruction of an availability zone as it stores the data in a single availability zone.

S3 Glacier storage class is the cheapest storage class, but it can be used for archive only.

You can store any amount of data at a lower cost than other storage classes.

S3 Glacier provides three types of models:

Expedited: In this model, data is stored for a few minutes, and it has a very higher fee.

Standard: The retrieval time of the standard model is 3 to 5 hours.

Bulk: The retrieval time of the bulk model is 5 to 12 hours.

You can upload the objects directly to the S3 Glacier.

It is designed for 99.999999999% durability of objects across multiple availability zones.

Performance across the Storage classes

	S3 Standard	S3 Standard IA	S3 One Zone-IA	S3 Glacier
Designed for durability	99.99999999%	99.99999999%	99.99999999%	99.99999999%
Designed for availability	99.99%	99.9%	99.5%	N/A
Availability SLA	99.9%	99%	99%	N/A
Availability zones	>=3	>=3	1	>=3
Minimum capacity charge per object	N/A	128KB	128KB	40KB
Minimum storage duration charge	N/A	30 days	30 days	90 days
Retrieval fee	N/A	per GB retrieved	per GB retrieved	per GB retrieved
First byte latency	milliseconds	milliseconds	milliseconds	Select minutes or hours
Storage type	Object	Object	Object	Object
Lifecycle transitions	Yes	Yes	Yes	Yes

Versioning

Versioning is a means of keeping the multiple forms of an object in the same S3 bucket. Versioning can be used to retrieve, preserve and restore every version of an object in S3 bucket.

For example, bucket consists of two objects with the same key but with different version ID's such as photo.jpg (version ID is 11) and photo.jpg (version ID is 12).

Versioning-enabled buckets allow you to recover the objects from the deletion or overwrite. It serves two purposes:

If you delete an object, instead of deleting the object permanently, it creates a delete marker which becomes a current version of an object.

If you overwrite an object, it creates a new version of the object and also restores the previous version of the object.

Note: Once you enable the versioning of a bucket, then it cannot be disabled. You can suspend the versioning.

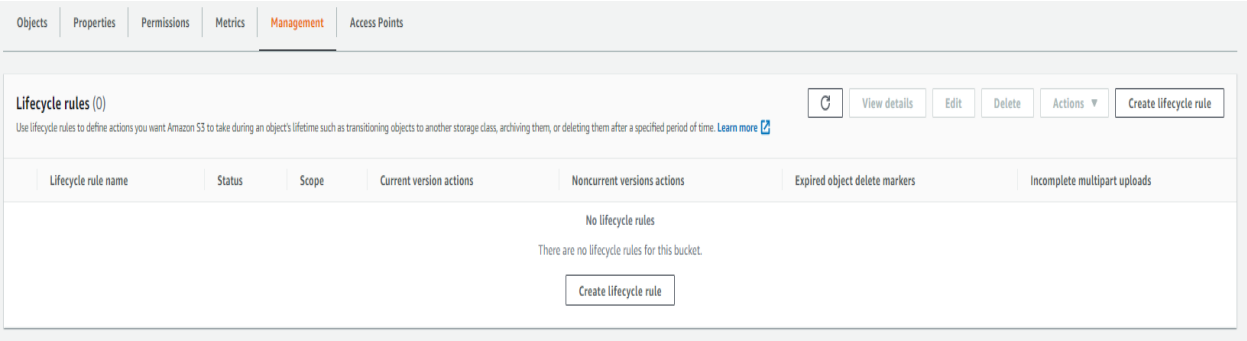
Versioning state can be applied to all the objects in a bucket. Once the versioning state is enabled, all the objects in a bucket will remain versioned, and they are provided with the unique version ID. Following are the important points:

If the versioning state is not enabled, then the version ID of the objects is set to null. When the versioning is not enabled, existing objects are not changed or are not affected.

The bucket owner can suspend the versioning to stop the object versions. When you suspend the versioning, existing objects are not affected.

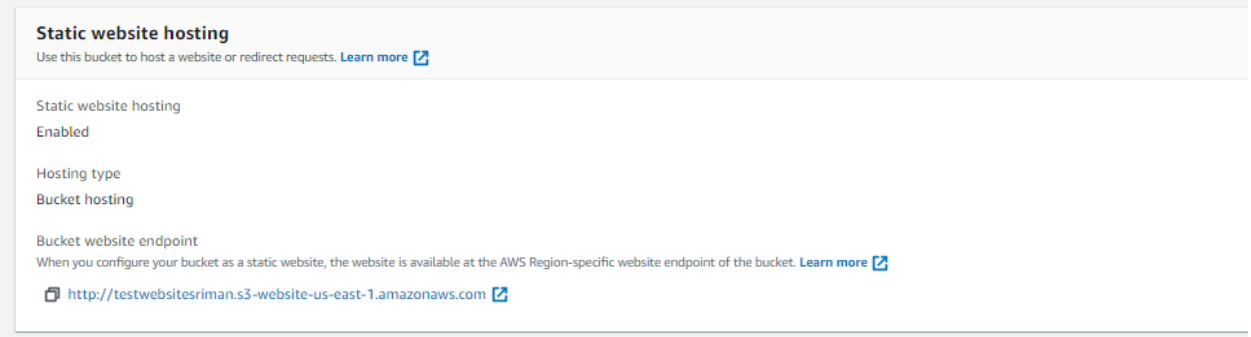
S3 Life cycle Management

First transition: 30 days after the creation of an object, object's storage class is converted to Standard Infrequently access storage class. Second transition: 60 days after the creation of an object, object's storage class is converted to Glacier storage class.



Static Website Hosting

On a static website, individual webpages include static content. ... They might also contain client-side scripts. By contrast, a dynamic website relies on server-side processing, including server-side scripts such as PHP, JSP, or ASP.NET.



S3 CORS

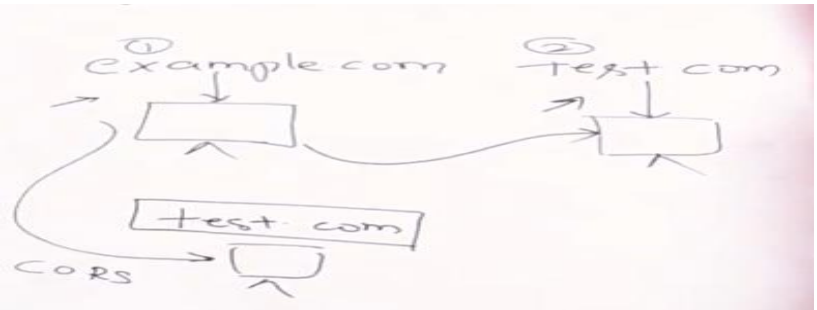
C - Cross

O - Origin

R - Resource

S – Sharing

Cross-origin resource sharing (CORS) defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. With CORS support, you can build rich client-side web applications with Amazon S3 and selectively allow cross-origin access to your Amazon S3 resources.



Scenario

AWS IAM

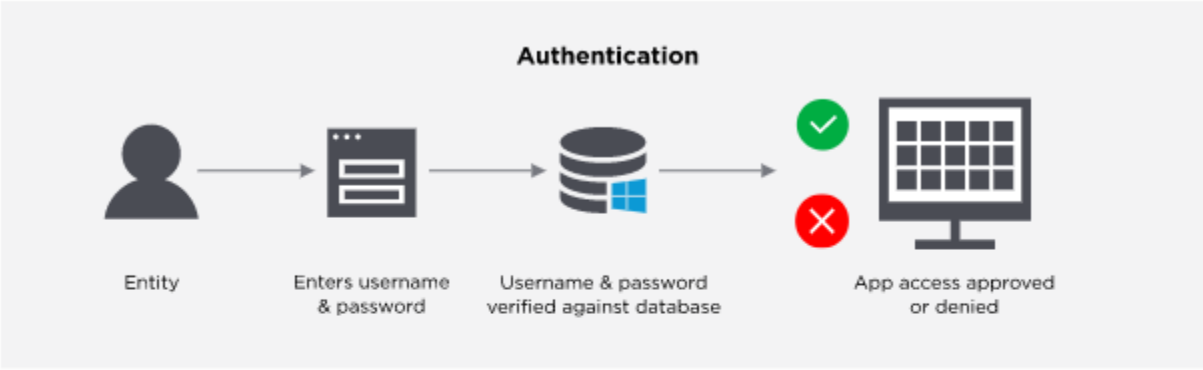
What is IAM?

- IAM stands for Identity Access Management.
- IAM allows you to manage users and their level of access to the aws console.
- It is used to set users, permissions and roles. It allows you to grant access to the different parts of the aws platform.
- AWS Identity and Access Management is a web service that enables Amazon Web Services (AWS) customers to manage users and user permissions in AWS.
- With IAM, Organizations can centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users can access.
- Without IAM, Organizations with multiple users must either create multiple user accounts, each with its own billing and subscriptions to AWS products or share an account with a single security credential. Without IAM, you also don't have control about the tasks that the users can do.
- IAM enables the organization to create multiple users, each with its own security credentials, controlled and billed to a single aws account. IAM allows the user to do only what they need to do as a part of the user's job.

Features of IAM

- Centralised control of your AWS account: You can control creation, rotation, and cancellation of each user's security credentials. You can also control what data in the aws system users can access and how they can access.
- Shared Access to your AWS account: Users can share the resources for the collaborative projects.
- Granular permissions: It is used to set a permission that user can use a particular service but not other services.
- Multifactor Authentication: An AWS provides multifactor authentication as we need to enter the username, password, and security check code to log in to the AWS Management Console.
- Permissions based on Organizational groups: Users can be restricted to the AWS access based on their job duties, for example, admin, developer, etc.
- Networking controls: IAM also ensures that the users can access the AWS resources within the organization's corporate network.
- Provide temporary access for users/devices and services where necessary: If you are using a mobile app and storing the data in AWS account, you can do this only when you are using temporary access.
- Integrates with many different aws services: IAM is integrated with many different aws services.

- Eventually Consistent: IAM service is eventually consistent as it achieves high availability by replicating the data across multiple servers within the Amazon's data center around the world.
- Free to use: AWS IAM is a feature of AWS account which is offered at no additional charge. You will be charged only when you access other AWS services by using IAM user.

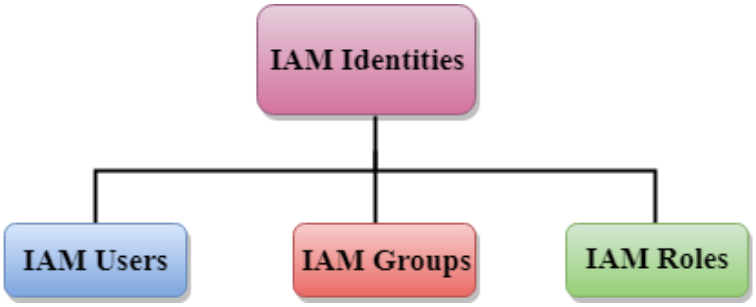


AWS Account Root User

- When you first create an AWS account, you create an account as a root user identity which is used to sign in to AWS.
- You can sign to the AWS Management Console by entering your email address and password. The combination of email address and password is known as root user credentials.
- When you sign in to AWS account as a root user, you have unrestricted access to all the resources in AWS account.
- The Root user can also access the billing information as well as can change the password also.
- IAM Identities
- IAM identities are created to provide authentication for people and processes in your aws account.

IAM identities are categorized as given below:

- IAM Users
- IAM Groups
- IAM Policies
- IAM Roles



Users

An IAM user is an identity with an associated credential and permissions attached to it. This could be an actual person who is a user, or it could be an application that is a user. With IAM, you can securely manage access to AWS services by creating an IAM user name for each employee in your organization. Each IAM user is associated with only one AWS account. By default, a newly created user is not authorized to perform any action in AWS. The advantage of having one-to-one user specification is that you can individually assign permissions to each user.

Practical:

We will create a user

We will only S3 bucket creation access

We will login as user and try to create bucket

We will delete the user

Policies

An IAM policy sets permission and controls access to AWS resources. Policies are stored in AWS as JSON documents. Permissions specify who has access to the resources and what actions they can perform. For example, a policy could allow an IAM user to access one of the buckets in Amazon S3.

The policy would contain the following information:

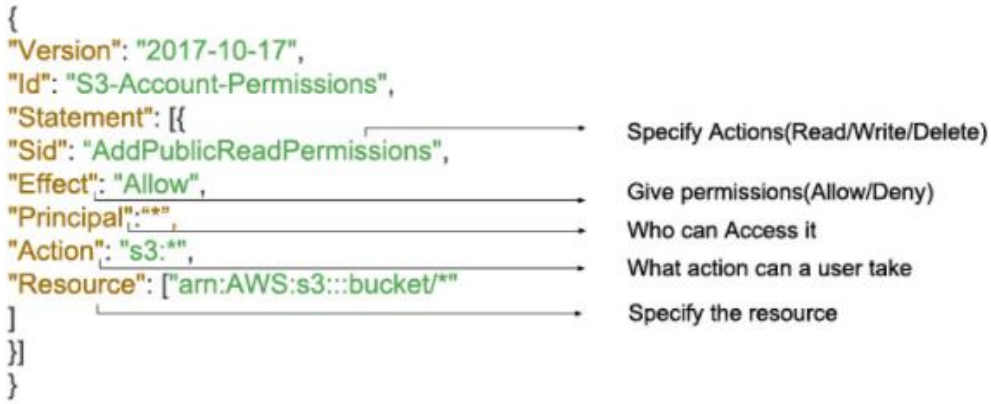
1. Who can access it

2. What actions that user can take
3. Which AWS resources that user can access
4. When they can be accessed

In JSON format that would look like this:

There are two types of policies: managed policies and inline policies.

1. A managed policy is a default policy that you attach to multiple entities (users, groups, and roles) in your AWS account. Managed policies, whether they are AWS-managed or customer-managed, are stand-alone identity-based policies attached to multiple users and/or groups.
2. Inline policies are policies that you create that are embedded directly into a single entity (user, group or role).



Groups

A collection of IAM users is an IAM group. You can use IAM groups to specify permissions for multiple users so that any permissions applied to the group are applied to the individual users in that group as well. Managing groups is quite easy. You set permissions for the group, and those permissions are automatically applied to all the users in the group. If you add another user to the group, the new user will automatically inherit all the policies and the permissions already assigned to that group. This lessens the administrative burden.

Practical:

We will create 2 users

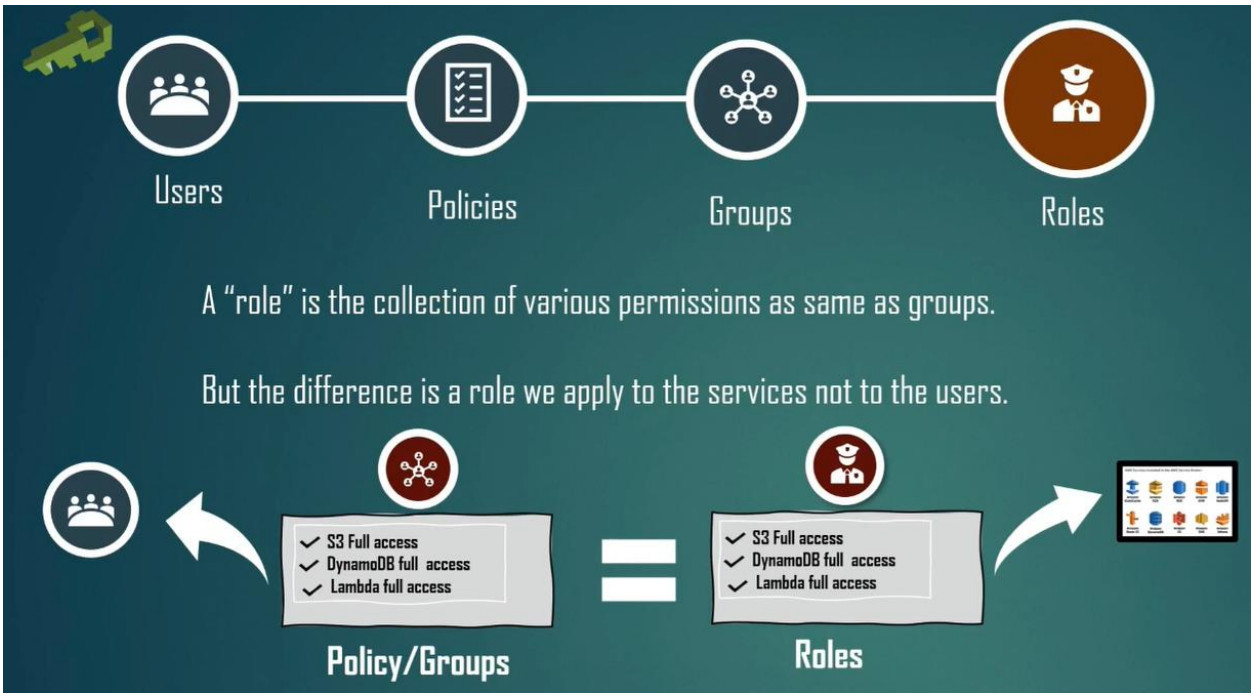
We will create a group

We will add 2 users to that group

We will provide S3 bucket creation access to that group

Roles

An IAM role is a set of permissions that define what actions are allowed and denied by an entity in the AWS console. It is similar to a user in that it can be accessed by any type of entity (an individual or AWS service). Role permissions are temporary credentials.



Practical:

- Create EC2 role with S3 permission
- Create two instances and install AWS CLI
- Attach IAM role for 1 instance, leave other instance as it is:

Install PIP:

curl -O https://bootstrap.pypa.io/get-pip.py

python3 get-pip.py --user

ls -a ~

export PATH=~/.local/bin:$PATH

source ~/.bash_profile

pip3 --version

Install and update the AWS CLI version 1 using pip

$ pip3 install awscli --upgrade --user

For a specific version of the AWS CLI, append a less-than symbol < and the version number to the filename. For this example the filename for version 1.16.312 would be <1.16.312 resulting in the following command:

$ aws --version

aws-cli/1.22.7 Python/3.8.8 Linux/4.14.133-113.105.amzn2.x86_64 botocore/1.13

- Practical:
- Create an upload file1 from both instances
- aws s3 cp /home/ec2-user/file1 s3://testsrimanit

Status Checks

Status checks will get performed each and every minute, while returning a status of either a pass or a fail. In case every single one of the checks pass, the final status of the instance is going to be OK. In case 1 or more checks tend to fail, the final status will be impaired. Status checks will be built into Amazon EC2, which means that they are not capable of being disabled or deleted.

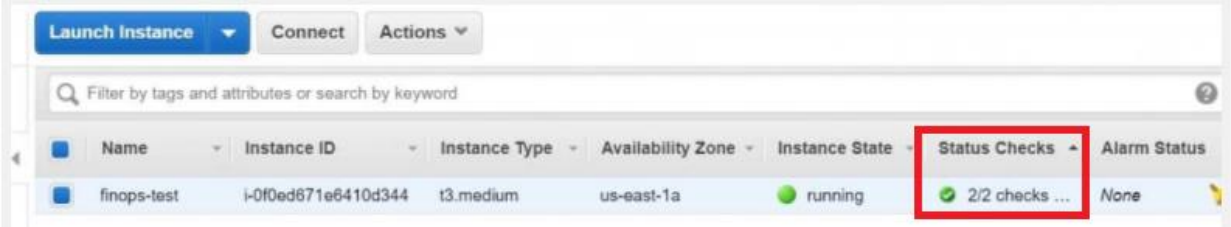
Status Checks types:

You can find that there are 2 types of status checks:

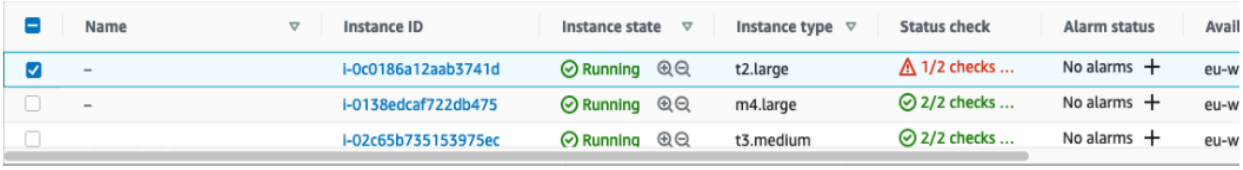
System status checks

Instance status checks

Good:



Need to check:



System Status Checks Work?

- To monitor the AWS systems where your instance tends to be running.

- Those checks can detect underlying problems found within your instance and need the AWS involvement for getting it repaired.

- The moment a system status check tends to fail, you get the possibility to either go with waiting for AWS till it fixes the issue, or you can simply go ahead and resolve it on your own. Instances that are backed by Amazon EBS, are capable of being stopped and started by your own doing, and this will in a lot of cases end up in the instance getting migrated to another new host. Instances that are backed by the instance store, may be terminated and replaced by you.

The below mentioned are a couple of examples regarding problems that may result in system status checks failure:

- Network connectivity lost

- System power lost

- The physical host having software issues

- The physical host having hardware issues which affect the network's reachability

Instance Status Checks Work?

- To monitor the network and also software configuration of individual instances.

- Amazon EC2 will start checking the health of your instance through sending an address resolution protocol -ARP- request straight to the network interface (NIC).

- Those checks are going to start detecting problems which need your help in getting them repaired.

- The moment an instance status check tends to fail, you will mainly need to get the problem addressed by yourself, such as tending to reboot the instance or through performing a couple of instance configuration alterations.

The below listed show examples of hardships which may cause the failure of instance status checks:

- Failed system status checks

- Incorrect networking / startup configuration

- Overloaded memory

- File system which is Corrupt

- Kernel which is Incompatible

Amazon SNS (Simple Notification Service)

A web service that makes it easy to set up, operate, and send notifications from the cloud.

SNS is integrated into many AWS services. We are able to use it to receive notifications when events occur in our AWS Environment. With CloudWatchand SNS a full environment monitoring solution could be created that notifies administrators of alerts, capacity is sues, downtime, changes in the environment, and more!

TOPIC: A topic what a “message is sent to”

Subscription end point: SNS sends all messages to subscriptions subscribed to a specific topic

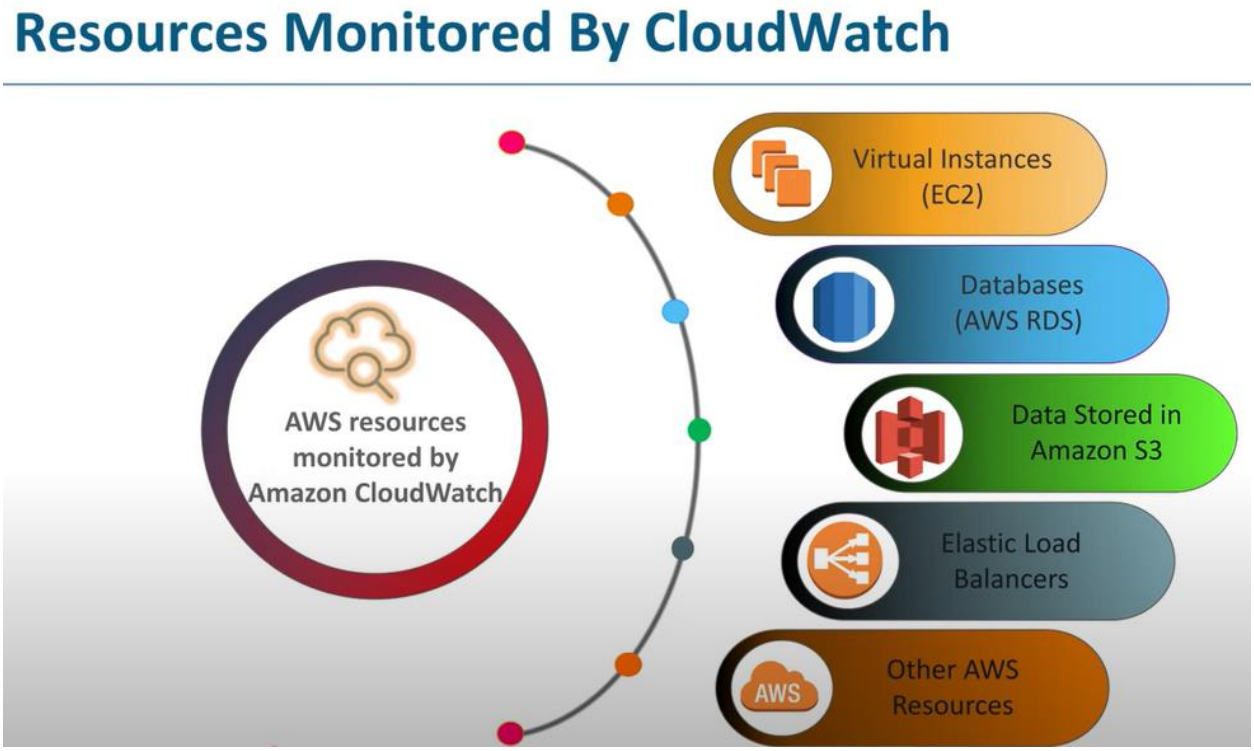
Subscriber end points include the following

- Application, Mobile APP notifications (IOS/Android/Amazon/Microsoft)
- SMS
- HTTPS
- JSON
- E-Mail JSON
- SQS Queue

- Pricing
- You pay based on the number of notifications you publish, the number of notifications you deliver, and any additional API calls for managing topics and subscriptions. Delivery pricing varies by endpoint type.
- Limits
- By default, SNS offers 10 million subscriptions per topic, and 100,000 topics per account.
- A single SMS message can contain a maximum of 140 bytes of information.
- With the exception of SMS messages, SNS messages can contain up to 256 KB of text data.
- Monitoring
- Monitoring SNS topics using CloudWatch
- Logging SNS API calls using CloudTrail

CLOUD WATCH

- Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS. You can use Amazon CloudWatch to collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes in your AWS resources. Amazon CloudWatch can monitor AWS resources such as Amazon EC2 instances, Amazon DynamoDB tables, and Amazon RDS DB instances, as well as custom metrics generated by your applications and services.
- Once you logged into AWS under Console Home page, choose CloudWatch under management tools.



Amazon CloudWatch Concepts

Metrics

A **metric** represents a time-ordered set of data points that are published to CloudWatch. Think of a metric as a variable to monitor, and the data points represent the values of that variable over time.

Dimensions

A **dimension** is a name/value pair that uniquely identifies a metric. They can be considered as categories of characteristics that describe a metric. We can assign up to 10 dimensions to a metric

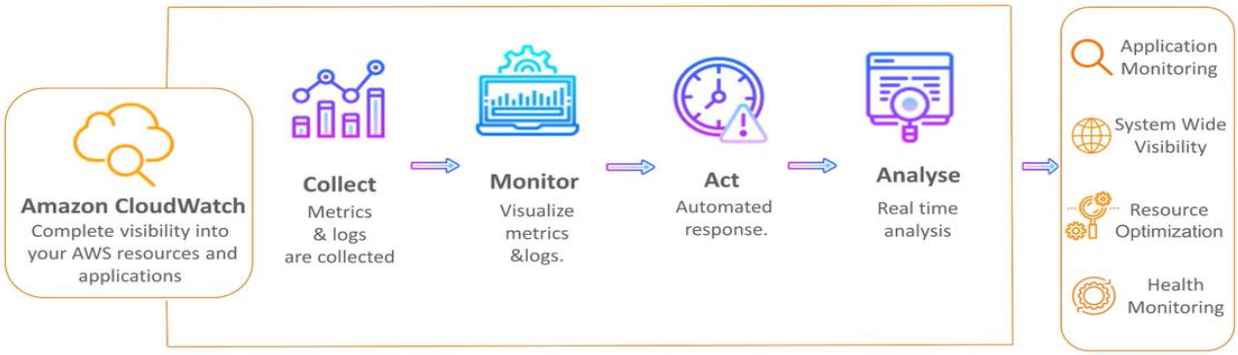
Statistics

Statistics are metric data aggregations over specified periods of time. Aggregations are made using the namespace, metric name, dimensions within the time period you specify.

Alarm

An **alarm** can be used to automatically initiate actions on your behalf. It watches a single metric over a specified time period, and performs one or more specified actions.

How Amazon CloudWatch Works?



Amazon Simple Email Service

Amazon Simple Email Service (Amazon SES) is a cloud-based email sending service designed to help digital marketers and application developers send marketing, notification, and transactional emails.

What is Amazon SES? *Bulk and transactional email-sending service*. Amazon SES eliminates the complexity and expense of building an in-house email solution or licensing, installing, and operating a third-party email service. The service integrates with other AWS services, making it easy to send emails from applications being hosted on services such as Amazon EC2.

Benefits

Integrate quickly

Using either the Amazon SES console, APIs, or SMTP, you can configure email sending in minutes. Amazon SES also supports email receiving, enabling you to interact with your customers at scale. Regardless of use case or sending volume, you only pay for what you use with Amazon SES.

Optimize your deliverability

Use the reputation dashboard, which includes account performance insights and anti-spam feedback, to maximize your deliverability. You have flexible deployment options that range from shared, dedicated, and customer-owned IPs that helps you influence your sending reputation. Amazon SES has relationships with experts like M3AAWG to improve delivery to your customers through industry best practices.

Send messages efficiently

Email sending statistics, including email deliveries, bounces, and feedback loop results, help you to measure the effectiveness of each email outreach. Additional insights like email open or click-through rates measure how engaged your customers are in your email communications.

Scale securely

Amazon SES authentication options such as Sender Policy Framework (SPF) and DomainKeys Identified Mail (DKIM) confirms your right to send on behalf of your domain. Virtual private cloud (VPC) support makes email sending from any application secure. Amazon SES is globally available with HIPAA eligibility, in-region compliance (C5, IRAP) and global certifications (Fed-Ramp, ISO, GDPR).

RDS

What Is Amazon RDS?

Amazon RDS is a service which provides database connectivity through the Internet. RDS makes it very simple and easy to set-up a relational database in the cloud.

Instead of concentrating on database features, you can concentrate more on the application to provide high availability, security, and compatibility. RDS is a fully managed RDBMS service.

Benefits of Amazon RDS

1. Reduced Administration Burden

Using RDS, you can easily deploy the database from project conception to production.

There is no need to install any database software and provide the infrastructure.

AWS automatically installs the latest software patches to the RDS instance which you have launched.

2. Cost-effective

You just pay for what you use, and nothing more. No upfront payment is needed, just the monthly usage payment.

3. Security

Using AWS Key Management Service (KMS), you can create encryption keys for maintaining security and authorized access for your database.

4. High Availability and Durability

The automated recovery feature of RDS enables point-in-time recovery for your database instance.

Multi-AZs provide high availability and durability across the globe.

5. Scalability

It just takes a few minutes to scale your infrastructure up or down, and you can scale up to a maximum of 32 vCPUs and 244 GiB.

6. Free Tier

AWS gives you a free tier usage of Amazon RDS for 750 hours/month for 12 months.

Database Engines

There are six database engines which RDS provides, and they are:

- Amazon Aurora
- PostgreSQL
- MySQL
- MariaDB
- Oracle Database
- Microsoft SQL Server

Port Numbers:

- MariaDB – 3306
- Amazon Aurora 3306
- Microsoft SQL Server – 1433
- MySQL – 3306
- Oracle – 1521
- PostgreSQL – 5432

Amazon Aurora

Amazon Aurora is a MySQL and PostgreSQL compatible relational database which is built for the cloud for providing simplicity and cost-effectiveness of open-source databases with the availability and performance of traditional enterprise databases.

- 5 times faster than standard MySQL
- 3 times faster than standard PostgreSQL

Are you preparing for AWS interview? Then here are latest AWS interview questions

Pricing

Prices vary according to the region; if you have North Virginia as the region:

- Database Storage: $0.10/GB/month
- Backup Storage: $0.021/GB/month

PostgreSQL

You can deploy scalable and highly available PostgreSQL relational databases using RDS and connect them. It only takes minutes to deploy a PostgreSQL instance. RDS PostgreSQL provides the same functionalities as the traditional PostgreSQL, so you can just dump or upload your local database into the RDS.

Pricing

- Database Storage: $0.115/GB/month
- Backup Storage: $0.095/GB/month

If you have any doubts or queries related to AWS, do post on AWS Community.

MySQL

MySQL is the world's most used and popular relational database, and Amazon RDS provides an easy way to set-up, operate, and scale. You can use the code which you are writing because RDS for MySQL covers all versions of MySQL.

Pricing

- Database Storage: $0.115/GB/month
- Backup Storage: $0.095/GB/month

MariaDB

MariaDB is created by original creators of MySQL and it is very popular in creating PHP based applications. All versions of MariaDB is covered by Amazon EDS.

Pricing

- Database Storage: $0.115/GB/month
- Backup Storage: $0.095/GB/month

Oracle

Oracle's relational database is called Oracle Database.

Pricing

- Database Storage: $0.115/GB/month
- Backup Storage: $0.095/GB/month

Microsoft SQL Server

Microsoft's relational database is called the SQL Server. Multiple editions of SQL Server (2008 R2, 2012, 2014, 2016, and 2017), including Express, Web, Standard, and Enterprise, are merged with Amazon RDS, so whatever code your application is built with can be applicable in the RDS.

Pricing

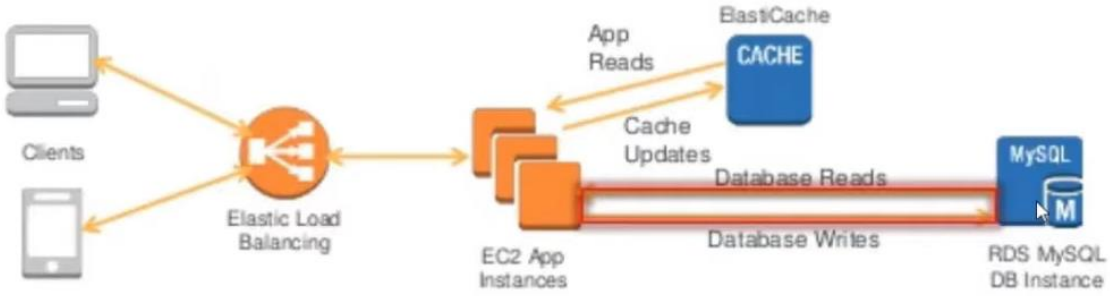
- Database Storage: $0.115/GB/month
- Backup Storage: $0.095/GB/month

But for all database engines, 'data in' from the Internet is free, only 'data out' requires a payment.

Hands-on: Creating a MySQL RDS Instance and Connecting It to Your MySQL WorkBench

Elastic Cache

ElastiCache is a distributed cache environment for providing faster access to data by using cloud-based caching. Querying for data directly from databases or through remote API calls is much slower than querying the data from cache. AWS provides ElastiCache service which has high performance, scalability and cost-effectiveness. It removes the complexities associated with managing a distributed cache.



Amazon ElastiCache Engines

When	Amazon ElastiCache Engine	
Building real-time apps across versatile use cases like gaming, geospatial service, caching, session stores, or queuing, with advanced data structures, replication, and point-in-time snapshot support. Learn how to set up your first Redis Cluster here »		Amazon ElastiCache for Redis
Building a simple, scalable caching layer for your data-intensive apps.		Amazon ElastiCache for Memcached

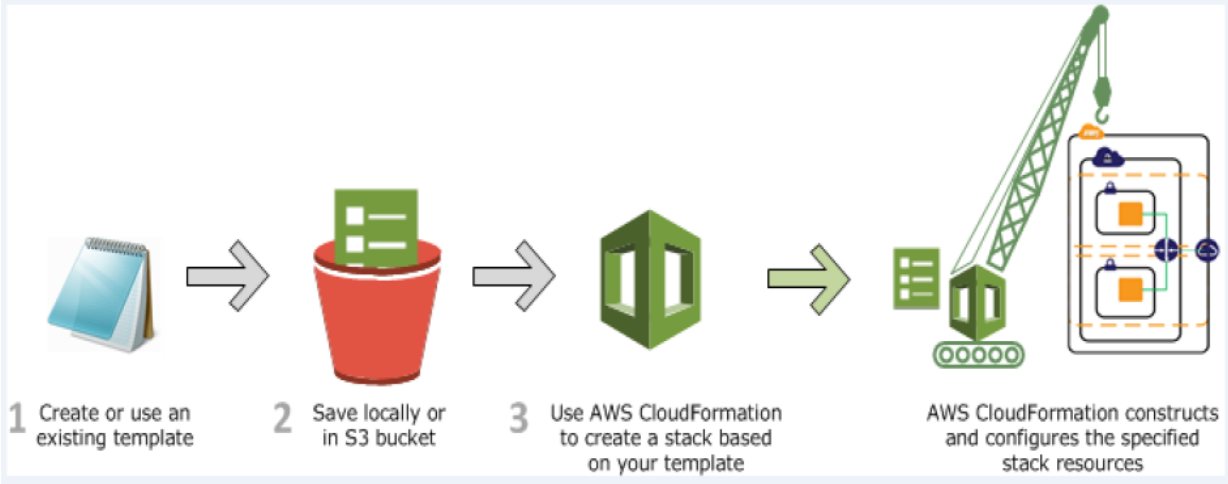
Redis Vs Memcached

Redis	Memcached
1. Support distribute your data among multiple nodes	1. Support distribute your data among multiple nodes
2. Allowing user to keep data on disk with a point in time snapshot	2. No support for allowing user to keep data on disk with a point in time snapshot
3. Allows to execute a group of commands as an isolated & atomic operation	3. No support to execute a group of commands as an isolated & atomic operation
4. It doesn't have native clustering	4. Memcached supports native clustering
5. We can cache bigger values in Redis, Values upto 512MB per key	5. Size is limited for the values cached. Values cached upto 1MB/ Key
6. Powerful data types and powerful commands to leverage them (Hashes, sorted sets, Lists and more)	6. Great for multi-threaded environmnets and it is very faster compared to Redis

AWS CloudFormation

AWS CloudFormation enables you to manage your complete infrastructure or AWS resources in a text file, or template. A collection of AWS resources is called a stack. AWS resources can be created or updated by using a stack.

All the resources you require in an application can be deployed easily using templates. Also, you can reuse your templates to replicate your infrastructure in multiple environments. To make templates reusable, use the parameters, mappings and conditions sections in the template so that you can customize your stacks when you create them.



Benefits of CloudFormation

- Deployment speed
- Scaling up
- Service integration
- Consistency
- Security
- Easy updates
- Auditing and change management

CloudFormation Template Terms and Concepts

Template

A CloudFormation template is simply a text file, formatted in a specific way (see below for details on formatting), that defines how AWS services or resources should be configured and deployed.

Stacks

A stack is a term AWS uses to refer to a collection of multiple AWS resources -- such as EC2 virtual machines, S3 storage, and IAM access controls -- that you can manage together using a single template.

Formatting

CloudFormation supports templates that are formatted using either JSON or YAML. These are widely used file formats for structuring text files. Most other IaC tools use the same formatting languages, as do platforms like Kubernetes.

Parameters

If you need to apply unique settings for each deployment, you can do so using parameters. Parameters let you define custom values for each deployment that CloudFormation will apply at runtime.

Conditions

You can also fine-tune deployments by setting conditions, which let you define conditional rules to govern precisely how each deployment proceeds.

Change sets

If you want to update a deployment using CloudFormation, you can update the template you used to create the deployment. You can then create a change set, which summarizes the changes that the updated template will apply before making the change.

Functions

There are several ways to get data into a CloudFormation template, with parameters being the primary. But those parameters may not be known at deployment time. CloudFormation Functions allow CloudFormation Designers to retrieve data from resources deployed in the current CloudFormation or from external sources in the AWS account. Ref is used extensively to reference other resources inside the template like the example below. It creates an EIP for the instance created earlier in the template.

How to Create a CloudFormation Template

There are two ways to create a CloudFormation template:

- Using a pre-existing template as the foundation or
- Writing an entirely new template from scratch
- https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/sample-templates-services-us-west-2.html#w2ab1c33c58c13c17

How to Deploy a CloudFormation Template

Once your template is ready, you can begin the deployment process to create the resources you have defined in the template in your actual AWS environment.

There are multiple ways to deploy a CloudFormation template. The approach you take will depend in part on how you created your template and in part on which AWS interface you prefer:

- The AWS console: If your template is a text file stored on your local computer, the easiest way to deploy it is to log into the AWS CloudFormation console https://console.aws.amazon.com/cloudformation and click Create New Stack. The console will walk you through the steps for naming your template and uploading your computer's template file. Once you have completed the steps and reviewed your configuration, click the Create button to deploy your template.
- From CloudFormation Designer: If you create your template in CloudFormation Designer, you can deploy it directly from there by clicking the Create Stack button, following the directions on the screen, and pressing Create when you are ready to apply the template.
- The AWS CLI: Using the AWS CLI tool, you can use the aws cloudformation deploy command to deploy a template. Use command-line arguments to define where your template is stored (typically, you would upload it to S3 first and point the CLI to that file) and other options you may want to configure. The AWS documentation provides complete details on CLI deployments.

An AWS CloudFormation template consists of nine main objects:

- Format version: Format version defines the capability of a template.
- Description: Any comments about your template can be specified in the description.
- Metadata: Metadata can be used in the template to provide further information using JSON or YAML objects.
- Parameters: Templates can be customized using parameters. Each time you create or update your stack, parameters help you give your template custom values at runtime.

- Mappings: Mapping enables you to map keys to a corresponding named value that you specify in a conditional parameter. Also, you can retrieve values in a map by using the “Fn:: FindInMap” intrinsic function.
- Conditions: In a template, conditions define whether certain resources are created or when resource properties are assigned to a value during stack creation or updating. Conditions can be used when you want to reuse the templates by creating resources in different contexts. You can use intrinsic functions to define conditions.
- Transform: Transform builds a simple declarative language for AWS CloudFormation and enables reuse of template components. Here, you can declare a single transform or multiple transforms within a template.
- Resources: Using this section, you can declare the AWS resource that you want to create and specify in the stack, such as an Amazon S3 bucket or AWS Lambda.
- Output: In a template, the output section describes the output values that you can import into other stacks or the values that are returned when you view your own stack properties. For example, for an S3 bucket name, you can declare an output and use the “Description-stacks” command from the AWS CloudFormation service to make the bucket name easier to find.

Other Important Services

Compute Services

AWS Elastic Load Balancing

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

AWS Lambda

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

AWS Auto scaling

- The Auto scaling feature is used to scale up and down automatically as and when required.

Database Services

Amazon Aurora

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

Amazon Redshift

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

Amazon Dynamo DB

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

Networking Services

VPC (Virtual Private cloud)

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

AWS Direct Connect

- It helps you establish a private connection between your premises and AWS, therefore giving better network performance.

Amazon Route 53

- Route 53 is a highly scalable and highly available Domain Name System by Amazon AWS. It effectively connects user requests to infrastructure running in AWS – such as Amazon EC2 instances, Elastic Load Balancing load balancers.

Management services

Amazon Cloud Watch

- It is a monitoring tool by AWS which is used to keep a track on the AWS resources and the applications you run on Amazon AWS.
- AWS Cloud Formation
- It is a service which helps you setup and model your Amazon AWS resources so that you can spend less time managing these resources and more time focusing on the development.

AWS CloudTrail

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

AWS Command Line Tool

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

Security and Identity

Identity and Access Management(IAM)

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

AWS Key Management Service

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

Application Services

Amazon SES

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

Amazon SNS

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

Amazon SQS

- It is a fast, reliable and scalable message queuing service, it can be used to transmit any volume of data at any level of throughput, without losing any messages or without the use of any other service.