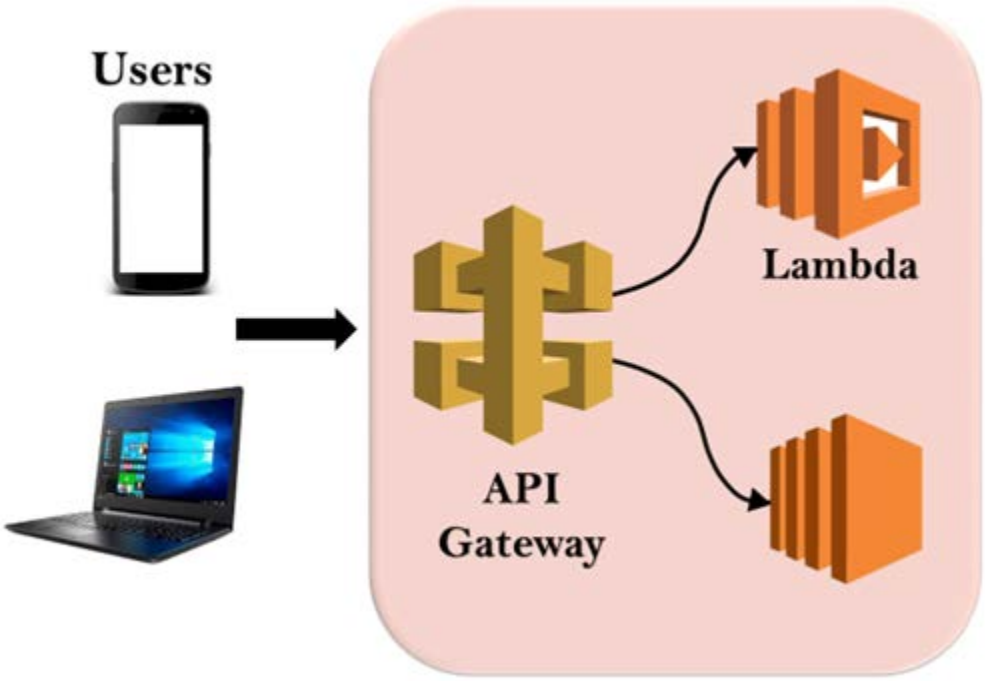
API Gateway

- API Gateway is a gateway that consists of a bunch of Lambda functions that create a serverless learning management system.
- API Gateway is a fully managed service that makes it easy for developers to publish, maintain, monitor, and secure APIs at any scale.
- With a few clicks in the AWS Management Console, you can create an API that acts as a "front door" for applications to access data, business logic, or functionality from your back-end services such as applications running on Amazon Elastic Compute Cloud (Amazon EC2), code running on AWS Lambda, or any web application.
- If your browser is making API calls to API Gateway, then API Gateway is routing down those calls to Lambda.

Architecture of API Gateway



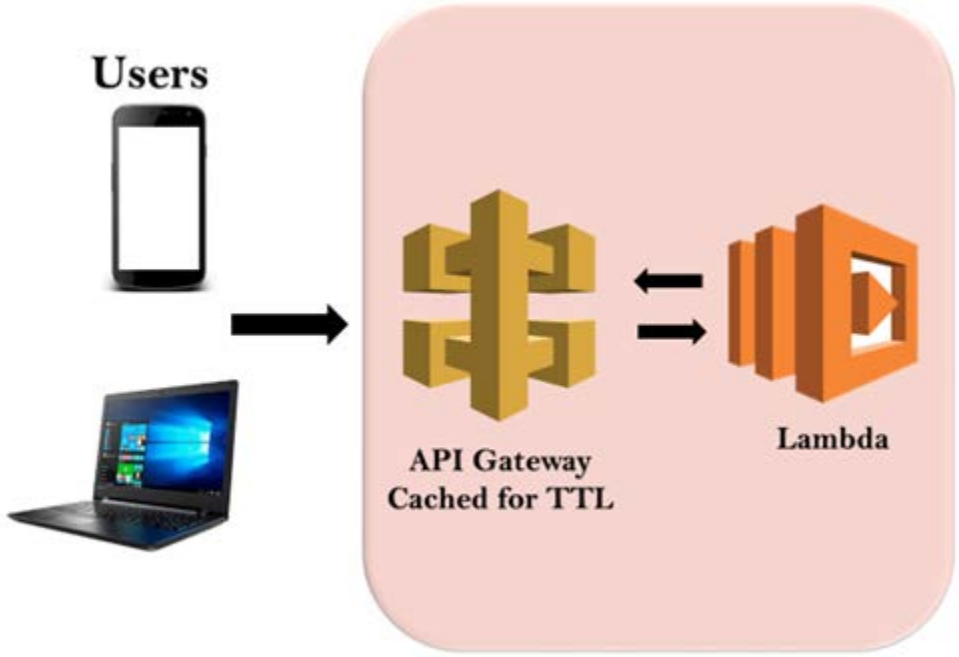
Suppose we got the users operating through phones or laptop makes an API call to API Gateway. API Gateway triggers either a Lambda function or a function inside the EC2.

What is API Caching?

- In Amazon API Gateway, you can enable API caching to cache your endpoint's responses.

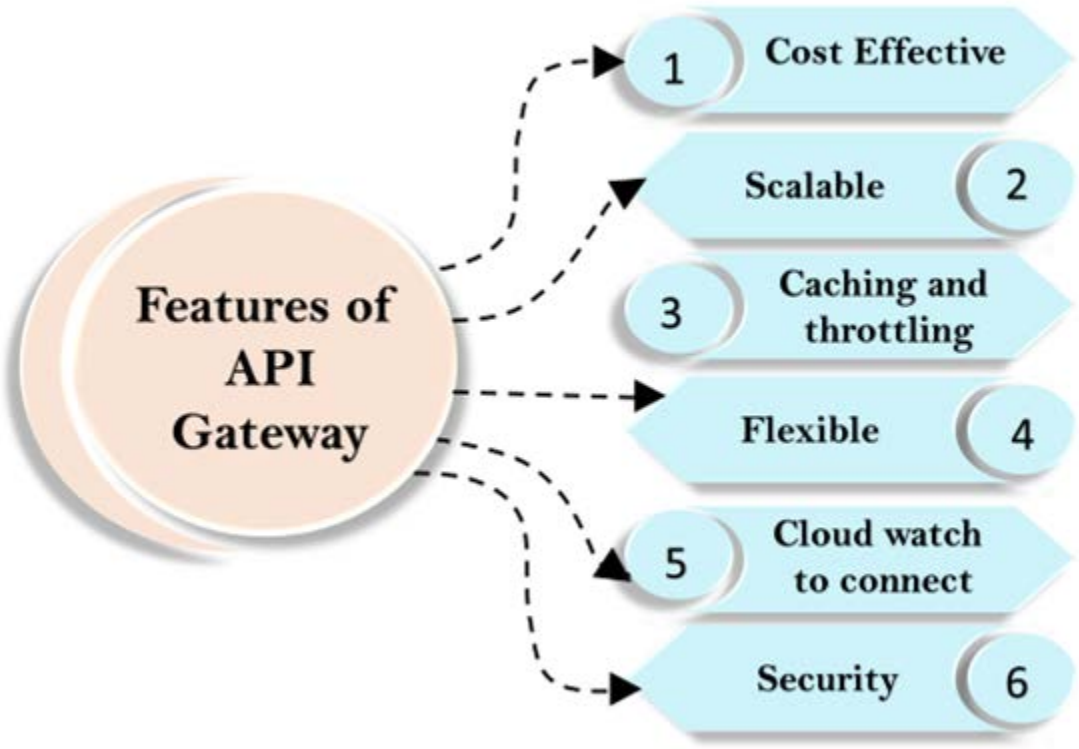
- API Caching can reduce the number of calls made to your endpoint and also improve the latency of the requests to your API.
- When API caching is enabled, API Gateway caches the responses from your endpoint for a specified time-to-live period, in seconds. API Gateway responds to the requests by looking up the response from the cache instead of making requests to your endpoint.

Architecture of API Caching



Suppose we have got the users making API calls to API Gateway. API Gateway triggers a Lambda function, and then Lambda function sends back the response to the API Gateway. Now, another user came and makes the same API call to API Gateway then we turn on Caching and time-to-live (TTL). Suppose the TTL is 60 seconds. Therefore, the response is sitting inside the API Gateway for 60 seconds. Now, to serve another user's request, you do not need to call the Lambda function as the response for that request has already been cached. In this way, we can get a much faster response from the end users.

Features of API Gateway



- **Cost Effective**
  It is very low cost and efficient as an API Gateway provides a tiered pricing model for API requests. The price of an API request is as low as $1.51 per million requests, you can also decrease the costs by decreasing the number of requests.
- **Scalable**
  You do not have to worry about having EC2 service or Autoscaling groups responding to API requests. An API Gateway scales automatically.
- **Caching and Throttling**
  Caching is the most important feature of API Gateway. Caching is used to cache the endpoint's responses which improve the latency of requests to your API. It is also a primary factor that determines the price of the service.
  You can also prevent the security risks to your API Gateway. If you want to prevent from flooding with the fraud API calls to API Gateway, you can configure throttle service that can throttle requests to prevent attacks.
- **Flexible**
  To implement the API Gateway, you do not have to launch an EC2 instance or setting up the Gateway software. API Gateway can be implemented in few minutes through the AWS Management Console.
- **CloudWatch to Connect**
  An Amazon API Gateway is integrated with the CloudWatch service which is a

monitoring service. This tool is used to monitor the metrics of incoming API calls, latency and errors.

- **Security**
  You can authorize access to your APIs. API Gateway is used to verify incoming requests by executing various authorization options such as Lambda function and Identity Access Management service (IAM). An IAM is integrated with a gateway that provides tools such as AWS credentials, i.e., access and secret keys to access an API. A Lambda function is used to verify tokens, and if tokens are successfully verified, then access to an API will be granted.

Kinesis

Before knowing about the Kinesis, you should know about the streaming data.

What is streaming data?

Streaming data is data which is generated continuously from thousands of data sources, and these data sources can send the data records simultaneously and in small size.

Following are the examples of streaming data:

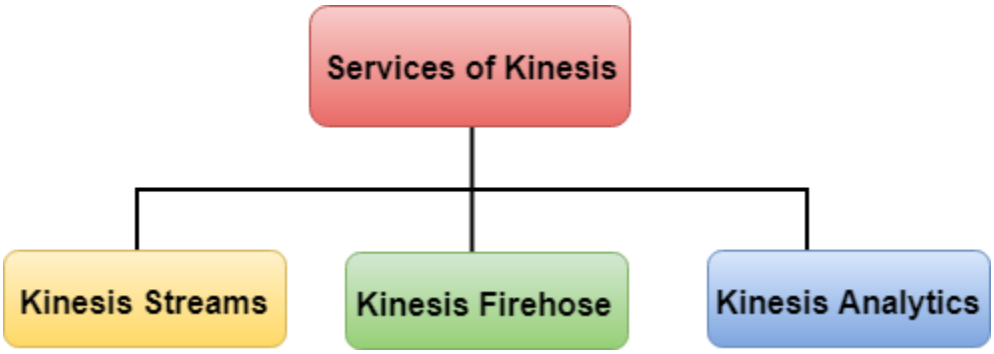
- **Purchases from online stores**
  People buying stuff on amazon.com and generates streaming data and that streaming data can be transactions, product, etc.
- **Stock prices**
  Stock price is also an example of streaming data.
- **Game data**
  Suppose the user is playing an angry bird game and the application is generating streaming data back to the central server. This streaming data could be "what the user is doing", "what is the score".
- **Social network data**
  Social network data is also another example of streaming data. Suppose you visit on Facebook, update your status, and put a post on your friend's wall. All these data would then be streamed.
- **Geospatial data**
  When you are using uber, and your device is connected to the internet. Uber application is constantly saying that where the uber driver is, where you are, and it

is interrogating the map to give you the best possible route to your destination. This is also a good example of streaming data.

- **iOT Sensor Data**
  It senses the all around world monitoring temperature.

What is Kinesis?

Kinesis is a platform on AWS that sends your streaming data. It makes it easy to analyze load streaming data and also provides the ability for you to build custom applications based on your business needs.



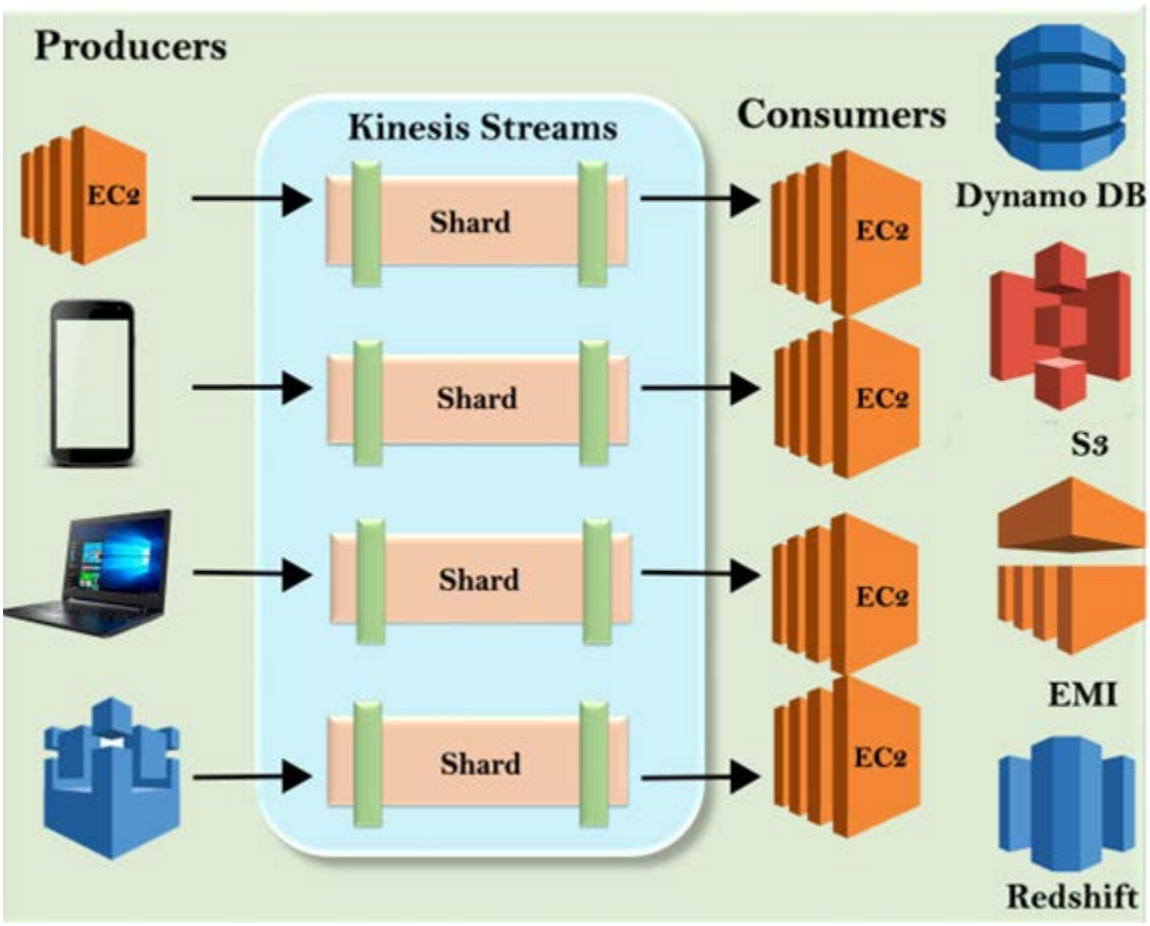
Core Services of Kinesis

- **Kinesis Streams**
- **Kinesis Firehose**
- **Kinesis Analytics**

Kinesis Streams

- Kinesis streams consist of shards.
- Shards provide 5 transactions per second for reads, up to a maximum total data read rate of 2MB per second and up to 1,000 records per second for writes up to a maximum total data write rate of 1MB per second.
- The data capacity of your stream is a function of the number of shards that you specify for the data stream. The total capacity of the Kinesis stream is the sum of the capacities of all shards.

Architecture of Kinesis Stream

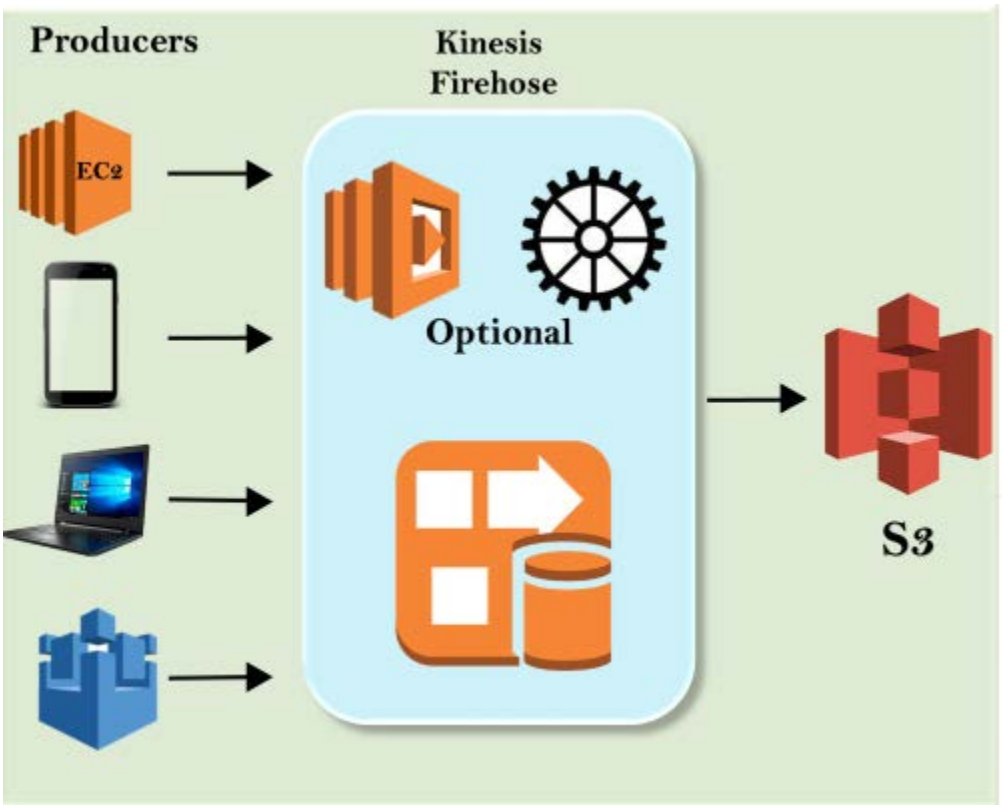


Suppose we have got the EC2, mobile phones, Laptops, IOT which are producing the data. They are known as producers as they produce the data. The data is moved to the Kinesis streams and stored in the shard. By default, the data is stored in shards for 24 hours. You can increase the time to 7 days of retention. Once the data is stored in shards, then you have EC2 instances which are known as consumers. They take the data from shards and turned it into useful data. Once the consumers have performed its calculation, then the useful data is moved to either of the AWS services, i.e., DynamoDB, S3, EMR, Redshift.

Kinesis Firehouse

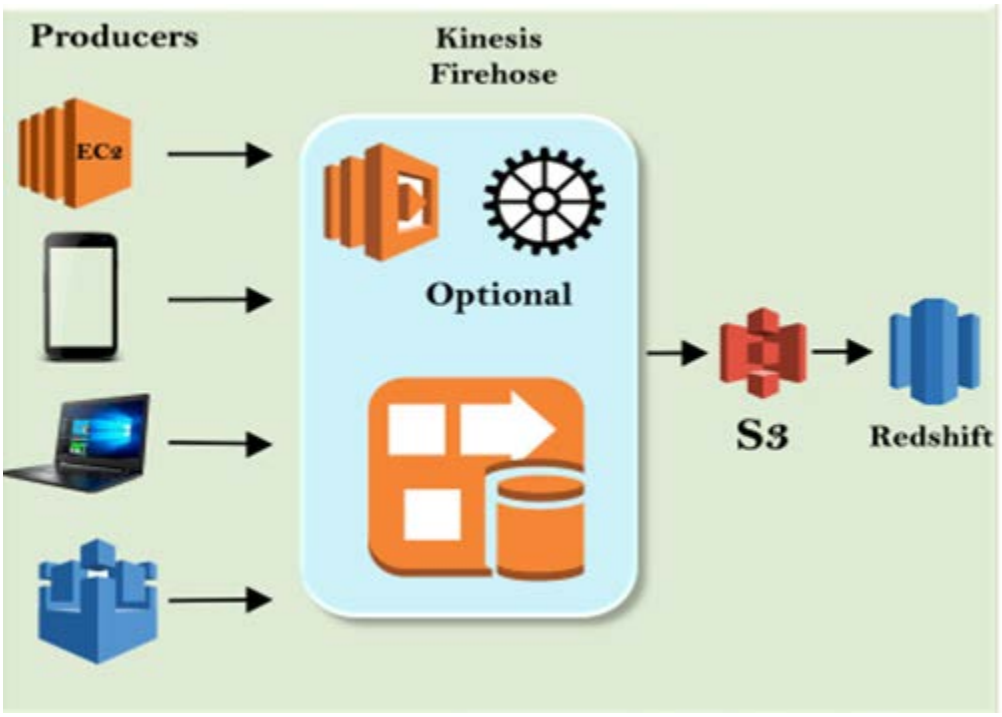
- Kinesis Firehose is a service used for delivering streaming data to destinations such as Amazon S3, Amazon Redshift, Amazon Elasticsearch.
- With Kinesis Firehouse, you do not have to manage the resources.

Architecture of Kinesis Firehose

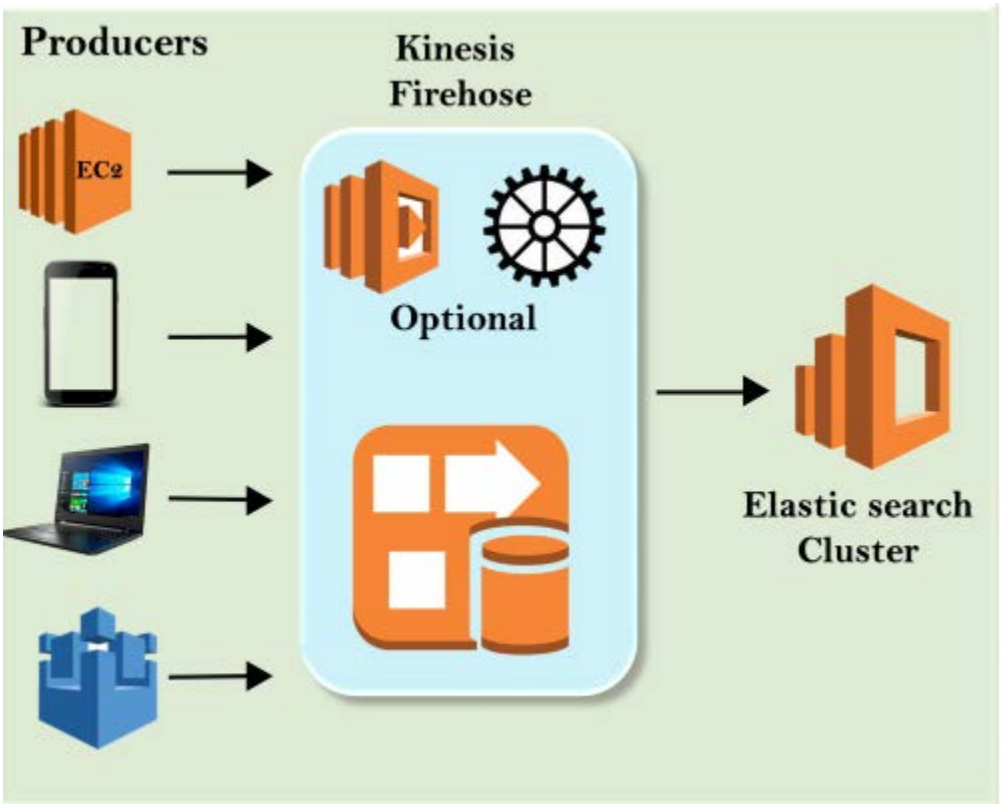


Suppose you have got the EC2, mobile phones, Laptop, IOT which are producing the data. They are also known as producers. Producers send the data to Kinesis Firehose. Kinesis Firehose does not have to manage the resources such as shards, you do not have to worry about streams, you do not have to worry about manual editing the shards to keep up with the data, etc. It?s completely automated. You do not have to worry even about the consumers. Data can be analyzed by using a Lambda function. Once the data has been analyzed, the data is sent directly over to the S3. The analytics of data is optional. One important thing about Kinesis Firehouse is that there is no automatic retention window, but the Kinesis stream has an automatic retention window whose default time is 24 hours and it can be extended up to 7 days. Kinesis Firehose does not work like this. It essentially either analyzes or sends the data over directly to S3 or other location.

The other location can be Redshift. First, you have to write to S3 and then copy it to the Redshift.



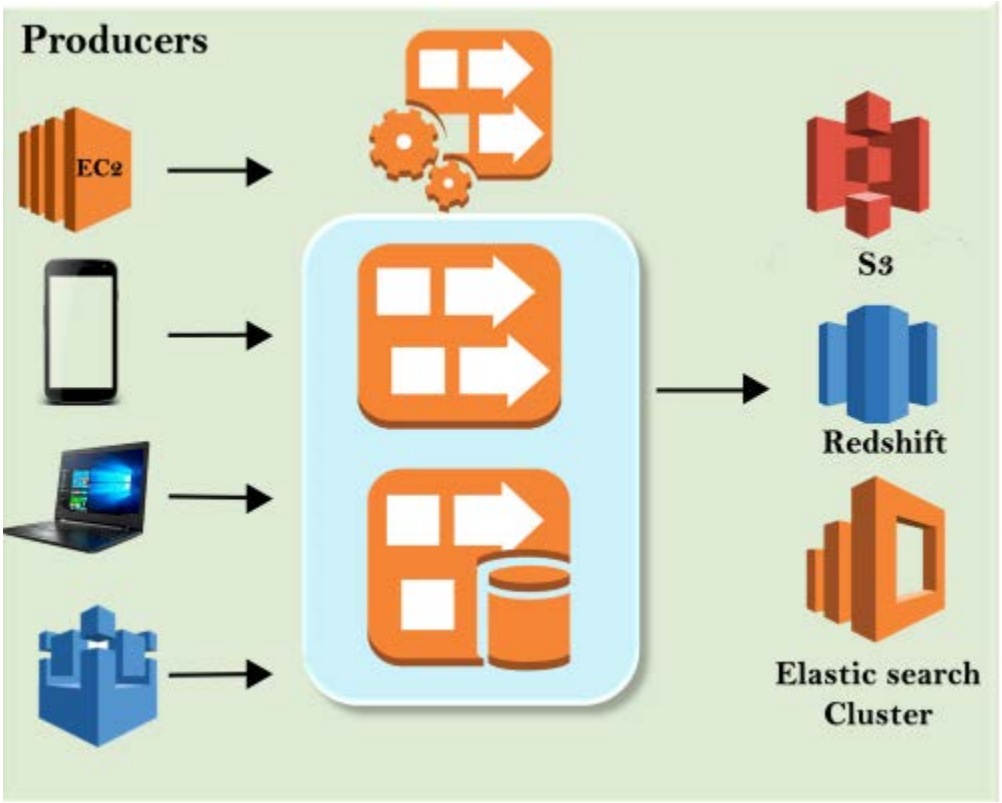
If the location is Elastic search cluster, then the data is directly sent to the Elastic search cluster.



Kinesis Analytics

Kinesis Analytics is a service of Kinesis in which streaming data is processed and analyzed using standard SQL.

Architecture of Kinesis Analytics



We have got the kinesis firehose and kinesis stream. Kinesis Analytics allows you to run the SQL Queries of that data which exist within the kinesis firehose. You can use the SQL Queries to store the data in S3, Redshift or Elasticsearch cluster. Essentially, data is analyzed inside the kinesis using SQL type query language.

Differences b/w Kinesis Streams & Kinesis Firehose

- **Kinesis stream is manually managed while Kinesis Firehose is fully automated managed.**
- **Kinesis stream sends the data to many services while Kinesis Firehose sends the data only to S3 or Redshift.**
- **Kinesis stream consists of an automatic retention window whose default time is 24 hours and can be extended to 7 days while Kinesis Firehose does not have automatic retention window.**

- **Kinesis streams send the data to consumers for analyzing and processing while kinesis firehose does not have to worry about consumers as kinesis firehose itself analyzes the data by using a lambda function.**

What is SAML?

- SAML stands for Security Assertion Markup language.
- Generally, users need to enter a username and password to login in any application.
- SAML is a technique of achieving **Single Sign-On (SSO)**.
- Security Assertion Markup Language (SAML) is an Xml-based framework that allows the identity providers to provide the authorization credentials to the service provider.
- With SAML, you need to enter one security attribute to log in to the application
- SAML is a link between the authentication of the user's identity and authorization to use a service.
- SAML provides a service known as Single Sign-On means that users have to log in once and can use the same credentials to log in to another service provider.

Why SAML?

- With SAML, both the service provider and identity provider exist separately, but centralizes the user management and provides access to the SaaS solutions.
- SAML authentication is mainly used for verifying the user's credentials from the identity provider.

Advantages of SAML:

- **SAML SSO (SINGLE SIGN-ON):** SAML provides security by eliminating passwords for an app and replacing them with the security tokens. Since we do not require any passwords for SAML logins, there is no risk of credentials to be stolen by unauthorized users. It provides faster, easier and trusted access to cloud applications.
- **Open Standard SINGLE SIGN-ON:** SAML implementations confirms to the open standard. Therefore, it is not restricted to a single identity provider. This open standard allows you to choose the SAML provider.
- **Strong Security:** SAML uses federated identities and secure tokens to make SAML one of the best secure forms for web-based authentication.
- **Improved online experience for end users:** SAML provides SINGLE SIGN-ON (SSO) to authenticate at an identity provider, and the identity provider sends the authentication to the service provider without additional credentials.

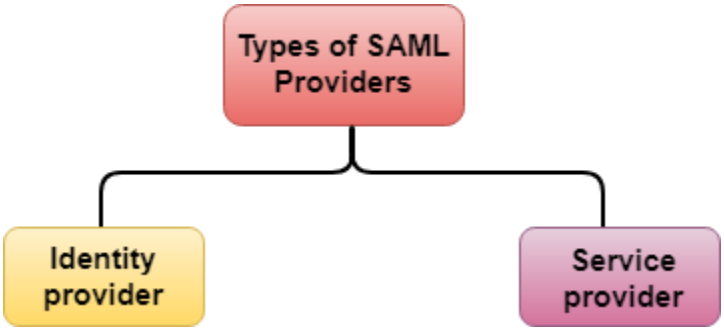
- **Reduced administrative costs for service providers:** Using single authentication multiple times for multiple services can reduce the administrative costs for maintaining the account information.
- **Risk transference:** SAML has put the responsibility of handling the identities to the identity provider.

Types of SAML providers

SAML provider is an entity within a system that helps the user to access the services that he or she wants.

There are two types of SAML providers:

- Service provider
- Identity provider



Service provider

- It is an entity within a system that provides the services to the users for which they are authenticated.
- Service provider requires the authentication from the identity provider that grants the access to the user.
- Salesforce and other CRM are the common service providers.

Identity provider

- An identity provider is an entity within a system that sends the authentication to the service provider is about who they are along with the user access rights.
- It maintains a directory of the user and provides an authentication mechanism.
- Microsoft Active Directory and Azure are the common identity providers.

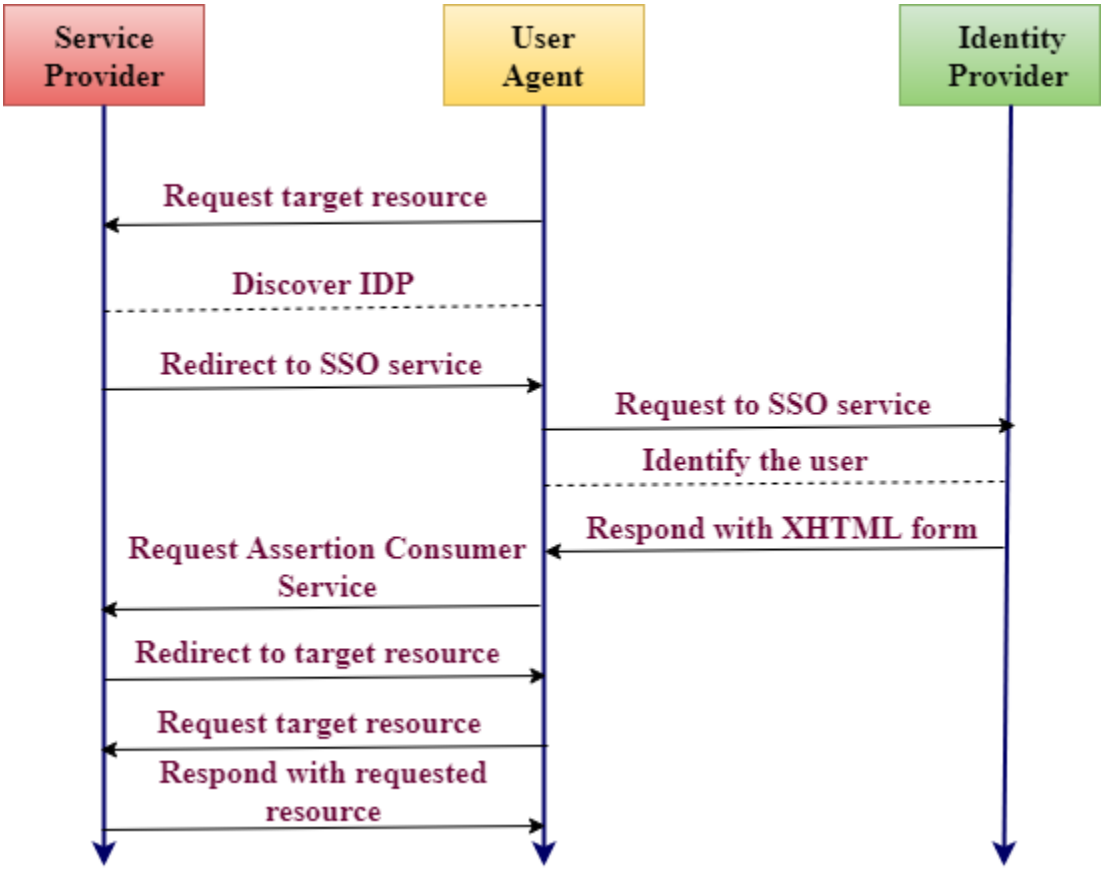
What is a SAML Assertion?

A SAML Assertion is an XML document that the identity provider sends to the service provider containing user authorization.

SAML Assertion is of three types:

- **Authentication**
  - It proves the identification of the user
  - It provides the time at which the user logged in.
  - It also determines which method of authentication has been used.
- **Attribute**
  - An attribute assertion is used to pass the SAML attributes to the service provider where attribute contains a piece of data about the user authentication.
- **Authorization decision**
  - An authorization decision determines whether the user is authorized to use the service or identity provider denied the request due to the password failure.

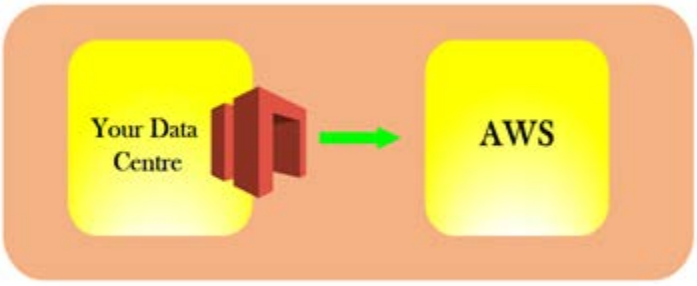
Working of SAML



- If a user tries to access the resource on the server, the service provider checks whether the user is authenticated within the system or not. If you are, you skip to step 7, and if you are not, the service provider starts the authentication process.
- The service provider determines the appropriate identity provider for you and redirects the request to the identity provider.
- An authentication request has been sent to the SSO (SINGLE SIGN-ON) service, and SSO service identifies you.
- The SSO service returns with an XHTML document, which contains authentic information required by the service provider in a SAMLResponse parameter.
- The SAMLResponse parameter is passed to the Assertion Consumer Service (ACS) at the service provider.
- The service provider processes the request and creates a security context; you automatically logged in.
- After login, you can request for a resource that you want.
- Finally, the resource is returned to you.

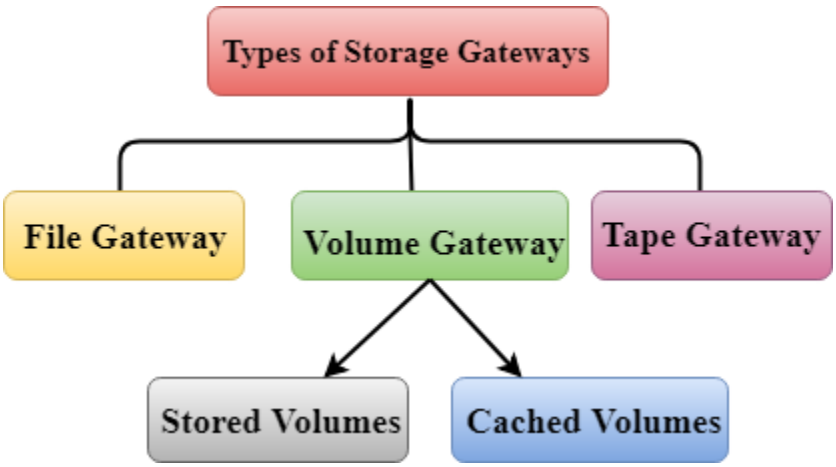
Storage Gateway

- Storage Gateway is a service in AWS that connects an on-premises software appliance with the cloud-based storage to provide secure integration between an organization's on-premises IT environment and AWS storage infrastructure.



- Storage Gateway service allows you to securely store the data in AWS cloud for the scalable and cost-effective storage.
- Storage Gateway is a virtual appliance which is installed in a hypervisor running in a Data center used to replicate the information to the AWS particularly S3.
- Amazon Storage Gateway's virtual appliance is available for download as a virtual machine (VM) image which you can install on a host in your data center.
- Storage Gateway supports either Vmware EXI or Microsoft Hyper-V.
- Once you have installed the storage gateway, link it with your AWS account through the activation process, and then you can use the AWS Management Console to create the storage gateway option.

There are three types of Storage Gateways:



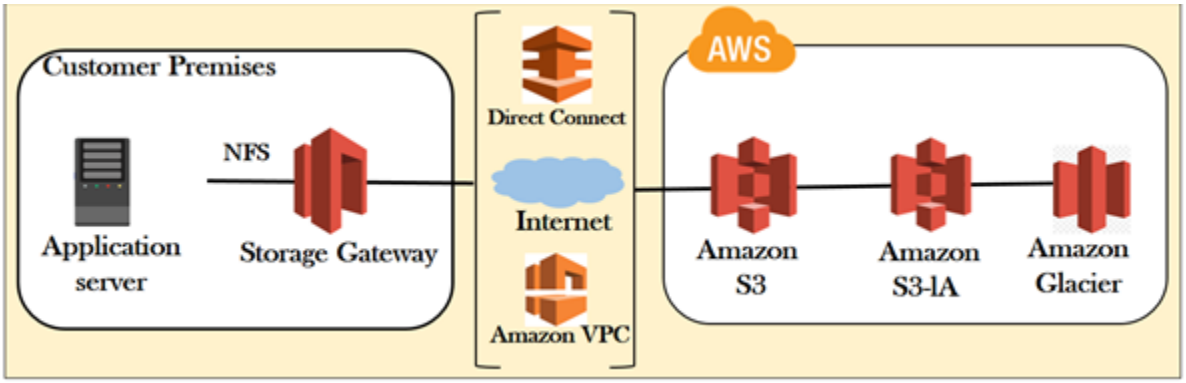
- File Gateway (NFS)
- Volume Gateway (iSCSI)
- Tape Gateway (VTL)

The above image shows that the storage gateway is categorized into three parts: File Gateway, Volume Gateway, and Tape Gateway. Volume Gateway is further classified into two parts: Stored Volumes and Cached Volumes.

File Gateway

- It is using the technique NFS.
- It is used to store the flat files in S3 such as word files, pdf files, pictures, videos, etc.
- It is used to store the files to S3 directly.
- Files are stored as objects in S3 buckets, and they are accessed through a Network File System (NFS) mount point.
- Ownership, permissions, and timestamps are durably stored in S3 in the user metadata of the object associated with the file.
- Once the objects are transferred to the S3, they can be used as the native S3 objects, and bucket policies such as versioning, lifecycle management, and cross-region replication can be directly applied to the objects stored in your bucket.

Architecture of File Gateway



- Storage Gateway is a virtual machine running on-premises.
- Storage Gateway is mainly connected to aws through the internet.
- It can use Direct Connect. Direct Connect is a direct connection line between the Data center and aws.
- It can also use an Amazon VPC (Virtual Private Cloud) to connect a storage gateway to aws. VPC is a virtual data center. It represents that the Application server and storage gateway do not need to be on-premises. In Amazon VPC, storage gateway sits inside the VPC, and then storage gateway sends the information to S3.

Volume Gateway

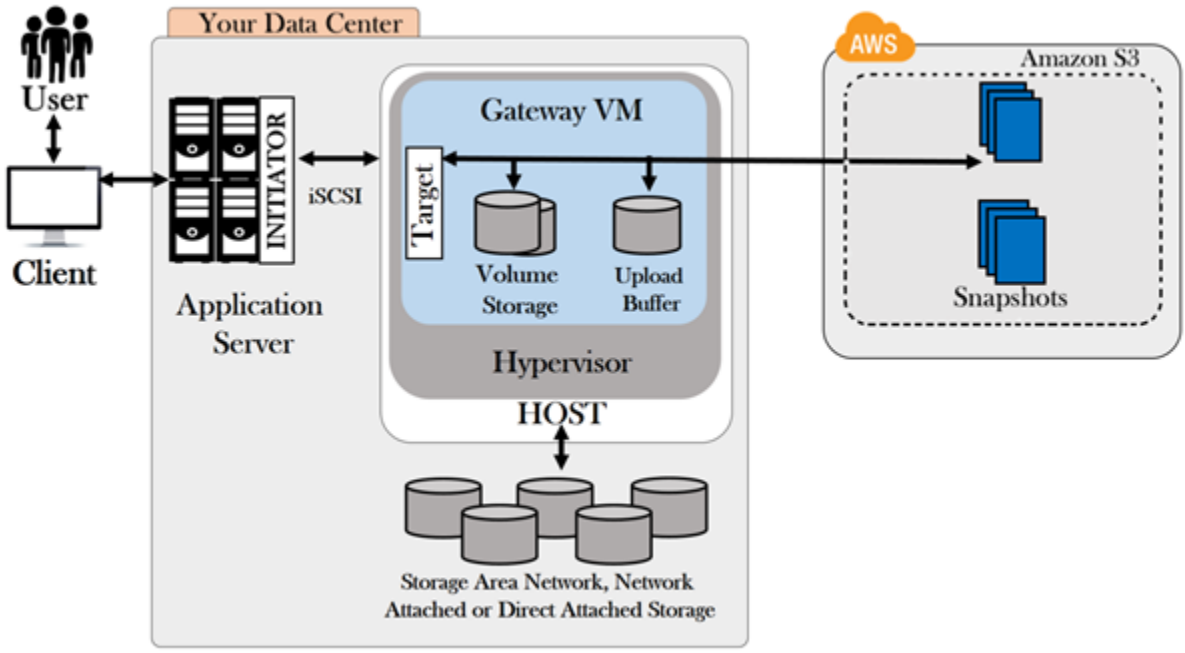
- Volume Gateway is an interface that presents your applications with disk volumes using the Iscsi block protocol. The iSCSI block protocol is block-based storage that can store an operating system, applications and also can run the SQL Server, database.
- Data written to the hard disk can be asynchronously backed up as point-in-time snapshots in your hard disks and stored in the cloud as EBS snapshots where EBS (Elastic Block Store) is a virtual hard disk which is attached to the EC2 instance. In short, we can say that the volume gateway takes the virtual hard disks that you back them up to the aws.
- Snapshots are incremental backups so that the changes made in the last snapshot are backed up. All snapshot storage is also compressed to minimize your storage charges.

Volume Gateway is of two types:

Stored Volumes

- It is a way of storing the entire copy of the data locally and asynchronously backing up the data to aws.
- Stored volumes provide low-latency access to the entire datasets of your on-premise applications and offsite backups.
- You can create a stored volume that can be a virtual storage volume which is mounted as iSCSI devices to your on-premise application services such as data services, web services.
- Data written to your stored volume is stored on your local storage hardware, and this data is asynchronously backed up to the Amazon Simple storage services in the form of Amazon Elastic Block store snapshots.
- The size of the stored volume is 1GB - 16 TB.

Architecture of Volume Gateway



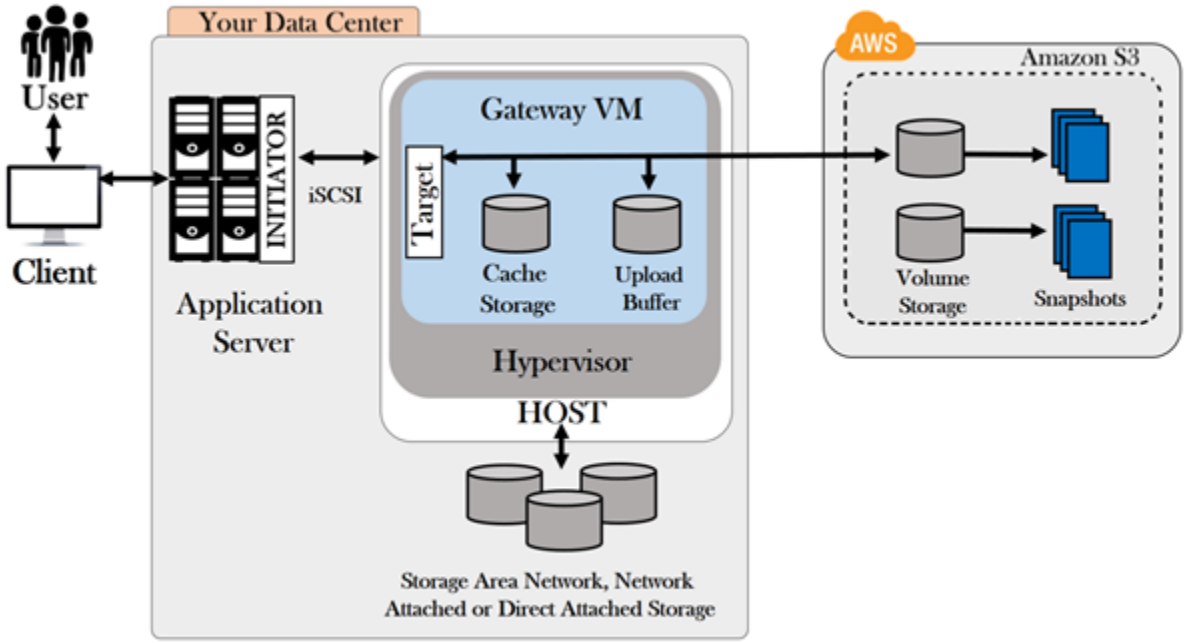
- A client is talking to the server that could be an application server or a web server.
- An application server is having an Iscst connection with the volume Gateway.
- Volume Gateway is installed on the Hypervisor.
- The volume storage is also known as a virtual hard disk which is stored in physical infrastructure, and the size of the virtual hard disk is 1TB.
- The volume storage takes the snapshots and sends them to the Upload buffer.
- The upload buffer performs the multiple uploads to the S3, and all these uploads are stored as EBS snapshots.

Cached Gateway

- It is a way of storing the most recently accessed data on site, and the rest of the data is stored in aws.
- Cached Volume allows using the Amazon Simple Storage service as your primary data storage while keeping the copy of the recently accessed data locally in your storage gateway.
- Cached Volume minimizes the need to scale your on-premises storage infrastructure while still providing the low-latency access to their frequently accessed data.
- Cached Gateway stores the data that you write to the volume and retains only recently read data in on-premises storage gateway.

- The size of the cached volume is 1GB - 32 TB.

Architecture of Cached Gateway



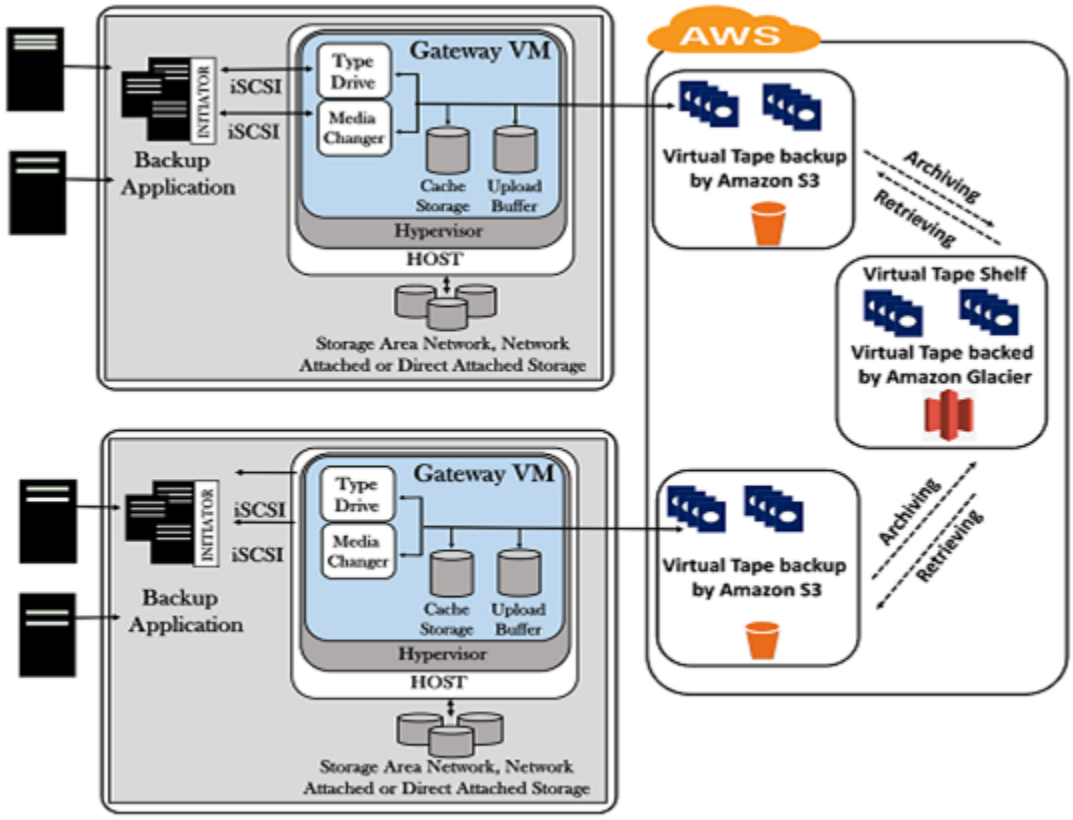
- A client is connected to the Application server, and an application server is having an iSCSI connection with the Gateway.
- The data send by the client is stored in the cache storage and then uploaded in an upload buffer.
- The data from the upload buffer is transferred to the virtual disks, i.e., volume storage which sits inside the Amazon S3.
- Volume storage is block-based storage which cannot be stored in S3 as S3 is object-based storage. Therefore, the snapshots, i.e., the flat files are taken, and these flat files are then stored in S3.
- The most recently read data is stored in the Cache Storage.

Tape Gateway

- Tape Gateway is mainly used for taking backups.
- It uses a Tape Gateway Library interface.

- Tape Gateway offers a durable, cost-effective solution to archive your data in AWS cloud.
- The VTL interface provides a tape-based backup application infrastructure to store data on virtual tape cartridges that you create on your tape Gateway.
- It is supported by NetBackup, Backup Exec, Veeam, etc. Instead of using physical tape, they are using virtual tape, and these virtual tapes are further stored in Amazon S3.

Architecture of Tape Gateway



- Servers are connected to the Backup Application, and the Backup Application can be NetBackup, Backup Exec, Veeam, etc.
- Backup Application is connected to the Storage Gateway over the iSCSI connection.
- Virtual Gateway is represented as a virtual appliance connected over iSCSI to the Backup application.
- Virtual tapes are uploaded to an Amazon S3.

- Now, we have a Lifecycle Management policy where we can archive to the virtual tape shelf in Amazon Glacier.

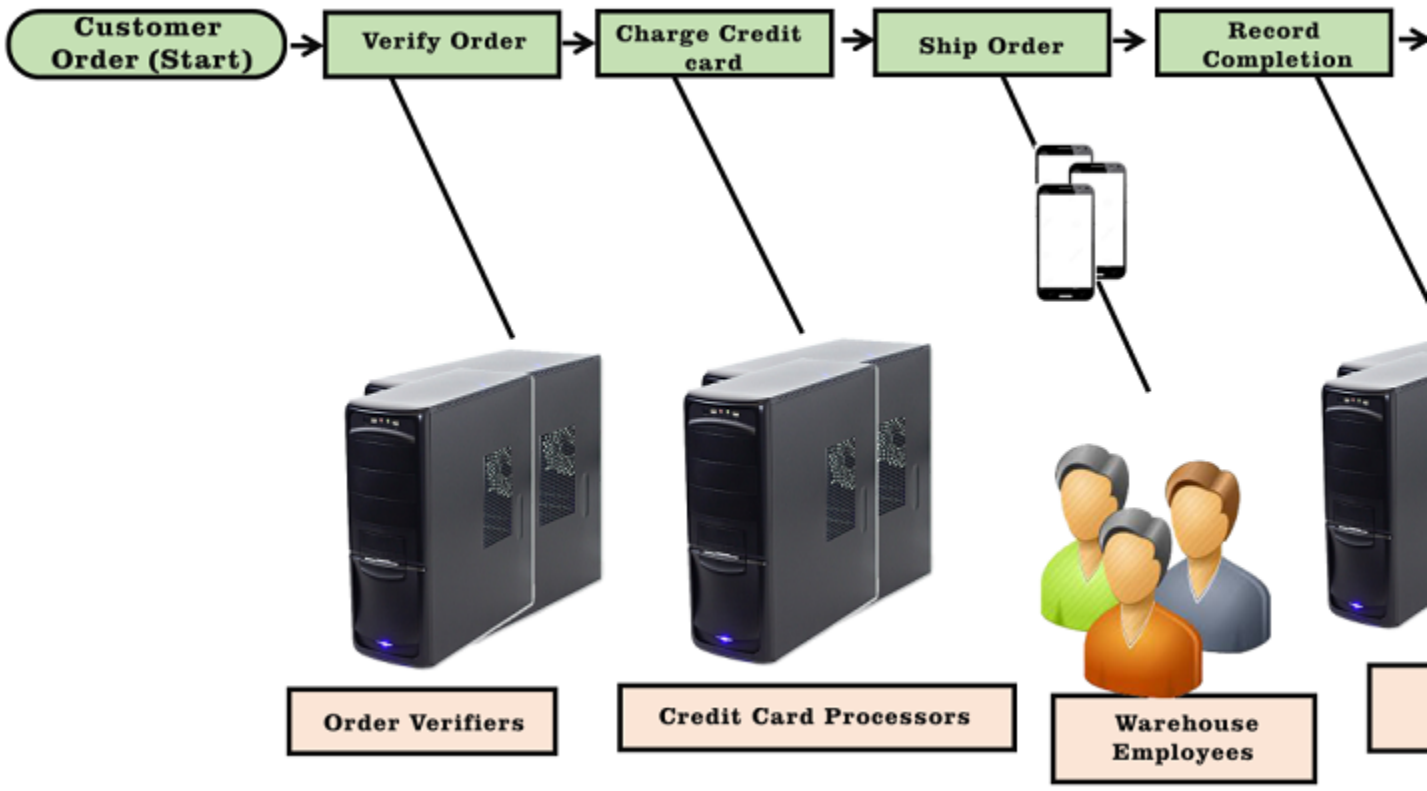
Important points to remember:

- File Gateway is used for object-based storage in which all the flat files such as word files, pdf files, etc, are stored directly on S3.
- Volume Gateway is used for block-based storage, and it is using an iSCSI protocol.
- Stored Volume is a volume gateway used to store the entire dataset on site and backed up to S3.
- Cached volume is a volume gateway used to store the entire dataset in a cloud (Amazon S3) and only the most frequently accessed data is kept on site.
- Tape Gateway is used for backup and uses popular backup applications such as NetBackup, Backup Exec, Veeam, etc.

What is SWF?

- SWF stands for **Simple Workflow Service**.
- It is a web service used to build scalable and resilient applications.
- It provides simple API calls which can be executed from code written in any language and can be run on your EC2 instance or any of your machines located anywhere in the world that access the internet. For example, you are building an application which consists of various modules, and to coordinate among various modules; we rely on SWF in aws. SWF acts as a coordinator, and it has control over all the modules of an application.
- It allows you to build applications and makes it easy to coordinate the work across distributed components.
- SWF provides a logical separation among all the components of a project.
- SWF involves in coordinating various tasks such as managing inter-task dependencies, scheduling, and concurrency in accordance with the logical flow of the application. You do not have to manage the tasks manually; SWF will do everything for you.

Let's understand through an example.



Suppose customer placed an order.

Step 1: You have to verify an order. You have your EC2 instances, and they go and check whether the order is in stock or not. Once the order has been verified, i.e., you have got a stock, then move to step 2.

Step 2: Now, it works on the **Charge Credit card**. It checks whether the charge of a credit card has been successful or not.

Step 3: If the charge of a credit card has been successful, we will ship an order. Shipping an order needs human interaction. Human brings order from the warehouse, and if the product has been boxed up means that it is ready for the shipment.

Step 4: Record Completion is a database which says that the product has been boxed up and shipped to the destination address. It also provides the tracking number. This is the end of the typical workflow.

SWF Workers and Deciders

- Workers are the programs that interact with the Amazon SWF to get the tasks, process the received tasks, and return the results.
- The decider is a program that provides coordination of tasks such as ordering, concurrency, scheduling, etc according to the application logic.

- Both workers and deciders run on the cloud infrastructure such as Amazon EC2, or machines behind firewalls.
- Deciders take a consistent view into the progress of tasks and initiate new tasks while Amazon SWF stores the tasks and assigns them to the workers to process them.
- Amazon SWF ensures that the task is assigned only once and is never duplicated.
- Workers and Deciders do not have to keep track of the execution state as Amazon SWF maintains the state durably.
- Both the workers and deciders run independently and scale quickly.

SWF Domains

- Domains are containers which isolate a set of types, executions, and task lists from others within the same account.
- Workflow, activity types, and workflow execution are all scoped to a domain.
- You can register a domain either by using the AWS Management Console or RegisterDomain action in the Amazon SWF API.

The parameters are specified in a JSON (**Javascript Object Notation**) format. The format is shown below:

```
RegisterDomain
{
   "name" : "867530901";
   "Description": "music";
    "workflowExecutionRetentionPeriodInDays": "60";
}
```

Where,

workflowExecutionRetentionPeriodInDays defines the number of days of retention period.

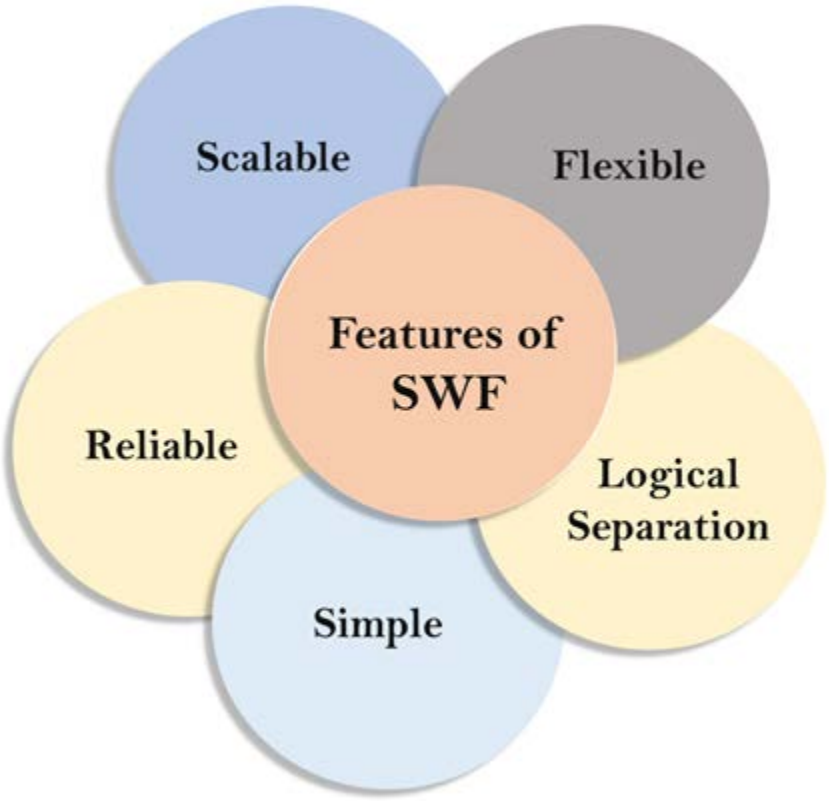
> ***Note: The maximum workflow can be 1 year and its value is measured in seconds.***

Differences b/w SQS and SWF

- Amazon SWF provides a task-oriented API while Amazon SQS provides a message-oriented API./li>

- Amazon SWF ensures that the task is assigned only once and is never duplicated. With Amazon SQS, the message can be duplicated and it may also need to ensure that a message is processed only once./li>
- SWF keeps track of all tasks and events in an application while SQS implements its own application level tracking when an application uses multiple queues.

Features of SWF



- **Scalable**
  Amazon SWF automatically scales the resources along with your application's usage. There is no manual administration of the workflow service required when you add more cloud workflows or increase the complexity of the workflows.
- **Reliable**
  Amazon SWF runs at Amazon's highly available data centres, therefore the state tracking is provided whenever applications need them. Amazon SWF stores the tasks, sends them to their respective application components, keeps a track on their progress.
- **Simple**
  Amazon SWF completely replaces the complexity of the old workflow solutions and

process automation software with new cloud workflow internet service. It eliminates the need for the developers to manage the automation process so that you can focus on the unique functionality of an application.

- **Logical separation**
  Amazon SWF provides a logical separation between the control flow of your background job's stepwise logic and the actual units of work that contains business logic. Due to the logical separation, you can separately manage, maintain, and scale "state machinery" of your application from the business logic. According to the change in the business requirements, you can easily manage the business logic without having worry about the state machinery, task dispatch, and flow control.
- **Flexible**
  Amazon SWF allows you to modify the application components, i.e., you can modify the application logic in any programming language and runs them within the cloud or on-premises.

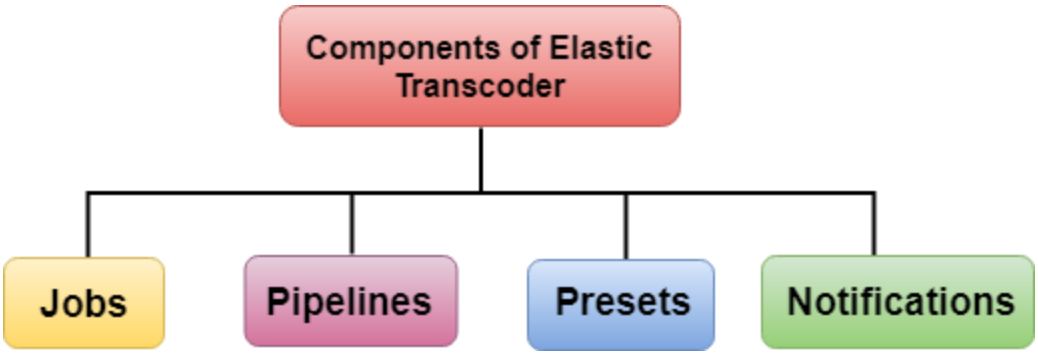
What is Elastic Transcoder?

- Elastic Transcoder is an aws service used to convert the media files stored in an S3 bucket into the media files in different formats supported by different devices.
- Elastic Transcoder is a media transcoder in the cloud.
- It is used to convert media files from their original source format into different formats that will play on smartphones, tablets, PC's, etc.
- It provides transcoding presets for popular output formats means that you don't need to guess about which settings work best on particular devices.
- If you use Elastic Transcoder, then you need to pay based on the minutes that you transcode and the resolution at which you transcode.

Components of Elastic Transcoder

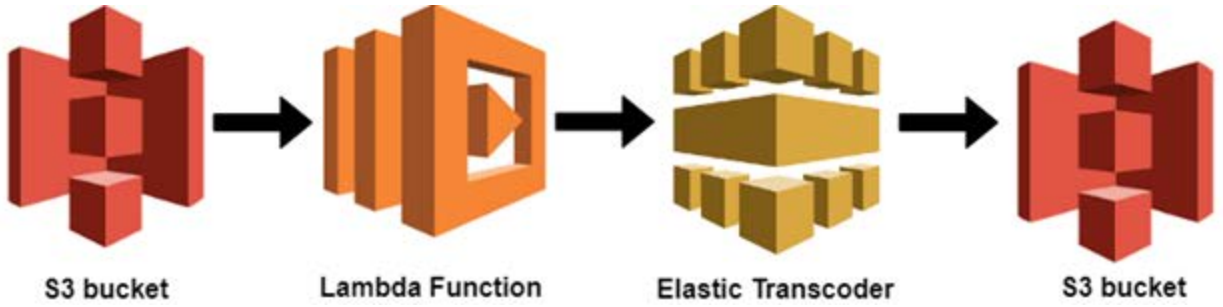
Elastic Transcoder consists of four components:

- **Jobs**
- **Pipelines**
- **Presets**
- **Notifications**



- **Jobs**
  The main task of the job is to complete the work of transcoding. Each job can convert a file up to 30 formats. For example, if you want to convert a media file into eight different formats, then a single job creates files in eight formats. When you create a job, you need to specify the name of the file that you want to transcode.
- **Pipelines**
  Pipelines are the queues that consist of your transcoding jobs. When you create a job, then you need to specify which pipeline you want to add your job. If you want a job to create more than one format, Elastic Transcoder creates the files for each format in the order you specify the formats in a job.
  You can create either of the two pipelines, i.e., standard-priority jobs and high-priority jobs. Mainly jobs go into the standard-priority jobs. Sometimes you want to transcode the file immediately; the high-priority pipeline is used.
- **Presets**
  Presets are the templates that contain the settings for transcoding the media file from one format to another format. Elastic transcoder consists of some default presets for common formats. You can also create your own presets that are not included in the default presets. You need to specify a preset that you want to use when you create a job.
- **Notifications**
  Notification is an optional field which you can configure with the Elastic Transcoder. Notification Service is a service that keeps you updated with the status of your job: when Elastic Transcoder starts processing your job, when Elastic Transcoder finishes its job, whether the Elastic Transcoder encounters an error condition or not.You can configure Notifications when you create a pipeline.

How A Cloud Uses Elastic Transcoder



Suppose I uploaded the mp4 file in S3 bucket. As soon as uploading is completed, it triggers a Lambda function. Lambda function will then invoke Elastic Transcoder. Elastic Transcoder converts the mp4 file into different formats so that the file can be opened in iphone, Laptop, etc. Once it has completed the transcoding, it stores the transcoded files in S3 bucket.