AWS Solution Architect

Serverless and Application Services

A Day in the Life of a Cloud Architect

You are working as a Cloud Architect in an organization, and you have been asked to launch the lambda function and perform the trigger operation on the lambda.

Highly Available Architectures on AWS have business-transforming capabilities. As a solution architect, you have to replicate the contents of a bucket to the other buckets deployed in different regions, thereby making the resources highly available.

To achieve all of the above, along with some additional concepts, we would be learning a few concepts in this lesson that will help you find a solution for the given scenario.

Learning Objectives

By the end of this lesson, you will be able to:

- Explain the lambda serverless compute
- Enable event notification for AWS resources using SNS
- Explain EventBridge, Event Bus, SaaS Partner Bus
- Explain the difference between SNS and SQS
- Enables real-time processing of streaming big data.

Lambda

What Are AWS Serverless Services?

AWS serverless services help users build and run applications without worrying about provisioning, maintaining, and managing the servers.

AWS serverless services

Why Use AWS Serverless Services?

AWS serverless services eliminate the following infrastructure management tasks:

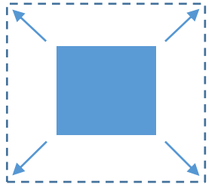
- Server or cluster provisioning
- Operating system maintenance
- Software and hardware patching
- Compute capacity provisioning

Benefits of AWS Serverless Services

The following are the benefits of AWS serverless services:

No Server Management

Flexible Scalability



Pay for Value

High Availability

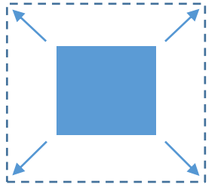
Users need not provide or maintain servers. There is no software or runtime that needs to be installed, maintained, or administered from the user's end.

Benefits of AWS Serverless Services

The following are the benefits of AWS serverless services:

No Server Management

Flexible Scalability



Pay for Value

High Availability

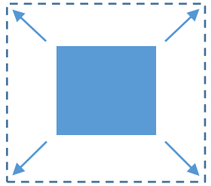
AWS serverless services allow the applications to be scaled up or down automatically.

Benefits of AWS Serverless Services

The following are the benefits of AWS serverless services:

No Server Management

Flexible Scalability



Pay for Value

High Availability

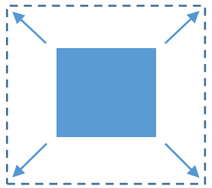
Instead of by server units, users are charged for the services' constant throughput or execution time.

Benefits of AWS Serverless Services

The following are the benefits of AWS serverless services:

No Server Management

Flexible Scalability



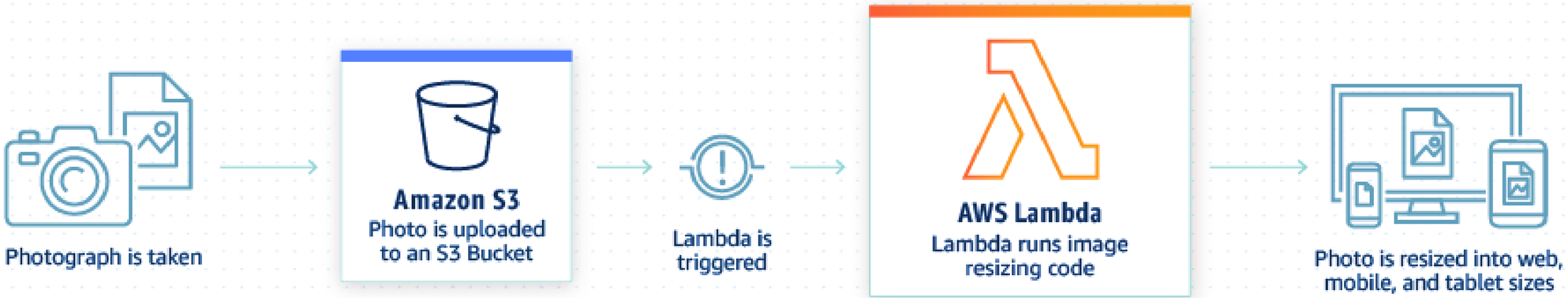
Pay for Value

High Availability

AWS serverless services provide built-in availability and fault tolerance to the applications running on them.

What Is AWS Lambda?

AWS Lambda is a serverless compute service that allows users to run code without provisioning or managing servers. It executes the code only when needed and scales automatically, from a few requests per day to thousands per second.



Terminologies in AWS Lambda

The following are the terminologies used in context with AWS Lambda:

01	02	03
Function	**Qualifier**	**Runtime**

A function is a resource that contains a code to process events and runtime to pass requests between Lambda and the function code.

Terminologies in AWS Lambda

The following are the terminologies used in context with AWS Lambda:

01	02	03
Function	**Qualifier**	**Runtime**

The qualifier is used to specify a version or an alias for a Lambda function.

Terminologies in AWS Lambda

The following are the terminologies used in context with AWS Lambda:

01	02	03
Function	**Qualifier**	**Runtime**

Runtimes allow function code written in different languages to run in the same base execution environment. Users are required to choose a runtime that matches the programming language of the code.

Terminologies in AWS Lambda

The following are the terminologies used in context with AWS Lambda:

04	05	06
Event	**Concurrency**	**Trigger**

An event is a JSON formatted document that contains data for a function to process. It is converted to an object and passed to the function code.

Terminologies in AWS Lambda

The following are the terminologies used in context with AWS Lambda:

04	05	06
Event	Concurrency	Trigger

Concurrency is the number of requests that a function is serving at any given time. Users can configure their functions to limit their concurrency.

Terminologies in AWS Lambda

The following are the terminologies used in context with AWS Lambda:

04	05	06
Event	Concurrency	Trigger

A trigger is a resource that invokes a Lambda function. It can be an AWS service, an application, or an event.

Stateless Design/Statelessness in Functions

- Functions are effectively stateless.
- Every time your Lambda function is triggered by an event it is invoked in a completely new environment.
- The environment exists only for a single invocation.
- No access to the execution context of the previous event.
- Rapid bursts of events at scale in a stateless manner.

Amazon RDS Proxy

Amazon RDS Proxy is a fully managed, highly available database proxy for AWS RDS.

Amazon RDS Proxy makes applications more:

01 Scalable

02 Resilient to database failures

03 Secure

RDS Proxy

To effectively manage connections to the relational database and increase the scalability of your application, Amazon RDS Proxy stands between it and the database.

Client Applications

Your application is pointed to the RDS Proxy endpoint

RDS Proxy

RDS Proxy sits between your application and database to efficiently manage DB connections

RDS Database

RDS Proxy pools and shares DB connections, improving database efficiency and application scalability

Networking and VPC configurations

The Networking and VPC configurations for Lamda functions:

- Lambda functions always run inside the VPCs owned by the Lambda service.
- These VPCs are not visible to customers.
- Configurations are maintained automatically, and monitoring is managed by the service.
- The Lambda service uses a Network Function Virtualization platform to provide NAT capabilities from the Lambda VPC to customer VPCs.

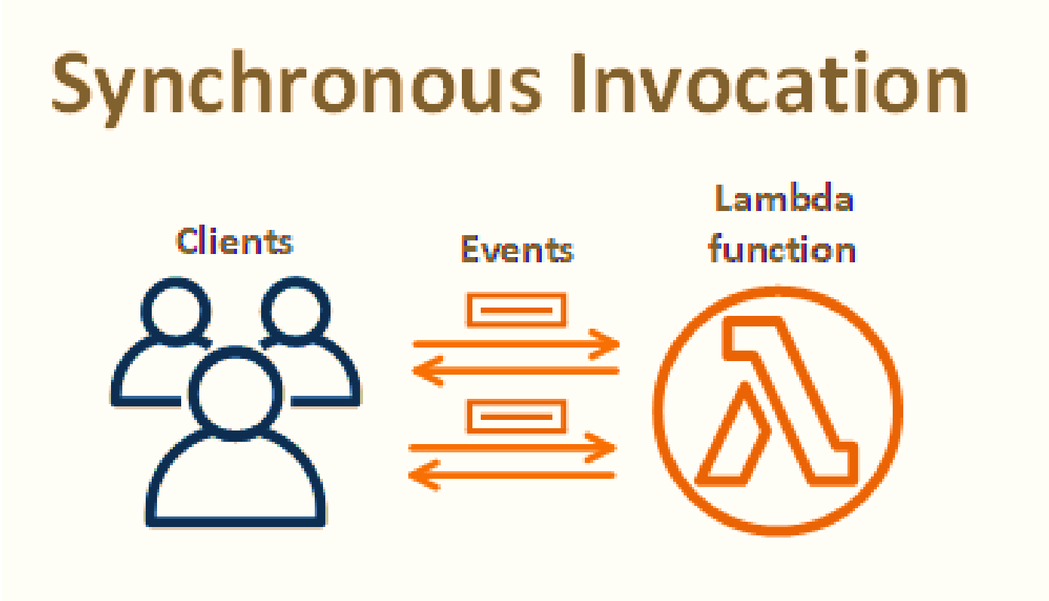
Security

The Security for Lamda functions:

- Lambda function execution environments are never shared across functions.
- Multiple mechanisms are used by the Lambda service to protect customer data.
- Principles of least privilege (in terms of permissions and scoping functions) to ensure that your user accounts are secured appropriately.
- Workloads secured with public endpoint, with authentication and authorization implemented.
- Data Encryption in Lambda-based applications.

Synchronous Invocations

The synchronous invocations are well suited for short-lived Lambda functions.



Examples of Synchronous Invocations

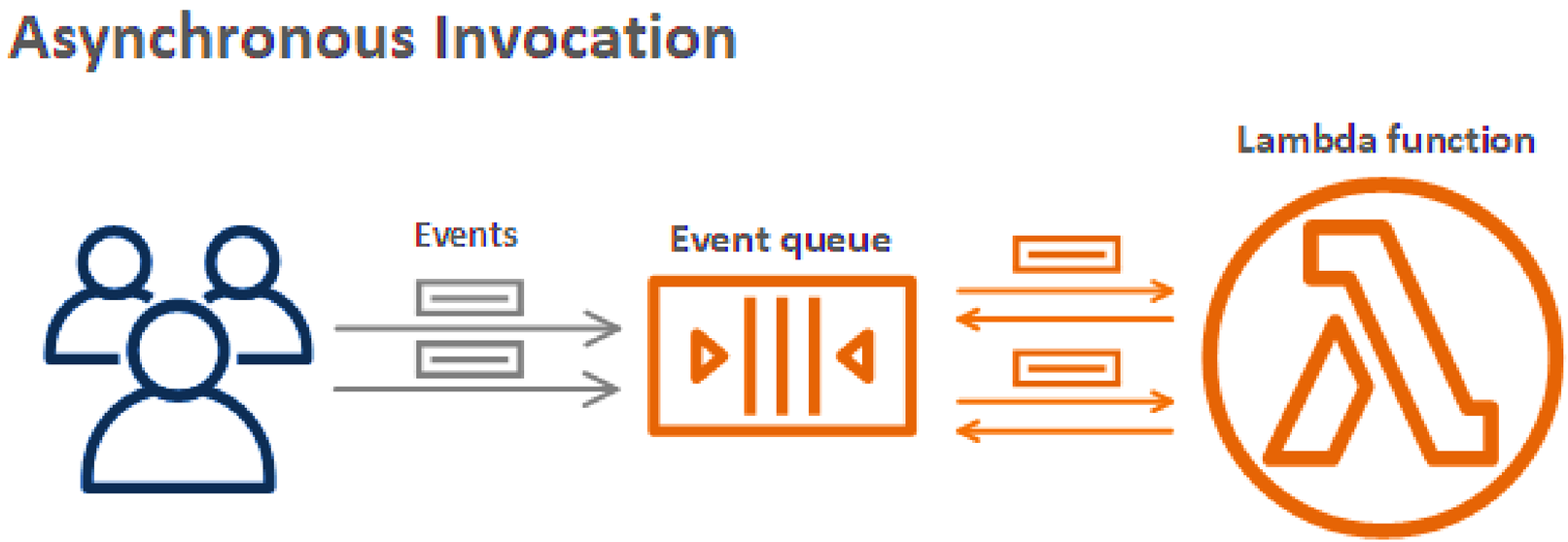
The examples of synchronous invocations are:

Amazon API Gateway

Amazon DynamoDB Streams

Asynchronous Invocations

Asynchronous invocations are well suited for not-so-short-lived lambda functions.



Examples of Asynchronous Invocations

The examples of asynchronous invocations are:

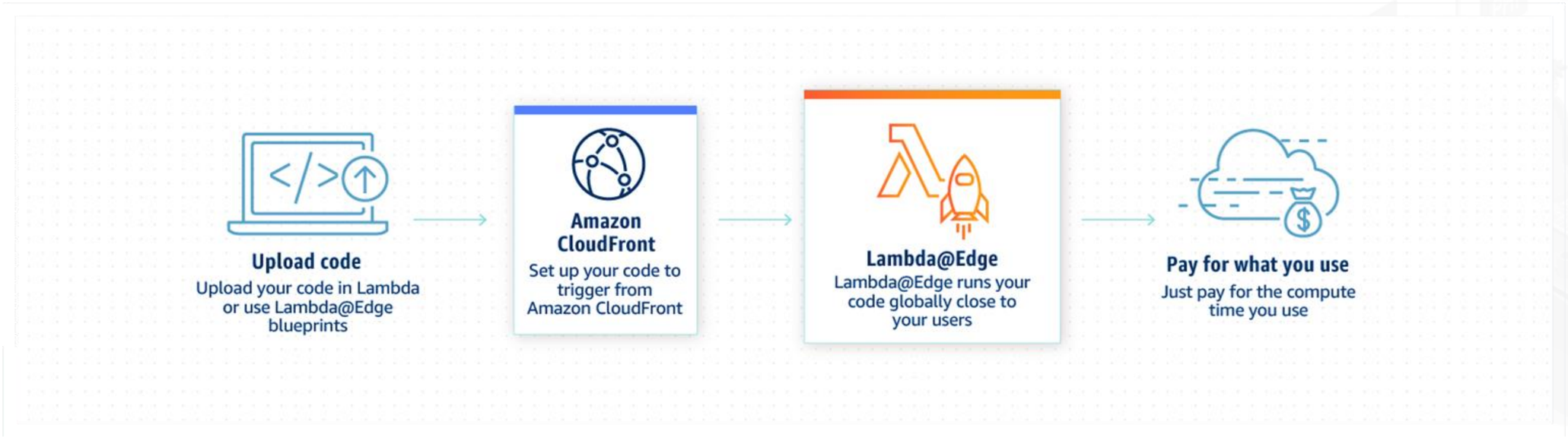
Lambda@Edge

Lamda@Edge can be used as an extension or replacement for origin infrastructure.

It enables you to do everything from simple HTTP requests and response processing at the edge to more advanced functionality.

Lambda@Edge

It is used to simplify and reduce origin infrastructure.



Create Lambda function, test and view CloudWatch logs

Duration: 15 mins

Problem Statement:

You have been asked to create lambda function, test and view CloudWatch logs

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Login to your AWS lab
2. Go to Lambda service
3. Create Lambda function
4. Deploy the code
5. Test the code
6. View the CloudWatch logs

Create a Lambda Layers

Duration: 15 mins

Problem Statement:

You have been asked to create a Lambda Layers

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Create a Lambda Function
2. Creating a Lambda Layer

API Gateway

What Is an API?

API stands for Application Programming Interface. It allows communication between two applications and is created for apps to access data, logic, and more.

Application Programming Interface

Amazon API Gateway

Amazon API Gateway is a fully-managed, scalable API management service that allows you to create, publish, maintain, monitor, and secure your APIs.

Amazon API Gateway

Features of API Gateway

01	Stores responses for the most common HTTP requests
02	Scales automatically
03	Cheaper than other gateways
04	Throttles requests to prevent attacks
05	Enables CORS to serve HTTP requests from other domains

Create a Serverless Web App

Duration: 13 mins

Problem statement:

You have been assigned a task to create a serverless web app

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

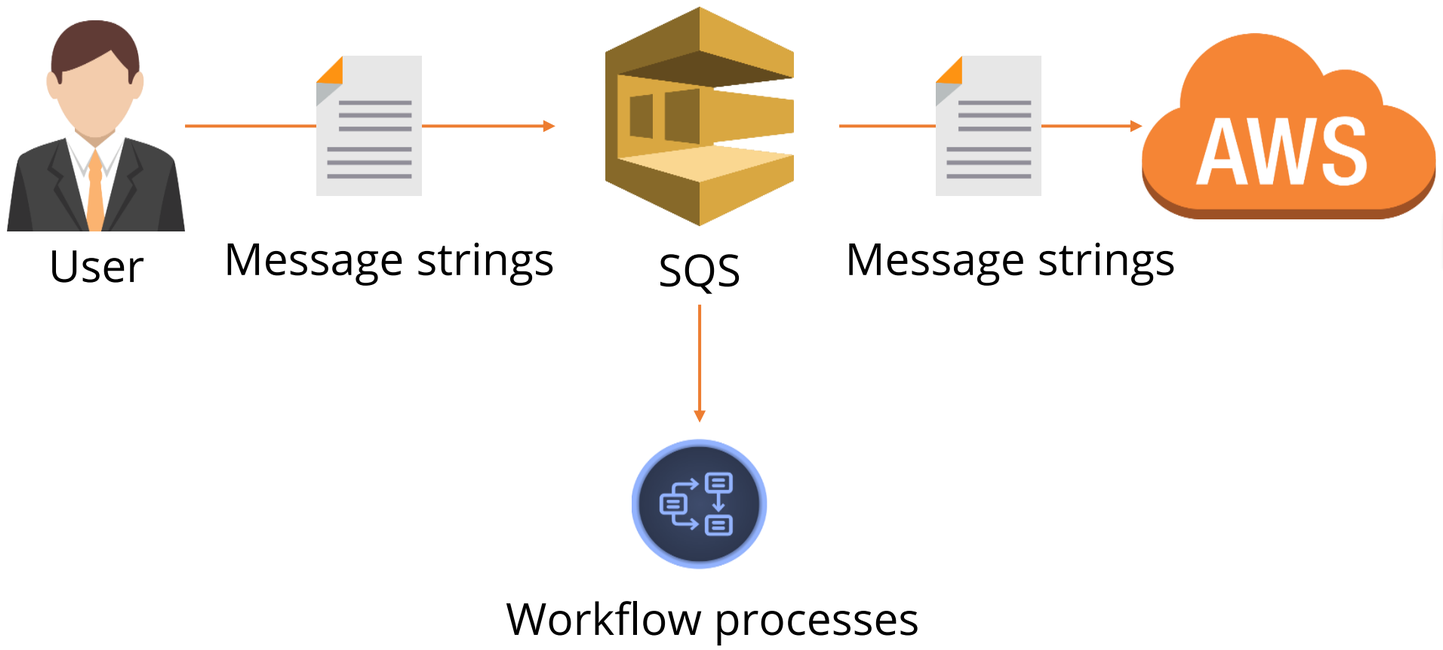
1. Setting up AWS CloudShell
2. To create a serverless web app

TECHNOLOGY

SQS Introduction

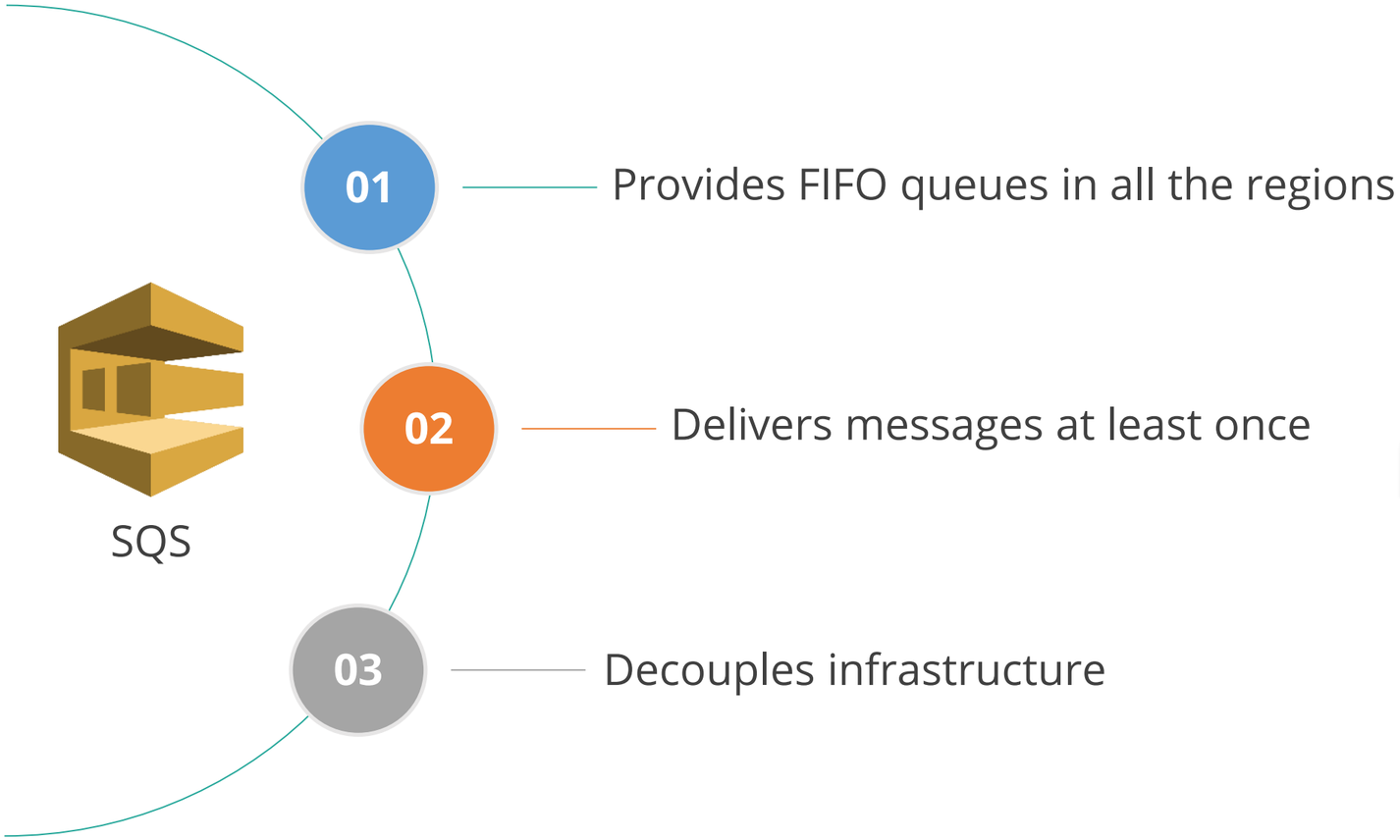
SQS Standard

Amazon SQS(Simple Queue Service) is a fully managed queue service that receives, stores, and sends message strings containing job descriptions across application components and AWS services. SQS follows first-in-first-out (FIFO) standard for sending out messages.



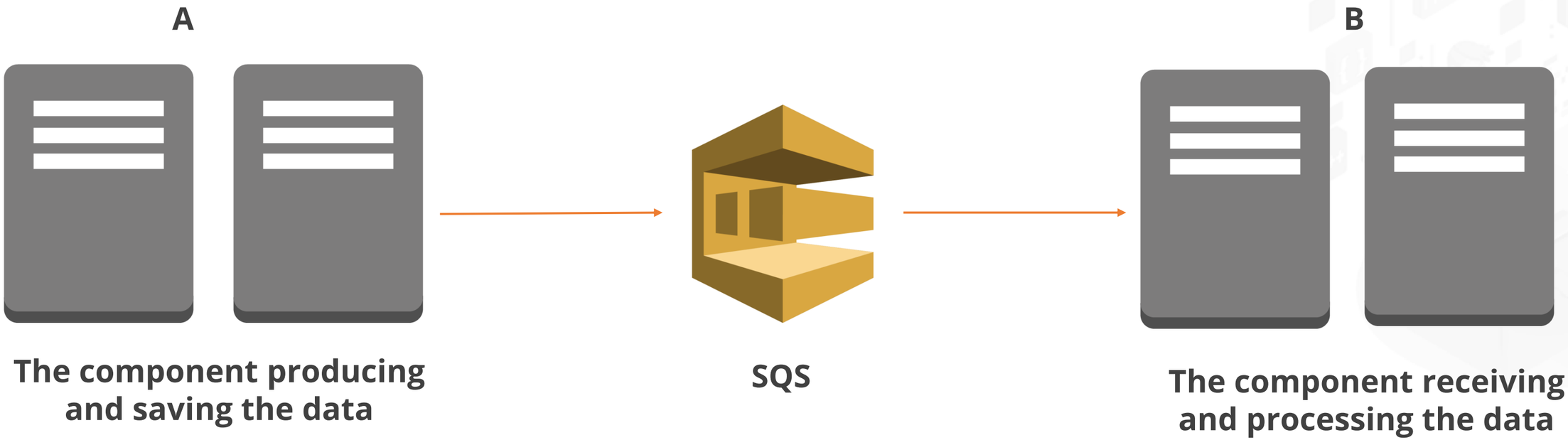
SQS Features

The following are the key features of SQS:



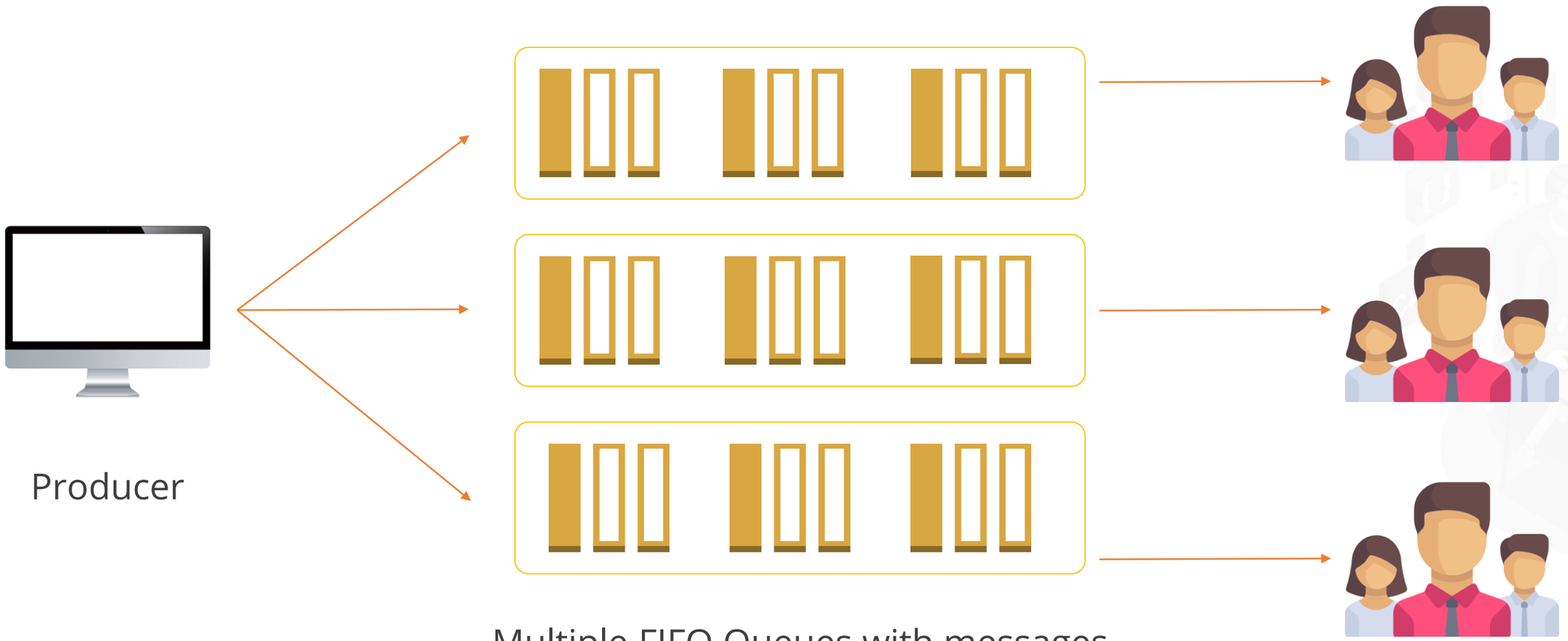
SQS Workflow

SQS allows one to decouple the components of a cloud application, which is an important concept of the AWS best practices in building architectures on the cloud.



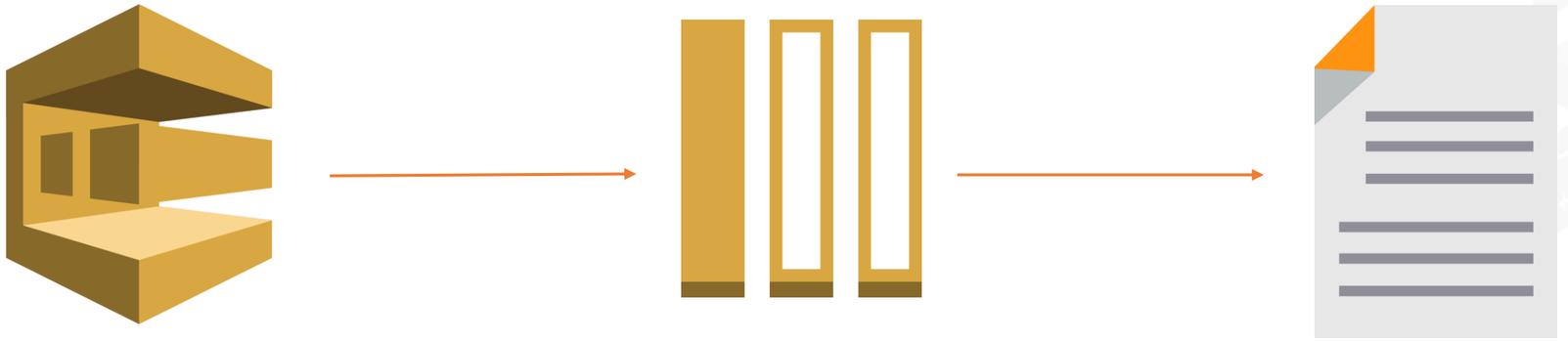
FIFO Queues

FIFO Queues are responsible for temporary storage of messages and text strings until delivered to the intended consumer for processing.



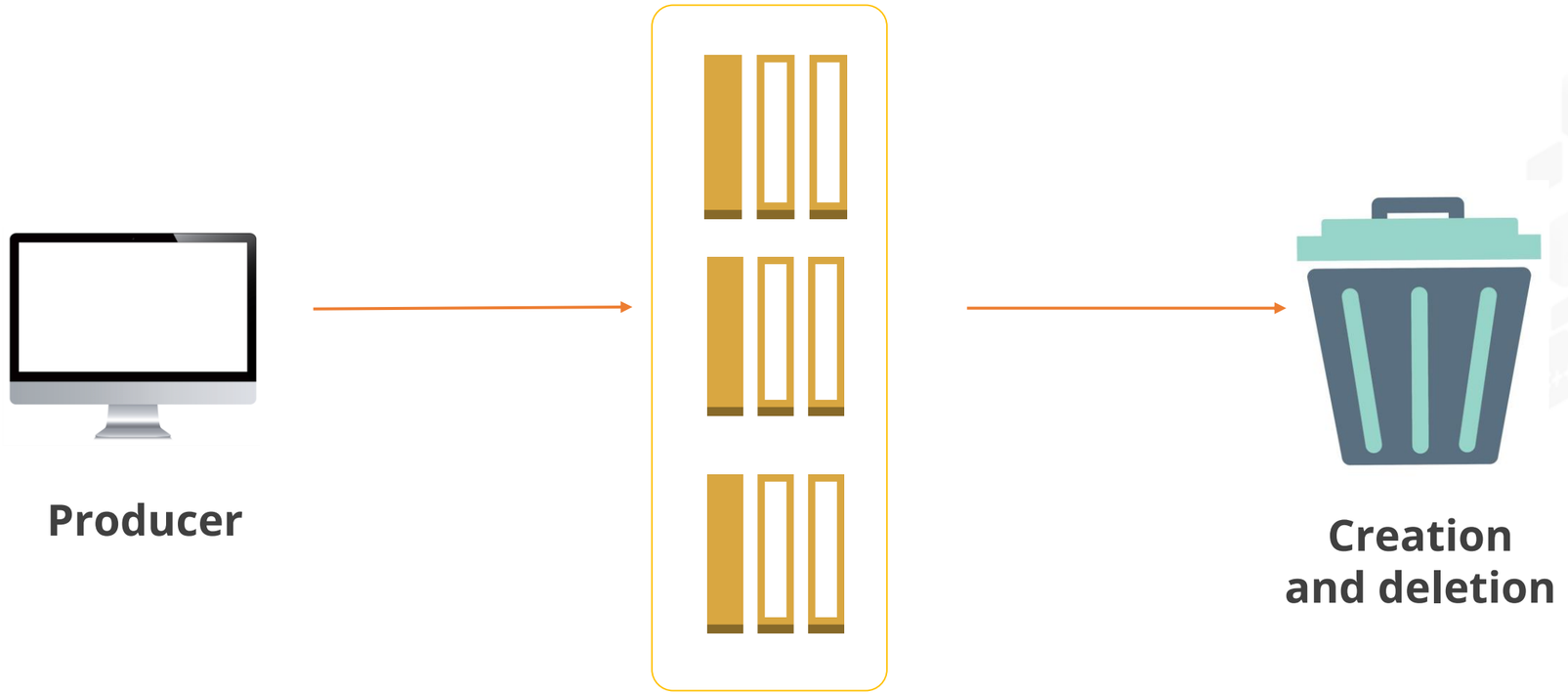
SQS Messages

SQS messages can contain up to 256 KB of text and are billed in chunks of 64 KB of data.



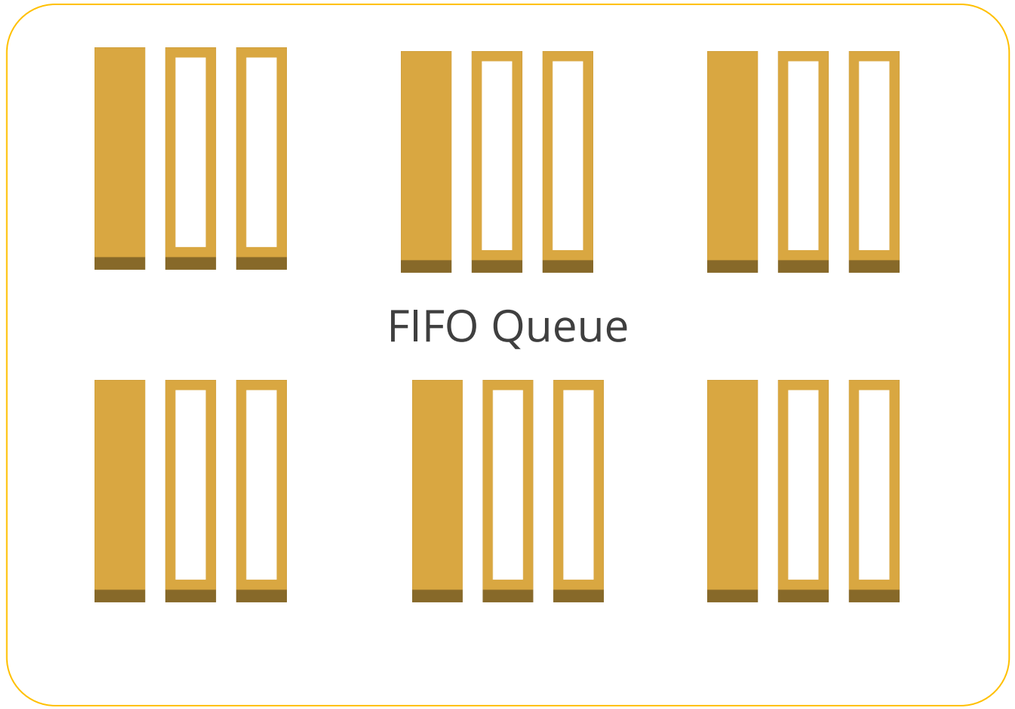
Messages Lifecycle

The message lifecycle is referred to the different stages of a message from its creation to its deletion.



Messages Lifecycle

FIFO Queue store all the sent messages to be sent until it is transferred to the user.



FIFO stands for First In First Out which adheres to the message first produced will be delivered first.

Messages in FIFO Queue are hidden using Visibility Timeout.

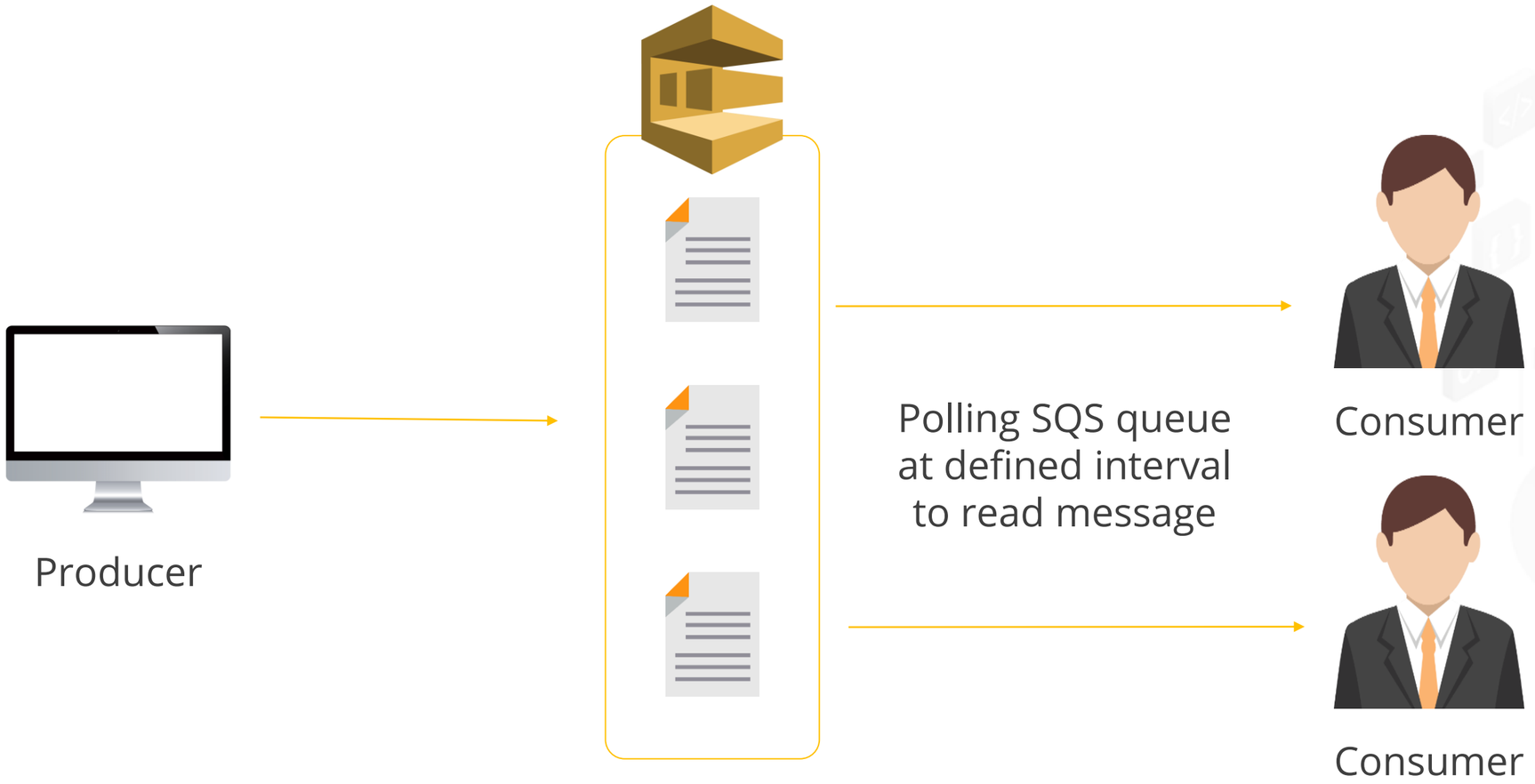
Messages Lifecycle

The sent messages are processed by the user and are automatically deleted by the system.

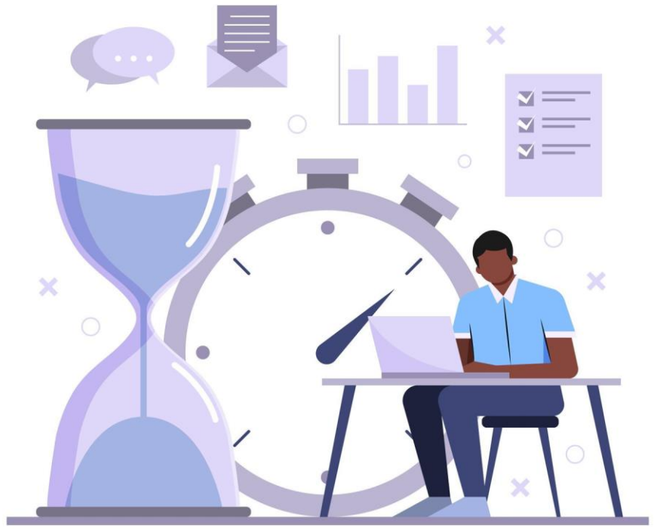
The messages if not received by the user will return an error message to the producer.
The received message can be deleted by both the user and the producer.

Visibility Timeout

Visibility Timeout is the time during which SQS hides the sent message from other users to prevent it from getting read and processed.



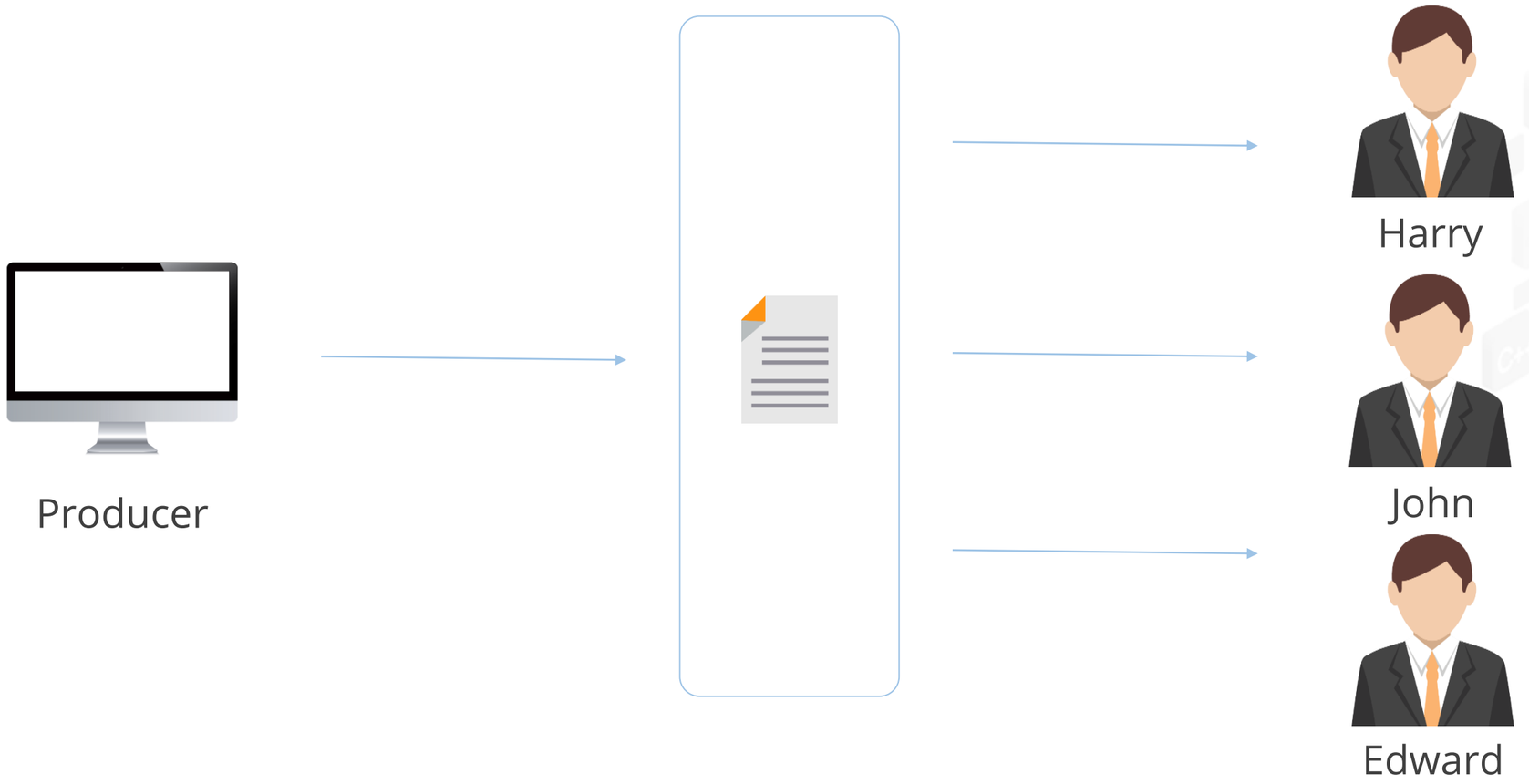
Visibility Timeout



- The default value for a visibility timeout is 30 seconds.
- The minimum value for timeout is 0 seconds and the maximum is 12 hours.
- Visibility Timeout protects the integrity of the message.

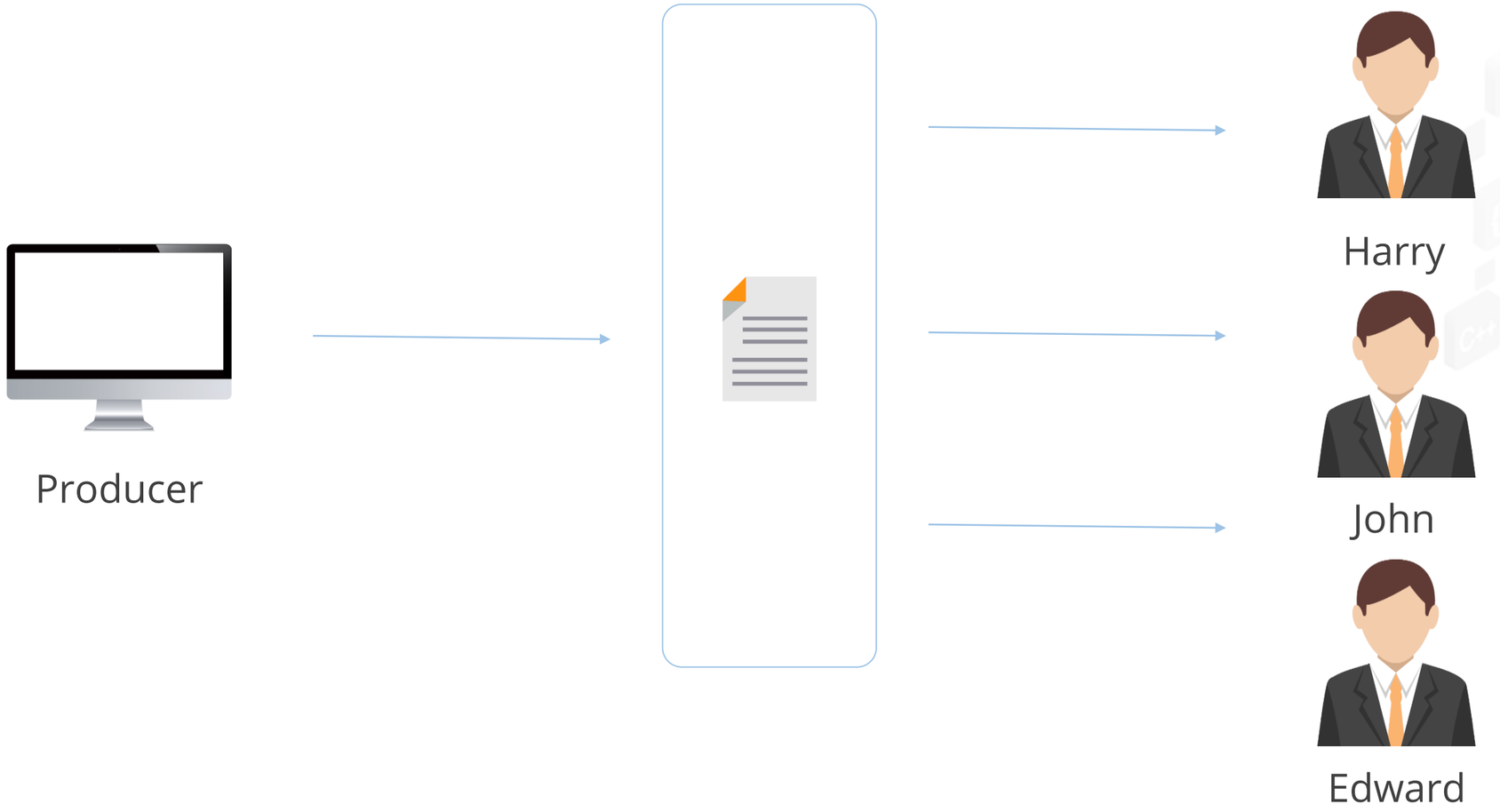
Visibility Timeout: Use Case

A message sent in the queue is meant for Harry. Visibility Timeout keeps the message hidden from others to be read until Harry can receive and process the message.



Visibility Timeout: Use Case

Due to some circumstances, Harry is unable to receive the message. In such a case, the Visibility Timeout timer runs out and the message is sent to John instead.



SQS Security and Performance

Amazon SQS (Simple Query Service)

Amazon SQS is a fully managed queue service that receives, stores, and sends message strings containing job descriptions across application components and AWS services. SQS follows the first-in-first-out (FIFO) standard for sending out messages.

Features of SQS

The features of SQS (Simple Query Services) are as follows:

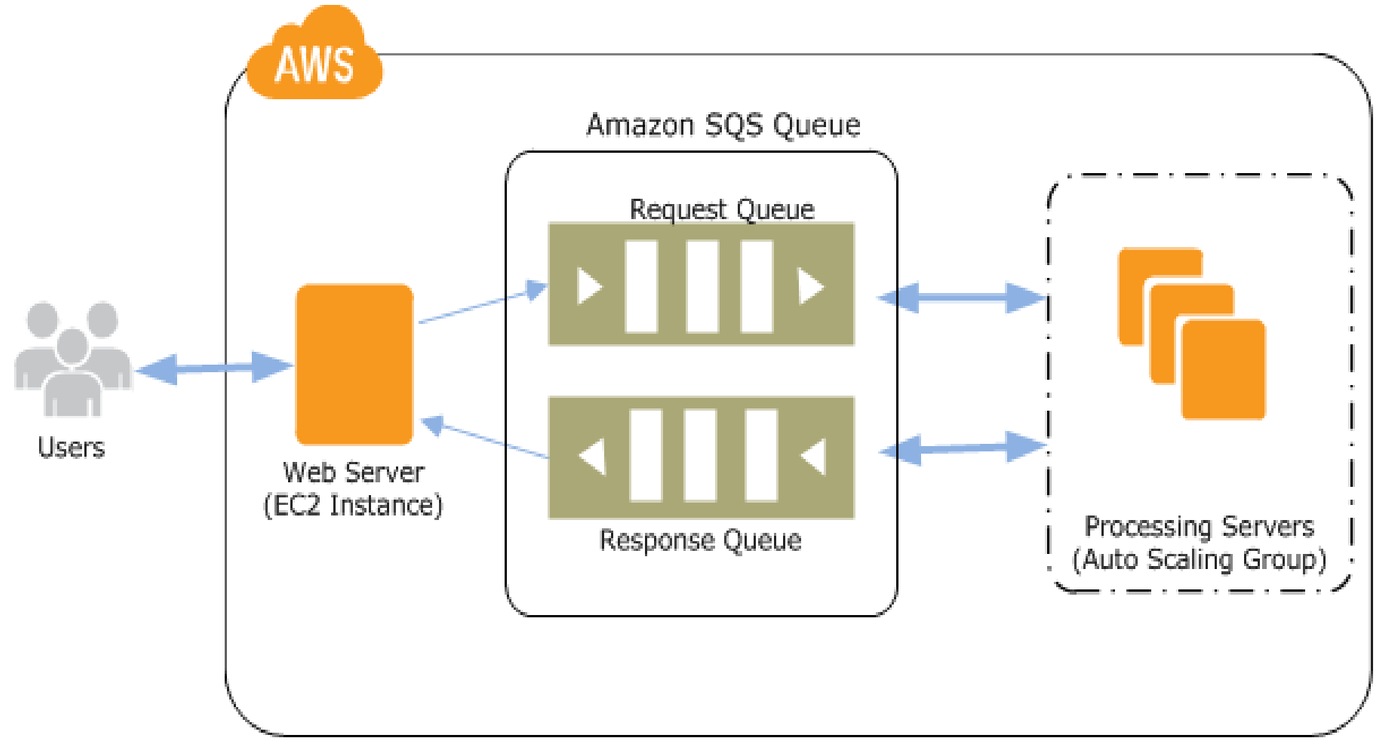
01 Provides FIFO queues in all the regions

02 Delivers messages at least once

03 Decouples the infrastructure

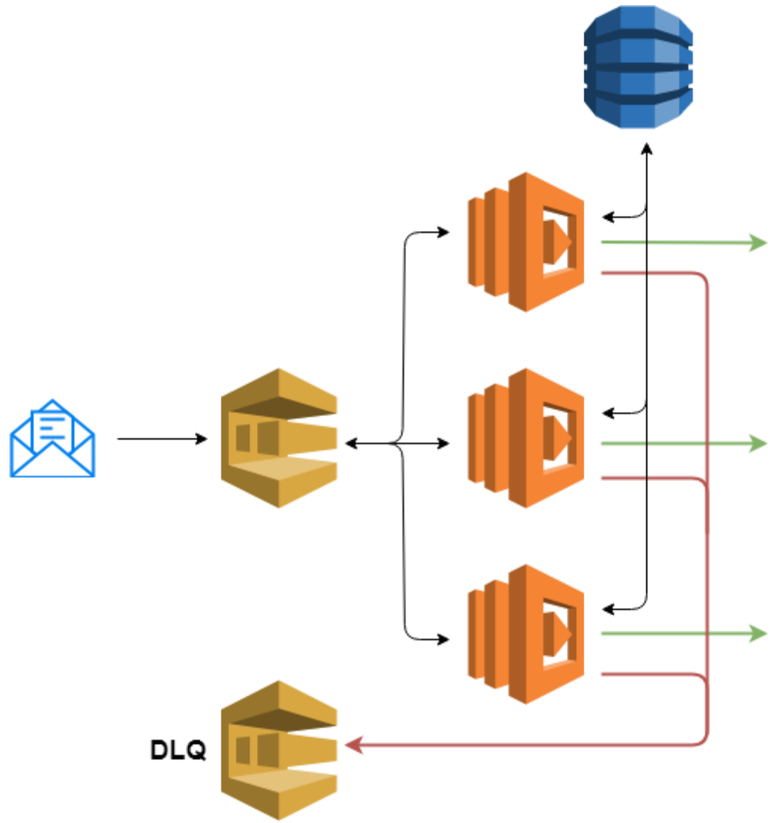
Amazon SQS Workflow

SQS allows users to decouple the components of a cloud application, which is an important concept of the AWS best practices in building architectures on the cloud.



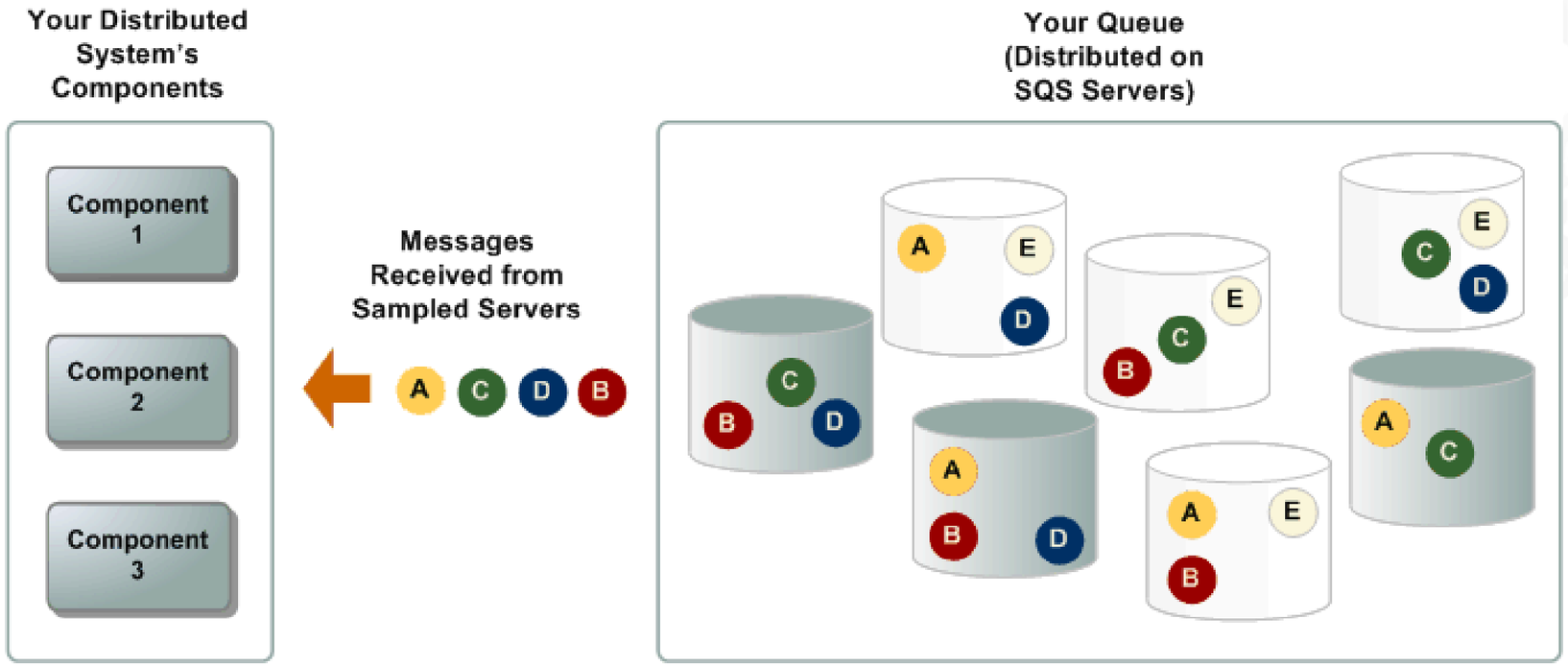
Amazon SQS Deduplication Messages

SQS messages can contain up to 256 KB of text and are billed in chunks of 64 KB of data.



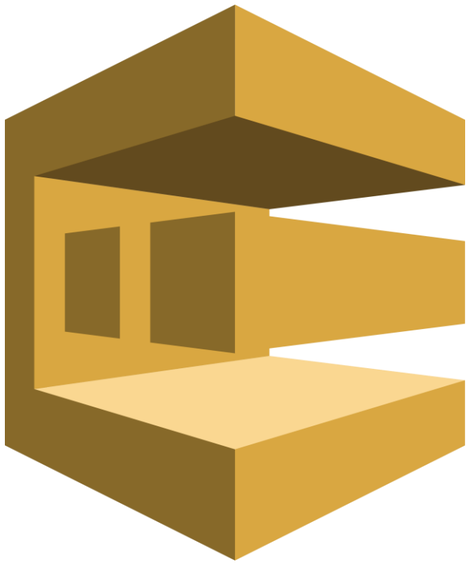
Amazon Long Polling

Long polling receives message wait time is set to greater than 0 (max 20 seconds). It reduces the number of requests, cost and removes a false empty response.



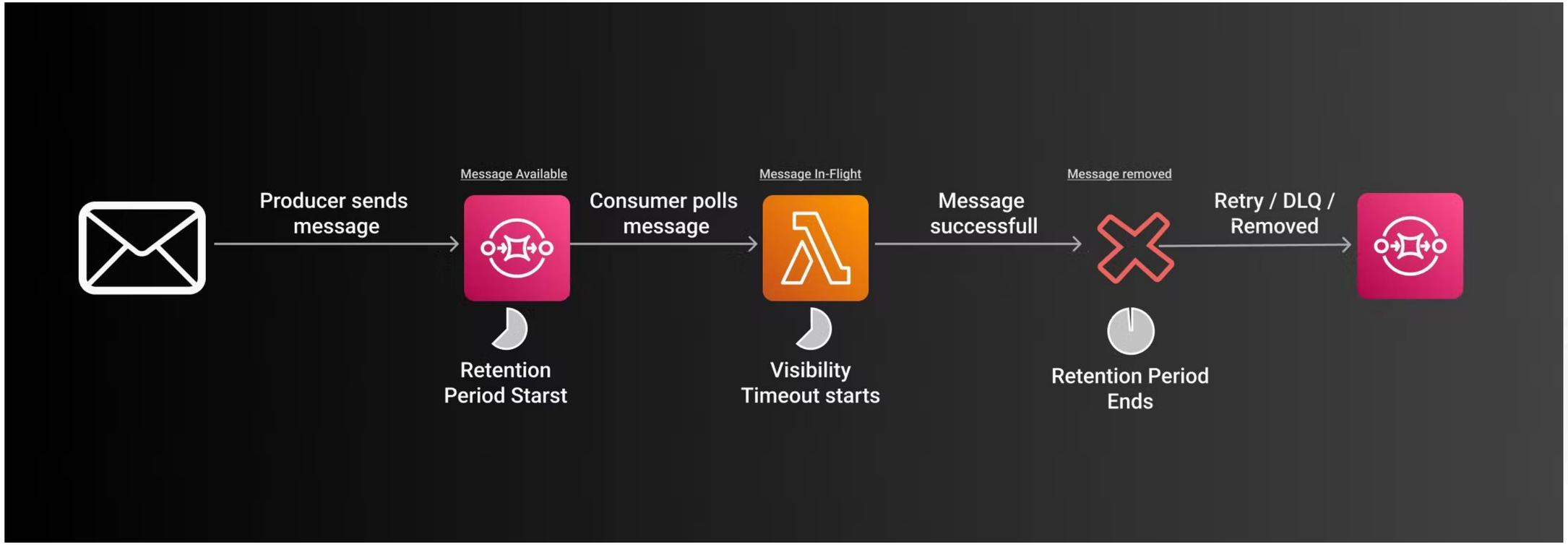
SQS Batching

SQS batching provides a way to handle partial failures when processing batches of messages from SQS. It provides successfully processed messages from being returned to SQS.



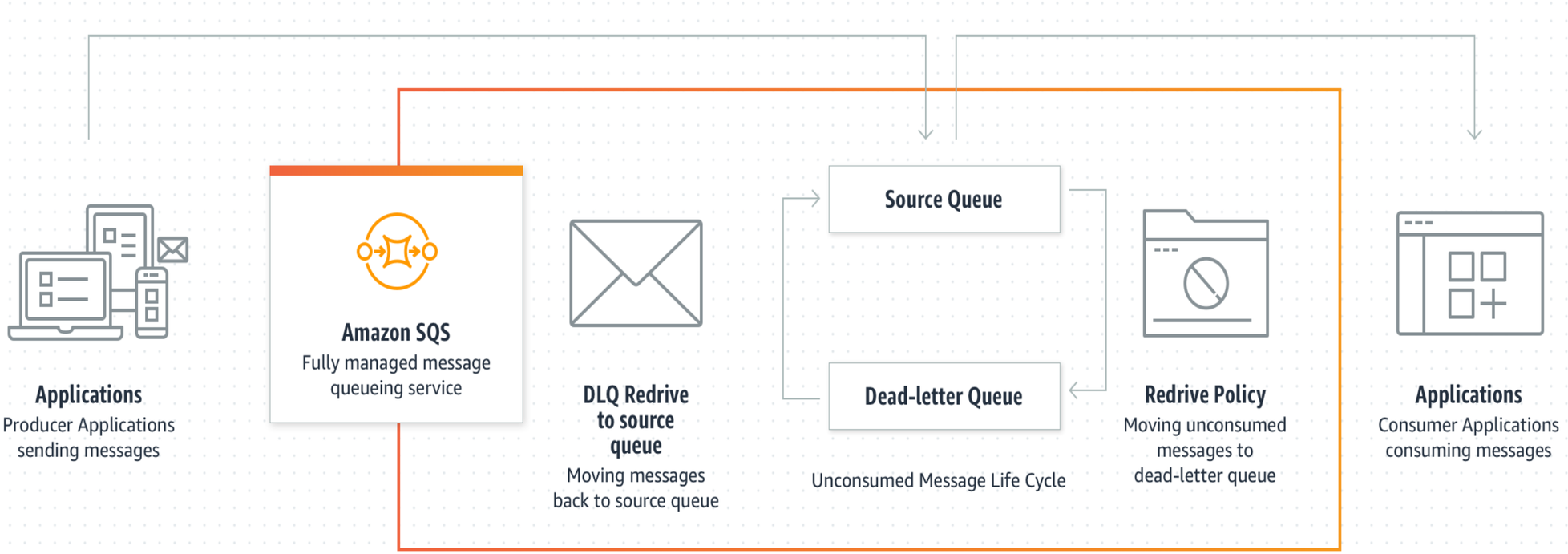
SQS Retention

It can configure the Amazon SQS message retention quota to a value from one minute to fourteen days and automatic deletion of messages after the message retention quota is reached.



SQS DLQ (Dead Letter Queue)

A separate SQS queue that has one or many source queues can send messages that cannot be processed or consumed.



It permits to debugging applications by letting them isolate messages. The isolation of messages that cannot be processed correctly.

SQS Encryption

SQS server-side encryption uses the 256-bit Advanced Encryption Standard (AES-256 GCM algorithm).

The integration with AWS Key Management Service (KMS) centrally manages the keys that protect SQS messages.

SQS Standard Queue, Visibility Timeout, DLQ

Duration: 15 mins

Problem Statement:

You have been asked to create a standard queue and perform the visibility timeout and DLQ

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Create Standard SQS Queue
2. Send and Receive Messages
3. Message Polling
4. Visibility Timeout
5. Create DLQ

SQS FIFO Queue, Deduplication, Message Group

Duration: 15 mins

Problem Statement:

You have been asked to create a SQS FIFO Queue, Deduplication, Message Group

ASSISTED PRACTICE

Assisted Practice: Guidelines

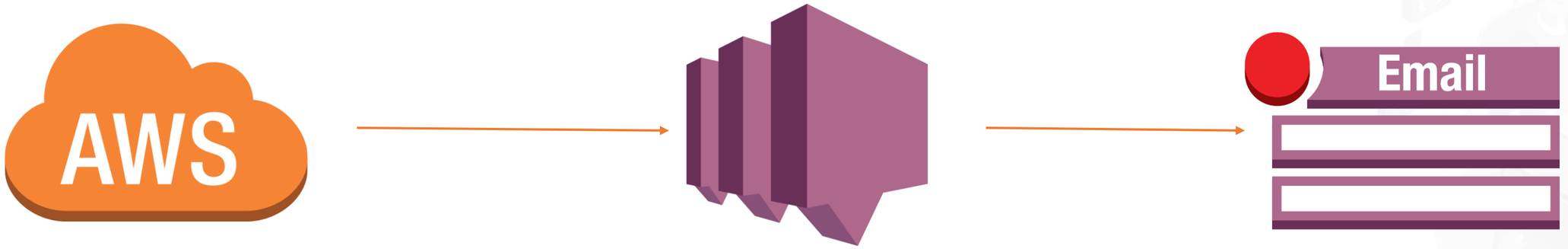
Steps to be followed:

1. Create a Queue
2. Send Messages Including Message Group ID and Duplication ID

Amazon Simple Notification Services (SNS)

Amazon Simple Notification Service

Amazon SNS is a fully managed publication-subscription-based messaging service, used to send push notifications, emails, and SMS messages.



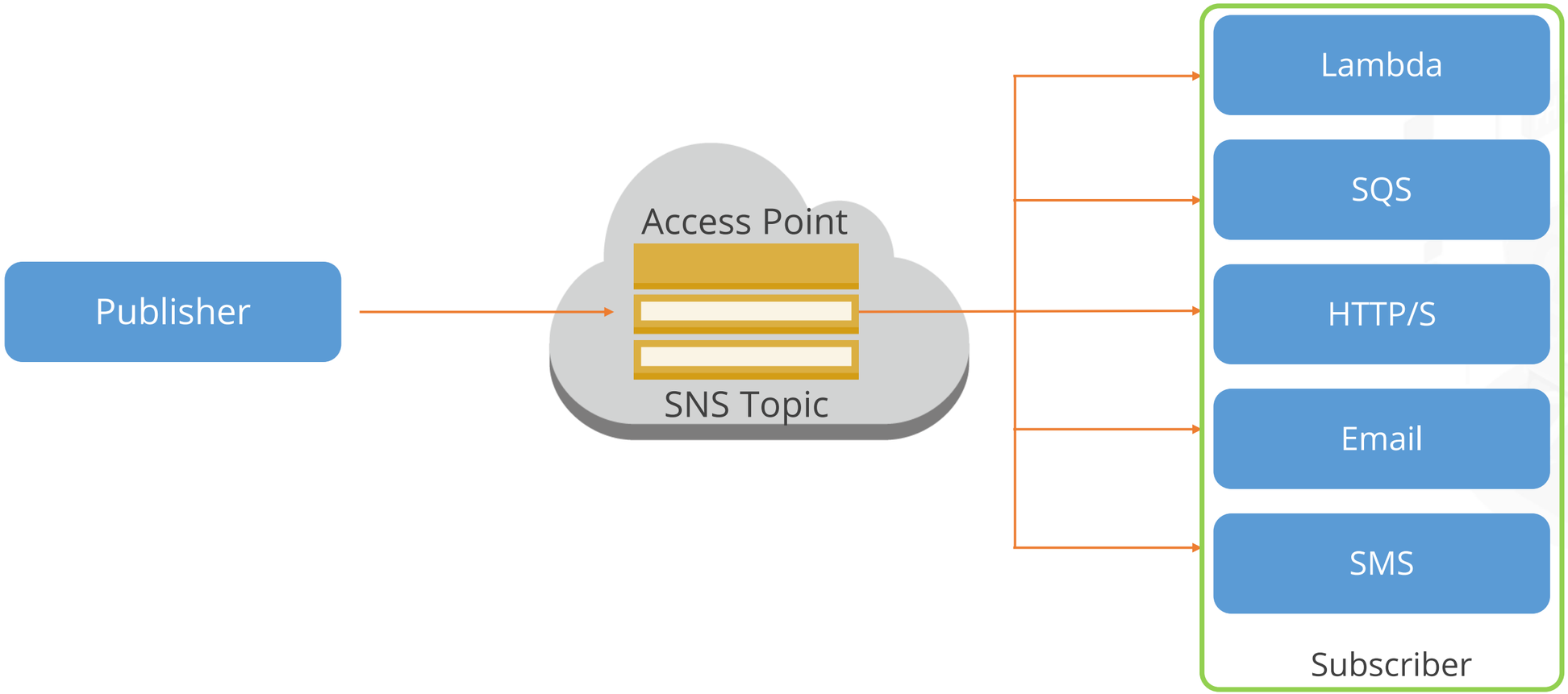
Amazon SNS

The features of Amazon SNS are as follows:

01	Instantaneous push-based delivery
02	Multiple transfer protocol
03	Pay-as-you-go model
04	Simple web-based interface
05	Message durability

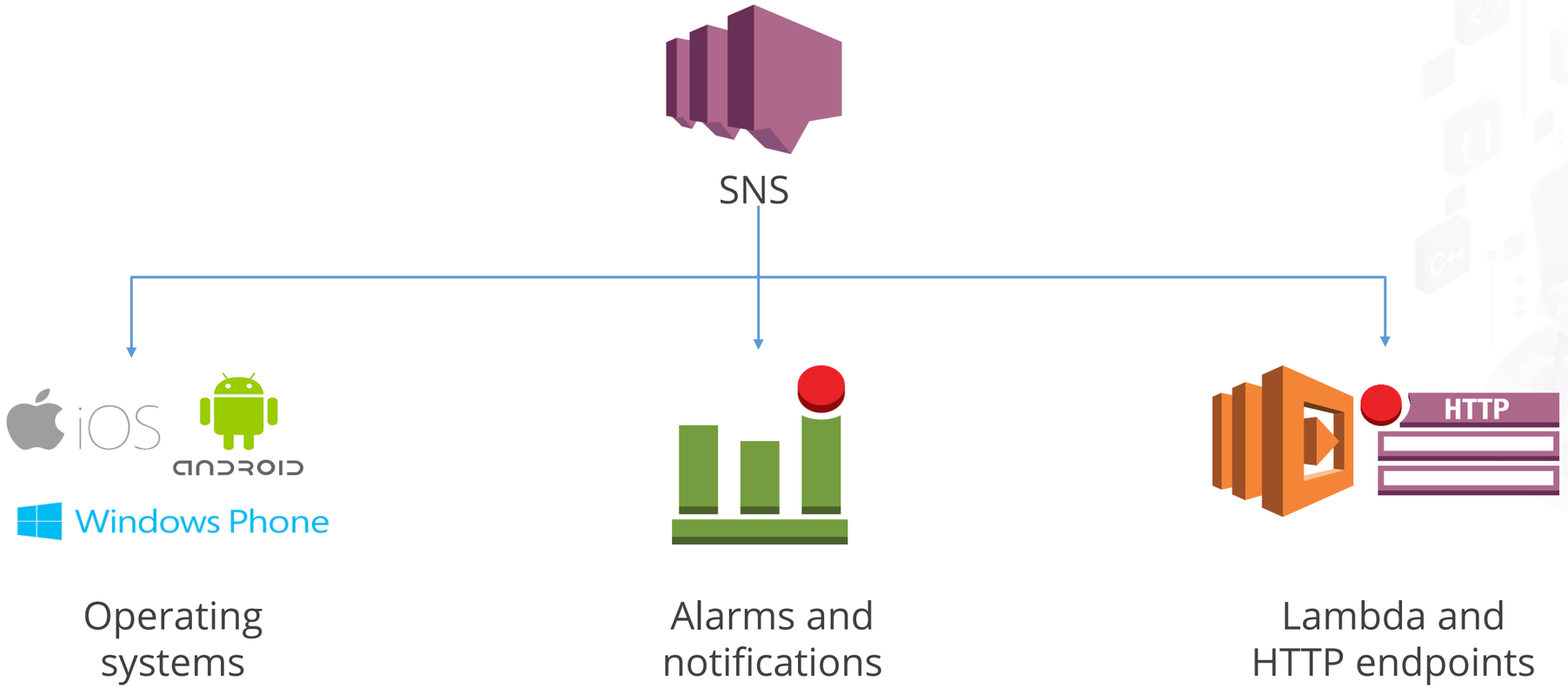
Amazon SNS Topic

An Amazon SNS topic is a communication channel that allows users to send messages and subscribe to notifications.



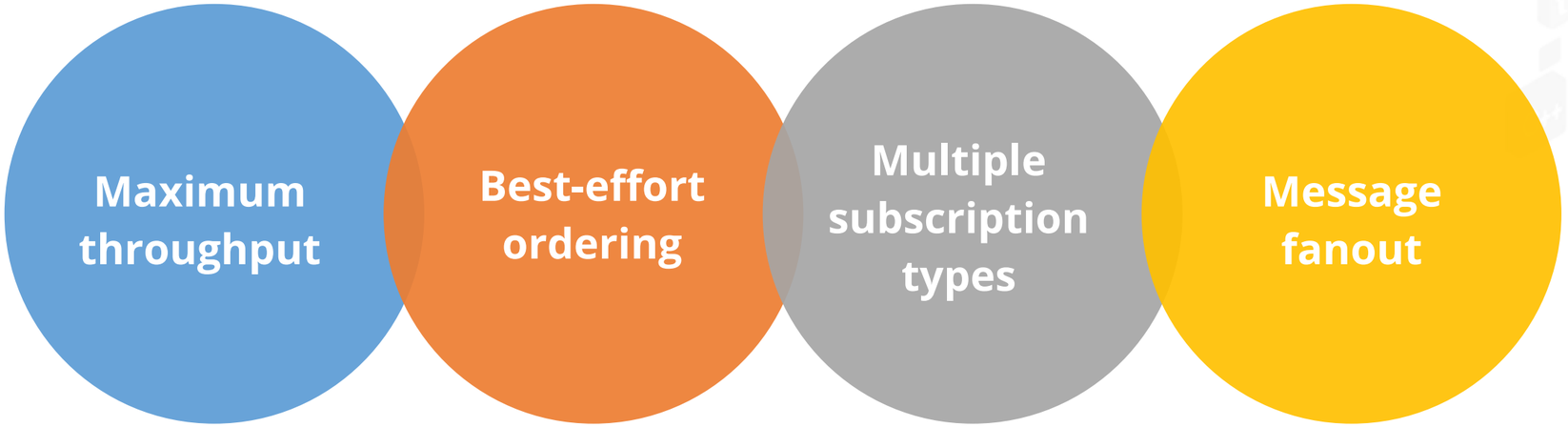
Push Model

In the push model, notifications are sent by Amazon SNS. They can be sent to various operating systems running on mobile devices such as iOS, Android, Windows, and more, in the form of emails and SMS messages.



Standard SNS

Standard topics can be used in many scenarios, as long as users' applications can process a message that arrives more than once and is out of order. The features of standard SNS topic are:



FIFO SNS

FIFO topics are designed to enhance messaging between applications when the order of operations and events is critical or when duplicates cannot be tolerated. The features of FIFO SNS are:

- High throughput
- Strict ordering
- Strict deduplication
- Message fanout

Message Filtering

Message filtering empowers the subscriber applications to create filter policies, so that the applications can receive only the notifications that they are interested in, as opposed to receiving every message published on the topic.

SNS- Dead-Letter Queues (DLQ)

A dead-letter queue is an Amazon SNS subscription that can target messages that cannot be delivered to subscribers successfully.

Logging Amazon SNS API

Amazon SNS is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or Amazon service. CloudTrail log files contain one or more log entries.

Event Bridge

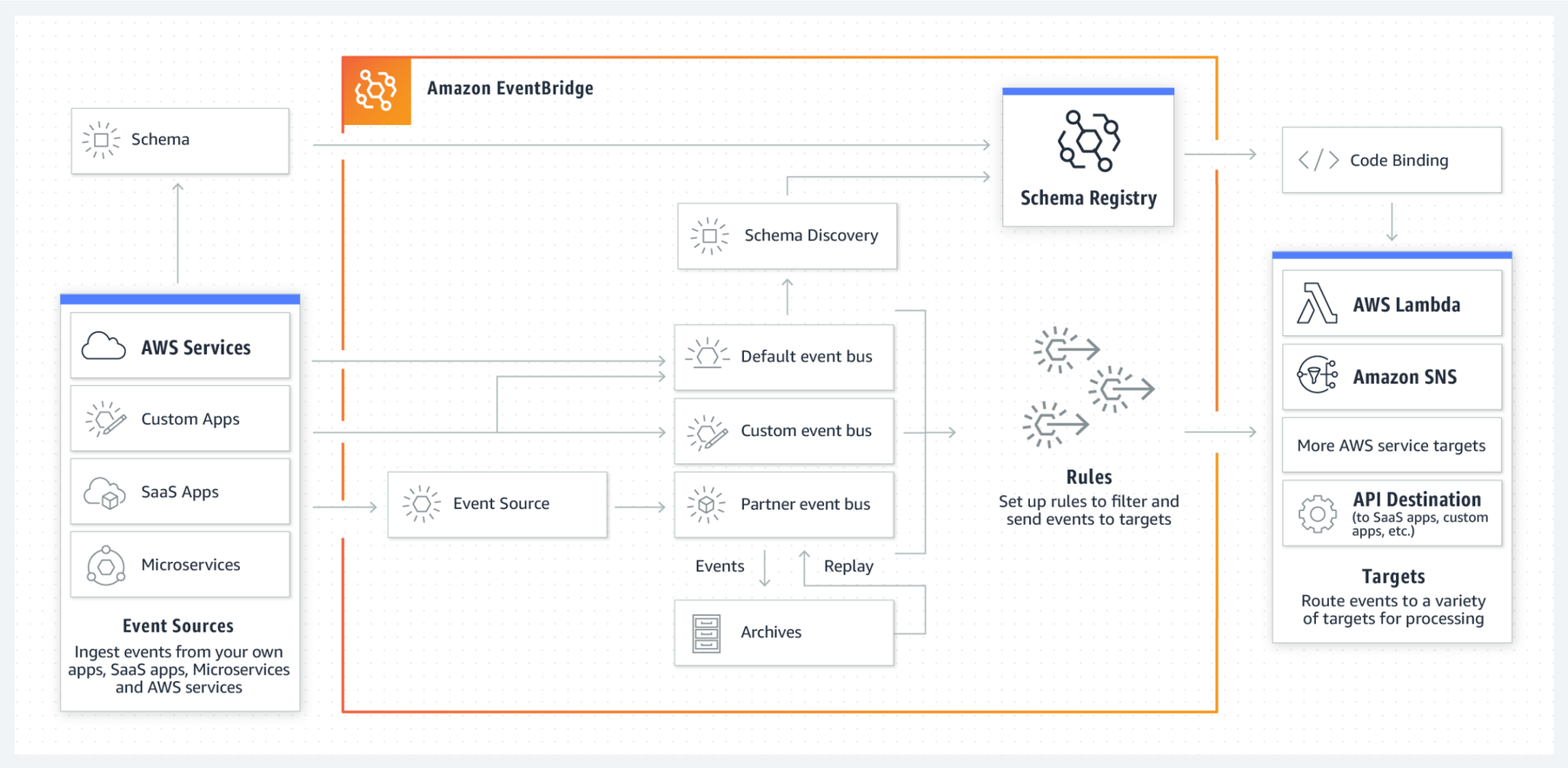
Amazon Event Bridge

Amazon Event Bridge is a serverless event bus service that allows a user to connect applications to data from various sources.

Event Bridge delivers real-time data from a user's apps, software as a service (SaaS) apps, AWS services to AWS Lambda functions, HTTP invocation endpoints with API destinations, and event buses in other AWS accounts.

Amazon Event Bridge

The following diagram depicts the working of an event bridge:



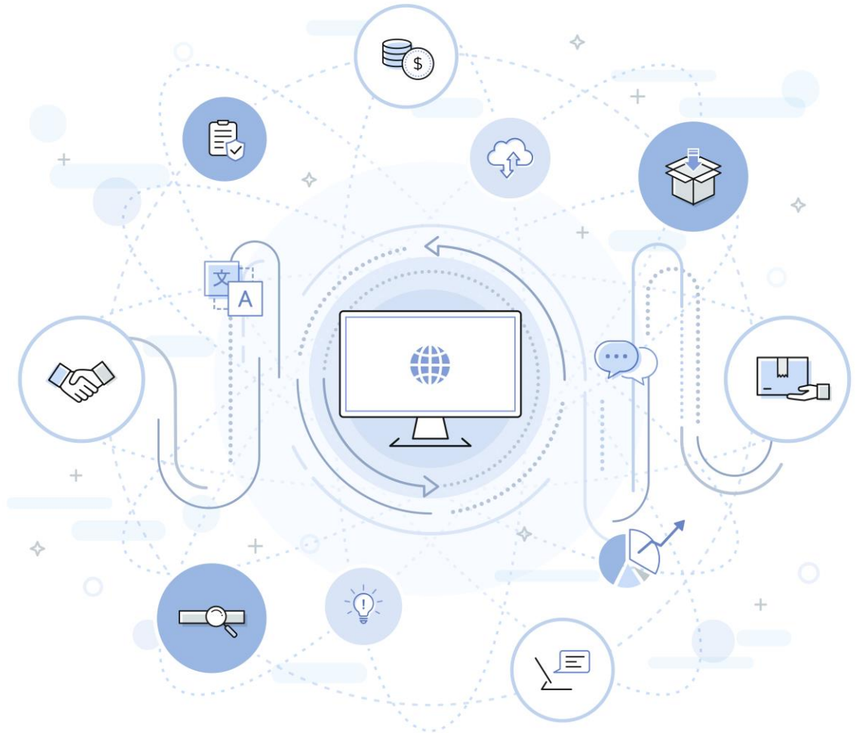
Amazon Event Bridge

The features of event bridge are as follows:

1. Advanced event rules filtering
2. Content-based event filtering
3. Schema registry
4. Message transformation
5. Custom events
6. Archive and replay events
7. SaaS Apps Integration
8. API destination

Event Bus

An event bus receives events from a source and routes them to rules associated with that event bus. A custom event bus can receive events from custom applications and services.



SaaS Partner Bus

Event bridge also enables the users to connect the applications with a range of SaaS partners without having to worry about building and maintaining custom infrastructure.

SNS Vs Event Bridge

	Amazon SNS	Amazon Event Bridge
Number of targets	10 million(soft)	5 targets per rule
Public throughput	No	Yes
Receives events from AWS CloudTrail	No	Yes
Public visibility	Can create public topics	Cannot create public buses
Cross-region	You can subscribe your AWS Lambda functions to an amazon SNS topics in any region	Targets must be in the same region. You can publish across regions to another event bus.
FIFO ordering available	Yes	No
SaaS integration	No	Yes

SNS Topic, Fanout, S3 Event Notification

Duration: 15 mins

Problem Statement:

You have been asked to create a SNS Topic, Fanout, S3 Event Notification

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Login to your AWS lab
2. Create an SNS Topic
3. Subscribe to an SNS Topic

Kinesis

Amazon Kinesis

Amazon Kinesis is a fully managed and scalable service that allows real-time collection, processing, and analysis of streaming data.

Amazon Kinesis

Amazon Kinesis Capabilities

Kinesis Data Streams: It allows users to build custom applications to process data in real time.

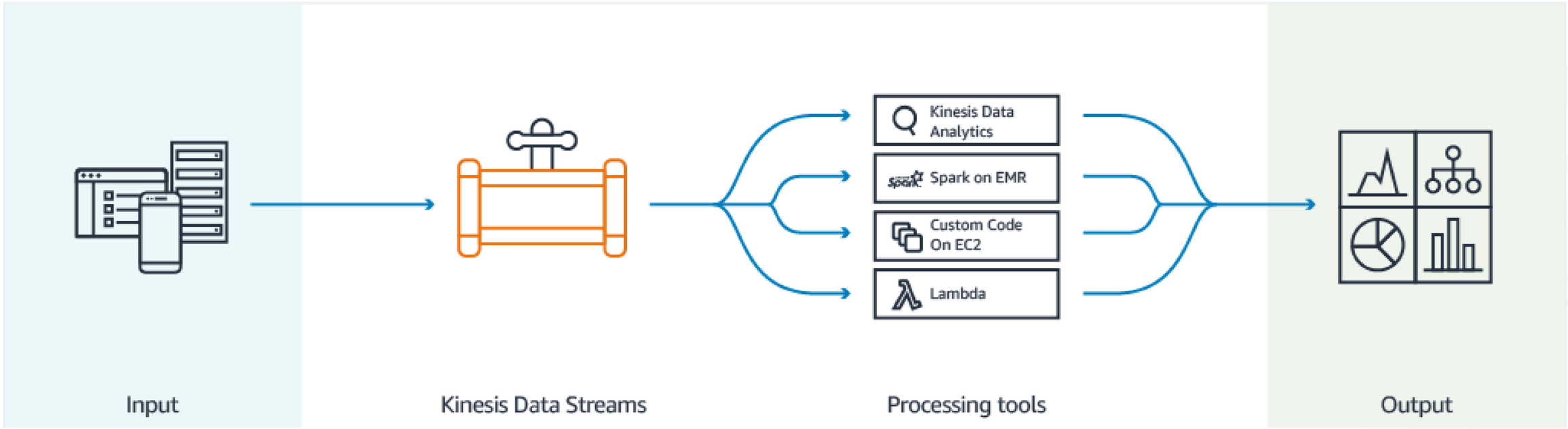
Kinesis Videos Streams: It allows users to securely stream videos from connected devices to AWS for processing.

Kinesis Data Firehose: It allows users to capture, transform, and load data into AWS data stores.

Kinesis Data Analytics: It runs queries against the data in real time.

Amazon Kinesis Data Streams

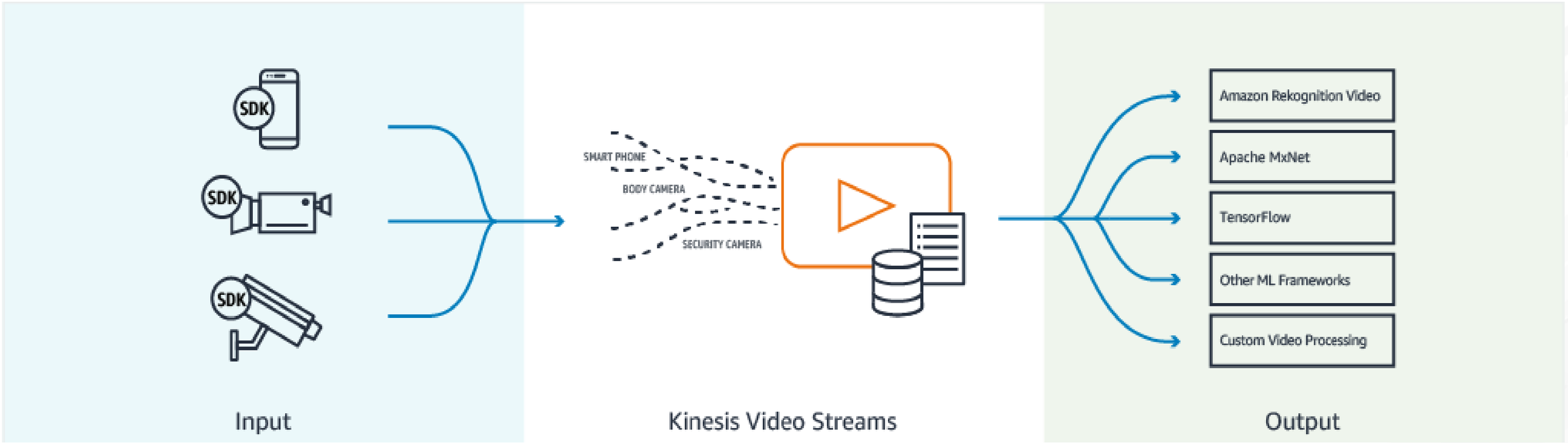
The following diagram shows the working of Amazon Kinesis Data Streams:



Source: aws.amazon.com

Amazon Kinesis Video Streams

The following diagram shows the working of Amazon Kinesis Video Streams:

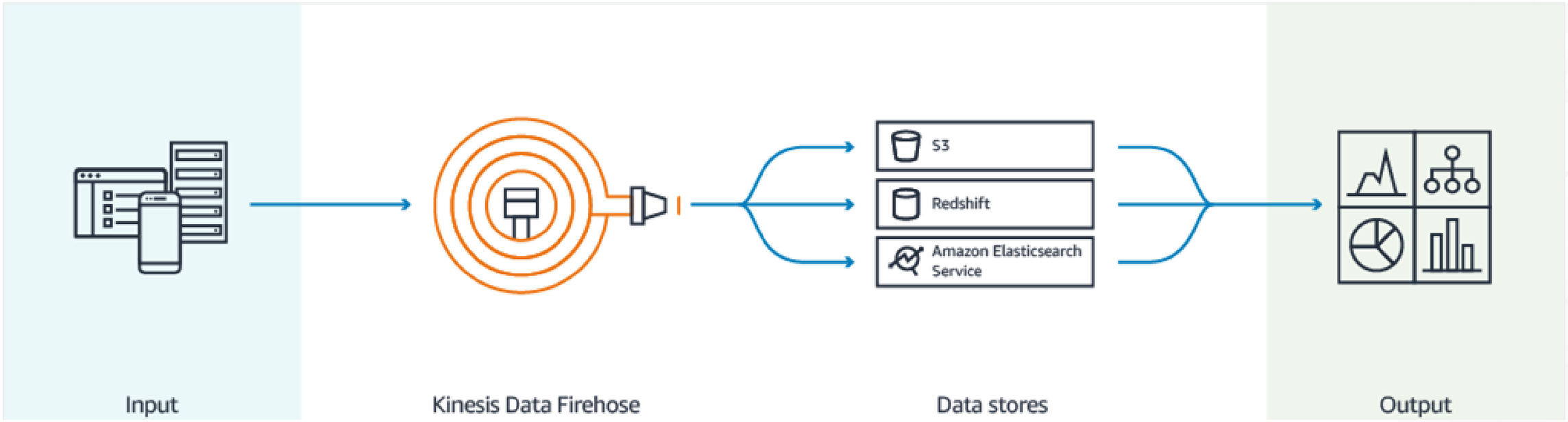


Step 1 **Step 2** **Step 3**

Source: aws.amazon.com

Amazon Kinesis Data Firehose

The following diagram shows the working of the Amazon Kinesis Data Firehose:



Step 1 **Step 2** **Step 3** **Step 4**

Source: aws.amazon.com

Amazon Kinesis Data Analytics

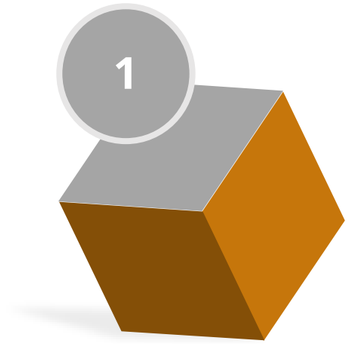
The following diagram shows the working of Amazon Kinesis Data Analytics:

Step 1 **Step 2** **Step 3**

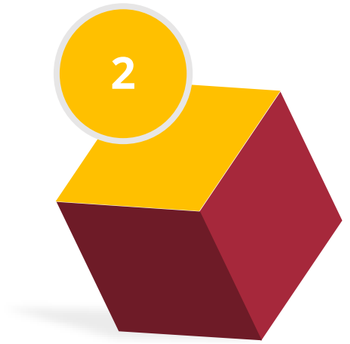
Source: aws.amazon.com

Kinesis Shards

A shard consists of a sequence of data records in a stream. It is a base throughput unit of a Kinesis data stream. The features of kinesis shards are as follows:



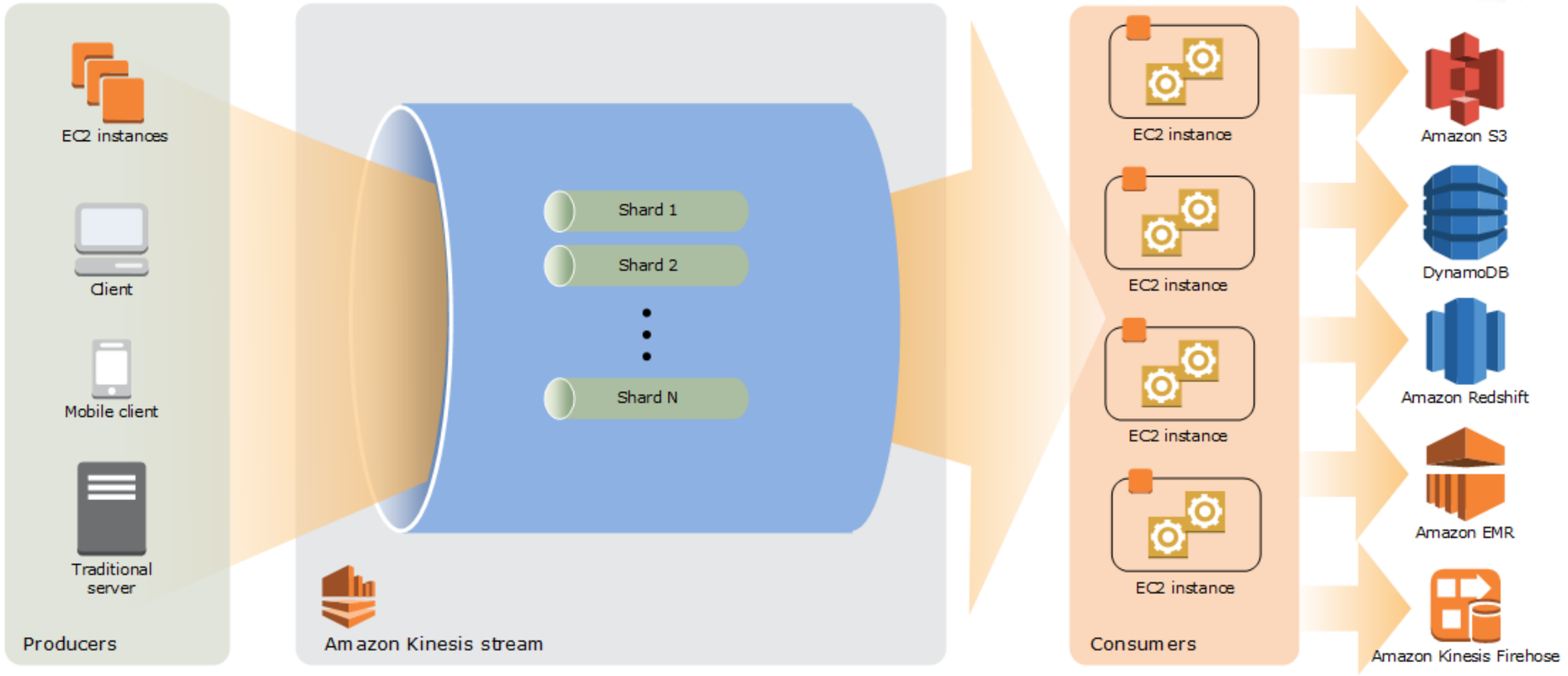
Supports up to five transactions per second for reads



Supports up to a maximum total data read rate of 2MB per second

Kinesis Shards

The following diagram depicts the workflow of kinesis shards:



Partition Key

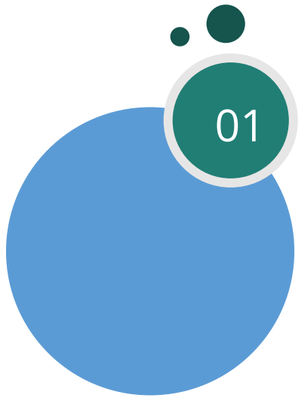
A partition key is used to group data by shard within a stream. Partition keys are Unicode strings, with a maximum length limit of 256 characters for each key.

The partition key associated with each data record determines to which shard the data record belongs.

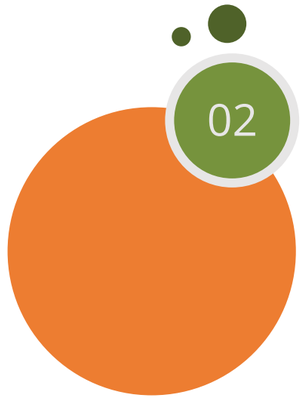
When an application puts data into a stream, it must specify a partition key.

Retention

The retention period is the length of time, that data records are accessible after they are added to the stream. The features of retention are as follows:



Can be increased up to 8760 hours (365 days) and can be decreased up to 24 hours



Set to a default of 24 hours after creation

Iterator

The iterator specifies the shard position from which to start with the reading data records sequentially. The position is specified using the sequence number of a data record in a shard.

- The iterator expires 5 minutes after it is returned to the requester.
- The users must specify the shard iterator type.

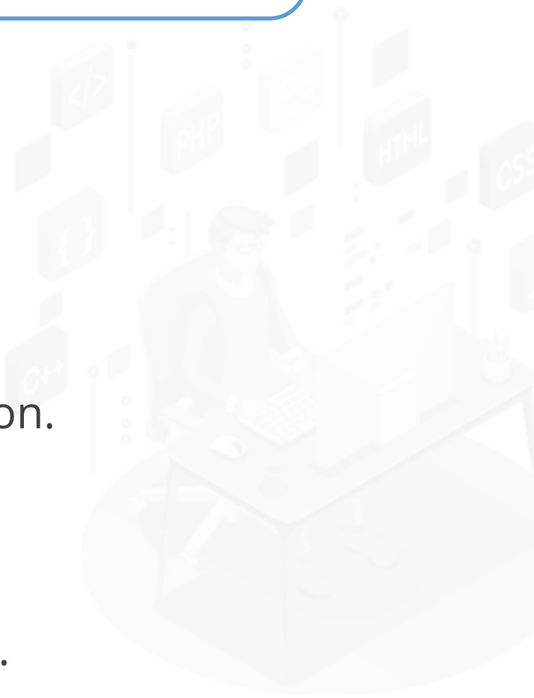
Resharding

Resharding lets the users adjust the number of shards in the stream. It helps to adapt to changes in the rate of data flow through the stream. The features of resharding are as follows:

01 Resharding is always pairwise and it’s an advanced operation.

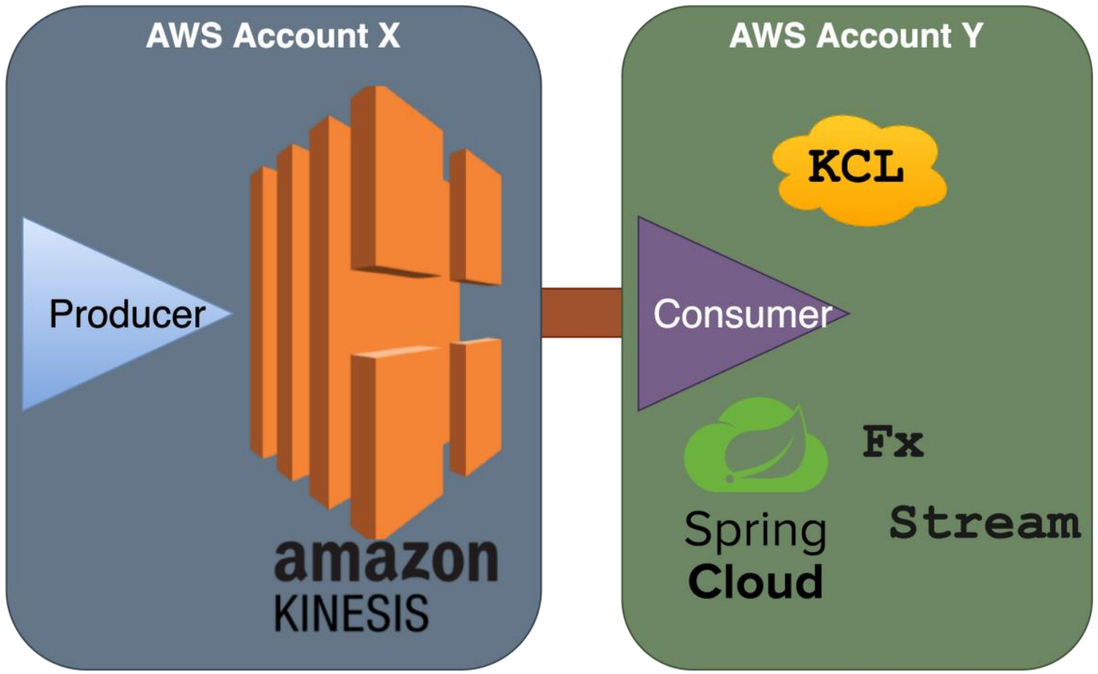
02 The users cannot split into more than two shards in a single operation.

03 The users cannot merge more than two shards in a single operation.



Kinesis Client Library (KCL)

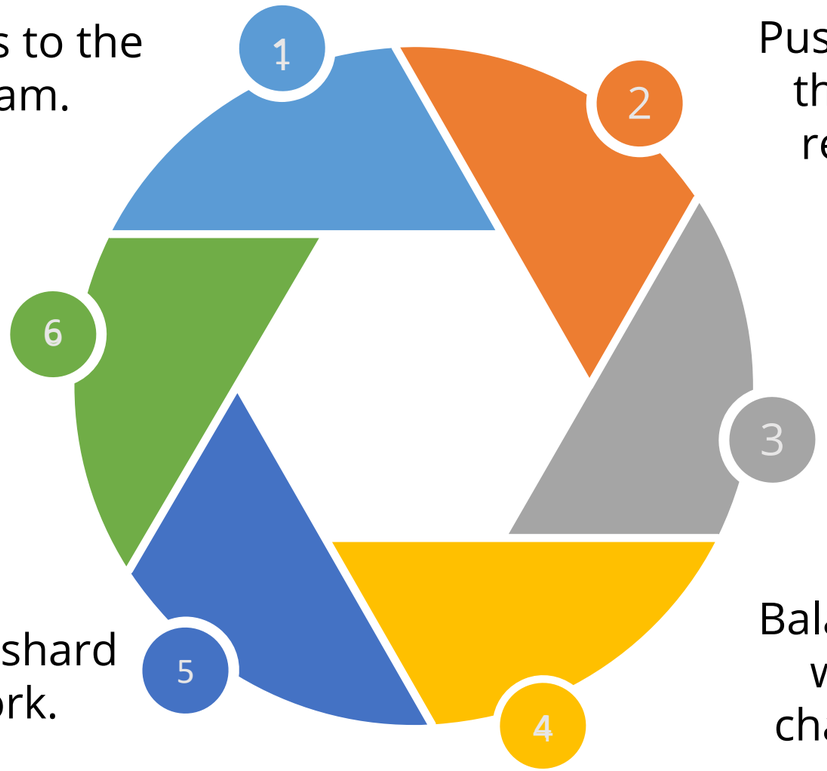
KCL helps in developing custom consumer applications that can process data from KDS data streams. It instantiates a record processor for every shard it manages and pulls data records from the data stream.



KCL-Kinesis Client Library

The features of KCL are as follows:

1. KCL connects to the data stream.
2. Pushes the records to the corresponding record processor.
3. Checkpoints processed record.
4. Balances shard – worker associations when the worker instance count changes or when the data stream is resharded.
5. Uses leases to coordinate shard associations with its work.
6. Enumerates the shards within the data stream.



Kinesis Producer Library

KPL (Kinesis Producer Library) simplifies producer application development, allowing developers to achieve high write throughput to a Kinesis data stream. The key concepts of KPL are as follows:

Collection: It refers to batching multiple Kinesis Data Streams records and sending them in a single HTTP request with a call to the API operations.

Records: The team record without a qualifier, we refer to a KPL user record.

Batching: Performing a single action on multiple items instead of repeatedly performing the action on each individual item.

Aggregation: Storage of multiple records in a Kinesis Data Streams record.

Permission and Access Management

The IAM integration with Kinesis Data Streams allows users to control who in their organization can perform a task using certain Kinesis Data Streams API actions and who can use specific AWS resources.

Permission and Access Management

An IAM policy is a JSON document that contains one or more statements. A statement is composed of the following components:

```
{
 "Statement":[{
   "Effect":"effect",
   "Action":"action",
   "Resource":"arn",
   "Condition":{
     "condition":{
       "key":"value"
       }
     }
   }
 ]
}
```

1. **Effect**: The effect can be **Allow** or **Deny**. By default, IAM users do not have authority to use resources or API activities, therefore all requests are denied.

2. **Action**: The action is an API action for which permission is being granted or denied.

3. **Resource**: The Amazon Resource Name (ARN) must be used to define a resource in the statement.

4. **Condition**: The conditions are optional. They can be used to control when the policy will be in effect.

Policies for Kinesis Data Streams

Example 1:

The following policy could be applied to users who want to get data from a specific stream.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:Get*",
                "kinesis:DescribeStreamSummary"
            ],
            "Resource": [
                "arn:aws:kinesis:us-east-
1:111122223333:stream/stream1"
            ]
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:ListStreams"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Policies for Kinesis Data Streams

Example 2:

The following policy could be applied to users who want to add data records to all streams.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:PutRecord"
            ],
            "Resource": [
                "arn:aws:kinesis:us-east-1:111122223333:stream/*"
            ]
        }
    ]
}
```

Policies for Kinesis Data Streams

Example 3:

The following policy could be applied to users who want administrative control over a specific stream.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kinesis:*",
            "Resource": [
                "arn:aws:kinesis:us-east-1:111122223333:stream/stream1"
            ]
        }
    ]
}
```

Policies for Kinesis Data Streams

Example 4:

The following policy allows a user or group to use any Kinesis Data Streams operation on any stream.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kinesis:*",
            "Resource": [
                "arn:aws:kinesis:*:111122223333:stream/*"
            ]
        }
    ]
}
```

Create Kinesis Data Stream

Duration: 15 mins

Problem Statement:

You have been asked to set up the prerequisites and create a Kinesis Data Stream

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Setting up the Kinesis data Firehouse and storing in S3

Publish Streaming data to firehouse from CLI and store in S3

Duration: 15 mins

Problem Statement:

You have been asked to publish streaming data to firehouse from CLI and store in S3

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Login to your AWS lab
2. Create an SNS Topic
3. Subscribe to an SNS Topic

Key Takeaways

- Amazon SQS is a fast, reliable, scalable, and fully managed message queuing service. It can be used to control workflow processes.
- Amazon SNS is a fully managed publication-subscription based messaging service used to send push notifications, emails, and SMS messages.
- Amazon Lambda is a computing service that runs code in response to events and automatically manages the computing resources required by that code.
- Amazon Kinesis is a fully managed and scalable service that allows real-time collection, processing, and analysis of streaming data.

Launch a Lambda Function Performing a Task of Copying Contents Across S3 Bucket

Duration: 30 mins

Project agenda: To launch the lambda function, perform the trigger operation on the lambda

Description:

Highly Available Architectures on AWS have business-transforming capabilities. As a solution architect, you have to replicate the contents of a bucket to the other buckets deployed in different regions, thereby making the resources highly available.

Perform the following:

- Create an S3 Bucket
- Create the Lambda Function
- Test the Lambda Function
- Add Trigger
- Add Destination

LESSON-END PROJECT

TECHNOLOGY

Thank You

simplilearn