
aws
certified
Solutions
Architect
Professional
DOLFINed

APPLICATION INTEGRATION

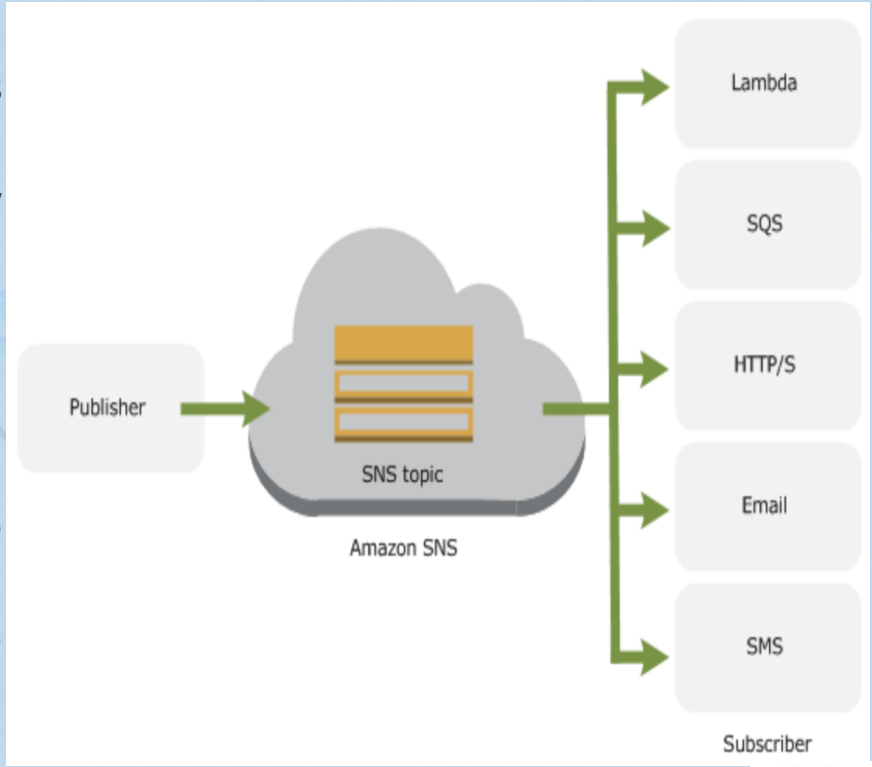
AMAZON SIMPLE NOTIFICATION SERVICE (SNS)

AWS SNS

In Amazon SNS, there are two types of clients—publishers and subscribers—also referred to as producers and consumers.

- Publishers communicate asynchronously with subscribers by producing and sending a message to a topic,

- Subscribers (web servers, email addresses, Amazon SQS queues, AWS Lambda functions) consume or receive the message or notification over one of the supported protocols (Amazon SQS, HTTP/S, email, SMS, Lambda, Application) when they are subscribed to the topic.



AWS SNS

SNS allows you to:

- Send push messages (not Poll messages like SQS)
- Scale as your needs grow
- Engage audiences directly or all at once
- Deliver messages worldwide and across multiple transport protocols
- Easily connect with other AWS services such as CloudWatch, SQS, Lambda, S3
- Message delivery analysis
- Usage based pricing

SNS Reliability

- Amazon SNS stores all topic and message information within Amazon's proven network infrastructure and datacenters.
  - At least three copies of the data are stored across multiple availability zones,
    - This means that no single computer or network failure renders Amazon SNS inaccessible.

SNS Security

- Securing messages to topics:
  - All API calls made to Amazon SNS are validated for the user's AWS ID and the signature.
  - AWS recommends that users secure their data over the wire by connecting to the secure SSL end-points

- Authenticating API calls:
  - All API calls made to Amazon SNS will validate authenticity by requiring that:
    - Requests be signed with the secret key of the AWS ID account
    - And verifying the signature included in the requests.

- Amazon SNS requires publishers with AWS IDs to validate their messages by signing messages with their secret AWS key; the signature is then validated by Amazon SNS.

SNS Mobile Push Notifications

- SNS Mobile Push lets you use Simple Notification Service (SNS) to deliver push notifications to Apple, Google, Fire OS, and Windows devices
- With push notifications, an installed mobile application can notify its users immediately by popping a notification about an event, without opening the application.
- Push notifications can only be sent to devices that have your app installed, and whose users have opted in to receive them.
- SNS Mobile Push does not require explicit opt-in for sending push notifications, but iOS, Android and Kindle Fire operating systems do require it.
- In order to send push notifications with SNS, you must also register your app and each installed device with SNS.

AWS SNS Billing

- Amazon SNS currently allows a maximum limit of 256 KB for published messages.

- Each 64KB chunk of published data is billed as 1 request.
  - For example, a single API call with a 256KB payload will be billed as four requests.

AWS SNS & AWS CloudTrail

- You can get the history for SNS API calls made to your account by enabling Cloudtrail
  - Cloudtrail will delivery log files for your SNS API Calls

- Cloudtrail logs will provide:
  - SNS API Caller
  - Source IP address
  - Time of the API call
  - Request parameters
  - Response elements returned by SNS

- This would be handy for security analysis, auditing , and operational/troubleshooting purposes

- Cloutrail logs for SNS are available for authenticated API calls only

AMAZON SIMPLE QUEUE SERVICE (SQS)

Amazon SQS

Introduction

Review Topic : Amazon SQS

AWS SQS

- SQS is a fast, reliable, durable, secure, and fully managed hosted message queue service
- Is a web service that gives you access to message queues that store messages waiting to be processed
- It offers a reliable, highly scalable, hosted queue for storing messages between distributed software systems and components.
- It allows the decoupling of application components such that a failure in one components does not cause a bigger problem to application functionality (like in coupled applications)
- Using SQS, you no longer need a highly available message cluster or the burden of building/running it
- SQS is a pay as you go service
- You can delete all the messages in an SQS queue without deleting the SQS queue itself
- You can use applications on EC2 instances to read and process the SQS queue messages
- You can use Auto Scaling to scale the EC2 fleet processing the SQS messages, as the queue size increases
- These applications on EC2 instances can process the SQS messages/jobs then post the results to other SQS queues or other AWS services
- SQS is a polling-based service (while SNS service is a push based service)

source: aws.amazon.co

Review Topic : Amazon SQS

AWS SQS - Reliability

- Amazon SQS stores all message queues and messages within a single, highly-available AWS region with multiple redundant Availability Zones (AZs),
  - No single computer, network, or AZ failure can make messages inaccessible.

Review Topic : Amazon SQS

AWS SQS – Queue Types

Standard Queue

- High (unlimited) throughput
- At least once delivery
- Duplicates are possible (can't guarantee no duplication)
- Best effort ordering

FIFO Queue (Not available in all regions)

- Limited throughput – 300 Transactions/sec/API (without batching) 3000 TPS/Sec/API with batching
- Exactly-once processing
- No duplicates guarantee
- Strict ordering - First-in-First-out

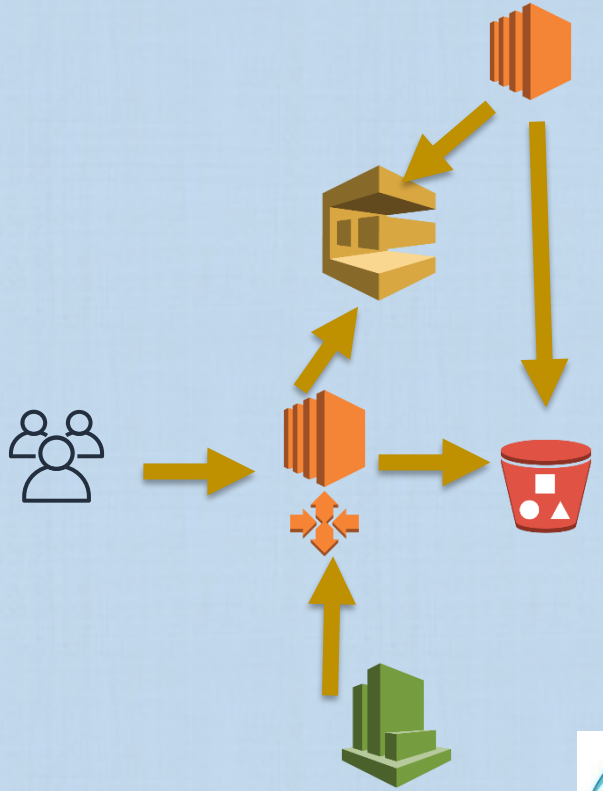
source: aws.amazon.co

Review Topic : Amazon SQS

AWS SQS – Use case

A video transcoding website uses Amazon EC2, Amazon SQS, Amazon S3, and Amazon DynamoDB together:

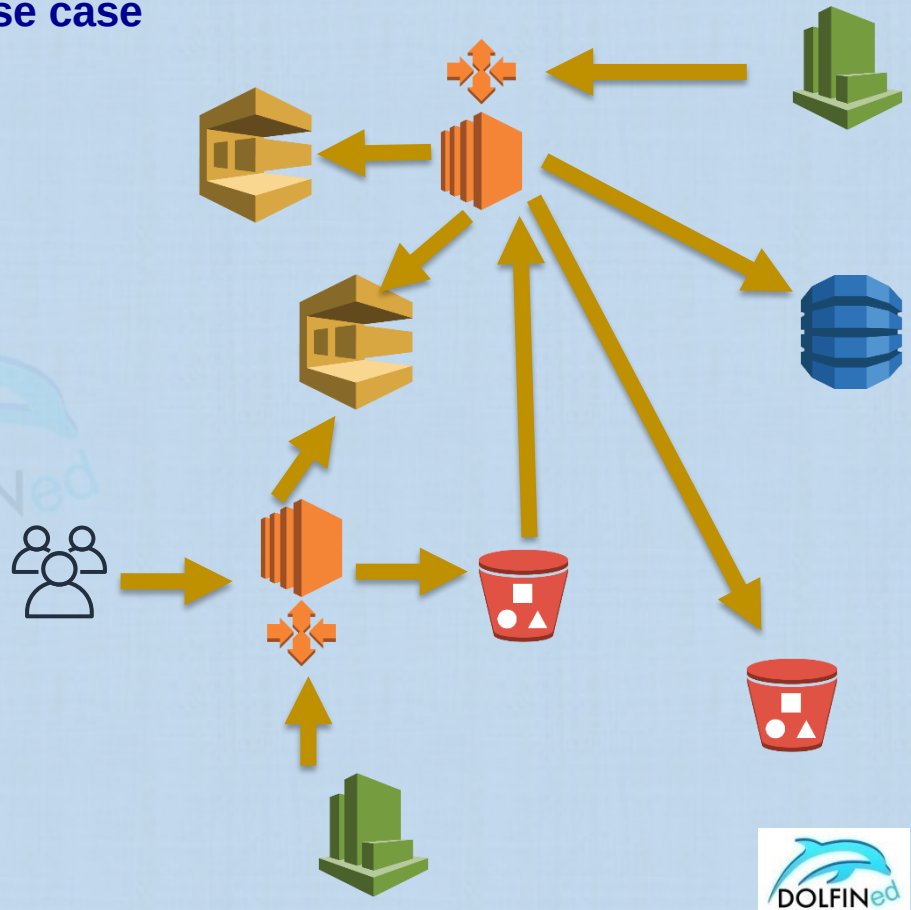
- End users submit videos to be transcoded to the website.
- The videos are stored in Amazon S3, and a request message is placed in an incoming Amazon SQS queue with a pointer to the video and to the target video format within the message.
- The transcoding engine that runs on a set of Amazon EC2 instances reads the request message from the incoming queue, retrieves the video from Amazon S3 using the pointer, and transcodes the video into the target format.



Review Topic : Amazon SQS

SQS – Use case

- The converted video is uploaded into another Amazon S3 bucket.
- Another response message is placed in another outgoing Amazon SQS queue with a pointer to the converted video.
- At the same time, metadata about the video (format, date created, length, and so on) is indexed into Amazon DynamoDB for querying.
- During this workflow, a dedicated Auto Scaling instance can constantly monitor the incoming queue. Based on the number of messages in the incoming queue, the Auto Scaling instance dynamically adjusts the number of transcoding Amazon EC2 instances to meet the response time requirements of the website's customers.



Amazon SQS

SQS Polling types and Timers

Review Topic : Amazon SQS

AWS SQS – Short/Regular Polling

- A request is returned immediately even if the queue is empty
  - It does not wait for messages to appear in the queue
  - It queries only a subset of the available servers for messages (based on weighted random distribution)
  - Default of SQS
  - ReceiveMessageWaitTime is set to 0

- More requests are used, which implies higher cost and empty reads

source: aws.amazon.co

Review Topic : Amazon SQS

AWS SQS – Long Polling

- Is preferred to regular/short polling, it uses fewer requests and reduces cost by:
  - Eliminating false empty responses by querying all the servers
  - Reduces the number of empty responses, by allowing Amazon SQS to wait until a message is available in the queue before sending a response. Unless the connection times out, the response to the ReceiveMessage
- request contains at least one of the available messages, up to the maximum number of messages specified in the ReceiveMessage action.
- Do not use if your application expects an immediate response to receive message calls
- ReceiveMessageWaitTime is set to a non zero value (up to 20 seconds)
- Same charge per million requests as the regular/short polling

source: aws.amazon.co

Review Topic : Amazon SQS

SQS – Retention Period

- SQS messages can remain in the queue for up to 14 days (SQS retention period)
  - Range is 1 min to 14 days (default is 4 days)
  - Once the maximum retention period of a message is reached, it will be deleted automatically from the queue
- Messages can be sent to the queue and read from the queue simultaneously
- SQS can be used with :
  - Redshift, DynamoDB, EC2, ECS, RDS, S3, Lambda to make distributed/decoupled applications
- You can have multiple SQS queues with different priorities in case you want one SQS queue messages to be handled with higher priority over other SQS queue messages
- You can scale up the send/receive messages by creating more queues for different processes/actions

Review Topic : Amazon SQS

SQS Visibility Timeout

- Is the duration of time (length) a message is locked for read by other consumers (after it has been read by a consumer to process it) so they can not be read again (by another consumer)
  - Max is 12 hours, default is 30 seconds , range 0 to 12 hours
  - Consumer is an application processing the SQS queue messages

- A consumer that reads a message to process it, can change the message visibility timeout if it needs more time to process the message

- After a message is read, there are the following possibilities:
  - An ACK is received from the consumer that a message is processed, so it must be deleted from the queue to avoid duplicates
  - If a FAIL is received or the visibility timeout expires, the message will then be unlocked for read , such that it can be read and processed by another consumer

source: aws.amazon.com

Review Topic : Amazon SQS

AWS SQS Delay Queue – Message Delay

- Delay queues allows for delaying the delivery of new messages to a queue for a number of seconds.
- In a delay queue, any messages that sent to the queue remain invisible to consumers for the duration of the delay period.
- Default delay is 0 seconds
- If you want to space the messages in the queue in time,
  - You can configure individual message delay of up to 15 minutes
    - This helps when need to schedule jobs with a delay

- For standard queues, the per-queue delay setting is not retroactive (does not affect messages already in the queue)
- For FIFO queues, the per-queue delay setting is retroactive

Review Topic : Amazon SQS

AWS SQS - Security

- You can use IAM policies to control who can read/write messages from/to an SQS queue
- Authentication mechanisms ensure that messages stored in Amazon SQS message queues are secured against unauthorized access.
  - You can control who can send messages to a message queue and who can receive messages from a message queue.
  - For additional security, you can build your application to encrypt messages before they are placed in a message queue.
- SQS supports HTTPS & supports TLS versions, 1.0, 1.1, 1.2 in all regions
- SQS is PCI DSS (Payment Card Industry Data Security Standard) level 1 compliant
- SQS is HIPAA (US Health Insurance Portability and Accountability Act) Eligible

source: aws.amazon.co

Review Topic : Amazon SQS

AWS SQS – SSE Encryption

- Server-side encryption (SSE) lets you transmit sensitive data in encrypted queues.
  - SSE protects the contents of messages in Amazon SQS queues using KMS managed keys
  - SSE encrypts messages as soon as Amazon SQS receives them.
  - The messages are stored in encrypted form and Amazon SQS decrypts messages only when they are sent to an authorized consumer
  - It uses AES-256 bits encryption

- AWS KMS combines secure, highly available hardware and software to provide a key management system scaled for the cloud.

- Both standard and FIFO queues support SSE.

- SSE for SQS may not be available in all regions, check the intended region for availability

Review Topic : Amazon SQS

AWS SQS – SSE Encryption

- SSE encrypts the body of a message in an Amazon SQS queue, SSE doesn't encrypt the following components:
  - Queue metadata (queue name and attributes)
  - Message metadata (message ID, timestamp, and attributes)
  - Per-queue metrics

- Encrypting a message makes its contents unavailable to unauthorized or anonymous users.
  - Encrypting messages doesn't affect the normal functioning of Amazon SQS:

- A message is encrypted only if it is sent after the encryption of a queue is enabled. Amazon SQS doesn't encrypt backlogged messages.

- Any encrypted message remains encrypted even if the encryption of its queue is disabled.

source: aws.amazon.co

Amazon SQS

SQS Service quotas & Monitoring

Review Topic : Amazon SQS

AWS SQS – Limits

- Unlimited number of messages can be received by an SQS Queue
- Maximum message size is 256 KB
- To send messages larger than 256KB, you can use Amazon SQS Extended Client Library for Java where a reference to the message is sent to the queue, while the message payload is stored in S3
  - Max message payload is 2GB

source: aws.amazon.co

Review Topic : Amazon SQS

AWS SQS – In-Flight messages

- Are the messages that are received from the queue by a consumer application but not deleted from the queue yet

- Standard Queues : An SQS limit of 120, 000 in-flight messages per queue
  - If you reach this limit, Amazon SQS returns the OverLimit error message.
  - You can also increase the number of queues you use to process your messages.

- FIFO Queue: An SQS limit of 20,000 in-flight messages per queue
  - If you reach this limit, Amazon SQS returns no error messages.

source: aws.amazon.co

Review Topic : Amazon SQS

AWS SQS - Monitoring

- AWS SQS and CloudWatch are integrated and you can view and monitor SQS queues' metrics using CloudWatch
  - This is supported for both Standard and FIFO based SQS Queues

- Amazon SQS is integrated with CloudTrail, a service that captures API calls made by or on behalf of Amazon SQS in your AWS account and delivers the log files to the specified Amazon S3 bucket.

source: aws.amazon.co

AMAZON MECHANICAL TURK (MTURK)

Amazon Mechanical Turk (MTurk)

- Human beings can still do some tasks much more effectively than computers, such as moderating content, performing data deduplication, or research.
- Amazon Turk, is a crowd sourcing marketplace that makes it easier for individuals and businesses to outsource their processes and jobs to a distributed virtual workforce who can perform these tasks.
- Amazon Mechanical Turk is a web service that provides an on-demand, scalable, human workforce to complete tasks.
- The tasks can include anything from conducting simple data validation and research to more subjective tasks like survey participation, content moderation, and more.

Amazon Mechanical Turk (MTurk)

The service enables companies to:

- Save cost by using virtual workforce, on-demand, rather than hiring.
- Harness the collective intelligence, skills, and insights from a global workforce to streamline business processes, augment data collection and analysis, and accelerate machine learning development.

- Crowdsourcing is a good way to break down a manual, time-consuming project into smaller, more manageable tasks to be completed by distributed workers over the Internet (also known as 'microtasks').

- Good workers and clear instructions are the key to obtaining successful results for any kind of project.

AWS SIMPLE WORKFLOW SERVICE (SWF)

Amazon SWF – Benefits

- Amazon SWF gives full control over implementing tasks and coordinating them without worrying about underlying complexities such as tracking their progress and maintaining their state.

- The Amazon SWF makes it easier to develop asynchronous and distributed applications by providing a programming model and infrastructure for coordinating distributed components and maintaining their execution state in a reliable way.

- Amazon SWF offers capabilities to support a variety of application requirements.
  - It is suitable for a range of use cases that require coordination of tasks, including:
    - Media processing,
    - Web application back-ends,
    - Business process workflows, and
    - Analytics pipelines.

- Amazon SWF access control uses AWS IAM

Amazon SWF – How to implement

- You have a number of options for implementing your workflow solutions with the Amazon Simple Workflow Service.
- AWS SDKs
- AWS Flow Framework
  - The AWS Flow Framework is an enhanced SDK for writing distributed, asynchronous programs that can run as workflows on Amazon SWF.
- HTTP Service API
- Development Environments
  - You will need to set up a development environment appropriate to the programming language that you will use.

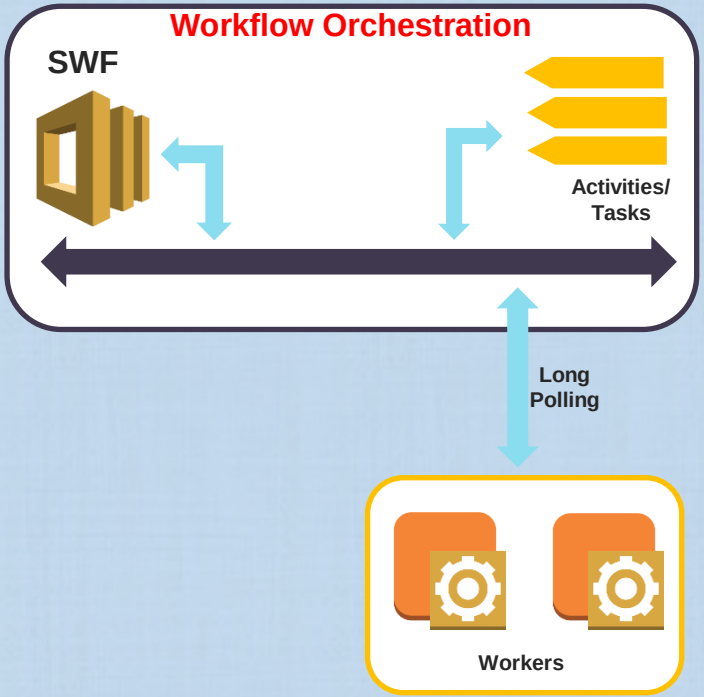
Amazon SWF Concepts

- The fundamental concept in Amazon SWF is the workflow.
  - A workflow is a set of activities that carry out some objective, together with logic that coordinates the activities.
  - Each workflow runs in an AWS resource called a domain, which controls the workflow's scope.
  - An AWS account can have multiple domains, each of which can contain multiple workflows, but workflows in different domains can't interact.

- When designing an Amazon SWF workflow,
  - Precisely define each of the required activities.
  - Then register each activity with Amazon SWF as an activity type.
    - Provide information such as a name and version, and some timeout values based on how long you expect the activity to take.
  - The process of carrying out the workflow, some activities may need to be performed more than once, perhaps with varying inputs.

Amazon SWF Concepts (cont.)

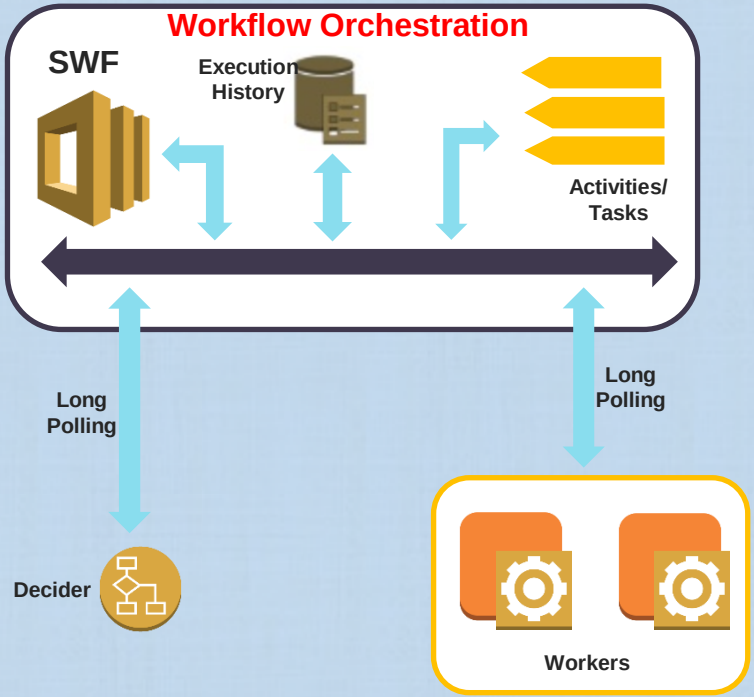
- **Worker(s)**
  - is a program that receives activity tasks, performs them, and provides results back.
  - Note that the task itself might actually be performed by a person, in which case the person would use the activity worker software for the receipt and disposition of the task.
    - An example might be a statistical analyst, who receives sets of data, analyzes them, and then sends back the analysis.
- When using Amazon SWF, workers are implemented to perform tasks.
  - These workers (workers are code specific to carry out a function or more) can run either on cloud infrastructure, such as Amazon EC2 instances, Lambda functions, or Mobile or tablet devices
  - Or they can be also **on premise** behind corporate firewalls
- **Activity tasks**
  - Are performed by workers and can run synchronously or asynchronously.
  - They can be distributed across multiple computers, **potentially in different geographic regions**, or they can all run on the same computer.



Amazon SWF Concepts (cont.)

- **Decider:**
  - Is the implementation of the coordination logic in a workflow and is contained in a software program.
    - Deciders control the flow of activity tasks in a workflow execution.
  - The decider schedules activity tasks, provides input data to the activity workers, processes events that arrive while the workflow is in progress, and ultimately ends (or closes) the workflow when the objective has been completed.
  - The decider communicates these steps back to Amazon SWF using decisions.
- The mechanism by which both the activity workers and the decider receive their tasks (activity tasks and decision tasks respectively) is by polling the Amazon SWF service.

- **Execution History**
  - Amazon SWF also maintains the state of each workflow execution,
    - This saves the burden of storing the state by the application itself



Amazon SWF – More on Tasks

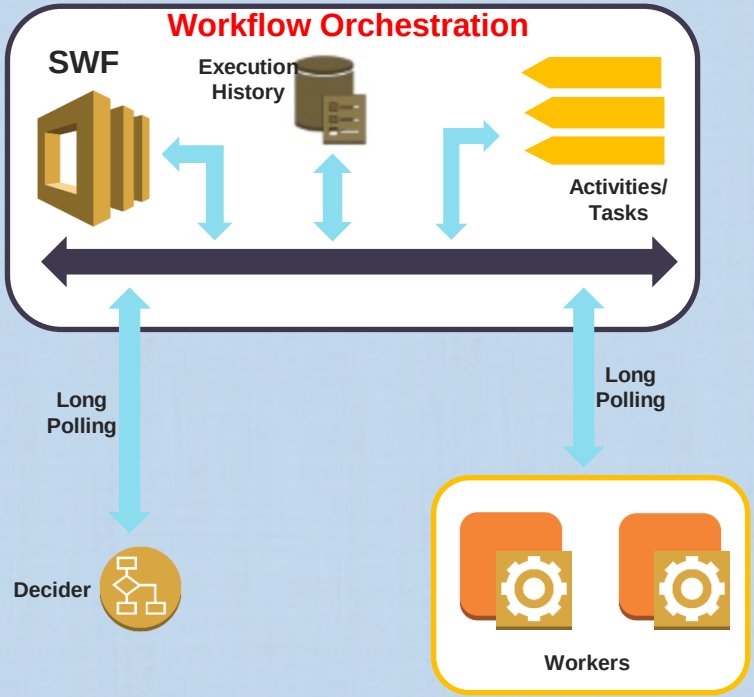
- Tasks can be created to be long-running, or that may fail, time out, or require restarts
- Amazon SWF stores tasks and assigns them to workers when they are ready, tracks their progress, and maintains their state, including details on their completion.
- To coordinate tasks, **programs needs to coded that gets the latest state o**f each task from Amazon SWF and uses it to initiate subsequent tasks.
- Amazon SWF maintains an application's execution state durably so that the application is resilient to failures in individual components.
  - With Amazon SWF, you can implement, deploy, scale, and modify these application components independently.
- **Task Types:**
- Activity task
  - An Activity task tells an activity worker to perform its function, such as to check inventory or charge a credit card.
  - The activity task contains all the information that the activity worker needs to perform its function.
- Lambda task
  - A Lambda task is similar to an Activity task, but executes a Lambda function instead of a traditional Amazon SWF activity.
- Decision task
  - A Decision task tells a decider that the state of the workflow execution has changed so that the decider can determine the next activity that needs to be performed.
  - The decision task contains the current workflow history.

Amazon SWF Concepts (cont.)

- The role of the Amazon SWF service is to function as a reliable central hub through which data is exchanged between the decider, the activity workers, and other relevant entities such as the person administering the workflow.

- The decider directs the workflow by receiving decision tasks from Amazon SWF and responding back to Amazon SWF with decisions.
  - A decision represents an action or set of actions which are the next steps in the workflow.
  - A typical decision would be to schedule an activity task.
  - Decisions can also be used to set timers to delay the execution of an activity task, to request cancellation of activity tasks already in progress, and to complete or close the workflow.

- Amazon SWF informs the decider of the state of the workflow by including, with each decision task, a copy of the current workflow execution history.



Amazon SWF -

Endpoints:

- Each workflow runs in an AWS resource called a domain, which controls the workflow's scope.
- To reduce latency and to store data in a location that meets your requirements, Amazon SWF provides endpoints in different regions.
- Each endpoint in Amazon SWF is completely independent; any domains, workflows and activities registered in one region don't share any data or attributes with those in another.
- When Amazon SWF domain workflow or activity is registered, it exists only within the region you registered it in.

Polling for Tasks in Amazon SWF

- Deciders and activity workers communicate with Amazon SWF using long polling. The decider or activity worker periodically initiates communication with Amazon SWF, notifying Amazon SWF of its availability to accept a task, and then specifies a task list to get tasks from.
- If a task is available on the specified task list, Amazon SWF returns it immediately in the response. If no task is available, Amazon SWF holds the TCP connection open for up to 60 seconds so that, if a task becomes available during that time, it can be returned in the same connection. If no task becomes available within 60 seconds, it returns an empty response and closes the connection.

AWS SWF

- Use cases
- SWF vs SQS

Amazon SWF – Use Case

- Amazon SWF has been applied to use cases in media processing, business process automation, data analytics, migration to the cloud, and batch processing.

- Video encoding using Amazon S3 and Amazon EC2.
  - Large videos are uploaded to Amazon S3 in chunks.
  - The upload of chunks has to be monitored.
  - After a chunk is uploaded, it is encoded by downloading it to an Amazon EC2 instance.
  - The encoded chunk is stored to another Amazon S3 location.
  - After all of the chunks have been encoded in this manner, they are combined into a complete encoded file which is stored back in its entirety to Amazon S3.
  - Failures could occur during this process due to one or more chunks encountering encoding errors.
  - Such failures need to be detected and handled.

Amazon SWF – Use Case

- Processing large product catalogs using **Amazon Mechanical Turk.**
  - While validating data in large catalogs, the products in the catalog are processed in batches.
    - Different batches can be processed concurrently.
    - For each batch, Amazon MTurk Requester User Interface (RUI) requires the product data is extracted from servers in the datacenter and transformed into CSV (Comma Separated Values) files.
    - The CSV is uploaded to populate and run the HITs (Human Intelligence Tasks).
    - When HITs complete, the resulting CSV file is reverse transformed to get the data back into the original format.
  - The results are then assessed and Amazon MTurk workers are paid for acceptable results.
    - Failures are weeded out and reprocessed, while the acceptable HIT results are used to update the catalog.
  - As batches are processed, the system needs to track the quality of the Amazon MTurk workers and adjust the payments accordingly.
  - Failed HITs are re-batched and sent through the pipeline again.

Amazon SWF – Use Case (cont.)

Using SWF:

- This is implemented as a set of workflows.
  - **A BatchProcess workflow** handles the processing for a single batch.
    - It has workers that extract the data, transform it and send it through Amazon Mechanical Turk.
    - The BatchProcess workflow outputs the acceptable HITs and the failed ones.
    - This is used as the input for three other workflows:
      - MTurkManager,
      - UpdateCatalogWorkflow, and
      - RerunProducts.
  - **The MTurkManager workflow** makes payments for acceptable HITs,
    - It responds to the human workers who produced failed HITs, and
    - Updates its own database for tracking results quality.
  - **The UpdateCatalogWorkflow** updates the master catalog based on acceptable HITs.
  - **The RerunProducts workflow** waits until there is a large enough batch of products with failed HITs.
    - It then creates a batch and sends it back to the BatchProcess workflow.
- The entire end-to-end catalog processing is performed by a CleanupCatalog workflow that initiates child executions of the above workflows.

Amazon SWF – Use Case

- Migrating components from the datacenter to the cloud.
  - Business critical operations are hosted in a private datacenter but need to be moved entirely to the cloud without causing disruptions.

- **Using SWF:**
  - Amazon SWF-based applications can combine workers that wrap components running in the datacenter with workers that run in the cloud.
  - To transition a datacenter worker seamlessly, new workers of the same type are first deployed in the cloud.
    - The workers in the datacenter continue to run as usual, along with the new cloud-based workers.
    - The cloud-based workers are tested and validated by routing a portion of the load through them.
      - During this testing, the application is not disrupted because the workers in the datacenter continue to run.
    - After successful testing, the workers in the datacenter are gradually stopped and those in the cloud are scaled up, so that the workers are eventually run entirely in the cloud.
    - This process can be repeated for all other workers in the datacenter so that the application moves entirely to the cloud.
    - If for some business reason, certain processing steps must continue to be performed in the private data center, those workers can continue to run in the private data center and still participate in the application.

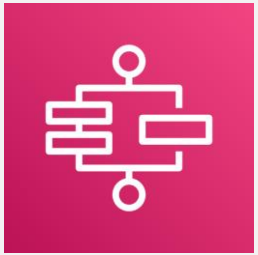
Amazon SWF vs. SQS

- Both Amazon SQS and Amazon SWF are services that facilitate the integration of applications or microservices:

- The following are the main differences between Amazon SQS and Amazon SWF:
  - Amazon SWF API actions are task-oriented.
    - **Amazon SQS API actions are message-oriented.**
  - Amazon SWF keeps track of all tasks and events in an application.
    - **Amazon SQS requires customers to implement their own application-level tracking, especially if the application uses multiple queues.**
  - The Amazon SWF Console and visibility APIs provide an application-centric view that allows to search for executions, drill down into an execution's details, and administer executions.
    - **Amazon SQS requires implementing such additional functionality.**
  - Amazon SWF offers several features that facilitate application development, such as passing data between tasks, signaling, and flexibility in distributing tasks.
    - **Amazon SQS requires customers to implement some application-level functionality.**

Amazon SWF – Execution time

- Each workflow execution can run for a maximum of 1 year.
- Each workflow execution history can grow up to 25,000 events.
- Amazon SWF does not take any special action if a workflow execution is idle for an extended period of time.
  - Idle executions are subject to the timeouts that are configured.



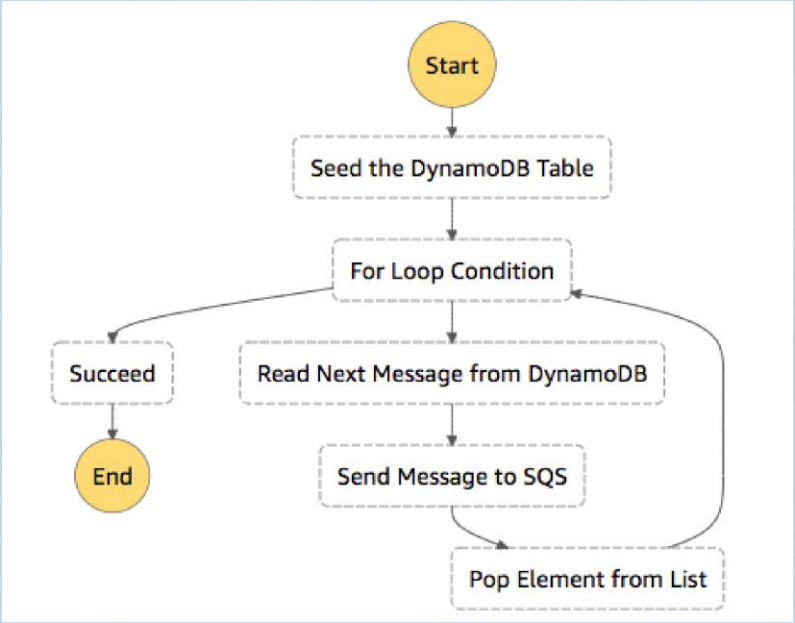
AWS STEP FUNCTIONS

AWS Step Functions

- Introduction
- How it works
- Activity Tasks and Service Tasks
- How a state machine connects to other AWS Services

AWS Step Functions

- AWS Step Functions is a fully managed service that makes it easy to coordinate the components of distributed applications and microservices using **visual workflows**.
  - Step Functions is a reliable way to coordinate components and step through the functions of the application.
- Step Functions provides a graphical console to arrange and visualize the components of the application as a series of steps.
  - Makes it simple to build and run multi-step applications.
- Step Functions service automatically triggers and tracks each step, and retries when there are errors, so the application executes in order and as expected.
  - Step Functions logs the state of each step for easier diagnosis and debugging
- It is possible to change and add steps without even writing code, so application can easily evolve and innovate faster.



AWS Step Functions – How it works

- AWS Step Functions allows to coordinate individual tasks by expressing the workflow as a finite state machine, written in the Amazon States Language.
  - Using AWS Step Functions, define state machines that describe the workflow as a series of steps, their relationships, and their inputs and outputs.
  - A finite state machine can express an algorithm as a number of states, their relationships, and their input and output.
- State machines contain a number **of states**,
  - Each state represents an individual step in a workflow diagram.
- States can perform work, make choices, pass parameters, initiate parallel execution, manage timeouts, or terminate your workflow with a success or failure.
- The visual console automatically graphs each state in the order of execution, making it easy to design multi-step applications.
- The console highlights the real-time status of each step and provides a detailed history of every execution.

AWS Step Functions – Activity Tasks and Service Tasks

- State machines work by creating/using activity tasks and service tasks.
  - A task performs work by using an activity or an AWS Lambda function, or by passing parameters to the API actions of other services.
- Activity Tasks
  - Is how a specific step (task) in the workflow (state machine) can be assigned to an activity worker (code running some where else)
    - Activity workers can be any application, anywhere, that can communicate via HTTP
    - Activity workers can be run on AWS EC2 instances, a mobile device, in an on-premise DC
    - Activity workers poll work from and takes input from Step Functions, does the work, and returns results to Step functions.
- Service Tasks
  - Is how a step in the state machine (workflow) can connect to a supported AWS service.
  - Step Functions pushes requests to other services so they can perform actions for the workflow, waits for the service task to complete, and then continues to the next step.
- An AWS Step Functions state machine can contain combinations of activity tasks and service tasks.
- AWS Step Functions applications can also combine activity workers running in a data center with service tasks that run in the cloud.
- The workers in the data center continue to run as usual, along with any cloud-based service tasks.

AWS Step Functions – How a State Machine connects to other AWS services

- Workflows that are created with AWS Step Functions can connect and coordinate other supported AWS services using service tasks, this includes:
  - Invoke an AWS Lambda function
  - Run an Amazon ECS or AWS Fargate task
  - Get or Put an item from an Amazon DynamoDB table
  - Submit an AWS Batch job and wait for it to complete
  - Publish a message to an Amazon SNS topic
  - Send a message to an Amazon SQS queue
  - Start an AWS Glue job run
  - Create an Amazon SageMaker job to train a machine learning model or batch transform a data set
  - Have Step Functions wait for a task to return a specific task token.

AWS Step Functions

- Use Cases
- Step Functions and API Gateway
- SWF vs Step Functions
- Monitoring and Logging

AWS Step Functions – Use Cases

- AWS Step Functions are suitable for any computational problem or business process that can be subdivided into a series of steps (or a State Machine as the workflow is called under Step Functions).
  - It's also useful for creating end-to-end workflows to manage jobs with interdependencies.

- Here are some common use cases:
  - **Data processing:**
    - Consolidate data from multiple databases into unified reports, refine and reduce large data sets into useful formats, or coordinate multi-step analytics and machine learning workflows
  - **DevOps and IT automation:**
    - Build tools for CI/CD, or create event-driven applications that automatically respond to changes in infrastructure
  - **E-commerce:**
    - Automate mission-critical business processes, such as order fulfillment and inventory tracking
  - **Web applications:**
    - Implement robust user registration processes and sign-on authentication

AWS Step Functions and API Gateway

- The Step Functions APIs can be associated with Amazon API Gateway so that these APIs invoke the respective state machines when an HTTPS request is sent to a customers defined API method.

- An Amazon API Gateway API can be used to start Step Functions state machines that coordinate the components of a distributed backend application, and integrate human based activity tasks into the application steps such as,
  - Approval requests and responses.

- Serverless asynchronous calls can be made to the APIs of services that the application uses.

Amazon SWF vs. AWS Step Functions

- AWS Step Functions is a fully managed service that makes it easy to coordinate the components of distributed applications and microservices using visual workflows.
- Instead of writing a Decider program as in SWF, Customers define state machines in JSON.
- Consider using Step Functions for new applications. If Step Functions does not fit the needs, then consider Amazon SWF
- **Use AWS SWF if:**
  - An external signal in the workflow is required
  - Or, when it is required to launch child processes that return a result to a parent
- Amazon SWF provides complete control over the orchestration logic, but increases the complexity of developing applications.
  - AWS will continue to provide the Amazon SWF service, Flow framework, and support all Amazon SWF customers.

AWS Step Functions - Monitoring

- AWS Step Functions sends metrics to **Amazon CloudWatch** and **AWS CloudTrail** for application monitoring.
- Amazon CloudWatch collects and track metrics, sets alarms, and automatically reacts to changes in AWS Step Functions.
- Step Functions also supports Amazon CloudWatch Events
  - A State Machine can be the target of AWS CloudWatch Events
- **AWS CloudTrail** captures all API calls for Step Functions as events, including calls from the Step Functions console and from code calls to the Step Functions APIs.