# Serverless Overview

What’s serverless?

* Serverless is a new paradigm in which the developers don’t have to manage servers anymore…
* They just deploy code
* Initially deploy… functions !
* Initially... Serverless == FAAS (Function as a Service)
* Serverless was pioneered by AWS Lambda but now also includes anything that’s managed: “databases, messaging, storage, etc.”
* Serverless does not mean there are no servers… it means you just don’t manage / provision / see them

**Serverless in AWS**

|  |  |
| --- | --- |
| * AWS Lambda * DynamoDB * AWS Cognito * AWS API Gateway * Amazon S3 * AWS SNS & SQS * AWS Kinesis Data Firehose Aurora Serverless * Step Functions * Fargate |  |

## Lambda

|  |  |
| --- | --- |
| EC2 | Lambda |
| • Virtual Servers in the Cloud  • Limited by RAM and CPU we provision  • Continuously running even when not used  • Scaling means intervention to add / remove servers | • Virtual functions – no servers to manage!  • Limited by time - short executions up to 15 minutes  • Run on-demand when you don't use Lambda, your Lambda function is not running, and you only are going to be billed when your function is running  • Scaling is automated! |

**Benefits of AWS Lambda**

* Easy Pricing:
  + Pay per request and compute time
  + Free tier of 1,000,000 AWS Lambda requests and 400,000 GBs of compute time
* Integrated with the whole AWS suite of services
* Integrated with many programming languages
* Easy monitoring through AWS CloudWatch
* Easy to get more resources per functions (up to 10GB of RAM!)
* Increasing RAM will also improve CPU and network!

**AWS Lambda language support**

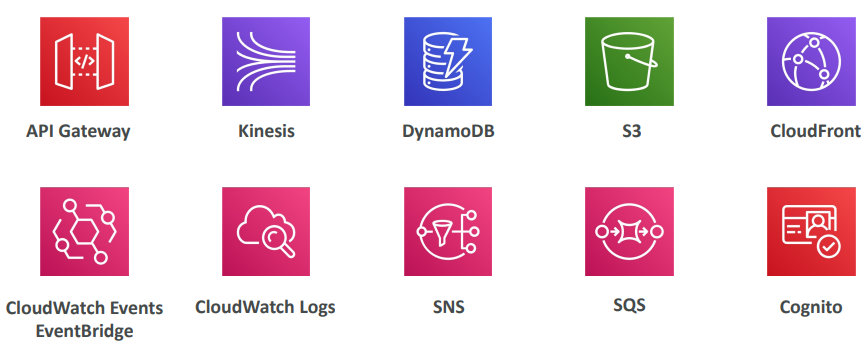
* Node.js (JavaScript)
* Python
* Java
* C# (.NET Core) / Powershell
* Ruby
* Custom Runtime API (community supported, example Rust or Golang)
* Lambda Container Image
  + The container image must implement the Lambda Runtime API
  + ECS / Fargate is preferred for running arbitrary Docker images

**AWS Lambda Pricing: example**

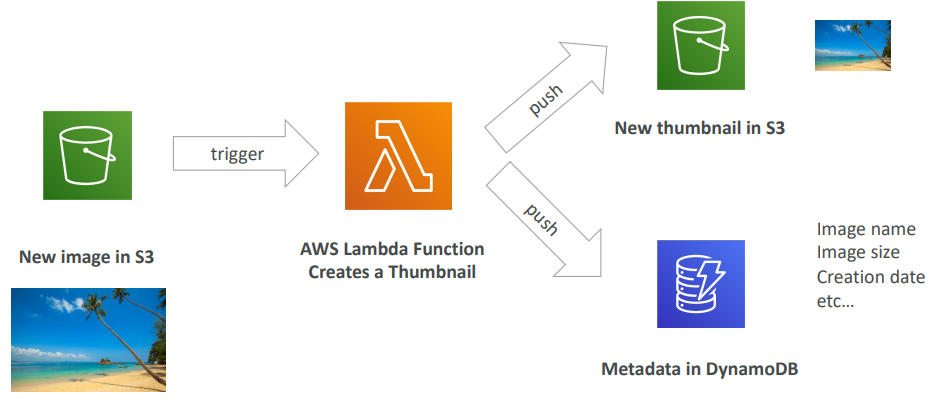
* You can find overall pricing information here:
* https://aws.amazon.com/lambda/pricing/
* Pay per calls:
* First 1,000,000 requests are free • $0.20 per 1 million requests thereafter ($0.0000002 per request)
* Pay per duration: (in increment of 1 ms)
  + 400,000 GB-seconds of compute time per month for FREE
  + == 400,000 seconds if function is 1GB RAM
  + == 3,200,000 seconds if function is 128 MB RAM
  + After that $1.00 for 600,000 GB-seconds
* It is usually very cheap to run AWS Lambda so it’s very popular

## Lambda Integrations examples

There are many more

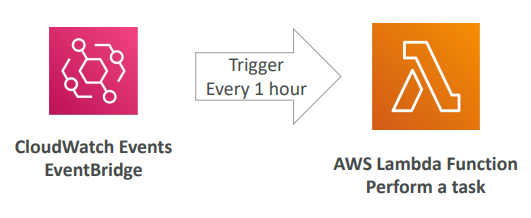


**Example: Serverless Thumbnail creation**



**Example: Serverless CRON Job**

You can create a CloudWatch Event rule or an EventBridge rule that will be triggered every one hour. And every one hour, it will be integrated with a Lambda function that will perform your task. With EC2 the server will continouly



## AWS Lambda Limits

* Execution:
  + Memory allocation: 128 MB – 10GB (1 MB increments)
  + Maximum execution time: 900 seconds (15 minutes)
  + Environment variables (4 KB)
  + Disk capacity in the “function container” (in /tmp): 512 MB to 10GB
  + Concurrency executions: 1000 (can be increased)
* Deployment:
  + Lambda function deployment size (compressed .zip): 50 MB
  + Size of uncompressed deployment (code + dependencies): 250 MB
  + Can use the /tmp directory to load other files at startup
  + Size of environment variables: 4 KB

## Lambda Concurrency

If we invoke our Lambda function at a low scale, we may have two concurrent executions of our Lambda functions. But if we have a very high scale of events happening, we may have up to 1000 concurrence of Lambda functions



Something can do though is to limit the number of concurrent executions a Lambda function can do, and that is recommended. For this, we can set what's called reserved concurrency and that is set at the function level. This is a limit, and we're saying, "Okay, this Lambda function can only have "up to 50 concurrent executions."

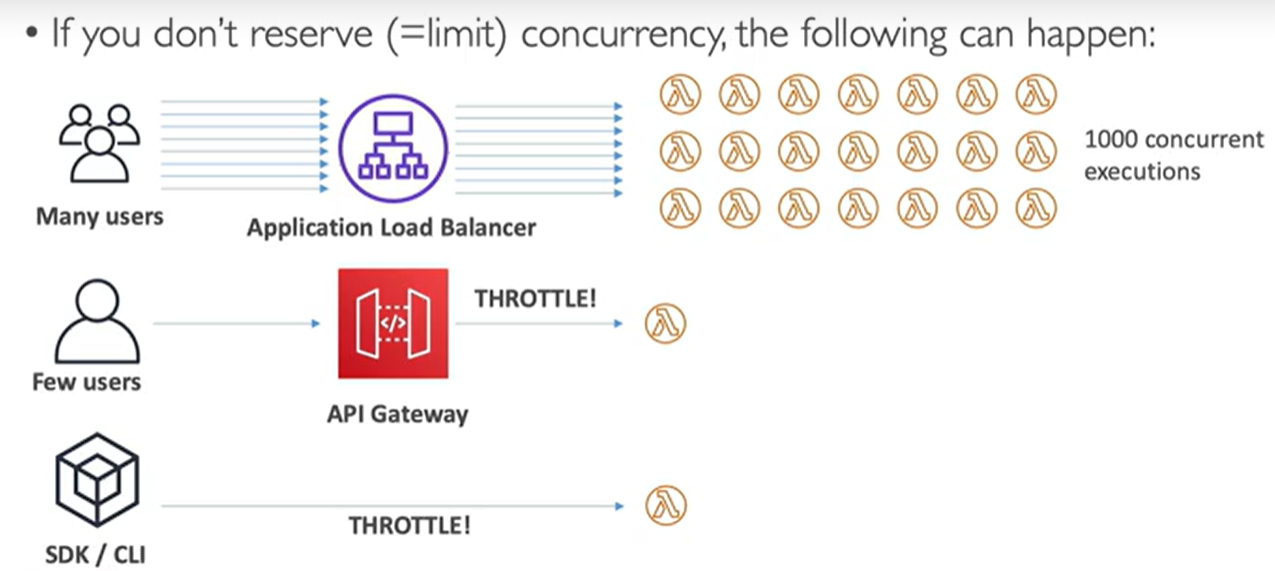
Each invocation over the concurrency limits will trigger what's called a throttle. And there are different behaviors with a throttle

* if it's a synchronous invocation it will return throttle error, 429.
* If it's an asynchronous invocation, it will retry automatically and then go to DLQ.

In case you need a higher than 1000 concurrent executions at a time, you can just open a support ticket to request a higher limit.

**Synchronous execution**

If you don't set any reserve concurrency, so any limit on your function concurrency, then this could happen. We have our application balancer, for example, connected to a Lambda function. We have another application where we have few users that connect to an API gateway, connected to another Lambda function, and one last application may be using the SDK and the CLI to invoke a Lambda function. When everything is low-level, like low throughput of invocation, everything is fine.

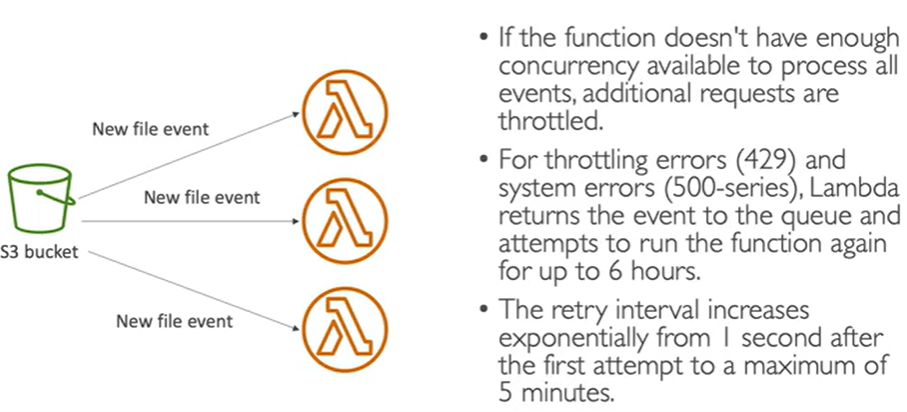


But let's say that we are running a huge promotion and somehow, we get many, many users hammering our application load balancers, we're very successful. What happens is that our load balancer will be invoking many, many Lambda functions and Lambda functions can scale automatically. We'll get up to 1000 concurrent executions and lambda has scaled. But here is the problem. All the concurrent executions went to the first application. So that means that the application users of our API gateway will be throttled. And that means that the CLI and SDK will also be throttled.

Concurrency limit applies to all the functions in your accounts, and so you have to be careful because if one function goes over the limit, it's possible that your other functions get throttled. So that's very, very important.

**Asynchronous execution**

Let's take the example of S3 event notifications. We are uploading files into our S3 buckets, and this creates a new file event that will invoke our Lambda functions, and say we are putting many, many files at the same time. So, we get many, many different Lambda concurrent executions happening. And if the function doesn't have enough concurrency available. So, if it cannot scale up because we have reached the limits, then the additional requests are throttled. But this is an asynchronous request. So, for any throttling errors and system error, so 429 and 500-series, Lambda will return the event to the event queue. So, remember in the asynchronous mode there is an internal event queue, and Lambda will attempt to run the function again for up to six hours. So, there's a lot of retries that happen due to the throttling and so on. Then this retries interval will increase in an exponential bucket fashion. So, from one second to our maximum of every five minutes. So this allows your Lambda functions to keep on retrying and hopefully one day find the concurrency and capacity available to run correctly.



**cold starts and provisioned concurrency**

Cold start, it means that when you create a new Lambda function instance, your code must be loaded and your code outside of the handler has to be run. This corresponds to all your initialization. And if your initialization is large, because you have a lot of code, a lot of dependencies, you're connecting to many databases and creating many SDK, this process can take a lot of time. So that means that the first request served by new instances has a higher latency than the rest and that may impact on your users. If your user is maybe waiting three seconds to get a request response, that may be very, very slow for them and they may experience a cold start and may be unhappy with your product.

Solution Well, you can use something called a provisioned concurrency. That means that you allocate concurrency before the function is even invoked. So, you allocate this concurrency in advance. This way, the cold never happens, and all the invocations will have a lower latency. And to manage this concurrency, you can... This provisioned concurrency, you can use Application Auto Scaling. For example, for a schedule or target position to make sure that you have enough reserved Lambda functions to be ready to be used and minimize this cold start problem. So please note that whenever before you used to launch a Lambda function in a VPC that used to take forever. So now there was a blog in October and November 2019 that has been released by AWS.

<https://aws.amazon.com/blogs/compute/announcing-improved-vpc-networking-for-aws-lambda-function>

And this blog shows the improvements they have done to dramatically reduce the cold starts in your VPC. So the good news is if you were using Lambda before the cold starts, really have a minimal impact on your VPC.

## Lambda Snap Start

|  |  |
| --- | --- |
| * Improves your Lambda functions performance up to 10x at no extra cost for Java 11 and above * When enabled, function is invoked from a preinitialized state (no function initialization from scratch) * When you publish a new version:   + Lambda initializes your function   + Takes a snapshot of memory and disk state of the initialized function   + Snapshot is cached for low-latency access |  |

## Customization at the edge

* Many modern applications execute some form of logic at the edge
* Edge Function:
  + A code that you write and attach to CloudFront distributions
  + Runs close to your users to minimize latency
* CloudFront provides two types: CloudFront Functions & Lambda@Edge
* You don’t have to manage any servers, deployed globally
* Use case: customize the CDN content
* Pay only for what you use
* Fully serverless

**CloudFront Functions & Lambda@Edge Use Cases**

* Website Security and Privacy
* Dynamic Web Application at the Edge
* Search Engine Optimization (SEO)
* Intelligently Route Across Origins and Data Centers
* Bot Mitigation at the Edge
* Real-time Image Transformation
* A/B Testing
* User Authentication and Authorization
* User Prioritization
* User Tracking and Analytics

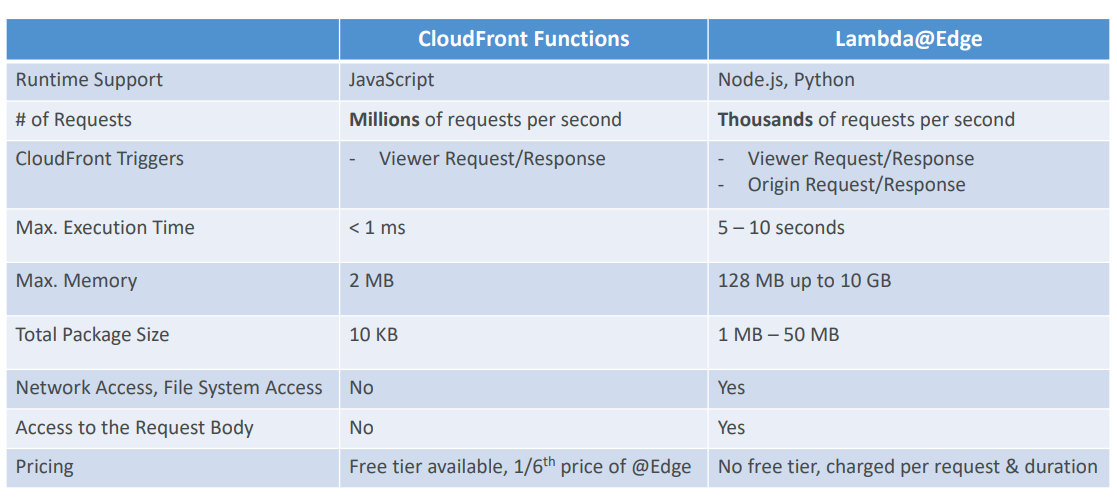
**CloudFront Functions**

|  |  |
| --- | --- |
| * Lightweight functions written in JavaScript * For high-scale, latency-sensitive CDN customizations * Sub-ms startup times, millions of requests/second * Used to change Viewer requests and responses:   + Viewer Request: after CloudFront receives a request from a viewer   + Viewer Response: before CloudFront forwards the response to the viewer * Native feature of CloudFront (manage code entirely within CloudFront) |  |

**Lambda@Edge**

|  |  |
| --- | --- |
| * Lambda functions written in NodeJS or Python * Scales to 1000s of requests/second * Used to change CloudFront requests and responses:   + Viewer Request – after CloudFront receives a request from a viewer   + Origin Request – before CloudFront forwards the request to the origin   + Origin Response – after CloudFront receives the response from the origin   + Viewer Response – before CloudFront forwards the response to the viewer * Author your functions in one AWS Region (us-east-1), then CloudFront replicates to its location |  |

**CloudFront Functions vs. Lambda@Edge**



**CoudFront Functions vs. Lambda@Edge - Use Cases**

|  |  |
| --- | --- |
| CloudFront Functions   * Cache key normalization   + Transform request attributes (headers, cookies, query strings, URL) to create an optimal Cache Key * Header manipulation   + Insert/modify/delete HTTP headers in the request or response * URL rewrites or redirects * Request authentication & authorization * Create and validate user-generated tokens (e.g., JWT) to allow/deny requests | Lambda@Edge   * Longer execution time (several ms) * Adjustable CPU or memory * Your code depends on a 3rd libraries (e.g., AWS SDK to access other AWS services) * Network access to use external services for processing * File system access or access to the body of HTTP requests |

## Lambda VPC

|  |  |
| --- | --- |
| * By default, your Lambda function is launched outside your own VPC (in an AWS -owned VPC) * • Therefore, it cannot access resources in your VPC (RDS, ElastiCache, internal ELB…) |  |

**Lambda in VPC**

|  |  |
| --- | --- |
| * You must define the VPC ID, the Subnets and the Security Groups * Lambda will create an ENI (Elastic Network Interface) in your subnets |  |

**Lambda with RDS Proxy**

|  |  |
| --- | --- |
| * If Lambda functions directly access your database, they may open too many connections under high load * RDS Proxy   + Improve scalability by pooling and sharing DB connections   + Improve availability by reducing by 66% the failover time and preserving connections   + Improve security by enforcing IAM authentication and storing credentials in Secrets Manager * The Lambda function must be deployed in your VPC, because RDS Proxy is never publicly accessible |  |

## Invoking Lambda from RDS & Aurora

We can invoke Lambda functions from within your database instance in some cases. And this would allow you to process data events happening from within your database. For example, a user is going to insert event data into your table, your registration table. And RDS will be set up to directly invoke your Lambda function. Your Lambda function may send a welcome email to your user.

This is something that you have to set up from within the database by connecting to it. This is not something that you set up from the AWS console.

|  |  |
| --- | --- |
| * Invoke Lambda functions from within your DB instance * Allows you to process data events from within a database * Support for RDS for PostgreSQL and Aurora MySQL * Must allow outbound traffic to your Lambda function from within your DB instance (Public, NAT GW, VPC Endpoints) * DB instance must have the required permissions to invoke the Lambda function (Lambda Resource-based Policy & IAM Policy) |  |

**RDS Event Notifications**

This is completely different from using the RDS event notifications. These notifications are happening within AWS and there are notifications that tell you information about the database instance itself. For example, when it was created, when it was started, and so on. And you don't have any information of the data happening within your database.

|  |  |
| --- | --- |
| * Notifications that tell information about the DB instance itself (created, stopped, start, …) * You don’t have any information about the data itself * Subscribe to the following event categories: DB instance, DB snapshot, DB Parameter Group, DB Security Group, RDS Proxy, Custom Engine Version * Near real-time events (up to 5 minutes) * Send notifications to SNS or subscribe to events using EventBridge |  |

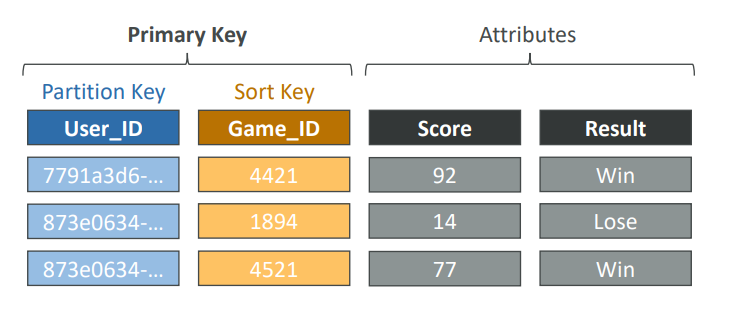
## Amazon DynamoDB

* Fully managed, highly available with replication across multiple AZs
* NoSQL database - not a relational database - with transaction support
* Scales to massive workloads, distributed database
* Millions of requests per seconds, trillions of row, 100s of TB of storage
* Fast and consistent in performance (single-digit millisecond)
* Integrated with IAM for security, authorization and administration
* Low cost and auto-scaling capabilities
* No maintenance or patching, always available
* Standard & Infrequent Access (IA) Table Class

**DynamoDB - Basics**

* DynamoDB is made of Tables
* Each table has a Primary Key (must be decided at creation time)
* Each table can have an infinite number of items (= rows)
* Each item has attributes (can be added over time – can be null) you can add the attributes over time and they can be null. So this is a big difference in an RDS or Aurora database
* Maximum size of an item is 400KB
* Data types supported are:
  + Scalar Types – String, Number, Binary, Boolean, Null
  + Document Types – List, Map
  + Set Types – String Set, Number Set, Binary Set
* Therefore, in DynamoDB you can rapidly evolve schemas

**DynamoDB –Table example**



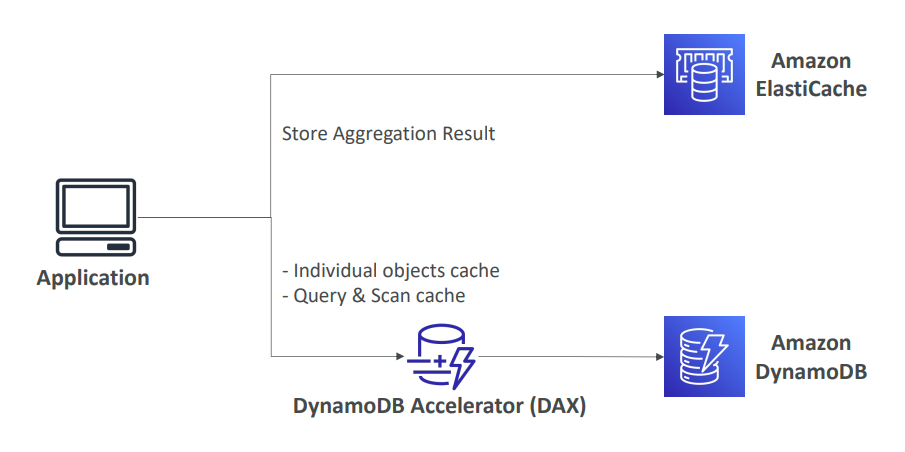
**DynamoDB – Read/Write Capacity Modes**

* Control how you manage your table’s capacity (read/write throughput)
* Provisioned Mode (default)
  + You specify the number of reads/writes per second
  + You need to plan capacity beforehand
  + Pay for provisioned Read Capacity Units (RCU) & Write Capacity Units (WCU)
  + Possibility to add auto-scaling mode for RCU & WCU
* On-Demand Mode
  + Read/writes automatically scale up/down with your workloads
  + No capacity planning needed
  + Pay for what you use, more expensive ($$$)
  + Great for unpredictable workloads, steep sudden spikes

**DynamoDB Accelerator (DAX)**

|  |  |
| --- | --- |
| * Fully managed, highly available, seamless in memory cache for DynamoDB * Help solve read congestion by caching * Microseconds latency for cached data * Doesn’t require application logic modification (compatible with existing DynamoDB APIs) * 5 minutes TTL for cache (default) |  |

**DynamoDB Accelerator (DAX) vs. ElastiCache**

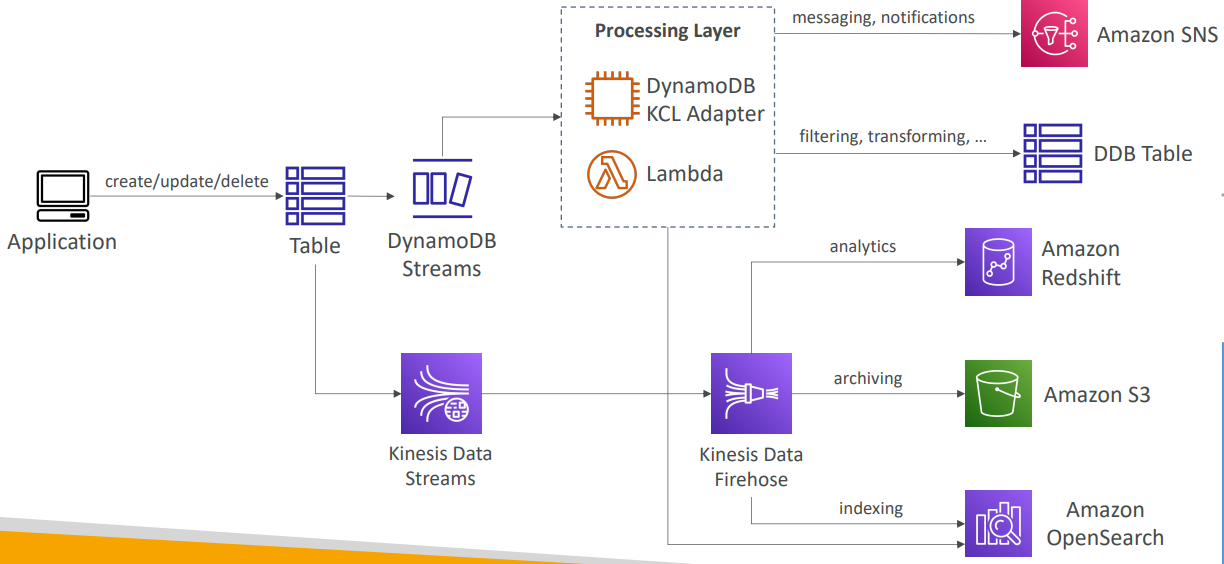


**DynamoDB – Stream Processing**

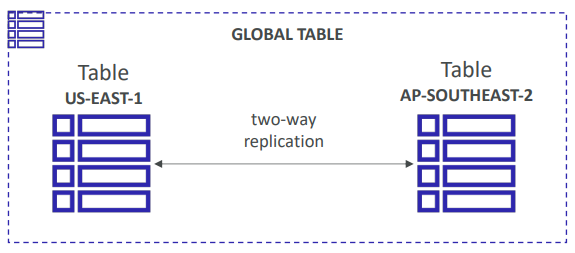
* Ordered stream of item-level modifications (create/update/delete) in a table
* Use cases:
* React to changes in real-time (welcome email to users)
* Real-time usage analytics
* Insert into derivative tables
* Implement cross-region replication
* Invoke AWS Lambda on changes to your DynamoDB table

|  |  |
| --- | --- |
| DynamoDB Streams   * 24-hour retention * Limited # of consumers * Process using AWS Lambda Triggers, or DynamoDB Stream Kinesis adapter | Kinesis Data Streams (newer)   * 1 year retention * High # of consumers * Process using AWS Lambda, Kinesis Data Analytics, Kineis Data Firehose, AWS Glue Streaming ETL… |

**DynamoDB Streams**



**DynamoDB Global Tables**



* Make a DynamoDB table accessible with low latency in multiple-regions
* Active-Active replication
* Applications can READ and WRITE to the table in any region
* Must enable DynamoDB Streams as a pre-requisite

**DynamoDB –Time To Live (TTL)**

|  |  |
| --- | --- |
| * Automatically delete items after an expiry timestamp * Use cases: reduce stored data by keeping only current items, adhere to regulatory obligations, web session handling… |  |

**DynamoDB – Backups for disaster recovery**

* Continuous backups using point-in-time recovery (PITR)
  + Optionally enabled for the last 35 days
  + Point-in-time recovery to any time within the backup window
  + The recovery process creates a new table
* On-demand backups
  + Full backups for long-term retention, until explicitly deleted
  + Doesn’t affect performance or latency
  + Can be configured and managed in AWS Backup (enables cross-region copy)
  + The recovery process creates a new table

**DynamoDB – Integration with Amazon S3**

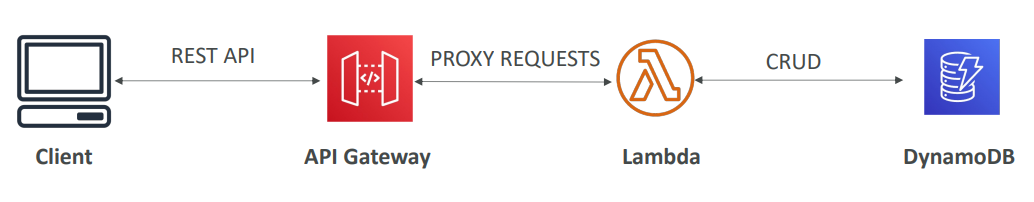
|  |  |
| --- | --- |
| * Export to S3 (must enable PITR) * Works at any point of time in the last 35 days * Doesn’t affect the read capacity of your table * Perform data analysis on top of DynamoDB * Retain snapshots for auditing * ETL on top of S3 data before importing back into * DynamoDB * Export in DynamoDB JSON or ION format * Import from S3 * Import CSV, DynamoDB JSON or ION format * Doesn’t consume any write capacity * Creates a new table * Import errors are logged in CloudWatch Logs |  |

## API gateway

**Example: Building a Serverless API**

We can have the client directly invoke the Lambda function, but that means that the client would need IAM permissions, or we've seen we can use an application load balancer to have it in between the client and the Lambda function and that would expose our Lambda function as an HTTP endpoint.

There's one more thing we can use. It's called the API Gateway. And this is a serverless offering from AWS which allows us to create REST APIs that are going to be public and accessible for our clients. So, the client will talk to the API Gateway and API Gateway will then proxy the request to our Lambda functions.



Note even tough Lambda function can have time out of 15 minutes. API gateway will have max timeout of 29 seconds.

Also when we configure Lambda functions in API gateway, it will automatically create IAM permissions to access Lambda function.

It will also show logs

**AWS API Gateway**

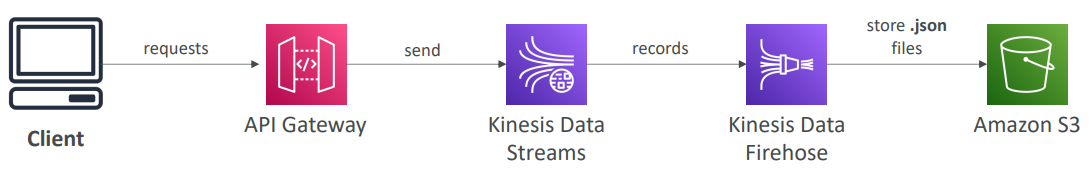
* AWS Lambda + API Gateway: No infrastructure to manage
* Support for the WebSocket Protocol
* Handle API versioning (v1, v2…)
* Handle different environments (dev, test, prod…)
* Handle security (Authentication and Authorization)
* Create API keys, handle request throttling
* Swagger / Open API import to quickly define APIs
* Transform and validate requests and responses
* Generate SDK and API specifications
* Cache API responses

**API Gateway – Integrations High Level**

* Lambda Function
  + Invoke Lambda function
  + Easy way to expose REST API backed by AWS Lambda
* HTTP
  + Expose HTTP endpoints in the backend
  + Example: internal HTTP API on premise, Application Load Balancer…
  + Why would we do this ? To add rate limiting, caching, user authentications, API keys, etc…
* AWS Service
  + Expose any AWS API through the API Gateway
  + Example: start an AWS Step Function workflow, post a message to SQS
  + Why would we do this ? To add authentication, deploy publicly, rate control…

**Example API Gateway integration with Kinesis Data Streams**

So, we want to have people send data into Kinesis Data Streams but in a secure way, without giving them access to AWS credentials. How to achieve this? In between our clients and our Kinesis Data Streams, we're going to have the API Gateway. And the clients will send HTTP requests into the API Gateway. And it's been configured to then send the messages into a Kinesis Data Streams from threse we can do many things like sending to S3



**API Gateway - Endpoint Types**

There are three ways to deploy your API Gateway.

* **Edge-Optimized (default)**: This is for your global clients. So that means that your API Gateway is going to be accessible from anywhere in the world. And to be efficient, the requests are going to be routed through all the CloudFront Edge locations, which will improve the latency. Your API Gateway is still only in one region where you created it, but it's accessible efficiently from every CloudFormation Edge location.
* **Regional:** This is when we don't want to use CloudFront Edge locations. When we expect all our users to be within the same region where we created our API Gateway. And if you wanted to, you could create your own platform distribution, and this will give you the same result as an Edge-Optimized distribution, but this time you have more control over location strategies and the platform settings themselves.
* **Private**: So, a private API Gateway can only be accessed from within your VPC. And it will use interface VPC endpoints for your ENIs. And to define access to your API Gateway, you can use a resource policy.

**API Gateway – Security**

* User Authentication can be done in 3 ways
  + IAM Roles (useful for internal applications)
  + Cognito (identity for external users – example mobile users)
  + Custom Authorizer (your own logic via lambda function)
* Also, you can have HTTPS security through your own custom domain name by integration with a service called the AWS Certificate Manager or ACM. And so that certificate, if you're using an Edge-Optimized endpoint must be in us-east-1 but if you're using a regional endpoint then that certificate may be in the same region as the API Gateway stage. Finally, you must set up, of course, a CNAME or an A-alias record in Route 53 to point to your domain and API Gateway
  + If using Edge-Optimized endpoint, then the certificate must be in us-east-1
  + If using regional endpoint, the certificate must be in the API Gateway region
  + Must set up CNAME or A-alias record in Route 53

## AWS Step Functions

|  |  |
| --- | --- |
| * Build serverless visual workflow to orchestrate your Lambda functions * Features: sequence, parallel, conditions, timeouts, error handling, … * Can integrate with EC2, ECS, On -premises servers, API Gateway, SQS queues, etc. … * Possibility of implementing human approval features * Use cases: order fulfillment, data processing, web applications, any workflow |  |

## Amazon Cognito

The purpose is to give users an identity to interact with our web or mobile application. These users usually sit out side of our AWS account hence the name. There are 2 kinds of subservices in Cognito.

1. Cognito User Pools which provides
   1. Sign in functionality for app users
   2. Integrate with API Gateway & Application Load Balancer

Basically if we have external apps and we want users to connect to AWS services

1. Cognito Identity Pools (used to be called Federated Identity): which provides
   1. Provide AWS credentials to users so they can access AWS resources directly
   2. Integrate with Cognito User Pools as an identity provider

Basically provide temporary access to Users to access AWS services

* Cognito vs IAM: “hundreds of users”, ”mobile users”, “authenticate with SAML”

**Cognito User Pools (CUP) – User Features**

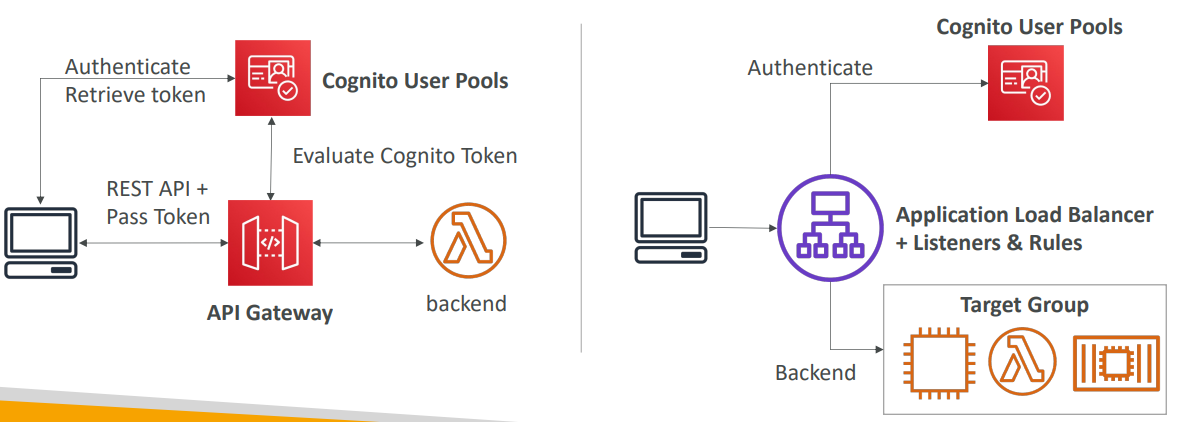
* Create a serverless database of users for your web & mobile apps
* Simple login: Username (or email) / password combination
* Password reset feature
* Email & Phone Number Verification feature
* Multi-factor authentication (MFA) feature
* Federated Identities: users from Facebook, Google, SAML…

**Cognito User Pools (CUP) - Integrations**

• CUP integrates with API Gateway and Application Load Balancer.

The user is going to connect to your Cognito User Pool, retrieve a token, pass this token to the API Gateway, which is going to verify it, and then if it's good, it's going to translate it into the user identity, which will be passed to your lambda function as a backend. And now your lambda function knows the user it's dealing with, and the fact that it has a very specific user that's been well authenticated.

Alternatively, you can also do the exact same thing with your Cognito User Pool on top of your Application Load Balancer. So that means that your application will connect again with your Cognito User Pool, then pass it on to your Application Load Balancer, and make sure that there is a true login. And if it's true, it will redirect the request to your backend, and pass in some additional headers with the identity of the user



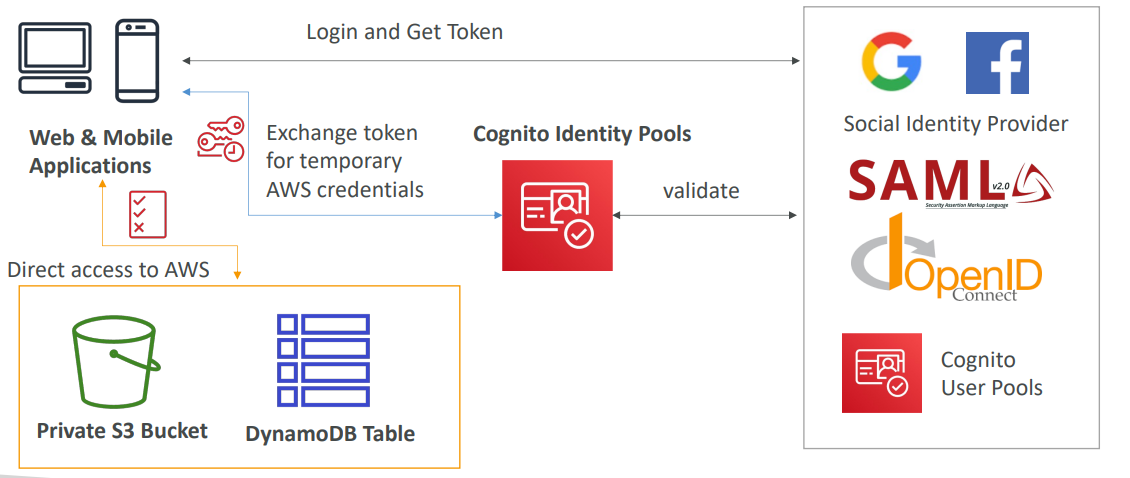
**Cognito Identity Pools (Federated Identities)**

Users don't access our application through the API Gateway or the Application Load Balancer,

* Get identities for “users” so they obtain temporary AWS credentials
* Users source can be Cognito User Pools, 3rd party logins, etc…
* Users can then access AWS services directly or through API Gateway
* The IAM policies applied to the credentials are defined in Cognito
* They can be customized based on the user\_id for fine grained control
* Default IAM roles for authenticated and guest users

**Cognito Identity Pools – Diagram**

Let's say your web and mobile applications want to have direct access to an S3 bucket or DynamoDB table. We're going to have Cognito Identity Pools for this. The web and mobile application are going to log in and get a token out of it. It could be login against Cognito User Pools, or social identity provider, or SAML, or open ID connect and so on. And then it's going to take this token and pass it on to the Cognito Identity Pool service to exchange this token for temporary AWS credentials. And to do so, the Cognito Identity Pool is first going to evaluate whether the token passed is being correct, meaning the login is valid. And the second thing it's going to do is to craft this IAM policy that is specific to that user. These credentials now will, thanks to these very temporary credentials with associated IAM policy, will be able to access AWS on your, for example, your S3 bucket, or your DynamoDB table, without going through an API Gateway or an Application Load Balancer. And this enables for you a use case where you can do row level security in DynamoDB. You would, for example, have this policy coming out of the Cognito Identity Pools. And the idea is that in it, you have a condition, and this condition is saying that the leading key for DynamoDB must be equal to the Cognito identity user ID. And therefore, what's going to happen is that the user with this policy cannot read and write all the items on your DynamoDB table, that would be too much power, but instead they can only deal with the items that they have access to through this condition.



**Cognito Identity Pools Row Level Security in DynamoDB**



## Questions

**Top of Form**

**You have created a Lambda function that typically will take around 1 hour to process some data. The code works fine when you run it locally on your machine, but when you invoke the Lambda function it fails with a "timeout" error after 3 seconds. What should you do?**

* **Configure your Lambda's timeout to 25 minutes**
* **Configure your Lambda's memory to 10 GB**
* **Run your code somewhere else (e.g., EC2 instance)**

**Bottom of Form**

Ans A The maximum timeout you can configure in Lambda is 15 minutes.

B This may speed up the function processing time, but it wouldn't reduce execution time below 15 minutes.

Anc C correct Lambda's maximum execution time is 15 minutes. You can run your code somewhere else such as an EC2 instance or use Amazon ECS.

**Top of Form**

**Before you create a DynamoDB table, you need to provision the EC2 instance the DynamoDB table will be running on.**

* **True**
* **False**

**Bottom of Form**

Ans B DynamoDB is serverless with no servers to provision, patch, or manage and no software to install, maintain or operate. It automatically scales tables up and down to adjust for capacity and maintain performance. It provides both provisioned (specify RCU & WCU) and on-demand (pay for what you use) capacity modes.

**Top of Form**

**You have provisioned a DynamoDB table with 10 RCUs and 10 WCUs. A month later you want to increase the RCU to handle more read traffic. What should you do?**

* **Increase RCU and keep WCU the same**
* **You need to increase both RCU and WCU**
* **Increase RCU and decrease WCU**

**Bottom of Form**

Ans A RCU and WCU are decoupled, so you can increase/decrease each value separately.

**Top of Form**

**You have an e-commerce website where you are using DynamoDB as your database. You are about to enter the Christmas sale and you have a few items which are very popular and you expect that they will be read often. Unfortunately, last year due to the huge traffic you had the ProvisionedThroughputExceededException exception. What would you do to prevent this error from happening again?**

* **Increase the RCU to a very high value**
* **Create a DAX Cluster**
* **Migrate the database away from DynamoDB for the time of the sale**

**Bottom of Form**

Ans A This is not a good solution to solve the Hot Key problem and it will cost you a lot of money.

B Correct DynamoDB Accelerator (DAX) is a fully managed, highly available, in-memory cache for DynamoDB that delivers up to 10x performance improvement. It caches the most frequently used data, thus offloading the heavy reads on hot keys off your DynamoDB table, hence preventing the "ProvisionedThroughputExceededException" exception.

C This requires too much effort and it might not deliver the required performance.

**Top of Form**

**You have developed a mobile application that uses DynamoDB as its datastore. You want to automate sending welcome emails to new users after they sign up. What is the most efficient way to achieve this?**

* **Schedule a Lambda function to run every minute using CloudWatch Events, scan the entire table looking for new users**
* **Enable SNS and DynamoDB integration**
* **Enable DynamoDB Streams and configure it to invoke a Lambda function to send emails**

**Bottom of Form**

Ans A This would work but it wouldn't be a scalable, cheap, or efficient solution.

B There's no such feature.

C DynamoDB Streams allows you to capture a time-ordered sequence of item-level modifications in a DynamoDB table. It's integrated with AWS Lambda so that you create triggers that automatically respond to events in real-time.

**Top of Form**

**To create a serverless API, you should integrate Amazon API Gateway with ......................**

* **EC2 Instance**
* **Elastic Load Balancing**
* **AWS Lambda**

**Bottom of Form**

Ans A EC2 instances are not serverless.

B An ELB does not help create an API on its own.

C correct

**Top of Form**

**When you are using an Edge-Optimized API Gateway, your API Gateway lives in CloudFront Edge Locations across all AWS Regions.**

* **False**
* **True**

**Bottom of Form**

Ans A correct An Edge-Optimized API Gateway is best for geographically distributed clients. API requests are routed to the nearest CloudFront Edge Location which improves latency. The API Gateway still lives in one AWS Region.

**Top of Form**

**You are running an application in production that is leveraging DynamoDB as its datastore and is experiencing smooth sustained usage. There is a need to make the application run in development mode as well, where it will experience the unpredictable volume of requests. What is the most cost-effective solution that you recommend?**

* **Use Provisioned Capacity Mode with Auto Scaling enabled for both development and production**
* **Use Provisioned Capacity Mode with Auto Scaling enabled for production and use On-Demand Capacity Mode for development**
* **Use Provisioned Capacity Mode with Auto Scaling enabled for development and use On-Demand Capacity Mode for production**
* **Use On-Demand Capacity Mode for both development and production**

**Bottom of Form**

Ans B correct

D This will work, but it would be extremely expensive as On-Demand Capacity Mode is much more expensive than Provisioned Capacity Mode. As we know production has a smooth sustained curve, Provisioned Capacity Mode with Auto Scaling enabled would be perfect.

**Top of Form**

**You have an application that is served globally using CloudFront Distribution. You want to authenticate users at the CloudFront Edge Locations instead of authentication requests go all the way to your origins. What should you use to satisfy this requirement?**

* **Lambda@Edge**
* **API Gateway**
* **DynamoDB**
* **AWS Global Accelerator**

Ans A correct Lambda@Edge is a feature of CloudFront that lets you run code closer to your users, which improves performance and reduces latency.Bottom of Form

Ans B API Gateway is a fully managed service that makes it easy to create, publish, maintain, monitor, and secure APIs at any scale. It can't be used to run code globally on CloudFront Edge Locations.

Ans D AWS Global Accelerator allows you to improve the performance of your users' traffic by up to 60% using AWS global network infrastructure. It can't be used to run code globally on CloudFront Edge Locations.

**Top of Form**

**The maximum size of an item in a DynamoDB table is ...................**

* **1 MB**
* **500 KB**
* **400 KB**
* **400 MB**

**Bottom of Form**

Ans c

**Top of Form**

**Which AWS service allows you to build Serverless workflows using AWS services (e.g., Lambda) and supports human approval?**

* **AWS Lambda**
* **Amazon ECS**
* **AWS Step Functions**
* **AWS Storage Gateway**

**Bottom of Form**

Ans C

**Top of Form**

**A company has a serverless application on AWS which consists of Lambda, DynamoDB, and Step Functions. In the last month, there are an increase in the number of requests against the application which results in an increase in DynamoDB costs, and requests started to be throttled. After further investigation, it shows that the majority of requests are read requests against some queries in the DynamoDB table. What do you recommend to prevent throttles and reduce costs efficiently?**

* **Use an EC2 instance with Redis installed and place it between the Lambda function and the DynamoDB table**
* **Migrate from DynamoDB to Aurora and use ElastiCache to cache the most requested read data**
* **Migrate from DynamoDB to S3 and use CloudFront to cache the most requested read data**
* **Use DynamoDB Accelerator (DAX) to cache the most requested read data**

**Bottom of Form**

Ans D

**Top of Form**

**You are a DevOps engineer in a football company that has a website that is backed by a DynamoDB table. The table stores viewers’ feedback for football matches. You have been tasked to work with the analytics team to generate reports on the viewers’ feedback. The analytics team wants the data in DynamoDB in json format and hosted in an S3 bucket to start working on it and create the reports. What is the best and most cost-effective way to convert DynamoDB data to json files?**

* **Select DynamoDB table then select Export to S3**
* **Create a Lambda function to read DynamoDB data, convert them to json files, then store the files in S3 bucket**
* **Use AWS Transfer Family**
* **Use AWS DataSync**

**Bottom of Form**

Ans A

**Top of Form**

**A website is currently in the development process and it is going to be hosted on AWS. There is a requirement to store user sessions for users logged in to the website with an automatic expiry and deletion of expired user sessions. Which of the following AWS services are best suited for this use case?**

* **Store users’ sessions in an S3 bucket and enable S3 Lifecycle Policy**
* **Store users’ sessions locally in an EC2 instance**
* **Store users’ sessions in a DynamoDB table and enable TTL**
* **Store users’ sessions in an EFS file system**

**Bottom of Form**

Ans C correct

**Top of Form**

**You have a mobile application and would like to give your users access to their own personal space in the S3 bucket. How do you achieve that?**

* **Generate IAM user credentials for each of your application's users**
* **Use Amazon Cognito Identity Federation**
* **Use SAML Identity Federation**
* **Use a Bucket Policy to make your bucket public**

**Bottom of Form**

Ans A This does not scale well, we cannot create an IAM user per mobile application user.

B Amazon Cognito can be used to federate mobile user accounts and provide them with their own IAM permissions, so they can be able to access their own personal space in the S3 bucket.

C SAML Identity Federation is used to integrate an Identity Provider service such as Microsoft Active Directory with AWS. It does not work for mobile applications.

D Never make an S3 bucket public to your mobile application users. This would result in data leaks! Read horror stories here: <https://businessinsights.bitdefender.com/worst-amazon-breaches>

**Top of Form**

**You are developing a new web and mobile application that will be hosted on AWS and currently, you are working on developing the login and signup page. The application backend is serverless and you are using Lambda, DynamoDB, and API Gateway. Which of the following is the best and easiest approach to configure the authentication for your backend?**

* **Store users’ credentials in a DynamoDB table encrypted using KMS**
* **Store users’ credentials in an S3 bucket encrypted using KMS**
* **Use Cognito User Pools**
* **Store users’ credentials in AWS Secrets Manager**

**Bottom of Form**

Ans C

**Top of Form**

**You are running a mobile application where you want each registered user to upload/download images to/from his own folder in the S3 bucket. Also, you want to give your users to sign-up and sign in using their social media accounts (e.g., Facebook). Which AWS service should you choose?**

* **AWS Identity and Access Management (IAM)**
* **AWS IAM Identity Center**
* **Amazon Cognito**
* **Amazon CloudFront**

**Bottom of Form**

Ans A AWS IAM enables you to manage access to AWS services and resources securely. It allows you to create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources. IAM cannot be used to federate mobile users.

B AWS IAM Identity Center makes it easy to centrally manage SSO access to all of your AWS accounts and cloud applications. It helps you manage SSO access and user permissions across all your AWS accounts in AWS Organizations. It also helps when you manage access and permissions to commonly used 3rd party software as a service (SaaS) application, AWS SSO-integrated applications as well as custom applications that support Security Assertion Markup Language (SAML) 2.0. AWS SSO also includes built-in SAML integrations to many business applications, such as Salesforce, Box, and Office 365.

Correct Amazon Cognito lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and easily. Amazon Cognito scales to millions of users and supports sign-in with social identity providers, such as Apple, Facebook, Google, and Amazon, and enterprise identity providers via SAML 2.0 and OpenID Connect.

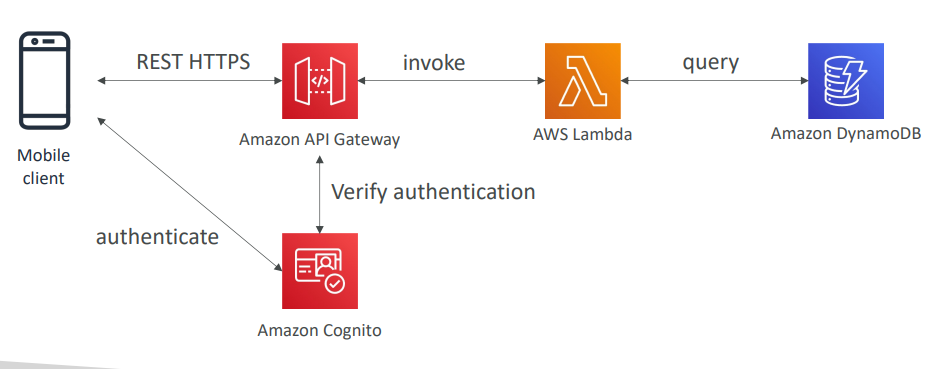
D Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds. Amazon CloudFront does not federate mobile users or generate security credentials.

# Serverless Architectures

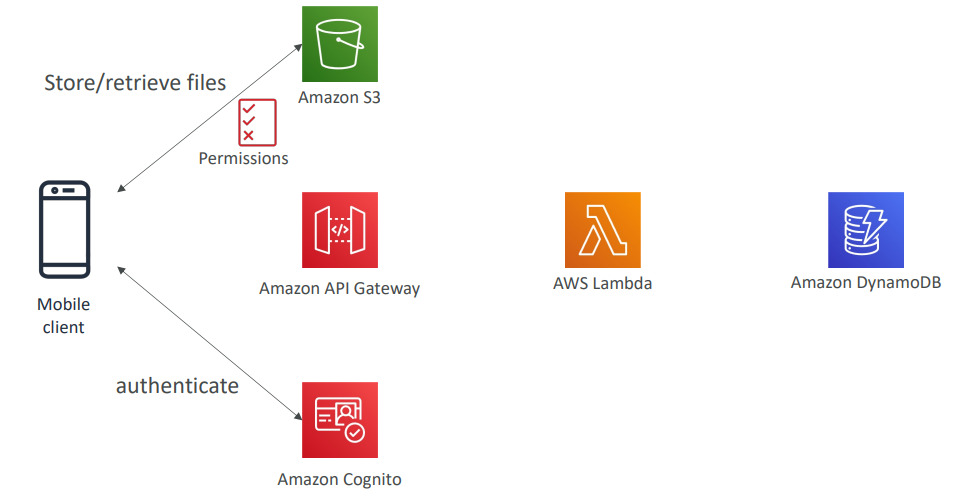
**Mobile application: MyTodoList**

* We want to create a mobile application with the following requirements
* Expose as REST API with HTTPS
* Serverless architecture
* Users should be able to directly interact with their own folder in S3
* Users should authenticate through a managed serverless service
* The users can write and read to-dos, but they mostly read them
* The database should be scaled, and have some high read throughput

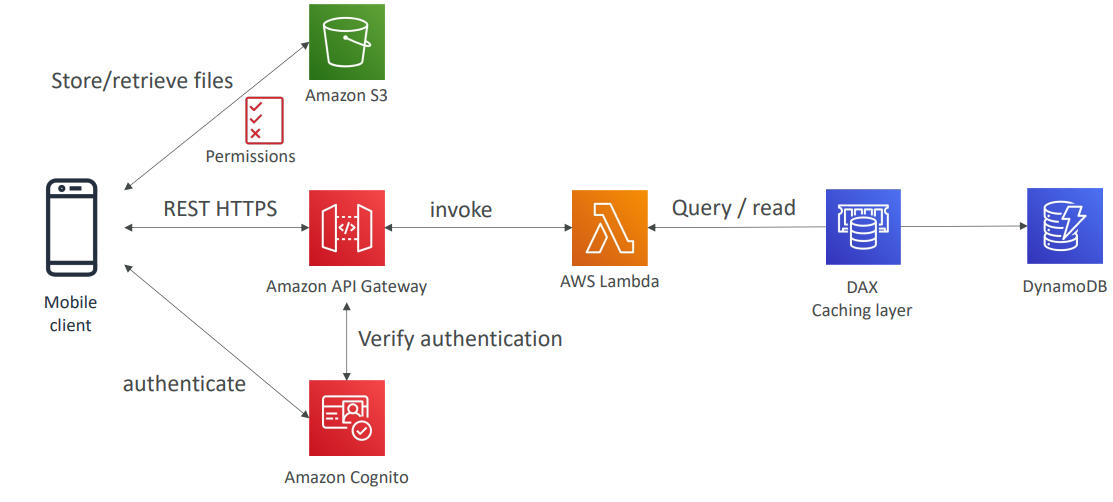
**Mobile app: REST API layer**



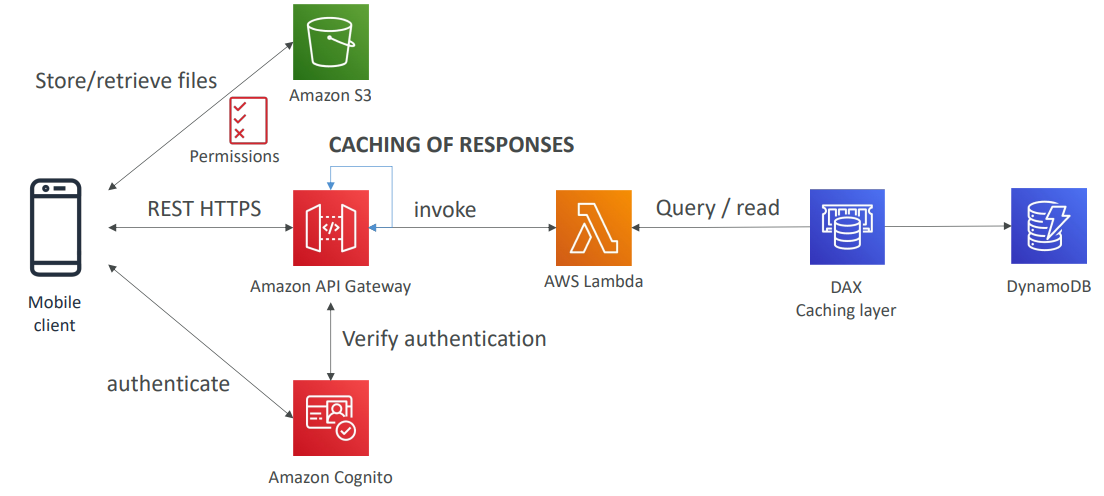
**Mobile app: giving users access to S3**



**Mobile app: high read throughput, static data**



**Mobile app: caching at the API Gateway**



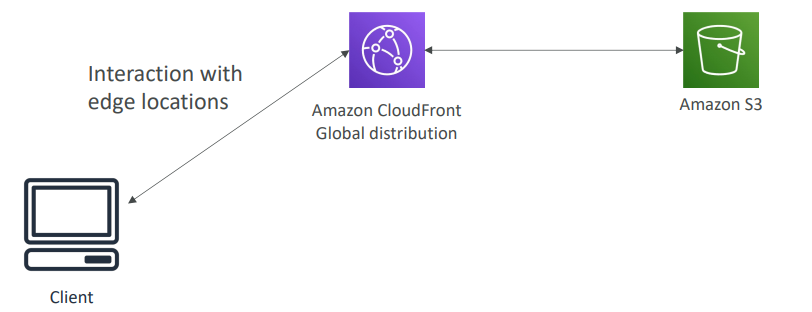
**In this lecture**

* Serverless REST API: HTTPS, API Gateway, Lambda, DynamoDB
* Using Cognito to generate temporary credentials to access S3 bucket with restricted policy. App users can directly access AWS resources this way. Pattern can be applied to DynamoDB, Lambda…
* Caching the reads on DynamoDB using DAX
* Caching the REST requests at the API Gateway level
* Security for authentication and authorization with Cognito

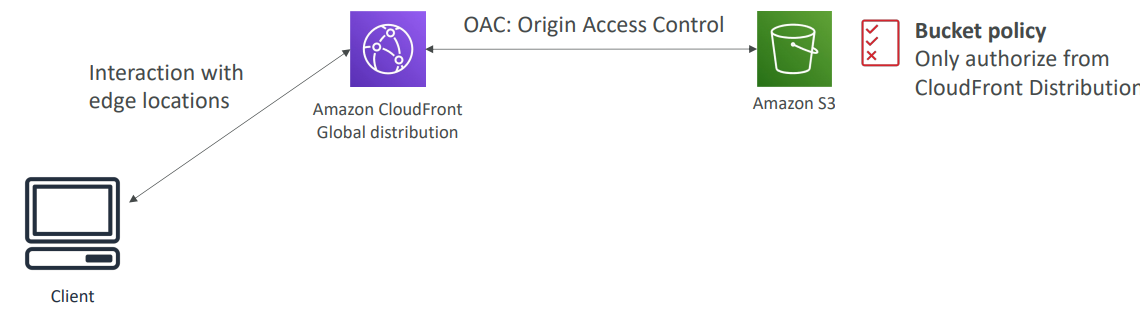
**Serverless hosted website: MyBlog.com**

* This website should scale globally
* Blogs are rarely written, but often read
* Some of the website is purely static files, the rest is a dynamic REST API
* Caching must be implement where possible
* Any new users that subscribes should receive a welcome email
* Any photo uploaded to the blog should have a thumbnail generated

**Serving static content, globally**



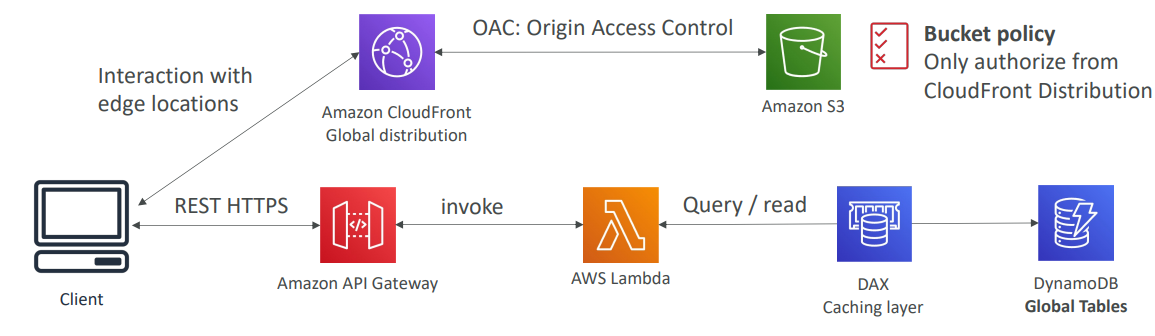
**Serving static content, globally, securely**



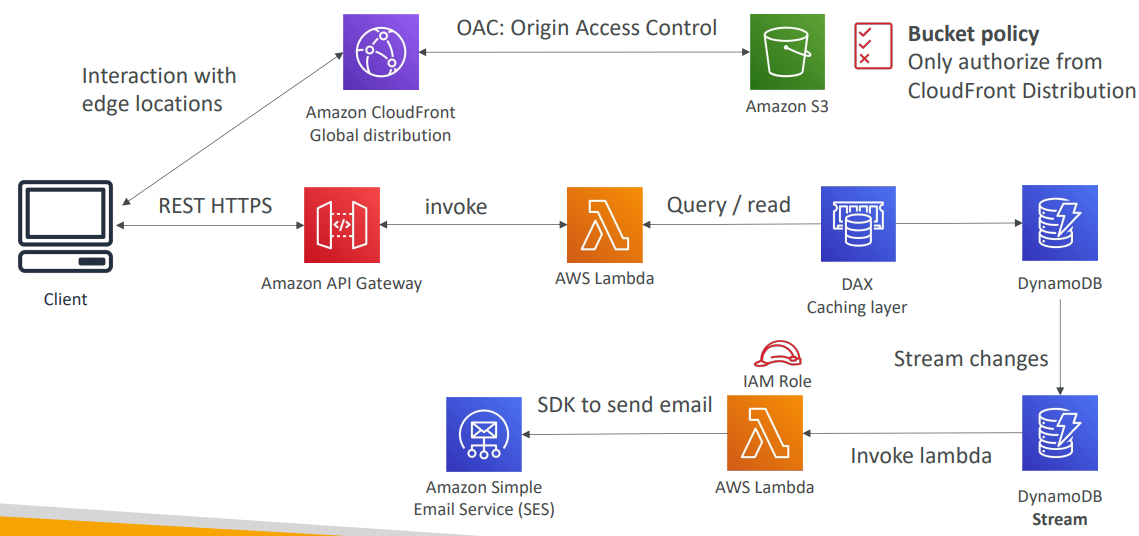
**Adding a public serverless REST API**



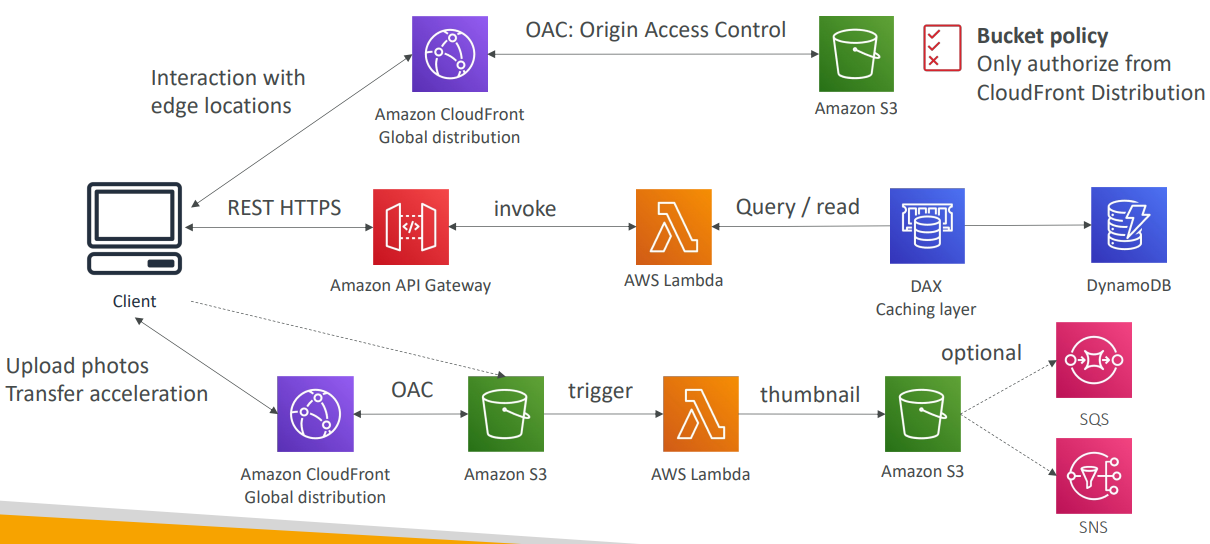
**Leveraging DynamoDB Global Tables**



**User Welcome email flow**



**Thumbnail Generation flow**



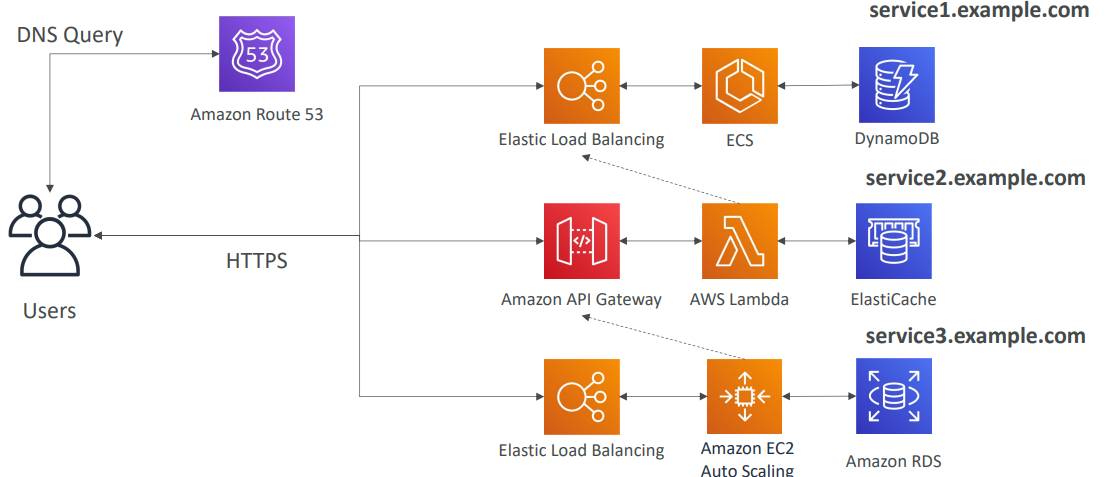
**AWS Hosted Website Summary**

* We’ve seen static content being distributed using CloudFront with S3
* The REST API was serverless, didn’t need Cognito because public
* We leveraged a Global DynamoDB table to serve the data globally
* (we could have used Aurora Global Database)
* We enabled DynamoDB streams to trigger a Lambda function
* The lambda function had an IAM role which could use SES
* SES (Simple Email Service) was used to send emails in a serverless way
* S3 can trigger SQS / SNS / Lambda to notify of events

**Micro Services architecture**

* We want to switch to a micro service architecture
* Many services interact with each other directly using a REST API
* Each architecture for each micro service may vary in form and shape
* We want a micro-service architecture so we can have a leaner development lifecycle for each service

**Micro Services Environment**



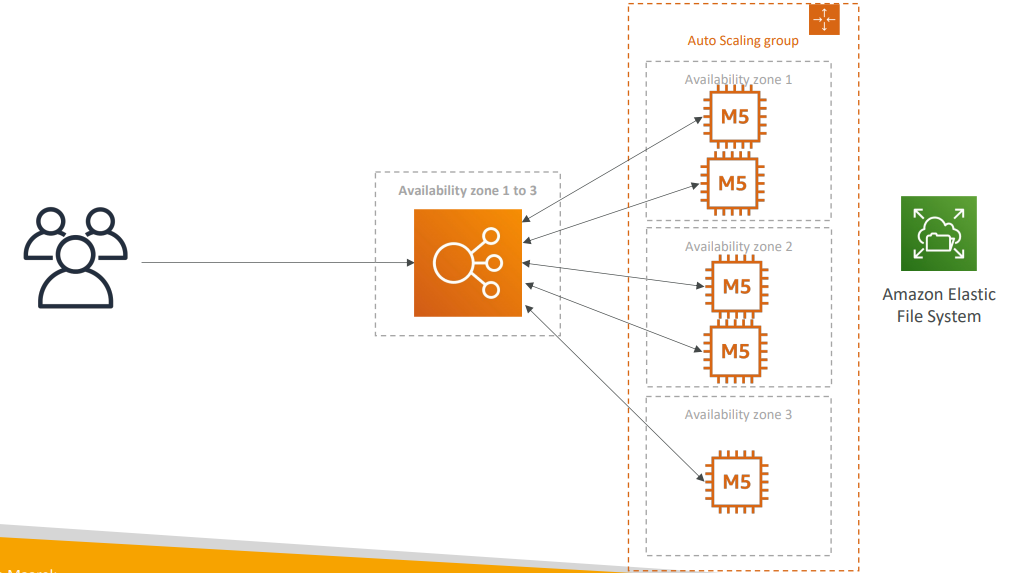
**Discussions on Micro Services**

* You are free to design each micro-service the way you want
* Synchronous patterns: API Gateway, Load Balancers
* Asynchronous patterns: SQS, Kinesis, SNS, Lambda triggers (S3)
* Challenges with micro-services:
  + repeated overhead for creating each new microservice,
  + issues with optimizing server density/utilization
  + complexity of running multiple versions of multiple microservices simultaneously
  + proliferation of client-side code requirements to integrate with many separate services.
* Some of the challenges are solved by Serverless patterns:
  + API Gateway, Lambda scale automatically and you pay per usage
  + You can easily clone API, reproduce environments
  + Generated client SDK through Swagger integration for the API Gateway

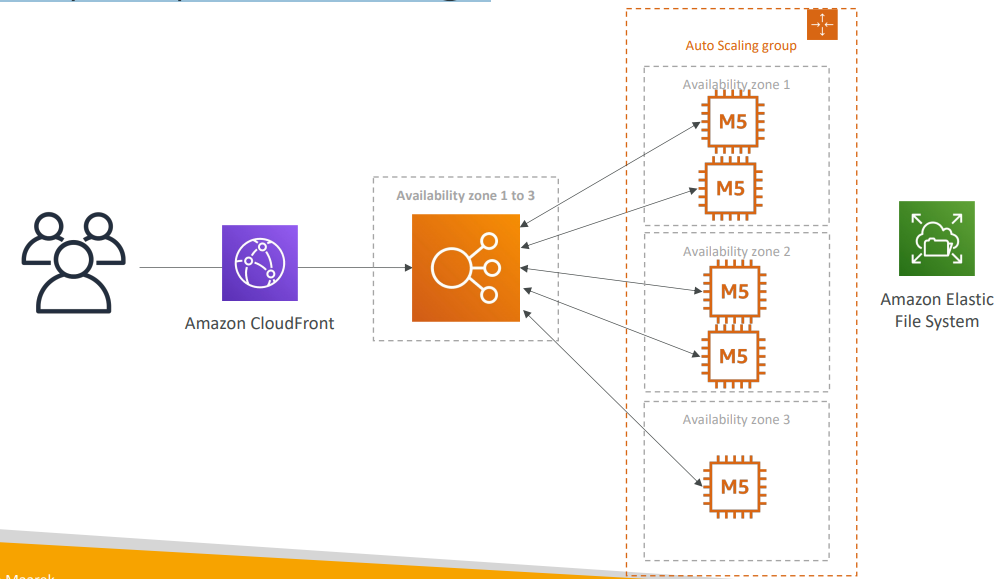
**Software updates offloading**

* We have an application running on EC2, that distributes software updates once in a while
* When a new software update is out, we get a lot of request and the content is distributed in mass over the network. It’s very costly
* We don’t want to change our application, but want to optimize our cost and CPU, how can we do it?

**Our application current state**



**Easy way to fix things!**



**Why CloudFront?**

* No changes to architecture
* Will cache software update files at the edge
* Software update files are not dynamic, they’re static (never changing)
* Our EC2 instances aren’t serverless
* But CloudFront is, and will scale for us
* Our ASG will not scale as much, and we’ll save tremendously in EC2
* We’ll also save in availability, network bandwidth cost, etc
* Easy way to make an existing application more scalable and cheaper!

# Databases in AWS

* We have a lot of managed databases on AWS to choose from
* Questions to choose the right database based on your architecture:
  + Read-heavy, write-heavy, or balanced workload? Throughput needs? Will it
  + change, does it need to scale or fluctuate during the day?
  + How much data to store and for how long? Will it grow? Average object size? How are they accessed?
  + Data durability? Source of truth for the data ?
  + Latency requirements? Concurrent users?
  + Data model? How will you query the data? Joins? Structured? Semi-Structured?
  + Strong schema? More flexibility? Reporting? Search? RDBMS / NoSQL?
  + License costs? Switch to Cloud Native DB such as Aurora?

**Database Types**

* RDBMS (= SQL / OLTP): RDS, Aurora – great for joins
* NoSQL database – no joins, no SQL : DynamoDB (~JSON), ElastiCache (key / value pairs), Neptune (graphs), DocumentDB (for MongoDB), Keyspaces (for Apache Cassandra)
* Object Store: S3 (for big objects) / Glacier (for backups / archives)
* Data Warehouse (= SQL Analytics / BI): Redshift (OLAP), Athena, EMR
* Search: OpenSearch (JSON) – free text, unstructured searches
* Graphs: Amazon Neptune – displays relationships between data
* Ledger: Amazon Quantum Ledger Database
* Time series: Amazon Timestream

Note: some databases are being discussed in the Data & Analytics section

## Amazon RDS

* Managed PostgreSQL / MySQL / Oracle / SQL Server / DB2 / MariaDB / Custom
* Provisioned RDS Instance Size and EBS Volume Type & Size
* Auto-scaling capability for Storage
* Support for Read Replicas and Multi AZ for HA(can’t perform query)
* Security through IAM, Security Groups, KMS , SSL in transit
* Automated Backup with Point in time restore feature (up to 35 days)
* Manual DB Snapshot for longer-term recovery
* Managed and Scheduled maintenance (with downtime)
* Support for IAM Authentication, integration with Secrets Manager
* RDS Custom for access to and customize the underlying instance (Oracle & SQL Server)
* Use case: Store relational datasets (RDBMS / OLTP), perform SQL queries, transactions

## Amazon Aurora

* Compatible API for PostgreSQL / MySQL, separation of storage and compute
* Storage: data is stored in 6 replicas, across 3 AZ – highly available, self-healing, auto-scaling
* Compute: Cluster of DB Instance across multiple AZ, auto-scaling of Read Replicas
* Cluster: Custom endpoints for writer and reader DB instances
* Same security / monitoring / maintenance features as RDS
* Know the backup & restore options for Aurora
* **Aurora Serverless** – for unpredictable / intermittent workloads, no capacity planning
* **Aurora Global**: up to 16 DB Read Instances in each region, < 1 second storage replication
* **Aurora Machine Learning**: perform ML using SageMaker & Comprehend on Aurora
* **Aurora Database Cloning**: new cluster from existing one, faster than restoring a snapshot
* Use case: same as RDS, but with less maintenance / more flexibility / more performance / more features

## Amazon ElastiCache

* Managed Redis / Memcached (similar offering as RDS, but for caches)
* In-memory data store, sub-millisecond latency
* Select an ElastiCache instance type (e.g., cache.m6g.large)
* Support for Clustering (Redis) and Multi AZ, Read Replicas (sharding)
* Security through IAM, Security Groups, KMS, Redis Auth
* Backup / Snapshot / Point in time restore feature
* Managed and Scheduled maintenance
* Requires some application code changes to be leveraged
* Use Case: Key/Value store, Frequent reads, less writes, cache results for DB queries, store session data for websites, cannot use SQL

## Amazon DynamoDB

* AWS proprietary technology, managed serverless NoSQL database, millisecond latency
* Capacity modes: provisioned capacity with optional auto-scaling or on-demand capacity
* Can replace ElastiCache as a key/value store (storing session data for example, using TTL feature)
* Highly Available, Multi AZ by default, Read and Writes are decoupled, transaction capability
* DAX cluster for reading cache, microsecond read latency
* Security, authentication and authorization is done through IAM
* Event Processing: DynamoDB Streams to integrate with AWS Lambda, or Kinesis Data Streams
* Global Table feature: active-active setup. Which means anyone can read from any region
* Automated backups up to 35 days with PITR (restore to new table), or on-demand backups
* Export to S3 without using RCU within the PITR window, import from S3 without using WCU
* Great to rapidly evolve schemas
* Use Case: Serverless applications development (small documents 100s KB), distributed serverless cache

## Amazon S3

* S3 is a… key / value store for objects
* Great for bigger objects, not so great for many small objects
* Serverless, scales infinitely, max object size is 5 TB, versioning capability
* Tiers: S3 Standard, S3 Infrequent Access, S3 Intelligent, S3 Glacier + lifecycle policy
* Features: Versioning, Encryption, Replication, MFA-Delete, Access Logs…
* Security: IAM, Bucket Policies, ACL, Access Points, Object Lambda, CORS, Object/Vault Lock
* Encryption: SSE-S3, SSE-KMS, SSE-C, client-side, TLS in transit, default encryption
* Batch operations on objects using S3 Batch, listing files using S3 Inventory
* Performance: Multi-part upload, S3 Transfer Acceleration, S3 Select
* Automation: S3 Event Notifications (SNS, SQS, Lambda, EventBridge)
* Use Cases: static files, key value store for big files, website hosting

## Document DB

* Aurora is an “AWS-implementation” of PostgreSQL / MySQL …
* Document DB is the same as MongoDB (which is a NoSQL database)
* MongoDB is used to store, query, and index JSON data
* Similar “deployment concepts” as Aurora
* Fully Managed, highly available with replication across 3 AZ
* Document DB storage automatically grows in increments of 10GB
* Automatically scales to workloads with millions of requests per seconds

## Amazon Neptune

|  |  |
| --- | --- |
| * Fully managed graph database * A popular graph dataset would be a social network   + Users have friends   + Posts have comments   + Comments have likes from users   + Users share and like posts… * Highly available across 3 AZ, with up to 15 read replicas * Build and run applications working with highly connected datasets – optimized for these complex and hard queries * Can store up to billions of relations and query the graph with milliseconds latency * Highly available with replications across multiple AZs * Great for knowledge graphs (Wikipedia), fraud detection, recommendation engines, social networking |  |

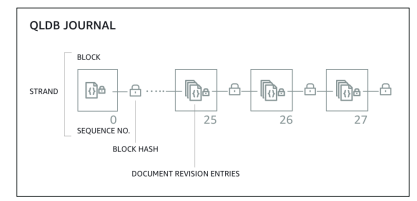
|  |  |
| --- | --- |
| * Real-time ordered sequence of every change to your graph data * Changes are available immediately after writing * No duplicates, strict order * Streams data is accessible in an HTTP REST API * Use cases:   + Send notifications when certain changes are made   + Maintain your graph data synchronized in another data store (e.g., S3, OpenSearch, ElastiCache)   + Replicate data across regions in Neptune |  |

## Amazon Key spaces (for Apache Cassandra)

* Apache Cassandra is an open-source NoSQL distributed database
* A managed Apache Cassandra-compatible database service
* Serverless, Scalable, highly available, fully managed by AWS
* Automatically scale tables up/down based on the application’s traffic
* Tables are replicated 3 times across multiple AZ
* Using the Cassandra Query Language (CQL)
* Single-digit millisecond latency at any scale, 1000s of requests per second
* Capacity: On-demand mode or provisioned mode with auto-scaling
* Encryption, backup, Point-In-Time Recovery (PITR) up to 35 days
* Use cases: store IoT devices info, time-series data, …

## Amazon QLDB

* QLDB stands for ”Quantum Ledger Database”
* A ledger is a book recording financial transactions
* Fully Managed, Serverless, High available, Replication across 3 AZ
* Used to review the history of all the changes made to your application data over time
* Immutable system: no entry can be removed or modified, cryptographically verifiable



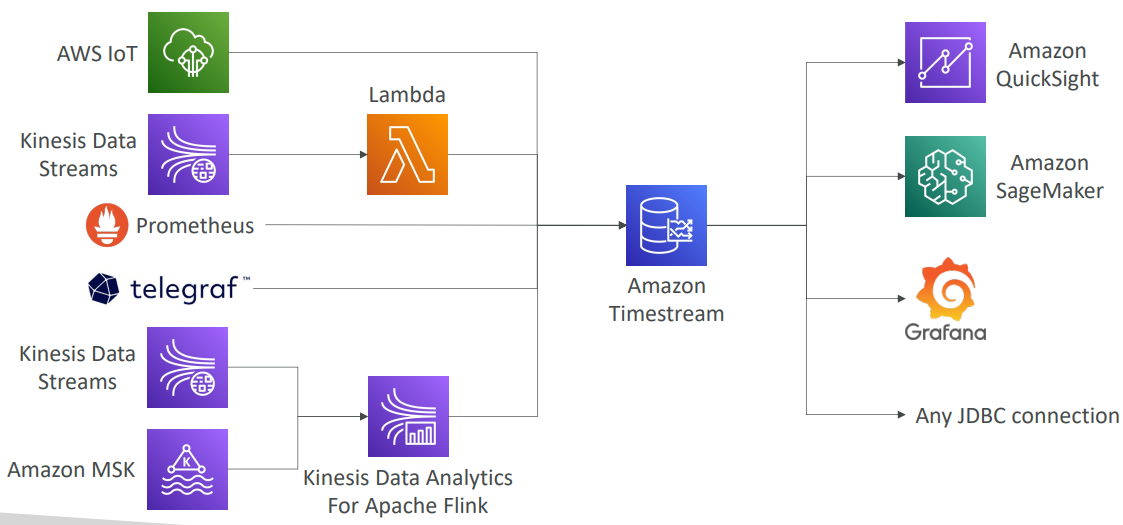
* 2-3x better performance than common ledger blockchain frameworks, manipulate data using SQL
* Difference with Amazon Managed Blockchain: no decentralization component, in accordance with financial regulation rules

## Amazon Timestream

It's actually a time series database. It's fully managed, it's fast, it's scalable and it's serverless. It's a bunch of points that have a time included in them.

|  |  |
| --- | --- |
| * Fully managed, fast, scalable, serverless time series database * Automatically scales up/down to adjust capacity * Store and analyze trillions of events per day * 1000 times faster & 1/10th the cost of relational databases * Scheduled queries, multi-measure records, SQL compatibility * Data storage tiering: recent data kept in memory and historical data kept in a cost -optimized storage * Built -in-time series analytics functions (helps you identify patterns in your data in near real-time) * Encryption in transit and at rest * Use cases: IoT apps, operational applications, real -time analytics, … |  |

**Amazon Timestream – Architecture**



## Questions

**Top of Form**

**Which database helps you store relational datasets, with SQL language compatibility and the capability of processing transactions such as insert, update, and delete?**

* **Amazon DocumentDB**
* **Amazon RDS**
* **Amazon DynamoDB**
* **Amazon ElastiCache**

**Bottom of Form**

Ans B correct

C Amazon DynamoDB is a key-value, document, NoSQL database. It does not have SQL and does not store the data in a relational format but can be used for transactional processing.

D Amazon ElastiCache is a caching technology, and it doesn't have SQL capabilities.

**Top of Form**

**Which AWS service provides you with caching capability that is compatible with Redis API?**

* **Amazon RDS**
* **Amazon DynamoDB**
* **Amazon OpenSearch**
* **Amazon ElastiCache**

**Bottom of Form**

Ans B Amazon DynamoDB can be used as a substitute for Redis but it does not support Redis API.

C Amazon OpenSearch helps you in indexing and searching through your data. It leverages HTTP for client interactions.

D Correct Amazon ElastiCache is a fully managed in-memory data store, compatible with Redis or Memcached.

**Top of Form**

**You want to migrate an on-premises MongoDB NoSQL database to AWS. You don't want to manage any database servers, so you want to use a managed NoSQL Serverless database, that provides you with high availability, durability, and reliability, and the capability to take your database global. Which database should you choose?**

* **Amazon RDS**
* **Amazon DynamoDB**
* **Amazon DocumentDB**
* **Amazon Aurora**

**Bottom of Form**

Ans A Amazon RDS is a relational, SQL-based database.

B Correct Amazon DynamoDB is a key-value, document, NoSQL database.

C Amazon DocumentDB is a NoSQL databases and it might work perfectly here. But the questions states that the database must be Serverless and wants to take the database global. DocumentDB doesn't have a Serverless option and it doesn't have a global database feature.

D Amazon Aurora is a relational, SQL-based database.

**Top of Form**

**You are looking to perform Online Transaction Processing (OLTP). You would like to use a database that has built-in auto-scaling capabilities and provides you with the maximum number of replicas for its underlying storage. What AWS service do you recommend?**

* **Amazon ElastiCache**
* **Amazon Neptune**
* **Amazon Aurora**
* **Amazon RDS**

**Bottom of Form**

Ans D Amazon RDS does not have built-in auto-scaling and data replication capabilities. You need to create RDS Read Replicas and enable Multi-AZ.

A Amazon ElastiCache is an in-memory data store, provides you with caching capabilities. It is compatible with Redis and Memcached. It does not support OLTP.

C Correct Amazon Aurora is a MySQL and PostgreSQL-compatible relational database. It features a distributed, fault-tolerant, self-healing storage system that auto-scales up to 128TB per database instance. It delivers high performance and availability with up to 15 low-latency read replicas, point-in-time recovery, continuous backup to Amazon S3, and replication across 3 AZs.

**Top of Form**

**As a Solutions Architect, a startup company asked you for help as they are working on an architecture for a social media website where users can be friends with each other, and like each other's posts. The company plan on performing some complicated queries such as *"What are the number of likes on the posts that have been posted by the friends of Mike?"*. Which database do you recommend?**

* **Amazon RDS**
* **Amazon QLDB**
* **Amazon Neptune**
* **Amazon OpenSearch**

**Bottom of Form**

Ans A This would work, but a graph database is better suited.

C Correct Amazon Neptune is a fast, reliable, fully-managed graph database service that makes it easy to build and run applications that work with highly connected datasets.

D This would work, but a graph database is better suited.

**Top of Form**

**You have a set of files, 100MB each, that you want to store in a reliable and durable key-value store. Which AWS service do you recommend?**

* **Amazon Aurora**
* **Amazon S3**
* **Amazon DynamoDB**
* **Amazon ElastiCache**

**Bottom of Form**

Ans D Amazon ElastiCache is a fully managed in-memory data store, compatible with Redis or Memcached. It can not be used to store big objects, in addition to it is not durable.

C Amazon DynamoDB can not be used to store big objects. The maximum item size in DynamoDB is 400KB.

B Amazon S3 is indeed a key-value store! (where the key is the full path of the object in the bucket)

**Top of Form**

**A company has an on-premises website that uses ReactJS as its frontend, NodeJS as its backend, and MongoDB for the database. There are some issues with the self-hosted MongoDB database as there is a lot of maintenance required and they don’t have and can’t afford the resources or experience to handle those issues. So, a decision was made to migrate the website to AWS. They have decided to host the frontend ReactJS application in an S3 bucket and the NodeJS backend on a set of EC2 instances. Which AWS service can they use to migrate the MongoDB database that provides them with high scalability and availability without making any code changes?**

* **Amazon ElastiCache**
* **Amazon DocumentDB**
* **Amazon RDS for MongoDB**
* **Amazon Neptune**

**Bottom of Form**

Ans B Correct

**Top of Form**

**A company using a self-hosted on-premises Apache Cassandra database which they want to migrate to AWS. Which AWS service can they use which provides them with a fully managed, highly available, and scalable Apache Cassandra database?**

* **Amazon DocumentDB**
* **Amazon DynamoDB**
* **Amazon Timestream**
* **Amazon Keyspaces**

**Bottom of Form**

Ans D Correct

**Top of Form**

**An online payment company is using AWS to host its infrastructure. Due to the application’s nature, they have a strict requirement to store an accurate record of financial transactions such as credit and debit transactions. Those transactions must be stored in secured, immutable, encrypted storage which can be cryptographically verified. Which AWS service is best suited for this use case?**

* **Amazon DocumentDB**
* **Amazon Aurora**
* **Amazon QLDB**
* **Amazon Neptune**

**Bottom of Form**

Ans C correct

**Top of Form**

**A startup is working on developing a new project to reduce forest fires due to climate change. The startup is developing sensors that will be spread across the entire forest to make some readings such as temperature, humidity, and pressures which will help detect the forest fires before it happens. They are going to have thousands of sensors that are going to store a lot of readings each second. There is a requirement to store those readings and do fast analytics so they can predict if there is a fire. Which AWS service can they use to store those readings?**

* **Amazon Timestream**
* **Amazon Neptune**
* **Amazon S3**
* **Amazon ElastiCache**

**Bottom of Form**

Ans A correct

# Data & Analytics

## Amazon Athena

Users are going to load data into your S3 bucket, or you are going to load data into your S3 bucket, and then you would use the Athena service to query and analyze this data in Amazon S3 without moving it. Athena is serverless, and it analyzes directly your data living in your S3 bucket.

|  |  |
| --- | --- |
| * Serverless query service to analyze data stored in Amazon S3 * It uses standard SQL language to query the files (built on Presto which uses SQL) * Supports CSV, JSON, ORC, Avro, and Parquet * Pricing: $5.00 per TB of data scanned * It’s serverless * Commonly used with Amazon Quick sight for reporting/dashboards * Use cases: Business intelligence / analytics / reporting, analyze & query VPC Flow Logs, ELB Logs, CloudTrail trails, etc... * Anytime you need to analyze data in S3 using serverless SQL, use Athena |  |

**Amazon Athena – Performance Improvement**

* Because you pay for data scanned per terabyte, you need to use a type of data where you're going to scan less data. And for this, you can use a columnar data type for cost-savings, because you only scan the columns you need. The recommended formats for Amazon Athena are Apache Parquet and ORC, and it's going to give you a huge performance improvement. And to get your files into the Apache Parquet or ORC format, you must use a service, for example, Glue. Glue can be very helpful to convert your data as an ETL job, between, for example, CSV and Parquet.
* Compress data for smaller retrievals (bzip2, gzip, lz4, snappy, zlip, zstd…)
* If you know you're going to query all the time on some specific columns, you can partition your datasets, and partition datasets means that in your S3 bucket, you're going to have the full path with slashes, and each slash will be a different column name with a specific value. And so, you're organizing, you're partitioning your data in Amazon S3, so that when you query it, you can know exactly in which folder at which path Amazon S3, you need to scan for data.

For e.g. we have flight data in Parquet formats, and then we do /year=1991. We partition by year, and we'll have one folder for each year. Then within each year we'll have month, so month equals one, and within each month we'll have days, this equals one. And so when I do a query on Athena, and I filter for a specific year, a specific month, and a specific day, then we'll know exactly to which folder in Amazon S3 to get the data from, and therefore we'll only recover a subset of the data. Therefore, we'll have really, really good partitioning.

* + s3://yourBucket/pathToTable

/<PARTITION\_COLUMN\_NAME>=<VALUE>

/<PARTITION\_COLUMN\_NAME>=<VALUE>

<PARTITION\_COLUMN\_NAME>=<VALUE>

/etc…

* + Example: s3://athena-examples/flight/parquet/year=1991/month=1/day=1/
* Use larger files (> 128 MB) to minimize overhead. It’s to minimize your overheads. If you have many, many, many small files in Amazon S3, Athena is not going to be as fast in performance, as if you had larger files,

**Amazon Athena – Federated Query**

|  |  |
| --- | --- |
| * It allows you to run SQL queries across data stored in relational, non-relational, object, and custom data sources (AWS or on-premises) * Have to use what's called a Data Source Connector. It's a Lambda function, and that Lambda function is going to run the Federated Queries in other services. So that could be, for example, CloudWatch Logs, DynamoDB, RDS, and so on.   For example, we have Athena here, and we have a Lambda function, and you'll have one Lambda function per Data Source Connector. And then through Amazon Athena, you can run the query across ElastiCache, Document DB, DynamoDB, Redshift, Aurora, SQL Server, MySQL, HBase on the EMR service, or any on-premises database. They're clear from Athena, as well as, of course, Amazon S3, and you can do your drawings, and you can do your competitions, and so on. That's why it's called a Federated Query. And then the results of this query can be stored into your Amazon S3 buckets for later analysis.   * Store the results back in Amazon S3 |  |

## Redshift

* Redshift is based on PostgreSQL, but it’s not used for OLTP (Online Transaction Processing)
* It’s OLAP – online analytical processing (analytics and data warehousing)
* 10x better performance than other data warehouses, and it can scale to PBs of data
* Columnar storage of data (instead of row based) & parallel query engine
* Two modes: Provisioned cluster or Serverless cluster
* Has a SQL interface for performing the queries
* BI tools such as Amazon Quicksight or Tableau integrate with it
* vs Athena: faster queries / joins / aggregations thanks to indexes

**Redshift Cluster**

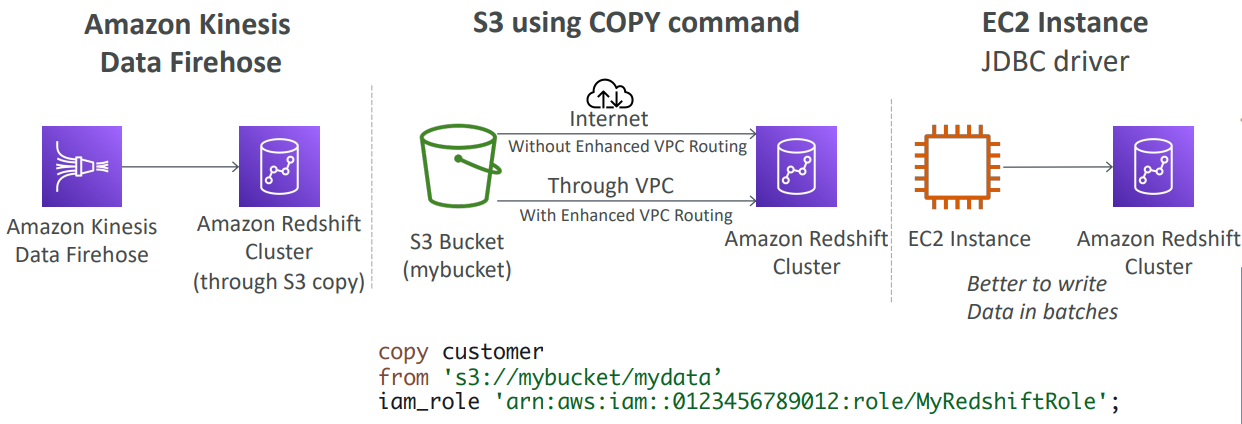
The leader node has access to the compute nodes, and it makes it a whole Redshift cluster. And so, when you send a query, for example, select count from MY\_TABLE GROUP by something. The leader node is going to plan the query, send it to the compute nodes, you're going to execute the query and then send back the results to the leader node.

|  |  |
| --- | --- |
| * Leader node: for query planning, results aggregation * Compute node: for performing the queries, send results to the leader * Provisioned mode: * Choose instance types in advance * Can reserve instances for cost savings |  |

**Redshift – Snapshots & DR**

|  |  |
| --- | --- |
| * Redshift has “Multi-AZ” mode for some clusters * Snapshots are point-in-time backups of a cluster, stored internally in S3 * Snapshots are incremental (only what has changed is saved) * You can restore a snapshot into a new cluster * Automated: every 8 hours, every 5 GB, or on a schedule. Set retention between 1 to 35 days * Manual: snapshot is retained until you delete it * You can configure Amazon Redshift to automatically copy snapshots (automated or manual) of a cluster to another AWS Region |  |

**Loading data into Redshift: Large inserts are MUCH better**



**Redshift Spectrum**

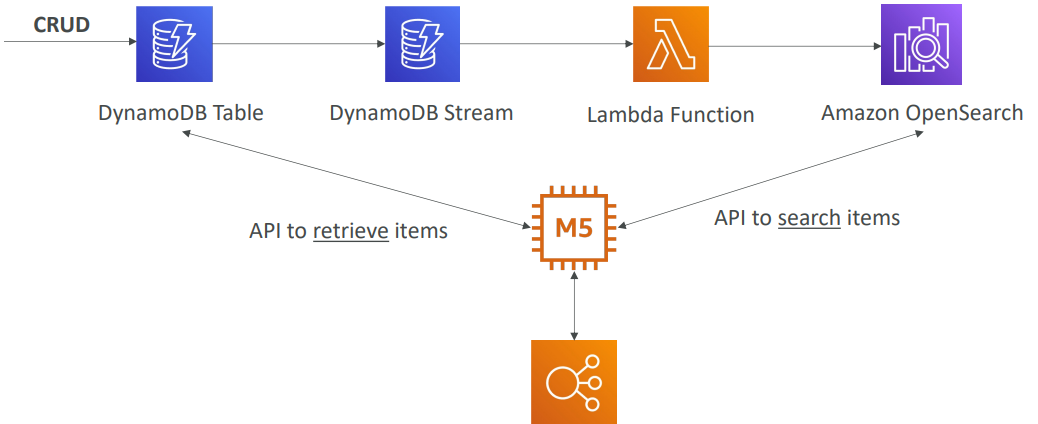
|  |  |
| --- | --- |
| * Query data that is already in S3 without loading it * Must have a Redshift cluster available to start the query * The query is then submitted to thousands of Redshift Spectrum nodes   You have your Redshift cluster with a leader node and a bunch of compute nodes as we've seen, and the data you want to analyze is in Amazon S3. In this case, we're going to run a query onto our cluster, as we can see. What we want to do is that the table we want to query is living in S3. We have from S3 dot and whatever you want. In that case, Spectrum will launch automatically. The query is going to be submitted to thousands of Redshift Spectrum nodes who are going to read the data from Amazon S3 and perform some aggregations. And when they're done, they're going to send back the results into your own Amazon Redshift cluster, and then you will get back into whoever initiated the query. But with this feature, we can leverage a lot more processing power from a shift than the one we have provisioned into our cluster in the first place, loading the data from Amazon S3 into Redshift. |  |

## OpenSearch Service

* Amazon OpenSearch is a successor to Amazon Elasticsearch
* In DynamoDB, queries only exist by primary key or indexes…
* With OpenSearch, you can search any field, even partially matches
* It’s very common to use OpenSearch to provide search to your application. And so, you would use OpenSearch as a compliment to another database. So, OpenSearch can be used for search, but also as the name doesn't indicate, you can also do analytic queries on top of OpenSearch.
* Two modes: managed cluster or serverless cluster
* Does not natively support SQL (can be enabled via a plugin)
* Ingestion from Kinesis Data Firehose, AWS IoT, and CloudWatch Logs
* Security through Cognito & IAM, KMS encryption, TLS
* Comes with OpenSearch Dashboards (visualization)

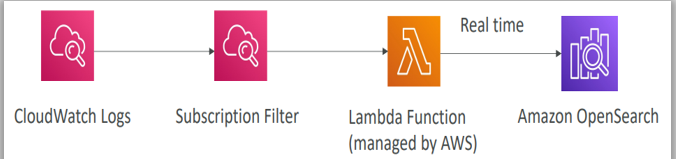
**OpenSearch patterns DynamoDB**

You will have DynamoDB, and it will contain your data. This is where your users will insert and delete and update data. And then you send all the streams to a DynamoDB Stream, which is then picked up by a Lambda Function. And that Lambda Function will insert the data into Amazon OpenSearch in real time. And through this process, what happens is that your application now can search for a specific item. For example, to do a partial search with the item name and then find the item ID out of it. And then once the item ID is obtained, then it will call DynamoDB to retrieve the full item from your DynamoDB Table. So that's a common pattern in which OpenSearch provides search capability, whereas your main source of data still remains your DynamoDB Table.



**OpenSearch patterns CloudWatch Logs**

You can ingest CloudWatch Logs into OpenSearch. So, the first one is to use what's called a CloudWatch Log Subscription Filter, sending data in real time to a Lambda Function that is managed by AWS. And then the Lambda Function in real time sends all the data into Amazon OpenSearch.



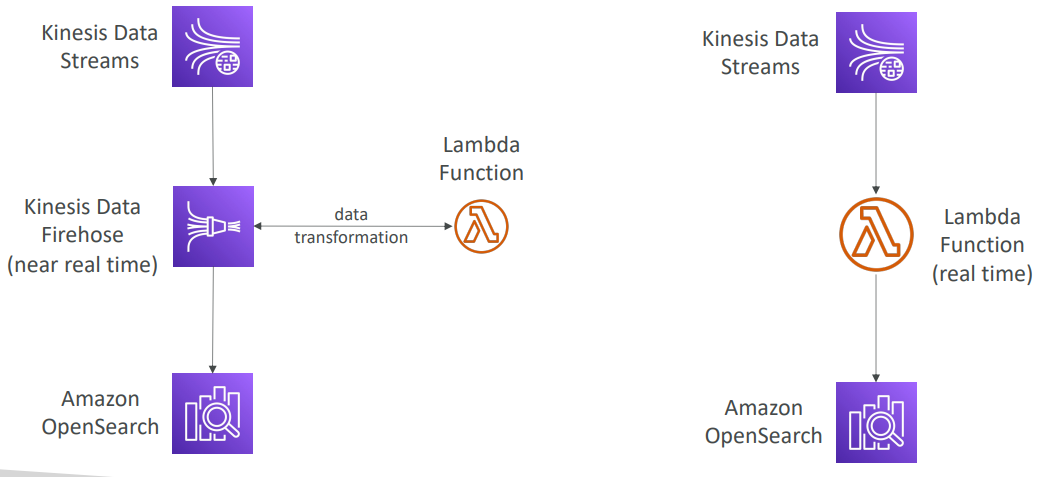
Or you can also use a CloudWatch Logs and then Subscription Filter. But this time Kinesis Data Firehose can read it from the Subscription Filter. And then near real time, because this is Data Firehose, data will be inserted in Amazon OpenSearch.

A diagram of a diagram

AI-generated content may be incorrect.

**OpenSearch patterns Kinesis Data Streams & Kinesis Data Firehose**

To send Kinesis Data Streams into Amazon OpenSearch, you have two strategies. First one is to use Kinesis Data Firehose. It’s a near real time type of service. You can optionally do some data transformation, using a Lambda Function and then send data into Amazon OpenSearch. Or you can use Kinesis Data Streams again, but this time, you would create a Lambda Function that would read the data stream in real time. And then you would write custom code to have the Lambda Function write to Amazon OpenSearch in real time.



## Amazon EMR

* EMR stands for “Elastic MapReduce”
* EMR helps creating Hadoop clusters (Big Data) to analyze and process vast amount of data
* The clusters can be made of hundreds of EC2 instances
* EMR comes bundled with a lot of tools that big data specialist use. For example, Apache Spark, or HBase, or Presto, or Apache Flink.
* They're very difficult to set up, so Amazon EMR will take care of all the provisioning and the configuration of these services for you
* Auto-scaling and integrated with Spot instances
* The use cases of Amazon EMR from an exam perspective are going to be around data processing, doing machine learning, web indexing and big data but all of them using big data related technologies such as Hadoop, Spark, HBase Preso Flink, and so on.

**Amazon EMR – Node types & purchasing**

Amazon EMR is made of clusters of EC2 instances and you have different kind of nodes.

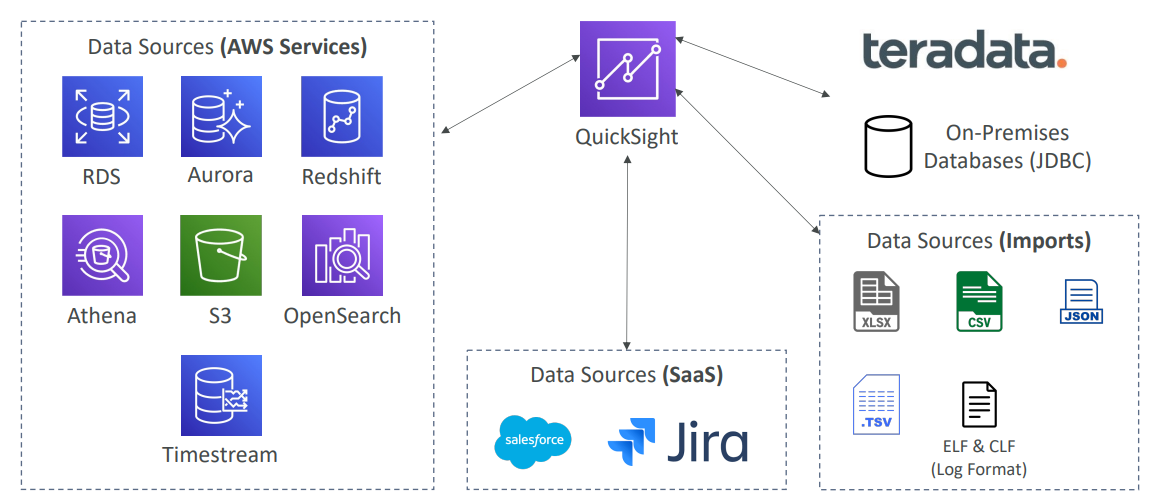
* Master Node: Manage the cluster, coordinate, manage health – long running
* Core Node: Run tasks and store data – long running
* Task Node (optional): Just to run tasks – usually Spot
* Purchasing options:
  + On-demand: reliable, predictable, won’t be terminated
  + Reserved (min 1 year): cost savings (EMR will automatically use if available)
  + Spot Instances: cheaper, can be terminated, less reliable
* Can have long-running cluster, or transient (temporary) cluster

## Amazon Quick Sight

**https://aws.amazon.com/quicksight/**

|  |  |
| --- | --- |
| * Serverless machine learning-powered business intelligence service to create interactive dashboards * Fast, automatically scalable, embeddable, with per-session pricing * Use cases:   + Business analytics   + Building visualizations   + Perform ad-hoc analysis   + Get business insights using data * Integrated with RDS, Aurora, Athena, Redshift, S3… * In-memory computation using SPICE engine if data is imported into Quick Sight and not with external connection * Enterprise edition: Possibility to set up Column-Level security (CLS) i.e. displayed based on access |  |

**Quick Sight Integrations**



**Quick Sight – Dashboard & Analysis**

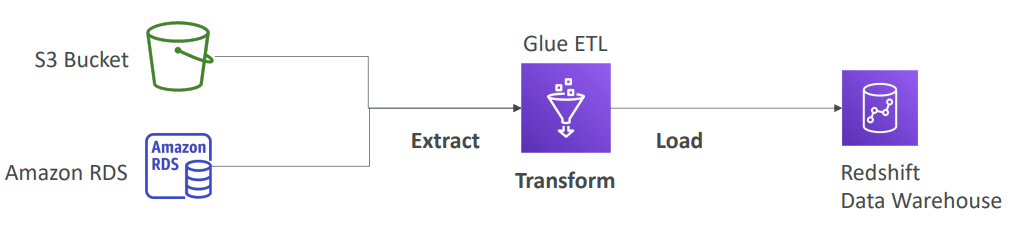
* Define Users (in standard versions) and Groups (available in enterprise version)
* These users & groups only exist within Quick Sight, not IAM !!
* A dashboard…
  + is a read-only snapshot of an analysis that you can share
  + preserves the configuration of the analysis (filtering, parameters, controls, sort)
* You can share the analysis or the dashboard with Users or Groups
* To share a dashboard, you must first publish it
* Users who see the dashboard can also see the underlying data

## AWS Glue

Glue is a managed extract, transform, and load service also called commonly ETL service

* Managed extract, transform, and load (ETL) service
* Useful to prepare and transform data for analytics
* Fully serverless service

Say you had data in an S3 bucket, or an Amazon RDS database and you wanted to load this into a Redshift data warehouse. So, you could extract it using Glue. Then you would transform it, if need be, to maybe filter some data or add some columns and so on, whatever you want. And then you would load the final output data into a Redshift data warehouse. All of this happened from within the Glue ETL Service. You just have to write some code, launch your ETL job and off you go.



**AWS Glue – Convert data into Parquet format**

Another example is how to convert data into Parquet formats. So why would you do this? Because, well, the Parquet format is a columnar data format and, therefore, it is much better when in use, for example, with services like Athena. So, say for example that you are doing inserts into your S3 buckets and then these files are in the CSV formats. Then you would use the Glue ETL Service to import the CSV and convert it into a Parquet format from within the Glue Service. Then you would send it into an output S3 bucket. And when in Parquet format then Amazon Athena is going to analyze this file in a much better fashion.

The other thing you can do to automate this entire process is that anytime a file is inserted into the S3 bucket, then you can send events notifications to a Lambda function which will then trigger a Glue ETL job. But you could replace the Lambda function with Event Bridge as well.



**Glue Data Catalog: catalog of datasets**

There’s another feature of Glue called the Glue Data Catalog, which is to catalog data sets. The Glue Data Catalog will run Glue Data Crawlers, and they will be connected to various data sources such as Amazon S3, Amazon RDS, Amazon DynamoDB or a compatible JDC database that you own on premises for example. The Glue Data Crawler is going to crawl these databases and is going to write all the metadata of your tables, of your columns, of your data types, and so on into the Glue Data Catalog. It will have all the databases, the tables, and the metadata, and that will be leveraged by the Glue jobs to perform ETF. Now, also when you use Amazon Athena behind the scenes to do the data discovery and the SQL discovery, Amazon Athena is going to be leveraging the AWS Glue Data Catalog. So will Amazon Redshift Spectrum. And so will Amazon EMR. As you can see, the Glue Data Catalog service is central to many other AWS services

A diagram of a data catalog

Description automatically generated

**Glue – things to know at a high-level**

* Glue Job Bookmarks: prevent re-processing old data
* Glue Elastic Views:
  + Combine and replicate data across multiple data stores using SQL
  + No custom code, Glue monitors for changes in the source data, serverless
  + Leverages a “virtual table” (materialized view)
* Glue DataBrew: clean and normalize data using pre-built transformation
* Glue Studio: new GUI to create, run and monitor ETL jobs in Glue
* Glue Streaming ETL (built on Apache Spark Structured Streaming): compatible with Kinesis Data Streaming, Kafka, MSK (managed Kafka)

## AWS Lake Formation

* Data lake = central place to have all your data at one place for analytics purposes
* Fully managed service that makes it easy to set up a data lake in days. Usually it takes months but with Lake formation can happen in days
* Discover, cleanse, transform, and ingest data into your Data Lake
* It automates many complex manual steps (collecting, cleansing, moving,cataloging data, …) and de-duplicate (using ML Transforms)
* Combine structured and unstructured data in the data lake
* Out-of-the-box source blueprints: S3, RDS, Relational & NoSQL DB. Which helps to migrate data.
* Fine-grained Access Control for your applications (row and column-level). That means that any application that is connecting to the AWS Lake Formation will have Fine-grain access control,
* Built on top of AWS Glue . It’s a layer on top of AWS Glue but you don't interact with Glue directly

**AWS Lake Formation**

Lake Formation allows you to create a data lake that is stored in Amazon S3. And the data sources are, could be Amazon S3, RDS, Aurora, your on-premises database, such as SQL or NoSQL, anyways, and thanks to the blueprints available on Lake Formation you will ingest the data. Lake Formation comes with Source Crawlers, it comes with ETL and data preparation tools and data cataloging tools, and all of this comes from the underlying Glue service. Then we have security settings and access controls to make sure that your data is protected on your data lake. And in services that can leverage Lake Formation can be Athena, Redshift, EMR, or other analytics tools, such as for example, the Apache Spark framework, and so you as users, you are connecting to these services which are in turn connecting to Lake Formation and your data lake.

A diagram of a data flow

Description automatically generated

**AWS Lake Formation Centralized Permissions Example**

So why do we want to use Lake Formation? Well, one central, key aspect is around the centralized permissions. Say for example, that your company is using Athena and Quick sights to analyze data, and your users must only view the data they need, and they should have permissions to see, and your data sources include Amazon S3, RDS, Aurora, and so on. You could try to set up security in Athena or you could try to set up security in Quick Sight or at the user level, or you can set up security with S3 bucket policies or with users and so on in RDS, or in Aurora. And you have multiple places where you can manage security, and it becomes a mess. What is the right thing, right? Lake Formation solves this problem because you have access control, and you get column and role level security. With Lake Formation, again, you ingest all this data, it sits in a central S3 bucket but from within Lake Formation you manage all the access control for row and column-level security so that any service that then connects to Lake Formation only will have the right to see whatever has the right to see. So, if you use Athena, if you use Quick sights, if you use whatever other tools and you connect to Lake Formation, you manage the security in one place, and that is Lake Formation and that is a big advantage.

A diagram of a computer

Description automatically generated

**Kinesis Data Analytics for SQL applications**

A diagram of data processing

Description automatically generated

**Kinesis Data Analytics (SQL application)**

* Real-time analytics on Kinesis Data Streams & Firehose using SQL
* Add reference data from Amazon S3 to enrich streaming data
* Fully managed, no servers to provision
* Automatic scaling
* Pay for actual consumption rate
* Output:
  + Kinesis Data Streams: create streams out of the real-time analytics queries
  + Kinesis Data Firehose: send analytics query results to destinations
* Use cases:
  + Time-series analytics
  + Real-time dashboards
  + Real-time metrics

## Apache Flink

It used to be called Kinesis Data Analytics for Apache Flink, but since then it has been renamed to just **Managed Service for Apache Flink**

**Kinesis Data Analytics for Apache Flink**

It's a framework used for processing data streams in real time. Use Flink (Java, Scala or SQL) to process and analyze streaming data

A white square with black text

Description automatically generated

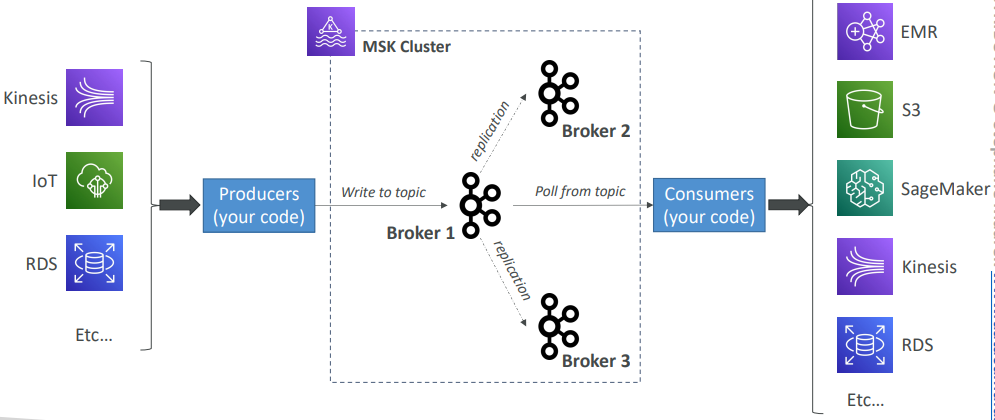
* Run any Apache Flink application on a managed cluster on AWS
  + provisioning compute resources, parallel computation, automatic scaling
  + application backups (implemented as checkpoints and snapshots)
  + Use any Apache Flink programming features
  + Flink does not read from Firehose (use Kinesis Analytics for SQL instead)

## Amazon MSK

**Amazon Managed Streaming for Apache Kafka (Amazon MSK)**

* Alternative to Amazon Kinesis
* Fully managed Apache Kafka on AWS
  + Allow you to create, update, delete clusters
  + MSK creates & manages Kafka brokers nodes & Zookeeper nodes for you
  + Deploy the MSK cluster in your VPC, multi-AZ (up to 3 for HA)
  + Automatic recovery from common Apache Kafka failures
  + Data is stored on EBS volumes for as long as you want
* MSK Serverless
  + Run Apache Kafka on MSK without managing the capacity
  + MSK automatically provisions resources and scales compute & storage

**Apache Kafka at a high level**



**Kinesis Data Streams vs. Amazon MSK**

|  |  |
| --- | --- |
| Kinesis data stream | Amazon MSK |
| * 1 MB message size limit * Data Streams with Shards * Shard Splitting & Merging * TLS In-flight encryption * KMS at-rest encryption | * 1MB default, configure for higher (ex: 10MB) * Kafka Topics with Partitions * Can only add partitions to a topic * PLAINTEXT or TLS In-flight Encryption * KMS at-rest encryption |

**Amazon MSK Consumers**



## Big Data Ingestion Pipeline

* We want the ingestion pipeline to be fully serverless
* We want to collect data in real time
* We want to transform the data
* We want to query the transformed data using SQL
* The reports created using the queries should be in S3
* We want to load that data into a warehouse and create dashboards

**Big Data Ingestion Pipeline**



**Okay, so how do we do this?**

Let's assume that the producers are data are IoT devices. And then there is this cool service in AWS, and it's called IoT Core. IoT Core helps you manage these IoT devices. Now, these devices can send data in real time to IoT Core and IoT Core directly into a Kinesis data stream. So, data stream for Kinesis, remember it allows us to basically pipe big data in real time very fast into this Kinesis service. Now, Kinesis can be talking to Kinesis data Firehose, and Firehose allows us to, for example, every one minute put and offload our data into an Amazon S3 bucket and that will be an ingestion bucket. So here, what we've done here is that we've just had a whole pipeline to get a lot of data from a lot of devices in real time and put it every one minute into an S3 bucket. On top of it, it's possible for us to cleanse or quickly transform the data using an AWS Lambda function that is directly linked to Kinesis Data Firehose.

Okay, so now we have that ingestion bucket, what can we do with it? Well, for example, we can trigger an SQS topic, an SQS Queue, sorry, and it's optional. And maybe the SQS Queue can trigger AWS Lambda function. I say optional because Lambda can be directly triggered by our S3 bucket, but I just wanted to show you the possibility of invoking SQS through that slide. Okay, so Lambda, what we'll do? Well Lambda will trigger an Amazon Athena SQL query. And this Athena query will pull data from the ingestion bucket, and we'll do an SQL query that's all serverless. And the outputs of this serverless query will go into a reporting bucket, maybe again in Amazon S3 B is different bucket. Okay, so from this we have the data it's being reported on, it's being cleansed and analyzed. We can directly visualize it using Quick Sight. So Quick Sight is a way for us to visualize our data into an Amazon S3 bucket, or we can load the data into a proper data warehouse for analytics, such as Amazon Redshift. And so, this Redshift data warehouse can also serve as an endpoint for Quick Sight. But this shows you overall what you can expect in data ingestion pipeline at a high level, including real-time ingestion, transformation, serverless Lambda, and some data warehousing using Redshift and visualization using Quick Sight.

**Big Data Ingestion Pipeline discussion**

* IoT Core allows you to harvest data from IoT devices
* Kinesis is great for real-time data collection
* Firehose helps with data delivery to S3 in near real-time (1 minute)
* Lambda can help Firehose with data transformations
* Amazon S3 can trigger notifications to SQS
* Lambda can subscribe to SQS (we could have connecter S3 to Lambda)
* Athena is a serverless SQL service and results are stored in S3
* The reporting bucket contains analyzed data and can be used by reporting tool such as AWS Quick Sight, Redshift, etc.…

## Questions

**Top of Form**

**You would like to have a database that is efficient at performing analytical queries on large sets of columnar data. You would like to connect to this Data Warehouse using a reporting and dashboard tool such as Amazon QuickSight. Which AWS technology do you recommend?**

* **Amazon RDS**
* **Amazon S3**
* **Amazon Redshift**
* **Amazon Neptune**

Ans C**Bottom of Form**

**Top of Form**

**You have a lot of log files stored in an S3 bucket that you want to perform a quick analysis, if possible Serverless, to filter the logs and find users that attempted to make an unauthorized action. Which AWS service allows you to do so?**

* **Amazon DynamoDB**
* **Amazon Redshift**
* **S3 Glacier**
* **Amazon Athena**

**Bottom of Form**

Ans D

**Top of Form**

**As a Solutions Architect, you have been instructed you to prepare a disaster recovery plan for a Redshift cluster. What should you do?**

* **Enable Multi-AZ**
* **Enable Automated Snapshots, then configure your Redshift cluster to automatically copy snapshots to another AWS region**
* **Take a snapshot then restore to a Redshift Global cluster**

**Bottom of Form**

Ans B

**Top of Form**

**Which feature in Redshift forces all COPY and UNLOAD traffic moving between your cluster and data repositories through your VPCs?**

* **Enhanced VPC Routing**
* **Improved VPC Routing**
* **Redshift Spectrum**

Ans A**Bottom of Form** correct

**Top of Form**

**You are running a gaming website that is using DynamoDB as its data store. Users have been asking for a search feature to find other gamers by name, with partial matches if possible. Which AWS technology do you recommend to implement this feature?**

* **Amazon DynamoDB**
* **Amazon Redshift**
* **Amazon OpenSearch Service**
* **Amazon Neptune**

**Bottom of Form**

**Ans**  C correct

**Top of Form**

**An AWS service allows you to create, run, and monitor ETL (extract, transform, and load) jobs in a few clicks.**

* **AWS Glue**
* **Amazon Redshift**
* **Amazon RDS**
* **Amazon DynamoDB**

**Bottom of Form**

**Ans** A

**Top of Form**

**A company is using AWS to host its public websites and internal applications. Those different websites and applications generate a lot of logs and traces. There is a requirement to centrally store those logs and efficiently search and analyze those logs in real-time for detection of any errors and if there is a threat. Which AWS service can help them efficiently store and analyze logs?**

* **Amazon S3**
* **Amazon OpenSearch service**
* **Amazon ElastiCache**
* **Amazon QLDB**

**Bottom of Form**

Ans B

**Top of Form**

**……………………….. makes it easy and cost-effective for data engineers and analysts to run applications built using open source big data frameworks such as Apache Spark, Hive, or Presto without having to operate or manage clusters.**

* **AWS Lambda**
* **Amazon EMR**
* **Amazon Athena**
* **Amazon OpenSearch Service**

**Bottom of Form**

Ans B

**Top of Form**

**An e-commerce company has all its historical data such as orders, customers, revenues, and sales for the previous years hosted on a Redshift cluster. There is a requirement to generate some dashboards and reports indicating the revenues from the previous years and the total sales, so it will be easy to define the requirements for the next year. The DevOps team is assigned to find an AWS service that can help define those dashboards and have native integration with Redshift. Which AWS service is best suited?**

* **Amazon OpenSearch Service**
* **Amazon Athena**
* **Amazon QuickSight**
* **Amazon EMR**

**Bottom of Form**

Ans C correct

**Top of Form**

**Which AWS Glue feature allows you to save and track the data that has already been processed during a previous run of a Glue ETL job?**

* **Glue Job Bookmarks**
* **Glue Elastic Views**
* **Glue Streaming ETL**
* **Glue DataBrew**

Bottom of Form

Ans A correct

**Top of Form**

**You are a DevOps engineer in a machine learning company which 3 TB of JSON files stored in an S3 bucket. There’s a requirement to do some analytics on those files using Amazon Athena and you have been tasked to find a way to convert those files’ format from JSON to Apache Parquet. Which AWS service is best suited?**

* **S3 Object Versioning**
* **Kinesis Data Streams**
* **Amazon MSK**
* **AWS Glue**

**Bottom of Form**

Ans D Correct

**Top of Form**

**You have an on-premises application that is used together with an on-premises Apache Kafka to receive a stream of clickstream events from multiple websites. You have been tasked to migrate this application as soon as possible without any code changes. You decided to host the application on an EC2 instance. What is the best option you recommend to migrate Apache Kafka?**

* **Kinesis Data Streams**
* **AWS Glue**
* **Amazon MSK**
* **Kinesis Data Analytics**

**Bottom of Form**

Ans C correct

**Top of Form**

**You have data stored in RDS, S3 buckets and you are using AWS Lake Formation as a data lake to collect, move and catalog data so you can do some analytics. You have a lot of big data and ML engineers in the company and you want to control access to part of the data as it might contain sensitive information. What can you use?**

* **Lake Formation Fine-grained Access Control**
* **Amazon Cognito**
* **AWS Shield**
* **S3 Object Lock**

**Bottom of Form**

Ans A correct

**Top of Form**

**Which AWS service is most appropriate when you want to perform real-time analytics on streams of data?**

* **Amazon SQS**
* **Amazon SNS**
* **Amazon Kinesis Data Analytics**
* **Amazon Kinesis Data Firehose**

**Bottom of Form**

Ans C Correct

# Machine Learning

## Amazon Rekognition

* Find objects, people, text, scenes in images and videos using ML
* Facial analysis and facial search to do user verification, counting number of people
* Create a database of “familiar faces” or compare against celebrities
* Use cases:
  + Labeling
  + Content Moderation
  + Text Detection
  + Face Detection and Analysis (gender, age range, emotions…)
  + Face Search and Verification
  + Celebrity Recognition
  + Pathing (ex: for sports game analysis)

<https://aws.amazon.com/rekognition/>

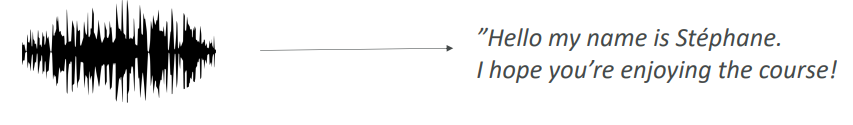
1. We can automate our image and video analysis with machine learning.
2. We can identify the elements of this image. For example, a person, a rock, a mountain bike, a crest and outdoors. We can label, for example, what we see in images, for example, golden retrievers or dogs.
3. Then I can look at content moderation to make sure that it's appropriate for all ages.
4. We can detect text, for example, the number tag identifying the runner in a marathon.
5. We can do face detection analysis, for example, this person looks happy. She's smiling, her eyes are open and she's a female. Face search and verification, if you have a security application.
6. If you want to recognize a celebrity, you can take a picture of them.
7. Also, for example, if you're monitoring a soccer game, you could see where everyone is going to do maybe some real time analytics.
8. Content moderation is used to detect content that is inappropriate, unwanted, or offensive in images and videos. This will be used, for example, if you have a social network or if you broadcast media or if you do advertising or if you're doing e-commerce and you need to create a safe user experience and make sure the image is displayed, don't show any kind of content that would be deemed offensive by some people. For example, racist content or pornography or other kind of things like this.

**Amazon Rekognition – Content Moderation**

|  |  |
| --- | --- |
| * Detect content that is inappropriate, unwanted, or offensive (image and videos) * Used in social media, broadcast media, advertising, and e-commerce situations to create a safer user experience * Set a Minimum Confidence Threshold for items that will be flagged * Flag sensitive content for manual review in Amazon Augmented AI (A2I) * Help comply with regulations |  |

## Amazon Transcribe

* Automatically convert speech to text
* Uses a deep learning process called automatic speech recognition (ASR) to convert speech to text quickly and accurately
* Automatically remove Personally Identifiable Information (PII) using Redaction
* Supports Automatic Language Identification for multi-lingual audio
* Use cases:
  + transcribe customer service calls
  + automate closed captioning and subtitling
  + generate metadata for media assets to create a fully searchable archive



## Amazon Polly

* Turn text into lifelike speech using deep learning
* Allowing you to create applications that talk

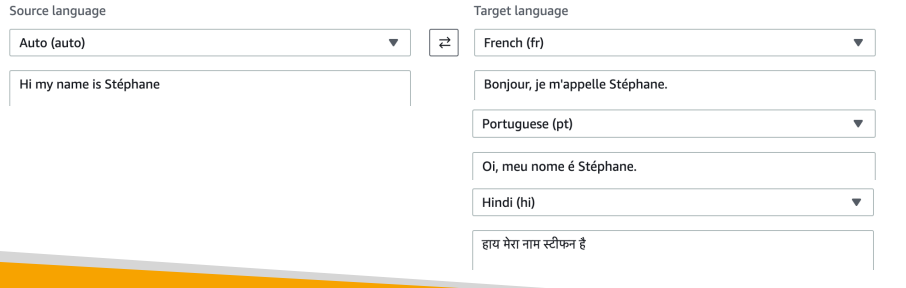


**Amazon Polly – Lexicon & SSML**

* Customize the pronunciation of words with Pronunciation lexicons
  + Stylized words: St3ph4ne => “Stephane”
  + Acronyms: AWS => “Amazon Web Services”
* Upload the lexicons and use them in the Synthesize Speech operation
* Generate speech from plain text or from documents marked up with Speech Synthesis Markup Language (SSML) – enables more customization
  + emphasizing specific words or phrases
  + using phonetic pronunciation
  + including breathing sounds, whispering
  + using the Newscaster speaking style

## Amazon Translate

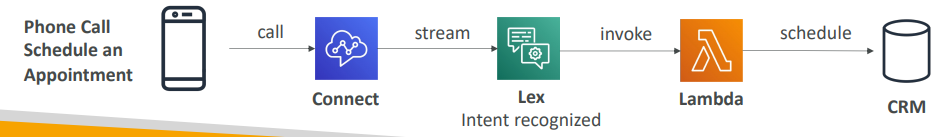
* Natural and accurate language translation
* Amazon Translate allows you to localize content - such as websites and applications - for international users, and to easily translate large volumes of text efficiently



## Amazon Lex & Connect

* Amazon Lex: (same technology that powers Alexa)
  + Automatic Speech Recognition (ASR) to convert speech to text
  + Natural Language Understanding to recognize the intent of text, callers
  + Helps build chatbots, call center bots
* Amazon Connect:
  + Receive calls, create contact flows, cloud-based virtual contact center
  + Can integrate with other CRM systems or AWS
  + No upfront payments, 80% cheaper than traditional contact center solutions

For example, a phone call to schedule an appointment that is made into a number that is defined by Amazon Connect. So they call Amazon Connect. Lex is streaming all the information from this call and understand the intent of the phone call, and therefore, it will invoke the right Lambda function. And that Lambda function, for example, can be very smart and say, "Hey, someone has said to schedule a meeting tomorrow with Tom at 3:00 PM." Okay, I will go into my CRM and schedule that meeting by writing some code.



## Amazon Comprehend

* For Natural Language Processing – NLP
* Fully managed and serverless service
* Uses machine learning to find insights and relationships in text
  + Language of the text
  + Extracts key phrases, places, people, brands, or events
  + Understands how positive or negative the text is
  + Analyzes text using tokenization and parts of speech
  + Automatically organizes a collection of text files by topic

So, Comprehend is really about getting a lot of data in and then Comprehend will do the rest to try to understand the meaning of that data. So it's about taking text or unstructured data and structuring it around these features.

* Sample use cases:
  + analyze customer interactions (emails) to find what leads to a positive or negative experience
  + Create and groups articles by topics that Comprehend will uncover

**Amazon Comprehend Medical**

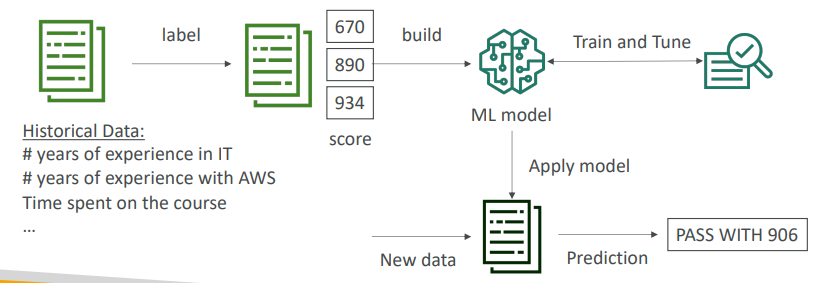
* Amazon Comprehend Medical detects and returns useful information in unstructured clinical text:
  + Physician’s notes
  + Discharge summaries
  + Test results
  + Case notes
* Uses NLP to detect Protected Health Information (PHI) – DetectPHI API
* Store your documents in Amazon S3, analyze real-time data with Kinesis Data Firehose, or use Amazon Transcribe to transcribe patient narratives into text that can be analyzed by Amazon Comprehend Medical.

## Amazon Sage Maker

Sage Maker is a fully managed service for developers and data scientists to build machine learning models. Sage Maker is a higher-level machine learning service where you have your actual developers or your data scientists within your organization create and build machine learning models. It is a lot more involved and a lot more difficult to use. When you want to do this kind of process to build a machine learning model, you must do a bunch of steps, and all these are quite difficult to do in one place, plus you need to provision some servers to perform these competitions to create these models, and that can be cumbersome as well. This is where Sage Maker comes in. Sage Maker will try to help you all along the way with the process.

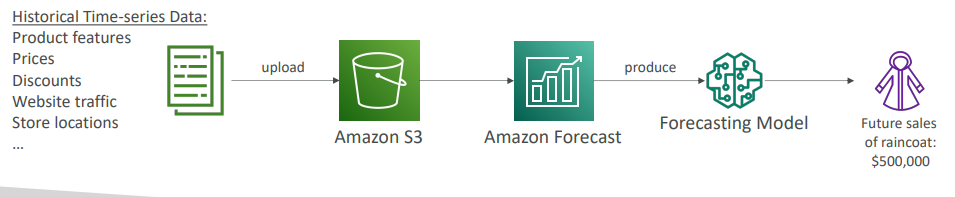
Let's say I wanted to build a model to predict what score you're going to get at your certified CLAP practitioner exam. So how will I do it? Say for example that I am a developer or a data scientist, so I'm going to gather all your data from the actual performance of my students. So, I will ask maybe 10,000 students to give me information about how many years of experience in IT they had, how many years of experience with AWS they had, how much time they spent on the course, how many practice exams they did, etc. etc., so I'll try to gather as much data as possible, and then I'm going to label the data. So that means you need to say which columns correspond to what, and also you need to give some kind of score, and the score is the actual score for me at the exam that someone obtains. For example, someone did not pass 670, maybe he didn't do the course completely, that would be the reason. Maybe someone passed with a high grade, so 990 or maybe someone with an even higher grade 934, so, each student will get a specific score, and my guess is that based on the data that I've collected I can predict what the score will be. So, I've first done the labeling, and that labeling process can be quite complicated to do in practice. Then I need to build a machine learning model, which is how I can predict these scores from historical data. So, then you build that machine learning model and then you must train it and tune it. So, this is another part that's quite difficult to do, which is how to refine my model over time to better fit my data and my outputs. Okay, so now Sage Maker and all of this will help you with the labeling, the building, the training and tuning, but not only. Now we have a machine learning model, and it has been created, it's fully working. So now I need to use it. So, this is called deploying machine learning models. So, we're going to get new data coming in. For example you are the new student, and I'm going to survey you, and I'm going to say okay, how many years of experience do you have in IT, with AWS, how much time have you spent on this course, and then I will apply based on this data you just given me, I will apply the machine learning model that I have created from before. And then the machine learning model will say, for example, hey it sends based on the data you have. I’m going to predict that this student will pass with a score of 906. And this whole process, of labeling, building the model, training it, tuning it, applying it can be all done within Sage Maker

* Fully managed service for developers / data scientists to build ML models
* Typically, it is difficult to do all the processes in one place + provision servers
* Machine learning process (simplified): predicting your exam score



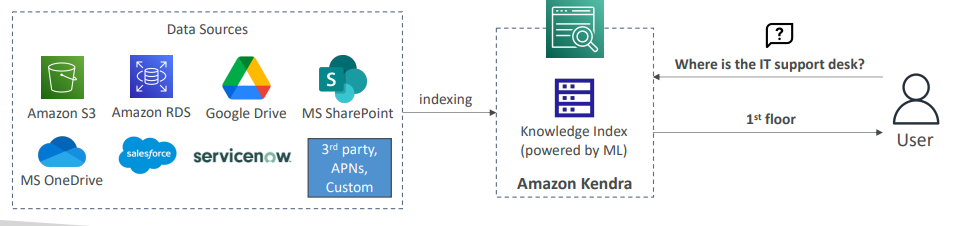
## Amazon Forecast

* Fully managed service that uses ML to deliver highly accurate forecasts
* Example: predict the future sales of a raincoat
* 50% more accurate than looking at the data itself
* Reduce forecasting time from months to hours
* Use cases: Product Demand Planning, Financial Planning, Resource Planning, …



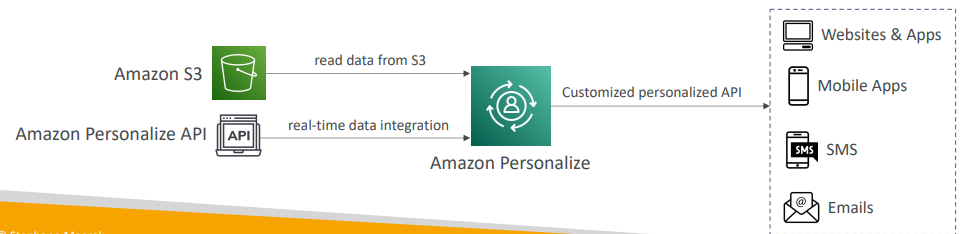
## Amazon Kendra

* Fully managed document search service powered by Machine Learning
* Extract answers from within a document (text, pdf, HTML, PowerPoint, MS Word, FAQs…)
* Natural language search capabilities
* Learn from user interactions/feedback to promote preferred results (Incremental Learning)
* Ability to manually fine-tune search results (importance of data, freshness, custom, …)



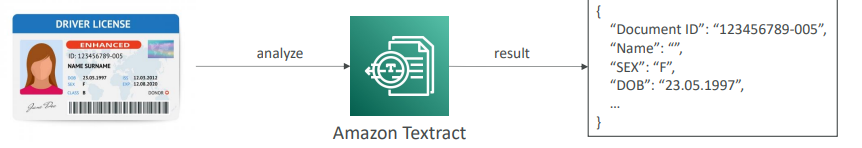
## Amazon Personalize

* Fully managed ML-service to build apps with real-time personalized recommendations
* Example: personalized product recommendations/re-ranking, customized direct marketing
* Example: User bought gardening tools, provide recommendations on the next one to buy
* Same technology used by Amazon.com
* Integrates into existing websites, applications, SMS, email marketing systems, …
* Implement in days, not months (you don’t need to build, train, and deploy ML solutions)
* Use cases: retail stores, media and entertainment…



## Amazon Textract

• Automatically extracts text, handwriting, and data from any scanned documents using AI and ML



• Extract data from forms and tables

• Read and process any type of document (PDFs, images, …)

• Use cases:

• Financial Services (e.g., invoices, financial reports)

• Healthcare (e.g., medical records, insurance claims)

• Public Sector (e.g., tax forms, ID documents, passports)

AWS Machine Learning- Summary

* Rekognition : face detection, labeling, celebrity recognition
* Transcribe: audio to text (ex: subtitles) • Polly: text to audio
* Polly: text to audio
* Translate: translations
* Lex: build conversational bots – chatbots
* Connect: cloud contact center
* Comprehend: natural language processing
* SageMaker: machine learning for every developer and data scientist
* Forecast: build highly accurate forecasts
* Kendra: ML-powered search engine
* Personalize: real-time personalized recommendations
* Textract: detect text and data in documents

## Questions

**Amazon Transcribe is an AWS service that makes it easy for customers to convert speech-to-text. Amazon Polly is a service that turns text into lifelike speech.**

Ans A Amazon Transcribe is an AWS service that makes it easy for customers to convert speech-to-text. Amazon Polly is a service that turns text into lifelike speech.

B Correct Amazon Transcribe is an AWS service that makes it easy for customers to convert speech-to-text. Amazon Polly is a service that turns text into lifelike speech.

**Top of Form**

**A company would like to implement a chatbot that will convert speech-to-text and recognize the customers' intentions. What service should it use?**

* **Transcribe**
* **Rekognition**
* **Connect**
* **Lex**

Bottom of Form

Ans A Amazon Transcribe is an AWS service that makes it easy for customers to convert speech-to-text. However, it does not have natural language understanding.

B Amazon Rekognition makes it easy to add image and video analysis to your applications using proven, highly scalable, deep learning technology that requires no machine learning expertise to use.

C Amazon Connect is a self-service, cloud-based contact center service that makes it easy for any business to deliver better customer service at lower cost. It does not provide speech-to-text conversion or natural language understanding.

D correct Amazon Lex is a service for building conversational interfaces into any application using voice and text. Lex provides the advanced deep learning functionalities of automatic speech recognition (ASR) for converting speech to text, and natural language understanding (NLU) to recognize the intent of the text, to enable you to build applications with highly engaging user experiences and lifelike conversational interactions.

**Top of Form**

**Which fully managed service can deliver highly accurate forecasts?**

* **Personalize**
* **SageMaker**
* **Lex**
* **Forecast**

**Bottom of Form**

Ans A Amazon Personalize is a machine learning service that makes it easy for developers to create individualized recommendations for customers using their applications. It is not used to deliver highly accurate forecasts.

B Amazon SageMaker is a fully managed service that provides every developer and data scientist with the ability to build, train, and deploy machine learning (ML) models quickly. It is not used to deliver highly accurate forecasts.

C Amazon Lex is a service for building conversational interfaces into any application using voice and text. It is not used to deliver highly accurate forecasts.

D correct Amazon Forecast is a fully managed service that uses machine learning to deliver highly accurate forecasts.

**Top of Form**

**You would like to find objects, people, text, or scenes in images and videos. What AWS service should you use?**

* **Rekognition**
* **Polly**
* **Kendra**
* **Lex**

**Bottom of Form**

Ans A correct Amazon Rekognition makes it easy to add image and video analysis to your applications using proven, highly scalable, deep learning technology that requires no machine learning expertise to use.

B Amazon Polly is a service that turns text into lifelike speech. It is not used to find objects, people, text, or scenes in images and videos.

C Amazon Kendra is a highly accurate and easy to use enterprise search service that’s powered by machine learning. It is not used to find objects, people, text, or scenes in images and videos.

D Amazon Lex is a service for building conversational interfaces into any application using voice and text. Lex provides the advanced deep learning functionalities of automatic speech recognition (ASR) for converting speech to text, and natural language understanding (NLU) to recognize the intent of the text, to enable you to build applications with highly engaging user experiences and lifelike conversational interactions. It is not used to find objects, people, text, or scenes in images and videos.

**Top of Form**

**A start-up would like to rapidly create customized user experiences. Which AWS service can help?**

* **Personalize**
* **Kendra**
* **Connect**

**Bottom of Form**

Ans A correct Amazon Personalize is a machine learning service that makes it easy for developers to create individualized recommendations for customers using their applications.

B Amazon Kendra is a highly accurate and easy to use enterprise search service that’s powered by machine learning. It is not used to create fast personalized user experiences and recommendations.

C Amazon Connect is an easy-to-use omnichannel cloud contact center that helps companies provide superior customer service at a lower cost. It is not used to create fast personalized user experiences and recommendations.

**Top of Form**

**A research team would like to group articles by topics using Natural Language Processing (NLP). Which service should they use?**

* **Translate**
* **Comprehend**
* **Lex**
* **RekognitionBottom of Form**

Ans A Amazon Translate is a neural machine translation service that delivers fast, high-quality, and affordable language translation. It does not use natural language processing (NLP).

B Correct Amazon Comprehend is a natural language processing (NLP) service that uses machine learning to find meaning and insights in text.

C Amazon Lex is a service for building conversational interfaces into any application using voice and text. Lex provides the advanced deep learning functionalities of automatic speech recognition (ASR) for converting speech to text, and natural language understanding (NLU) to recognize the intent of the text, to enable you to build applications with highly engaging user experiences and lifelike conversational interactions. It does not use natural language processing (NLP).

D Amazon Rekognition makes it easy to add image and video analysis to your applications using proven, highly scalable, deep learning technology that requires no machine learning expertise to use. It does not use natural language processing (NLP).

**Top of Form**

**A company would like to convert its documents into different languages, with natural and accurate wording. What should they use?**

* **Transcribe**
* **Polly**
* **Translate**
* **WordTranslator**

**Bottom of Form**

Ans A Amazon Transcribe is an AWS service that makes it easy for customers to convert speech-to-text.

B Amazon Polly is a service that turns text into lifelike speech.

C Correct Amazon Translate is a neural machine translation service that delivers fast, high-quality, and affordable language translation.

D This is a distractor. This service does not exist.

**Top of Form**

**A developer would like to build, train, and deploy a machine learning model quickly. Which service can he use?**

* **SageMaker**
* **Polly**
* **Comprehend**
* **Personalize**

**Bottom of Form**

Ans A Correct Amazon SageMaker is a fully managed service that provides every developer and data scientist with the ability to build, train, and deploy machine learning (ML) models quickly. SageMaker removes the heavy lifting from each step of the machine learning process to make it easier to develop high quality models.

B Amazon Polly is a service that turns text into lifelike speech. It is not used to build, train, and deploy machine learning models quickly.

C Amazon Comprehend is a natural language processing (NLP) service that uses machine learning to find meaning and insights in text. It is not used to build, train, and deploy machine learning models quickly.

D Amazon Personalize is a machine learning service that makes it easy for developers to create individualized recommendations for customers using their applications. It is not used to build, train, and deploy machine learning models quickly.

**Top of Form**

**Which AWS service makes it easy to convert speech-to-text?**

* **Connect**
* **Translate**
* **Transcribe**
* **Polly**

**Bottom of Form**

Ans A Amazon Connect is a self-service, cloud-based contact center service that makes it easy for any business to deliver better customer service at lower cost. It is not used to convert speech-to-text.

B Amazon Translate is a neural machine translation service that delivers fast, high-quality, and affordable language translation.

C Correct Amazon Transcribe is an AWS service that makes it easy for customers to convert speech-to-text.

D Amazon Polly is a service that turns text into lifelike speech.

**Top of Form**

**Which of the following services is a document search service powered by machine learning?**

* **Forecast**
* **Kendra**
* **Comprehend**
* **Polly**

**Bottom of Form**

Ans A Amazon Forecast is a fully managed service that uses machine learning to deliver highly accurate forecasts. It is not a document search service.

B Correct Amazon Kendra is a highly accurate and easy to use enterprise search service that’s powered by machine learning.

C Amazon Comprehend is a natural language processing (NLP) service that uses machine learning to find meaning and insights in text.

D Amazon Polly is a service that turns text into lifelike speech. It is not a document search service.

**Top of Form**

**A company is managing an image and video sharing platform which is used by customers around the globe. The platform is running on AWS using an S3 bucket to host both images and videos and using CloudFront as the CDN to deliver content to customers all over the world with low latency. In the last couple of months, a lot of customers have complained that they have started to see inappropriate content on the platform which started to increase in the last week. It will be very expensive and time-consuming to manually approve those images and videos by employees before its published on the platform. There is a requirement to find a solution that can automatically detect inappropriate and offensive images and videos and give you the ability to set a minimum confidence threshold for items that will be flagged and allows for manual review. Which AWS service can fit the requirement?**

* **Amazon Polly**
* **Amazon Translate**
* **Amazon Lex**
* **Amazon Rekognition**

**Bottom of Form**

Ans D

**Top of Form**

**An online medical company that allows you to book an appointment with doctors using through a phone call is using AWS to host their infrastructure. They are using Amazon Connect and Amazon Lex to receive calls and create a workflow, book an appointment, and pay. According to the company’s policy, all calls must be recorded for review. But, there is a requirement to remove any Personally Identifiable Information (PII) from the call before it's saved. What do you recommend to use which helps in removing PII from calls?**

* **Amazon Polly**
* **Amazon Transcribe**
* **Amazon Recognition**
* **Amazon Forecast**

**Bottom of Form**

Ans B Correct

**Top of Form**

**Amazon Polly allows you to turn text into speech. It has two important features. First is ……………….. which allows you to customize the pronunciation of words (e.g., “Amazon EC2” will be “Amazon Elastic Compute Cloud”). The second is ……………….. which allows you to emphasize words, including breathing sounds, whispering, and more.**

* **Speech Synthesis Markup Language (SSML), Pronunciation Lexicons**
* **Pronunciation Lexicons, Security Assertion Markup Language (SAML)**
* **Pronunciation Lexicons, Speech Synthesis Markup Language (SSML)**
* **Security Assertion Markup Language (SAML), Pronunciation Lexicons**

**Bottom of Form**

Ans C correct

**Top of Form**

**A medical company is in the process of implementing a solution to detect, extract, and analyze information from unstructured medical text like doctors’ notes, clinical trial reports, and radiology reports. Those documents are uploaded and stored on S3 buckets. According to the company’s regulations, the solution must be designed and implemented to keep patients’ privacy by identifying Protected Health Information (PHI) so the solution will be eligible with HIPAA. Which AWS service should you use?**

* **Amazon Comprehend Medical**
* **Amazon Rekognition**
* **Amazon Polly**
* **Amazon Translate**

**Bottom of Form**

Ans A Correct

# AWS Monitoring, Audit and Performance

## Amazon CloudWatch Metrics

* CloudWatch provides metrics for every service in AWS
* Metric is a variable to monitor ( e.g. CPU Utilization, Networking…)
* Metrics belong to namespaces
* Dimensions are an attribute of metric. For example, a metric about CPU Utilization can be related to a specific instance id, or a specific environment
* Up to 30 dimensions per metric
* Metrics have time stamps
* Can create CloudWatch dashboards of metrics
* Can create CloudWatch Custom Metrics (for RAM for example)

**CloudWatch Metric Streams**

The other thing you can do with your CloudWatch Metrics is to stream them outside of CloudWatch.

|  |  |
| --- | --- |
| * Continually stream CloudWatch metrics to a destination of your choice, with near-real-time delivery and low latency. * Amazon Kinesis Data Firehose (and then its destinations) * 3rd party service provider: Datadog, Dynatrace, New Relic, Splunk, Sumo Logic… * Option to filter metrics to only stream a subset of them |  |

Your CloudWatch Metrics are going to be streamed near real-time into Kinesis Data Firehose, and you obviously have to set that up for it to work. And from Kinesis Data Firehose you can, for example, send them into an Amazon S3 bucket from which you can use Amazon Athena to analyze your data in Amazon S3, or you can use Amazon Redshift to have data warehousing for your metrics, or Amazon OpenSearch to build dashboards from Amazon OpenSearch, or do analytics there

## CloudWatch Logs

Place to store application logs on AWS

* Log groups: arbitrary name, usually representing an application
* Log stream: instances within application / log files / containers
* Can define log expiration policies (never expire, 1 day to 10 years…)
* CloudWatch Logs can send logs to:
  + Amazon S3 (exports)
  + Kinesis Data Streams
  + Kinesis Data Firehose
  + AWS Lambda
  + OpenSearch
* Logs are encrypted by default
* Can setup KMS-based encryption with your own keys

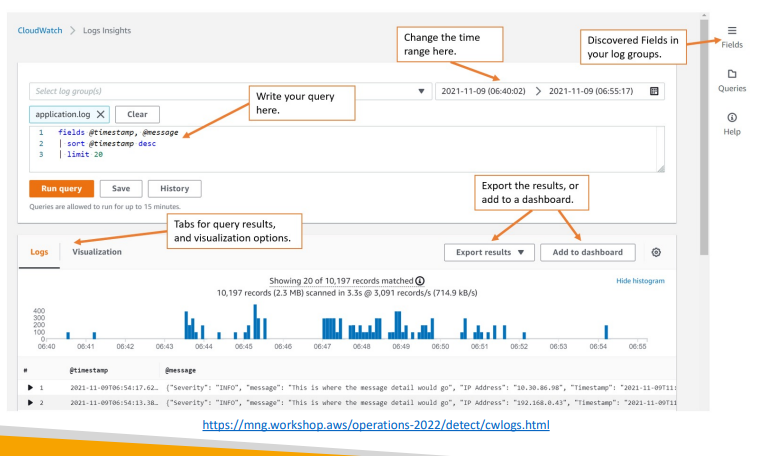
**CloudWatch Logs – Sources**

* SDK, CloudWatch Logs Agent, CloudWatch Unified Agent
* Elastic Beanstalk: collection of logs from application
* ECS: collection from containers
* AWS Lambda: collection from function logs
* VPC Flow Logs: VPC specific logs
* API Gateway
* CloudTrail based on filter
* Route53: Log DNS queries

**CloudWatch Logs Insights**

It's a querying capability within CloudWatch Logs

<https://mng.workshop.aws/operations-2022/detect/cwlogs.html>



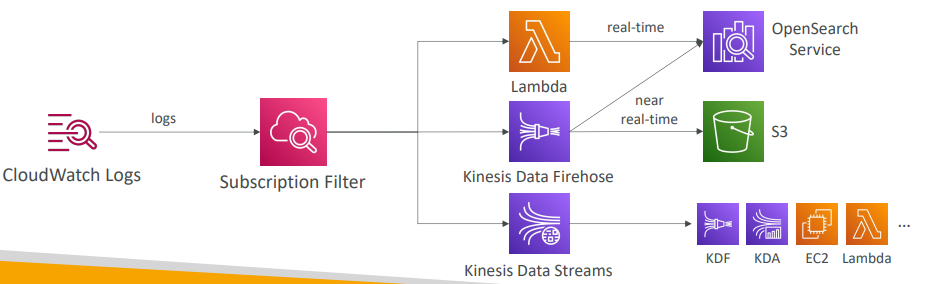
* Write query and specify time stamp to get visualization and also the log lines
* Results can be exported or added to dashboard so thar it can be rerun in future
* There are lots of simple queries provided as part of the console for CloudWatch Logs Insights. For example, you can find the 25 most recent events, or you can have a look at how many events had exceptions or errors in your logs, or you can look for a specific IP and so on.
* It provides a purpose-built query language. All the fields to allow you to build your queries are automatically detected from CloudWatch Logs, and then you can filter based on conditions. You can calculate aggregate statistics, you can sort events, limit the number of events, and so on.
* CloudWatch Logs Insights is a query engine, not a real-time engine. And so as such, it will only query historical data when you run the query.
* Can query multiple Log Groups in different AWS accounts

**CloudWatch Logs – S3 Export**

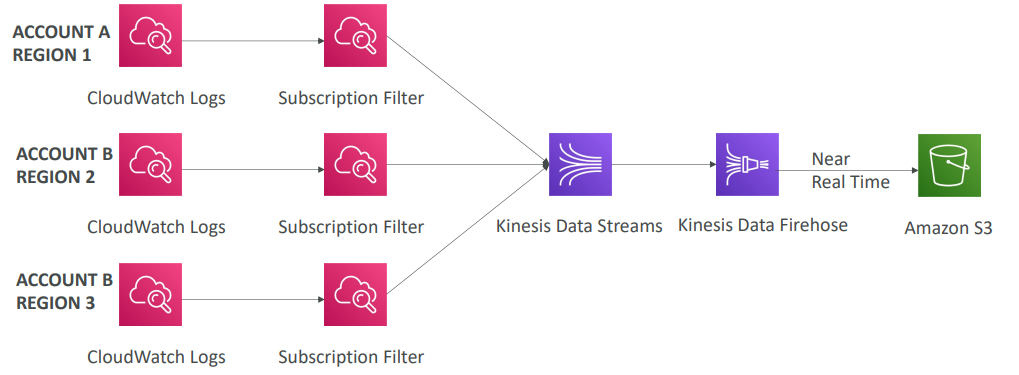
|  |  |
| --- | --- |
| • Log data can take up to 12 hours to become available for export  • The API call is CreateExportTask  • Not near-real time or real-time… use Logs Subscriptions instead |  |

**CloudWatch Logs Subscriptions**

* Get real-time log events from CloudWatch Logs for processing and analysis
* Send to Kinesis Data Streams, Kinesis Data Firehose, or Lambda
* Subscription Filter – filter which logs are events delivered to your destination

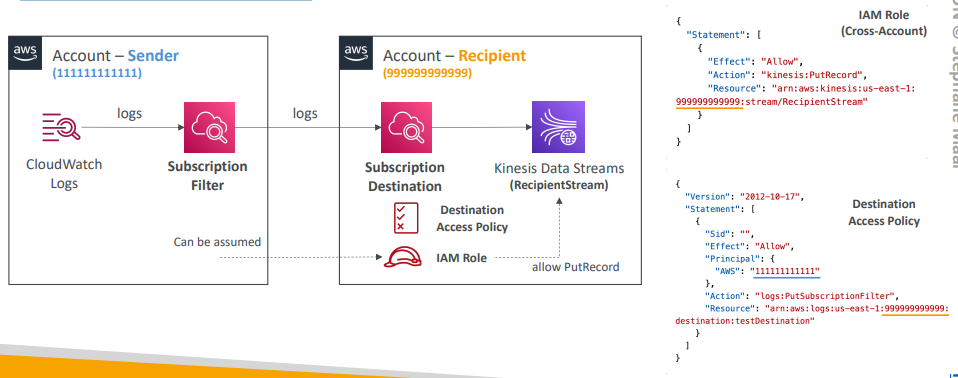


**CloudWatch Logs Aggregation Multi-Account & Multi Region**



**CloudWatch Logs Subscriptions**

**•** Cross-Account Subscription – send log events to resources in a different AWS account (KDS, KDF)



## CloudWatch Agent

**CloudWatch Logs for EC2**

|  |  |
| --- | --- |
| * By default, no logs from your EC2 machine will go to CloudWatch * You need to run a CloudWatch agent on EC2 to push the log files you want * Make sure IAM permissions are correct * The CloudWatch log agent can be setup on-premises too |  |

**CloudWatch Logs Agent & Unified Agent**

* For virtual servers (EC2 instances, on-premises servers…)
* CloudWatch Logs Agent
  + Old version of the agent
  + Can only send to CloudWatch Logs
* CloudWatch Unified Agent newer one
  + Collect additional system-level metrics such as RAM, processes, etc....
  + Collect logs to send to CloudWatch Logs
  + Centralized configuration using SSM Parameter Store

**CloudWatch Unified Agent – Metrics**

* Collected directly on your Linux server / EC2 instance
* CPU (active, guest, idle, system, user, steal)
* Disk metrics (free, used, total), Disk IO (writes, reads, bytes, IOPS)
* RAM (free, inactive, used, total, cached)
* Netstat (number of TCP and UDP connections, net packets, bytes)
* Processes (total, dead, bloqued, idle, running, sleep)
* Swap Space (free, used, used %)
* Reminder: out-of-the box metrics for EC2 – disk, CPU, network (high level)

## CloudWatch Alarms

* Alarms are used to trigger notifications for any metric
* Various options (sampling, %, max, min, etc…)
* Alarm States:
  + OK
  + INSUFFICIENT\_DATA
  + ALARM
* Period:
  + Length of time in seconds to evaluate the metric
  + High resolution custom metrics: 10 sec, 30 sec or multiples of 60 sec

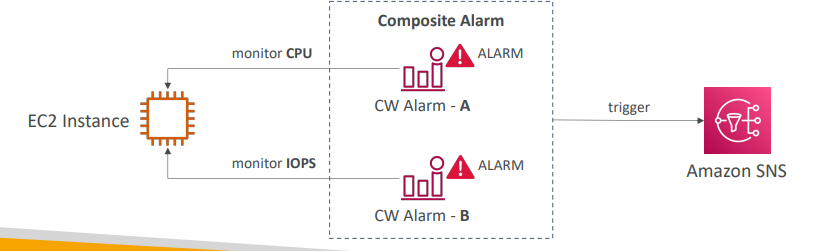
**CloudWatch Alarm Targets**

* Stop, Terminate, Reboot, or Recover an EC2 Instance
* Trigger Auto Scaling Action
* Send notification to SNS (from which you can do pretty much anything)



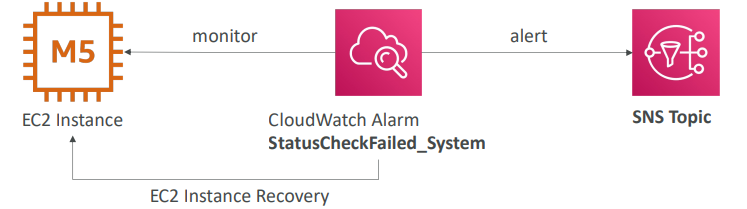
**CloudWatch Alarms – Composite Alarms**

* CloudWatch Alarms are on a single metric
* Composite Alarms monitor the states of multiple other alarms
* AND/OR conditions
* Helpful to reduce “alarm noise” by creating complex composite alarms



**EC2 Instance Recovery**

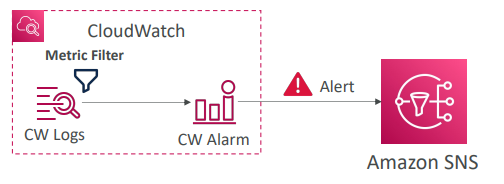
* Status Check:
  + Instance status = check the EC2 VM
  + System status = check the underlying hardware
  + Attached EBS status = check attached EBS volumes



* Recovery: Same Private, Public, Elastic IP, metadata, placement group

**CloudWatch Alarm: good to know**

* Alarms can be created based on CloudWatch Logs Metrics Filters

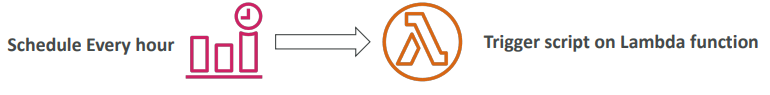


* To test alarms and notifications, set the alarm state to Alarm using CLI aws cloudwatch set-alarm-state --alarm-name "myalarm" --state-value
* ALARM --state-reason "testing purposes"

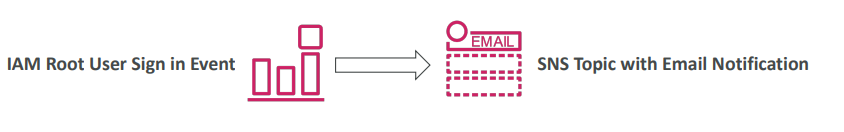
## Amazon Event Bridge

formerly CloudWatch Events. We can do a lot of things with it like

* Schedule: Cron jobs (scheduled scripts)



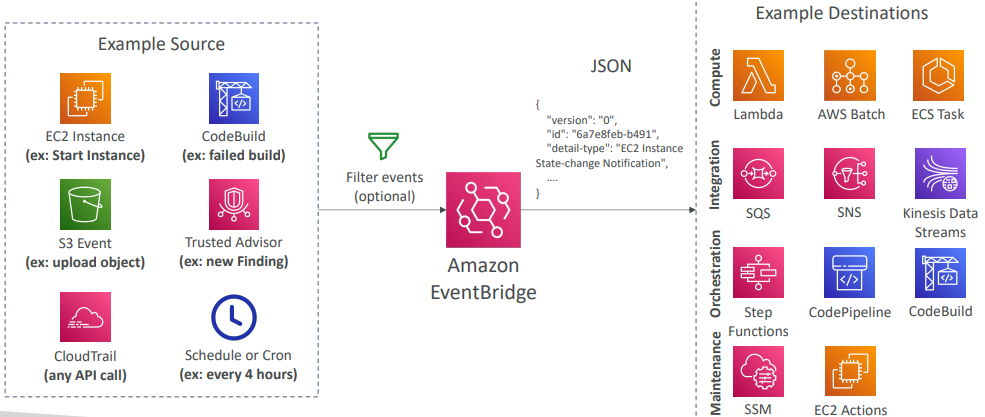
* Event Pattern: Event rules to react to a service doing something



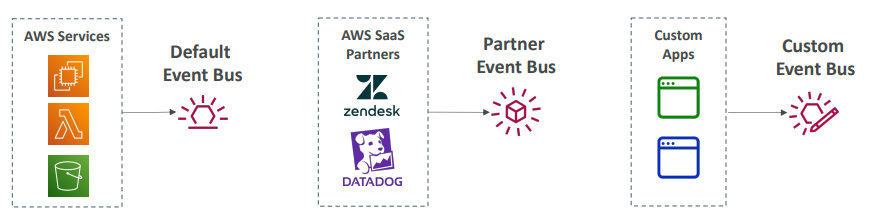
* Trigger Lambda functions, send SQS/SNS messages…

**Amazon Event Bridge Rules**

Event Bridge sits in the middle, and we have all the sources that can send events into Amazon Event Bridge. For example, EC2 instances when they start, when they're stop, when they're terminated, and so on. Code Build, for example, if you have a build that fails or S3, whenever there's an event, for example, when an object is uploaded, or Trusted Advisor when you have a new finding of security in your accounts or as a good combo, you can combine Event Bridge and CloudTrail and actually intercept any API call made within your AWS accounts, which is huge. You can have a schedule or a cron, so you can say every four hours or every Monday at 8:00 am, the first Monday of the month, this is something you can do as well. Then these events get sent into Amazon Event Bridge and you can set up a filter. For example, you say, "Hey, I only want these events for a specific bucket," for Amazon S3, for example. Then Event Bridge is going to generate adjacent document that represent the details about your events. So which instance, for example, gets started, whether its ID, and so on. A lot of information, the time, the IP, and so on. So once this is done, then this JSON document, this event, can be sent to so many kinds of different destinations, allowing you to do awesome integrations. For example, you can schedule and trigger a Lambda function, you can schedule a batch in AWS Batch, you can launch an ECS task for Amazon ECS, you can send a message to SQS, to SNS, or even to a Kinesis Data Stream, you can, for example, start a Step Function, you can start a CI/CD Pipeline with Code Pipeline or a build with Code Build so you don't actually know all these things, of course, these are different AWS services, or you can also, for example, start an SSM automation or a specific EC2 action, such as starting or stopping or restarting an EC2 instance.



Amazon Event Bridge is what we call the default event bus, which is what we just saw, which represents services from AWS that send their events into the default event bus, but Amazon Event Bridge has more capability. There is something called the partner event bus and this is AWS that has integrated with partners, most likely they are going to be software as service partners, and they are going to send their events directly into your partner event bus. If you're using, for example, Zendesk, Datadog, Auth0, or others, you need to check the partner list. Then there's a chance that they can send their events directly into a specified partner event bus and so you can react to changes happening outside of AWS directly in your accounts.



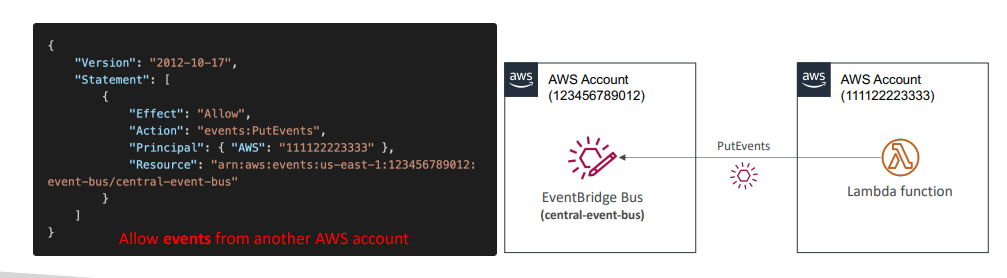
* Event buses can be accessed by other AWS accounts using Resource-based Policies
* You can archive events (all/filter) sent to an event bus (indefinitely or set period)
* Ability to replay archived events

**Amazon Event Bridge – Schema Registry**

|  |  |
| --- | --- |
| Event Bridge is going to analyze events in your bus and then it's going to infer the schema and the schema out of the Schema Registry is going to allow you to generate code for your application that will know in advance how the data is structured in the event bus. For example, this is an example here for a specific Code Pipeline in action. There's a schema and you can download the code directly using the orange button and this will know directly how to infer the schema and structure the data out of your event bus. Also, the schemas can be versioned so you can over time iterate between the schemas of your application, of course. |  |

**Amazon Event Bridge – Resource-based Policy**

* Manage permissions for a specific Event Bus
* Example: allow/deny events from another AWS account or AWS region
* Use case: aggregate all events from your AWS Organization in a single AWS account or AWS region



## CloudWatch Container Insights

|  |  |
| --- | --- |
| * Collect, aggregate, summarize metrics and logs from containers * Available for containers on…   + Amazon Elastic Container Service (Amazon ECS)   + Amazon Elastic Kubernetes Services (Amazon EKS)   + Kubernetes platforms on EC2   + Fargate (both for ECS and EKS) * In Amazon EKS and Kubernetes, CloudWatch Insights is using a containerized version of the CloudWatch Agent to discover containers |  |

**CloudWatch Lambda Insights**

|  |  |
| --- | --- |
| * Monitoring and troubleshooting solution for serverless applications running on AWS Lambda * Collects, aggregates, and summarizes system -level metrics including CPU time, memory, disk, and network * Collects, aggregates, and summarizes diagnostic information such as cold starts and Lambda worker shutdowns * Lambda Insights is provided as a Lambda Layer |  |

**CloudWatch Contributor Insights**

|  |  |
| --- | --- |
| * Analyze log data and create time series that display contributor data.   + See metrics about the top-N contributors   + The total number of unique contributors, and their usage. * This helps you find top talkers and understand who or what is impacting system performance. * Works for any AWS-generated logs (VPC, DNS, etc..) * For example, you can find bad hosts, identify the heaviest network users, or find the URLs that generate the most errors. * You can build your rules from scratch, or you can also use sample rules that AWS has created – leverages your CloudWatch Logs * CloudWatch also provides built-in rules that you can use to analyze metrics from other AWS services. |  |

**CloudWatch Application Insights**

* Provides automated dashboards that show potential problems with monitored applications, to help isolate ongoing issues
* Your applications run on Amazon EC2 Instances with select technologies only (Java, .NET, Microsoft IIS Web Server, databases…)
* And you can use other AWS resources such as Amazon EBS, RDS, ELB, ASG, Lambda, SQS, DynamoDB, S3 bucket, ECS, EKS, SNS, API Gateway…
* Powered by SageMaker
* Enhanced visibility into your application health to reduce the time it will take you to troubleshoot and repair your applications
* Findings and alerts are sent to Amazon EventBridge and SSM OpsCenter

**Summary**

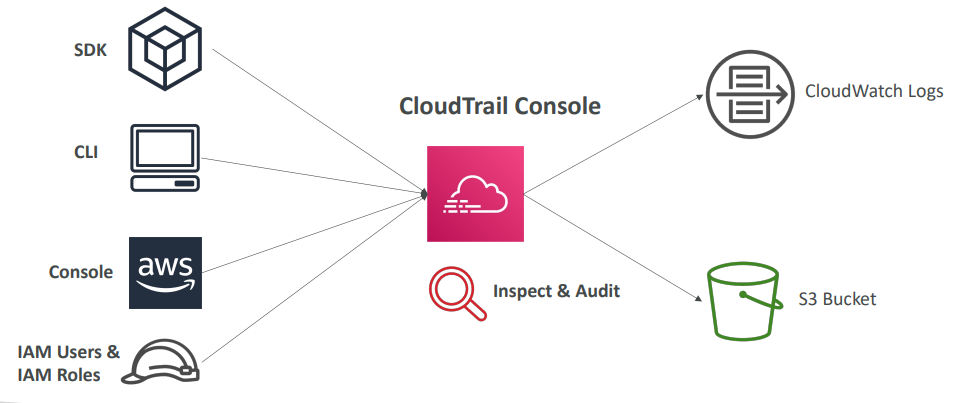
* CloudWatch Container Insights
  + ECS, EKS, Kubernetes on EC2, Fargate, needs agent for Kubernetes
  + Metrics and logs
* CloudWatch Lambda Insights
  + Detailed metrics to troubleshoot serverless applications
* CloudWatch Contributors Insights
  + Find “Top-N” Contributors through CloudWatch Logs
* CloudWatch Application Insights
  + Automatic dashboard to troubleshoot your application and related AWS services

## AWS CloudTrail

* Provides governance, compliance and audit for your AWS Account
* CloudTrail is enabled by default!
* Get an history of events / API calls made within your AWS Account by:
  + Console
  + SDK
  + CLI
  + AWS Services
* Can put logs from CloudTrail into CloudWatch Logs or S3
* A trail can be applied to All Regions (default) or a single Region.
* If a resource is deleted in AWS, investigate CloudTrail first!

E.g. use case, say that an EC2 instance was being terminated and you want to figure out who did it?

**CloudTrail Diagram**



**CloudTrail Events**

3 types of events in cloud trail

* Management Events:
  + Operations that are performed on resources in your AWS account
  + Examples:
    - Configuring security (IAM AttachRolePolicy)
    - Configuring rules for routing data (Amazon EC2 CreateSubnet)
    - Setting up logging (AWS CloudTrail CreateTrail)
  + By default, trails are configured to log management events.
  + Can separate Read Events (that don’t modify resources) from Write Events (that may modify resources)
* Data Events:
  + By default, data events are not logged (because of high volume operations)
  + Amazon S3 object-level activity (ex: GetObject, DeleteObject, PutObject): can separate Read and Write Events
  + AWS Lambda function execution activity (the Invoke API)
* CloudTrail Insights

So when we have so many Management Events across all types of services and so many APIs happening very quickly in your accounts, it can be quite difficult to understand what looks odd, what looks unusual and what doesn't. And so this is where CloudTrail Insights comes in.

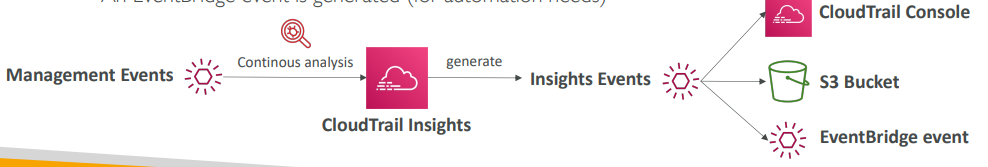
So with CloudTrail Insights and you have to enable it and you have to pay for it, it will analyze your events and try to detect unusual activity in your accounts.

Enable CloudTrail Insights to detect unusual activity in your account like

* + inaccurate resource provisioning
  + hitting service limits
  + Bursts of AWS IAM actions
  + Gaps in periodic maintenance activity

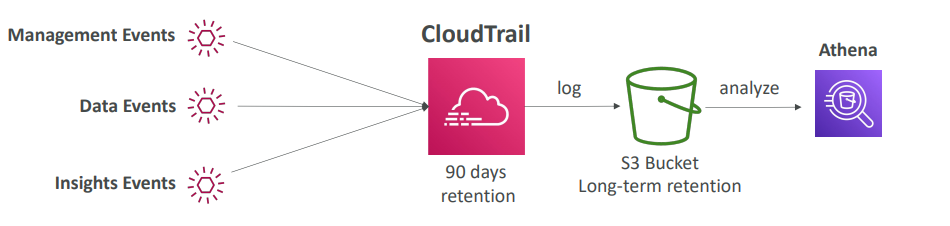
CloudTrail Insights analyzes normal management events to create a baseline And then continuously analyzes write events to detect unusual patterns which we can handle like below

* + Anomalies appear in the CloudTrail console
  + Event is sent to Amazon S3
  + An EventBridge event is generated (for automation needs)



**CloudTrail Events Retention**

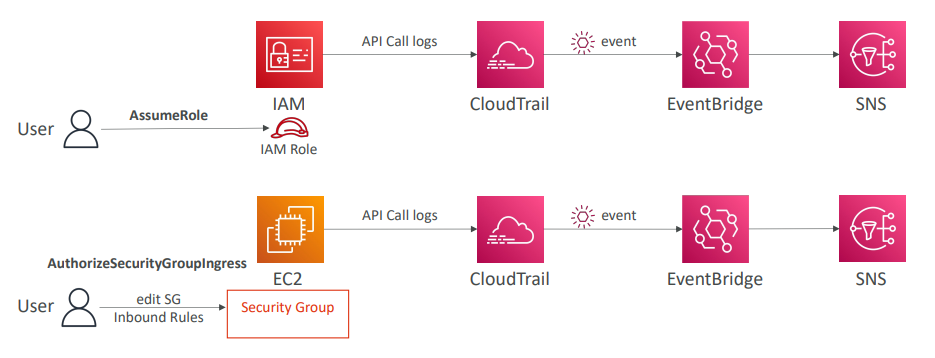
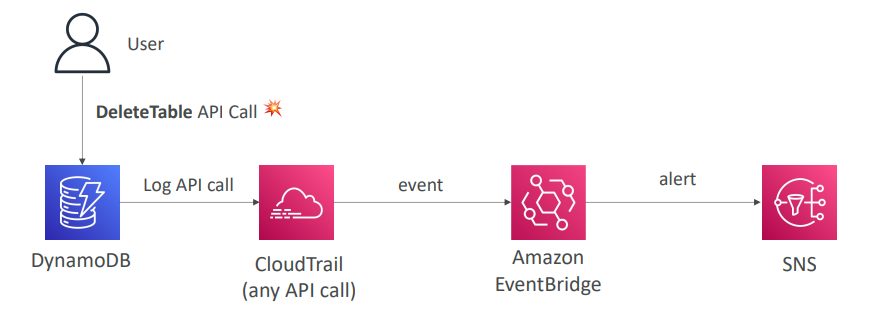
* Events are stored for 90 days on CloudTrail
* To keep events beyond this period, log them to S3 and use Athena



## Amazon Event Bridge – Cloud train integration

Let's say you wanted to receive an SNS notification, anytime a user would delete a table in DynamoDB by using the Delete Table API Call. What happens that whenever we do an API call in AWS, as you know, the API call itself is going to be logged in CloudTrail. That's for any API call. But then all these API calls will end up as events as well in Amazon Event Bridge. So we can look for that very specific delete table API call, and create a rule out of it. And this rule will have a destination, the destination being Amazon SNS and therefore, we can create alerts. Some integration example

For example, say, you wanted to be notified whenever a user was assuming a role in your accounts. So the Assume Role is an API in the IAM service and therefore, is going to be logged by CloudTrail. And then using Event Bridge integration, we can trigger a message into an SNS topic. Similarly, we can also intercept API calls that, for example, change the Security Group inbound rules. So the Security Group call is called AuthorizeSecurityGroupIngress, and it's an EC2 API call. So these are going to be logged again by CloudTrail and then they will appear in EventBridge and then we can trigger a notification in SNS. So as you can see, the possibilities are endless, but now you have a few ideas of how the integration can be leveraged.



## AWS Config

* Helps with auditing and recording compliance of your AWS resources based on the rules set
* Helps record configurations and changes over time to rollback and identify issue
* Questions that can be solved by AWS Config:
  + Is there unrestricted SSH access to my security groups?
  + Do my buckets have any public access?
  + How has my ALB configuration changed over time?
* You can receive alerts (SNS notifications) for any changes
* AWS Config is a per-region service
* Can be aggregated across regions and accounts
* Possibility of storing the configuration data into S3 (analyzed by Athena)

**Config Rules**

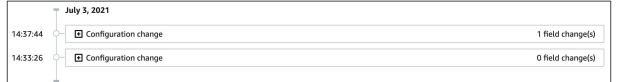
* Can use AWS managed config rules (over 75)
* Can make custom config rules (must be defined in AWS Lambda)
  + Ex: evaluate if each EBS disk is of type gp2
  + Ex: evaluate if each EC2 instance is t2.micro
* Rules can be evaluated / triggered:
  + For each config change
  + And / or: at regular time intervals
* AWS Config Rules does not prevent actions from happening (no deny)
* Pricing: no free tier, $0.003 per configuration item recorded per region, $0.001 per config rule evaluation per region

**AWS Config Resource**

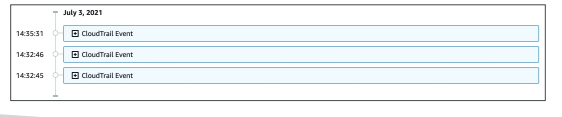
* View compliance of a resource over time



* View configuration of a resource over time

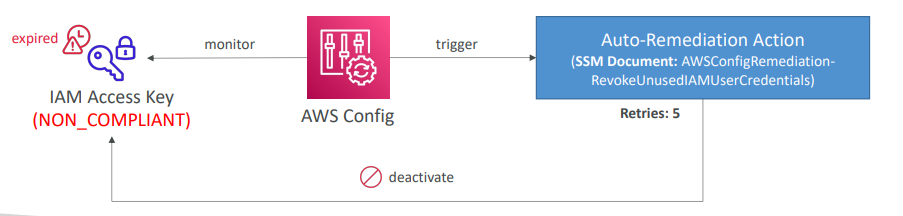


* View CloudTrail API calls of a resource over time



**Config Rules – Remediations**

* Automate remediation of non-compliant resources using SSM Automation Documents
* Use AWS-Managed Automation Documents or create custom Automation Documents
  + Tip: you can create custom Automation Documents that invokes Lambda function
* You can set Remediation Retries if the resource is still non-compliant after auto- remediation

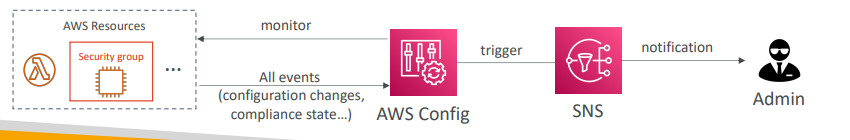


**Config Rules – Notifications**

* Use EventBridge to trigger notifications when AWS resources are noncompliant



* Ability to send configuration changes and compliance state notifications to SNS (all events – use SNS Filtering or filter at client-side)



## CloudWatch vs CloudTrail vs Config

* CloudWatch
  + Performance monitoring (metrics, CPU, network, etc…) & dashboards
  + Events & Alerting
  + Log Aggregation & Analysis
* CloudTrail
  + Record API calls made within your Account by everyone
  + Can define trails for specific resources
  + Global Service
* Config
  + Record configuration changes
  + Evaluate resources against compliance rules
  + Get timeline of changes and compliance

**For an Elastic Load Balancer how can we use each service**

* CloudWatch:
  + Monitoring Incoming connections metric
  + Visualize error codes as % over time
  + Make a dashboard to get an idea of your load balancer performance
* Config:
  + Track security group rules for the Load Balancer
  + Track configuration changes for the Load Balancer
  + Ensure an SSL certificate is always assigned to the Load Balancer (compliance)
* CloudTrail:
  + Track who made any changes to the Load Balancer with API calls

## Questions

**Top of Form**

**You have an RDS DB instance that's configured to push its database logs to CloudWatch. You want to create a CloudWatch alarm if there's an Error found in the logs. How would you do that?**

* **Create a scheduled CloudWatch Event that triggers an AWS Lambda every 1 hour, scans the logs, and notify you through SNS topic**
* **Create a CloudWatch Logs Metric Filter that filter the logs for the keyword Error, then create a CloudWatch Alarm based on that Metric Filter**
* **Create an AWS Config Rule that monitors Error in your database logs and notify you through SNS topic**

**Bottom of Form**

Ans B correct

**Top of Form**

**You have an application hosted on a fleet of EC2 instances managed by an Auto Scaling Group that you configured its minimum capacity to 2. Also, you have created a CloudWatch Alarm that is configured to scale in your ASG when CPU Utilization is below 60%. Currently, your application runs on 2 EC2 instances and has low traffic and the CloudWatch Alarm is in the ALARM state. What will happen?**

* **One EC2 instance will be terminated and the ASG desired and minimum capacity will go to 1**
* **The CloudWatch Alarm will remain in ALARM state but never decrease the number of EC2 instances in the ASG**
* **The CloudWatch Alarm will be detached from my ASG**
* **The CloudWatch Alarm will go in OK state**

**Bottom of Form**

Ans B Correct The number of EC2 instances in an ASG can not go below the minimum capacity, even if the CloudWatch alarm would in theory trigger an EC2 instance termination.

**Top of Form**

**How would you monitor your EC2 instance memory usage in CloudWatch?**

* **Enable EC2 Detailed Monitoring**
* **By default, the EC2 instance pushes memory usage to CloudWatch**
* **Use the Unified CloudWatch Agent to push memory usage as a custom metric to CloudWatch**

**Bottom of Form**

Ans C

**Top of Form**

**You have made a configuration change and would like to evaluate the impact of it on the performance of your application. Which AWS service should you use?**

* **Amazon CloudWatch**
* **AWS CloudTrail**

**Bottom of Form**

Ans A Correct Amazon CloudWatch is a monitoring service that allows you to monitor your applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health. It is used to monitor your applications' performance and metrics.

B AWS CloudTrail allows you to log, continuously monitor, and retain account activity related to actions across your AWS infrastructure. It provides the event history of your AWS account activity, audit API calls made through the AWS Management Console, AWS SDKs, AWS CLI. You can use CloudTrail to detect unusual activity in your AWS accounts.

**Top of Form**

**Someone has terminated an EC2 instance in your AWS account last week, which was hosting a critical database that contains sensitive data. Which AWS service helps you find who did that and when?**

* **CloudWatch Metrics**
* **CloudWatch Alarms**
* **CloudWatch Events**
* **AWS CloudTrail**

**Bottom of Form**

Ans A CloudWatch Metrics are data about the performance of your systems, it won't indicate "who" did what.

B CloudWatch Alarms allows you to trigger notifications/actions based on any CloudWatch Metrics or other CloudWatch Alarms, it won't indicate "who" did what.

C CloudWatch Events delivers a stream of system events that describe changes in your AWS resources, it won't indicate "who" did what.

D AWS CloudTrail allows you to log, continuously monitor, and retain account activity related to actions across your AWS infrastructure. It provides the event history of your AWS account activity, audit API calls made through the AWS Management Console, AWS SDKs, AWS CLI. So, the EC2 instance termination API call will appear here. You can use CloudTrail to detect unusual activity in your AWS accounts.

**Top of Form**

**You have CloudTrail enabled for your AWS Account in all AWS Regions. What should you use to detect unusual activity in your AWS Account?**

* **CloudTrail Data Events**
* **CloudTrail Insights**
* **CloudTrail Management Events**

**Bottom of Form**

Ans B Correct

**Top of Form**

**One of your teammates terminated an EC2 instance 4 months ago which has critical data. You don't know who made this so you are going to review all API calls within this period using CloudTrail. You already have CloudTrail set up and configured to send logs to the S3 bucket. What should you do to find out who made this?**

* **Use CloudTrail Event History in CloudTrail Console**
* **Analyze CloudTrail logs in S3 bucket using Amazon Athena**

**Bottom of Form**

Ans B Correct You can use the CloudTrail Console to view the last 90 days of recorded API activity. For events older than 90 days, use Athena to analyze CloudTrail logs stored in S3.

**Top of Form**

**You are running a website on a fleet of EC2 instances with OS that has a known vulnerability on port 84. You want to continuously monitor your EC2 instances if they have port 84 exposed. How should you do this?**

* **Setup CloudWatch Metrics**
* **Setup CloudTrail Trails**
* **Setup Config Rules**
* **Schedule a CloudWatch Event to trigger a Lambda function to scan your EC2 instances**

**Bottom of Form**

Ans C Correct

**Top of Form**

**You would like to evaluate the compliance of your resource's configurations over time. Which AWS service will you choose?**

* **AWS Config**
* **Amazon CloudWatch**
* **AWS CloudTrail**

**Bottom of Form**

Ans A Correct

**Top of Form**

**Someone changed the configuration of a resource and made it non-compliant. Which AWS service is responsible for logging who made modifications to resources?**

* **Amazon CloudWatch**
* **AWS CloudTrail**
* **AWS Config**

**Bottom of Form**

Ans B Correct

**Top of Form**

**You have enabled AWS Config to monitor Security Groups if there's unrestricted SSH access to any of your EC2 instances. Which AWS Config feature can you use to automatically re-configure your Security Groups to their correct state?**

* **AWS Config Remediations**
* **AWS Config Rules**
* **AWS Config Notifications**

**Bottom of Form**

Ans A Correct

**Top of Form**

**You are running a critical website on a set of EC2 instances with a tightened Security Group that has restricted SSH access. You have enabled AWS Config in your AWS Region and you want to be notified via email when someone modified your EC2 instances' Security Group. Which AWS Config feature helps you do this?**

* **AWS Config Remediations**
* **AWS Config Rules**
* **AWS Config Notifications**

**Bottom of Form**

Ans C Correct

**Top of Form**

**…………………………. is a CloudWatch feature that allows you to send CloudWatch metrics in near real-time to S3 bucket (through Kinesis Data Firehose) and 3rd party destinations (e.g., Splunk, Datadog, …).**

* **CloudWatch Metric Stream**
* **CloudWatch Log Stream**
* **CloudWatch Metric Filter**
* **CloudWatch Log Group**

**Bottom of Form**

Ans A Correct

**Top of Form**

**A DevOps engineer is working for a company and managing its infrastructure and resources on AWS. There was a sudden spike in traffic for the main application for the company which was not normal in this period of the year. The application is hosted on a couple of EC2 instances in private subnets and is fronted by an Application Load Balancer in a public subnet. To detect if this is normal traffic or an attack, the DevOps engineer enabled the VPC Flow Logs for the subnets and stored those logs in CloudWatch Log Group. The DevOps wants to analyze those logs and find out the top IP addresses making requests against the website to check if there is an attack. Which of the following can help the DevOps engineer to analyze those logs?**

* **CloudWatch Metric Stream**
* **CloudWatch Alarm**
* **CloudWatch Contributor Insights**
* **CloudWatch Metric Filter**

**Bottom of Form**

Ans C Correct

**Top of Form**

**A company is developing a Serverless application on AWS using Lambda, DynamoDB, and Cognito. A junior developer joined a few weeks ago and accidentally deleted one of the DynamoDB tables in the dev AWS account which contained important data. The CTO asks you to prevent this from happening again and there must be a notification system to monitor if there is an attempt to make such deletion actions for the DynamoDB tables. What would you do?**

* **Assign developers to a certain IAM group which prevents deletion of DynamoDB tables. Configure EventBridge to capture any DeleteTable API calls through S3 and send a notification using KMS**
* **Assign developers to a certain IAM group which prevents deletion of DynamoDB tables. Configure EventBridge to capture any DeleteTable API calls through CloudTrail and send a notification using SNS**
* **Assign developers to a certain IAM group which prevents deletion of DynamoDB tables. Configure EventBridge to capture any DeleteTable API calls through CloudTrail and send a notification using KMS**

**Bottom of Form**

Ans B Correct

**Top of Form**

**A company has a running Serverless application on AWS which uses EventBridge as an inter-communication channel between different services within the application. There is a requirement to use the events in the prod environment in the dev environment to make some tests. The tests will be done every 6 months, so the events need to be stored and used later on. What is the most efficient and cost-effective way to store EventBridge events and use them later?**

* **Use EventBridge Archive and Replay feature**
* **Create a Lambda function to store the EventBridge events in an S3 bucket for later usage**
* **Configure EventBridge to store events in a DynamoDB table**

**Bottom of Form**

Ans A correct