SAA-7

Example: Building a Serverless API

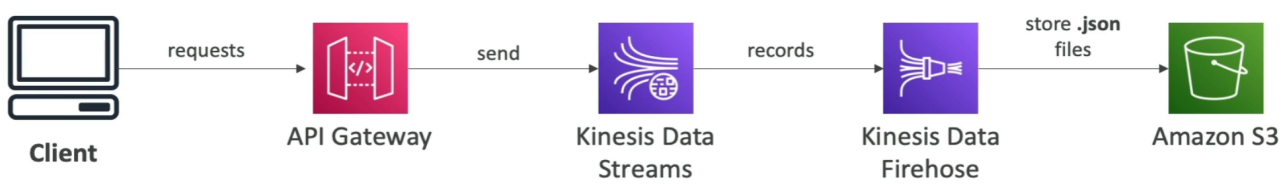
AWS API Gateway

- 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? 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? Add authentication, deploy publicly, rate control ...

API Gateway - AWS Service Integration Kinesis Data Streams example



API Gateway - Endpoint Types

- **Edge - Optimized(default):** For global clients
  - Requests are routed through the CloudFront Edge locations (improves latency)
  - The API Gateway still lives in only one region
- **Regional:**
  - For clients within the same region
  - Could manually combine with CloudFront (more control over the caching strategies and the distribution)
- **Private:**
  - Can only be accessed from your VPC using an interface VPC endpoint (ENI)
  - Use a resource policy to define access

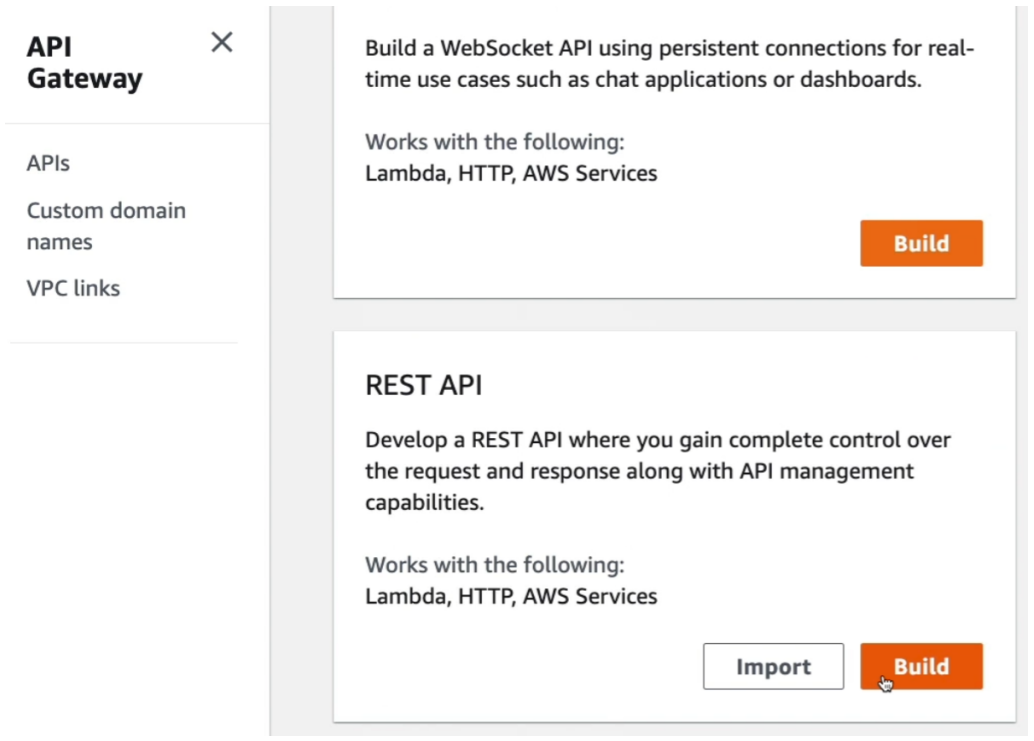
API Gateway - Security

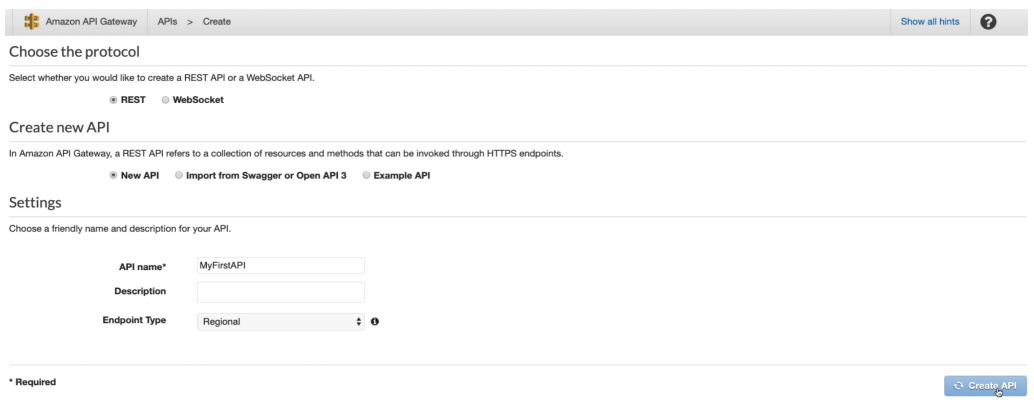
- **User Authentication through**
  - IAM Roles (useful for internal applications)

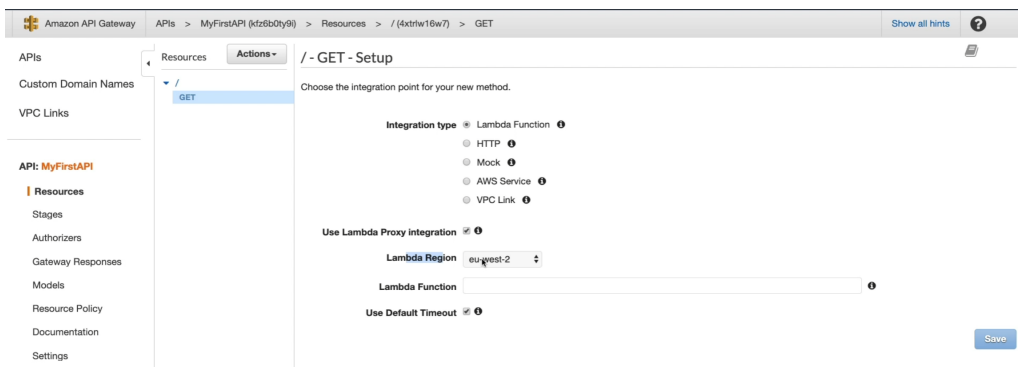
    - Cognito (identity for external users - example mobile users)
    - Custom Authorizer (your own logic)

- **Custom Domain Name HTTPS** security through integration with **AWS Certificate Manager (ACM)**
    - 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 setup CNAME or A-alias record in Route 53

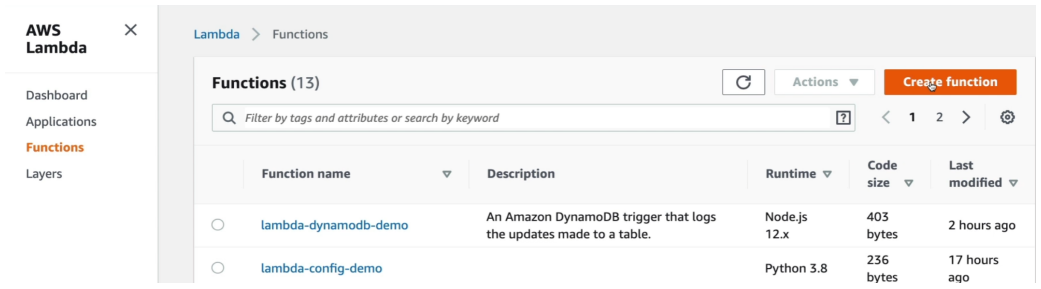
Hands On

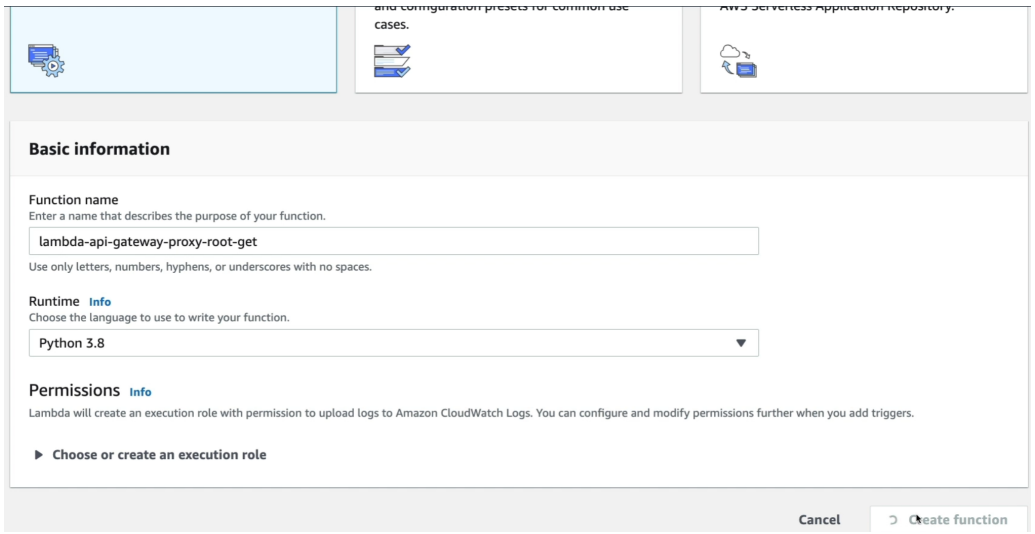


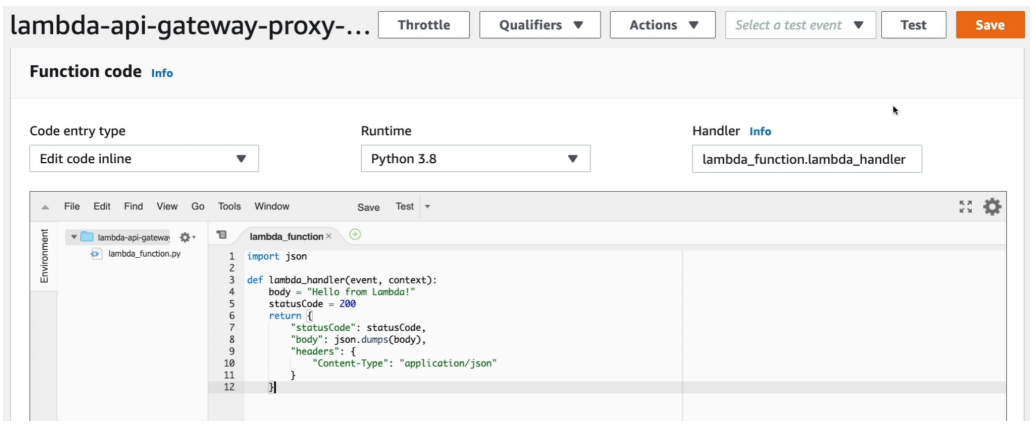


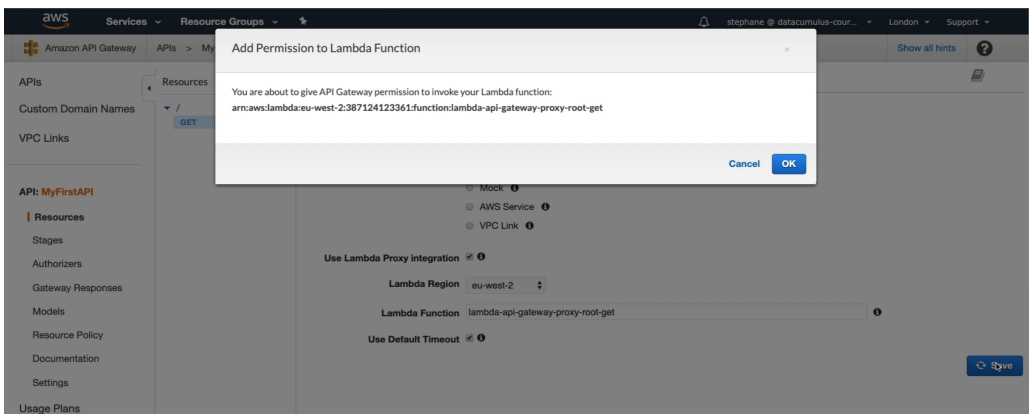


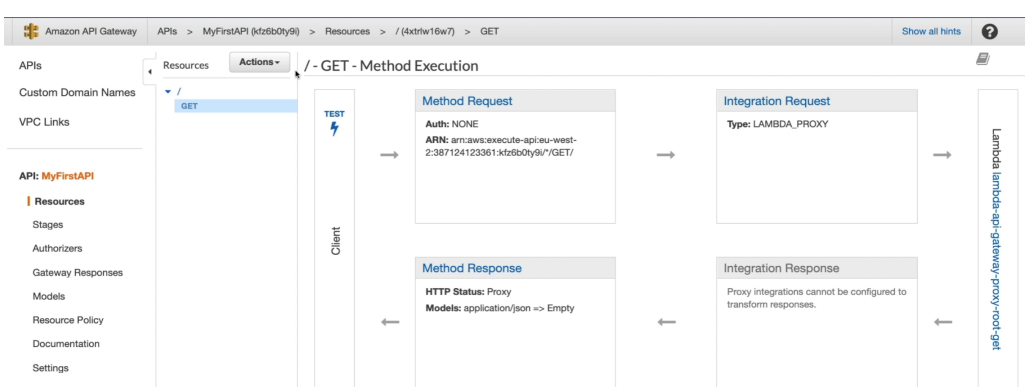
Create Lambda Function

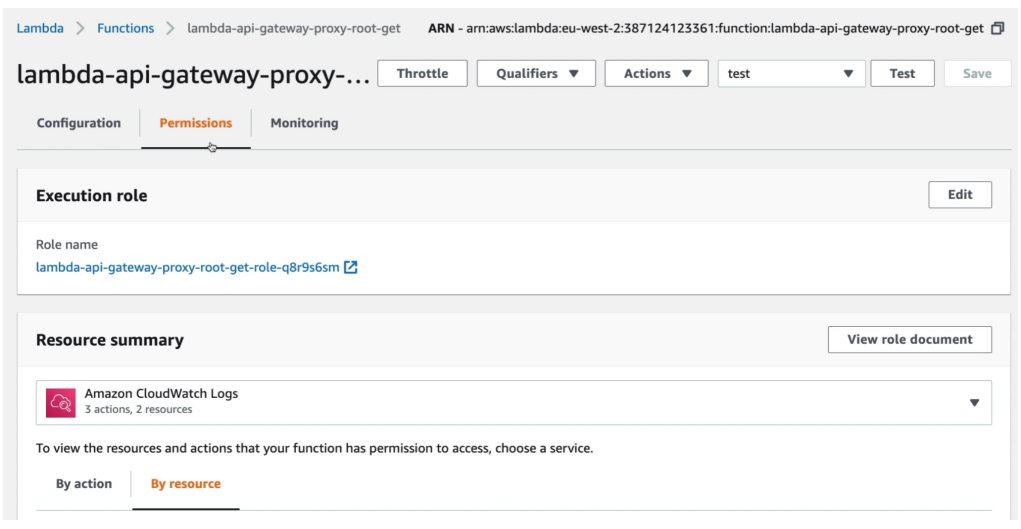



cases.
Basic information
Function name
Enter a name that describes the purpose of your function.
lambda-api-gateway-proxy-root-get
Use only letters, numbers, hyphens, or underscores with no spaces.
Runtime Info
Choose the language to use to write your function.
Python 3.8
Permissions Info
Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further when you add triggers.
Choose or create an execution role
Cancel
Create function


lambda-api-gateway-proxy-...
Throttle
Qualifiers
Actions
Select a test event
Test
Save
Function code Info
Code entry type
Edit code inline
Runtime
Python 3.8
Handler Info
lambda_function.lambda_handler
File Edit Find View Go Tools Window
Save
Test
Environment
lambda-api-gateway
lambda_function.py
lambda_function
import json
def lambda_handler(event, context):
body = "Hello from Lambda!"
statusCode = 200
return {
"statusCode": statusCode,
"body": json.dumps(body),
"headers": {
"Content-Type": "application/json"
}
}


Add Permission to Lambda Function
You are about to give API Gateway permission to invoke your Lambda function:
arn:aws:lambda:eu-west-2:387124123361:function:lambda-api-gateway-proxy-root-get
Cancel
OK
Amazon API Gateway
APIs
Resources
Custom Domain Names
VPC Links
API: MyFirstAPI
Stages
Authorizers
Gateway Responses
Models
Resource Policy
Documentation
Settings
Usage Plans
Show all hints
Mock
AWS Service
VPC Link
Use Lambda Proxy integration
Lambda Region
eu-west-2
Lambda Function
lambda-api-gateway-proxy-root-get
Use Default Timeout


Amazon API Gateway
APIs > MyFirstAPI (kfz6b0ty9i) > Resources > / (4xtrlw16w7) > GET
Show all hints
Actions
/ - GET - Method Execution
TEST
Client
Method Request
Auth: NONE
ARN: arn:aws:execute-api:eu-west-2:387124123361:kfz6b0ty9i/*/GET/
Integration Request
Type: LAMBDA_PROXY
Lambda lambda-api-gateway-proxy-root-get
Method Response
HTTP Status: Proxy
Models: application/json => Empty
Integration Response
Proxy integrations cannot be configured to transform responses.


Lambda > Functions > lambda-api-gateway-proxy-root-get
ARN - arn:aws:lambda:eu-west-2:387124123361:function:lambda-api-gateway-proxy-root-get
lambda-api-gateway-proxy-...
Throttle
Qualifiers
Actions
test
Test
Save
Configuration
Permissions
Monitoring
Execution role
Edit
Role name
lambda-api-gateway-proxy-root-get-role-q8r9s6sm
Resource summary
View role document
Amazon CloudWatch Logs
3 actions, 2 resources
To view the resources and actions that your function has permission to access, choose a service.
By action
By resource

Resource-based policy Info

```
{
  "Version": "2012-10-17",
  "Id": "default",
  "Statement": [
    {
      "Sid": "1fdca940-9cec-4e7c-89d5-9b0c35438775",
      "Effect": "Allow",
      "Principal": {
        "Service": "apigateway.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:eu-west-2:387124123361:function:lambda-api-gateway-proxy-root-get",
      "Condition": {
        "ArnLike": {
          "AWS:SourceArn": "arn:aws:execute-api:eu-west-2:387124123361:kfz6b0ty9i/*/GET/"
        }
      }
    }
  ]
}
```

Test

Resources | Actions

/
GET

← Method Execution / - GET - Method Test

Make a test call to your method with the provided input

Path

No path parameters exist for this resource. You can define path parameters by using the syntax **{myPathParam}** in a resource path.

Query Strings

No query string parameters exist for this method. You can add them via Method Request.

Headers

No header parameters exist for this method. You can add them via Method Request.

Stage Variables

No stage variables exist for this method.

Client Certificate

No client certificates have been generated.

Request Body

Request Body is not supported for GET methods.

Test

Request: /
Status: 200
Latency: 59 ms
Response Body

```
"Hello from Lambda!"
```

Response Headers

```
{"X-Amzn-Trace-Id":"Root=1-5eb52ea9-f894c411e4232d620f6ad9f2;Sampled=0","Content-Type":"application/json"}
```

Logs

```
Execution log for request 7cdd17f7-2b04-4c20-9e70-283984eb5d34
Fri May 08 10:04:25 UTC 2020 : Starting execution for request: 7cdd17f7-2b04-4c20-9e70-283984eb5d34
Fri May 08 10:04:25 UTC 2020 : HTTP Method: GET, Resource Path: /
Fri May 08 10:04:25 UTC 2020 : Method request path: {}
Fri May 08 10:04:25 UTC 2020 : Method request query string: {}
Fri May 08 10:04:25 UTC 2020 : Method request headers: {}
Fri May 08 10:04:25 UTC 2020 : Method request body before transformations:
Fri May 08 10:04:25 UTC 2020 : Endpoint request URI: https://lambda.eu-west-2.amazonaws.com/2015-03-31/functions/arn:aws:lambda:eu-west-2:387124123361:function:lambda-api-gateway-proxy-root-get/invocations
Fri May 08 10:04:25 UTC 2020 : Endpoint request headers: {x-amzn-lambda-integration-tag=7cdd17f7-2b04-4c20-9e70-283984eb5d34, Au
```

Create Resource

APIs > MyFirstAPI (kfz6b0ty9i) > Resources > / (4xtrlw16w7) > Create | Show all hints

Resources | Actions

/
GET

New Child Resource

Use this page to create a new child resource for your resource.

Configure as proxy resource

Resource Name* houses

Resource Path* / houses

You can add path parameters using brackets. For example, the resource path **{username}** represents a path parameter called 'username'. Configuring /{proxy+} as a proxy resource catches all requests to its sub-resources. For example, it works for a GET request to /foo. To handle requests to /, add a new ANY method on the / resource.

Enable API Gateway CORS

* Required

Cancel | Create Resource

Resource Groups | stephane @ datacumulus-cour... | London | Support

APIs > My | Show all hints

Resources

/
GET
/hous
GET

Add Permission to Lambda Function

You are about to give API Gateway permission to invoke your Lambda function:
arn:aws:lambda:eu-west-2:387124123361:function:lambda-api-gateway-proxy-houses-get

Cancel | OK

Mock
AWS Service
VPC Link

Use Lambda Proxy integration

Lambda Region eu-west-2

Lambda Function lambda-api-gateway-proxy-houses-get

Use Default Timeout

Save

APIs > MyFirstAPI (kfz6b0ty9i) > Resources > /houses (zr32nh) > GET | Show all hin

Resources | Actions

/houses - GET - Method Execution

/
GET
/houses
GET

METHOD ACTIONS
Edit Method Documentation
Delete Method

RESOURCE ACTIONS
Create Method
Create Resource
Enable CORS
Edit Resource Documentation
Delete Resource

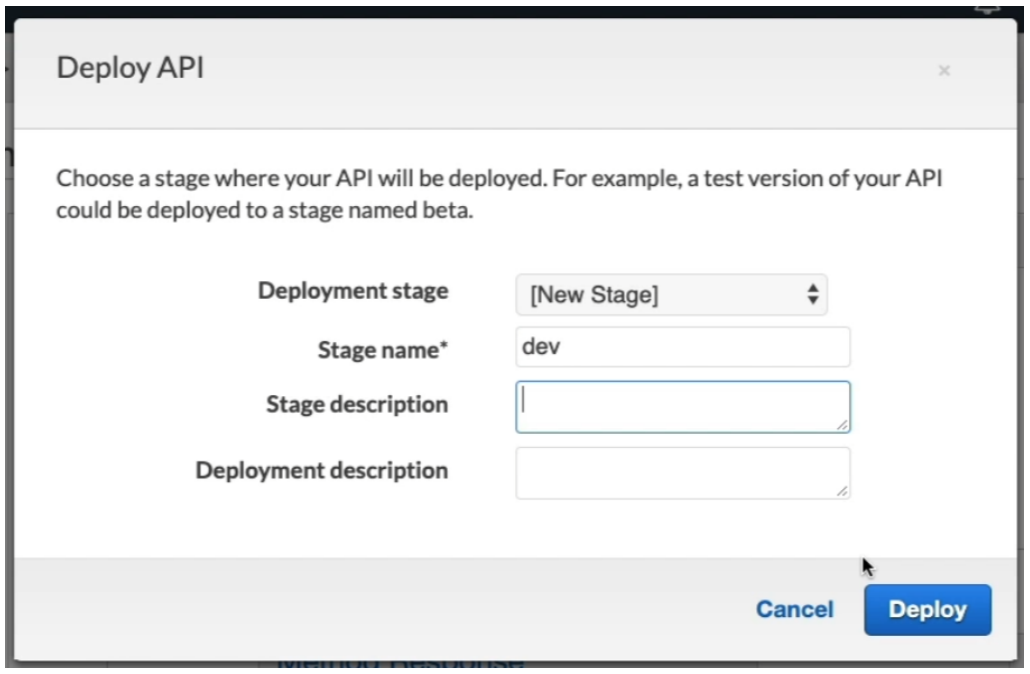
API ACTIONS
Deploy API
Import API
Edit API Documentation
Delete API

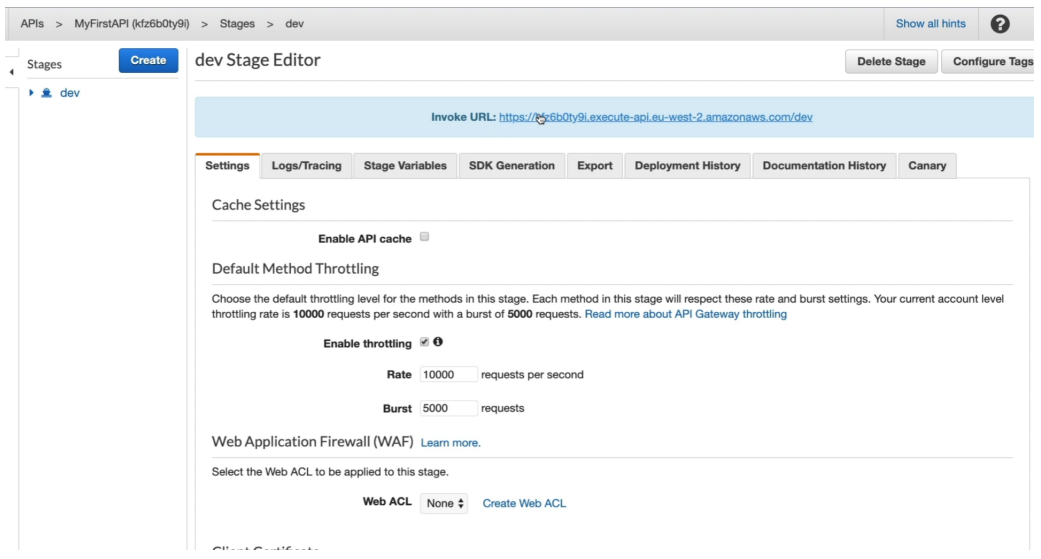
Method Request
Auth: NONE
ARN: arn:aws:execute-api:eu-west-2:387124123361:kfz6b0ty9i/*/GET/houses

Integration Request
Type: LAMBDA_PROXY

Method Response
HTTP Status: Proxy
Models: application/json => Empty

Integration Response
Proxy integrations cannot be configured to transform responses.





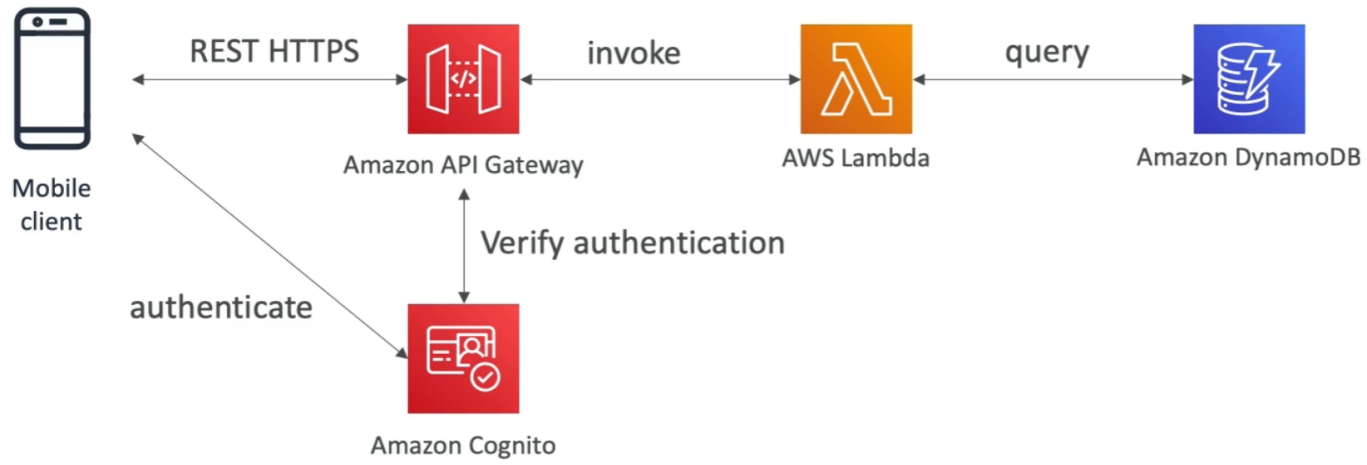
AWS Step Functions

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

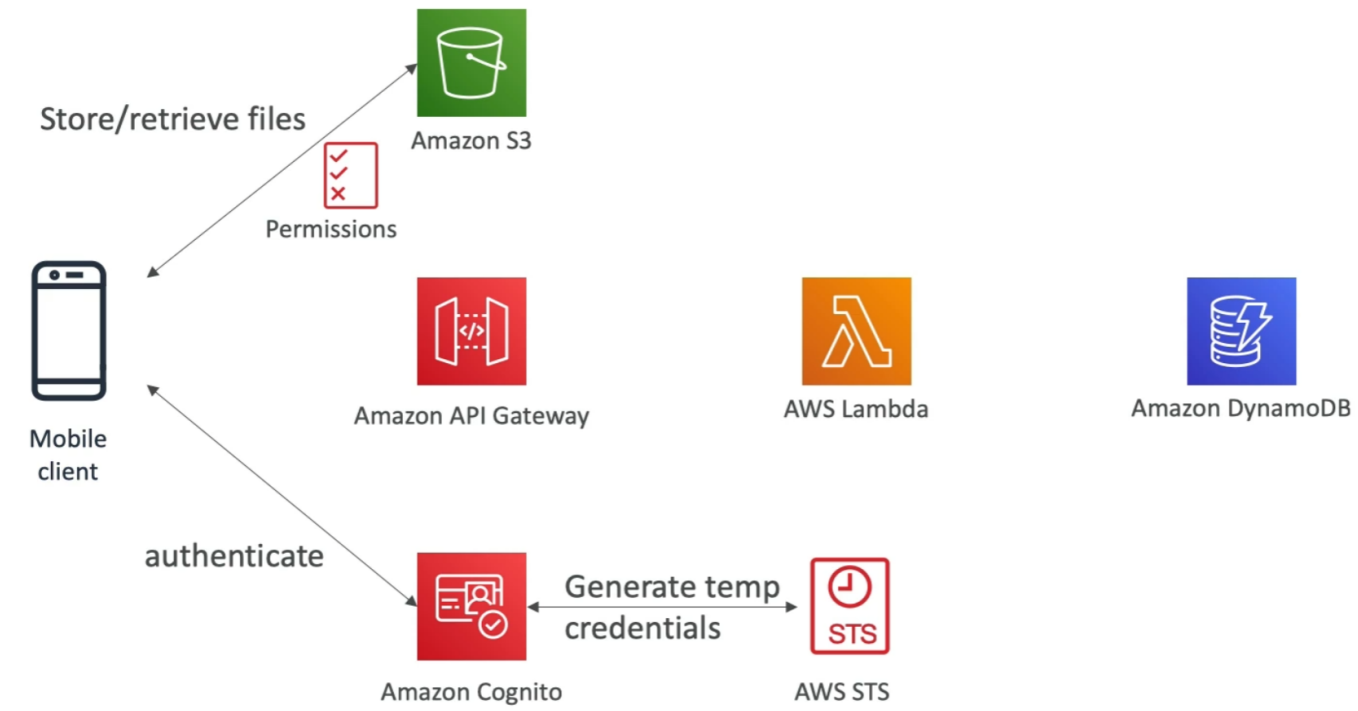
Mobile application: MyTodoList

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

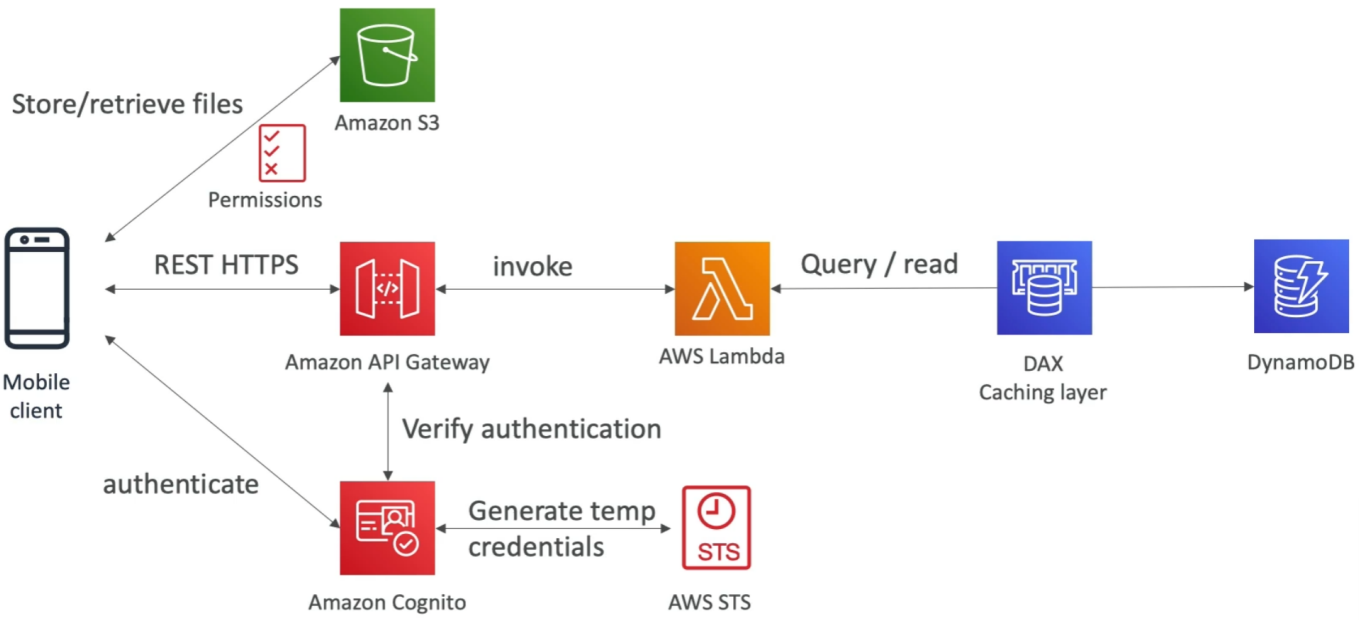
Mobile app : REST API layer



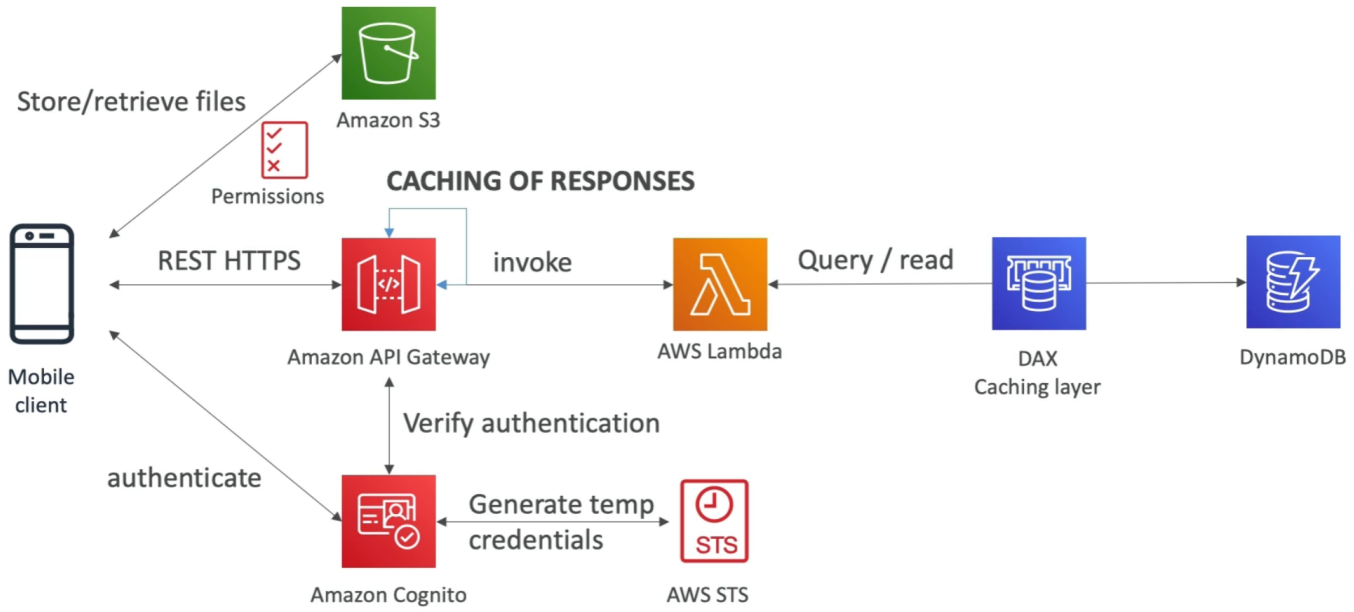
Mobile app: giving users access to S3



Mobile app: high read throughput, static data



Mobile app: caching at the API Gateway

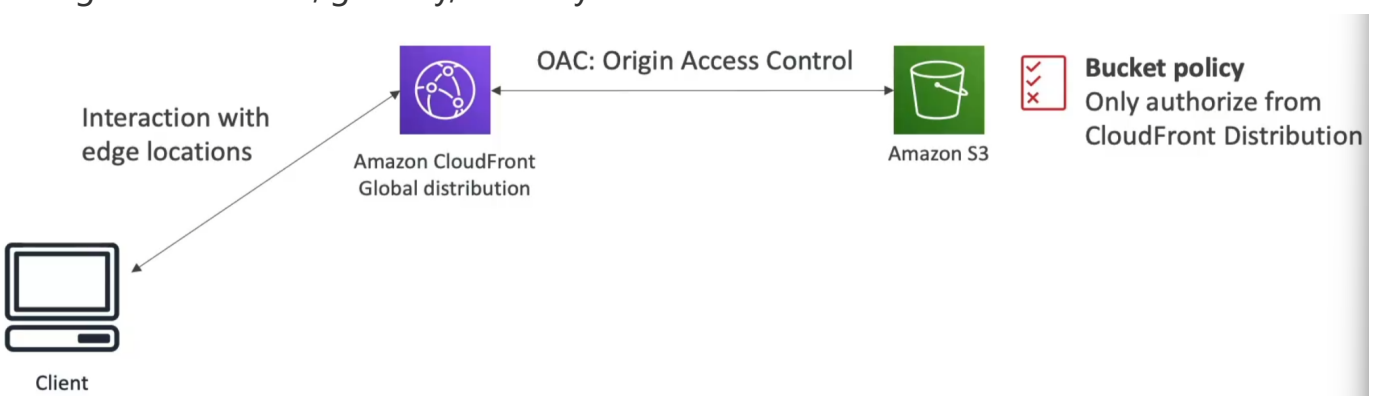


Serverless hosted website: MyBlog.com

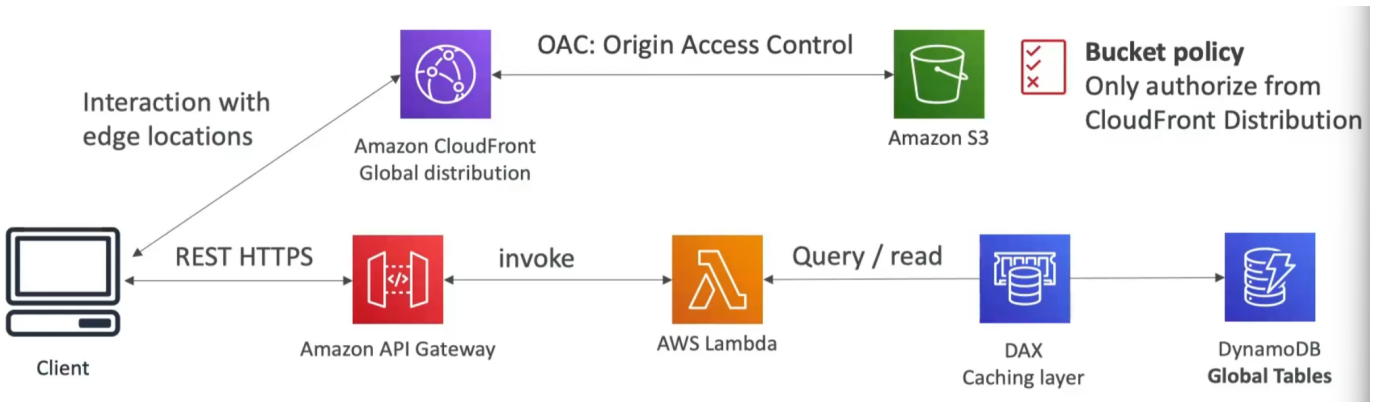
- This website should scale globally
- Blogs are rarely written, but often read
- Some of the website is purely static files, the rest is a dynamic REST API
- Caching must be implement where possible
- Any new users that subscribes should receive a welcome email
- Any photo uploaded to the blog should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

Serving static content, globally

Serving static content, globally, securely

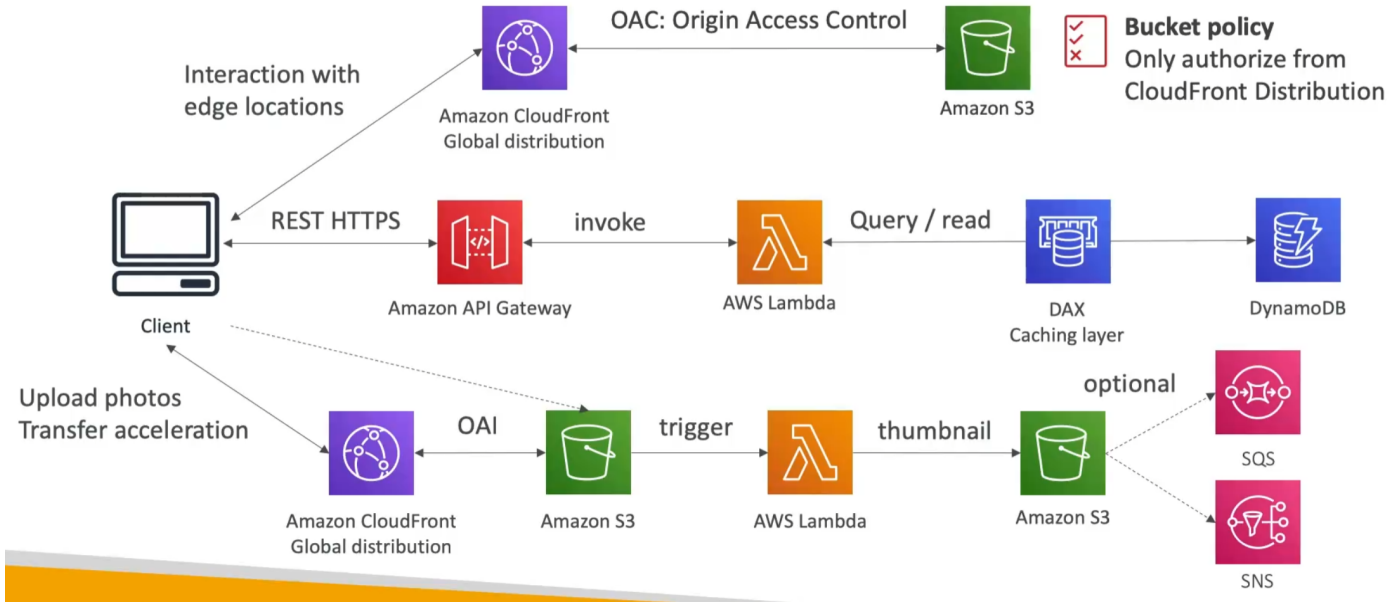


Leveraging DynamoDB Global Tables



User Welcome email flow

Thumbnail Generation flow



AWS Hosted Website Summary

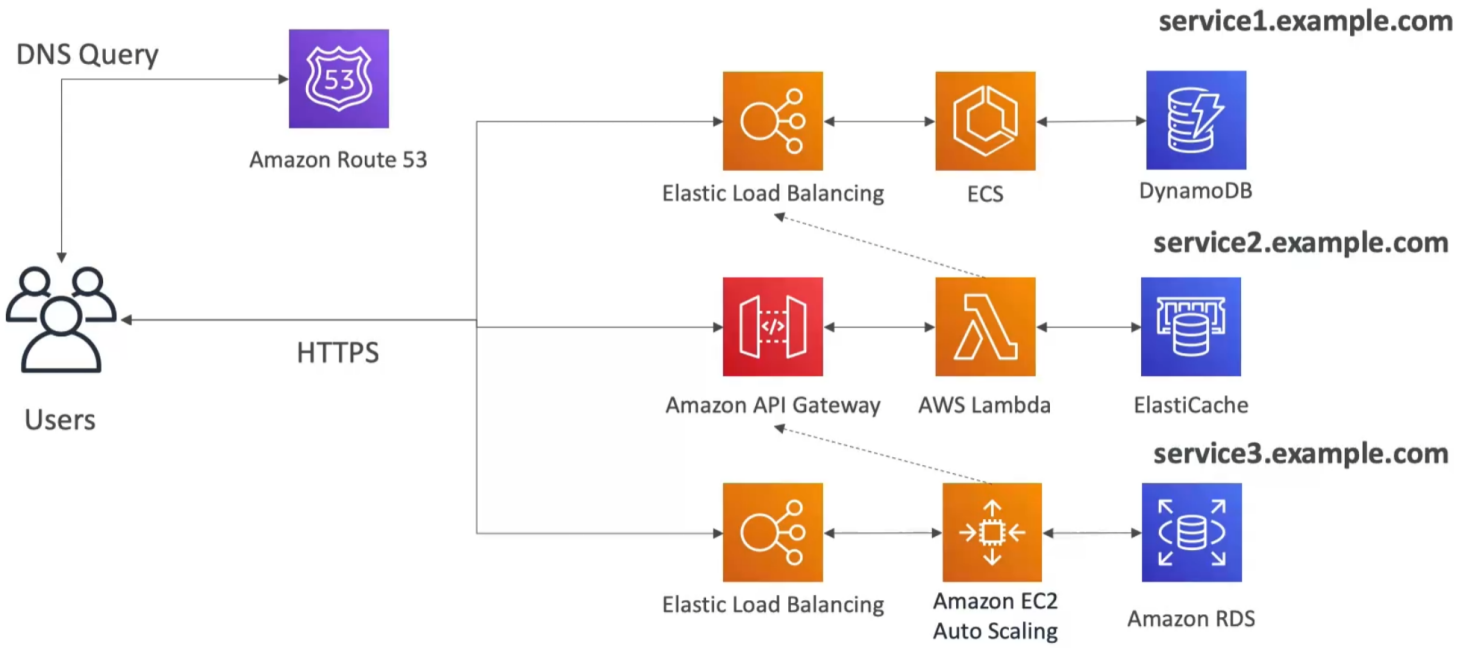
- We've seen static content being distributed using CloudFront with S3
- The REST API was serverless, didn't need Cognito because public
- We leveraged a Global DynamoDB table to serve the data globally
- (we could have used Aurora Global Database)
- We enabled DynamoDB streams to trigger a Lambda function
- 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



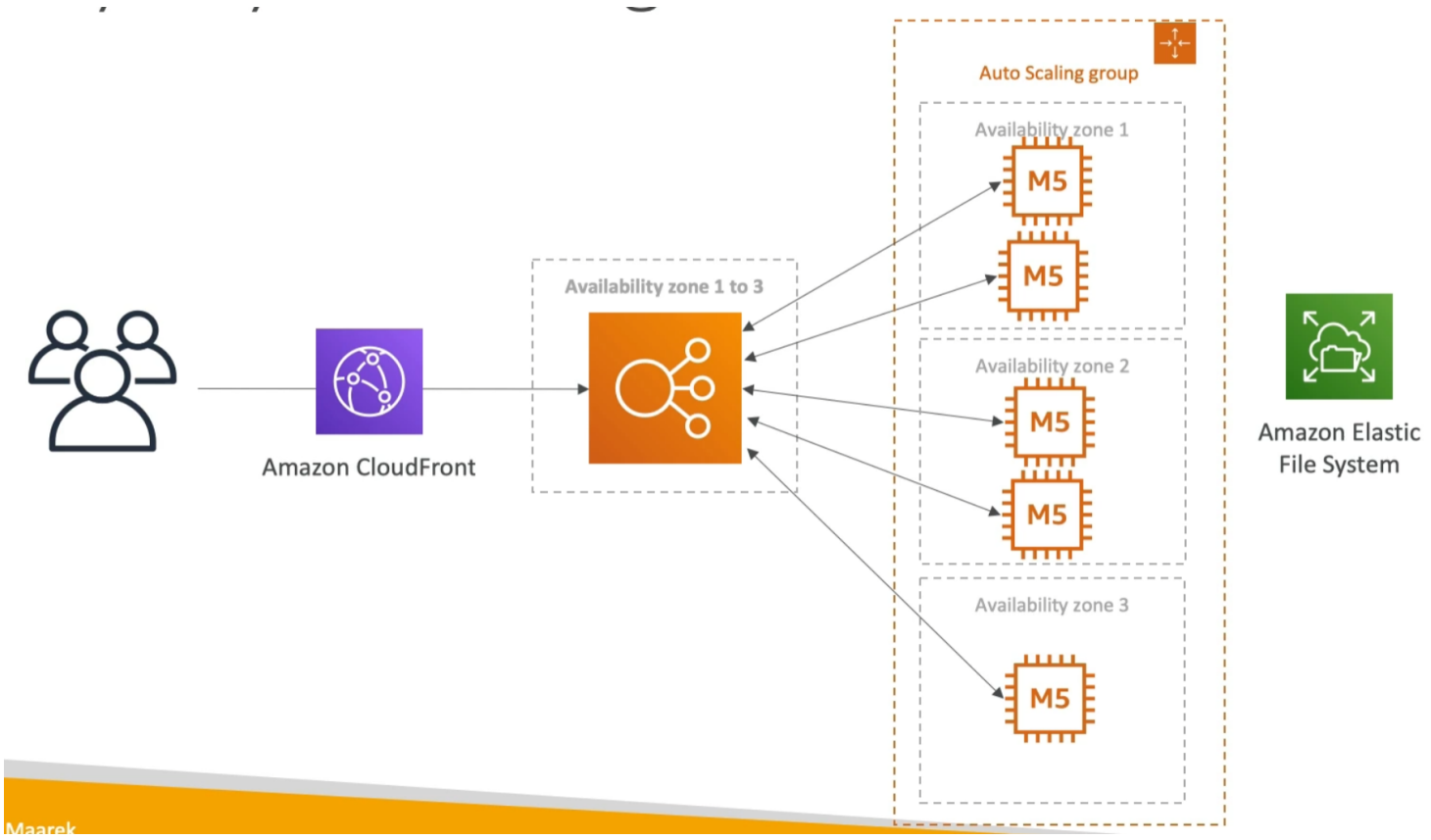
Discussions on Micro Services

- **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?

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!

Choosing the Right Database

- We have a lot of managed databases on AWS to choose from
- Questions to choose the rigth 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 - Summary

- Managed PosgreSQL / MySQL / Oracle / SQL Server / MariaDB / Custom
- Provisioned RDS Instance Size and EBS Volume Type & Size
- Auto-scaling capability for Storage
- Support for Read Replicas and Multi AZ
- 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 - Summary

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 Multi-Master - for continuous writes failover (high write availability)
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 - Summary

- Managed Redis / Memcached (similar offering as RDS, but for caches)
- In-memory data store, sub - millisecond latency
- Must provision an EC2 instance type
- 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 - Summary

- 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 read 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
- Automated backups up to 35 days with PITR (restroe 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, doesn't have SQL query language available

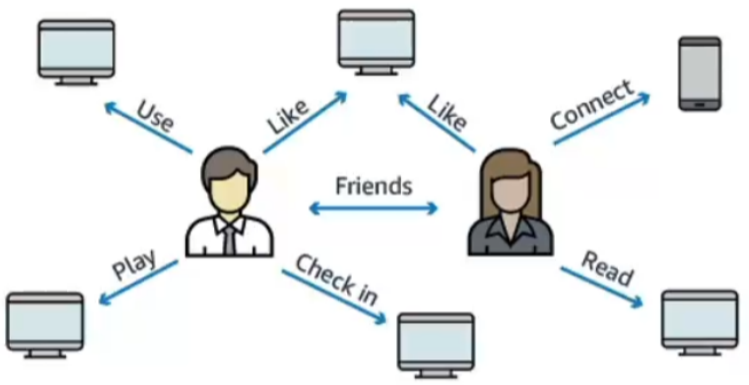
Amazon S3 - Summary

- 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 5TB, 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

DocumentDB

- Aurora is an "AWS-implementation" of PostgreSQL / MySQL ...
- **DocumentDB is the same for 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
- DocumentDB storage automatically grows in increments of 10 GB, up to 64 TB
- 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
  - User 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

Amazon Keyspaces (for Apache Cassandra)

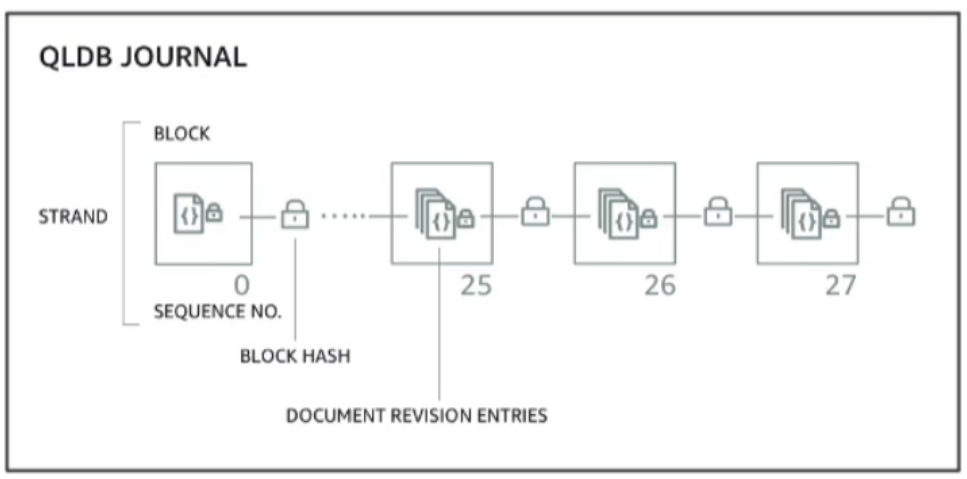
- Apache Cassandra is an open-source NoSQL distributed database

- A managed Apache Cassandra-compatible database service
- Serverless, Scalable, highly available, fully managed by AWS
- 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 Case： **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 history of all the changes made to your application data** over time
- **Immutable** system: no entry can be removed or modified, cryptographically verifiable
-

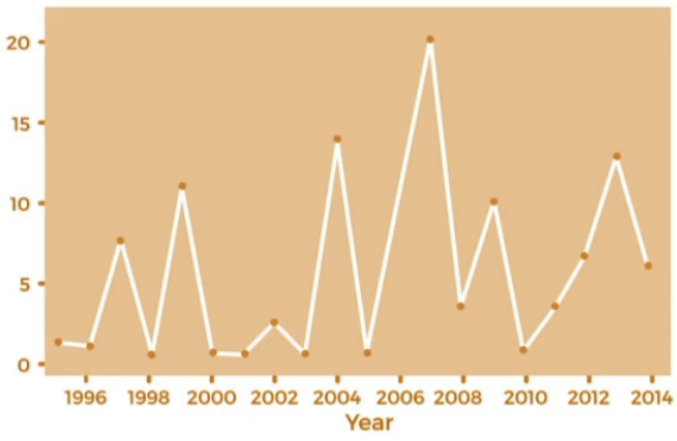


-
- 2 - 3x better performance than common ledger blockchain frameworks, manipulate data using SQL
- Difference with Amazon Managed Blockchain: **no decentralization component**, in accordance with financial regulation rules

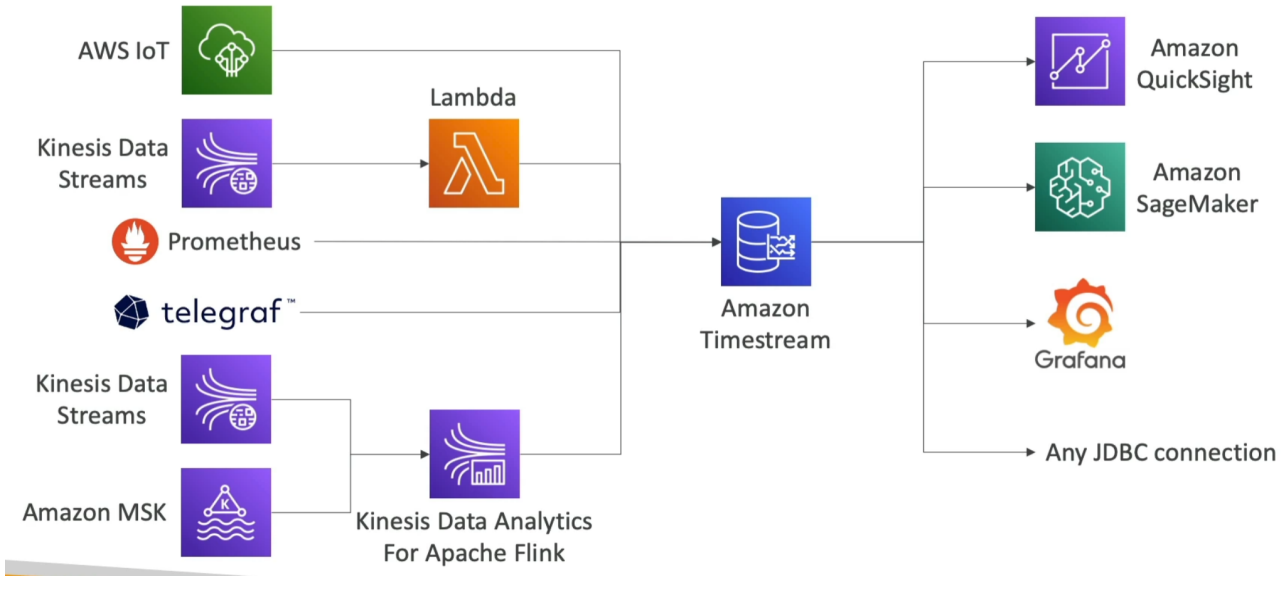
Amazon Timestream

- Fully managed, fast, scalable,serverless time series database
- Automatically scales up/down to adjust capacity
- Store and analyze trillions of events per day
- 1000s 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 applicatiuons, real-time analytics, ...



Amazon Timestream - Architecture

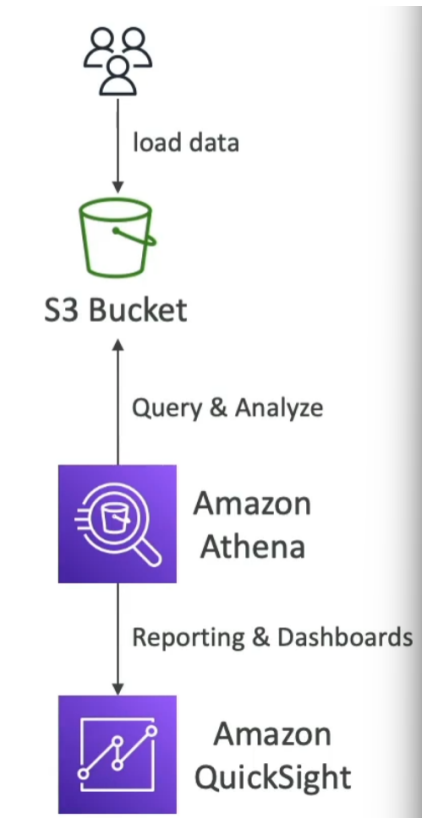


Amazon Athena

- Serverless query service to analyze data stored in Amazon S3
- Uses standard SQL language to query the files (built on Presto)
- Supports CSV,JSON, ORC, Avto , and Parquet
- Pricing: $5.00 per TB of data scanned
- Commonly used with Amazon Quicksight for reporting/dashboards

- **Use cases**: Business intelligentce / analytics / reporting, analyze & query VPC Flow Logs, ELB Logs, CloudTrail trails, etc ...

- **Exam Tip**: analyze data in S3 using serverless SQL, use Athena

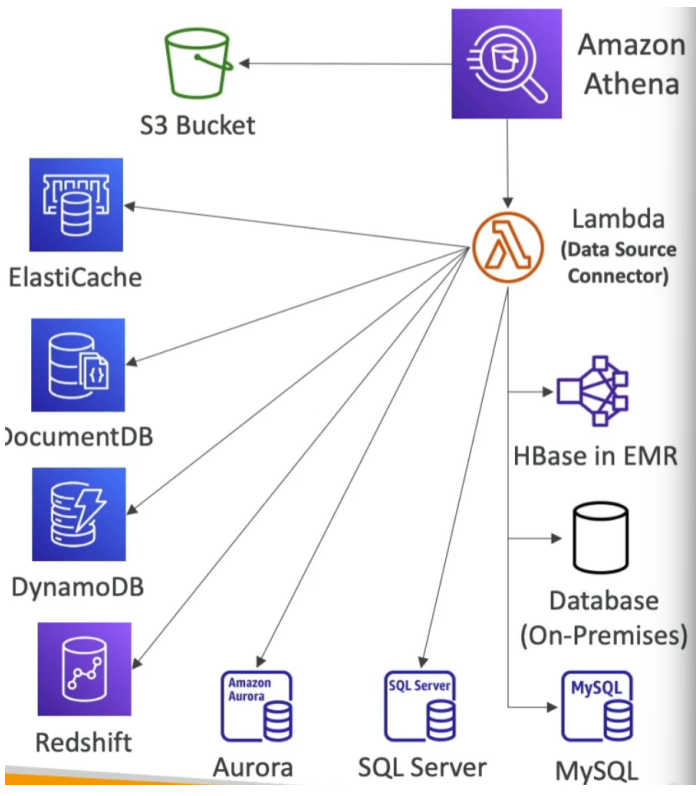


Amazon Athena - Performance Improvement

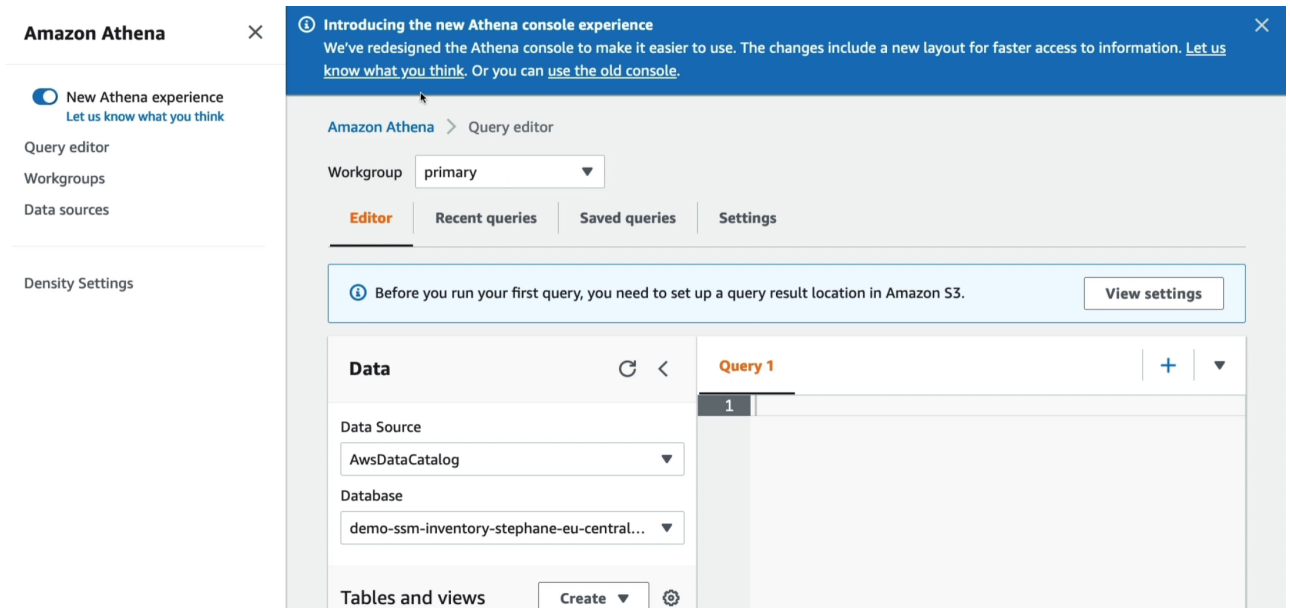
- Use **columnar data** for cost-savings (less scan)
  - Apache Parquet or ORC is recommended
  - Huge performance improvement
  - Use Glue to convert your data your Parquet or ORC
- **Compress data** for smaller retrievals (bzip2, gzip, lz4, snappy, zlip,zstd...)
- **Partition** datasets in S3 for easy querying on virtual columns
  - s3://yourBucket/pathToTable
    /<PARTITION_COLUMN_NAME>=<VALUE>
    /<PARTITION_COLUMN_NAME>=<VALUE>
    /<PARTITION_COLUMN_NAME>=<VALUE>
    /ect...
  - Example:s3://athena-examples/flight/parquet/year=1991/month=1/day=1/
- **Use larger files**(> 128 MB) to minimize overhead

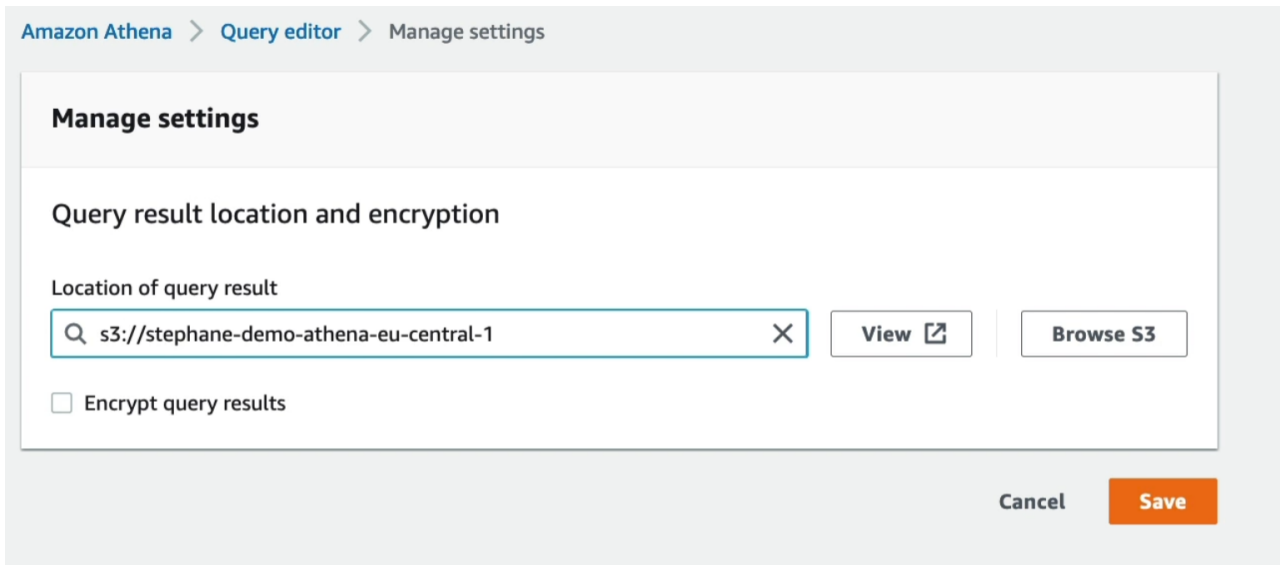
Amazon Athena - Federated Query

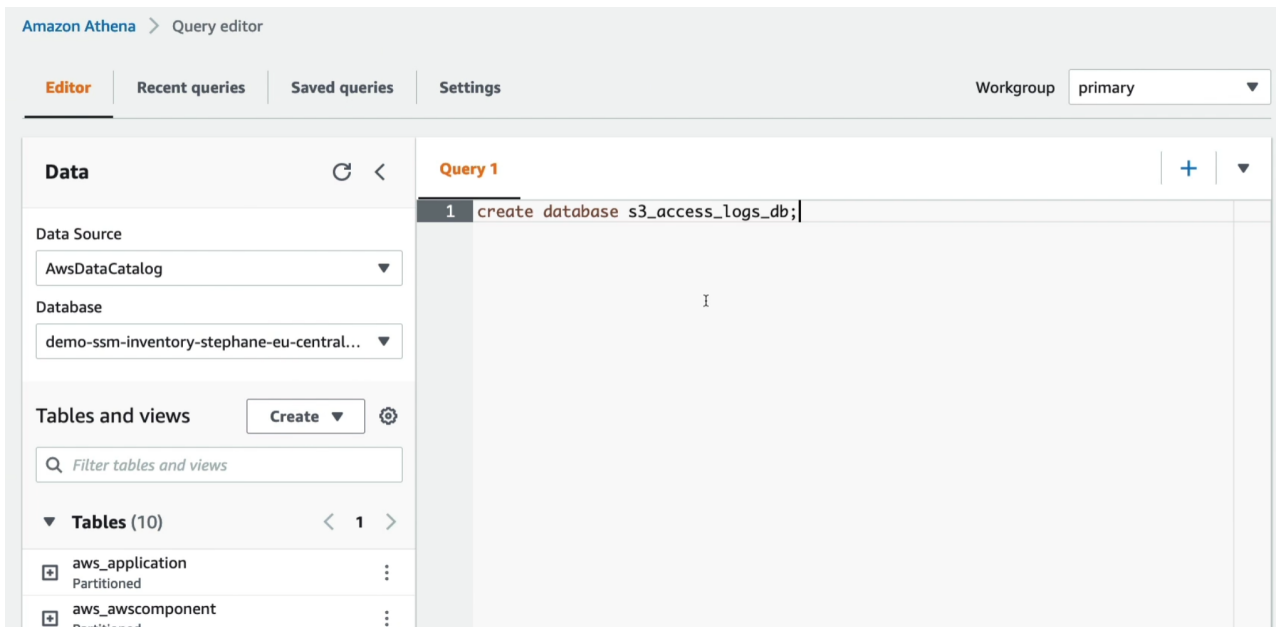
- Allows you to run SQL queries across data stored in relational, non-relational,object, and custom data sources (AWS or on-premises)
- Uses Data Source Connectors that run on AWS Lambda to run Federated Queries (e.g., CloudWatch Logs, DynamoDB, RDS, ...)
- Store the results back in Amazon S3



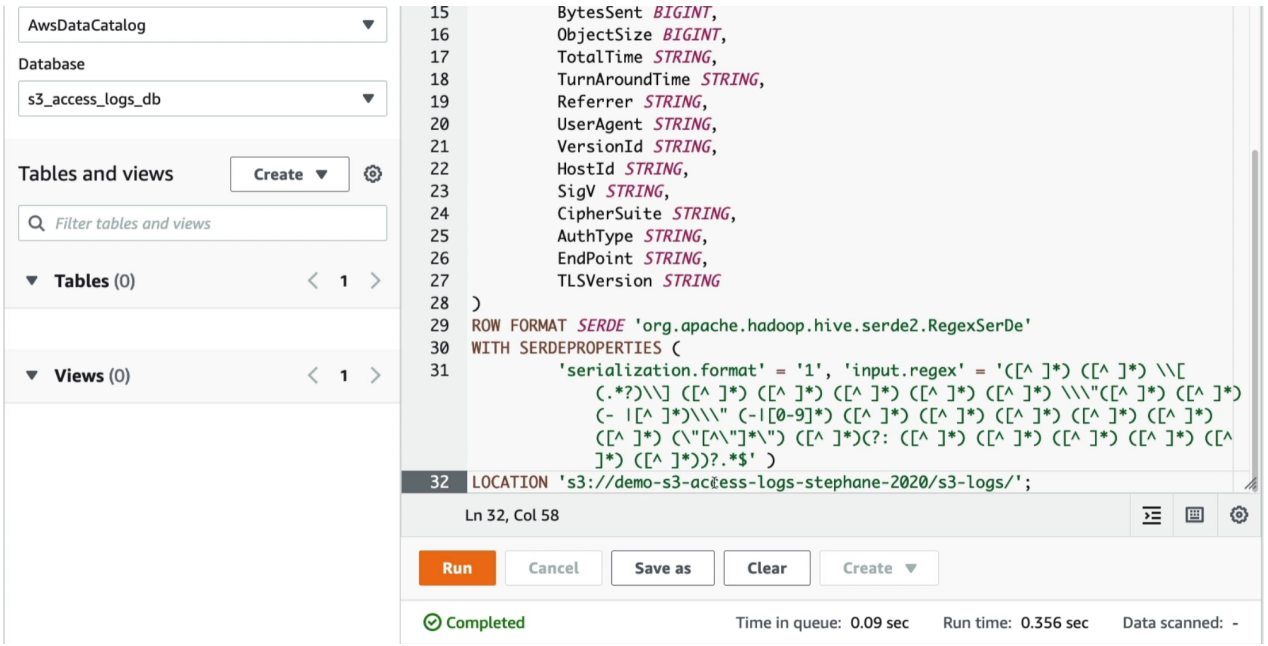
Hands on

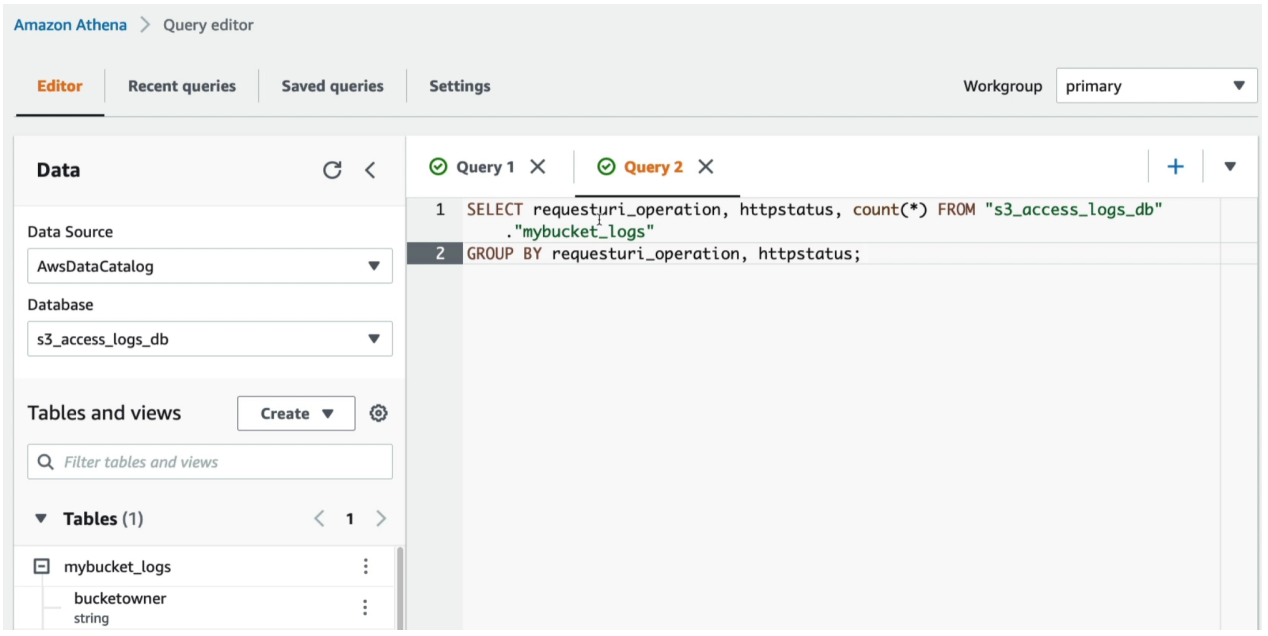

Amazon Athena
New Athena experience
Let us know what you think
Query editor
Workgroups
Data sources
Density Settings
Introducing the new Athena console experience
We've redesigned the Athena console to make it easier to use. The changes include a new layout for faster access to information. Let us know what you think. Or you can use the old console.
Amazon Athena > Query editor
Workgroup primary
Editor
Recent queries
Saved queries
Settings
Before you run your first query, you need to set up a query result location in Amazon S3.
View settings
Data
Query 1
1
Data Source
AwsDataCatalog
Database
demo-ssm-inventory-stephane-eu-central...
Tables and views
Create


Amazon Athena > Query editor > Manage settings
Manage settings
Query result location and encryption
Location of query result
s3://stephane-demo-athena-eu-central-1
View
Browse S3
Encrypt query results
Cancel
Save


Amazon Athena > Query editor
Editor
Recent queries
Saved queries
Settings
Workgroup primary
Data
Query 1
1 create database s3_access_logs_db;
Data Source
AwsDataCatalog
Database
demo-ssm-inventory-stephane-eu-central...
Tables and views
Create
Filter tables and views
Tables (10)
aws_application
Partitioned
aws_awscomponent

```
CREATE EXTERNAL TABLE IF NOT EXISTS s3_access_logs_db.
mybucket_logs(
        BucketOwner STRING,
        Bucket STRING,
        RequestDateTime STRING,
        RemoteIP STRING,
        Requester STRING,
        RequestID STRING,
        Operation STRING,
        Key STRING,
        RequestURI_operation STRING,
        RequestURI_key STRING,
        RequestURI_httpProtoversion STRING,
        HTTPstatus STRING,
        ErrorCode STRING,
        BytesSent BIGINT,
        ObjectSize BIGINT,
        TotalTime STRING,
        TurnAroundTime STRING,
        Referrer STRING,
        UserAgent STRING,
        VersionId STRING,
```





Results (16) Download results

Search rows 1

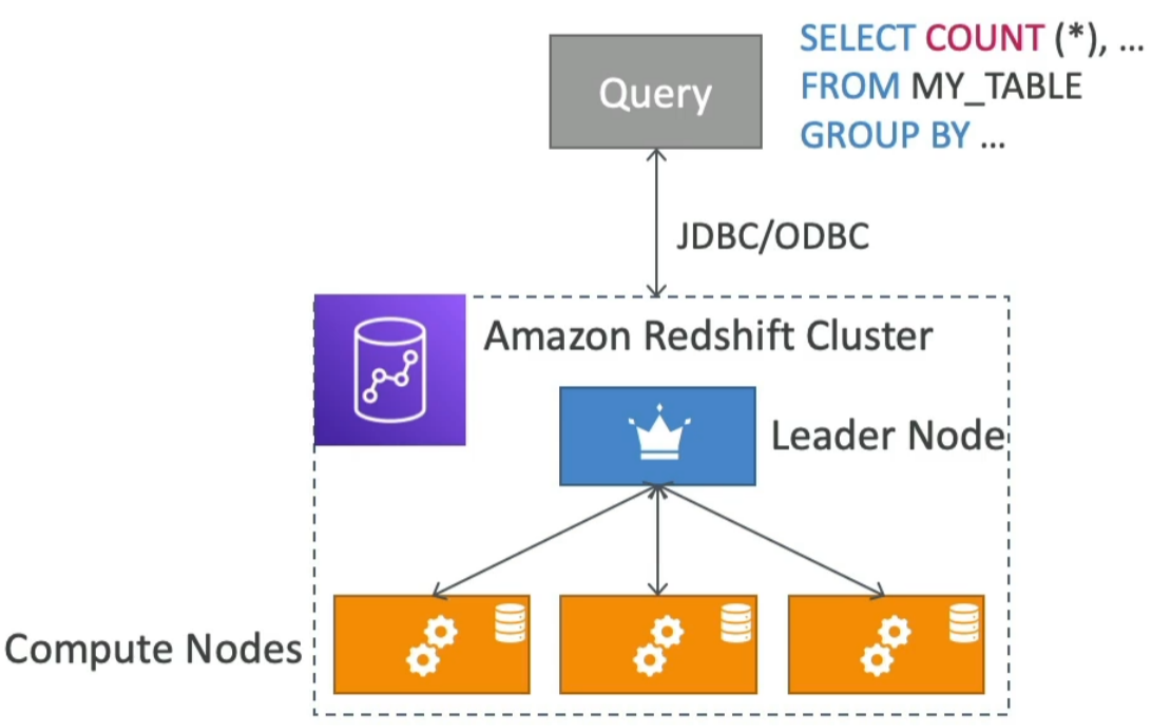
requesturi_operation	httpstatus	_col2
GET	404	49
PUT	403	12
POST	202	1
PUT	400	1
POST	200	1
GET	400	1
		17

Redshift Overview

- Redshift is based on PostgreSQL, but it's not used for OLTP
- **It's OLAP - online analytical processing** (**analytics and data warehousing**)
- 10x better performance than other data warehouses, scale to PBs of data
- **Columnar** storage of data (instead of row based) & parallel query engine
- Pay as you go based on the instances provisioned
- 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

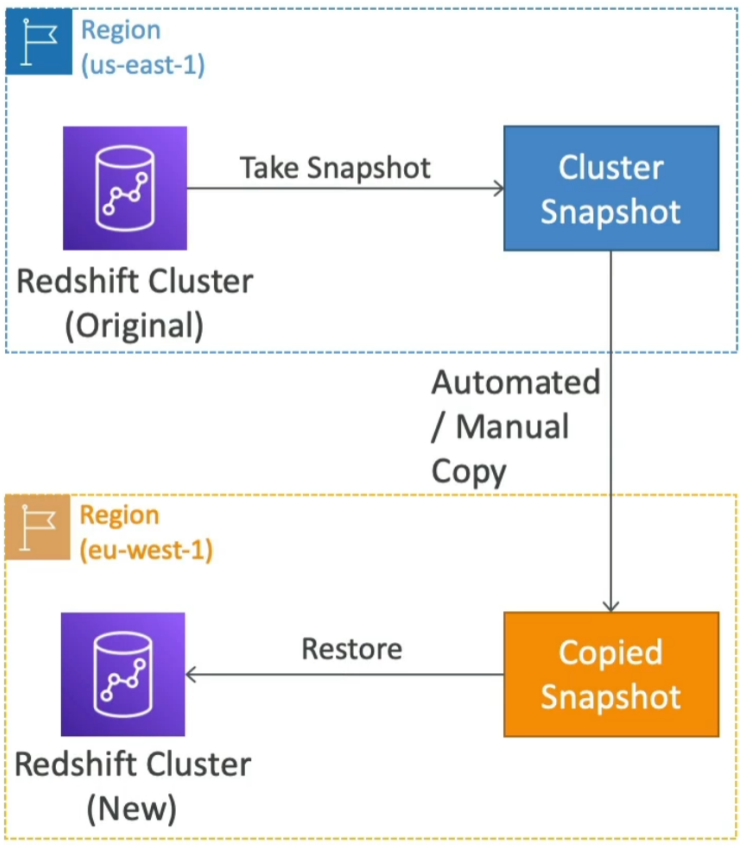
- Leader node: for query planning, results aggregation
- Compute node: for perfroming the queries,send results to leader
- You provision the node size in advance
- You can used Reserved Instances for cost savings



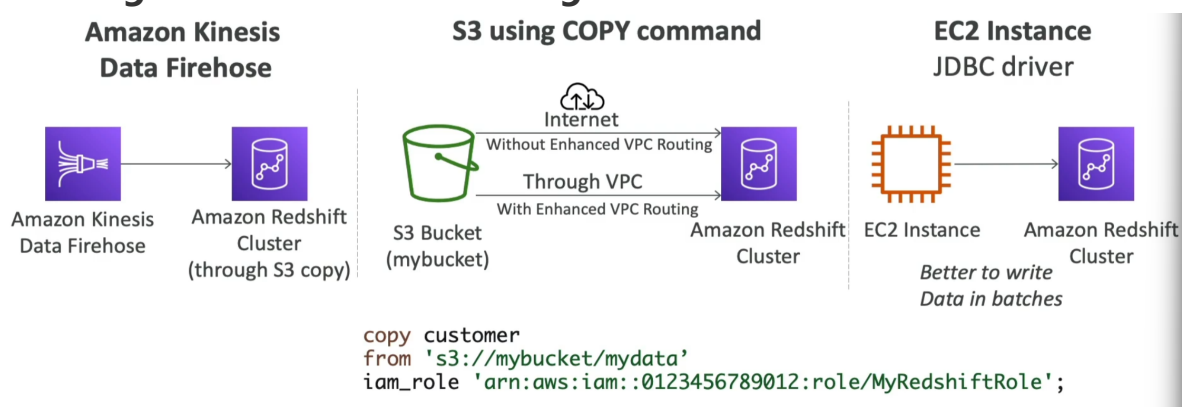
Redshift - Snapshots&DR

- **Redshift has no "Multi-AZ" mode**
- 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
- 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**

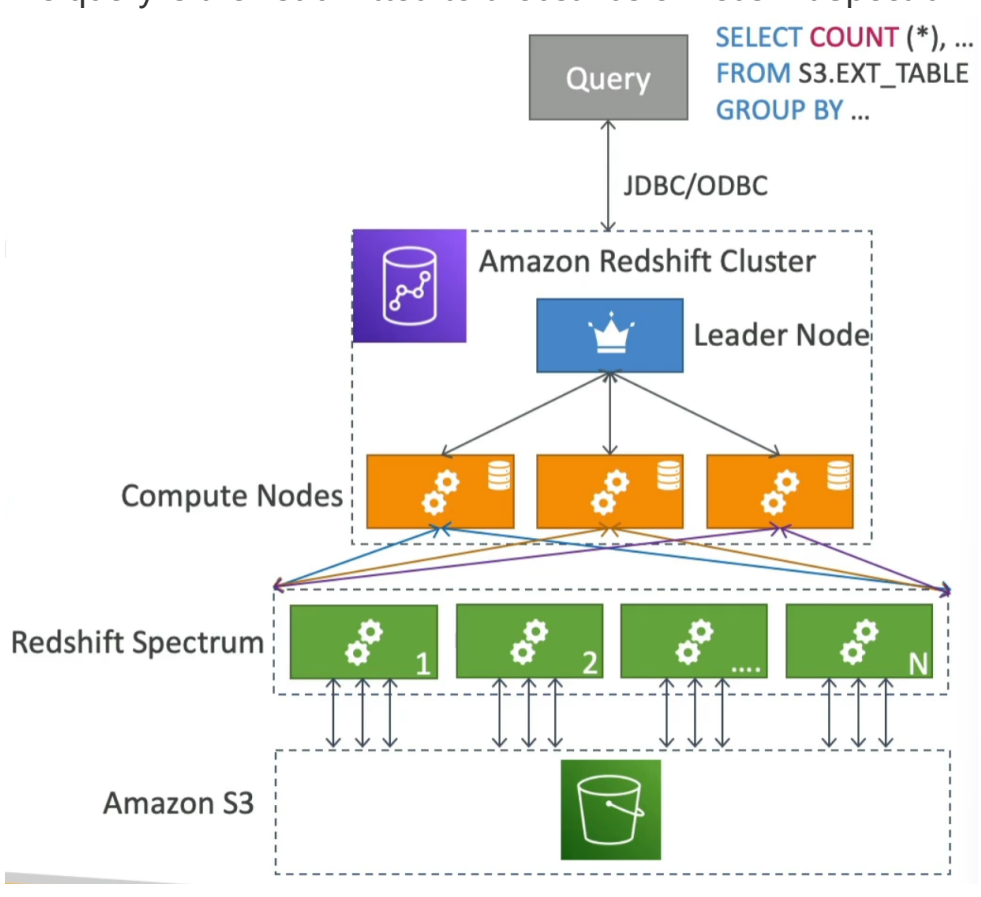


Loading data into Redshift: Large inserts are MUCH better



Redshift Spectrum

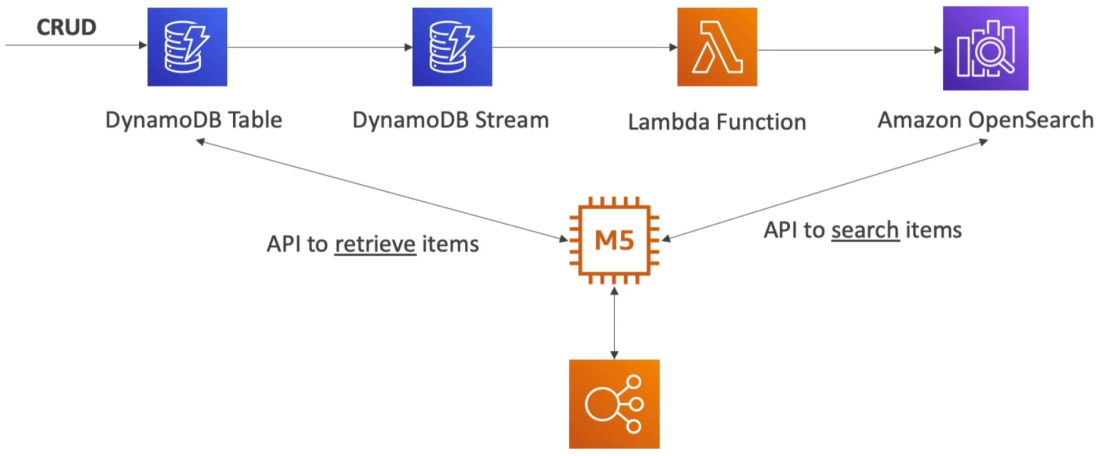
- Query data that is already in S3 without loading it
- **Must have a Redshift cluster available to start the query**
- The query is then submitted to thousands of Redshift Spectrum nodes



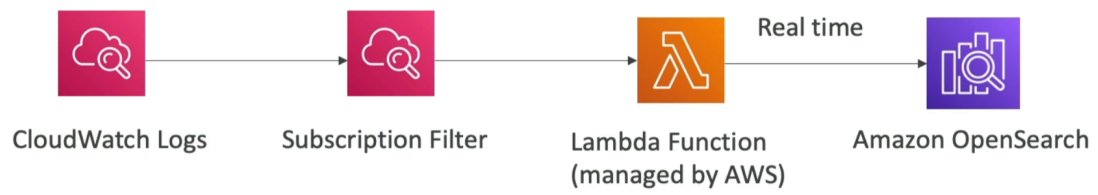
Amazon OpenSearch Service

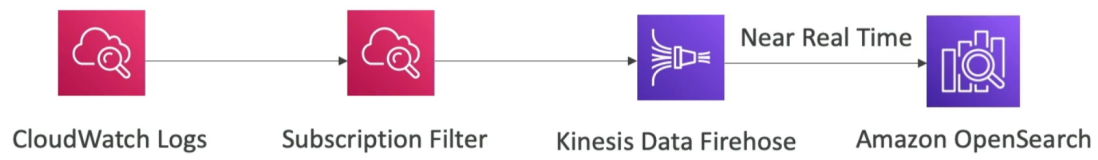
- **Amazon OpenSearch is successor to Amazon ElasticSearch**
- In DynamoDB, queries only exist by primary key or indexes ...
- **With OpenSearch, you can search any field, even partially matches**
- It's common to use OpenSearch as a complement to another database
- OpenSearch requires a cluster of instances (not serverless)
- Does not support SQL (it has its own query language)
- 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

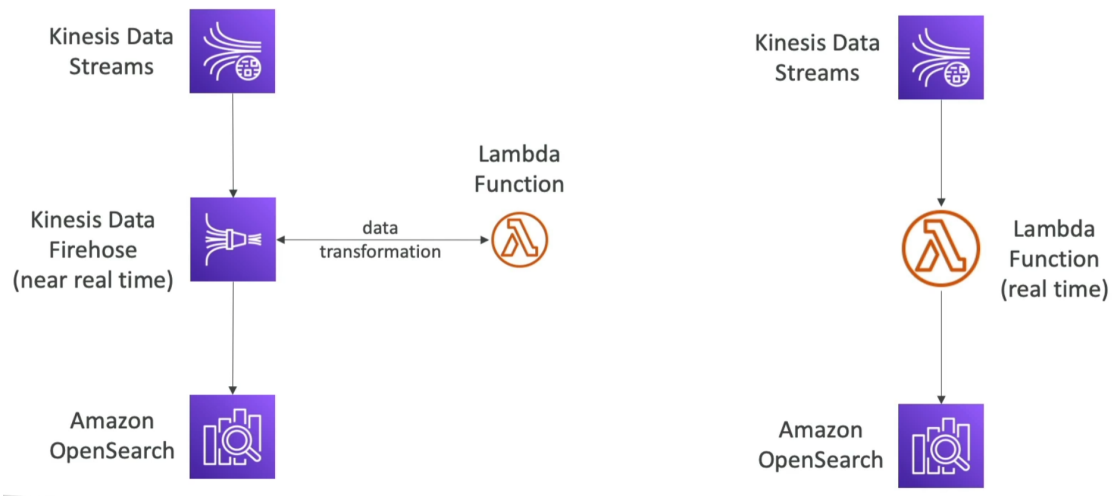


OpenSearch patterns CloudWatch Logs





OpenSearch patterns Kinesis Data Streams & Kinesis Data Firehose



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 Apache Spark,Hbase,Presto, Flink...
- EMR takes care of all the provisioning and configuration
- Auto-scaling and integrated with Spot instances

- **Use cases: data processing, machine learning, web indexing, big data ...**

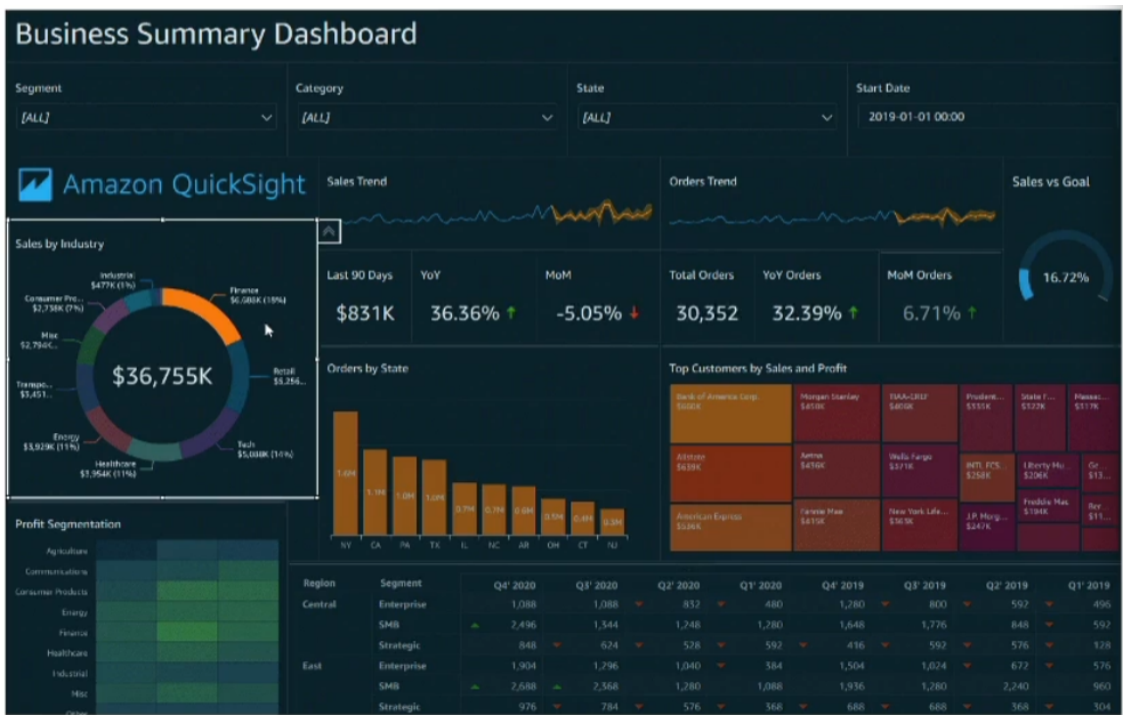
Amazon EMR - Node types & purchasing

- **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

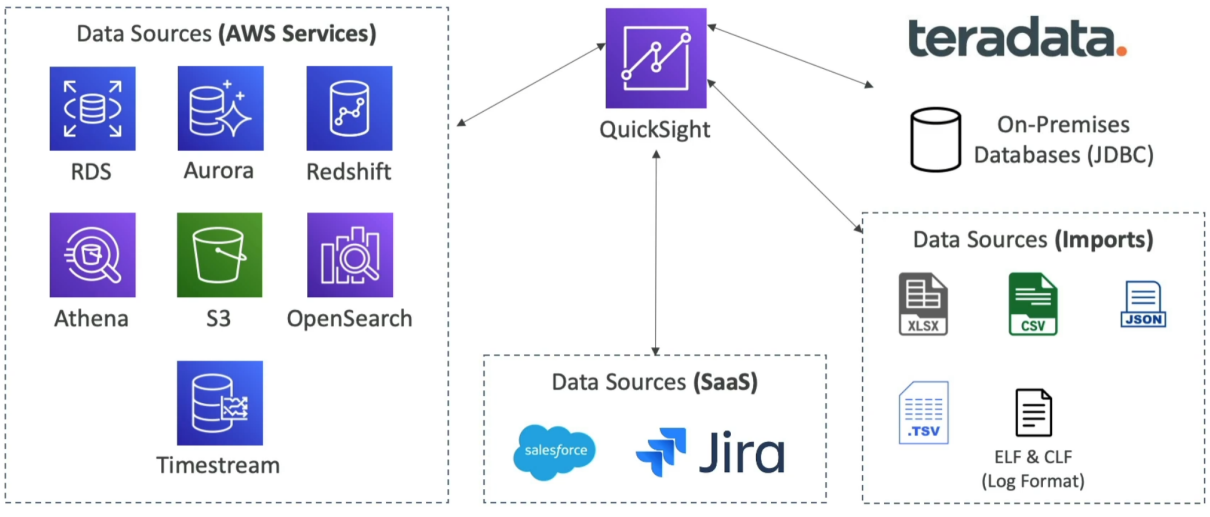
- Can have long-running cluster, or transient (temporary) cluster

Amazon 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 QuickSight
- Enterprise edition: Possibility to setup **Column-Level security (CLS)**



QuickSight Integrations



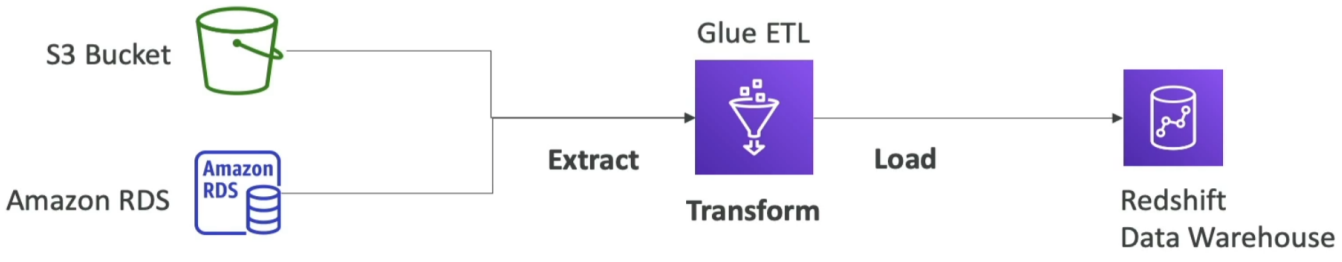
QuickSight - Dasjboard & Analysis

- Define Users (standard versions) and Groups (enterprise version)
  - These users & groups only exist within QuickSight,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)

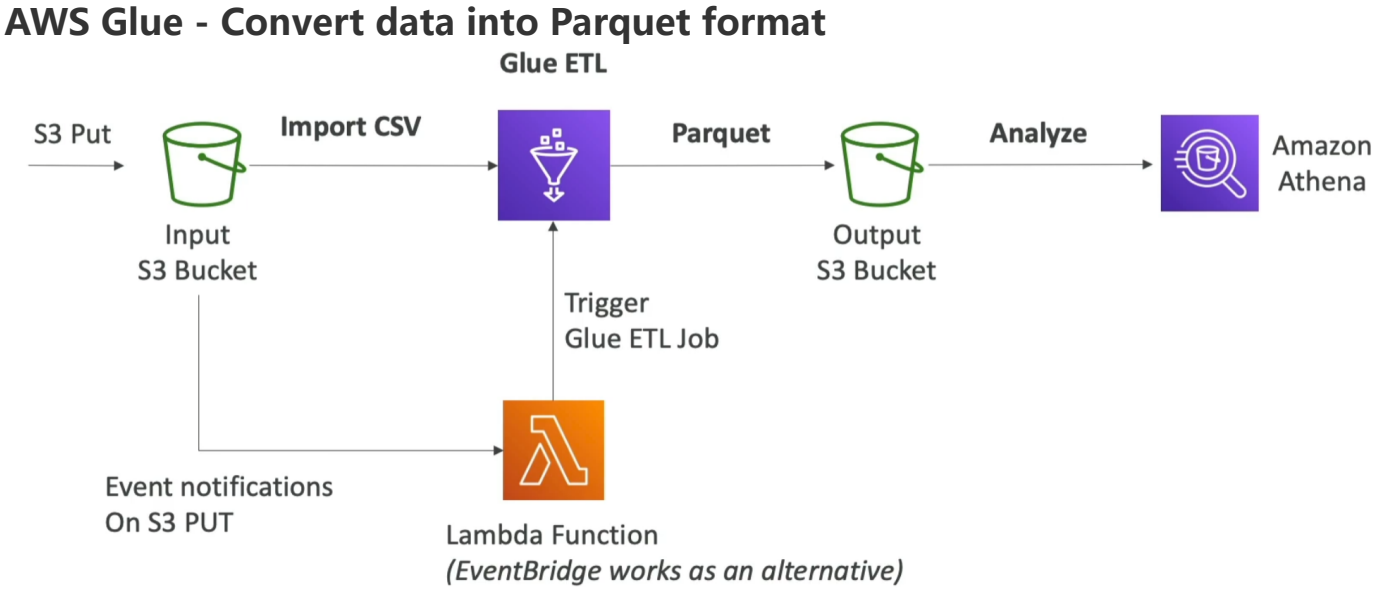
- **You can share the analysis or the dashboard with Users or Groups**
- To share a dashboard, you must first publish it
- Users who see the dashboard can also see the underlying data

AWS Glue

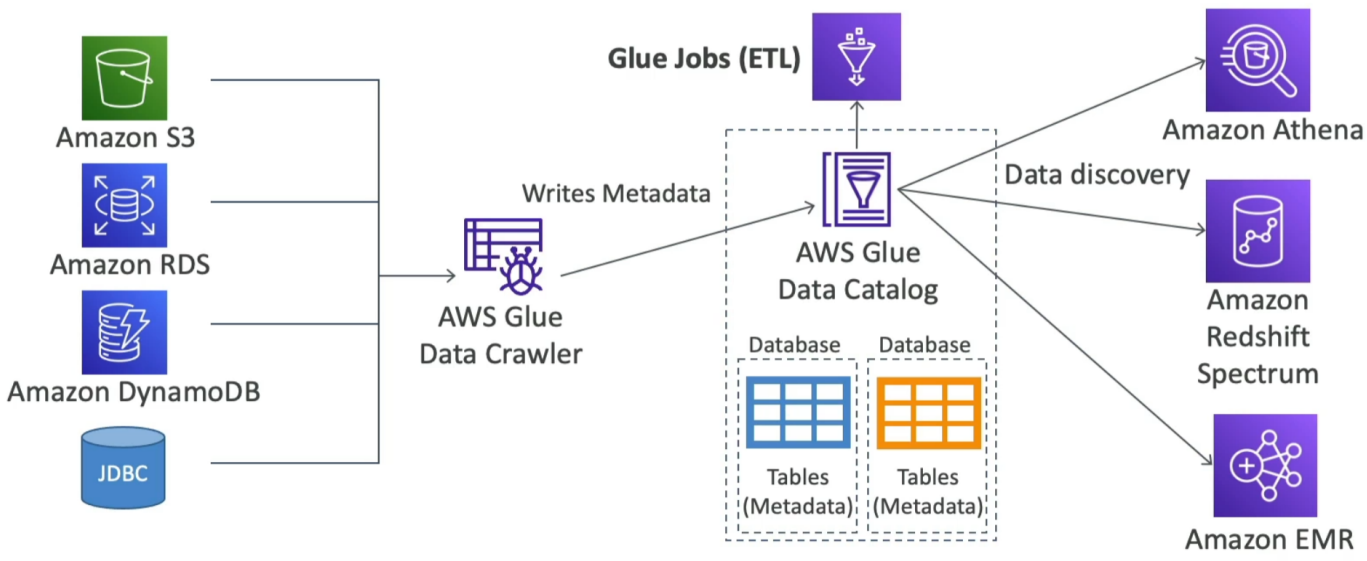
- Managed **extract,transform,and load (ETL)** service
- Useful to prepare and transform data for analytics
- Fully **serverless** service



AWS Glue - Convert data into Parquet format



Glue Data Catalog: catalog of datasets

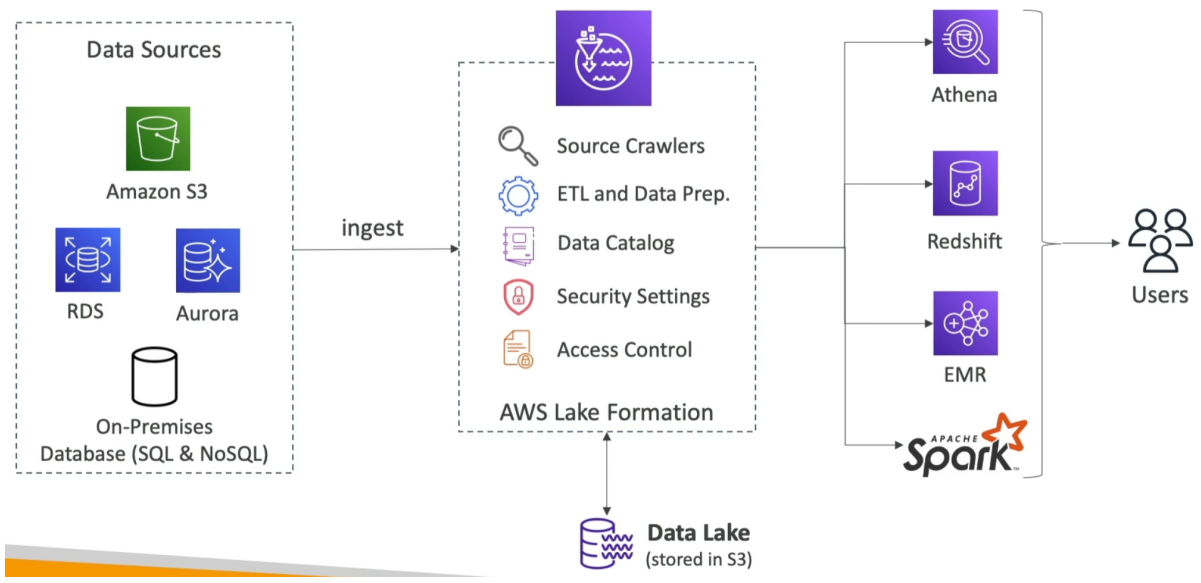


Glue - things to know at a high-level

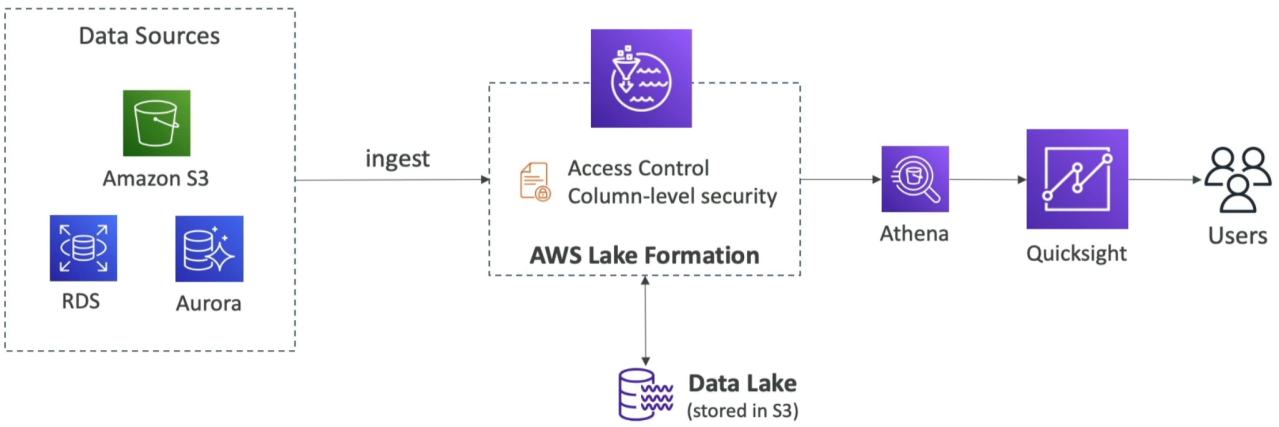
- **Glue Job Bookmarks**: prevent re-processing old data
- **Glue Elastic Views**:
  - Combine and replicate data across multiple data stores using SQL
  - No custom code, Glue monitors for changes in the source data, serverless
  - Leverages a "virtual table" (materialized view)
- **Glue DataBrew**: 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

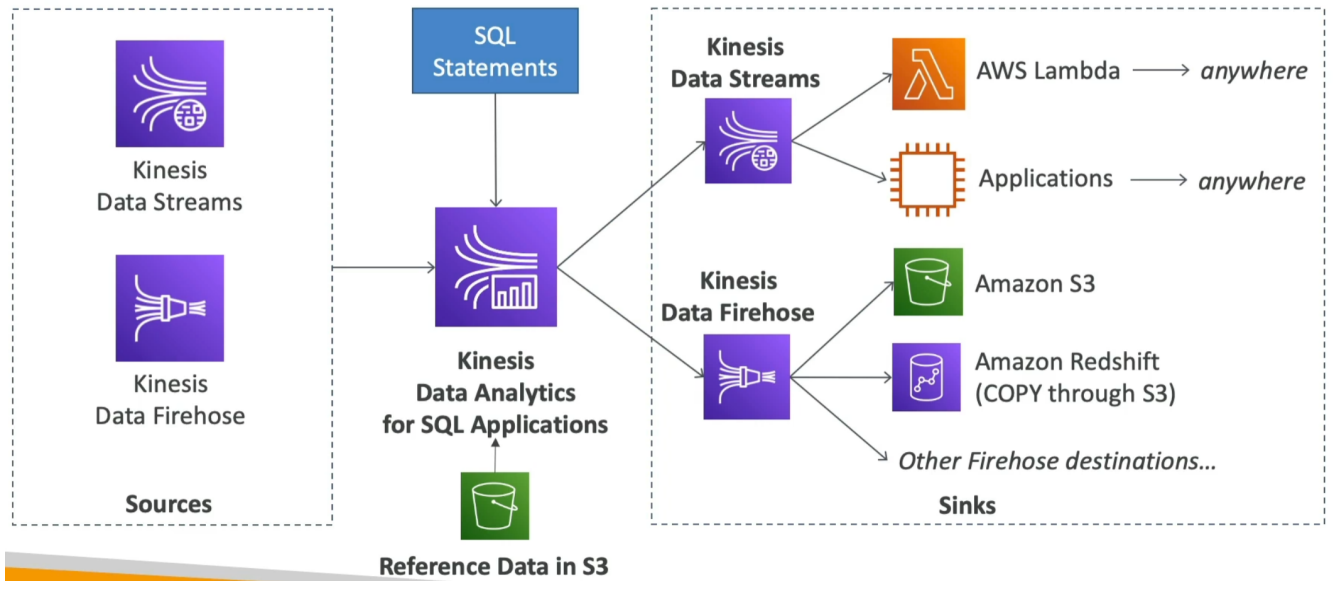
- **Data lake = central place to have all your data for analytics purposes**
- Fully managed service that makes it easy to setup a **data lake** in days
- Discover, cleanse, transform,and ingest data into your Data Lake
- It automates many complex manual steps (collecting, cleansing,moving, cataloging 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 ...
- **Fing-grained Access Control for your applications (row and column-level)**
- Built on top of AWS Glue



AWS Lake Formation Centralized Permissions Example



Kinesis Data Analytics for SQL applications



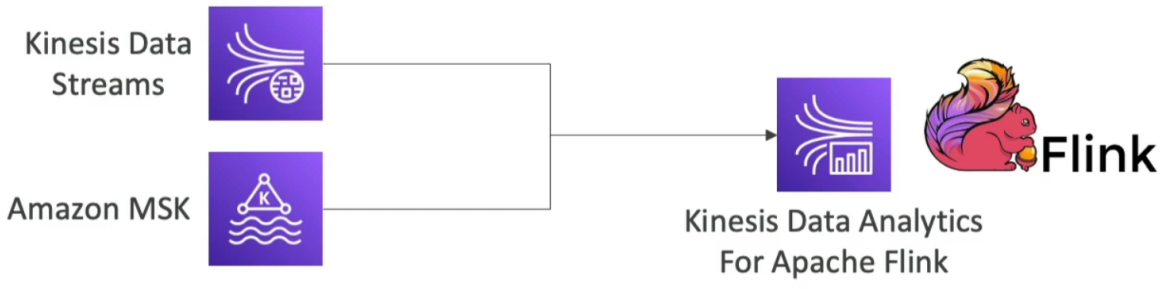
Kinesis Data Analytics (SQL application)

- 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 stream 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

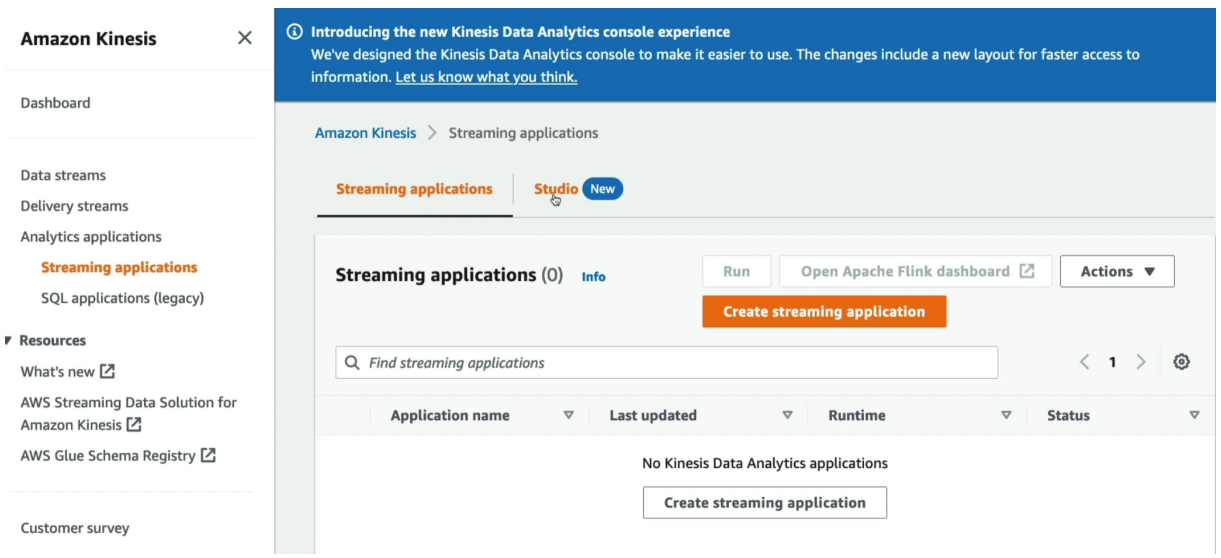
Kinesis Data Analytics for Apache Flink

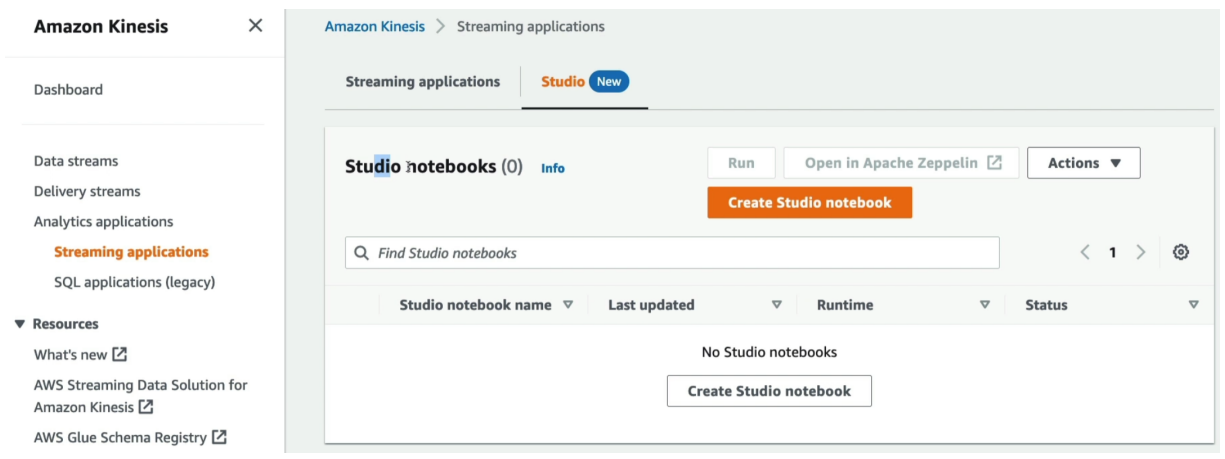
- Use Flink (Java, Scala or SQL) to process and analyze streaming data



- Run any Apache Flink applciation 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)

Hands on

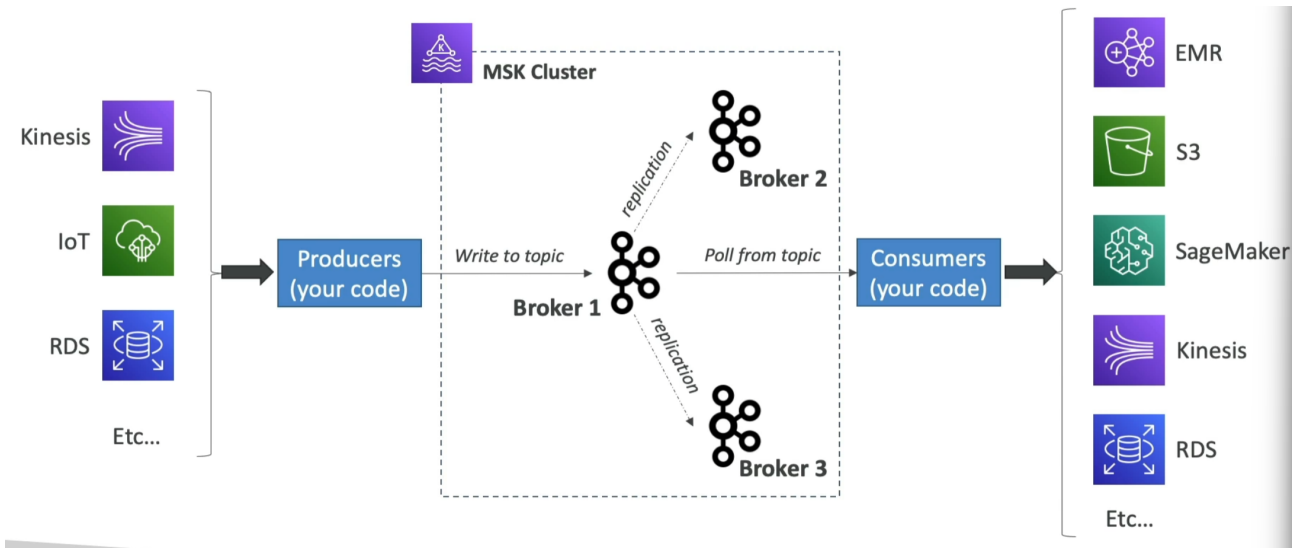




Amazon Managed Streaming for Apache Kafka (Amazon MSK)

- 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

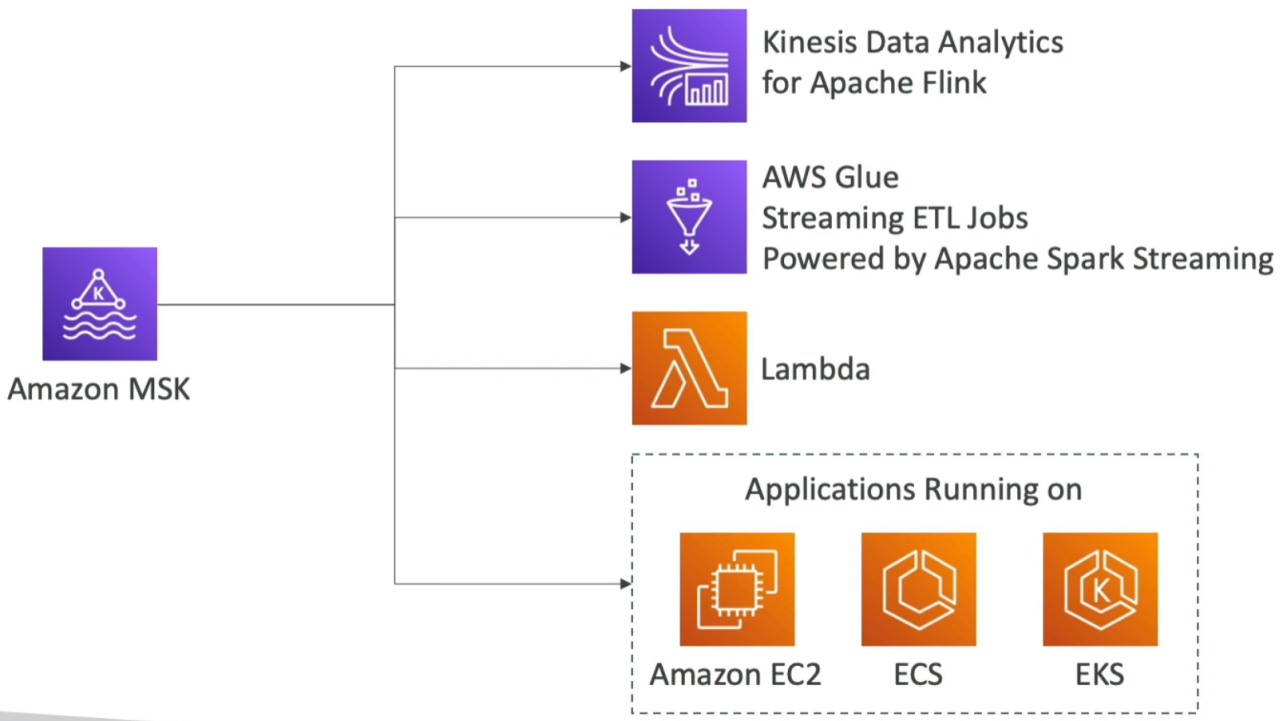
Kinesis Data Streams

- 1 MB message size limit
- Data Streams with Shards
- Shard Splitting & Merging
- TLS In-flight encryption
- KMS at-rest encryption

Amazon MSK

- 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

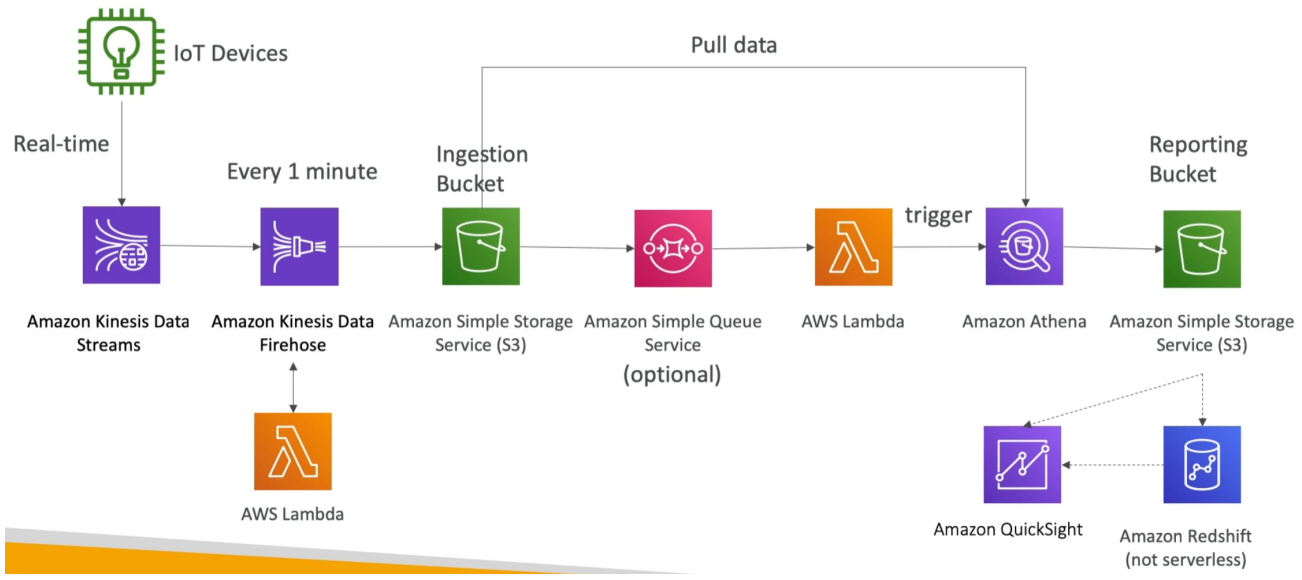
Amazon MSK Consumers



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 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 QuickSight, Redshift, etc ...

Amazon Rekognition

- Find **object,people,text,scenes** in **images and videos** using ML
- **Facial analysis** and **facial search** to do user verification, people counting
- 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)

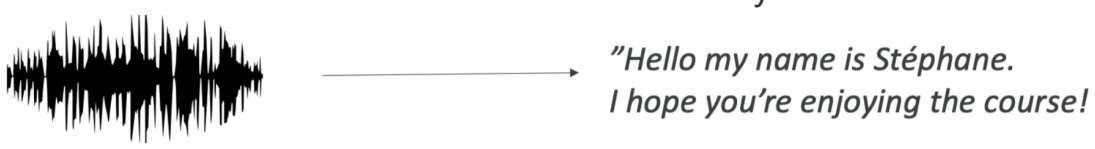
Amazon Rekognition - Content Moderation

- Detect content that is inappropriated, 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

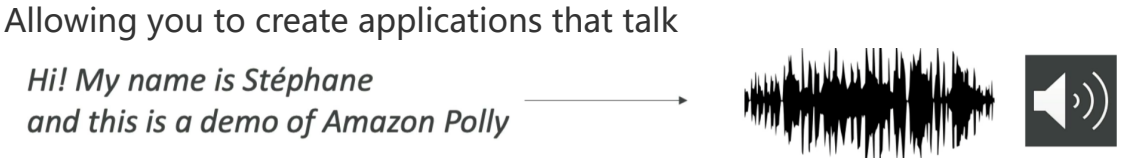
Amazon Transcribe

- 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

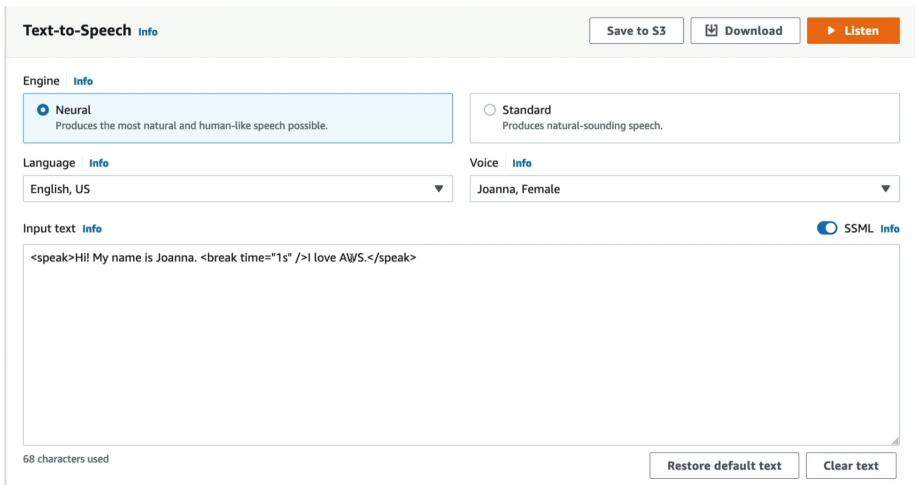


Amazon Polly - Lexicon & SSML

- 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 **SynthensizeSpeech** operation

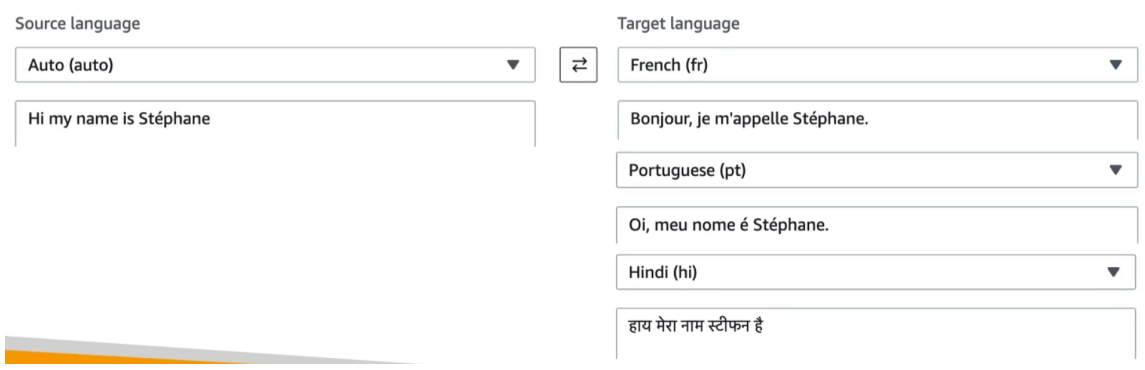
- Generate speech from plain text or from documents marked up with **Speech Synthesis Markup Language (SSML)** - enables more customization
  - emphasizing specific words or phrases
  - using phonetic pronunciation
  - including breathing sounds, whispering

- using the Newscaster speaking style

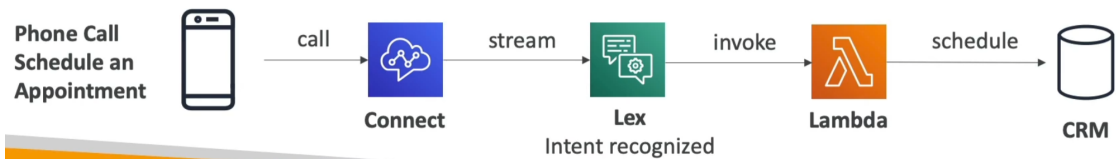


Amazon Translate

- Natural and ccurate **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 solutions
- 

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
- 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

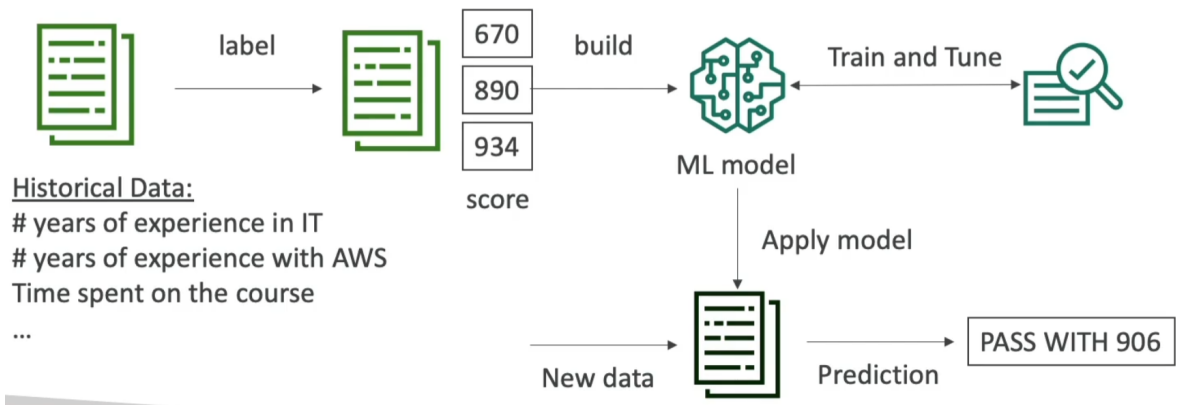
Amazon Comprehend Medical

- 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

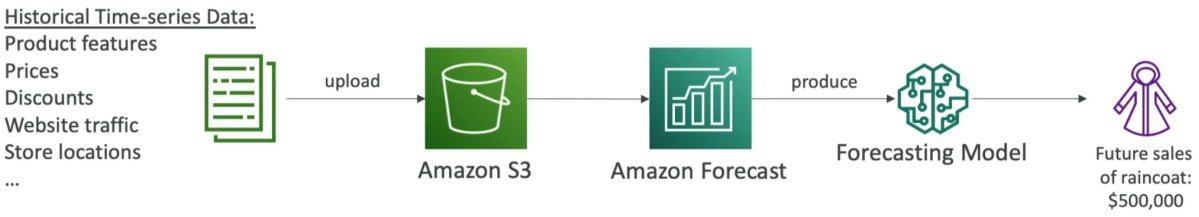
Amazon SageMaker

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



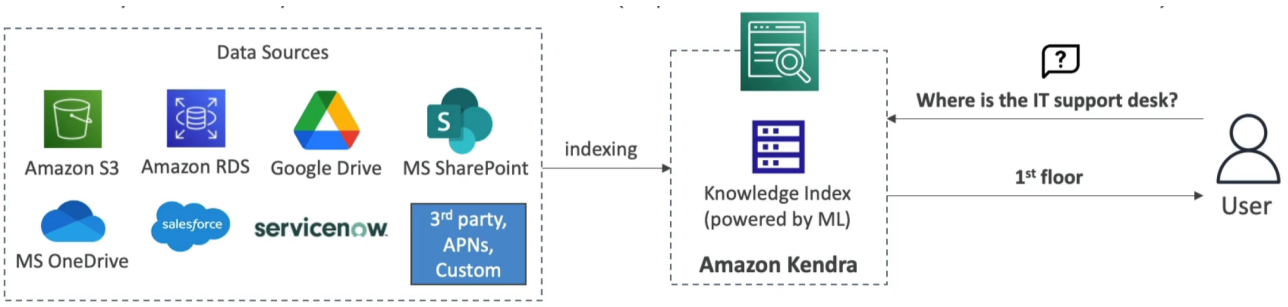
Amazon Forecast

- 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
- Learning from user interactions / feedback to promote preferred results (Incremental Learning)
- Ability to manually fine-tune search results (importance of data, freshness, custom,...)



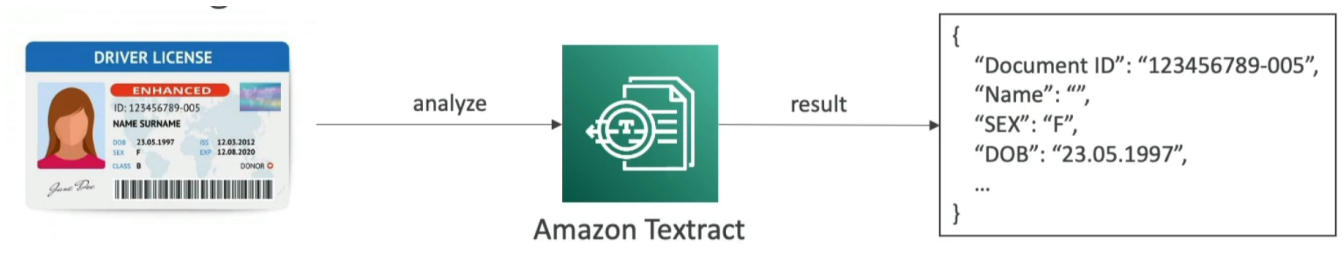
Amazon Personalize

- 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)
  - Healthycare (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
- **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