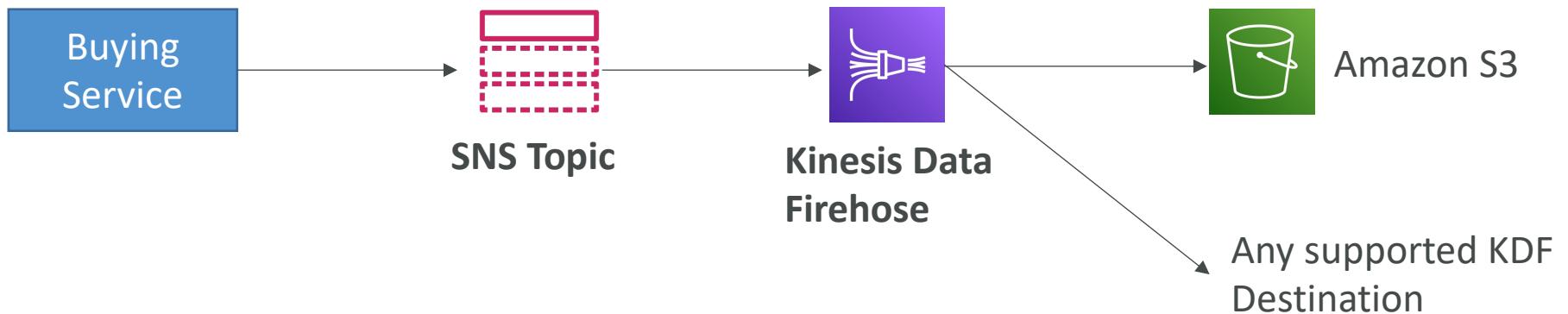


Application: SNS to Amazon S3 through Kinesis Data Firehose

- SNS can send to Kinesis and therefore we can have the following solutions architecture:



Amazon SNS – FIFO Topic

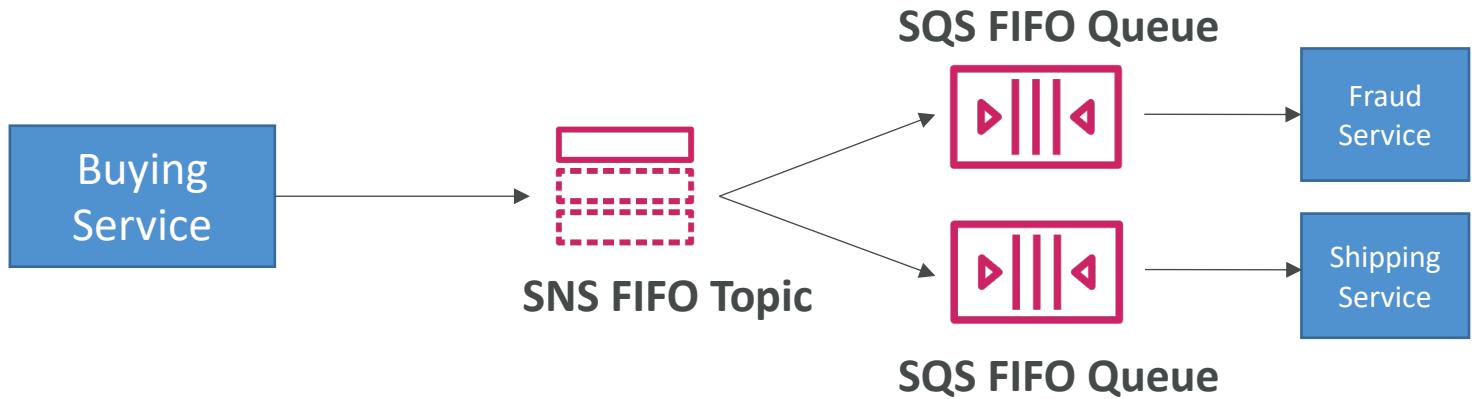
- FIFO = First In First Out (ordering of messages in the topic)



- Similar features as SQS FIFO:
 - Ordering by Message Group ID (all messages in the same group are ordered)
 - Deduplication using a Deduplication ID or Content Based Deduplication
- Can only have SQS FIFO queues as subscribers
- Limited throughput (same throughput as SQS FIFO)

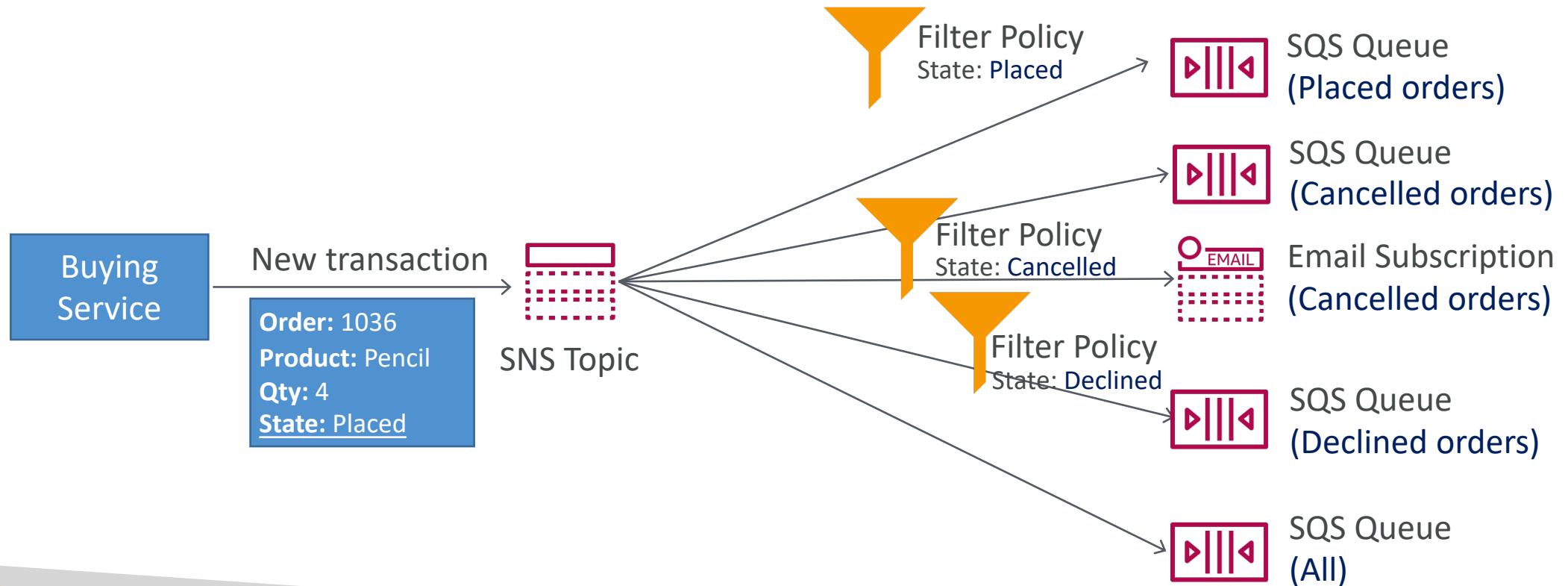
SNS FIFO + SQS FIFO: Fan Out

- In case you need fan out + ordering + deduplication



SNS – Message Filtering

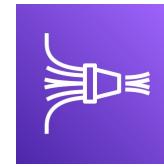
- JSON policy used to filter messages sent to SNS topic's subscriptions
- If a subscription doesn't have a filter policy, it receives every message



Kinesis Overview

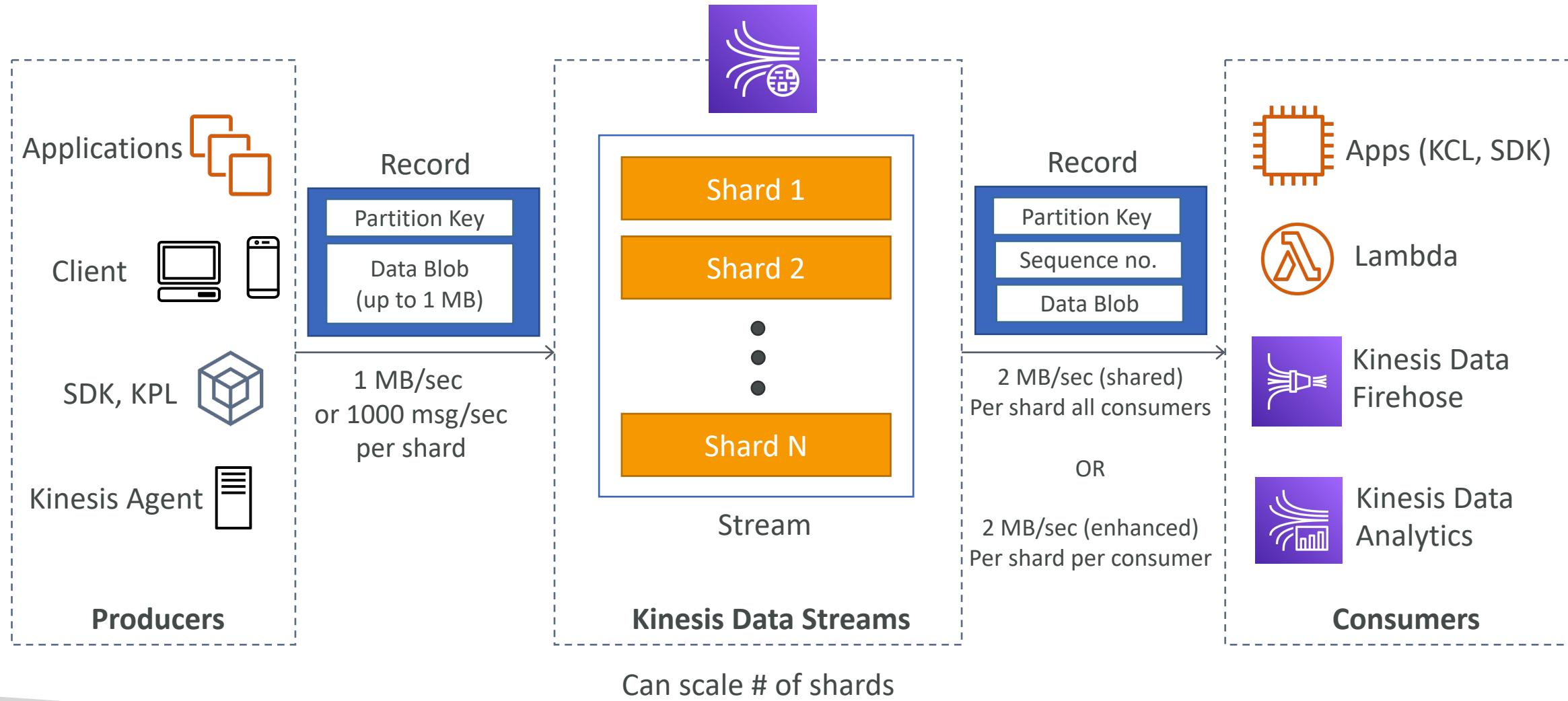


- Makes it easy to **collect, process, and analyze** streaming data in real-time
- Ingest real-time data such as: Application logs, Metrics, Website clickstreams, IoT telemetry data...



- **Kinesis Data Streams:** capture, process, and store data streams
- **Kinesis Data Firehose:** load data streams into AWS data stores
- **Kinesis Data Analytics:** analyze data streams with SQL or Apache Flink
- **Kinesis Video Streams:** capture, process, and store video streams

Kinesis Data Streams





Kinesis Data Streams

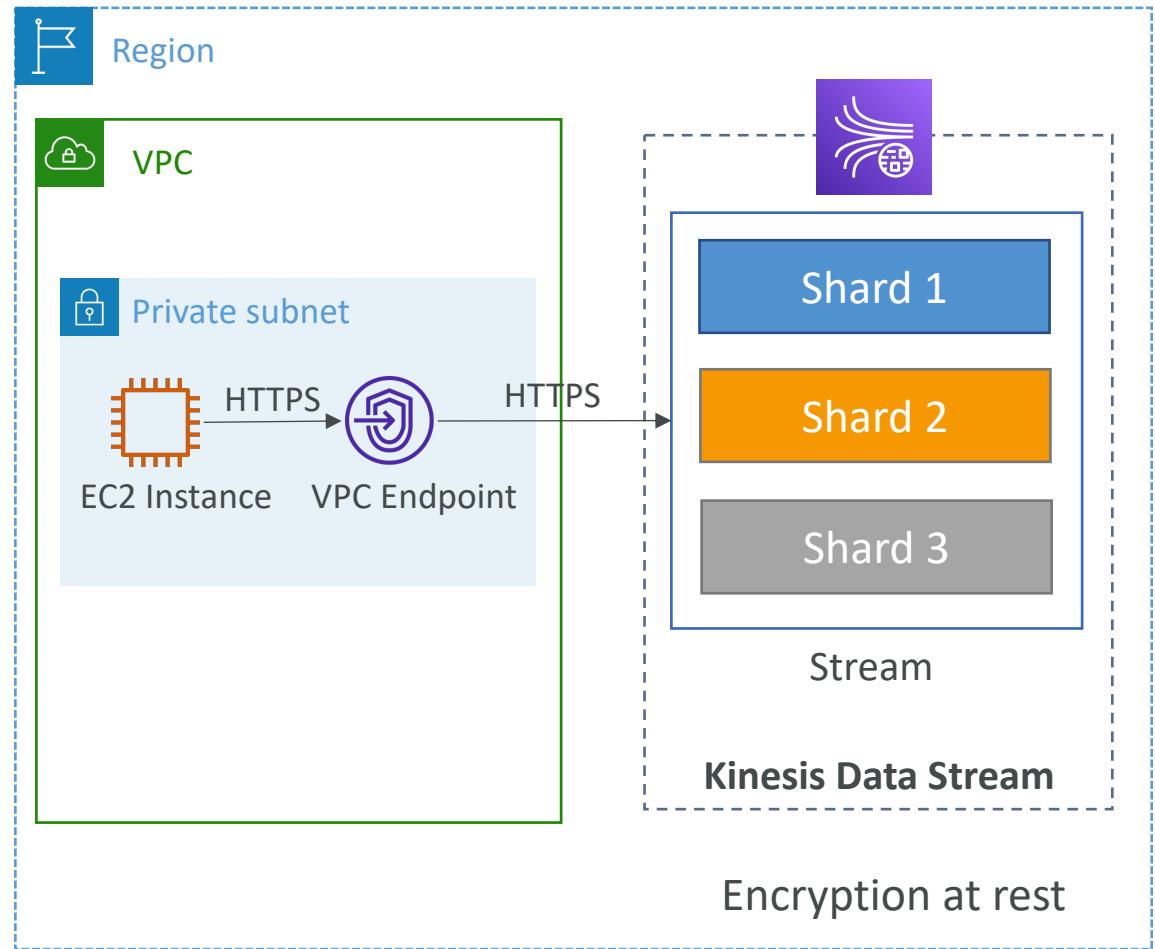
- Retention between 1 day to 365 days
- Ability to reprocess (replay) data
- Once data is inserted in Kinesis, it can't be deleted (immutability)
- Data that shares the same partition goes to the same shard (ordering)
- Producers: AWS SDK, Kinesis Producer Library (KPL), Kinesis Agent
- Consumers:
 - Write your own: Kinesis Client Library (KCL), AWS SDK
 - Managed: AWS Lambda, Kinesis Data Firehose, Kinesis Data Analytics,

Kinesis Data Streams – Capacity Modes

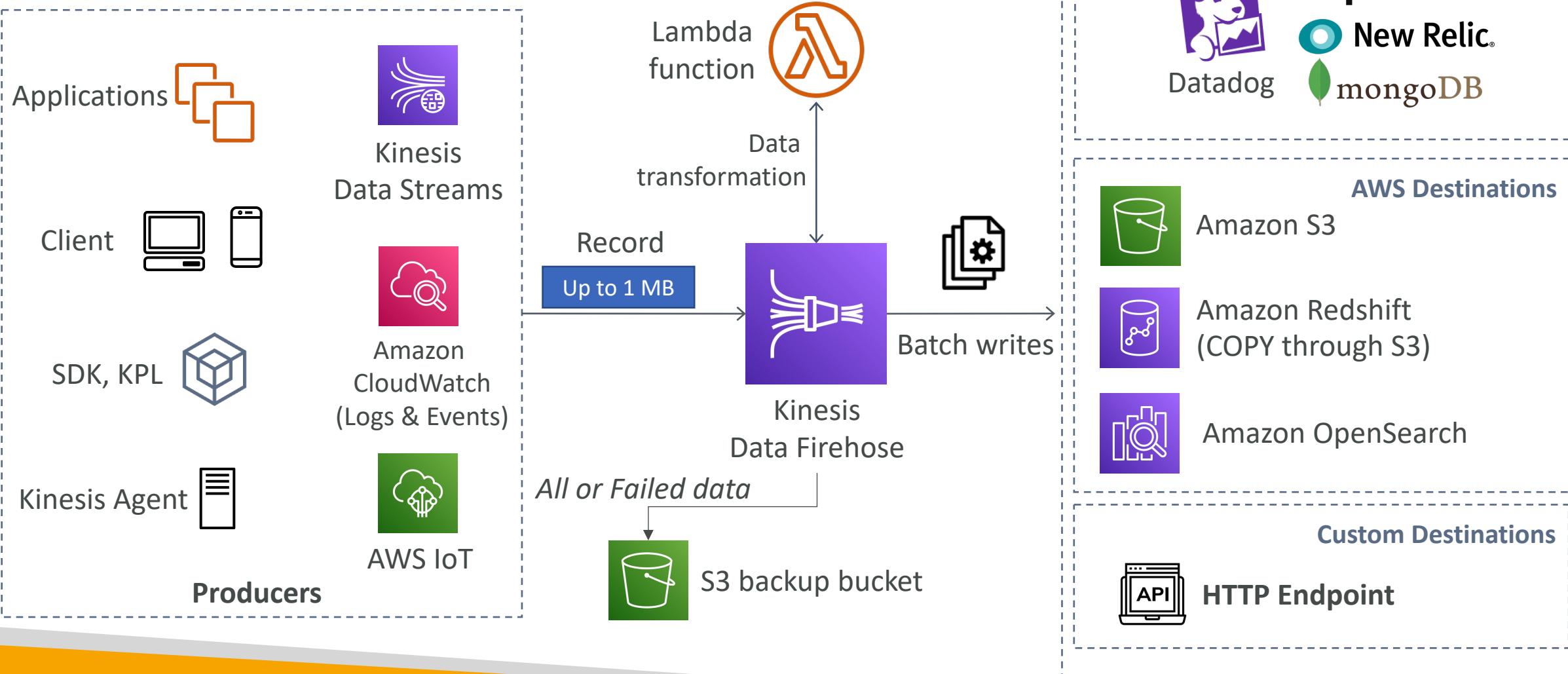
- **Provisioned mode:**
 - You choose the number of shards provisioned, scale manually or using API
 - Each shard gets 1MB/s in (or 1000 records per second)
 - Each shard gets 2MB/s out (classic or enhanced fan-out consumer)
 - You pay per shard provisioned per hour
- **On-demand mode:**
 - No need to provision or manage the capacity
 - Default capacity provisioned (4 MB/s in or 4000 records per second)
 - Scales automatically based on observed throughput peak during the last 30 days
 - Pay per stream per hour & data in/out per GB

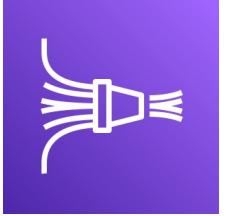
Kinesis Data Streams Security

- Control access / authorization using IAM policies
- Encryption in flight using HTTPS endpoints
- Encryption at rest using KMS
- You can implement encryption/decryption of data on client side (harder)
- VPC Endpoints available for Kinesis to access within VPC
- Monitor API calls using CloudTrail



Kinesis Data Firehose





Kinesis Data Firehose

- Fully Managed Service, no administration, automatic scaling, serverless
 - AWS: Redshift / Amazon S3 / OpenSearch
 - 3rd party partner: Splunk / MongoDB / DataDog / NewRelic / ...
 - Custom: send to any HTTP endpoint
- Pay for data going through Firehose
- **Near Real Time**
 - 60 seconds latency minimum for non full batches
 - Or minimum 1 MB of data at a time
- Supports many data formats, conversions, transformations, compression
- Supports custom data transformations using AWS Lambda
- Can send failed or all data to a backup S3 bucket

Kinesis Data Streams vs Firehose



Kinesis Data Streams

- Streaming service for ingest at scale
- Write custom code (producer / consumer)
- Real-time (~200 ms)
- Manage scaling (shard splitting / merging)
- Data storage for 1 to 365 days
- Supports replay capability

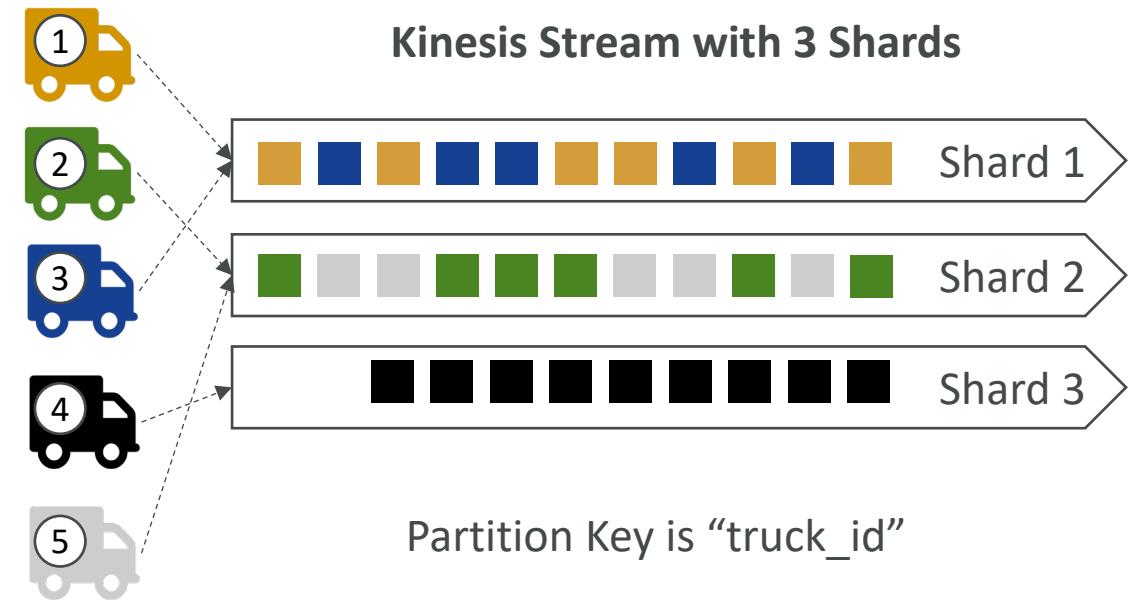


Kinesis Data Firehose

- Load streaming data into S3 / Redshift / OpenSearch / 3rd party / custom HTTP
- Fully managed
- Near real-time (buffer time min. 60 sec)
- Automatic scaling
- No data storage
- Doesn't support replay capability

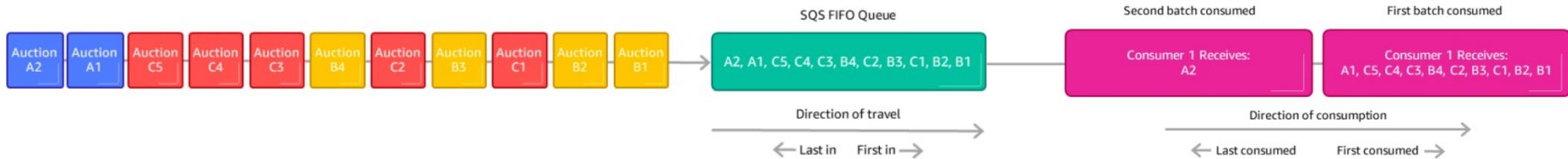
Ordering data into Kinesis

- Imagine you have 100 trucks (truck_1, truck_2, ... truck_100) on the road sending their GPS positions regularly into AWS.
- You want to consume the data in order for each truck, so that you can track their movement accurately.
- How should you send that data into Kinesis?
- Answer: send using a “Partition Key” value of the “truck_id”
- The same key will always go to the same shard

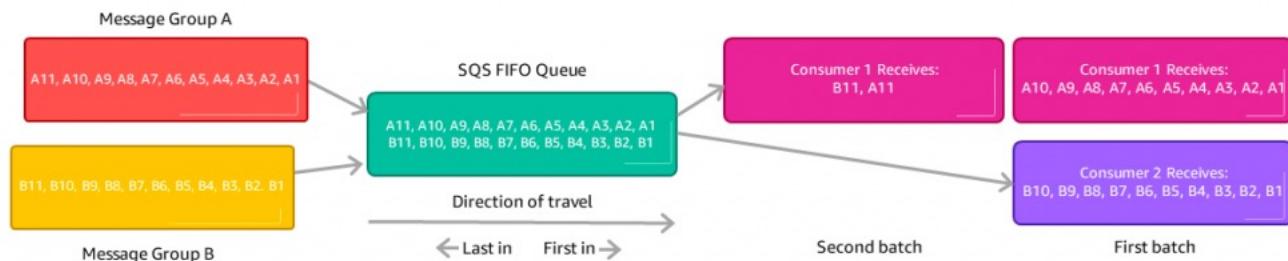


Ordering data into SQS

- For SQS standard, there is no ordering.
- For SQS FIFO, if you don't use a Group ID, messages are consumed in the order they are sent, with **only one consumer**



- You want to scale the number of consumers, but you want messages to be “grouped” when they are related to each other
- Then you use a Group ID (similar to Partition Key in Kinesis)



Kinesis vs SQS ordering

- Let's assume 100 trucks, 5 kinesis shards, 1 SQS FIFO
- Kinesis Data Streams:
 - On average you'll have 20 trucks per shard
 - Trucks will have their data ordered within each shard
 - The maximum amount of consumers in parallel we can have is 5
 - Can receive up to 5 MB/s of data
- SQS FIFO
 - You only have one SQS FIFO queue
 - You will have 100 Group ID
 - You can have up to 100 Consumers (due to the 100 Group ID)
 - You have up to 300 messages per second (or 3000 if using batching)

SQS vs SNS vs Kinesis

SQS:

- Consumer “pull data”
- Data is deleted after being consumed
- Can have as many workers (consumers) as we want
- No need to provision throughput
- Ordering guarantees only on FIFO queues
- Individual message delay capability



SNS:

- Push data to many subscribers
- Up to 12,500,000 subscribers
- Data is not persisted (lost if not delivered)
- Pub/Sub
- Up to 100,000 topics
- No need to provision throughput
- Integrates with SQS for fan-out architecture pattern
- FIFO capability for SQS FIFO

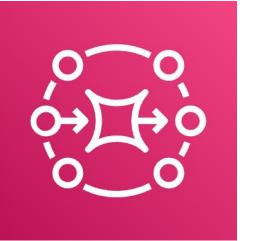


Kinesis:

- Standard: pull data
 - 2 MB per shard
- Enhanced-fan out: push data
 - 2 MB per shard per consumer
- Possibility to replay data
- Meant for real-time big data, analytics and ETL
- Ordering at the shard level
- Data expires after X days
- Provisioned mode or on-demand capacity mode



Amazon MQ

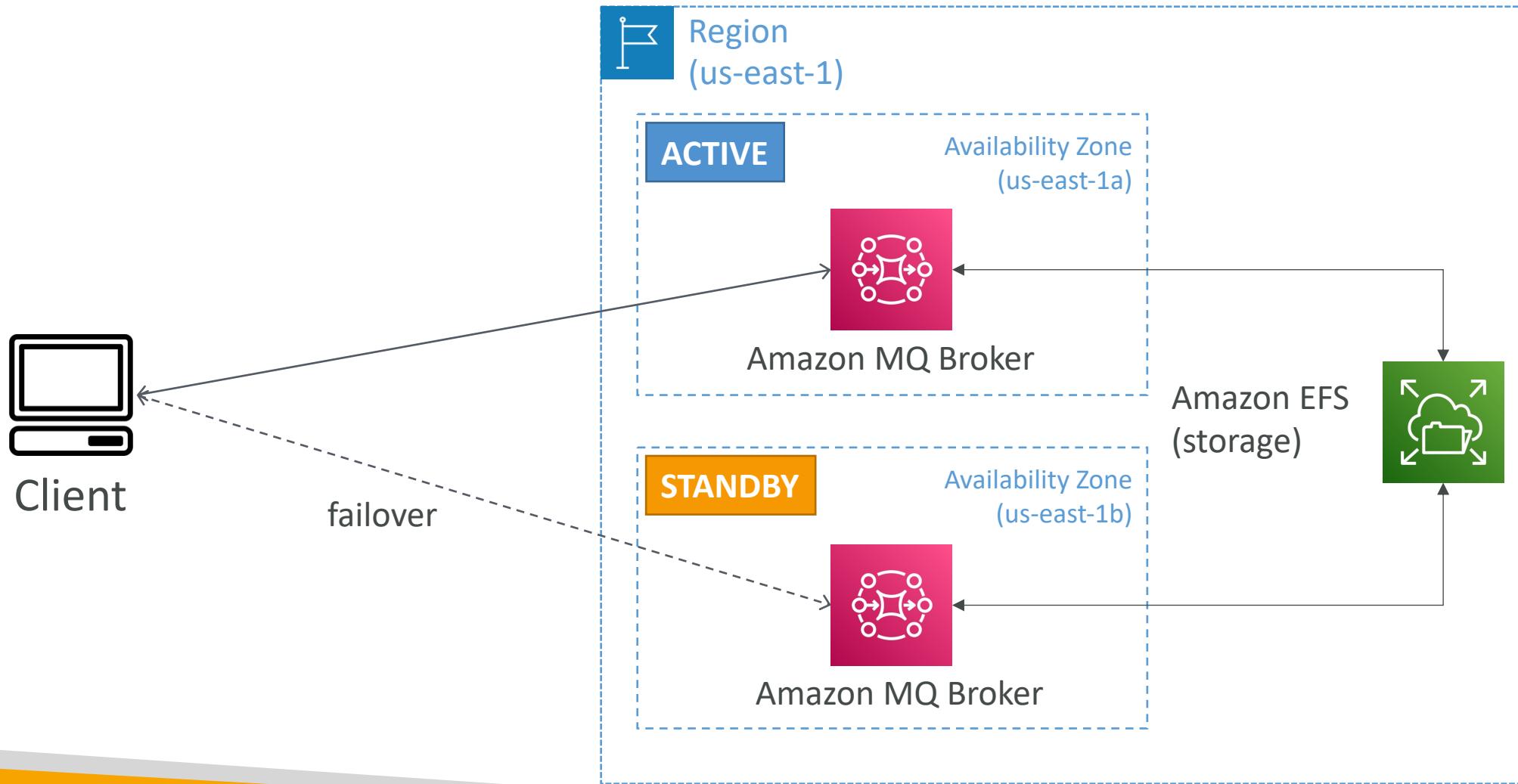


- SQS, SNS are “cloud-native” services: proprietary protocols from AWS
- Traditional applications running from on-premises may use open protocols such as: MQTT, AMQP, STOMP, Openwire, WSS
- When migrating to the cloud, instead of re-engineering the application to use SQS and SNS, we can use Amazon MQ
- Amazon MQ is a managed message broker service for



- Amazon MQ doesn’t “scale” as much as SQS / SNS
- Amazon MQ runs on servers, can run in Multi-AZ with failover
- Amazon MQ has both queue feature (~SQS) and topic features (~SNS)

Amazon MQ – High Availability



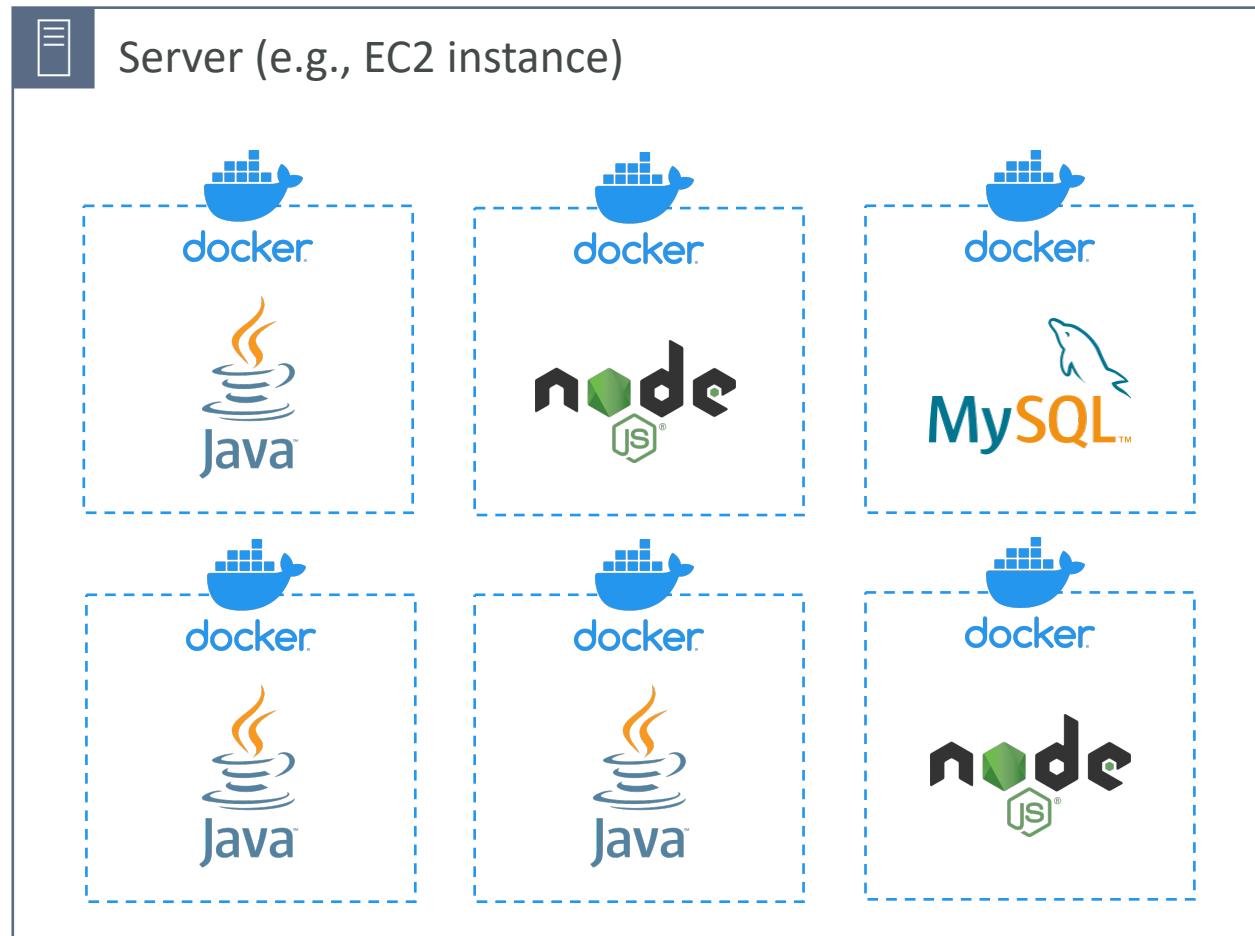
Container Section



What is Docker?

- Docker is a software development platform to deploy apps
- Apps are packaged in **containers** that can be run on any OS
- Apps run the same, regardless of where they're run
 - Any machine
 - No compatibility issues
 - Predictable behavior
 - Less work
 - Easier to maintain and deploy
 - Works with any language, any OS, any technology
- Use cases: microservices architecture, lift-and-shift apps from on-premises to the AWS cloud, ...

Docker on an OS

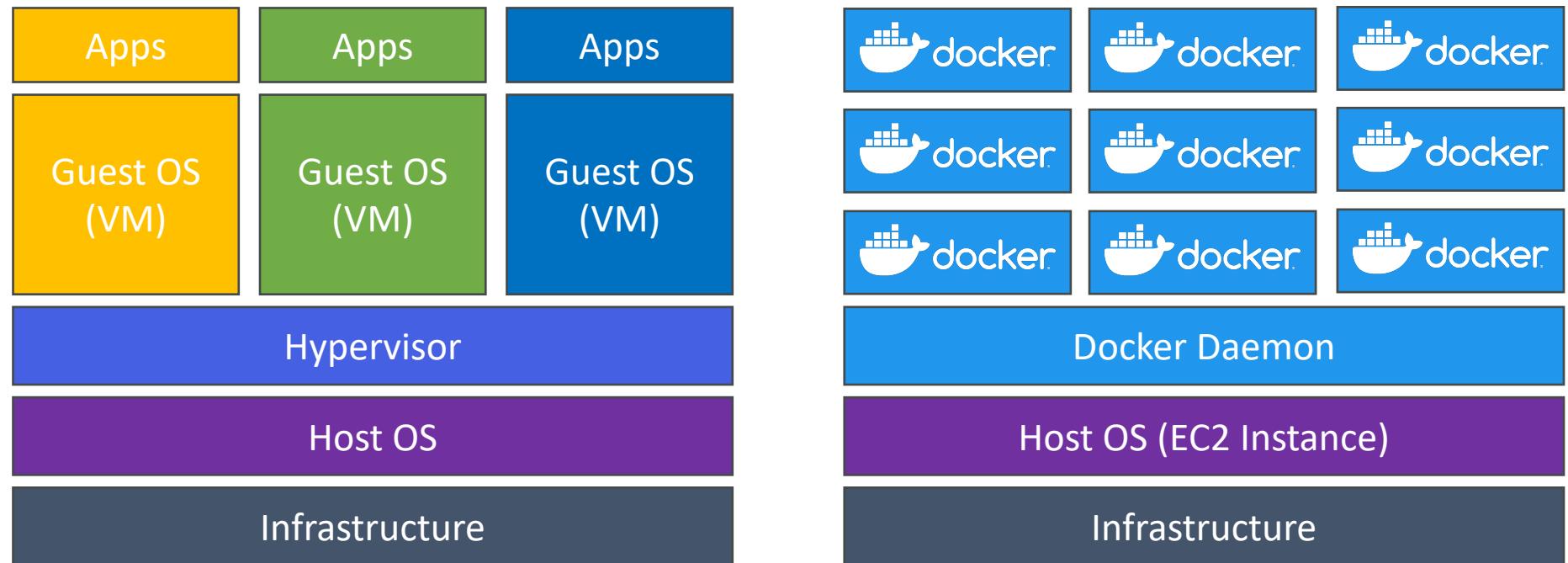


Where are Docker images stored?

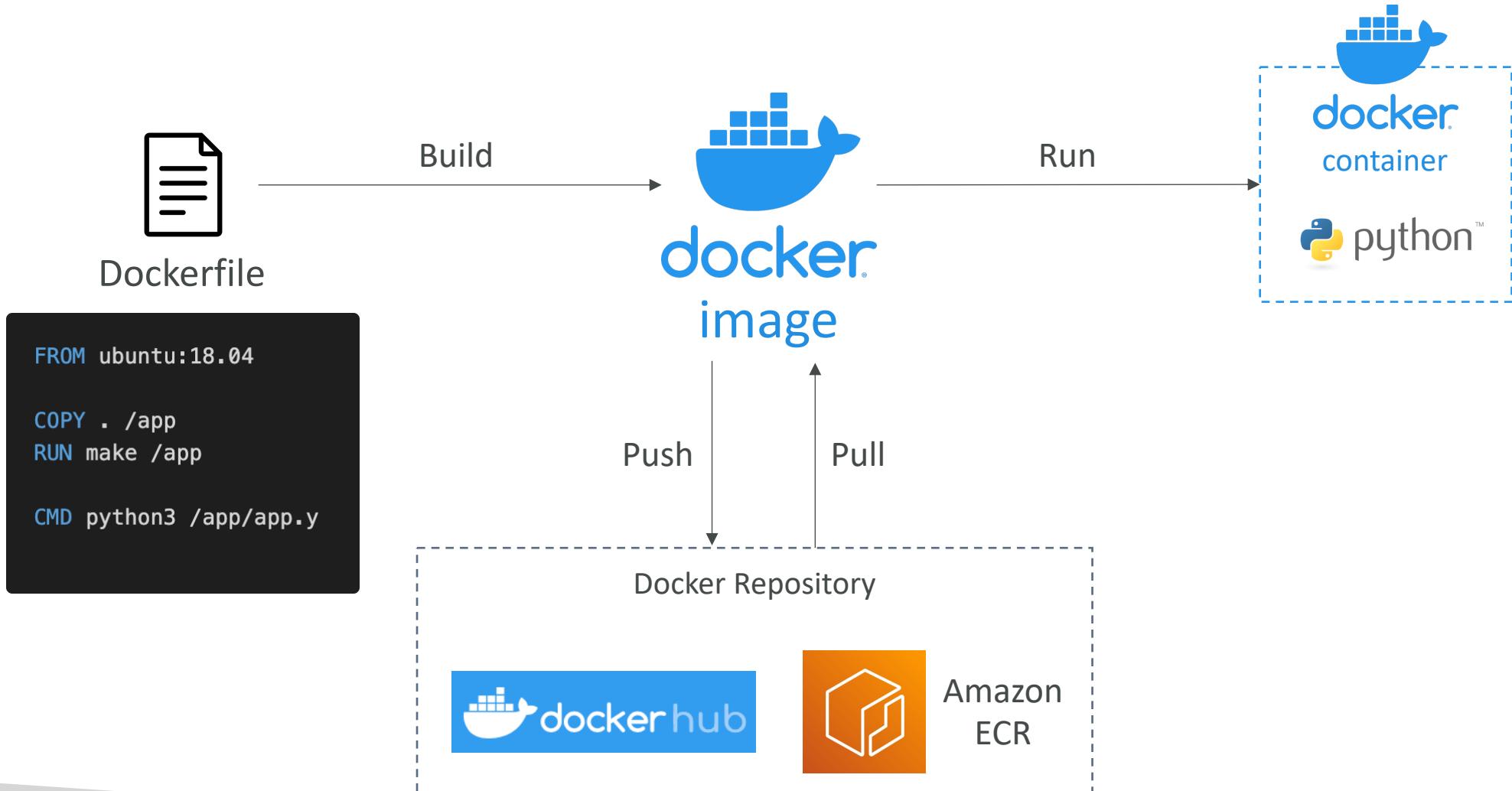
- Docker images are stored in Docker Repositories
- Docker Hub (<https://hub.docker.com>)
 - Public repository
 - Find base images for many technologies or OS (e.g., Ubuntu, MySQL, ...)
- Amazon ECR (Amazon Elastic Container Registry)
 - Private repository
 - Public repository (Amazon ECR Public Gallery <https://gallery.ecr.aws>)

Docker vs. Virtual Machines

- Docker is "sort of" a virtualization technology, but not exactly
- Resources are shared with the host => many containers on one server



Getting Started with Docker



Docker Containers Management on AWS

- Amazon Elastic Container Service (Amazon ECS)
 - Amazon's own container platform
- Amazon Elastic Kubernetes Service (Amazon EKS)
 - Amazon's managed Kubernetes (open source)
- AWS Fargate
 - Amazon's own Serverless container platform
 - Works with ECS and with EKS
- Amazon ECR:
 - Store container images



Amazon ECS



Amazon EKS



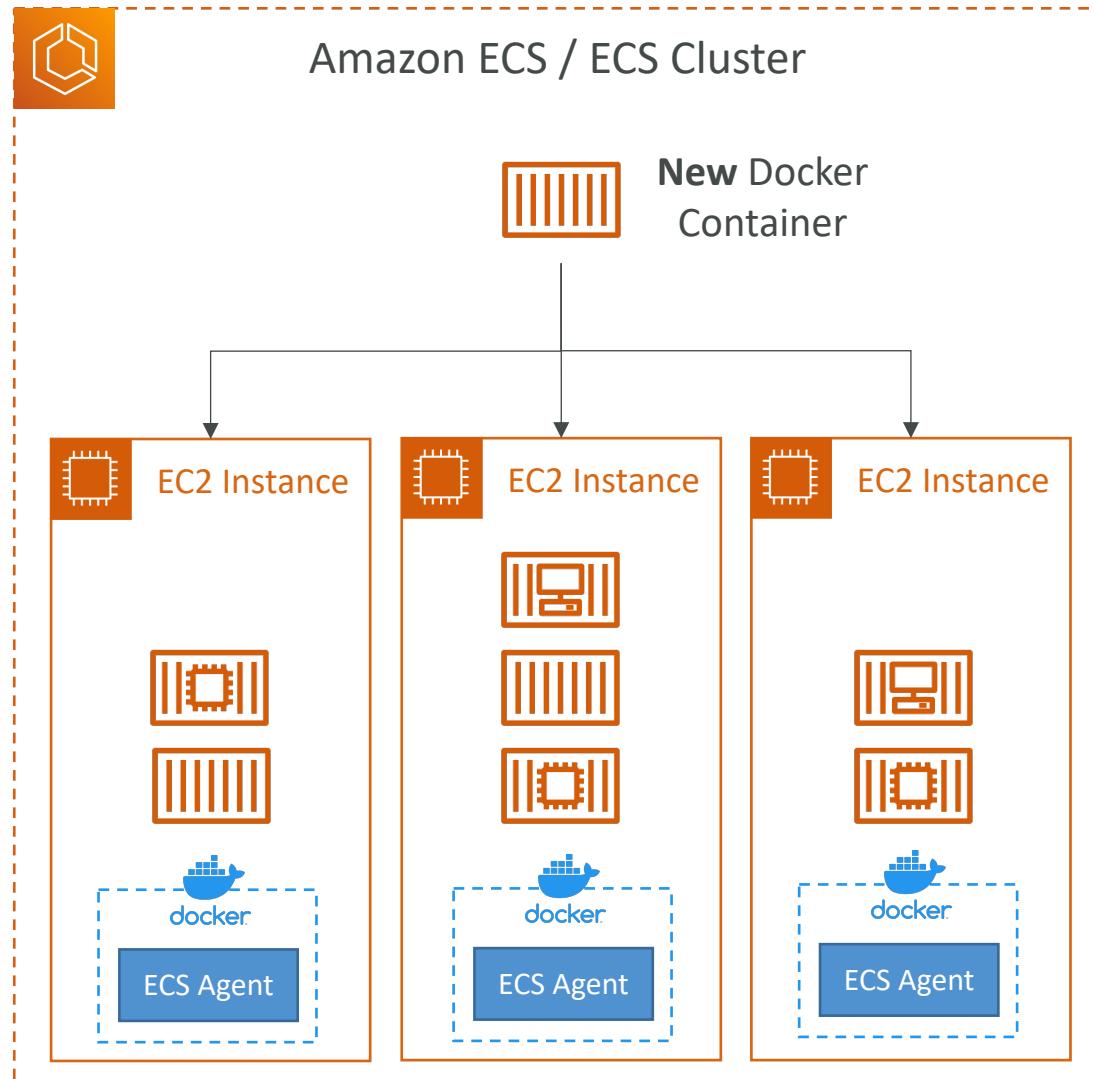
AWS Fargate



Amazon ECR

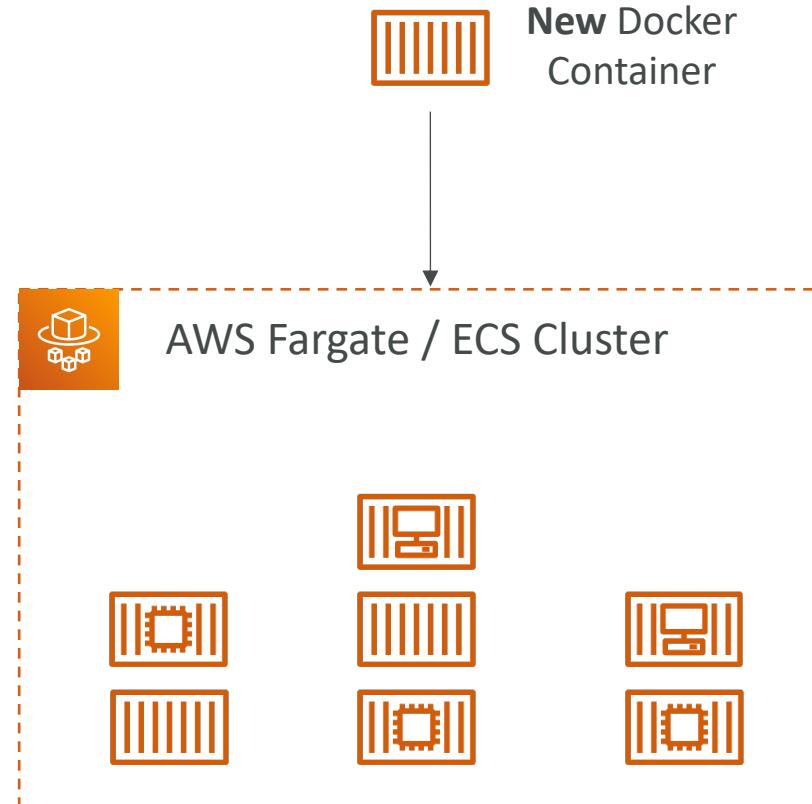
Amazon ECS - EC2 Launch Type

- ECS = Elastic Container Service
- Launch Docker containers on AWS = Launch **ECS Tasks** on ECS Clusters
- **EC2 Launch Type:** you must provision & maintain the infrastructure (the EC2 instances)
- Each EC2 Instance must run the ECS Agent to register in the ECS Cluster
- AWS takes care of starting / stopping containers



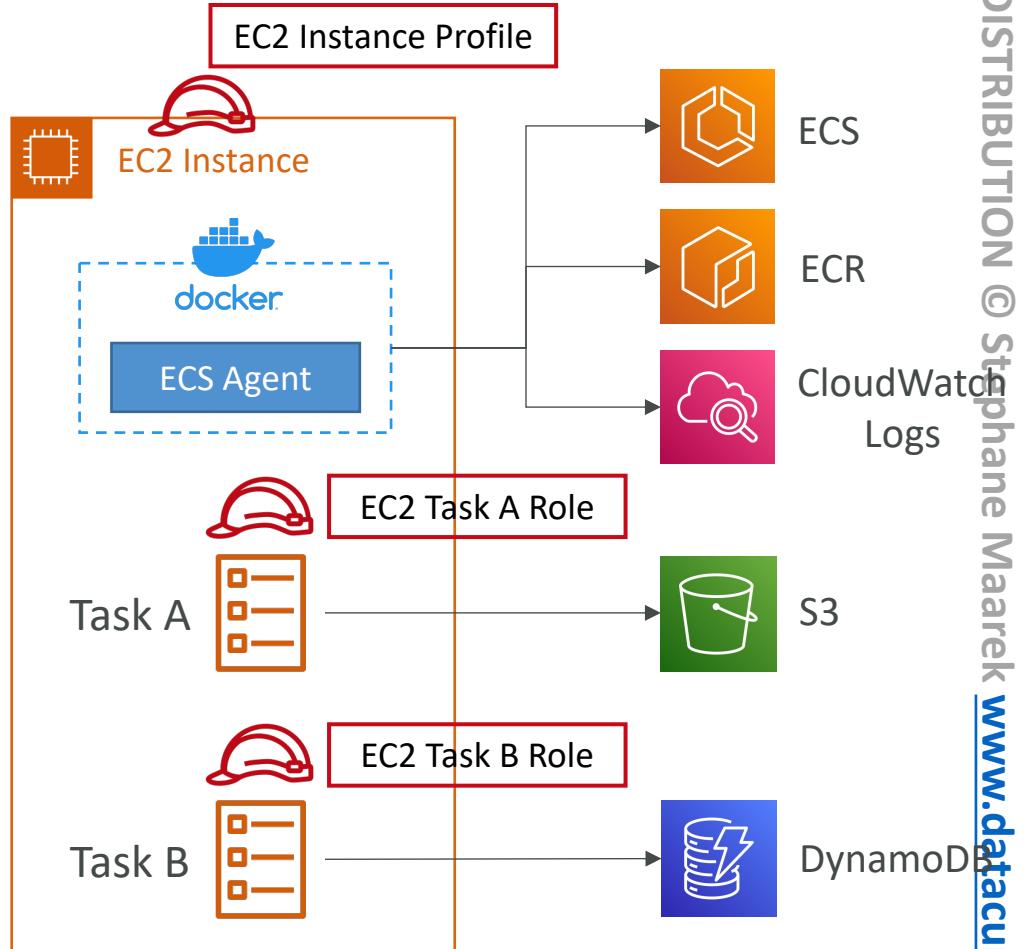
Amazon ECS – Fargate Launch Type

- Launch Docker containers on AWS
- You do not provision the infrastructure
(no EC2 instances to manage)
- It's all Serverless!
- You just create task definitions
- AWS just runs ECS Tasks for you based
on the CPU / RAM you need
- To scale, just increase the number of
tasks. Simple - no more EC2 instances



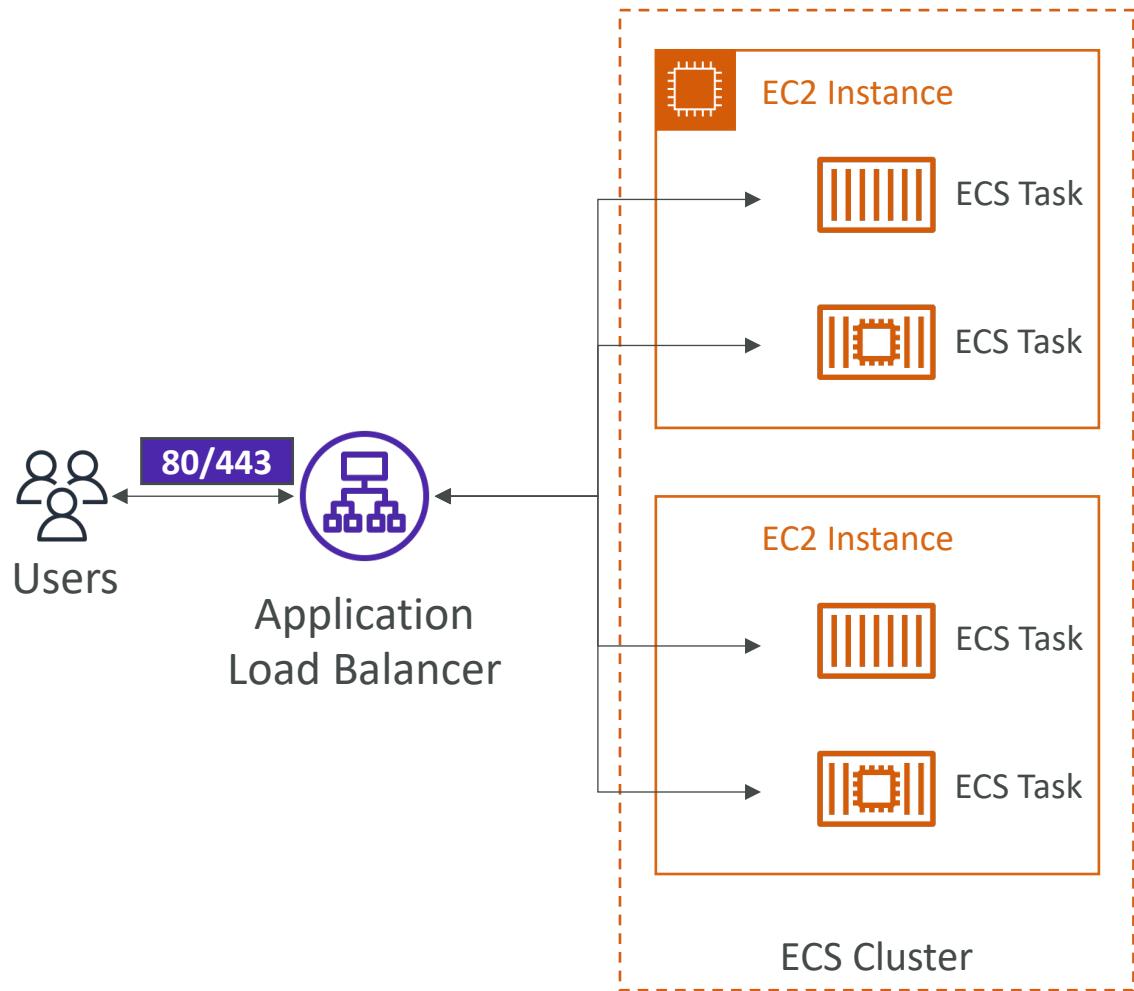
Amazon ECS – IAM Roles for ECS

- **EC2 Instance Profile (EC2 Launch Type only):**
 - Used by the ECS agent
 - Makes API calls to ECS service
 - Send container logs to CloudWatch Logs
 - Pull Docker image from ECR
 - Reference sensitive data in Secrets Manager or SSM Parameter Store
- **ECS Task Role:**
 - Allows each task to have a specific role
 - Use different roles for the different ECS Services you run
 - Task Role is defined in the task definition



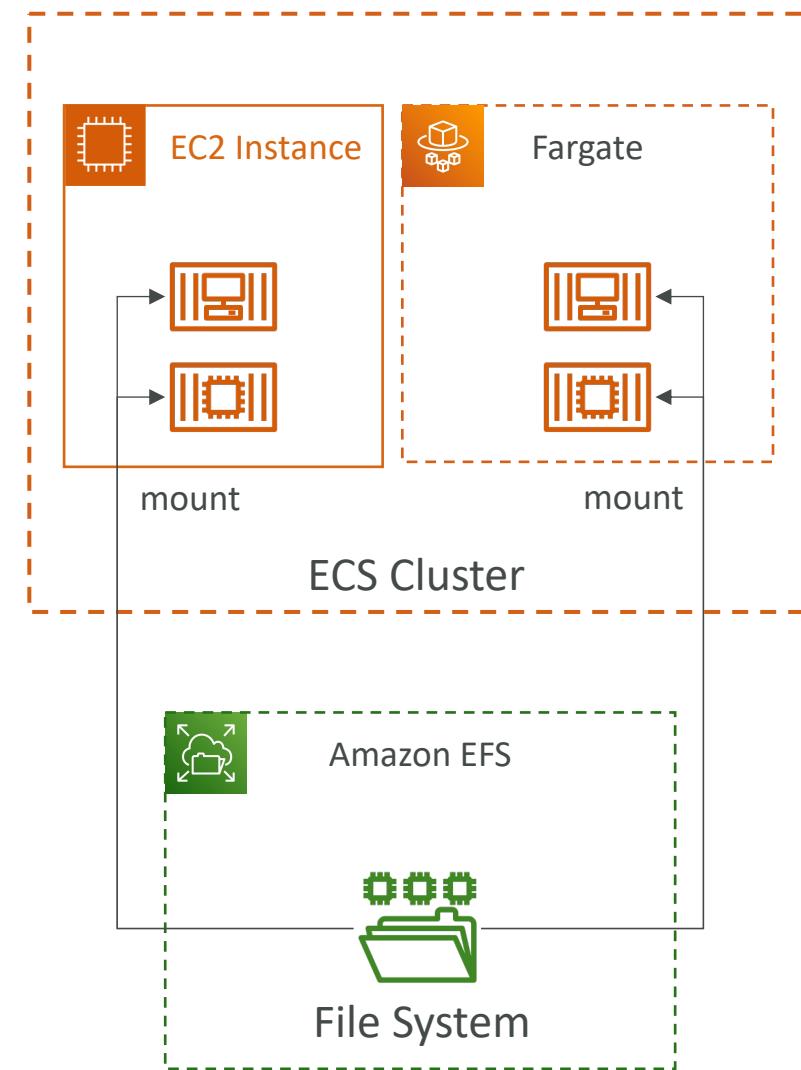
Amazon ECS – Load Balancer Integrations

- **Application Load Balancer** supported and works for most use cases
- **Network Load Balancer** recommended only for high throughput / high performance use cases, or to pair it with AWS Private Link
- **Classic Load Balancer** supported but not recommended (no advanced features – no Fargate)

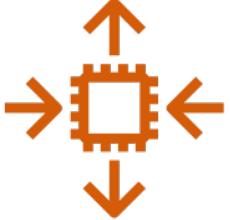


Amazon ECS – Data Volumes (EFS)

- Mount EFS file systems onto ECS tasks
- Works for both **EC2** and **Fargate** launch types
- Tasks running in any AZ will share the same data in the EFS file system
- **Fargate + EFS = Serverless**
- Use cases: persistent multi-AZ shared storage for your containers
- Note:
 - Amazon S3 cannot be mounted as a file system



ECS Service Auto Scaling

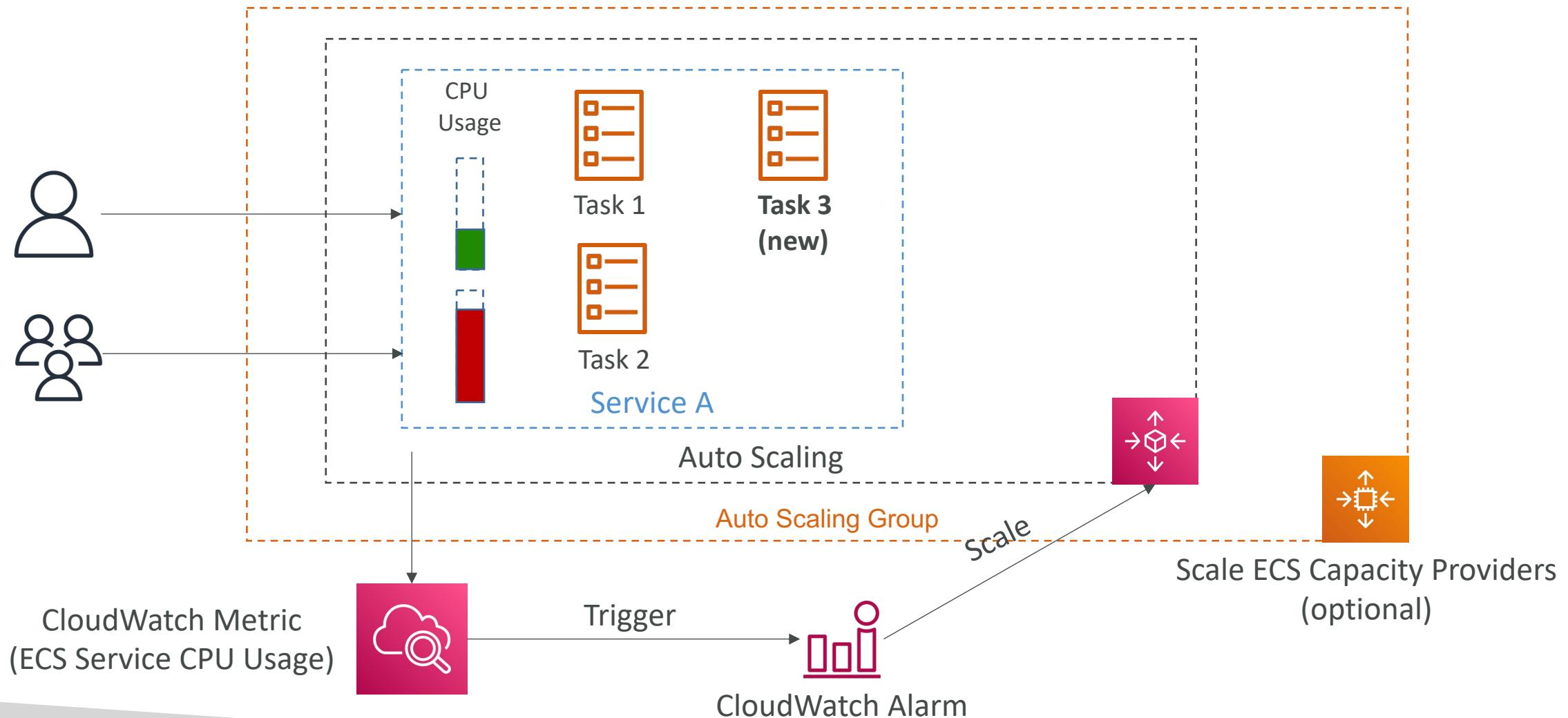


- Automatically increase/decrease the desired number of ECS tasks
- Amazon ECS Auto Scaling uses **AWS Application Auto Scaling**
 - ECS Service Average CPU Utilization
 - ECS Service Average Memory Utilization - Scale on RAM
 - ALB Request Count Per Target – metric coming from the ALB
- **Target Tracking** – scale based on target value for a specific CloudWatch metric
- **Step Scaling** – scale based on a specified CloudWatch Alarm
- **Scheduled Scaling** – scale based on a specified date/time (predictable changes)
- ECS Service Auto Scaling (task level) **≠** EC2 Auto Scaling (EC2 instance level)
- Fargate Auto Scaling is much easier to setup (because **Serverless**)

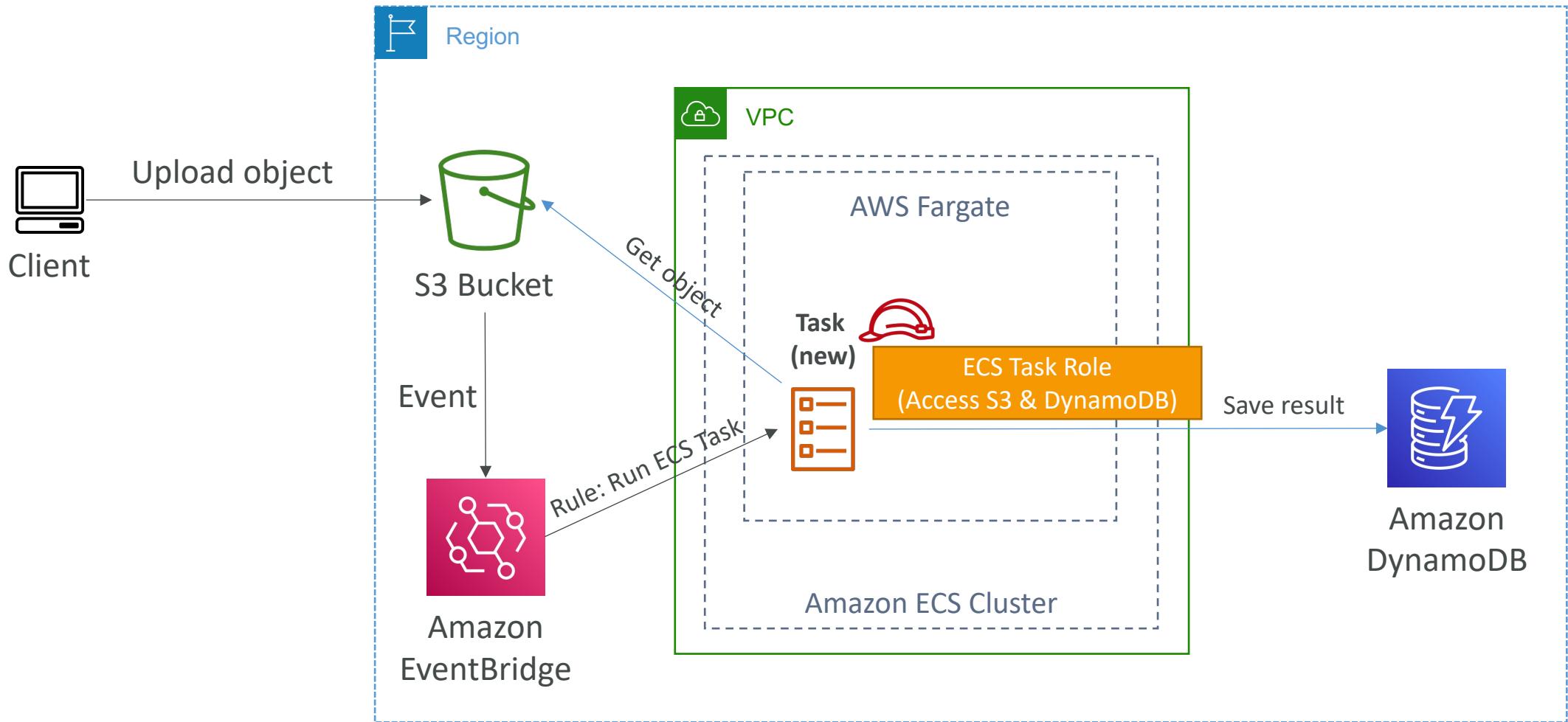
EC2 Launch Type – Auto Scaling EC2 Instances

- Accommodate ECS Service Scaling by adding underlying EC2 Instances
- **Auto Scaling Group Scaling**
 - Scale your ASG based on CPU Utilization
 - Add EC2 instances over time
- **ECS Cluster Capacity Provider**
 - Used to automatically provision and scale the infrastructure for your ECS Tasks
 - Capacity Provider paired with an Auto Scaling Group
 - Add EC2 Instances when you're missing capacity (CPU, RAM...)

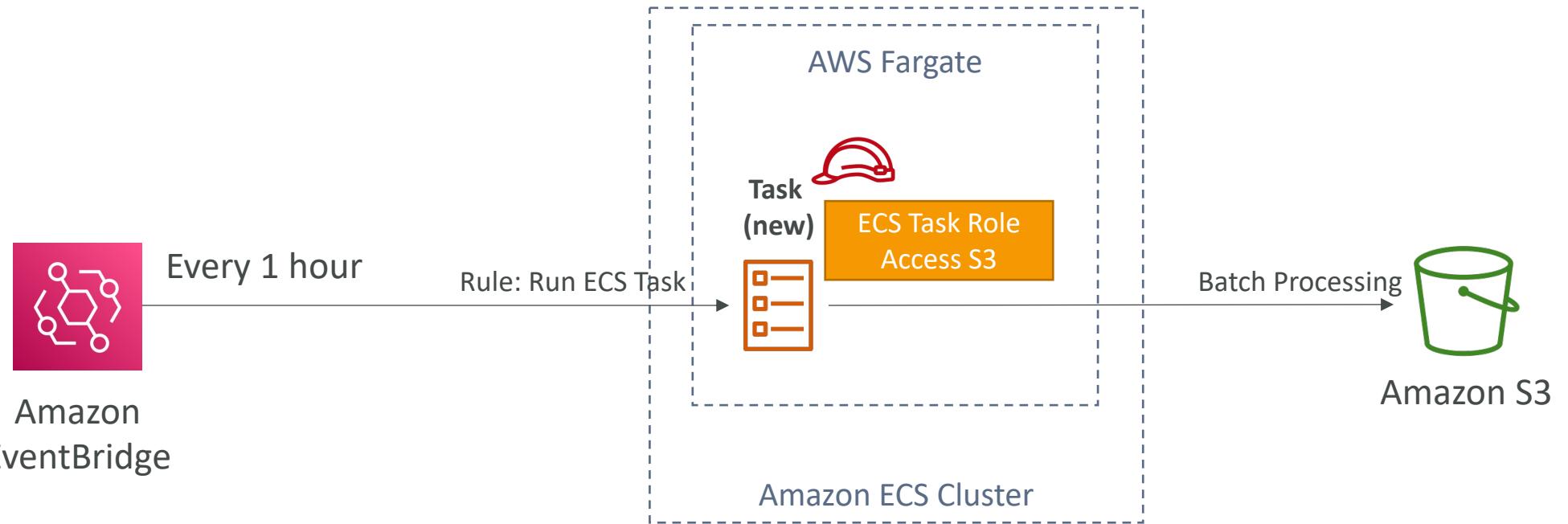
ECS Scaling – Service CPU Usage Example



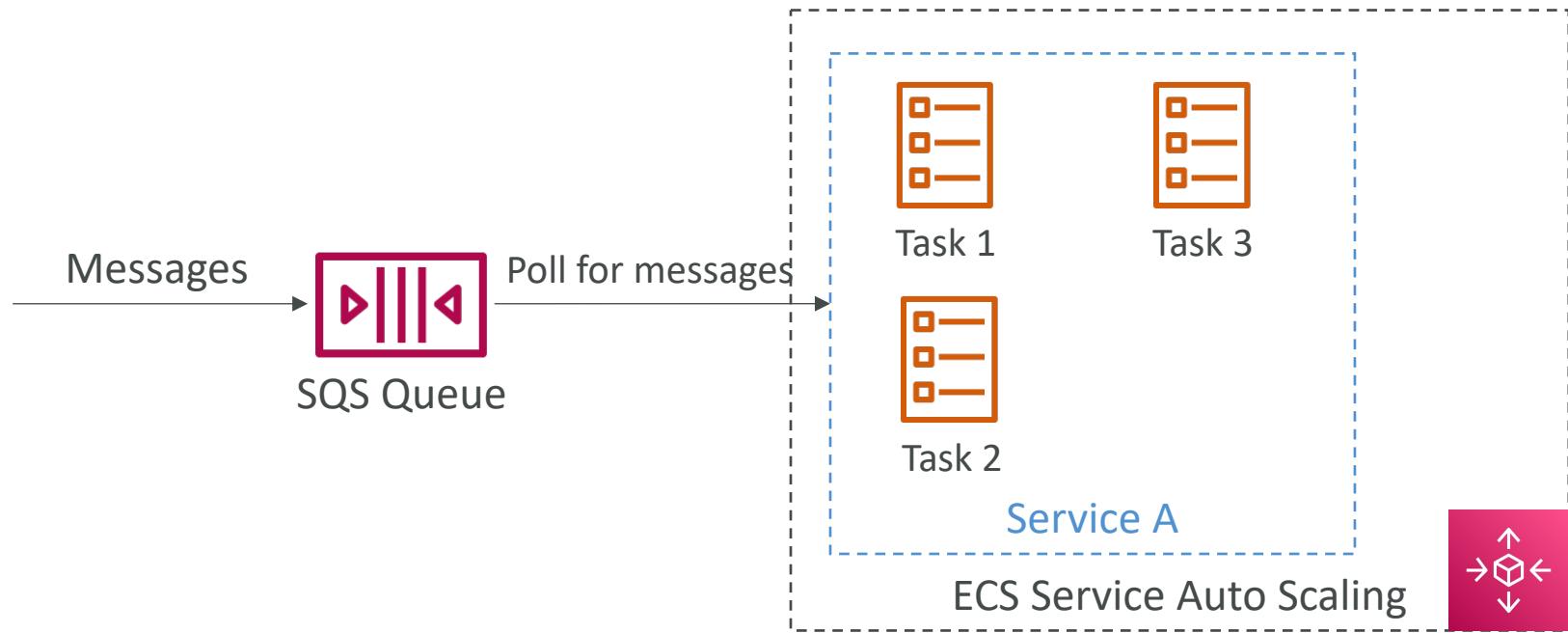
ECS tasks invoked by Event Bridge



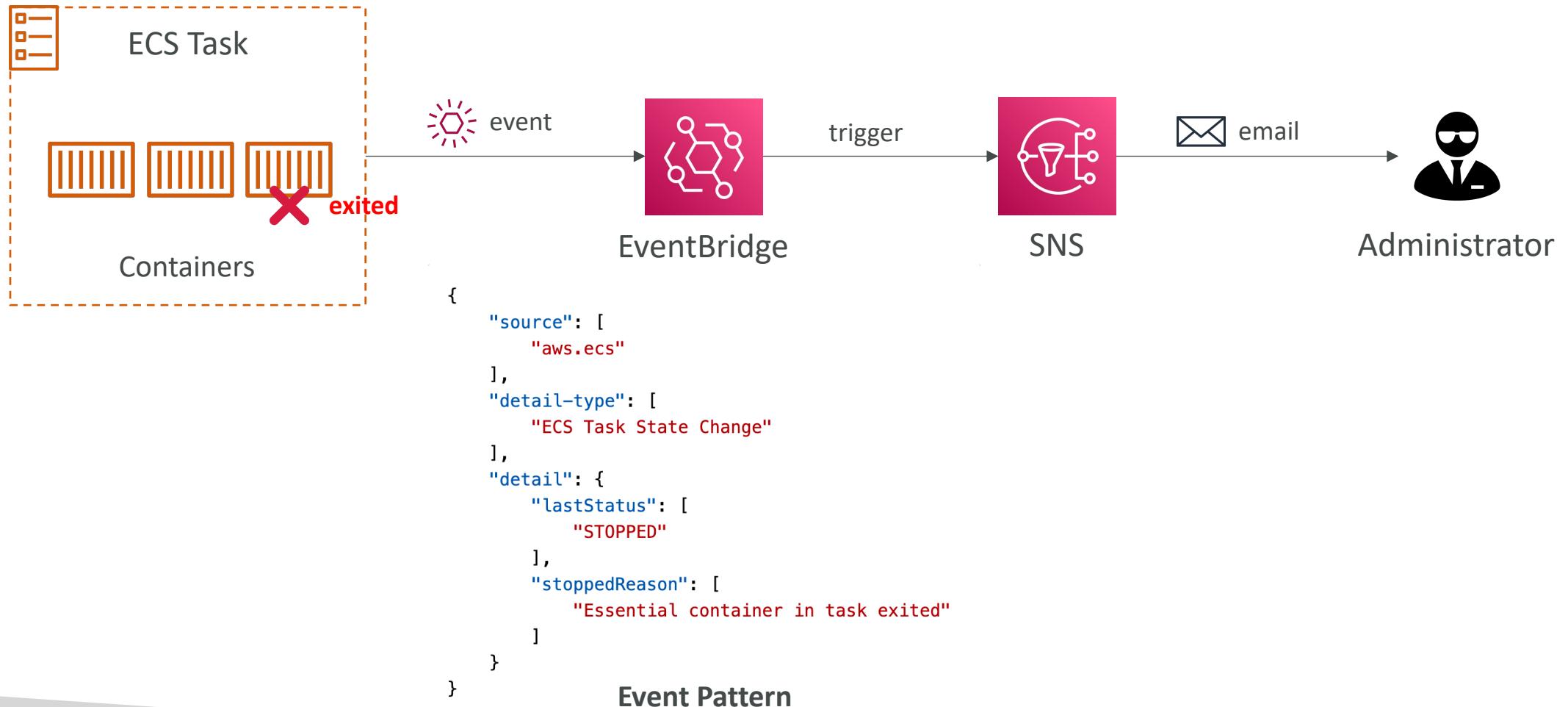
ECS tasks invoked by Event Bridge Schedule



ECS – SQS Queue Example



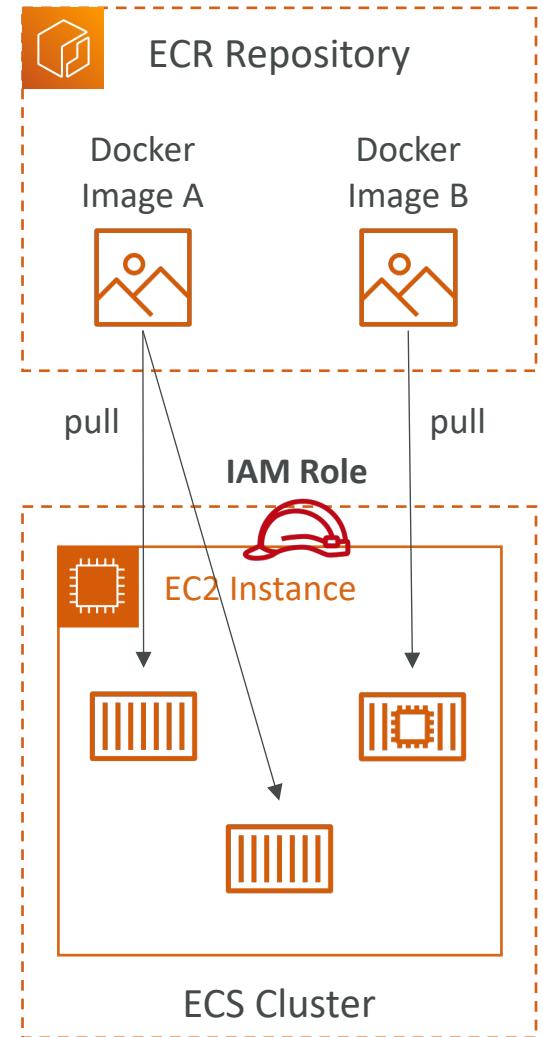
ECS – Intercept Stopped Tasks using EventBridge



Amazon ECR



- ECR = Elastic Container Registry
- Store and manage Docker images on AWS
- Private and Public repository (Amazon ECR Public Gallery <https://gallery.ecr.aws>)
- Fully integrated with ECS, backed by Amazon S3
- Access is controlled through IAM (permission errors => policy)
- Supports image vulnerability scanning, versioning, image tags, image lifecycle, ...

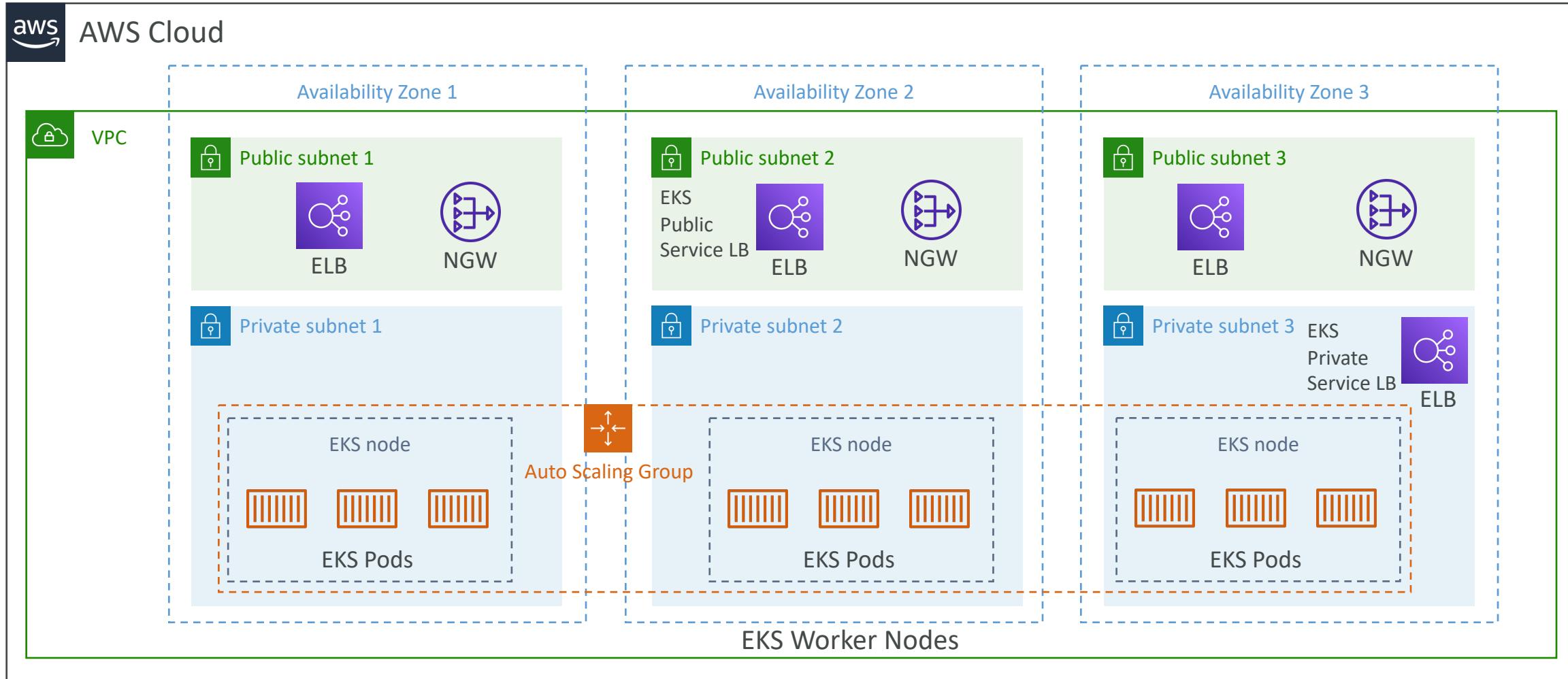


Amazon EKS Overview



- Amazon EKS = Amazon Elastic **Kubernetes** Service
- It is a way to launch **managed Kubernetes clusters** on AWS
- Kubernetes is an **open-source system** for automatic deployment, scaling and management of containerized (usually Docker) application
- It's an alternative to ECS, similar goal but different API
- EKS supports **EC2** if you want to deploy worker nodes or **Fargate** to deploy serverless containers
- **Use case:** if your company is already using Kubernetes on-premises or in another cloud, and wants to migrate to AWS using Kubernetes
- **Kubernetes is cloud-agnostic** (can be used in any cloud – Azure, GCP...)
- For multiple regions, deploy one EKS cluster per region
- Collect logs and metrics using **CloudWatch Container Insights**

Amazon EKS - Diagram

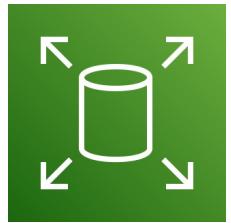


Amazon EKS – Node Types

- **Managed Node Groups**
 - Creates and manages Nodes (EC2 instances) for you
 - Nodes are part of an ASG managed by EKS
 - Supports On-Demand or Spot Instances
- **Self-Managed Nodes**
 - Nodes created by you and registered to the EKS cluster and managed by an ASG
 - You can use prebuilt AMI - Amazon EKS Optimized AMI
 - Supports On-Demand or Spot Instances
- **AWS Fargate**
 - No maintenance required; no nodes managed

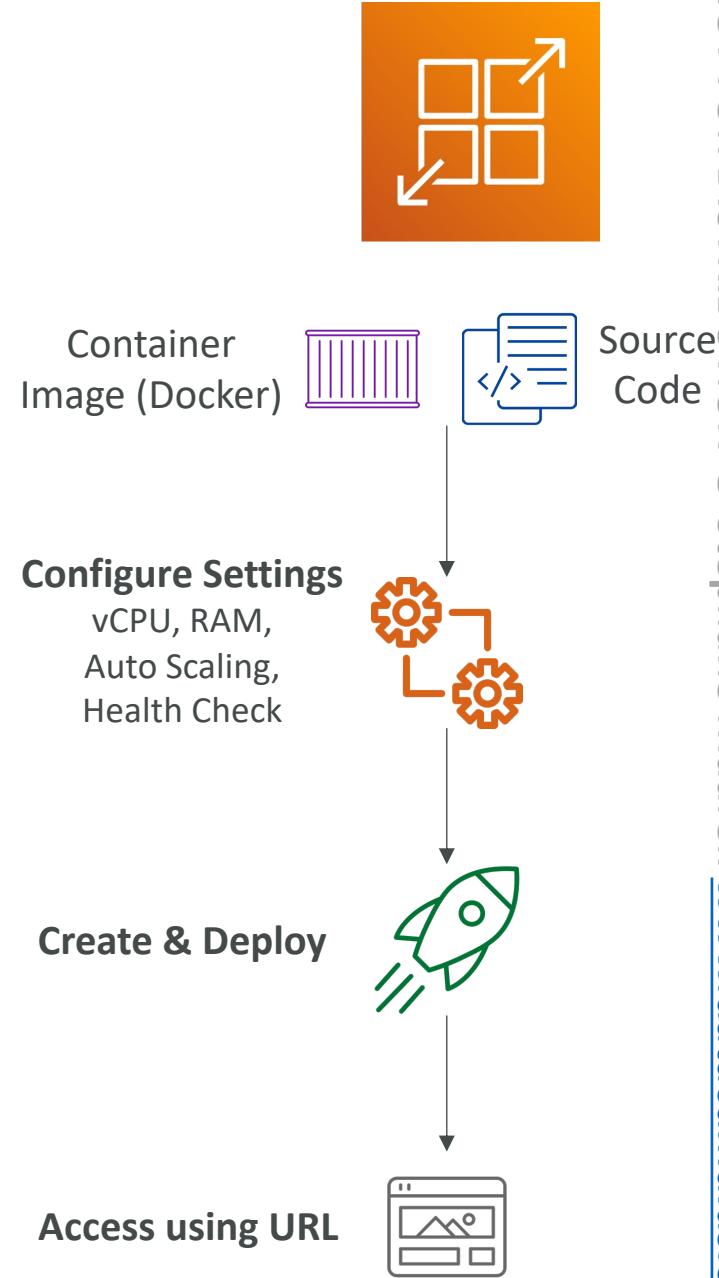
Amazon EKS – Data Volumes

- Need to specify **StorageClass** manifest on your EKS cluster
- Leverages a **Container Storage Interface (CSI)** compliant driver
- Support for...
- Amazon EBS
- Amazon EFS (works with Fargate)
- Amazon FSx for Lustre
- Amazon FSx for NetApp ONTAP



AWS App Runner

- Fully managed service that makes it easy to deploy web applications and APIs at scale
- No infrastructure experience required
- Start with your source code or container image
- Automatically builds and deploy the web app
- Automatic scaling, highly available, load balancer, encryption
- VPC access support
- Connect to database, cache, and message queue services
- Use cases: web apps, APIs, microservices, rapid production deployments



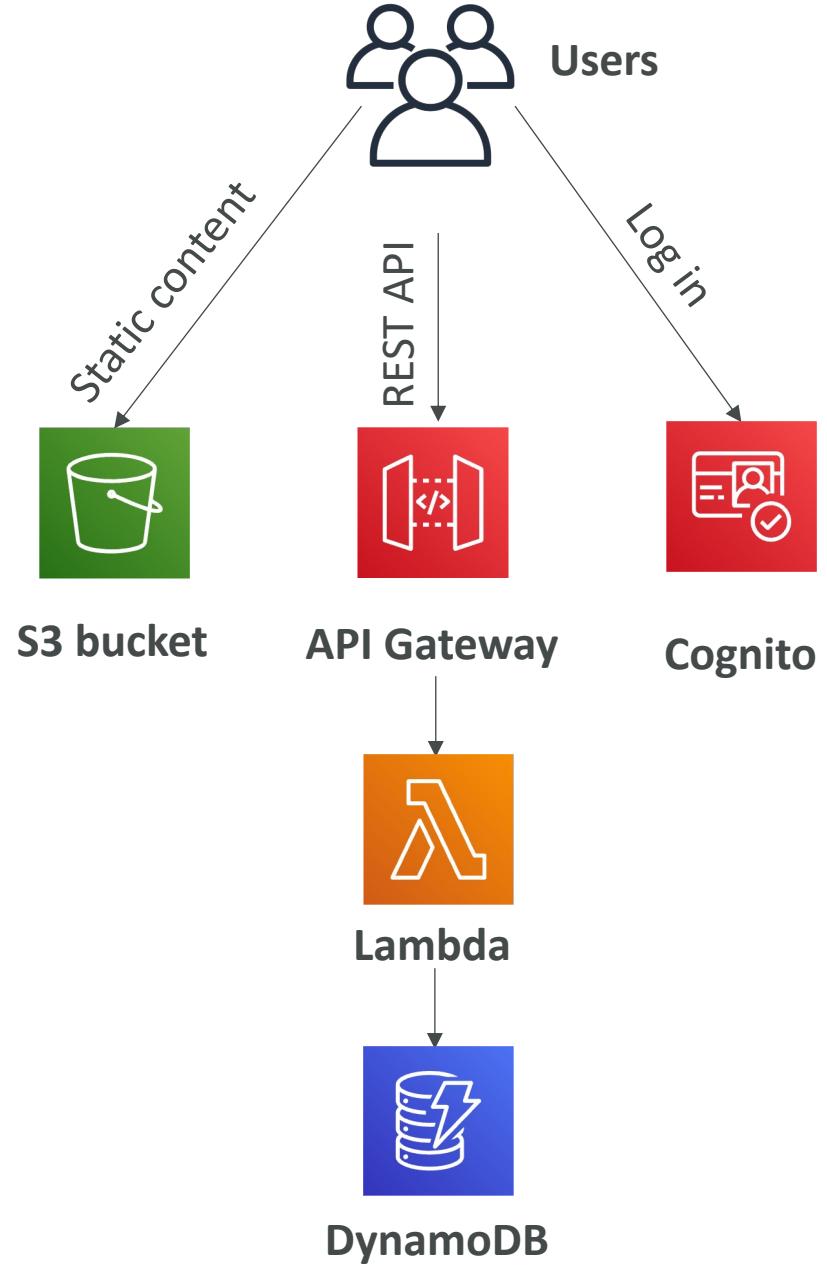
Serverless Overview

What's serverless?

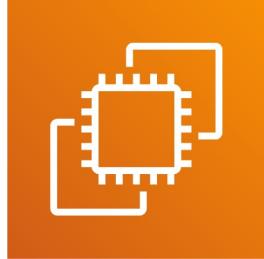
- Serverless is a new paradigm in which the developers don't have to manage servers anymore...
- They just deploy code
- They just deploy... functions !
- Initially... Serverless == FaaS (Function as a Service)
- Serverless was pioneered by AWS Lambda but now also includes anything that's managed: “databases, 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



Why AWS Lambda



Amazon EC2

- Virtual Servers in the Cloud
 - Limited by RAM and CPU
 - Continuously running
 - Scaling means intervention to add / remove servers
-



Amazon Lambda

- Virtual **functions** – no servers to manage!
- Limited by time - short **executions**
- Run **on-demand**
- Scaling is **automated!**

Benefits of AWS Lambda

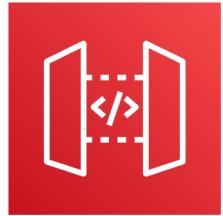
- Easy Pricing:
 - Pay per request and compute time
 - Free tier of 1,000,000 AWS Lambda requests and 400,000 GBs of compute time
- Integrated with the whole AWS suite of services
- Integrated with many programming languages
- Easy monitoring through AWS CloudWatch
- Easy to get more resources per functions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

AWS Lambda language support

- Node.js (JavaScript)
- Python
- Java (Java 8 compatible)
- C# (.NET Core)
- Golang
- C# / Powershell
- Ruby
- Custom Runtime API (community supported, example Rust)
- Lambda Container Image
 - The container image must implement the Lambda Runtime API
 - ECS / Fargate is preferred for running arbitrary Docker images

AWS Lambda Integrations

Main ones



API Gateway



Kinesis



DynamoDB



S3



CloudFront



CloudWatch Events
EventBridge



CloudWatch Logs



SNS

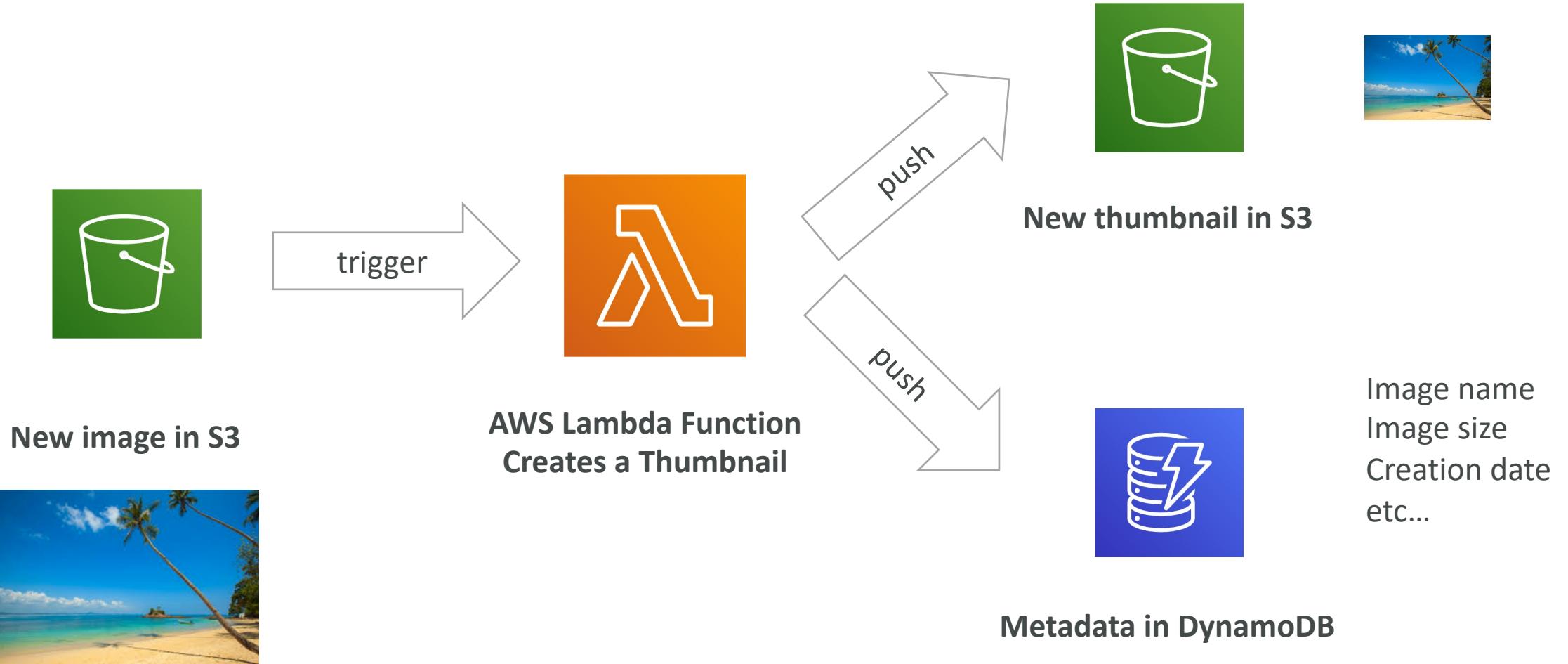


SQS



Cognito

Example: Serverless Thumbnail creation



Example: Serverless CRON Job

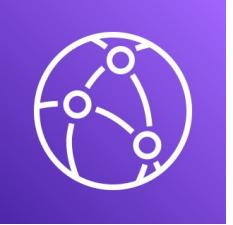


AWS Lambda Pricing: example

- 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 1 GB 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

AWS Lambda Limits to Know - per region

- **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

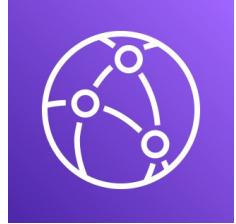


Customization At The Edge

- Many modern applications execute some form of the 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

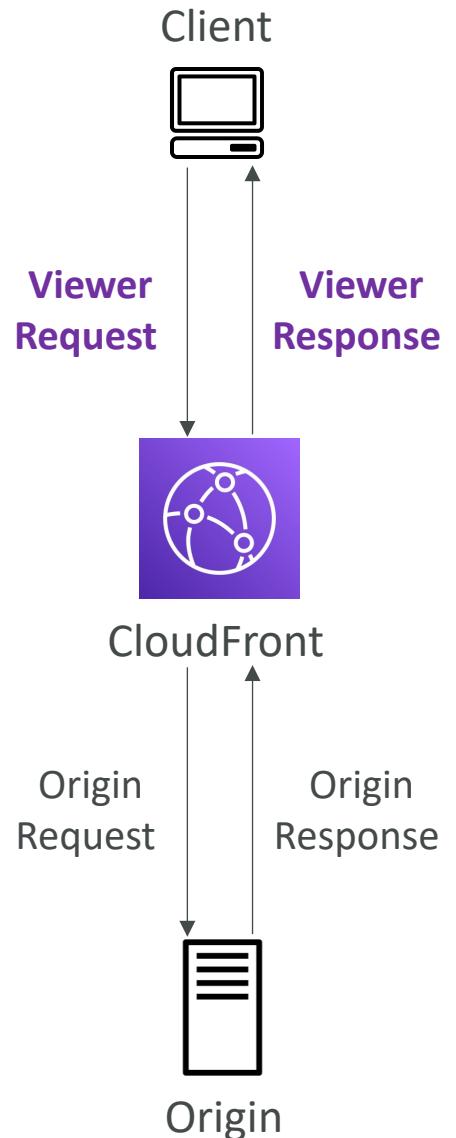
CloudFront Functions & Lambda@Edge Use Cases



- 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

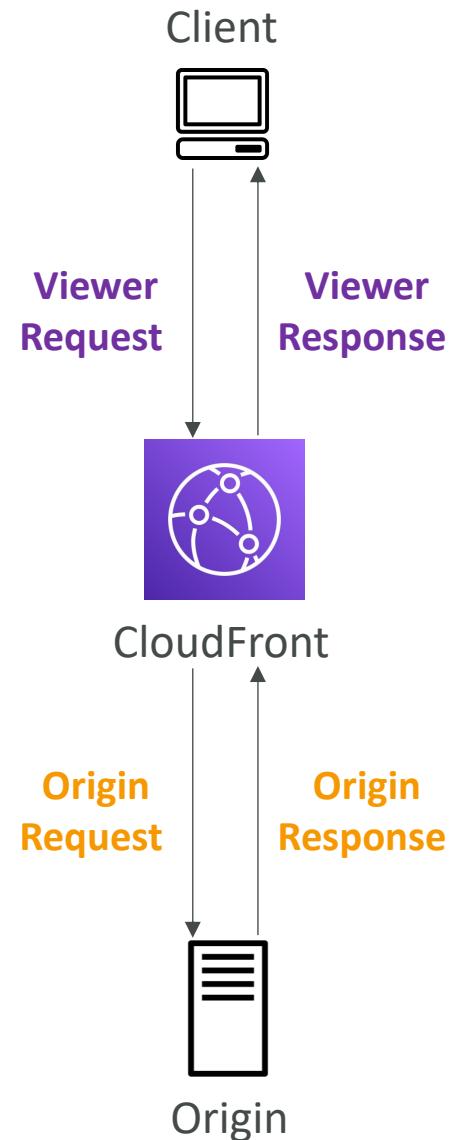
CloudFront Functions

- 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 locations



CloudFront Functions vs. Lambda@Edge

	CloudFront Functions	Lambda@Edge
Runtime Support	JavaScript	Node.js, Python
# of Requests	Millions of requests per second	Thousands of requests per second
CloudFront Triggers	- Viewer Request/Response	- Viewer Request/Response - Origin Request/Response
Max. Execution Time	< 1 ms	5 – 10 seconds
Max. Memory	2 MB	128 MB up to 10 GB
Total Package Size	10 KB	1 MB – 50 MB
Network Access, File System Access	No	Yes
Access to the Request Body	No	Yes
Pricing	Free tier available, 1/6 th price of @Edge	No free tier, charged per request & duration

CloudFront Functions vs. Lambda@Edge - Use Cases

CloudFront Functions

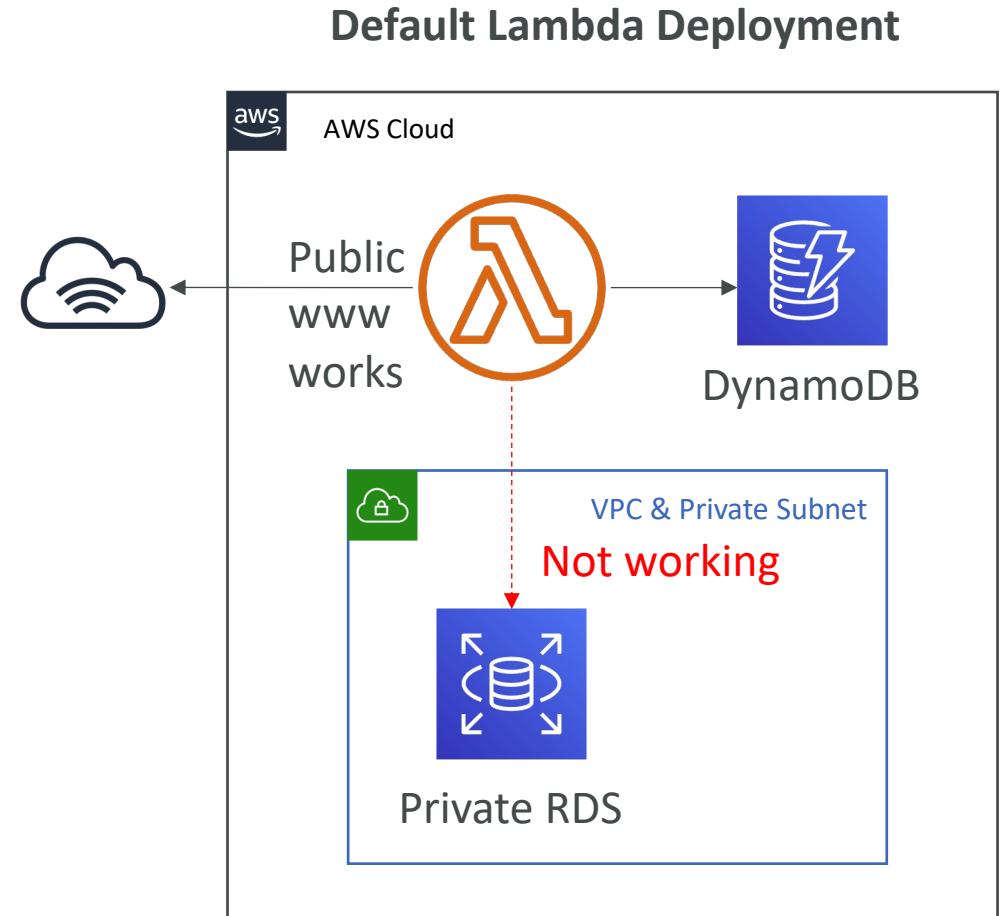
- Cache key normalization
 - Transform request attributes (headers, cookies, query strings, URL) to create an optimal Cache Key
- Header manipulation
 - Insert/modify/delete HTTP headers in the request or response
- URL rewrites or redirects
- Request authentication & authorization
 - Create and validate 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

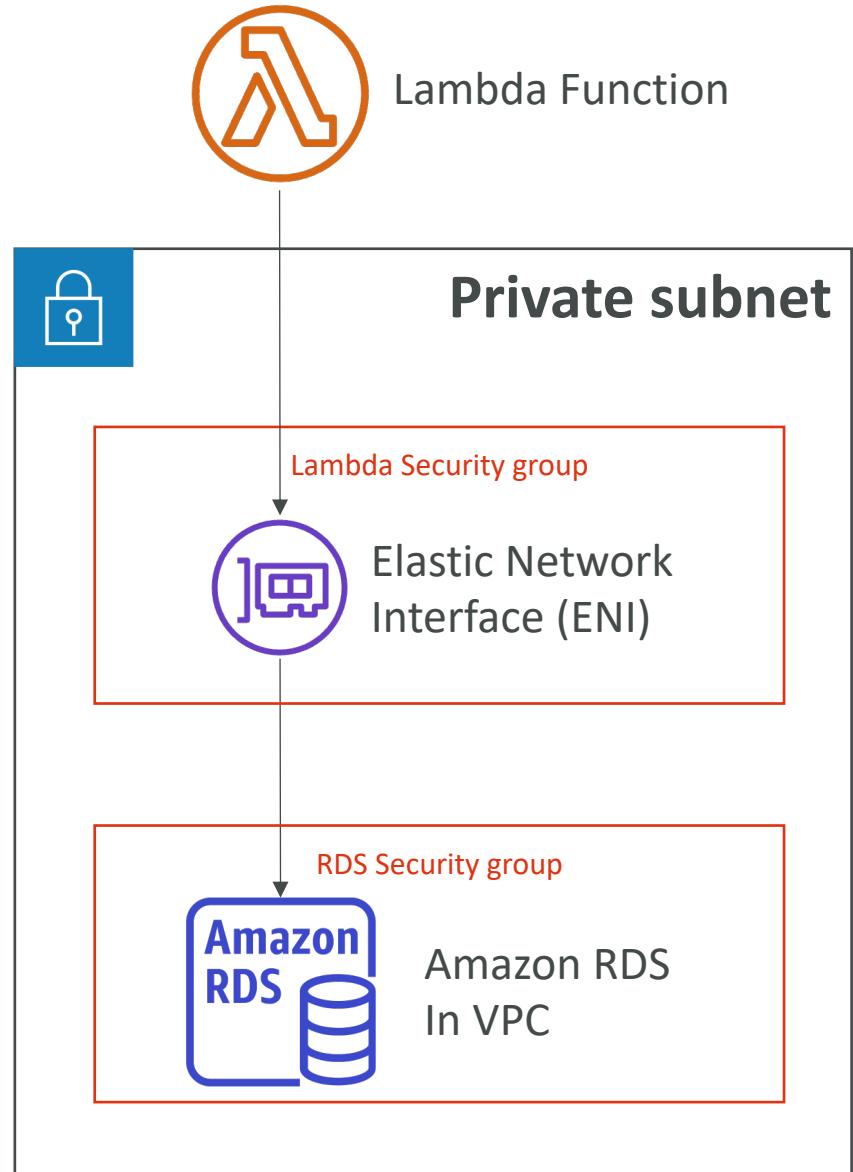
Lambda by default

- 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...)



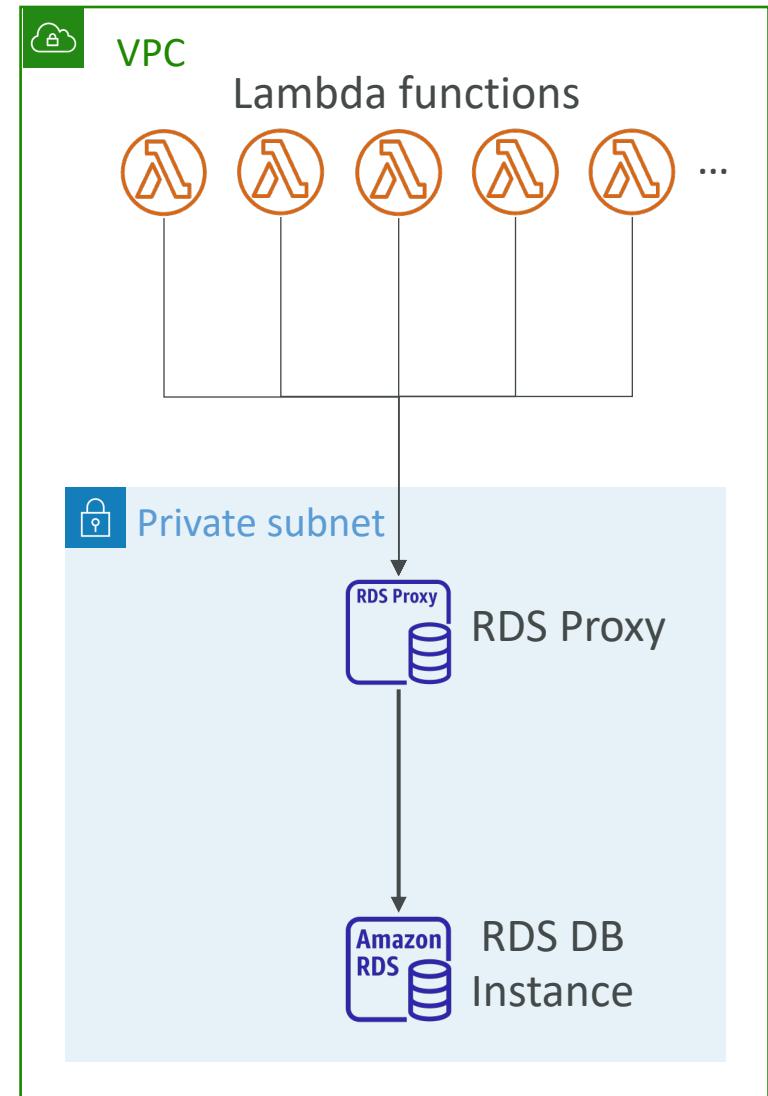
Lambda in VPC

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



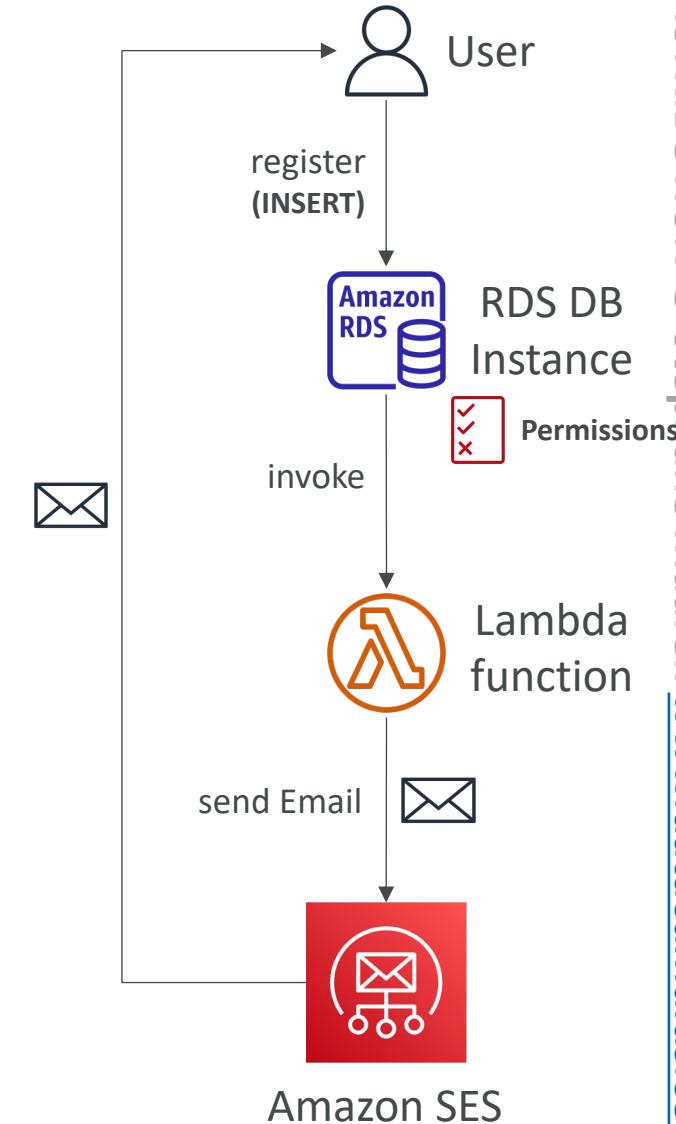
Lambda with RDS Proxy

- 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**



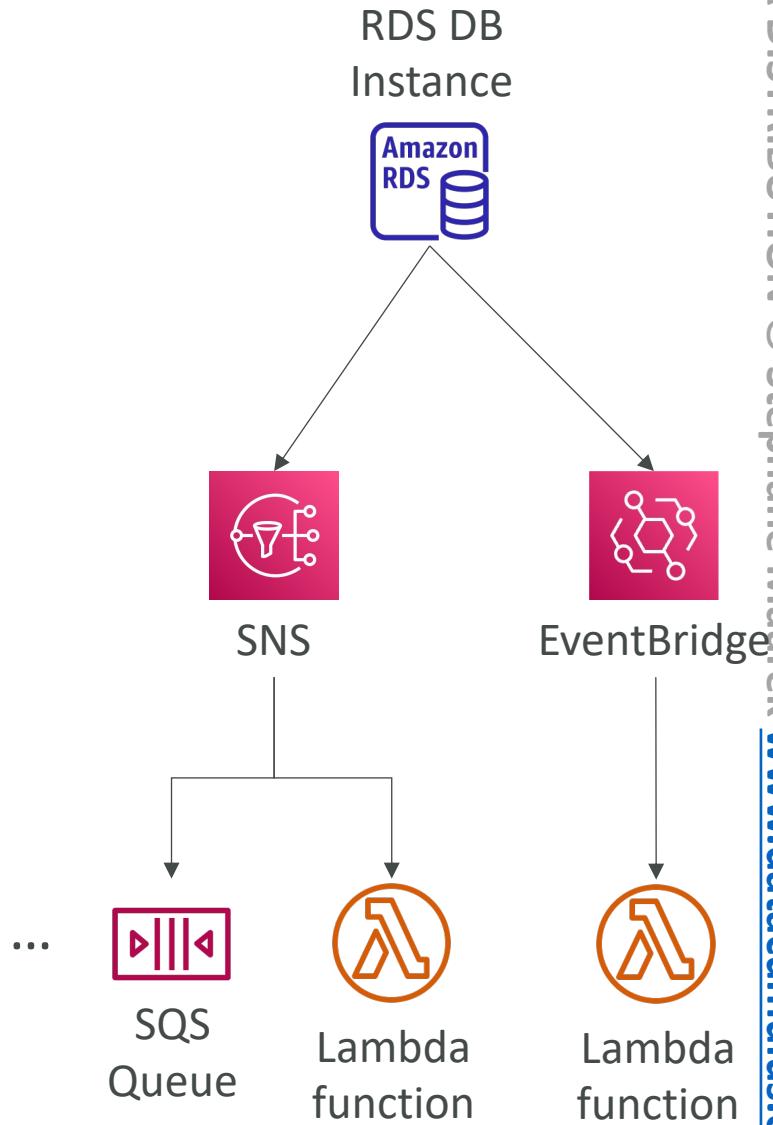
Invoking Lambda from RDS & Aurora

- Invoke Lambda functions from within your DB instance
- Allows you to process **data events** from within a database
- Supported 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

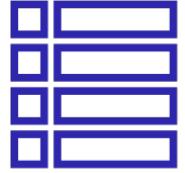
- Notifications that tells information about the DB instance itself (created, stopped, start, ...)
- You don't have any information about the data itself
- Subscribe to the following event categories: DB instance, DB snapshot, DB Parameter Group, DB Security Group, RDS Proxy, Custom Engine Version
- 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)
- 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

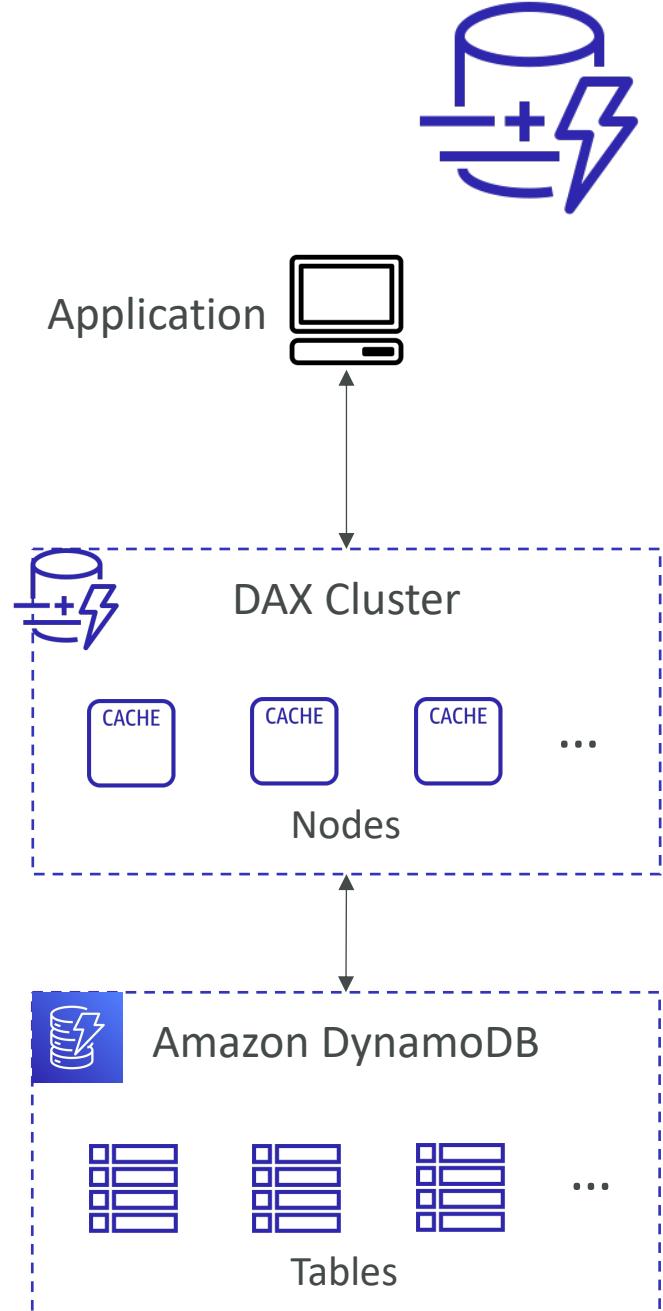
Primary Key		Attributes	
Partition Key	Sort Key	Score	Result
User_ID	Game_ID	Score	Result
7791a3d6...	4421	92	Win
873e0634...	1894	14	Lose
873e0634...	4521	77	Win

DynamoDB – Read/Write Capacity Modes

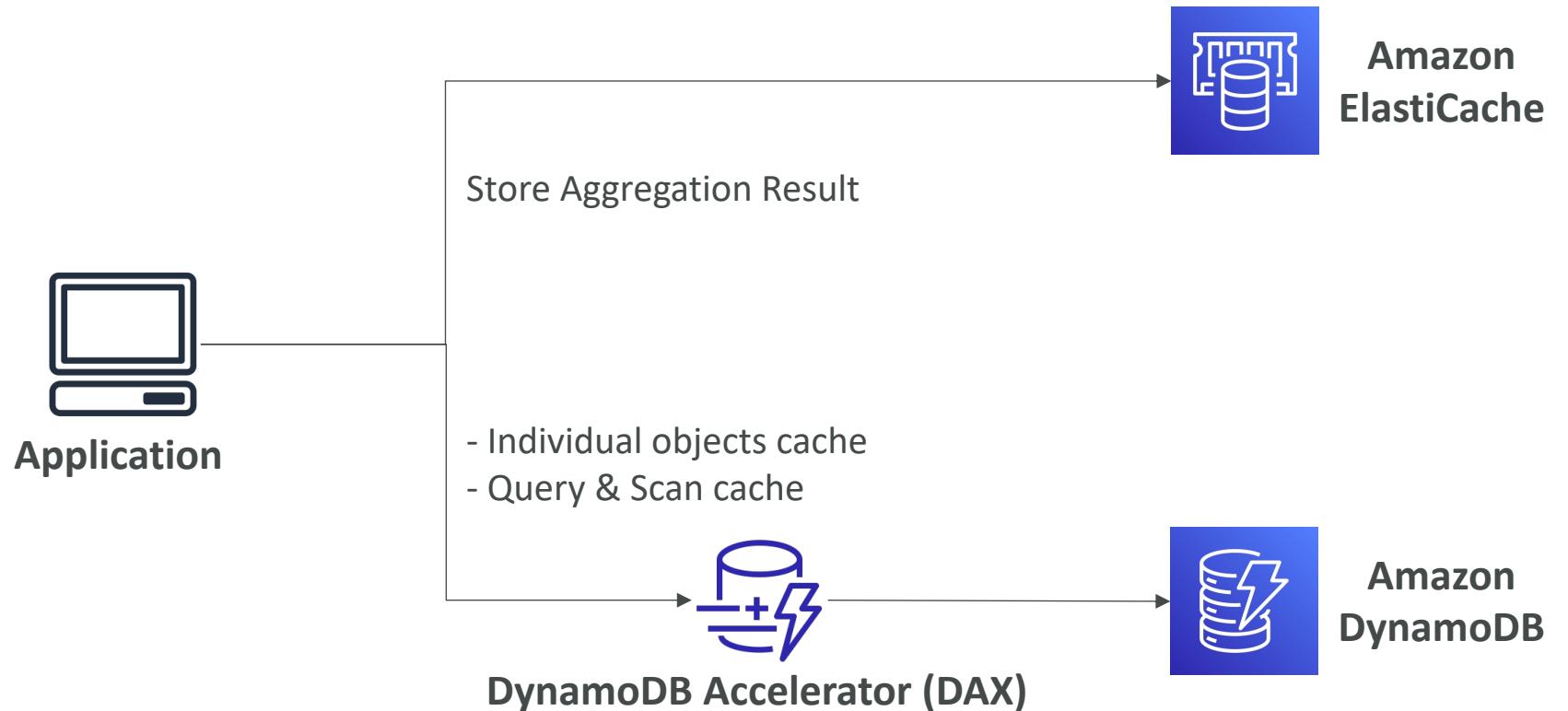
- Control how you manage your table's capacity (read/write throughput)
- Provisioned Mode (default)
 - You specify the number of reads/writes per second
 - You need to plan capacity beforehand
 - Pay for provisioned Read Capacity Units (RCU) & Write Capacity Units (WCUs)
 - Possibility to add auto-scaling 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

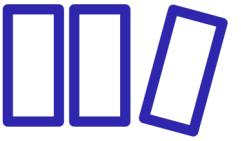
DynamoDB Accelerator (DAX)

- 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

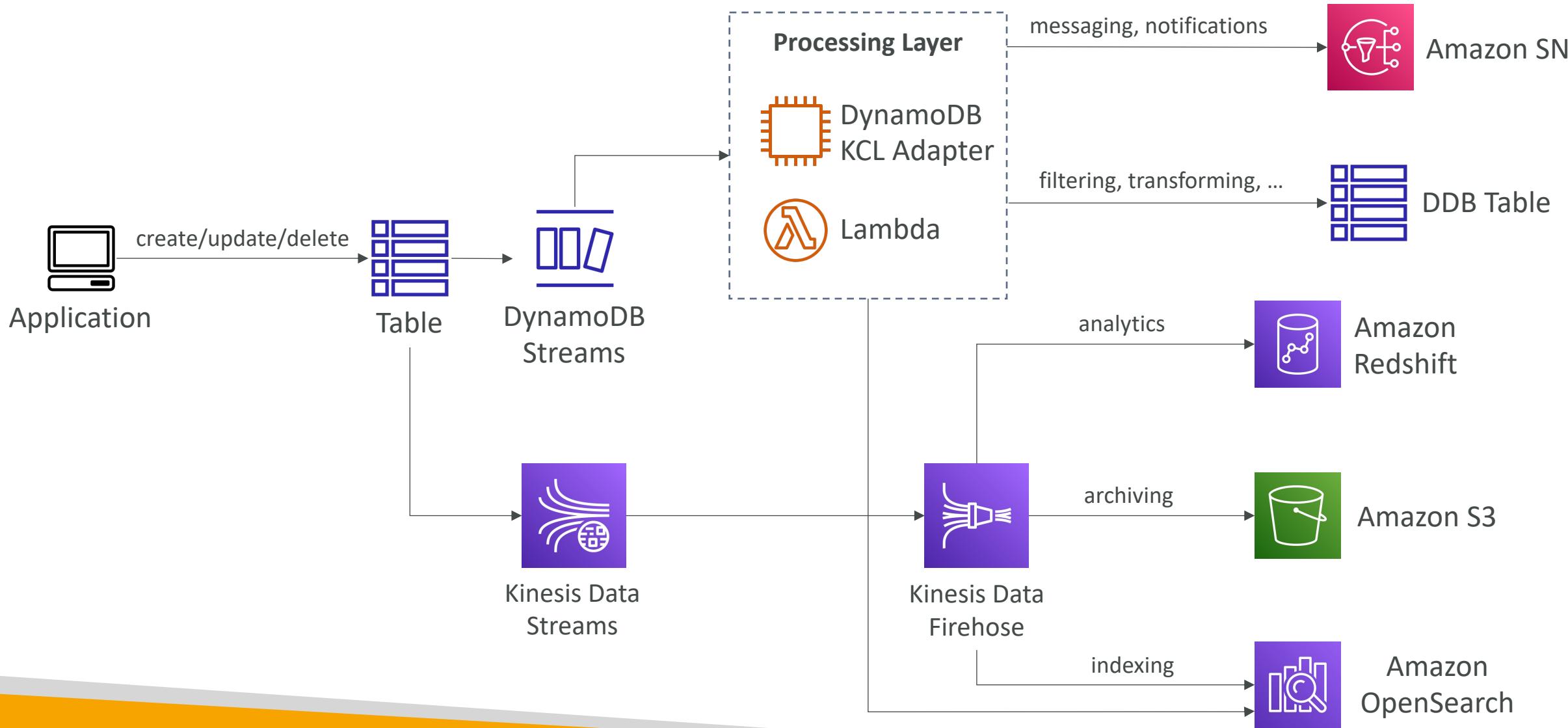
DynamoDB Streams

- 24 hours retention
- Limited # of consumers
- Process using AWS Lambda Triggers, or DynamoDB Stream Kinesis adapter

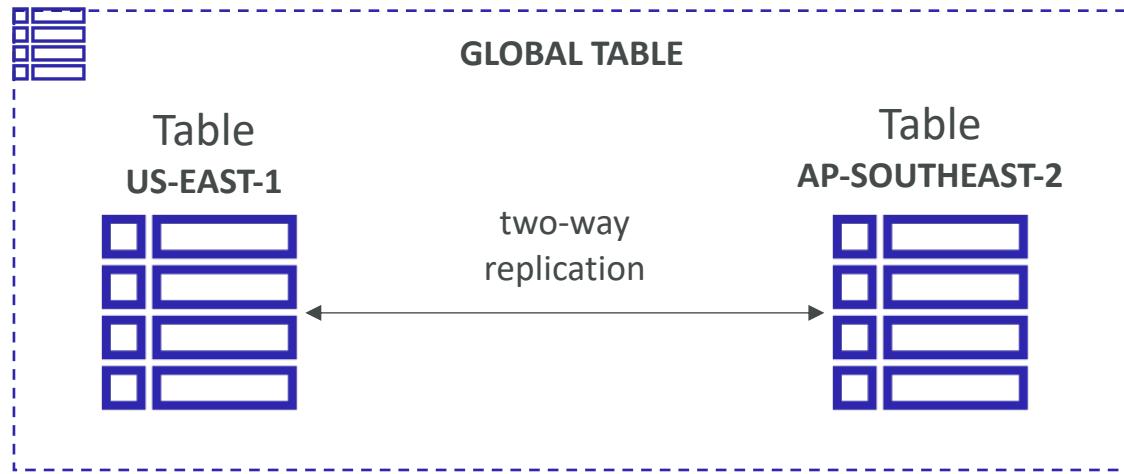
Kinesis Data Streams (newer)

- 1 year retention
- High # of consumers
- Process using AWS Lambda, Kinesis Data Analytics, Kinesis Data Firehose, AWS Glue Streaming ETL...

DynamoDB Streams



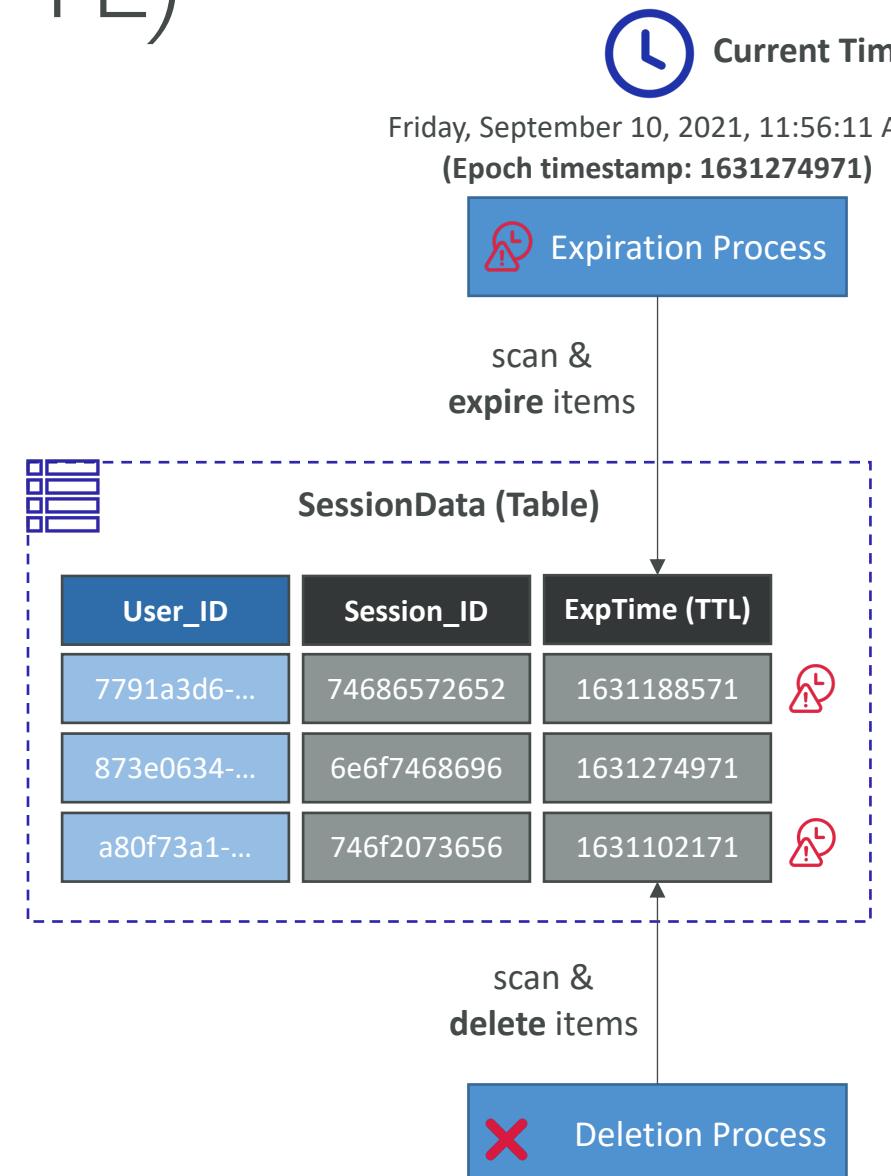
DynamoDB Global Tables



- 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...

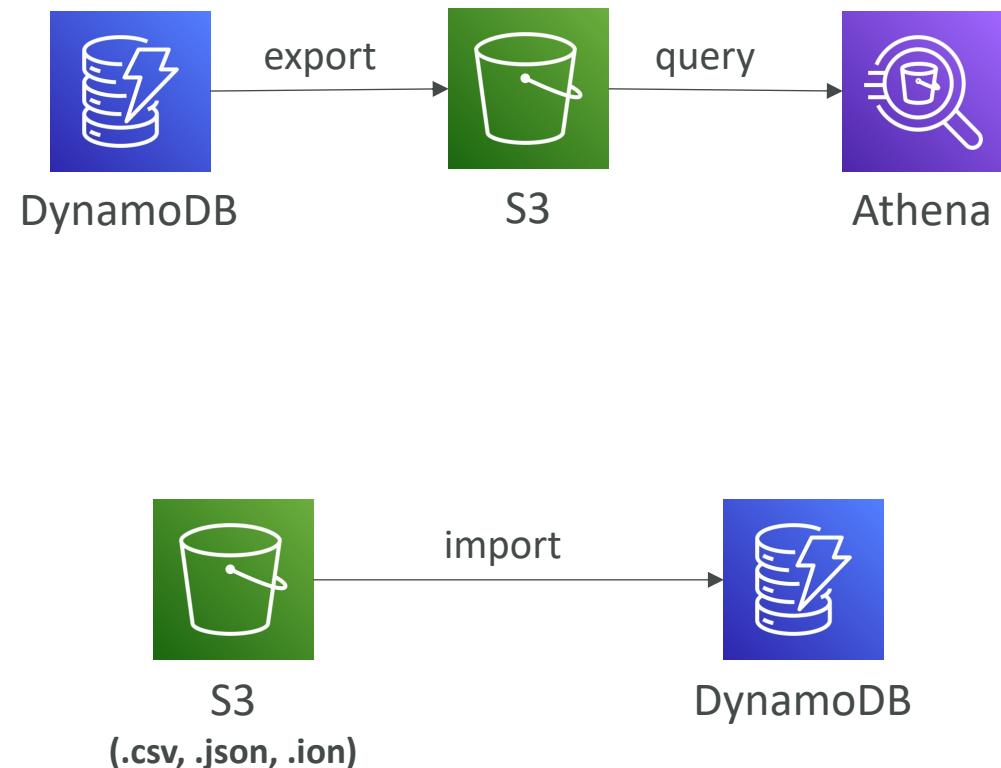


DynamoDB – Backups for disaster recovery

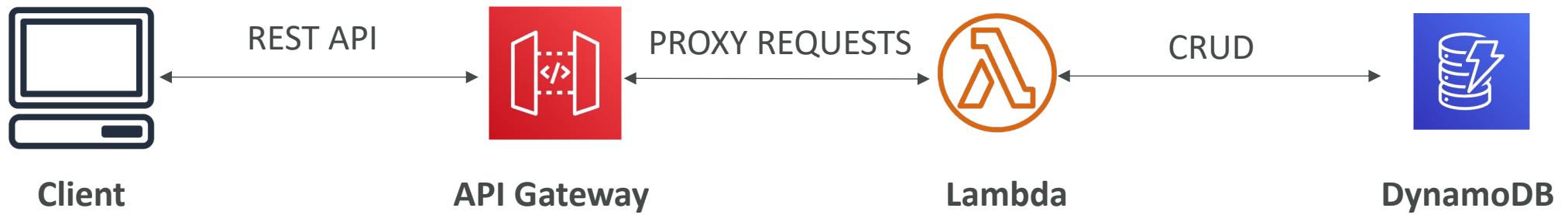
- Continuous backups using point-in-time recovery (PITR)
 - Optionally enabled for the last 35 days
 - Point-in-time recovery to any time within the backup window
 - The recovery process creates a new table
- On-demand backups
 - Full backups for long-term retention, until explicitly deleted
 - Doesn't 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

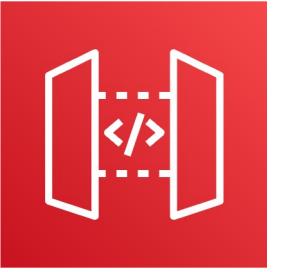
- Export to S3 (must enable PITR)
 - Works for any point of time in the last 35 days
 - Doesn't affect the read capacity of your table
 - Perform data analysis on top of DynamoDB
 - Retain snapshots for auditing
 - ETL on top of S3 data before importing back into DynamoDB
 - Export in DynamoDB JSON or 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



Example: Building a Serverless API



AWS API Gateway



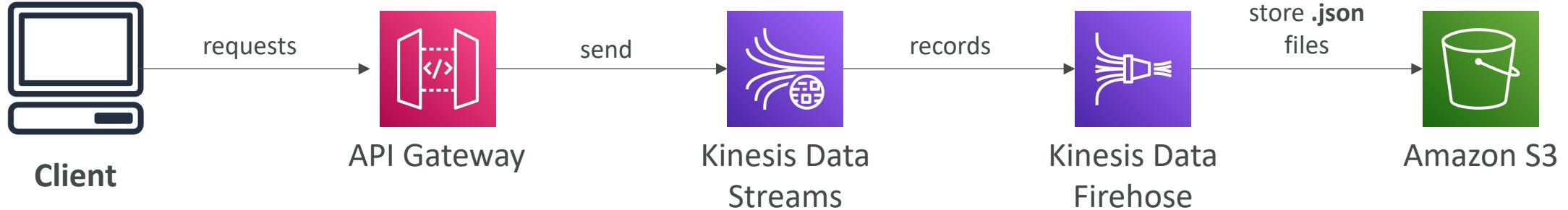
- AWS Lambda + API Gateway: No infrastructure to manage
- Support for the WebSocket Protocol
- Handle API versioning (v1, v2...)
- Handle different environments (dev, test, prod...)
- Handle security (Authentication and Authorization)
- Create API keys, handle request throttling
- Swagger / Open API 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



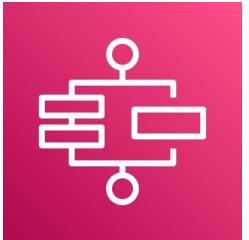
API Gateway - Endpoint Types

- **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

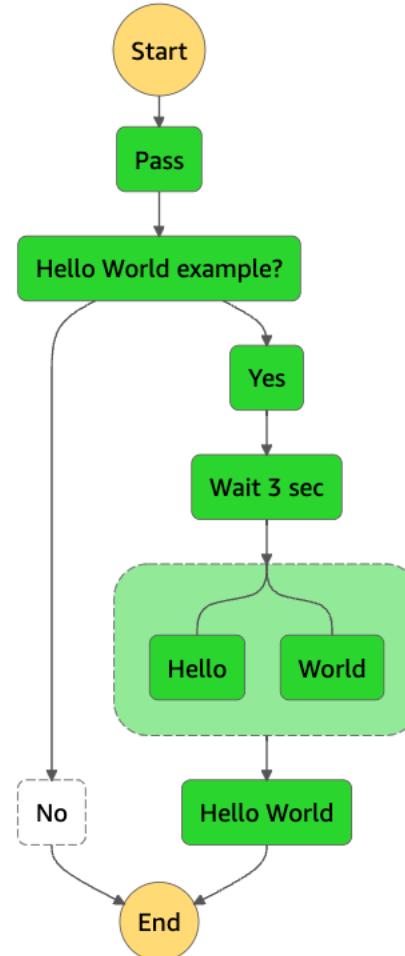
- User Authentication through
 - IAM Roles (useful for internal applications)
 - Cognito (identity for external users – example mobile users)
 - Custom Authorizer (your own logic)
- Custom Domain Name HTTPS security through integration with AWS Certificate Manager (ACM)
 - If using Edge-Optimized endpoint, 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

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 feature
- **Use cases:** order fulfillment, data processing, web applications, any workflow

■ In Progress ■ Succeeded ■ Failed ■ Cancelled ■ Caught Error



Amazon Cognito



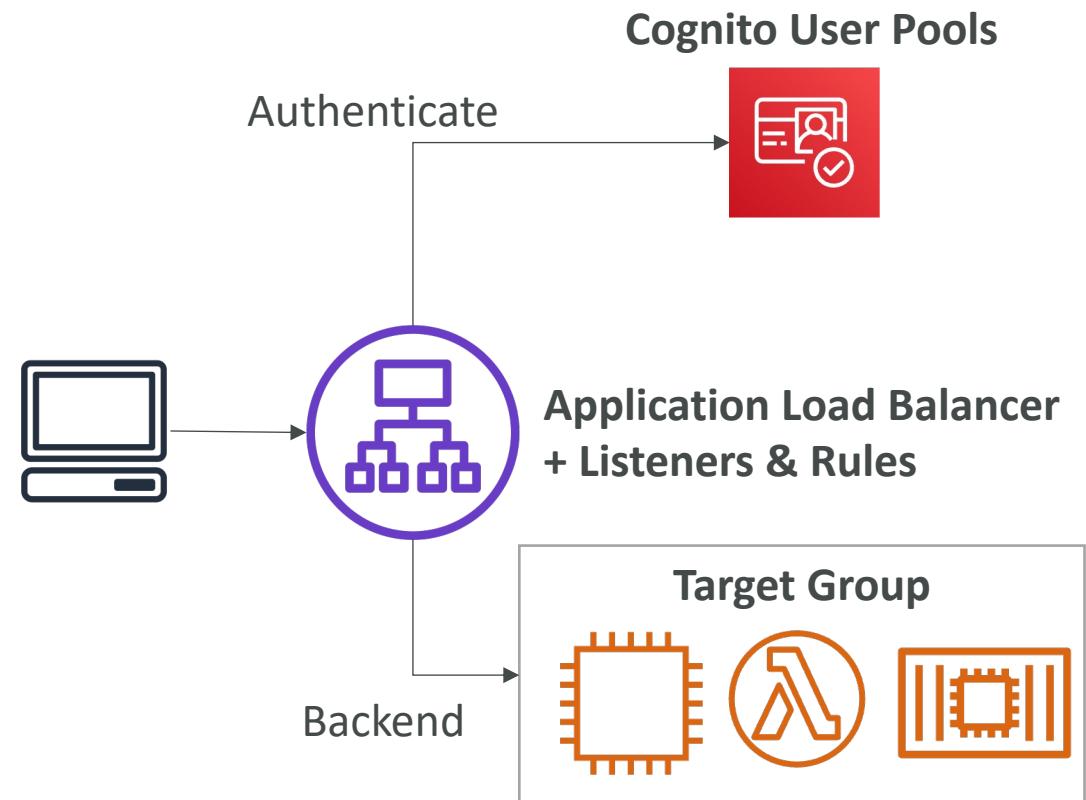
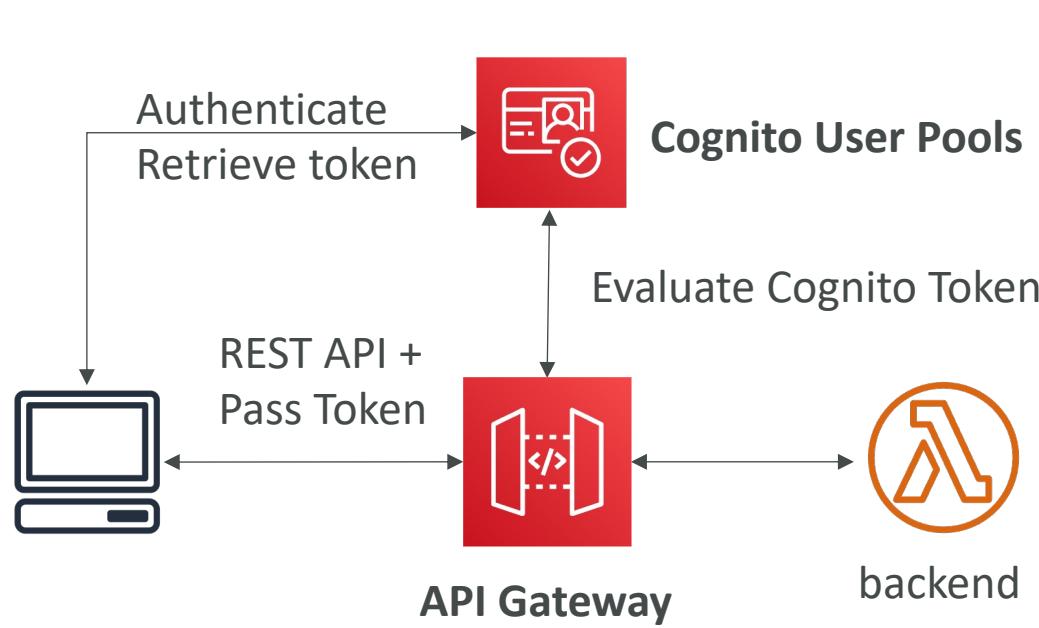
- Give users an identity to interact with our web or mobile application
- **Cognito User Pools:**
 - Sign in functionality for app users
 - Integrate with API Gateway & Application Load Balancer
- **Cognito Identity Pools (Federated Identity):**
 - Provide AWS credentials to users so they can access AWS resources directly
 - Integrate with Cognito User Pools as an identity provider
- **Cognito vs IAM:** “hundreds of users”, “mobile users”, “authenticate with SAML”

Cognito User Pools (CUP) – User Features

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

Cognito User Pools (CUP) - Integrations

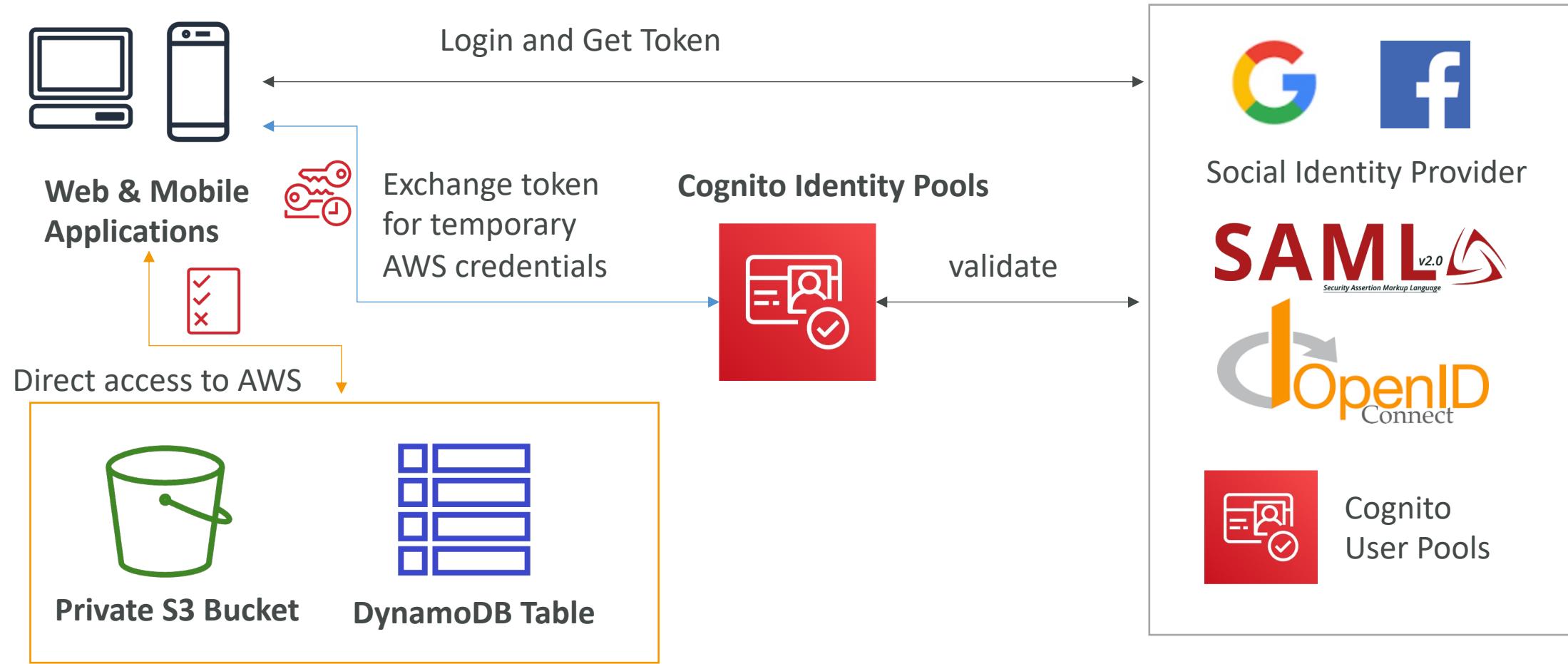
- CUP integrates with API Gateway and Application Load Balancer



Cognito Identity Pools (Federated Identities)

- 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



Cognito Identity Pools

Row Level Security in DynamoDB

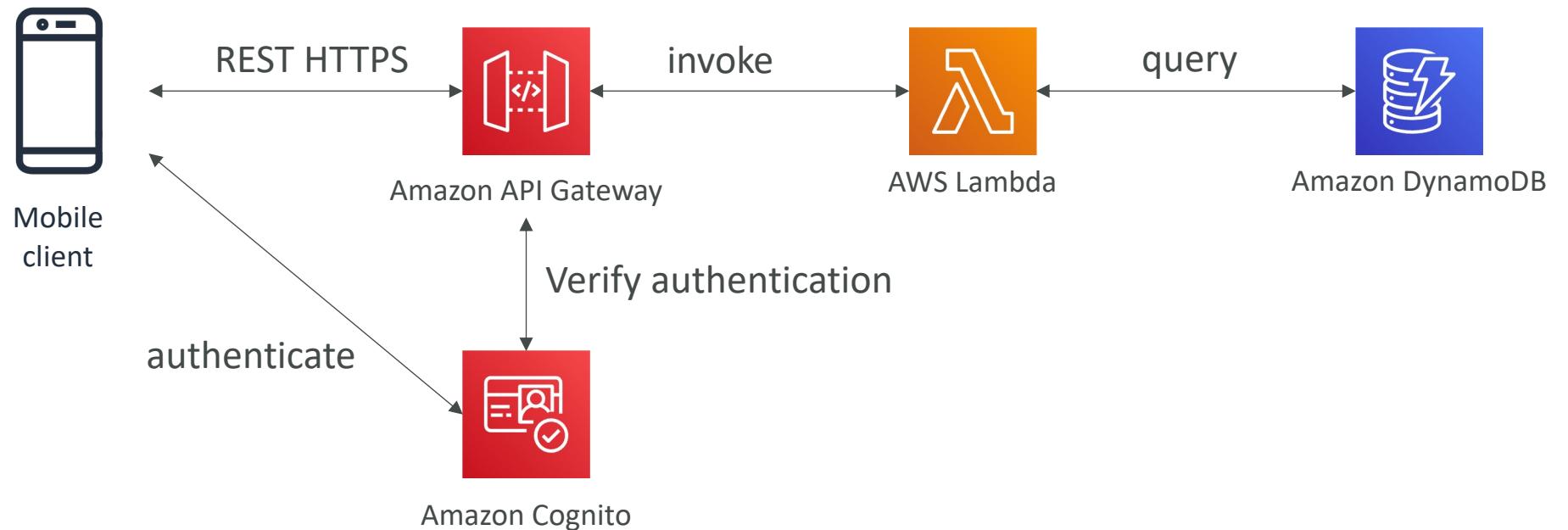
```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "dynamodb:GetItem", "dynamodb:BatchGetItem", "dynamodb:Query",  
                "dynamodb:PutItem", "dynamodb:UpdateItem", "dynamodb:DeleteItem",  
                "dynamodb:BatchWriteItem"  
            ],  
            "Resource": [  
                "arn:aws:dynamodb:us-west-2:123456789012:table/MyTable"  
            ],  
            "Condition": {  
                "ForAllValues:StringEquals": {  
                    "dynamodb:LeadingKeys": [  
                        "${cognito-identity.amazonaws.com:sub}"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

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 scale, and have some high read throughput

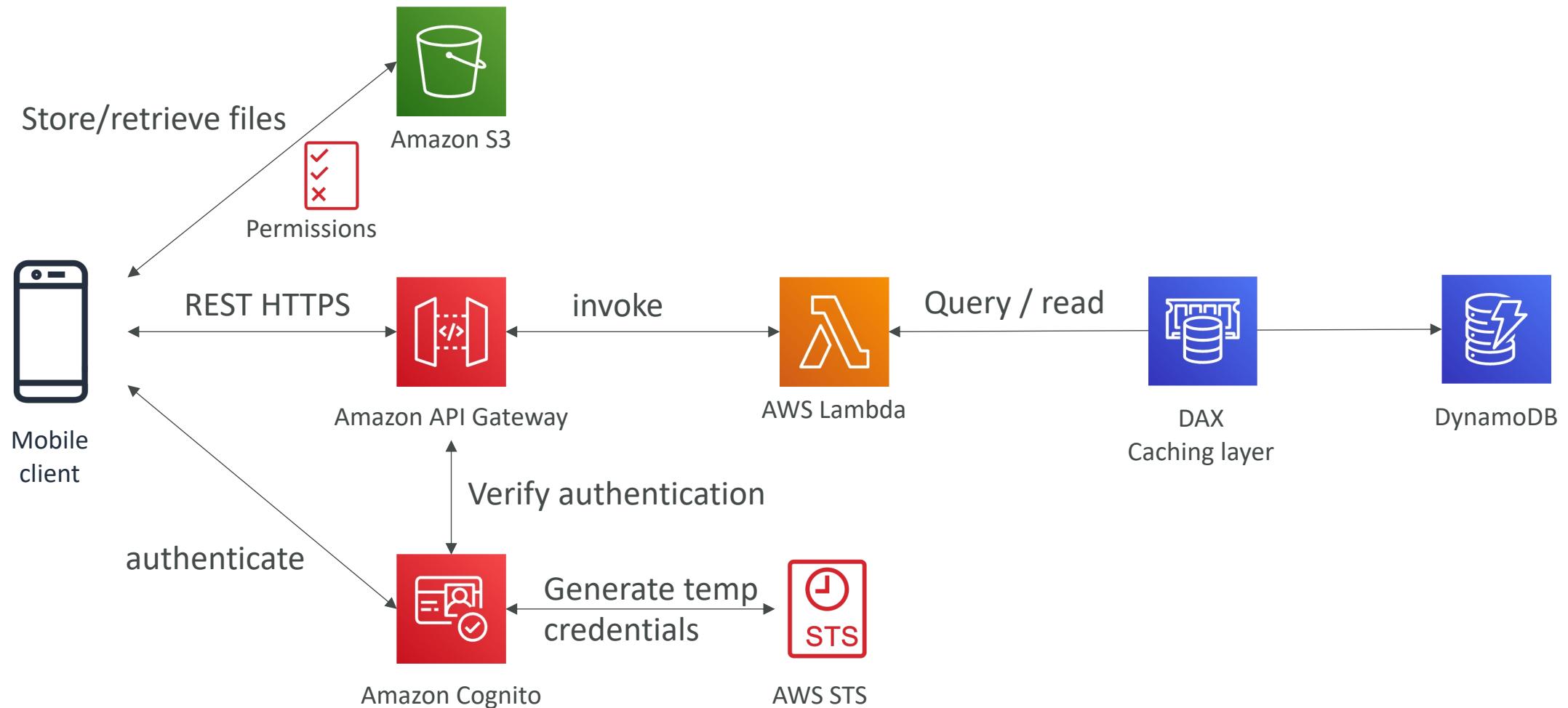
Mobile app: REST API layer



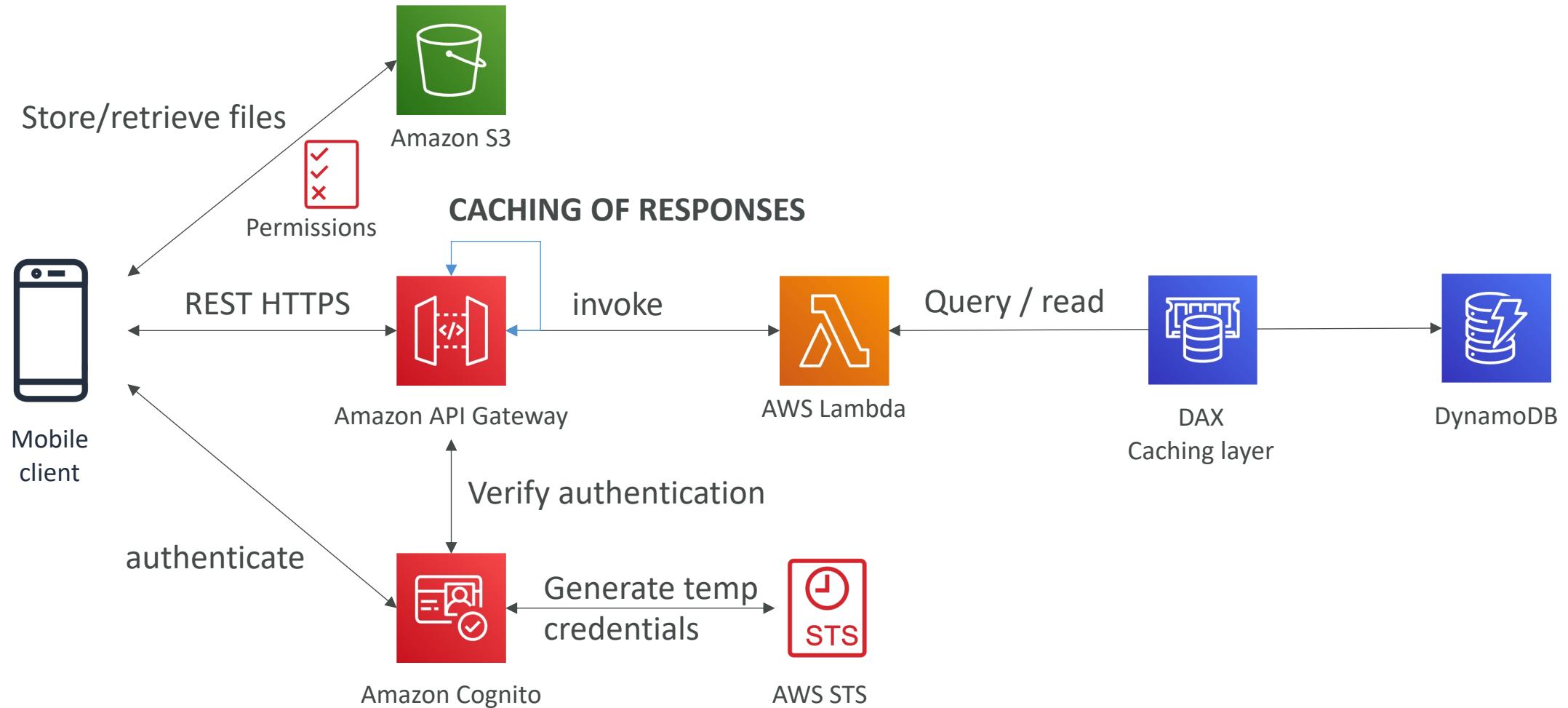
Mobile app: giving users access to S3



Mobile app: high read throughput, static data



Mobile app: caching at the API Gateway



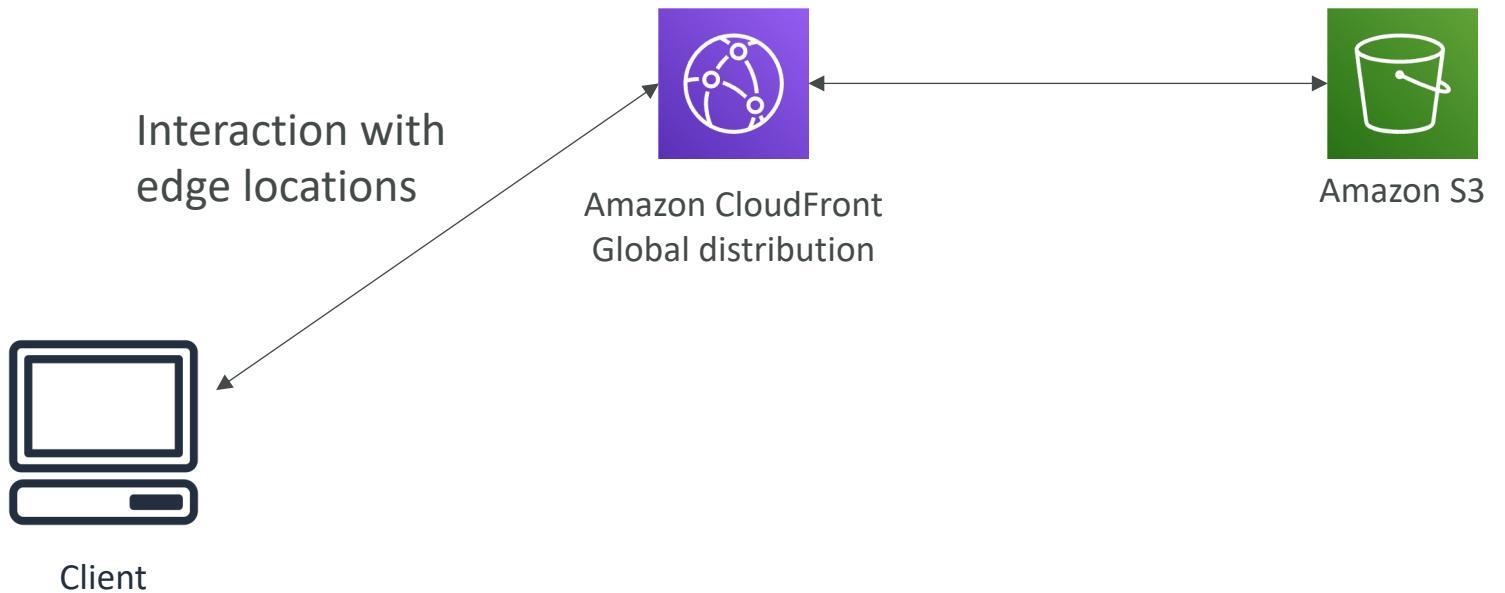
In this lecture

- Serverless REST API: HTTPS, API Gateway, Lambda, DynamoDB
- Using Cognito to generate temporary credentials with STS 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, STS

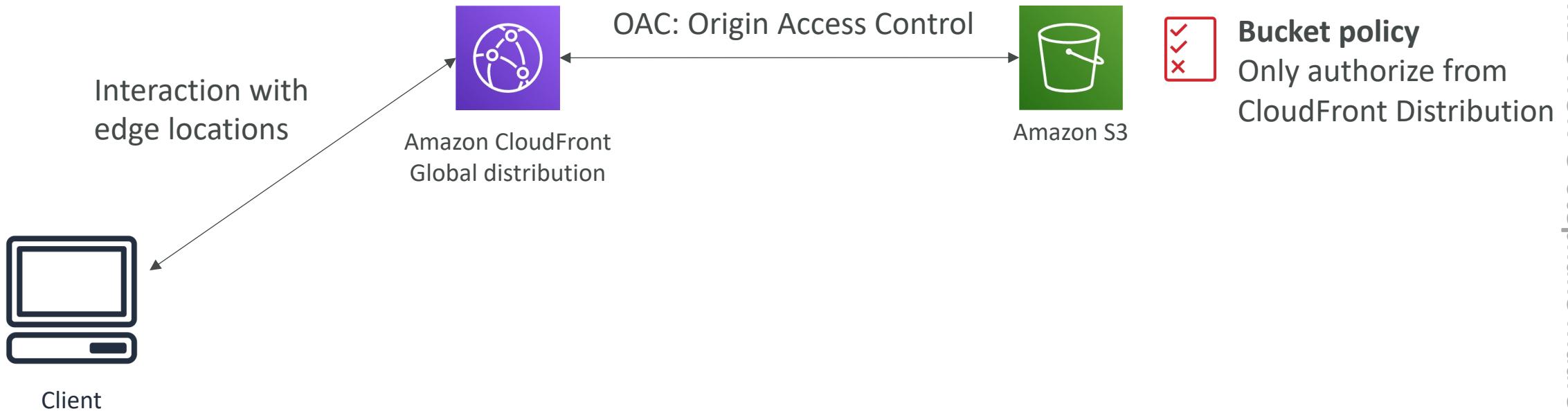
Serverless hosted website: MyBlog.com

- 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 implemented where possible
- Any new users that subscribe should receive a welcome email
- Any photo uploaded to the blog should have a thumbnail generated

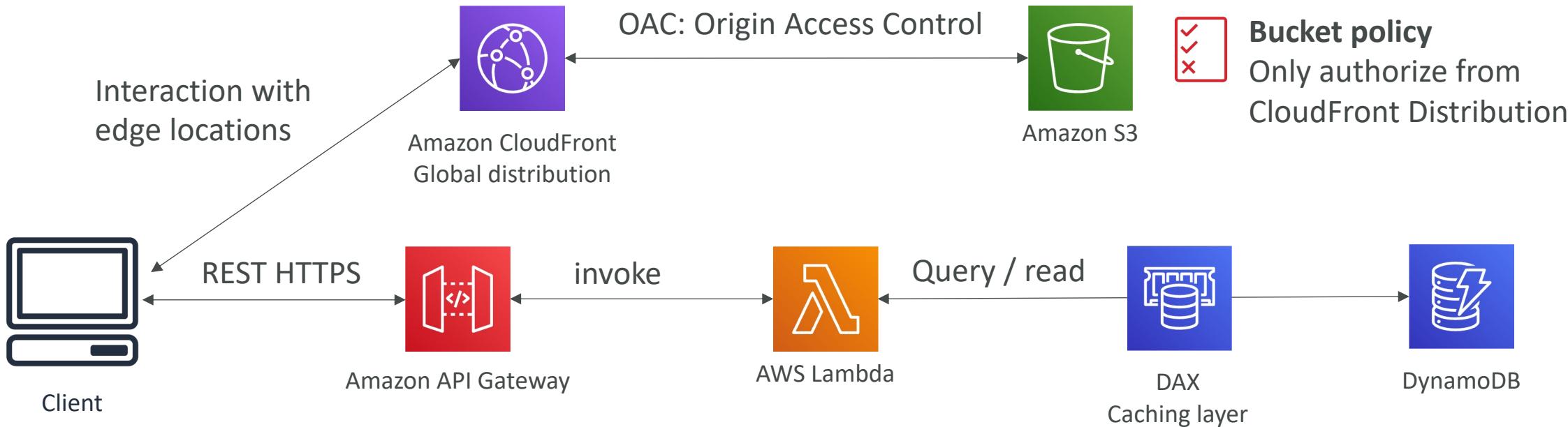
Serving static content, globally



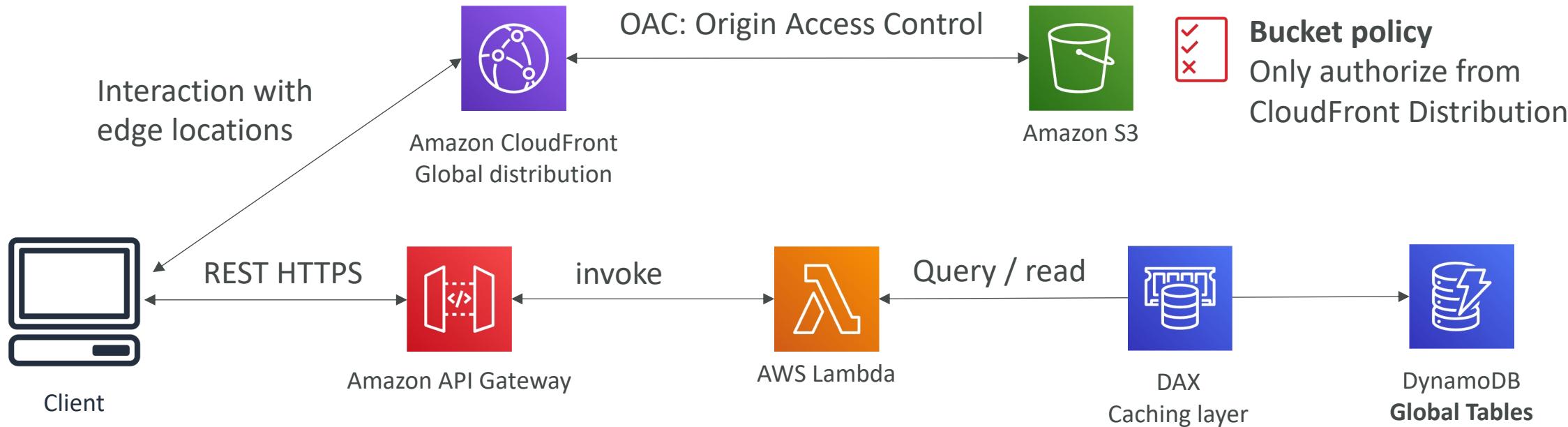
Serving static content, globally, securely



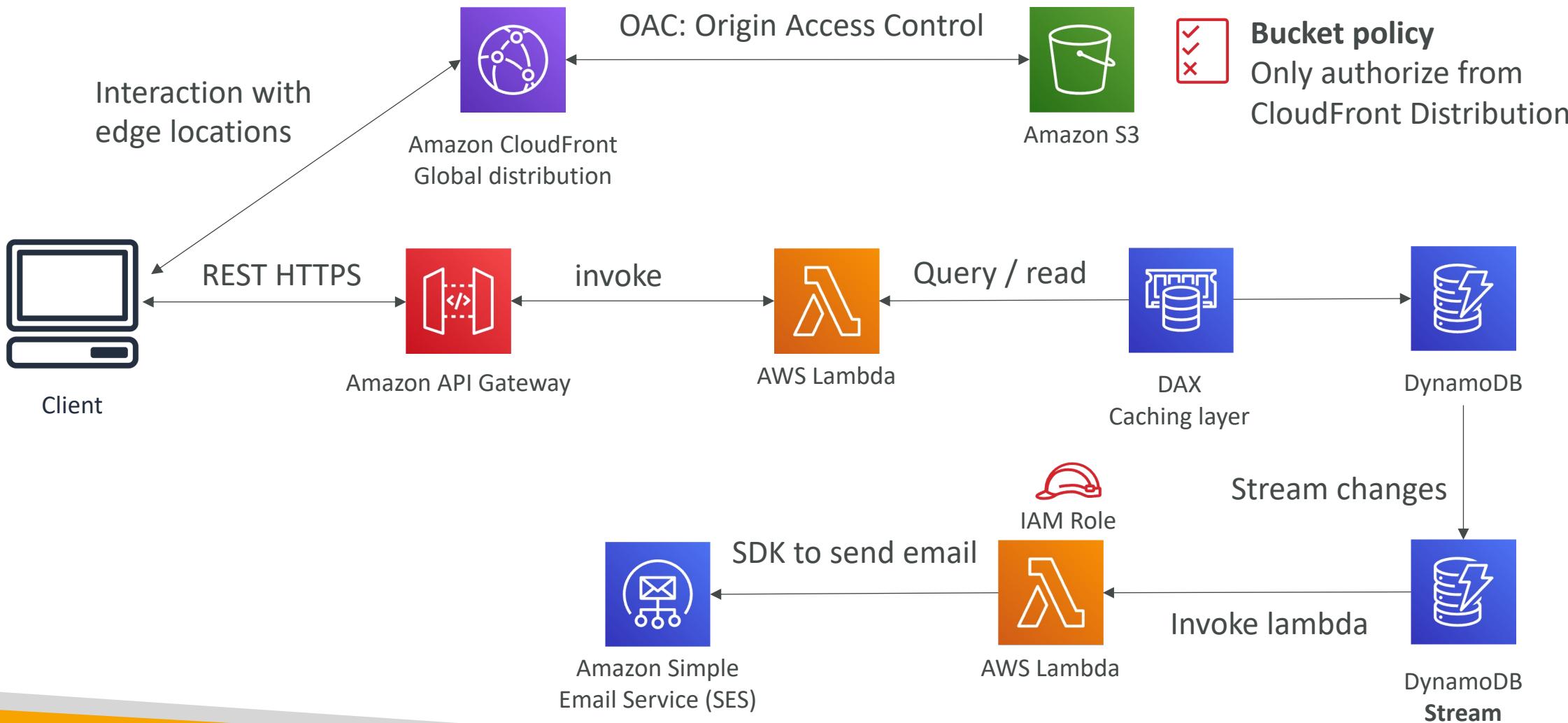
Adding a public serverless REST API



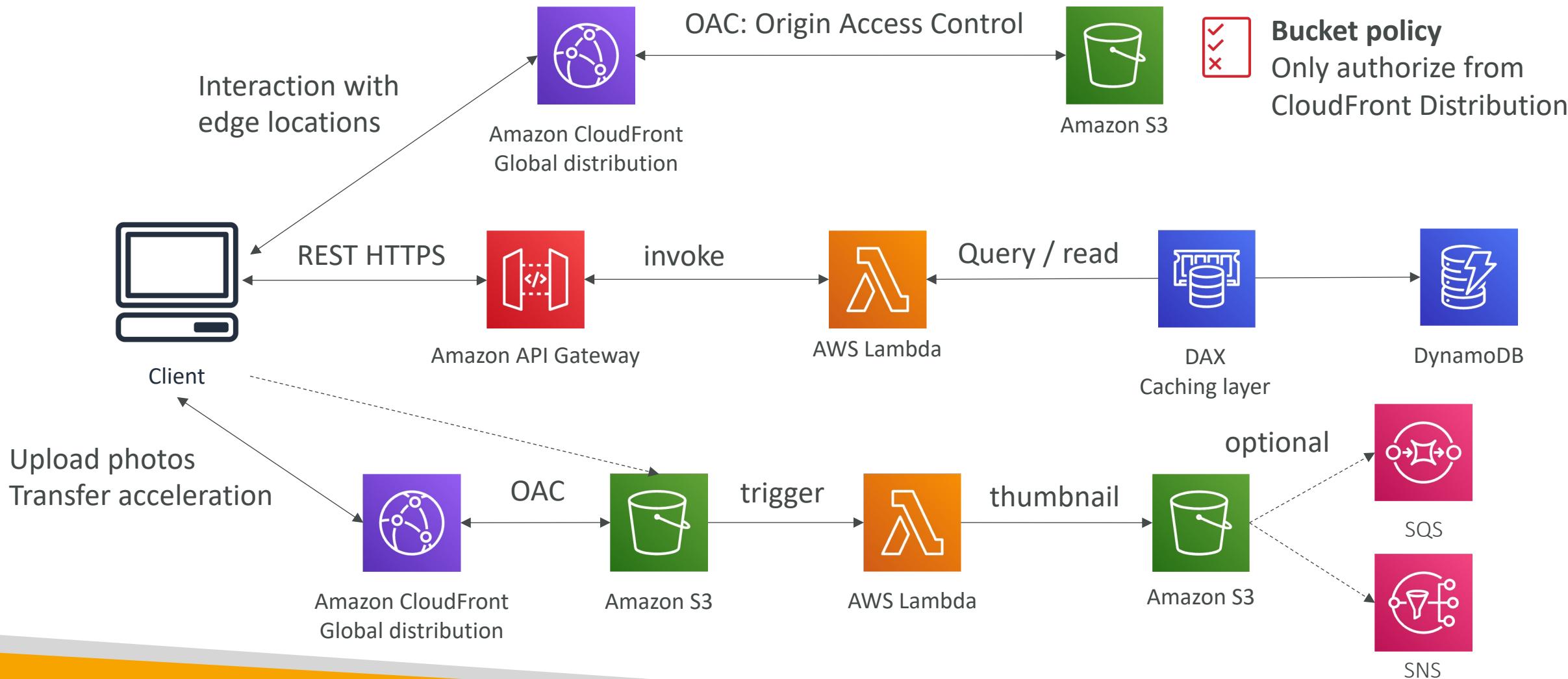
Leveraging DynamoDB Global Tables



User Welcome email flow



Thumbnail Generation flow



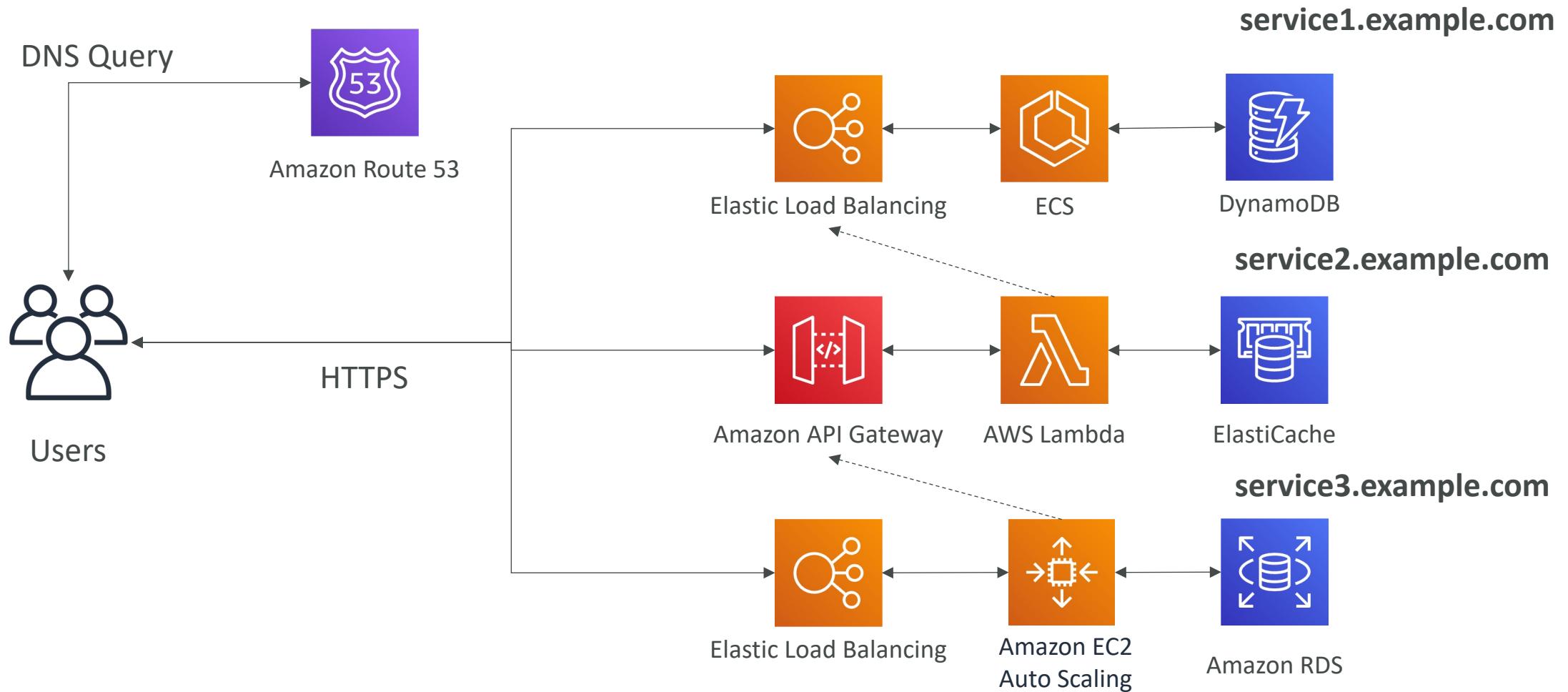
AWS Hosted Website Summary

- 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



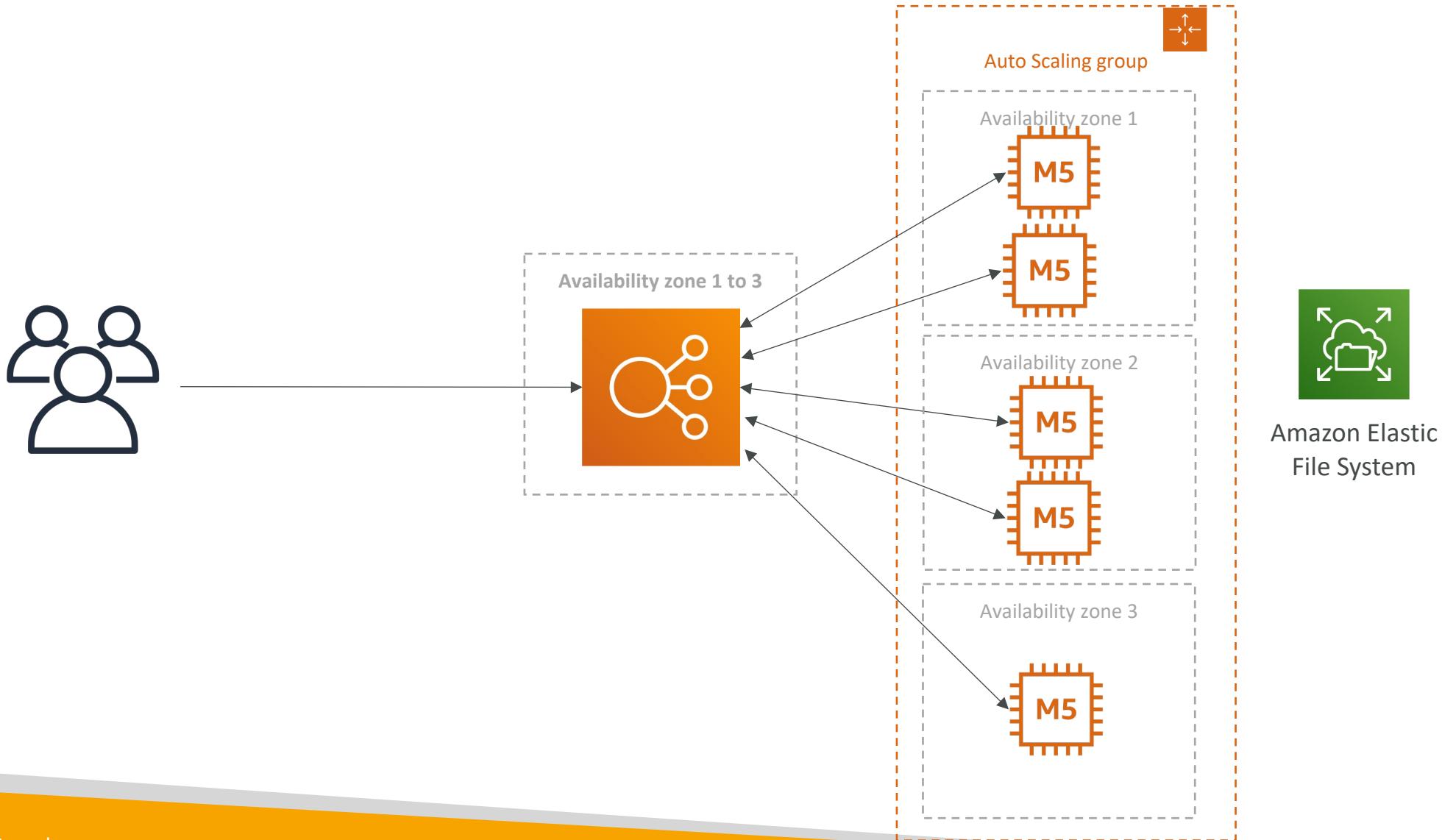
Discussions on Micro Services

- 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

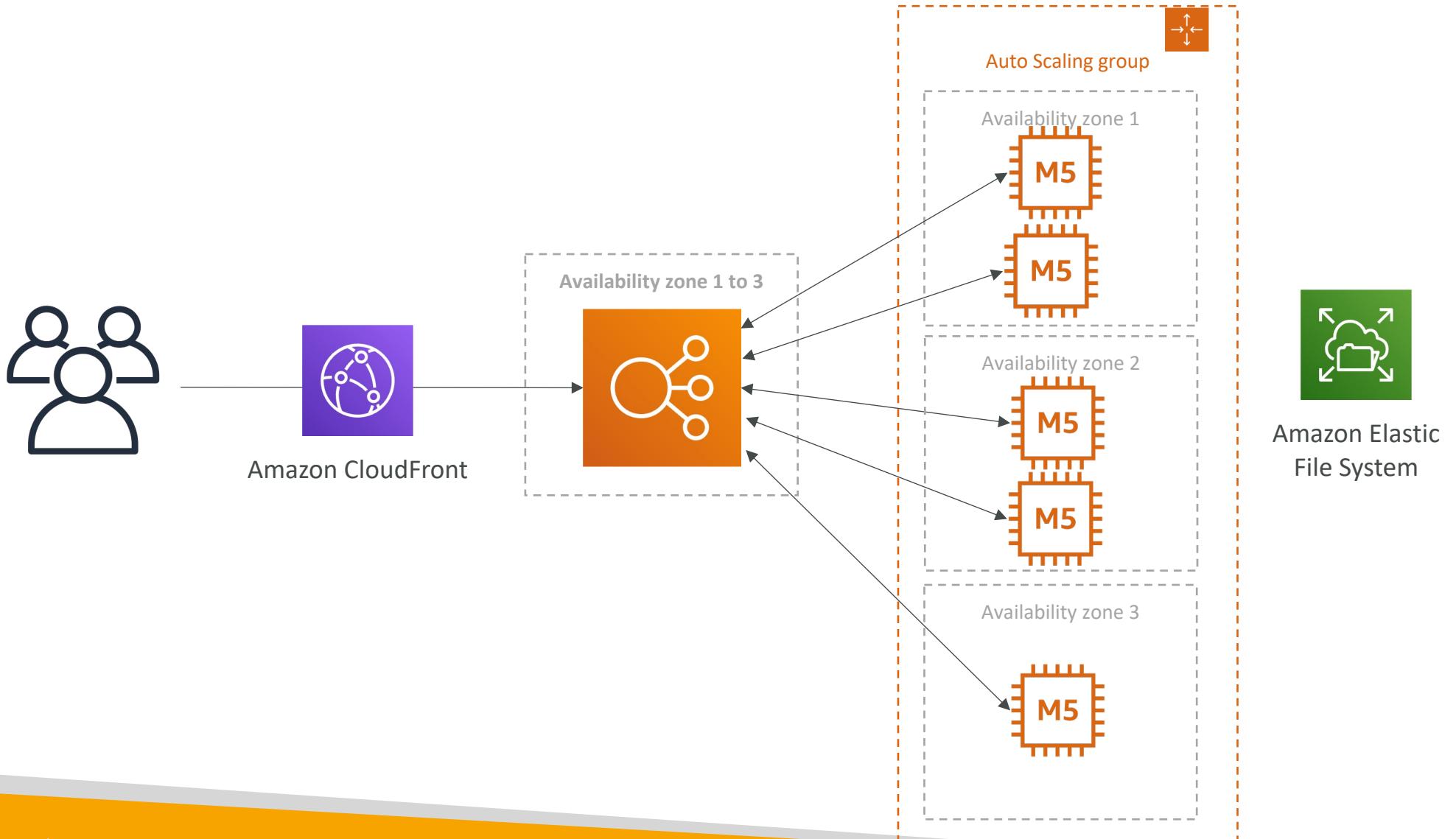
Software updates offloading

- 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

Choosing the Right Database

- 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 – Summary



- Managed PostgreSQL / 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
- 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 – 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 (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 – 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 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

DocumentDB



- Aurora is an “AWS-implementation” of PostgreSQL / MySQL ...
- DocumentDB is the same for MongoDB (which is a NoSQL database)

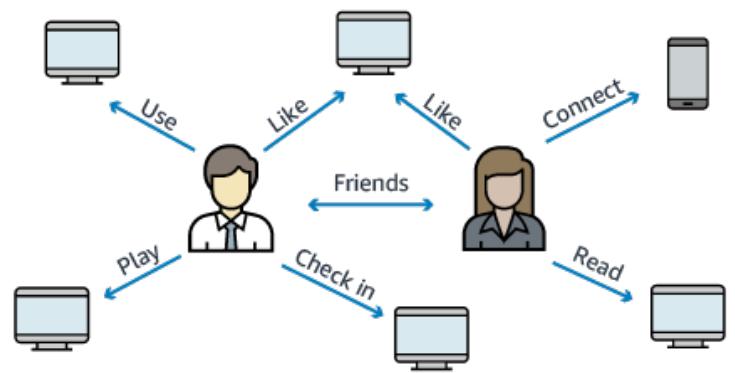
- MongoDB is used to store, query, and index JSON data
- Similar “deployment concepts” as Aurora
- Fully Managed, highly available with replication across 3 AZ
- DocumentDB storage automatically grows in increments of 10GB, up to 64 TB.

- Automatically scales to workloads with millions of requests per seconds

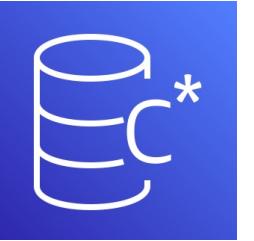
Amazon Neptune



- Fully managed **graph** database
- A popular **graph dataset** would be a **social network**
 - 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



Amazon Keyspaces (for Apache Cassandra)

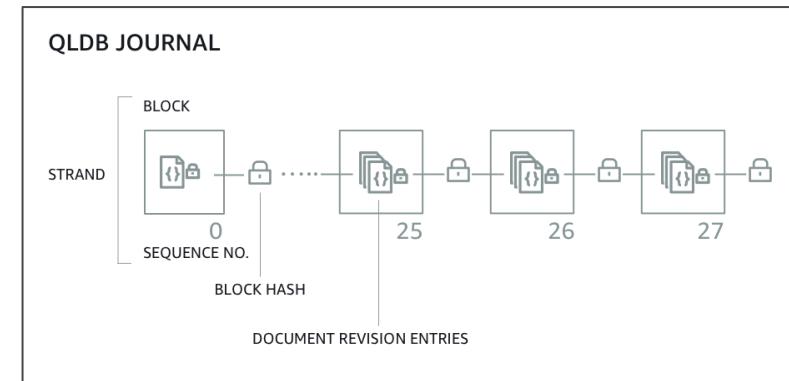


- Apache Cassandra is an open-source NoSQL distributed database
- A managed Apache Cassandra-compatible database service
- Serverless, Scalable, highly available, fully managed by AWS
- Automatically scale tables up/down based on the application's traffic
- Tables are replicated 3 times across multiple 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 history of all the changes made to your application data** over time
- **Immutable** system: no entry can be removed or modified, cryptographically verifiable



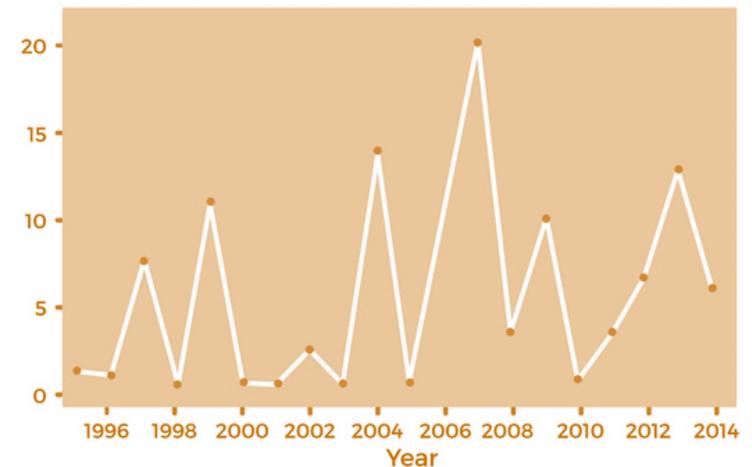
- 2-3x better performance than common ledger blockchain frameworks, manipulate data using SQL
- Difference with Amazon Managed Blockchain: **no decentralization component**, in accordance with financial regulation rules

<https://docs.aws.amazon.com/qldb/latest/developerguide/ledger-structure.html>

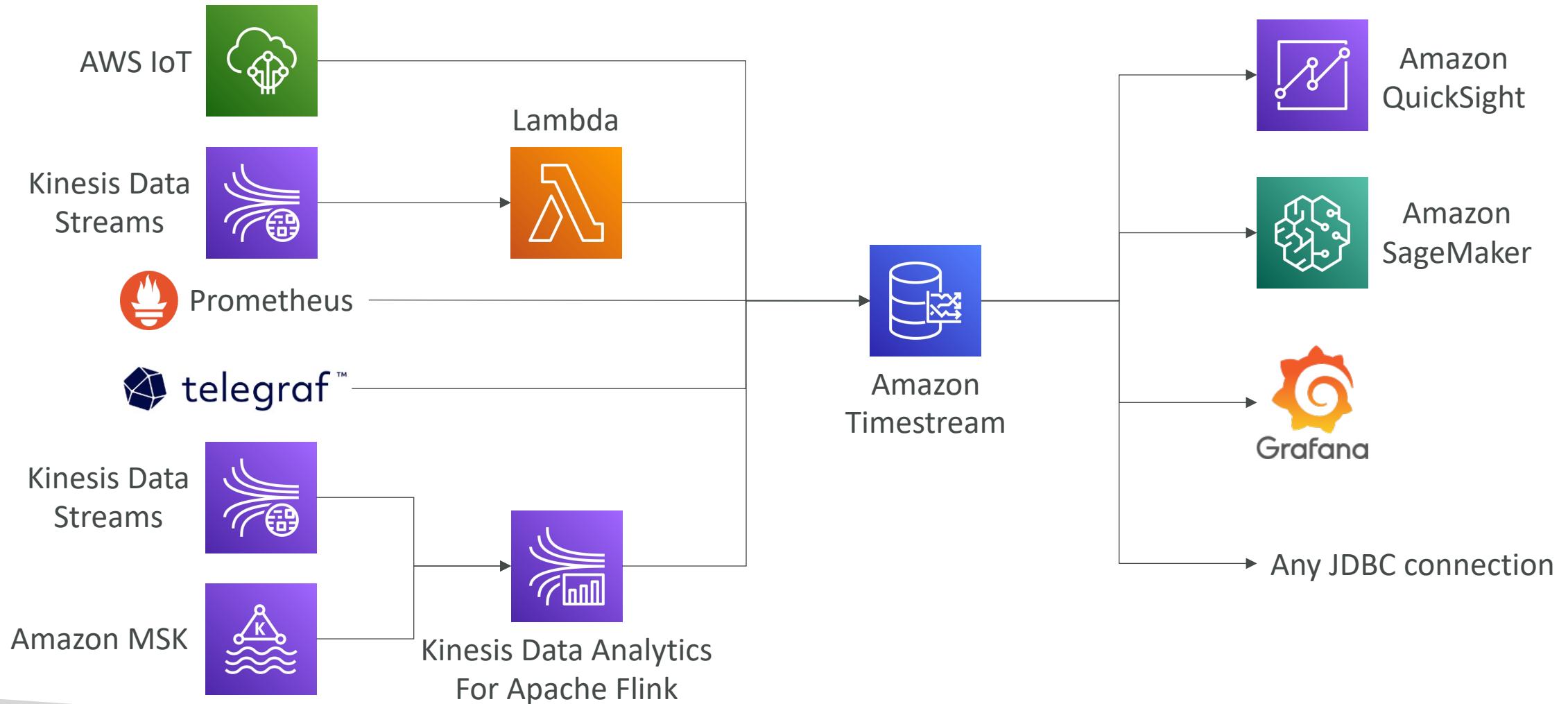
Amazon Timestream



- 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 applications, real-time analytics, ...



Amazon Timestream – Architecture

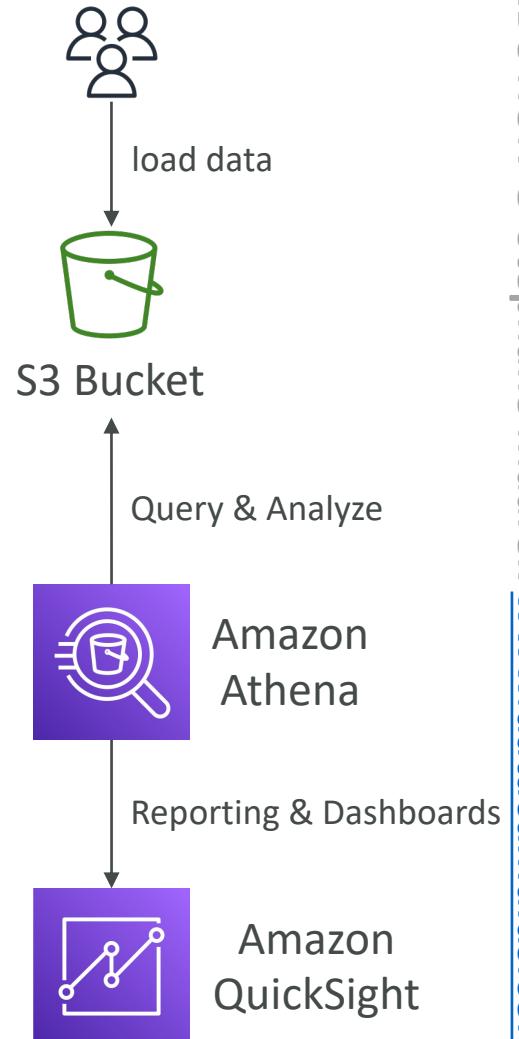


Data & Analytics

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, Avro, and Parquet
- Pricing: \$5.00 per TB of data scanned
- Commonly used with Amazon Quicksight for reporting/dashboards
- **Use cases:** Business intelligence / analytics / reporting, analyze & query VPC Flow Logs, ELB Logs, CloudTrail trails, etc...
- **Exam Tip:** analyze data in S3 using serverless SQL, use Athena

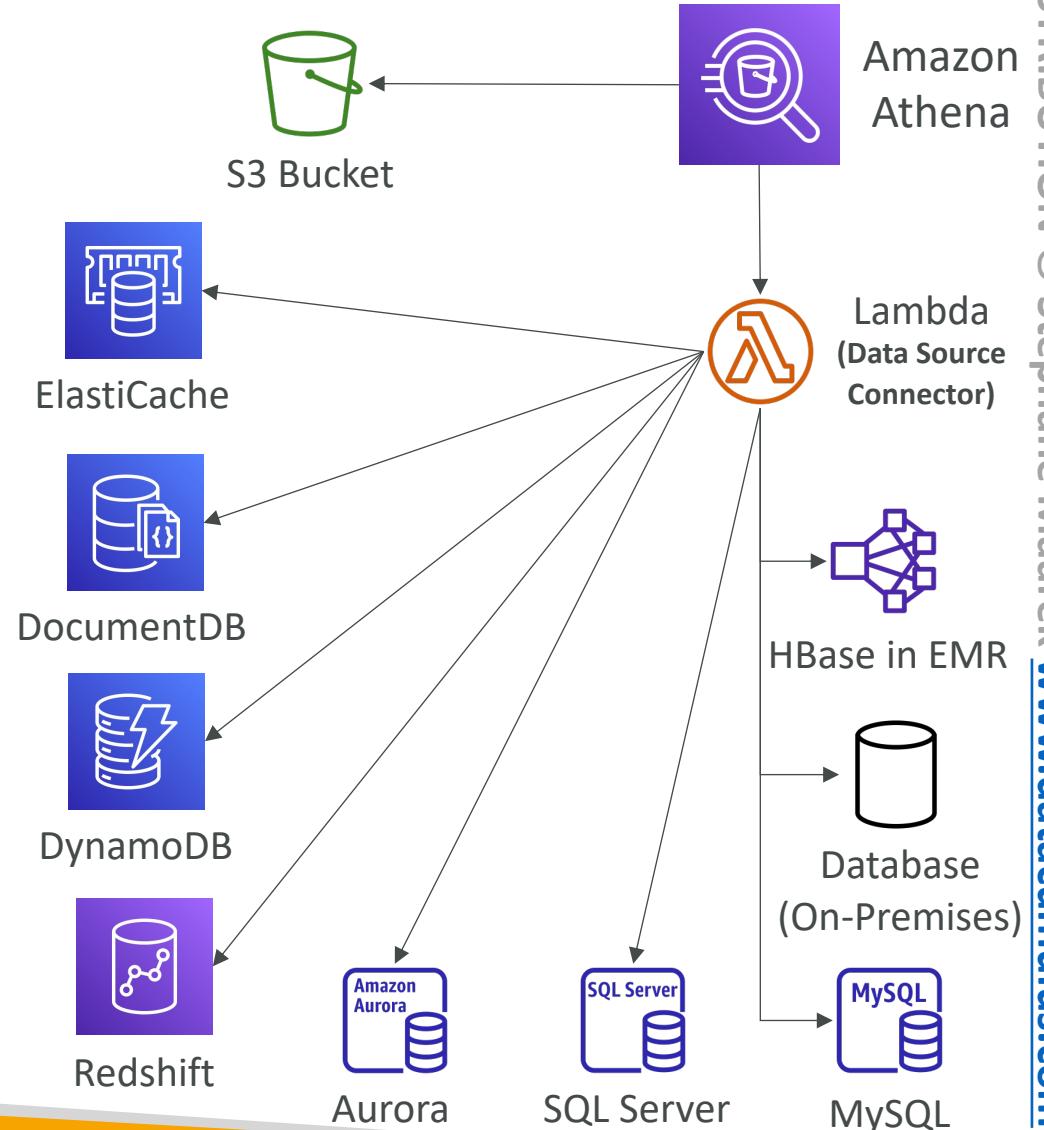


Amazon Athena – Performance Improvement

- Use **columnar data** for cost-savings (less scan)
 - Apache Parquet or ORC is recommended
 - Huge performance improvement
 - Use Glue to convert your data to 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>
 /etc...
 - 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

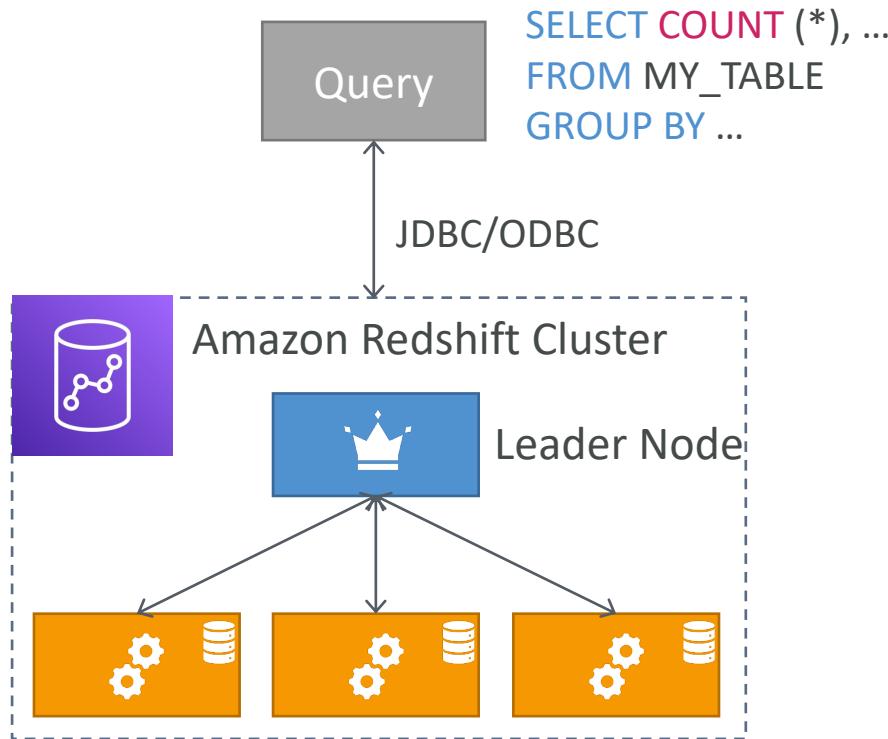


Redshift Overview



- 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

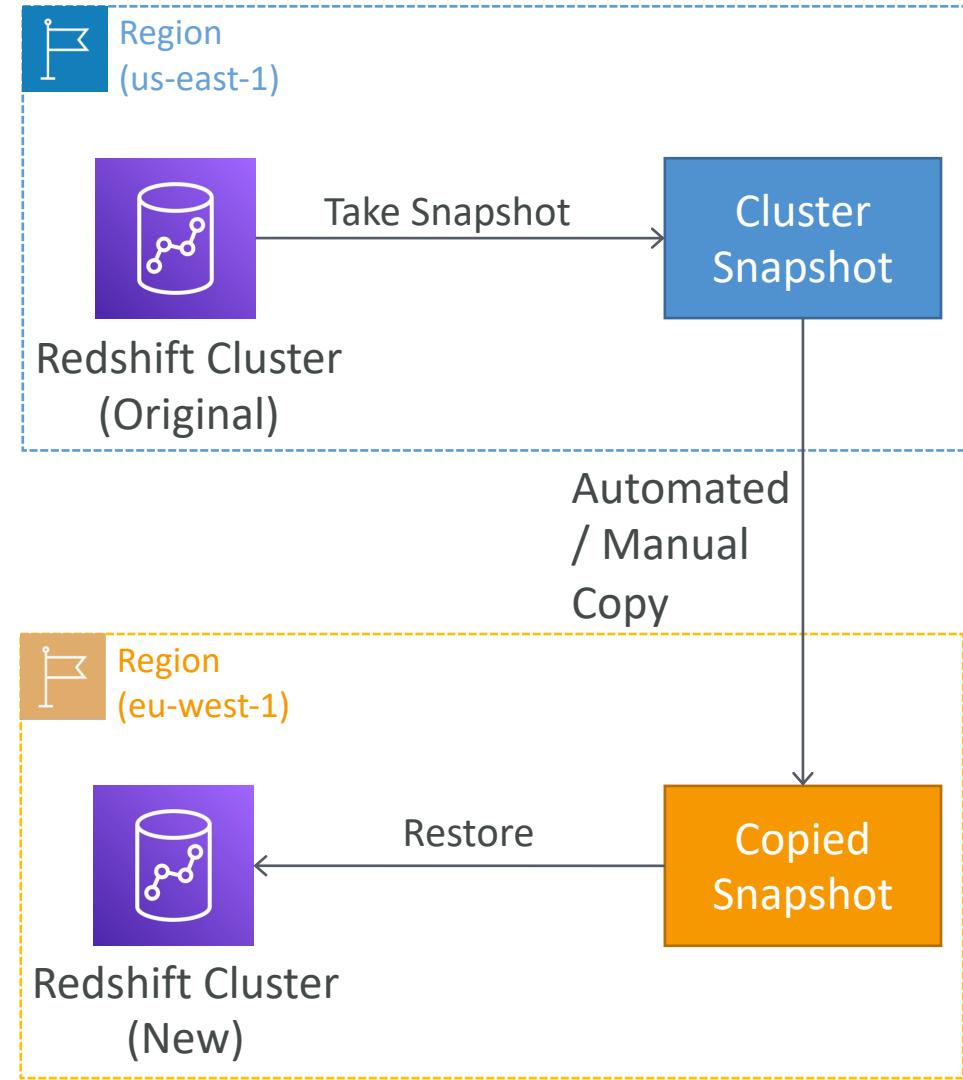
Redshift Cluster



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

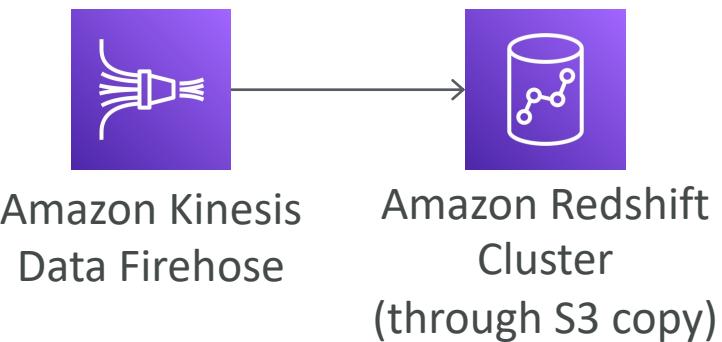
Redshift – Snapshots & DR

- 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

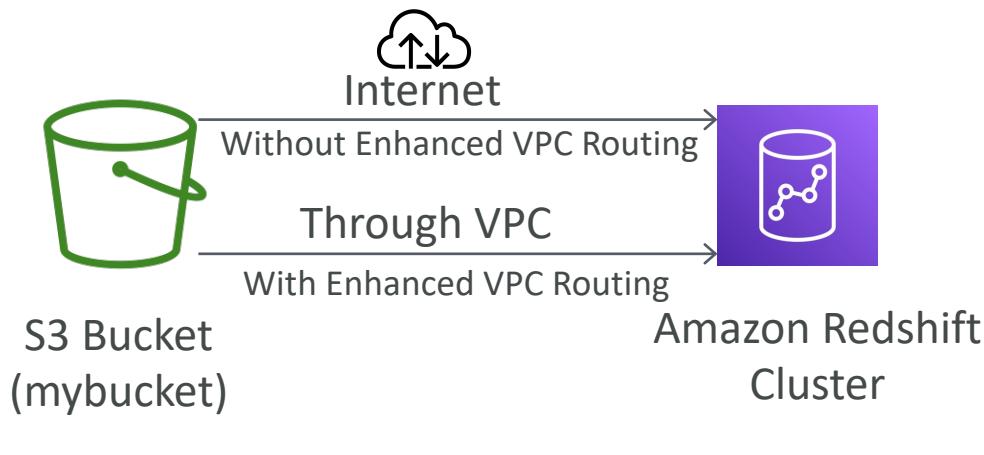


Loading data into Redshift: Large inserts are MUCH better

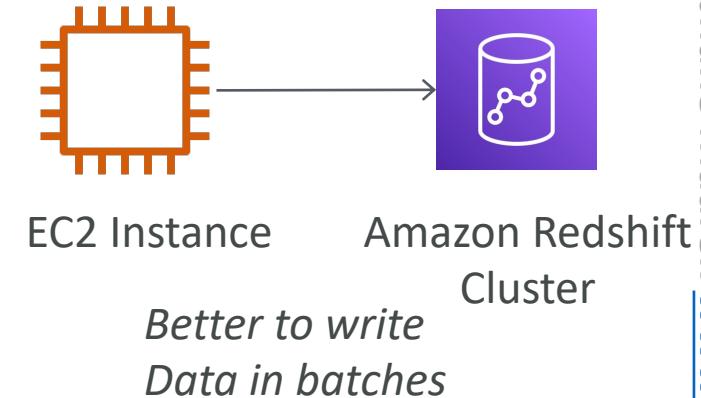
Amazon Kinesis Data Firehose



S3 using COPY command



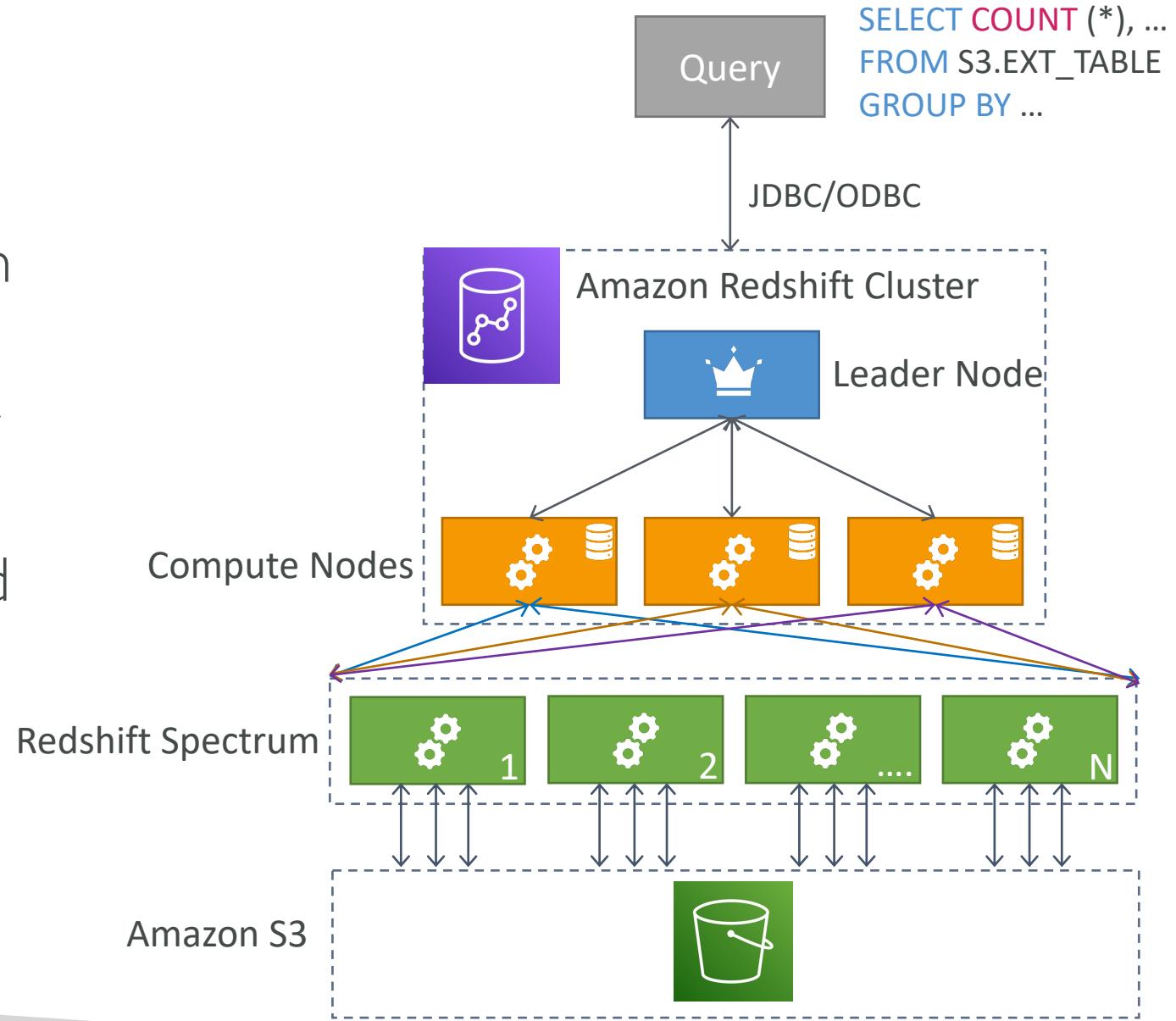
EC2 Instance JDBC driver



```
copy customer
from 's3://mybucket/mydata'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

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



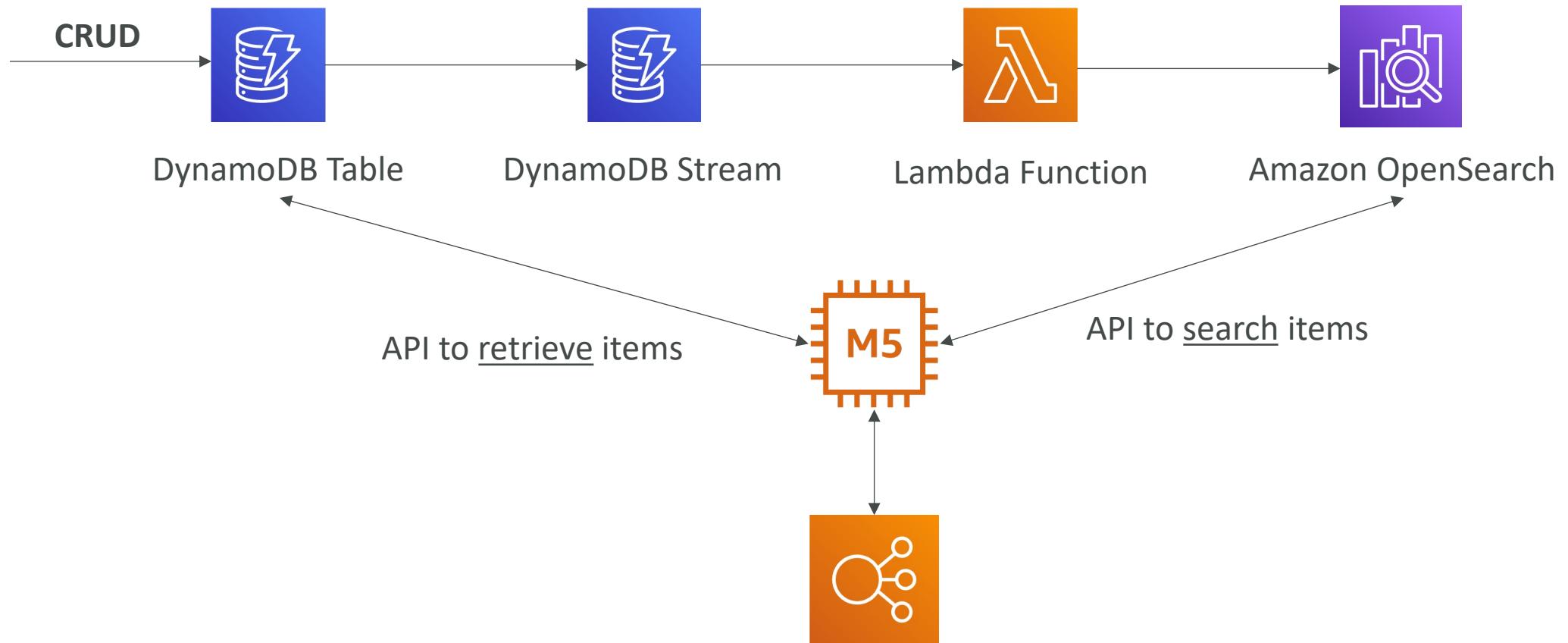
Amazon OpenSearch Service



- *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
- 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)

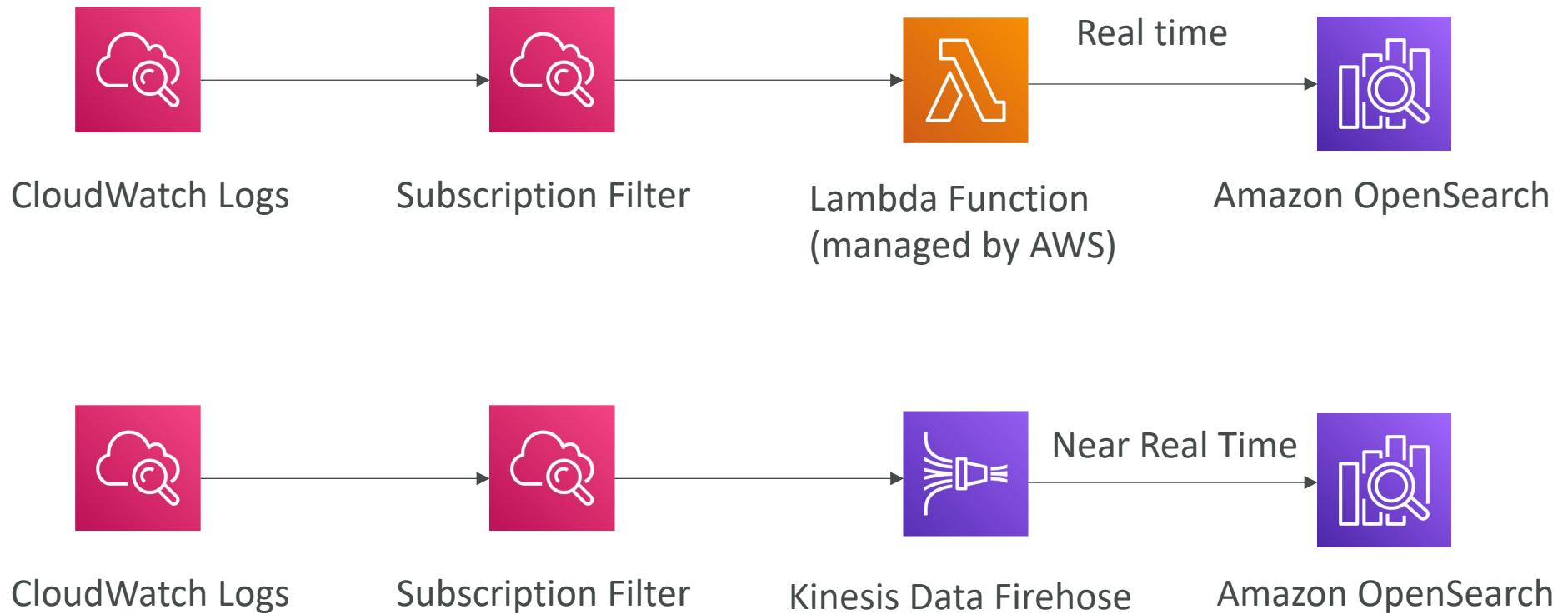
OpenSearch patterns

DynamoDB



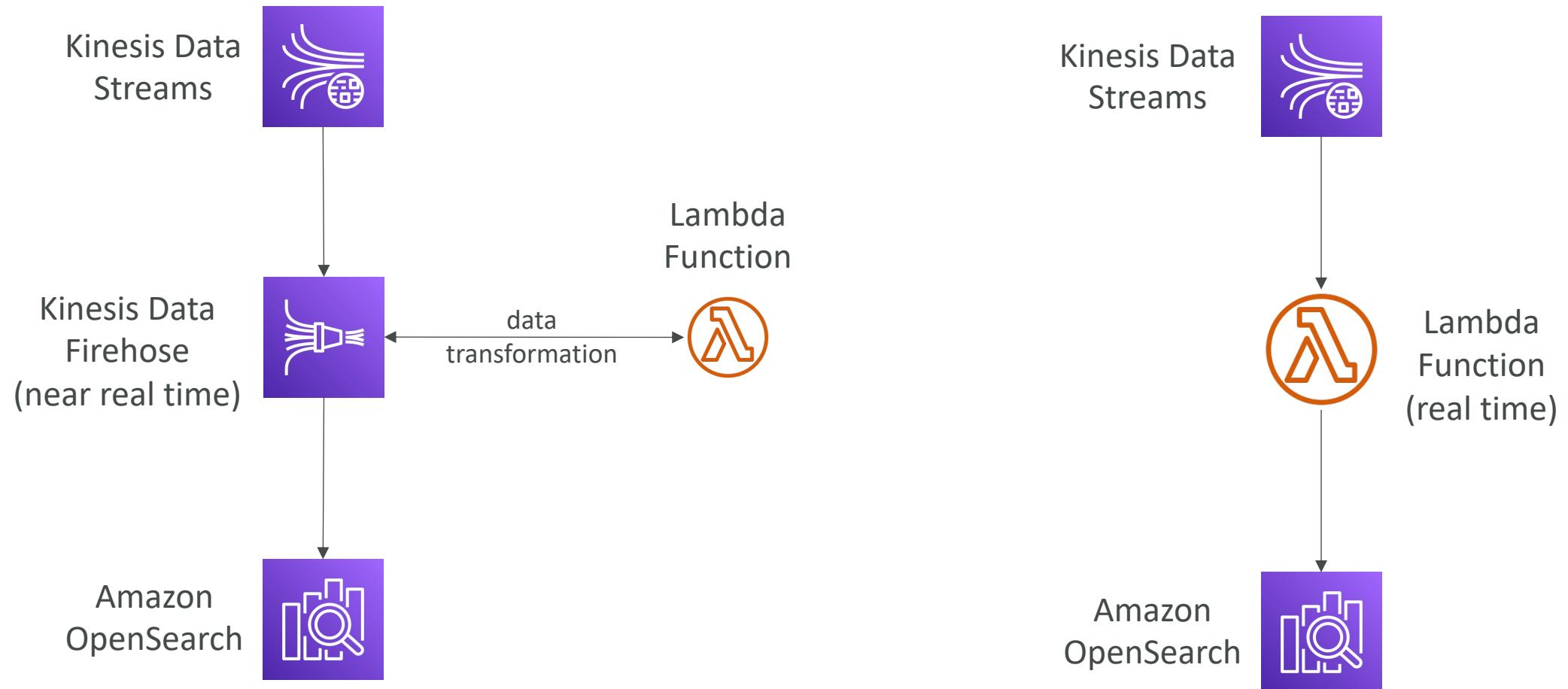
OpenSearch patterns

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...

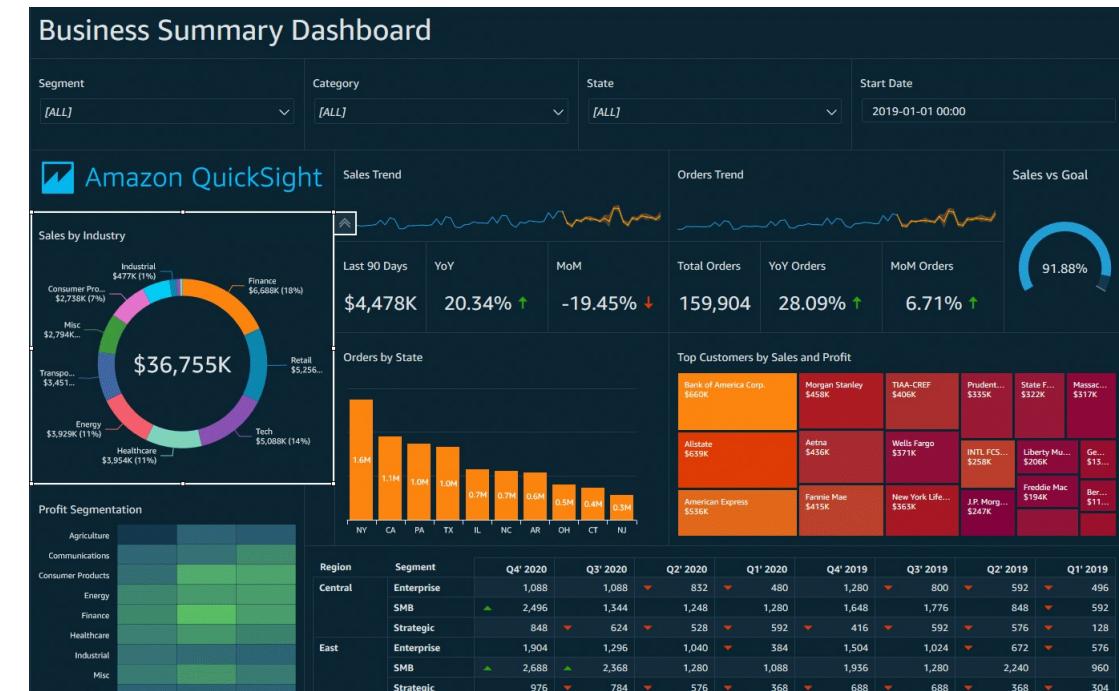
Amazon EMR – Node types & purchasing

- **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

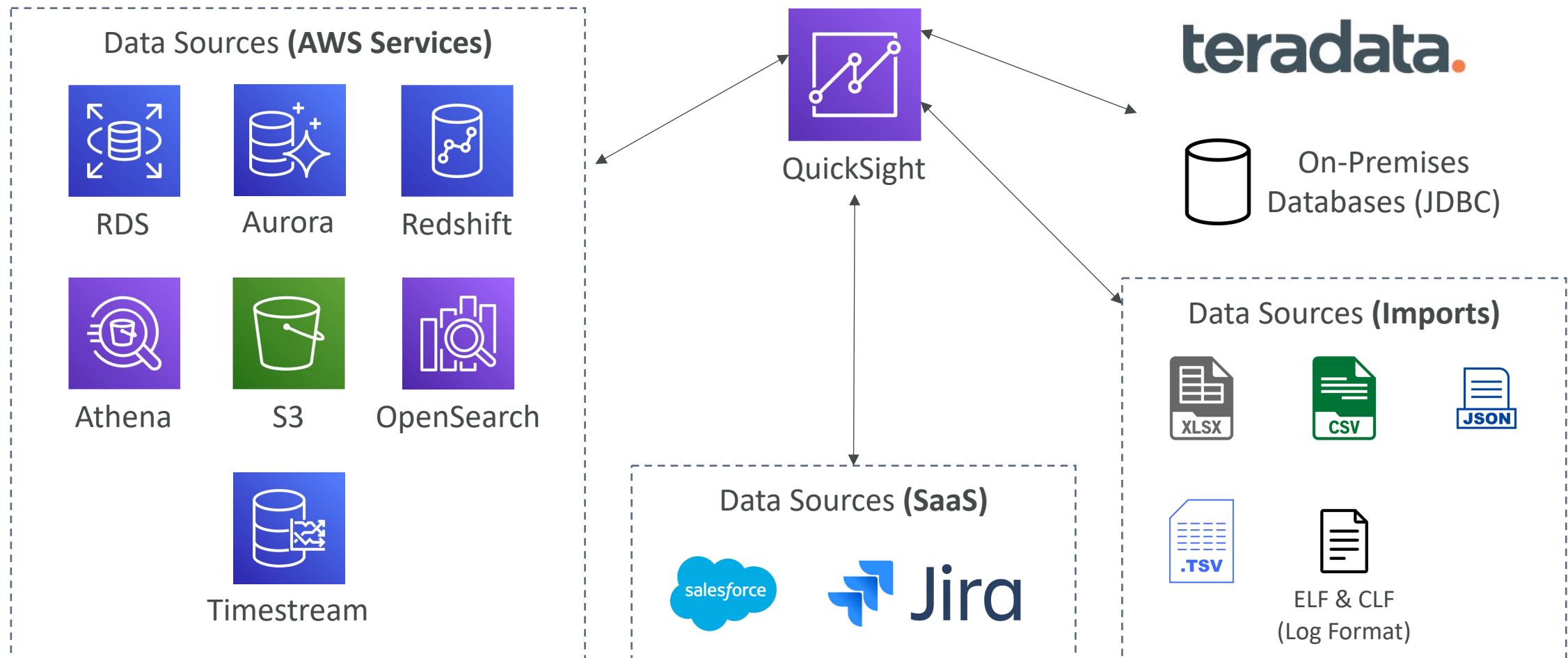


- Serverless machine learning-powered business intelligence service to create interactive dashboards
- Fast, automatically scalable, embeddable, with per-session pricing
- Use cases:
 - Business analytics
 - Building visualizations
 - Perform ad-hoc analysis
 - Get business insights using data
- Integrated 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)



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

QuickSight Integrations



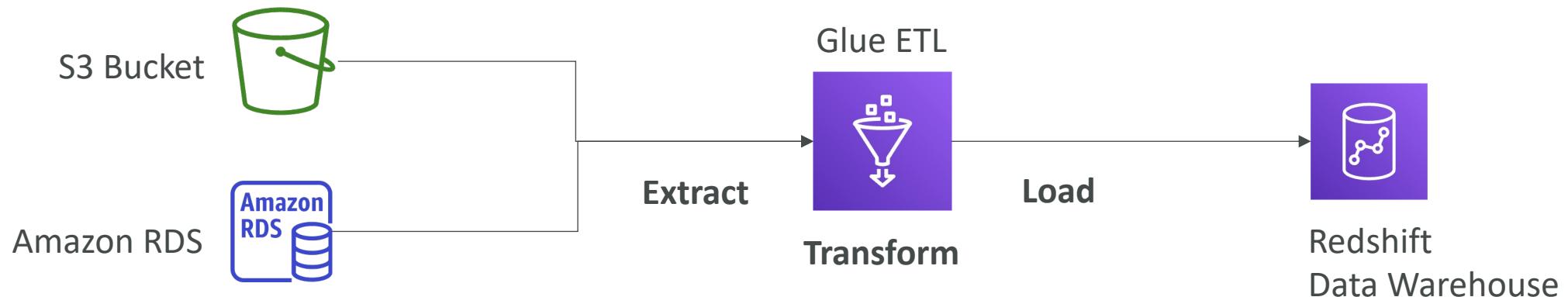
QuickSight – Dashboard & Analysis

- 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

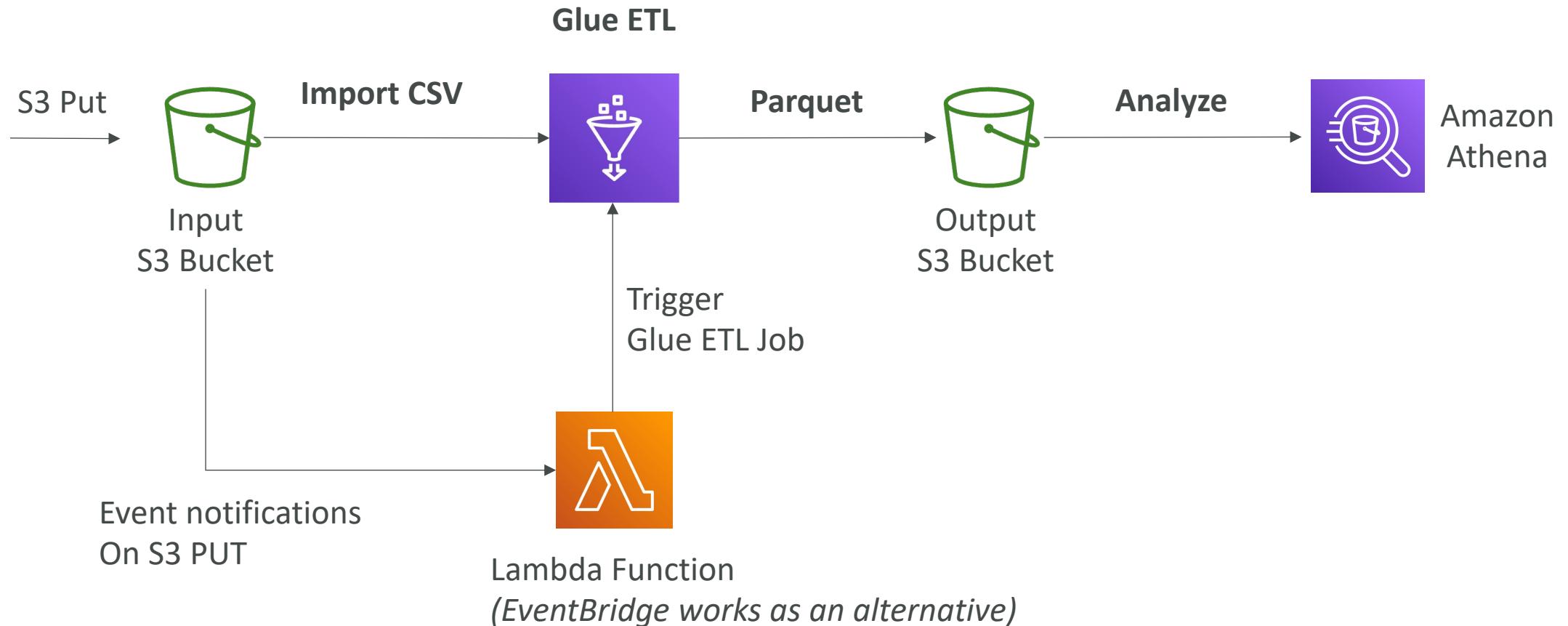
AWS Glue



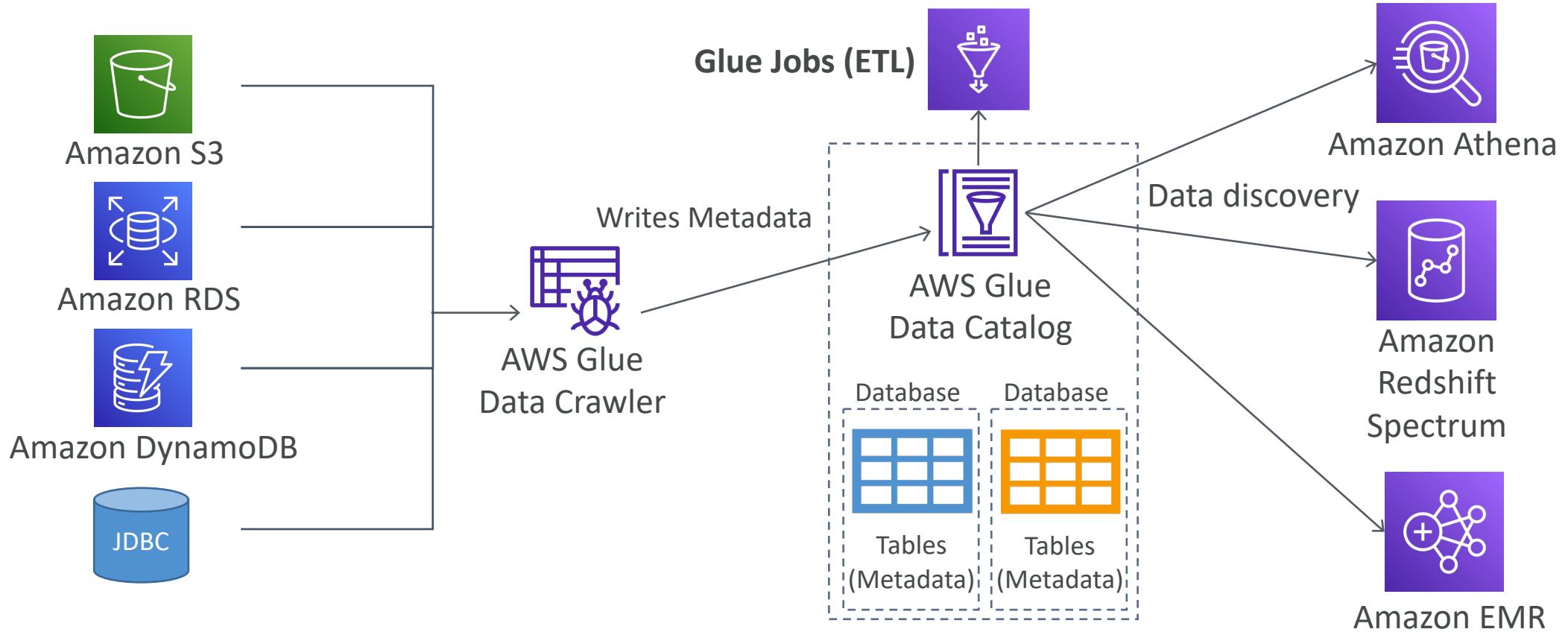
- Managed **extract, transform, and load (ETL)** service
- Useful to prepare and transform data for analytics
- Fully **serverless** service



AWS Glue – Convert data into Parquet format



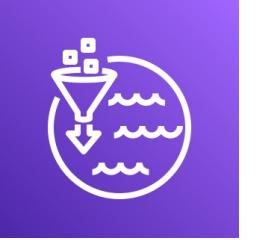
Glue Data Catalog: catalog of datasets



Glue – things to know at a high-level

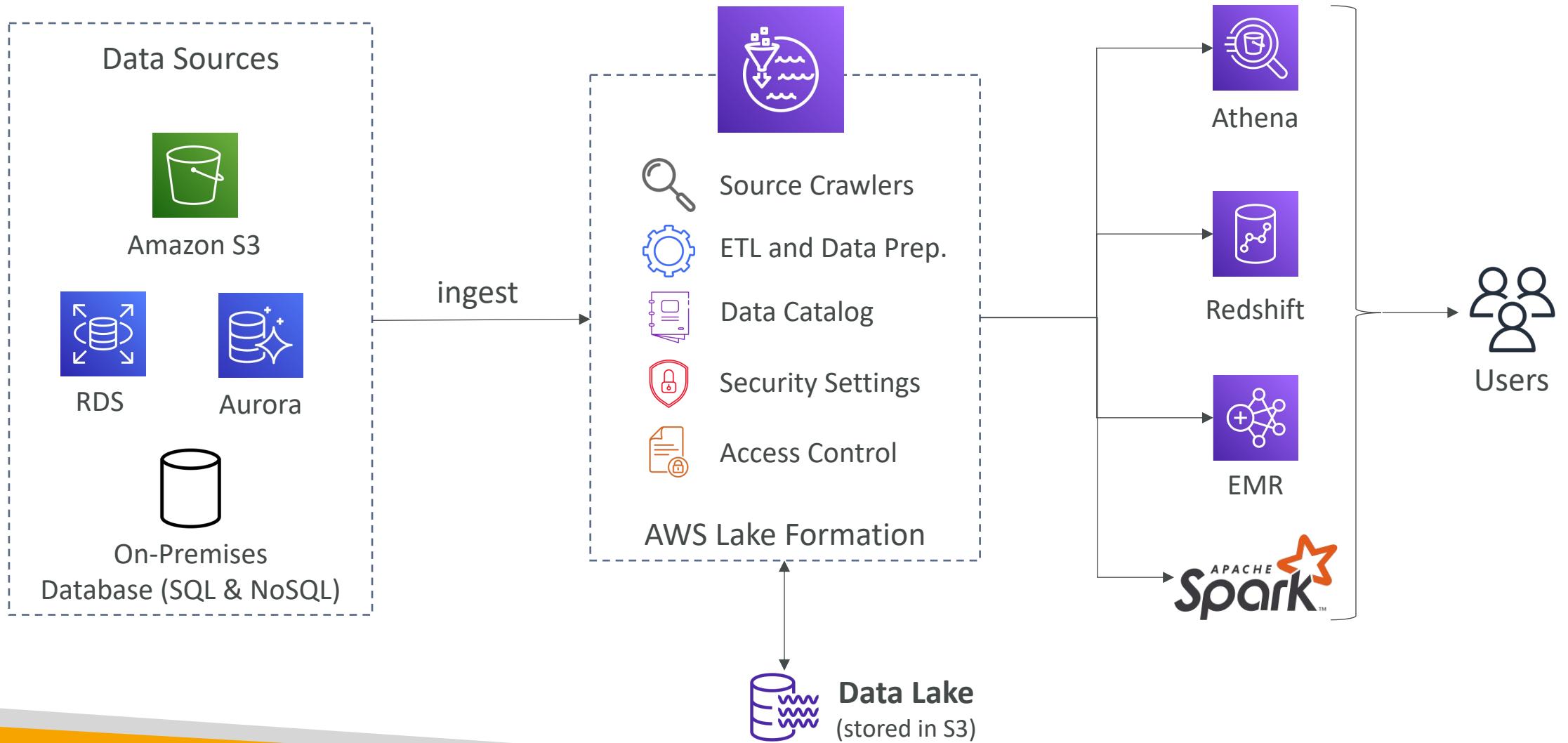
- **Glue Job Bookmarks:** prevent re-processing old data
- **Glue Elastic Views:**
 - Combine and replicate data across multiple data stores using SQL
 - No custom code, Glue monitors for changes in the source data, serverless
 - Leverages a “virtual table” (materialized view)
- **Glue DataBrew:** clean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



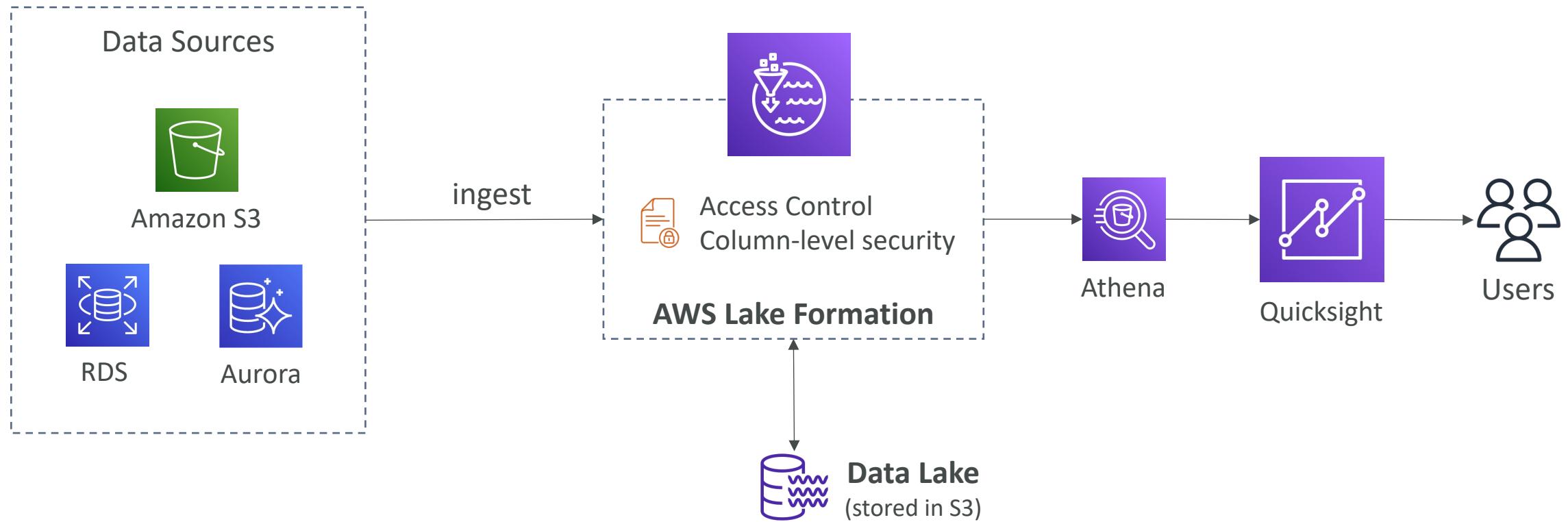
- Data lake = central place to have all your data for analytics purposes
- Fully managed service that makes it easy to setup a **data lake** in days
- Discover, cleanse, transform, and ingest data into your Data Lake
- It automates many complex manual steps (collecting, cleansing, moving, cataloging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

AWS Lake Formation

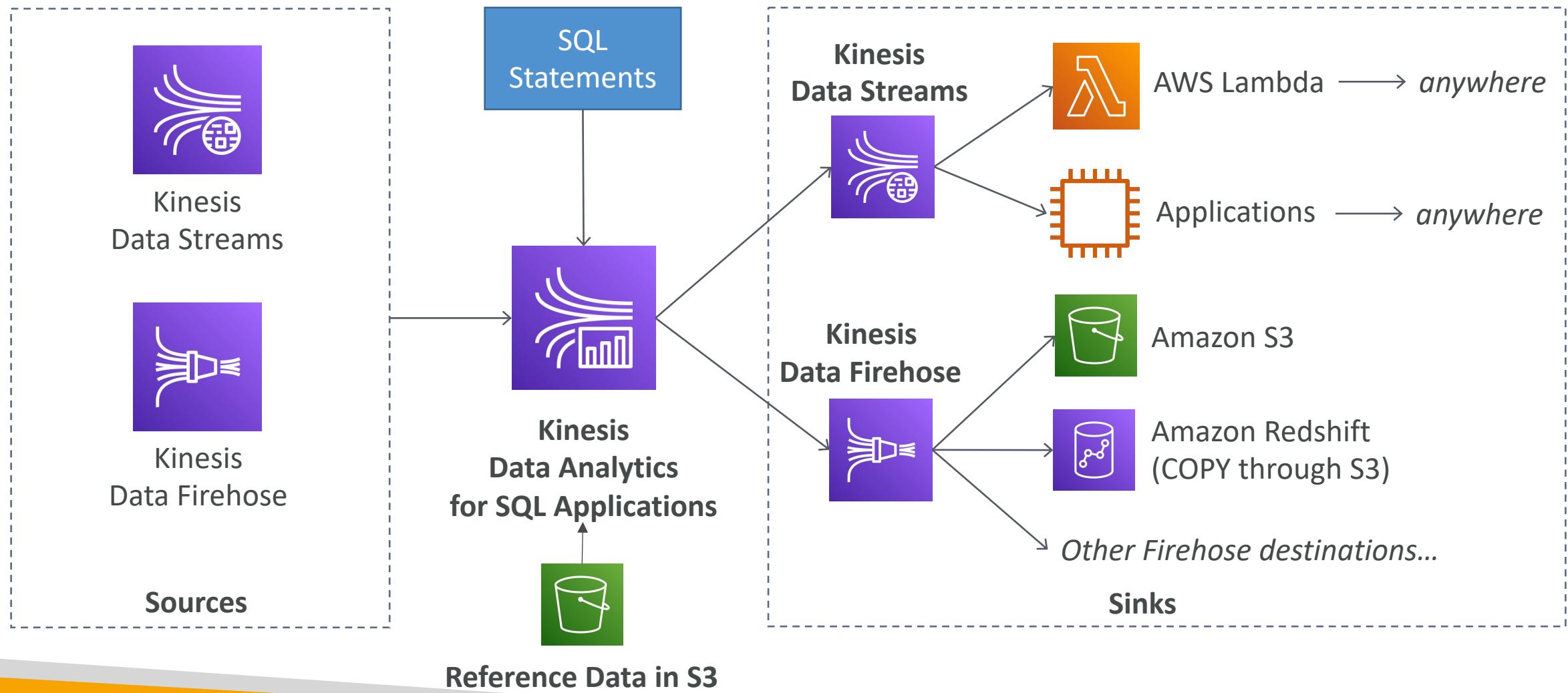


AWS Lake Formation

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 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

Kinesis Data Analytics for Apache Flink

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



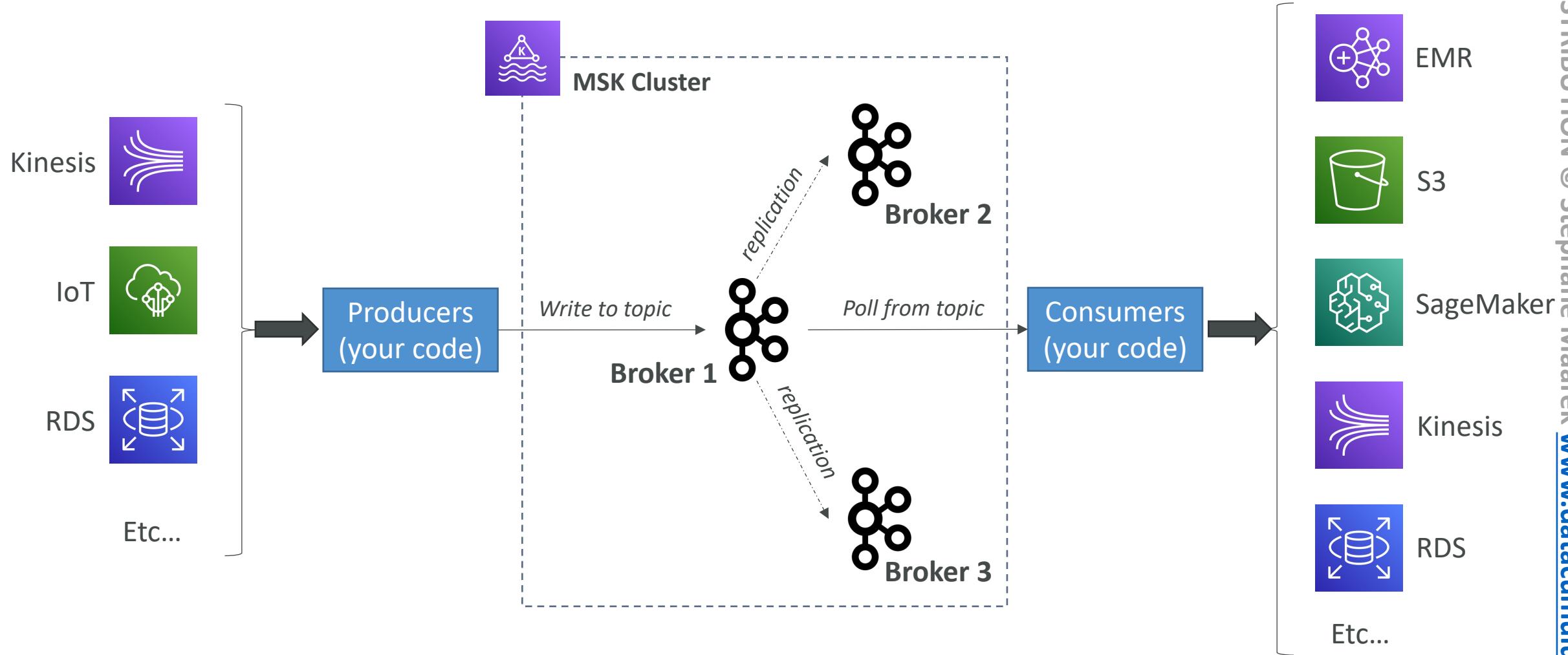
- Run any Apache Flink application on a managed cluster on AWS
 - provisioning compute resources, parallel computation, automatic scaling
 - application backups (implemented as checkpoints and snapshots)
 - Use any Apache Flink programming features
 - Flink does not read from Firehose (use Kinesis Analytics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

Amazon 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

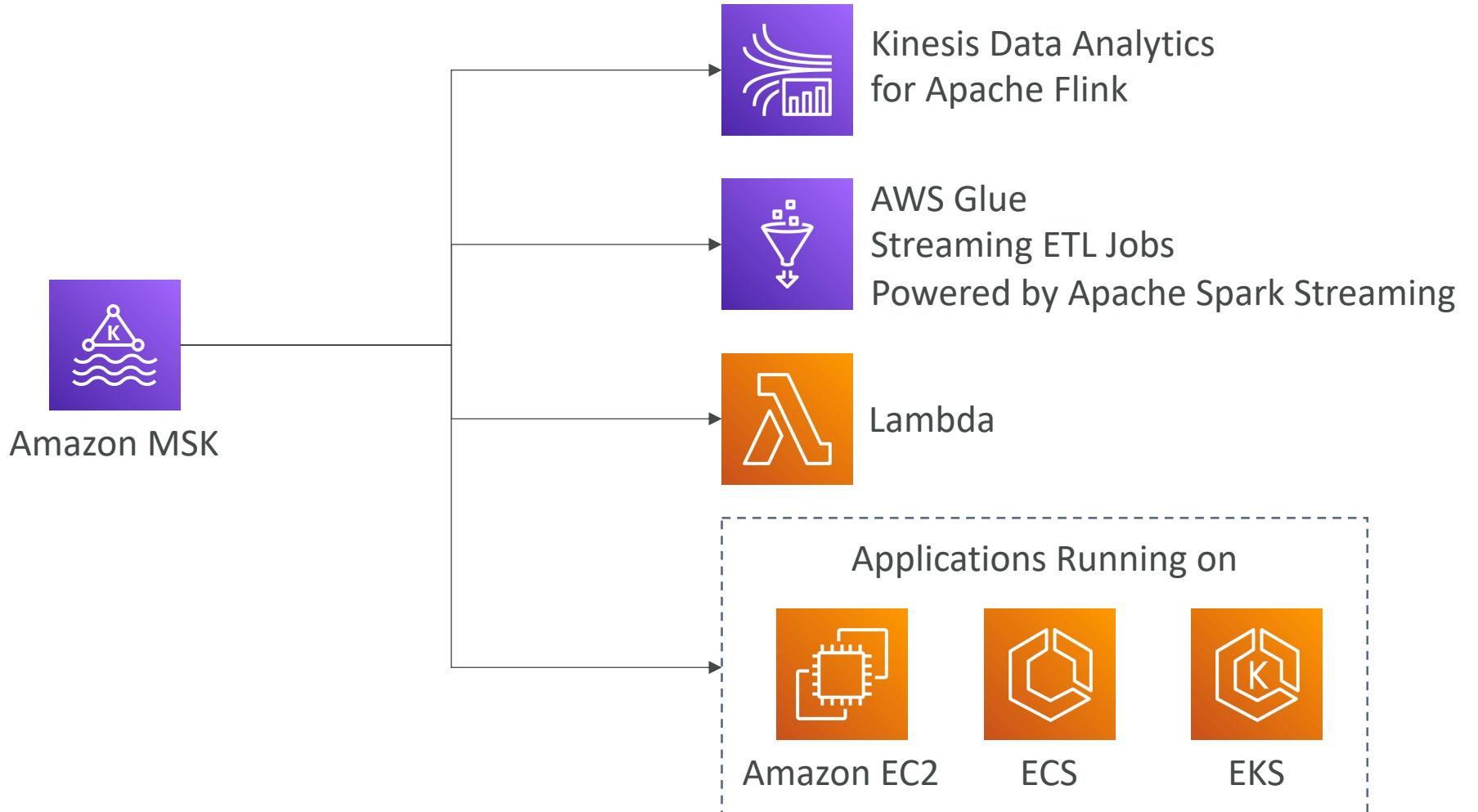
- 1 MB message size limit
- Data Streams with Shards
- Shard Splitting & Merging
- TLS In-flight encryption
- KMS at-rest encryption



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

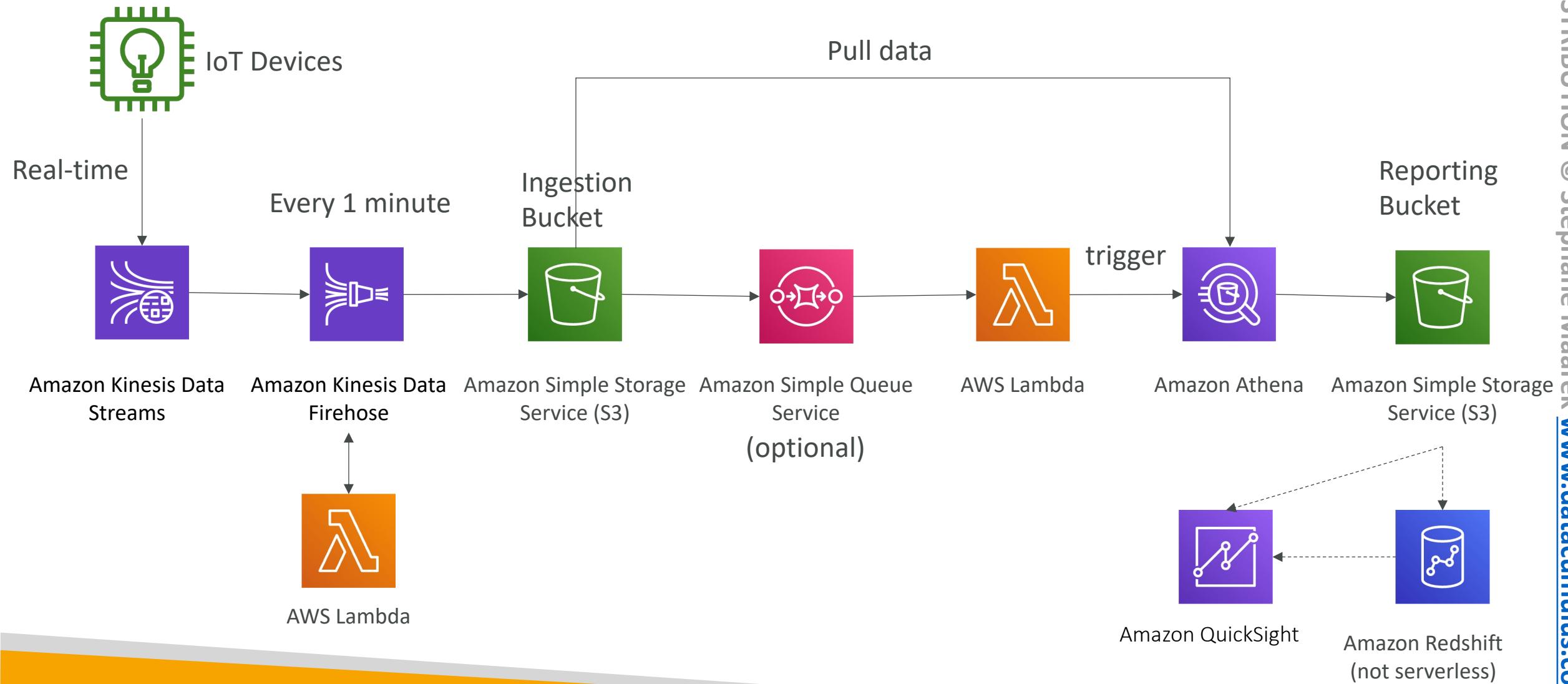
Amazon MSK Consumers



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



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...

Machine Learning

Amazon Rekognition



- Find objects, 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 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



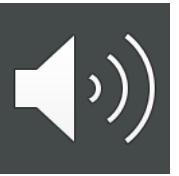
*"Hello my name is Stéphane.
I hope you're enjoying the course!"*

Amazon Polly



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

*Hi! My name is Stéphane
and this is a demo of Amazon Polly*



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 **SynthesizeSpeech** 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.

Source language

Auto (auto) ▾

Hi my name is Stéphane

Target language

French (fr) ▾

Bonjour, je m'appelle Stéphane.

Portuguese (pt) ▾

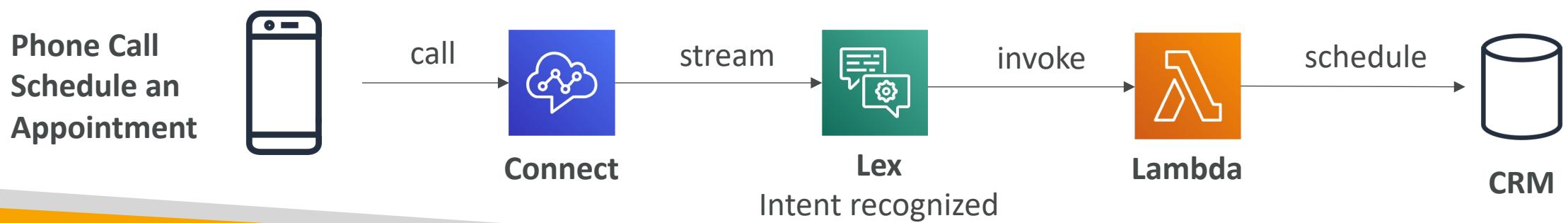
Oi, meu nome é Stéphane.

Hindi (hi) ▾

हाय मेरा नाम स्टीफन है

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





Amazon Comprehend

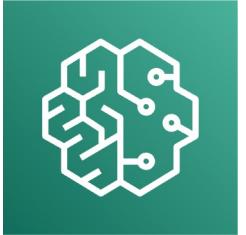
- For Natural Language Processing – NLP
- Fully managed and serverless service
- Uses machine learning to find insights and relationships in text
 - Language of the text
 - Extracts key phrases, places, people, brands, or events
 - Understands how positive or negative the text is
 - Analyzes text using 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



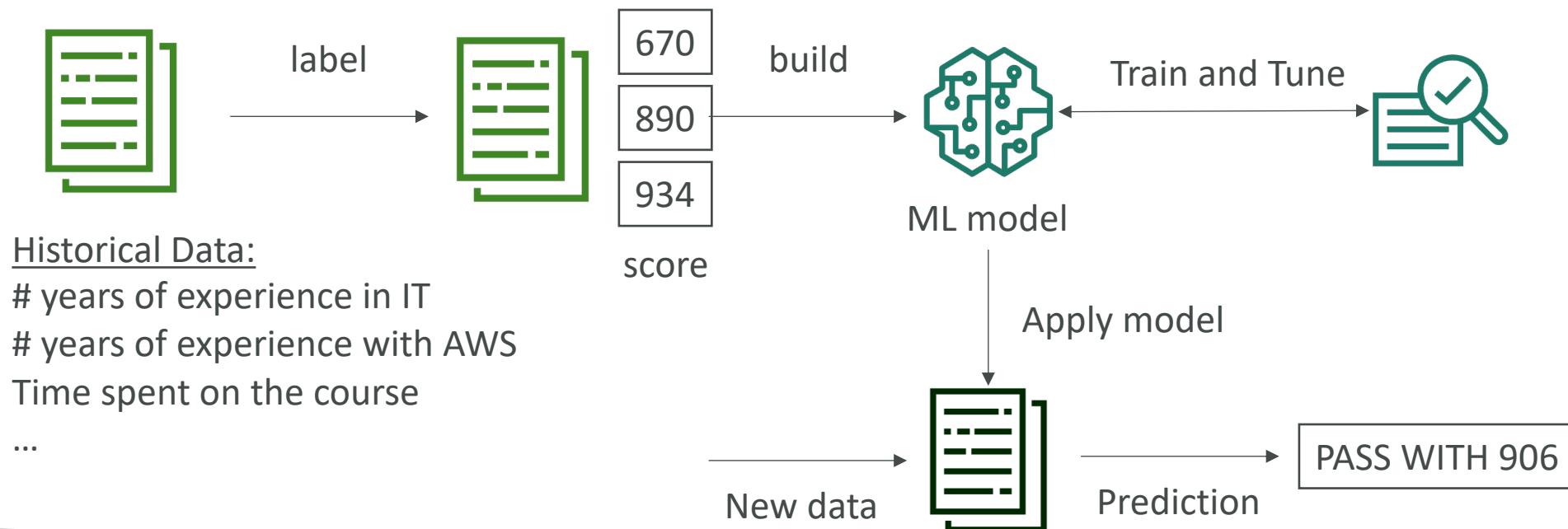
Amazon Comprehend Medical

- 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, ...

Historical Time-series Data:

Product features

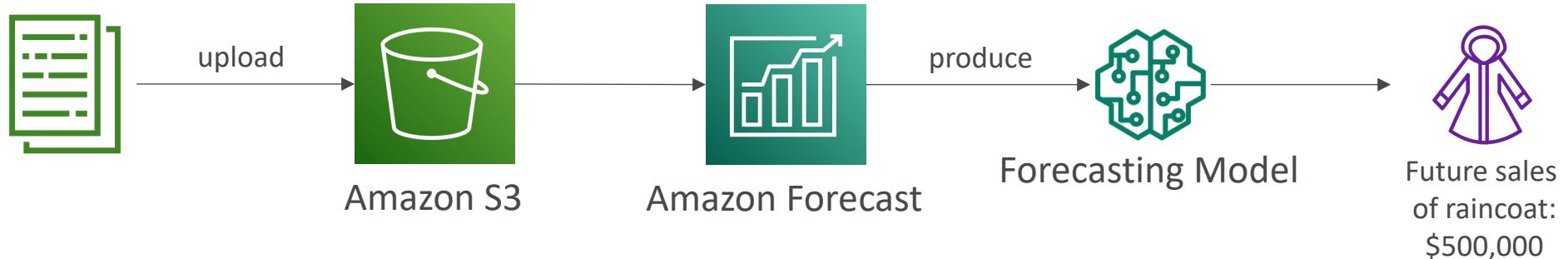
Prices

Discounts

Website traffic

Store locations

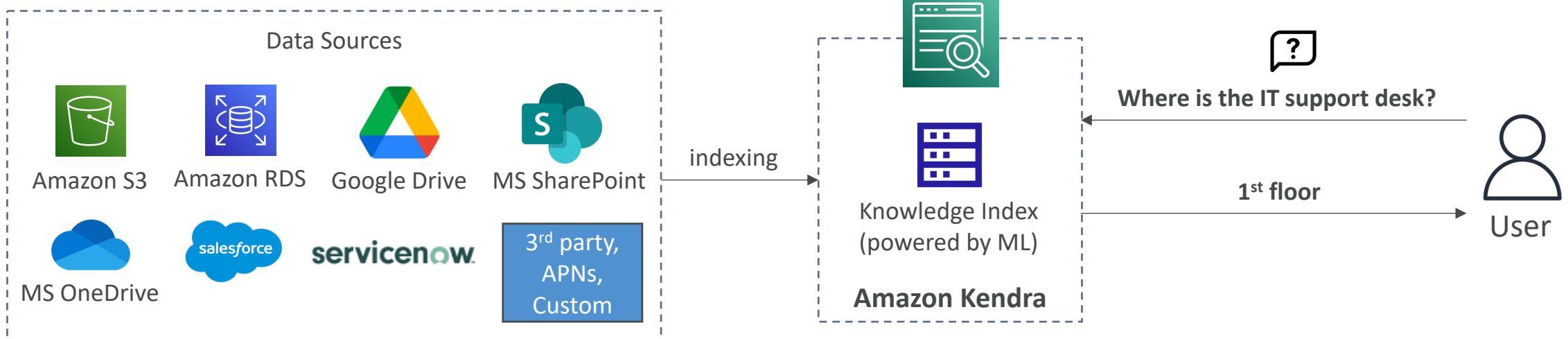
...



Amazon Kendra



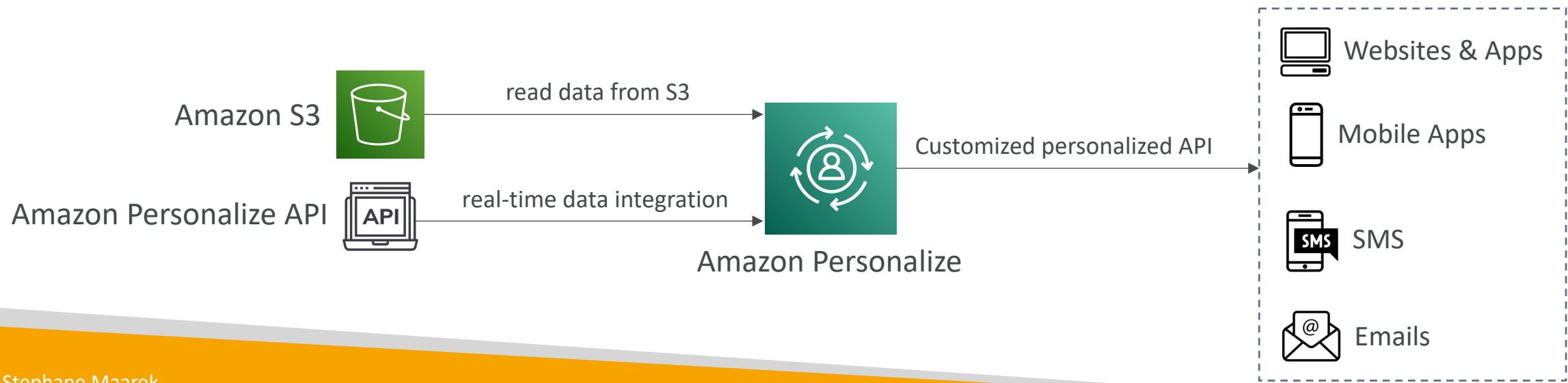
- Fully managed **document search service** powered by Machine Learning
- Extract answers from within a document (text, pdf, HTML, PowerPoint, MS Word, FAQs...)
- Natural language search capabilities
- Learn from user interactions/feedback to promote preferred results (**Incremental Learning**)
- Ability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

AWS Monitoring, Audit and Performance

CloudWatch, CloudTrail & AWS Config

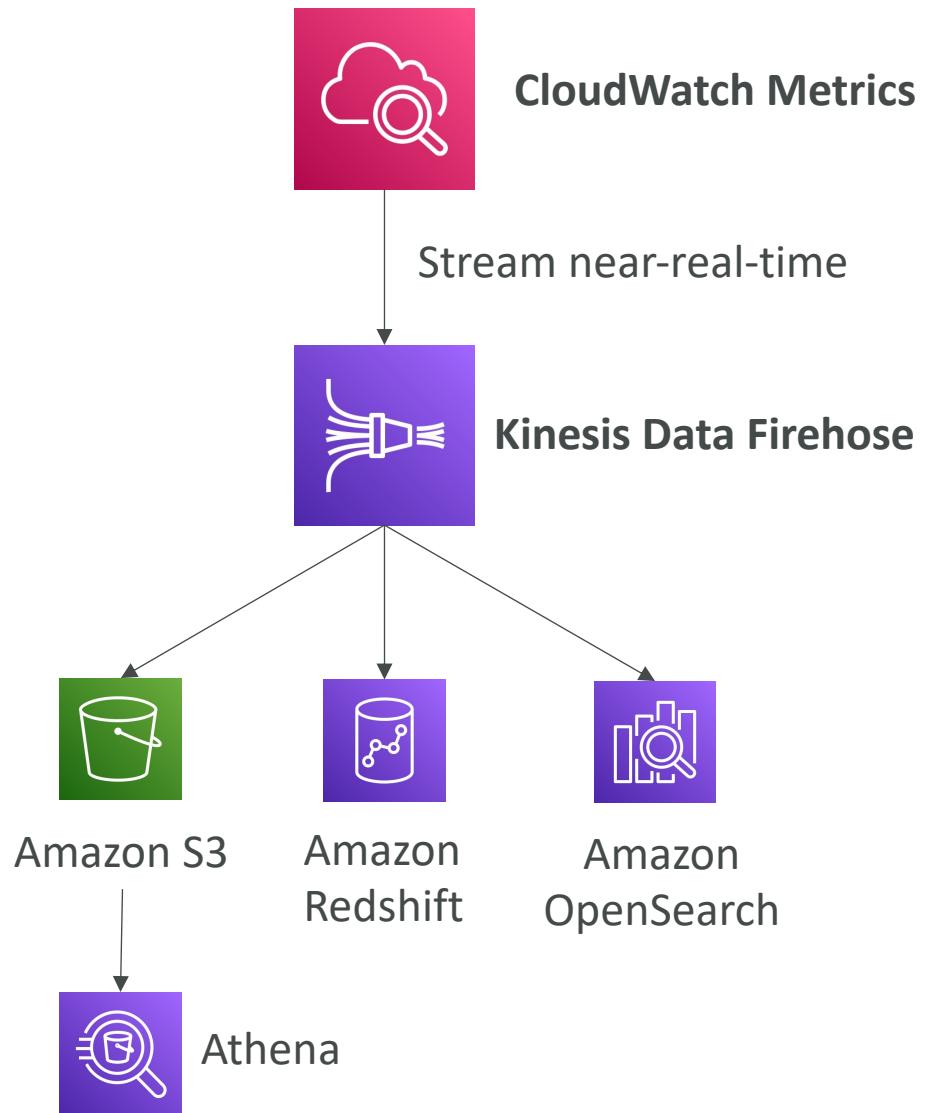
Amazon CloudWatch Metrics



- CloudWatch provides metrics for every services in AWS
- **Metric** is a variable to monitor (CPUUtilization, NetworkIn...)
- Metrics belong to **namespaces**
- Dimension is an attribute of a metric (instance id, environment, etc....).
- Up to 30 dimensions per metric
- Metrics have **timestamps**
- Can create CloudWatch dashboards of metrics
- Can create **CloudWatch Custom Metrics** (for the RAM for example)

CloudWatch Metric Streams

- 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





CloudWatch Logs

- **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

The screenshot shows the CloudWatch Logs Insights interface. At the top, there's a navigation bar with 'CloudWatch > Logs Insights'. Below it is a search bar labeled 'Select log group(s)' with 'application.log' selected. To the right of the search bar are time range controls set to '2021-11-09 (06:40:02) > 2021-11-09 (06:55:17)'. A large orange box highlights the query input area, which contains the following AWS Lambda-like query:

```
1 fields @timestamp, @message
2 | sort @timestamp desc
3 | limit 20
```

Below the query are three buttons: 'Run query' (orange), 'Save', and 'History'. A note says 'Queries are allowed to run for up to 15 minutes.' To the right of the query area, two orange boxes provide instructions: 'Change the time range here.' pointing to the time range controls, and 'Discovered Fields in your log groups.' pointing to a sidebar titled 'Fields'.

The main content area shows a histogram of log records over time, with the x-axis from 06:40 to 06:55 and the y-axis from 0 to 400. The histogram shows several peaks, notably around 06:45 and 06:47. Below the histogram, a table lists the first two matching records:

#	@timestamp	@message
► 1	2021-11-09T06:54:17.62...	{"Severity": "INFO", "message": "This is where the message detail would go", "IP Address": "10.30.86.98", "Timestamp": "2021-11-09T11:54:17.622Z"}
► 2	2021-11-09T06:54:13.38...	{"Severity": "INFO", "message": "This is where the message detail would go", "IP Address": "192.168.0.43", "Timestamp": "2021-11-09T11:54:13.382Z"}

Below the table, an orange box points to the 'Logs' tab, which is currently selected. Other tabs include 'Visualization'. To the right of the table, two more orange boxes provide options: 'Export the results, or add to a dashboard.' pointing to 'Export results' and 'Add to dashboard' buttons, and 'Hide histogram'.

On the far right, a sidebar has three items: 'Fields' (selected), 'Queries', and 'Help'.

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

CloudWatch Logs Insights

- Search and analyze log data stored in CloudWatch Logs
- Example: find a specific IP inside a log, count occurrences of “ERROR” in your logs...
- Provides a purpose-built query language
 - Automatically discovers fields from AWS services and JSON log events
 - Fetch desired event fields, filter based on conditions, calculate aggregate statistics, sort events, limit number of events...
 - Can save queries and add them to CloudWatch Dashboards
- Can query multiple Log Groups in different AWS accounts
- It's a query engine, not a real-time engine

Sample queries [Learn more ↗](#)

- ▶ Lambda
- ▶ VPC Flow Logs
- ▶ CloudTrail
- ▼ Common queries

▼ 25 most recently added log events

```
fields @timestamp, @message
| sort @timestamp desc
| limit 25
```

[Apply](#)

▼ Number of exceptions logged every 5 minutes

```
filter @message like /Exception/
| stats count(*) as exceptionCount by
bin(5m)
| sort exceptionCount desc
```

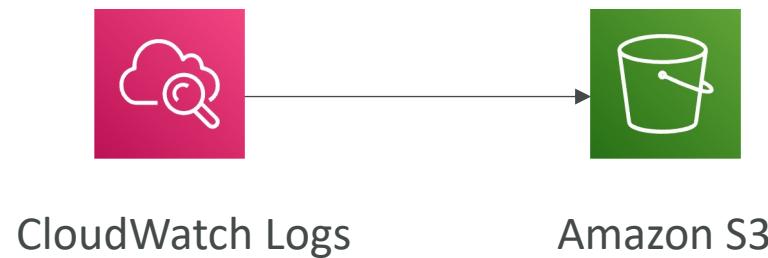
[Apply](#)

▼ List of log events that are not exceptions

```
fields @message
| filter @message not like /Exception/
```

[Apply](#)

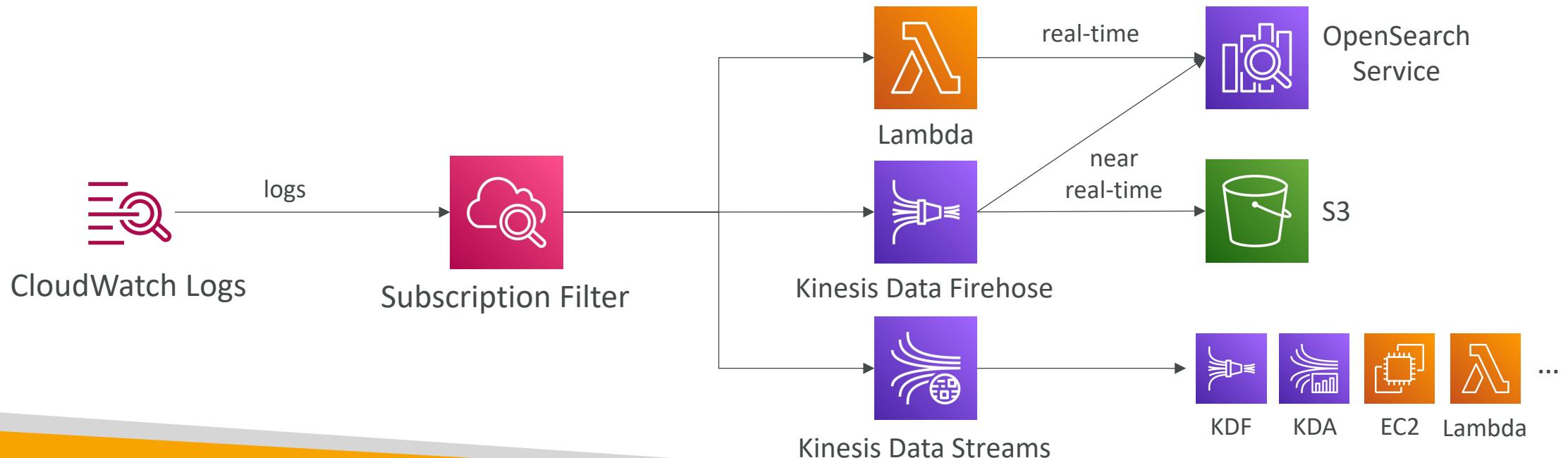
CloudWatch Logs – S3 Export



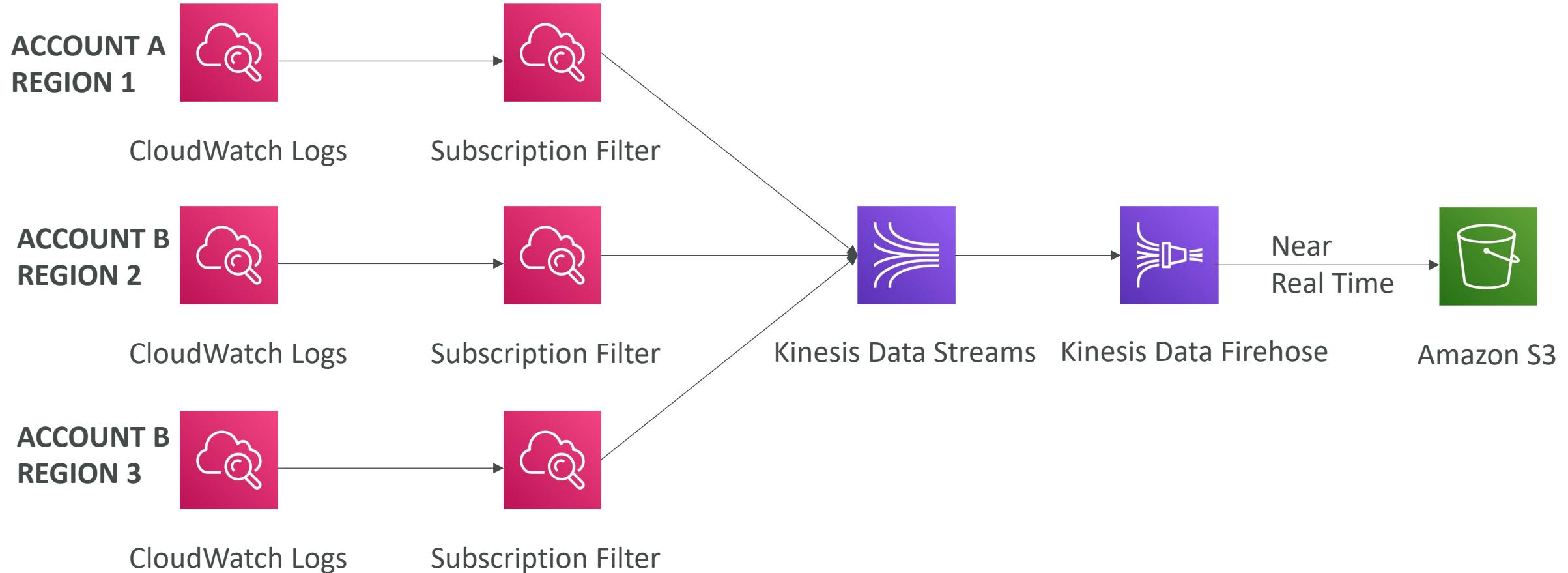
- Log data can take up to 12 hours to become available for export
- The API call is `CreateExportTask`
- Not near-real time or real-time... use Logs Subscriptions instead

CloudWatch Logs Subscriptions

- Get a 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

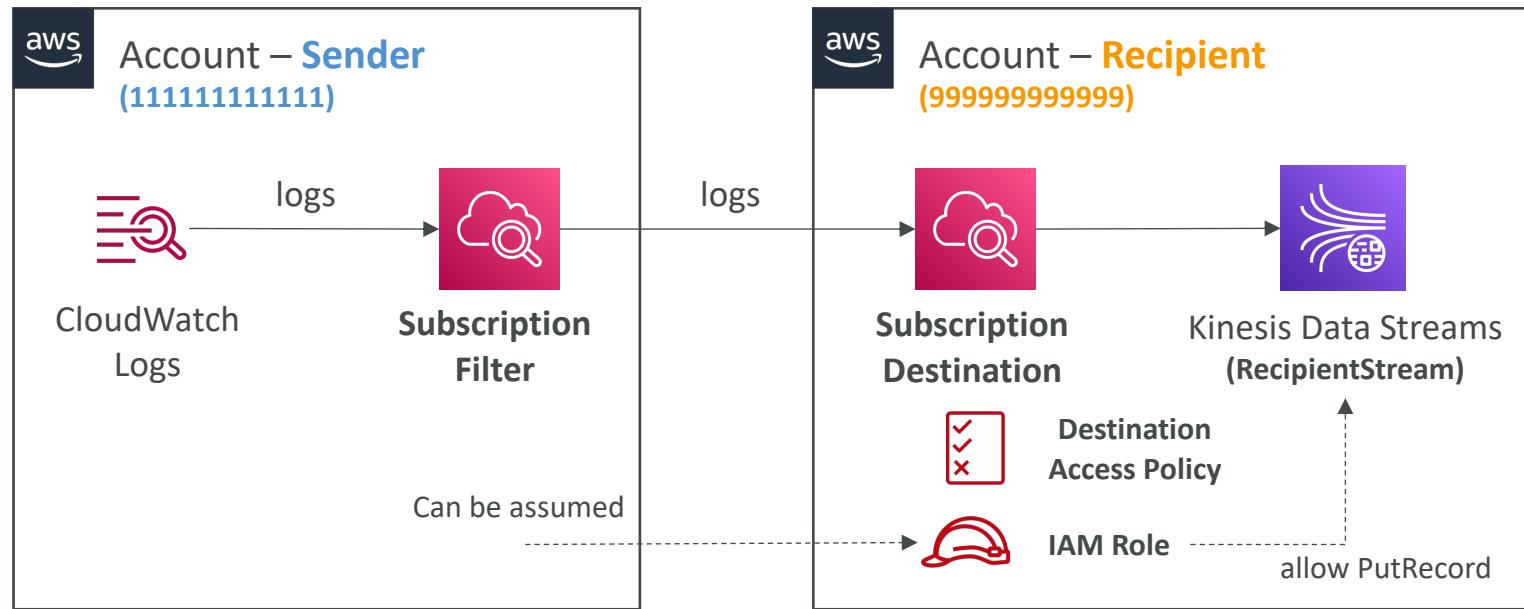


CloudWatch Logs Aggregation Multi-Account & Multi Region



CloudWatch Logs Subscriptions

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



```

IAM Role
(Cross-Account)

{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream"
    }
  ]
}

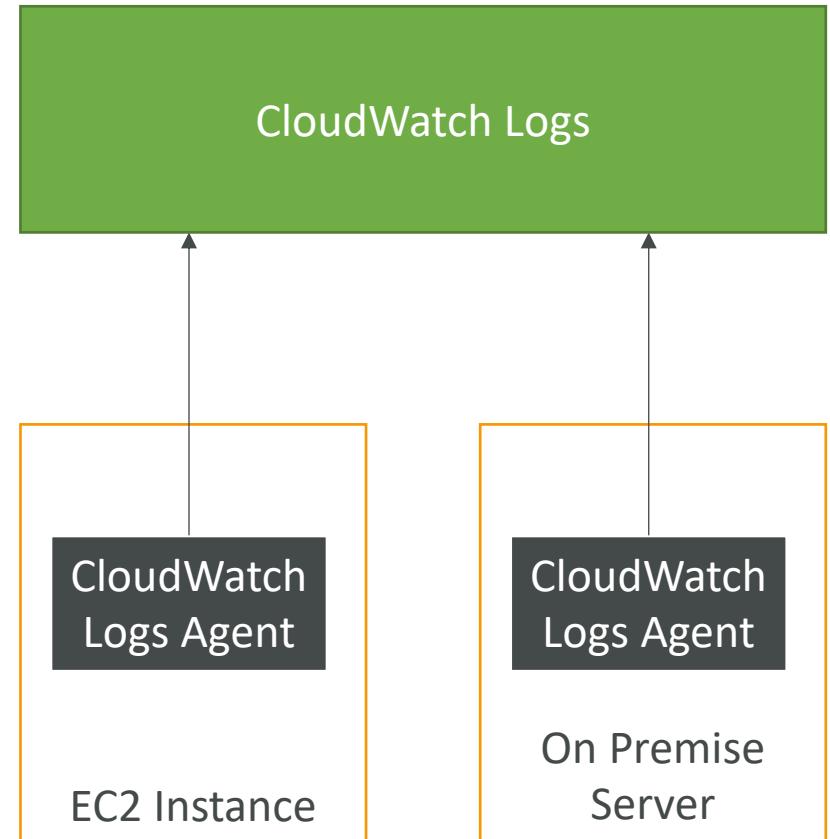
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "111111111111"
      },
      "Action": "logs:PutSubscriptionFilter",
      "Resource": "arn:aws:logs:us-east-1:999999999999:destination:testDestination"
    }
  ]
}

```

Destination Access Policy

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**
 - 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)

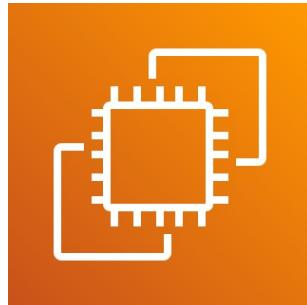
CloudWatch Alarms



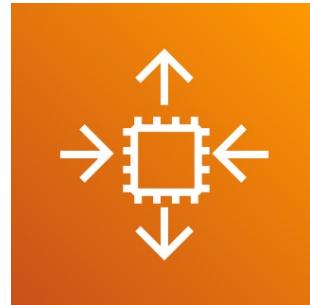
- Alarms are used to trigger notifications for any metric
- Various options (sampling, %, max, min, etc...)
- Alarm States:
 - OK
 - INSUFFICIENT_DATA
 - ALARM
- Period:
 - Length of time in seconds to evaluate the metric
 - High resolution custom metrics: 10 sec, 30 sec or multiples of 60 sec

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)



Amazon EC2



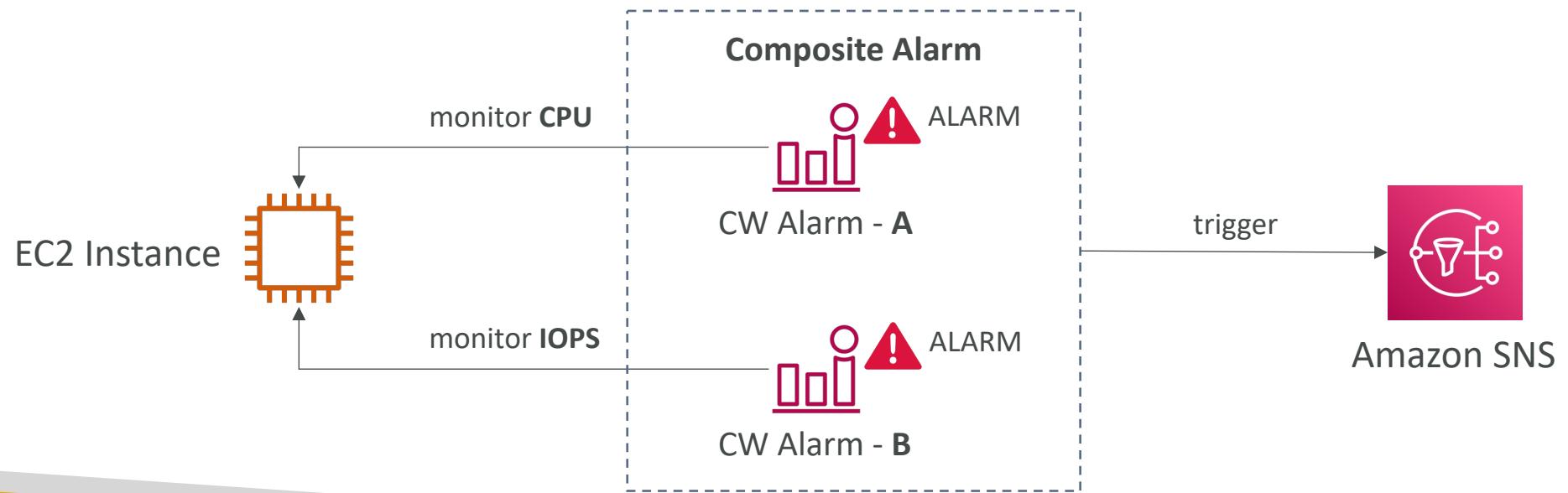
EC2 Auto Scaling



Amazon SNS

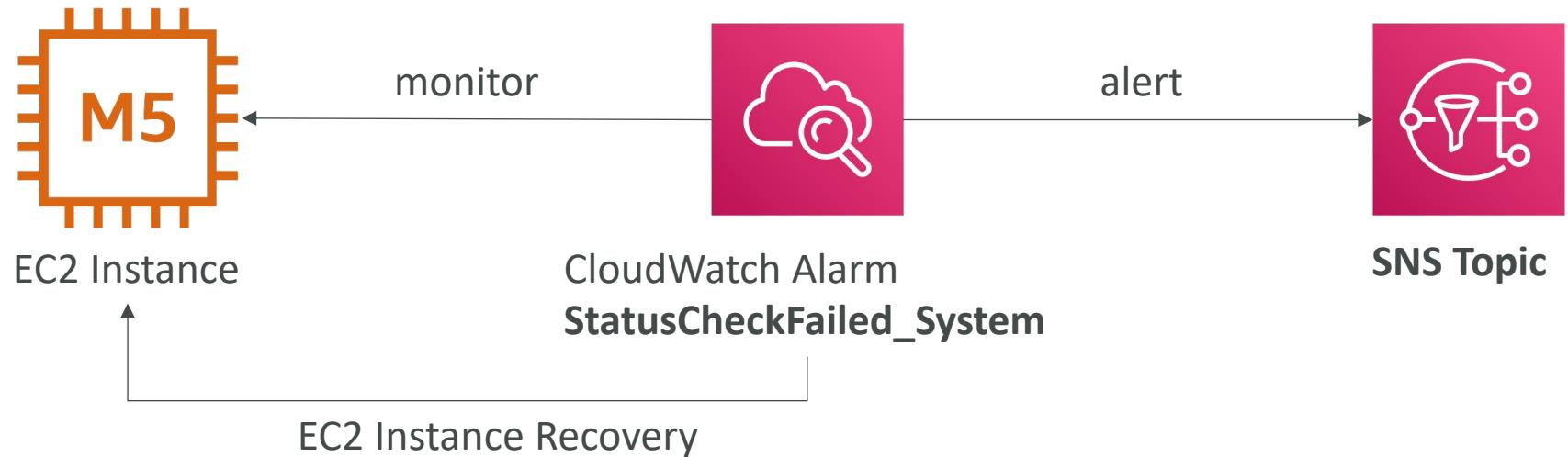
CloudWatch Alarms – Composite Alarms

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



EC2 Instance Recovery

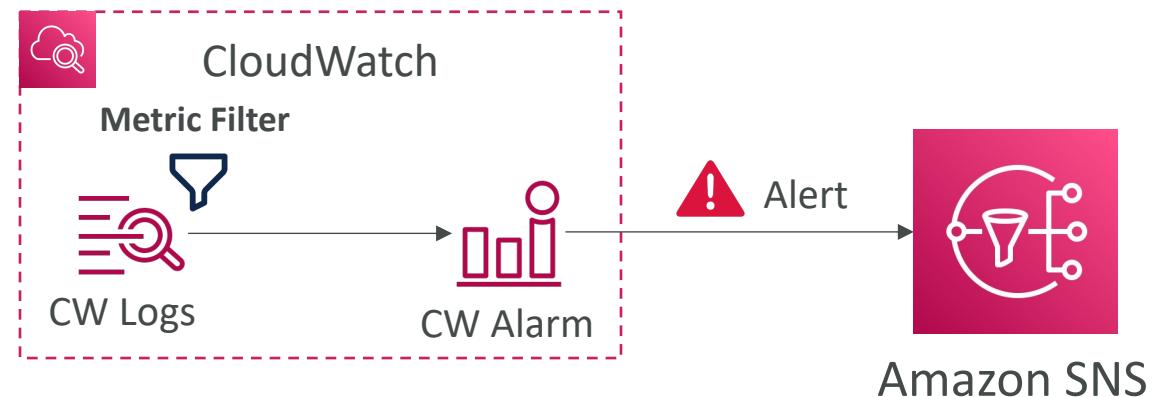
- Status Check:
 - Instance status = check the EC2 VM
 - System status = check the underlying hardware



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

CloudWatch Alarm: good to know

- Alarms can be created based on CloudWatch Logs Metrics Filters

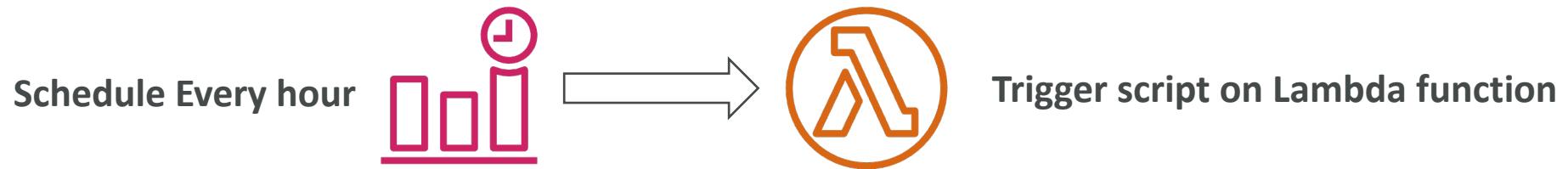


- To test alarms and notifications, set the alarm state to Alarm using CLI
`aws cloudwatch set-alarm-state --alarm-name "myalarm" --state-value ALARM --state-reason "testing purposes"`

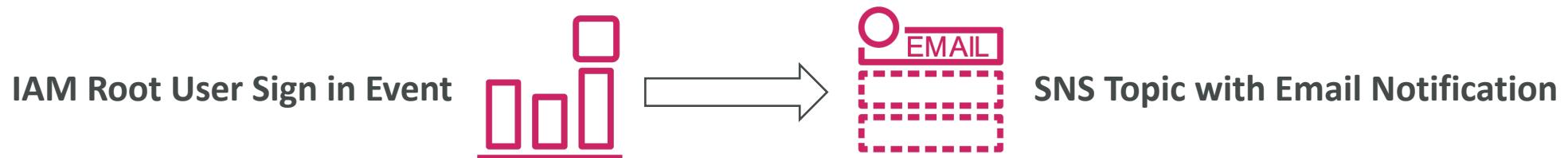
Amazon EventBridge (formerly CloudWatch Events)



- Schedule: Cron jobs (scheduled scripts)

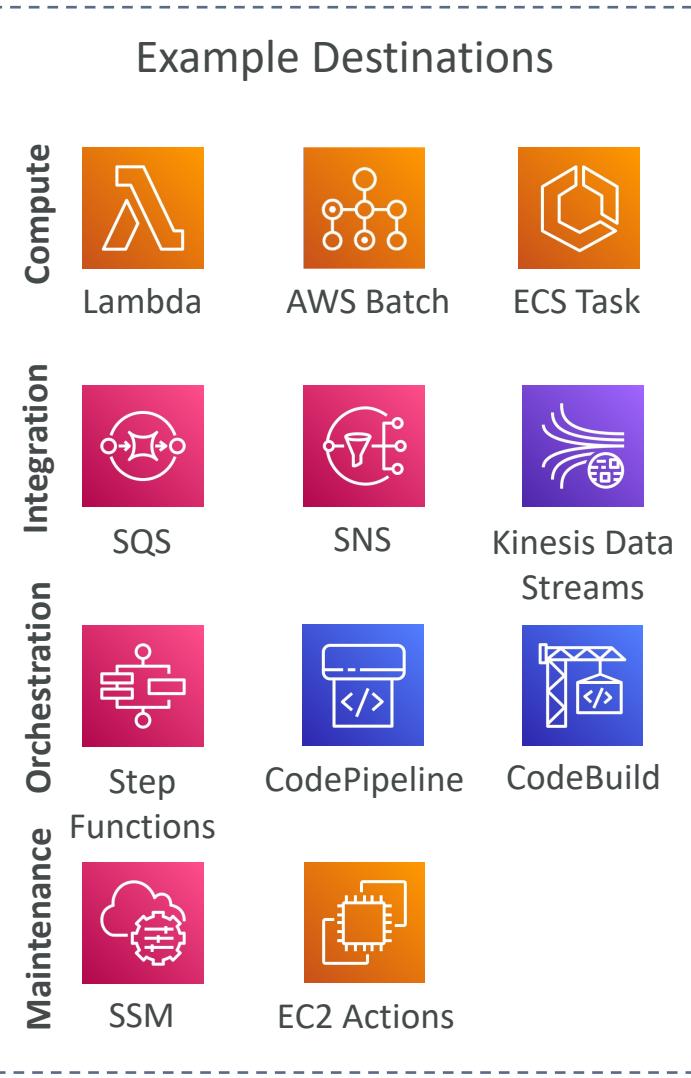
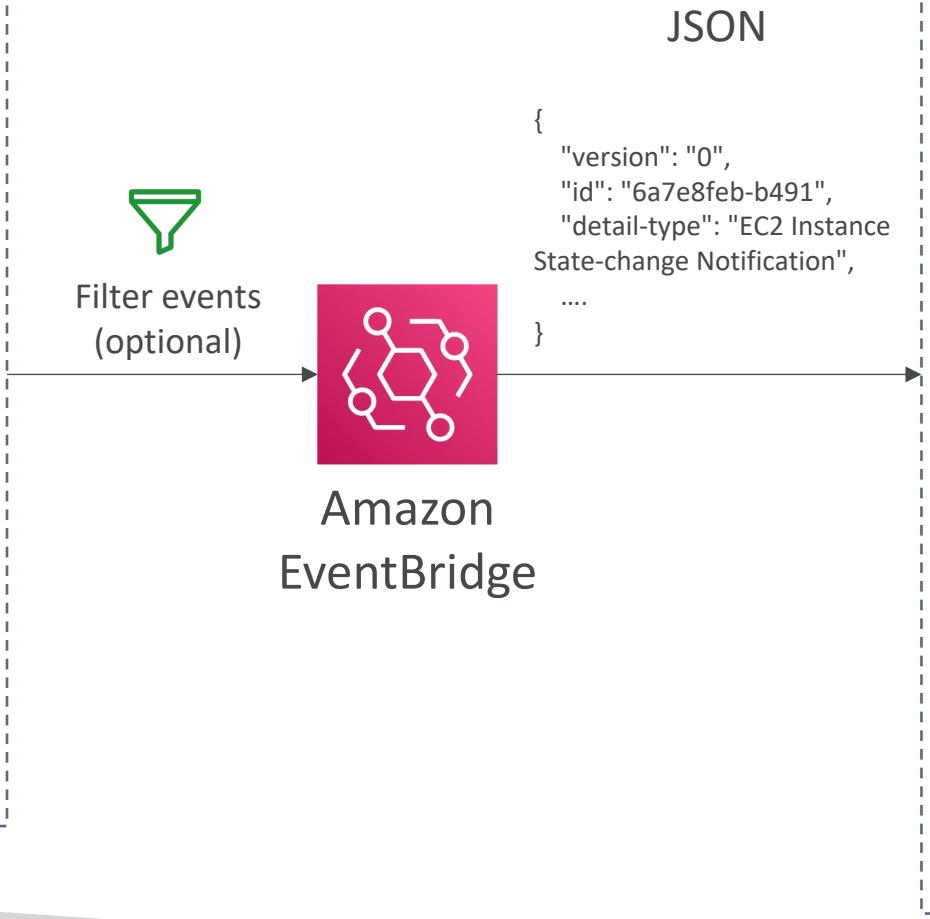
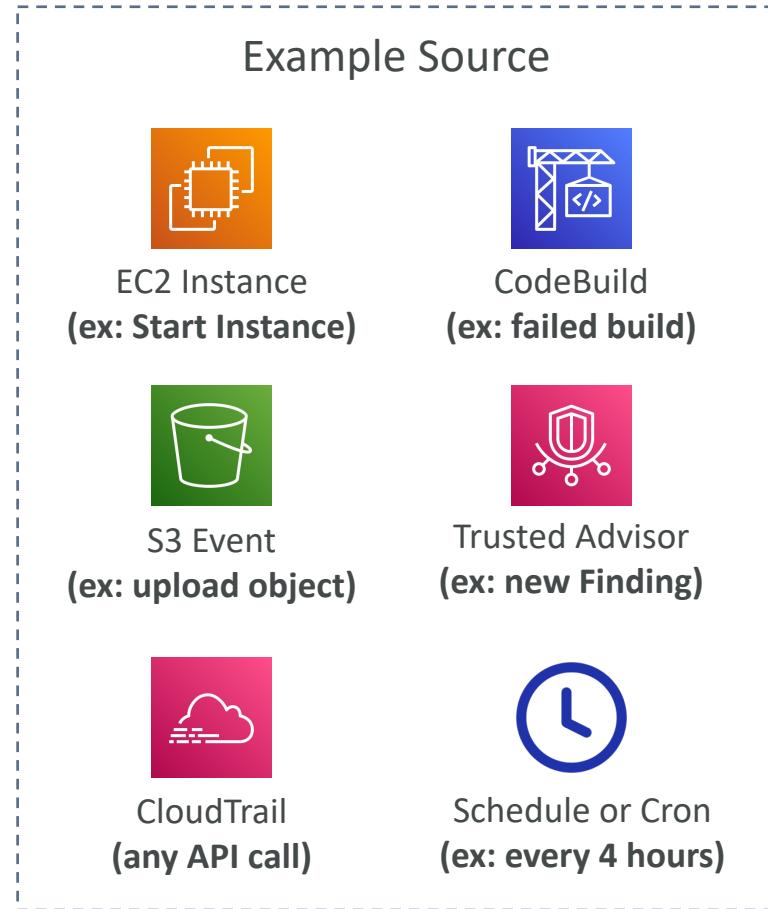


- Event Pattern: Event rules to react to a service doing something

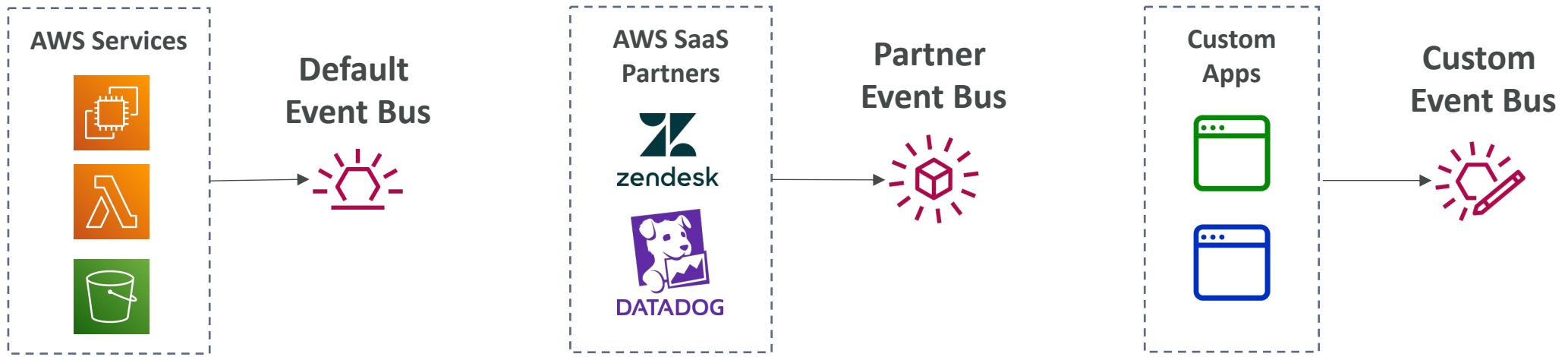


- Trigger Lambda functions, send SQS/SNS messages...

Amazon EventBridge Rules



Amazon EventBridge



- 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 EventBridge – Schema Registry

- EventBridge can analyze the events in your bus and infer the **schema**
- The **Schema Registry** allows you to generate code for your application, that will know in advance how data is structured in the event bus
- Schema can be versioned

The screenshot shows the AWS Schema Registry interface. At the top, it displays the schema name: `aws.codepipeline@CodePipelineActionExecutionStateChange`. Below this, the **Schema details** section provides the following information:

Schema name	Last modified	Schema ARN
<code>aws.codepipeline@CodePipelineActionExecutionStateChange</code>	Dec 1, 2019, 12:11 AM GMT	-
Description	Schema for event type CodePipelineActionExecutionStateChange, published by AWS service aws.codepipeline	Schema registry aws.events Number of versions 1 Schema type OpenAPI 3.0

Below the details, the **Version 1** section is shown, created on Dec 1, 2019, 12:11 AM GMT. It includes an **Action** dropdown and a **Download code bindings** button. The schema definition is displayed as follows:

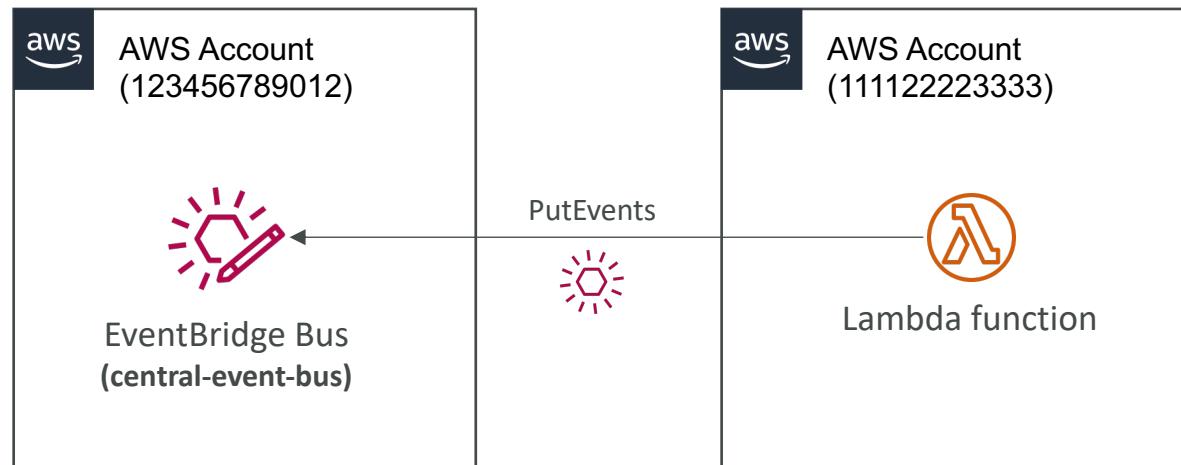
```
1 {
2   "openapi": "3.0.0",
3   "info": {
4     "version": "1.0.0",
5     "title": "CodePipelineActionExecutionStateChange"
6   },
7   "paths": {},
8   "components": {
9     "schemas": {
10       "AWSEvent": {
```

Amazon EventBridge – 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

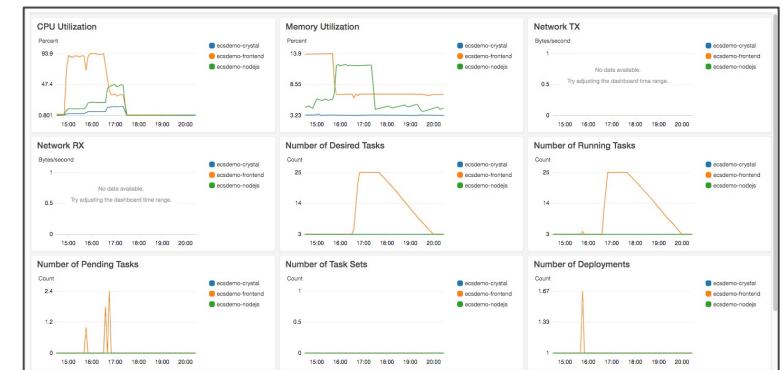
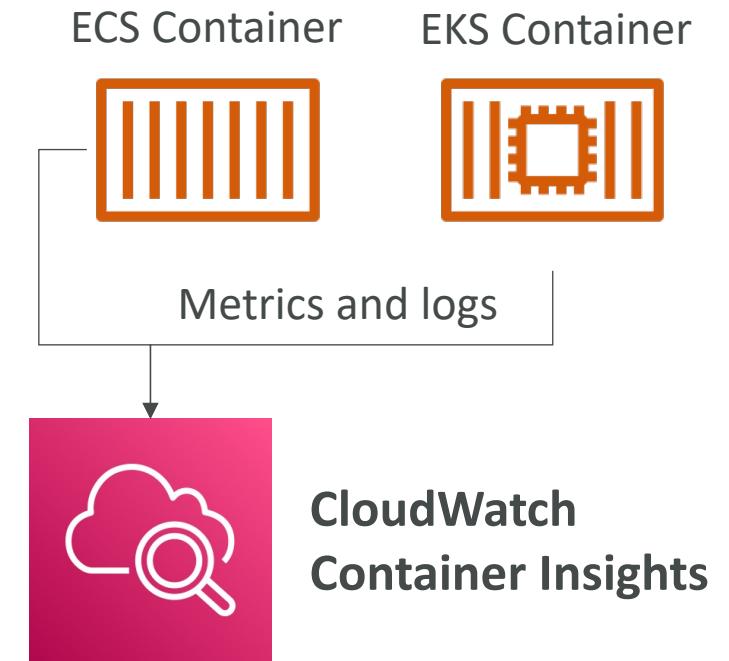
```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "events:PutEvents",  
            "Principal": { "AWS": "111122223333" },  
            "Resource": "arn:aws:events:us-east-1:123456789012:  
event-bus/central-event-bus"  
        }  
    ]  
}
```

Allow **events** from another AWS account



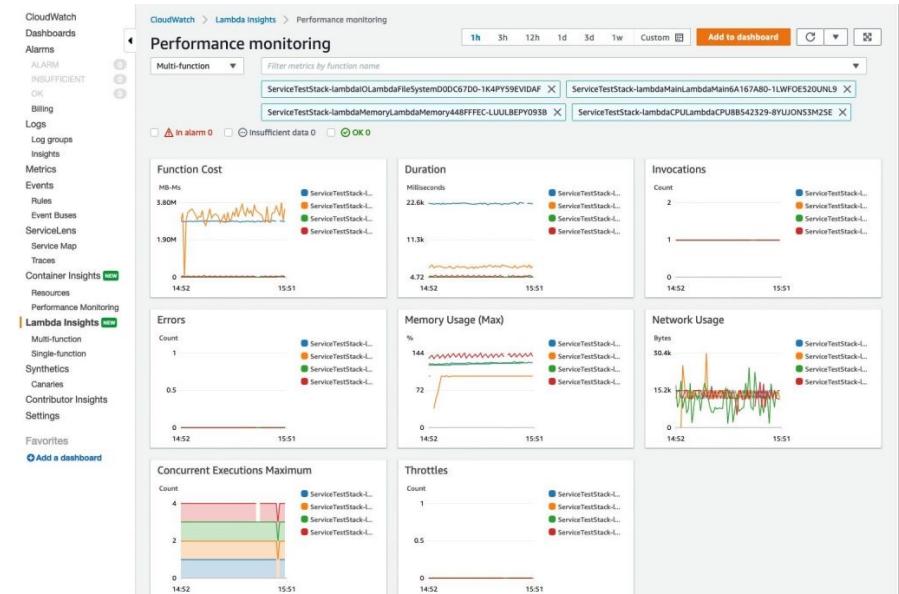
CloudWatch Container Insights

- 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.



VPC Flow Logs



CloudWatch Logs



CloudWatch
Contributor Insights

Top-10 IP addresses

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

CloudWatch Insights and Operational Visibility

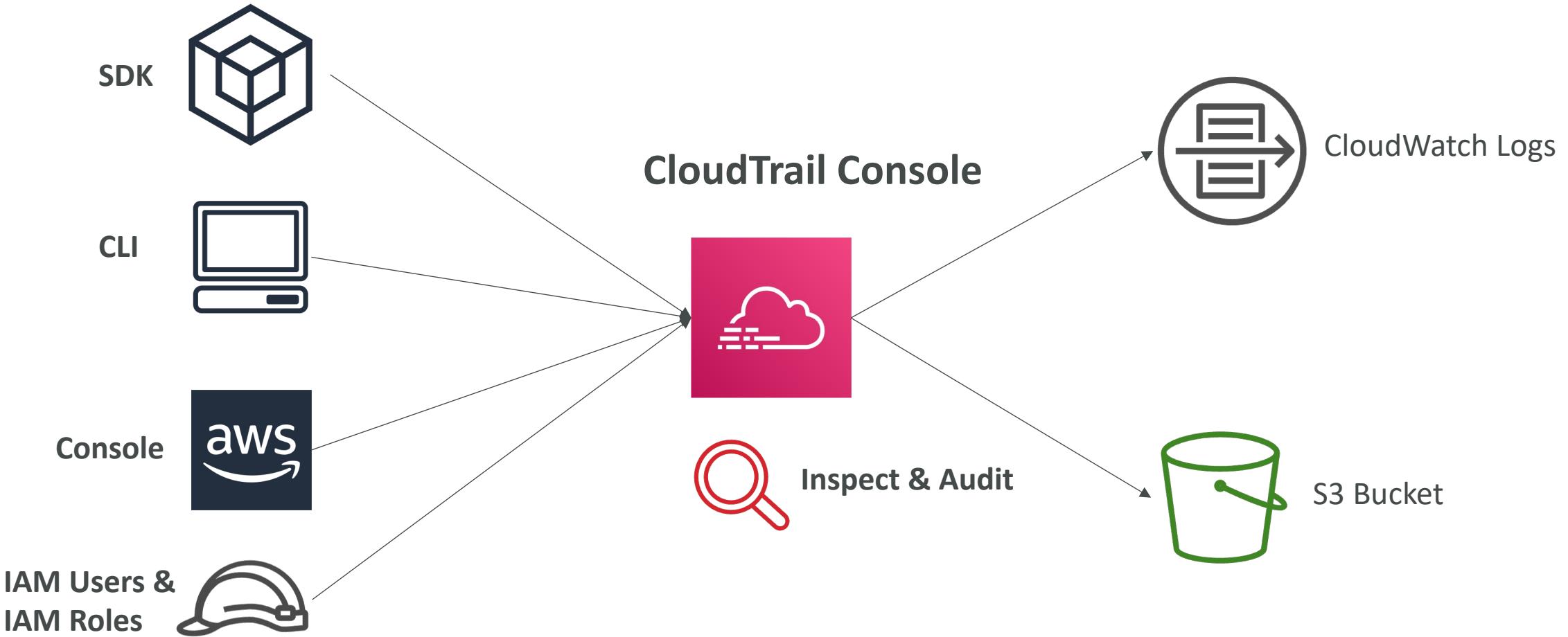
- CloudWatch Container Insights
 - ECS, EKS, Kubernetes on EC2, Fargate, needs agent for Kubernetes
 - Metrics and logs
- CloudWatch Lambda Insights
 - Detailed metrics to troubleshoot serverless applications
- CloudWatch Contributors Insights
 - Find “Top-N” Contributors through CloudWatch Logs
- CloudWatch 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!

CloudTrail Diagram





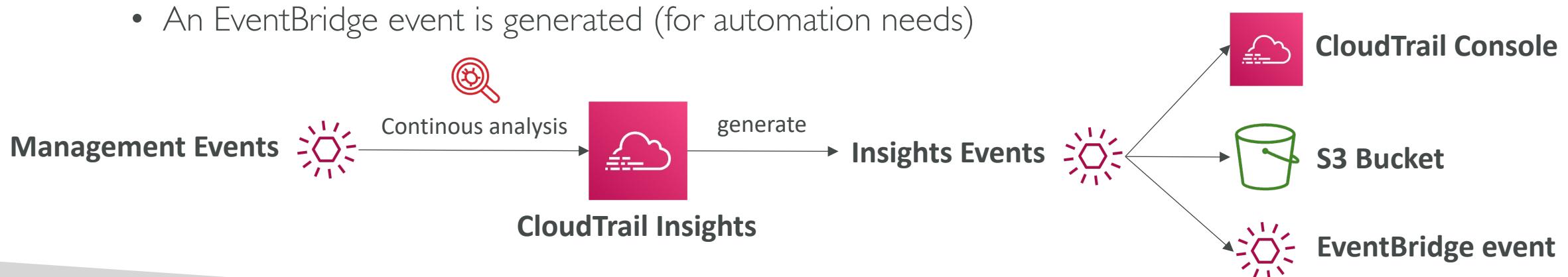
CloudTrail Events

- 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 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 Events:
 - See next slide ☺



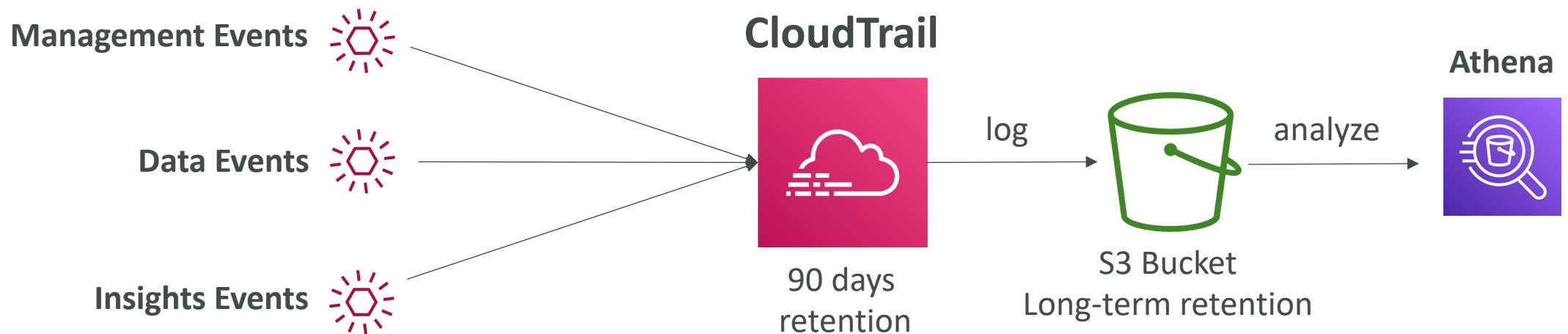
CloudTrail Insights

- Enable CloudTrail Insights to detect unusual activity in your account:
 - 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
 - Anomalies appear in the CloudTrail console
 - Event is sent to Amazon S3
 - An EventBridge event is generated (for automation needs)

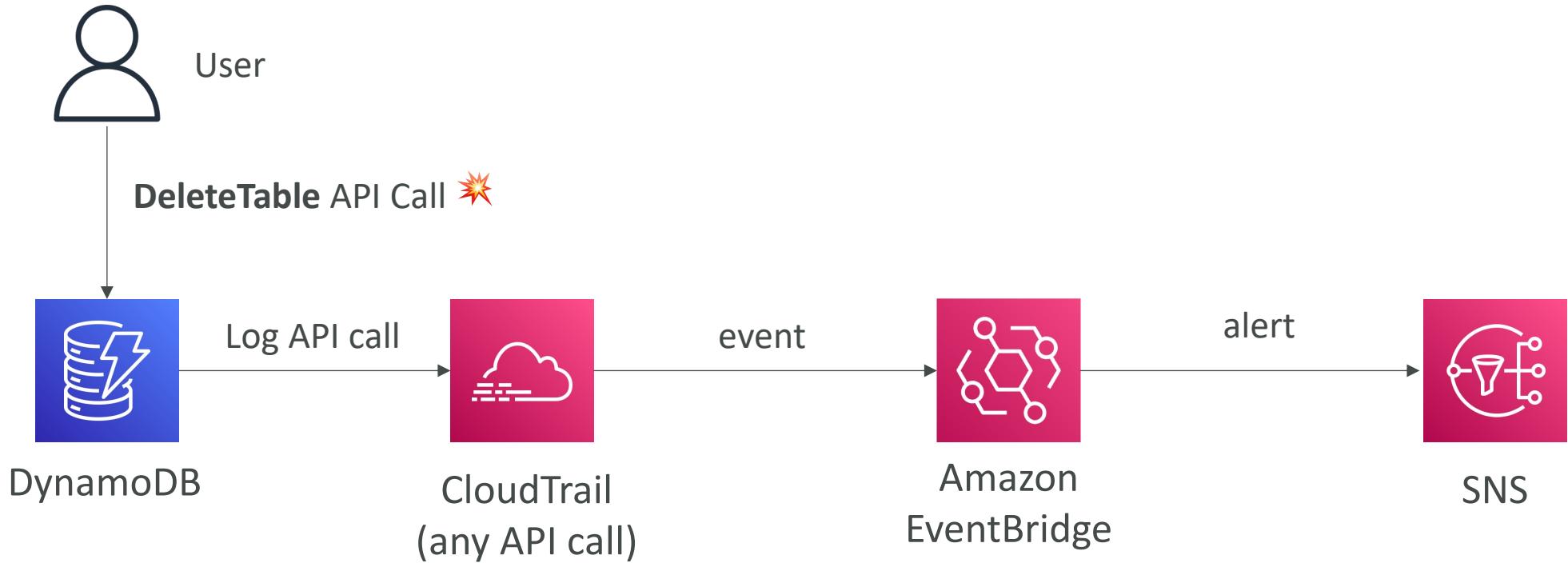


CloudTrail Events Retention

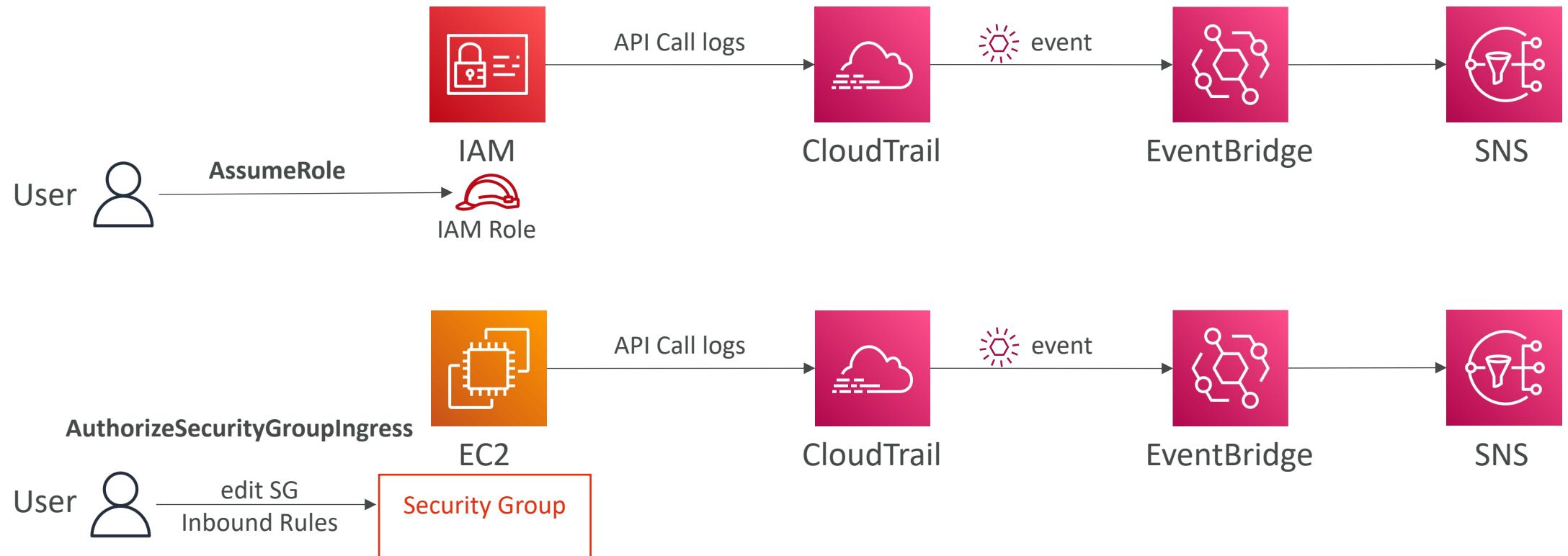
- Events are stored for 90 days in CloudTrail
- To keep events beyond this period, log them to S3 and use Athena

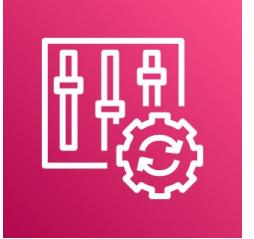


Amazon EventBridge – Intercept API Calls



Amazon EventBridge + CloudTrail





AWS Config

- Helps with auditing and recording **compliance** of your AWS resources
- Helps record configurations and changes over time
- 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

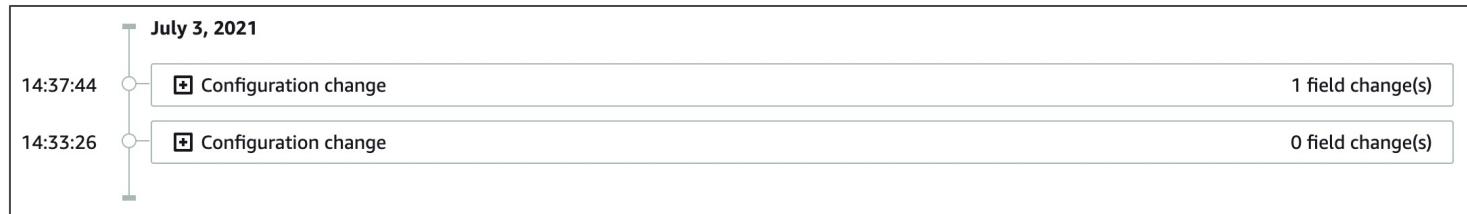
- Can use AWS managed config rules (over 75)
- Can make custom config rules (must be defined in AWS Lambda)
 - Ex: evaluate if each EBS disk is of type gp2
 - Ex: evaluate if each EC2 instance is t2.micro
- Rules can be evaluated / triggered:
 - For each config change
 - And / or: at regular time intervals
- 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

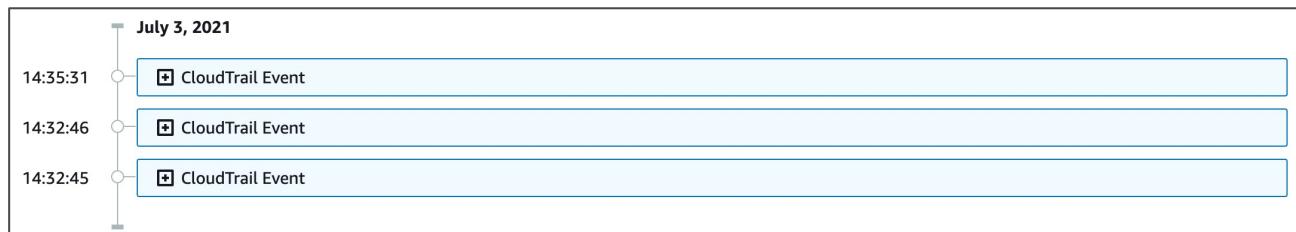
- View compliance of a resource over time

<input type="radio"/> sg-077b425b1649da83e	EC2 SecurityGroup	 Compliant
<input type="radio"/> sg-0831434f1876c0c74	EC2 SecurityGroup	 Noncompliant
<input type="radio"/> sg-09f10ed254d464f30	EC2 SecurityGroup	 Compliant

- View configuration of a resource over time

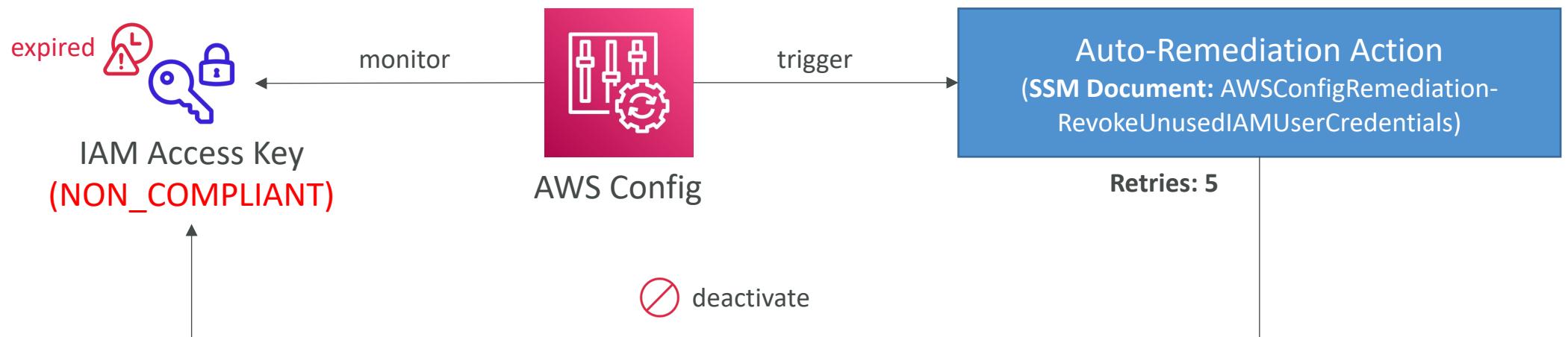


- View CloudTrail API calls of a resource over time



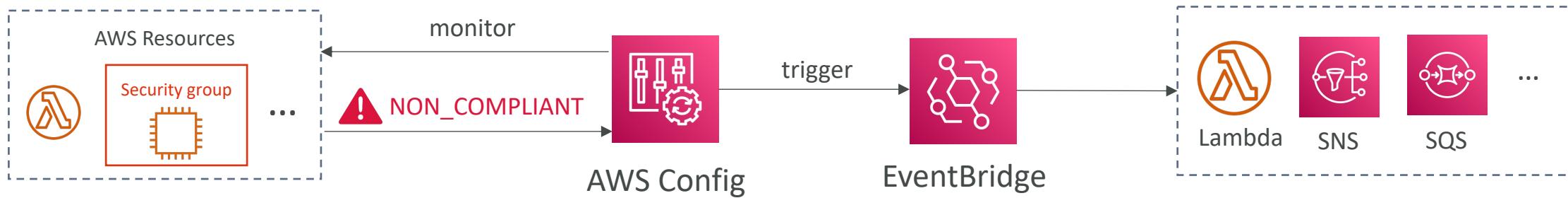
Config Rules – Remediations

- 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

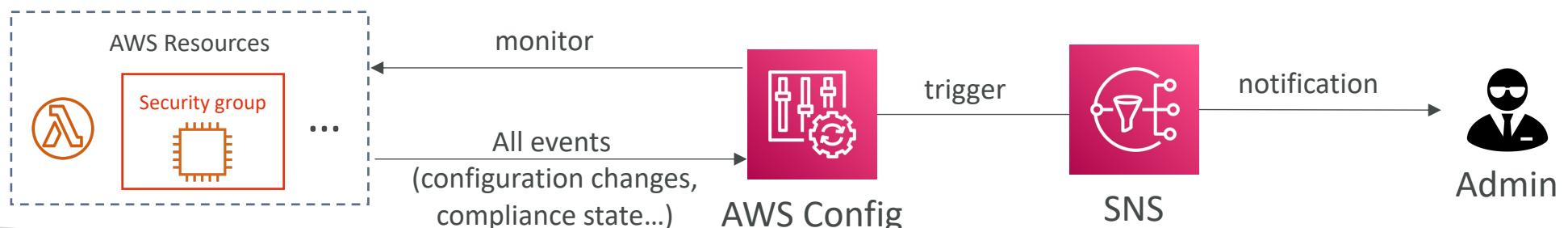


Config Rules – Notifications

- Use EventBridge to trigger notifications when AWS resources are non-compliant



- 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

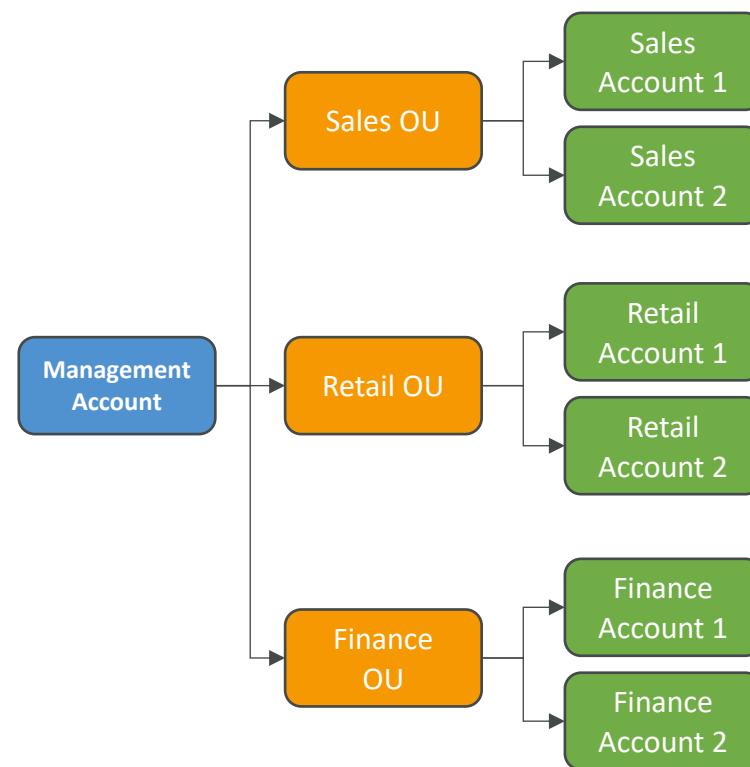
For an Elastic Load Balancer

- 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

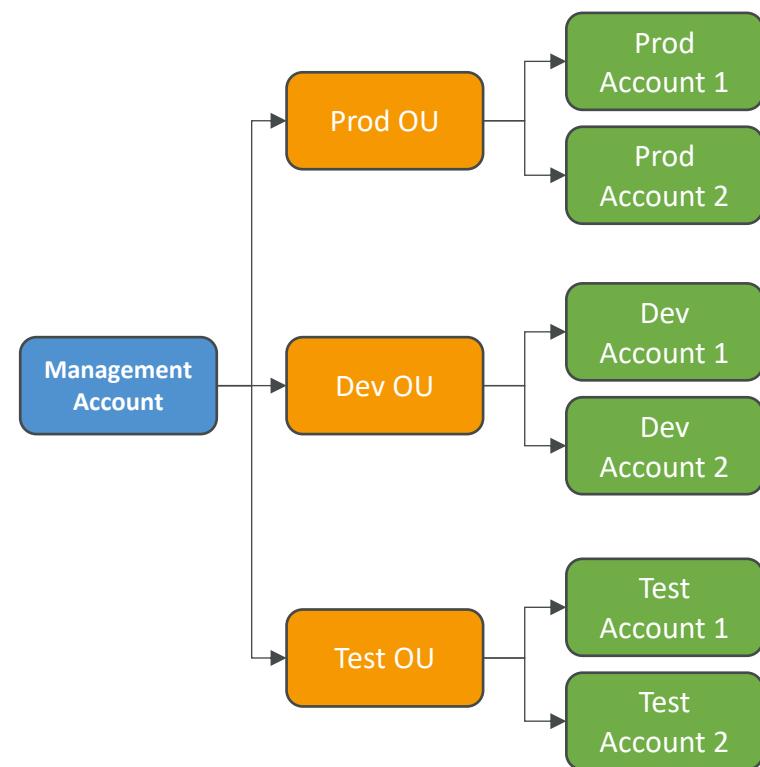
Advanced Identity in AWS

Organizational Units (OU) - Examples

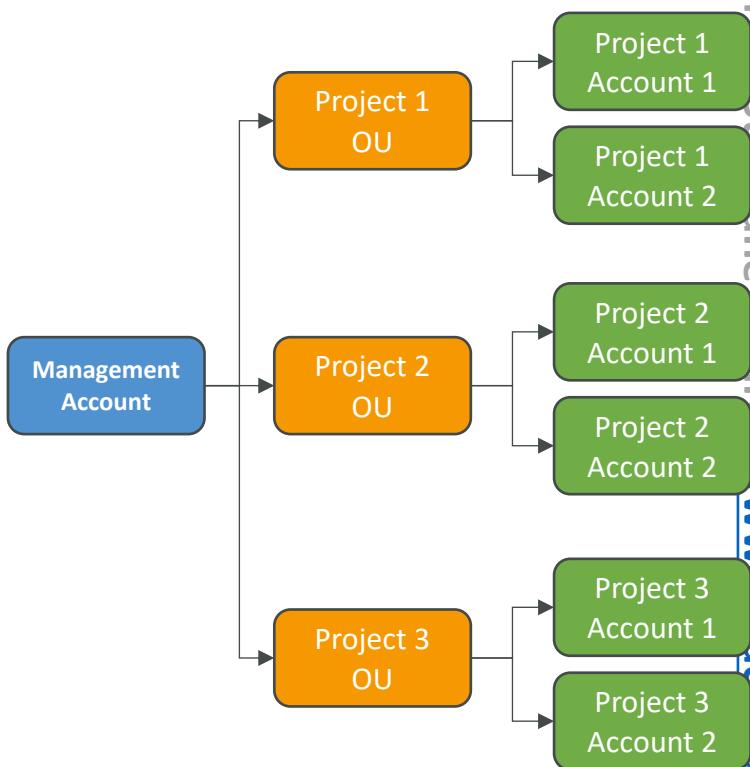
Business Unit



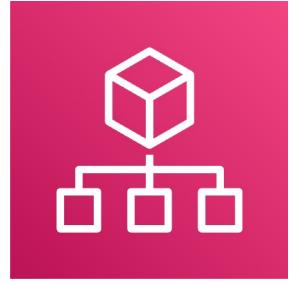
Environmental Lifecycle



Project-Based

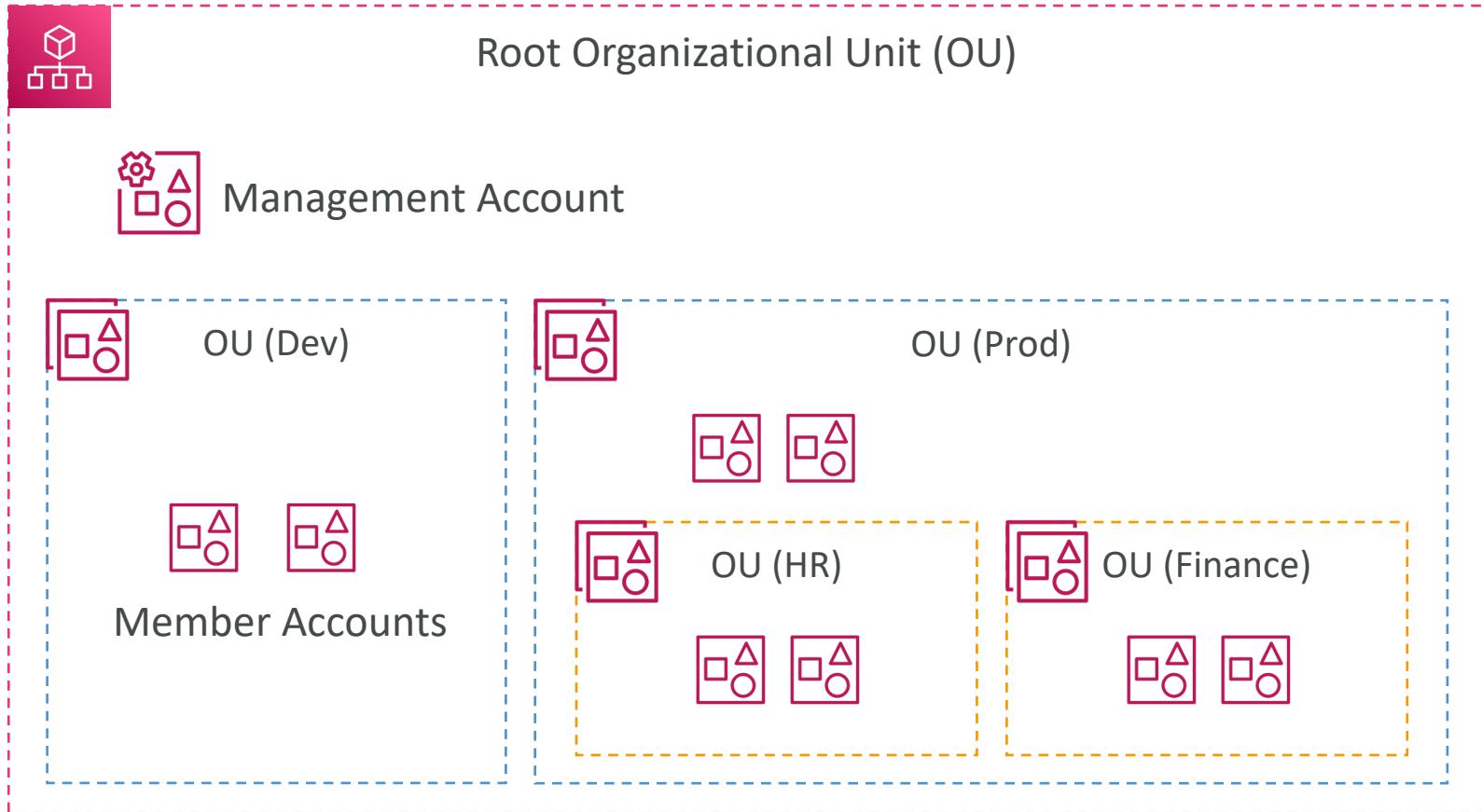


AWS Organizations



- Global service
- Allows to manage multiple AWS accounts
- The main account is the management account
- Other accounts are member accounts
- Member accounts can only be part of one organization
- Consolidated Billing across all accounts - single payment method
- Pricing benefits from aggregated usage (volume discount for EC2, S3...)
- Shared reserved instances and Savings Plans discounts across accounts
- API is available to automate AWS account creation

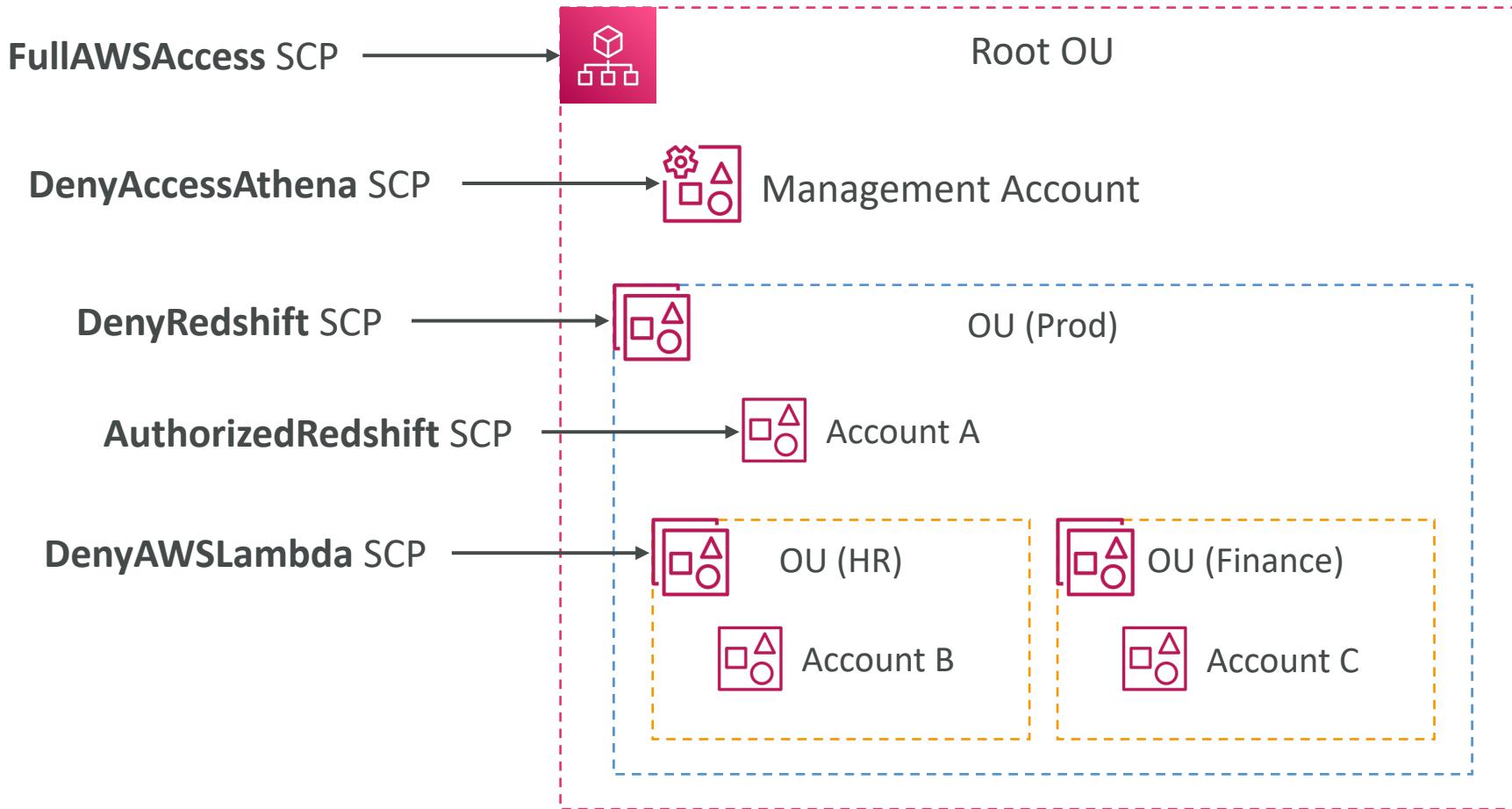
AWS Organizations



AWS Organizations

- Advantages
 - Multi Account vs One Account Multi VPC
 - Use tagging standards for billing purposes
 - Enable CloudTrail on all accounts, send logs to central S3 account
 - Send CloudWatch Logs to central logging account
 - Establish Cross Account Roles for Admin purposes
- Security: Service Control Policies (SCP)
 - IAM policies applied to OU or Accounts to restrict Users and Roles
 - They do not apply to the management account (full admin power)
 - Must have an explicit allow (does not allow anything by default – like IAM)

SCP Hierarchy



- Management Account
 - Can do anything
 - (no SCP apply)
- Account A
 - Can do anything
 - EXCEPT access Redshift (explicit Deny from OU)
- Account B
 - Can do anything
 - EXCEPT access Redshift (explicit Deny from Prod OU)
 - EXCEPT access Lambda (explicit Deny from HR OU)
- Account C
 - Can do anything
 - EXCEPT access Redshift (explicit Deny from Prod OU)

SCP Examples

Blocklist and Allowlist strategies

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowsAllActions",  
            "Effect": "Allow",  
            "Action": "*",  
            "Resource": "*"  
        },  
        {  
            "Sid": "DenyDynamoDB",  
            "Effect": "Deny",  
            "Action": "dynamodb:*",  
            "Resource": "*"  
        }  
    ]  
}
```

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:*",  
                "cloudwatch:/*"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

More examples: https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_example-scps.html

IAM Conditions

aws:SourceIp

restrict the client IP from
which the API calls are being made

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Deny",  
      "Action": "*",  
      "Resource": "*",  
      "Condition": {  
        "NotIpAddress": {  
          "aws:SourceIp": ["192.0.2.0/24", "203.0.113.0/24"]  
        }  
      }  
    }  
  ]  
}
```

aws:RequestedRegion

restrict the region the
API calls are made to

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Deny",  
      "Action": ["ec2:*", "rds:*", "dynamodb:*"],  
      "Resource": "*",  
      "Condition": {  
        "StringEquals": {  
          "aws:RequestedRegion": ["eu-central-1", "eu-west-1"]  
        }  
      }  
    }  
  ]  
}
```

IAM Conditions

ec2:ResourceTag

restrict based on **tags**

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": ["ec2:startInstances", "ec2:StopInstances"],  
      "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",  
      "Condition": {  
        "StringEquals": {  
          "ec2:ResourceTag/Project": "DataAnalytics",  
          "aws:PrincipalTag/Department": "Data"  
        }  
      }  
    }  
  ]  
}
```

aws:MultiFactorAuthPresent

to force MFA

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "ec2:*",  
      "Resource": "*"  
    },  
    {  
      "Effect": "Deny",  
      "Action": ["ec2:StopInstances", "ec2:TerminateInstances"],  
      "Resource": "*",  
      "Condition": {  
        "BoolIfExists": {  
          "aws:MultiFactorAuthPresent": false  
        }  
      }  
    }  
  ]  
}
```

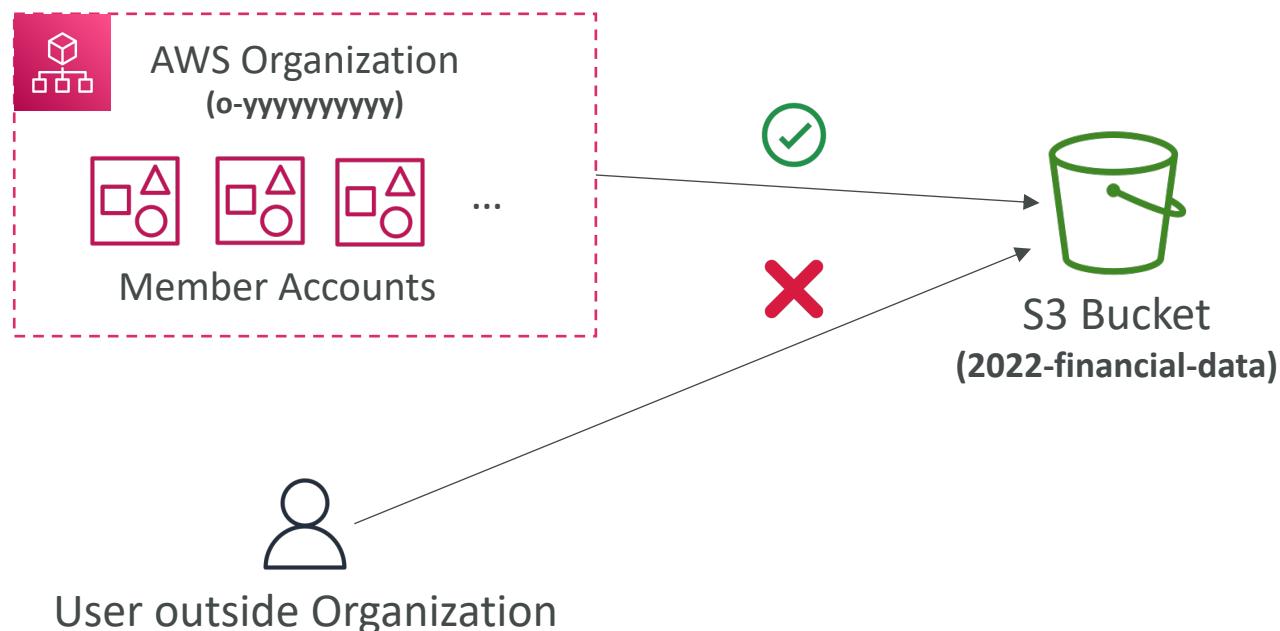
IAM for S3

- s3>ListBucket permission applies to
arn:aws:s3:::test
- => bucket level permission
- s3GetObject, s3PutObject,
s3DeleteObject applies to
arn:awn:s3:::test/*
- => object level permission

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": ["s3>ListBucket"],  
      "Resource": "arn:aws:s3:::test"  
    },  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3>PutObject",  
        "s3>GetObject",  
        "s3>DeleteObject"  
      ],  
      "Resource": "arn:aws:s3:::test/*"  
    }  
  ]  
}
```

Resource Policies & aws:PrincipalOrgID

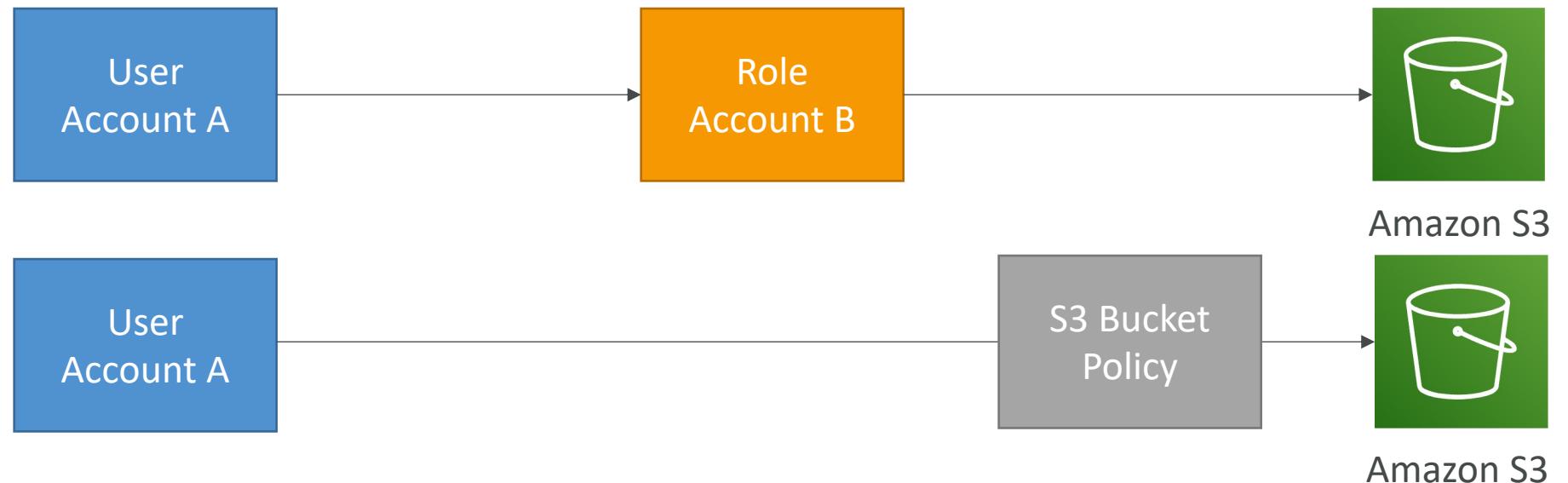
- aws:PrincipalOrgID can be used in any resource policies to restrict access to accounts that are member of an AWS Organization



```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": ["s3:PutObject", "s3:GetObject"],  
      "Resource": "arn:aws:s3:::2022-financial-data/*",  
      "Condition": {  
        "StringEquals": {  
          "aws:PrincipalOrgID": ["o-yyyyyyyyyy"]  
        }  
      }  
    }  
  ]  
}
```

IAM Roles vs Resource Based Policies

- Cross account:
 - attaching a resource-based policy to a resource (example: S3 bucket policy)
 - OR using a role as a proxy

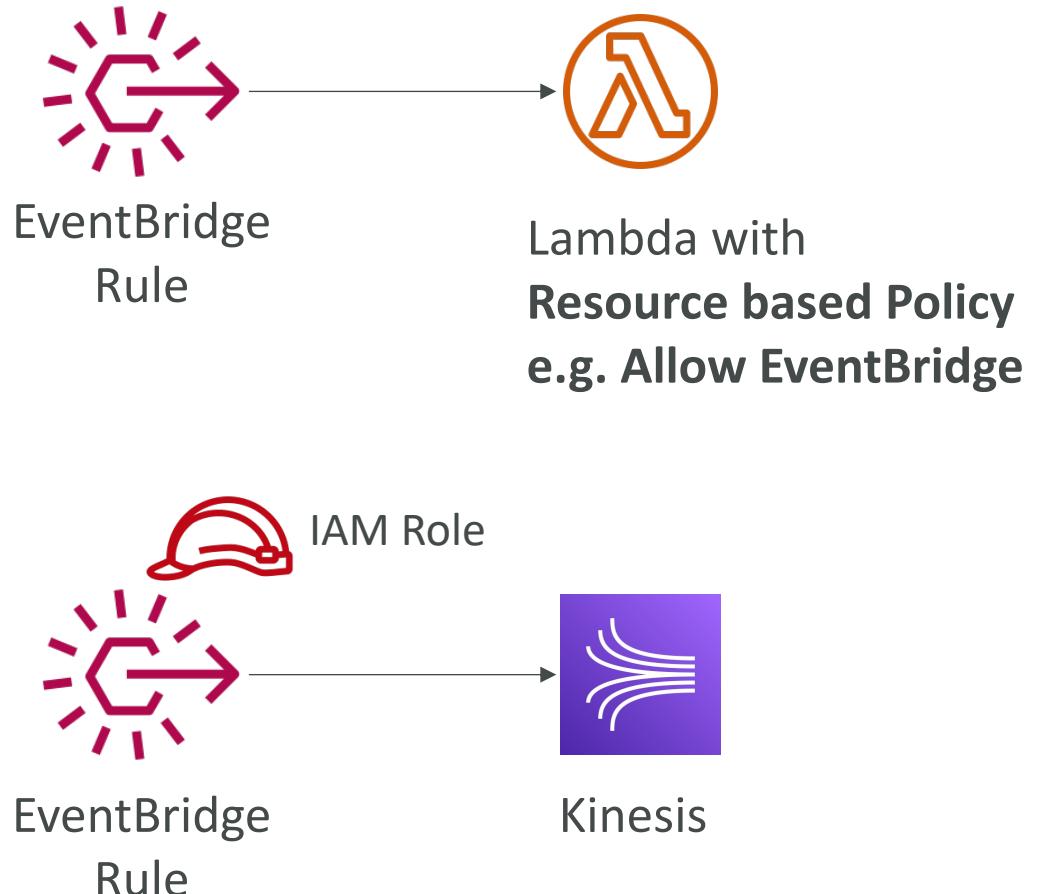


IAM Roles vs Resource-Based Policies

- When you assume a role (user, application or service), you give up your original permissions and take the permissions assigned to the role
- When using a resource-based policy, the principal doesn't have to give up his permissions
- Example: User in account A needs to scan a DynamoDB table in Account A and dump it in an S3 bucket in Account B.
- Supported by: Amazon S3 buckets, SNS topics, SQS queues, etc...

Amazon EventBridge – Security

- When a rule runs, it needs permissions on the target
- Resource-based policy: Lambda, SNS, SQS, CloudWatch Logs, API Gateway...
- IAM role: Kinesis stream, Systems Manager Run Command, ECS task...



IAM Permission Boundaries

- IAM Permission Boundaries are supported for users and roles (not groups)
- Advanced feature to use a managed policy to set the maximum permissions an IAM entity can get.

Example:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:*",  
                "cloudwatch:*",  
                "ec2:*"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```



IAM Permission Boundary

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iam>CreateUser",  
            "Resource": "*"  
        }  
    ]  
}
```

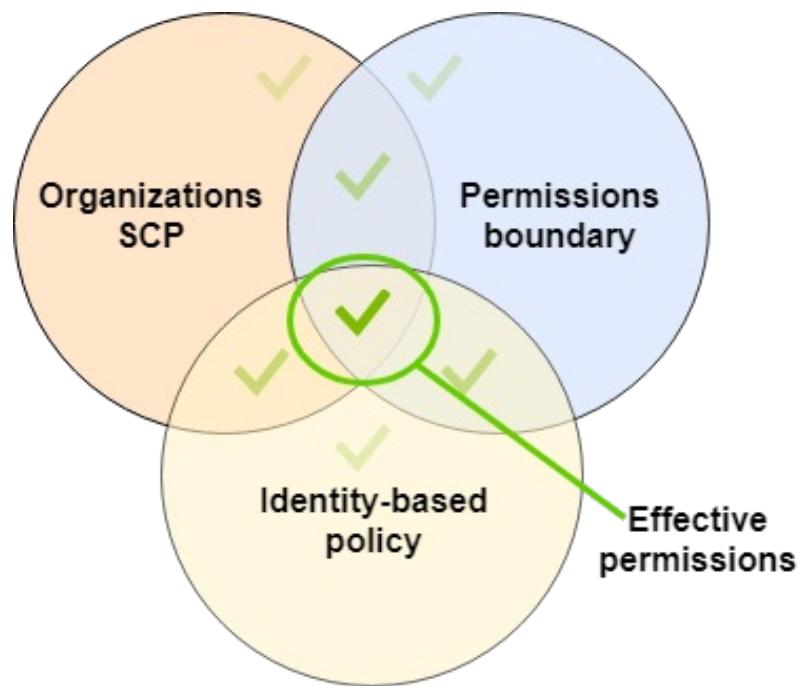


No Permissions

**IAM Permissions
Through IAM Policy**

IAM Permission Boundaries

- Can be used in combinations of AWS Organizations SCP

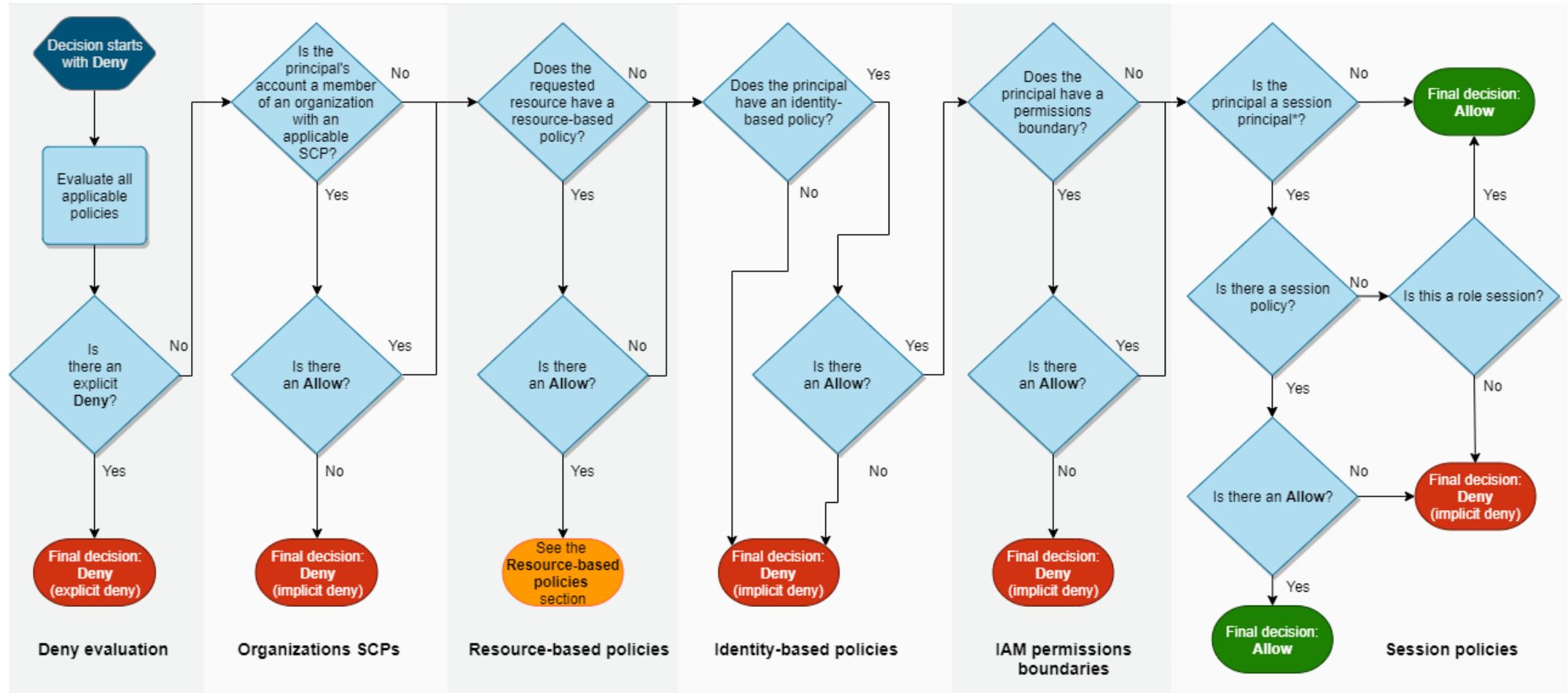


https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_boundaries.html

Use cases

- Delegate responsibilities to non administrators within their permission boundaries, for example create new IAM users
- Allow developers to self-assign policies and manage their own permissions, while making sure they can't "escalate" their privileges (= make themselves admin)
- Useful to restrict one specific user (instead of a whole account using Organizations & SCP)

IAM Policy Evaluation Logic



*A session principal is either a role session or an IAM federated user session.

https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_evaluation-logic.html

Example IAM Policy

- Can you perform sqs:CreateQueue?
- Can you perform sqs:DeleteQueue?
- Can you perform ec2:DescribeInstances?

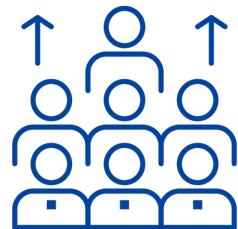
```
"Version": "2012-10-17",
"Statement": [
  {
    "Action": "sts:AssumeRole",
    "Effect": "Allow",
    "Resource": "*"
  },
  {
    "Action": [
      "sts:AssumeRole"
    ],
    "Effect": "Allow",
    "Resource": "*"
  }
]
```

AWS IAM Identity Center

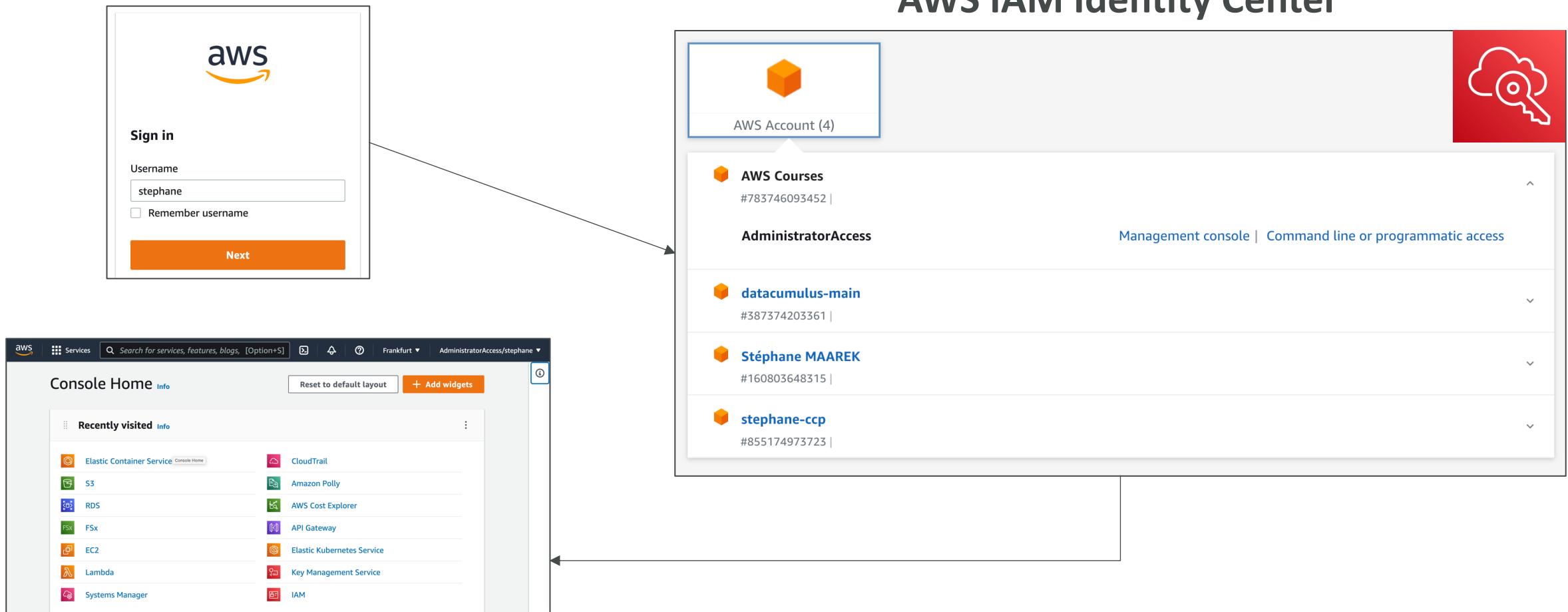
(successor to AWS Single Sign-On)



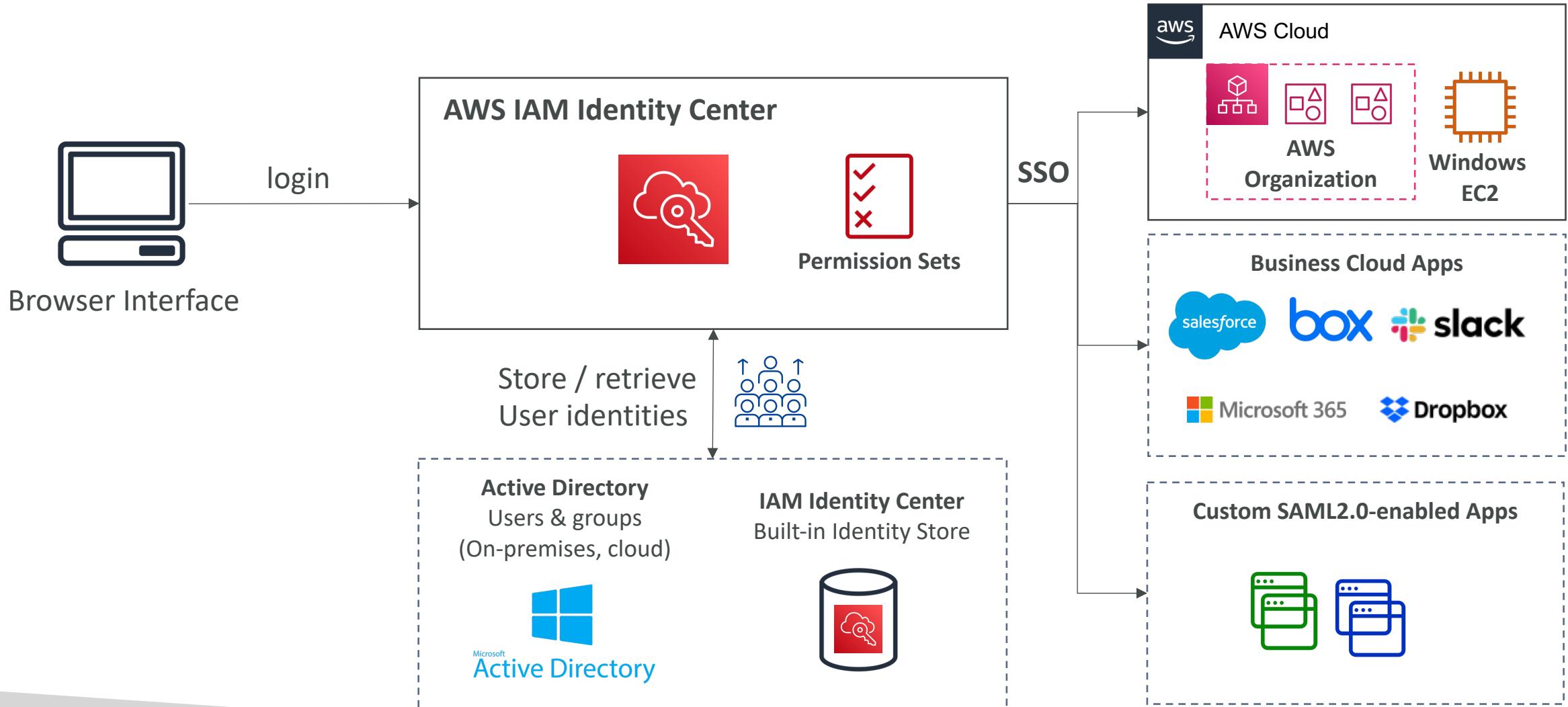
- One login (single sign-on) for all your
 - AWS accounts in AWS Organizations
 - Business cloud applications (e.g., Salesforce, Box, Microsoft 365, ...)
 - SAML2.0-enabled applications
 - EC2 Windows Instances
- Identity providers
 - Built-in identity store in IAM Identity Center
 - 3rd party: Active Directory (AD), OneLogin, Okta...



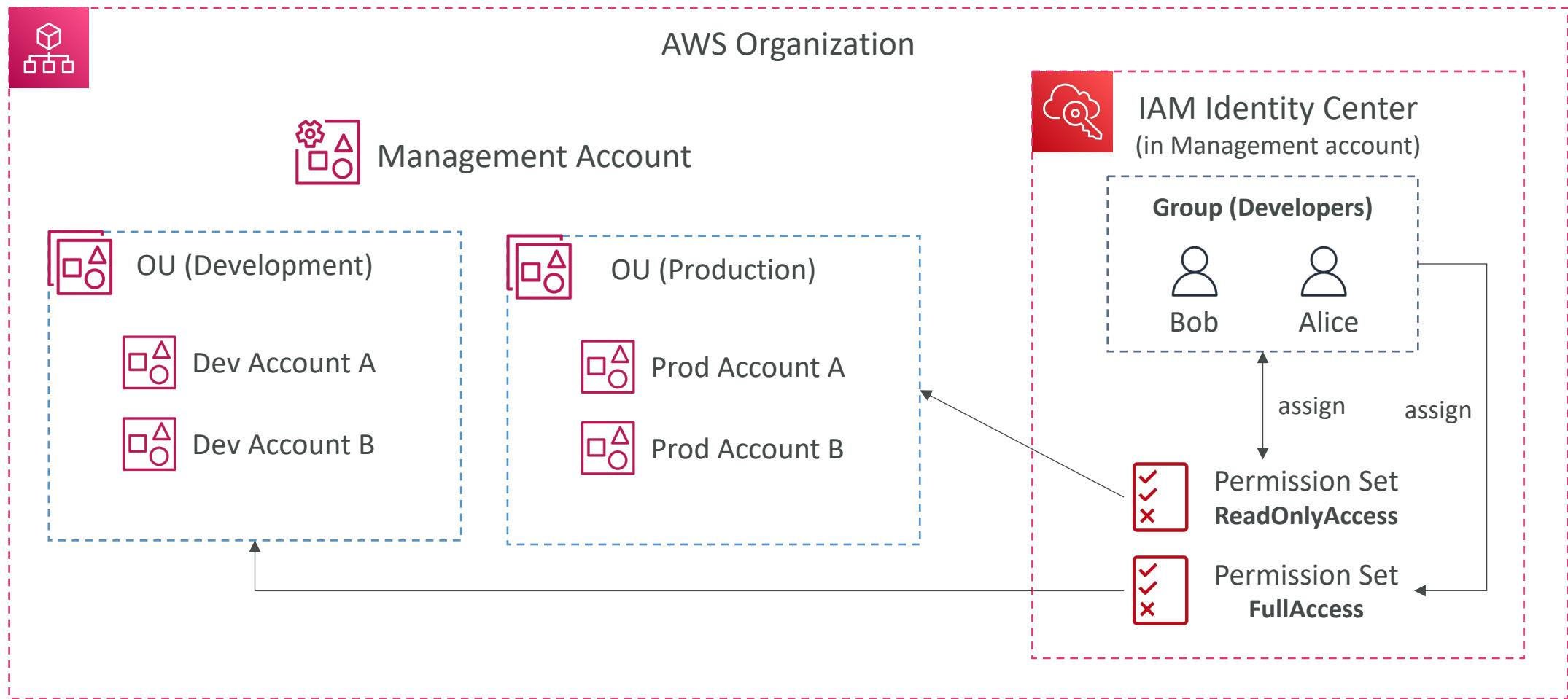
AWS IAM Identity Center – Login Flow



AWS IAM Identity Center



IAM Identity Center

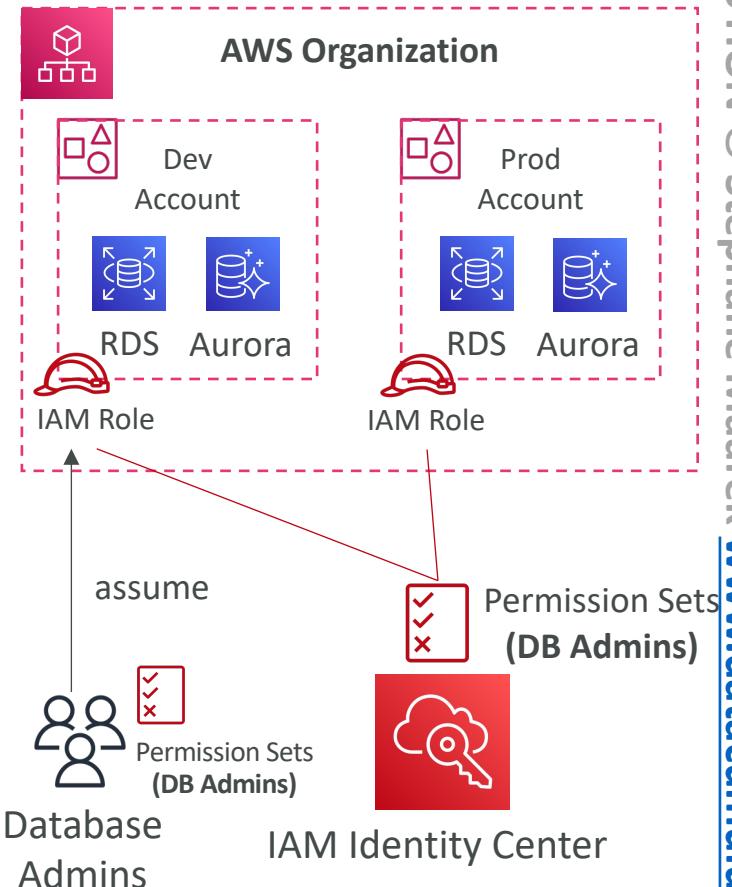


AWS IAM Identity Center

Fine-grained Permissions and Assignments

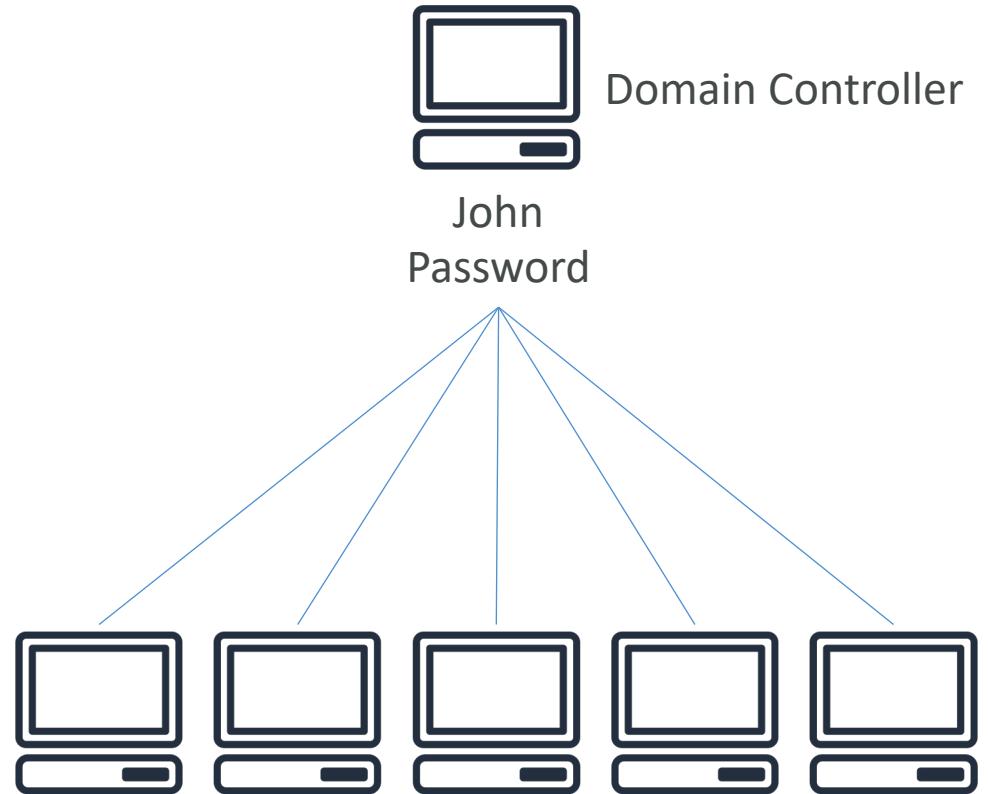


- Multi-Account Permissions
 - Manage access across AWS accounts in your AWS Organization
 - Permission Sets – a collection of one or more IAM Policies assigned to users and groups to define AWS access
- Application Assignments
 - SSO access to many SAML 2.0 business applications (Salesforce, Box, Microsoft 365, ...)
 - Provide required URLs, certificates, and metadata
- Attribute-Based Access Control (ABAC)
 - Fine-grained permissions based on users' attributes stored in IAM Identity Center Identity Store
 - Example: cost center, title, locale, ...
 - Use case: Define permissions once, then modify AWS access by changing the attributes



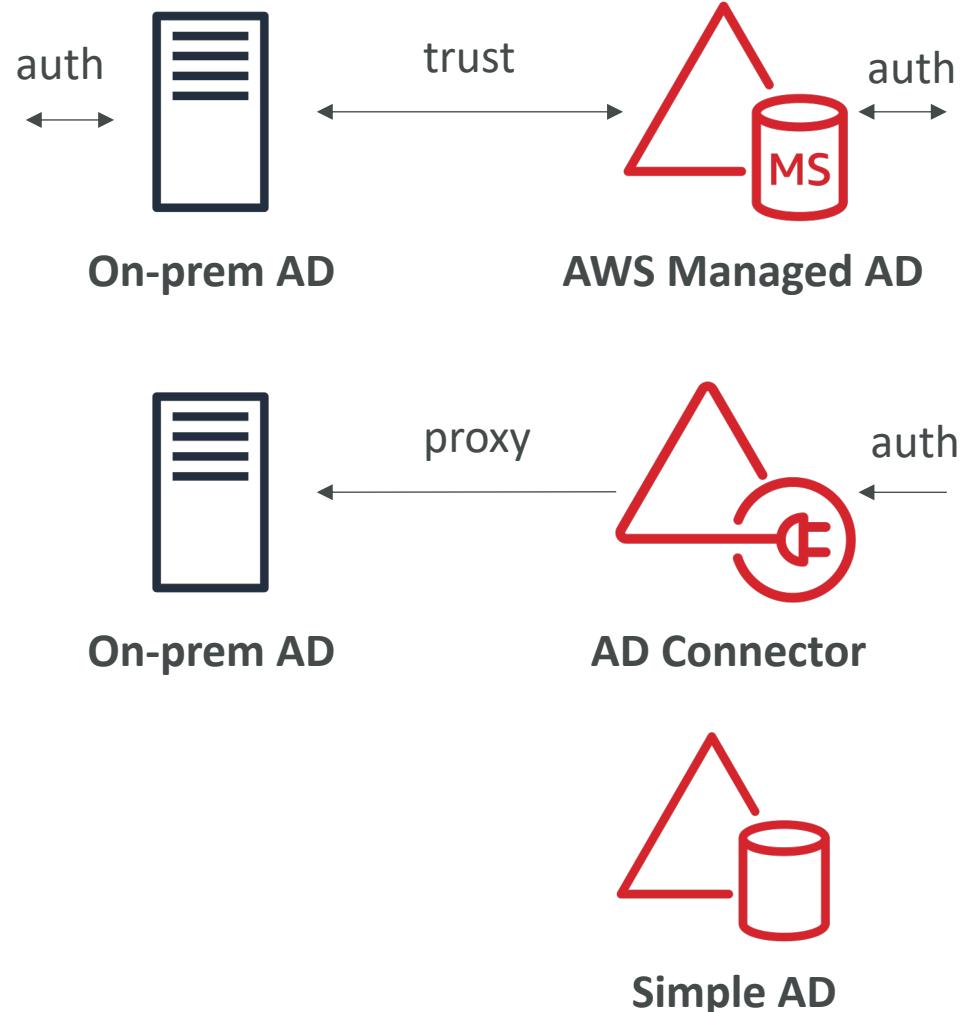
What is Microsoft Active Directory (AD)?

- Found on any Windows Server with AD Domain Services
- Database of **objects**: User Accounts, Computers, Printers, File Shares, Security Groups
- Centralized security management, create account, assign permissions
- Objects are organized in **trees**
- A group of trees is a **forest**



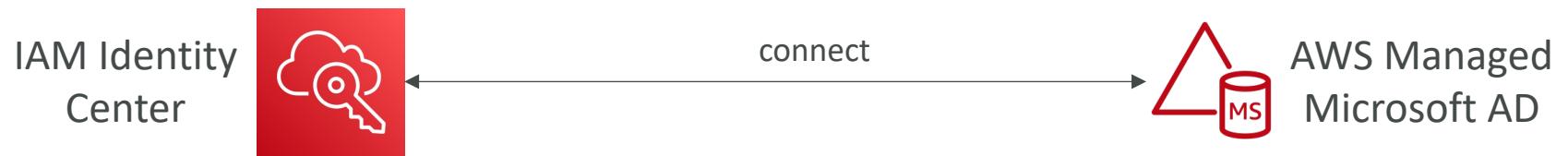
AWS Directory Services

- AWS Managed Microsoft AD
 - Create your own AD in AWS, manage users locally, supports MFA
 - Establish “trust” connections with your on-premises AD
- AD Connector
 - Directory Gateway (proxy) to redirect to on-premises AD, supports MFA
 - Users are managed on the on-premises AD
- Simple AD
 - AD-compatible managed directory on AWS
 - Cannot be joined with on-premises AD



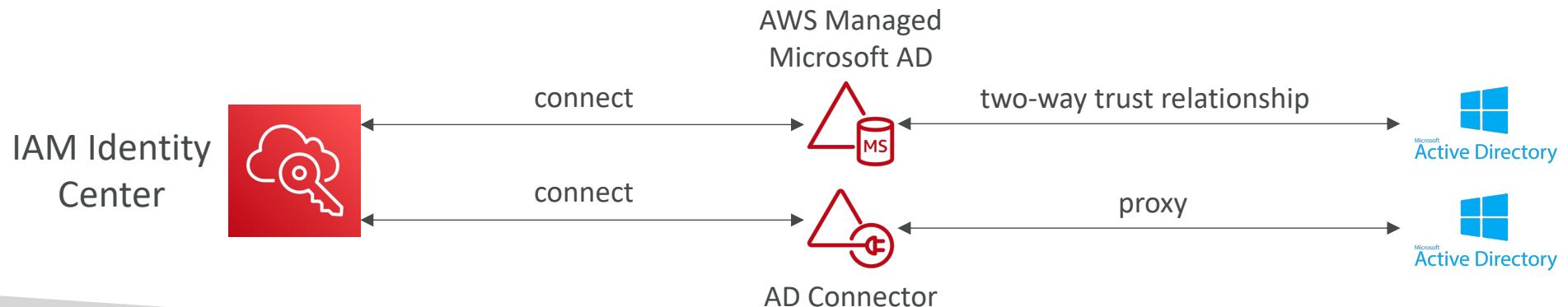
IAM Identity Center – Active Directory Setup

- Connect to an AWS Managed Microsoft AD (Directory Service)
 - Integration is out of the box



- Connect to a Self-Managed Directory

- Create Two-way Trust Relationship using AWS Managed Microsoft AD
- Create an AD Connector



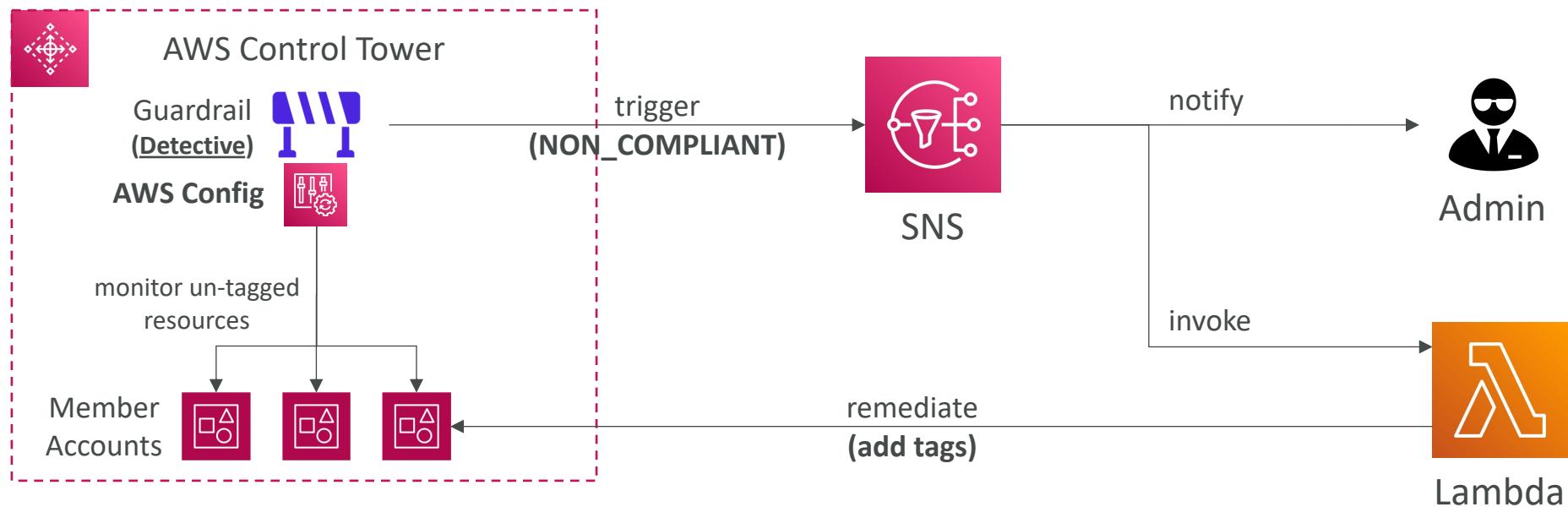
AWS Control Tower



- Easy way to set up and govern a secure and compliant multi-account AWS environment based on best practices
- AWS Control Tower uses AWS Organizations to create accounts
- Benefits:
 - Automate the set up of your environment in a few clicks
 - Automate ongoing policy management using guardrails
 - Detect policy violations and remediate them
 - Monitor compliance through an interactive dashboard

AWS Control Tower – Guardrails

- Provides ongoing governance for your Control Tower environment (AWS Accounts)
- Preventive Guardrail – using SCPs (e.g., Restrict Regions across all your accounts)
- Detective Guardrail – using AWS Config (e.g., identify untagged resources)



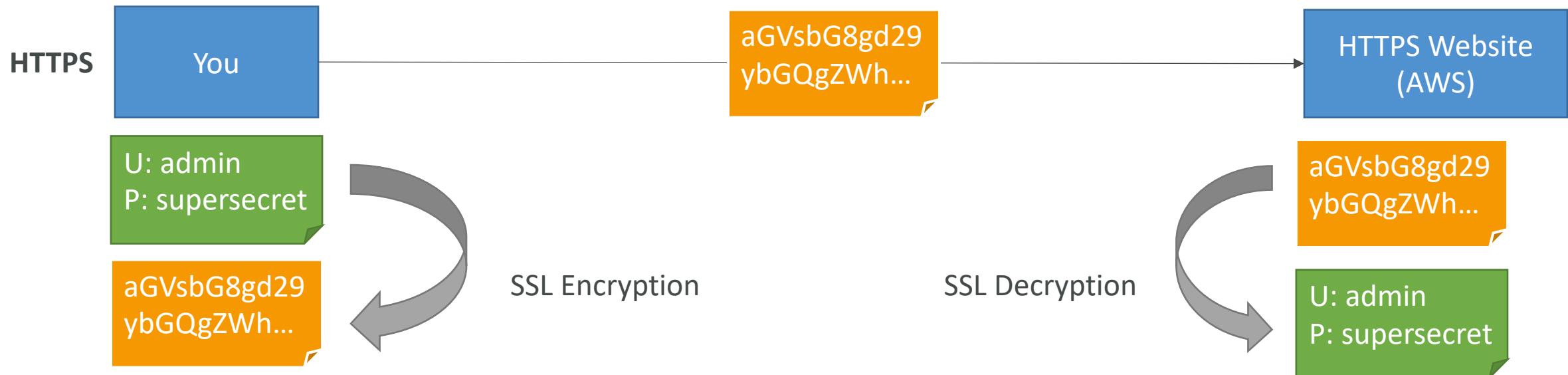
AWS Security & Encryption

KMS, Encryption SDK, SSM Parameter Store

Why encryption?

Encryption in flight (SSL)

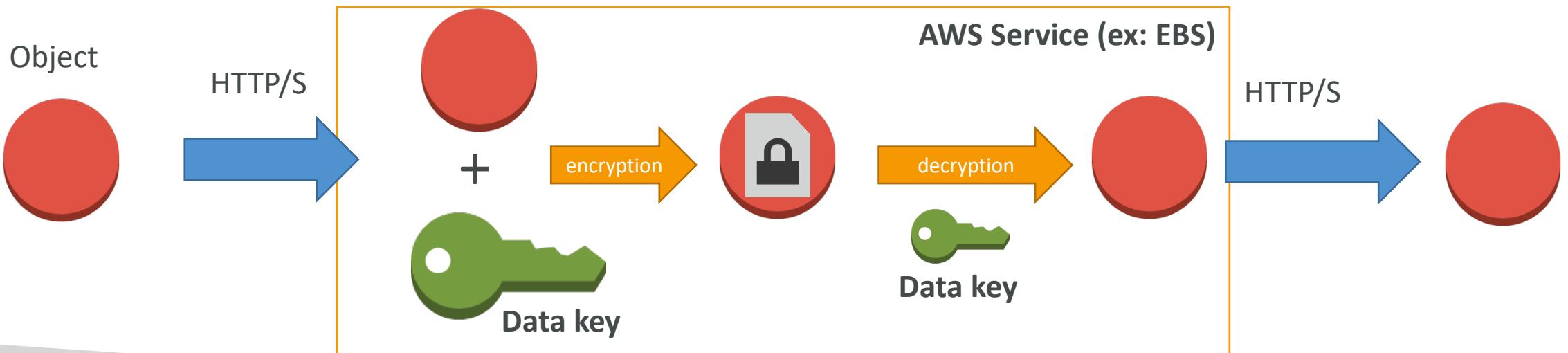
- Data is encrypted before sending and decrypted after receiving
- SSL certificates help with encryption (HTTPS)
- Encryption in flight ensures no MITM (man in the middle attack) can happen



Why encryption?

Server side encryption at rest

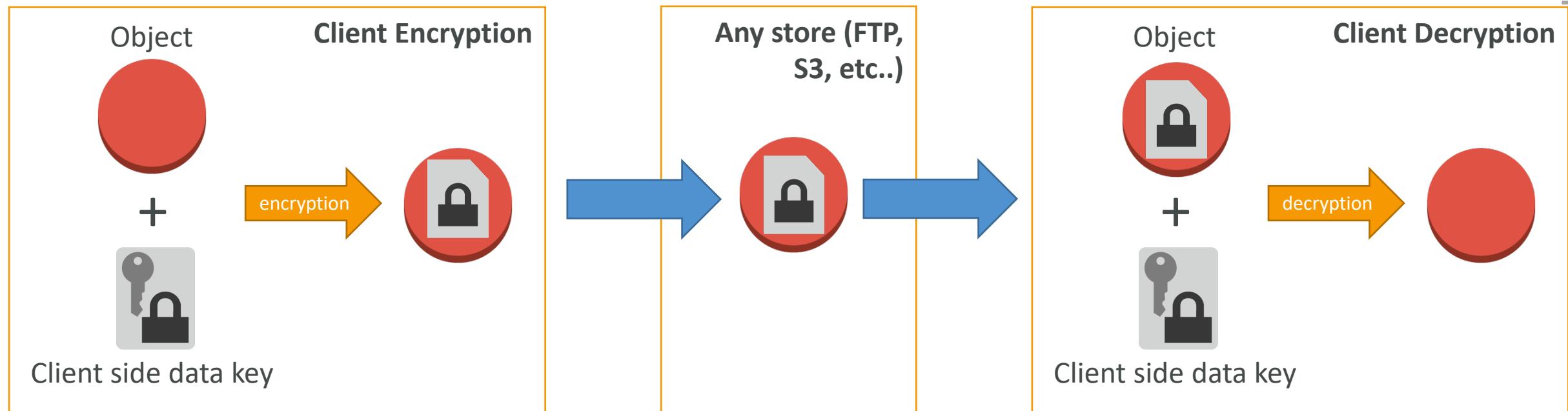
- Data is encrypted after being received by the server
- Data is decrypted before being sent
- It is stored in an encrypted form thanks to a key (usually a data key)
- The encryption / decryption keys must be managed somewhere and the server must have access to it



Why encryption?

Client side encryption

- Data is encrypted by the client and never decrypted by the server
- Data will be decrypted by a receiving client
- The server should not be able to decrypt the data
- Could leverage Envelope Encryption



AWS KMS (Key Management Service)



- Anytime you hear “encryption” for an AWS service, it’s most likely KMS
- AWS manages encryption keys for us
- Fully integrated with IAM for authorization
- Easy way to control access to your data
- Able to audit KMS Key usage using CloudTrail
- Seamlessly integrated into most AWS services (EBS, S3, RDS, SSM...)
- **Never ever store your secrets in plaintext, especially in your code!**
 - KMS Key Encryption also available through API calls (SDK, CLI)
 - Encrypted secrets can be stored in the code / environment variables

KMS Keys Types

- KMS Keys is the new name of KMS Customer Master Key
- Symmetric (AES-256 keys)
 - Single encryption key that is used to Encrypt and Decrypt
 - AWS services that are integrated with KMS use Symmetric CMKs
 - You never get access to the KMS Key unencrypted (must call KMS API to use)
- Asymmetric (RSA & ECC key pairs)
 - Public (Encrypt) and Private Key (Decrypt) pair
 - Used for Encrypt/Decrypt, or Sign/Verify operations
 - The public key is downloadable, but you can't access the Private Key unencrypted
 - Use case: encryption outside of AWS by users who can't call the KMS API

AWS KMS (Key Management Service)



- Types of KMS Keys:

- AWS Owned Keys (free): SSE-S3, SSE-SQS, SSE-DDB (default key)
- AWS Managed Key: **free** (aws/service-name, example: aws/rds or aws/ebs)
- Customer managed keys created in KMS: \$1 / month
- Customer managed keys imported (must be symmetric key): \$1 / month
- + pay for API call to KMS (\$0.03 / 10000 calls)

Encryption key management

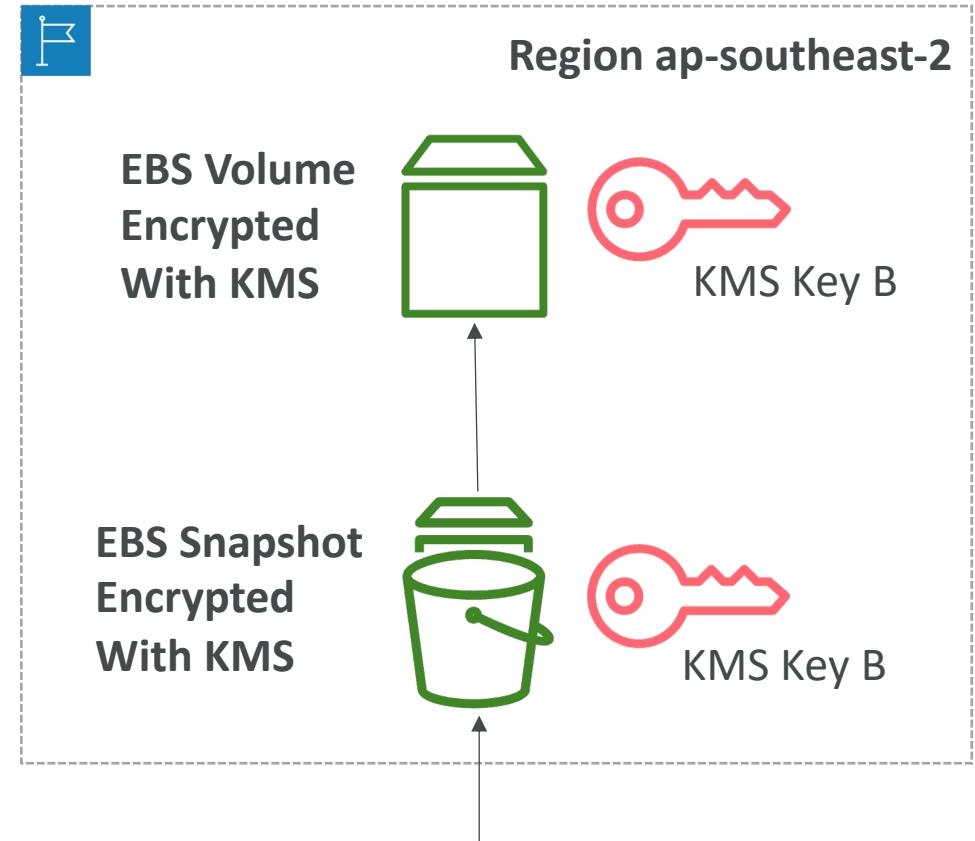
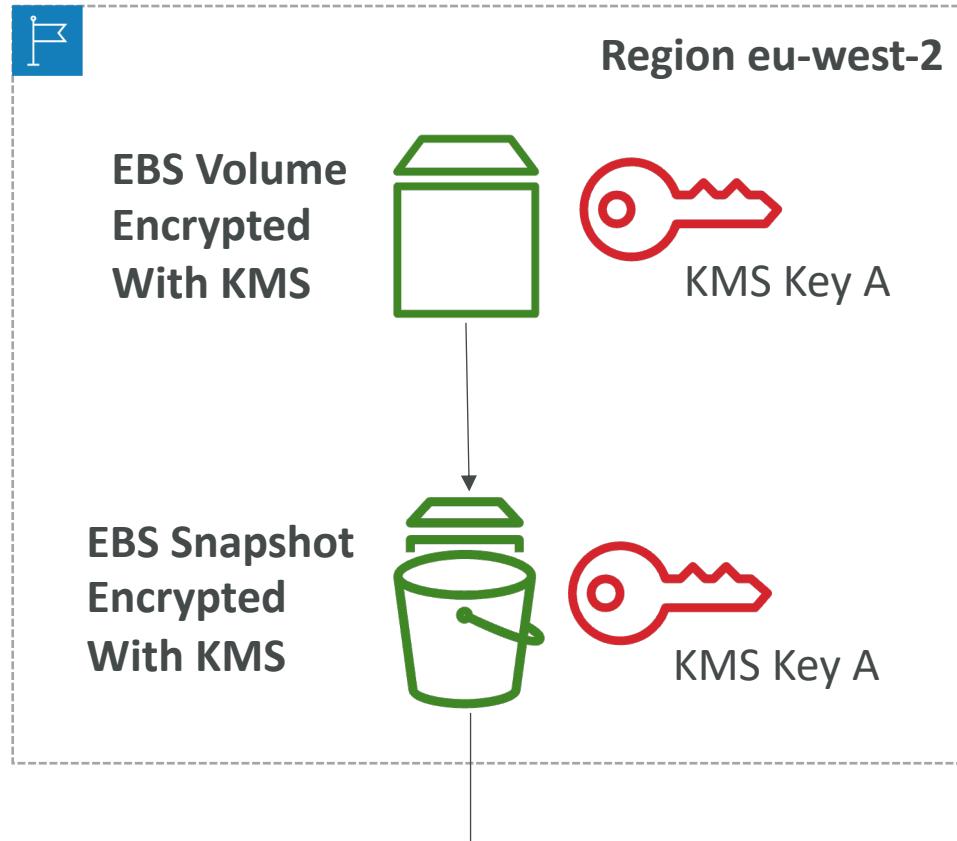
- Owned by Amazon DynamoDB
- AWS managed key **Lea**
Key alias: aws/dynamodb.
- Stored in your account,
and owned and managed by you

C

- Automatic Key rotation:

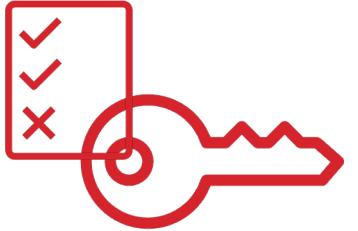
- AWS-managed KMS Key: automatic every 1 year
- Customer-managed KMS Key: (must be enabled) automatic every 1 year
- Imported KMS Key: only manual rotation possible using alias

Copying Snapshots across regions



KMS ReEncrypt with KMS Key B

KMS Key Policies



- Control access to KMS keys, “similar” to S3 bucket policies
- Difference: you cannot control access without them
- **Default KMS Key Policy:**
 - Created if you don't provide a specific KMS Key Policy
 - Complete access to the key to the root user = entire AWS account
- **Custom KMS Key Policy:**
 - Define users, roles that can access the KMS key
 - Define who can administer the key
 - Useful for cross-account access of your KMS key

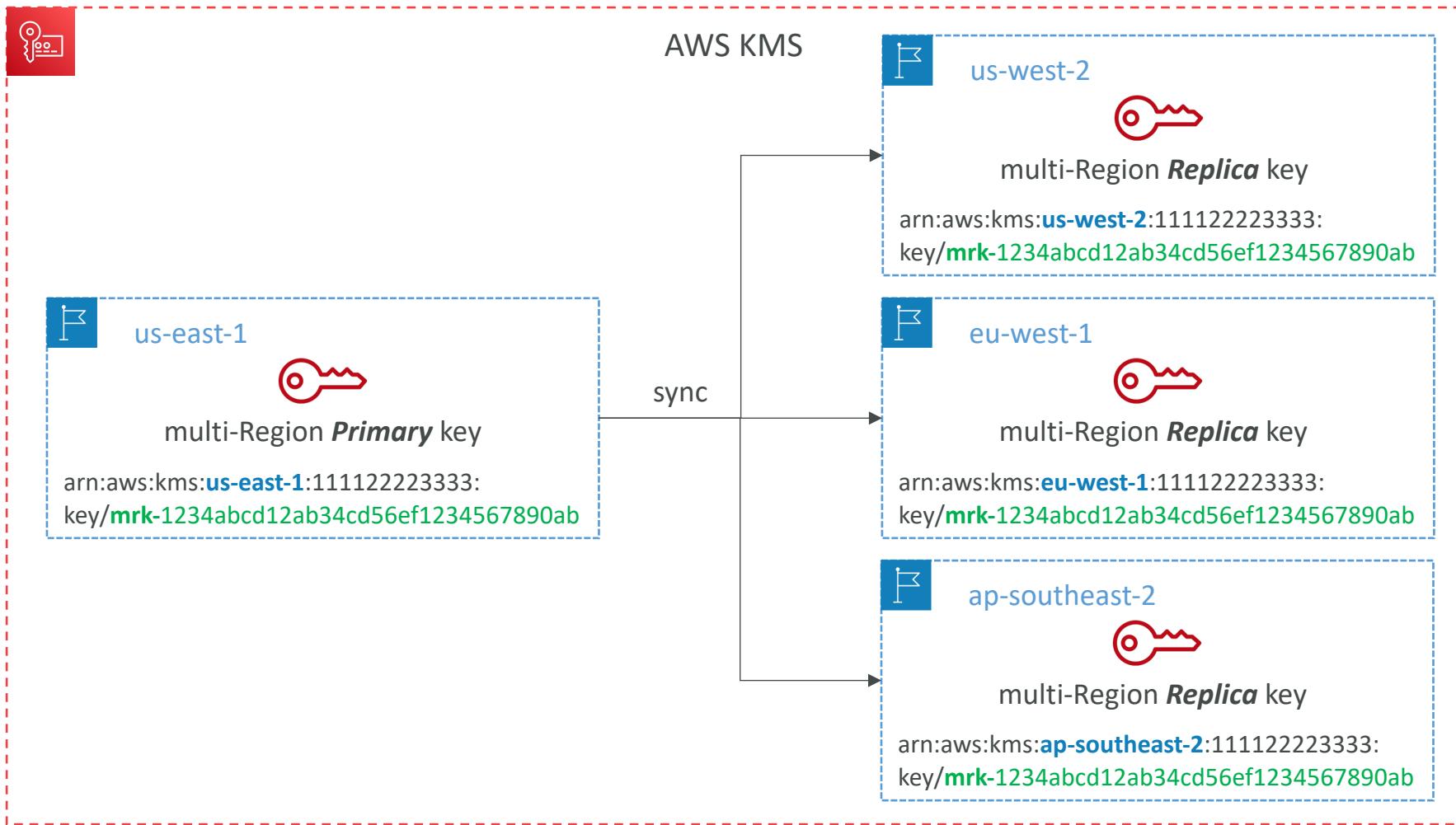
Copying Snapshots across accounts

1. Create a Snapshot, encrypted with your own KMS Key (Customer Managed Key)
2. Attach a KMS Key Policy to authorize cross-account access
3. Share the encrypted snapshot
4. (in target) Create a copy of the Snapshot, encrypt it with a CMK in your account
5. Create a volume from the snapshot

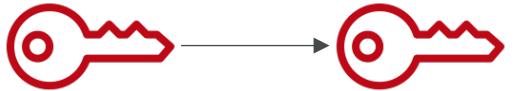
```
{  
  "Sid": "Allow use of the key with destination account",  
  "Effect": "Allow",  
  "Principal": {  
    "AWS": "arn:aws:iam::TARGET-ACCOUNT-ID:role/ROLENAMESPACE"  
  },  
  "Action": [  
    "kms:Decrypt",  
    "kms>CreateGrant"  
  ],  
  "Resource": "*",  
  "Condition": {  
    "StringEquals": {  
      "kms:ViaService": "ec2.REGION.amazonaws.com",  
      "kms:CallerAccount": "TARGET-ACCOUNT-ID"  
    }  
  }  
}
```

KMS Key Policy

KMS Multi-Region Keys



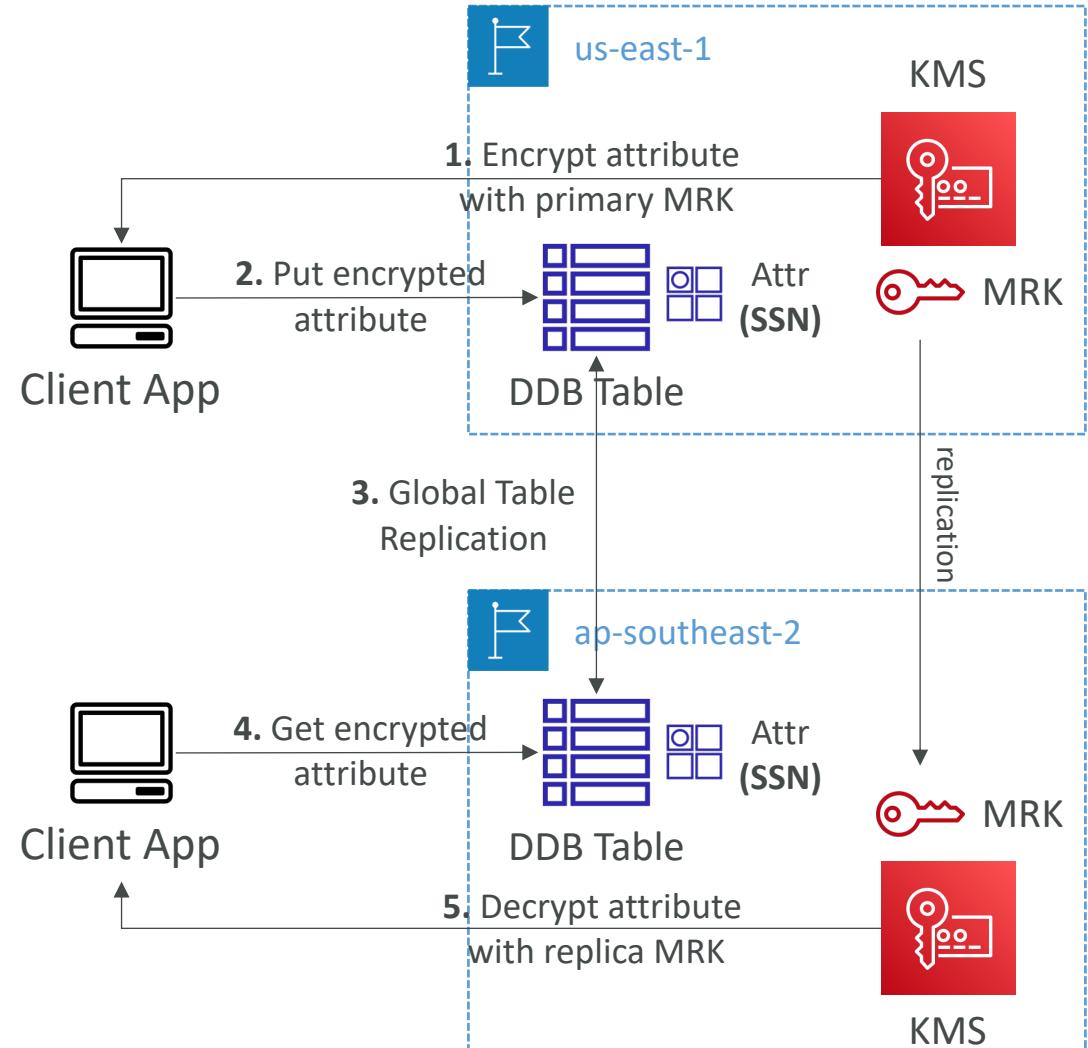
KMS Multi-Region Keys



- Identical KMS keys in different AWS Regions that can be used interchangeably
- Multi-Region keys have the same key ID, key material, automatic rotation...
- Encrypt in one Region and decrypt in other Regions
- No need to re-encrypt or making cross-Region API calls
- KMS Multi-Region are NOT global (Primary + Replicas)
- Each Multi-Region key is managed **independently**
- **Use cases:** global client-side encryption, encryption on Global DynamoDB, Global Aurora

DynamoDB Global Tables and KMS Multi-Region Keys Client-Side encryption

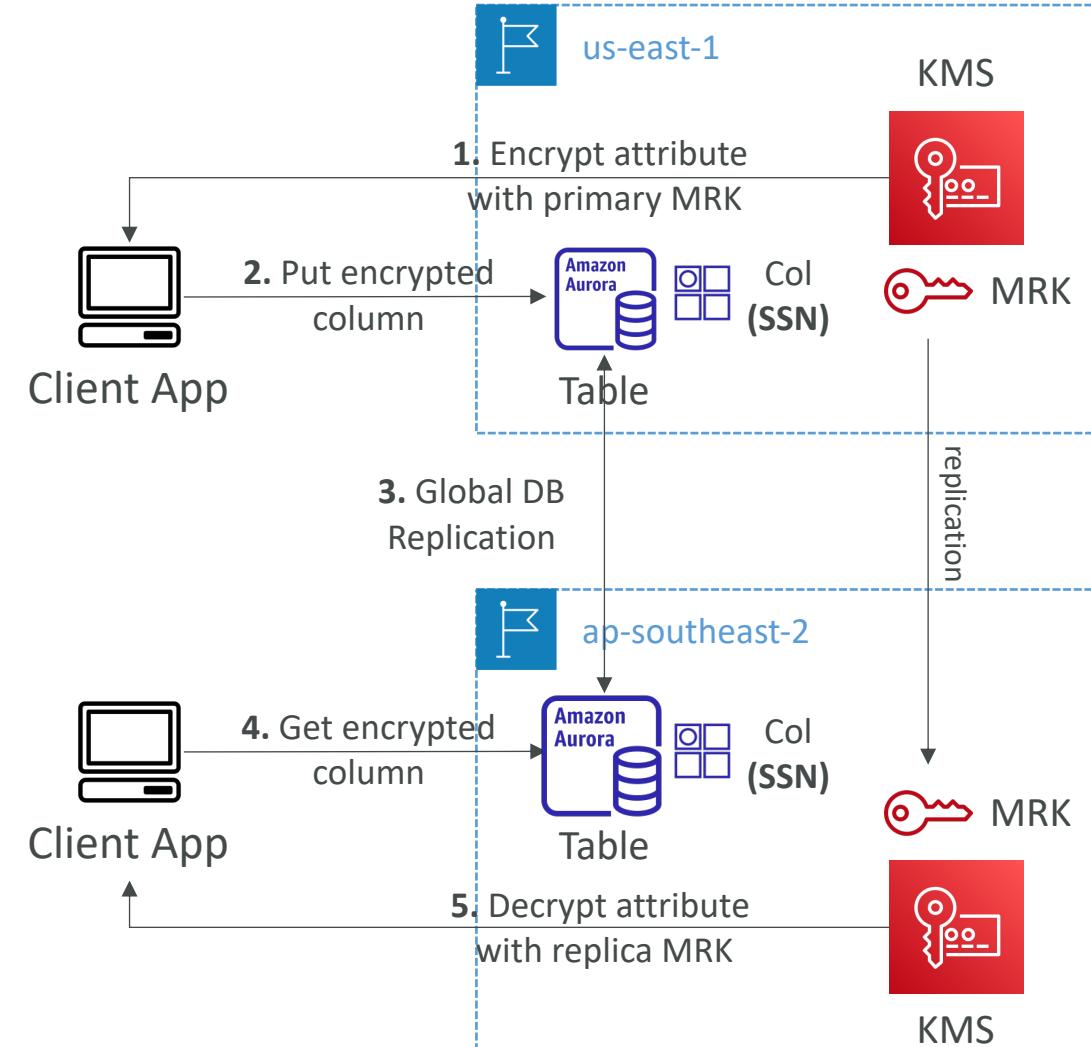
- We can encrypt specific attributes client-side in our DynamoDB table using the **Amazon DynamoDB Encryption Client**
- Combined with Global Tables, the client-side encrypted data is replicated to other regions
- If we use a multi-region key, replicated in the same region as the DynamoDB Global table, then clients in these regions can use low-latency API calls to KMS in their region to decrypt the data client-side
- Using client-side encryption we can protect specific fields and guarantee only decryption if the client has access to an API key



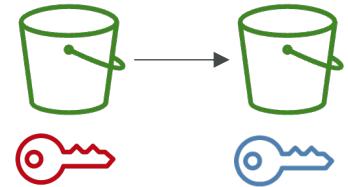
Global Aurora and KMS Multi-Region Keys

Client-Side encryption

- We can encrypt specific attributes client-side in our Aurora table using the **AWS Encryption SDK**
- Combined with Aurora Global Tables, the client-side encrypted data is replicated to other regions
- If we use a multi-region key, replicated in the same region as the Global Aurora DB, then clients in these regions can use low-latency API calls to KMS in their region to decrypt the data client-side
- Using client-side encryption we can protect specific fields and guarantee only decryption if the client has access to an API key, we can **protect specific fields even from database admins**



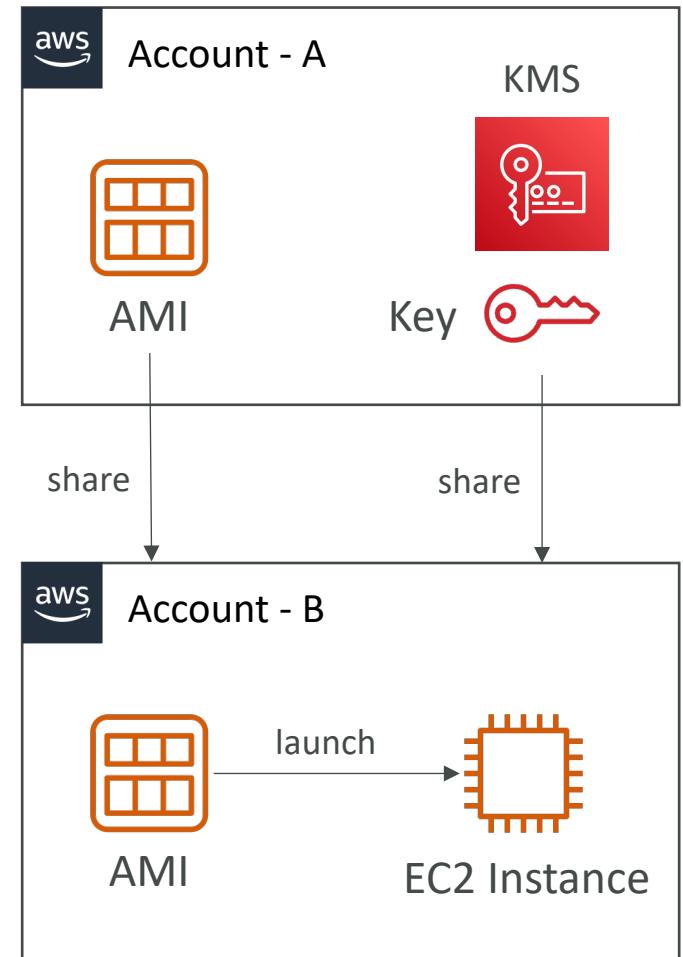
S3 Replication Encryption Considerations



- Unencrypted objects and objects encrypted with SSE-S3 are replicated by default
- Objects encrypted with SSE-C (customer provided key) are never replicated
- For objects encrypted with SSE-KMS, you need to enable the option
 - Specify which KMS Key to encrypt the objects within the target bucket
 - Adapt the KMS Key Policy for the target key
 - An IAM Role with kms:Decrypt for the source KMS Key and kms:Encrypt for the target KMS Key
 - You might get KMS throttling errors, in which case you can ask for a Service Quotas increase
- You can use multi-region AWS KMS Keys, but they are currently treated as independent keys by Amazon S3 (the object will still be decrypted and then encrypted)

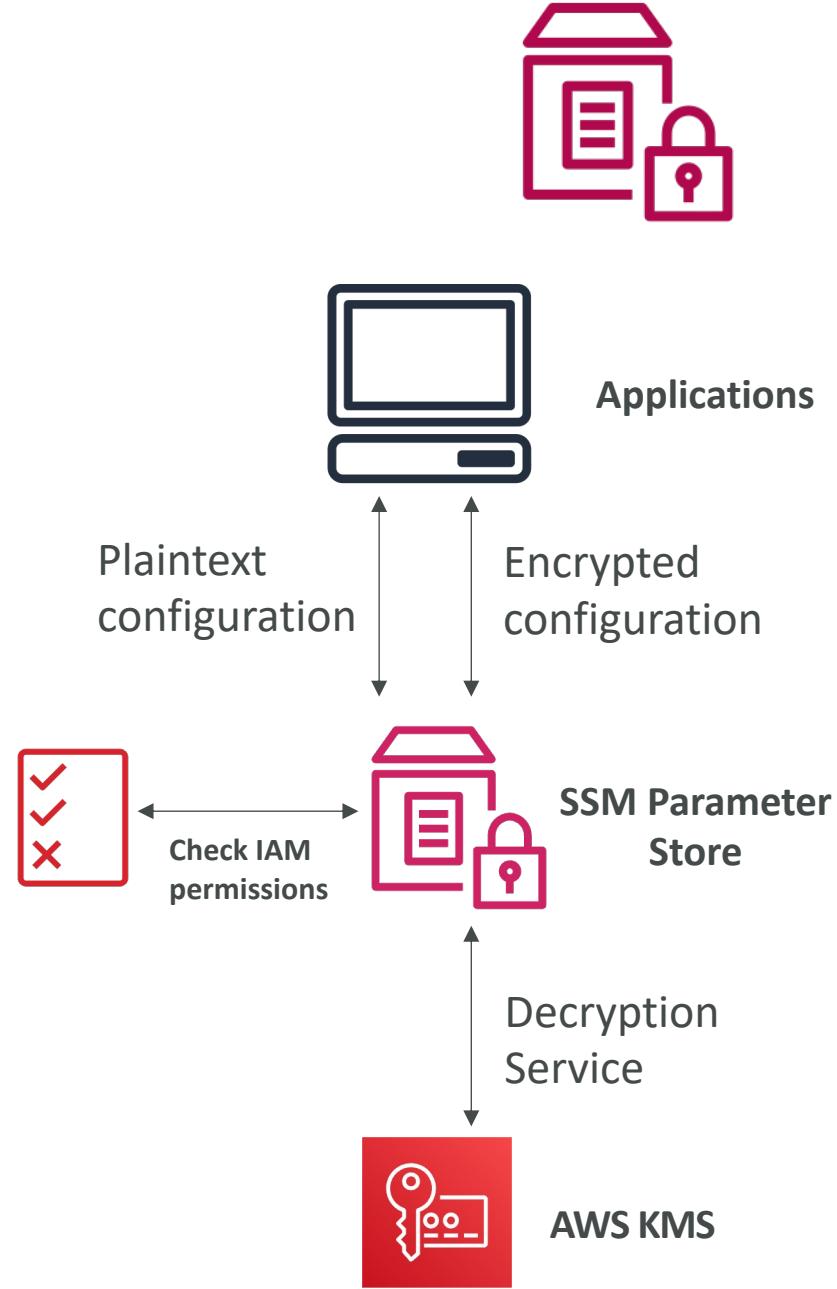
AMI Sharing Process Encrypted via KMS

1. AMI in Source Account is encrypted with KMS Key from Source Account
2. Must modify the image attribute to add a **Launch Permission** which corresponds to the specified target AWS account
3. Must share the KMS Keys used to encrypted the snapshot the AMI references with the target account / IAM Role
4. The IAM Role/User in the target account must have the permissions to `DescribeKey`, `ReEncrypted`, `CreateGrant`, `Decrypt`
5. When launching an EC2 instance from the AMI, optionally the target account can specify a new KMS key in its own account to re-encrypt the volumes



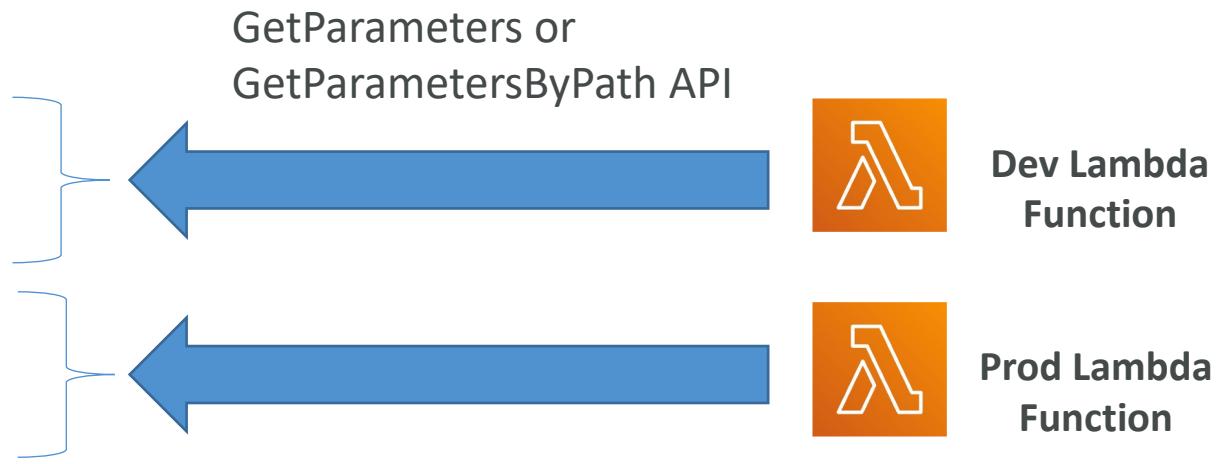
SSM Parameter Store

- Secure storage for configuration and secrets
- Optional Seamless Encryption using KMS
- Serverless, scalable, durable, easy SDK
- Version tracking of configurations / secrets
- Security through IAM
- Notifications with Amazon EventBridge
- Integration with CloudFormation



SSM Parameter Store Hierarchy

- /my-department/
 - my-app/
 - dev/
 - db-url
 - db-password
 - prod/
 - db-url
 - db-password
 - other-app/
 - /other-department/
 - /aws/reference/secretsmanager/secret_ID_in_Secrets_Manager
 - /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2 (public)



Standard and advanced parameter tiers

	Standard	Advanced
Total number of parameters allowed (per AWS account and Region)	10,000	100,000
Maximum size of a parameter value	4 KB	8 KB
Parameter policies available	No	Yes
Cost	No additional charge	Charges apply
Storage Pricing	Free	\$0.05 per advanced parameter per month

Parameters Policies (for advanced parameters)

- Allow to assign a TTL to a parameter (expiration date) to force updating or deleting sensitive data such as passwords
- Can assign multiple policies at a time

Expiration (to delete a parameter)

```
{  
  "Type": "Expiration",  
  "Version": "1.0",  
  "Attributes": {  
    "Timestamp": "2020-12-02T21:34:33.000Z"  
  }  
}
```

ExpirationNotification (EventBridge)

```
{  
  "Type": "ExpirationNotification",  
  "Version": "1.0",  
  "Attributes": {  
    "Before": "15",  
    "Unit": "Days"  
  }  
}
```

NoChangeNotification (EventBridge)

```
{  
  "Type": "NoChangeNotification",  
  "Version": "1.0",  
  "Attributes": {  
    "After": "20",  
    "Unit": "Days"  
  }  
}
```

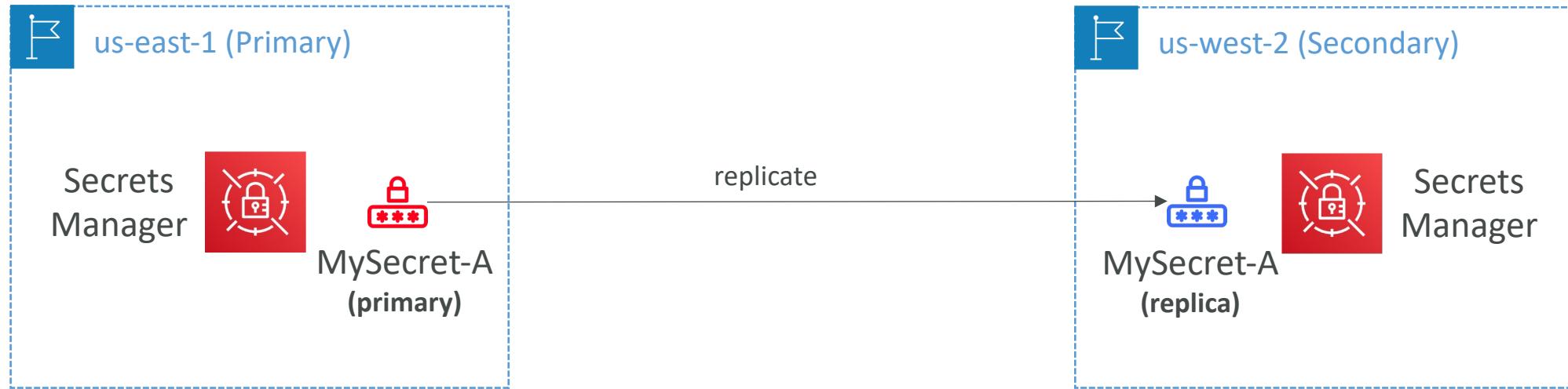
AWS Secrets Manager



- Newer service, meant for storing secrets
- Capability to force **rotation of secrets** every X days
- Automate generation of secrets on rotation (uses Lambda)
- Integration with **Amazon RDS** (MySQL, PostgreSQL, Aurora)
- Secrets are encrypted using KMS
- Mostly meant for RDS integration

AWS Secrets Manager – Multi-Region Secrets

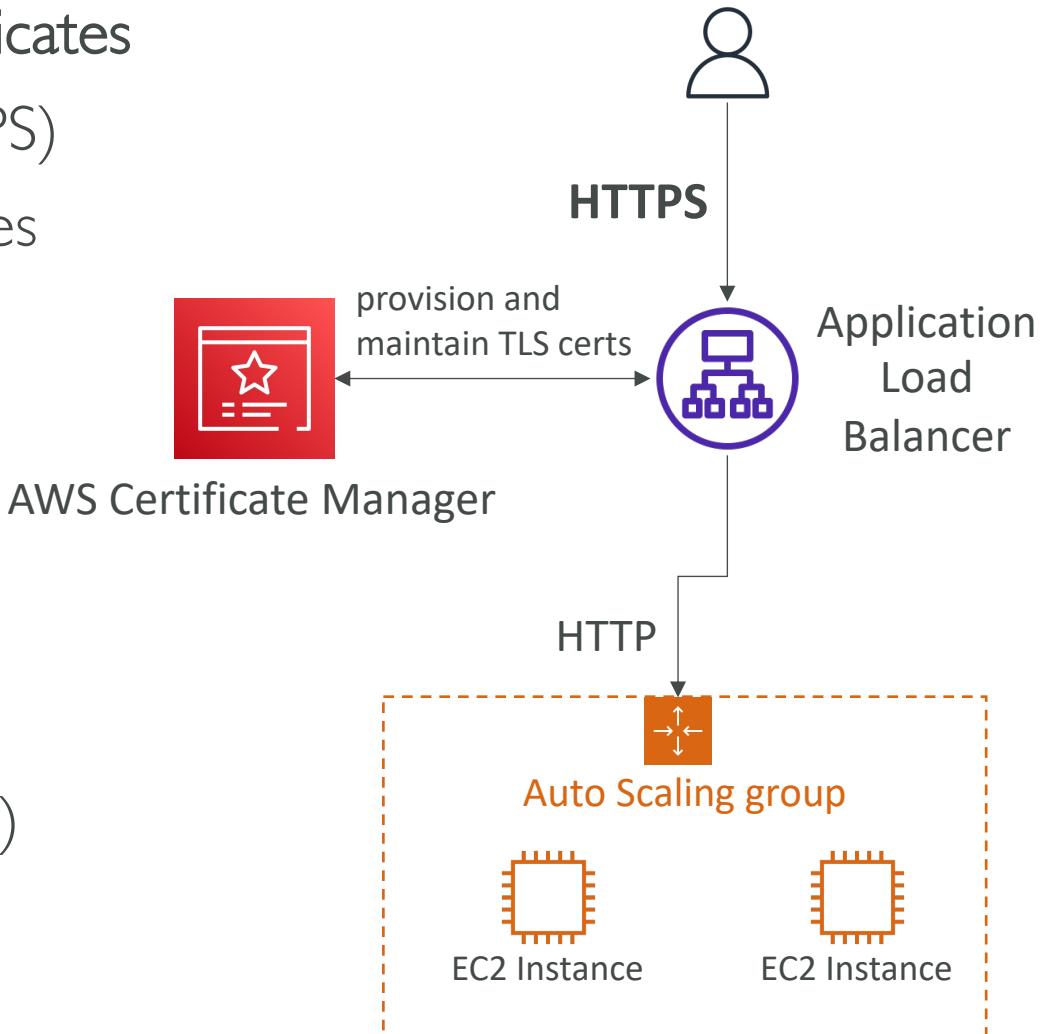
- Replicate Secrets across multiple AWS Regions
- Secrets Manager keeps read replicas in sync with the primary Secret
- Ability to promote a read replica Secret to a standalone Secret
- Use cases: multi-region apps, disaster recovery strategies, multi-region DB...



AWS Certificate Manager (ACM)



- Easily provision, manage, and deploy **TLS Certificates**
- Provide in-flight encryption for websites (HTTPS)
- Supports both public and private TLS certificates
- Free of charge for public TLS certificates
- Automatic TLS certificate renewal
- Integrations with (load TLS certificates on)
 - Elastic Load Balancers (CLB, ALB, NLB)
 - CloudFront Distributions
 - APIs on API Gateway
- Cannot use ACM with EC2 (can't be extracted)

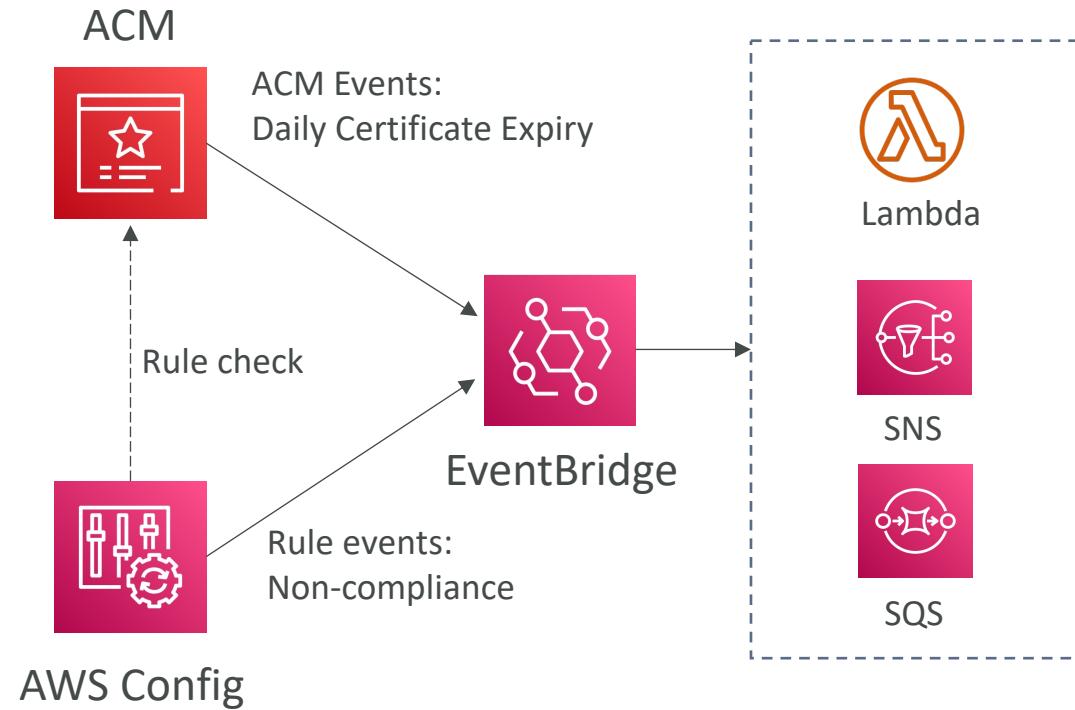


ACM – Requesting Public Certificates

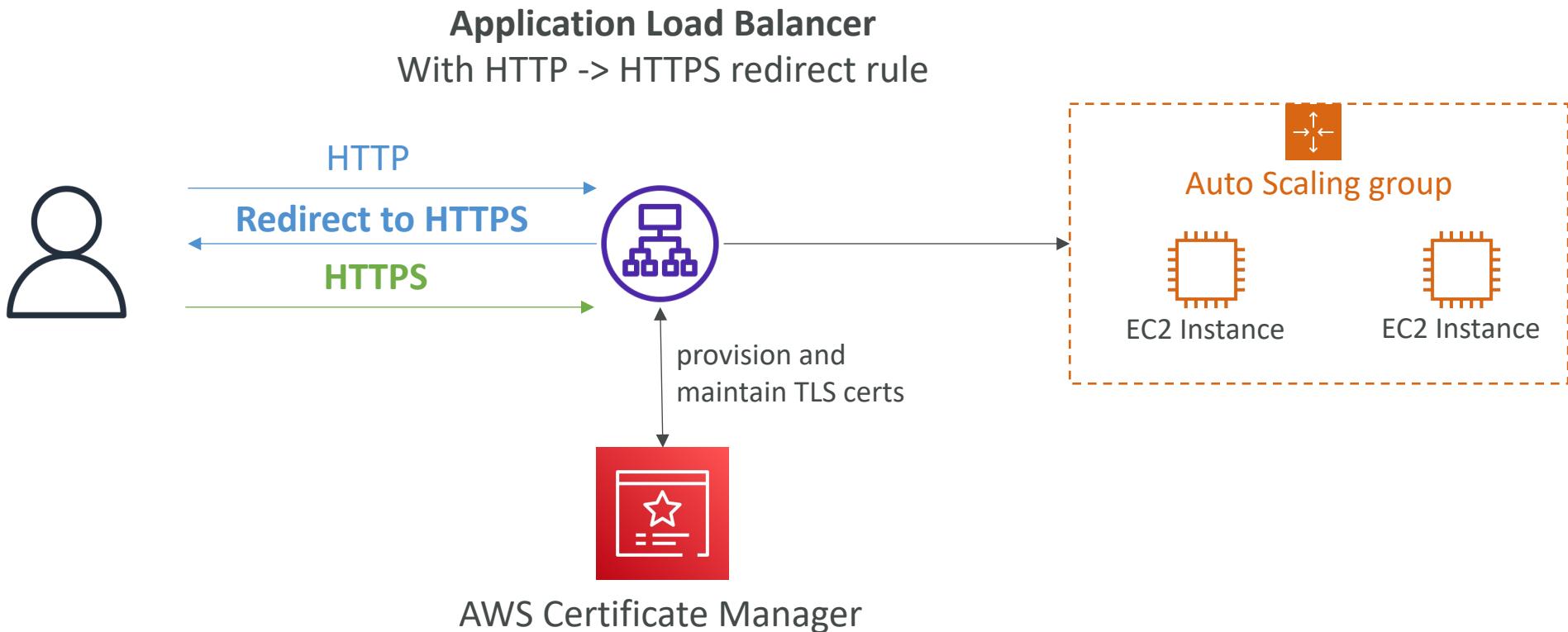
1. List domain names to be included in the certificate
 - Fully Qualified Domain Name (FQDN): corp.example.com
 - Wildcard Domain: *.example.com
2. Select Validation Method: DNS Validation or Email validation
 - DNS Validation is preferred for automation purposes
 - Email validation will send emails to contact addresses in the WHOIS database
 - DNS Validation will leverage a CNAME record to DNS config (ex: Route 53)
3. It will take a few hours to get verified
4. The Public Certificate will be enrolled for automatic renewal
 - ACM automatically renews ACM-generated certificates 60 days before expiry

ACM – Importing Public Certificates

- Option to generate the certificate outside of ACM and then import it
- No automatic renewal, must import a new certificate before expiry
- ACM sends daily expiration events starting 45 days prior to expiration
 - The # of days can be configured
 - Events are appearing in EventBridge
- AWS Config has a managed rule named `acm-certificate-expiration-check` to check for expiring certificates (configurable number of days)



ACM – Integration with ALB

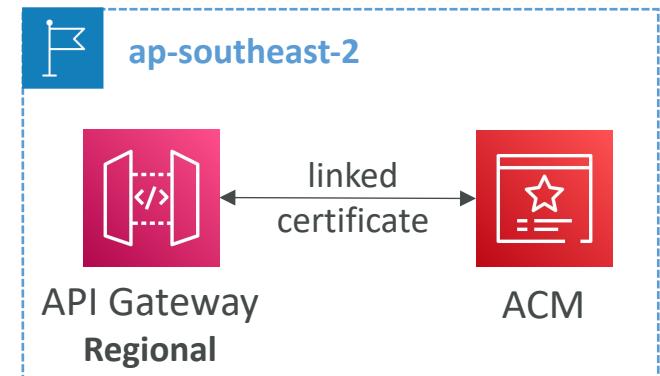
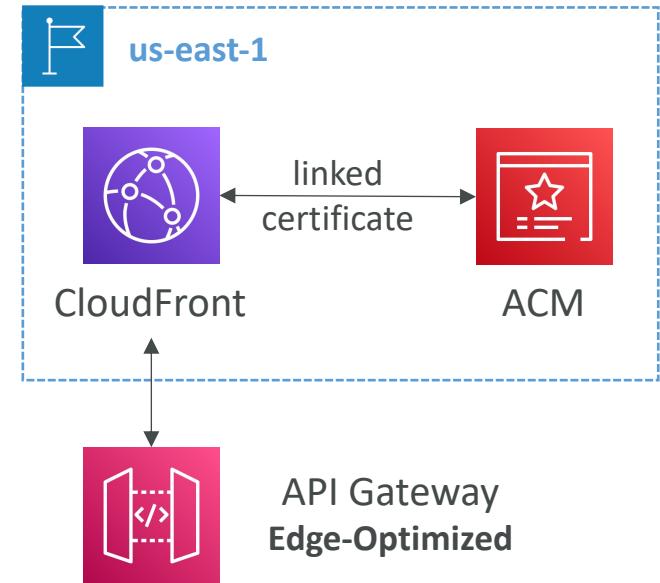


API Gateway - Endpoint Types

- **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

ACM – Integration with API Gateway

- Create a **Custom Domain Name** in API Gateway
- **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
 - The TLS Certificate must be in the same region as CloudFront, in us-east-1
 - Then setup CNAME or (better) A-Alias record in Route 53
- **Regional:**
 - For clients within the same region
 - The TLS Certificate must be imported on API Gateway, in the same region as the API Stage
 - Then setup CNAME or (better) A-Alias record in Route 53



AWS WAF – Web Application Firewall



- Protects your web applications from common web exploits (Layer 7)
- Layer 7 is HTTP (vs Layer 4 is TCP/UDP)
- Deploy on
 - Application Load Balancer
 - API Gateway
 - CloudFront
 - AppSync GraphQL API
 - Cognito User Pool

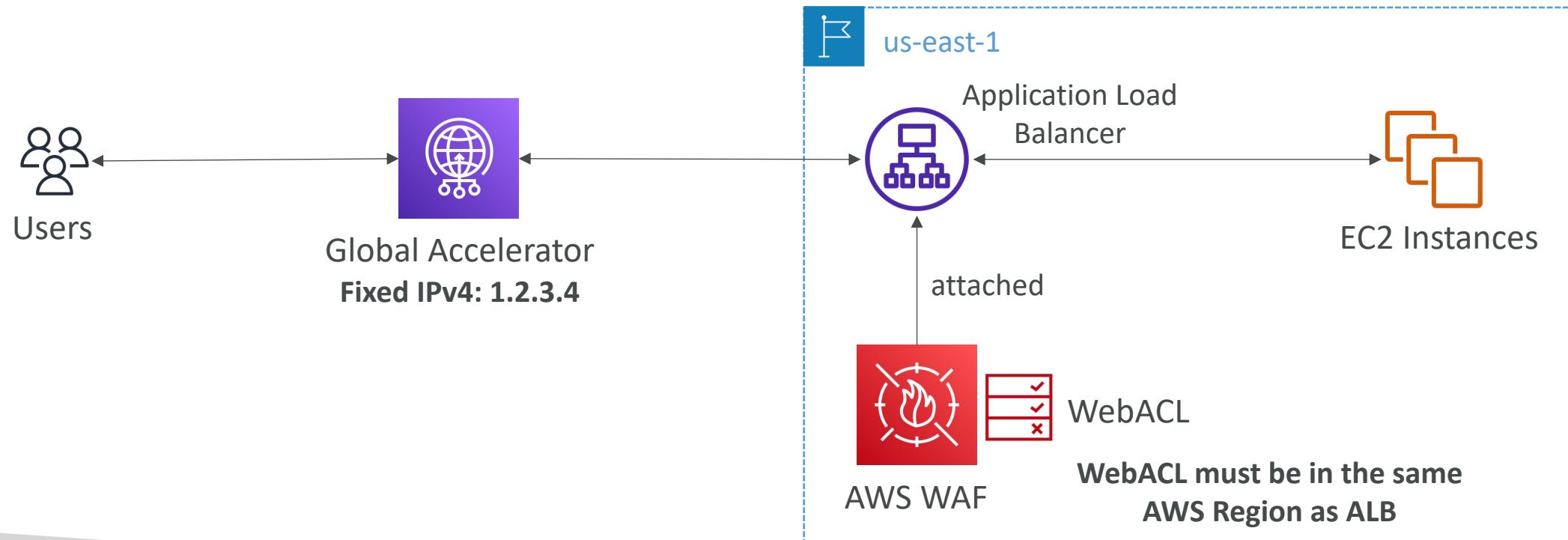
AWS WAF – Web Application Firewall



- Define Web ACL (Web Access Control List) Rules:
 - IP Set: up to 10,000 IP addresses – use multiple Rules for more IPs
 - HTTP headers, HTTP body, or URI strings Protects from common attack - SQL injection and Cross-Site Scripting (XSS)
 - Size constraints, geo-match (block countries)
 - Rate-based rules (to count occurrences of events) – for DDoS protection
- Web ACL are Regional except for CloudFront
- A rule group is a reusable set of rules that you can add to a web ACL

WAF – Fixed IP while using WAF with a Load Balancer

- WAF does not support the Network Load Balancer (Layer 4)
- We can use Global Accelerator for fixed IP and WAF on the ALB



AWS Shield: protect from DDoS attack



- DDoS: Distributed Denial of Service – many requests at the same time
- AWS Shield Standard:
 - Free service that is activated for every AWS customer
 - Provides protection from attacks such as SYN/UDP Floods, Reflection attacks and other layer 3/layer 4 attacks
- AWS Shield Advanced:
 - Optional DDoS mitigation service (\$3,000 per month per organization)
 - Protect against more sophisticated attack on [Amazon EC2](#), [Elastic Load Balancing \(ELB\)](#), [Amazon CloudFront](#), [AWS Global Accelerator](#), and [Route 53](#)
 - 24/7 access to AWS DDoS response team (DRP)
 - Protect against higher fees during usage spikes due to DDoS
 - Shield Advanced automatic application layer DDoS mitigation automatically creates, evaluates and deploys AWS WAF rules to mitigate layer 7 attacks

AWS Firewall Manager



- Manage rules in all accounts of an AWS Organization
- Security policy: common set of security rules
 - WAF rules (Application Load Balancer, API Gateways, CloudFront)
 - AWS Shield Advanced (ALB, CLB, NLB, Elastic IP, CloudFront)
 - Security Groups for EC2, Application Load Balancer and ENI resources in VPC
 - AWS Network Firewall (VPC Level)
 - Amazon Route 53 Resolver DNS Firewall
 - Policies are created at the region level
- Rules are applied to new resources as they are created (good for compliance) across all and future accounts in your Organization

WAF vs. Firewall Manager vs. Shield



AWS WAF



AWS Firewall Manager



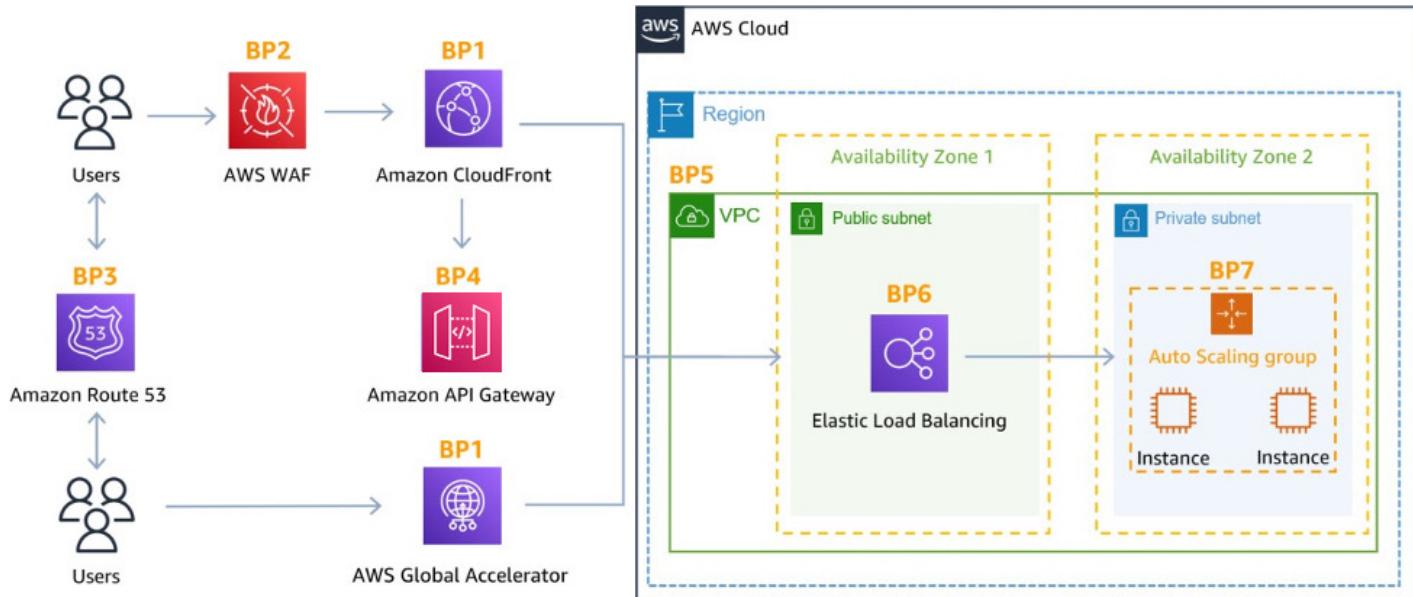
AWS Shield

- WAF, Shield and Firewall Manager are used together for comprehensive protection
- Define your Web ACL rules in WAF
- For granular protection of your resources, WAF alone is the correct choice
- If you want to use AWS WAF across accounts, accelerate WAF configuration, automate the protection of new resources, use Firewall Manager with AWS WAF
- Shield Advanced adds additional features on top of AWS WAF, such as dedicated support from the Shield Response Team (SRT) and advanced reporting.
- If you're prone to frequent DDoS attacks, consider purchasing Shield Advanced

AWS Best Practices for DDoS Resiliency

Edge Location Mitigation (BP1, BP3)

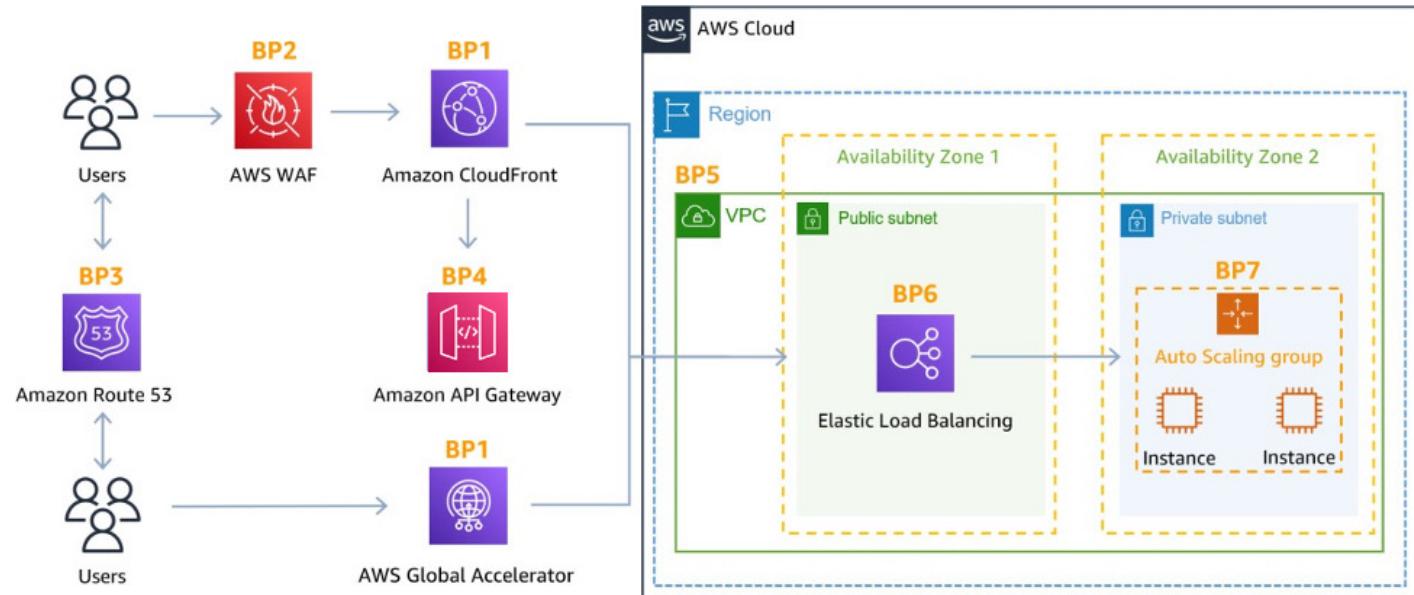
- **BP1 – CloudFront**
 - Web Application delivery at the edge
 - Protect from DDoS Common Attacks (SYN floods, UDP reflection...)
- **BP1 – Global Accelerator**
 - Access your application from the edge
 - Integration with Shield for DDoS protection
 - Helpful if your backend is not compatible with CloudFront
- **BP3 – Route 53**
 - Domain Name Resolution at the edge
 - DDoS Protection mechanism



AWS Best Practices for DDoS Resiliency

Best practices for DDoS mitigation

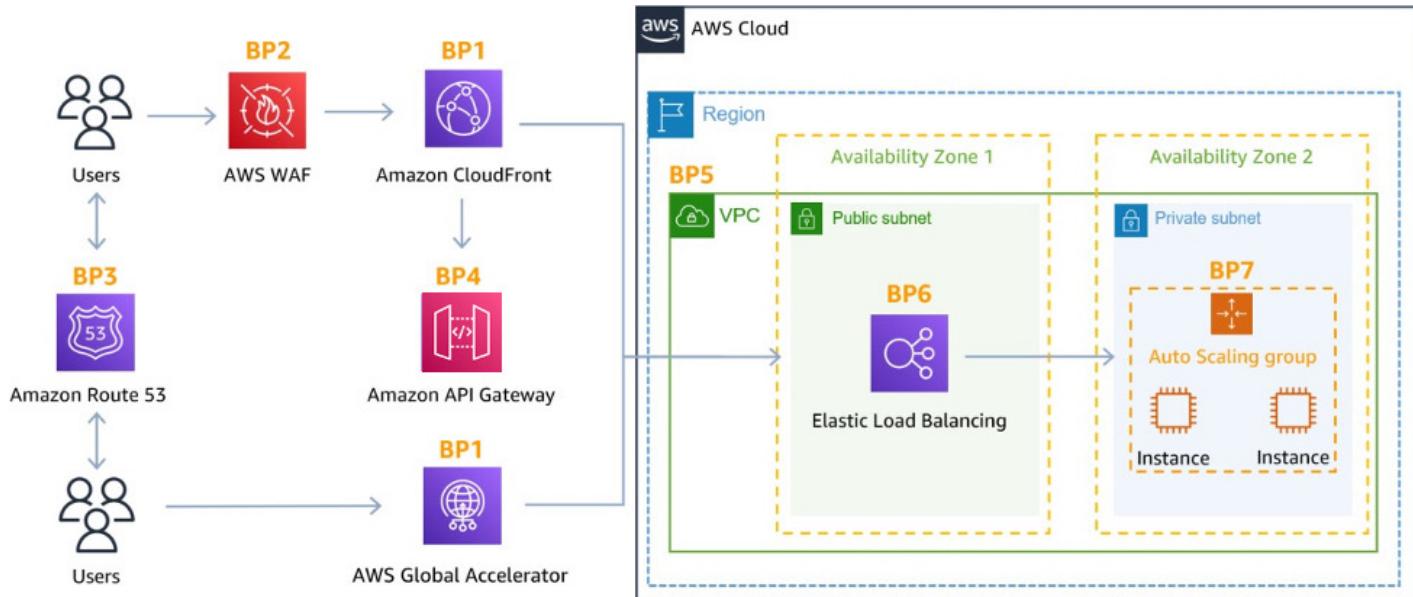
- Infrastructure layer defense (BP1, BP3, BP6)
 - Protect Amazon EC2 against high traffic
 - That includes using Global Accelerator, Route 53, CloudFront, Elastic Load Balancing
- Amazon EC2 with Auto Scaling (BP7)
 - Helps scale in case of sudden traffic surges including a flash crowd or a DDoS attack
- Elastic Load Balancing (BP6)
 - Elastic Load Balancing scales with the traffic increases and will distribute the traffic to many EC2 instances



AWS Best Practices for DDoS Resiliency

Application Layer Defense

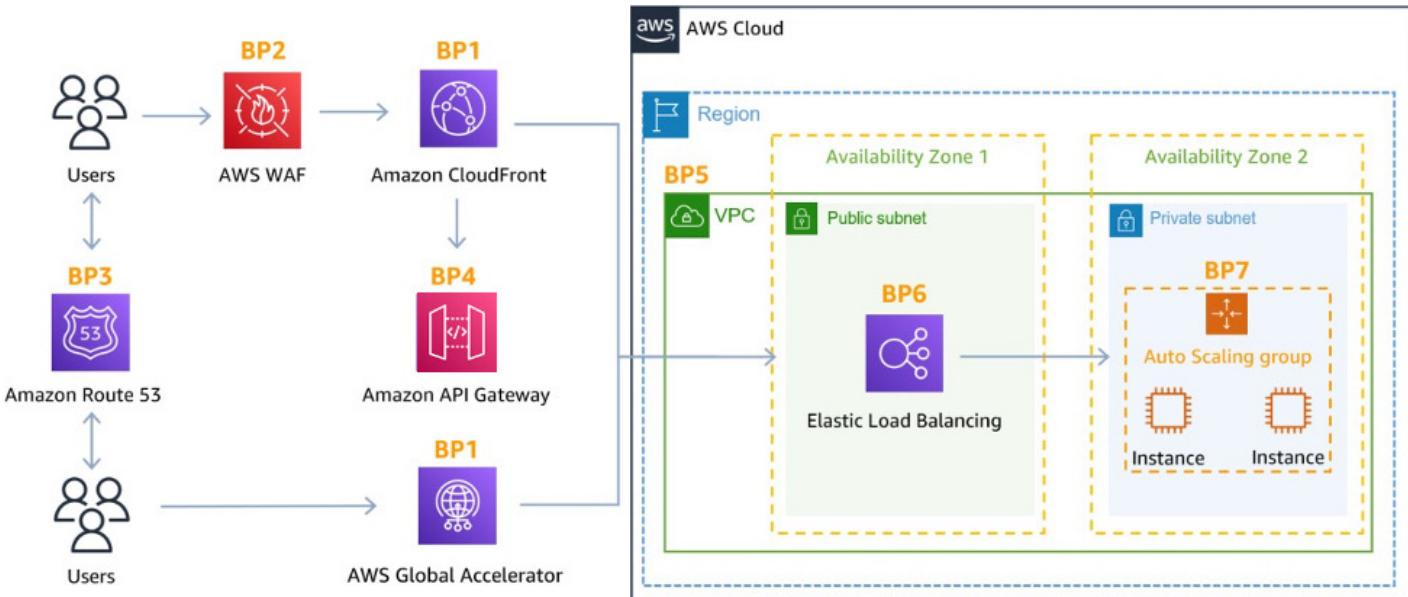
- Detect and filter malicious web requests (BP1, BP2)
 - CloudFront cache static content and serve it from edge locations, protecting your backend
 - AWS WAF is used on top of CloudFront and Application Load Balancer to filter and block requests based on request signatures
 - WAF rate-based rules can automatically block the IPs of bad actors
 - Use managed rules on WAF to block attacks based on IP reputation, or block anonymous IPs
 - CloudFront can block specific geographies
- Shield Advanced (BP1, BP2, BP6)
 - Shield Advanced automatic application layer DDoS mitigation automatically creates, evaluates and deploys AWS WAF rules to mitigate layer 7 attacks



AWS Best Practices for DDoS Resiliency

Attack surface reduction

- Obfuscating AWS resources (BP1, BP4, BP6)
 - Using CloudFront, API Gateway, Elastic Load Balancing to hide your backend resources (Lambda functions, EC2 instances)
- Security groups and Network ACLs (BP5)
 - Use security groups and NACLs to filter traffic based on specific IP at the subnet or ENI-level
 - Elastic IP are protected by AWS Shield Advanced
- Protecting API endpoints (BP4)
 - Hide EC2, Lambda, elsewhere
 - Edge-optimized mode, or CloudFront + regional mode (more control for DDoS)
 - WAF + API Gateway: burst limits, headers filtering, use API keys

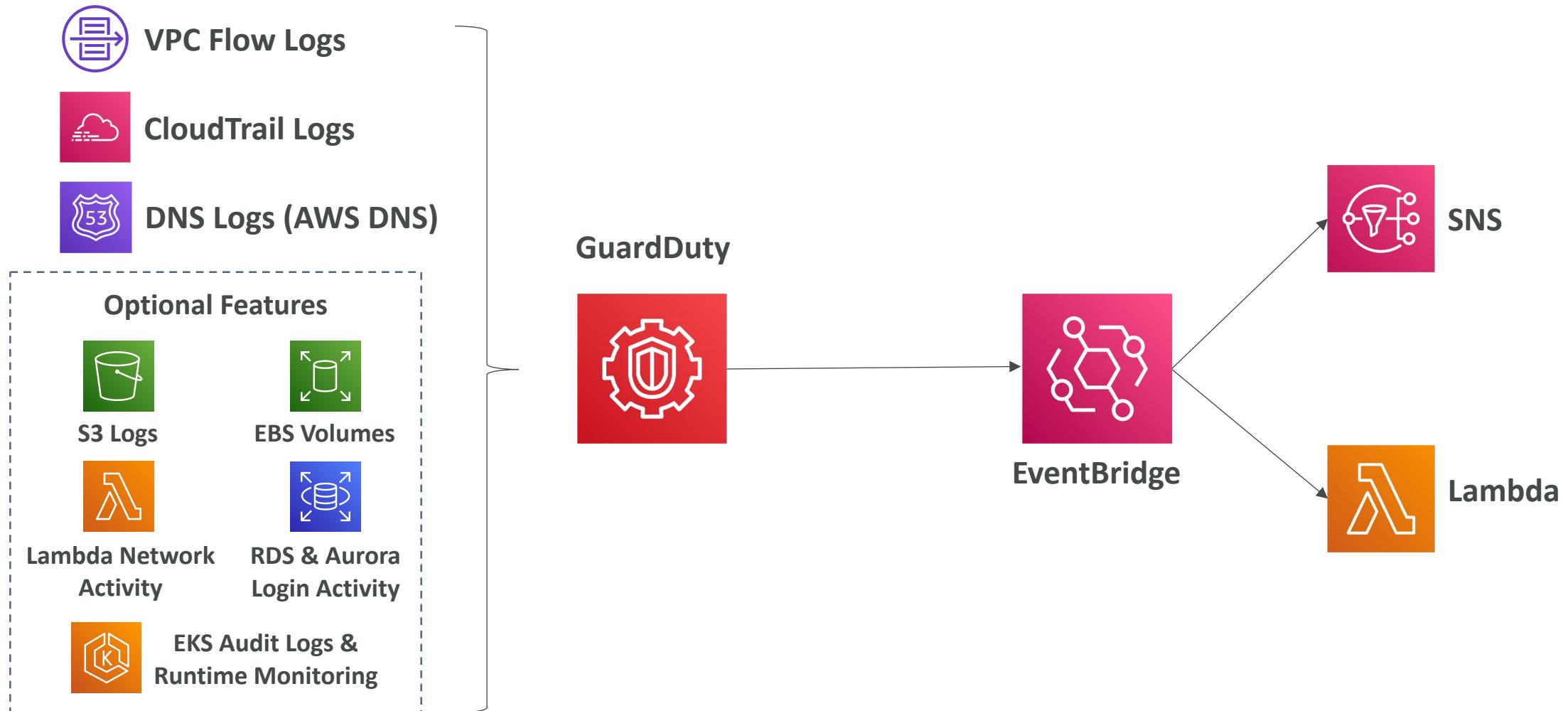




Amazon GuardDuty

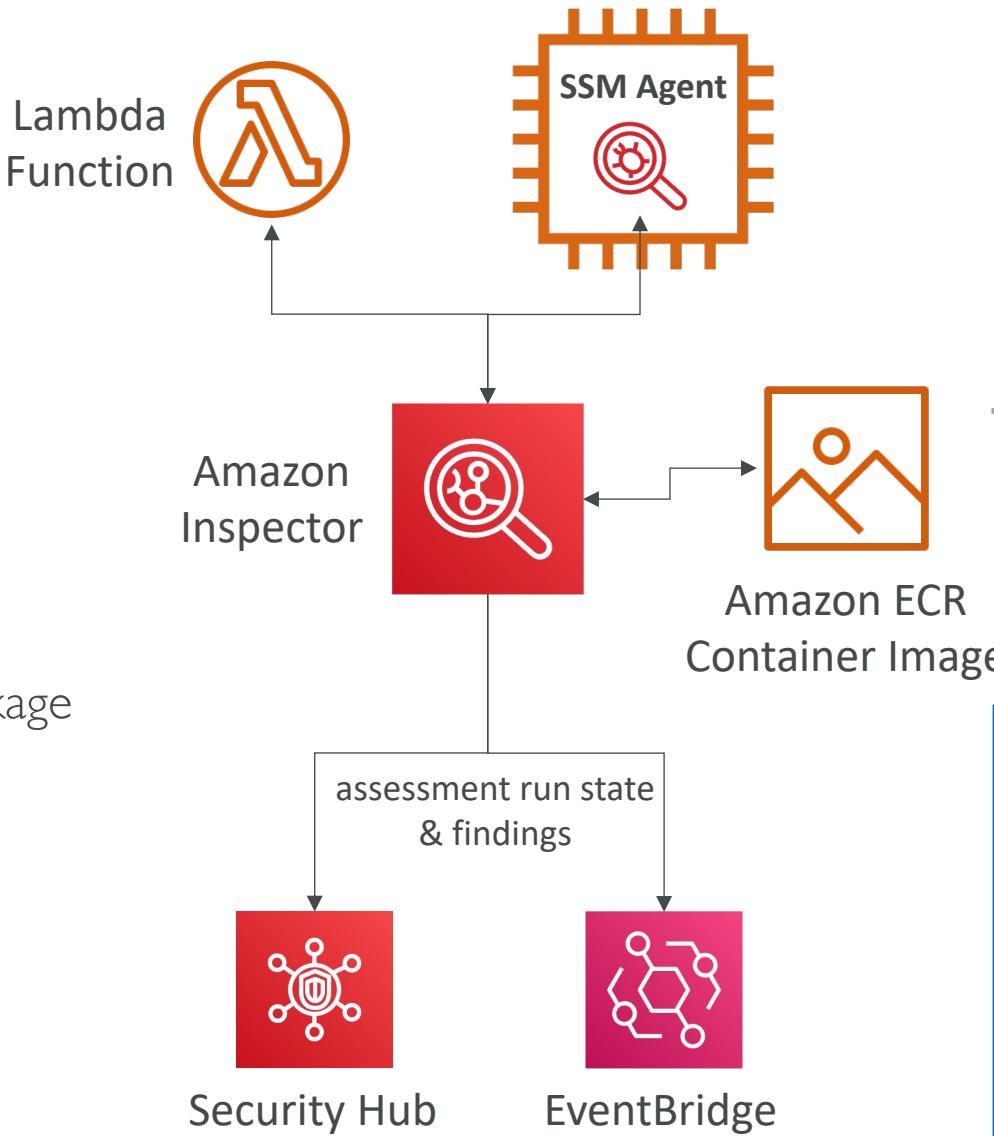
- Intelligent Threat discovery to protect your AWS Account
- Uses Machine Learning algorithms, anomaly detection, 3rd party data
- One click to enable (30 days trial), no need to install software
- Input data includes:
 - CloudTrail Events Logs – unusual API calls, unauthorized deployments
 - CloudTrail Management Events – create VPC subnet, create trail, ...
 - CloudTrail S3 Data Events – get object, list objects, delete object, ...
 - VPC Flow Logs – unusual internal traffic, unusual IP address
 - DNS Logs – compromised EC2 instances sending encoded data within DNS queries
 - Optional Features – EKS Audit Logs, RDS & Aurora, EBS, Lambda, S3 Data Events...
- Can setup **EventBridge rules** to be notified in case of findings
- EventBridge rules can target AWS Lambda or SNS
- Can protect against CryptoCurrency attacks (has a dedicated “finding” for it)

Amazon GuardDuty



Amazon Inspector

- Automated Security Assessments
- For EC2 instances
 - Leveraging the AWS System Manager (SSM) agent
 - Analyze against unintended network accessibility
 - Analyze the running OS against known vulnerabilities
- For Container Images push to Amazon ECR
 - Assessment of Container Images as they are pushed
- For Lambda Functions
 - Identifies software vulnerabilities in function code and package dependencies
 - Assessment of functions as they are deployed
- Reporting & integration with AWS Security Hub
- Send findings to Amazon Event Bridge



What does Amazon Inspector evaluate?

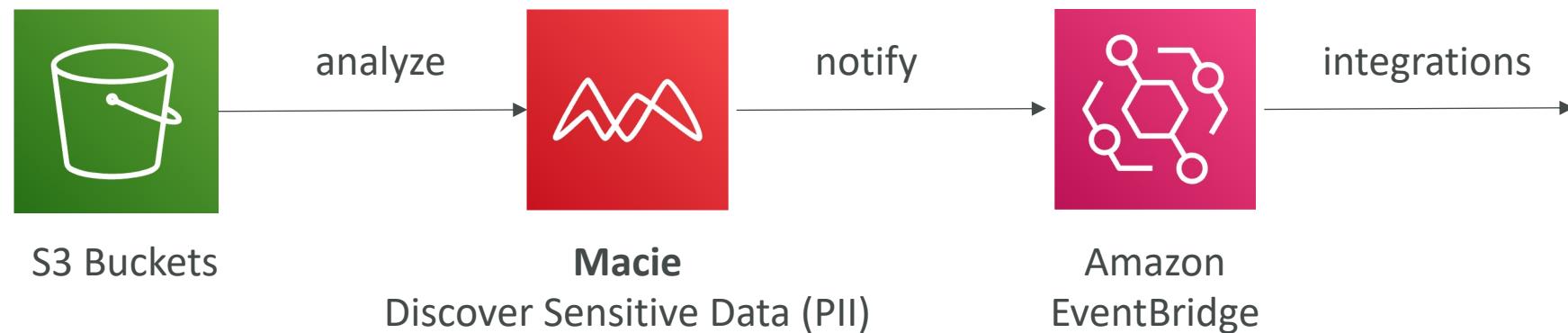


- Remember: only for EC2 instances, Container Images & Lambda functions
- Continuous scanning of the infrastructure, only when needed
- Package vulnerabilities (EC2, ECR & Lambda) – database of CVE
- Network reachability (EC2)
- A risk score is associated with all vulnerabilities for prioritization

AWS Macie

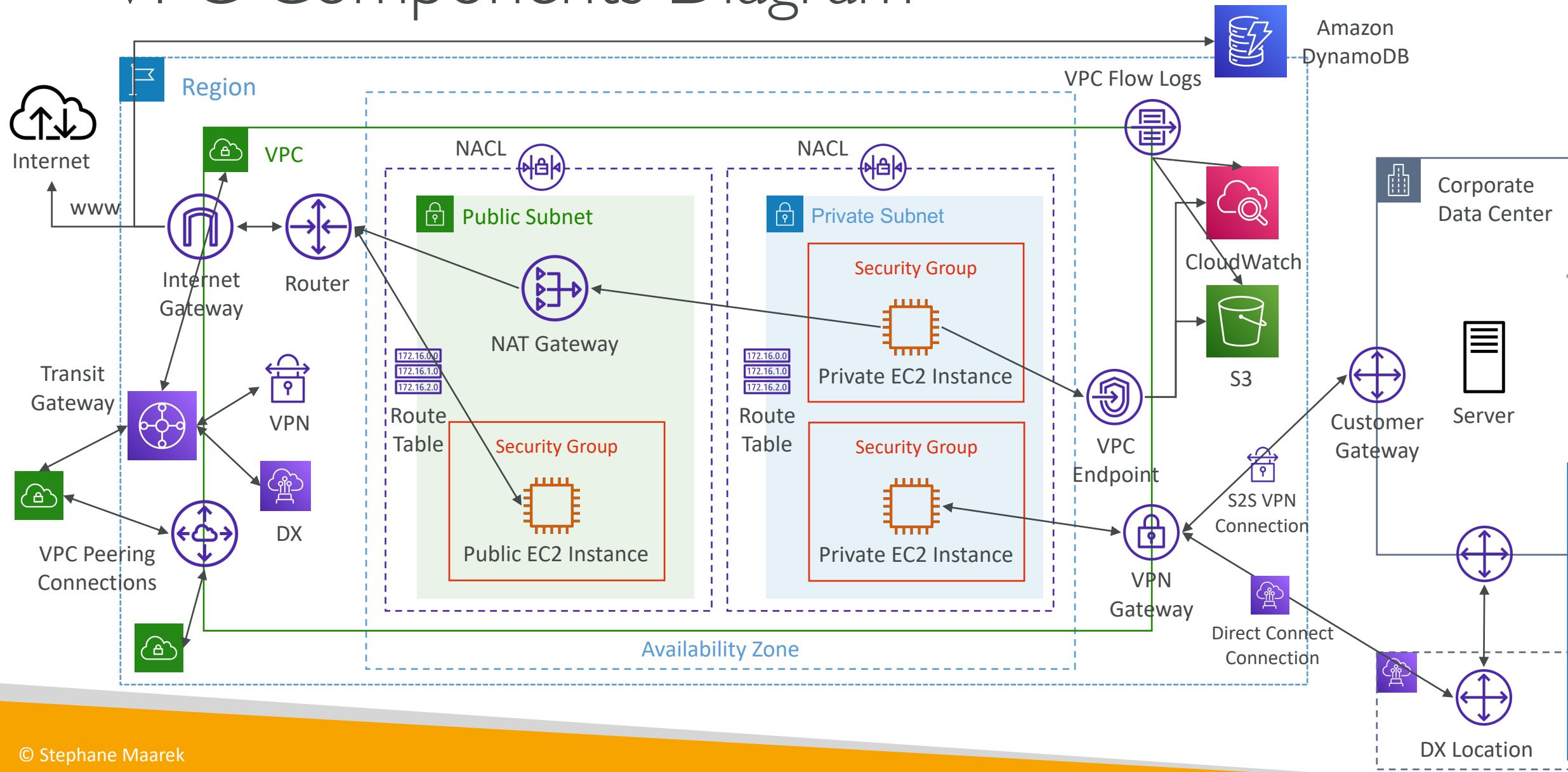


- Amazon Macie is a fully managed data security and data privacy service that uses machine learning and pattern matching to discover and protect your sensitive data in AWS.
- Macie helps identify and alert you to sensitive data, such as personally identifiable information (PII)



Virtual Private Cloud (VPC)

VPC Components Diagram



Understanding CIDR – IPv4

- Classless Inter-Domain Routing – a method for allocating IP addresses
- Used in **Security Groups** rules and AWS networking in general

IP version	Type	Protocol	Port range	Source	Description
IPv4	SSH	TCP	22	122.149.196.85/32	-
IPv4	HTTP	TCP	80	0.0.0.0/0	-

- They help to define an IP address range:
 - We've seen WW.XX.YY.ZZ/32 => one IP
 - We've seen 0.0.0.0/0 => all IPs
 - But we can define: 192.168.0.0/26 => 192.168.0.0 – 192.168.0.63 (64 IP addresses)

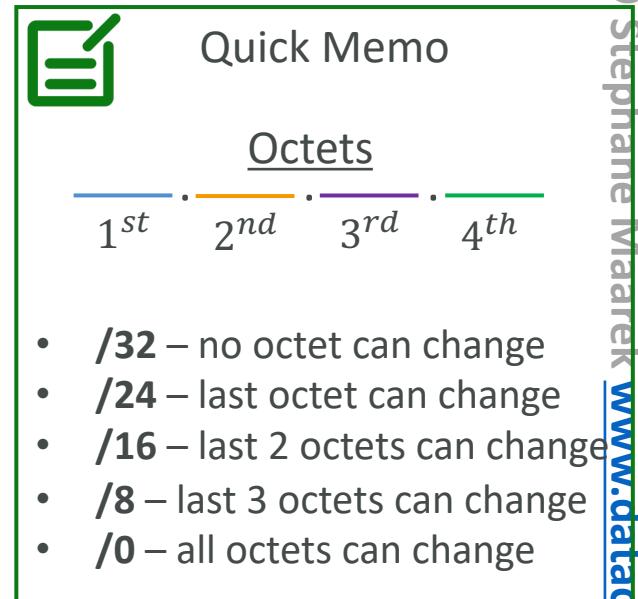
Understanding CIDR – IPv4

- A CIDR consists of two components
- Base IP
 - Represents an IP contained in the range (XX.XX.XX.XX)
 - Example: 10.0.0.0, 192.168.0.0, ...
- Subnet Mask
 - Defines how many bits can change in the IP
 - Example: /0, /24, /32
 - Can take two forms:
 - /8 \Leftrightarrow 255.0.0.0
 - /16 \Leftrightarrow 255.255.0.0
 - /24 \Leftrightarrow 255.255.255.0
 - /32 \Leftrightarrow 255.255.255.255

Understanding CIDR – Subnet Mask

- The Subnet Mask basically allows part of the underlying IP to get additional next values from the base IP

192	. 168 . 0 . 0 /32 => allows for 1 IP (2^0)	→ 192.168.0.0
192	. 168 . 0 . 0 /31 => allows for 2 IP (2^1)	→ 192.168.0.0 -> 192.168.0.1
192	. 168 . 0 . 0 /30 => allows for 4 IP (2^2)	→ 192.168.0.0 -> 192.168.0.3
192	. 168 . 0 . 0 /29 => allows for 8 IP (2^3)	→ 192.168.0.0 -> 192.168.0.7
192	. 168 . 0 . 0 /28 => allows for 16 IP (2^4)	→ 192.168.0.0 -> 192.168.0.15
192	. 168 . 0 . 0 /27 => allows for 32 IP (2^5)	→ 192.168.0.0 -> 192.168.0.31
192	. 168 . 0 . 0 /26 => allows for 64 IP (2^6)	→ 192.168.0.0 -> 192.168.0.63
192	. 168 . 0 . 0 /25 => allows for 128 IP (2^7)	→ 192.168.0.0 -> 192.168.0.127
192	. 168 . 0 . 0 /24 => allows for 256 IP (2^8)	→ 192.168.0.0 -> 192.168.0.255
...		
192	. 168 . 0 . 0 /16 => allows for 65,536 IP (2^{16})	→ 192.168.0.0 -> 192.168.255.255
...		
192	. 168 . 0 . 0 /0 => allows for All IPs	→ 0.0.0.0 -> 255.255.255.255



Understanding CIDR – Little Exercise

- $192.168.0.0/24 = \dots ?$
 - $192.168.0.0 - 192.168.0.255$ (256 IPs)
- $192.168.0.0/16 = \dots ?$
 - $192.168.0.0 - 192.168.255.255$ (65,536 IPs)
- $134.56.78.123/32 = \dots ?$
 - Just $134.56.78.123$
- $0.0.0.0/0$
 - All IPs!
- When in doubt, use this website <https://www.ipaddressguide.com/cidr>

Public vs. Private IP (IPv4)

- The Internet Assigned Numbers Authority (IANA) established certain blocks of IPv4 addresses for the use of private (LAN) and public (Internet) addresses
- **Private IP** can only allow certain values:
 - 10.0.0 – 10.255.255.255 (10.0.0/8) ↵ in big networks
 - 172.16.0.0 – 172.31.255.255 (172.16.0.0/12) ↵ AWS default VPC in that range
 - 192.168.0.0 – 192.168.255.255 (192.168.0.0/16) ↵ e.g., home networks
- All the rest of the IP addresses on the Internet are Public

Default VPC Walkthrough

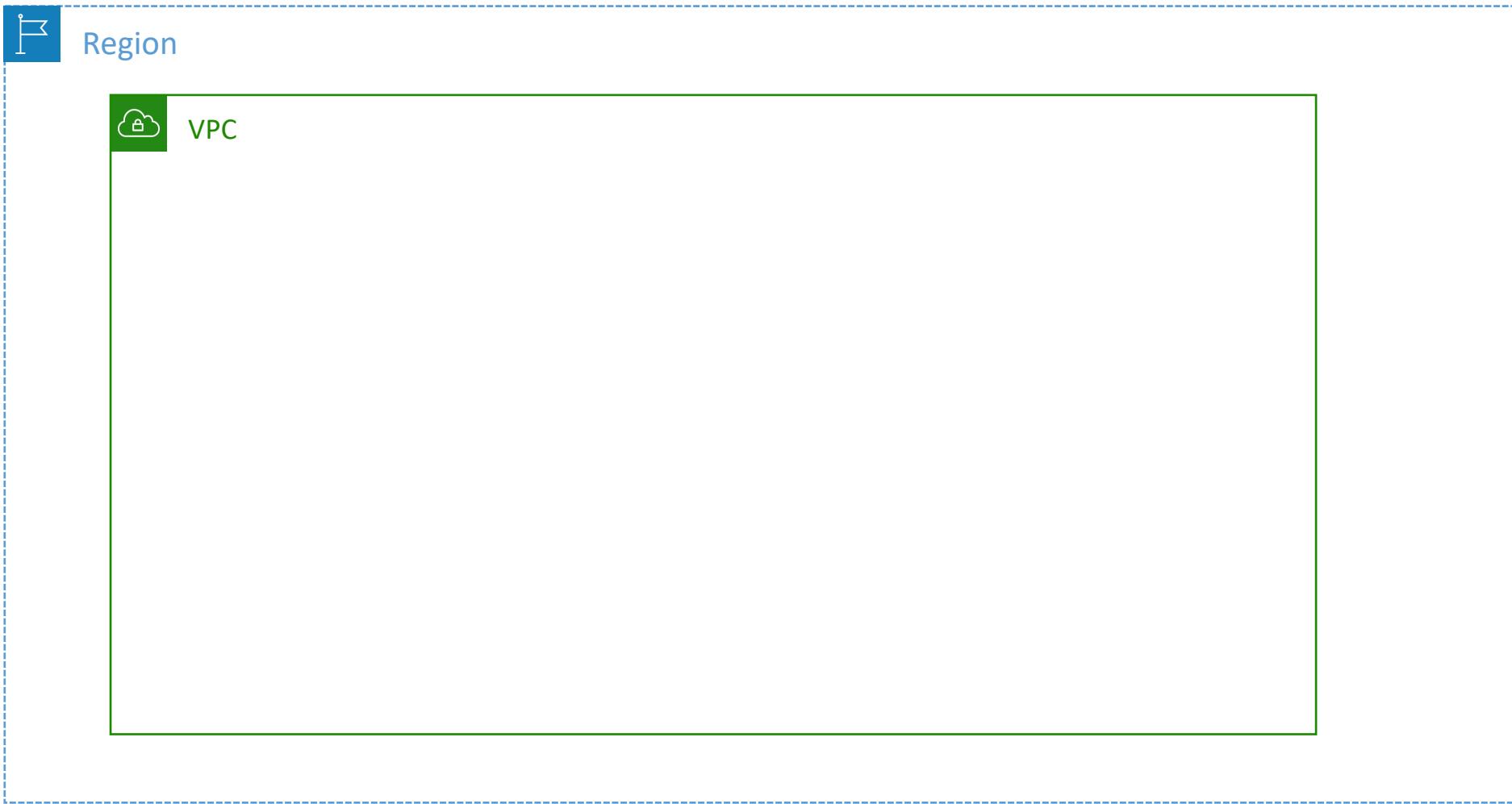
- All new AWS accounts have a default VPC
- New EC2 instances are launched into the default VPC if no subnet is specified
- Default VPC has Internet connectivity and all EC2 instances inside it have public IPv4 addresses
- We also get a public and a private IPv4 DNS names

VPC in AWS – IPv4

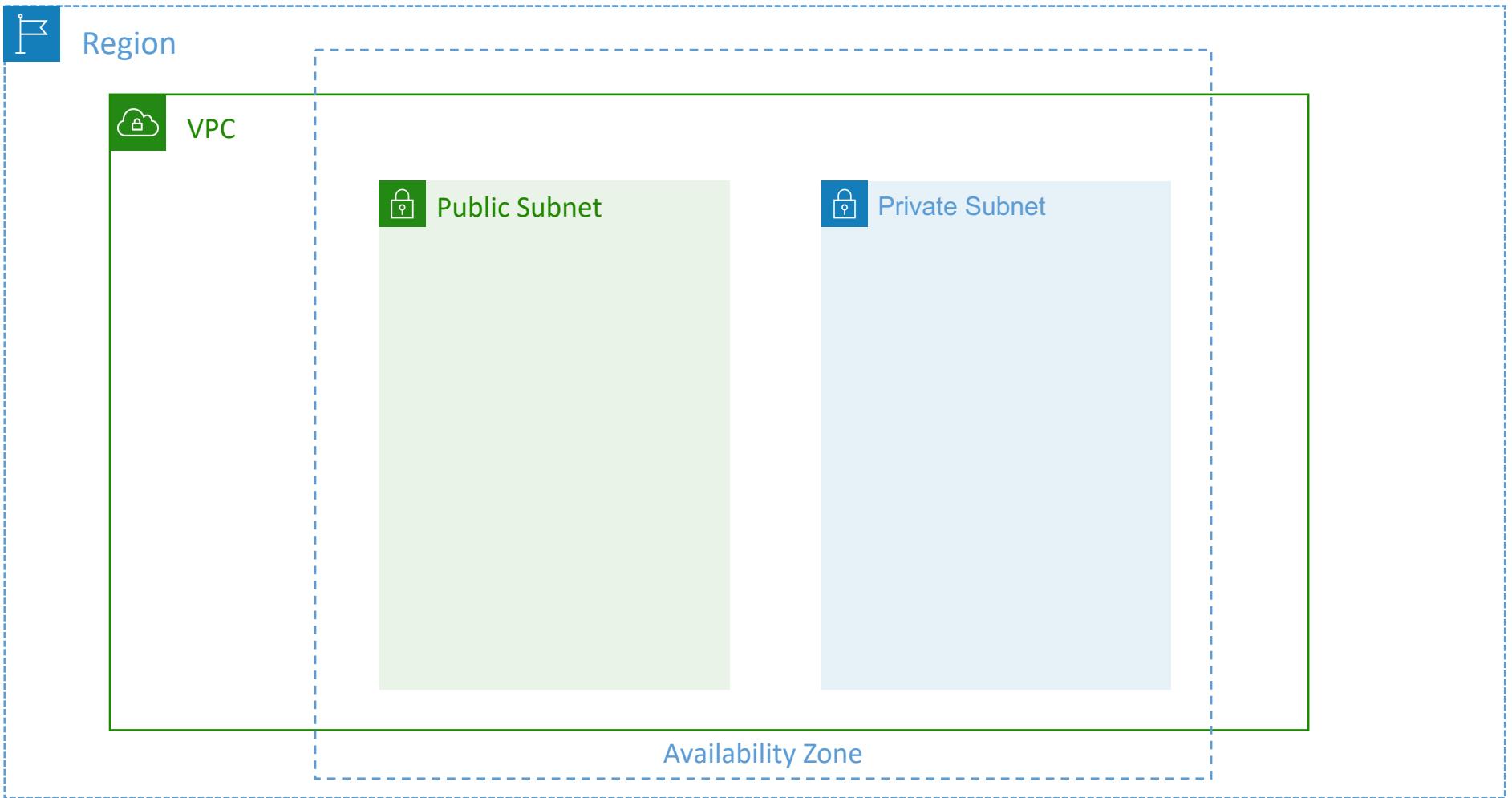


- VPC = Virtual Private Cloud
- You can have multiple VPCs in an AWS region (max. 5 per region – soft limit)
- Max. CIDR per VPC is 5, for each CIDR:
 - Min. size is /28 (16 IP addresses)
 - Max. size is /16 (65536 IP addresses)
- Because VPC is private, only the Private IPv4 ranges are allowed:
 - 10.0.0.0 – 10.255.255.255 (10.0.0.0/8)
 - 172.16.0.0 – 172.31.255.255 (172.16.0.0/12)
 - 192.168.0.0 – 192.168.255.255 (192.168.0.0/16)
- Your VPC CIDR should NOT overlap with your other networks (e.g., corporate)

State of Hands-on



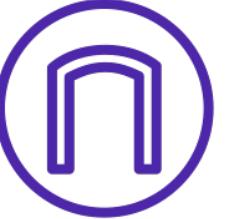
Adding Subnets



VPC – Subnet (IPv4)



- AWS reserves **5 IP addresses (first 4 & last 1)** in each subnet
- These 5 IP addresses are not available for use and can't be assigned to an EC2 instance
- Example: if CIDR block 10.0.0.0/24, then reserved IP addresses are:
 - 10.0.0.0 – Network Address
 - 10.0.0.1 – reserved by AWS for the VPC router
 - 10.0.0.2 – reserved by AWS for mapping to Amazon-provided DNS
 - 10.0.0.3 – reserved by AWS for future use
 - 10.0.0.255 – Network Broadcast Address. AWS does not support broadcast in a VPC, therefore the address is reserved
- **Exam Tip**, if you need 29 IP addresses for EC2 instances:
 - You can't choose a subnet of size /27 (32 IP addresses, $32 - 5 = 27 < 29$)
 - You need to choose a subnet of size /26 (64 IP addresses, $64 - 5 = 59 > 29$)

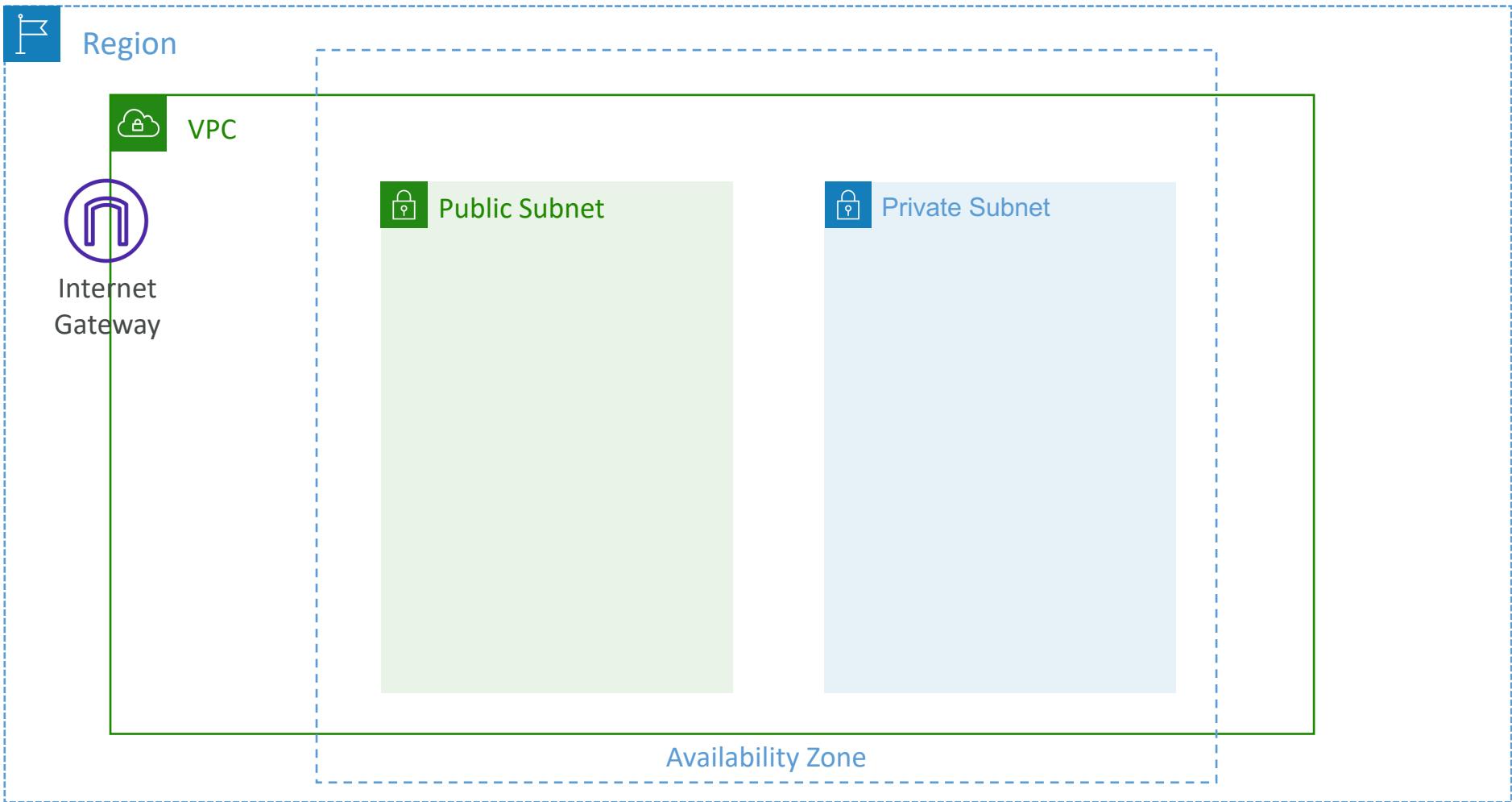


Internet Gateway (IGW)

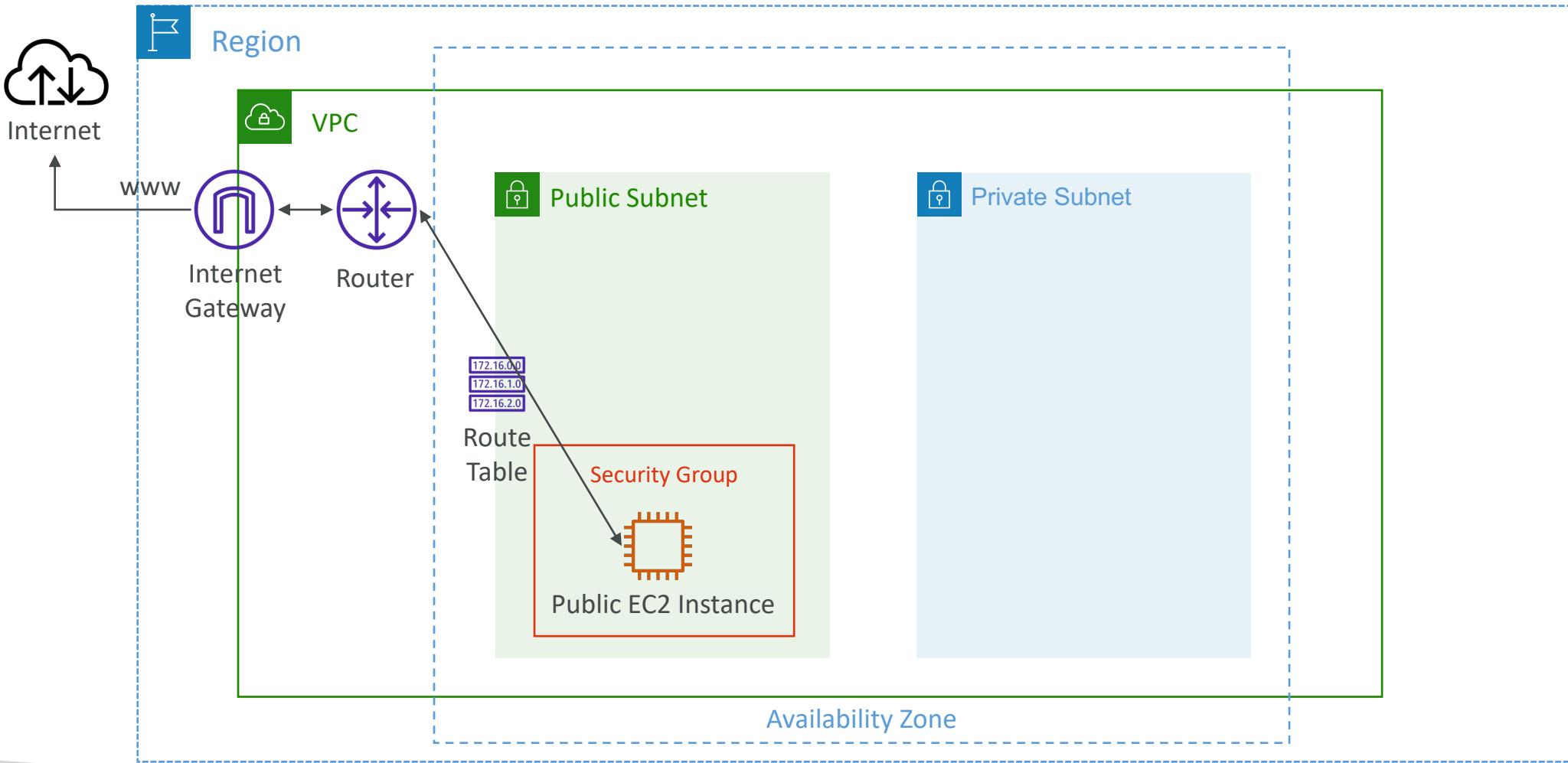
- Allows resources (e.g., EC2 instances) in a VPC connect to the Internet
- It scales horizontally and is highly available and redundant
- Must be created separately from a VPC
- One VPC can only be attached to one IGW and vice versa

- Internet Gateways on their own do not allow Internet access...
- Route tables must also be edited!

Adding Internet Gateway

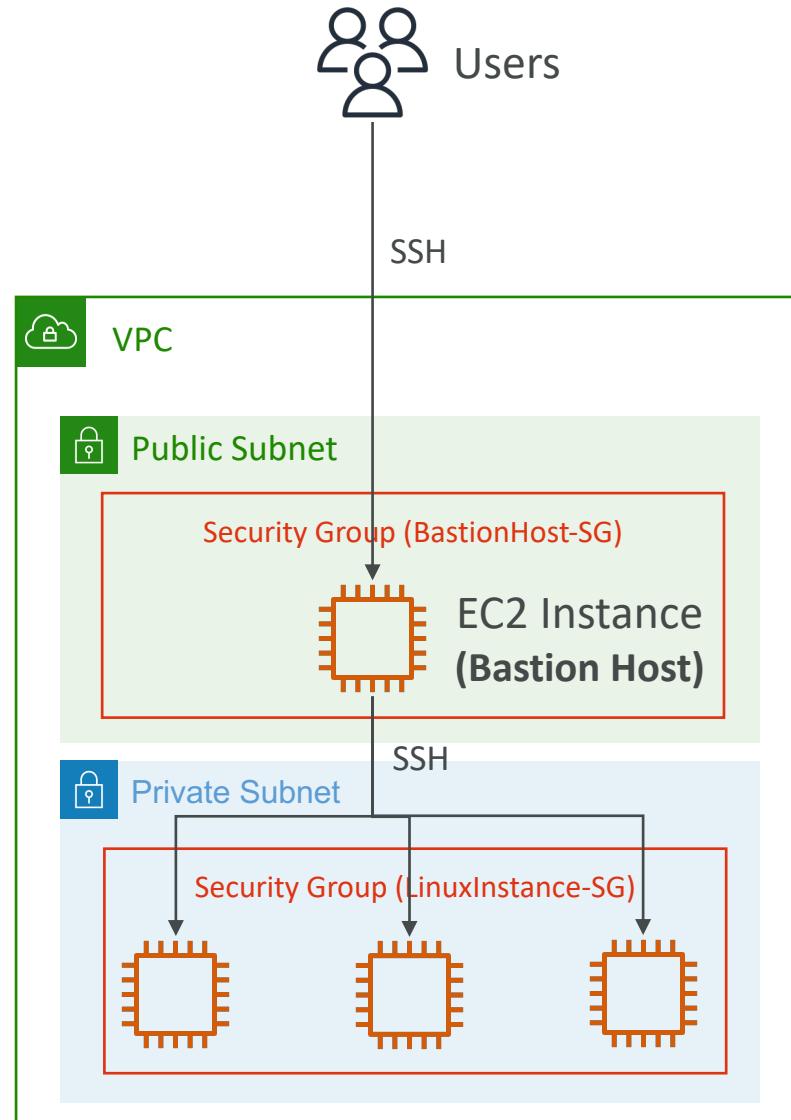


Editing Route Tables



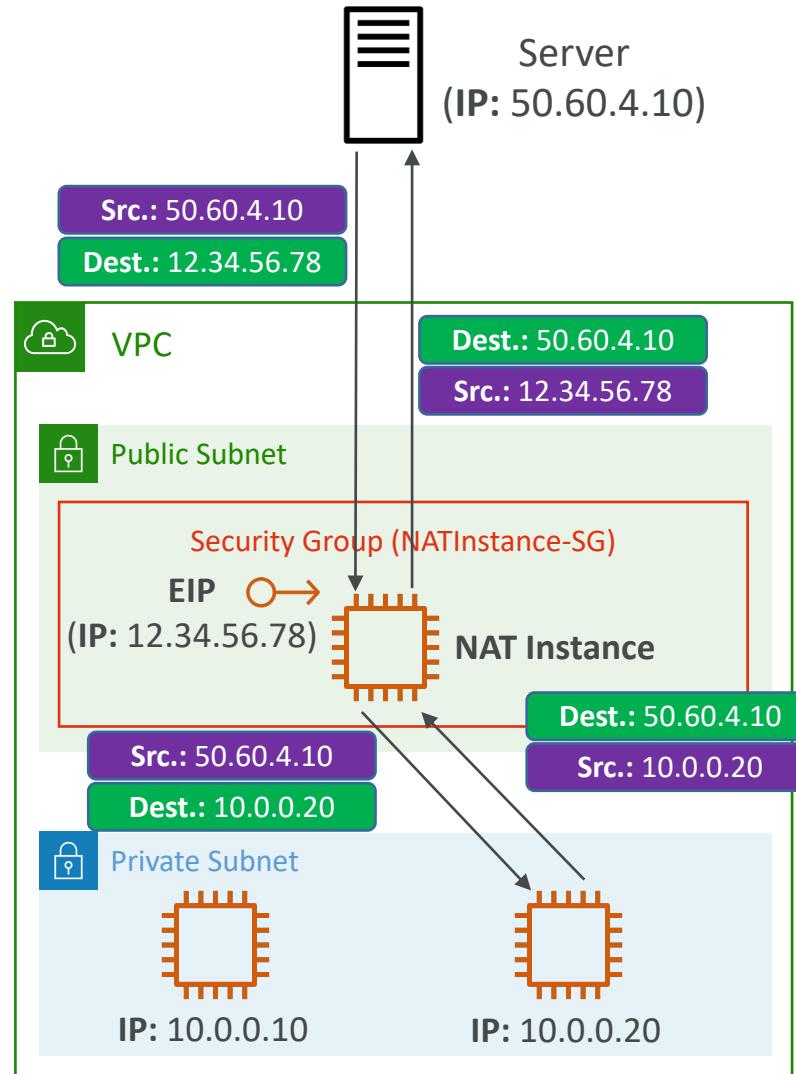
Bastion Hosts

- We can use a Bastion Host to SSH into our private EC2 instances
- The bastion is in the public subnet which is then connected to all other private subnets
- **Bastion Host security group must allow** inbound from the internet on port 22 from restricted CIDR, for example the public CIDR of your corporation
- **Security Group of the EC2 Instances** must allow the Security Group of the Bastion Host, or the private IP of the Bastion host

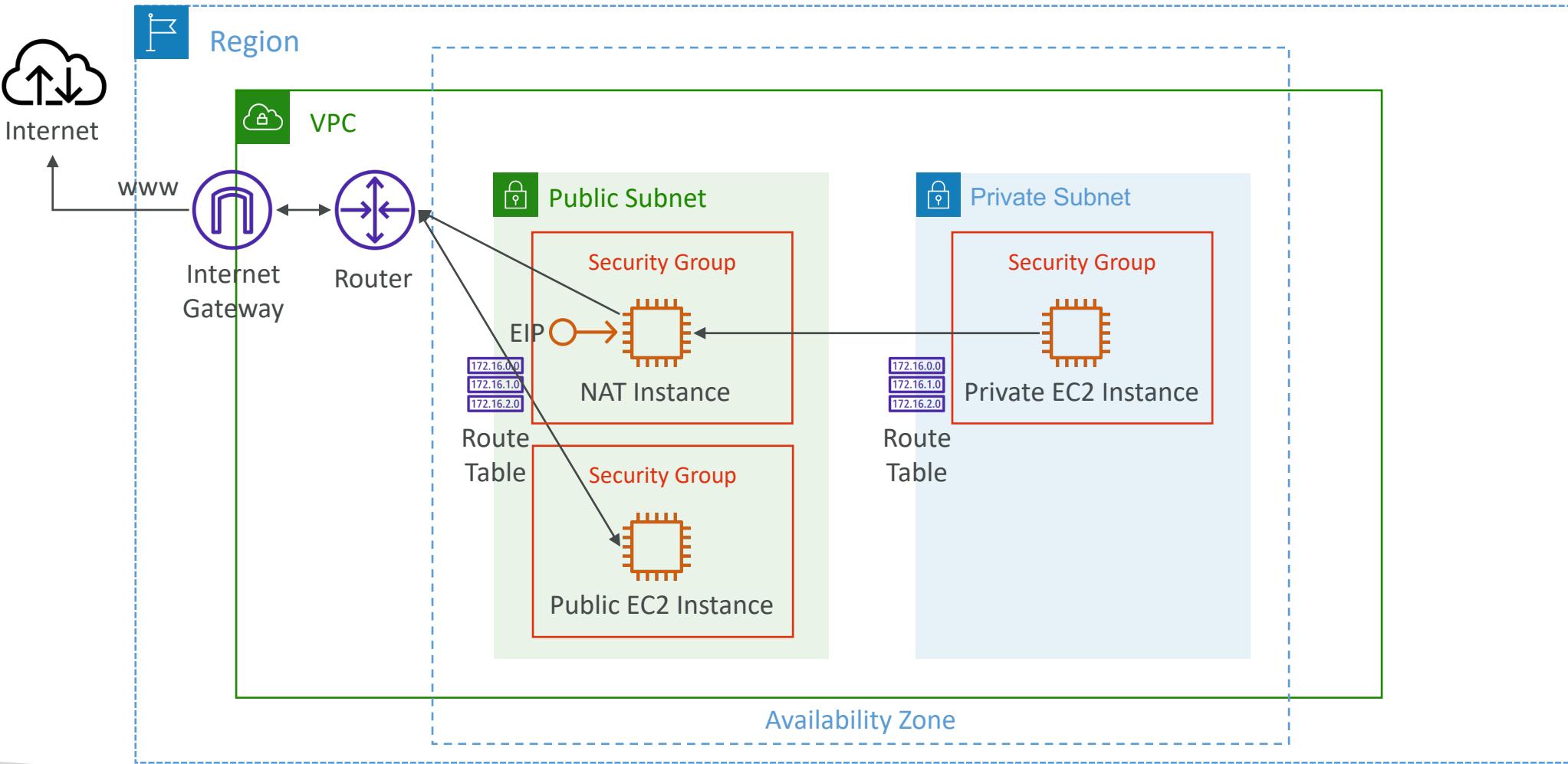


NAT Instance (outdated, but still at the exam)

- NAT = Network Address Translation
- Allows EC2 instances in private subnets to connect to the Internet
- Must be launched in a public subnet
- Must disable EC2 setting: **Source / destination Check**
- Must have Elastic IP attached to it
- Route Tables must be configured to route traffic from private subnets to the NAT Instance

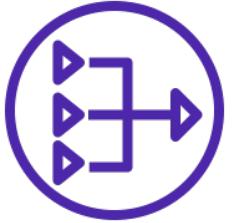


NAT Instance



NAT Instance – Comments

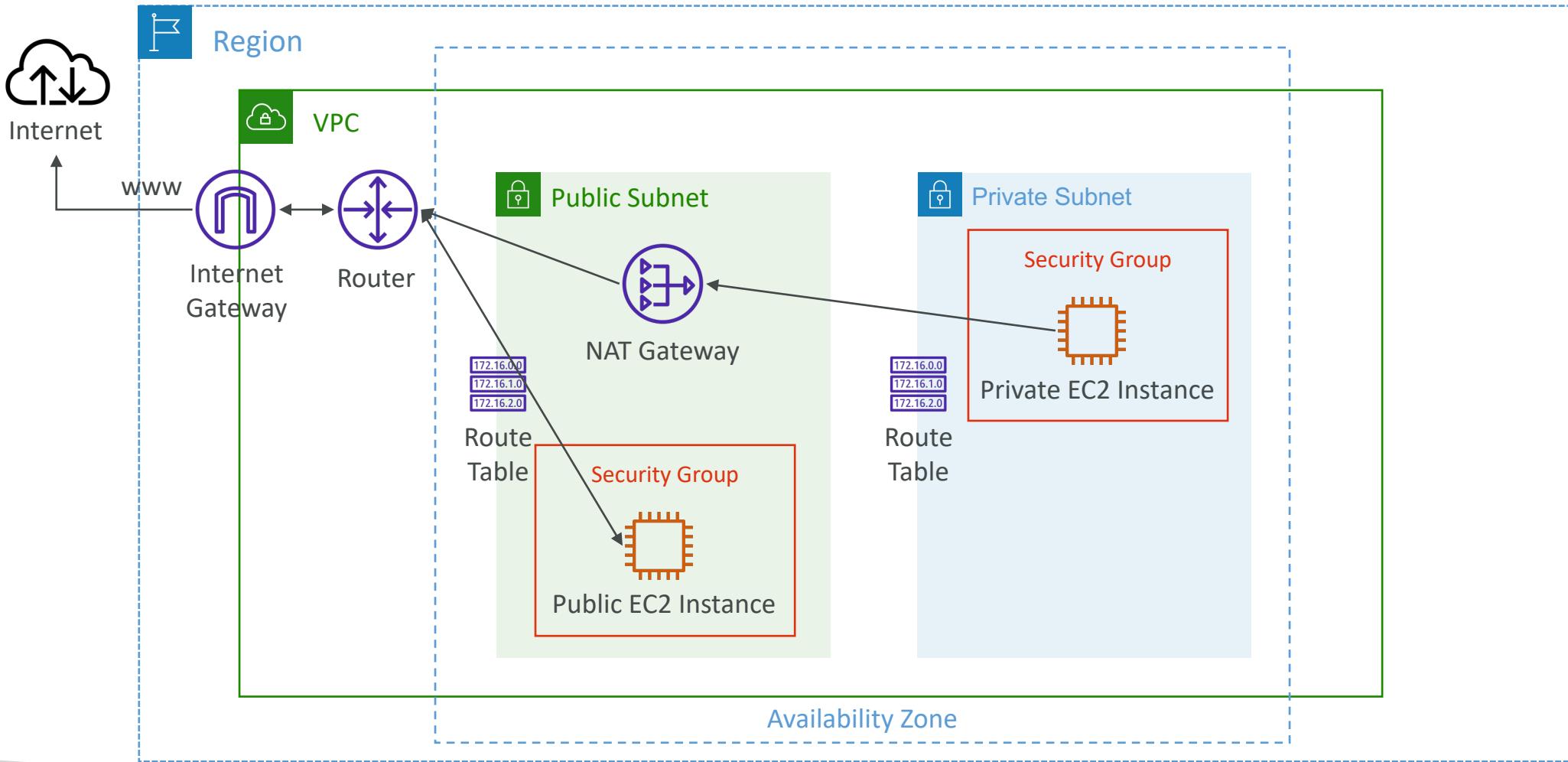
- Pre-configured Amazon Linux AMI is available
 - Reached the end of standard support on December 31, 2020
- Not highly available / resilient setup out of the box
 - You need to create an ASG in multi-AZ + resilient user-data script
- Internet traffic bandwidth depends on EC2 instance type
- You must manage Security Groups & rules:
 - Inbound:
 - Allow HTTP / HTTPS traffic coming from Private Subnets
 - Allow SSH from your home network (access is provided through Internet Gateway)
 - Outbound:
 - Allow HTTP / HTTPS traffic to the Internet



NAT Gateway

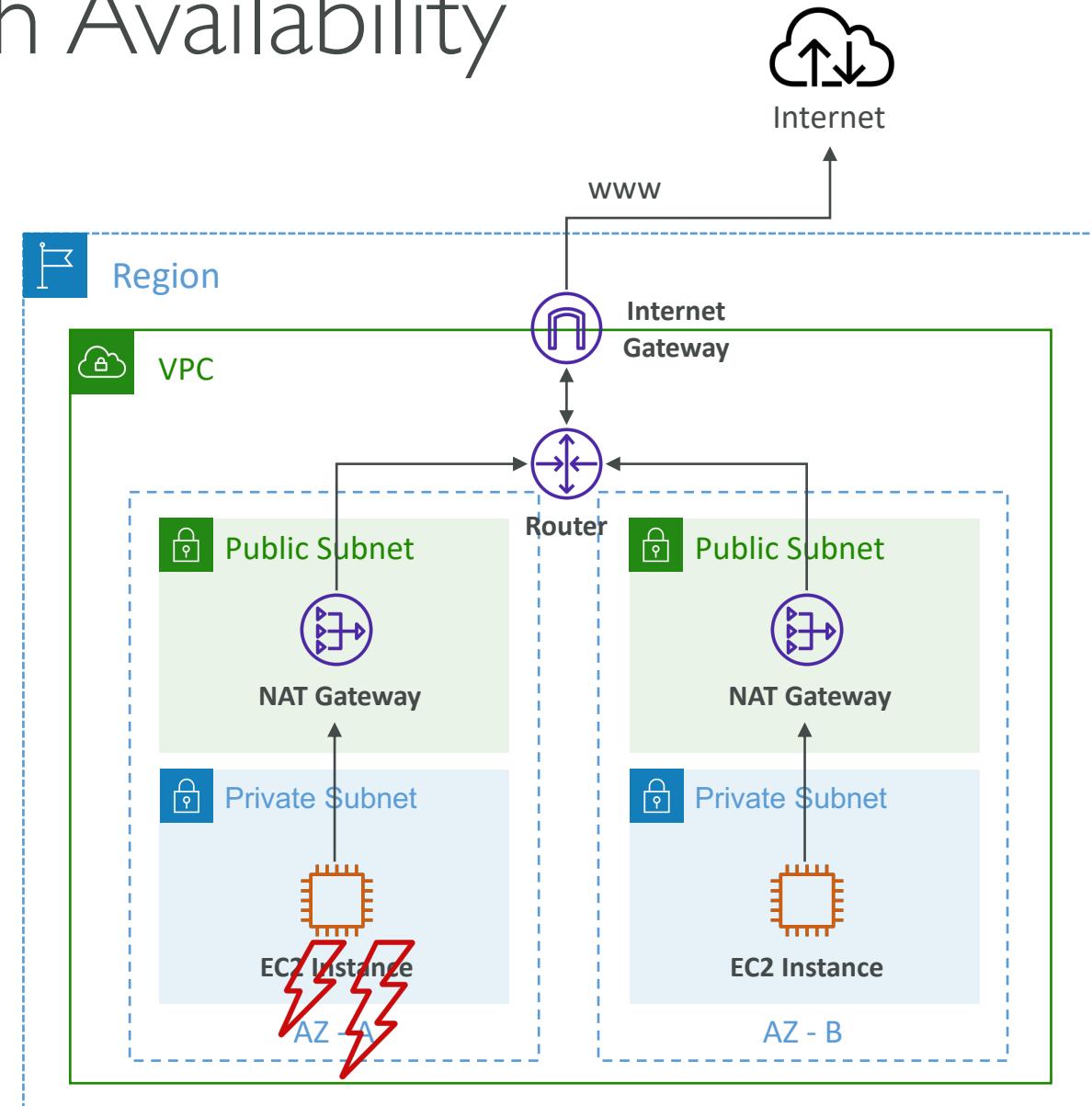
- AWS-managed NAT, higher bandwidth, high availability, no administration
- Pay per hour for usage and bandwidth
- NATGW is created in a specific Availability Zone, uses an Elastic IP
- Can't be used by EC2 instance in the same subnet (only from other subnets)
- Requires an IGW (Private Subnet => NATGW => IGW)
- 5 Gbps of bandwidth with automatic scaling up to 45 Gbps
- No Security Groups to manage / required

NAT Gateway



NAT Gateway with High Availability

- NAT Gateway is resilient within a single Availability Zone
- Must create multiple NAT Gateways in multiple AZs for fault-tolerance
- There is no cross-AZ failover needed because if an AZ goes down it doesn't need NAT



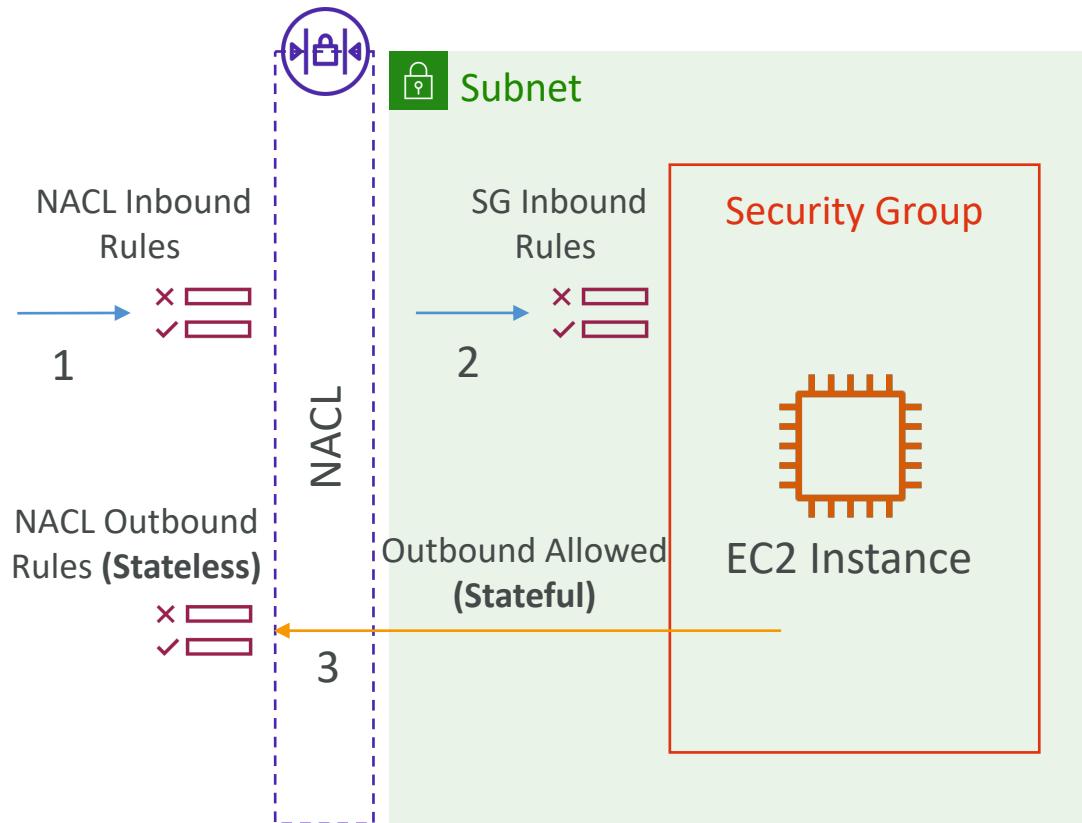
NAT Gateway vs. NAT Instance

	NAT Gateway	NAT Instance
Availability	Highly available within AZ (create in another AZ)	Use a script to manage failover between instances
Bandwidth	Up to 45 Gbps	Depends on EC2 instance type
Maintenance	Managed by AWS	Managed by you (e.g., software, OS patches, ...)
Cost	Per hour & amount of data transferred	Per hour, EC2 instance type and size, + network \$
Public IPv4	✓	✓
Private IPv4	✓	✓
Security Groups	✗	✓
Use as Bastion Host?	✗	✓

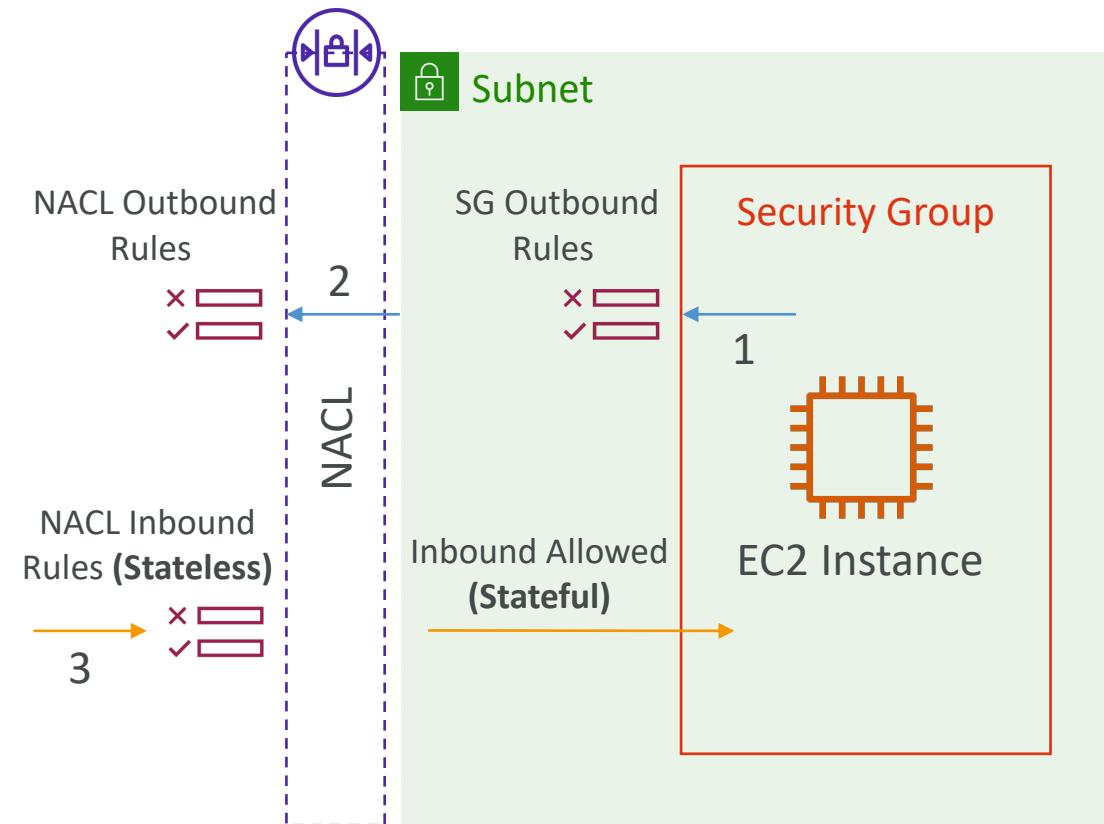
More at: <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-comparison.html>

Security Groups & NACLs

Incoming Request



Outgoing Request

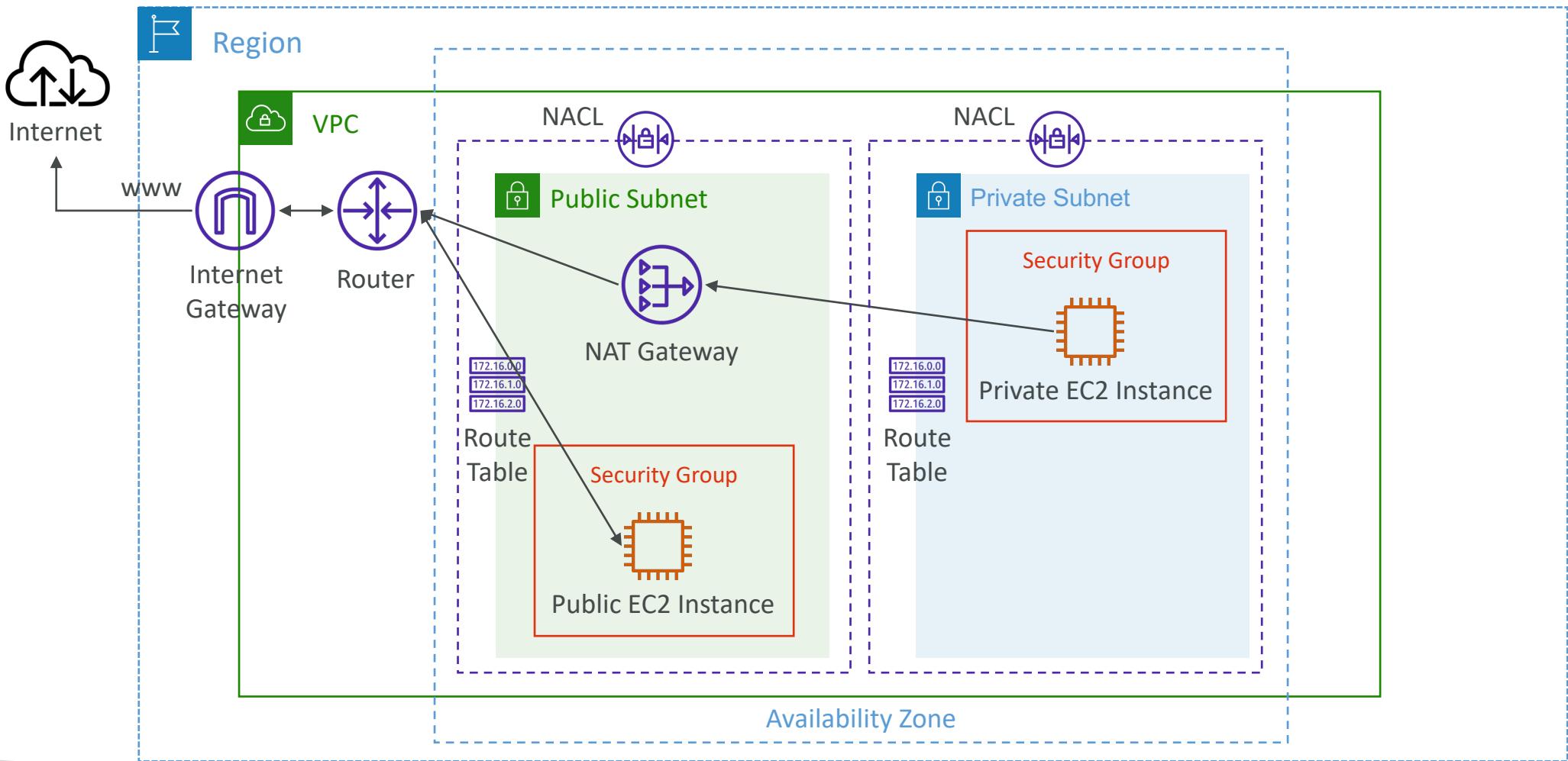


Network Access Control List (NACL)



- NACL are like a firewall which control traffic from and to subnets
- One NACL per subnet, new subnets are assigned the Default NACL
- You define NACL Rules:
 - Rules have a number (1-32766), higher precedence with a lower number
 - First rule match will drive the decision
 - Example: if you define #100 ALLOW 10.0.0.10/32 and #200 DENY 10.0.0.10/32, the IP address will be allowed because 100 has a higher precedence over 200
 - The last rule is an asterisk (*) and denies a request in case of no rule match
 - AWS recommends adding rules by increment of 100
- Newly created NACLs will deny everything
- NACL are a great way of blocking a specific IP address at the subnet level

NACLs



Default NACL

- Accepts everything inbound/outbound with the subnets it's associated with
- Do NOT modify the Default NACL, instead create custom NACLs



Default NACL for a VPC that supports IPv4

Inbound Rules

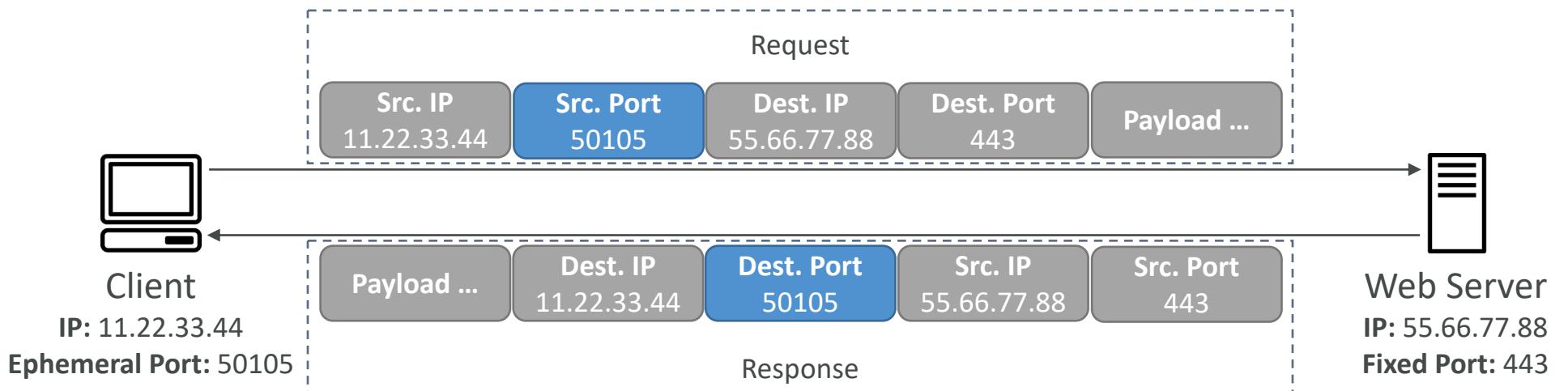
Rule #	Type	Protocol	Port Range	Source	Allow/Deny
100	All IPv4 Traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 Traffic	All	All	0.0.0.0/0	DENY

Outbound Rules

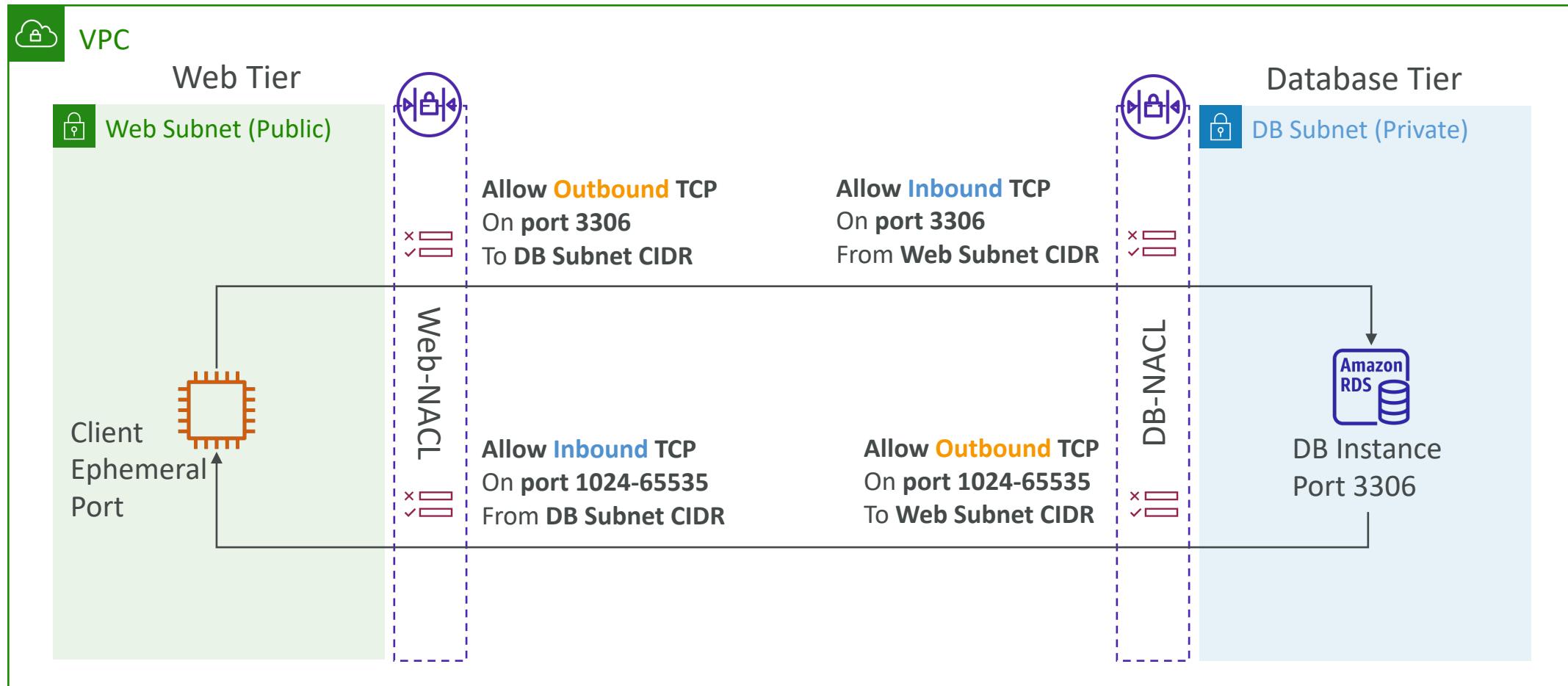
Rule #	Type	Protocol	Port Range	Destination	Allow/Deny
100	All IPv4 Traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 Traffic	All	All	0.0.0.0/0	DENY

Ephemeral Ports

- For any two endpoints to establish a connection, they must use ports
- Clients connect to a **defined port**, and expect a response on an **ephemeral port**
- Different Operating Systems use different port ranges, examples:
 - IANA & MS Windows 10 → 49152 – 65535
 - Many Linux Kernels → 32768 – 60999

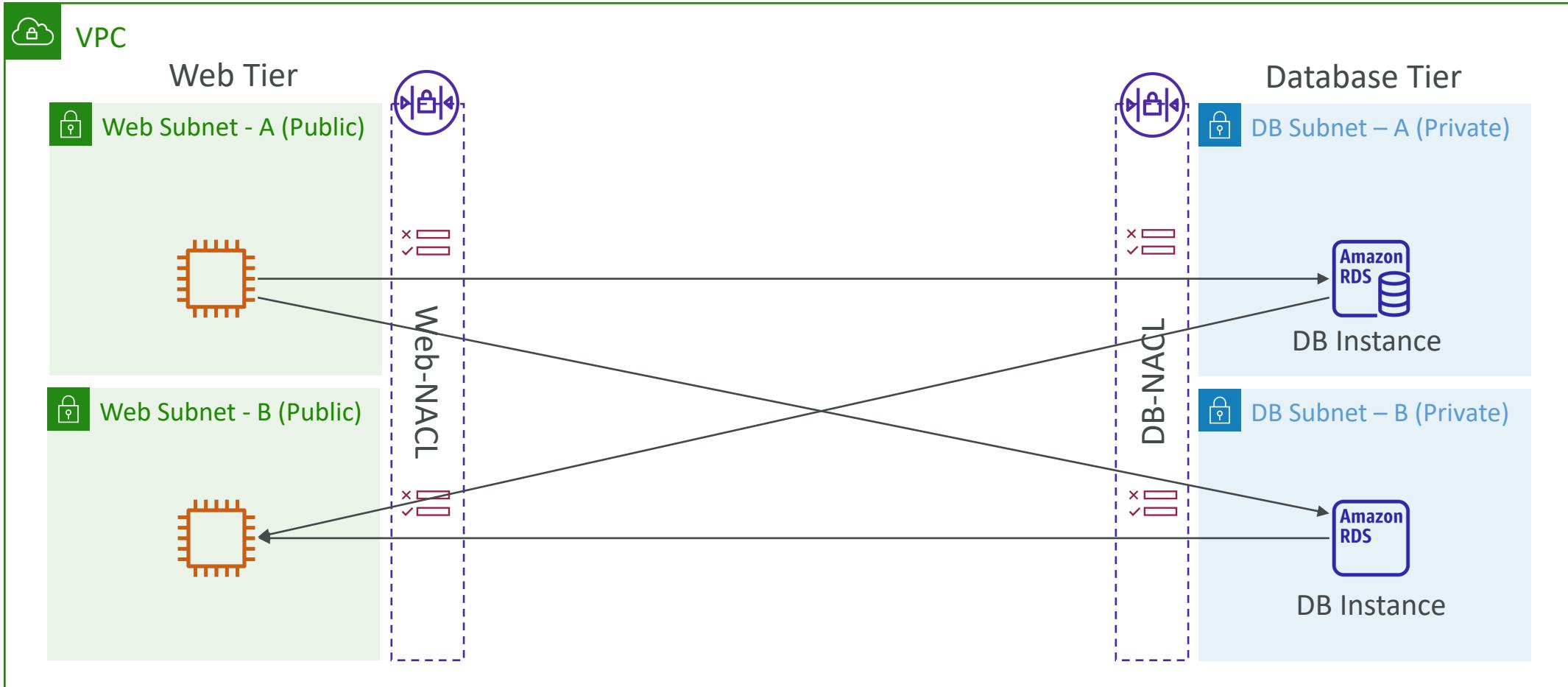


NACL with Ephemeral Ports



<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html#nacl-ephemeral-ports>

Create NACL rules for each target subnets CIDR



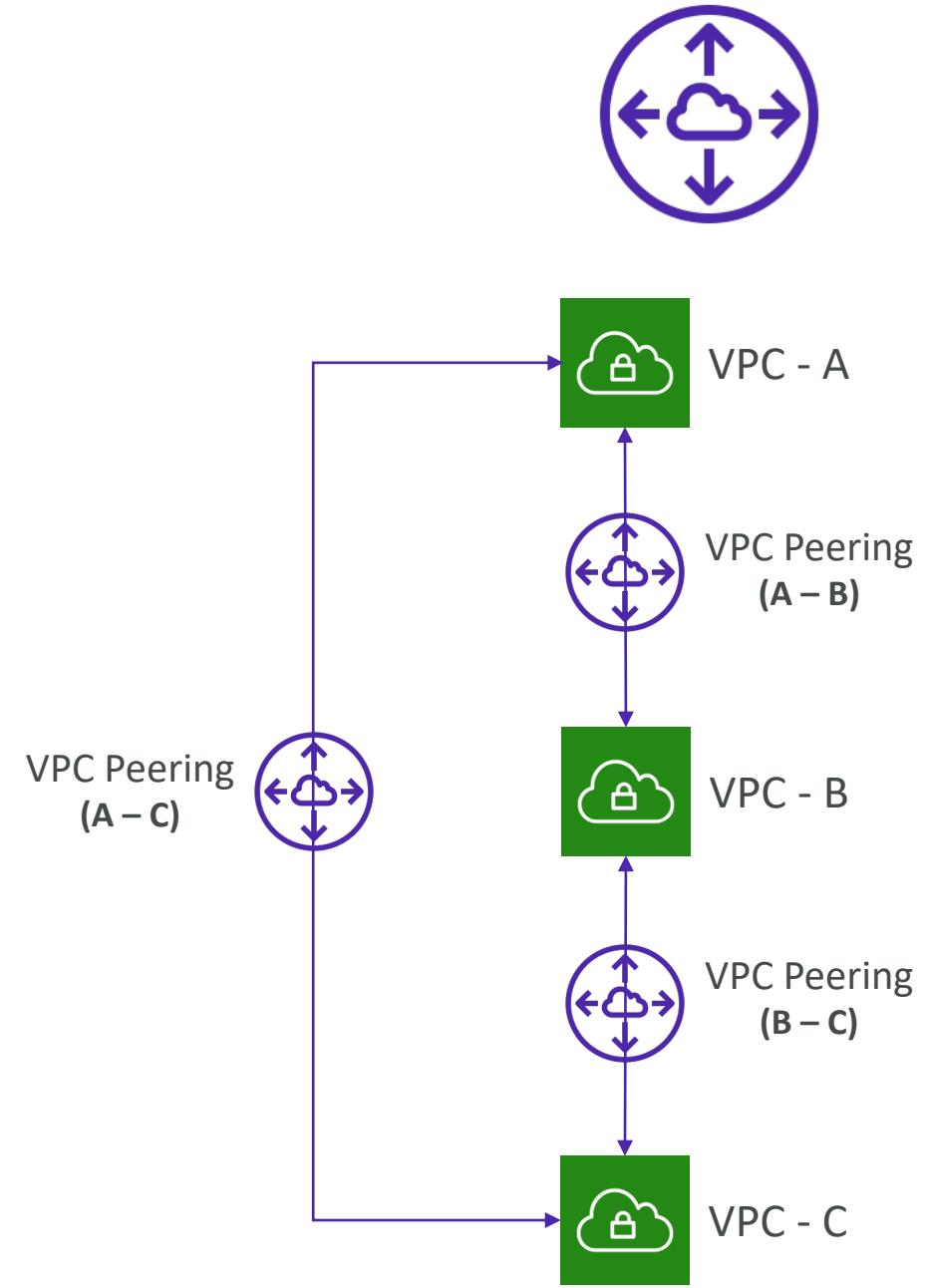
Security Group vs. NACLs

Security Group	NACL
Operates at the instance level	Operates at the subnet level
Supports allow rules only	Supports allow rules and deny rules
Stateful: return traffic is automatically allowed, regardless of any rules	Stateless: return traffic must be explicitly allowed by rules (think of ephemeral ports)
All rules are evaluated before deciding whether to allow traffic	Rules are evaluated in order (lowest to highest) when deciding whether to allow traffic, first match wins
Applies to an EC2 instance when specified by someone	Automatically applies to all EC2 instances in the subnet that it's associated with

NACL Examples: <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html>

VPC Peering

- Privately connect two VPCs using AWS' network
- Make them behave as if they were in the same network
- Must not have overlapping CIDRs
- VPC Peering connection is **NOT** transitive (must be established for each VPC that need to communicate with one another)
- You must update route tables in each VPC's subnets to ensure EC2 instances can communicate with each other



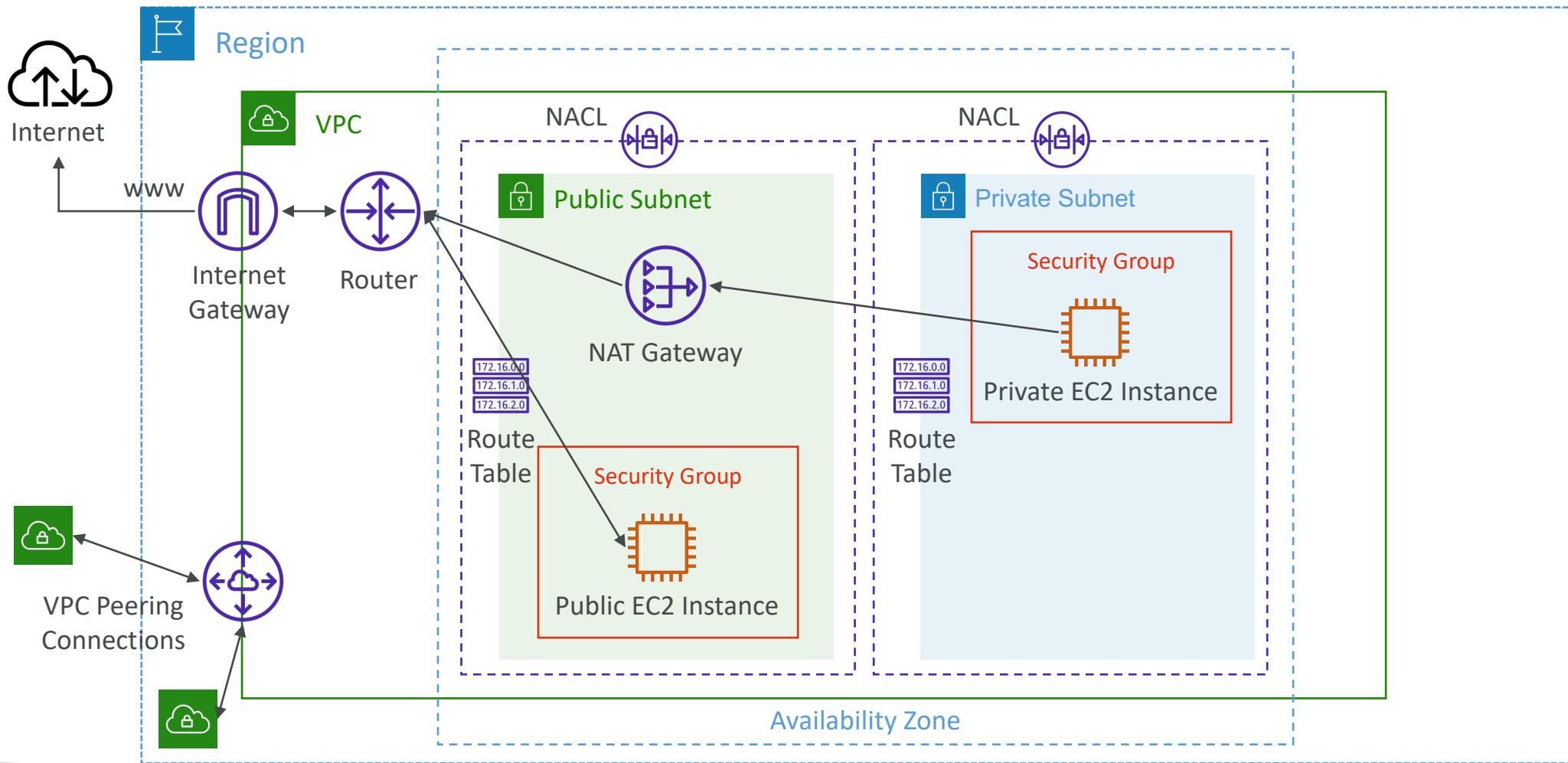
VPC Peering – Good to know

- You can create VPC Peering connection between VPCs in different AWS accounts/regions
- You can reference a security group in a peered VPC (works cross accounts – same region)

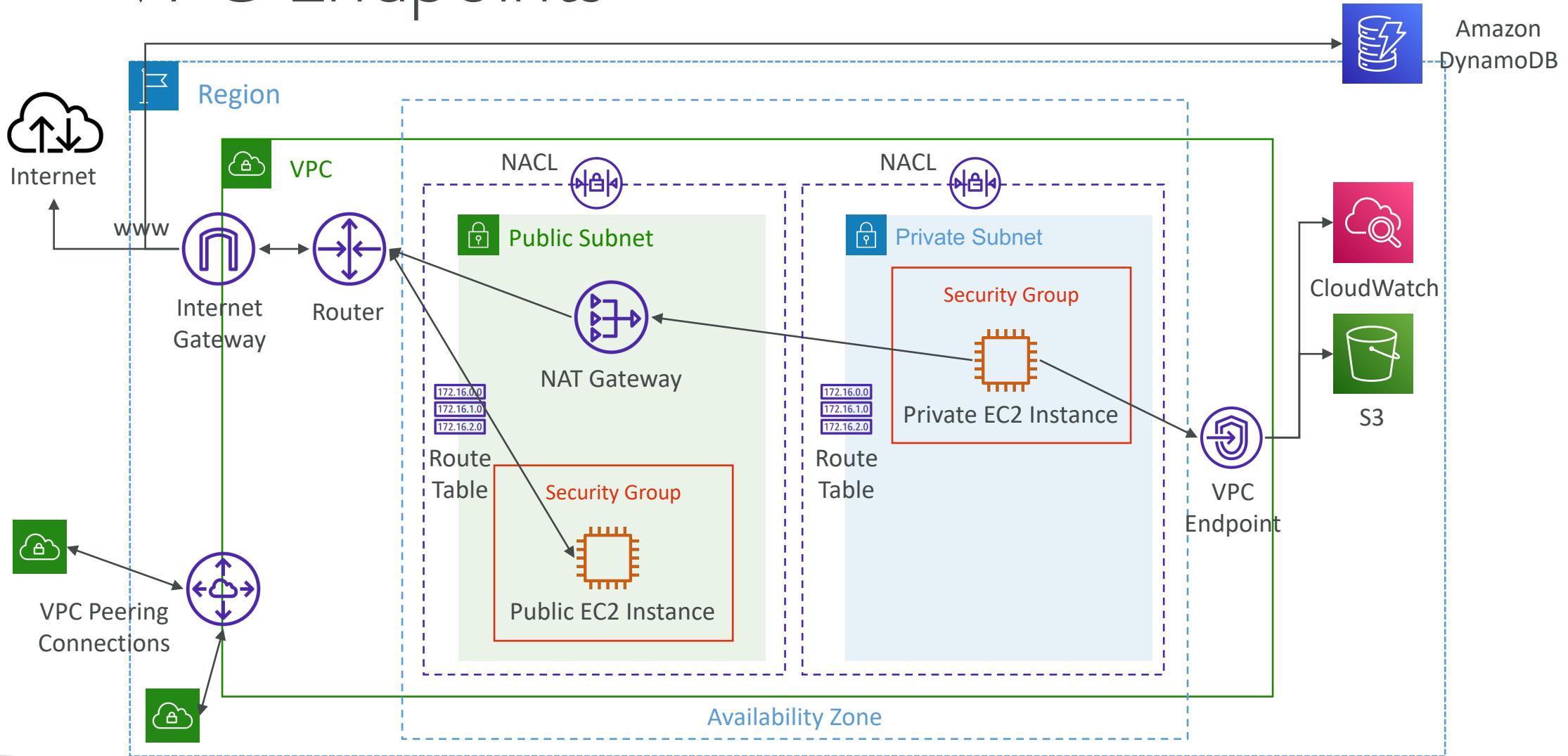
Type	Protocol	Port range	Source
HTTP	TCP	80	sg-04991f9af3473b939 / default
HTTP	TCP	80	[REDACTED] / sg-027ad1f7865d4be76

↑
Account ID

VPC Peering



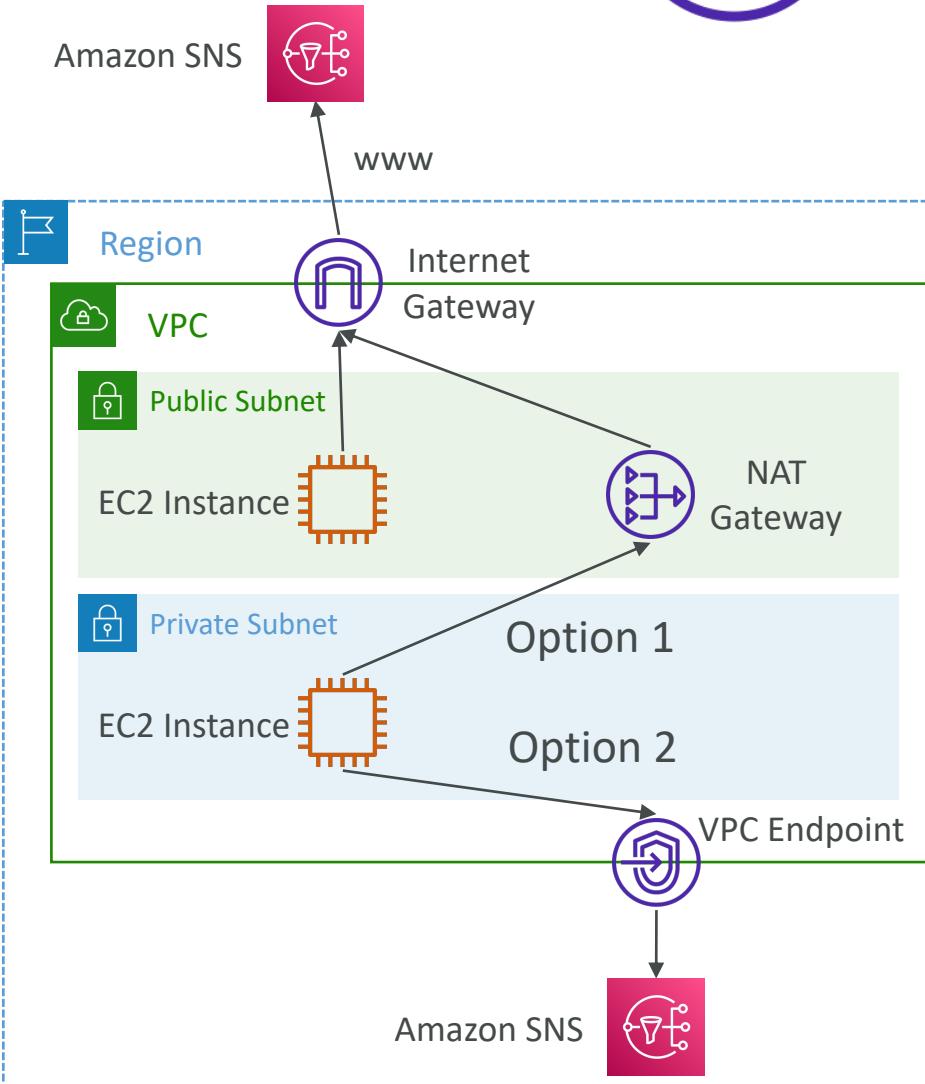
VPC Endpoints



VPC Endpoints (AWS PrivateLink)



- Every AWS service is publicly exposed (public URL)
- VPC Endpoints (powered by AWS PrivateLink) allows you to connect to AWS services using a **private network** instead of using the public Internet
- They're redundant and scale horizontally
- They remove the need of IGW, NATGW, ... to access AWS Services
- In case of issues:
 - Check DNS Setting Resolution in your VPC
 - Check Route Tables



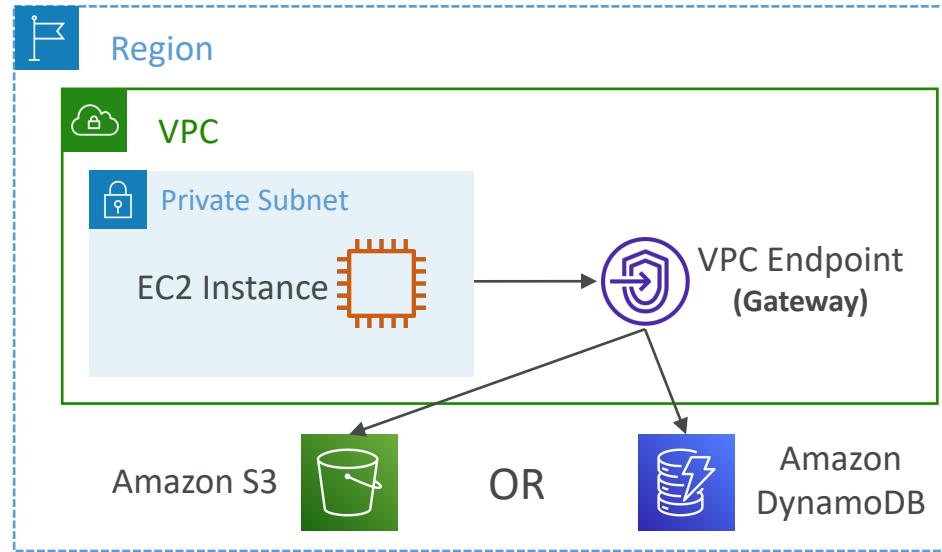
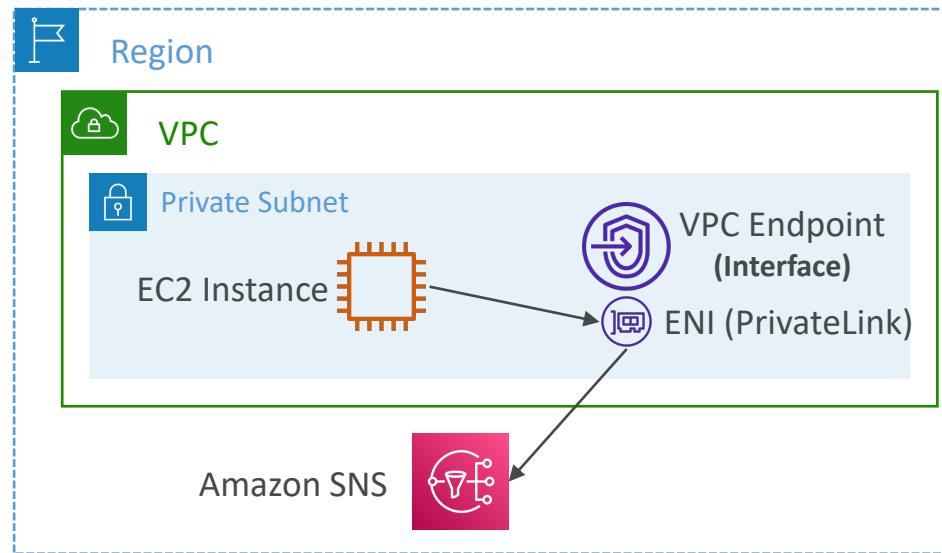
Types of Endpoints

- Interface Endpoints (powered by PrivateLink)

- Provisions an ENI (private IP address) as an entry point (must attach a Security Group)
- Supports most AWS services
- \$ per hour + \$ per GB of data processed

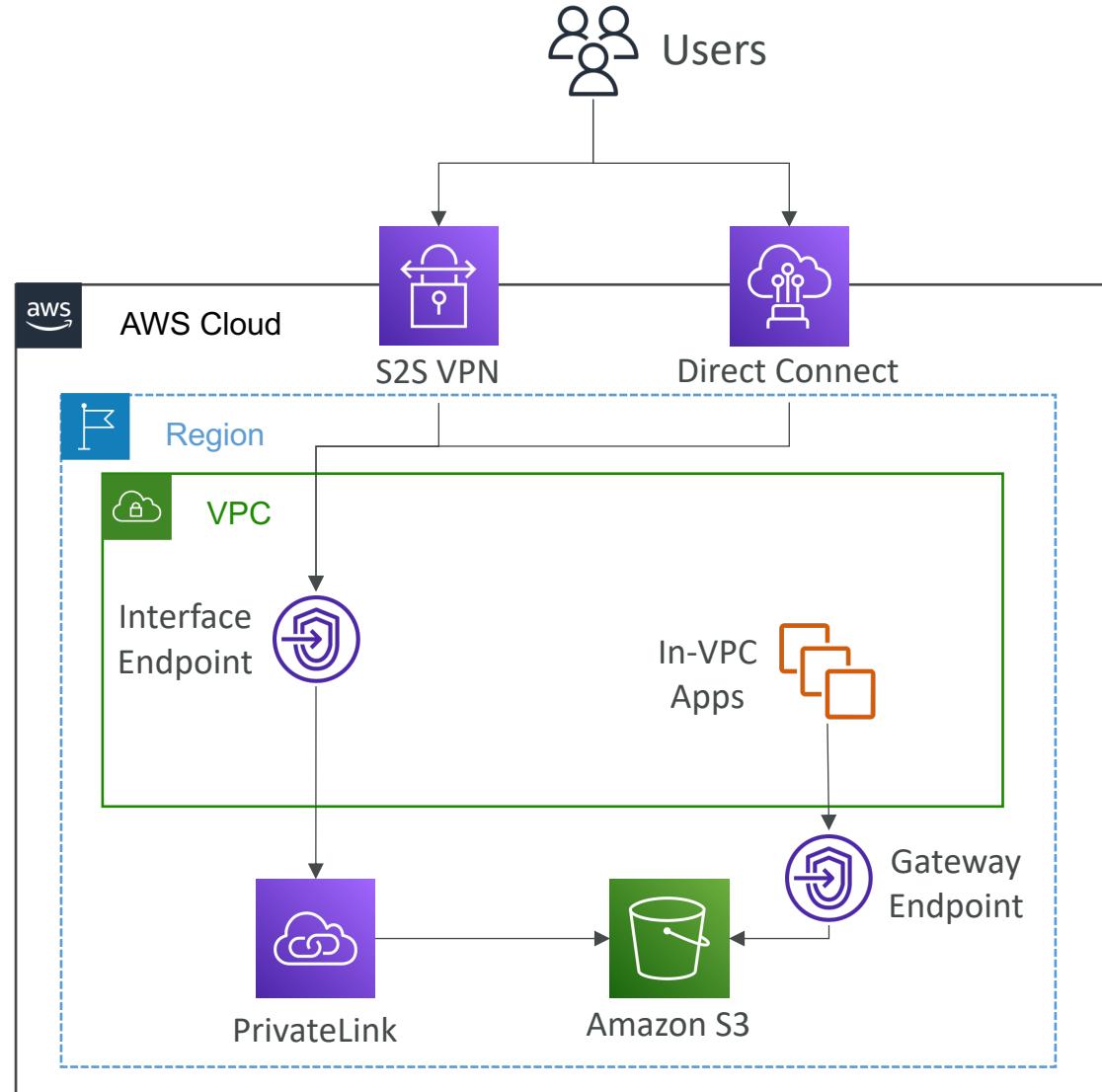
- Gateway Endpoints

- Provisions a gateway and must be used as a target in a route table (does not use security groups)
- Supports both S3 and DynamoDB
- Free



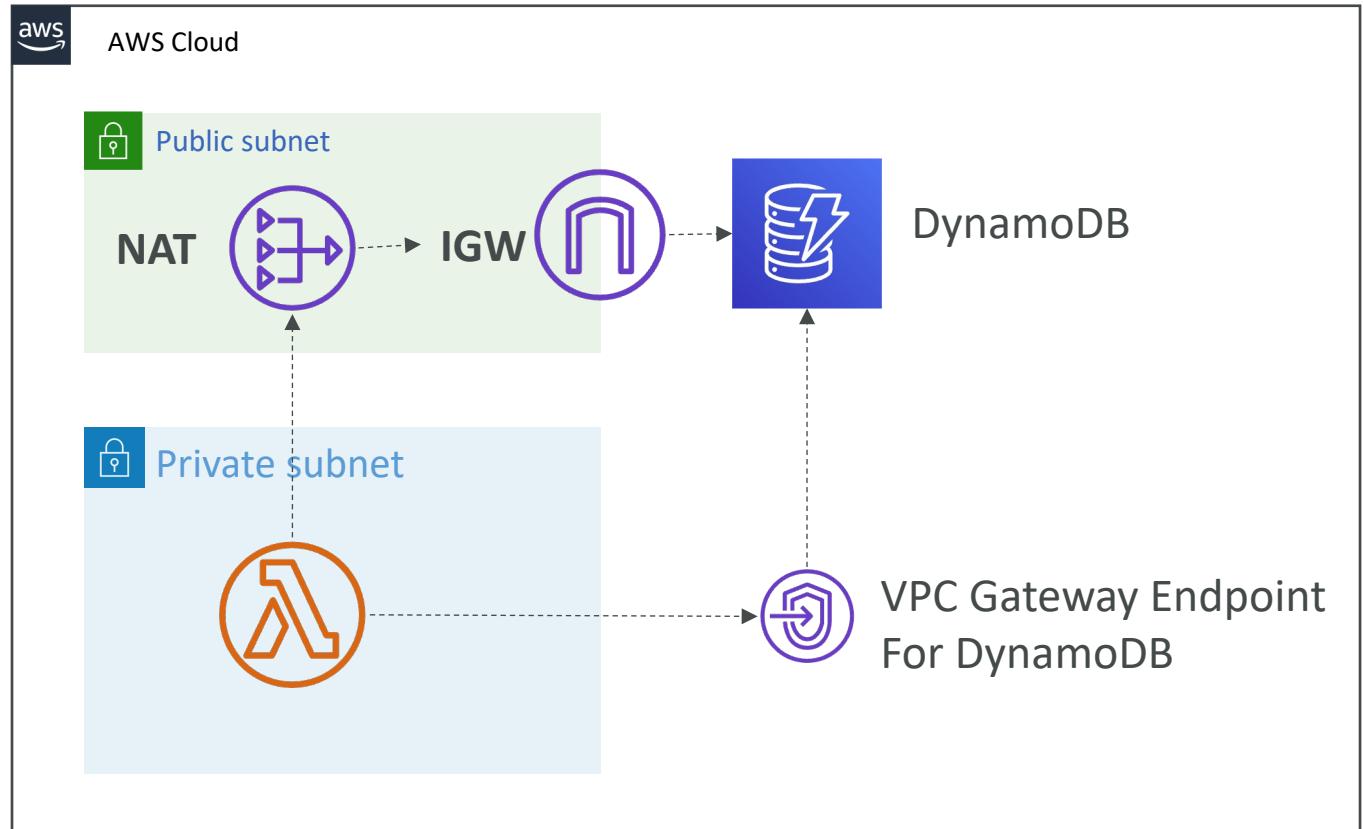
Gateway or Interface Endpoint for S3?

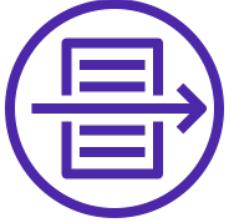
- Gateway is most likely going to be preferred all the time at the exam
- Cost: free for Gateway, \$ for interface endpoint
- Interface Endpoint is preferred access is required from on-premises (Site to Site VPN or Direct Connect), a different VPC or a different region



Lambda in VPC accessing DynamoDB

- DynamoDB is a public service from AWS
- Option 1: Access from the public internet
 - Because Lambda is in a VPC, it needs a NAT Gateway in a public subnet and an internet gateway
- Option 2 (better & free): Access from the private VPC network
 - Deploy a VPC Gateway endpoint for DynamoDB
 - Change the Route Tables

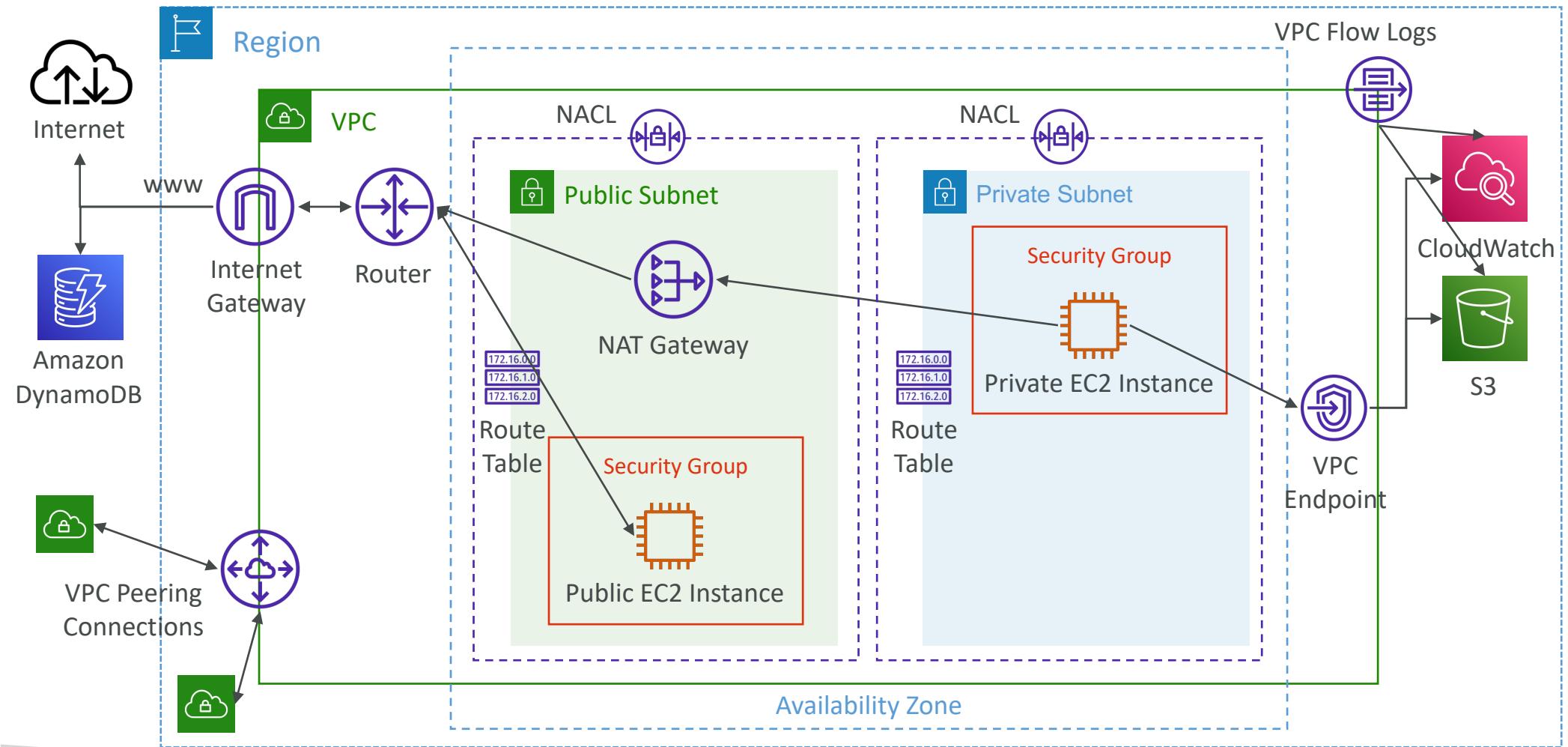




VPC Flow Logs

- Capture information about IP traffic going into your interfaces:
 - VPC Flow Logs
 - Subnet Flow Logs
 - Elastic Network Interface (ENI) Flow Logs
- Helps to monitor & troubleshoot connectivity issues
- Flow logs data can go to S3, CloudWatch Logs, and Kinesis Data Firehose
- Captures network information from AWS managed interfaces too: ELB, RDS, ElastiCache, Redshift, WorkSpaces, NATGW, Transit Gateway...

VPC Flow Logs



VPC Flow Logs Syntax

version	interface-id	dstaddr	dstport	packets	start	action
2	123456789010	eni-1235b8ca123456789	172.31.16.139	172.31.16.21	20641	ACCEPT OK
2	123456789010	eni-1235b8ca123456789	172.31.9.69	172.31.9.12	49761	REJECT OK
account-id	srcaddr	srcport	protocol	bytes	end	log-status

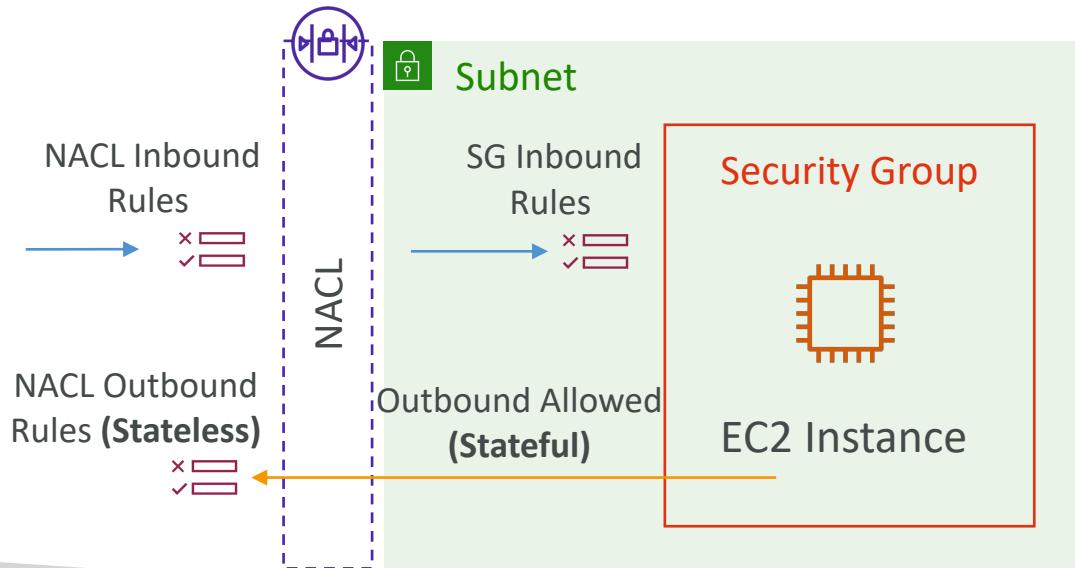
- srcaddr & dstaddr – help identify problematic IP
- srcport & dstport – help identify problematic ports
- Action – success or failure of the request due to Security Group / NACL
- Can be used for analytics on usage patterns, or malicious behavior
- Query VPC flow logs using Athena on S3 or CloudWatch Logs Insights
- Flow Logs examples: <https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs-records-examples.html>

VPC Flow Logs – Troubleshoot SG & NACL issues

Look at the “ACTION” field

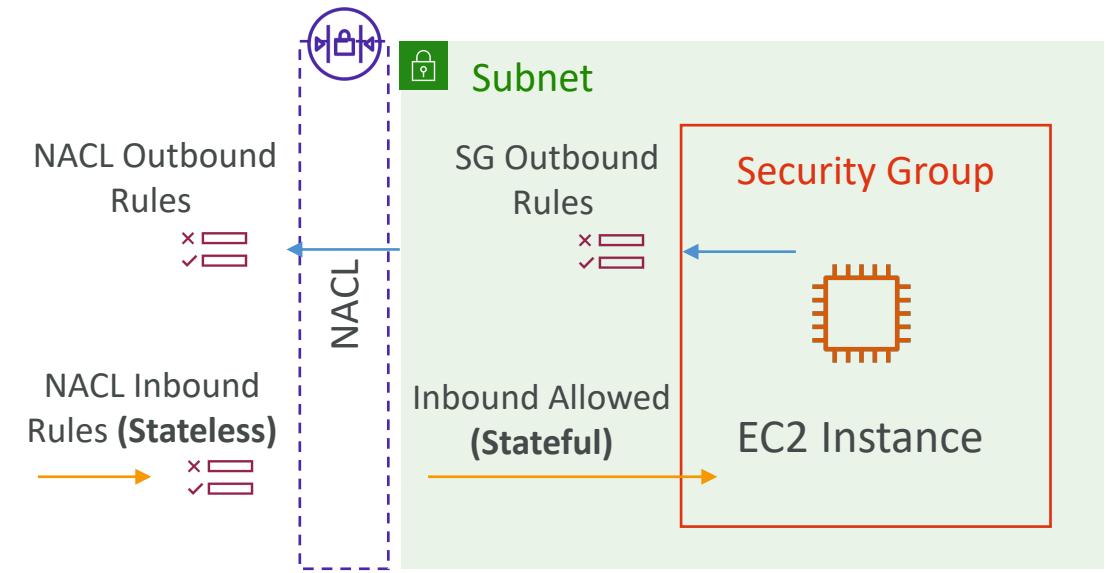
Incoming Requests

- Inbound REJECT => NACL or SG
- Inbound ACCEPT, Outbound REJECT => NACL

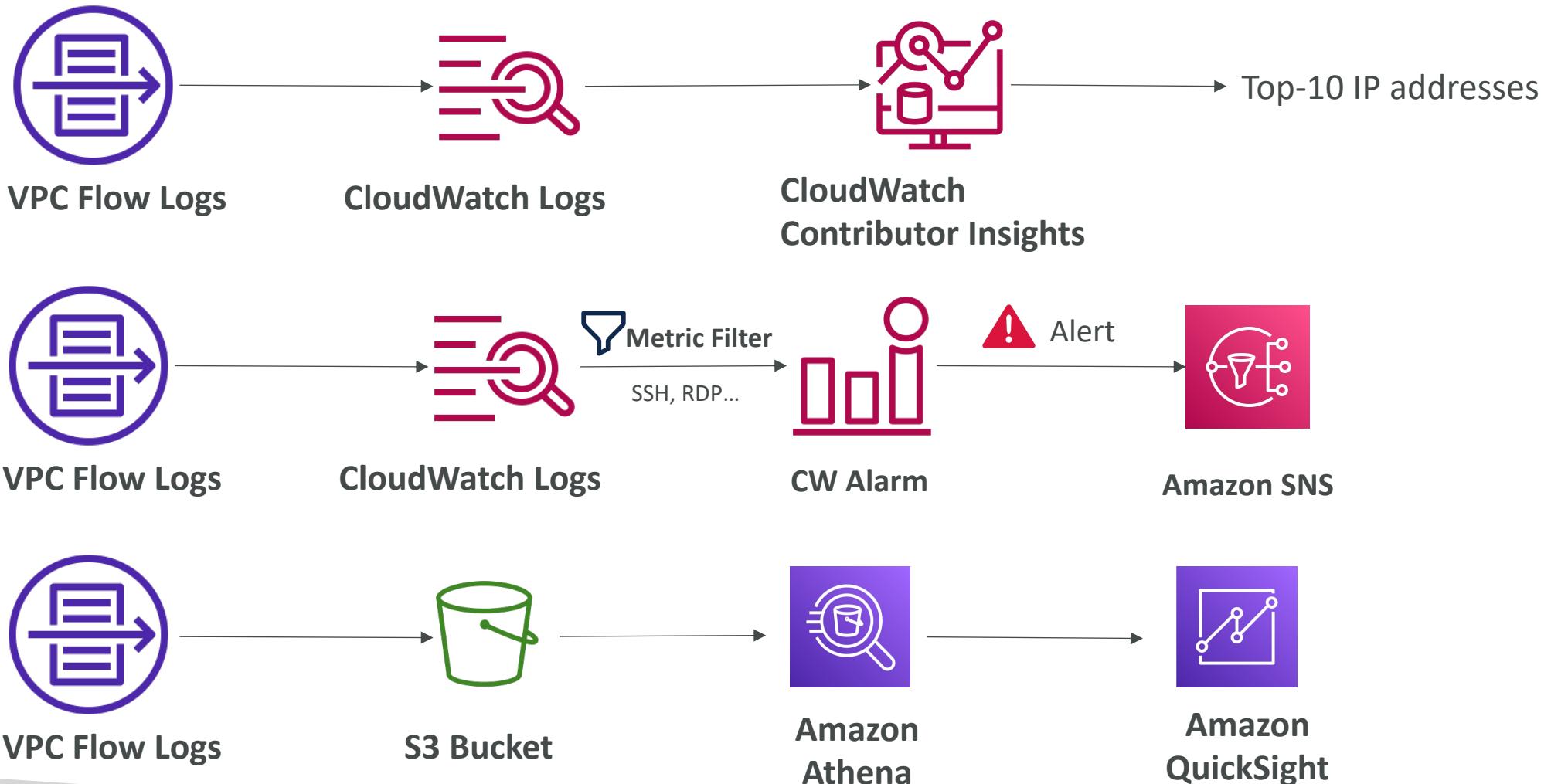


Outgoing Requests

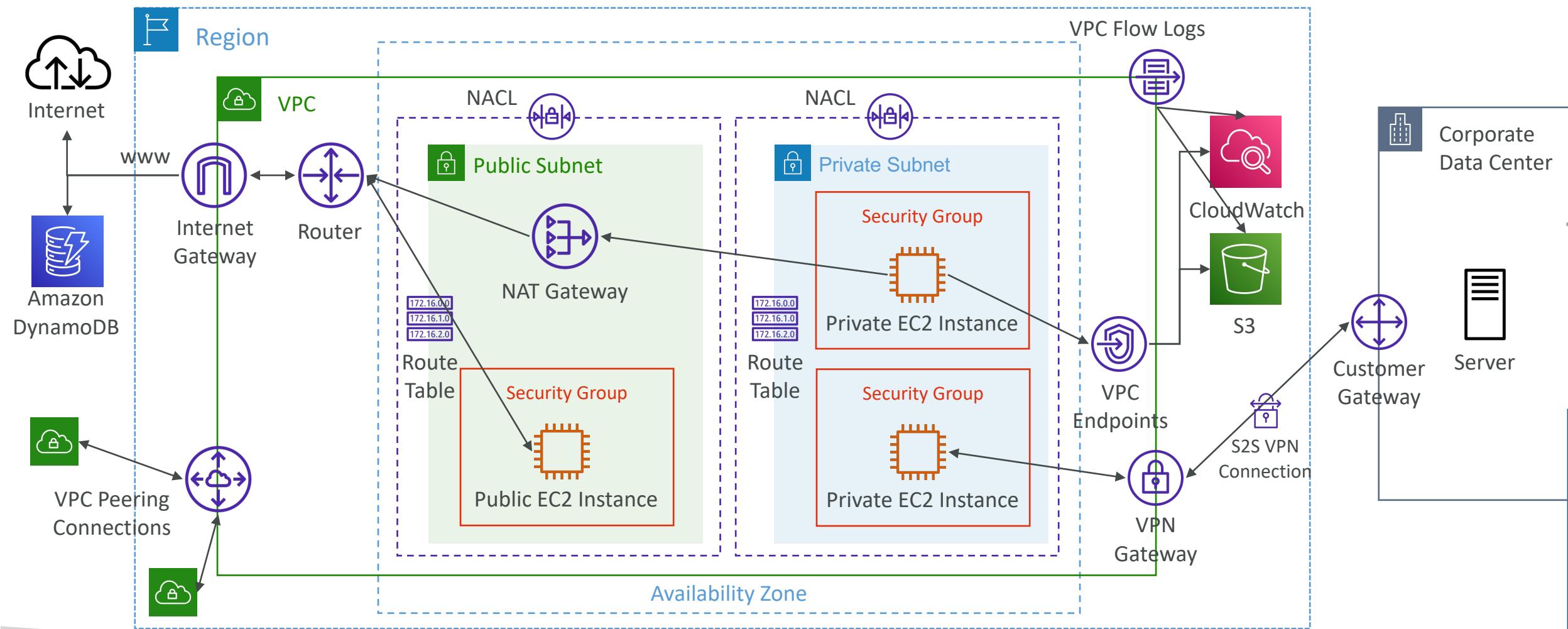
- Outbound REJECT => NACL or SG
- Outbound ACCEPT, Inbound REJECT => NACL



VPC Flow Logs – Architectures



AWS Site-to-Site VPN



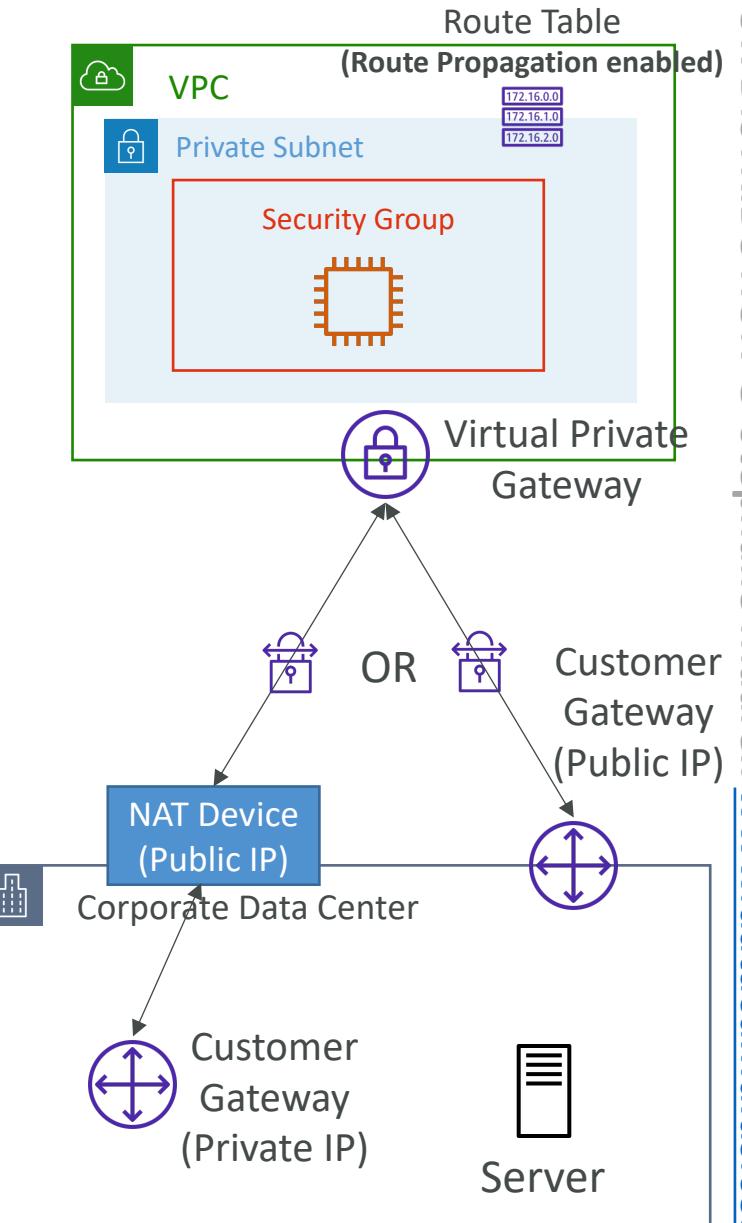
AWS Site-to-Site VPN



- **Virtual Private Gateway (VGW)**
 - VPN concentrator on the AWS side of the VPN connection
 - VGW is created and attached to the VPC from which you want to create the Site-to-Site VPN connection
 - Possibility to customize the ASN (Autonomous System Number)
- **Customer Gateway (CGW)**
 - Software application or physical device on customer side of the VPN connection
 - <https://docs.aws.amazon.com/vpn/latest/s2svpn/your-cgw.html#DevicesTested>

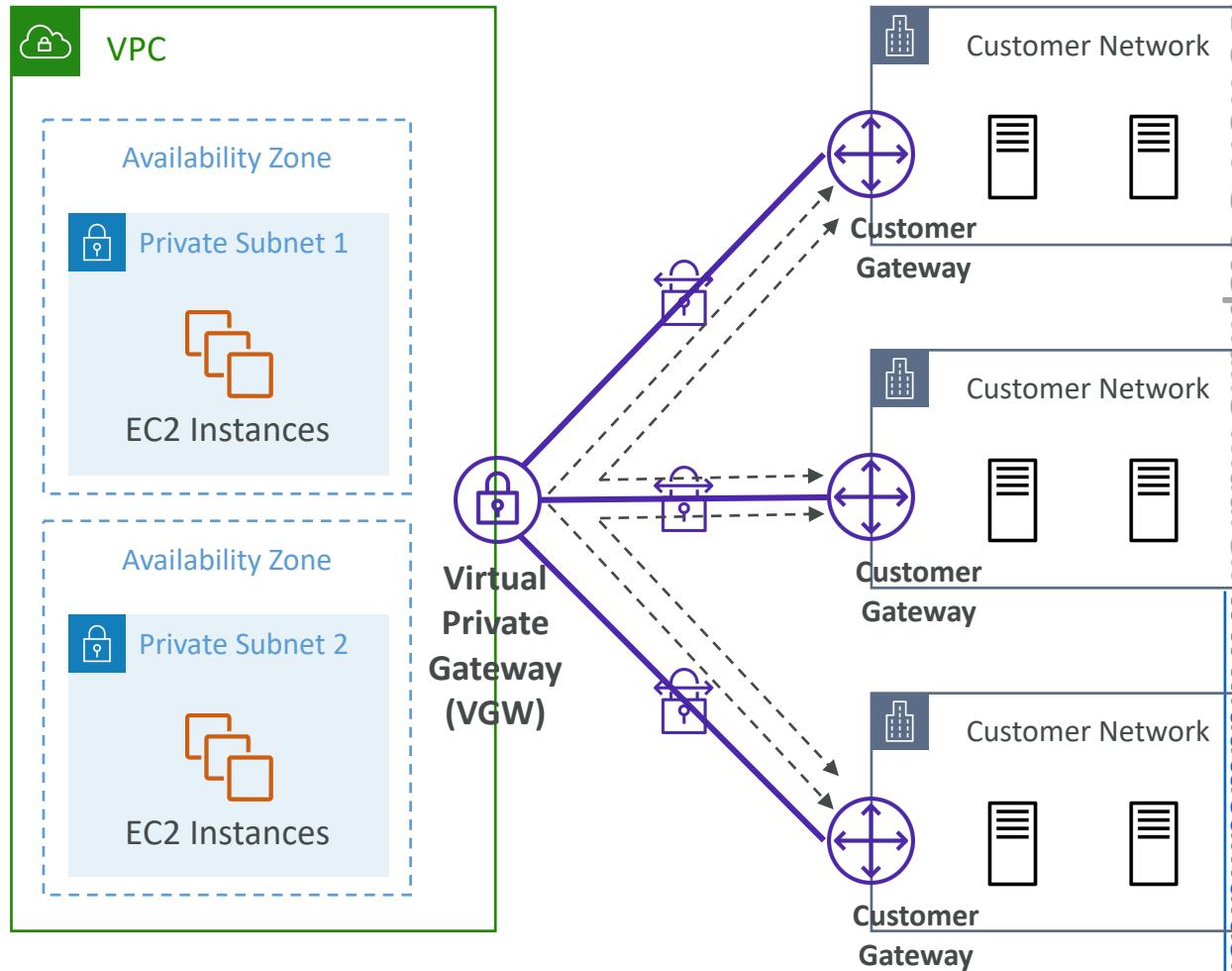
Site-to-Site VPN Connections

- Customer Gateway Device (On-premises)
 - What IP address to use?
 - Public Internet-routable IP address for your Customer Gateway device
 - If it's behind a NAT device that's enabled for NAT traversal (NAT-T), use the public IP address of the NAT device
- Important step: enable Route Propagation for the Virtual Private Gateway in the route table that is associated with your subnets
- If you need to ping your EC2 instances from on-premises, make sure you add the ICMP protocol on the inbound of your security groups



AWS VPN CloudHub

- Provide secure communication between multiple sites, if you have multiple VPN connections
- Low-cost hub-and-spoke model for primary or secondary network connectivity between different locations (VPN only)
- It's a VPN connection so it goes over the public Internet
- To set it up, connect multiple VPN connections on the same VGW, setup dynamic routing and configure route tables

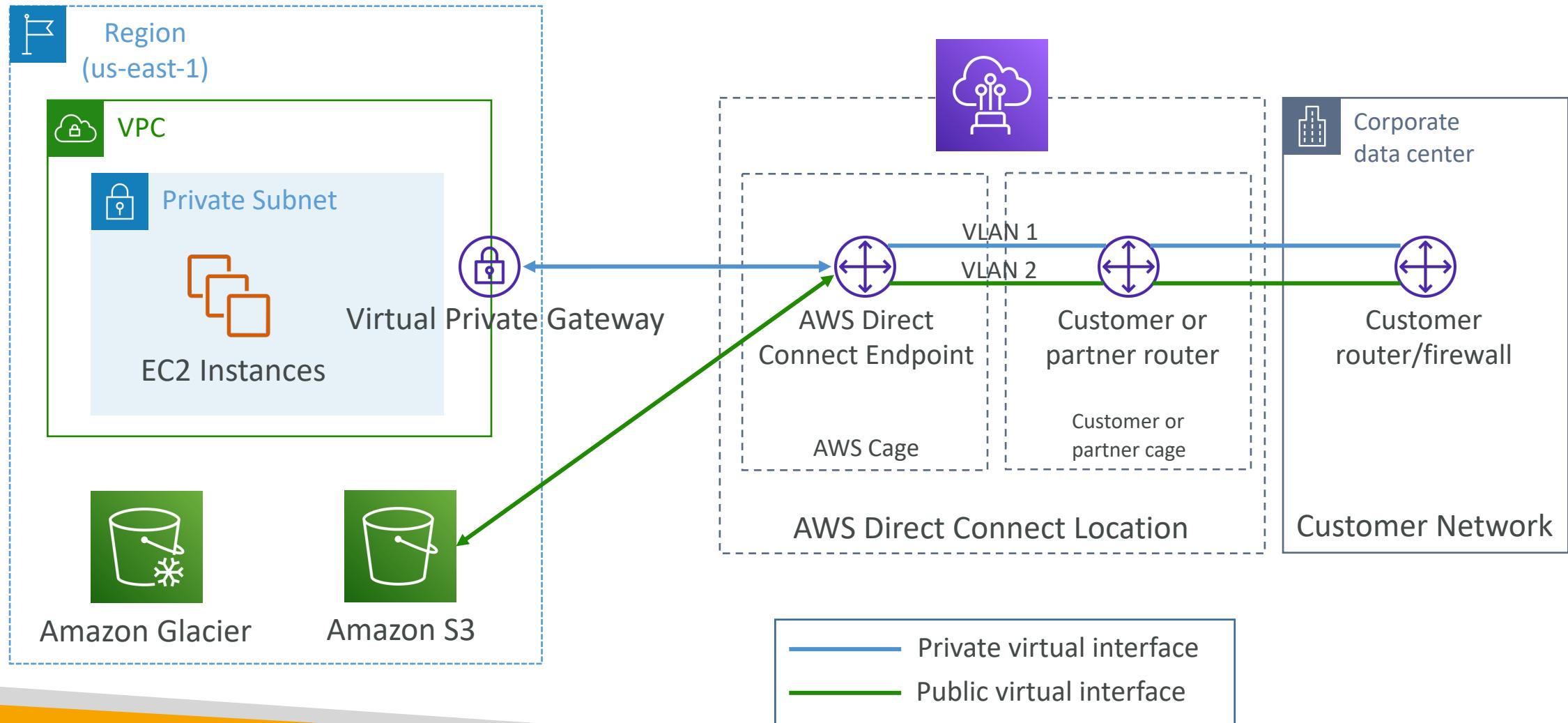




Direct Connect (DX)

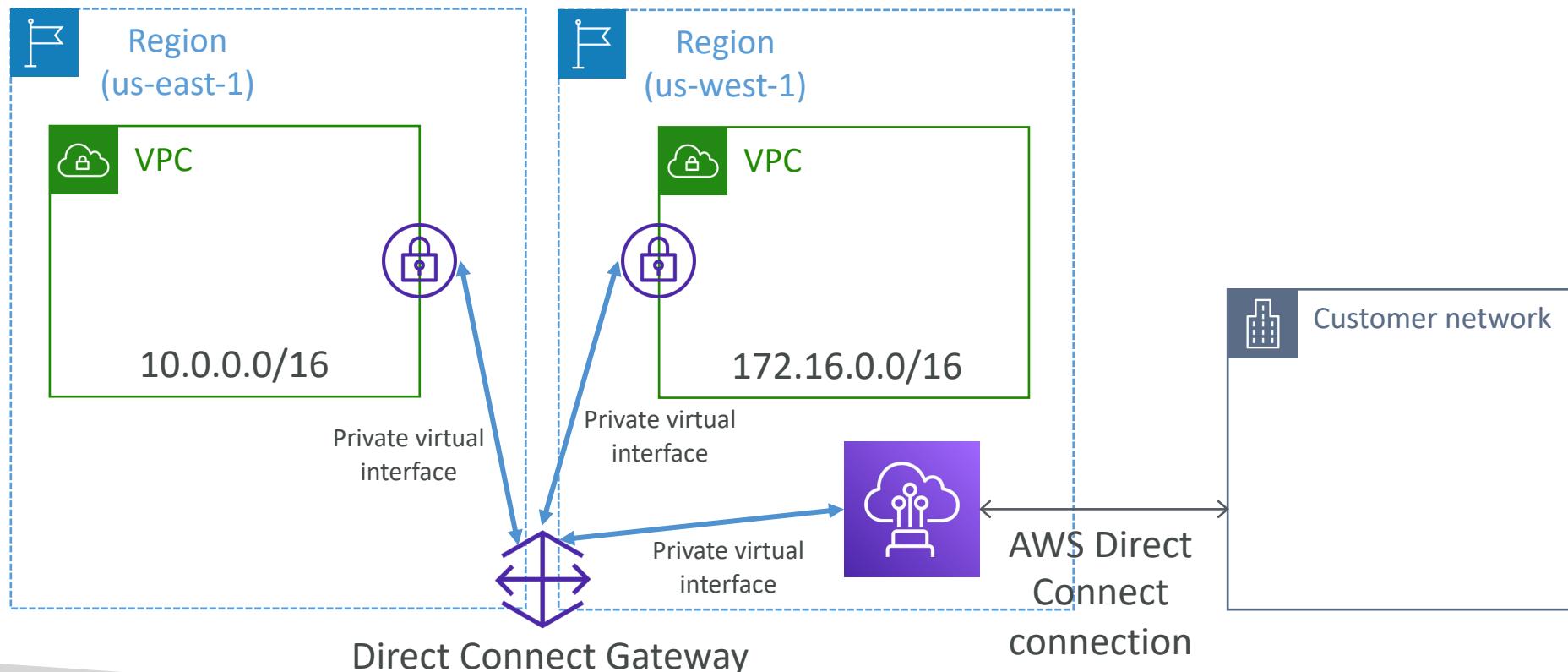
- Provides a dedicated private connection from a remote network to your VPC
- Dedicated connection must be setup between your DC and AWS Direct Connect locations
- You need to setup a Virtual Private Gateway on your VPC
- Access public resources (S3) and private (EC2) on same connection
- Use Cases:
 - Increase bandwidth throughput - working with large data sets – lower cost
 - More consistent network experience - applications using real-time data feeds
 - Hybrid Environments (on prem + cloud)
- Supports both IPv4 and IPv6

Direct Connect Diagram



Direct Connect Gateway

- If you want to setup a Direct Connect to one or more VPC in many different regions (same account), you must use a Direct Connect Gateway

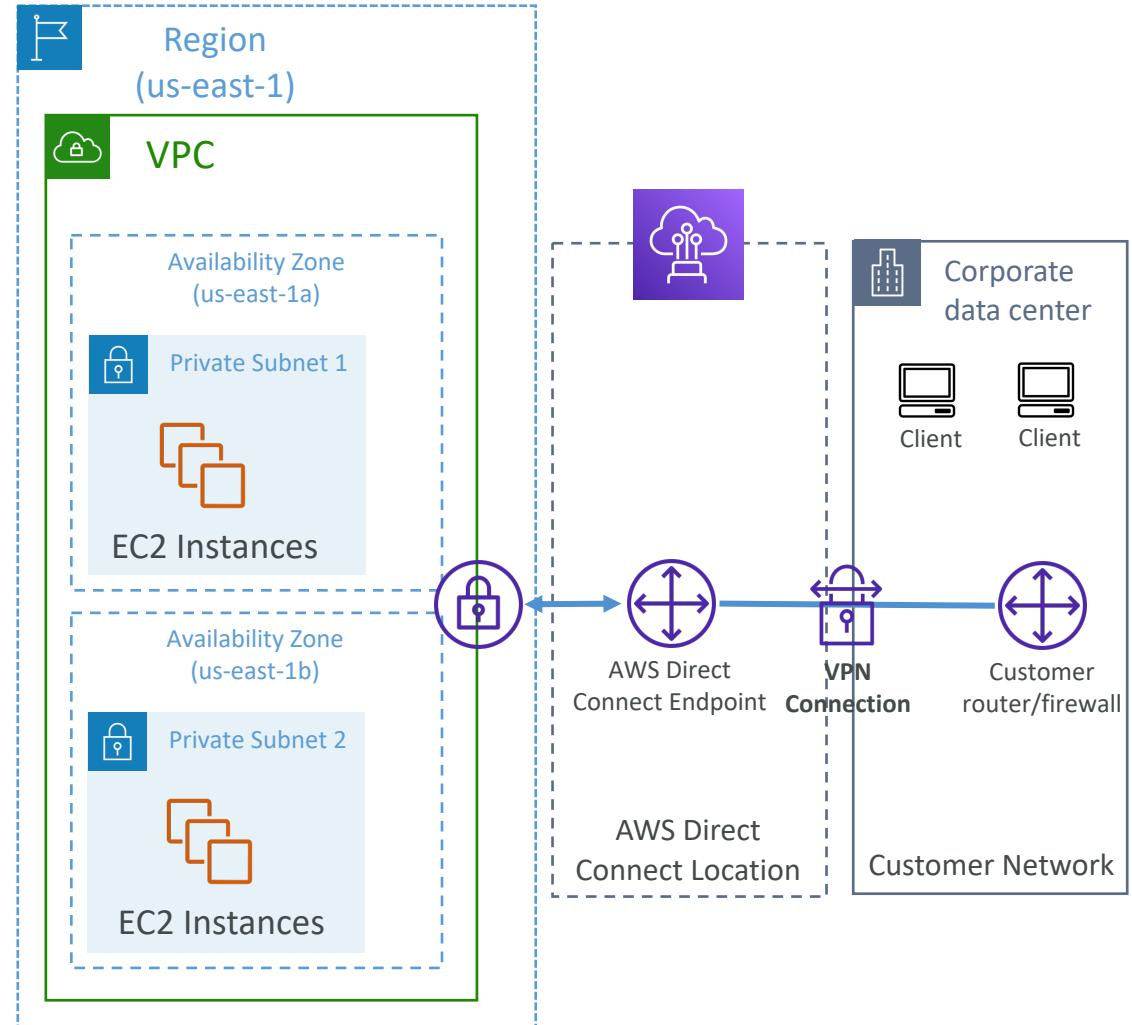


Direct Connect – Connection Types

- **Dedicated Connections:** 1 Gbps, 10 Gbps and 100 Gbps capacity
 - Physical ethernet port dedicated to a customer
 - Request made to AWS first, then completed by AWS Direct Connect Partners
- **Hosted Connections:** 50Mbps, 500 Mbps, to 10 Gbps
 - Connection requests are made via AWS Direct Connect Partners
 - Capacity can be **added or removed on demand**
 - 1, 2, 5, 10 Gbps available at select AWS Direct Connect Partners
- Lead times are often longer than 1 month to establish a new connection

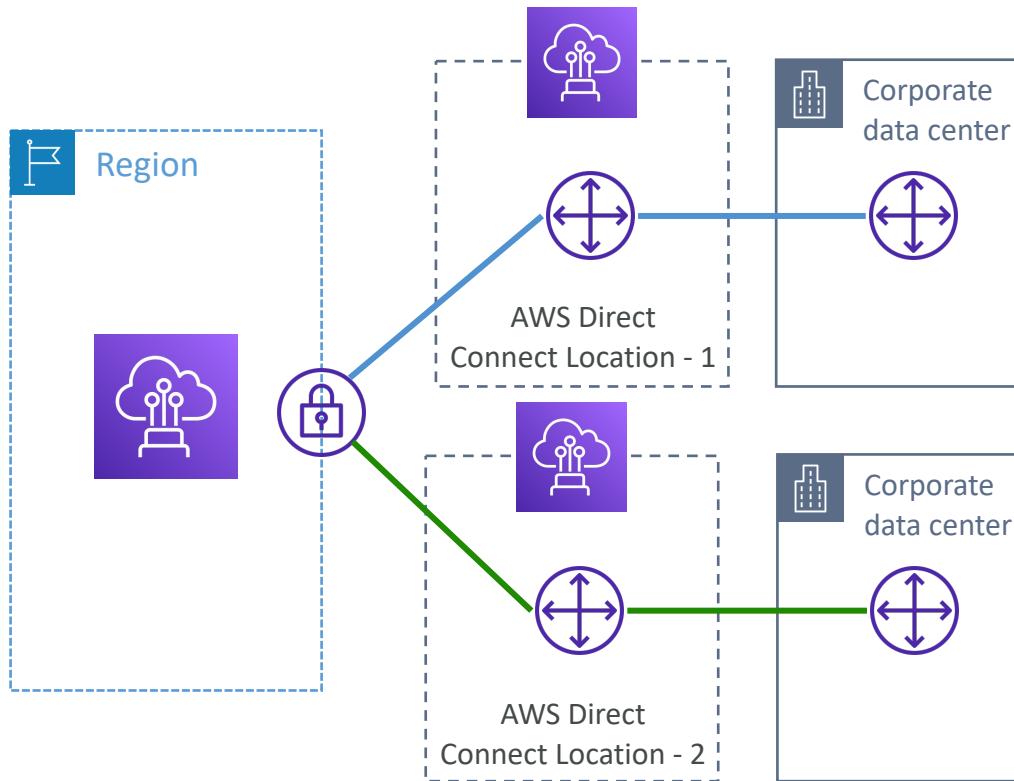
Direct Connect – Encryption

- Data in transit is not encrypted but is private
- AWS Direct Connect + VPN provides an IPsec-encrypted private connection
- Good for an extra level of security, but slightly more complex to put in place



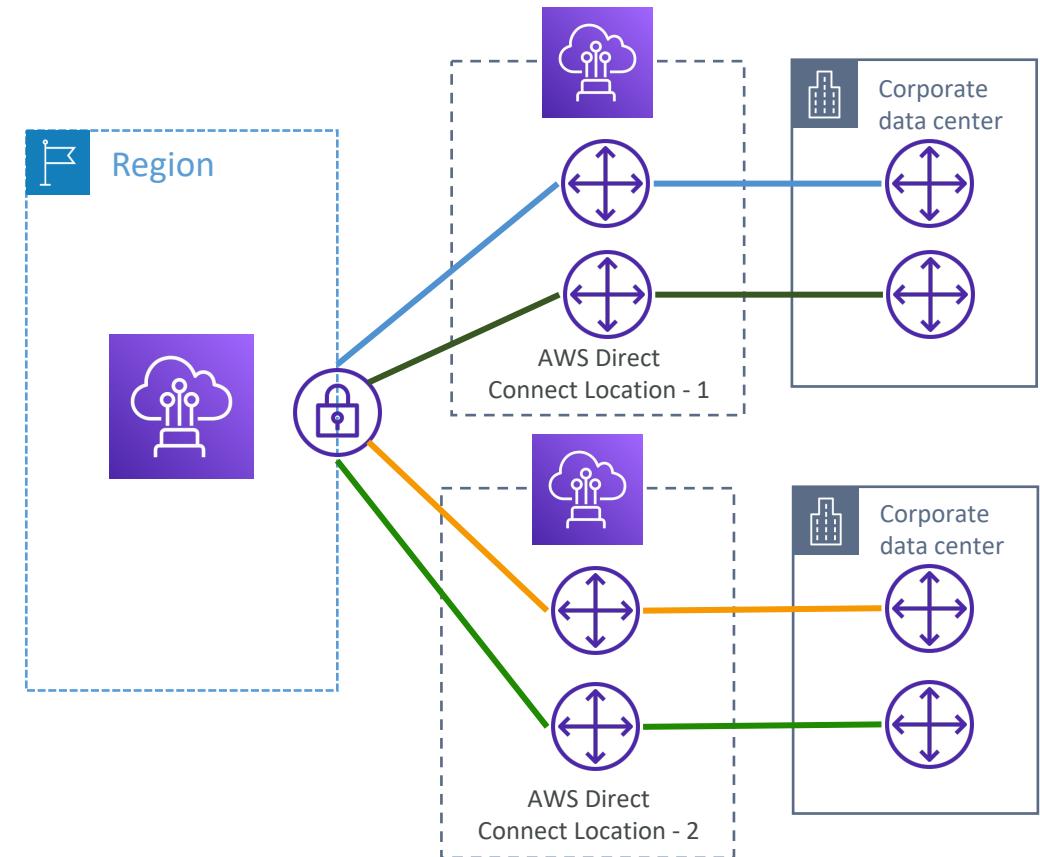
Direct Connect - Resiliency

High Resiliency for Critical Workloads



One connection at multiple locations

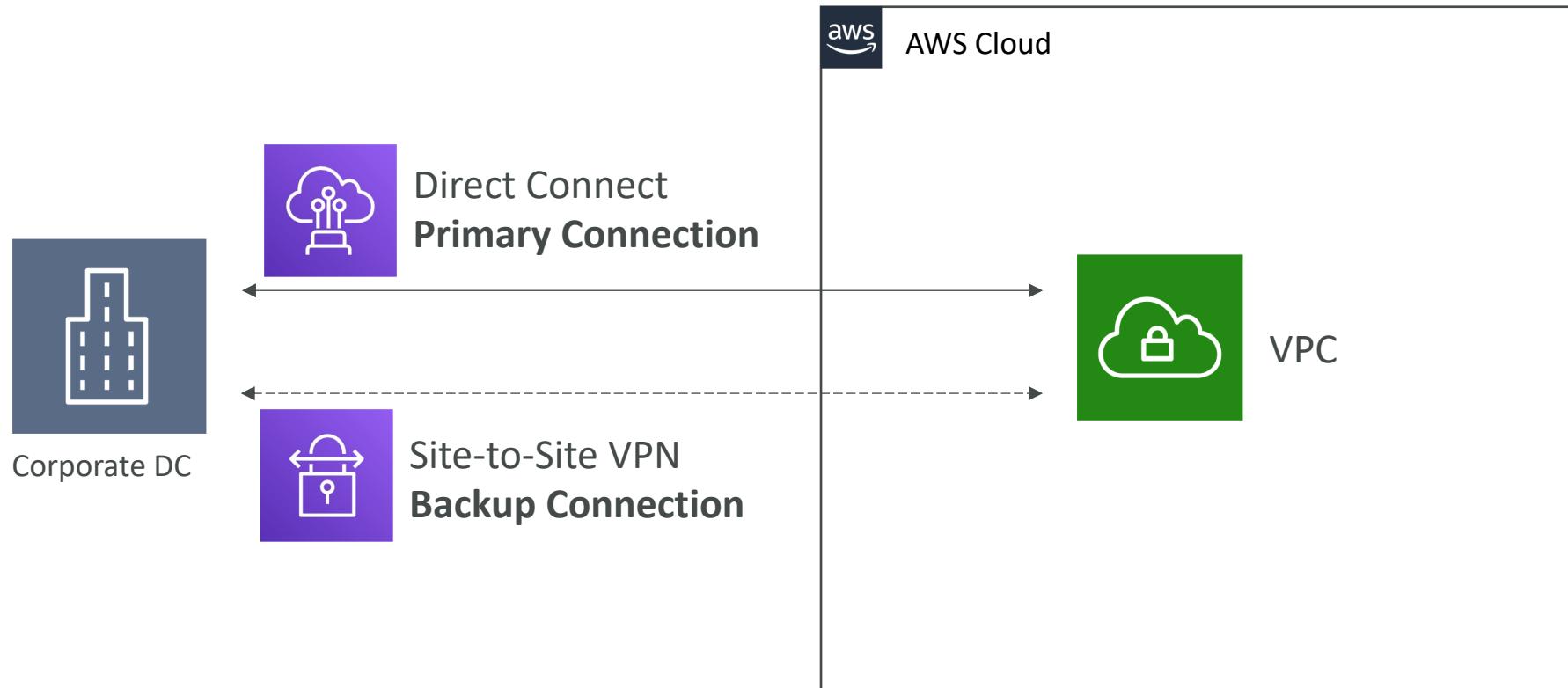
Maximum Resiliency for Critical Workloads



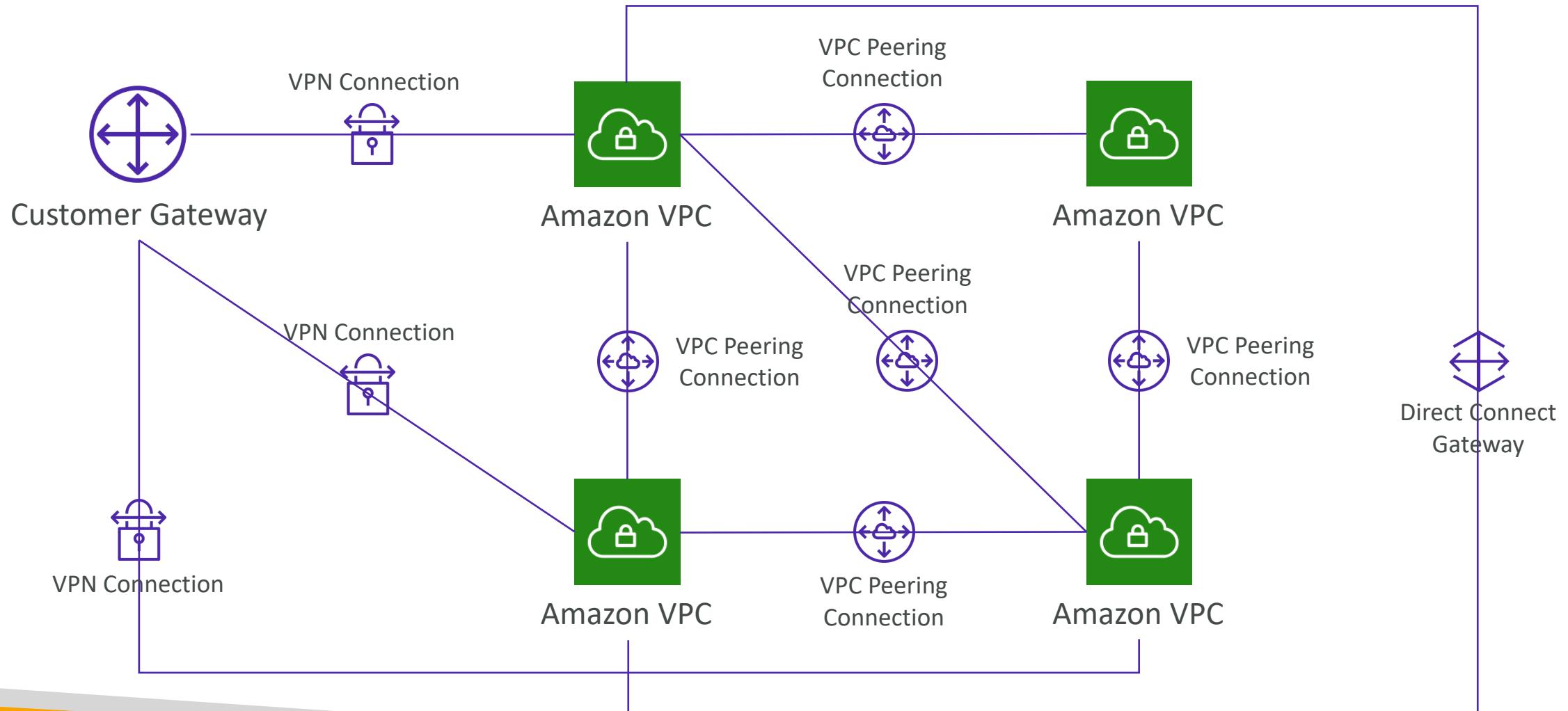
Maximum resilience is achieved by separate connections terminating on separate devices in more than one location.

Site-to-Site VPN connection as a backup

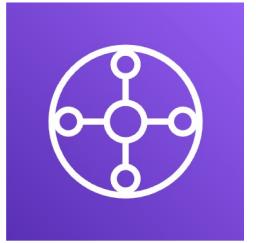
- In case Direct Connect fails, you can set up a backup Direct Connect connection (expensive), or a Site-to-Site VPN connection



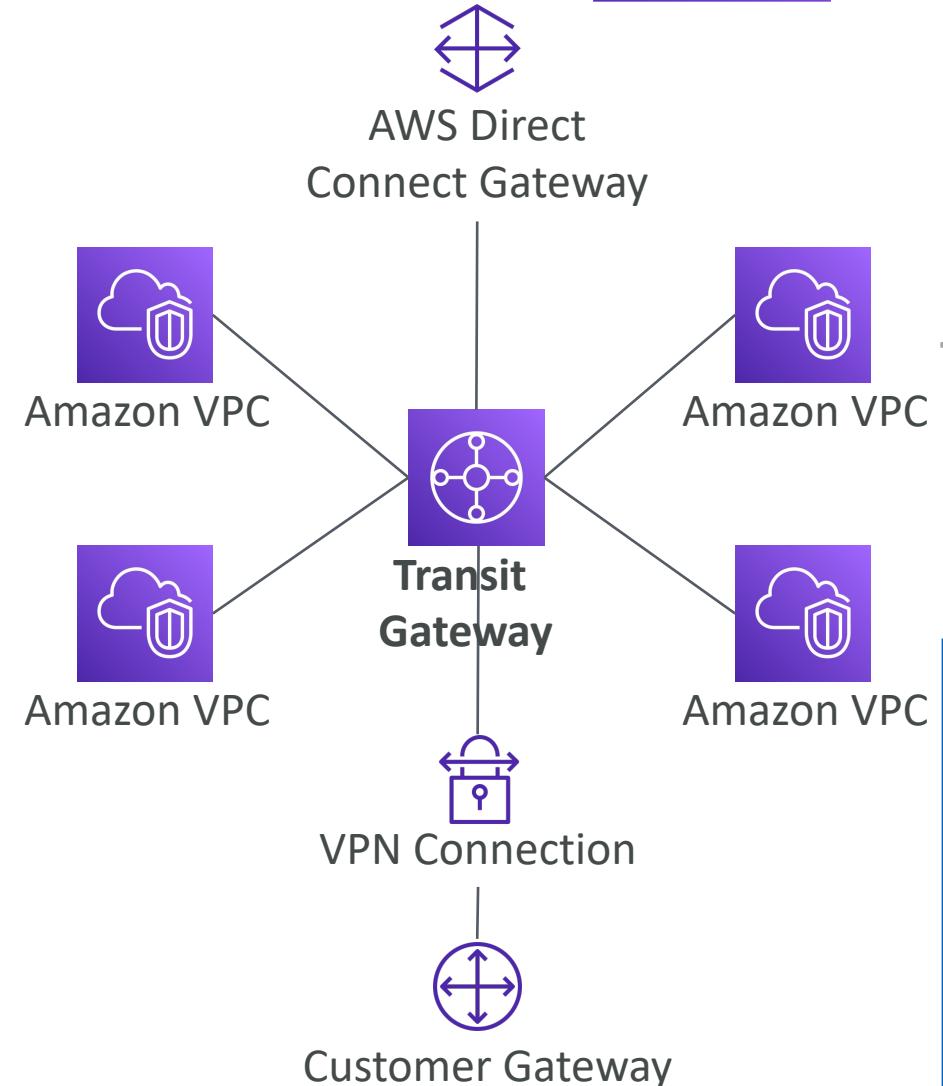
Network topologies can become complicated



Transit Gateway

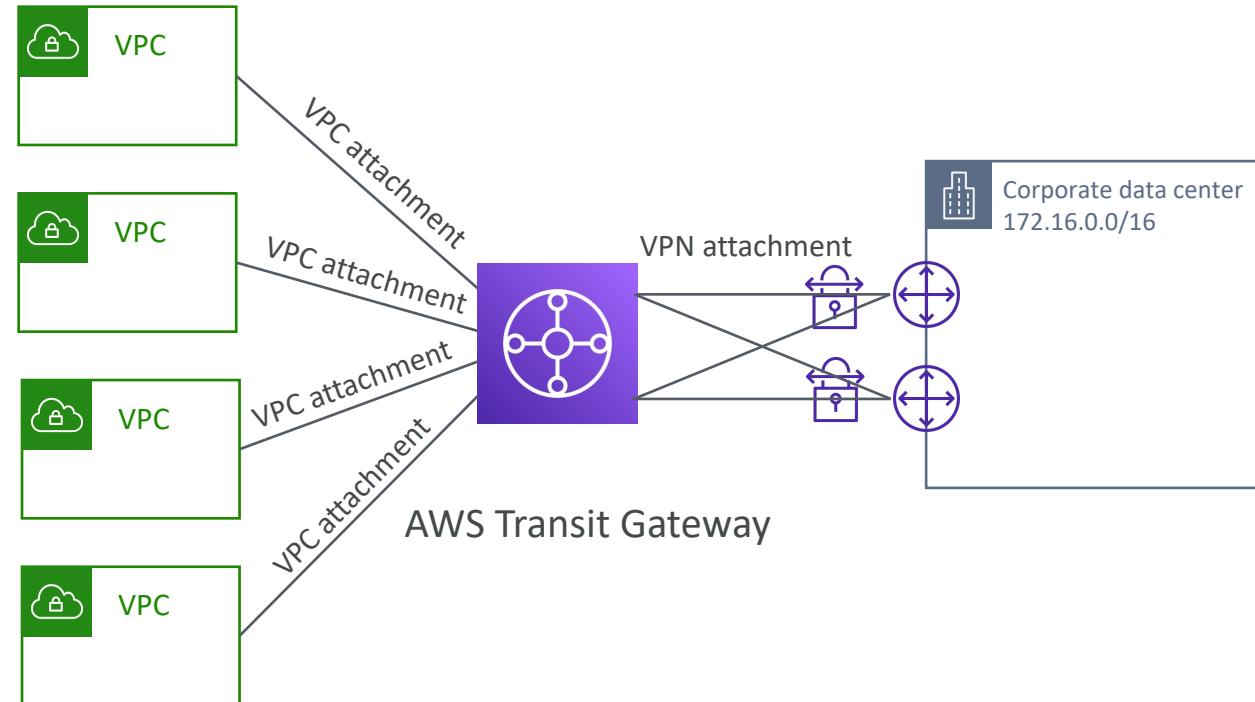


- For having transitive peering between thousands of VPC and on-premises, hub-and-spoke (star) connection
- Regional resource, can work cross-region
- Share cross-account using Resource Access Manager (RAM)
- You can peer Transit Gateways across regions
- Route Tables: limit which VPC can talk with other VPC
- Works with Direct Connect Gateway, VPN connections
- Supports IP Multicast (not supported by any other AWS service)



Transit Gateway: Site-to-Site VPN ECMP

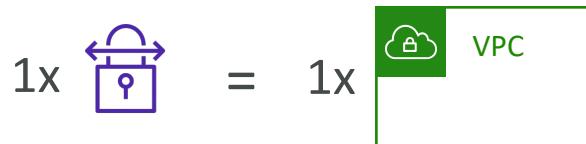
- ECMP = Equal-cost multi-path routing
- Routing strategy to allow to forward a packet over multiple best path
- Use case: create multiple Site-to-Site VPN connections to increase the bandwidth of your connection to AWS



Transit Gateway: throughput with ECMP



VPN to virtual private gateway



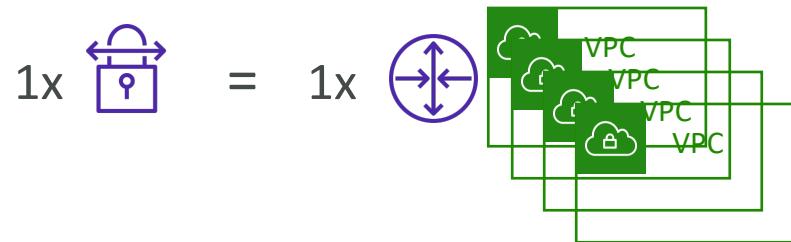
1x = 1.25 Gbps



VPN connection
(2 tunnels)



VPN to transit gateway



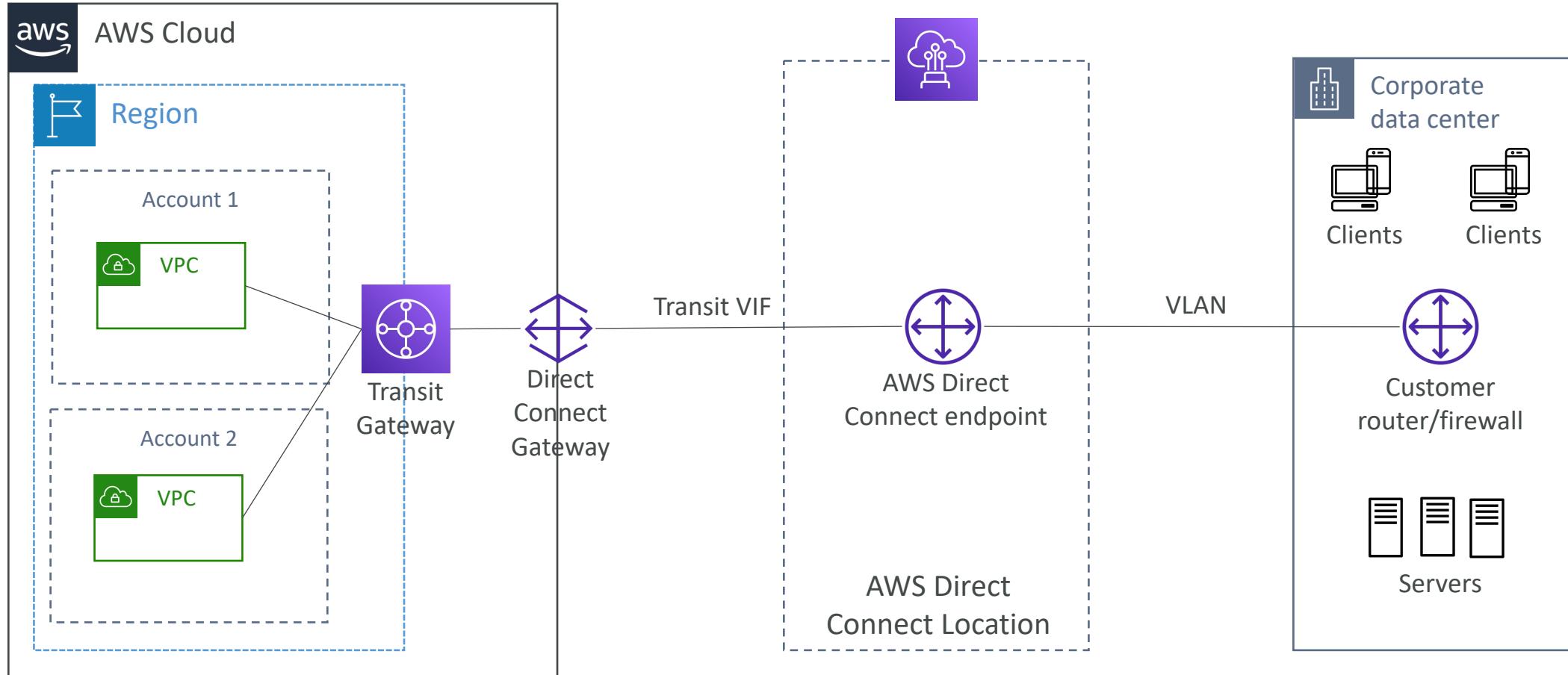
1x = 2.5 Gbps (ECMP) – 2 tunnels used

2x = 5.0 Gbps (ECMP)

3x = 7.5 Gbps (ECMP)

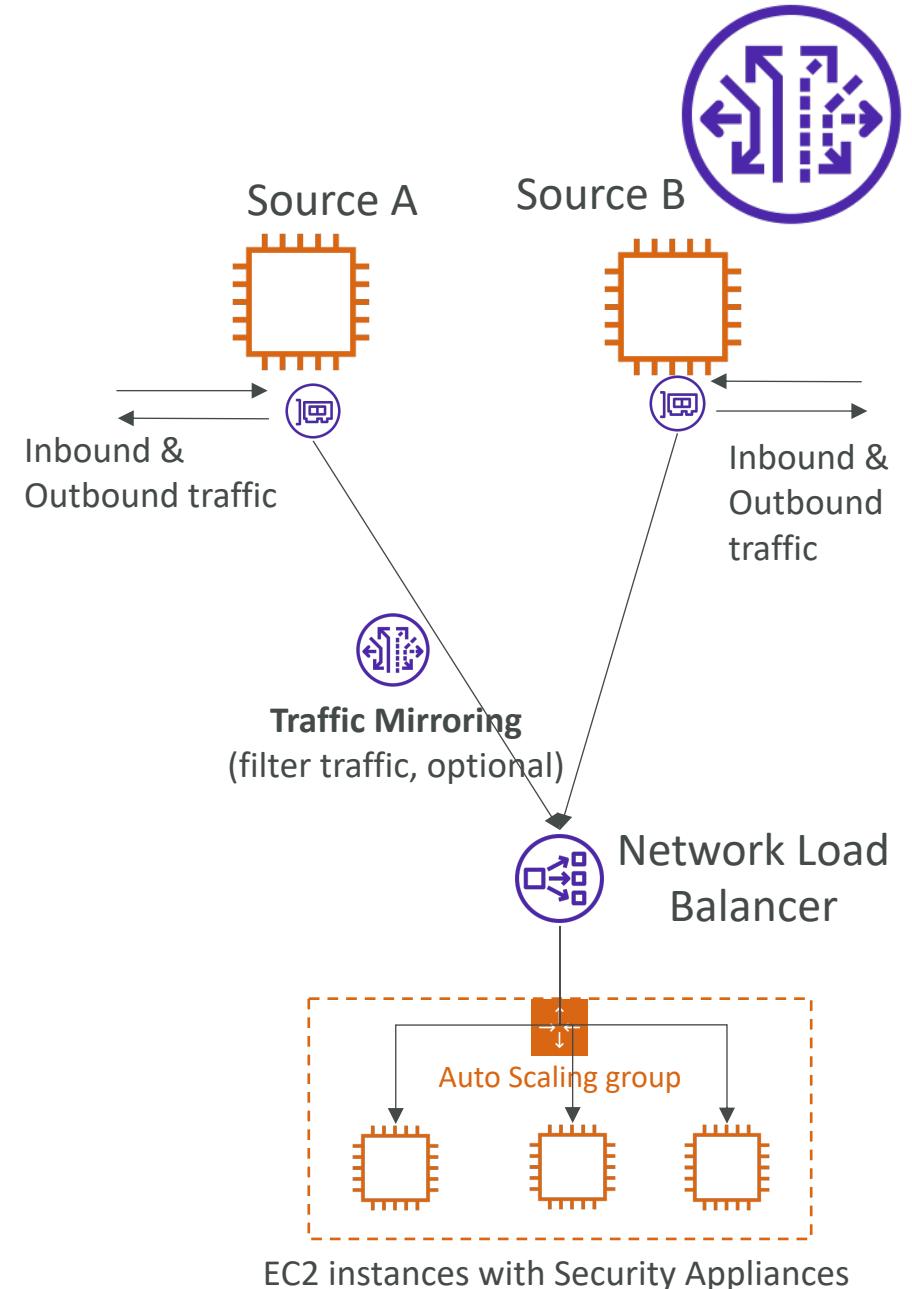
+\$\$ per GB of TGW
processed data

Transit Gateway – Share Direct Connect between multiple accounts



VPC – Traffic Mirroring

- Allows you to capture and inspect network traffic in your VPC
- Route the traffic to security appliances that you manage
- Capture the traffic
 - From (Source) – ENIs
 - To (Targets) – an ENI or a Network Load Balancer
- Capture all packets or capture the packets of your interest (optionally, truncate packets)
- Source and Target can be in the same VPC or different VPCs (VPC Peering)
- Use cases: content inspection, threat monitoring, troubleshooting, ...

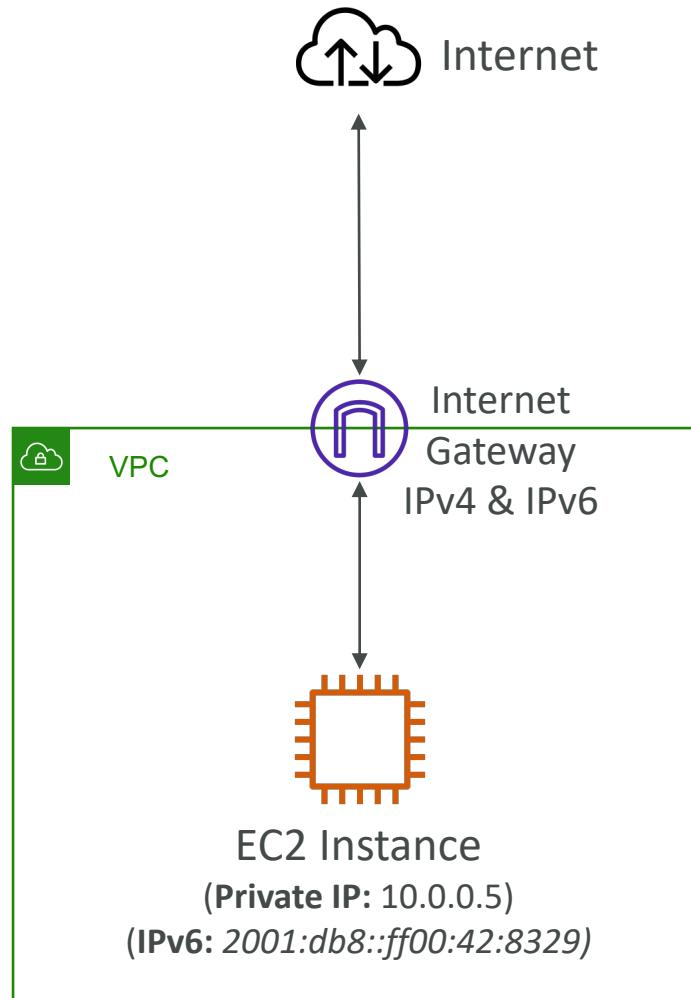


What is IPv6?

- IPv4 designed to provide 4.3 Billion addresses (they'll be exhausted soon)
- IPv6 is the successor of IPv4
- IPv6 is designed to provide 3.4×10^{38} unique IP addresses
- Every IPv6 address is public and Internet-routable (no private range)
- Format → x.x.x.x.x.x.x.x (x is hexadecimal, range can be from 0000 to ffff)
- Examples:
 - 2001:db8:3333:4444:5555:6666:7777:8888
 - 2001:db8:3333:4444:cccc:dddd:eeee:ffff
 - :: → all 8 segments are zero
 - 2001:db8:: → the last 6 segments are zero
 - ::1234:5678 → the first 6 segments are zero
 - 2001:db8::1234:5678 → the middle 4 segments are zero

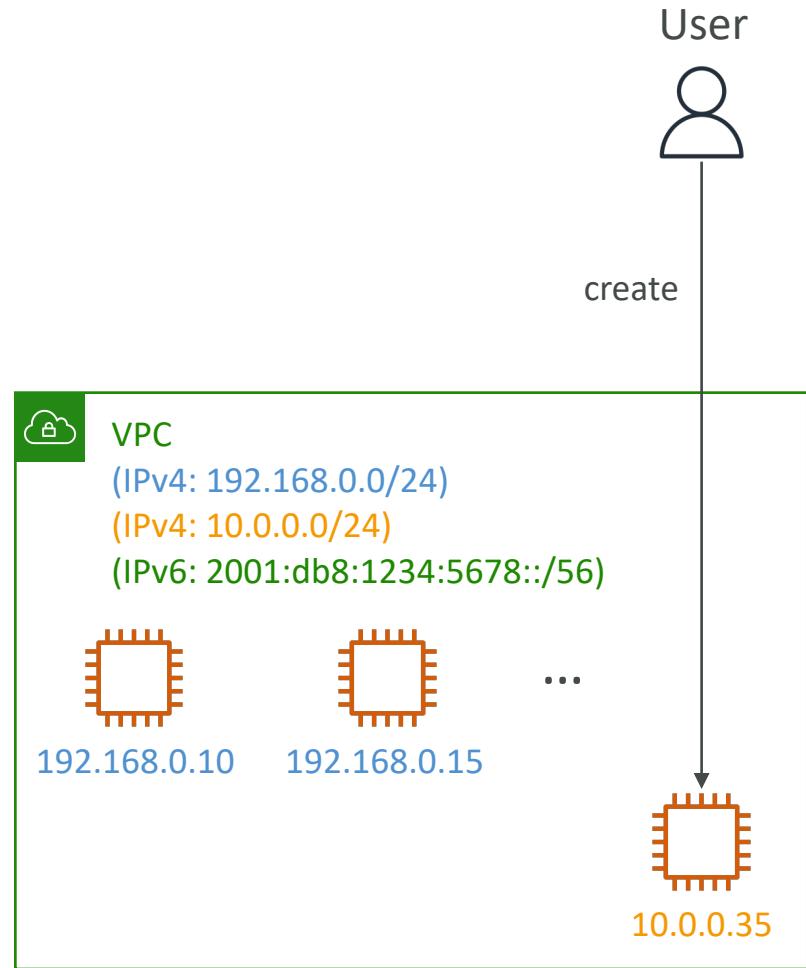
IPv6 in VPC

- IPv4 cannot be disabled for your VPC and subnets
- You can enable IPv6 (they're public IP addresses) to operate in dual-stack mode
- Your EC2 instances will get at least a private internal IPv4 and a public IPv6
- They can communicate using either IPv4 or IPv6 to the internet through an Internet Gateway

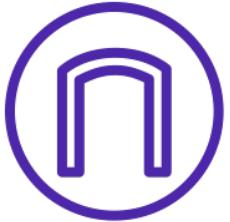


IPv6 Troubleshooting

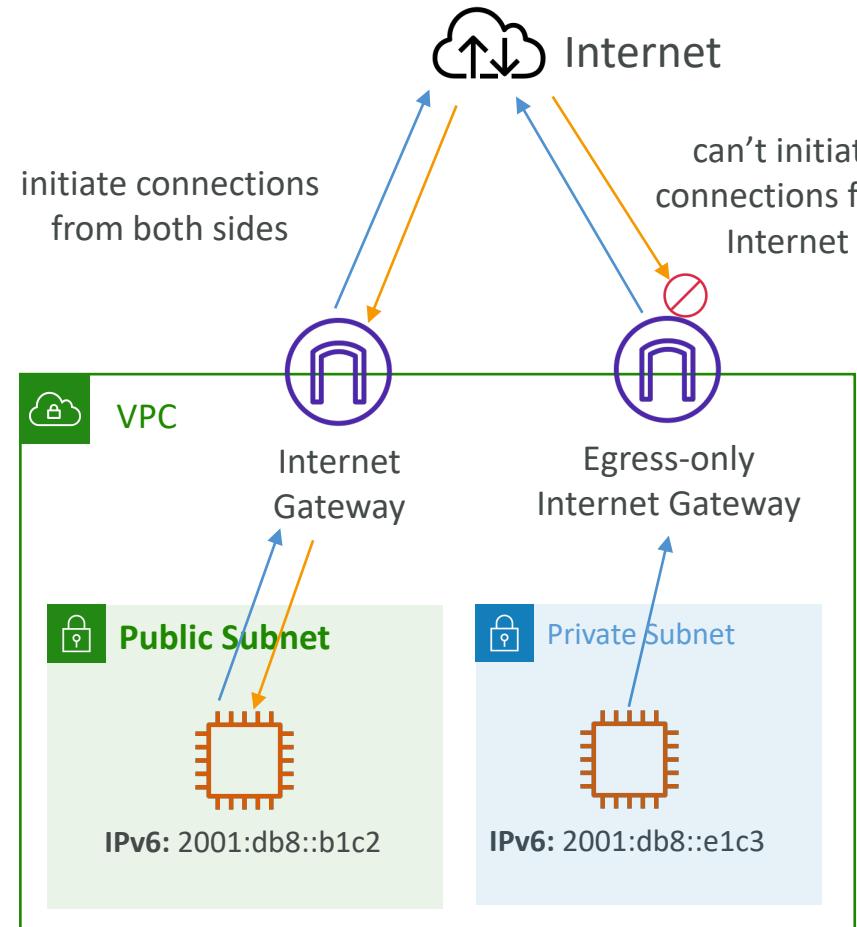
- IPv4 cannot be disabled for your VPC and subnets
- So, if you cannot launch an EC2 instance in your subnet
 - It's not because it cannot acquire an IPv6 (the space is very large)
 - It's because there are no available IPv4 in your subnet
- Solution: create a new IPv4 CIDR in your subnet



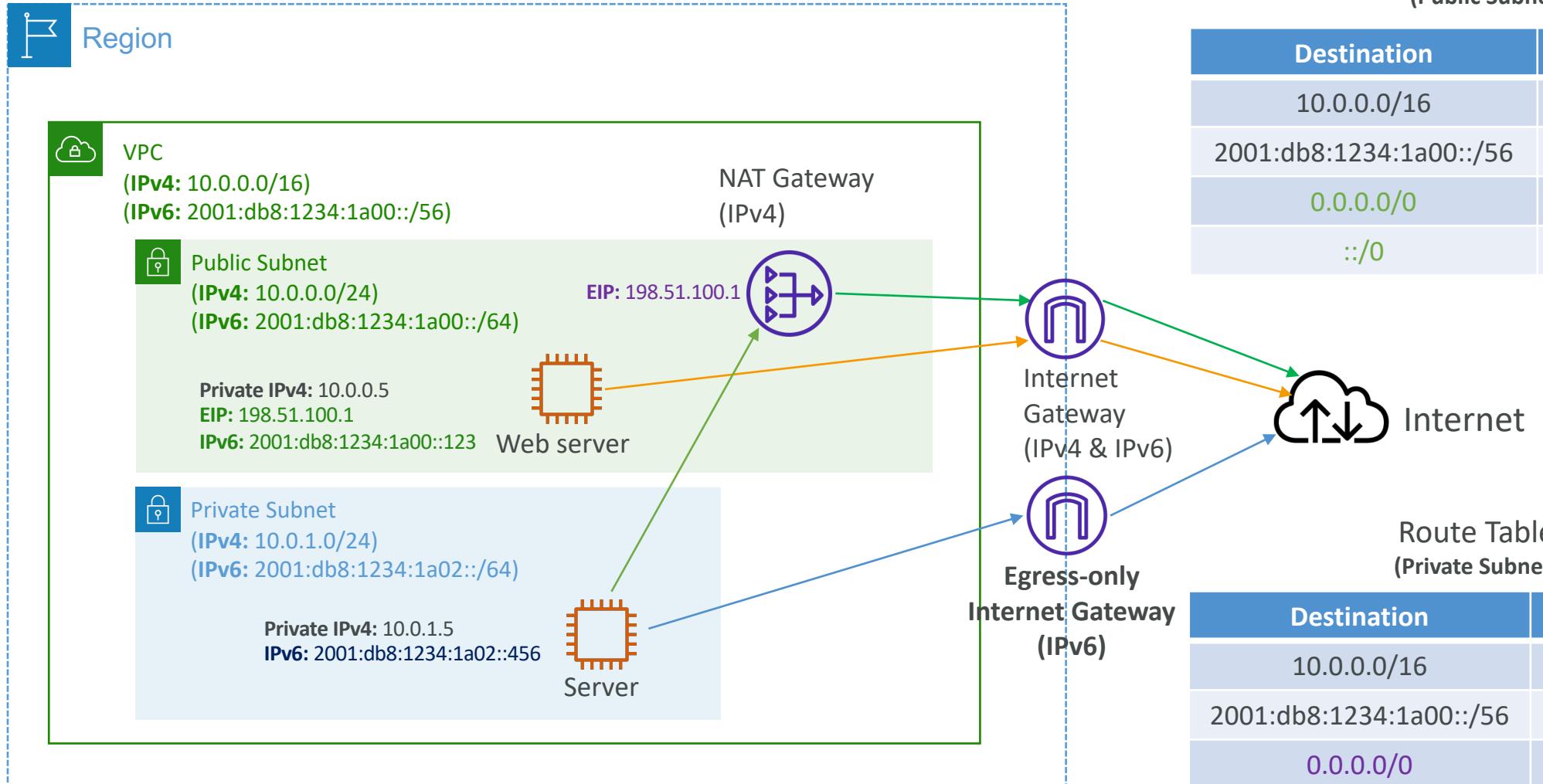
Egress-only Internet Gateway



- Used for IPv6 only
- (similar to a NAT Gateway but for IPv6)
- Allows instances in your VPC outbound connections over IPv6 while preventing the internet to initiate an IPv6 connection to your instances
- You must update the Route Tables



IPv6 Routing



VPC Section Summary (1/3)

- CIDR – IP Range
- VPC – Virtual Private Cloud => we define a list of IPv4 & IPv6 CIDR
- Subnets – tied to an AZ, we define a CIDR
- Internet Gateway – at the VPC level, provide IPv4 & IPv6 Internet Access
- Route Tables – must be edited to add routes from subnets to the IGW,VPC Peering Connections,VPC Endpoints, ...
- Bastion Host – public EC2 instance to SSH into, that has SSH connectivity to EC2 instances in private subnets
- NAT Instances – gives Internet access to EC2 instances in private subnets. Old, must be setup in a public subnet, disable Source / Destination check flag
- NAT Gateway – managed by AWS, provides scalable Internet access to private EC2 instances, IPv4 only

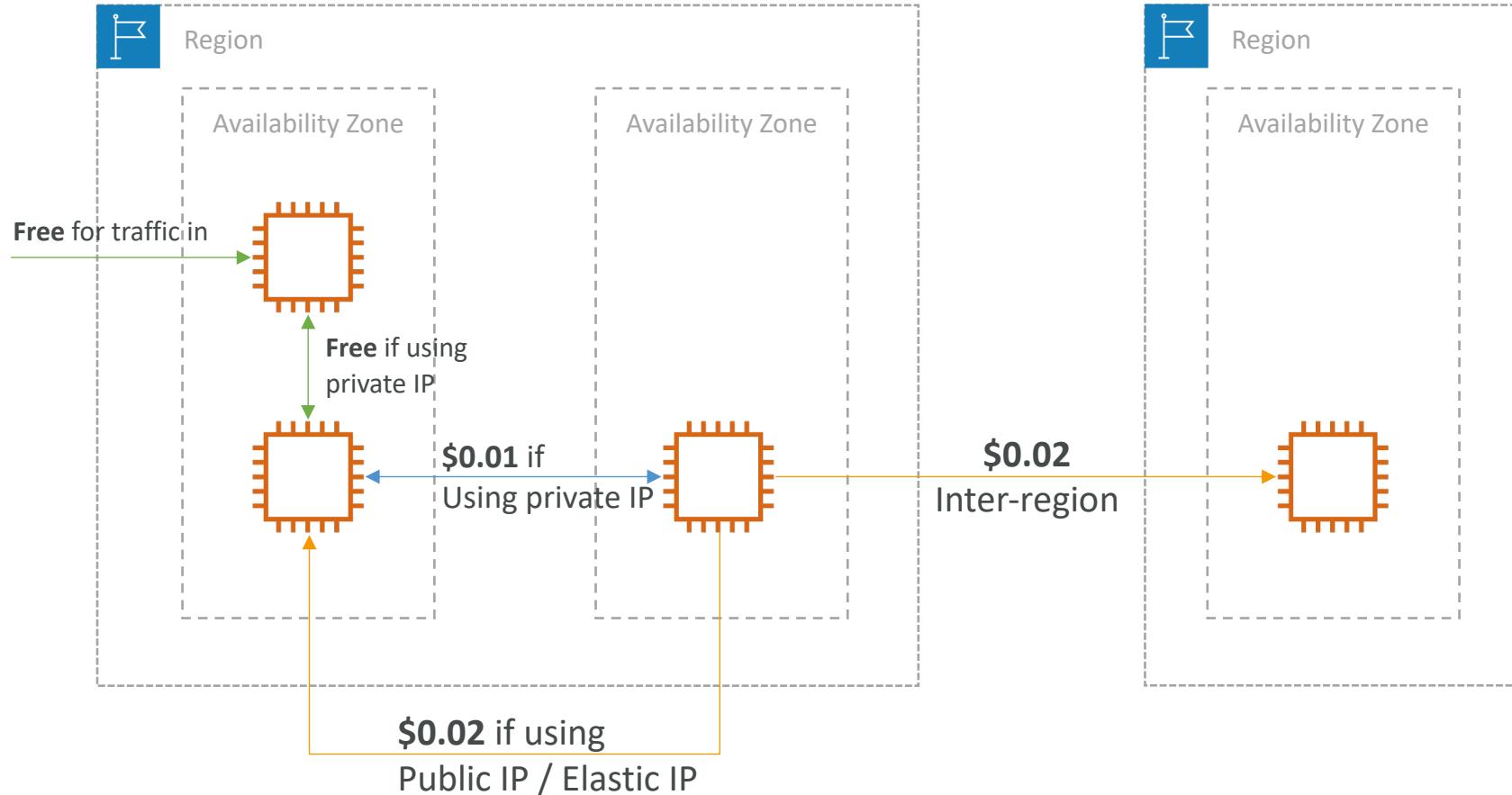
VPC Section Summary (2/3)

- **NACL** – stateless, subnet rules for inbound and outbound, don't forget Ephemeral Ports
- **Security Groups** – stateful, operate at the EC2 instance level
- **VPC Peering** – connect two VPCs with non overlapping CIDR, non-transitive
- **VPC Endpoints** – provide private access to AWS Services (S3, DynamoDB, CloudFormation, SSM) within a VPC
- **VPC Flow Logs** – can be setup at the VPC / Subnet / ENI Level, for ACCEPT and REJECT traffic, helps identifying attacks, analyze using Athena or CloudWatch Logs Insights
- **Site-to-Site VPN** – setup a Customer Gateway on DC, a Virtual Private Gateway on VPC, and site-to-site VPN over public Internet
- **AWS VPN CloudHub** – hub-and-spoke VPN model to connect your sites

VPC Section Summary (3/3)

- **Direct Connect** – setup a Virtual Private Gateway on VPC, and establish a direct private connection to an AWS Direct Connect Location
- **Direct Connect Gateway** – setup a Direct Connect to many VPCs in different AWS regions
- **AWS PrivateLink / VPC Endpoint Services:**
 - Connect services privately from your service VPC to customers VPC
 - Doesn't need VPC Peering, public Internet, NAT Gateway, Route Tables
 - Must be used with Network Load Balancer & ENI
- **ClassicLink** – connect EC2-Classic EC2 instances privately to your VPC
- **Transit Gateway** – transitive peering connections for VPC, VPN & DX
- **Traffic Mirroring** – copy network traffic from ENIs for further analysis
- **Egress-only Internet Gateway** – like a NAT Gateway, but for IPv6

Networking Costs in AWS per GB - Simplified

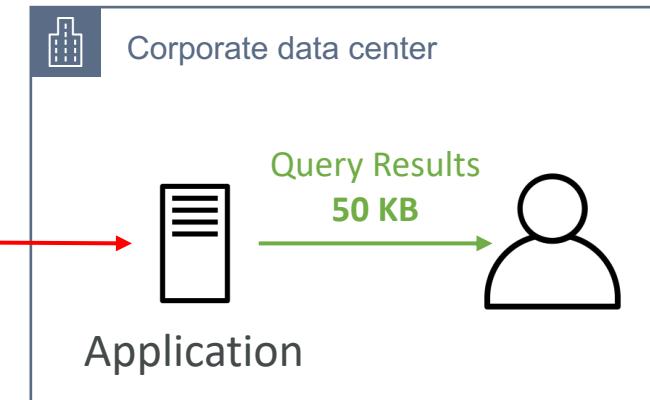
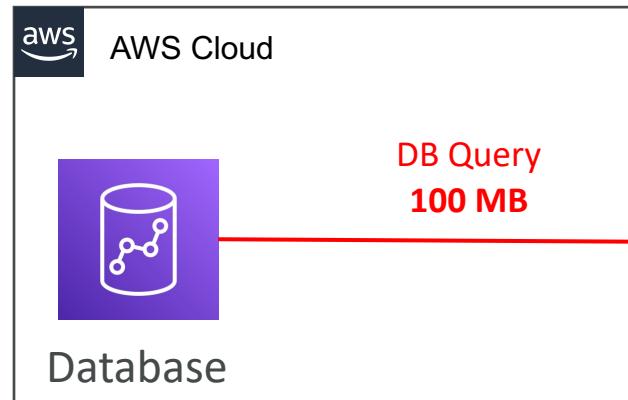


- Use Private IP instead of Public IP for good savings and better network performance
- Use same AZ for maximum savings (at the cost of high availability)

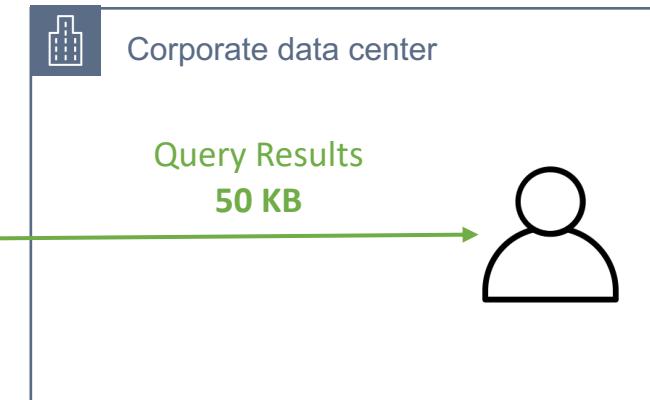
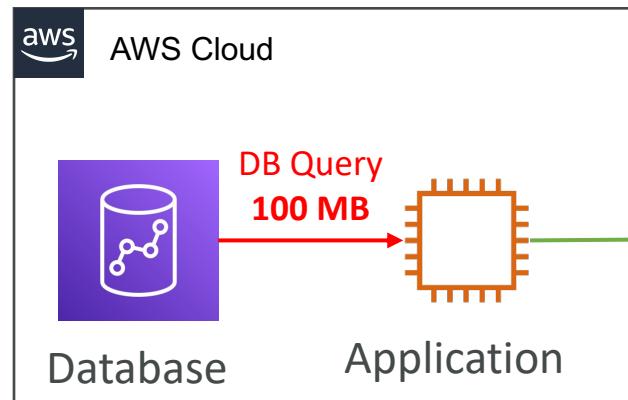
Minimizing egress traffic network cost

- Egress traffic: outbound traffic (from AWS to outside)
- Ingress traffic: inbound traffic - from outside to AWS (typically free)
- Try to keep as much internet traffic within AWS to minimize costs
- Direct Connect locations that are co-located in the same AWS Region result in lower cost for egress network

Egress cost is high

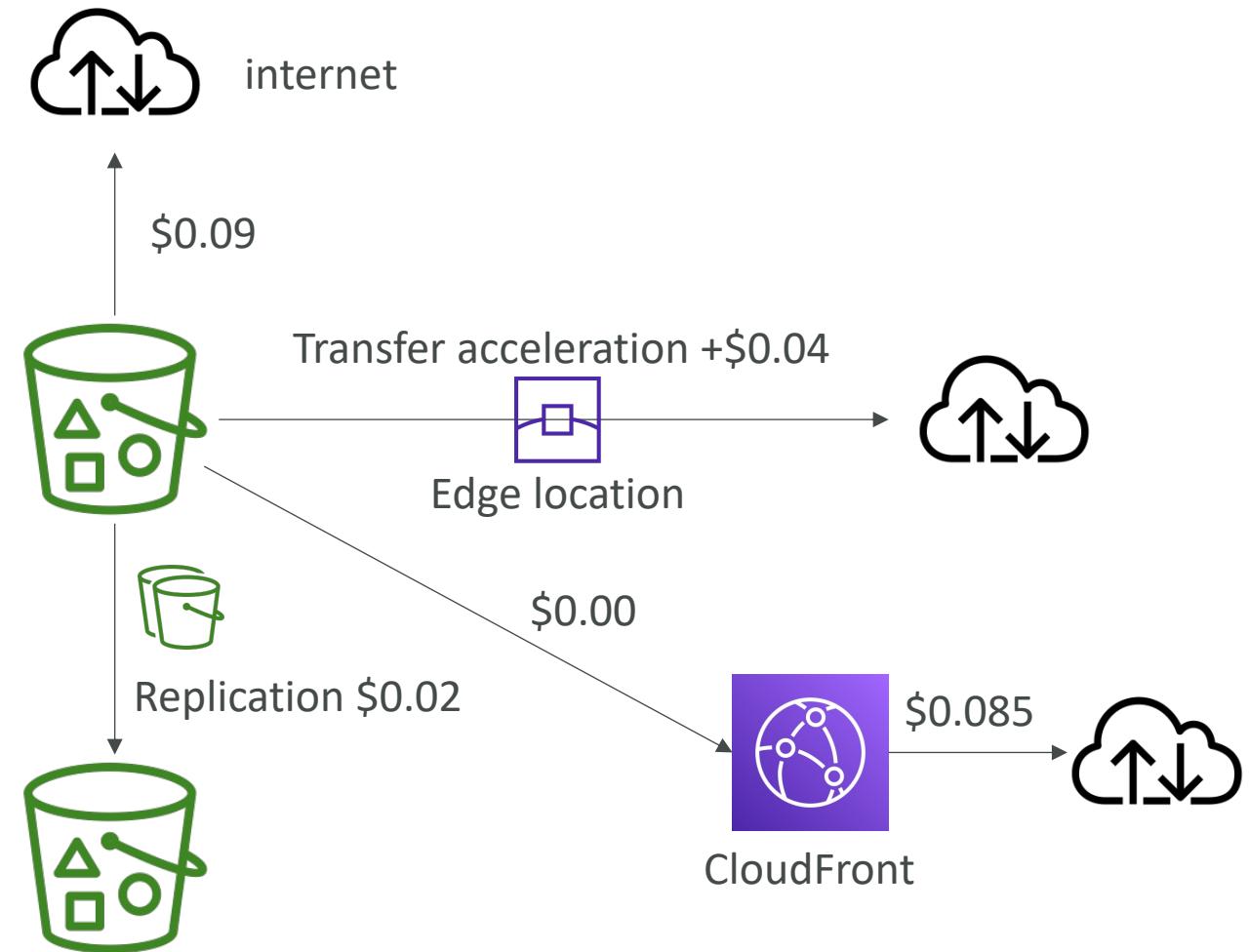


Egress cost is minimized



S3 Data Transfer Pricing – Analysis for USA

- S3 ingress: free
- S3 to Internet: \$0.09 per GB
- S3 Transfer Acceleration:
 - Faster transfer times (50 to 500% better)
 - Additional cost on top of Data Transfer Pricing: +\$0.04 to \$0.08 per GB
- S3 to CloudFront: \$0.00 per GB
- CloudFront to Internet: \$0.085 per GB (slightly cheaper than S3)
 - Caching capability (lower latency)
 - Reduce costs associated with S3 Requests Pricing (7x cheaper with CloudFront)
- S3 Cross Region Replication: \$0.02 per GB



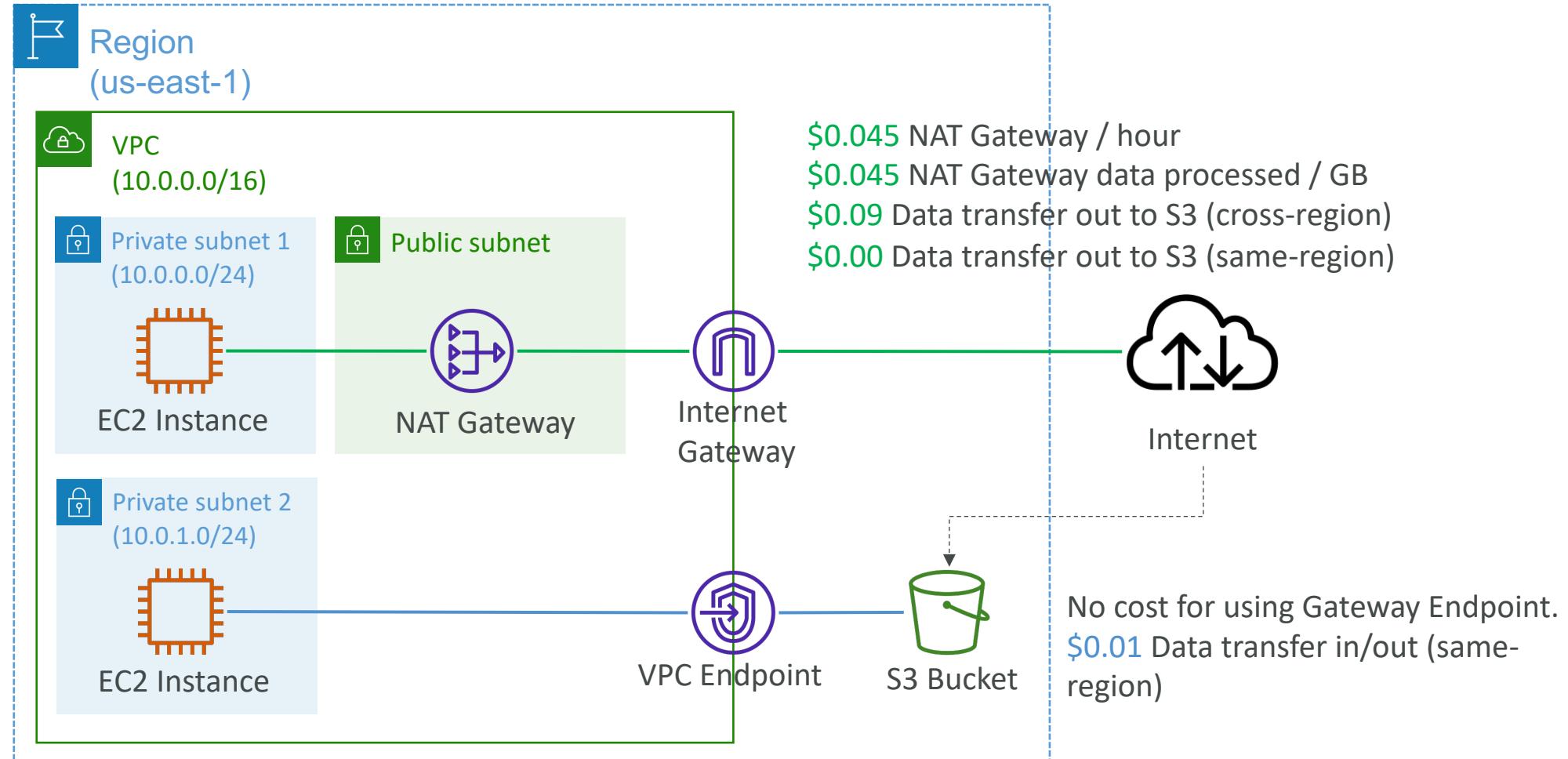
Pricing: NAT Gateway vs Gateway VPC Endpoint

Subnet 1 route table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	igw-id

Subnet 2 route table

Destination	Target
10.0.0.0/16	Local
pl-id for Amazon S3	vpce-id

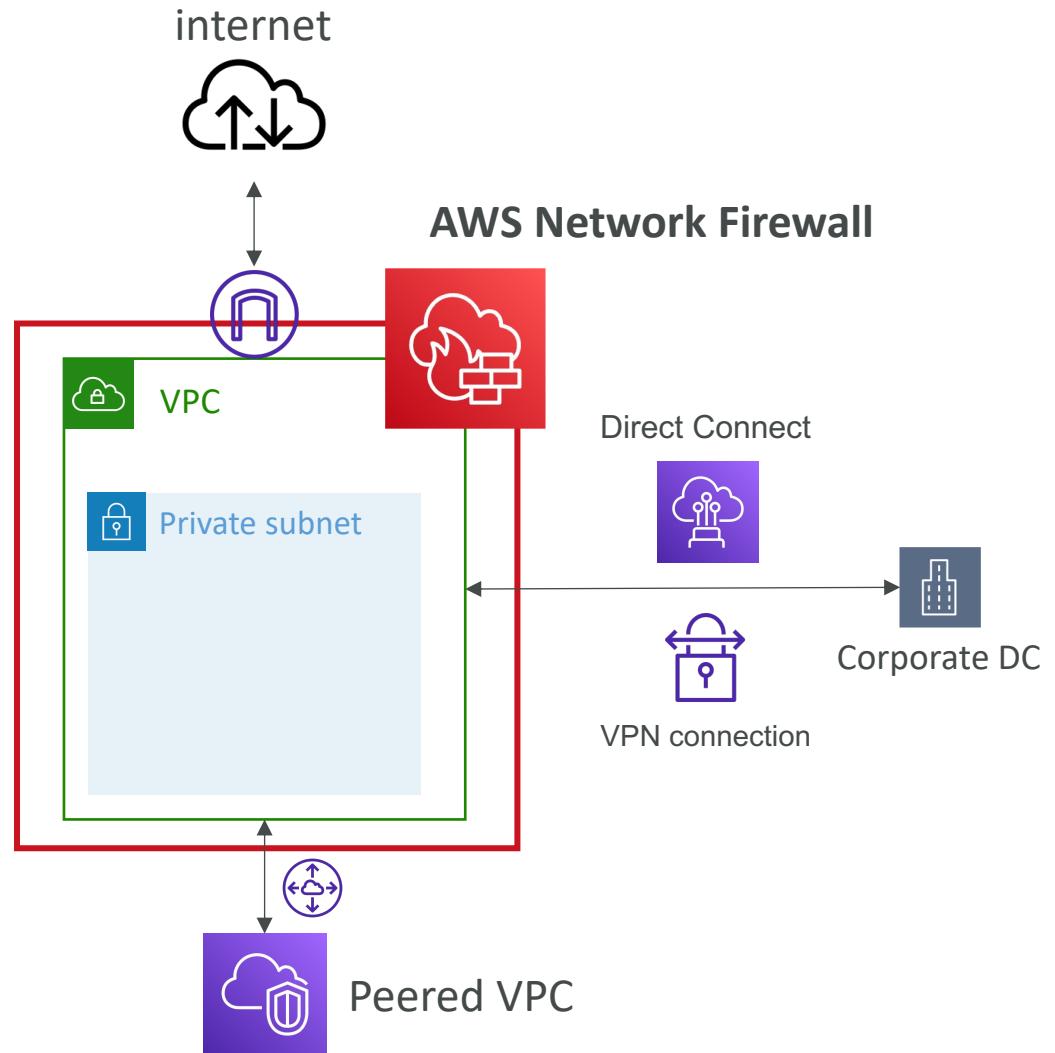


Network Protection on AWS

- To protect network on AWS, we've seen
 - Network Access Control Lists (NACLs)
 - Amazon VPC security groups
 - AWS WAF (protect against malicious requests)
 - AWS Shield & AWS Shield Advanced
 - AWS Firewall Manager (to manage them across accounts)
- But what if we want to protect in a sophisticated way our entire VPC?

AWS Network Firewall

- Protect your entire Amazon VPC
- From Layer 3 to Layer 7 protection
- Any direction, you can inspect
 - VPC to VPC traffic
 - Outbound to internet
 - Inbound from internet
 - To / from Direct Connect & Site-to-Site VPN
- Internally, the AWS Network Firewall uses the AWS Gateway Load Balancer
- Rules can be centrally managed cross-account by AWS Firewall Manager to apply to many VPCs





Network Firewall – Fine Grained Controls

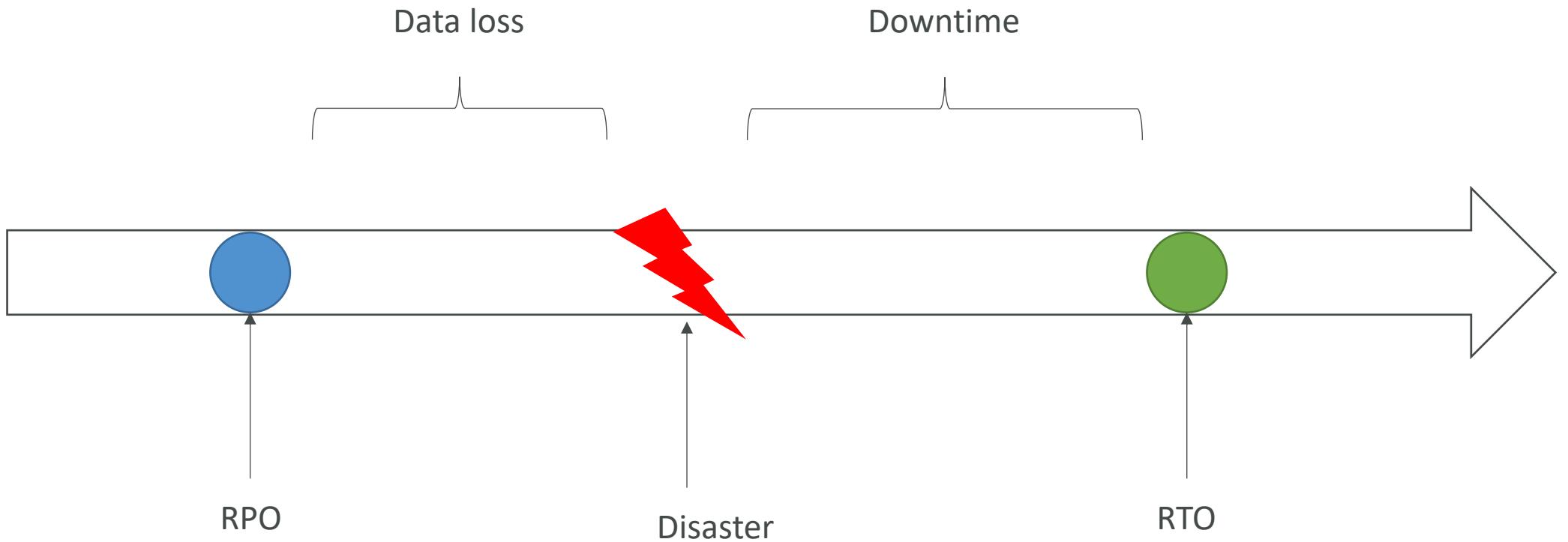
- Supports 1000s of rules
 - IP & port - example: 10,000s of IPs filtering
 - Protocol – example: block the SMB protocol for outbound communications
 - Stateful domain list rule groups: only allow outbound traffic to *.mycorp.com or third-party software repo
 - General pattern matching using regex
- **Traffic filtering:** Allow, drop, or alert for the traffic that matches the rules
- Active flow inspection to protect against network threats with intrusion-prevention capabilities (like Gateway Load Balancer, but all managed by AWS)
- Send logs of rule matches to Amazon S3, CloudWatch Logs, Kinesis Data Firehose

Disaster Recovery & Migrations

Disaster Recovery Overview

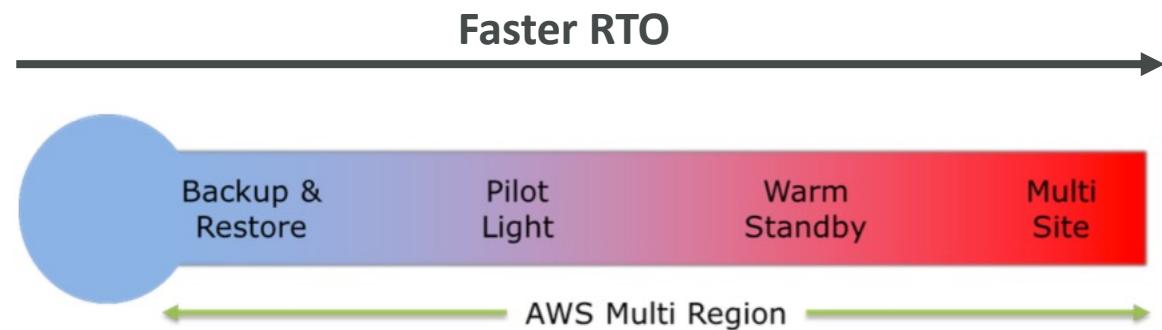
- Any event that has a negative impact on a company's business continuity or finances is a disaster
- Disaster recovery (DR) is about preparing for and recovering from a disaster
- What kind of disaster recovery?
 - On-premise => On-premise: traditional DR, and very expensive
 - On-premise => AWS Cloud: hybrid recovery
 - AWS Cloud Region A => AWS Cloud Region B
- Need to define two terms:
 - RPO: Recovery Point Objective
 - RTO: Recovery Time Objective

RPO and RTO

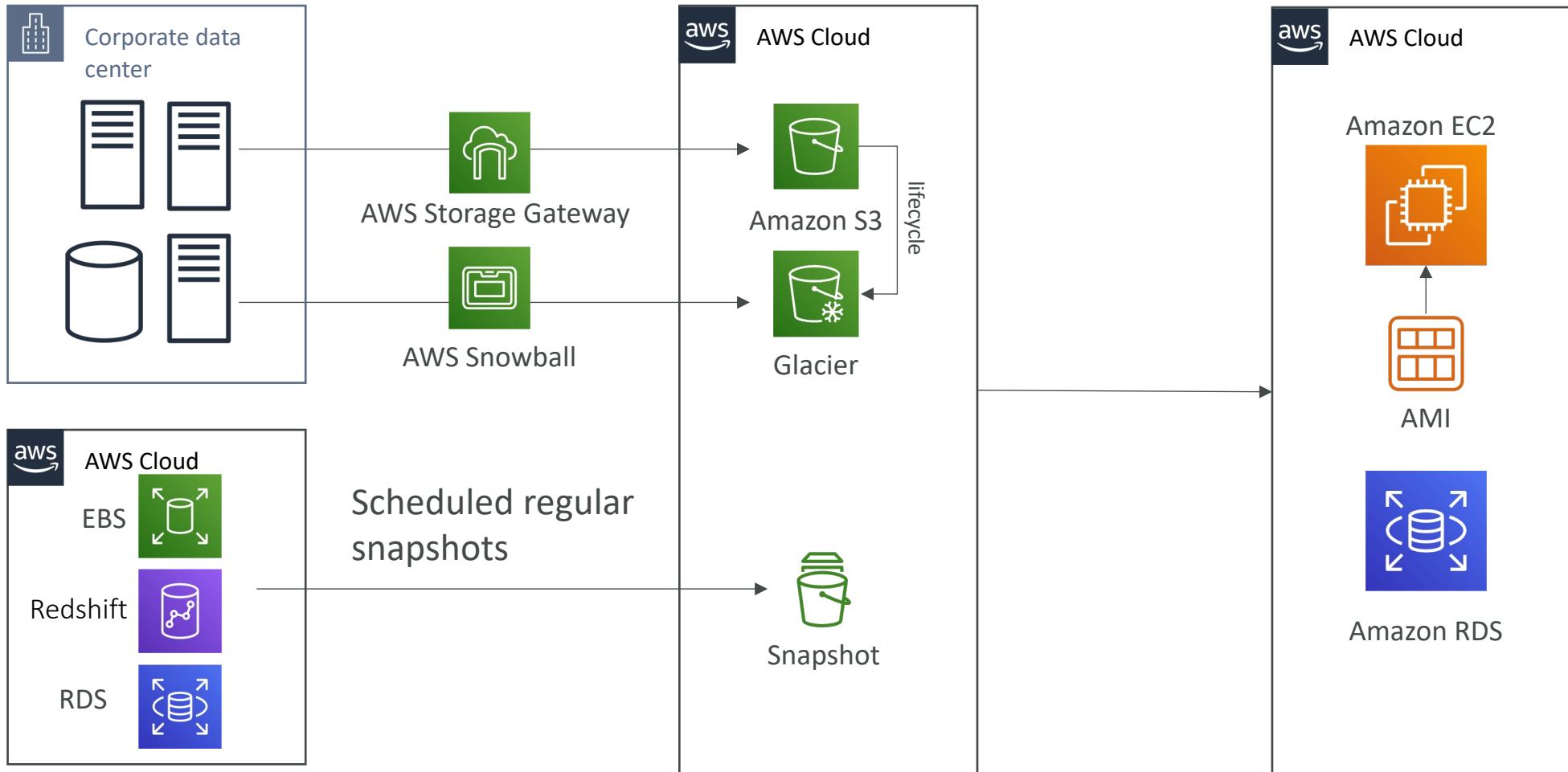


Disaster Recovery Strategies

- Backup and Restore
- Pilot Light
- Warm Standby
- Hot Site / Multi Site Approach

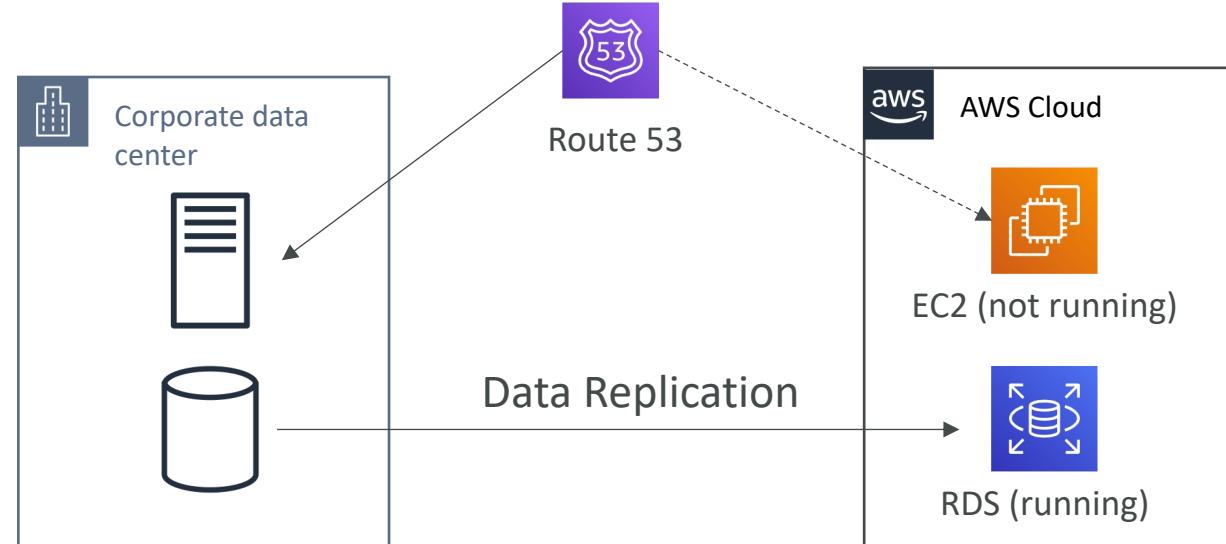


Backup and Restore (High RPO)



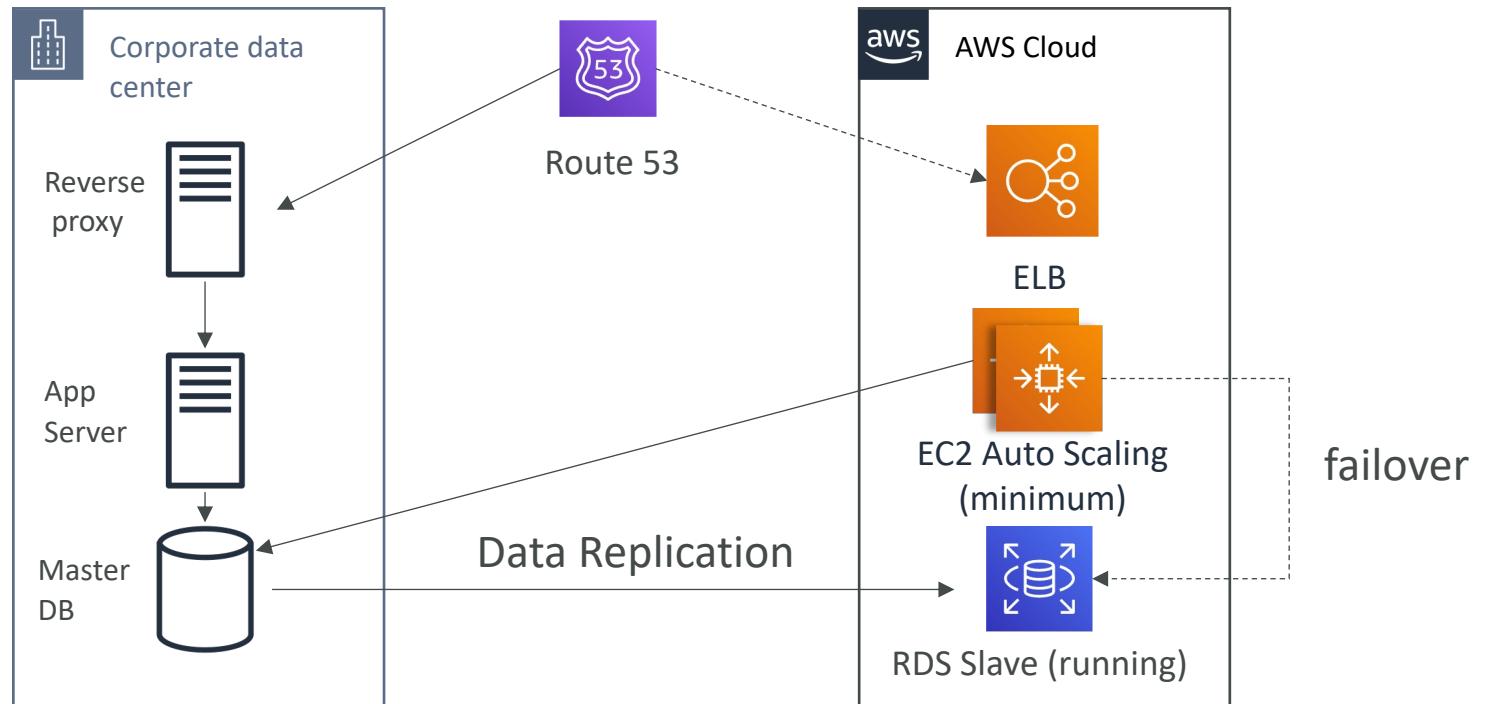
Disaster Recovery – Pilot Light

- A small version of the app is always running in the cloud
- Useful for the critical core (pilot light)
- Very similar to Backup and Restore
- Faster than Backup and Restore as critical systems are already up



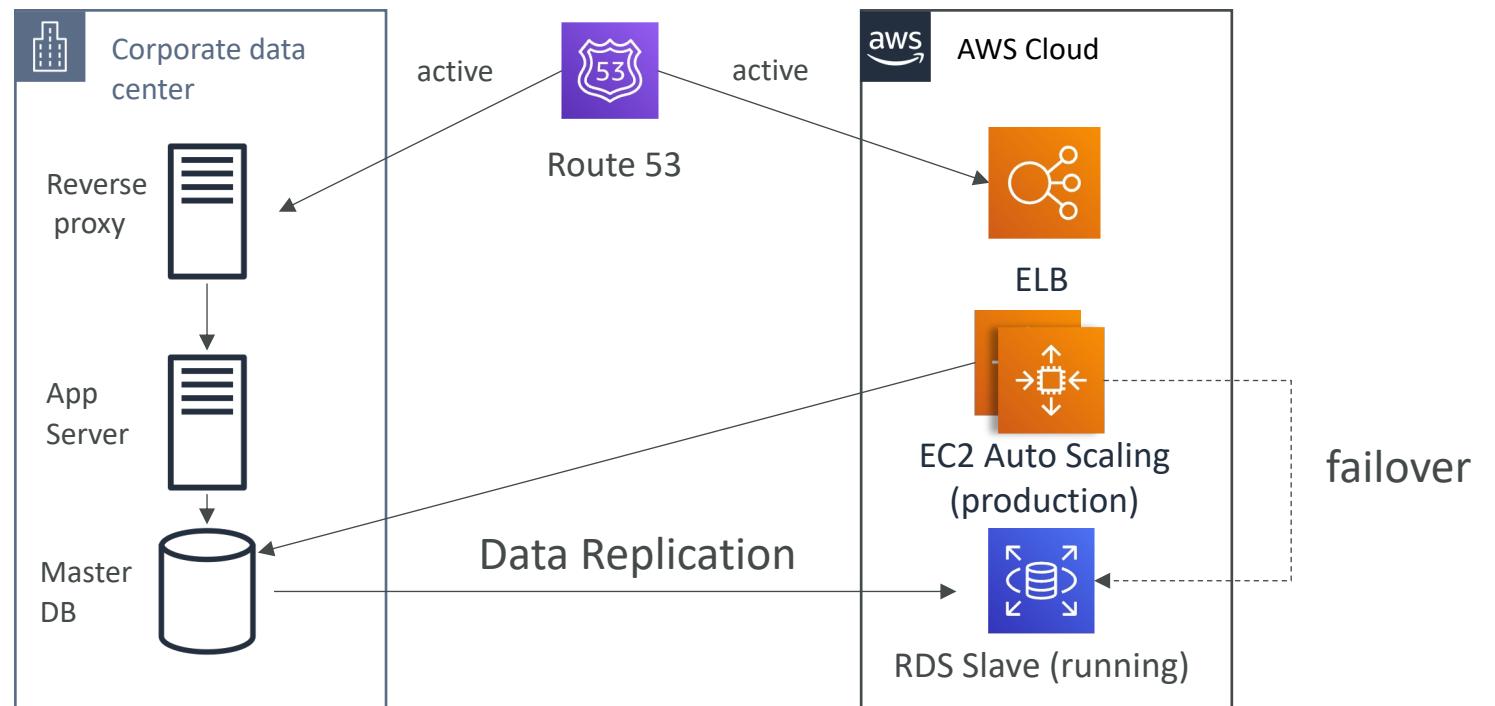
Warm Standby

- Full system is up and running, but at minimum size
- Upon disaster, we can scale to production load

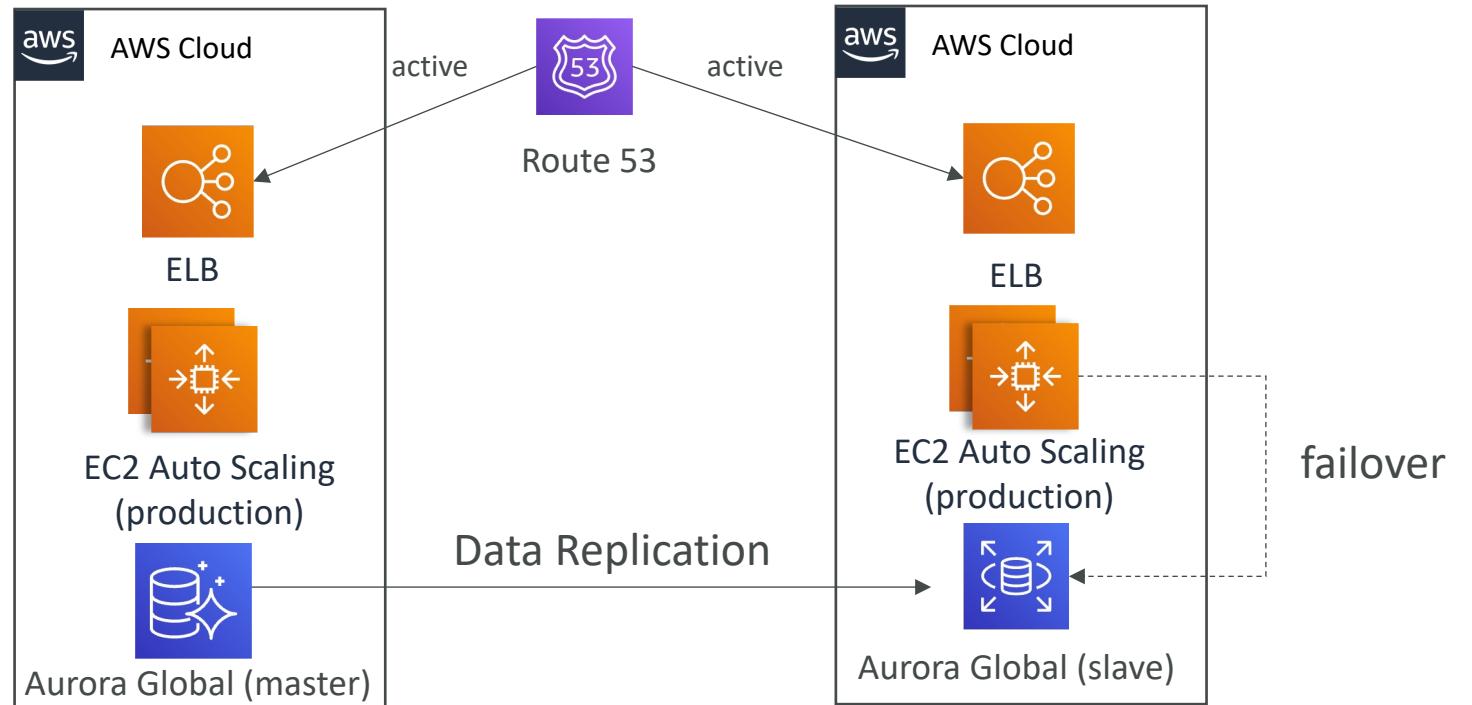


Multi Site / Hot Site Approach

- Very low RTO (minutes or seconds) – very expensive
- Full Production Scale is running AWS and On Premise



All AWS Multi Region



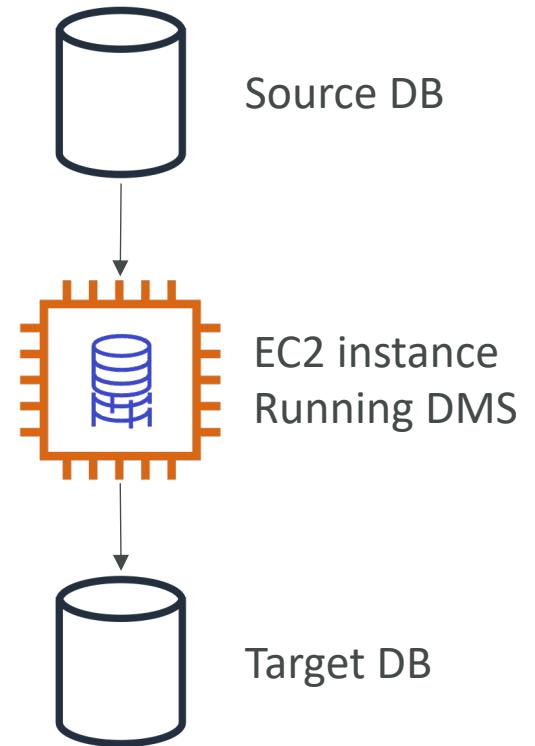
Disaster Recovery Tips

- **Backup**
 - EBS Snapshots, RDS automated backups / Snapshots, etc...
 - Regular pushes to S3 / S3 IA / Glacier, Lifecycle Policy, Cross Region Replication
 - From On-Premise: Snowball or Storage Gateway
- **High Availability**
 - Use Route53 to migrate DNS over from Region to Region
 - RDS Multi-AZ, ElastiCache Multi-AZ, EFS, S3
 - Site to Site VPN as a recovery from Direct Connect
- **Replication**
 - RDS Replication (Cross Region), AWS Aurora + Global Databases
 - Database replication from on-premises to RDS
 - Storage Gateway
- **Automation**
 - CloudFormation / Elastic Beanstalk to re-create a whole new environment
 - Recover / Reboot EC2 instances with CloudWatch if alarms fail
 - AWS Lambda functions for customized automations
- **Chaos**
 - Netflix has a “simian-army” randomly terminating EC2



DMS – Database Migration Service

- Quickly and securely migrate databases to AWS, resilient, self healing
- The source database remains available during the migration
- Supports:
 - Homogeneous migrations: ex Oracle to Oracle
 - Heterogeneous migrations: ex Microsoft SQL Server to Aurora
- Continuous Data Replication using CDC
- You must create an EC2 instance to perform the replication tasks



DMS Sources and Targets

SOURCES:

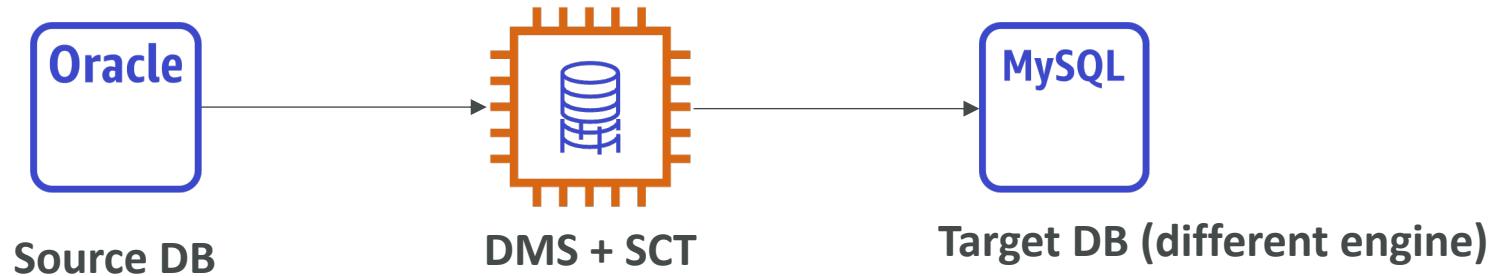
- On-Premises and EC2 instances databases: Oracle, MS SQL Server, MySQL, MariaDB, PostgreSQL, MongoDB, SAP, DB2
- Azure: *Azure SQL Database*
- Amazon RDS: all including Aurora
- Amazon S3
- DocumentDB

TARGETS:

- On-Premises and EC2 instances databases: Oracle, MS SQL Server, MySQL, MariaDB, PostgreSQL, SAP
- Amazon RDS
- Redshift, DynamoDB, S3
- OpenSearch Service
- Kinesis Data Streams
- Apache Kafka
- DocumentDB & Amazon Neptune
- Redis & Babelfish

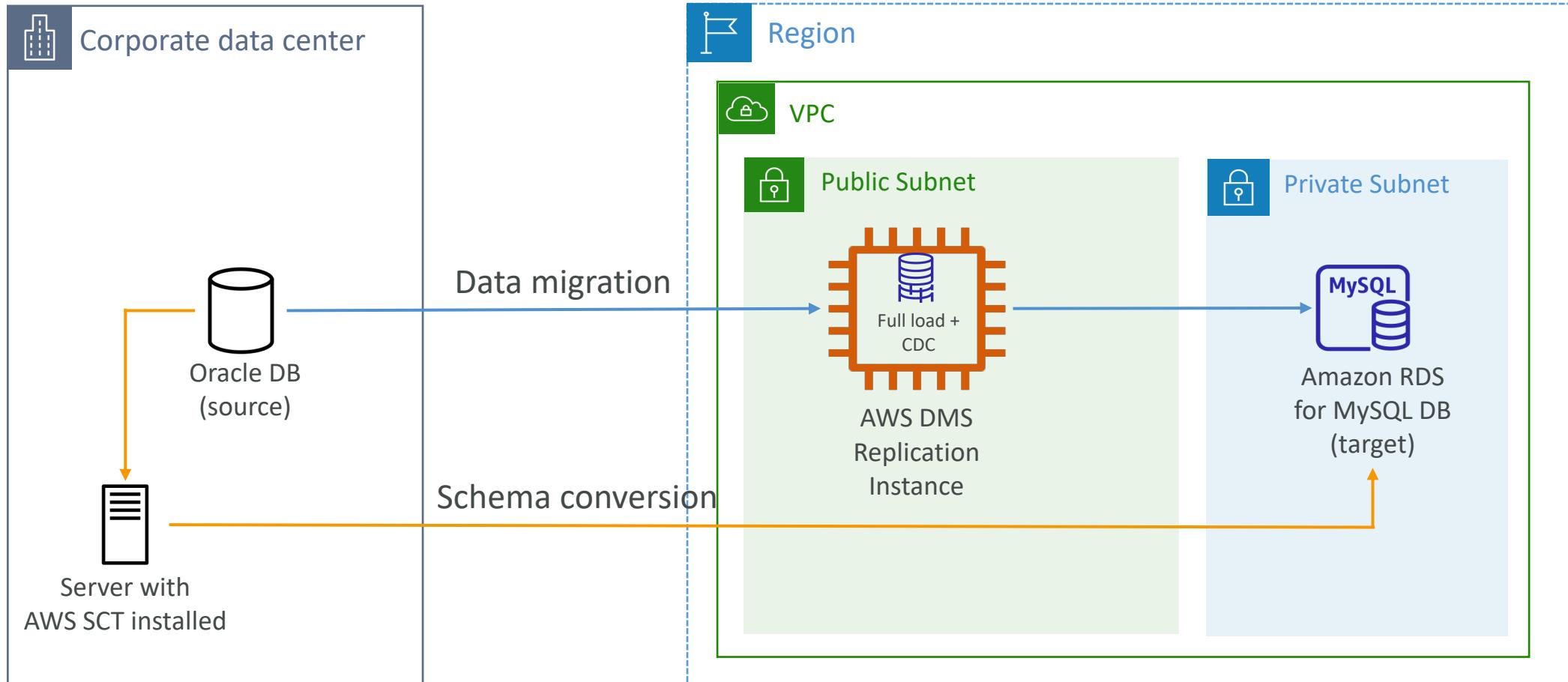
AWS Schema Conversion Tool (SCT)

- Convert your Database's Schema from one engine to another
- Example OLTP: (SQL Server or Oracle) to MySQL, PostgreSQL, Aurora
- Example OLAP: (Teradata or Oracle) to Amazon Redshift
- Prefer compute-intensive instances to optimize data conversions



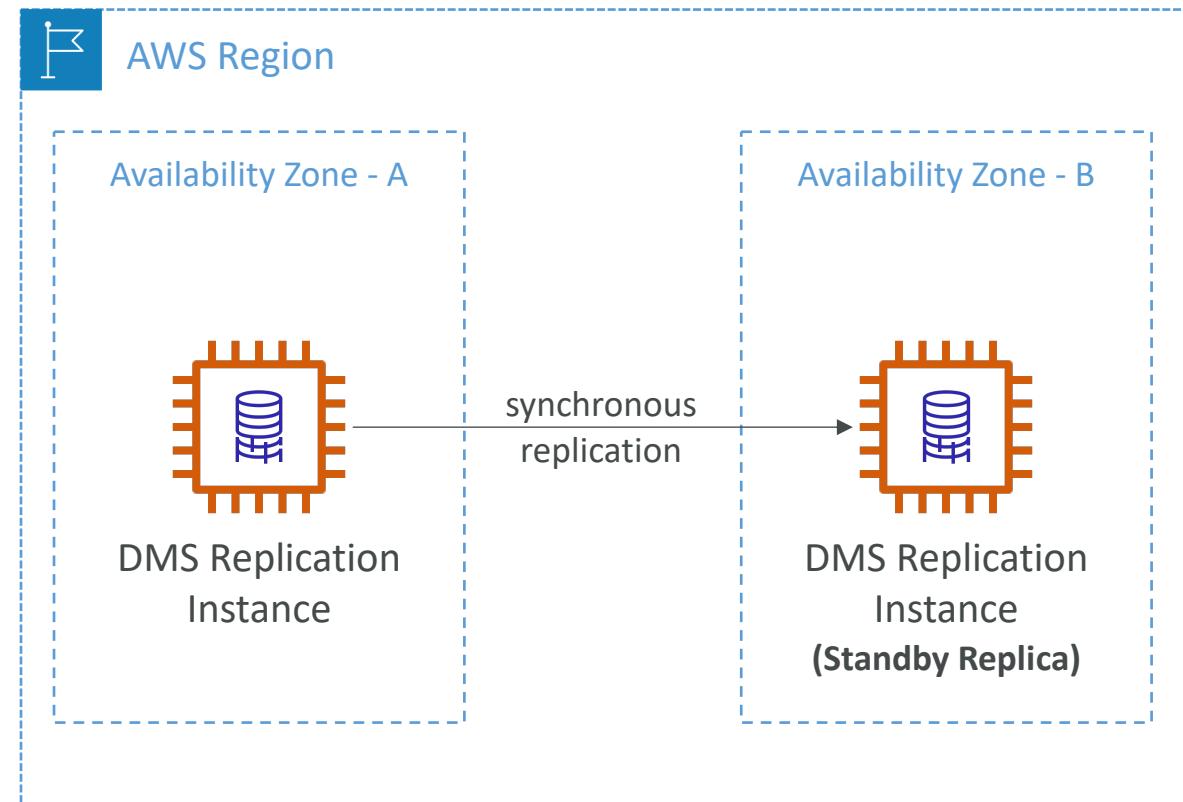
- You do not need to use SCT if you are migrating the same DB engine
 - Ex: On-Premise PostgreSQL => RDS PostgreSQL
 - The DB engine is still PostgreSQL (RDS is the platform)

DMS - Continuous Replication



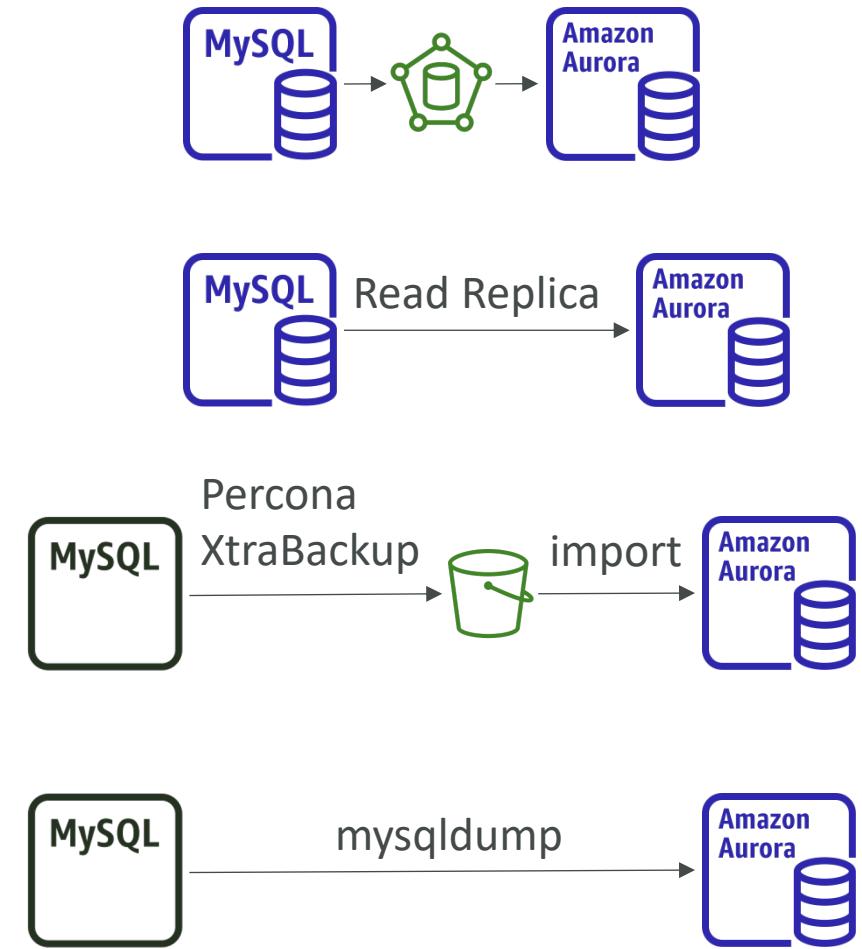
AWS DMS – Multi-AZ Deployment

- When Multi-AZ Enabled, DMS provisions and maintains a synchronously stand replica in a different AZ
- Advantages:
 - Provides Data Redundancy
 - Eliminates I/O freezes
 - Minimizes latency spikes



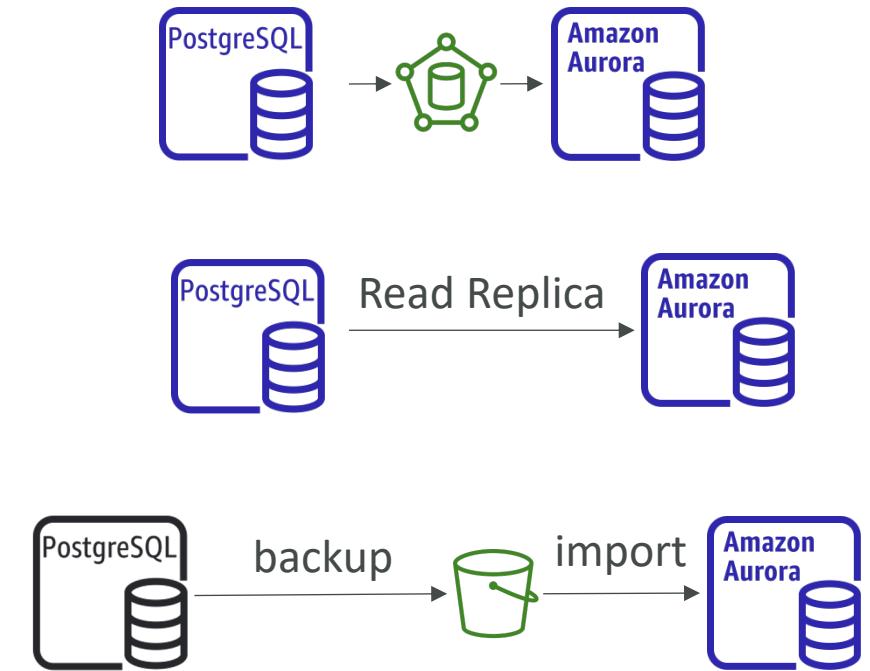
RDS & Aurora MySQL Migrations

- RDS MySQL to Aurora MySQL
 - Option 1: DB Snapshots from RDS MySQL restored as MySQL Aurora DB
 - Option 2: Create an Aurora Read Replica from your RDS MySQL, and when the replication lag is 0, promote it as its own DB cluster (can take time and cost \$)
- External MySQL to Aurora MySQL
 - Option 1:
 - Use Percona XtraBackup to create a file backup in Amazon S3
 - Create an Aurora MySQL DB from Amazon S3
 - Option 2:
 - Create an Aurora MySQL DB
 - Use the mysqldump utility to migrate MySQL into Aurora (slower than S3 method)
- Use DMS if both databases are up and running



RDS & Aurora PostgreSQL Migrations

- RDS PostgreSQL to Aurora PostgreSQL
 - Option 1: DB Snapshots from RDS PostgreSQL restored as PostgreSQL Aurora DB
 - Option 2: Create an Aurora Read Replica from your RDS PostgreSQL, and when the replication lag is 0, promote it as its own DB cluster (can take time and cost \$)
- External PostgreSQL to Aurora PostgreSQL
 - Create a backup and put it in Amazon S3
 - Import it using the aws_s3 Aurora extension
- Use DMS if both databases are up and running



On-Premise strategy with AWS

- Ability to download Amazon Linux 2 AMI as a VM (.iso format)
 - VMWare, KVM, VirtualBox (Oracle VM), Microsoft Hyper-V
- VM Import / Export
 - Migrate existing applications into EC2
 - Create a DR repository strategy for your on-premises VMs
 - Can export back the VMs from EC2 to on-premises
- AWS Application Discovery Service
 - Gather information about your on-premises servers to plan a migration
 - Server utilization and dependency mappings
 - Track with AWS Migration Hub
- AWS Database Migration Service (DMS)
 - replicate On-premise => AWS , AWS => AWS, AWS => On-premise
 - Works with various database technologies (Oracle, MySQL, DynamoDB, etc..)
- AWS Server Migration Service (SMS)
 - Incremental replication of on-premises live servers to AWS



AWS Backup

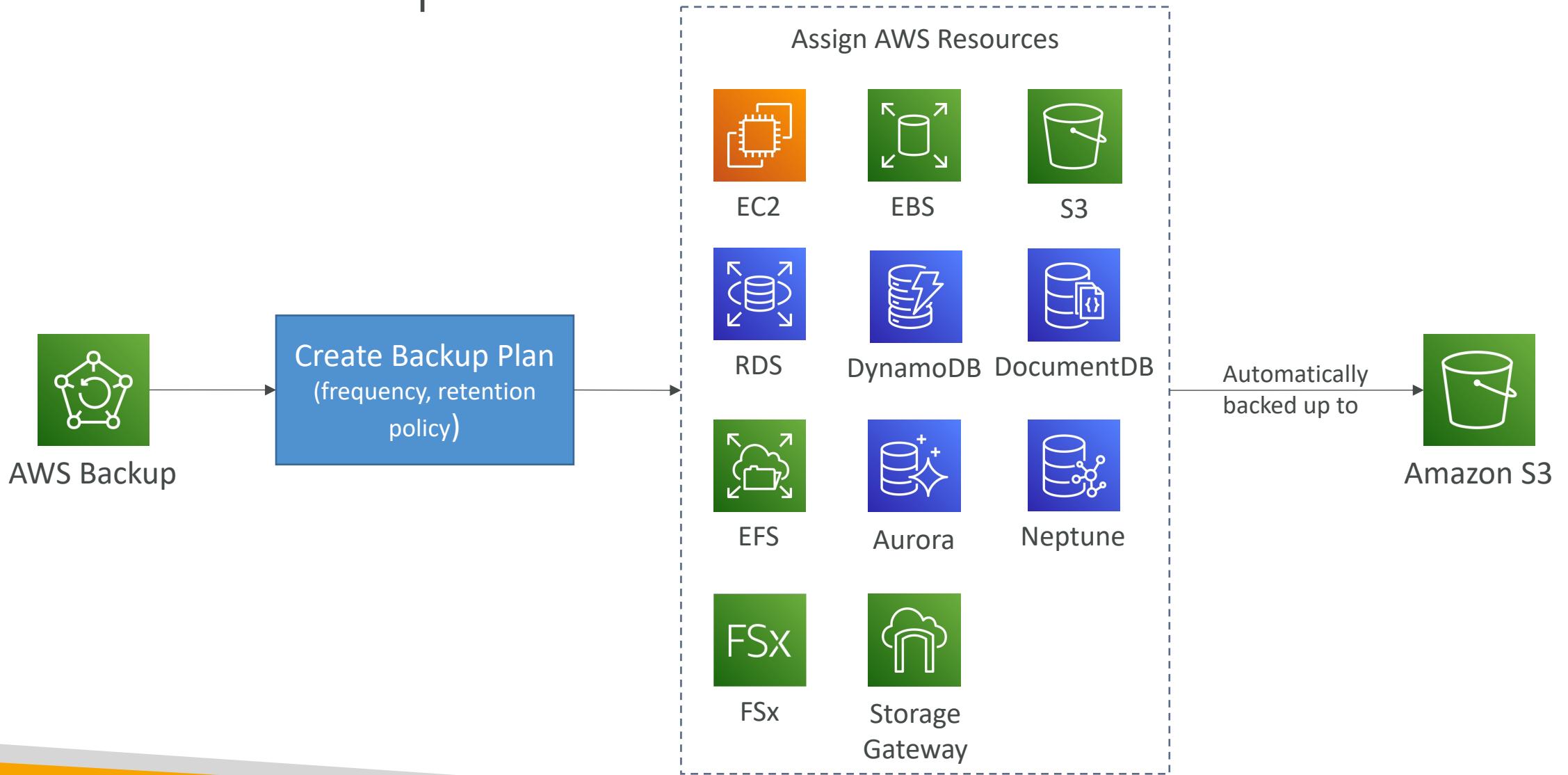
- Fully managed service
- Centrally manage and automate backups across AWS services
- No need to create custom scripts and manual processes
- Supported services:
 - Amazon EC2 / Amazon EBS
 - Amazon S3
 - Amazon RDS (all DBs engines) / Amazon Aurora / Amazon DynamoDB
 - Amazon DocumentDB / Amazon Neptune
 - Amazon EFS / Amazon FSx (Lustre & Windows File Server)
 - AWS Storage Gateway (Volume Gateway)
- Supports cross-region backups
- Supports cross-account backups

AWS Backup



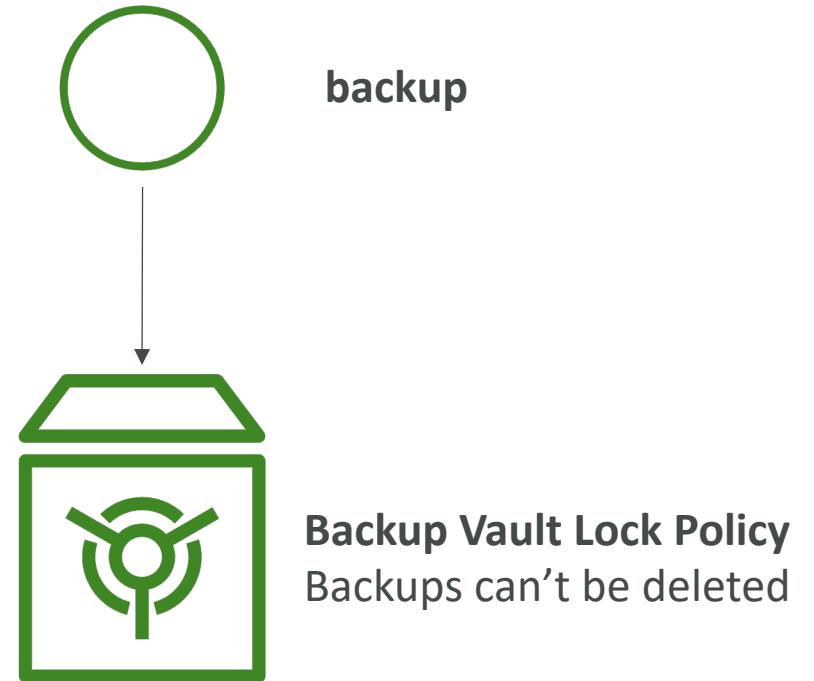
- Supports PITR for supported services
- On-Demand and Scheduled backups
- Tag-based backup policies
- You create backup policies known as **Backup Plans**
 - Backup frequency (every 12 hours, daily, weekly, monthly, cron expression)
 - Backup window
 - Transition to Cold Storage (Never, Days, Weeks, Months, Years)
 - Retention Period (Always, Days, Weeks, Months, Years)

AWS Backup



AWS Backup Vault Lock

- Enforce a WORM (Write Once Read Many) state for all the backups that you store in your AWS Backup Vault
- Additional layer of defense to protect your backups against:
 - Inadvertent or malicious delete operations
 - Updates that shorten or alter retention periods
- Even the root user cannot delete backups when enabled



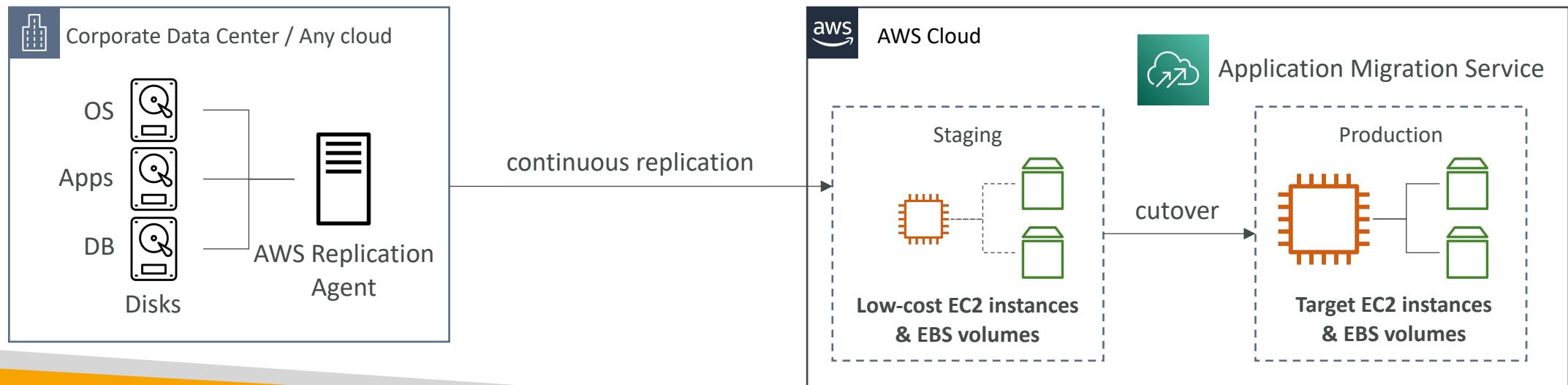
AWS Application Discovery Service



- Plan migration projects by gathering information about on-premises data centers
- Server utilization data and dependency mapping are important for migrations
- Agentless Discovery (AWS Agentless Discovery Connector)
 - VM inventory, configuration, and performance history such as CPU, memory, and disk usage
- Agent-based Discovery (AWS Application Discovery Agent)
 - System configuration, system performance, running processes, and details of the network connections between systems
- Resulting data can be viewed within AWS Migration Hub

AWS Application Migration Service (MGN)

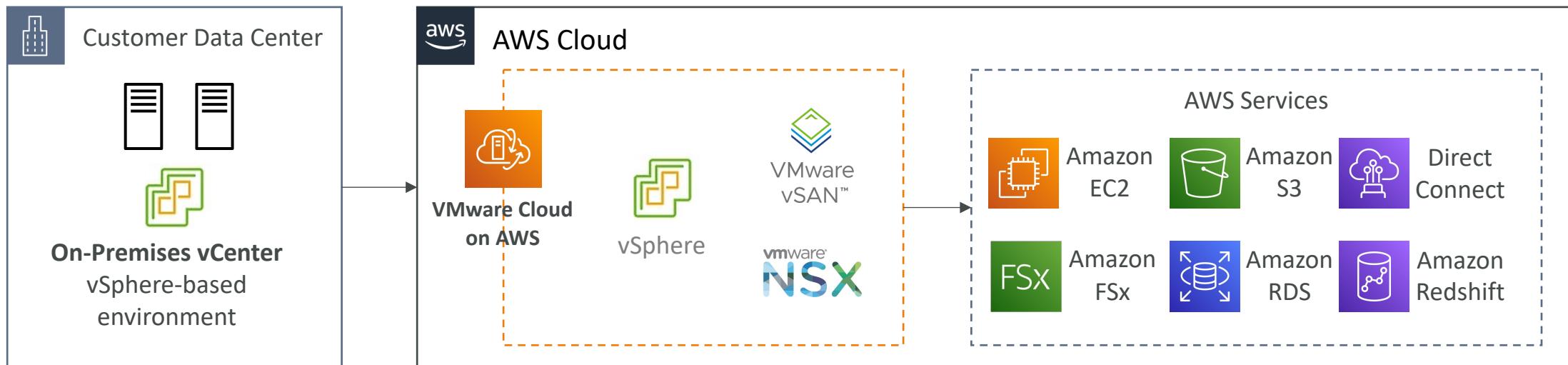
- The “AWS evolution” of CloudEndure Migration, replacing AWS Server Migration Service (SMS)
- Lift-and-shift (rehost) solution which simplify migrating applications to AWS
- Converts your physical, virtual, and cloud-based servers to run natively on AWS
- Supports wide range of platforms, Operating Systems, and databases
- Minimal downtime, reduced costs



VMware Cloud on AWS



- Some customers use VMware Cloud to manage their on-premises Data Center
- They want to extend the Data Center capacity to AWS, but keep using the VMware Cloud software
- ...Enter VMware Cloud on AWS
- Use cases
 - Migrate your VMware vSphere-based workloads to AWS
 - Run your production workloads across VMware vSphere-based private, public, and hybrid cloud environments
 - Have a disaster recover strategy

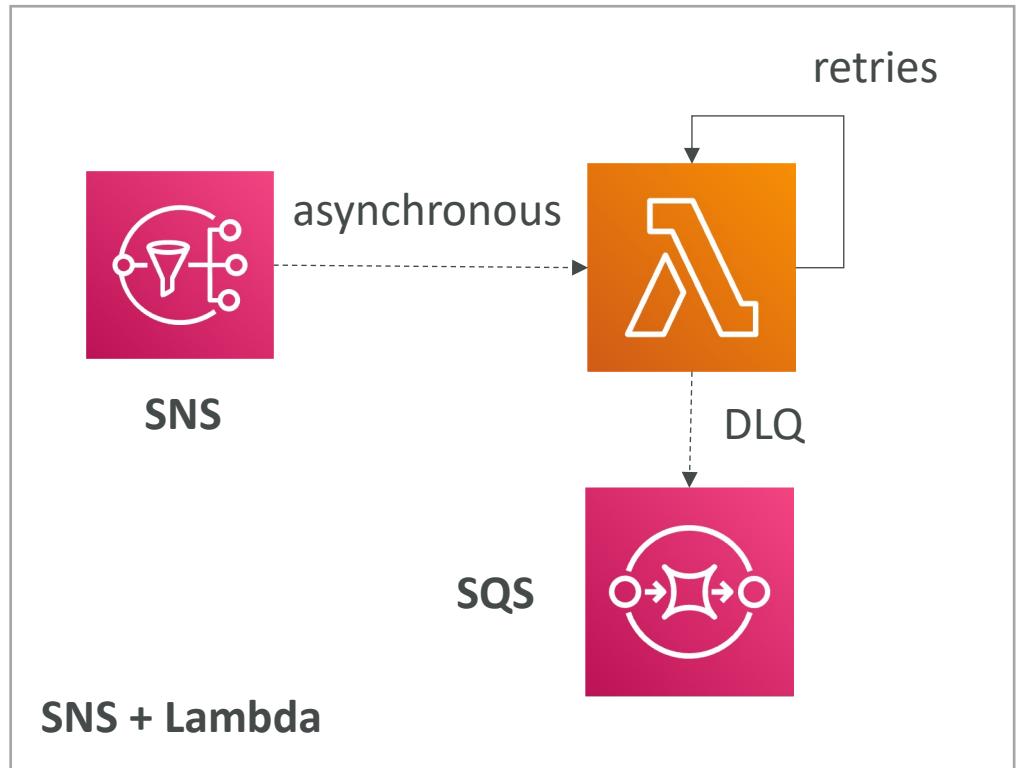
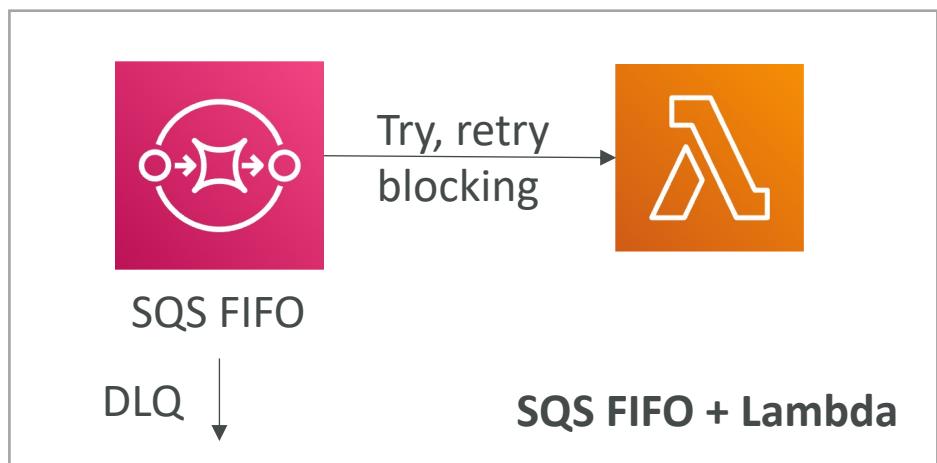
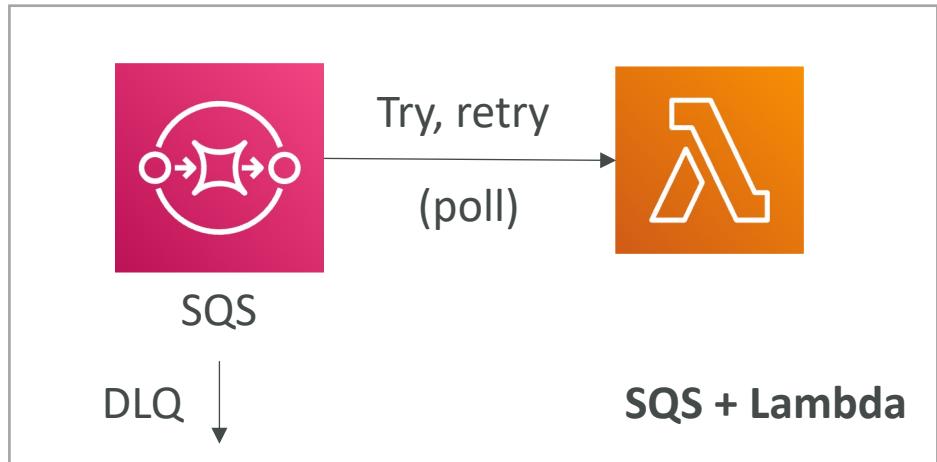


Transferring large amount of data into AWS

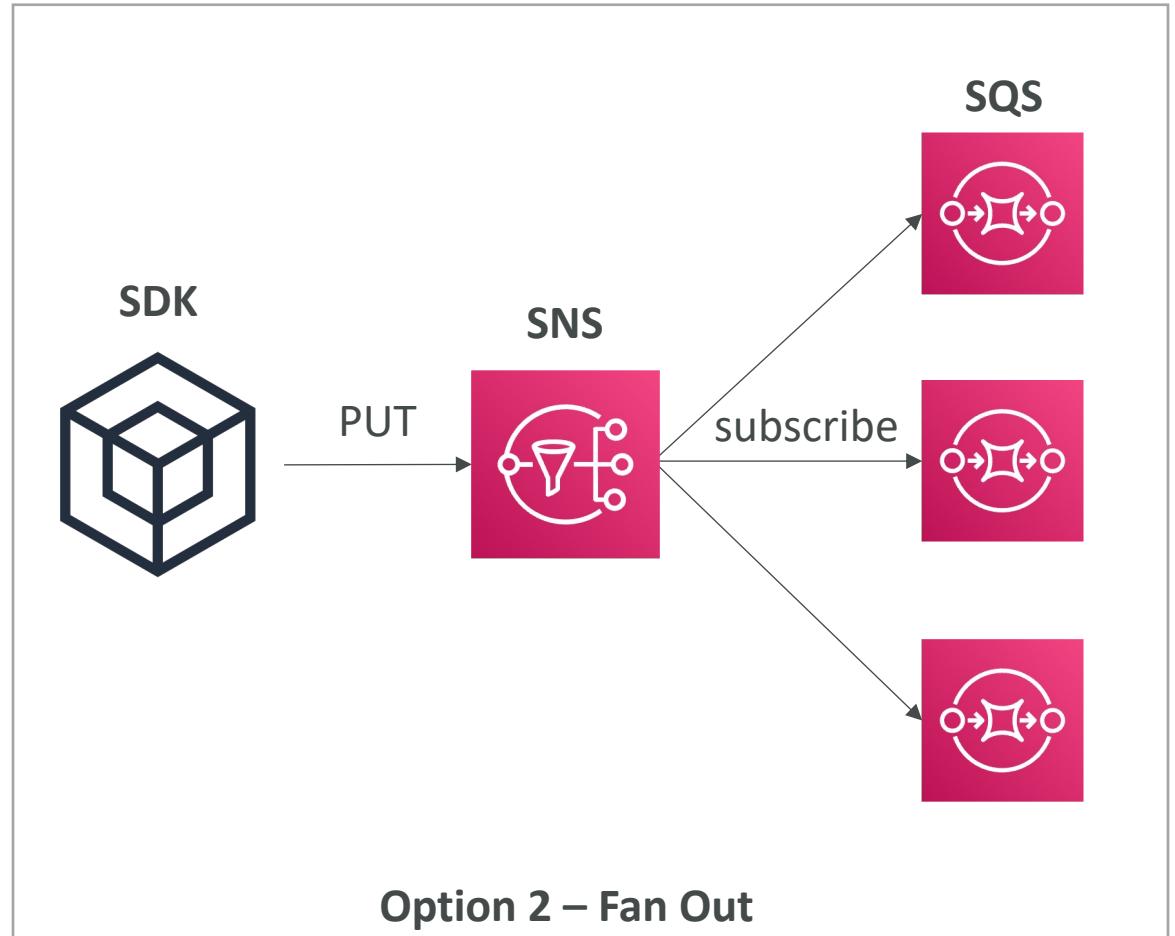
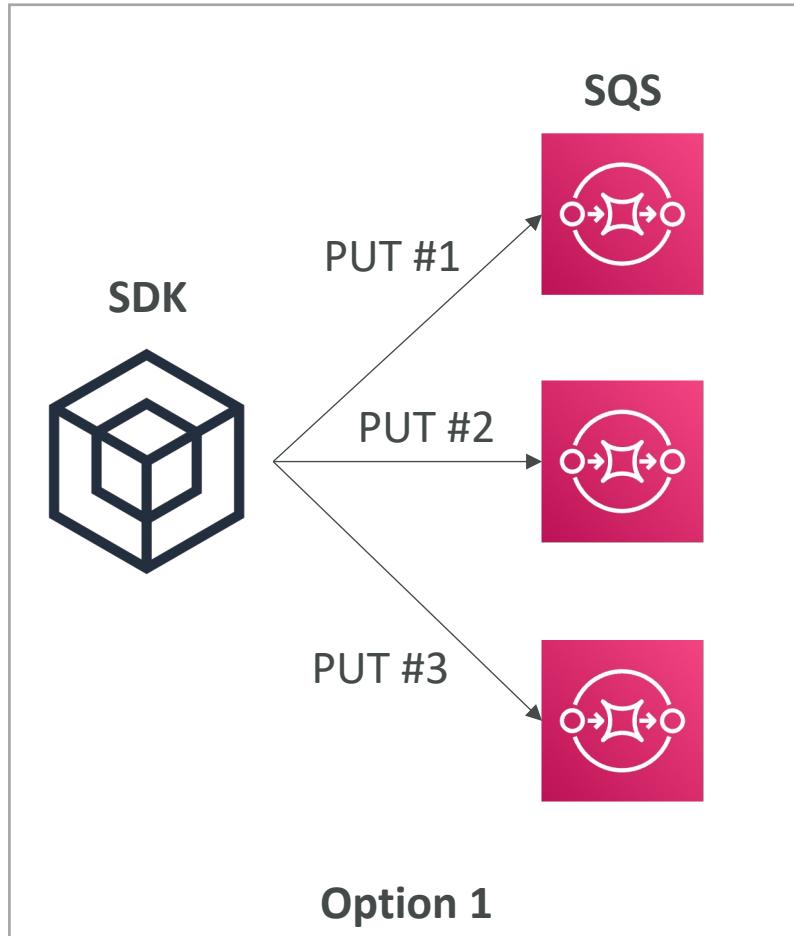
- Example: transfer 200 TB of data in the cloud. We have a 100 Mbps internet connection.
- Over the internet / Site-to-Site VPN:
 - Immediate to setup
 - Will take $200(\text{TB}) * 1000(\text{GB}) * 1000(\text{MB}) * 8(\text{Mb}) / 100 \text{ Mbps} = 16,000,000 \text{s} = 185 \text{d}$
- Over direct connect 1 Gbps:
 - Long for the one-time setup (over a month)
 - Will take $200(\text{TB}) * 1000(\text{GB}) * 8(\text{Gb}) / 1 \text{ Gbps} = 1,600,000 \text{s} = 18.5 \text{d}$
- Over Snowball:
 - Will take 2 to 3 snowballs in parallel
 - Takes about 1 week for the end-to-end transfer
 - Can be combined with DMS
- For on-going replication / transfers: Site-to-Site VPN or DX with DMS or DataSync

Extra Solution Architecture discussions

Lambda, SNS & SQS

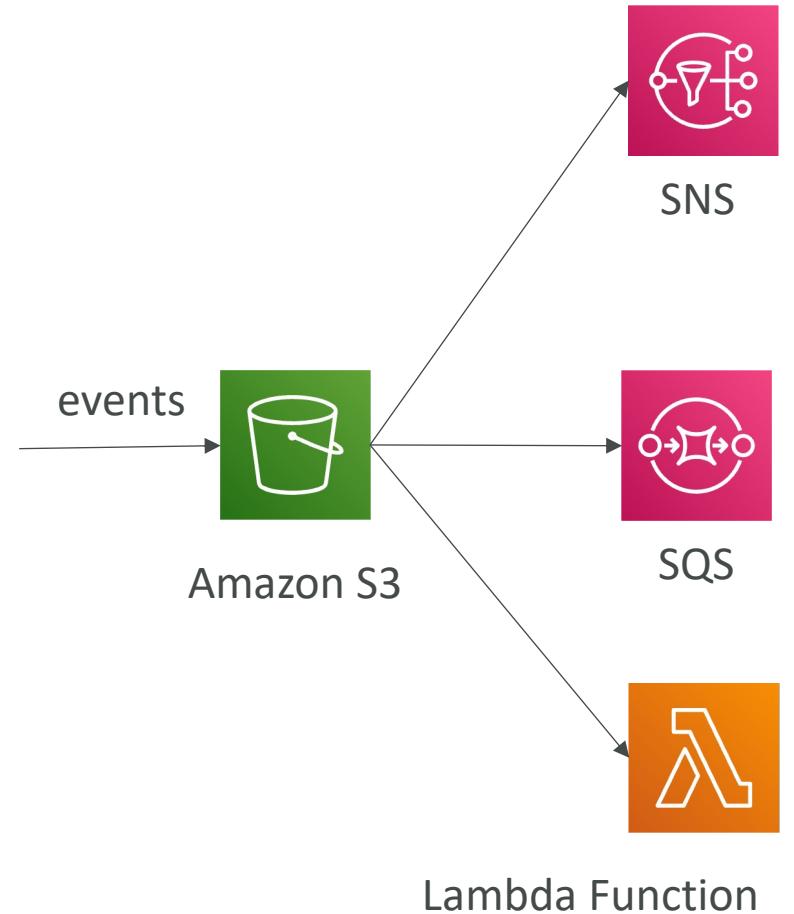


Fan Out Pattern: deliver to multiple SQS

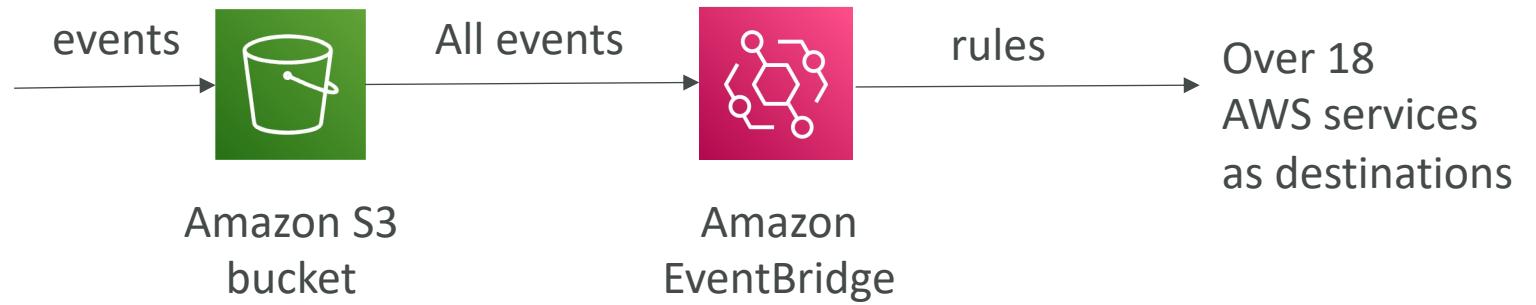


S3 Event Notifications

- S3:ObjectCreated, S3:ObjectRemoved, S3:ObjectRestore, S3:Replication...
- Object name filtering possible (*.jpg)
- Use case: generate thumbnails of images uploaded to S3
- Can create as many “S3 events” as desired
- S3 event notifications typically deliver events in seconds but can sometimes take a minute or longer

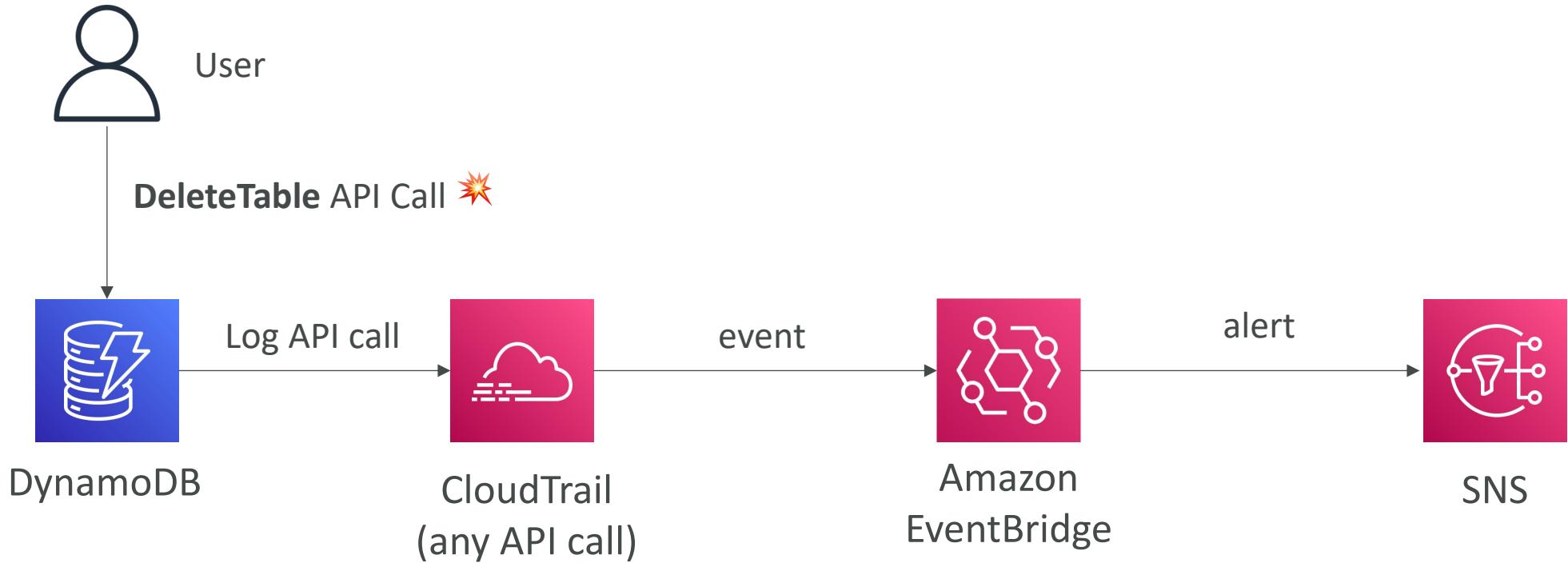


S3 Event Notifications with Amazon EventBridge



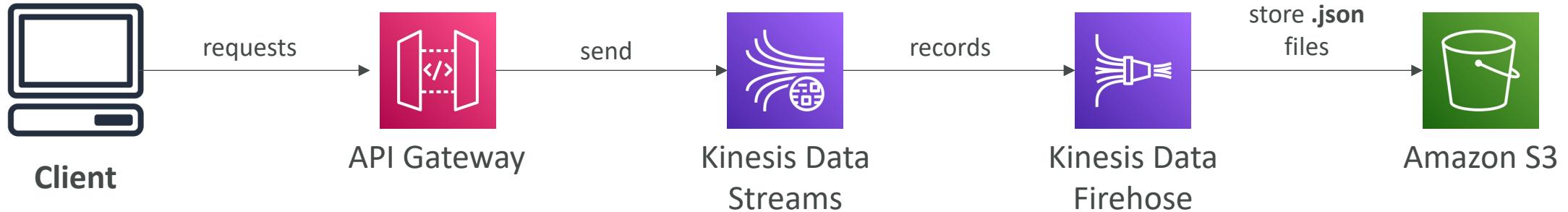
- Advanced filtering options with JSON rules (metadata, object size, name...)
- Multiple Destinations – ex Step Functions, Kinesis Streams / Firehose...
- EventBridge Capabilities – Archive, Replay Events, Reliable delivery

Amazon EventBridge – Intercept API Calls

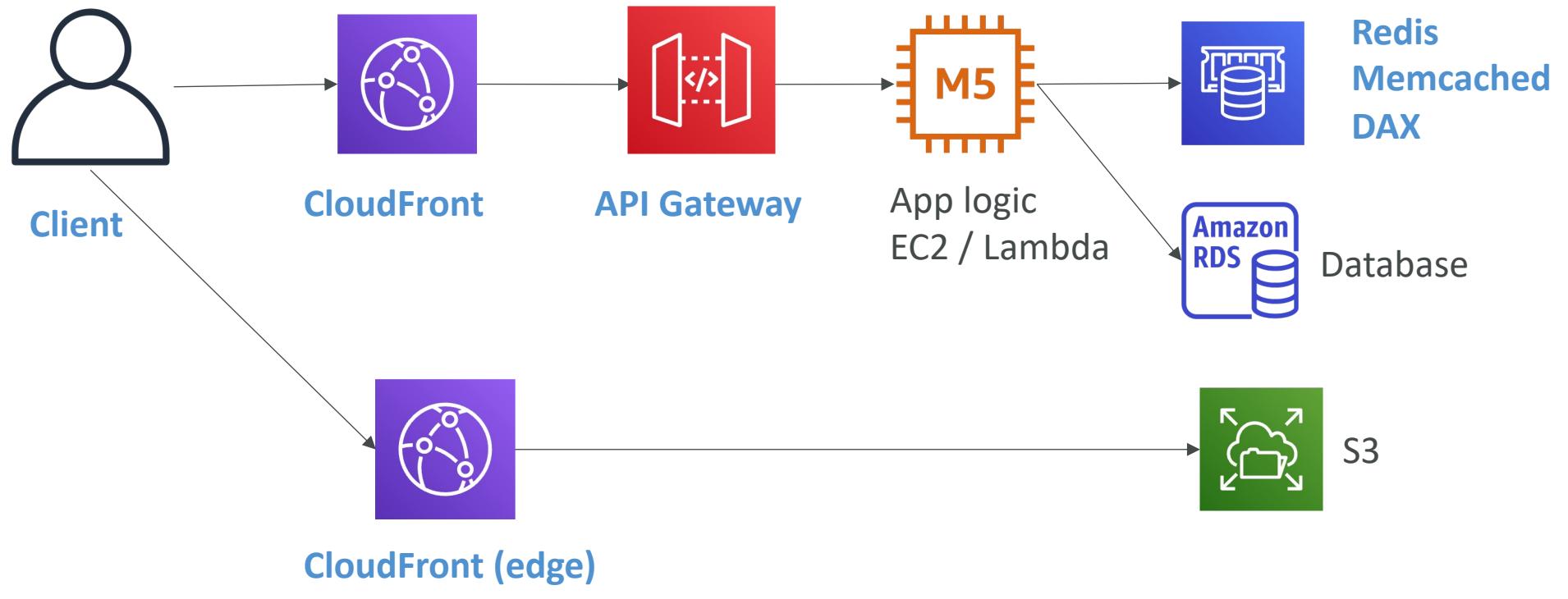


API Gateway – AWS Service Integration

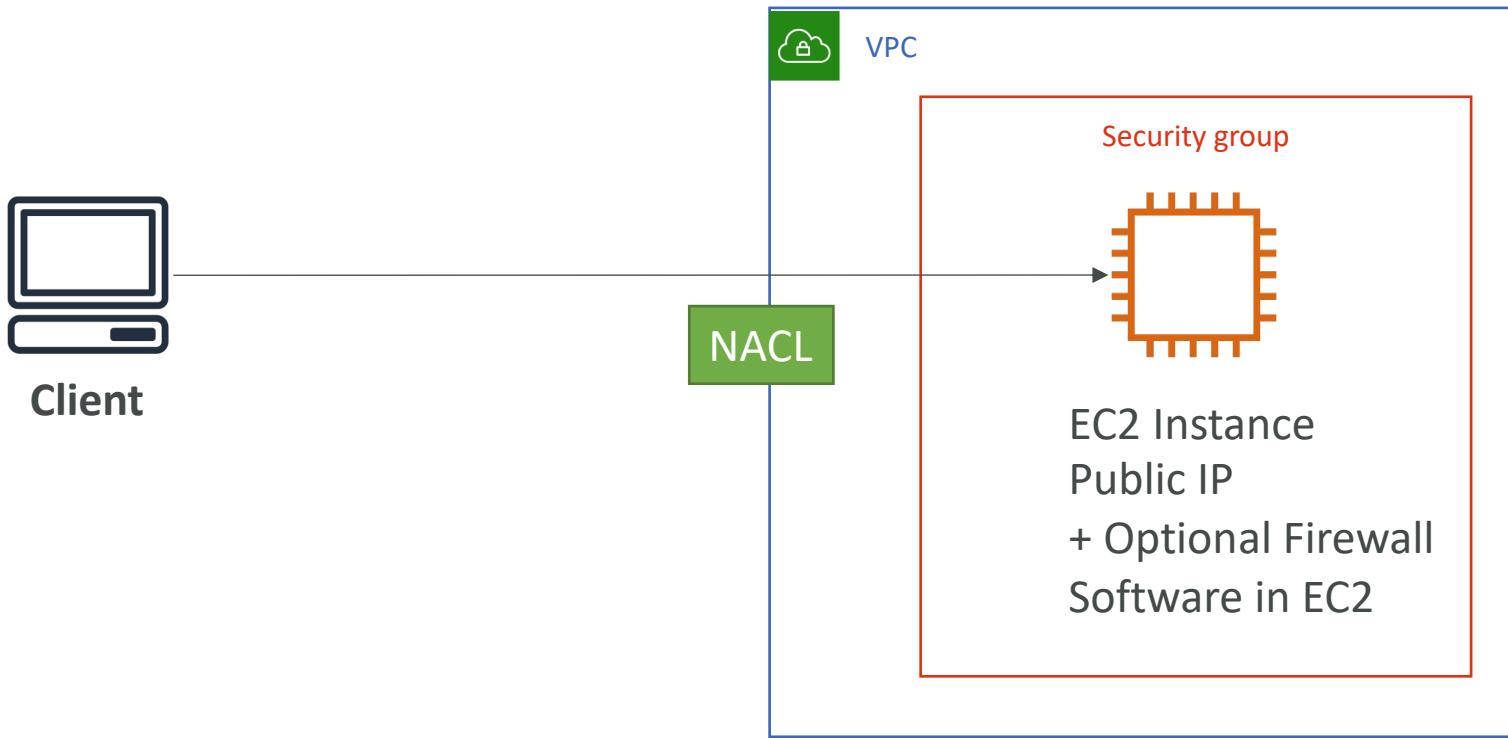
Kinesis Data Streams example



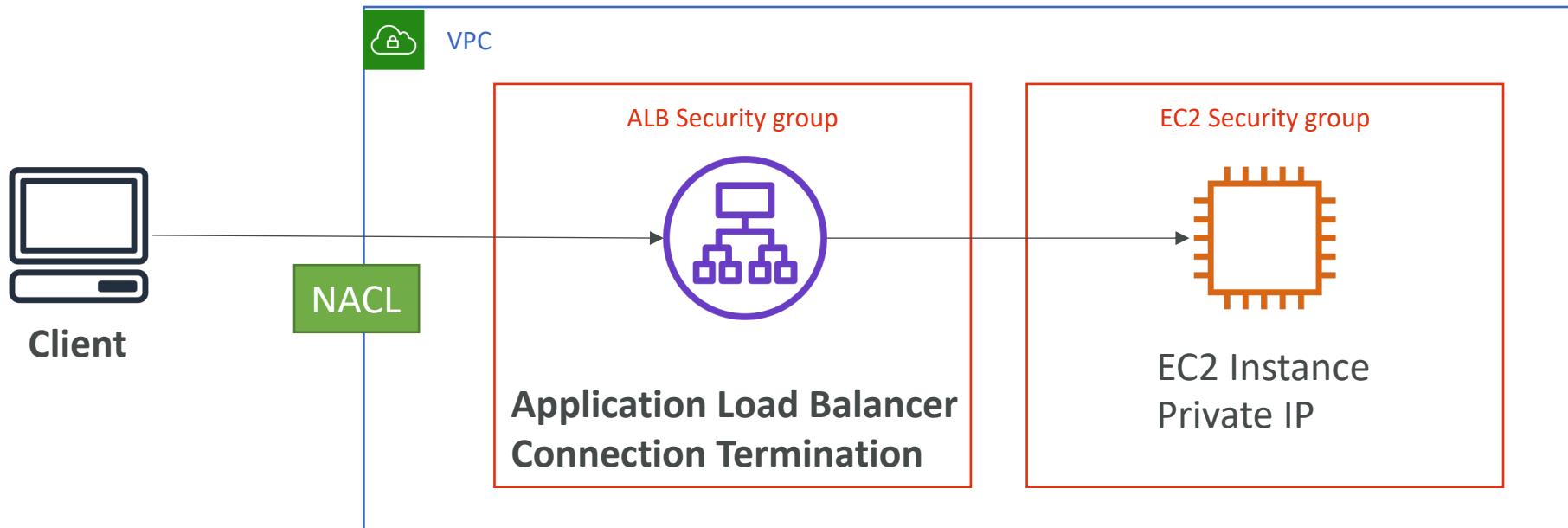
Caching Strategies



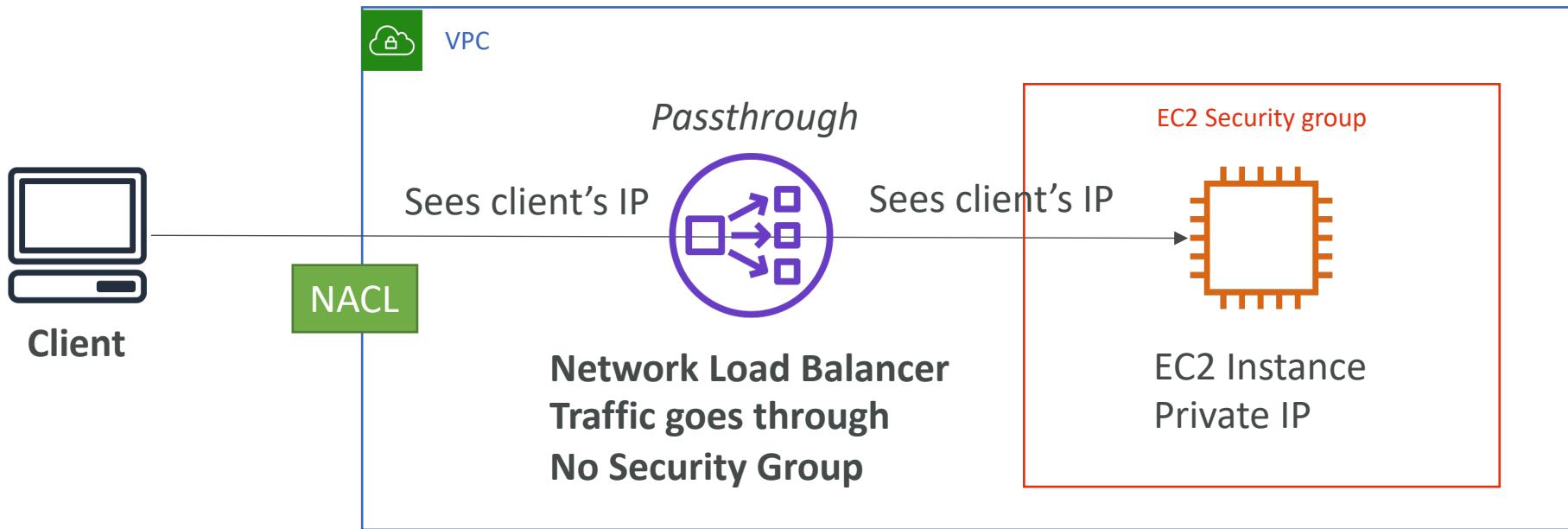
Blocking an IP address



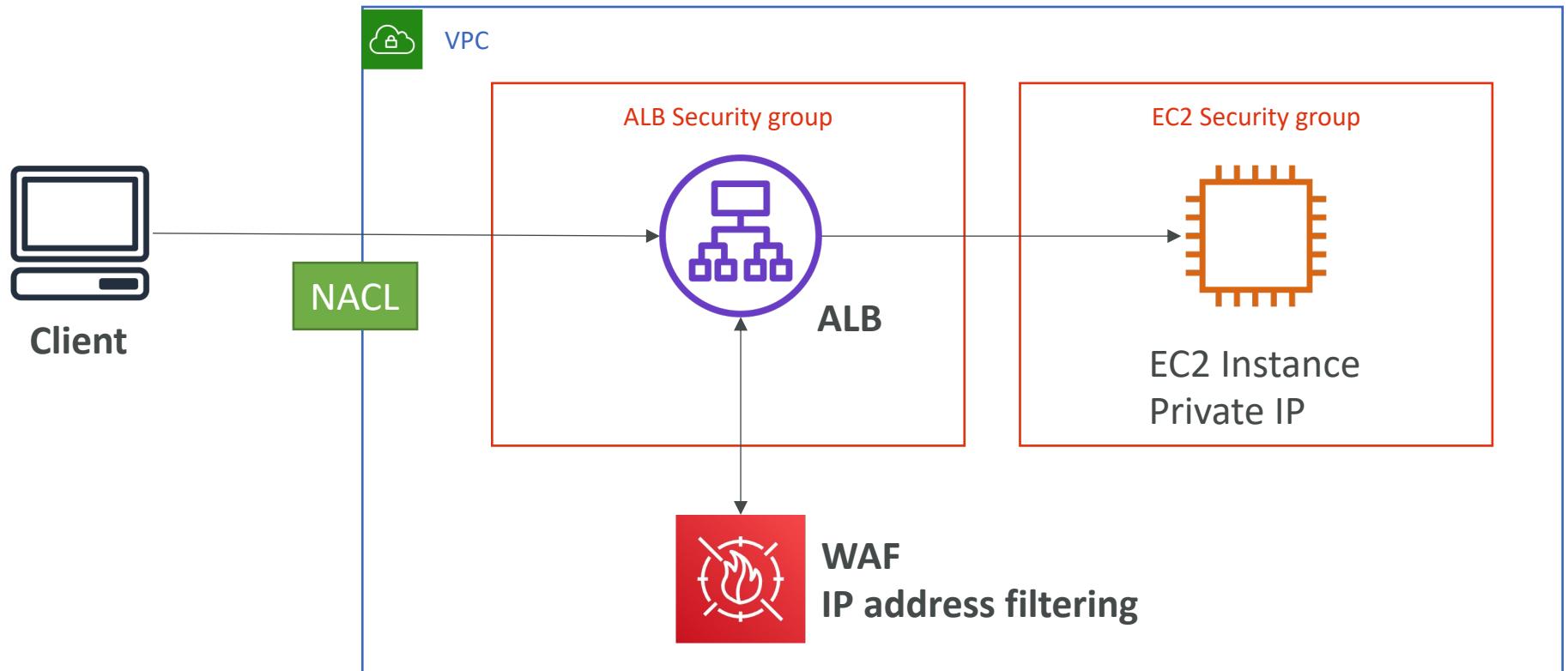
Blocking an IP address – with an ALB



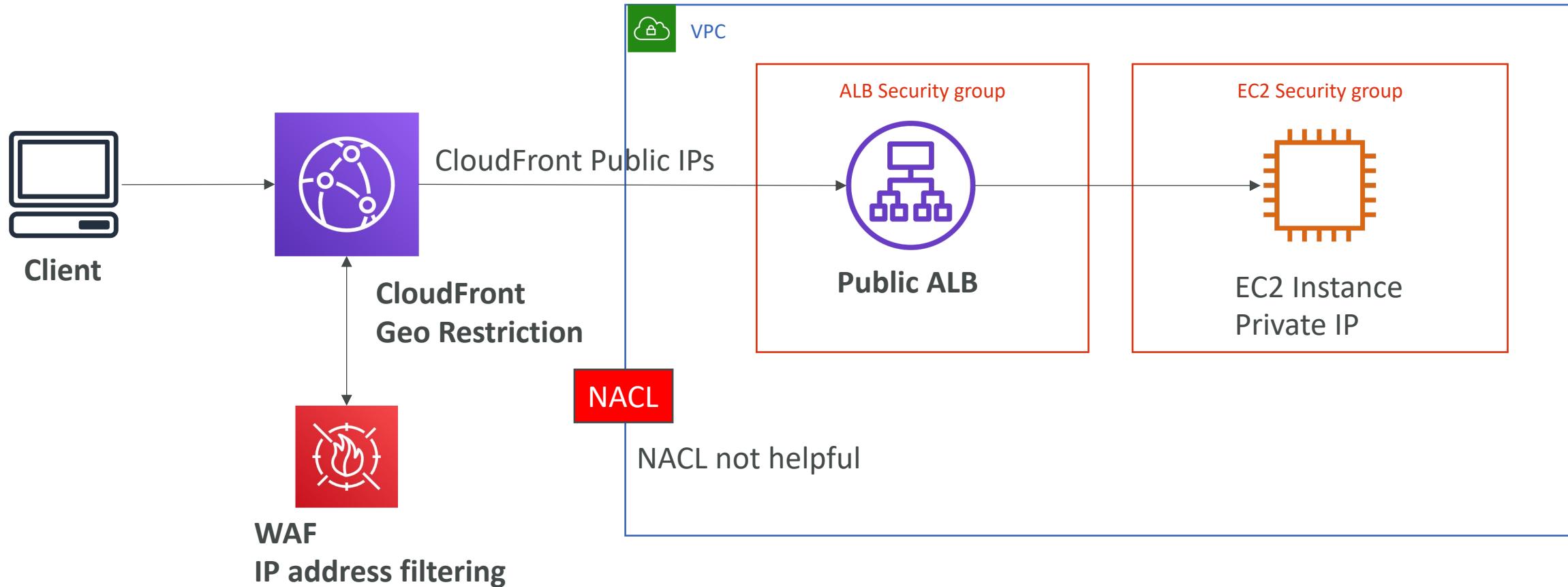
Blocking an IP address – with an NLB



Blocking an IP address – ALB + WAF



Blocking an IP address – ALB, CloudFront WAF



High Performance Computing (HPC)

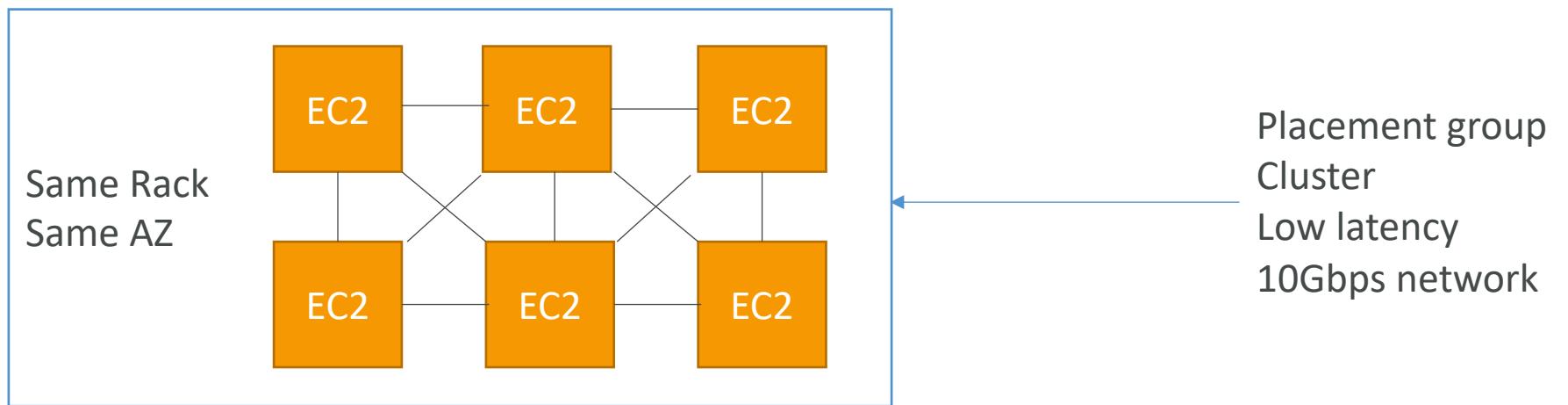
- The cloud is the perfect place to perform HPC
- You can create a very high number of resources in no time
- You can speed up time to results by adding more resources
- You can pay only for the systems you have used
- Perform genomics, computational chemistry, financial risk modeling, weather prediction, machine learning, deep learning, autonomous driving
- Which services help perform HPC?

Data Management & Transfer

- AWS Direct Connect:
 - Move GB/s of data to the cloud, over a private secure network
- Snowball & Snowmobile
 - Move PB of data to the cloud
- AWS DataSync
 - Move large amount of data between on-premises and S3, EFS, FSx for Windows

Compute and Networking

- EC2 Instances:
 - CPU optimized, GPU optimized
 - Spot Instances / Spot Fleets for cost savings + Auto Scaling
- EC2 Placement Groups: **Cluster** for good network performance



Compute and Networking

- EC2 Enhanced Networking (SR-IOV)
 - Higher bandwidth, higher PPS (packet per second), lower latency
 - Option 1: Elastic Network Adapter (ENA) up to 100 Gbps
 - Option 2: Intel 82599 VF up to 10 Gbps – LEGACY
- Elastic Fabric Adapter (EFA)
 - Improved ENA for HPC, only works for Linux
 - Great for inter-node communications, **tightly coupled workloads**
 - Leverages Message Passing Interface (MPI) standard
 - **Bypasses the underlying Linux OS to provide low-latency, reliable transport**

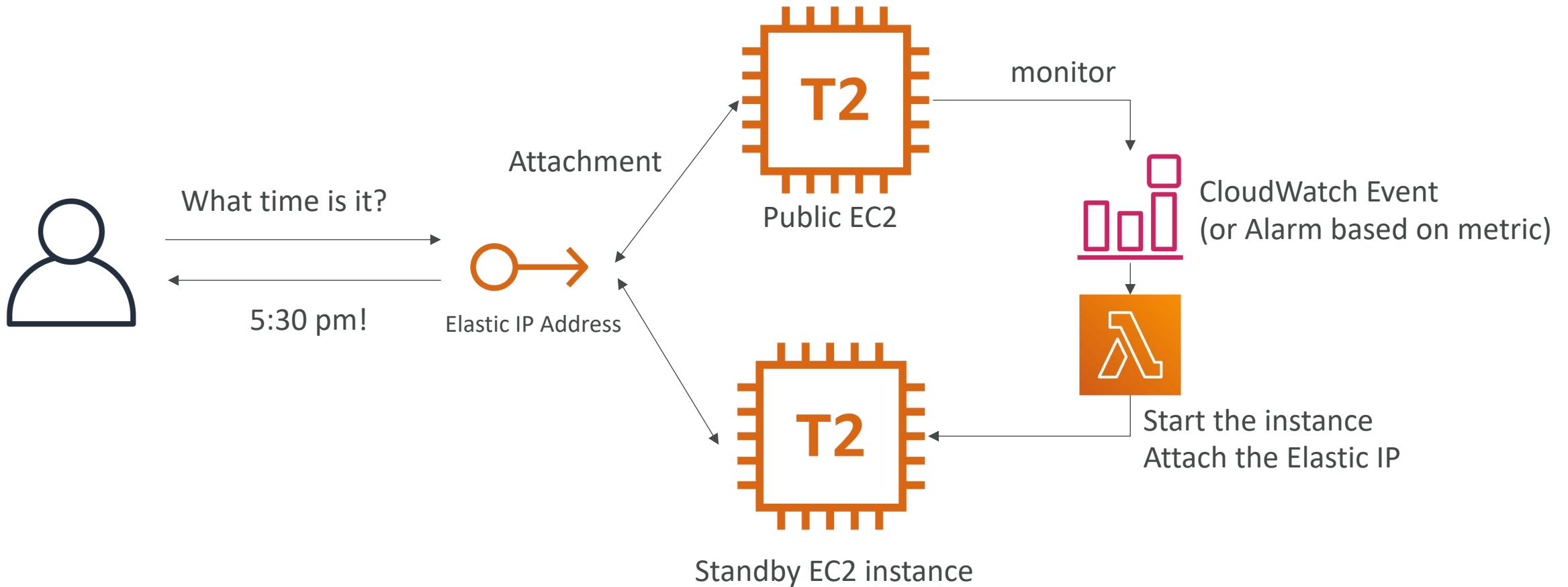
Storage

- Instance-attached storage:
 - EBS: scale up to 256,000 IOPS with io2 Block Express
 - Instance Store: scale to millions of IOPS, linked to EC2 instance, low latency
- Network storage:
 - Amazon S3: large blob, not a file system
 - Amazon EFS: scale IOPS based on total size, or use provisioned IOPS
 - Amazon FSx for Lustre:
 - HPC optimized distributed file system, millions of IOPS
 - Backed by S3

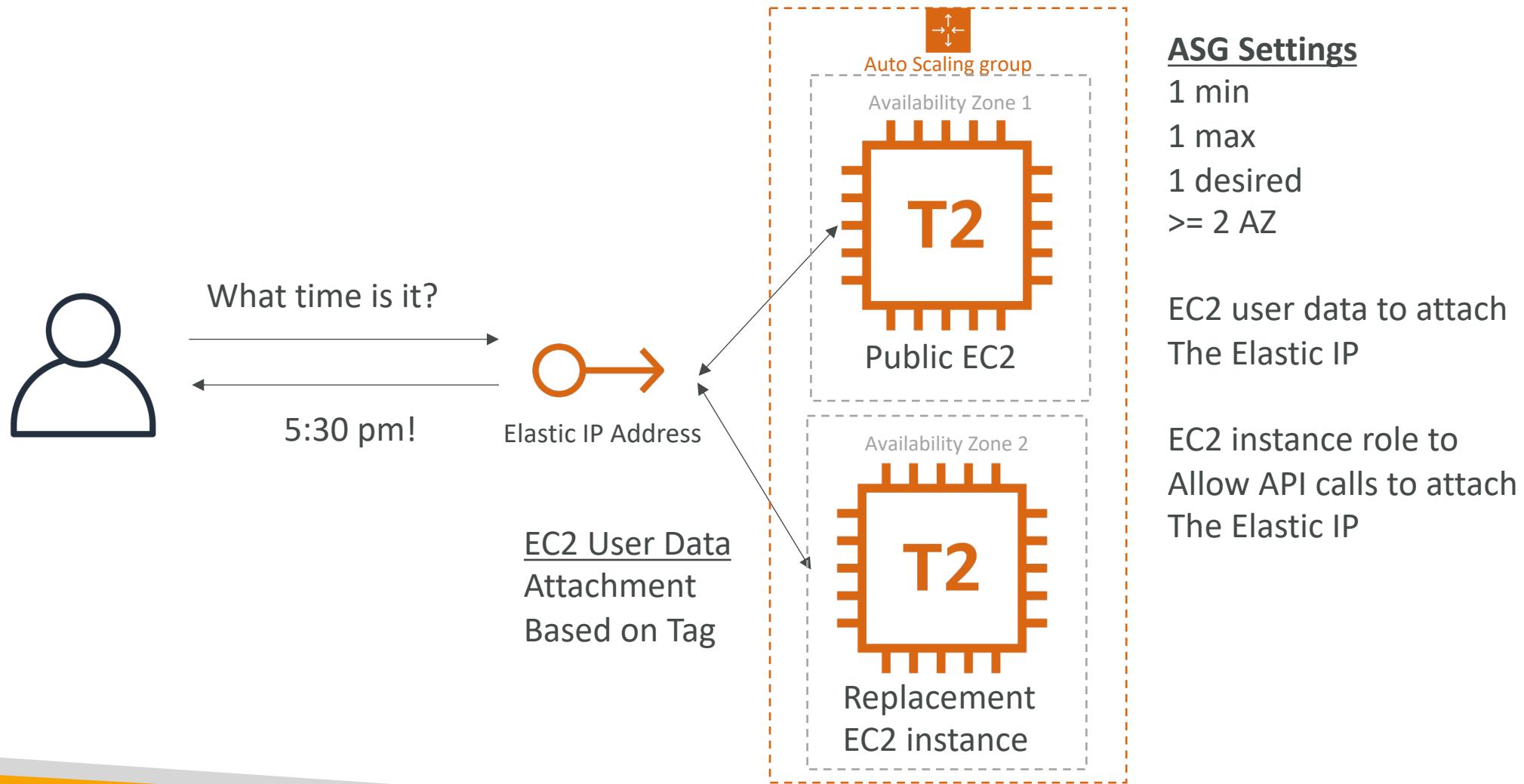
Automation and Orchestration

- AWS Batch
 - AWS Batch supports multi-node parallel jobs, which enables you to run single jobs that span multiple EC2 instances.
 - Easily schedule jobs and launch EC2 instances accordingly
- AWS ParallelCluster
 - Open-source cluster management tool to deploy HPC on AWS
 - Configure with text files
 - Automate creation of VPC, Subnet, cluster type and instance types
 - Ability to enable EFA on the cluster (improves network performance)

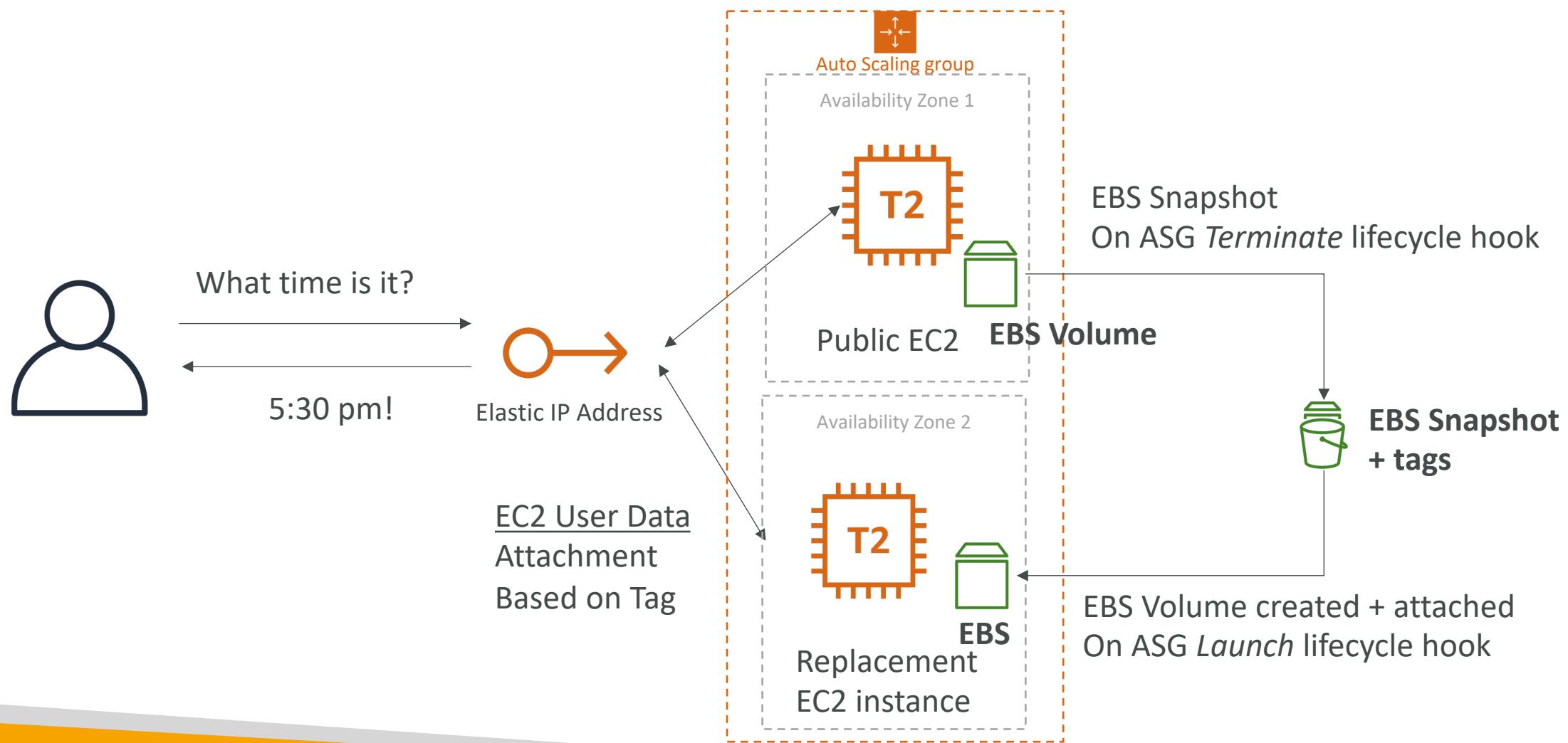
Creating a highly available EC2 instance



Creating a highly available EC2 instance With an Auto Scaling Group



Creating a highly available EC2 instance With ASG + EBS



Other Services

Overview of Services that might come up in a few questions



What is CloudFormation

- CloudFormation is a declarative way of outlining your AWS Infrastructure, for any resources (most of them are supported).
- For example, within a CloudFormation template, you say:
 - I want a security group
 - I want two EC2 instances using this security group
 - I want an S3 bucket
 - I want a load balancer (ELB) in front of these machines
- Then CloudFormation creates those for you, in the **right order**, with the **exact configuration** that you specify

Benefits of AWS CloudFormation (1/2)

- Infrastructure as code
 - No resources are manually created, which is excellent for control
 - Changes to the infrastructure are reviewed through code
- Cost
 - Each resources within the stack is tagged with an identifier so you can easily see how much a stack costs you
 - You can estimate the costs of your resources using the CloudFormation template
 - Savings strategy: In Dev, you could automation deletion of templates at 5 PM and recreated at 8 AM, safely

Benefits of AWS CloudFormation (2/2)

- Productivity
 - Ability to destroy and re-create an infrastructure on the cloud on the fly
 - Automated generation of Diagram for your templates!
 - Declarative programming (no need to figure out ordering and orchestration)
- Don't re-invent the wheel
 - Leverage existing templates on the web!
 - Leverage the documentation
- Supports (almost) all AWS resources:
 - Everything we'll see in this course is supported
 - You can use "custom resources" for resources that are not supported

CloudFormation Stack Designer

- Example: WordPress CloudFormation Stack
- We can see all the resources
- We can see the relations between the components



Amazon Simple Email Service (Amazon SES)



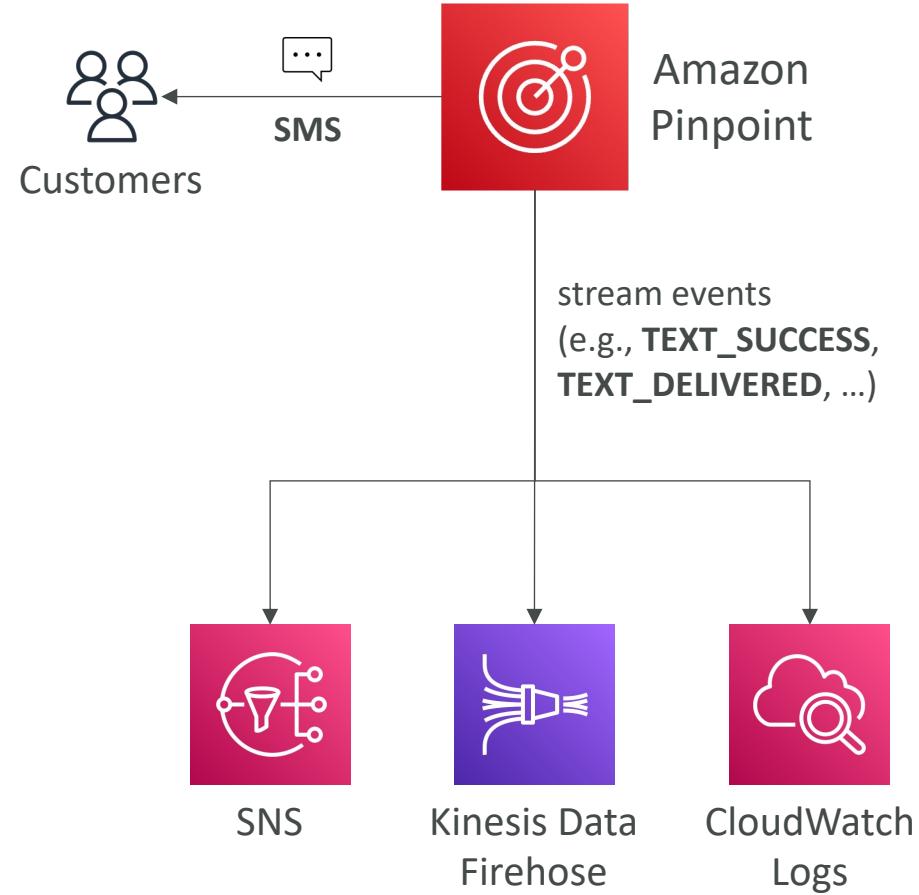
- Fully managed service to send emails securely, globally and at scale
- Allows inbound/outbound emails
- Reputation dashboard, performance insights, anti-spam feedback
- Provides statistics such as email deliveries, bounces, feedback loop results, email open
- Supports DomainKeys Identified Mail (DKIM) and Sender Policy Framework (SPF)
- Flexible IP deployment: shared, dedicated, and customer-owned IPs
- Send emails using your application using AWS Console, APIs, or SMTP
- Use cases: transactional, marketing and bulk email communications



Amazon Pinpoint

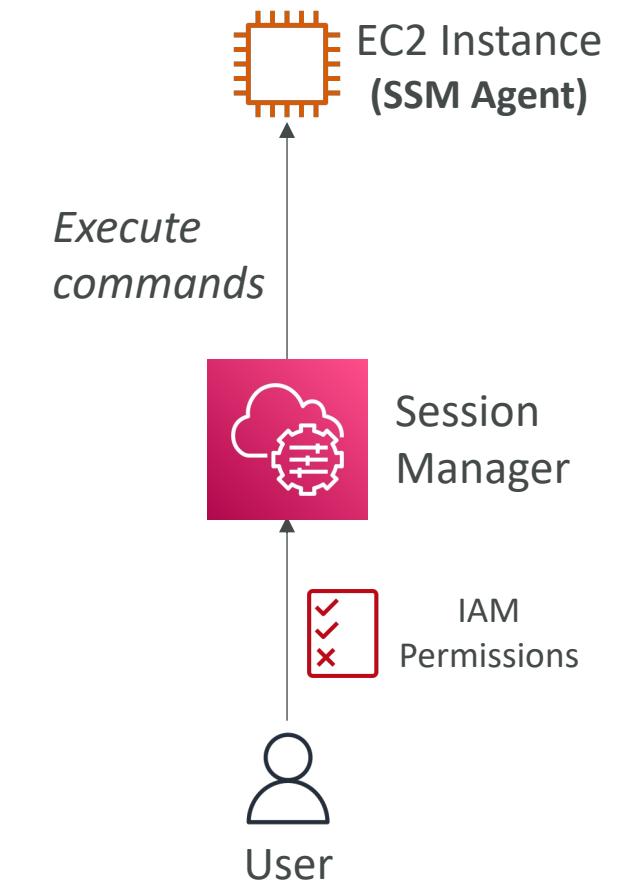


- Scalable 2-way (outbound/inbound) marketing communications service
- Supports email, SMS, push, voice, and in-app messaging
- Ability to segment and personalize messages with the right content to customers
- Possibility to receive replies
- Scales to billions of messages per day
- Use cases: run campaigns by sending marketing, bulk, transactional SMS messages
- **Versus Amazon SNS or Amazon SES**
 - In SNS & SES you managed each message's audience, content, and delivery schedule
 - In Amazon Pinpoint, you create message templates, delivery schedules, highly-targeted segments, and full campaigns



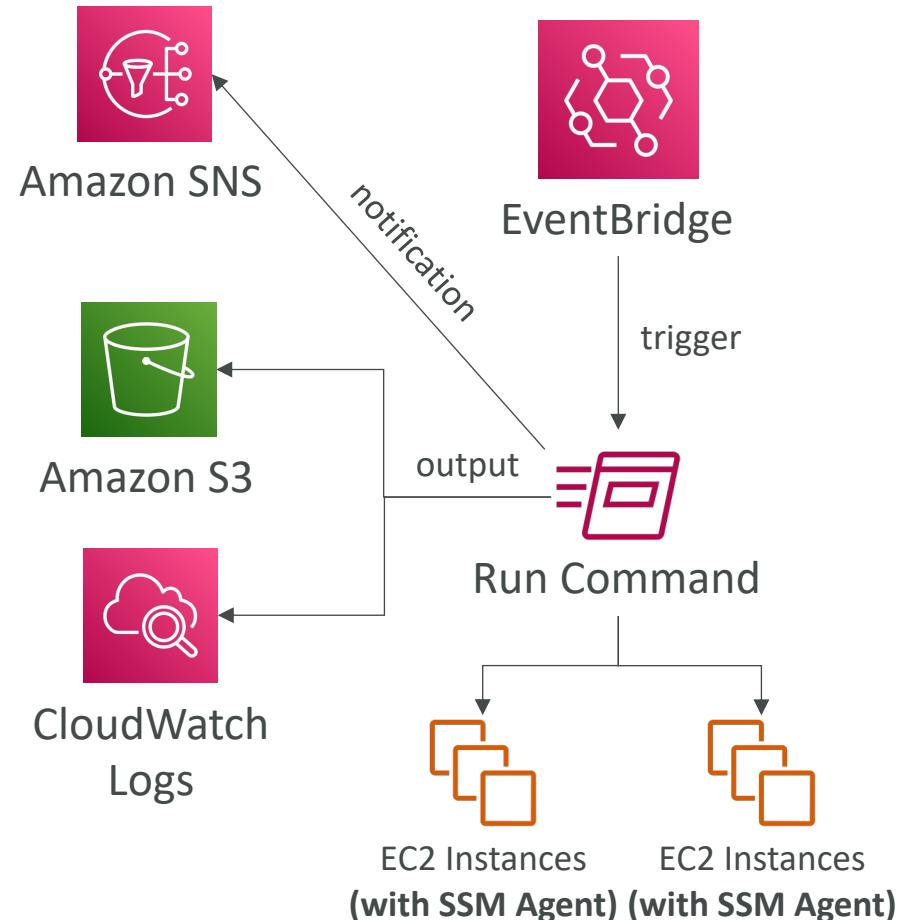
Systems Manager – SSM Session Manager

- Allows you to start a secure shell on your EC2 and on-premises servers
- No SSH access, bastion hosts, or SSH keys needed
- No port 22 needed (better security)
- Supports Linux, macOS, and Windows
- Send session log data to S3 or CloudWatch Logs



Systems Manager – Run Command

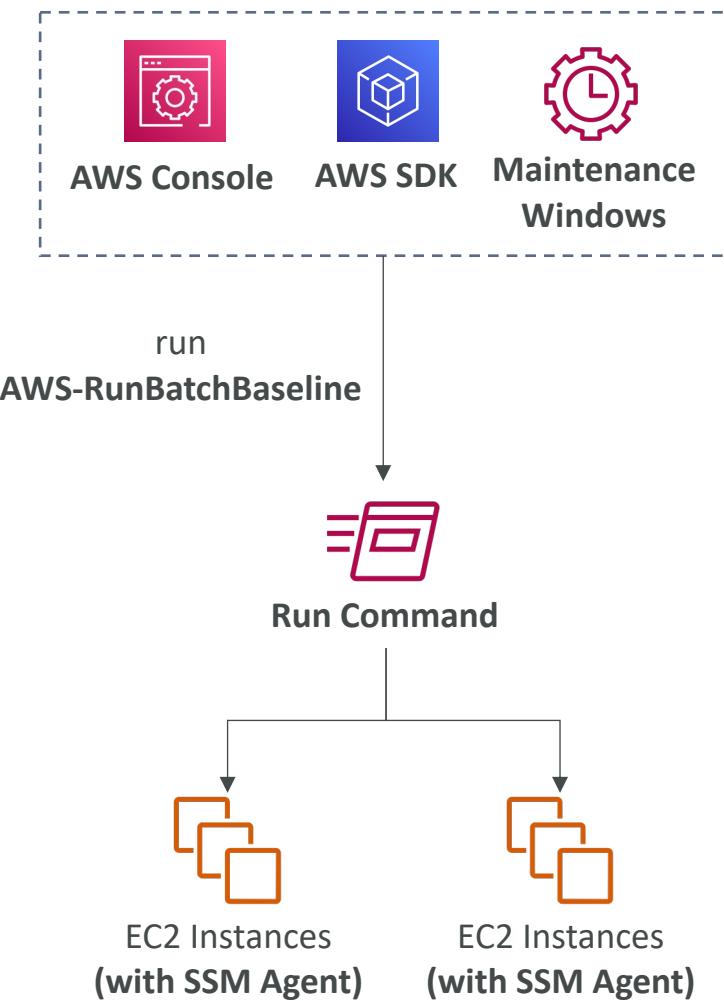
- Execute a document (= script) or just run a command
- Run command across multiple instances (using resource groups)
- No need for SSH
- Command Output can be shown in the AWS Console, sent to S3 bucket or CloudWatch Logs
- Send notifications to SNS about command status (In progress, Success, Failed, ...)
- Integrated with IAM & CloudTrail
- Can be invoked using EventBridge



Systems Manager – Patch Manager



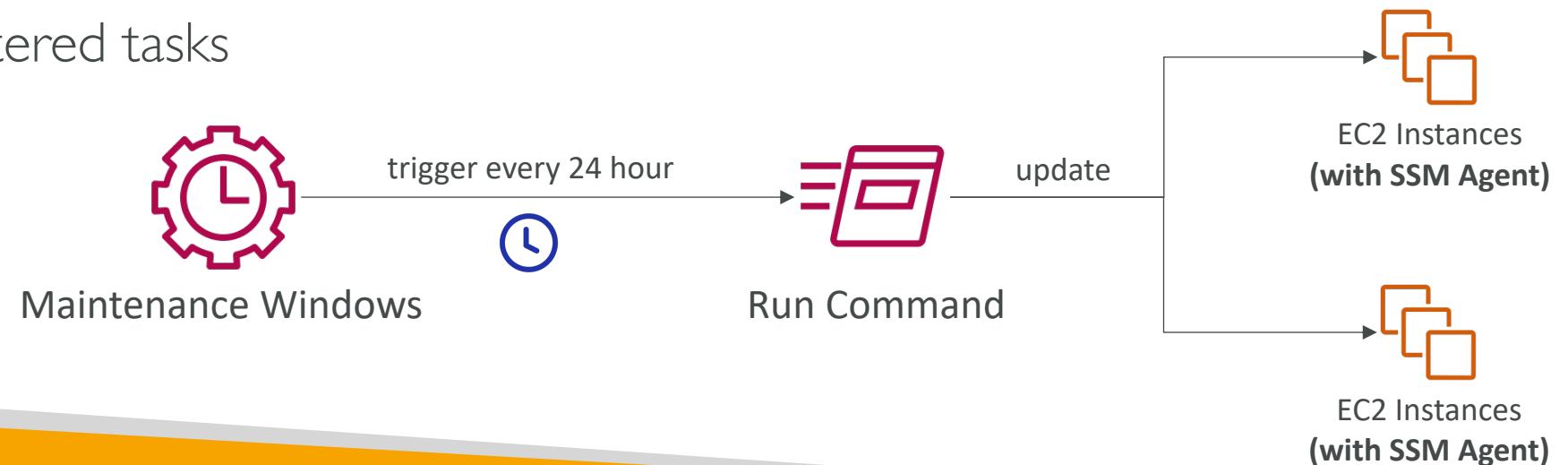
- Automates the process of patching managed instances
- OS updates, applications updates, security updates
- Supports EC2 instances and on-premises servers
- Supports Linux, macOS, and Windows
- Patch on-demand or on a schedule using **Maintenance Windows**
- Scan instances and generate patch compliance report (missing patches)





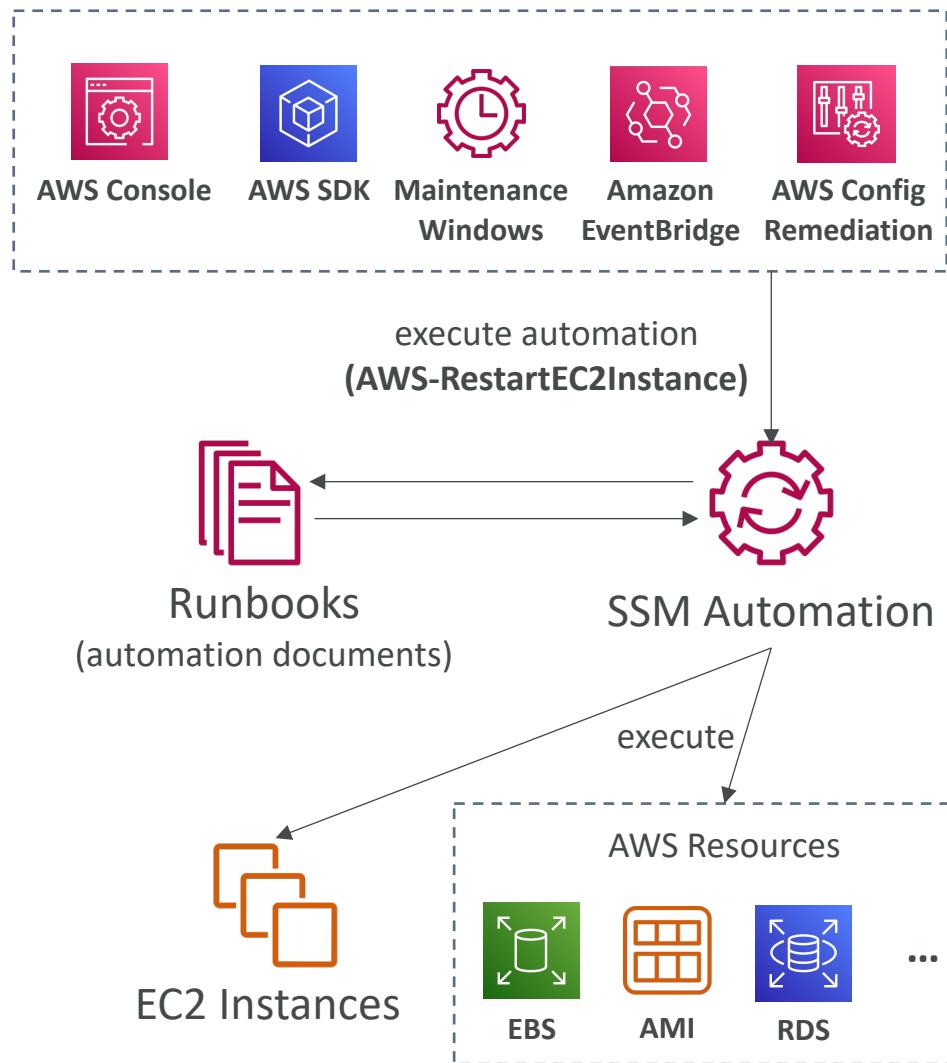
Systems Manager – Maintenance Windows

- Defines a schedule for when to perform actions on your instances
- Example: OS patching, updating drivers, installing software, ...
- Maintenance Window contains
 - Schedule
 - Duration
 - Set of registered instances
 - Set of registered tasks



Systems Manager - Automation

- Simplifies common maintenance and deployment tasks of EC2 instances and other AWS resources
- Examples: restart instances, create an AMI, EBS snapshot
- **Automation Runbook** – SSM Documents to define actions preformed on your EC2 instances or AWS resources (pre-defined or custom)
- Can be triggered using:
 - Manually using AWS Console, AWS CLI or SDK
 - Amazon EventBridge
 - On a schedule using Maintenance Windows
 - By AWS Config for rules remediations



Cost Explorer

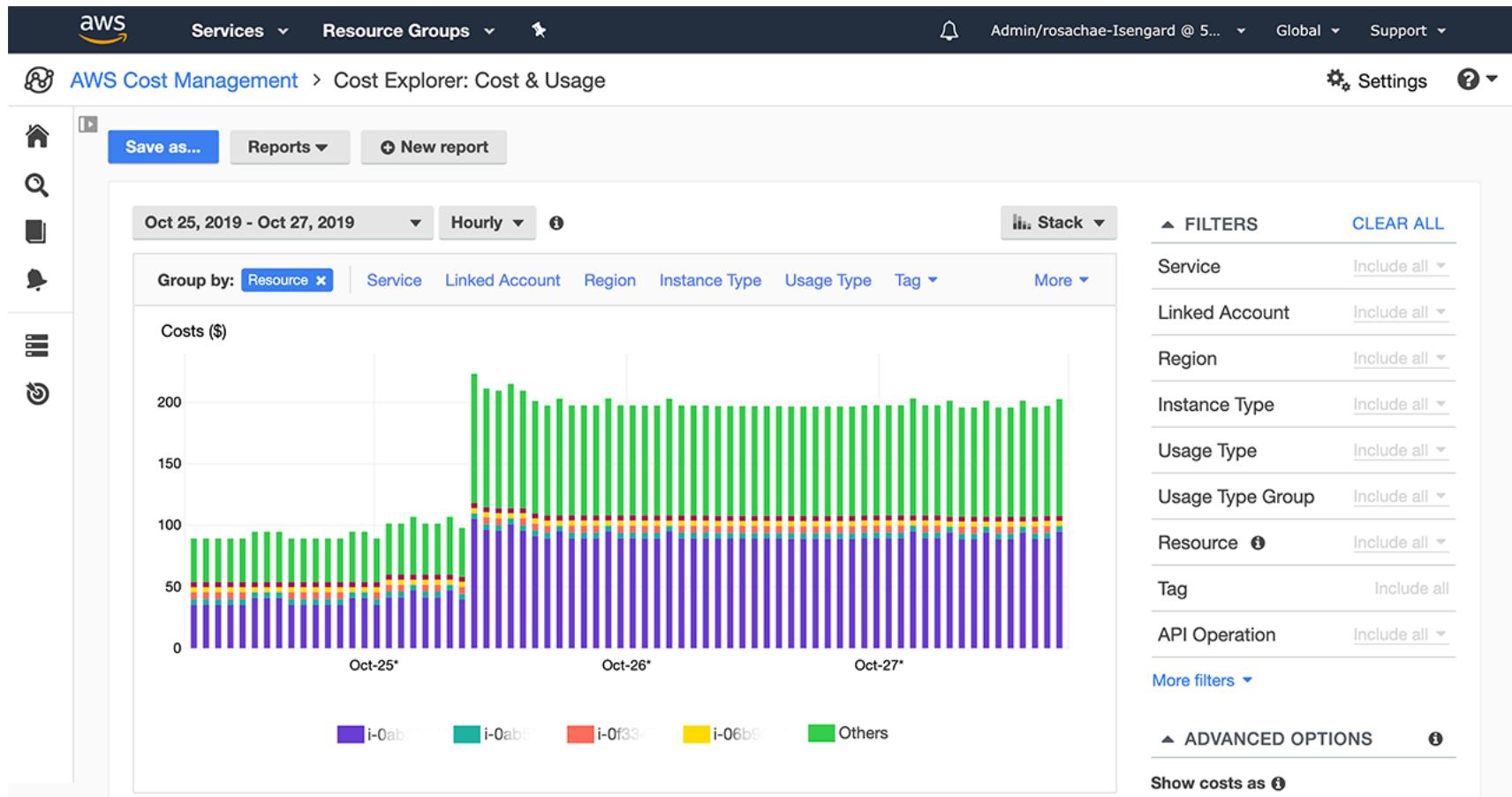


- Visualize, understand, and manage your AWS costs and usage over time
- Create custom reports that analyze cost and usage data.
- Analyze your data at a high level: total costs and usage across all accounts
- Or Monthly, hourly, resource level granularity
- Choose an optimal **Savings Plan** (to lower prices on your bill)
- Forecast usage up to 12 months based on previous usage

Cost Explorer – Monthly Cost by AWS Service



Cost Explorer– Hourly & Resource Level



Cost Explorer – Savings Plan Alternative to Reserved Instances

Recommendation options

Savings Plans type <input checked="" type="radio"/> Compute <input type="radio"/> EC2 Instance	Savings Plans term <input type="radio"/> 1-year <input checked="" type="radio"/> 3-year	Payment option <input checked="" type="radio"/> All upfront <input type="radio"/> Partial upfront <input type="radio"/> No upfront	Based on the past <input type="radio"/> 7 days <input type="radio"/> 30 days <input checked="" type="radio"/> 60 days
--	---	---	--

Recommendation: Purchase a Compute Savings Plan at a commitment of \$2.40/hour

You could save an estimated **\$1,173** monthly by purchasing the recommended Compute Savings Plan.

Based on your past **60 days** of usage, we recommend purchasing a Savings Plan with a commitment of **\$2.40/hour** for a **3-year term**. With this commitment, we project that you could save an average of **\$1.61/hour** - representing a **40%** savings compared to On-Demand. To account for variable usage patterns, this recommendation maximizes your savings by leaving an average **\$0.04/hour** of On-Demand spend.

Before recommended purchase	After recommended purchase (based on your past 60 days of usage)
Monthly On-Demand spend <small> ⓘ</small> \$2,955 (\$4.05/hour) Based on your On-Demand spend over the past 60 days	Estimated monthly spend <small> ⓘ</small> \$1,782 (\$2.44/hour) Your recommended \$2.40/hour Savings Plans commitment + an average \$0.04/hour of On-Demand spend Estimated monthly savings <small> ⓘ</small> \$1,173 (\$1.61/hour) 40% monthly savings over On-Demand \$2,955 - \$1,782 = \$1,173

This recommendation examines your usage over the past 60 days (including your existing Savings Plans and EC2 Reserved Instances) and calculates what your costs would have been had you purchased the recommended Savings Plans. See applicable rates for Savings Plans [here](#). To generate this recommendation, AWS simulates your bill for different commitment amounts and recommends the commitment amount that provides the greatest estimated savings. [Learn more](#)

Recommended Compute Savings Plans [Download CSV](#) [Add selected Savings Plan\(s\) to cart](#)

x	Term	Payment option	Recommended commitment	Estimated hourly savings <small> ⓘ</small>
<input checked="" type="checkbox"/>	3-year	All upfront	\$2.40/hour	\$1.61 (40%)

*Average hourly spend and minimum hourly spend based on your current on-demand spend for the given instance family.

Cost Explorer – Forecast Usage



Amazon Elastic Transcoder



- Elastic Transcoder is used to convert media files stored in S3 into media files in the formats required by consumer playback devices (phones etc..)
- Benefits:
 - Easy to use
 - Highly scalable – can handle large volumes of media files and large file sizes
 - Cost effective – duration-based pricing model
 - Fully managed & secure, pay for what you use

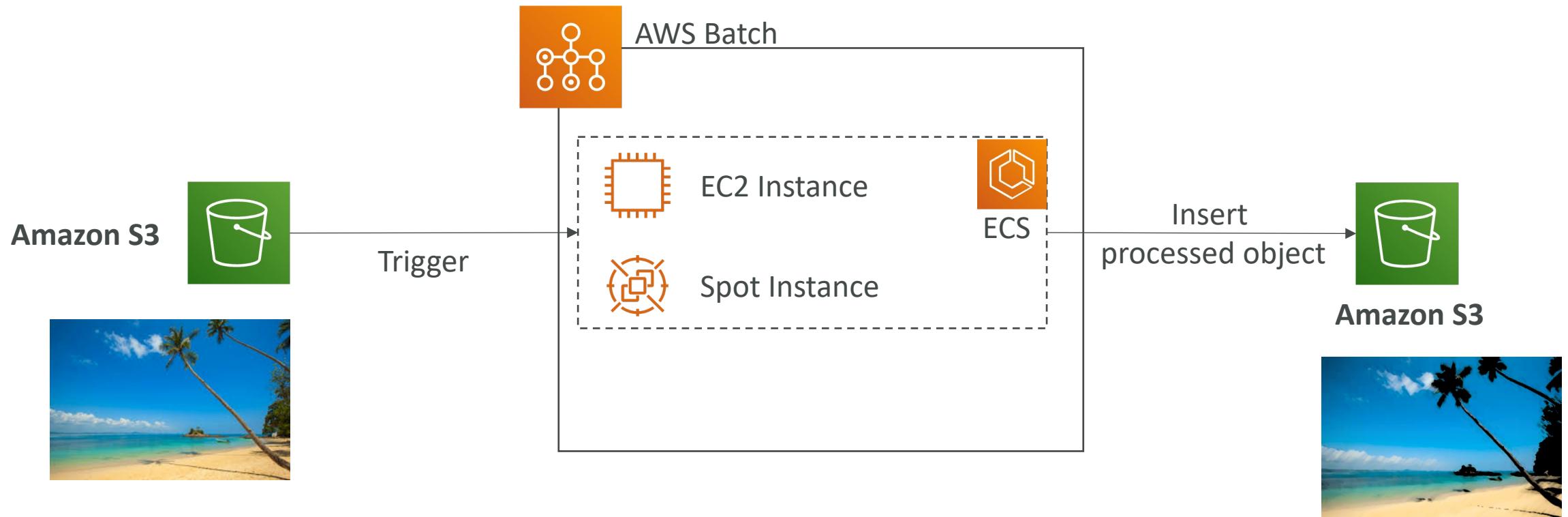


AWS Batch



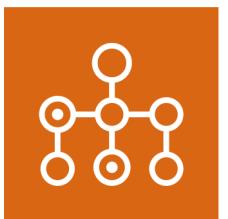
- Fully managed batch processing at any scale
- Efficiently run 100,000s of computing batch jobs on AWS
- A “batch” job is a job with a start and an end (opposed to continuous)
- Batch will dynamically launch **EC2 instances** or **Spot Instances**
- AWS Batch provisions the right amount of compute / memory
- You submit or schedule batch jobs and AWS Batch does the rest!
- Batch jobs are defined as **Docker images** and run on **ECS**
- Helpful for cost optimizations and focusing less on the infrastructure

AWS Batch – Simplified Example



Batch vs Lambda

- Lambda:
 - Time limit
 - Limited runtimes
 - Limited temporary disk space
 - Serverless
- Batch:
 - No time limit
 - Any runtime as long as it's packaged as a Docker image
 - Rely on EBS / instance store for disk space
 - Relies on EC2 (can be managed by AWS)





Amazon AppFlow

- Fully managed integration service that enables you to securely transfer data between **Software-as-a-Service (SaaS) applications and AWS**
- Sources: Salesforce, SAP, Zendesk, Slack, and ServiceNow
- Destinations: AWS services like Amazon S3, Amazon Redshift or non-AWS such as SnowFlake and Salesforce
- Frequency: on a schedule, in response to events, or on demand
- Data transformation capabilities like filtering and validation
- Encrypted over the public internet or privately over AWS PrivateLink
- Don't spend time writing the integrations and leverage APIs immediately

White Papers and Architectures

Well Architected Framework, Disaster Recovery, etc...

Section Overview

- Well Architected Framework Whitepaper
- Well Architected Tool
- AWS Trusted Advisor
- Reference architectures resources (for real-world)
- Disaster Recovery on AWS Whitepaper

Well Architected Framework

General Guiding Principles

- <https://aws.amazon.com/architecture/well-architected>
- Stop guessing your capacity needs
- Test systems at production scale
- Automate to make architectural experimentation easier
- Allow for evolutionary architectures
 - Design based on changing requirements
- Drive architectures using data
- Improve through game days
 - Simulate applications for flash sale days

Well Architected Framework

6 Pillars

- 1) Operational Excellence
 - 2) Security
 - 3) Reliability
 - 4) Performance Efficiency
 - 5) Cost Optimization
 - 6) Sustainability
-
- They are not something to balance, or trade-offs, they're a synergy

AWS Well-Architected Tool



- Free tool to **review your architectures** against the 6 pillars Well-Architected Framework and **adopt architectural best practices**
- How does it work?
 - Select your workload and answer questions
 - Review your answers against the 6 pillars
 - Obtain advice: get videos and documentations, generate a report, see the results in a dashboard
- Let's have a look: <https://console.aws.amazon.com/wellarchitected>

Name	Overall status	High risks	Medium risks	Improvement status	Last updated
Internal Employee Portal	Answered	13	2	None	Nov 24, 2018 3:40 PM UTC-8
Mobile app - Android	Answered	9	1	None	Nov 24, 2018 3:43 PM UTC-8
Mobile app - iOS	Answered	0	1	None	Nov 24, 2018 3:49 PM UTC-8
Retail Website- EU	Unanswered	0	0	None	Nov 24, 2018 3:52 PM UTC-8
Retail Website- North America	Unanswered	0	0	None	Nov 24, 2018 3:19 PM UTC-8

<https://aws.amazon.com/blogs/aws/new-aws-well-architected-tool-review-workloads-against-best-practices/>



Trusted Advisor

- No need to install anything – high level AWS account assessment
- Analyze your AWS accounts and provides recommendation on 5 categories
 - Cost optimization
 - Performance
 - Security
 - Fault tolerance
 - Service limits

Checks

- ▶ ✓ **Amazon EBS Public Snapshots**
Checks the permission settings for your Amazon Elastic
0 EBS snapshots are marked as public.
- ▶ ✓ **Amazon RDS Public Snapshots**
Checks the permission settings for your Amazon Relation
public.
0 RDS snapshots are marked as public.
- ▶ ✓ **IAM Use**
This check is intended to discourage the use of root acce
At least one IAM user has been created for this account.

Trusted Advisor – Support Plans

7 CORE CHECKS

Basic & Developer Support plan

- S3 Bucket Permissions
- Security Groups – Specific Ports Unrestricted
- IAM Use (one IAM user minimum)
- MFA on Root Account
- EBS Public Snapshots
- RDS Public Snapshots
- Service Limits

FULL CHECKS

Business & Enterprise Support plan

- Full Checks available on the 5 categories
- Ability to set CloudWatch alarms when reaching limits
- Programmatic Access using AWS Support API

More Architecture Examples

- We've explored the most important architectural patterns:
 - **Classic:** EC2, ELB, RDS, ElastiCache, etc...
 - **Serverless:** S3, Lambda, DynamoDB, CloudFront, API Gateway, etc...
- If you want to see more AWS architectures:
- <https://aws.amazon.com/architecture/>
- <https://aws.amazon.com/solutions/>

Exam Review & Tips

State of learning checkpoint

- Let's look how far we've gone on our learning journey
- <https://aws.amazon.com/certification/certified-solutions-architect-associate/>

Practice makes perfect

- If you're new to AWS, take a bit of AWS practice thanks to this course before rushing to the exam
 - The exam recommends you to have one or more years of hands-on experience on AWS
 - Practice makes perfect!
-
- If you feel overwhelmed by the amount of knowledge you just learned, just go through it one more time

Proceed by elimination

- Most questions are going to be scenario based
 - For all the questions, rule out answers that you know for sure are wrong
 - For the remaining answers, understand which one makes the most sense
-
- There are very few trick questions
 - Don't over-think it
 - If a solution seems feasible but highly complicated, it's probably wrong

Skim the AWS Whitepapers

- You can read about some AWS White Papers here:
 - Architecting for the Cloud: AWS Best Practices
 - AWS Well-Architected Framework
 - AWS Disaster Recovery (<https://aws.amazon.com/disaster-recovery/>)
- Overall we've explored all the most important concepts in the course
- It's never bad to have a look at the whitepapers you think are interesting!

Read each service's FAQ

- FAQ = Frequently asked questions
- Example: <https://aws.amazon.com/vpc/faqs/>
- FAQ cover a lot of the questions asked at the exam
- They help confirm your understanding of a service

Get into the AWS Community

- Help out and discuss with other people in the course Q&A
 - Review questions asked by other people in the Q&A
 - Do the practice test in this section
-
- Read forums online
 - Read online blogs
 - Attend local meetups and discuss with other AWS engineers
 - Watch re-invent videos on Youtube (AWS Conference)

How will the exam work?

- You'll have to register online at <https://www.aws.training/>
- Fee for the exam is 150 USD
- Provide one identity documents (ID, Passport, details are in emails sent to you...)
- No notes are allowed, no pen is allowed, no speaking
- 65 questions will be asked in 130 minutes
- Use the “Flag” feature to mark questions you want to re-visit
- At the end you can optionally review all the questions / answers

- To pass you need a score of at least 720 out of 1000
- You will know within 5 days if you passed / failed the exams (most of the time less)
- You will know the overall score a few days later (email notification)
- You will not know which answers were right / wrong
- If you fail, you can retake the exam again 14 days later

Congratulations!

Congratulations!

- Congrats on finishing the course!
- I hope you will pass the exam without a hitch ☺
- If you haven't done so yet, I'd love a review from you!
- If you passed, I'll be more than happy to know I've helped
 - Post it in the Q&A to help & motivate other students. Share your tips!
 - Post it on LinkedIn and tag me!
- Overall, I hope you learned how to use AWS and that you will be a tremendously good AWS Solutions Architect