Section 3: AWS Overview

- Global services:
    - IAM, Route 53 (DNS service)
    - CloudFront (Content Delivery Network)
    - WAF

Section 4: IAM & AWS CLI

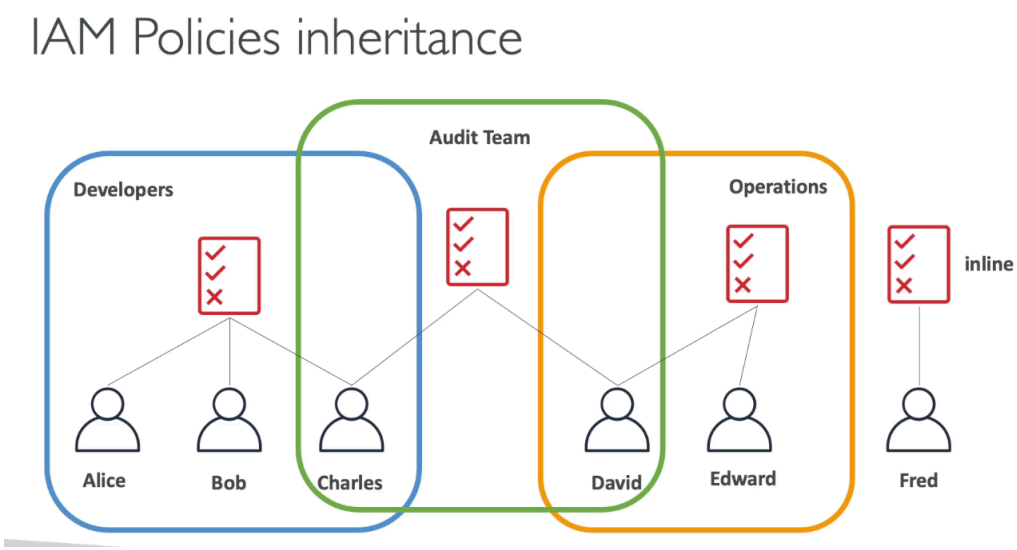
IAM

- Root account created by default, but shouldn't be used or shared
- Users are people within the org and can be grouped
- Groups only contain users, not other groups
- Users don't have to be in a group and users can be in multiple groups

Permissions

- Policy document (JSON) defines what users or groups can do
    - Least privilege principle

Policies



Password Policy

- Strong passwords = high security
- Allow IAM users to change passwords
- Require users to change password after some time
- Prevent password reuse

MFA

- Protect root and IAM users
- MFA = password + security device

Access AWS

- Console, CLI, SDK
    - CLI is a tool that allows AWS interaction in terminal
        - CloudShell is the in console terminal
    - Direct access to public APIs of AWS services
    - SDK is language specific APIs that embed within application to access and manage AWS services
- Access keys generated via Console

IAM Security Tools

- IAM Credentials report (account level)
    - Report that lists all account users and status of credentials
- IAM Access advisor (user level)
    - Shows the service permissions granted to a user and when those services were last accessed

IAM Guidelines & Best Practices

- Don’t use root account
- One physical user = one AWS user
- Assign users to groups and assign permissions to groups
- Strong password and MFA
- Create and use roles for giving permissions to AWS services

Shared Responsibility Model for IAM

- You: Users, groups, roles, policies management and monitoring

Summary

IAM Section – Summary

- **Users:** mapped to a physical user, has a password for AWS Console
- **Groups:** contains users only
- **Policies:** JSON document that outlines permissions for users or groups
- **Roles:** for EC2 instances or AWS services
- **Security:** MFA + Password Policy
- **AWS CLI:** manage your AWS services using the command-line
- **AWS SDK:** manage your AWS services using a programming language
- **Access Keys:** access AWS using the CLI or SDK
- **Audit:** IAM Credential Reports & IAM Access Advisor

Section 5: EC2 Fundamentals

EC2 Basics

- Elastic Compute Cloud = infrastructure as a service
    - Rent VMs (EC2)
    - Storing data on virtual drives (EBS)
    - Distributing load across machines (ELB)
    - Scaling services using auto scaling group (ASG)

EC2 sizing & config options

- OS, CPU, RAM, Storage, network, WAF, bootstrap script
    - Bootstrapping means launching commands when machine starts (run only once at instance first start)
    - Runs as root user (sudo)

EC2 User data

- Bootstrap instances via user script
    - Install updates, software, download files…
- Bootstrapping means launching commands when machine starts (run only once at instance first start)
- Runs as root user (sudo)

EC2 Instance Types

- Naming convention: m5.2xlarge
    - M: instance class

- 5: generation
- 2xlarge: size within the instance class

General Purpose
- Great all around workload: balance between compute, memory, networking

Compute Optimized
- Great for compute intensive tasks that require high performance processors

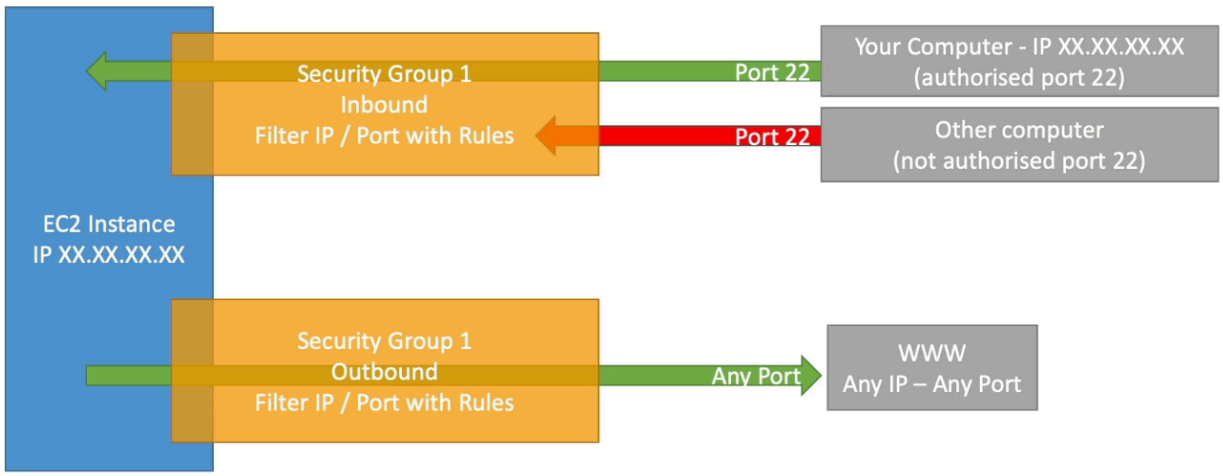
Memory Optimized
- Fast performance for workloads that process large data sets in memory

Storage Optimized
- Storage intensive tasks that require high sequential read and write access to large data sets in local storage

Introduction to Security Groups

- Control how traffic is allowed in/out of EC2 instances
- SG only contain ALLOW rules
- SG rules can reference by IP or SG
- SG act as firewall on EC2 instances
    - Regulate:
        - Access to ports
        - Authorized IP ranges
        - Control inbound and outbound network traffic

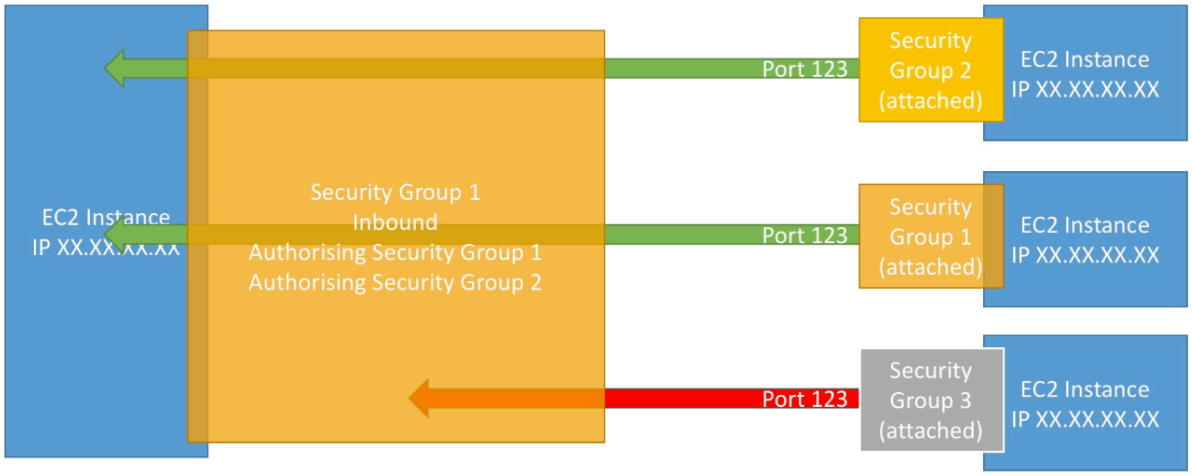
Security Groups Diagram



SG Good to know

- Can be attached to multiple instances
- Locked down to region/VPC combination
- Lives “outside” EC2 - if traffic blocked, EC2 instance won’t see it
- Good to maintain one separate SG for SSH access
- If application not accessible (timeout), then it’s security group issue
- Connection refused error, then it’s application error or not launched
- All inbound traffic is BLOCKED by default
- All outbound traffic is authorized by default

Referencing other security groups Diagram



Classic Ports to Know

- 22 == SSH - log into Linux instance
- 21 = FTP (file transfer protocol) - upload files to file share
- 22 = SFTP (secure file transfer protocol) - upload files using SSH
- 80 = HTTP - access unsecure websites
- 443 = HTTPS - access secure websites
- 3389 = RDP (remote desktop protocol) - log into windows instance

EC2 Instance Purchase Options

- On demand instances
  - Short workload, predictable pricing, pay by second (Linux or windows), after first minute
  - All other OS - billing per hour

    - Highest cost, no upfront payment, no long-term commitment
    - Recommended for short-term and un-interrupted workloads where you can't predict how the application will behave

- Reserved (1 & 3 years): long workloads
    - Convertible reserved instances - long workloads with flexible instances
        - Can change instance type, family, OS, scope and tenancy
    - Up to 72% discount compared to on demand
    - Reserve specific instance attributes (instance type, region, tenancy, OS)
    - Payment options: no upfront, partial, all upfront
    - Reserved instance's scope - regional or zonal (reserve capacity in AZ)
    - Recommended for steady state usage applications
    - Buy or sell in reserved instance marketplace

- Savings Plans (1 & 3 years) - commitment to an amount of usage, long workload
    - Commit to certain type of usage
    - Usage beyond EC2 savings plan billed at on demand price
    - Locked to specific instance family and region
    - Flexible across instance size, OS, tenancy

- Spot instances - short workloads, cheap, can lose instances if max price less than current spot price (less reliable)
    - Most cost efficient
    - Workloads that are resilient to failure
        - NOT DB OR CRITICAL JOBS

- Dedicated hosts - book entire physical server, control instance placement
    - Compliance requirements or licenses
    - Purchasing by on demand or reserved
        - Most expensive

- Dedicated instances - no other customers will share hardware
    - May share hardware with other instances in same account
    - No control over instance placement (can move after stop/start)
    - You get your own instance on your own hardware

- Capacity reservations - reserve capacity in specific AZ for any duration
    - No time commitment or discounts
    - Charged at on demand rate whether you run instances
    - For short-term, uninterrupted workloads in specific AZ

Which purchasing option is right for me?

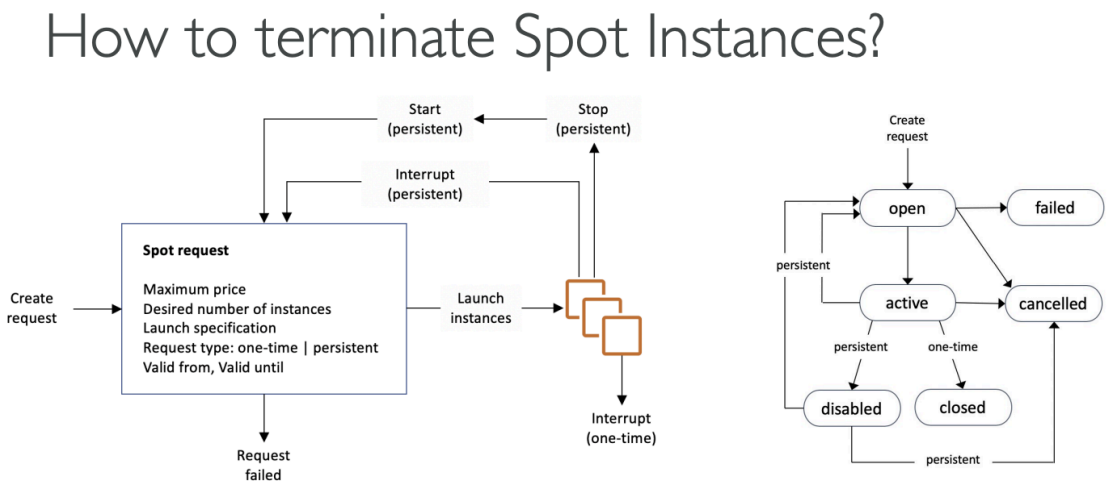
- **On demand:** coming and staying in resort whenever we like, we pay the full price
- **Reserved:** like planning ahead and if we plan to stay for a long time, we may get a good discount.
- **Savings Plans:** pay a certain amount per hour for certain period and stay in any room type (e.g., King, Suite, Sea View, …)
- **Spot instances:** the hotel allows people to bid for the empty rooms and the highest bidder keeps the rooms. You can get kicked out at any time
- **Dedicated Hosts:** We book an entire building of the resort
- **Capacity Reservations:** you book a room for a period with full price even you don't stay in it

EC2 Spot Instance Requests

- Define max spot price to get instance while current spot price < max
  - Hourly spot price varies
  - If current spot price > max price, stop or terminate instance with 2 minute grace period
- Spot Block
  - "Block" spot instance during a specified time frame (1-6 hours) without interruptions
  - Rare instances where spot instance is reclaimed
- Used for batch jobs, data analysis, or workloads that are resilient to failures, not critical workloads or DB

How to terminate Spot Instances?

- One time or persistent spot requests. If it is one time, the spot instance will start and the spot request is gone. If persistent, the spot request will remain and if an instance is terminated, another will be launched.
- Cancel spot requests in open, active, or disabled state
- Canceling stop request does not terminate instances, must cancel Spot request then stop instances



Spot Fleets

- Set of spot instances + on demand instances
- Automatically request spot instances with lowest price
- Fleet will try to meet target capacity with price constraints
    - Define possible launch pools: instance type, OS, AZ
    - Can have multiple launch pools for fleet to choose
    - Spot fleet stops launching instances when reaching capacity or max cost
- Strategies to allocate Spot Instances:
    - lowestPrice: from pool with lowest price (cost optimization, short workload)
    - Diversified: distributed across all pools (great for availability, log workloads)
    - capacityOptimized: pool with optimal capacity for number of instances
    - priceCapacityOptimized (recommended): pools with highest capacity available, then select pool with lowest price (best choice for most workloads)

Section 6: EC2 - SAA Level

Private vs Public IP (IPv4)

- Public IP:
    - Machine can be identified on internet
    - Must be unique across whole web, can be geo-located easily
- Private IP:
    - Machine can only be identified on private network only

- IP must be unique across private network, but 2 different private networks can have same IPs
- Machines connect to internet via internet gateway as proxy
- Only specified range of IPs can be used as private IP

Elastic IP

- When you stop and start EC2 instance, it can change its public IP
- If you need to have a fixed public IP for instance, you need Elastic IP
- Elastic IP is public IPv4 IP you own as long as you don't delete it
- Attached to 1 instance at a time
- With Elastic IP, you can mask the failure of an instance by rapidly remapping the address to another instance in your account
- Only 5 Elastic IP in account (can be increased)
- Avoid using, instead use random public IP and register DNS name
    - LB and no public IP
- Default EC2:
    - Private IP for AWS, public IP for internet
- SSH into EC2:
    - Can't use private IP because not on same network, only public IP
- If machine is stopped and started, public IP can change

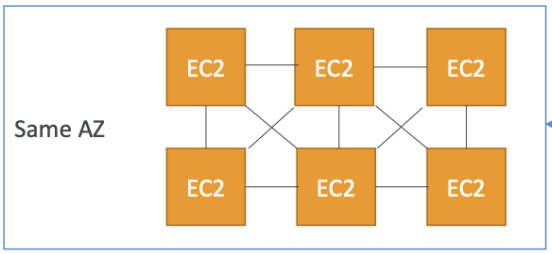
Placement Groups

- Control over EC2 instance placement
- Defined using placement groups
- Specify strategy:
    - Cluster: cluster instances in low latency group in single AZ
    - Spread: spreads instances across underlying hardware (7 instances per group per AZ) for critical apps
    - Partition: spreads instances across many different partitions within AZ; scales to 100s of instances per group

Cluster

- Single AZ, great network with enhanced networking enabled
- Con: if AZ fails, all instances fails at same time
- Use cases: big data to complete very fast, app needs low latency and high network throughput

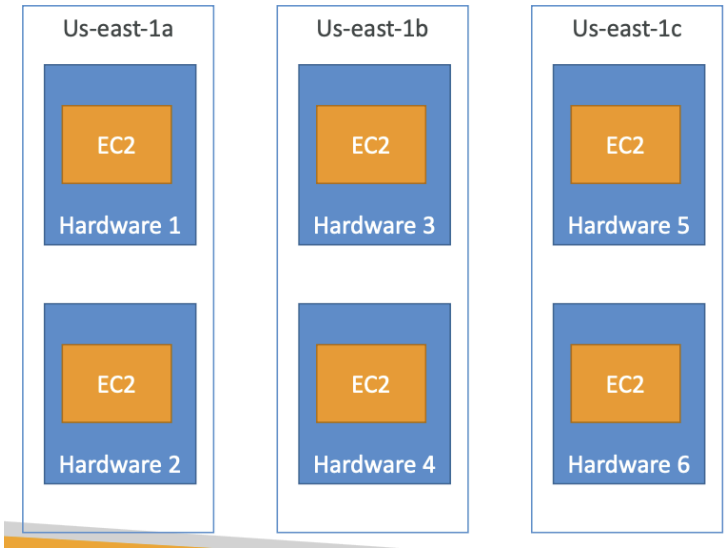
Placement Groups
Cluster



- Pros: Great network (10 Gbps bandwidth between instances with Enhanced Networking enabled - recommended)
- Cons: If the AZ fails, all instances fails at the same time
- Use case:
  - Big Data job that needs to complete fast
  - Application that needs extremely low latency and high network throughput

Spread

Placement Groups
Spread



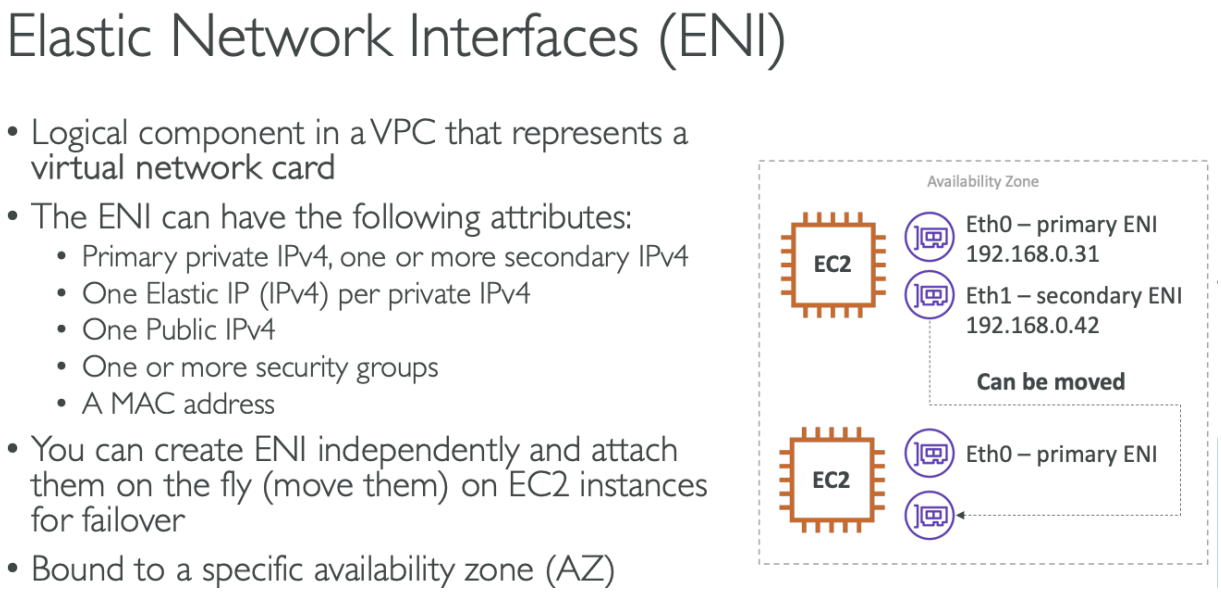
- Pros:
  - Can span across Availability Zones (AZ)
  - Reduced risk is simultaneous failure
  - EC2 Instances are on different physical hardware
- Cons:
  - Limited to 7 instances per AZ per placement group
- Use case:
  - Application that needs to maximize high availability
  - Critical Applications where each instance must be isolated from failure from each other

- Span multiple AZ, EC2 instances on different hardware for reduced risk of failure
- Limits 7 instances per AZ per placement group
- Use case: High availability, critical apps for isolated hardware

Partition

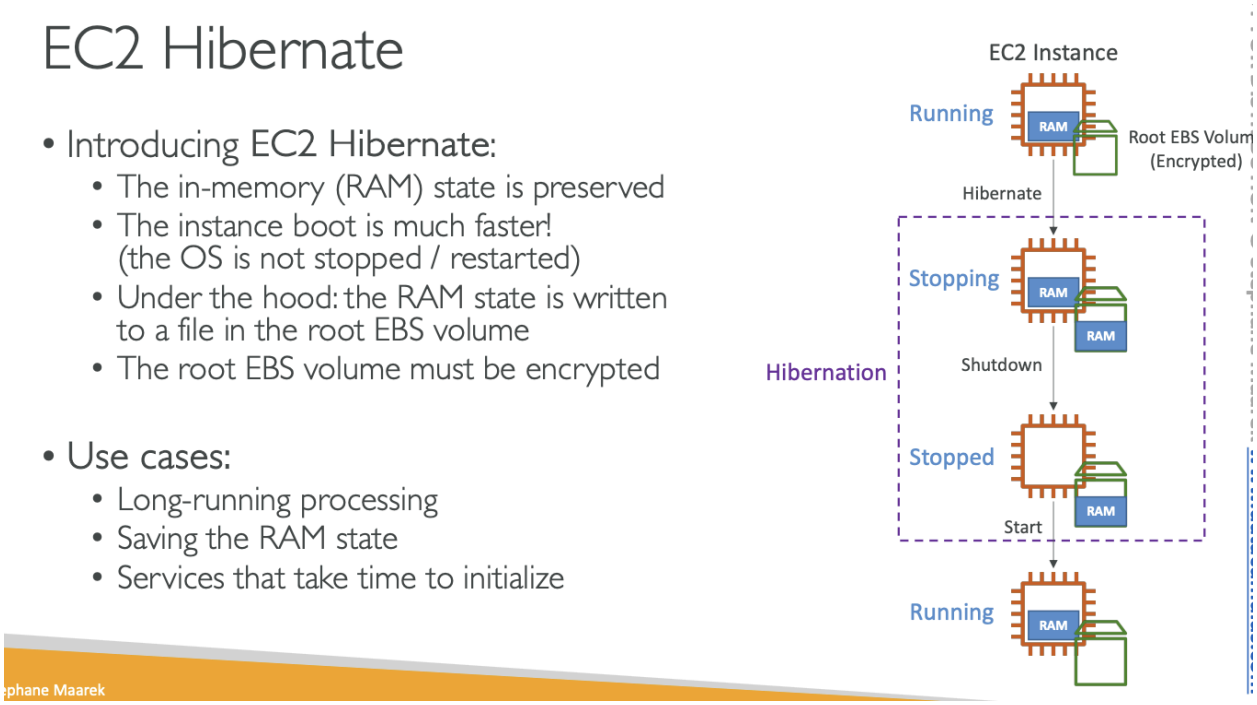
- Up to 7 partitions per AZ, spanning across multiple AZ in same region - up to 100s of instances
- Instances in partition do not share hardware with other partitions
    - Partition failure does not affect other partitions
- Metadata used by EC2 instances to get partition information
- Use case: Kafka

Elastic Network Interface (ENI)



- Component in VPC that represents virtual network card
- Attributes:
    - Primary private IPv4, 1+ secondary IPv4
    - 1 elastic IPv4 per IPv4
    - 1 public IPv4
    - 1+ SG
    - MAC address
- Create ENI independently and attach on fly on EC2 instances for failover
- Bound to AZ

EC2 Hibernate



- Stop: data on disk is kept intact in next start
- Terminate: any EBS volumes (root) also set up to be destroyed is lost
- On start:
    - First start: OS boots and user data script is run
    - App starts, cache warmed
- Hibernate:
    - RAM state is preserved
    - Instance boot much faster (OS not stopped)
    - Under the hood: RAM state written to file at root EBS volume
    - Root EBS volume must be encrypted
- Use case: long running processes, save RAM state, services that take time to initialize

Good to know:

- Many instance families
- RAM size must be < 150 GB
- Instance size - not supported for bare metal instances
- Many AMI's
- Root volume - EBS only, encrypted
- Available for on demand, reserved and spot instances
- Not hibernated for more than 60 days

Section 7: EC2 Instance Storage

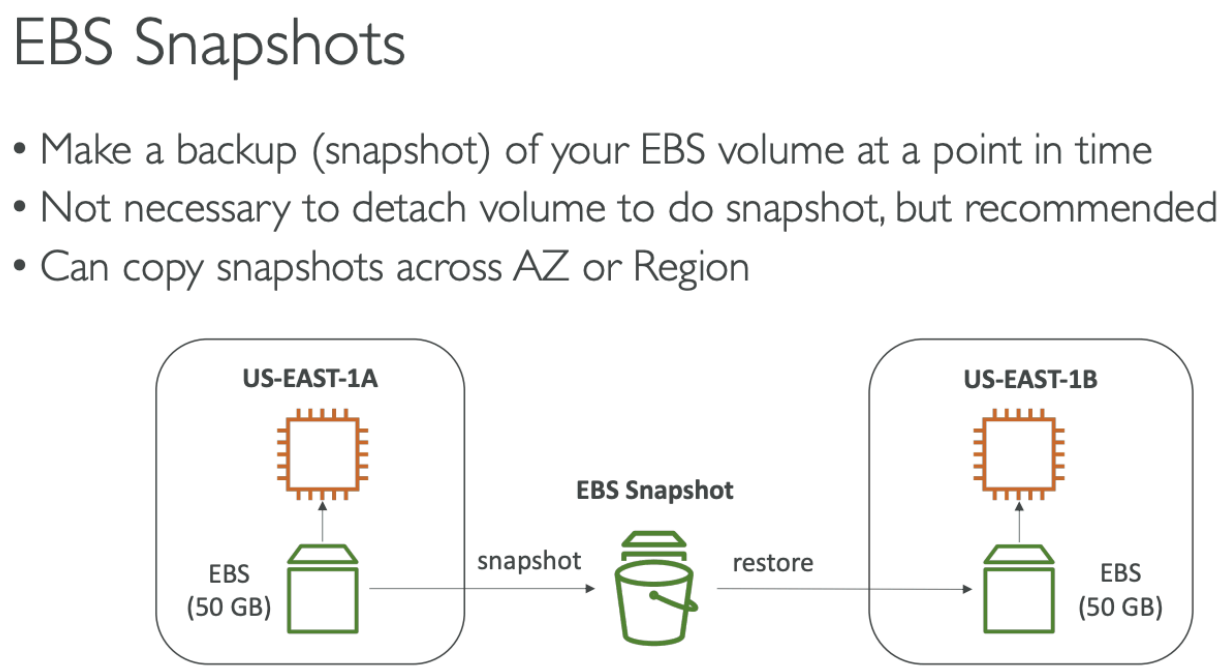
EBS Overview

- Network drive attached to instance while they run → “network USB stick”
    - Uses network to communicate to the instance, thus latency
    - Instances can persist data after termination
- Multi-attach available for some EBS types, but others can only be mounted to 1 instance at a time, bound by AZ
    - Locked by AZ
        - Snapshot to move volumes across
    - Can be detached and attached to another EC2 instance quickly
- Must provision capacity

Delete on Termination

- Default root EBS volume is deleted, other EBS volumes are not deleted; can be changed

EBS Snapshot



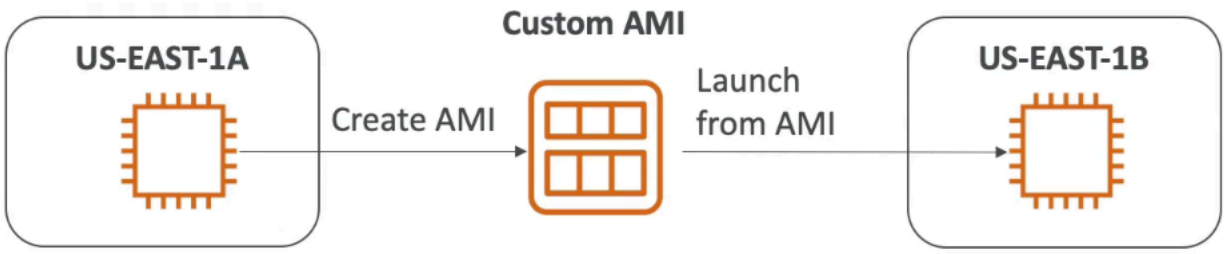
- Backup of EBS at point in time
    - Not necessary to detach volume to do snapshot, but recommended
    - Copy across AZ or region

Features

- Snapshot Archive
    - Move snapshot to archive tier 75% cheaper, takes within 24-72 hours for restoring archive
- Recycle Bin for EBS snapshots
    - Setup rules to retain deleted snapshots to recover from accidental deletion (1 day to 1 year retention)
- Fast Snapshot Restore (FSR)
    - Force full initialization of snapshot to have no latency on first use ($$$)

Amazon Machine Image (AMI) Overview

- Customization of EC2 instance
    - Add own software, configuration for faster boot/config
    - Start instance → customize → Stop instance → Build AMI → launch
- Built region specific
    - Launch from public AMI (AWS provided), personal, marketplace AMI



EC2 Instance Store

- High performance hardware disk
    - Better I/O performance
    - EC2 instance store lose storage if EC2 instance is stopped (ephemeral)
        - Good for cache, temporary content due to risk of data loss

EBS Volume Types

- 6 types
    - gp2/gp3 (SSD)
        - General purpose SSD volume that balances price and performance
        - GP3 can independently increase IOPS, while GP2 GP and IOPS linked

EBS Volume Types Use cases
General Purpose SSD

- Cost effective storage, low-latency
- System boot volumes, Virtual desktops, Development and test environments
- 1 GiB - 16 TiB
- gp3:
  - Baseline of 3,000 IOPS and throughput of 125 MiB/s
  - Can increase IOPS up to 16,000 and throughput up to 1000 MiB/s independently
- gp2:
  - Small gp2 volumes can burst IOPS to 3,000
  - Size of the volume and IOPS are linked, max IOPS is 16,000
  - 3 IOPS per GB, means at 5,334 GB we are at the max IOPS

    - IO1/IO2 block express (SSD)
        - High performance SSD for critical, low latency, high throughput workloads
            - For over 32k IOPS, need EC2 Nitro with IO1 or 2
        - EBS Multi Attach only available here
    - STL (HDD)
        - Low Cost HDD volume for frequently accessed throughput intensive workloads
    - SCL (HDD)
        - Lowest cost HDD volume for less frequently accessed workloads
- Only gp2/gp3 and IO1/IO2 block express can be used as boot volumes (where root OS is running)

Provisioned IOPS SSD

- Critical apps with sustained IOPS performance
    - DB workloads (sensitive to storage performance and consistency)

EBS Volume Types Use cases
Provisioned IOPS (PIOPS) SSD

- Critical business applications with sustained IOPS performance
- Or applications that need more than 16,000 IOPS
- Great for **databases workloads** (sensitive to storage perf and consistency)
- io1 (4 GiB - 16 TiB):
  - Max PIOPS: 64,000 for Nitro EC2 instances & 32,000 for other
  - Can increase PIOPS independently from storage size
- io2 Block Express (4 GiB – 64 TiB):
  - Sub-millisecond latency
  - Max PIOPS: 256,000 with an IOPS:GiB ratio of 1,000:1
- Supports EBS Multi-attach

EBS Volume Types Use cases

Hard Disk Drives (HDD)

- Cannot be a boot volume
- 125 GiB to 16 TiB
- Throughput Optimized HDD (st1)
  - Big Data, Data Warehouses, Log Processing
  - **Max throughput** 500 MiB/s – max IOPS 500
- Cold HDD (sc1):
  - For data that is infrequently accessed
  - Scenarios where lowest cost is important
  - **Max throughput** 250 MiB/s – max IOPS 250

EBS – Volume Types Summary

	General Purpose SSD volumes		Provisioned IOPS SSD volumes	
Volume type	gp3	gp2	io2 Block Express [3]	io1
Durability	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)		99.999% durability (0.001% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)
Use cases	• Transactional workloads • Virtual desktops • Medium-sized, single-instance databases • Low-latency interactive applications • Boot volumes • Development and test environments		Workloads that require: • Sub-millisecond latency • Sustained IOPS performance • More than 64,000 IOPS or 1,000 MiB/s of throughput	• Workloads that require sustained IOPS performance or more than 16,000 IOPS • I/O-intensive database workloads
Volume size	1 GiB - 16 TiB		4 GiB - 64 TiB [4]	4 GiB - 16 TiB
Max IOPS per volume (16 KiB I/O)	16,000		256,000 [5]	64,000
Max throughput per volume	1,000 MiB/s	250 MiB/s [1]	4,000 MiB/s	1,000 MiB/s [2]
Amazon EBS Multi-attach	Not supported		Supported	
NVMe reservations	Not supported		Supported	Not supported
Boot volume	Supported			

	Throughput Optimized HDD volumes	Cold HDD volumes
Volume type	st1	sc1
Durability	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	
Use cases	• Big data • Data warehouses • Log processing	• Throughput-oriented storage for data that is infrequently accessed • Scenarios where the lowest storage cost is important
Volume size	125 GiB - 16 TiB	
Max IOPS per volume (1 MiB I/O)	500	250
Max throughput per volume	500 MiB/s	250 MiB/s
Amazon EBS Multi-attach	Not supported	
Boot volume	Not supported	

EBS Multi Attach

- Attach same EBS volume to multiple EC2 instances in same AZ
- Each instance has full read/write permissions
- Up to 16 EC2 Instances at a time
- Must use a file system that's cluster aware

EBS Encryption

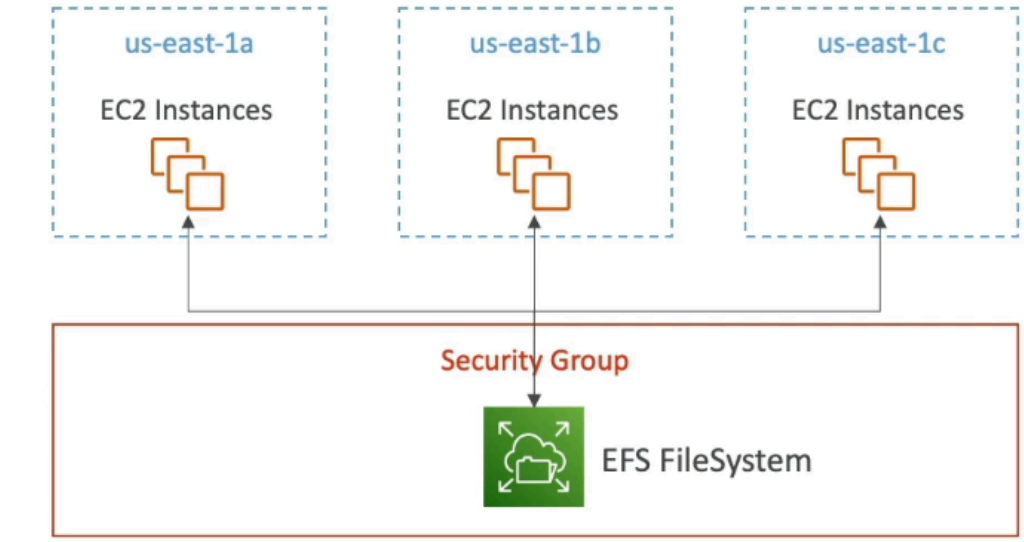
- When encrypting EBS:
    - Data at rest encryption
    - Data in flight between instance and volume encrypted
    - All snapshots encrypted
    - All volumes created from snapshot
- Encryption and decryption are handled transparently (do nothing)
    - Minimal latency
- KMS keys (AES 256)
- Copying unencrypted snapshot allows encryption
- Snapshots of encrypted volumes are encrypted

Encrypt unencrypted EBS volume

- Create EBS snapshot of volume
- Encrypt EBS snapshot (using copy)
- Create new EBS volume from snapshot (volume will also be encrypted)
- Attach encrypted volume to original instance

EFS (Elastic File System)

- Managed network file system mounted on EC2
- Works with EC2 instances in multi AZ, highly available, scalable
- Use cases: content management, web serving, data sharing
- NFSv4.1 protocol
- Uses SG to control access to EFS
- Only compatible with Linux based AMI
    - POSIX file system
    - Scales automatically, pay per use, no capacity planning
- Encryption at rest using KMS
- Performance mode and throughput mode settings

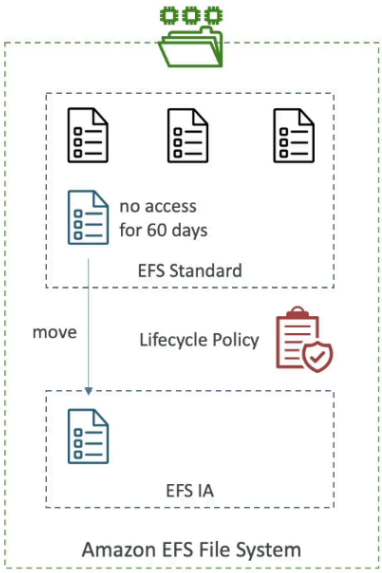


EFS – Performance & Storage Classes

- EFS Scale
  - 1000s of concurrent NFS clients, 10 GB+ /s throughput
  - Grow to Petabyte-scale network file system, automatically
- Performance Mode (set at EFS creation time)
  - General Purpose (default) – latency-sensitive use cases (web server, CMS, etc…)
  - Max I/O – higher latency, throughput, highly parallel (big data, media processing)
- Throughput Mode
  - Bursting – 1 TB = 50MiB/s + burst of up to 100MiB/s
  - Provisioned – set your throughput regardless of storage size, ex: 1 GiB/s for 1 TB storage
  - Elastic – automatically scales throughput up or down based on your workloads
    - Up to 3GiB/s for reads and 1GiB/s for writes
    - Used for unpredictable workloads

EFS – Storage Classes

- Storage Tiers (lifecycle management feature – move file after N days)
  - Standard: for frequently accessed files
  - Infrequent access (EFS-IA): cost to retrieve files, lower price to store. Enable EFS-IA with a Lifecycle Policy
- Availability and durability
  - Standard: Multi-AZ, great for prod
  - One Zone: One AZ, great for dev, backup enabled by default, compatible with IA (EFS One Zone-IA)
- Over 90% in cost savings



EFS vs EBS

- EBS volumes
  - One instance (except multi attach IO 1/IO 2)
  - Locked at AZ level
  - GP2: IO increases if disk size increases
  - GP3 & IO1: increase IO independently
  - Migrate EBS across AZ by taking snapshot and restore at another AZ
    - Backups use IO, so should be done when low traffic
  - Root EBS volumes terminated by default if EC2 instance is terminated
- EFS
  - Mounts to many instances across AZ
  - Only for linux
  - Higher price than EBS

Section 8: High Availability and Scalability: ELB & ASG

Scalability & High Availability

- Scalability means an application can handle greater loads by adapting
  - Vertical and horizontal (elasticity)
    - Vertical means upgrading size of instance (like DB)
      - Hardware limit
    - Horizontal means more instances

        - Implies distributed systems like web apps
    - Linked, but different to availability
- High Availability
    - Running app in at least 2 AZ (prevent data center loss)
    - Active or passive

Security Groups

Load Balancer Security Group:

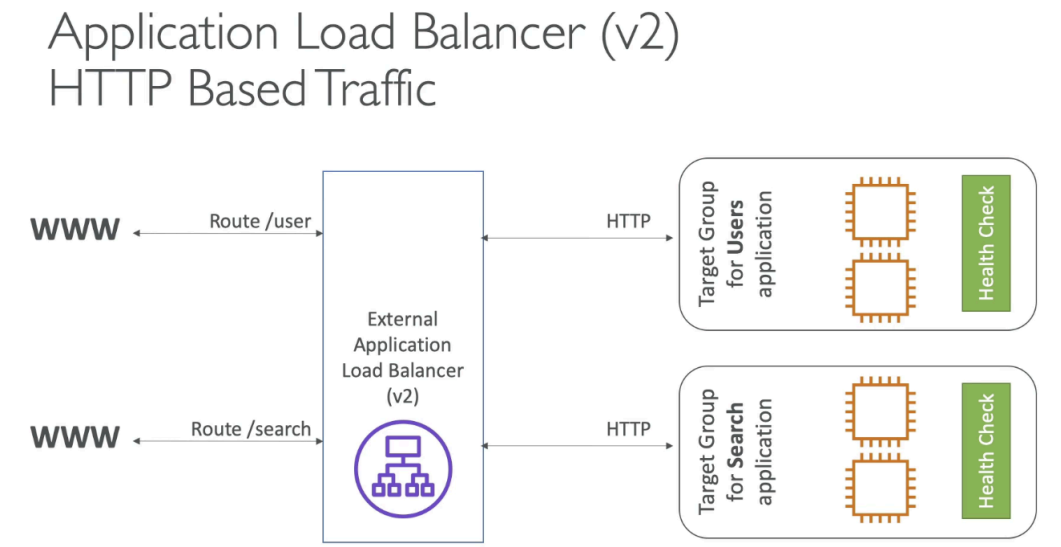
Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	Allow HTTP from an...
HTTPS	TCP	443	0.0.0.0/0	Allow HTTPS from a...

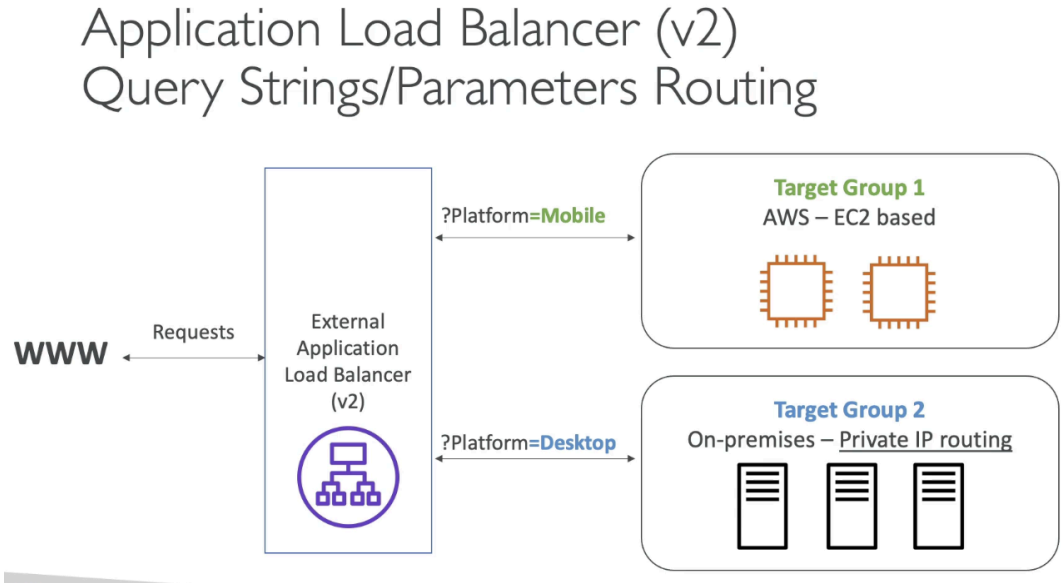
Application Security Group: Allow traffic only from Load Balancer

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	sg-054b5ff5ea02f2b6e (load-b	Allow Traffic only...

Application Load Balancer

- Layer 7 HTTP (supports HTTP/2 and websocket, and redirects from HTTP to HTTPS)
- Load balance to multiple HTTP apps across same or different machines
- Routing tables to different target groups
    - Based on path, hostname, query string or headers
- Good for container based apps
- Has port mapping to redirect to a dynamic port in ECS

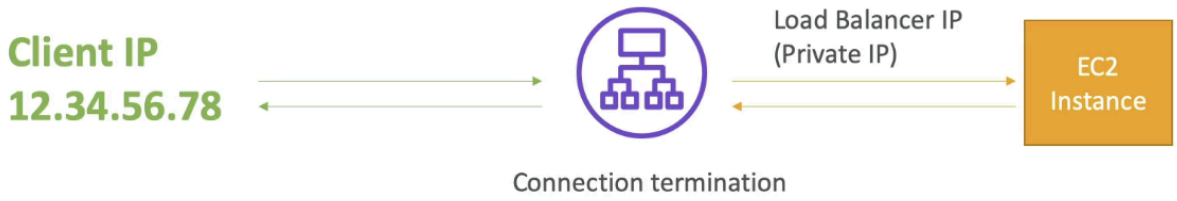




Target Groups

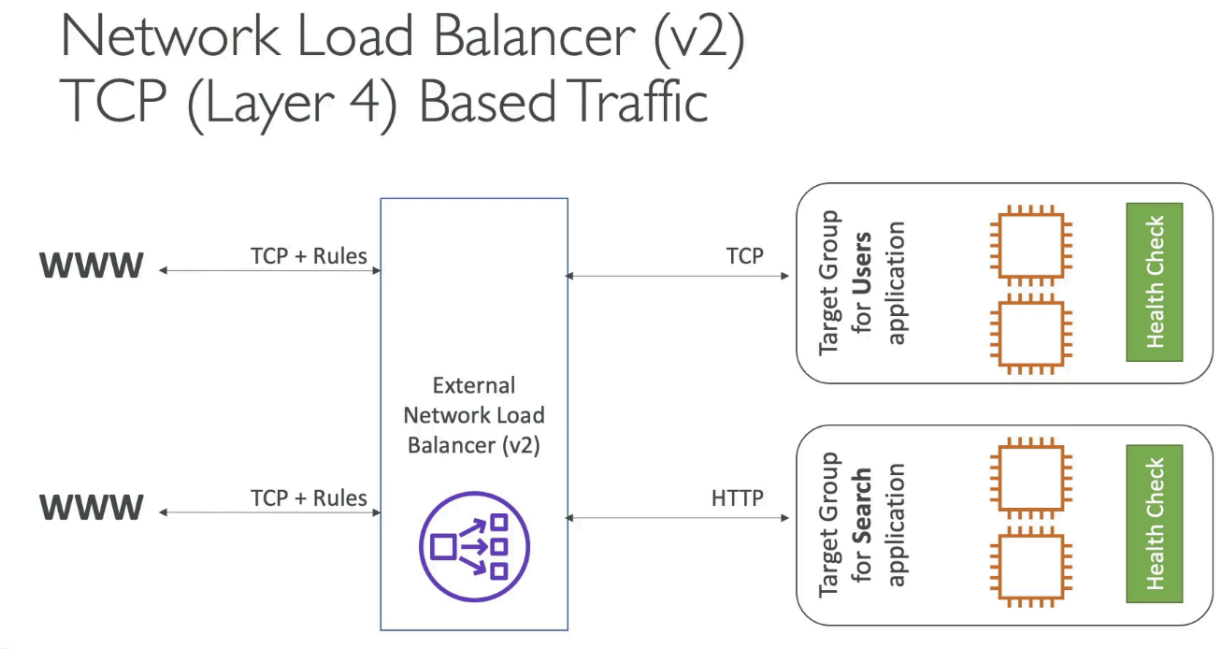
- EC2 instances (can be managed by auto scaling groups) - HTTP
- ECS tasks - HTTP
- Lambda functions - HTTP request translated into JSON event
- IP address - must be private
- ALB can route to multiple target groups and health checks done at target group level

Good to know



- Fixed hostname
- Application servers don't see the IP of client directly
    - True IP of client is inserted in header X-Forwarded-For
    - Can also get port, X-Forwarded-Port and proto (X-Forwarded-Proto)

Network Load Balancer



- Layer 4 allows to forward TCP/UDP traffic instances
    - High performance to handle millions of requests and 100 ms vs 400 ms for ALB
- One static IP per AZ and supports assigning elastic IP to each AZ
    - Expose app to a set of IPs, whitelisting specific IP

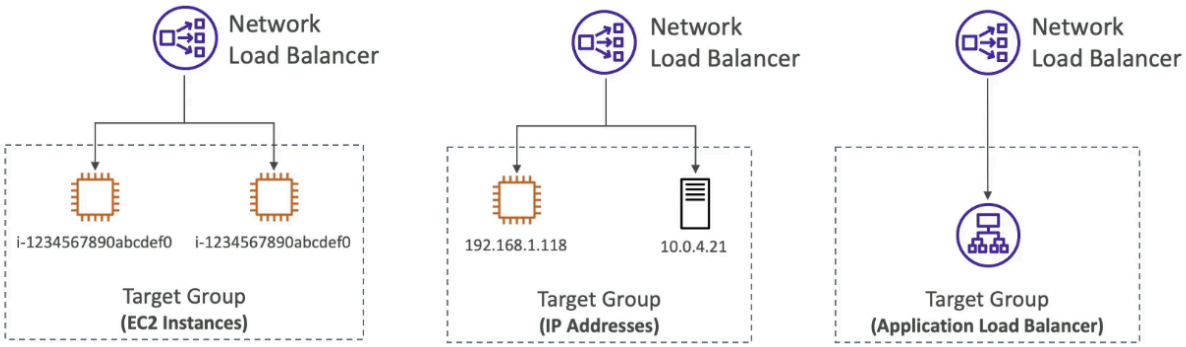
Target Groups

- EC2 instances
- IP addresses - must be private IP
- Application Load balancer

- Why do this? NLB gets fixed IP addresses and ALB gets all the rules around handling HTTP traffic
- Health checks by NLB support TCP, HTTP, and HTTPS protocols

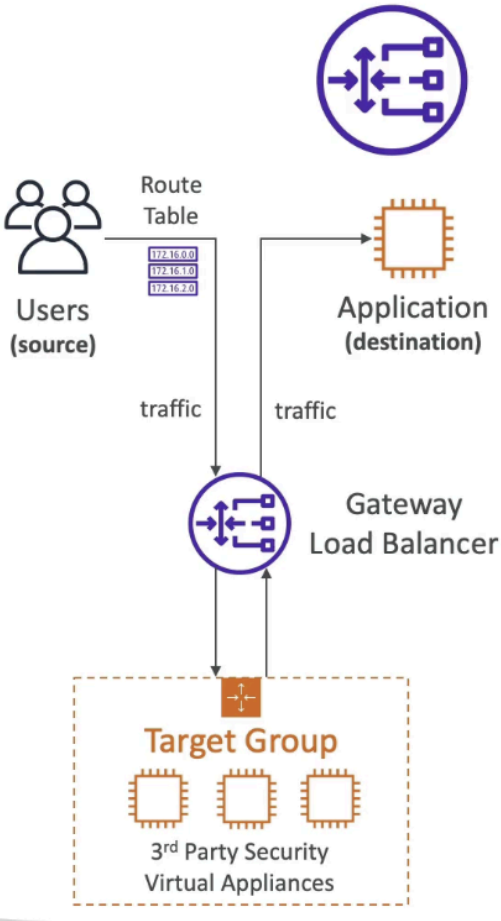
Network Load Balancer – Target Groups

- EC2 instances
- IP Addresses – must be private IPs
- Application Load Balancer



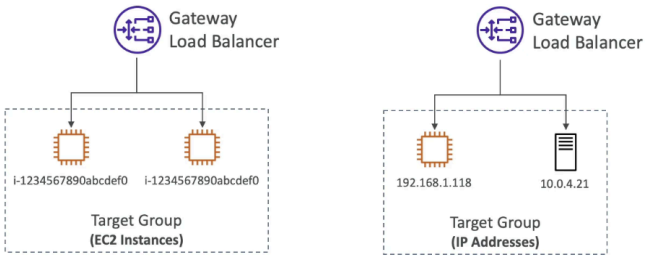
Gateway Load Balancer

- Deploy and manage 3rd party network apps in AWS
  - Firewalls, intrusion detection, etc…
- Operates at layer 3 (network layer) - IP Packets
- Combines:
  - Transparent network gateway - single entry/exit for all traffic
  - Load balancer - distributes traffic to all virtual apps
- Uses GENEVE protocol on port 6081

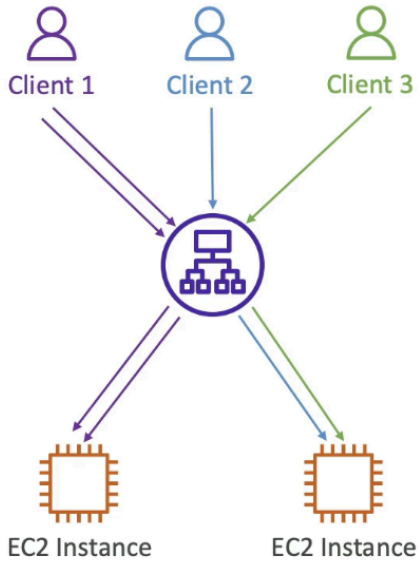


Gateway Load Balancer – Target Groups

- EC2 instances
- IP Addresses – must be private IPs



Elastic Load Balancer - Sticky Session (Session Affinity)



- Same client is always redirected to the same instance behind a LB
    - May bring imbalance to load over backend instances
    - Done at target group level
- A “cookie” used for stickiness and has an expiration date you control
- Use case: make sure the user doesn’t lose session data

Sticky Sessions - Cookie Names

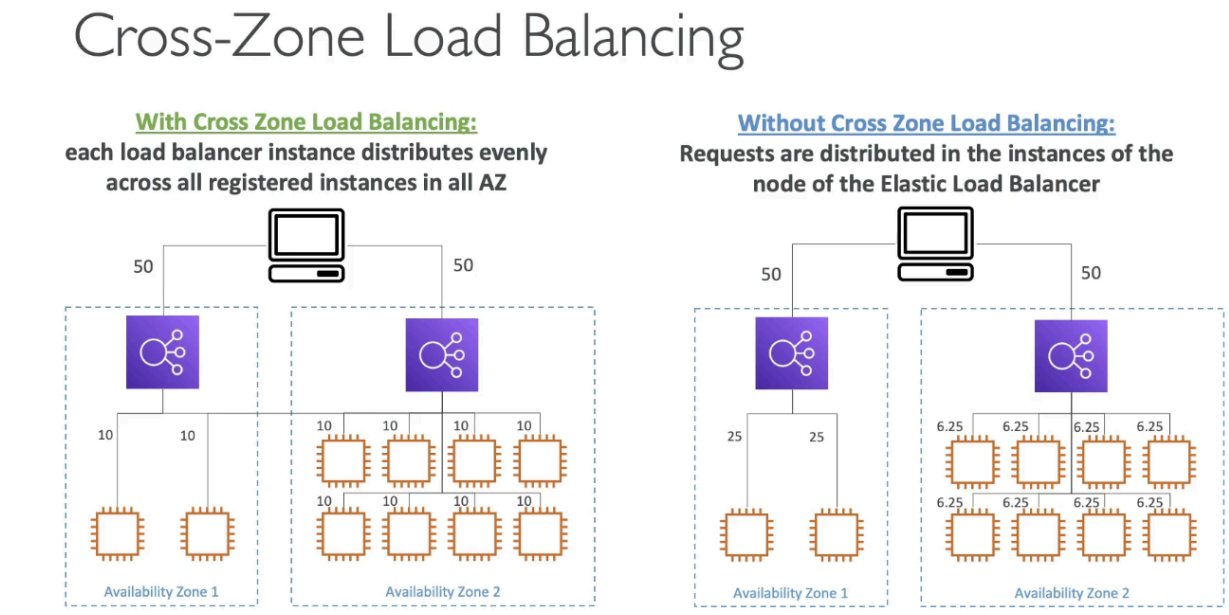
Application based cookies;

- Custom cookie
    - generated by target
    - Include any custom attributes required by app
    - Cookie name must be specified individually for each TG
- Application Cookie
    - Generated by LB, cookie name is AWSALBAPP

Duration based cookies

- Generated by LB, name is AWSALB for ALB, AWSELB for CLB

ELB - Cross Zone LB



- Each LB instance distributes evenly across all instances in all AZs
    - Client distributes evenly to ALBs, but each ALBs will evenly distribute regardless of AZ
- ALB
    - Enabled by default (disabled at TG level)
    - No charges for inter AZ data
- NLB & Gateway LB
    - Disabled by default, pay to use

ELB - SSL Certificates

SSL/TLS Basics

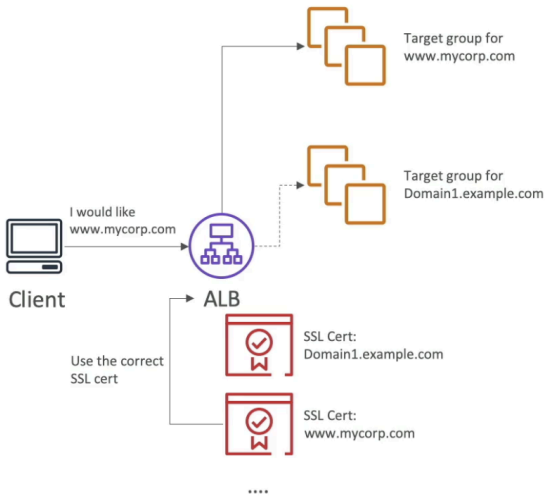
- SSL Cert allows traffic between client and LB to be encrypted in transit (in flight encryption)
    - SSL = secure sockets layer, used to encrypt connections
    - TLS = transport layer security, newer version, mainly used
- Public SSL certs issued by certificate authorities
- SSL Certs have expiration date (you set) and must be renewed

LB SSL Certificates

- Load balancer uses X.509 certificate and managed via ACM
- HTTPS listener:
    - Must specify default certificate
    - Clients can use SNI (server name indication) to specify hostname reached
    - Ability to specify security policy to support older versions
    -

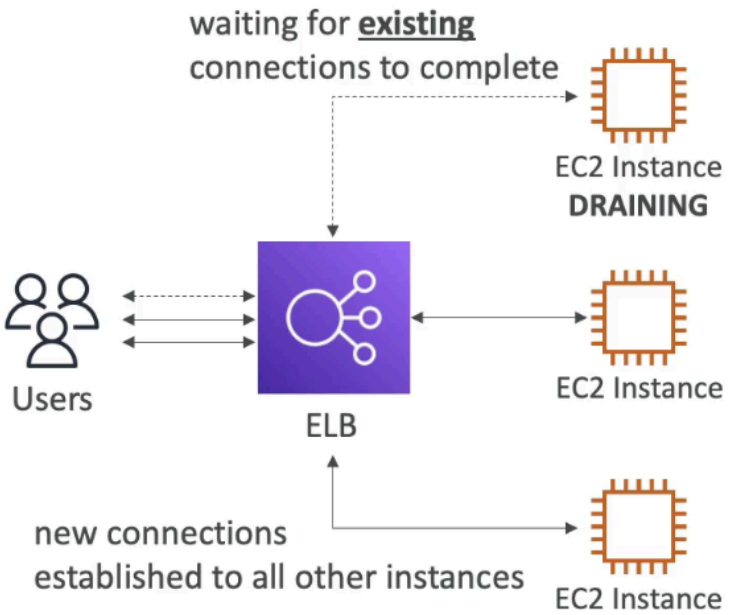
SSL - Server Name Indication (SNI)

- Solves problem of loading multiple SSL certificates to one web server (to serve multiple websites)
    - Newer protocol and requires client to indicate the hostname of the target server in initial SSL handshake
    - Only works for ALB, NLB, CloudFront



ELB - Connection Draining

- ALB/NLB called deregistration delay
- Time to complete “in-flight requests” while the instance is deregistering or unhealthy (default 300 seconds or disabled - only low values if requests are short)
    - Stops sending new requests to EC2 instance that is deregistering



Auto Scaling Groups (ASG)

- Goal to scale in/out instances to match load
    - Min and max EC2 instances and automatically register new instances to LB
    - Recreate EC2 instance in case of unhealthy instances
- Launch template for various attributes
- Can scale based on CloudWatch alarms based on metric that are computed for overall instances

Scaling Policies

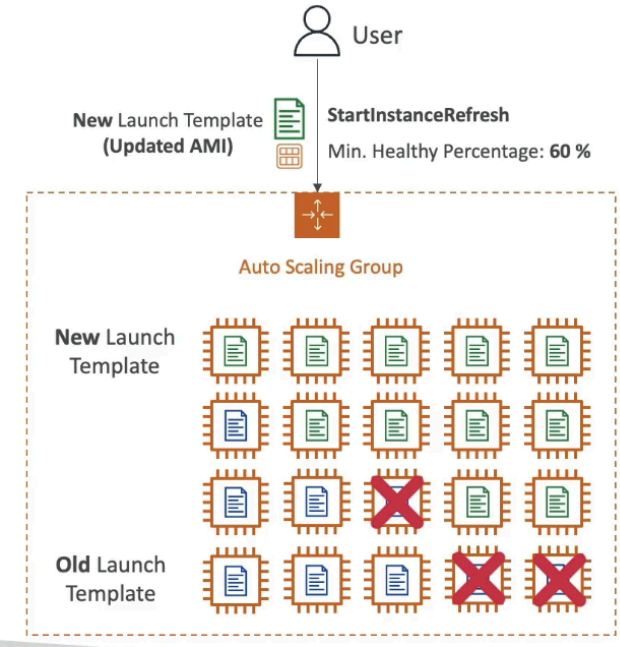
- Dynamic Scaling
    - Target Tracking Scaling
        - Based on metric
    - Simple/Step scaling
        - When CloudWatch alarm is triggered, add or remove
- Scheduled Scaling
    - Based on time and predictability of usage
- Predictive Scaling
    - Continuous forecast load and schedule scaling ahead

Metrics to scale on:

- CPU utilization: avg CPU usage across instances
- RequestCountPerTarget: number of request per instance is stable
- Average network in/out: if app is network bound
- Custom metric via CloudWatch

Scaling Cooldowns

- After scaling occurs, cooldown period of 300 seconds where ASG will not launch or terminate instances to allow metrics to stabilize



Instance Refresh

- Goal: update launch template and recreate all EC2 instances
- Setting of minimum healthy %: tells how many instances can be deleted over time
    - As new instances are created, old ones are terminated
    - Specify warm up time (how long until instance is ready to use)

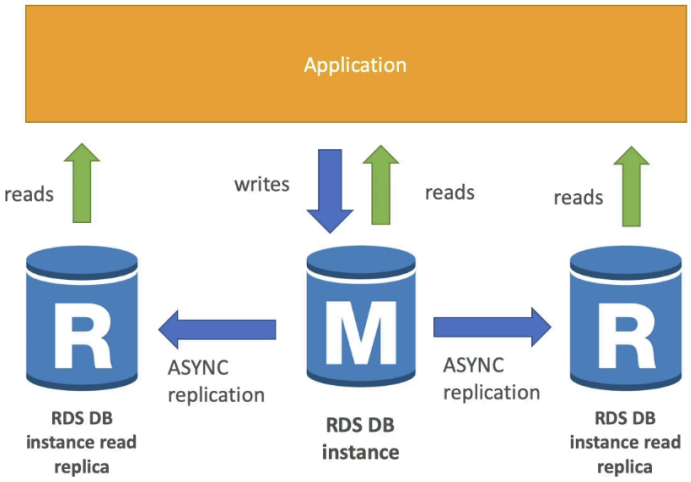
Section 9: AWS Fundamentals: RDS + Aurora + ElastiCache

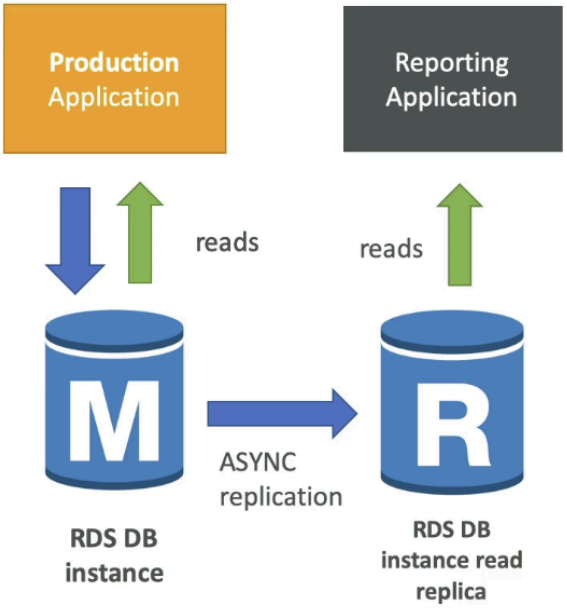
RDS (Relational DB Service) Overview

- Managed DB service using SQL
    - Create DB managed by AWS: PostGres, MySQL, etc…
- RDS vs DB on EC2
    - Managed by AWS for automated provisioning, OS patching, backups/restore, monitoring, read replicas, multi AZ for disaster recovery, scaling capabilities
    - Cannot SSH into RDS instances
- Storage Auto Scaling
    - Dynamically increase storage, done automatically
    - Avoid manual scaling and have a maximum storage threshold
    - Automatically modify if:
        - Free storage < 10% of allocated storage
        - Low storage lasts at least 5 minutes
        - 6 hours passed since last modification
    - Useful for apps with unpredictable workloads

RDS Read Replicas for read scalability

- Up to 15 read replicas within or cross AZ/region
- Async replication between main and read replicas





- Read replicas can be promoted to its own DB, out of replication mechanism

Use Cases:

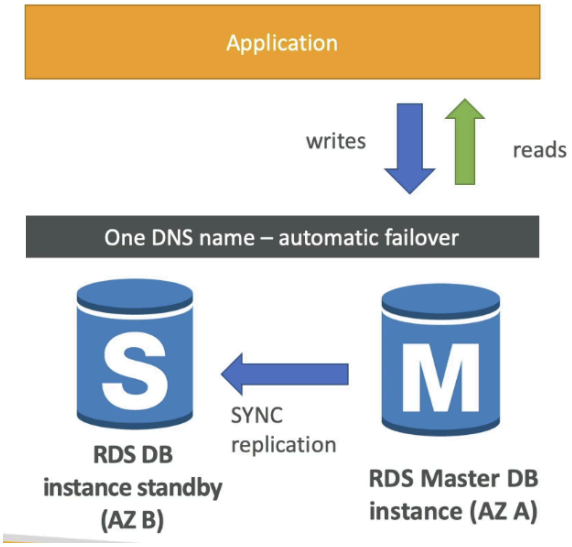
- Production DB on normal load. A new team wants to run analytics on DB, so a read replica is created to run new workloads

Read replicas network costs:

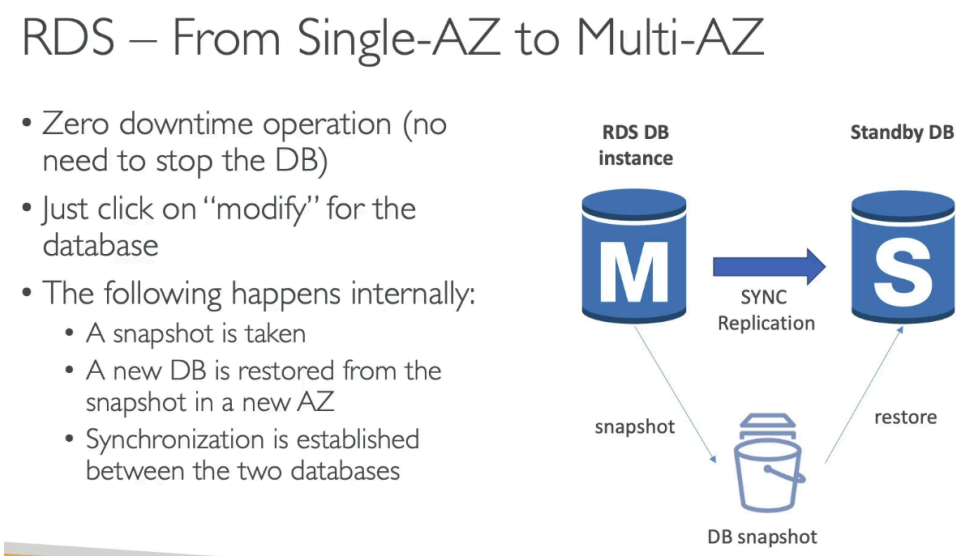
- For RDS replicas within the same region from one AZ to another AZ, no fee to move data. For cross region will incur replication fees

RDS Multi AZ (Disaster recovery)

- Synchronous replication to standby instance in a different AZ
- One DNS name - automatic failover to standby to increase availability
- Not used for scaling



From Single AZ to Multi AZ



- 0 downtime, just enable multi AZ and it will have a standby DB with synchronous replication
- Internal process:
    1. Snapshot taken
    2. New DB is restored from snapshot in new AZ
    3. Synchronization is established between the 2 DB

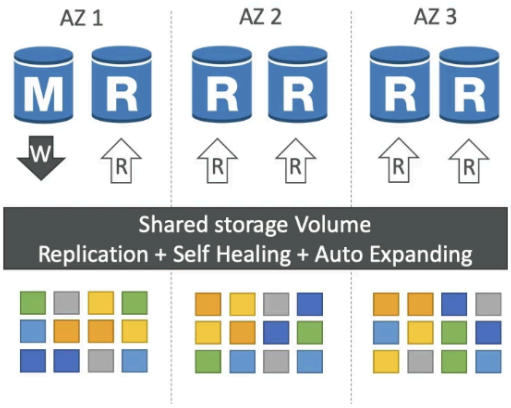
RDS Custom

- Managed Oracle and Microsoft SQL Server DB with OS and DB customization
    - RDS: automate setup, operations, scaling of DB in AWS
    - Custom:
        - Access to underlying DB and OS
            - Configure settings
            - Install patches
            - Enable native features
            - Access underlying EC2 instance using SSH or SSM Session Manager
    - De-activate automation mode to perform customization
        - Take DB snapshot before
- RDS vs RDS Custom
    - RDS: entire DB and OS owned by AWS
    - Custom: full admin access to underlying OS and DB

Amazon Aurora

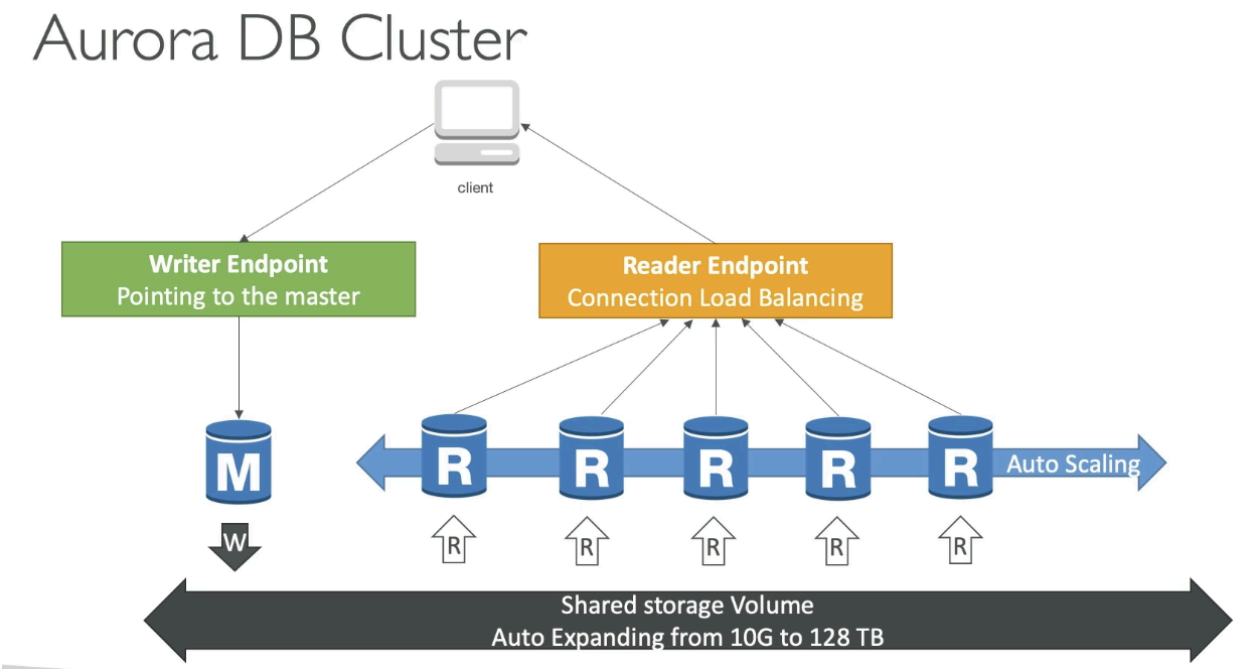
- Proprietary tech from AWS, but compatible w/ Postgres and MySQL
    - Cloud optimized, good performance over base postgres/mysql
- Automatically grows in increments of 10 GB up to 128TB
- 15 read replicas and faster than MySQL, failover is instantaneous, high availability
    - Slightly higher cost than RDS, but more efficient

High Availability and Read Scaling



- 6 copies of data across 3 AZ:
    - 4/6 copies needed for writes, 3/6 needed for reads, self-heal with peer to peer replication
    - Storage striped across 100s of volumes
- One Aurora instance takes writes (master) with automated failover for master < 30 sec
    - Master + up to 15 read replicas in cross origin replication

Aurora DB Cluster



- DNS name called writer endpoint, which points at the master, which will automatically write to the master instance, even if master changes.

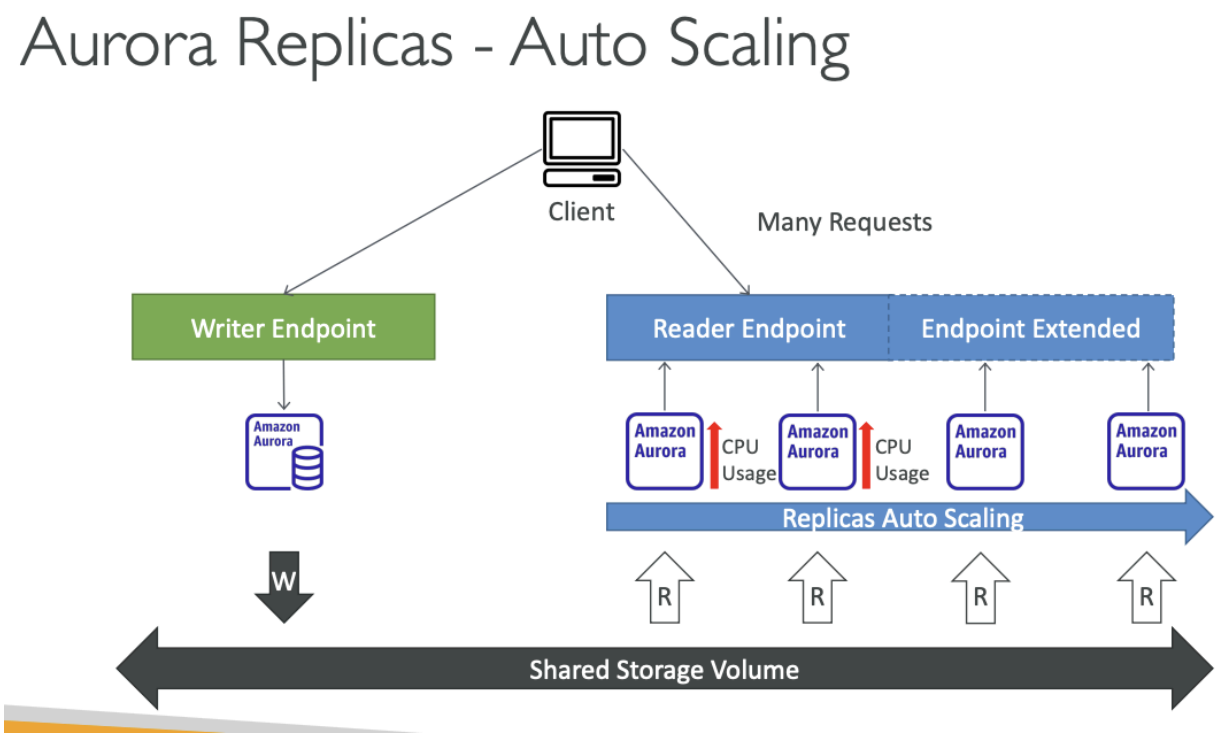
- Reader endpoint: connection load balancing and connects automatically to all the read replicas so anytime the client connects to reader endpoint, it will load balance across all read replicas
    - LB happens at connection level

Features of Aurora

- Auto failover
- Backup and recovery
- Isolation and security
- Industry compliance
- Push button scaling
- Advanced monitoring
- Automated patching with 0 downtime
- Routine maintenance
- Backtrack: restore data and any point in time without backups

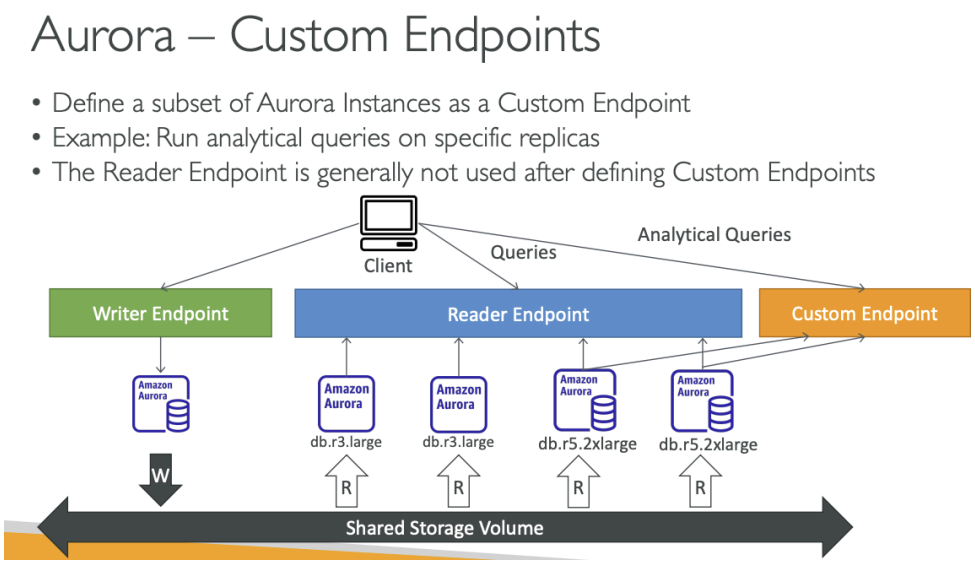
Aurora Advanced Concepts

Replica Auto Scaling



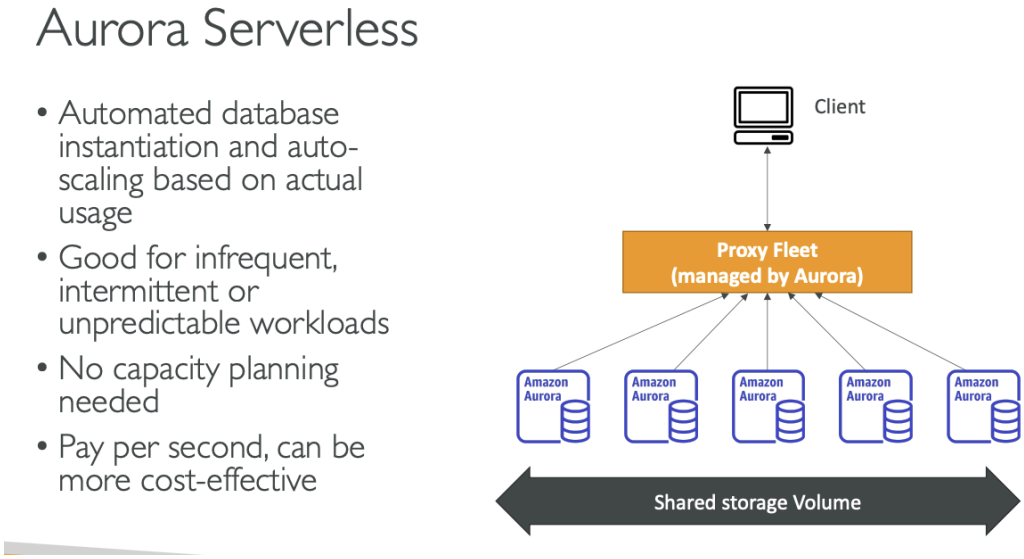
- Add replicas and have extended reader endpoints to cover the new replicas to distribute reads

Custom Endpoints



- With different DB sizes, a subset of reader instances are used as custom endpoints
- Reader endpoint generally not used after custom endpoints

Aurora Serverless



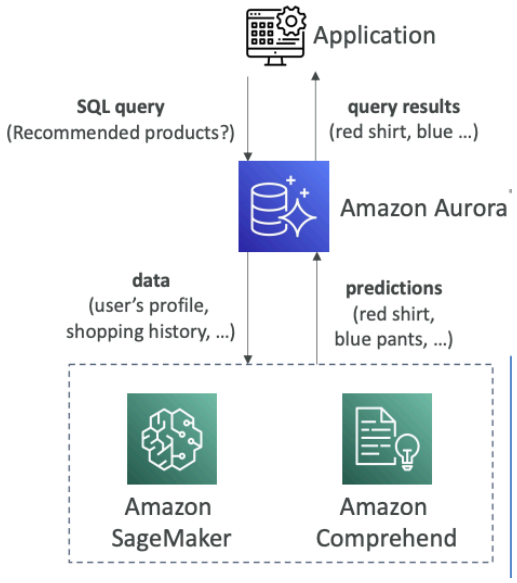
- Automated DB instantiation and auto scaling based on usage
  - Good for infrequent or intermittent / unpredictable workloads
  - No capacity planning, pay per second

Global Aurora

- Cross region read replicas
    - Disaster recovery, simple to put in place
- Global DB (recommended)
    - 1 primary region for read / write
    - Up to 5 secondary (read only) regions, replication lag is < 1 second
    - Up to 16 read replicas per secondary region
    - Helps for decreasing latency
    - Promoting another region < 1 min
- Typical cross region replication takes < 1 second

Aurora Machine Learning

- ML based predictions to apps via SQL
- Simple, optimized integration between Aurora and ML services
- Supports:
    - Sagemaker
    - Comprehend
- Use cases: fraud detection, sentiment analysis, etc…



RDS & Aurora – Backup and Monitoring

RDS Backups

- Automated backups
    - Daily full backup of DB (during backup window)
    - Transaction logs backed up by RDS every 5 min
        - Ability to restore to any point in time
    - 1 to 35 day retention, 0 to disable backup
- Manual DB snapshots
    - Retain backup as long as you want
- Trick: in stopped RDS DB, if you plan on stopping it for a long time, snapshot and delete original, then restore

Aurora Backups

- Automated backups

    - 1 to 35 day retention (cannot be disabled)
    - Point in time recovery in that timeframe
- Manual backups:
    - Manual trigger, retains as long as you want

RDS & Aurora Restore Options

- Restoring a backup or snapshot creates a new DB
- Restore MySQL RDS DB from S3
    - Create a backup from on premise DB
    - Store in S3
    - Restore backup file from S3 onto new RDS instance running MySQL
- Restoring MySQL Aurora cluster from S3
    - Create backup of on premise DB using Percona XtraBackup
    - Store backup file on S3
    - Restore backup file on new Aurora cluster running MySQL

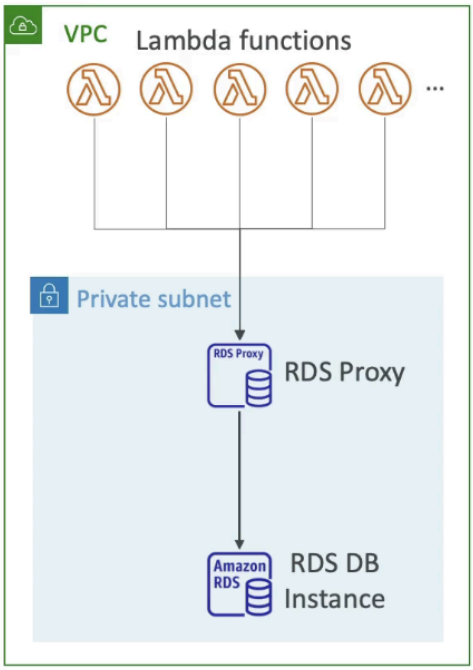
Aurora DB Cloning

- Create new Aurora DB cluster on existing one
- Faster than backup & restore
- Uses copy on write protocol
    - New DB cluster uses same data volume as original DB cluster (fast and efficient - no copying needed)
    - When updates are made to new DB cluster data, additional storage is allocated and data is copied to be separated
- Useful to create a “staging” DB from “prod” DB without impacting the production DB

RDS & Aurora Security

- At rest encryption:
    - DB master & replicas encrypted via KMS defined at launch time
    - If master is not encrypted, read replicas cannot be encrypted
        - To encrypt unencrypted DB, go through DB snapshot & restore as encrypted
- In flight encryption:
    - TLS ready by default, use AWS TLS root certificates client side
- IAM Auth: IAM roles to connect to DB (instead of username and password)
- SG: control network access to RDS / aurora DB
- No SSH available except on RDS custom
- Audit logs can be enabled and sent to CW logs for longer retention

Amazon RDS Proxy

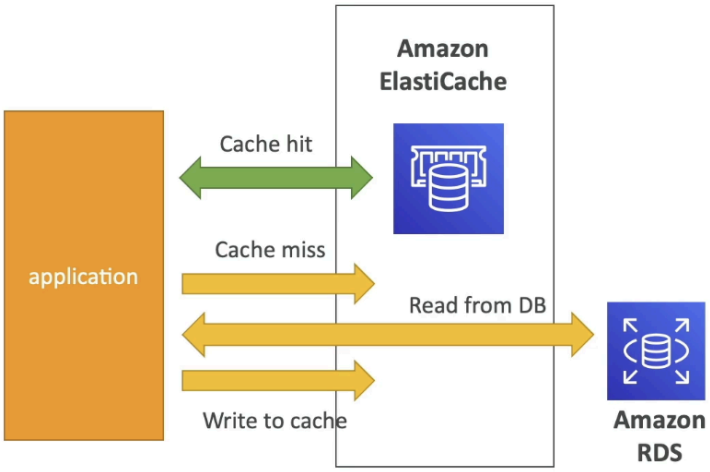


- Fully managed DB proxy for RDS and Aurora
  - RDS can already be reached directly, but proxy allows apps to pool and share DB connections with the DB. Basically, a proxy pools all connections and sends less connections to RDS DB instance at a time
    - Improves efficiency by reducing stress on resources and minimizes open connections
    - Never publicly accessible, must be accessed via VPC
  - Serverless, autoscaling, highly available (multi AZ)
  - Reduce RDS/Aurora failover time
  - Enforces IAM auth for DB and credentials stored via Secrets Manager
  - Never publicly accessible

ElastiCache Overview

ElastiCache
Solution Architecture - DB Cache

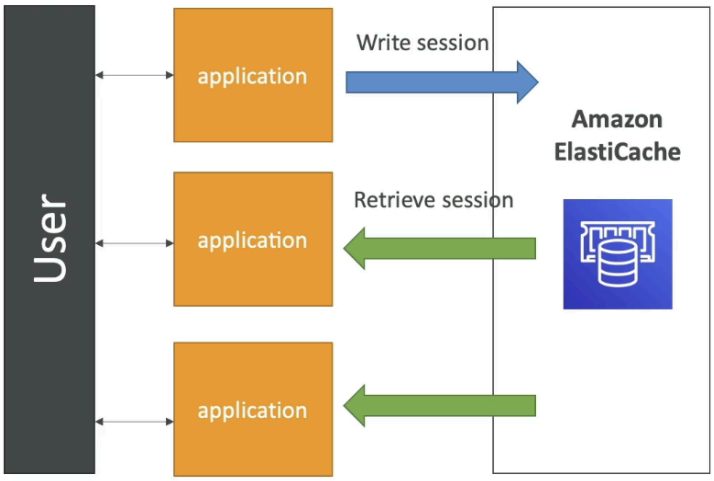
- Applications queries ElastiCache, if not available, get from RDS and store in ElastiCache.
- Helps relieve load in RDS
- Cache must have an invalidation strategy to make sure only the most current data is used in there.



- AWS managed Redis or mem-cached
- Caches are in-memory DB with high performance, low latency to help reduce load off DB for read intensive workloads and makes application stateless
- Needs cache invalidation method to make sure most current data is cached

ElastiCache
Solution Architecture – User Session Store

- User logs into any of the application
- The application writes the session data into ElastiCache
- The user hits another instance of our application
- The instance retrieves the data and the user is already logged in



ElastiCache - Redis vs Memcached

Redis

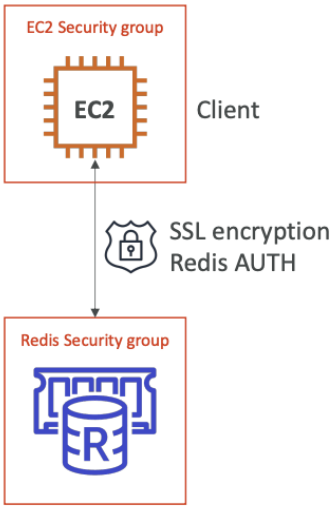
- Multi AZ with auto-failover
- Read replica to scale read and high availability
- Data durability using AOF persistence
- Backup and restore features
- Supports sets and sorted sets
- Max 5 read replicas in redis cluster with cluster mode disabled

Memcached

- Multi-node partitioning of data (sharding)
- No high availability (replication)
- Non-persistent
- No backup and restore
- Multithreaded architecture

Cache Security

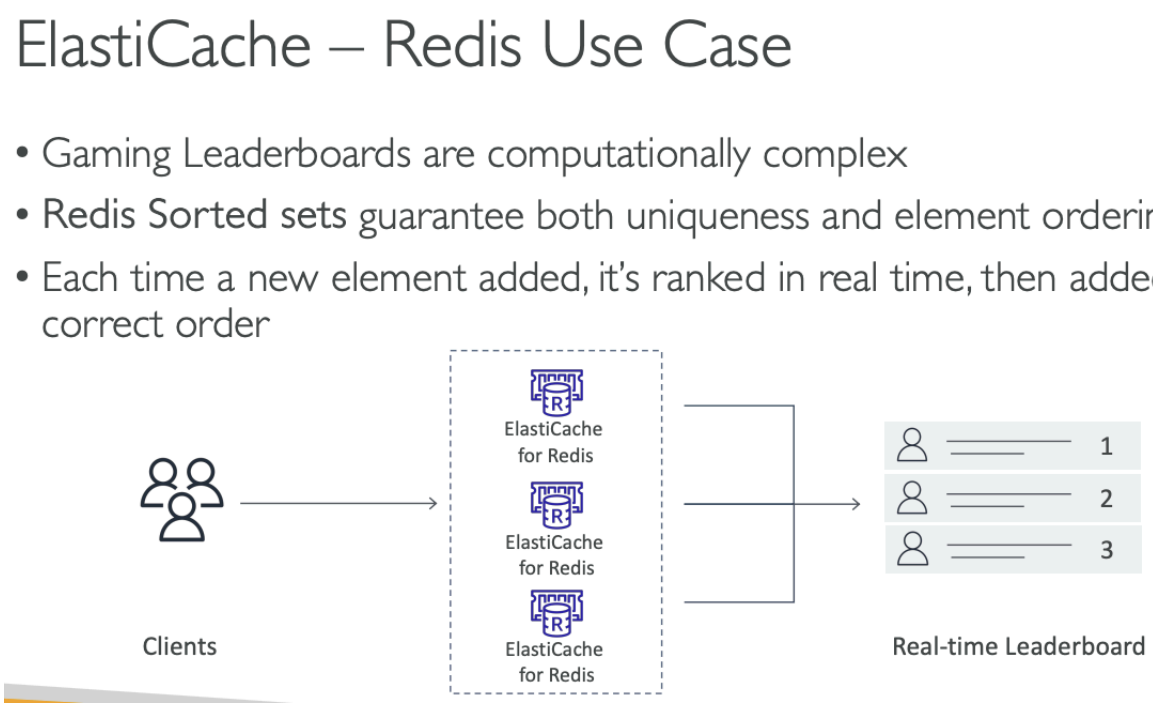
- Supports IAM Auth for Redis only
- IAM policies on ElastiCache only used for AWS API
- Redis AUTH
    - Password / token set when create Redis cluster
    - Extra level of security for cache (on top of SG)
    - Support SSL in flight encryption
- Memcached
    - Supports SASL based authentication (advanced)



Patterns for ElastiCache

- Lazy loading: all read data is cached
- Write Through: adds / updates when written to DB
- Session store: store temp session data in cache (via TTL)

Redis Use Case



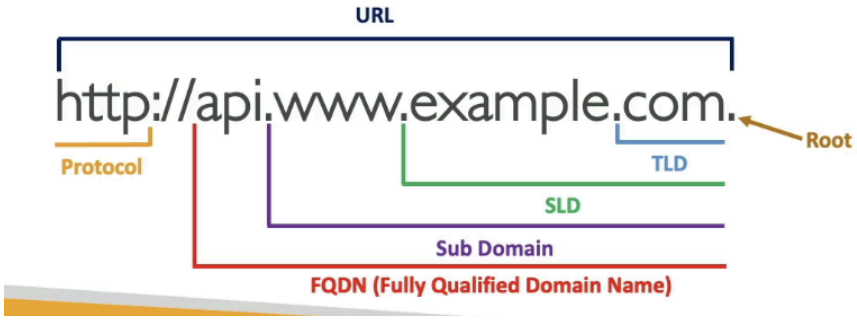
- Gaming leaderboards are complex
- Redis sorted sets guarantee both uniqueness and element ordering
- Each time new element is added, ranked in real time and added in correct order

RDS DB Ports

- PostgreSQL: 5432
- MySQL: 3306
- Oracle RDS: 1521
- MSSQL Server: 1433
- MariaDB: 3306
- Aurora: 5432 (if PostgreSQL compatible) or 3306 (MySQL compatible)

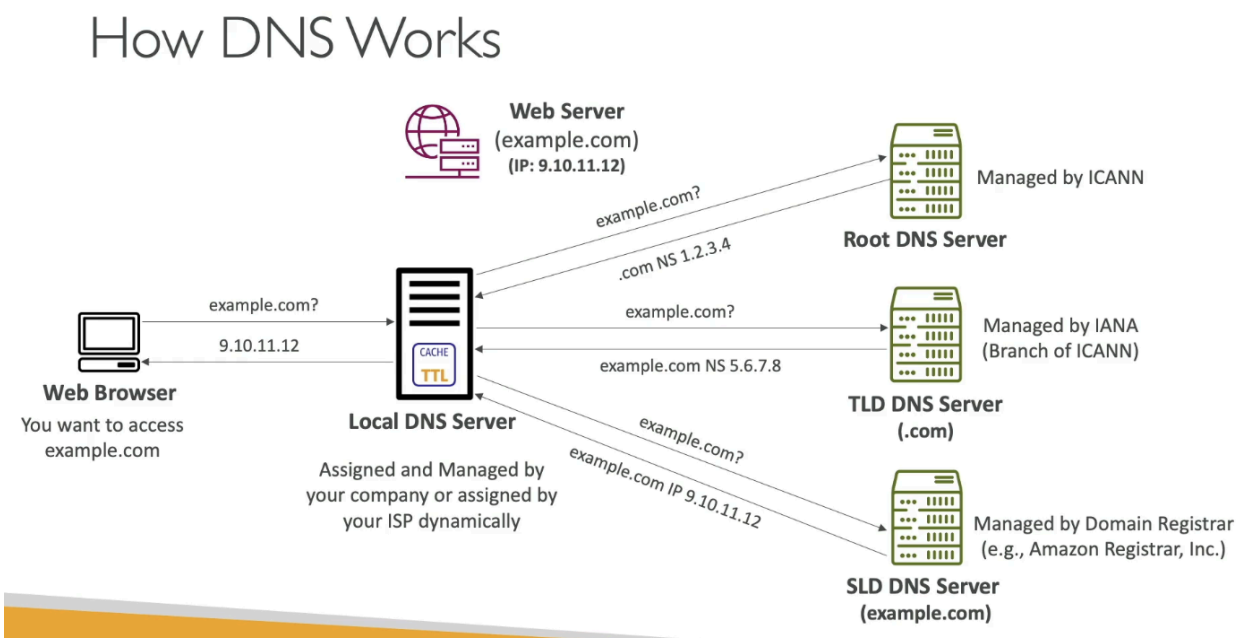
Section 10: Route 53

What is DNS?

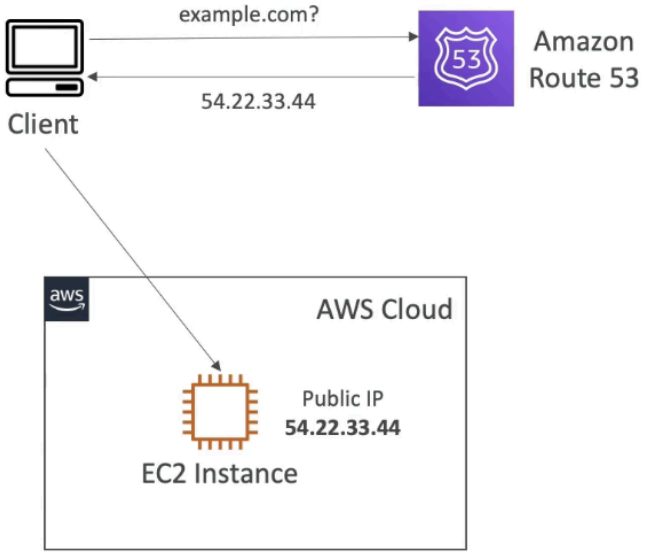


- Domain name system → translates human friendly hostnames into machine IP addresses (e.g. google.com → 123.456.78.90
    - Backbone of internet and has hierarchical naming structure: .com, x.com, www.x.com
- Domain registrar: register domain name
- DNS records: A, AAAA, CNAME, NS
- Zone file: contains DNS records → match hostname to IP
- Name server: resolves DNS queries (authoritative or non authoritative)
- Top level domain (TLD): .com, .org, etc..
- Second level domain (SLD): google.com

How DNS works



Route 53 Overview



- Highly available, scalable, fully managed and authoritative DNS
    - Authoritative = customer can update DNS records
- Also a domain registrar (can register domain)
- Ability to check health of resources
- Only service with 100% availability SLA

Route 53 Records

- How you want to route traffic for a domain
- Record contains:
    - Domain/subdomain name - example.com
    - Record type - A, AAAA
    - Value - 123.456.78
    - Routing policy - how Route 53 responds to queries
    - TTL - amount of time record cached at DNS resolvers
- Supports the following DNS record types:
    - Must know: A / AAAA / CNAME / NS

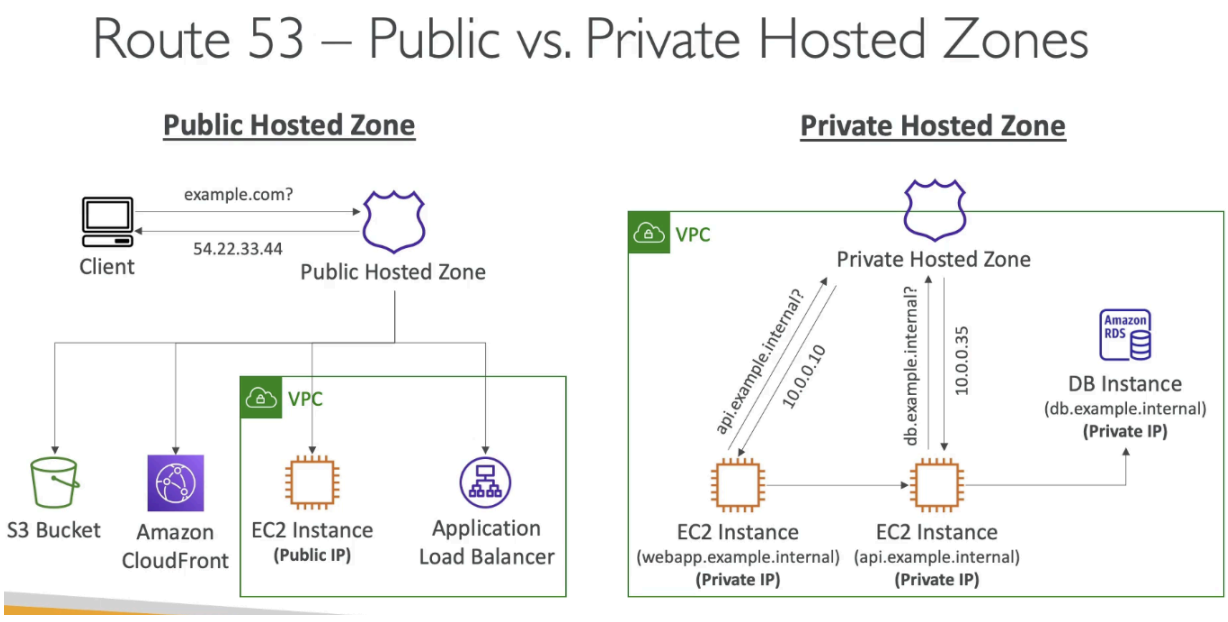
Record Types:

- A: maps a hostname to IPv4 (example.com → 123.456.78)
- AAAA: maps hostname to IPv6
- CNAME - map hostname to another hostname
    - Target is a domain name which must have A or AAAA record
    - Can't create a CNAME record for top node of DNS namespace (zone apex)
        - Ex: can't create for example.com, but can for www.example.com
- NS: name servers for hosted zone
    - This is DNS names or IP addresses of servers that can respond to DNS queries for hosted zone
    - Control how traffic is routed to domain

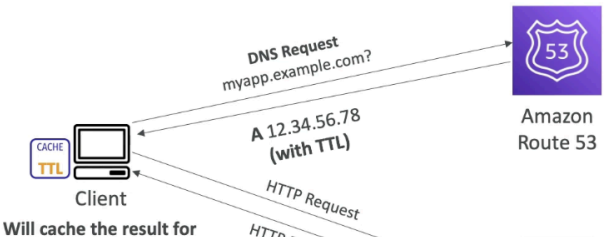
Hosted Zones

- A container for records that define how to route traffic to a domain and subdomains
- Public hosted zones: contains records that specify how to route traffic on internet (public domain names)
- Private hosted zones: records that specify how you route traffic within 1+ VPCs (private domain name)
- $0.50/month per hosted zone

Public vs Private Hosted Zones



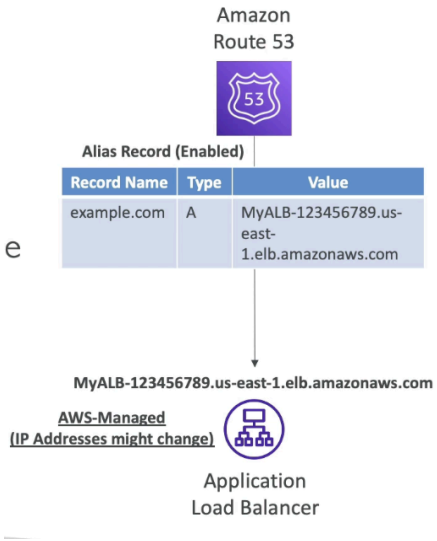
Records TTL (time to live)



- If the client requests the same host name again, the client will not query the DNS system if it is cached and still within TTL. Done because we don't want to query the DNS often
    - High TTL:
        - Less traffic on Route 53
        - Possibly outdated records
    - Low TTL:
        - More traffic on Route 53 ($$$)
        - Records are outdated for less time
        - Easy to change records
- Except for Alias records, TTL is mandatory for each DNS record

CNAME vs Alias

- AWS resources exposes AWS hostname (map hostname to domain name)
- CNAME:
    - Points a hostname to any other hostname (example.com → example1.com)
    - Only works for non-root domain (aka something.domain.com, not domain.com)
- Alias:
    - Points a hostname to AWS resource
    - Works for root and non root domains
    - Free of charge and have native health check
    - Extension to DNS functionality
    - Automatically recognizes changes in resource's IP address
    - Unlike CNAME, it can be used for top node of DNS namespace (zone apex) e.g: example.com
    - Always of type A / AAAA
    - Can't set TTL (done automatically)
    - Cannot set ALIAS record for EC2 DNS name



Route 53 – Alias Records Targets

- Elastic Load Balancers
- CloudFront Distributions
- API Gateway
- Elastic Beanstalk environments
- S3 Websites
- VPC Interface Endpoints
- Global Accelerator accelerator
- Route 53 record in the same hosted zone

- You cannot set an ALIAS record for an EC2 DNS name

Routing Policies

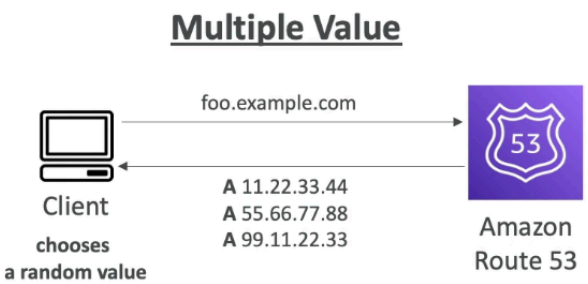
- Define how Route 53 responds to DNS queries
  - DNS does not route any traffic, only responds to DNS queries (translate hostnames to IP)

Simple

- Typically route traffic to a single resource
- Can specify multiple values in same record
- If multiple values are returned, a random one is chosen by the client
- When Alias enabled, specify only 1 AWS resource
- Can't associate with health checks

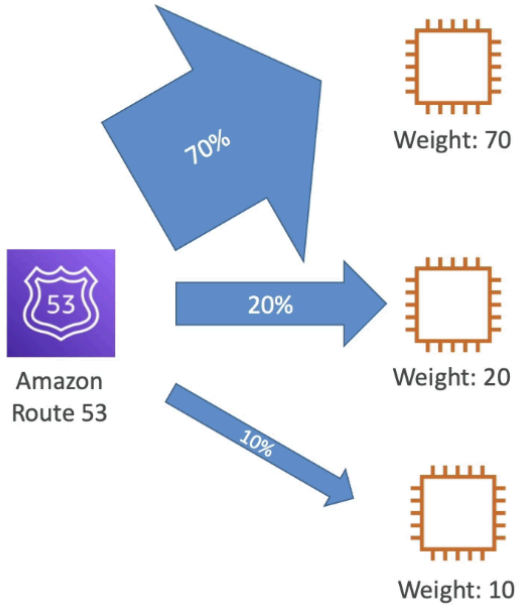
Weighted

- Control % of requests that go to each specific resource
- Assign each record a relative weight
  - Traffic % = weight for specific record / sum of all weights for all records
  - No need to add up to 100
  - Weight = 0, no traffic

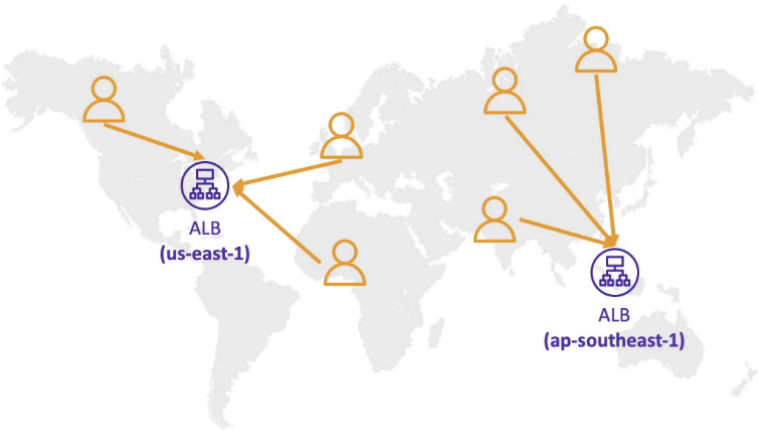


    - If all records have 0, all records are returned equally
- DNS records must have the same name and type
- Can be associated with health checks
- Use case: LB between regions, testing application versions…

Latency-based



- Redirect to the resource that has the least latency (helpful when latency is priority)
    - Latency based on traffic between users and AWS regions
- Can be associated with health checks (has failover capability)



Health Checks

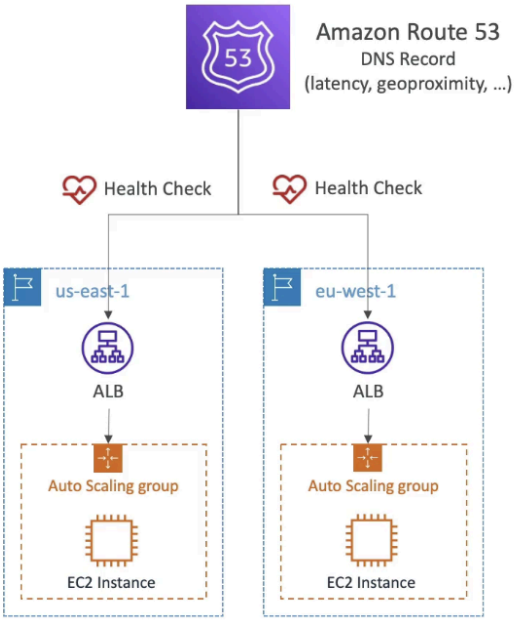
- Only for public resources
- Automated DNS failover
    - Monitor endpoint, other health checks, CloudWatch alarms
- 15 health checkers from all regions
    - Threshold for healthy/unhealthy
    - Interval for check
    - Supports HTTP/S, TCP
    - If > 18% is healthy, it is healthy; otherwise unhealthy
        - Passes when 2xx or 3xx status code is returned
        - Can be setup to pass/fail based on the text in first 5120 bytes of the response
    - Ability to choose which locations Route 53 uses
    - Configure your router/firewall to allow incoming requests from Route 53 health checkers

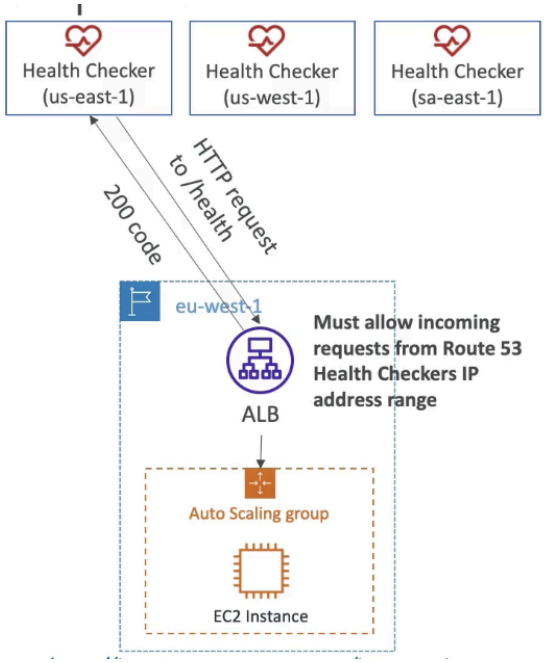
Calculated Health checks

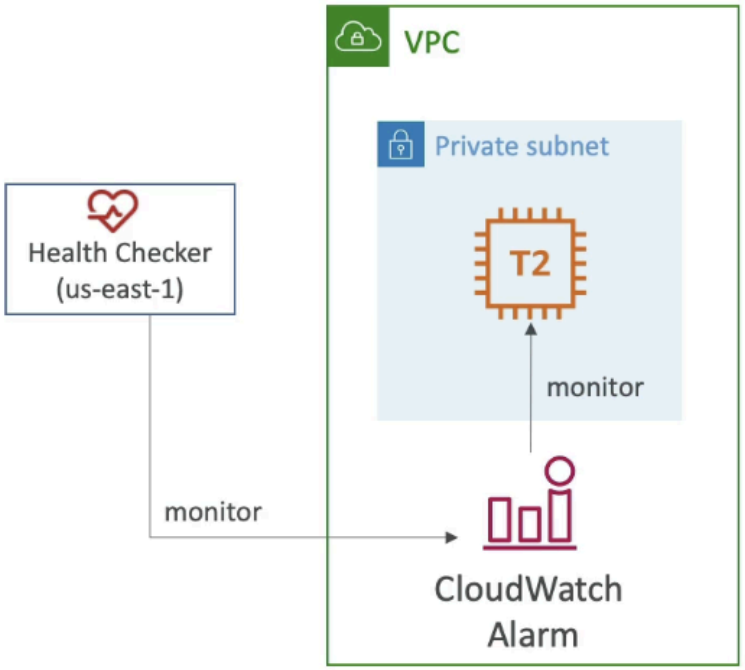
- Combine the results of multiple health checks into a single health checks
- Can use OR, AND, NOT
- Monitor up to 256 child health checks
- Specify how many health checks need to pass to make the parent pass
- Usage: perform maintenance without causing all checks to fail

Private Hosted Zone Health Checks

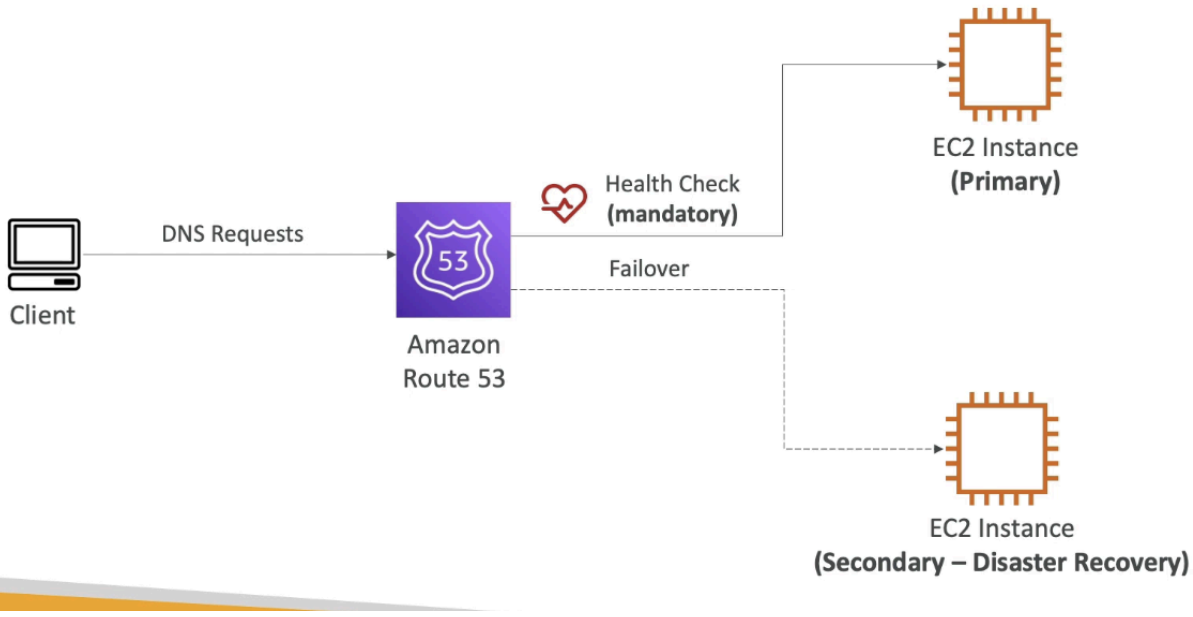
- Route 53 health checks are outside VPC, cannot access private endpoints
- Create a CloudWatch metric and associate an alarm, then create a health check that checks the alarm itself







Failover (Active - Passive)

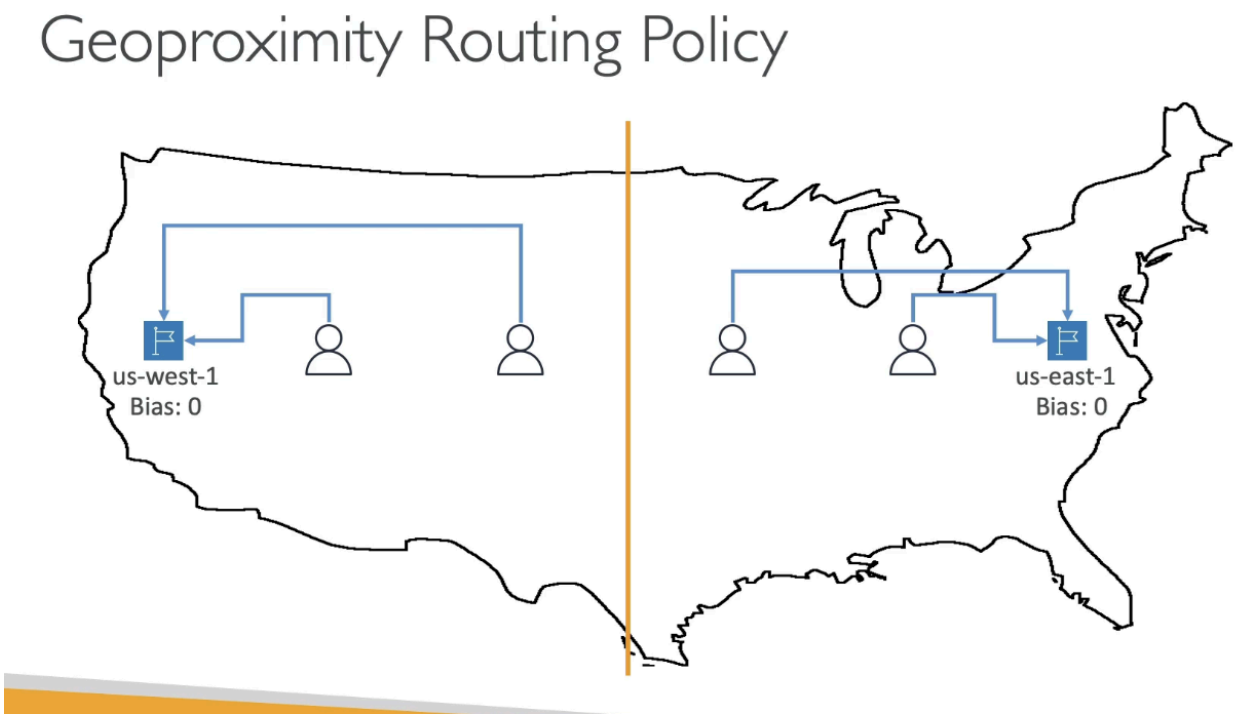


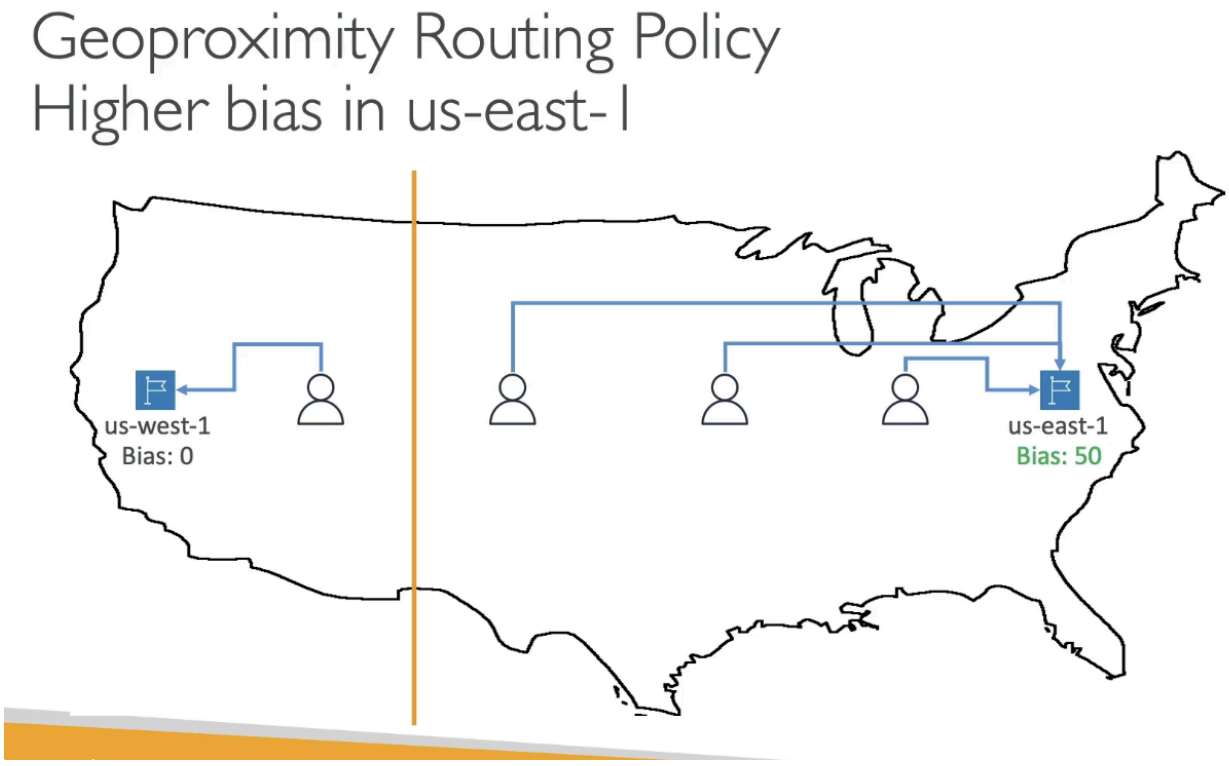
Geolocation

- Based on user location (continent, country, state), default record if no match
    - Restricts content distribution
    - Associated with health checks

Geo Proximity Routing

- Route traffic based on geographic location of users and resources
- Ability to shift more traffic to resources based on defined bias
- To change the size of geographic location, specify bias value:
    - To expand (1 to 99) - more traffic to resource
    - To shrink (-1 to -99) - less traffic to resource
- Resources can be:
    - AWS resources (specify AWS region)
    - Non AWS resources (specify latitude and longitude)
- You must use Route 53 Traffic flow to leverage bias



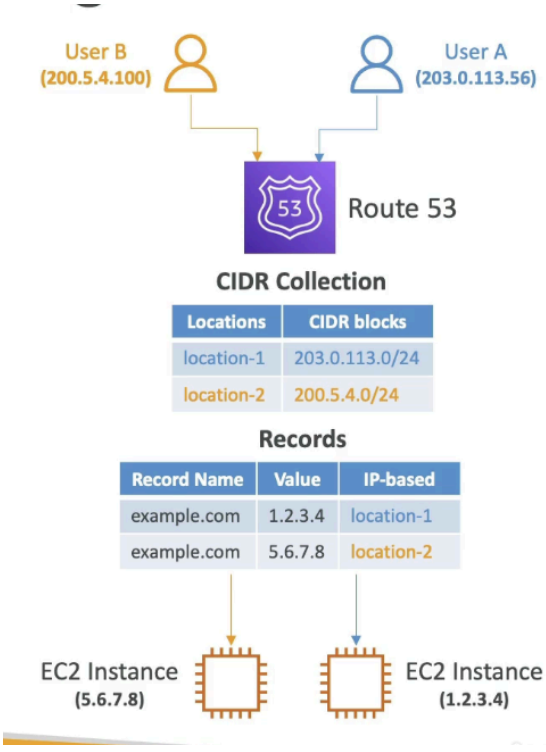


Traffic Flow

- Simplify the process of creating and maintaining records in large and complex configurations
    - Visual editor to manage complex routing decision trees
    - Configurations saved as traffic flow policy and can be applied to different Route 53 hosted zones (different domain names)

IP Based

- Routing based on clients' IP addresses
- Provide list of CIDRs and corresponding endpoints/locations (user IP to endpoint mapping)
- Use cases: optimize performance, reduce network costs



Multi Value Routing

- Use when routing traffic to multiple resources
- Route 53 return multiple values / resources
    - Can be associated with health checks (return only values for healthy records)
    - Up to 8 healthy records are returned for each multi-value query
- Not a substitute for having ELB
- Different than simple routing with multiple records because simple doesn't have health checks

3rd Party Domains & Route 53

Domain Registrar vs DNS Service

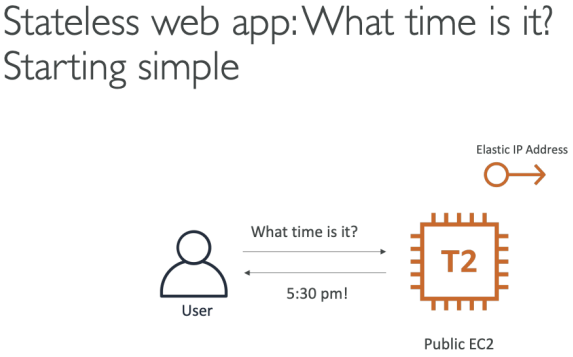
- You can buy / register domain name with Domain Registrar
- Domain registrar usually provides a DNS service to manage DNS records, but can use another DNS service to manage DNS records
- Domain registrar != DNS service, but every domain registrar usually comes with some DNS features

1. Create Hosted Zone in Route 53
2. Update NS records on 3rd party to use Route 53 Name Servers

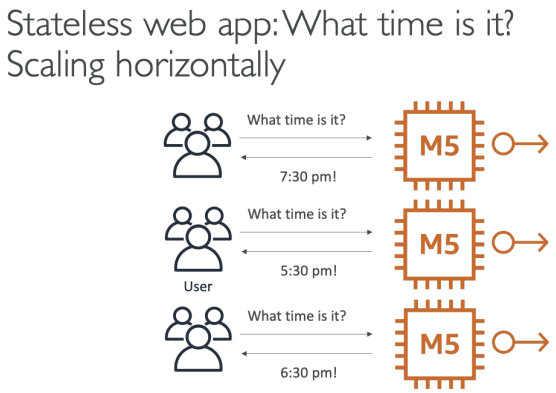
Section 11: Classic Solutions Architecture Discussions

Stateless web app: WhatsTheTime.com

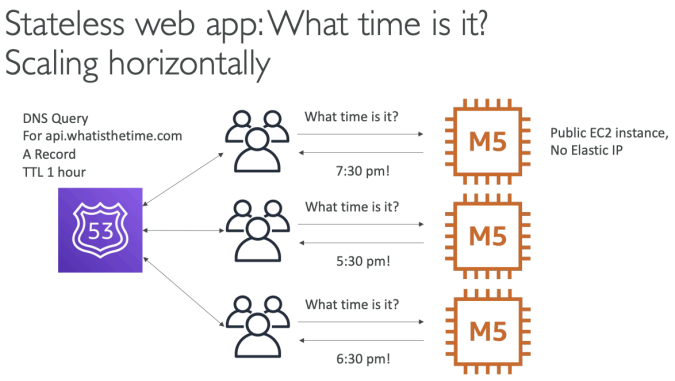
- Starting small with a T2 micro EC2 instance will host website and has an elastic IP to have static IP if any restarts need to happen

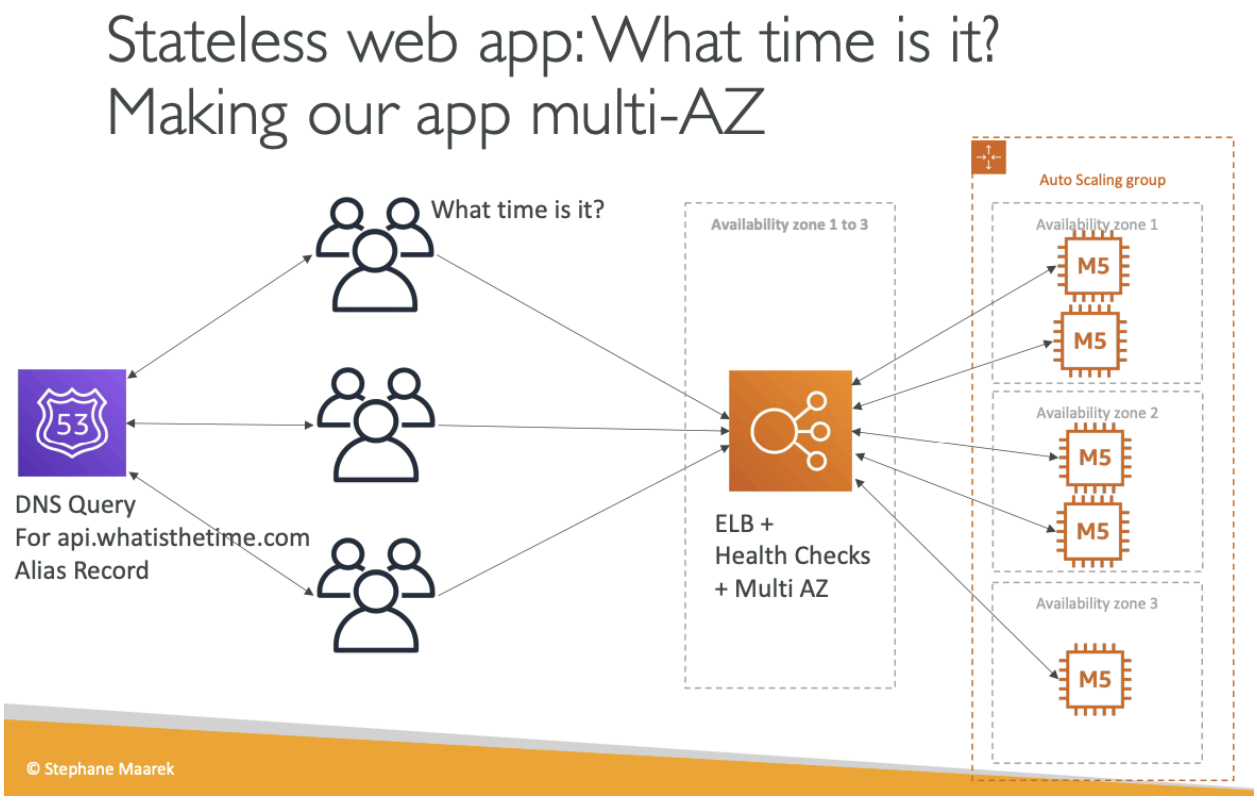


- More uses, scale to M5 instance, but there is downtime while M5 instance is being deployed. Add more M5 instances for more users with more elastic IPs



- Users leverage Route 53 with A record of TTL 1 hour for no elastic IPs. Route 53 will keep instances in sync. However if an instance is removed, TTL might be saved to a removed instance.

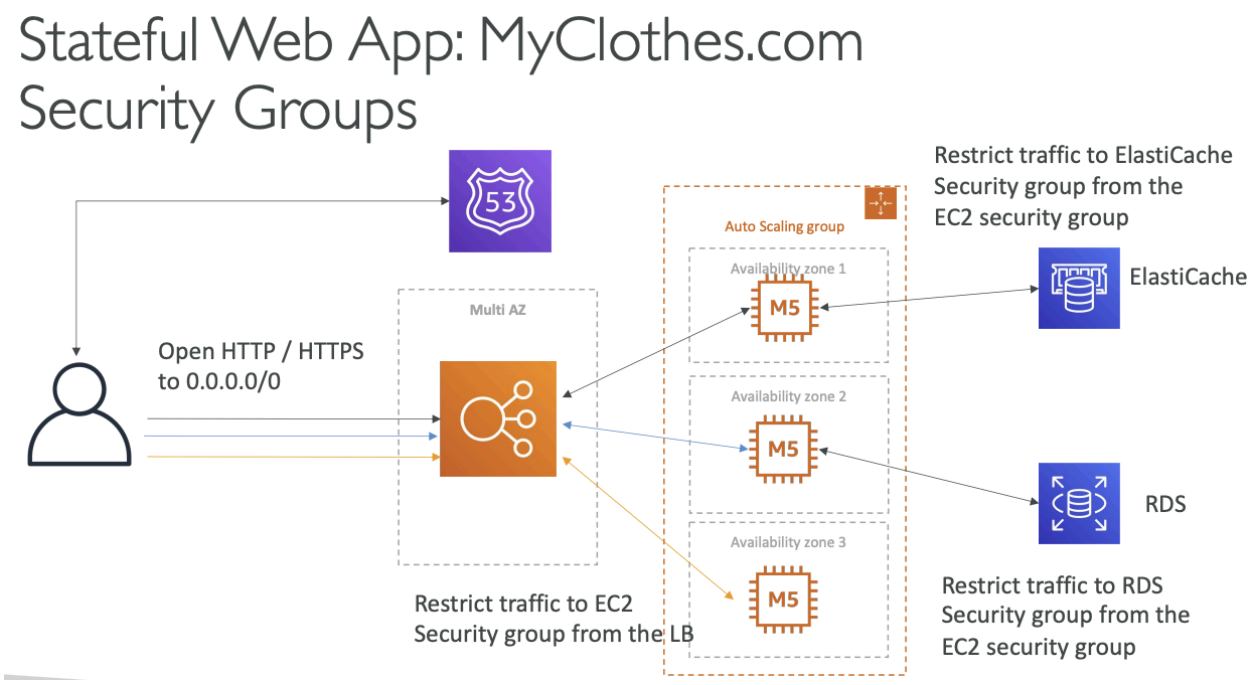




- Auto scaling instances in multiple AZs for disaster recovery. ELB as public facing with SG leading to instances. Alias record to track ELB for users. Additional cost savings can be made by reserving at least 1 instance per AZ.

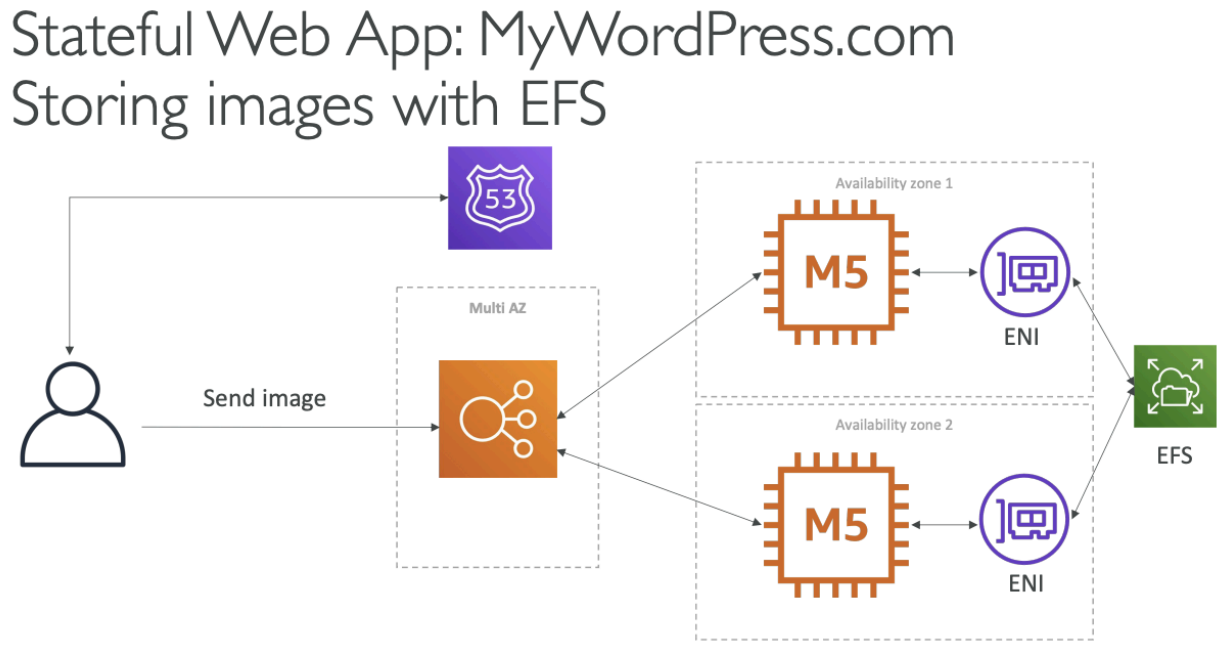
Stateful Web App

- The ending architecture from the other example poses a risk where if the user moves to another instance, session state is lost. Using stickiness fixes the issue, but if instance terminated, session is lost.
- Using user cookies solves the issue, but HTTP requests are heavier and must be validated if cookies are altered.



- Using a session ID in web cookies to store and retrieve data from the cache to save session state. RDS can be used with read replicas to have disaster recovery. Add SG to restrict traffic to EC2 instances and ElastiCache + RDS
    - 3 tier architecture

Stateful Web App



- Building off the previous example, you can exchange RDS for Aurora for a serverless architecture. To store images, EBS can be used as a block store. However, with increased instances you need additional EBS volumes on each AZ. EFS can be used to solve this issue by creasing ENI's in each AZ for AZ's to access single EFS.

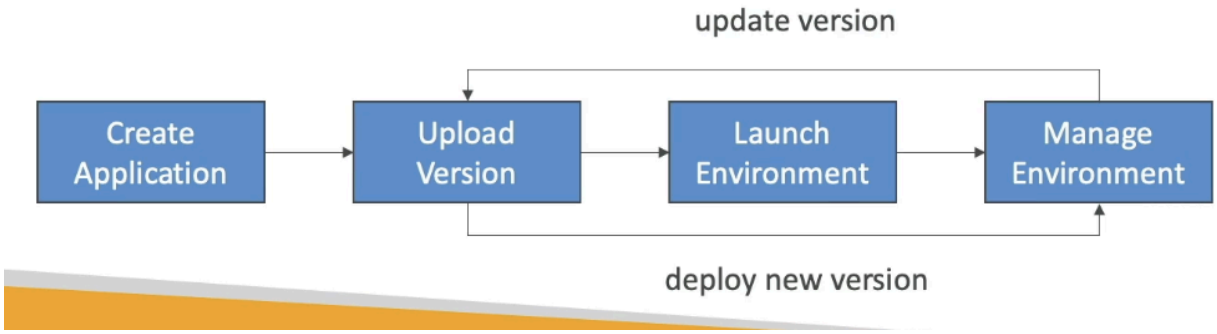
Instantiating Applications Quickly

- EC2 Instances
    - Golden AMI: install applications, OS dependencies… beforehand and launch EC2 instance from golden AMI
    - Bootstrap via user data: for dynamic configuration
    - Hybrid: mix of golden AMI and user data (Elastic Beanstalk)
- RDS
    - Restore from snapshot: DB will have schemas and data ready
- EBS Volumes
    - Restore from snapshot, disk will be read and formatted

Beanstalk Overview

- Developer centric view of deploying apps on AWS
    - Managed service
        - Automatically handles capacity provisioning, LB, scaling, application health monitoring, instance configuration
        - Developer in charge of application code
    - Full control over configuration, paid for underlying services

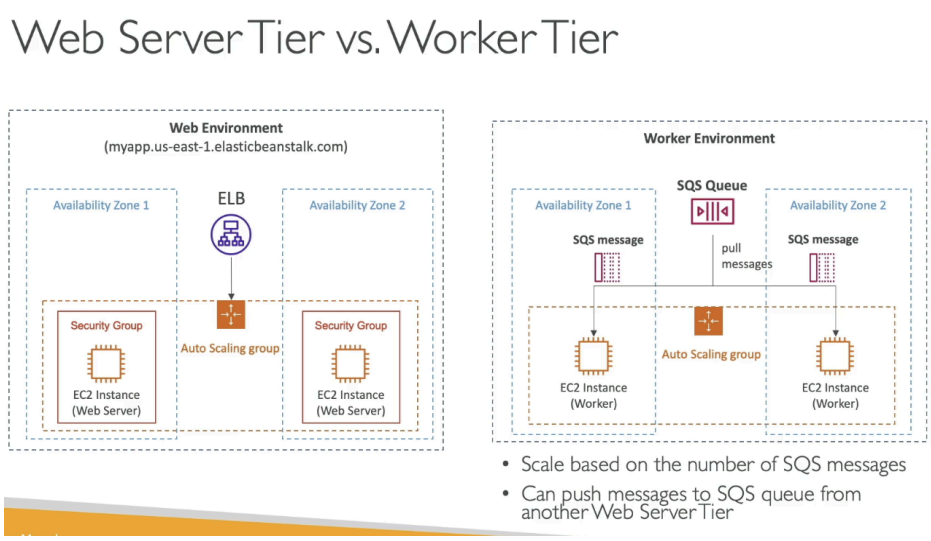
Beanstalk Components



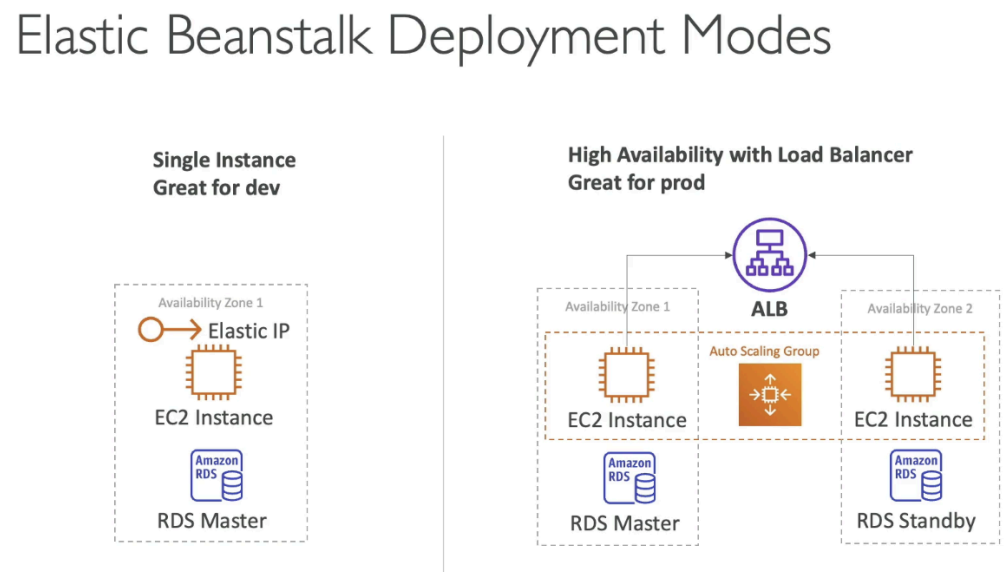
- Application: collection of Elastic Beanstalk components (environments, versions, configurations)
- Application version, environment (tiers)
    - Can create multiple environments and has worker vs web server tier

Supports many platforms

Web Server Tier vs Worker Tier



Deployment Modes



- Single Instance (for dev) or high availability with LB (prod)

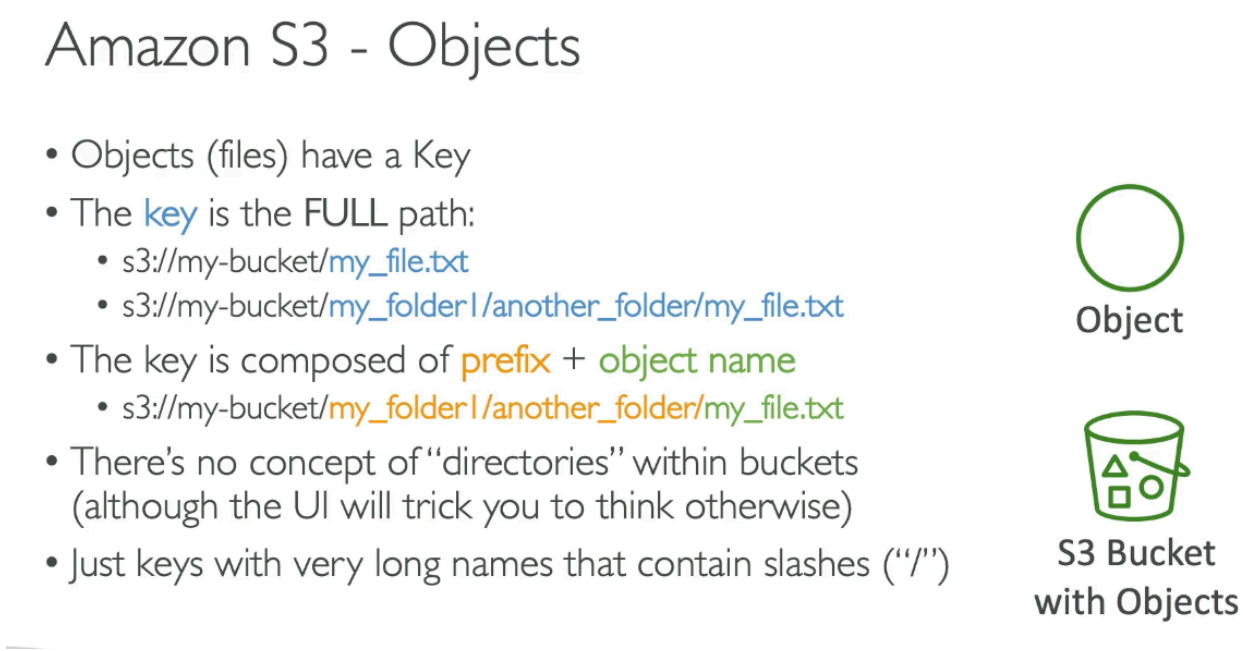
Section 12: S3 Intro

S3 Overview

- Backup and storage, disaster recovery, archive, hybrid cloud storage, application hosting

- Buckets (directories) allow storage of objects (files) and must be globally unique name across all regions
    - However, buckets are regional level created
- Naming convention: no uppercase, no underscore, 3-63 char long, not IP, not start with xn–, and not end in -s3alias

S3 Objects



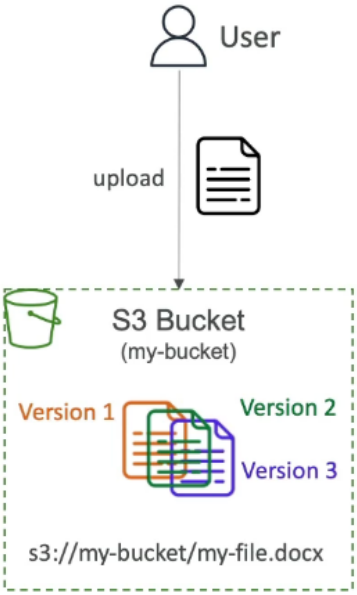
- Objects (files) have a key where key is the full path
    - Key is composed of prefix + object name
- There is no concept of directories within buckets, only keys that are just long names that contain slashes
- Object values are the content of the body:
    - Max size: 5TB, but if uploading more than 5GB, must use multi-part upload
- Metadata: list of text key / pair values - system or user metadata
- Tags: unicode key / value pair - up to 10 - useful for security / lifecycle
- Version ID (if enabled)

S3 Security: Bucket Policy

- User based
    - IAM Policies: which API calls allowed for specific user from IAM
- Resource Based
    - Bucket policies - bucket wide rules that allow cross account access
    - Object Access Control List (ACL) - finer grain (can be disabled)

    - Bucket Access Control List (ACL) - less common (can be disabled)
- Note: IAM principal can access S3 object if
    - User IAM permissions allow OR resource policy allows
    - AND no explicit deny
- Encryption: using encryption keys
- JSON based policies
    - Resource block: buckets and objects
    - Effect: allow/deny
    - Actions: set of API to allow/deny
    - Principal: account or user to apply policy to
- Use S3 bucket for policy to:
    - Grant public access to bucket
    - Force objects to be encrypted at upload
    - Grant access to another account (cross account)

S3 Static Website Hosting

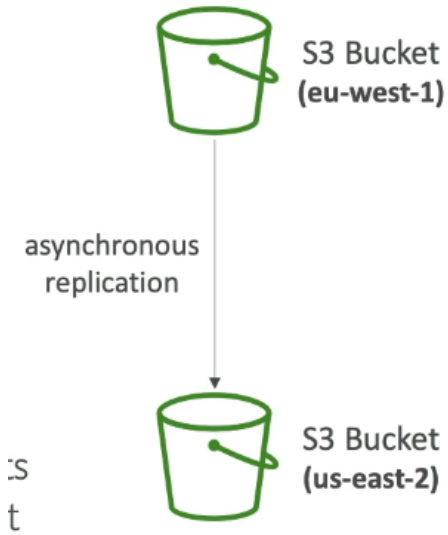


- S3 Can host static websites and accessible on internet
    - 403 forbidden error, make sure bucket policy allows public reads
- Website URL varies based on region, but is from the bucket website endpoint

S3 Versioning

- Enabled at bucket level
- Same key overwrite will change the version to: 1, 2, 3…
    - Protect against unintended deletes (ability to restore version)
    - Easy roll back
- Notes:
    - Any file not versioned prior to enabling versioning will have version “null”
    - Suspending versioning does not delete previous versions
    - Delete marker is on deleted items

S3 Replication (CRR & SRR)



- Versioning must be enabled on source and destination buckets
- CRR = Cross region replication
- SRR = same region replication
- Buckets can be in different AWS accounts
- Copying is asynchronous
- Must give proper IAM permissions to S3

- Use cases:
    - CRR: compliance, low latency access, replication across accounts
    - SRR: log aggregation, live replication between prod and test
- After replication is enabled, only new objects are replicated
- Optionally you can replicate existing objects using S3 Batch Replication
    - Replicates existing objects and objects that failed replication
- For delete operations:
    - Can replicate delete markers from source to target (optional) → delete marker replication
    - Deletions with version ID are not replicated (to avoid malicious deletes)
- No “chaining” of replication
    - If bucket 1 has replication into bucket 2, which has replication into bucket 3
    - Then objects created in bucket 1 are not replicated to bucket 3

S3 Storage Classes

S3 Storage Classes

- Amazon S3 Standard - General Purpose
- Amazon S3 Standard-Infrequent Access (IA)
- Amazon S3 One Zone-Infrequent Access
- Amazon S3 Glacier Instant Retrieval
- Amazon S3 Glacier Flexible Retrieval
- Amazon S3 Glacier Deep Archive
- Amazon S3 Intelligent Tiering

- Can move between classes manually or using S3 Lifecycle configurations

S3 Durability and Availability

- Durability:
    - High durability (99.999999999%, 11 9's) of objects across multiple AZ
    - If you store 10,000,000 objects with Amazon S3, you can on average expect to incur a loss of a single object once every 10,000 years
    - Same for all storage classes

- Availability:
    - Measures how readily available a service is
    - Varies depending on storage class
    - Example: S3 standard has 99.99% availability = not available 53 minutes a year

General Purpose

- Frequently accessed data, low latency and high throughput
- Sustain 2 concurrent facility failures

Infrequent Access

- Amazon S3 Standard-Infrequent Access (S3 Standard-IA)
  - 99.9% Availability
  - Use cases: Disaster Recovery, backups

- Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA)
  - High durability (99.999999999%) in a single AZ; data lost when AZ is destroyed
  - 99.5% Availability
  - Use Cases: Storing secondary backup copies of on-premise data, or data you can recreate

- Less frequently accessed data, but requires rapid access when needed
- Lower cost than S3 Standard, but cost on retrieval

Glacier Storage Classes

Amazon S3 Glacier Storage Classes

- Low-cost object storage meant for archiving / backup
- Pricing: price for storage + object retrieval cost

- Amazon S3 Glacier Instant Retrieval
  - Millisecond retrieval, great for data accessed once a quarter
  - Minimum storage duration of 90 days

- Amazon S3 Glacier Flexible Retrieval (formerly Amazon S3 Glacier):
  - Expedited (1 to 5 minutes), Standard (3 to 5 hours), Bulk (5 to 12 hours) – free
  - Minimum storage duration of 90 days

- Amazon S3 Glacier Deep Archive – for long term storage:
  - Standard (12 hours), Bulk (48 hours)
  - Minimum storage duration of 180 days

- Low cost object storage for archiving / backup
- Pricing: price for storage + object retrieval cost

S3 Intelligent Tiering

S3 Intelligent-Tiering

- Small monthly monitoring and auto-tiering fee
- Moves objects automatically between Access Tiers based on usage
- There are no retrieval charges in S3 Intelligent-Tiering

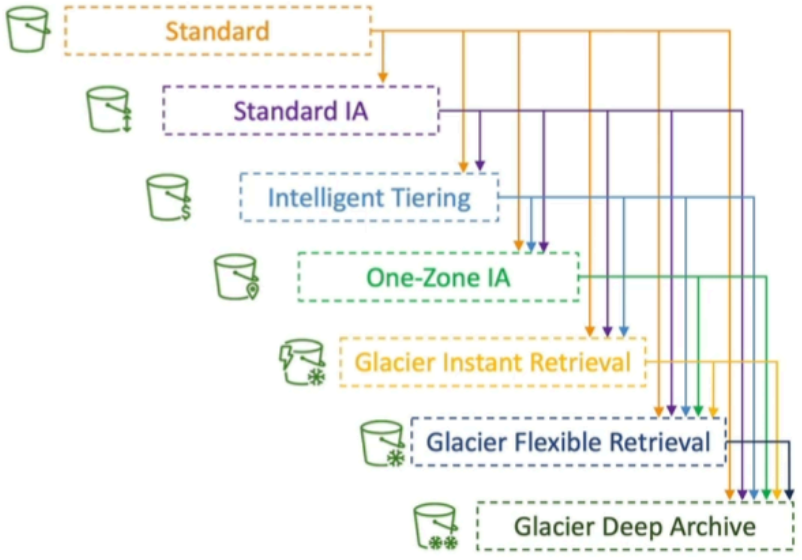
- *Frequent Access tier (automatic):* default tier
- *Infrequent Access tier (automatic):* objects not accessed for 30 days
- *Archive Instant Access tier (automatic):* objects not accessed for 90 days
- *Archive Access tier (optional):* configurable from 90 days to 700+ days
- *Deep Archive Access tier (optional):* config. from 180 days to 700+ days

- Small monthly monitoring and auto-tiering fee
- Moves objects automatically between Access Tiers based on usage
- No retrieval charges

Section 13: Advanced S3

S3 Lifecycle Rules (with S3 analytics)

- Transition objects between storage classes
- For infrequently accessed objects, move to Standard IA. For archive objects that you don't need fast access to, move to Glacier or Glacier Deep archive
- Moving objects automated with lifecycle rules
- Transition Action: configure object to transition to another storage class
- Expiration actions: configure objects to expire (delete) after some time
    - Can delete old versions of files (for versioning)



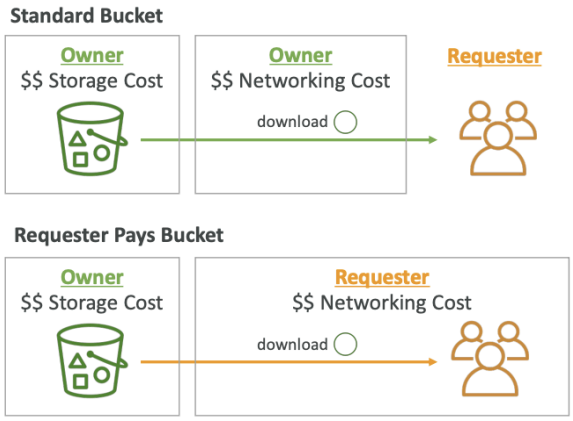
- Rules can be specified for certain prefix
- Rules can be created for certain object tags

S3 Storage Class Analytics

- Helps decide when to transition objects to the right storage class
- Recommendations for Standard and Standard IA
    - Does not work for one zone IA or Glacier
- Report updated daily
    - 24 to 48 hours to start seeing data analytics

S3 Requester Pays

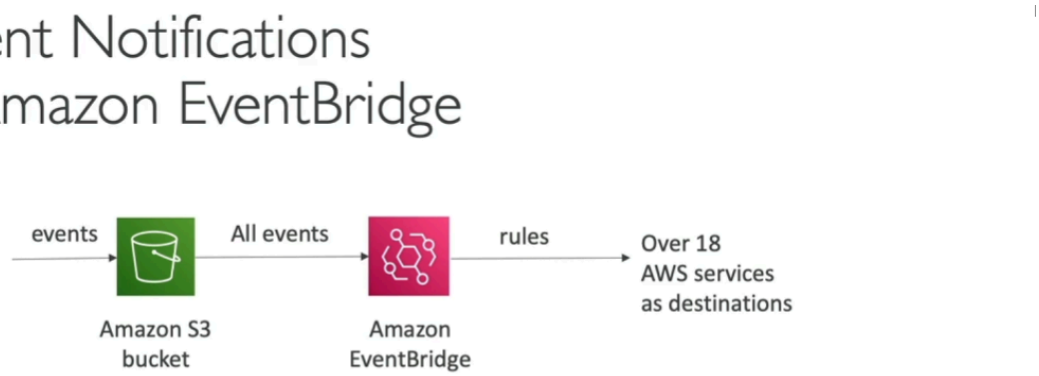
- In general, bucket owners pay for all S3 storage and data transfer costs with their bucket
- With requester pays buckets, requester pays the costs of the request and data download from bucket
    - Helpful when you want to share large datasets with other accounts
    - Requester must be authenticated with AWS



S3 Event Notifications

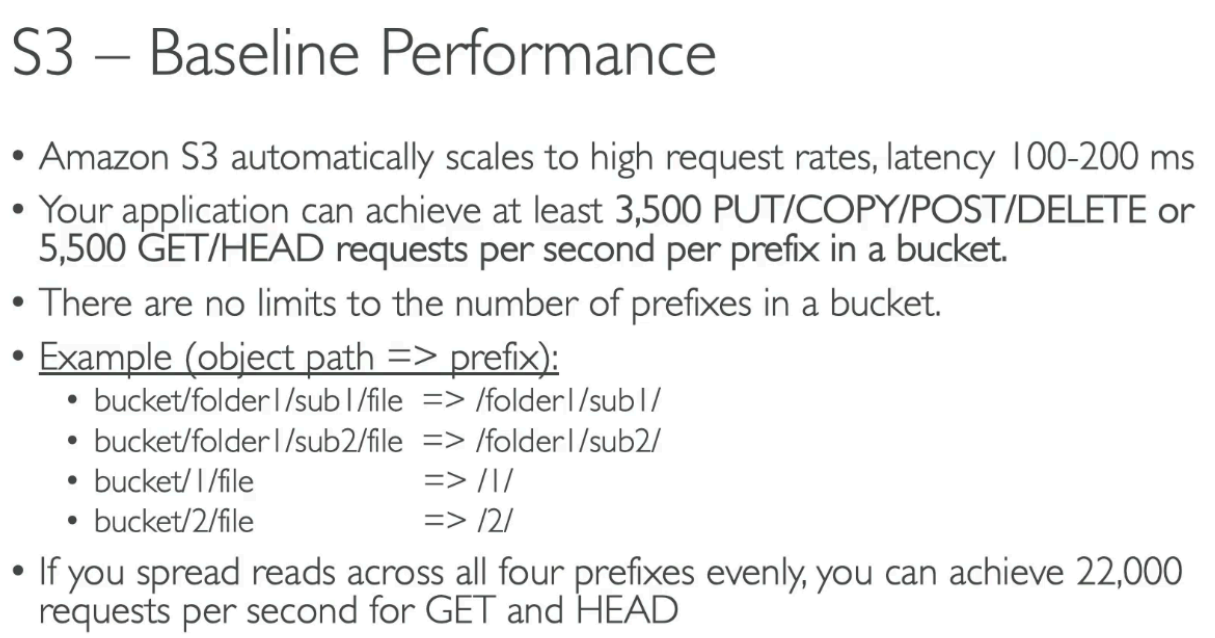
- React any time an action occurs in S3 bucket
    - S3:ObjectCreated, S3:ObjectRemoved, S3:ObjectRestore, S3:Replication
    - Object name filtering possible
- Must set IAM permissions
    - Other services must have a resource policy set to retrieve from bucket
- SNS, SQS, and Lambda function are Event Notification targets

S3 Event Notifications with Amazon EventBridge



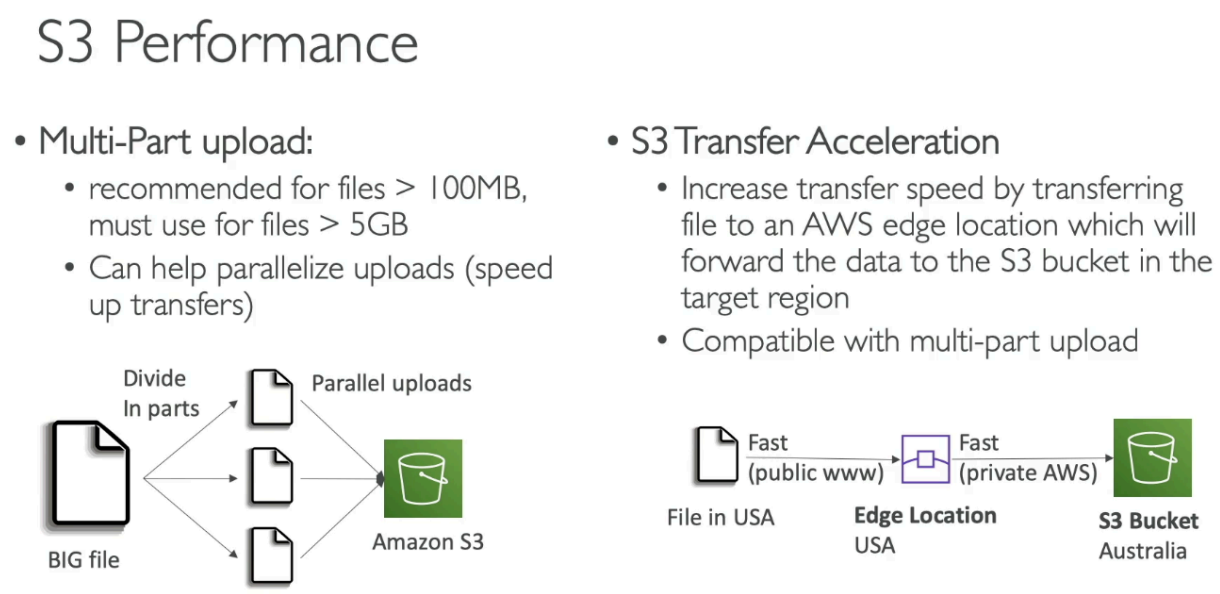
- Enhanced version with advanced filtering options, multiple destinations, and EventBridge Capabilities such as Archive, Replay events, Reliable Delivery
-

S3 Performance



- Automatically scales to high request rates with low latency

- 3500 PUT/COPY/POST/DELETE or 5500 GET/HEAD requests per second per prefix in a bucket
- No limit to number of prefixes in a bucket



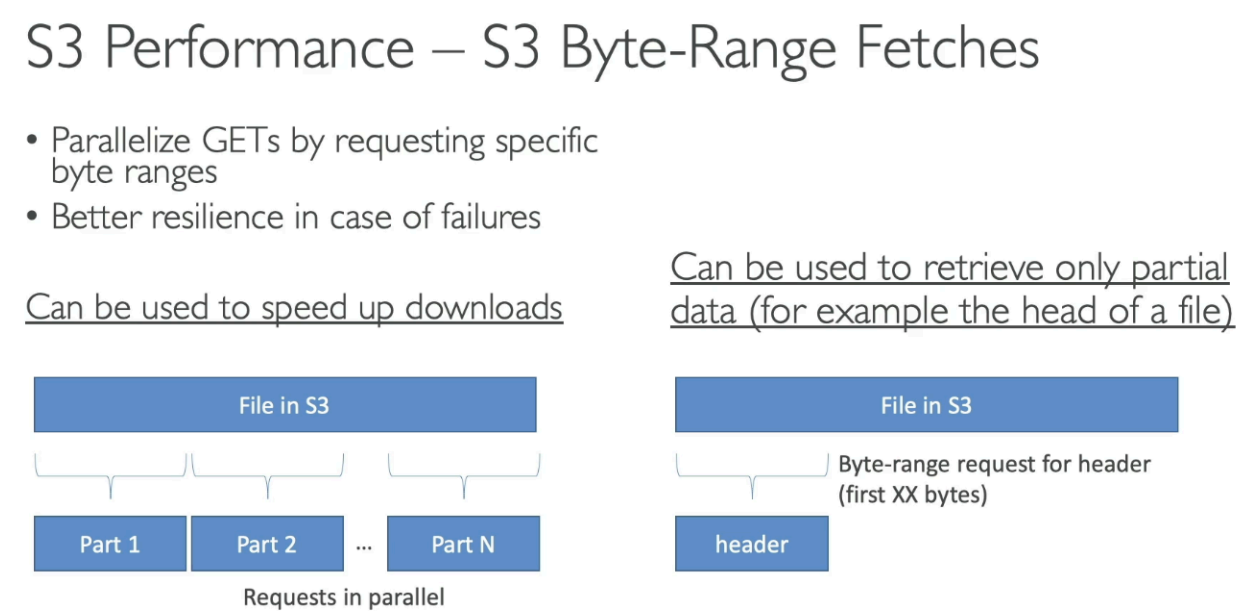
Multi-Part Upload

- Recommended for files > 100MB, must use for > 5GB
- Can help parallelize uploads (speed up transfers)

S3 Transfer Acceleration

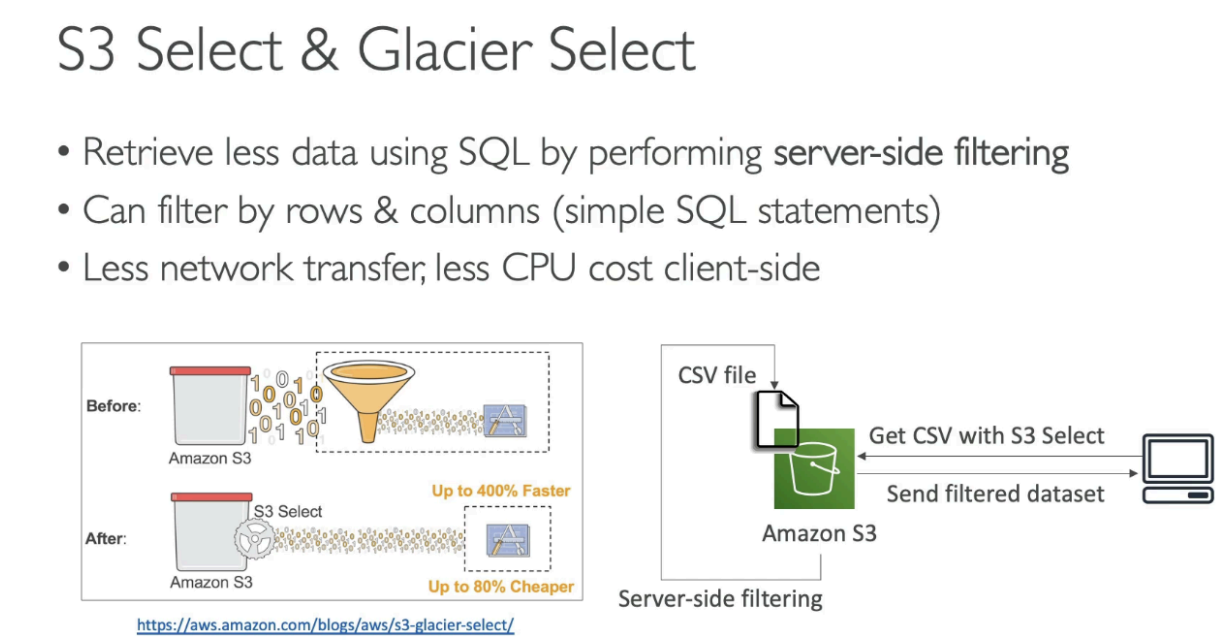
- Increase transfer speed by transferring file to AWS Edge location which will forward the data to S3 bucket in the target region
- Compatible with Multi-part upload

S3 Byte Range Fetches



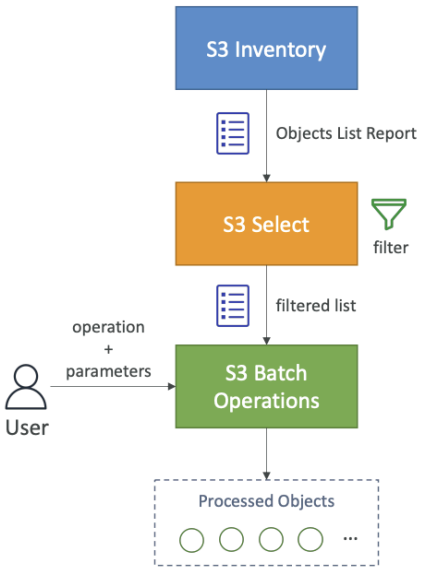
- Parallelize GETs by requesting specific byte ranges
- Better resilience in case of failures
- Can be used to speed up downloads
- Can be used to retrieve only partial data (ex: head of a file)

S3 Select & Glacier Select



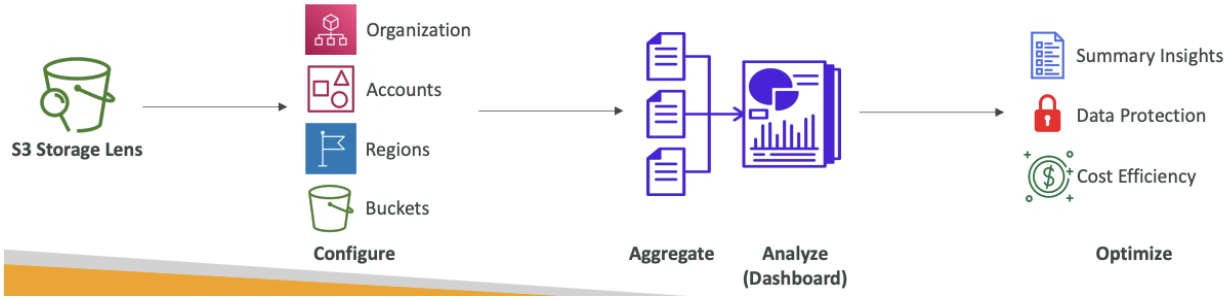
- Retrieve less data using SQL by performing server-side filtering
    - Can filter by rows/columns
- Less network transfer, less CPI cost client-side

S3 Batch Operations



- Perform bulk operations on existing AS3 objects with a single request
    - Modify object metadata & properties
    - Copy objects between S3 objects
    - Encrypt un-encrypted objects
    - Modify ACLs, tags
    - Restore objects from S3 glacier
    - Invoke lambda function to perform custom action on each object
- Job consists of a list of objects, action to perform, and optional parameters
- Manages retries, tracks progress, sends completion notifications, generate reports
    - S3 inventory to get object list and S3 Select to filter objects

S3 Storage Lens



- Understand, analyze, and optimize storage across entire AWS organization
    - Discover anomalies, identify cost efficiencies, and apply data protection best practices across entire AWS organization (30 days usage & activity metrics)
    - Aggregate data for Organization, specific accounts, regions, buckets, or prefixes
    - Default dashboard or create your own
    - Can be configured to export metrics daily to S3 bucket (export in CSV, parquet)

Default Dashboard

- Visualize summarized insights and trends for both free and advanced metrics

- Shows multi-region and multi-account data
- Preconfigured by S3
- Can't be deleted, only disabled

Metrics

Storage Lens – Metrics

- Summary Metrics
  - General insights about your S3 storage
  - StorageBytes, ObjectCount…
  - Use cases: identify the fastest-growing (or not used) buckets and prefixes
- Cost-Optimization Metrics
  - Provide insights to manage and optimize your storage costs
  - NonCurrentVersionStorageBytes, IncompleteMultipartUploadStorageBytes…
  - Use cases: identify buckets with incomplete multipart uploaded older than 7 days, Identify which objects could be transitioned to lower-cost storage class

- Summary Metrics
  - General insights of S3 storage
    - StorageBytes, ObjectCount…
  - Use cases: identify fastest growing (or not used) buckets and prefixes

- Cost Optimization Metrics
  - Provide insights to manage and optimize storage costs
    - NonCurrentVersionStorageBytes, IncompleteMultipartUploadStorageBytes…
  - Use cases: identify buckets with incomplete multipart uploaded older than 7 days, identify which object can be transitioned to lower cost storage class
- Data protection Metrics
  - Provide insights to data protection features
    - VersioningEnabledBucketCount, MFADeleteEnabledBucketCount, SSEKMSEnabledBucketCount, CrossRegionReplicationRuleCount…
  - Use cases: identify buckets that aren't following data protection best practices

Storage Lens – Metrics

- Data-Protection Metrics
  - Provide insights for data protection features
  - VersioningEnabledBucketCount, MFADeleteEnabledBucketCount, SSEKMSEnabledBucketCount, CrossRegionReplicationRuleCount…
  - Use cases: identify buckets that aren't following data-protection best practices

- Access-management Metrics
  - Provide insights for S3 Object Ownership
  - ObjectOwnershipBucketOwnerEnforcedBucketCount…
  - Use cases: identify which Object Ownership settings your buckets use

- Event Metrics
  - Provide insights for S3 Event Notifications
  - EventNotificationEnabledBucketCount (identify which buckets have S3 Event Notifications configured)

- Access Management metrics
  - Provide insights for S3 Object Ownership
    - ObjectOwnershipBucketOwnerEnforcedBucketCount…
  - Use case: identify which Object Ownership settings buckets use
- Event Metrics
  - Provide insights for S3 Event Notifications
    - EventNotificationEnabledBucketCount (identify which buckets have S3 Event Notifications configured)

Storage Lens – Metrics

- Performance Metrics
  - Provide insights for S3 Transfer Acceleration
  - TransferAccelerationEnabledBucketCount (identify which buckets have S3 Transfer Acceleration enabled)

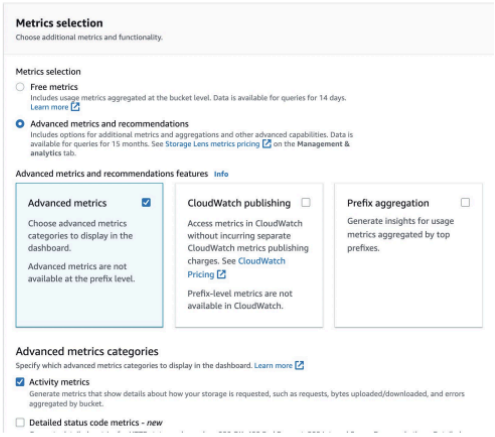
- Activity Metrics
  - Provide insights about how your storage is requested
  - AllRequests, GetRequests, PutRequests, ListRequests, BytesDownloaded…

- Detailed Status Code Metrics
  - Provide insights for HTTP status codes
  - 200OKStatusCount, 403ForbiddenErrorCount, 404NotFoundErrorCount…

Free vs Paid

Storage Lens – Free vs. Paid

- Free Metrics
  - Automatically available for all customers
  - Contains around 28 usage metrics
  - Data is available for queries for 14 days
- Advanced Metrics and Recommendations
  - Additional paid metrics and features
  - **Advanced Metrics** – Activity, Advanced Cost Optimization, Advanced Data Protection, Status Code
  - **CloudWatch Publishing** – Access metrics in CloudWatch without additional charges
  - **Prefix Aggregation** – Collect metrics at the prefix level
  - Data is available for queries for 15 months

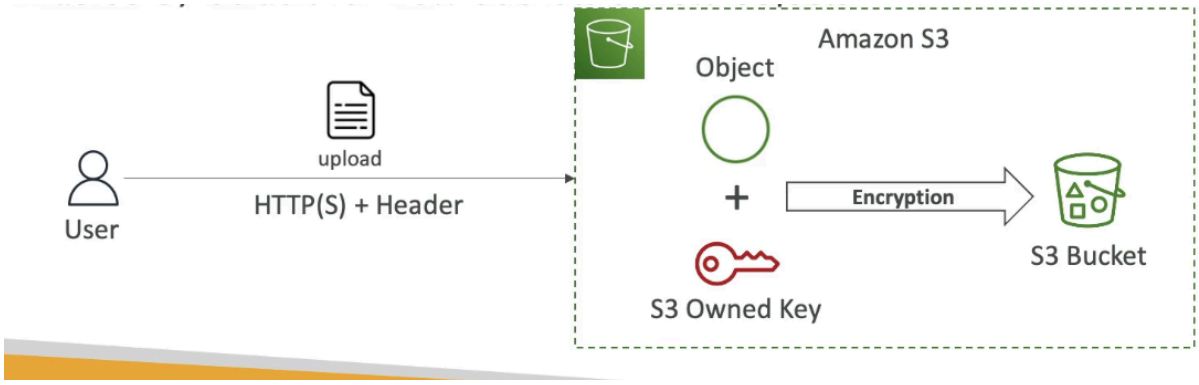


Section 14: Amazon S3 Security

S3 Encryption

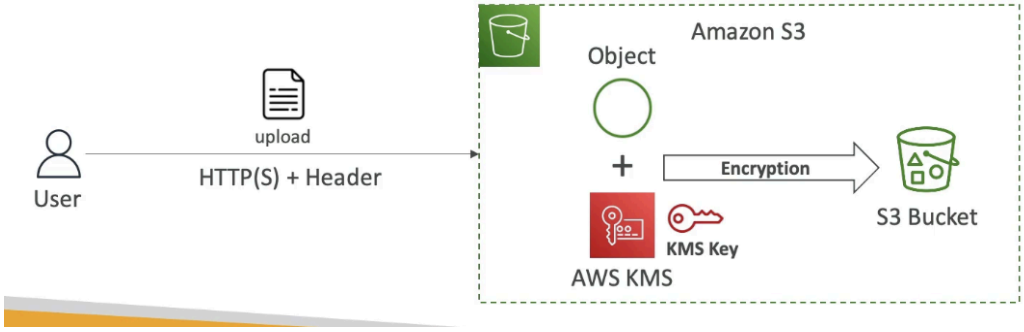
- Server Side Encryption (SSE)
  - Amazon S3 managed keys (SSE-S3) - enabled by default
    - Encrypts S3 objects using keys managed by AWS
  - SSE-KMS with KMS keys
    - Leverage AWS KMS to manage keys
  - SSE with Customer Provided Keys (SSE-C)
    - Manage own encryption keys
- Client Side Encryption

SSE-S3 Encryption



- Encryption is handled server side and keys are AWS managed and owned
  - Encryption type is AES-256
- Must set header “x-amz-server-side-encryption”: “AES256”
- Enabled by default for new buckets/objects

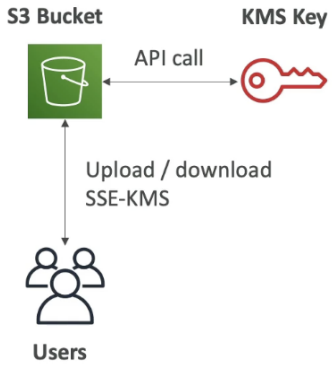
SSE-KMS Encryption



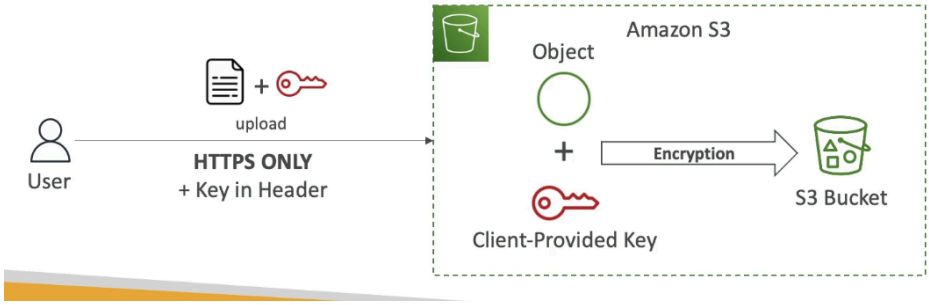
- Encryption using keys managed by AWS KMS
  - User control + audit key usage via CloudTrail
- Header: “x-amz-server-side-encryption”: “aws:kms”

KMS Limitation

- If you use KMS key, may be impacted with KMS limits
- When you upload, it calls GenerateDataKey KMS API and download calls Decrypt KMS API
  - Count towards KMS quota per second, but can request a quota increase using Service Quotas Console

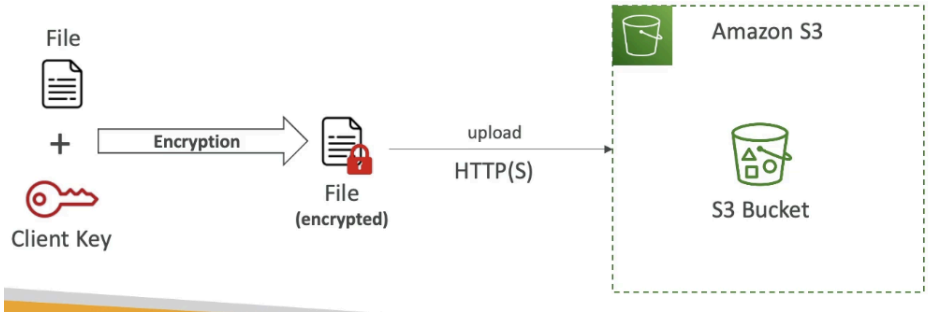


SSE-C



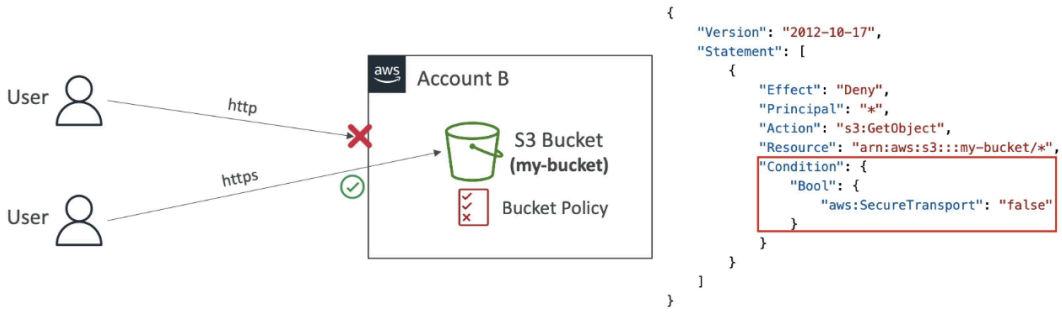
- Server side encryption fully managed outside of AWS by customer
    - S3 does not store encryption key
- HTTPS must be used
- Encryption key must be provided in HTTP headers for every HTTP request made

Client Side Encryption



- Client libraries to implement where clients must encrypt data themselves before sending to S3 and decryption of data when retrieving from S3
- Customer fully manages the keys and encryption cycle

Encryption in Transit (SSL / TLS)




```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::my-bucket/*",
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "false"
                }
            }
        }
    ]
}
```

- Encryption in flight also called SSL / TLS
- S3 exposes 2 endpoints:
    - HTTP (non encrypted)
    - HTTPS (encryption in flight)
- HTTPS recommended and mandatory for SSE-C
    - Most clients use HTTPS endpoints by default

Force Encryption in Transit

- aws:SecureTransport

S3 Default Encryption vs Bucket Policies

Amazon S3 – Default Encryption vs. Bucket Policies

- SSE-S3 encryption is automatically applied to new objects stored in S3 bucket
- Optionally, you can "force encryption" using a bucket policy and refuse any API call to PUT an S3 object without encryption headers (SSE-KMS or SSE-C)

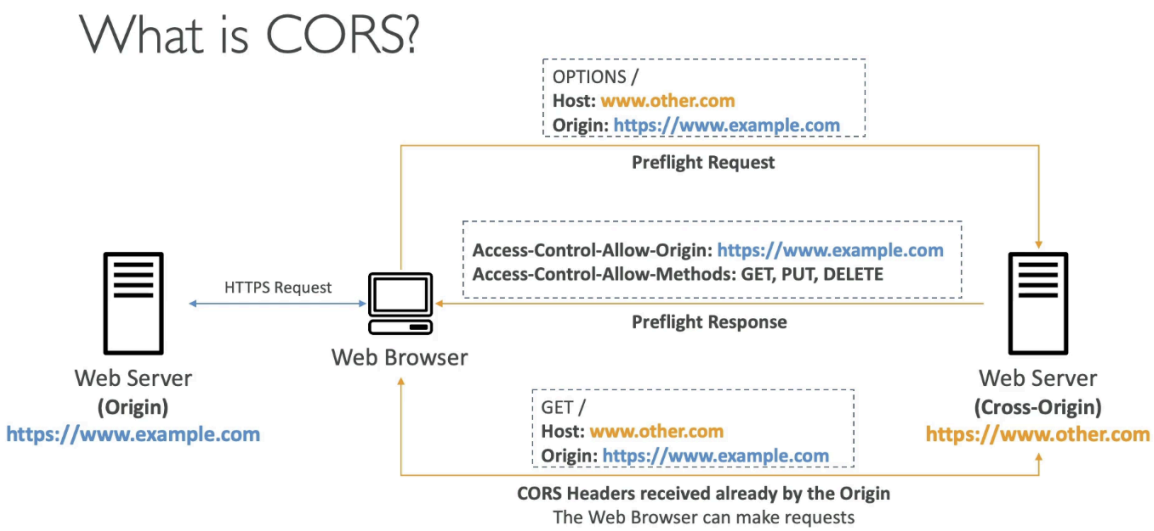
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "s3:PutObject",
            "Principal": "*",
            "Resource": "arn:aws:s3:::my-bucket/*",
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-server-side-encryption": "aws:kms"
                }
            }
        }
    ]
}
```

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "s3:PutObject",
            "Principal": "*",
            "Resource": "arn:aws:s3:::my-bucket/*",
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption-customer-algorithm": "true"
                }
            }
        }
    ]
}
```

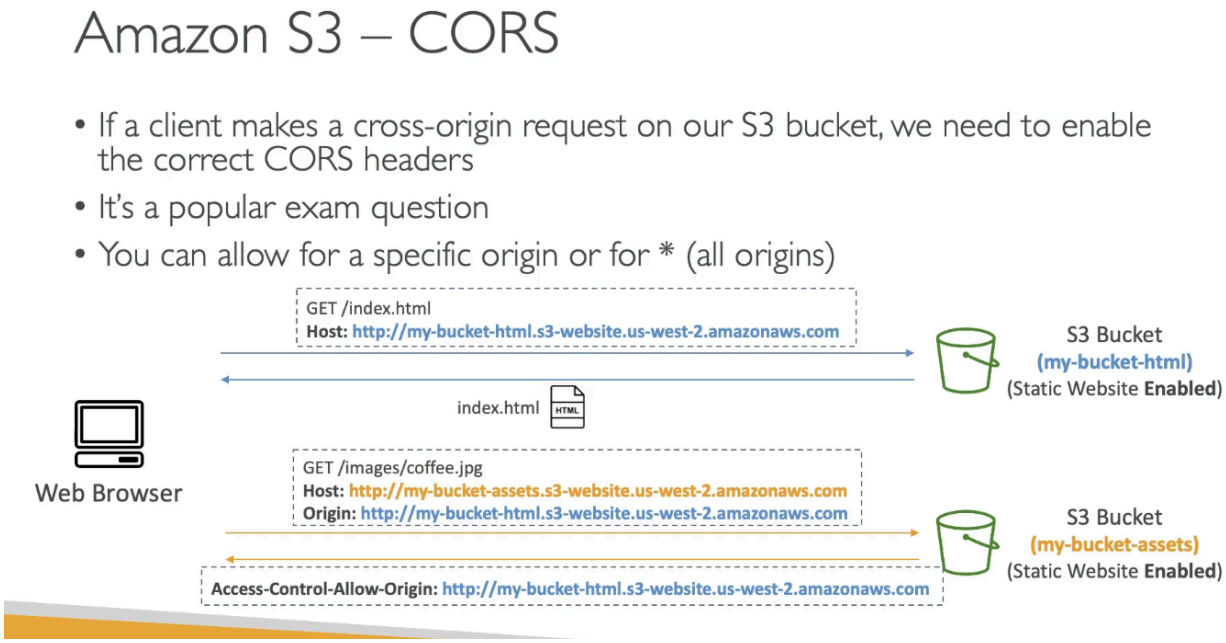
- Note: Bucket Policies are evaluated before "Default Encryption"

- SSE-S3 set by default, but a “force encryption” can be done using bucket policy to refuse any API call without encryption headers
- Bucket policies evaluated before default encryption

S3 CORS



- Origin = scheme (protocol) + host (domain) + port
  - https://example.com (implied port is 443 for HTTPS, 80 for HTTP)
- Web browser based mechanism to allow requests to other origins while visiting the main origin
  - Same origin: http://example.com/app & http://example.com/app2
  - Different origin: http://www.example.com & http://other.example.com
- Requests won’t be fulfilled unless the other origin allows for the requests, using CORS headers



- If a client makes a CORS request on S3 bucket, we need to enable the correct CORS headers
    - Can allow specific origin or * for all origins

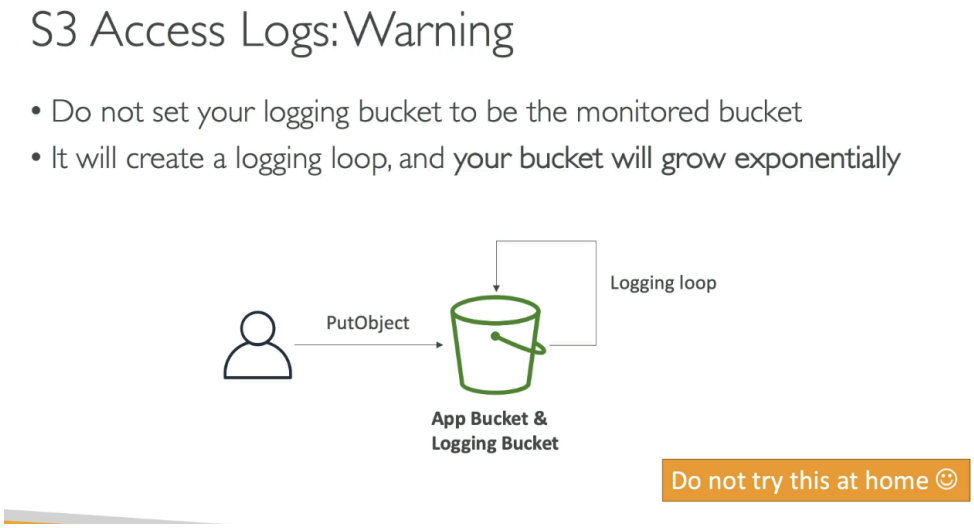
S3 MFA Delete

- MFA will be required to:
    - Permanently delete an object version
    - Suspend versioning on bucket
- MFA won't be required to:
    - Enable versioning
    - List deleted versions
- To use MFA delete, versioning must be enabled on the bucket
- Only bucket owner (root account) can enable / disable MFA delete
    - Only done via CLI, not UI

S3 Access Logs

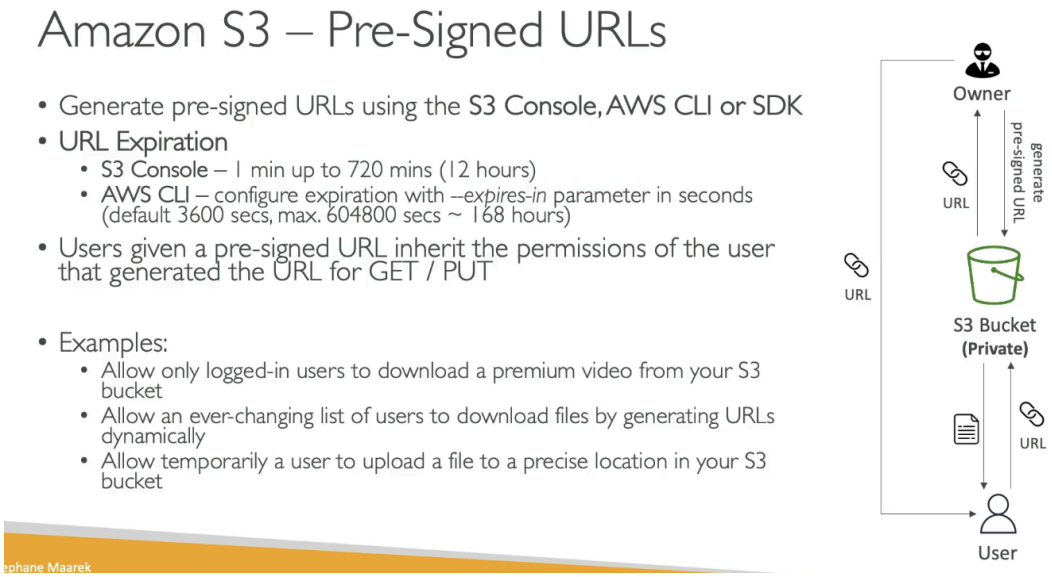
- For audit purposes, log all access to S3 buckets
- Any request made to S3 from any account will be logged to another S3 bucket in the same AWS region

Warning



- Don't set logging bucket to be monitored bucket, creates a logging loop and bucket grows exponentially

S3 Pre-signed URLs



- Users given a pre-signed URL inherit the permissions of the user that generated the URL for GET / PUT
  - Temporary access for specific files for upload or download
- Generate via Console, CLI, SDK
  - Console: 1 min to 12 hours
  - CLI: configure expiration

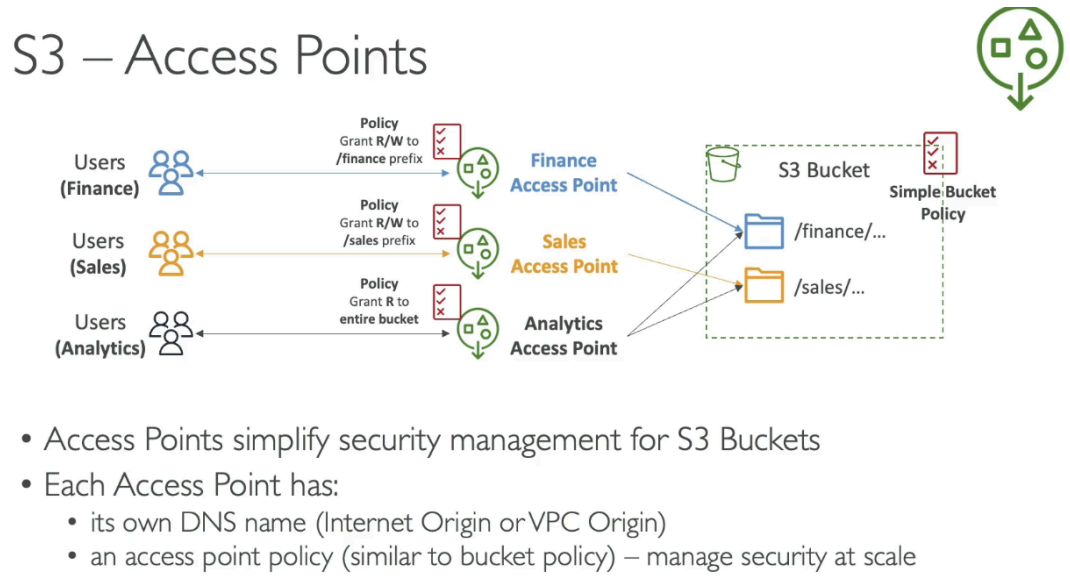
Glacier Vault Lock

- Lock glacier vault to adopt WORM (write once, read many)
- Create a vault lock policy, lock policy for future edits (can no longer be changed or deleted)
- Helpful for compliance and data retention

S3 Object Lock

- Versioning must be enabled, adopt WORM model to block object version deletion for a specified amount of time
- Retention mode:
    - Compliance
        - Object versions can't be overwritten or deleted by any user; including root user
        - Objects retention modes can't be changed and retention period can't be shortened
    - Governance
        - Most users can't overwrite or delete an object version or alter its lock settings
        - Some users can special permissions to change retention or delete the object
- Retention period: protect the object for a fixed period, can be extended
- Legal hold
    - Protect the object indefinitely, independent from retention period
    - Can be freely placed and removed using s3:PutObjectLegalHold

S3 Access Points



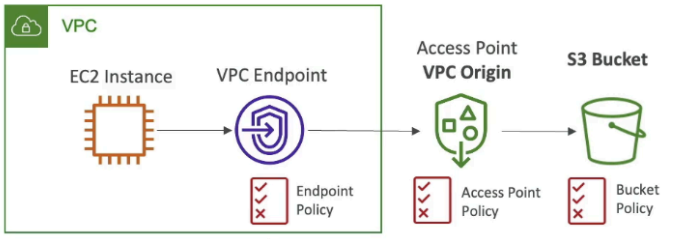
- Access Points simplify security management for S3 Buckets
- Each Access Point has:
    - its own DNS name (Internet Origin or VPC Origin)
    - an access point policy (similar to bucket policy) – manage security at scale

- Separate policies to allow access to specific prefix
- Simple bucket policy
- Each access point has:
    - Own DNS name (internet origin or VPC origin)
    - Access point policy (similar to bucket policy) - manage security at scale

Access Points - VPC Origin

S3 – Access Points – VPC Origin

- We can define the access point to be accessible only from within the VPC
- You must create a VPC Endpoint to access the Access Point (Gateway or Interface Endpoint)
- The VPC Endpoint Policy must allow access to the target bucket and Access Point

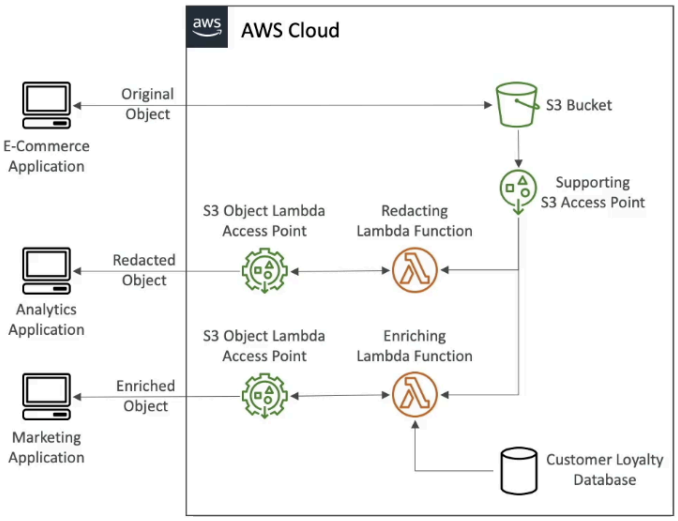



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::awsexamplebucket1/*",
                "arn:aws:s3:us-west-2:123456789012:accesspoint/example-vpc-ap/object/*"
            ]
        }
    ]
}
```

S3 Object Lambda

S3 Object Lambda

- Use AWS Lambda Functions to change the object before it is retrieved by the caller application
- Only one S3 bucket is needed, on top of which we create S3 Access Point and S3 Object Lambda Access Points.
- Use Cases:
  - Redacting personally identifiable information for analytics or non-production environments.
  - Converting across data formats, such as converting XML to JSON.
  - Resizing and watermarking images on the fly using caller-specific details, such as the user who requested the object.



- Use Lambda functions to change the object before it is retrieved by the caller application
- Only 1 bucket is needed, on top of S3 Access point and S3 Object Lambda Access Point
- Use cases: redact PI data, converting across data formats, etc…

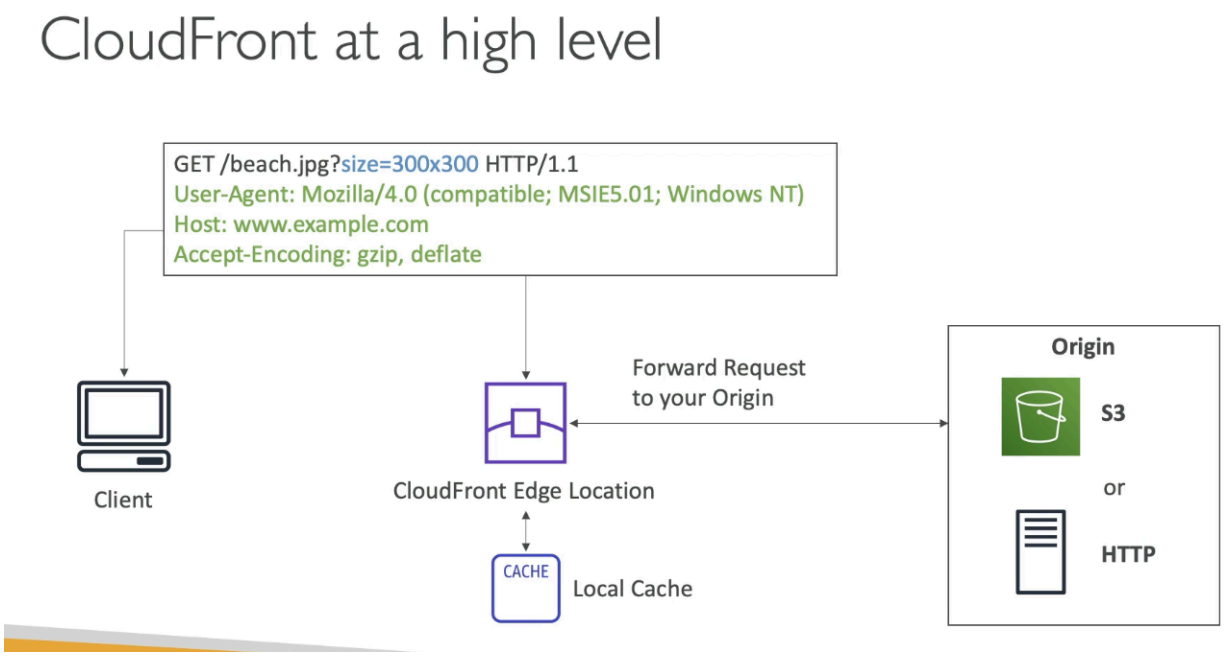
Section 15: CloudFront

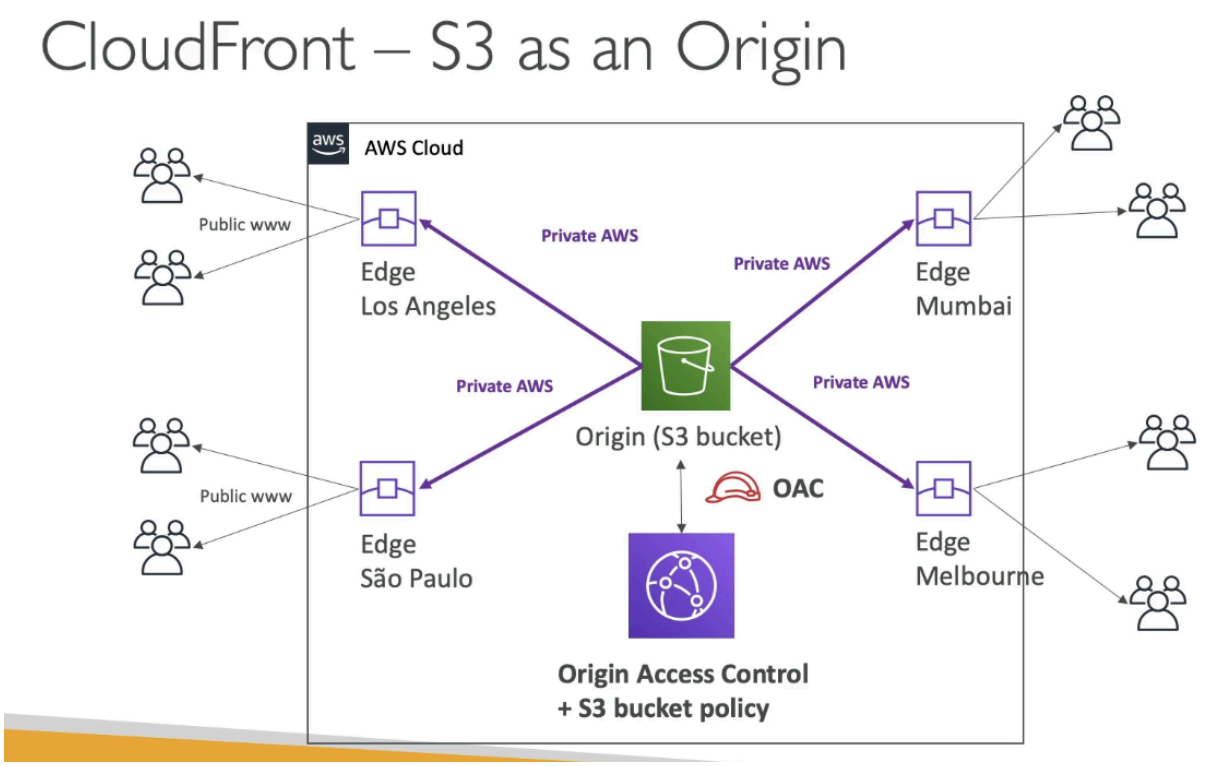
CloudFront Overview

- Content Delivery Network (CDN)
- Improves read performance, content is cached at the edge to improve user experience
    - 216 edge locations (point of presence)
- DDoS protection, integration with Shield and WAF

Origins

- S3 Bucket
    - For distributing files and caching at the edge
    - Enhanced security with CloudFront Origin Access Control (OAC)
    - OAC replaces Origin Access Identity (OAI)
    - CloudFront and be used as an ingress (upload files to S3)
- Custom Origin (HTTP)
    - ALB, EC2, S3 website, any HTTP backend



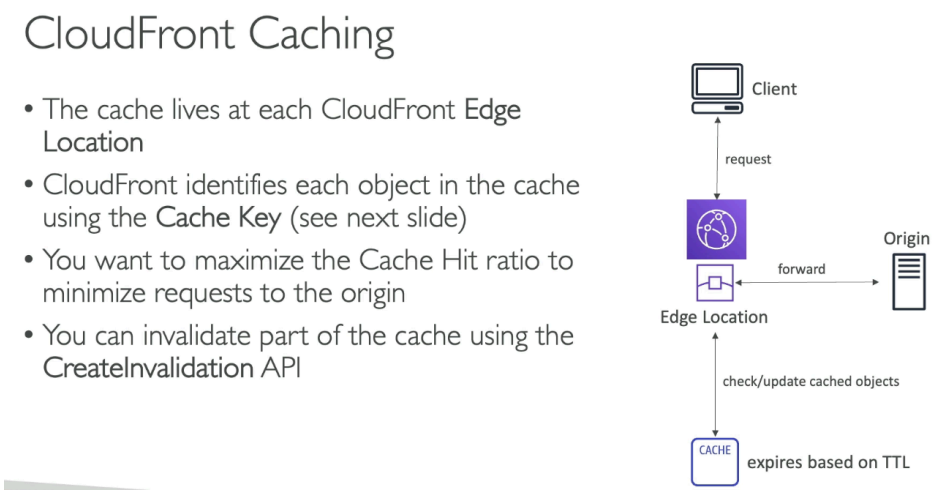


CloudFront vs S3 Cross Region Replication

- CloudFront:
  - Global Edge network
  - Files are cached for TTL
  - Great for static content that must be available everywhere

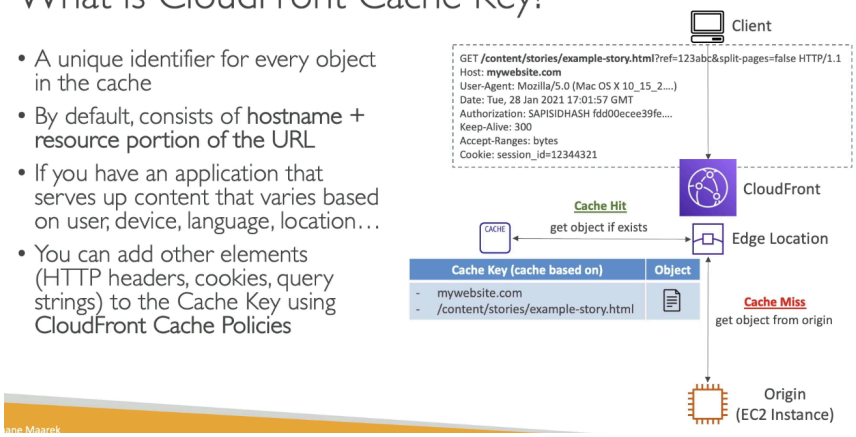
- S3 Cross Region Replication
    - Must be setup for each region you want replication
    - Files updated near real-time
    - Read only
    - Great for dynamic content that needs to be available at low-latency in a few regions

CloudFront - Caching & Caching Policies



- Cache lives at edge locations
- CloudFront identifies each object in the cache using Cache Key
- Maximize cache hit ratio to minimize requests to origin
- Can invalidate part of the cache using CreateInvalidation API

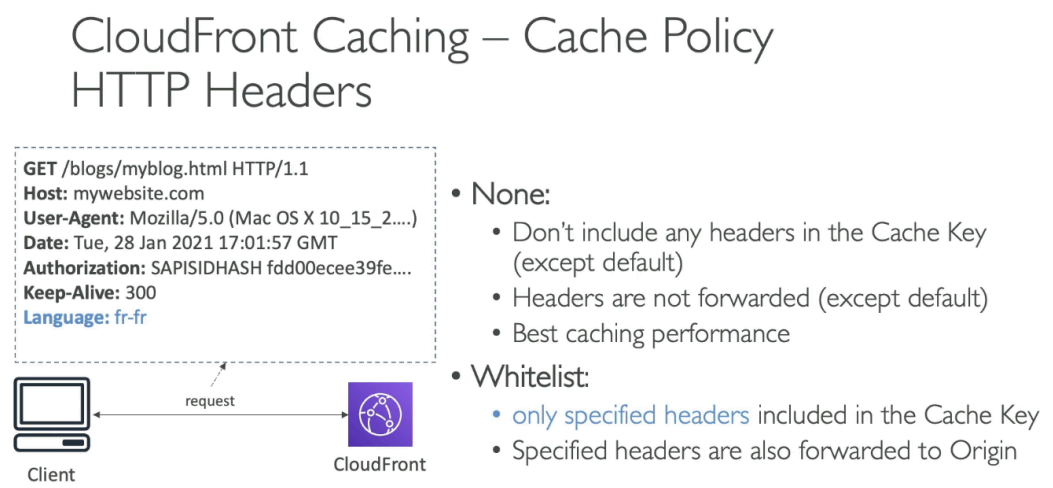
What is CloudFront Cache Key?

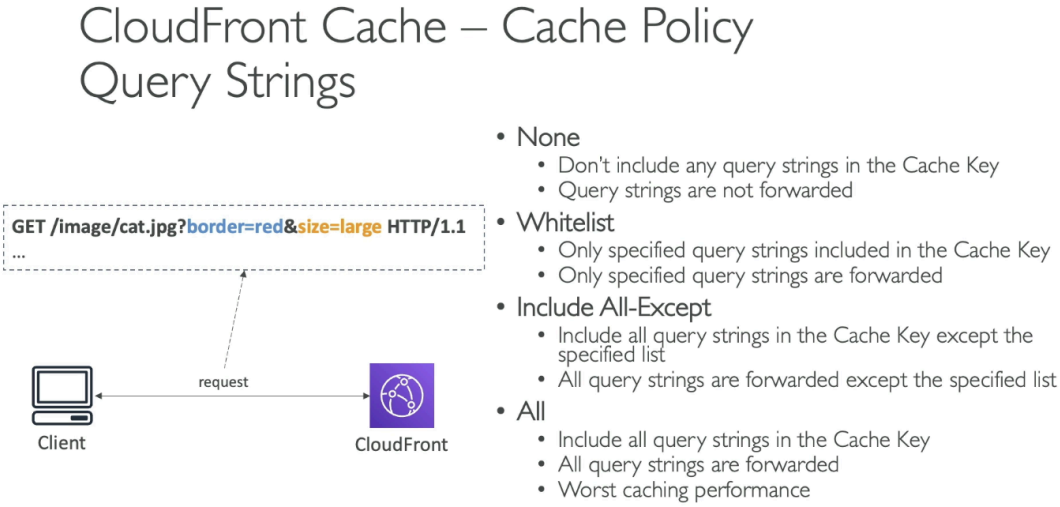


- Unique identifier for every object in the cache

- By default, consists of hostname + resource portion of URL
    - Other elements HTTP headers, cookies, etc… to the cache key using CloudFront Cache Policy

Cache Policy

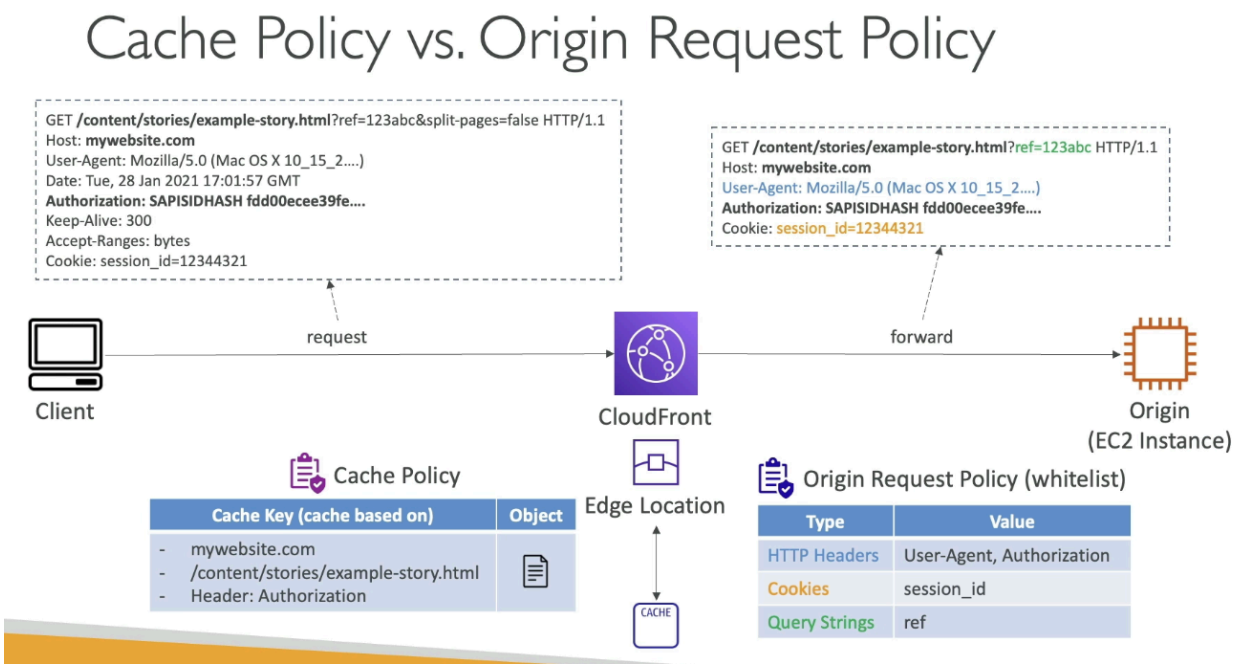




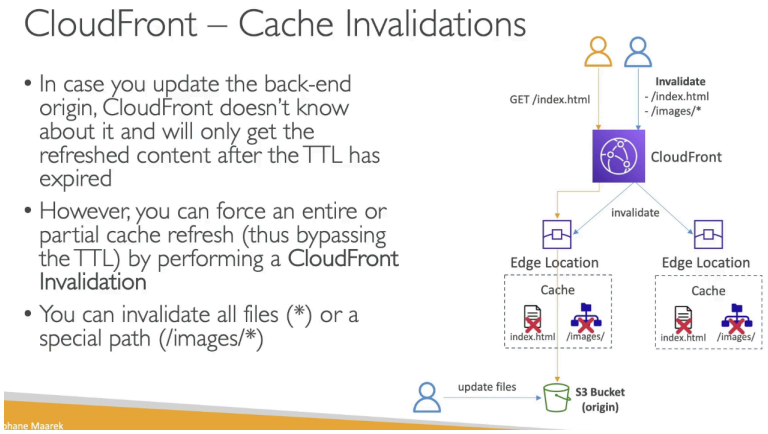
- Cache based on:
    - HTTP Headers: none - whitelist
    - Cookies: none, whitelist, include all except, all
    - Query String: none, whitelist, include all except, all
- Control the TTL, can be set by origin using the cache control header, expires header…
- Create own policy or predefined policies
- All HTTP headers, cookies, and query strings that you include in cache key are automatically included in origin requests

Origin Request Policy

- Specify values want to include in origin request without including in Cache Key (no duplicated cached content)
    - HTTP Headers: none, whitelist, all viewer headers options
    - Cookies: none, whitelist, all
    - Query strings: none, whitelist, all
- Ability to add CloudFront HTTP headers and custom headers to origin request that are not included in viewer request
- Can create own or use predefined policy

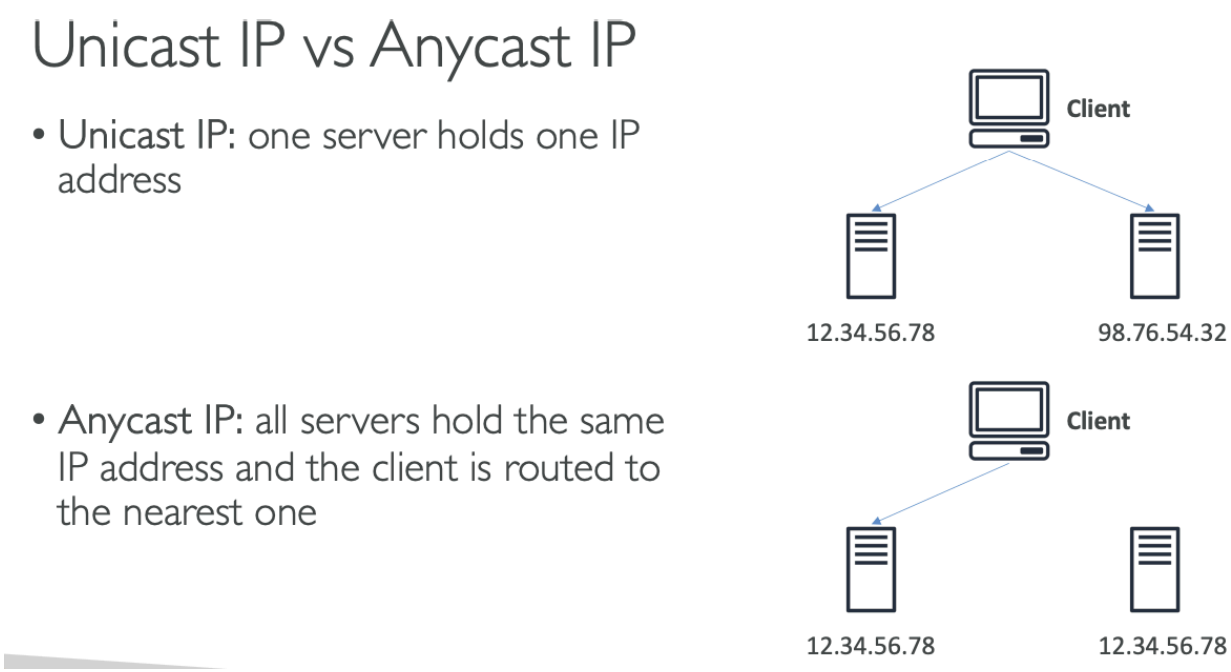


Cache Invalidations



- In case you update the backend origin, CloudFront doesn't know and will only get refreshed content after TTL expired. Force entire or partial cache refresh (bypasses TTL) via CloudFront Invalidation

Unicast IP vs Anycast IP



- Unicast IP: One server holds 1 IP address
- Anycast IP: all servers hold same IP address and client is routed to the nearest one

AWS Global Accelerator Overview

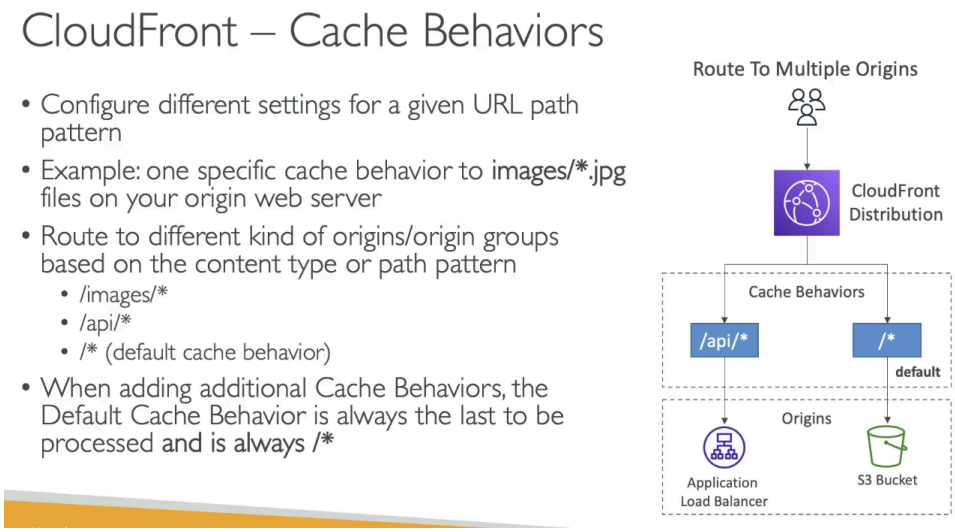
- If you have an application with global users, they go over public internet which adds latency with hops through routers.
- Leverages Anycast IP to use AWS internal network to route to application (edge location)
- 2 Anycast IP are created for application and anycast IP sends traffic directly to edge locations, then to application via AWS network
- Works with Elastic IP, EC2 Instances, ALB, NLB, public or private
- Consistent performance
    - Intelligent routing to lowest latency and fast regional failover
    - No issue with client cache (IP doesn't change)
- Health checks
    - Global accelerator performs health check of applications
    - Helps make application global (failover < 1 min for unhealthy)

    - Great for disaster recovery
- Security
    - Only 2 external IP need whitelisted
    - DDoS protection via AWS Shield

AWS Global Accelerator vs CloudFront

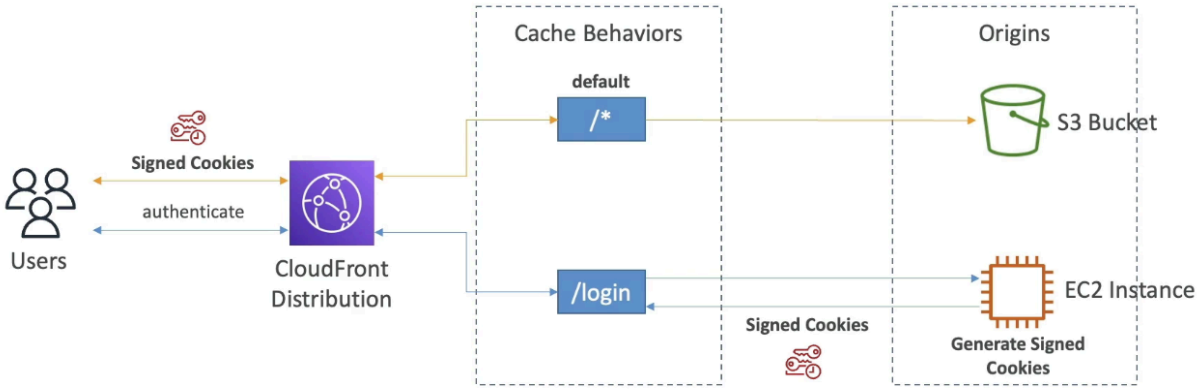
- Both use edge locations and AWS global network, integrations with AWS Shield
- CloudFront
    - Improves performance for both cacheable content (images / videos)
    - Dynamic content, with content served at edge
- Global Accelerator
    - Improves performance for wide range of applications over TCP or UDP
    - Proxying packers at edge to applications running in 1+ regions
    - Good for non HTTP use cases like gaming (UDP), voice over IP
    - Good for HTTP cases that require static IP or require deterministic, fast regional failover

Cache Behaviors

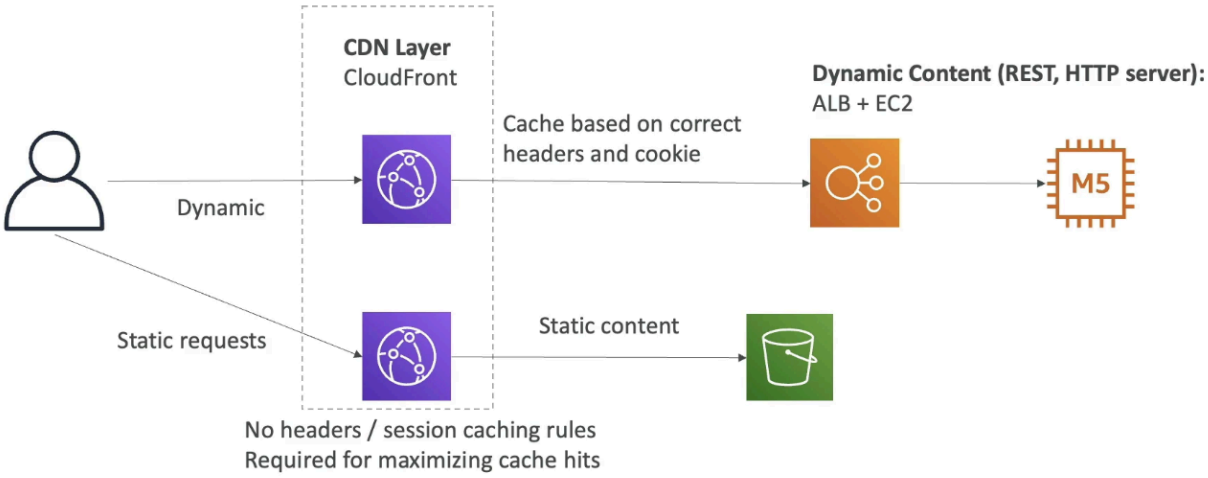


- Configure different settings for given URL path pattern
- Route to different kind of origins/origin groups based on the content type or path pattern
- When adding additional cache behavior, the default cache behavior is always the last processed and is always /*

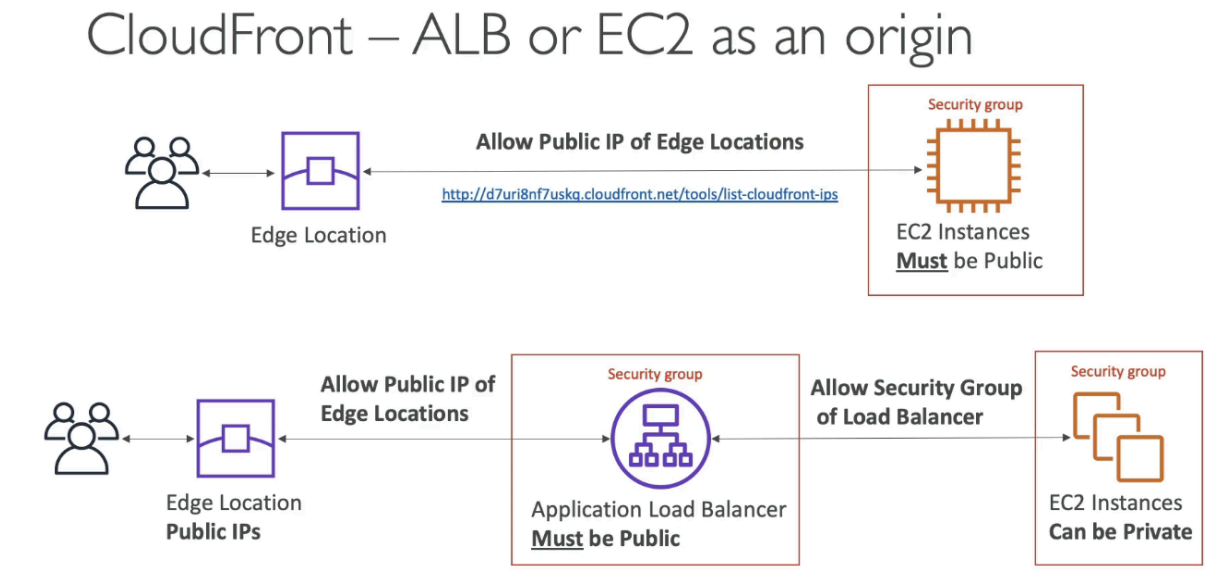
CloudFront – Cache Behaviors – Sign In Page



CloudFront – Maximize cache hits by separating static and dynamic distributions



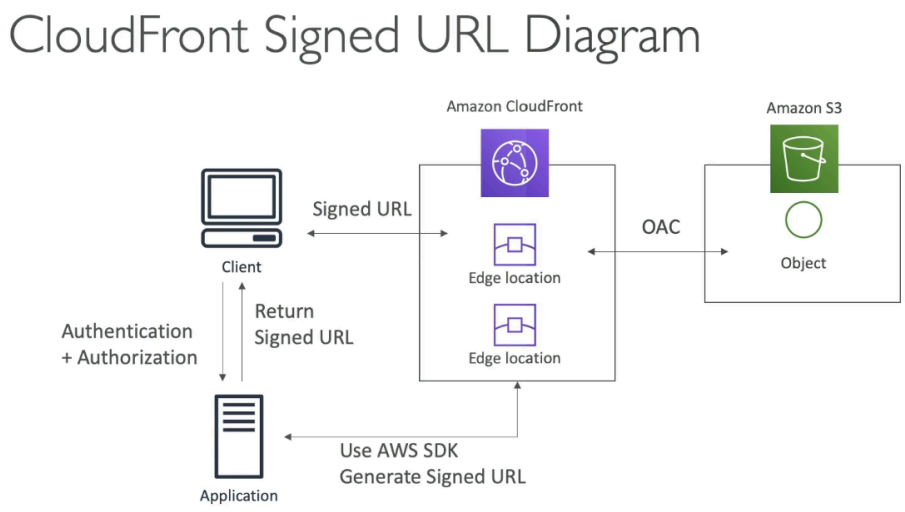
ALB as an Origin



Geo Restriction

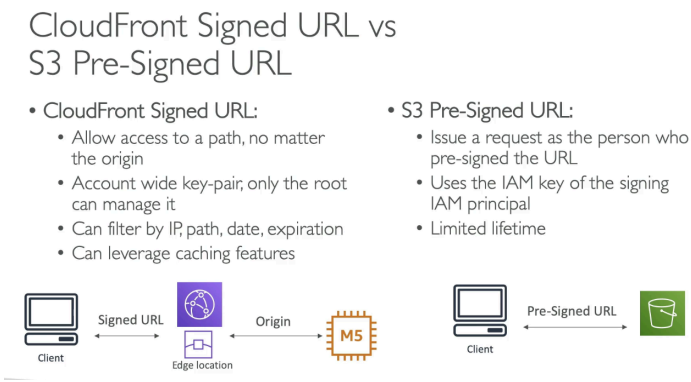
- Restrict who can access distribution
  - Allowlist: allow users to access content only if they're on a list of approved countries
  - Blocklist: prevent users from accessing content if they're on a banned country list
- Country determined via 3rd party Geo-IP database

Signed URL / Signed Cookies



- Must attach policy with:
    - URL expiration
    - IP ranges to access data
    - Trusted signers (which AWS accounts can create signed URLs)
- How long should the URL be valid for?
    - Shared content: short
    - Private content: years
- Signed URL = access to individual files (1 signed URL per file)
- Signed Cookies = access to multiple files (one signed cookie for many files)

Signed URL vs S3 Pre-signed URL



- CloudFront Signed URL
    - Allow access to path, no matter the origin
    - Account wide key pair, not only the root can manage
    - Can filter by IP, path, date, expiration
    - Can leverage caching features
- S3 Pre-Signed URL
    - Issue a request as person who pre-signed URL
    - Uses IAM key of the signing IAM principal
    - Limited Lifetime

Signed URL Process

- 2 types of signers
    - Trusted key group
        - Can leverage APIs to create and rotate keys (and IAM for API security)
    - AWS account that contains a CloudFront Key Pair
        - Need to manage keys using root account and console (not recommended)
- In CloudFront distribution, create 1+ trusted key groups
- Generate your own public / private key

- Private key used by applications to sign URLs
- Public key (uploaded) is used by CloudFront to verify URLs

CloudFront Advanced Concepts

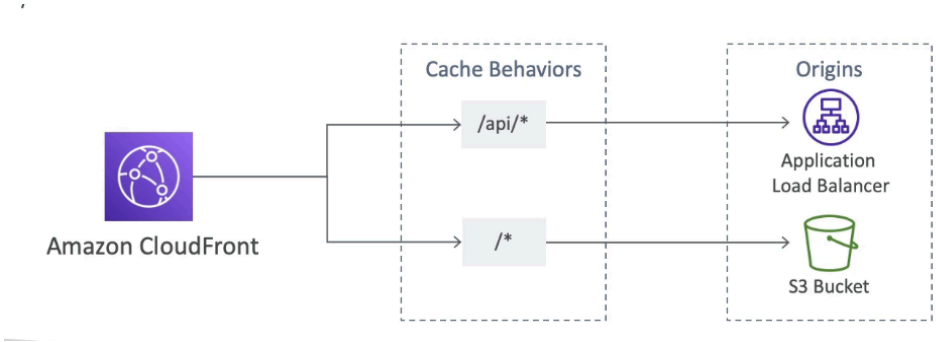
Pricing

- Cost of data out per edge location varies based on location of edge location

Price Classes

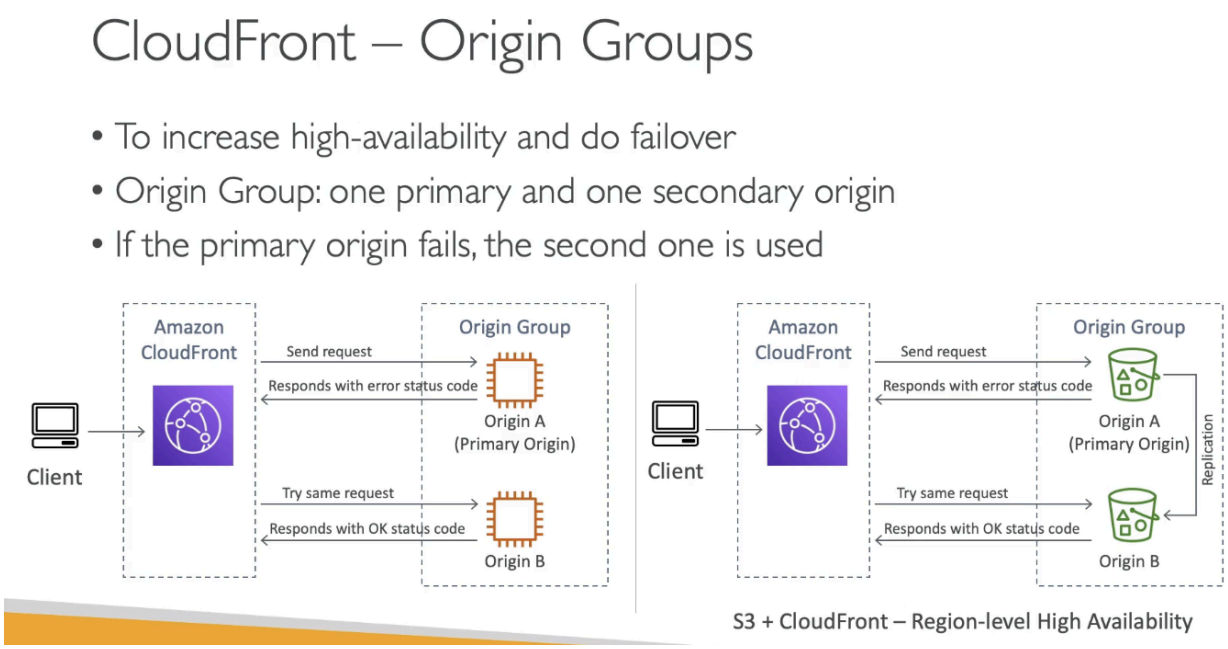
- Can reduce the number of edge locations for cost reduction
- 3 price classes:
    1. Price Class All: all regions, best performance
    2. Price Class 200: most regions, excludes most expensive regions
    3. Price Class 300: only the least expensive regions

Multiple Origin



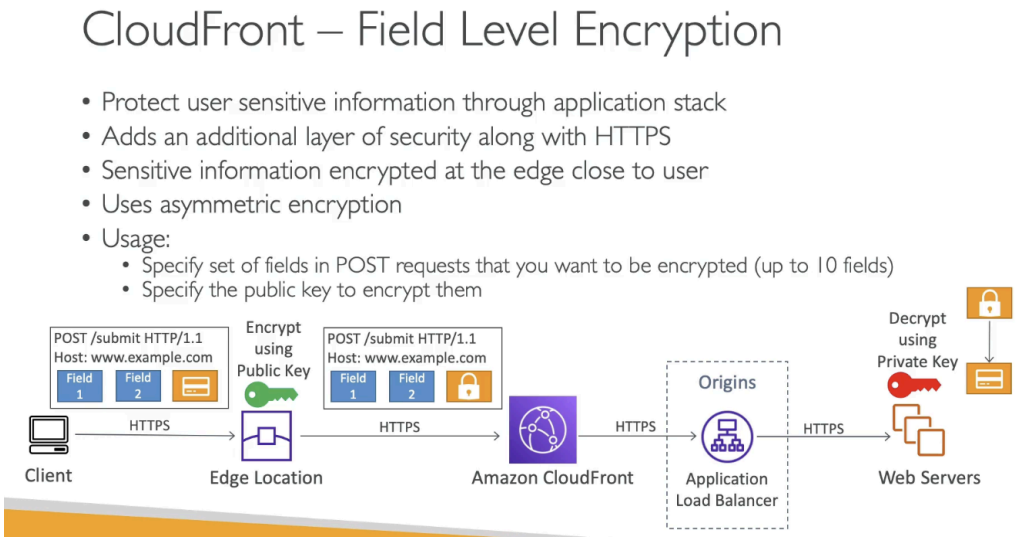
- To route to different kind of origins based on content type based on path pattern

Origin Groups



- Increase high availability and do failover
- Origin group: one primary and one secondary
- If the primary origin fails, second is used

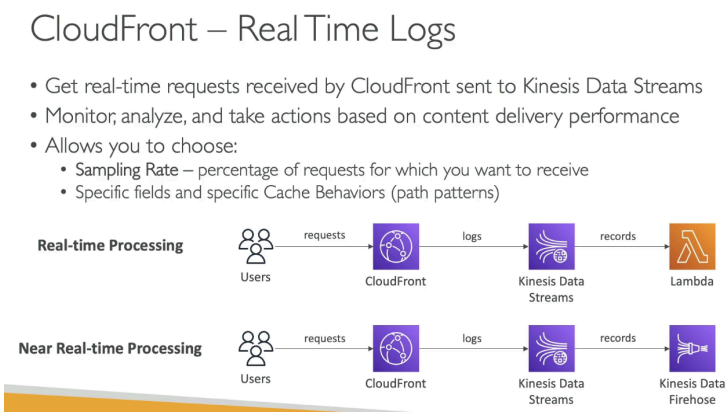
Field Level Encryption



- Protect user sensitive info through application stack
- Additional layer along with HTTPS
- Sensitive information encrypted at the edge close to user

- Uses asymmetric encryption
- Usage:
    - Specify set of fields in POST you want encrypted (up to 10 fields)
    - Specify the public key to encrypt

Real Time Logs



- Real time requests received by CloudFront sent to Kinesis Data stream
    - Monitor, analyze based on content delivery performance
- Choose:
    - Sampling rate: % of requests you want to receive
    - Specific fields and specific cache behaviors (path patterns)

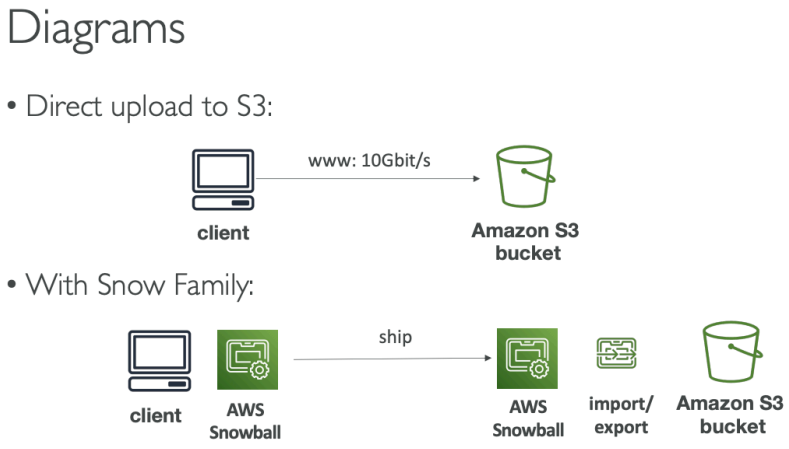
Section 16: AWS Storage Extras

AWS Snow Family Overview

- Highly secure portable devices to collect and process data at the edge AND migrate data in / out of AWS
    - Data migration: Snowcone, Snowball Edge, Snowmobile
    - Edge computing: Snowcone, snowball edge

Data Migration with Snow

- Offline devices to perform data migrations
    - More than a week to transfer data, use Snowball
    - Done physically



AWS Snow Family for Data Migrations

Snowcone

Snowball Edge

Snowmobile

	Snowcone & Snowcone SSD	Snowball Edge Storage Optimized	Snowmobile
Storage Capacity	8 TB HDD 14 TB SSD	80 TB - 210 TB	< 100 PB
Migration Size	Up to 24 TB, online and offline	Up to petabytes, offline	Up to exabytes, offline
DataSync agent	Pre-installed		

Snowball Edge

- Physical data transport to move TB or PB of data in / out of AWS
  - Alternative to moving data over network and paying network fees
- Pay per data transfer job
- Provide block storage and S3 compatible object storage
- Snowball Edge Storage Optimized
  - 80 TB of HDD or 210 TB of NVMe capacity for block volume and S3 compatible object storage
- Snowball Edge Compute Optimized
  - 42 TB or 28 TB NVMe capacity for block volume and S3 compatible object storage
- Use case: large data cloud migrations, disaster recovery

AWS Snow Cone & Snow Cone SSD

- Small portable computing anywhere, rugged and secure to withstand harsh environments
  - Light, device used for edge computing, storage, and data transfer
    - Must provide own battery / cables
  - Used where snowball does not fit (space limited)
- Sent back to AWS offline or connect to internet and use AWS DataSync to send data
- Snow Cone - 8 TB of HDD
- Snow Cone SSD - 14 TB SSD

AWS Snowmobile

- Transfer EBs of data with each has 100 PB capacity (can use multiple in parallel)
    - Better if need > 10 PB
- High security, temp controlled, GPS, 24/7 video surveillance

Usage Process

1. Request Snowball device from AWS for delivery
2. Install snowball client / AWS OpsHub on servers
3. Connect snowball to servers and copy files via client
4. Ship back to AWS
5. Data loaded in S3 bucket
6. Snowball completely wiped

What is Edge Computing?

- Process data while created on edge location
    - Edge location is somewhere that does not have internet access or computing power
- Use Snowball Edge / Snow Cone device
- Use cases: preprocess data, ML at edge, transcoding media streams…
- Ship back to AWS

Snow Family – Edge Computing

Snow Family – Edge Computing

- Snowcone & Snowcone SSD (smaller)
    - 2 CPUs, 4 GB of memory, wired or wireless access
    - USB-C power using a cord or the optional battery

- Snowball Edge – Compute Optimized
    - 104 vCPUs, 416 GiB of RAM
    - Optional GPU (useful for video processing or machine learning)
    - 28 TB NVMe or 42TB HDD usable storage
    - Storage Clustering available (up to 16 nodes)

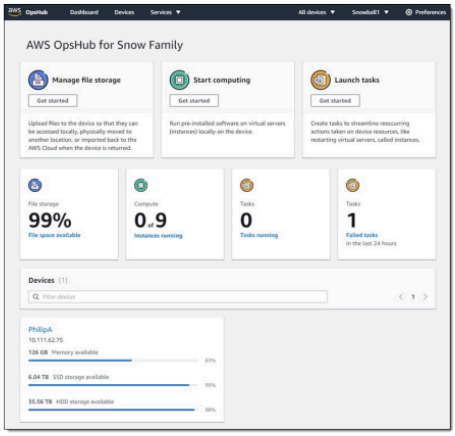
- Snowball Edge – Storage Optimized
    - Up to 40 vCPUs, 80 GiB of RAM, 80 TB storage
    - Up to 104 vCPUs, 416 GiB of RAM, 210 TB NVMe storage
- All: Can run EC2 Instances & AWS Lambda functions (using AWS IoT Greengrass)
- Long-term deployment options: 1 and 3 years discounted pricing

- Can run EC2 instances & lambda

AWS OpsHub

AWS OpsHub

- Historically, to use Snow Family devices, you needed a CLI (Command Line Interface tool)
- Today, you can use **AWS OpsHub** (a software you install on your computer / laptop) to manage your Snow Family Device
  - Unlocking and configuring single or clustered devices
  - Transferring files
  - Launching and managing instances running on Snow Family Devices
  - Monitor device metrics (storage capacity, active instances on your device)
  - Launch compatible AWS services on your devices (ex: Amazon EC2 instances, AWS DataSync, Network File System (NFS))



https://aws.amazon.com/blogs/aws/aws-snowball-edge-update/

- Software installed on computer to manage Snow Family device

Architecture: Snowball into Glacier

- Snowball cannot import to Glacier directly, must use S3 with lifecycle policy

Amazon FSx Overview

- Launch 3rd party high performance file systems in AWS; fully managed service

Amazon FSx for Windows (File Server)

Amazon FSx for Windows (File Server)

- **FSx for Windows** is a fully managed **Windows** file system share drive
- Supports SMB protocol & Windows NTFS
- Microsoft Active Directory integration, ACLs, user quotas
- **Can be mounted on Linux EC2 instances**
- Supports **Microsoft's Distributed File System (DFS) Namespaces** (group files across multiple FS)

- Scale up to 10s of GB/s, millions of IOPS, 100s PB of data
- Storage Options:
  - **SSD** – latency sensitive workloads (databases, media processing, data analytics, …)
  - **HDD** – broad spectrum of workloads (home directory, CMS, …)
- Can be accessed from your on-premises infrastructure (VPN or Direct Connect)
- Can be configured to be Multi-AZ (high availability)
- Data is backed-up daily to S3

- Fully managed Windows file system share drive
  - Supports SMB protocol & Windows NTFS
  - Microsoft Active Directory integration, ACLs, user quotas
- Can be mounted on Linux EC2 instances
- Supports Microsoft’s Distributed File System (DFS) Namespaces (groups files across multiple FS)
- Scale up to 10s of GB/s, millions of IOPS…
- Storage options:
  - SSD – latency sensitive workloads (DB, media processing, data analytics…)
  - HDD – broad spectrum of workloads
- Can be accessed from on premise infrastructure (VPN or Direct Connect)
- Can be configured for Multi AZ (high availability)
- Data backed up to S3 daily

Amazon FSx for Lustre

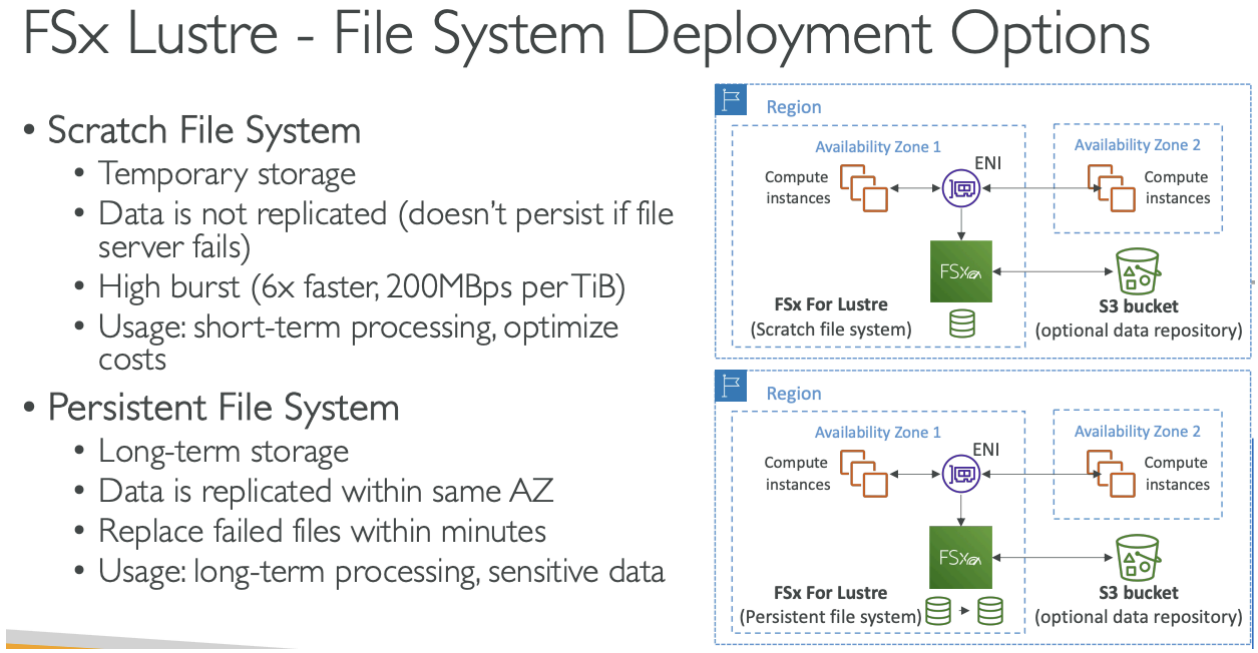
Amazon FSx for Lustre

- Lustre is a type of parallel distributed file system, for large-scale computing
- The name Lustre is derived from "Linux" and "cluster

- Machine Learning, High Performance Computing (HPC)
- Video Processing, Financial Modeling, Electronic Design Automation
- Scales up to 100s GB/s, millions of IOPS, sub-ms latencies
- Storage Options:
  - SSD – low-latency, IOPS intensive workloads, small & random file operations
  - HDD – throughput-intensive workloads, large & sequential file operations
- Seamless integration with S3
  - Can "read S3" as a file system (through FSx)
  - Can write the output of the computations back to S3 (through FSx)
- Can be used from on-premises servers (VPN or Direct Connect)

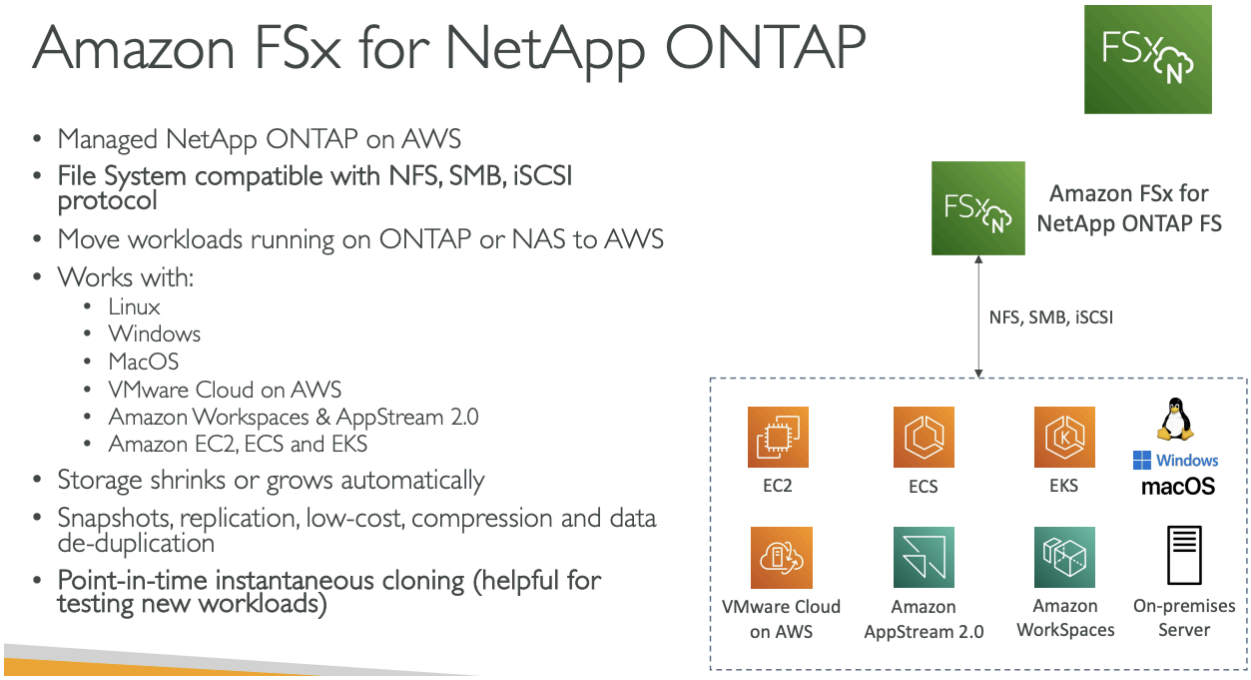
- Type of parallel distributed file system for large scale computing
    - Name derived from Linux and cluster
- ML, high performance computing (HPC)
    - Video processing, financial modeling…
    - Scales up to 100s GB, millions of IOPS, low latency
- Storage options:
    - SSD: low latency, IOPS sensitive workloads, small & random file operations
    - HDD: throughput intensive workloads, large & sequential file operations
- Seamless integration with S3
    - Can "read S3" as file system (through FSx)
    - Can write the output of computations back to S3 (through FSx)
- Can be used from on premise servers

FSx File System Deployment Options



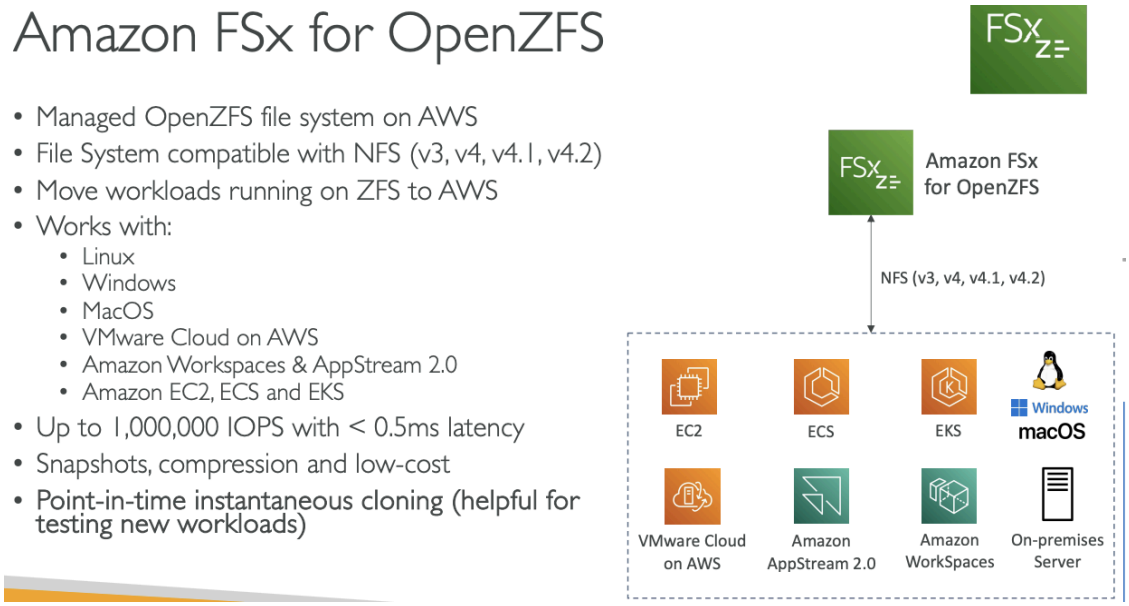
- Scratch File System
  - Temp storage, data not replicated (doesn't persist if file server fails)
  - High burst (6x faster, 200 MBps per TB)
  - Usage: short term processing, optimize cost
- Persistent File System
  - Long term storage, data replicated within same AZ
  - Replace failed files within minutes
  - Use case: long term processing, sensitive data

Amazon FSx for NetApp ONTAP



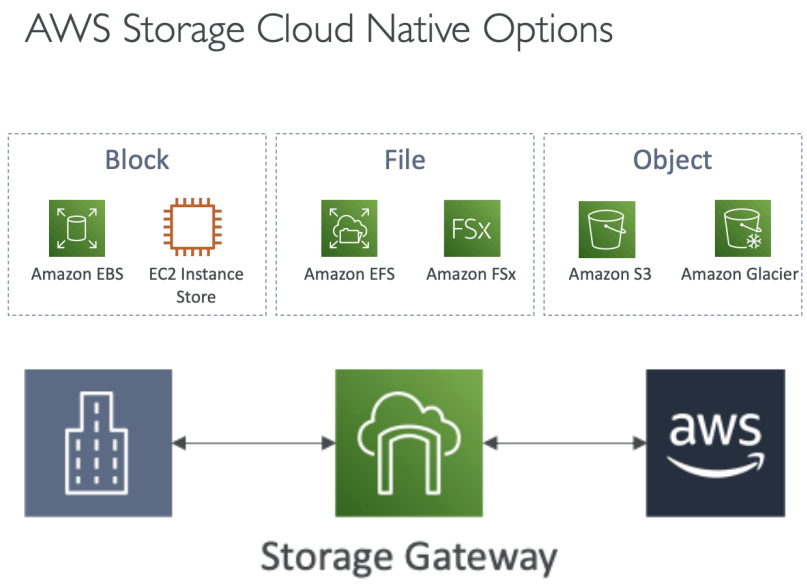
- Managed NetApp ONTAP on AWS
- Compatible with NFS, SMB, iSCSI protocol
- Move workloads running on ONTAP or NAS to AWS
- Works with many OS and services
- Storage auto scaling
- Snapshots, replication, low-cost, compression and data deduplication
- Point in time instantaneous cloning (for testing new workloads)

Amazon FSx for OpenZFS



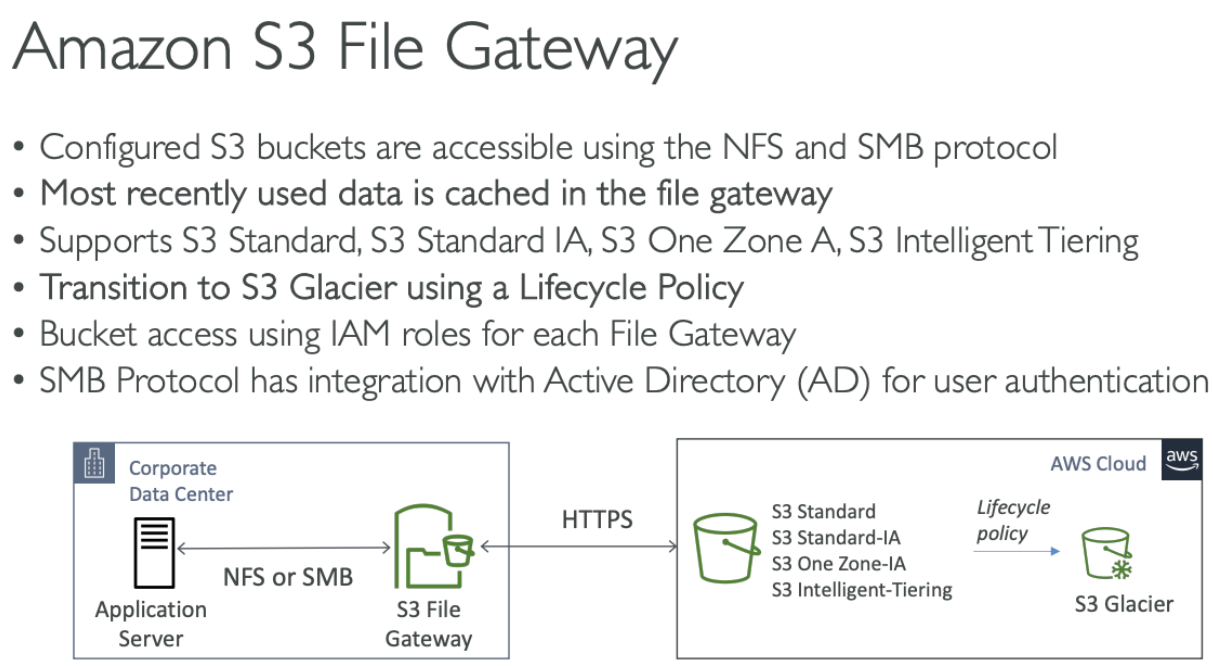
- Managed OpenZFS file system in AWS
- File system compatible with NFS
- Move workloads running on ZFS to AWS
- Works with all OS and services
- Up to 1 million IOPS with < 0.5 ms latency
- Snapshots, compression, low cost
- Point in time instantaneous cloning

AWS Storage Gateway



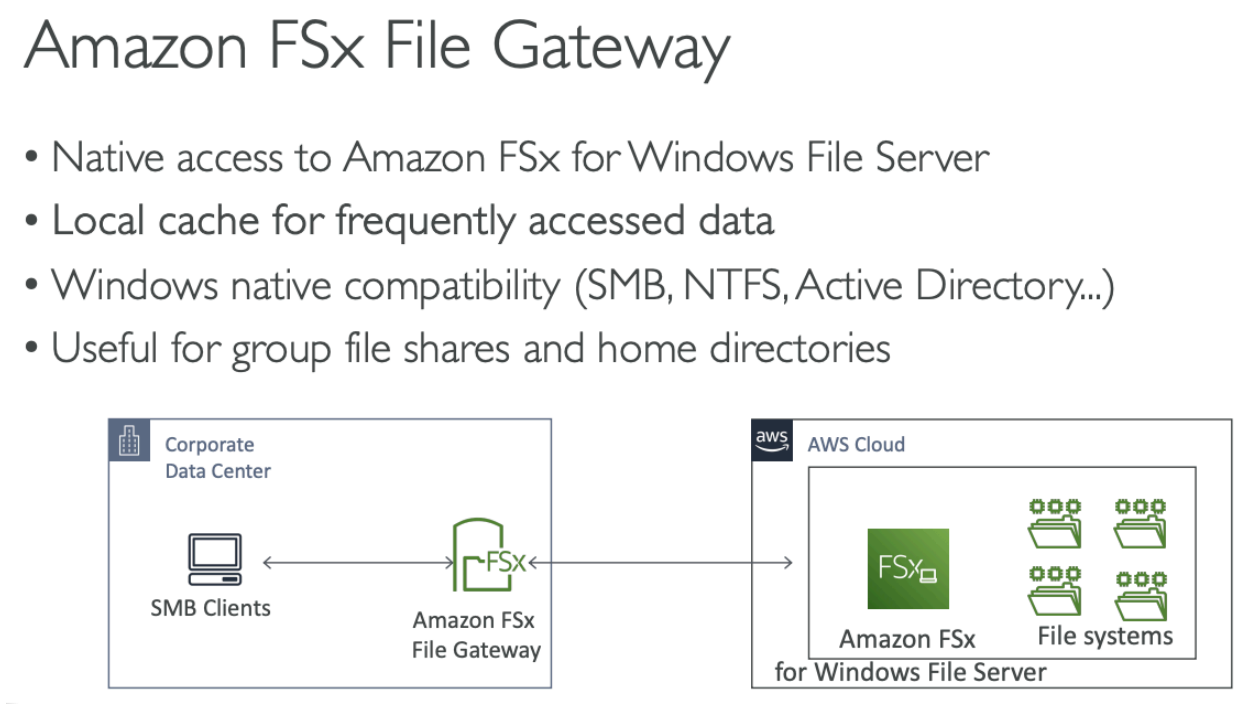
- Bridge between on premise and cloud
    - Use case: disaster recovery, backup and restore, tiered storage, on premise cache & low latency file access
- Types of Storage Gateway
    - S3 File Gateway
    - FSx File Gateway
    - Volume Gateway
    - Tape Gateway

S3 File Gateway



- Configured S3 buckets are accessible using NFS and SMB protocol
    - SMB has integration with Active Directory for user authentication
- Most recently used data is cached in file gateway
- Supports all S3 tiers except Glacier
    - Transition to S3 Glacier via lifecycle policy
- Bucket access using IAM roles for each file gateway

FSx File Gateway

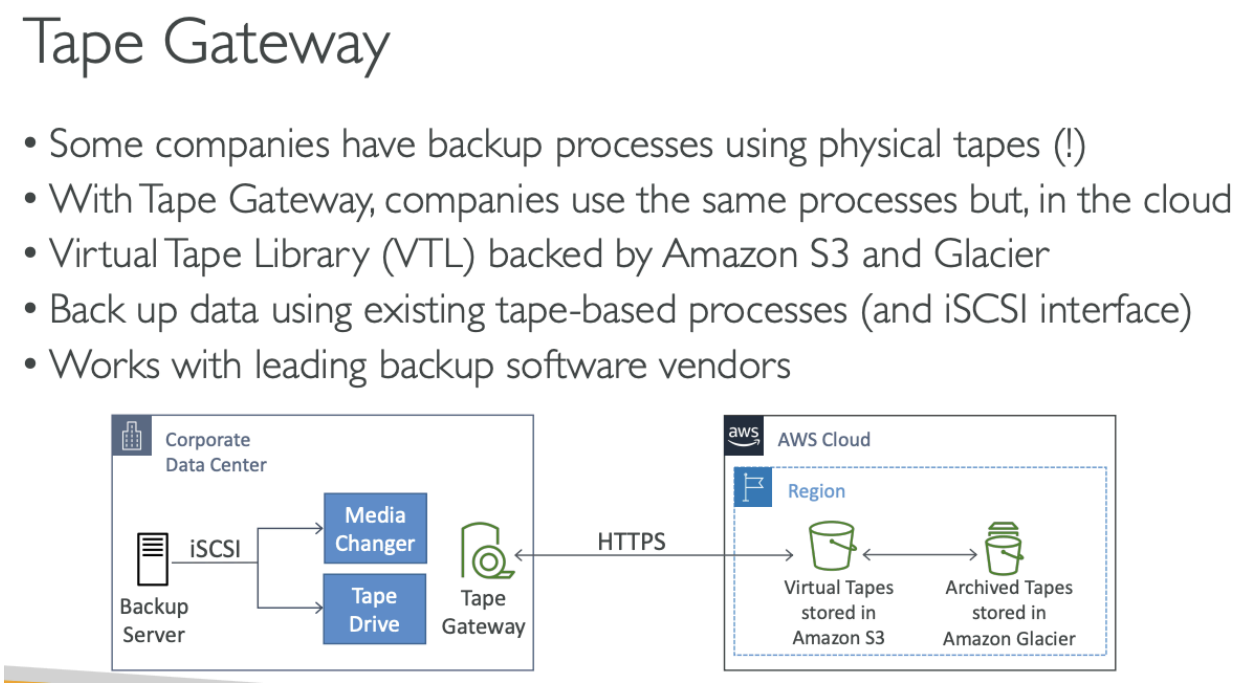


- Native access to Amazon FSx for Windows File Server
    - Windows native compatibility (SMB, NTFS, Active Directory…)
- Local cache for frequently accessed data
- Useful for group file shares and home directories

Volume Gateway

- Block storage using iSCSI protocol backed by S3
- Backed by EBS snapshots to help restore on premise volumes
- Cached Volumes: low latency access to most recent data
- Stored volumes: entire dataset is on premise, scheduled backups to S3

Tape Gateway

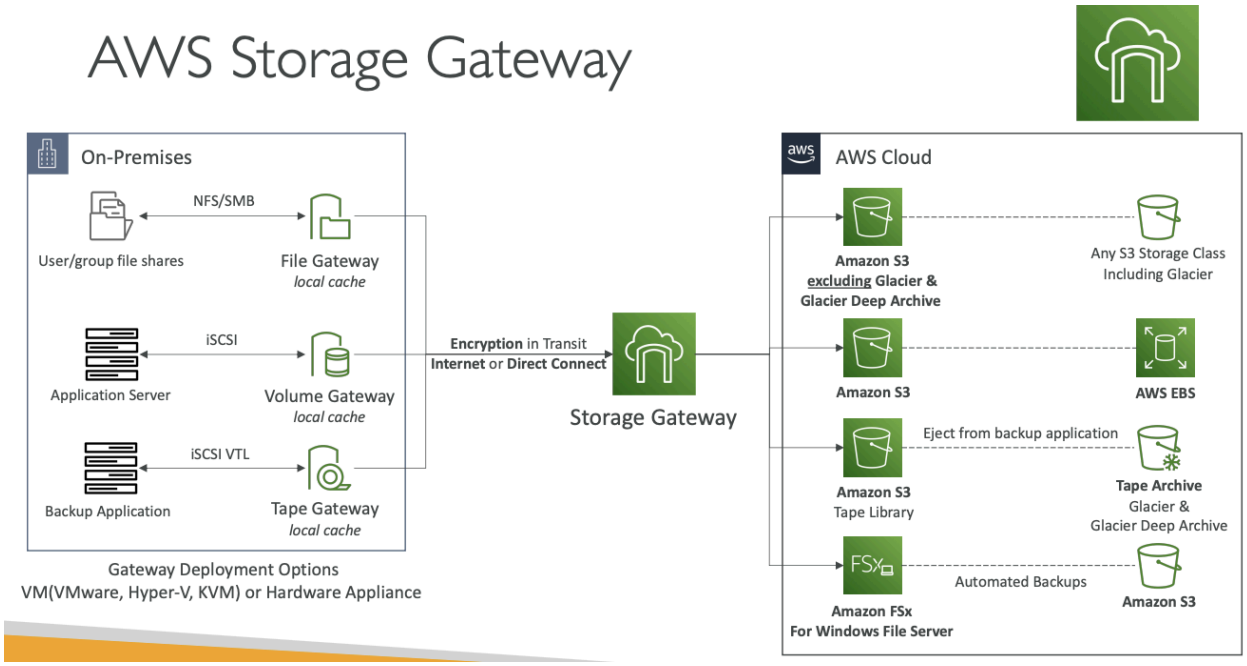


- Backup processes using physical tapes
- Virtual Tape Library (VTL) backed by S3 and Glacier
- Back up data using existing tape based processes
- Works with leading backup software vendors

Storage Gateway – Hardware Appliance

- All gateways require on premise data server for virtualization, but if it isn't there by default, use Storage Gateway Hardware Appliance
    - Works with all Gateways and has required resources already
    - Helps for daily NFS backups in small data centers

Storage Gateway Summary

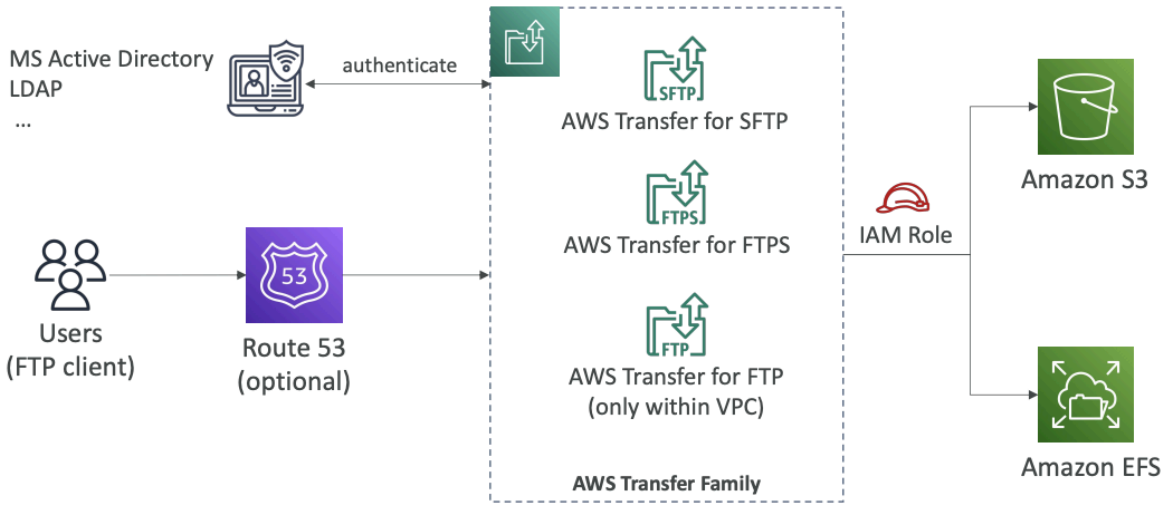


AWS Transfer Family

- Fully managed service for file transfer in / out of S3 or EFS via FTP protocol
- Supports:
    - AWS Transfer for FTP (File transfer protocol, unencrypted)

    - AWS Transfer for FTPS (encrypted)
    - AWS Transfer for SFTP (secure FTP, encrypted in flight)
- Managed infrastructure, scalable, reliable, high availability (multi AZ)
- Pay per provisioned endpoint per hour + data transfers in GB
- Store and manage users' credentials within the service
- Integration with existing authentication systems
- Usage: sharing files, public datasets…

AWS Transfer Family



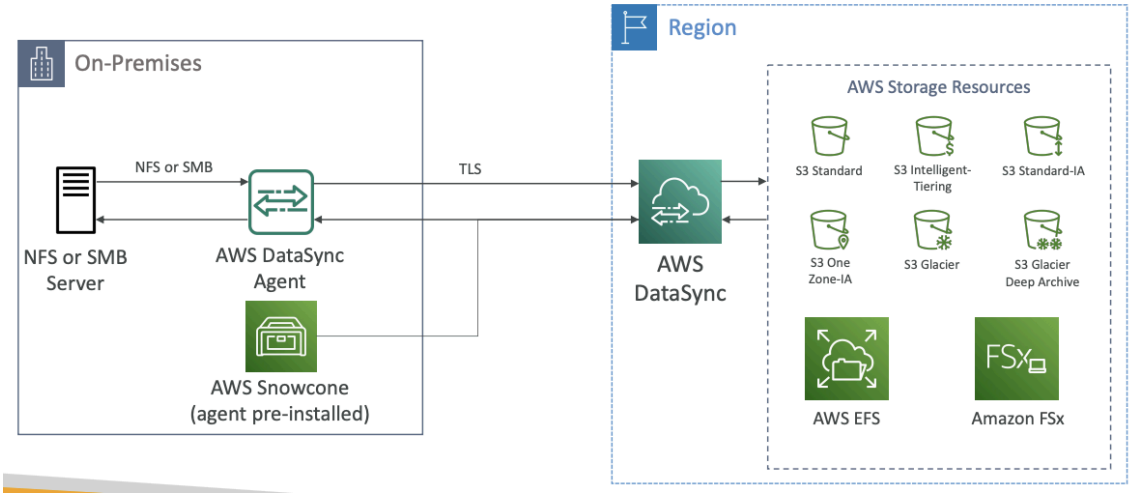
AWS DataSync

AWS DataSync

- Move large amount of data to and from
    - On-premises / other cloud to AWS (NFS, SMB, HDFS, S3 API…) – needs agent
    - AWS to AWS (different storage services) – no agent needed
- Can synchronize to:
    - Amazon S3 (any storage classes – including Glacier)
    - Amazon EFS
    - Amazon FSx (Windows, Lustre, NetApp, OpenZFS...)
- Replication tasks can be scheduled hourly, daily, weekly
- File permissions and metadata are preserved (NFS POSIX, SMB…)
- One agent task can use 10 Gbps, can setup a bandwidth limit

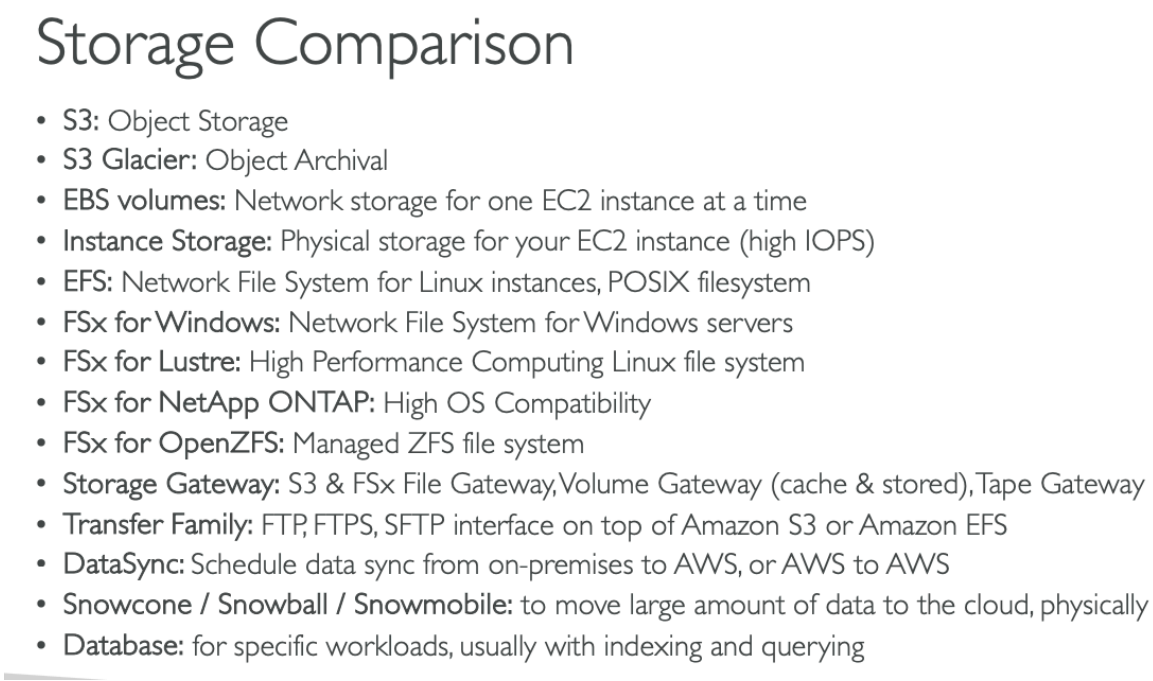
- Move large amount of data to and from on prem or other cloud to AWS (needs agent), or AWS to AWS (no agent needed)
- Can synchronize to:
    - S3 (any tier)
    - EFS
    - FSx (all)
- Replication tasks scheduled hourly, daily, weekly
- File permissions and metadata are preserved
- One agent task can use 10 Gbps, can set bandwidth limit

AWS DataSync
NFS / SMB to AWS (S3, EFS, FSx…)



AWS DataSync
Transfer between AWS storage services

All AWS Storage Options Compared



Section 17: Decoupling applications: SQS, SNS, Kinesis, Active MQ

Intro to Messaging

1. Synchronous communication (application to application)
    a. Synchronous between apps if there are sudden spikes in traffic
    b. Decoupling helps to scale independently
2. Asynchronous communication / event based (application → queue → application)

SQS

- A queue has producers that send messages and a consumer that polls messages
- Decouples between producers and consumers

Standard Queue

- Oldest offered, fully managed service for decoupling apps
- Attributes:
    - Unlimited throughput, unlimited number of messages in queue
    - Default retention of messages 4 days, max 14 days
    - Low latency (<10ms on publish and receive)
    - Limitation of 256 KB per message sent
- Can have duplicate messages (at least once delivery)

- Can have out of order messages (best effort ordering)

Producers

- Via SendMessageAPI in SDK
- Message is persisted in SQS until a consumer deletes
    - Retention of 4 default days, max 14

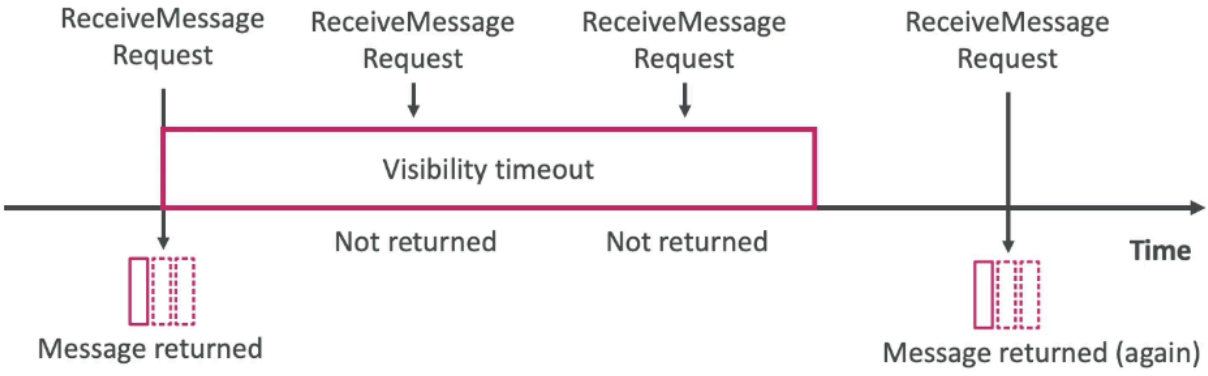
Consumers

- Consumers on EC2, Lambda, etc…
- Poll SQS for messages (up to 10 at a time) and process messages
- Delete messages via DeleteMessageAPI

Multiple EC2 Instance Consumers

- Consumers receive and process messages in parallel
- At least once delivery
- Best effort message ordering
- Consumers delete messages after processing them
- Scale consumers horizontally to improve processing

Message Visibility Timeout



- After a message is polled by a consumer, it becomes invisible to other consumers

    - Default timeout of 30 seconds, which means the message must be processed within 30 seconds.
    - After the timeout ends, the message is “visible” and back on queue in SQS
- If a message is not processed within timeout, it will be processed twice
- A consumer could call ChangeMessageVisibility API to get more time if it needs longer to process
    - If timeout is high (hours) and consumers crashes, reprocessing takes time
    - If timeout is too low (seconds), we may get duplicate processing

FIFO Queues

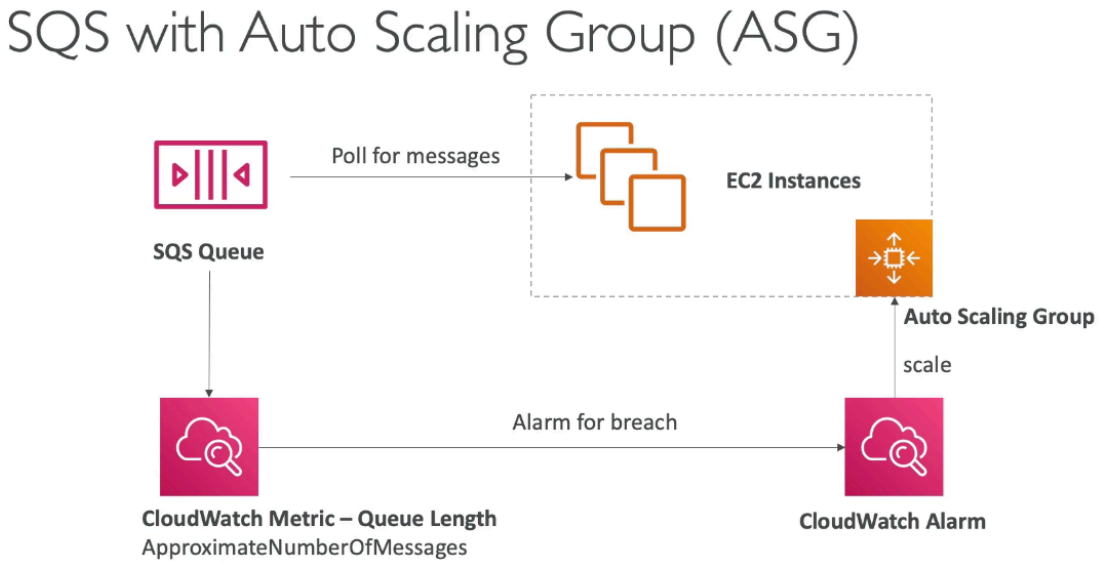
- First in first out (ordering of messages in the queue)
- Limited throughput: 300 msg/s without batching, 3000 msg/s with
- Exactly once send capability (by removing duplicates)
- Messages processed in order by the consumer
- Must have .fifo at the end

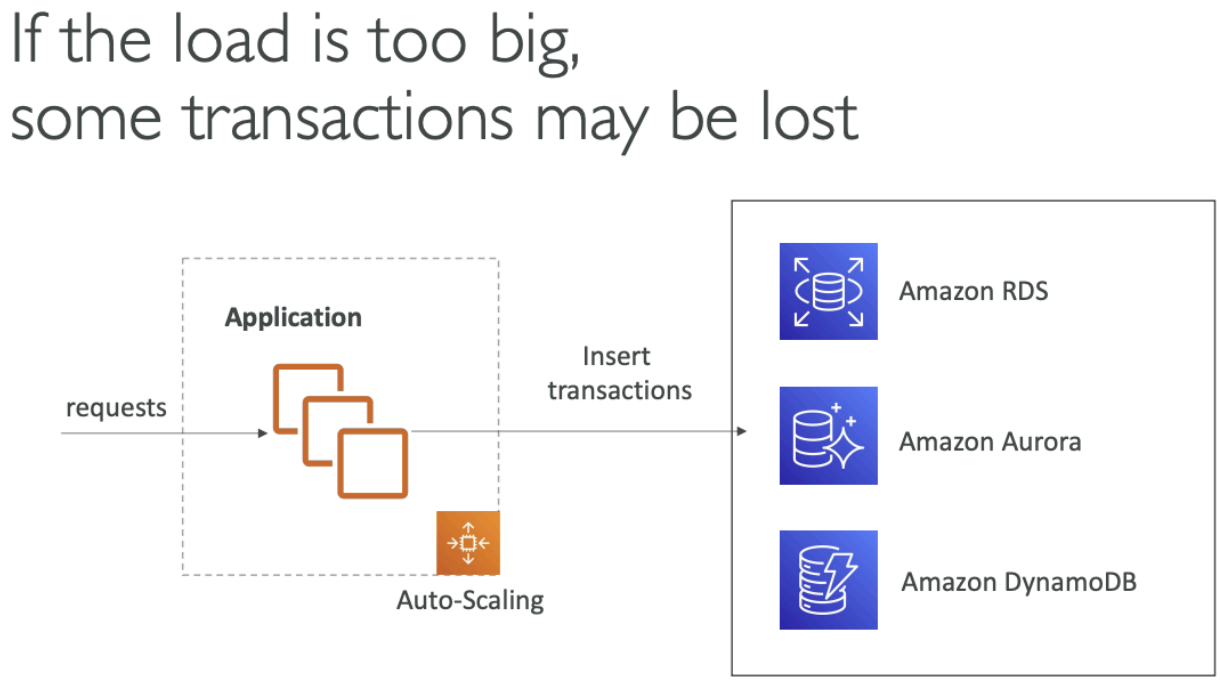
Long Polling

- When consumer requests message, it can “wait” for messages to arrive if queue empty, called long polling
- Done to decrease # API calls made to SQS while increasing efficiency and decreasing latency of application
    - Wait of 1 to 20 sec
    - Preferable to short polling (short polling is 0 seconds)
- Enabled at queue level or at API queue level using ReceiveMessageWaitTimeSeconds

SQS w/ ASG

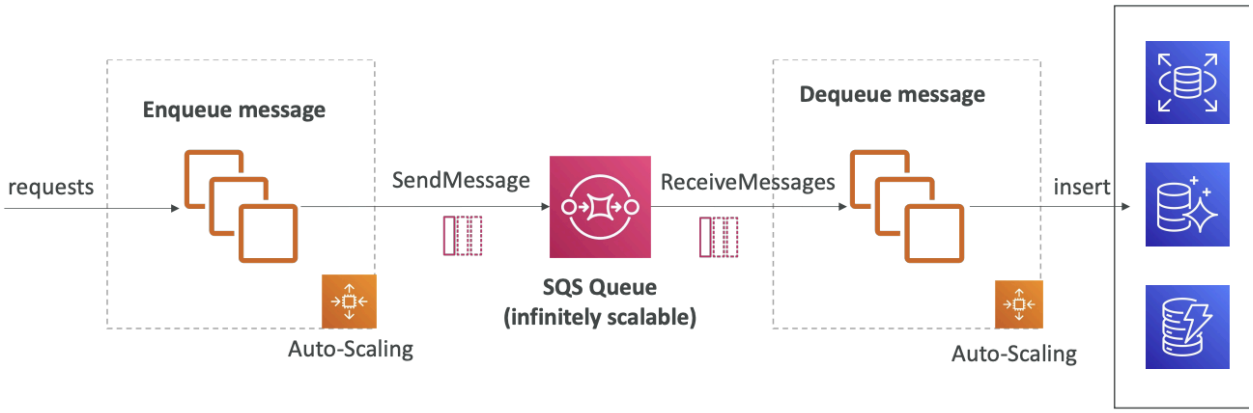
- Queue length CloudWatch metric alarm to increase capacity of ASG



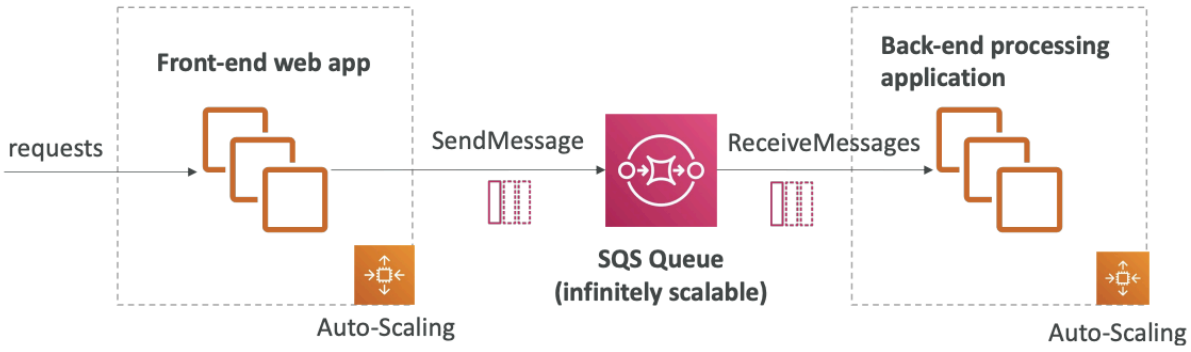


- DBs may be overloaded if too many requests come through. Resolved with SQS as buffer to DB writes:

SQS as a buffer to database writes



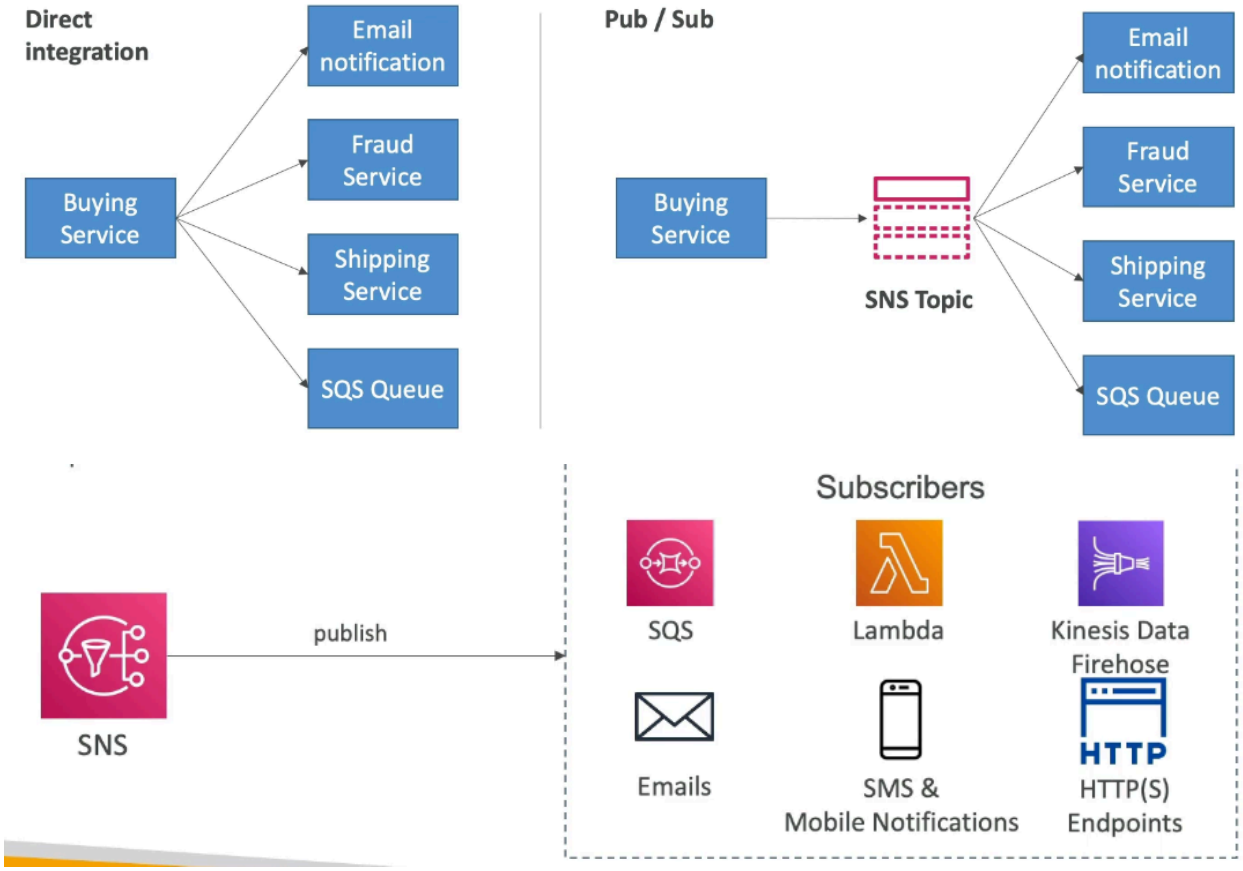
SQS to decouple between application tiers



SNS

Amazon SNS

- What if you want to send one message to many receivers?



- Pub / sub of a middle man between publisher and subscribers
- “Event producer” only sends message to 1 SNS topic and “event receiver” (subscriber) will listen to SNS topic notifications
    - Each sub to the topic will get the messages and many subs per topic allowed
- AWS services can send data directly to SNS for notifications

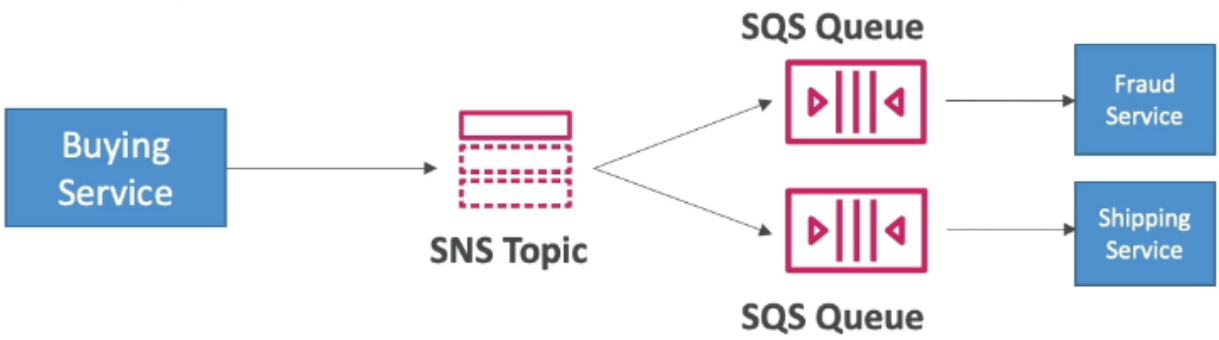
How to publish

- Topic Publish (SDK)
    - Create topic, subscription, publish
- Direct Publish (mobile apps SDK)
    - Create a platform application, platform endpoint publish

SNS Security

- Encryption:
    - In flight via HTTPS
    - At rest via KMS
    - Client side
- Access Controls: IAM policies
- SNS Access Policies (similar to S3 bucket policies)
    - Cross account, allow other services to write to SNS topic

SNS + SQS Fan Out Pattern

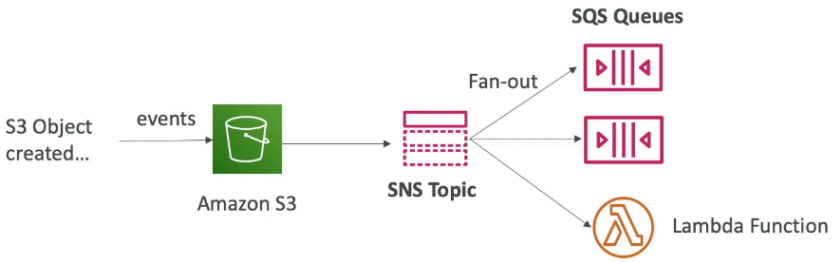


- Push once in SNS, receive in all SQS queues that are subscribers
- Fully decoupled, no data loss
    - SQS allows for: data persistence, delayed processing and retries
- Ability to add more SQS subscribers over time
- Must have SQS queue access policy allow for SNS to write
    - Cross region delivery for SQS and SNS different regions

S3 Events to multiple queues

Application: S3 Events to multiple queues

- For the same combination of: event type (e.g. object create) and prefix (e.g. images/) you can only have one S3 Event rule
- If you want to send the same S3 event to many SQS queues, use fan-out



SNS to S3 via Kinesis

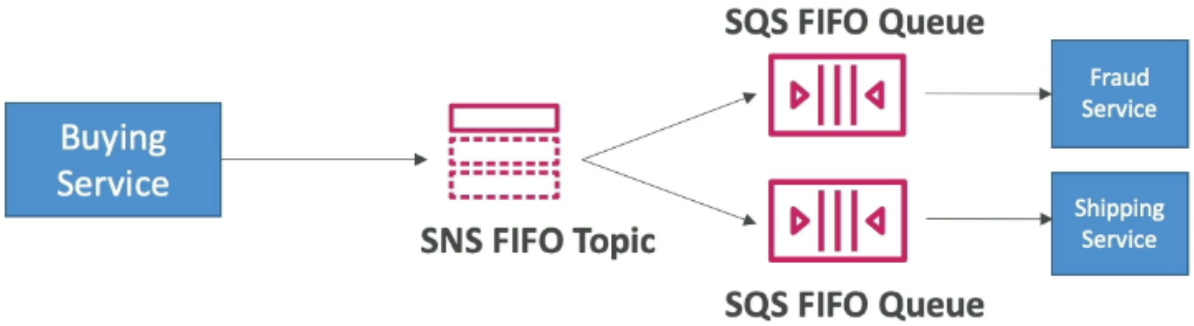
Application: SNS to Amazon S3 through Kinesis Data Firehose

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

SNS – FIFO Topic

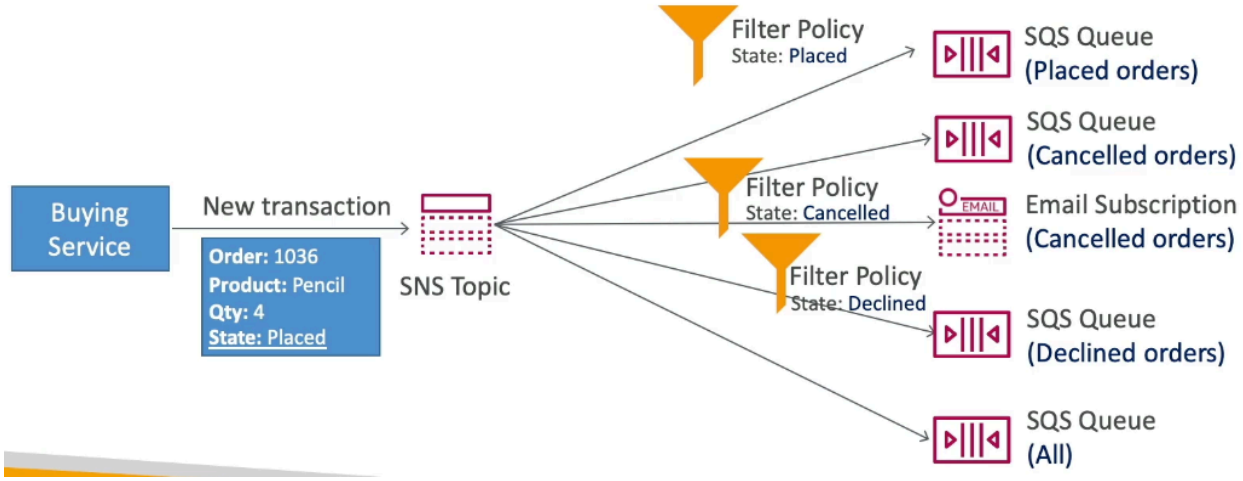
- Similar as SQS FIFO
  - Ordering by message group ID
  - Deduplication via ID or Content based
- Can ONLY have SQS Standard and FIFO queues as subscribers

SNS FIFO + SQS FIFO: Fan Out



- Fan out + ordering + deduplication

Message Filtering

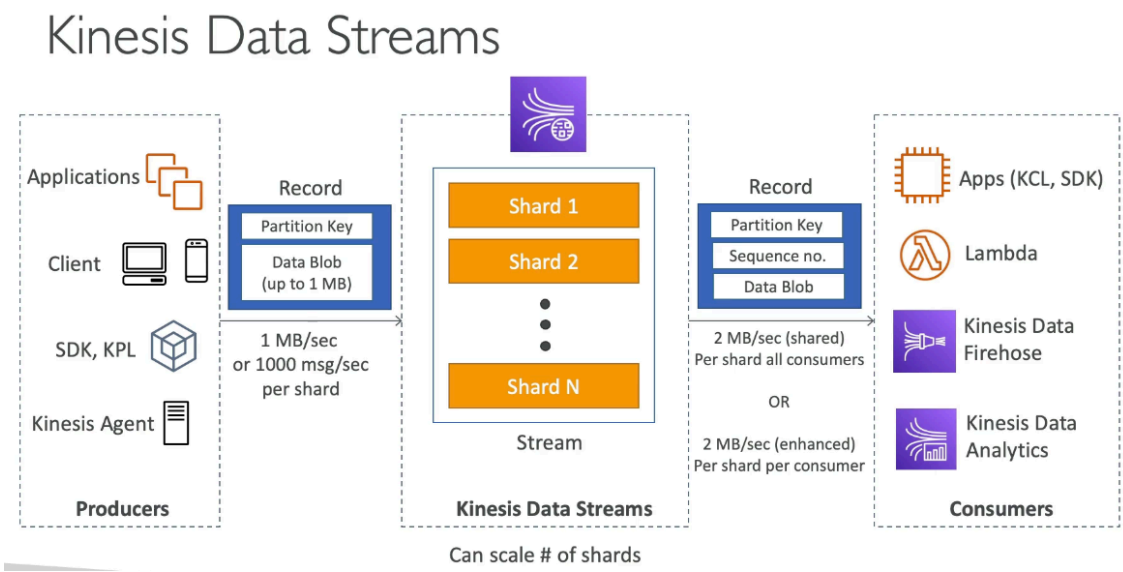


- JSON policy to filter messages sent to SNS topic's subscriptions

Kinesis

- Makes it easy to collect, process, and analyze streaming data in real time
    - Application logs, metrics…
- 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

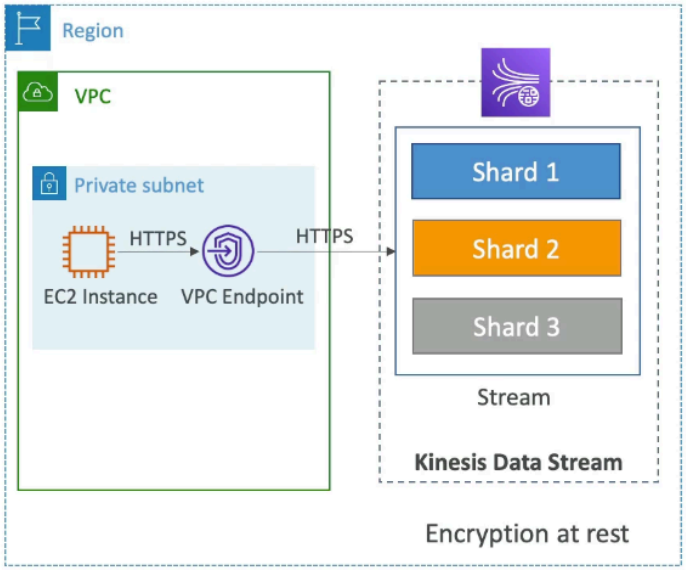
Kinesis Data Streams



- Made of shards, provisioned ahead of time. Data is split across all shards. Producers send records (made of partition key + data blob).
- Retention between 1 and 365 days with ability to reprocess data
- Once data in Kinesis, cannot be deleted (immutability) and data that shared the same partition key goes in the same shard (ordering) → key based ordering

Capacity Modes

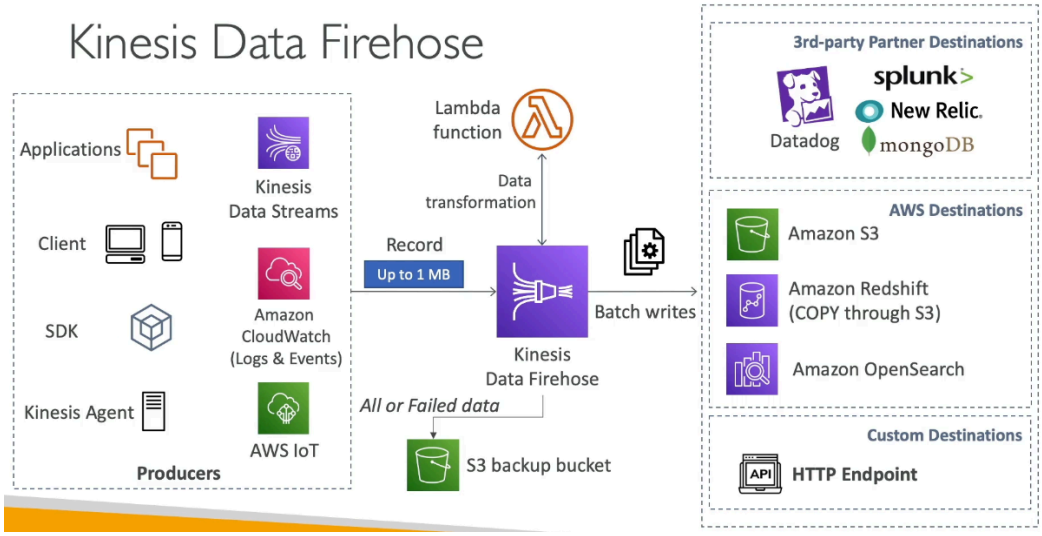
- Provisioned Mode:
  - Choose # shards provisioned, each shard gets 1 MB/s in and 2 MB/s out
  - Pay per shard per hour
- On demand:
  - No provisioning or capacity management
  - Default capacity provisioned (4 MB/s)
  - Scales automatically based on throughput peak in last 30 days
  - Pay per stream per hour & data in/out per GB



Security

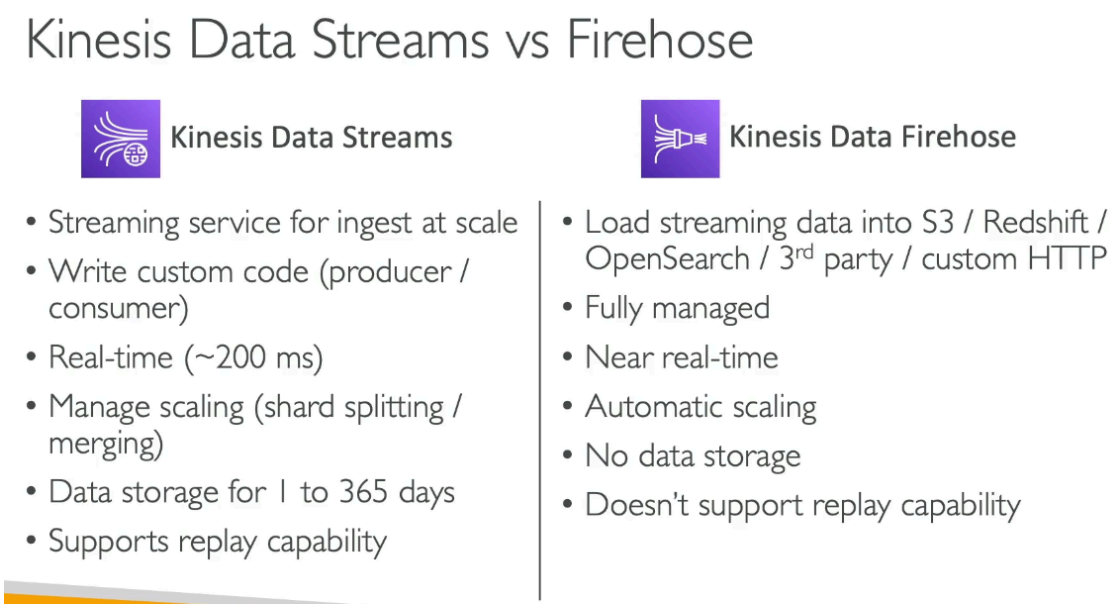
- Control access via IAM policies, in flight, at rest, client side encryption
- VPC endpoints available for kinesis to access within VPC
- Can be monitored via CloudTrail

Kinesis Data Firehose



- Takes data from sources (Kinesis Data Streams) and send in batches to S3, Redshift (COPY through S3), Amazon OpenSearch
    - There is a buffer size and buffer interval. If the size isn't filled by the time the interval ends, the buffer is automatically flushed.
- Fully managed service (serverless)
    - Pay for data going through Firehost
    - Near real time
        - Buffering interval from 0 to 900 seconds and buffer size minimum of 1 MB
- Supports many data formats, conversions, transformations, conversions

Kinesis Data Streams vs Firehose



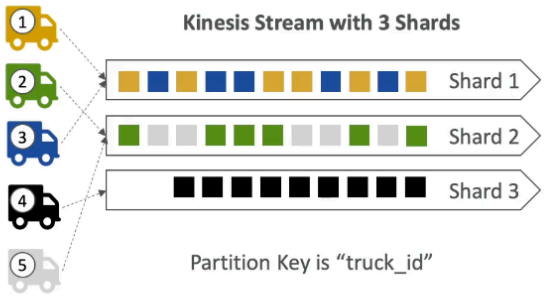
Data Ordering into Kinesis vs SQS FIFO

Ordering with Kinesis

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?

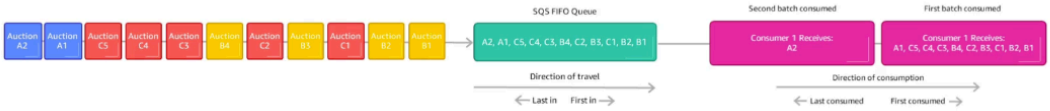
- Answer: send using a "Partition Key" value of the "truck_id"
- The same key will always go to the same shard



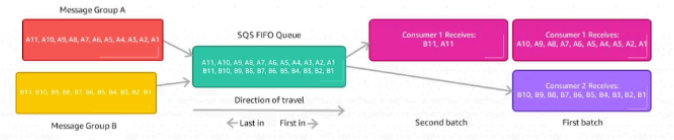
Ordering with SQS

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 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

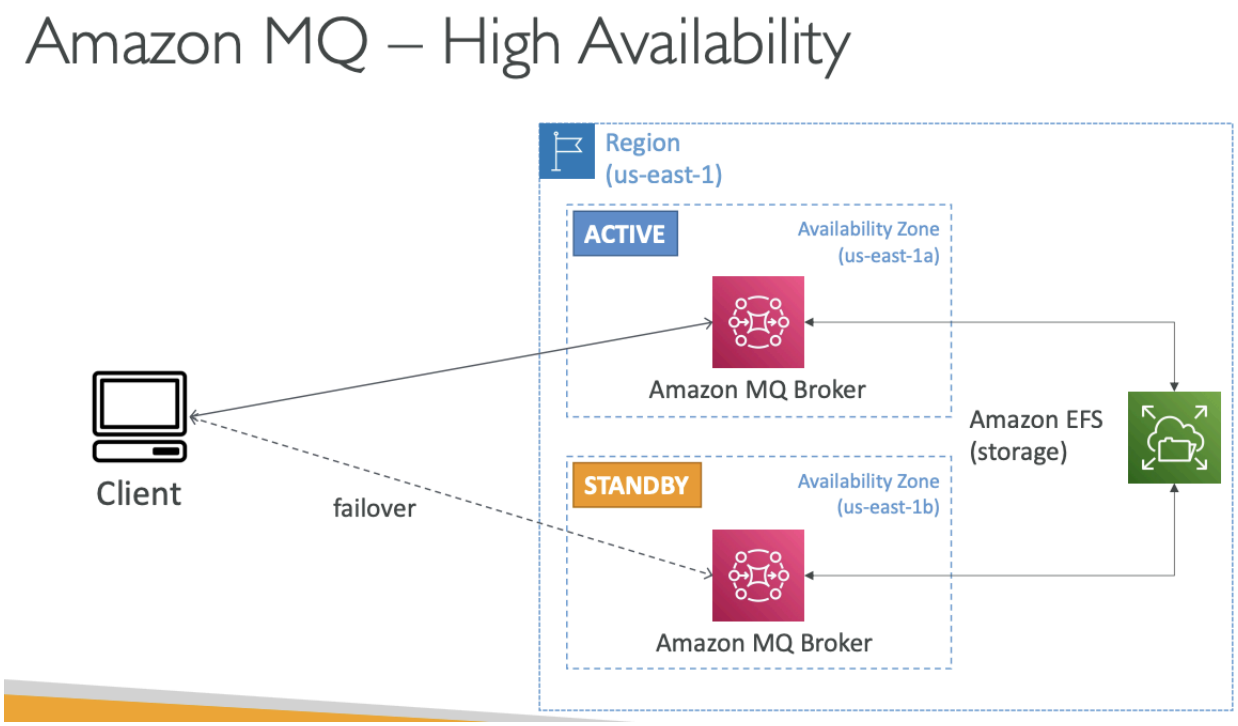
SNS:

- 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



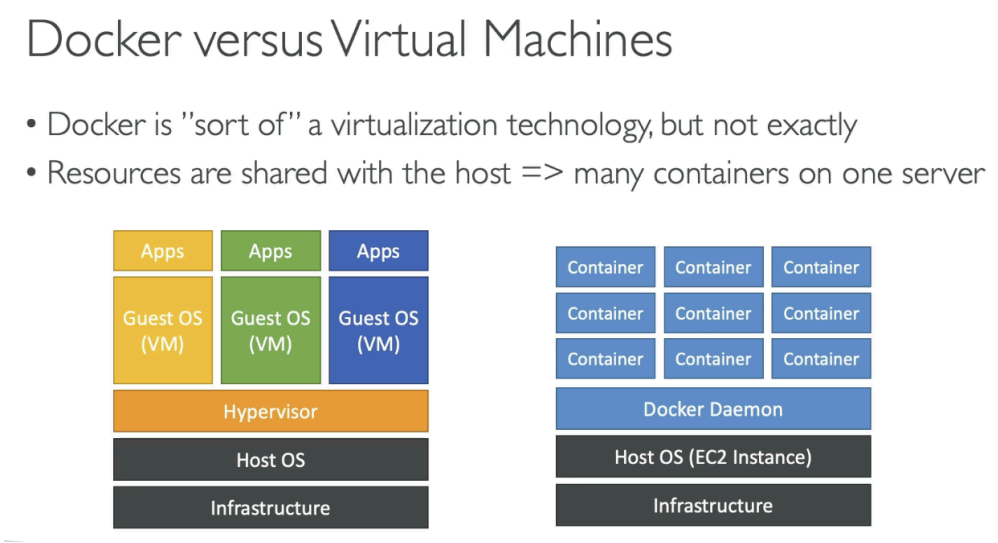
- SQS and SNS are AWS specific, but if you already use an open protocol for on premise, when migrating to the cloud use MQ instead of updating application
- Managed message broker service for RabbitMQ and ActiveMQ
    - Doesn't scale as much as SQS / SNS and runs on servers and can run in multi AZ with failover
    - Has both queue feature (SQS) and topic feature (SNS)

Section 18: Containers on AWs: ECS, Fargate, ECR & EKS

Docker Introduction

- Software development to deploy apps
- Apps are packaged in containers to run on any OS (predictable behavior)
- Docker images are stored in Docker repos @ Docker Hub or Amazon ECR

Docker vs VM



- Docker is “sort of” a virtualization tech, but not exactly
- Resources shared with the host → many containers on 1 server

Docker Container Management on AWS

- ECS, EKS, Fargate, ECR

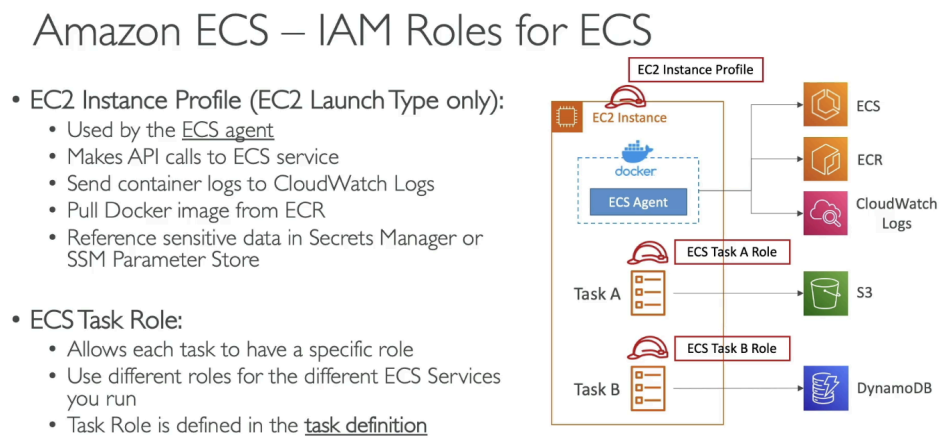
Amazon ECS - EC2 Launch Type

- Launch Docker containers on AWS = launch ECS tasks on ECS clusters
- EC2 launch type: must provision & maintain the infrastructure (EC2 instances)
    - Each EC2 instance must run the ECS Agent to register in the ECS Cluster
    - AWS takes care of starting / stopping containers

Fargate Launch Type

- Launch Docker containers on AWS
- Do not provision infrastructure (serverless)
- Create task definitions, AWS just runs ECS tasks based on CPU / RAM needed - scaling to increase number of tasks

IAM Roles for ECS



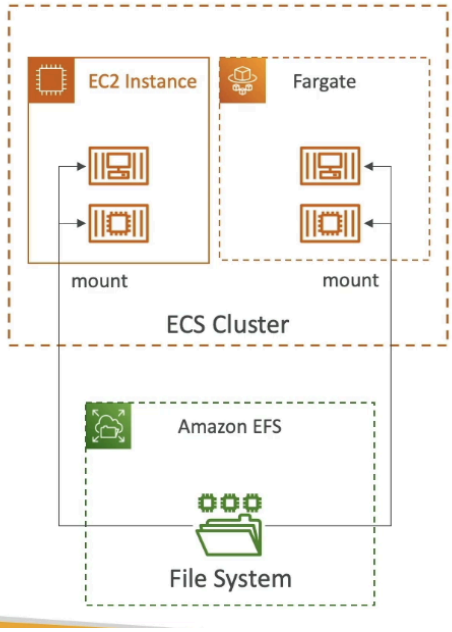
- EC2 instance profile (EC2 launch type only)
    - Used by ECS agent and makes API calls to ECS service
        - Can send container logs to CloudWatch, pull Docker image from ECR, reference data in Secrets Manager
- ECS Task Role
    - Allows each task to have a specific role
    - Use different roles for different ECS services you run
    - Task role defined in task definition

Load Balancer Integrations

- ALB supported and works for most cases
- NLB recommended for high throughput / high performance or with AWS Private Link
- Classic LB not recommended

ECS Data Volumes (EFS)

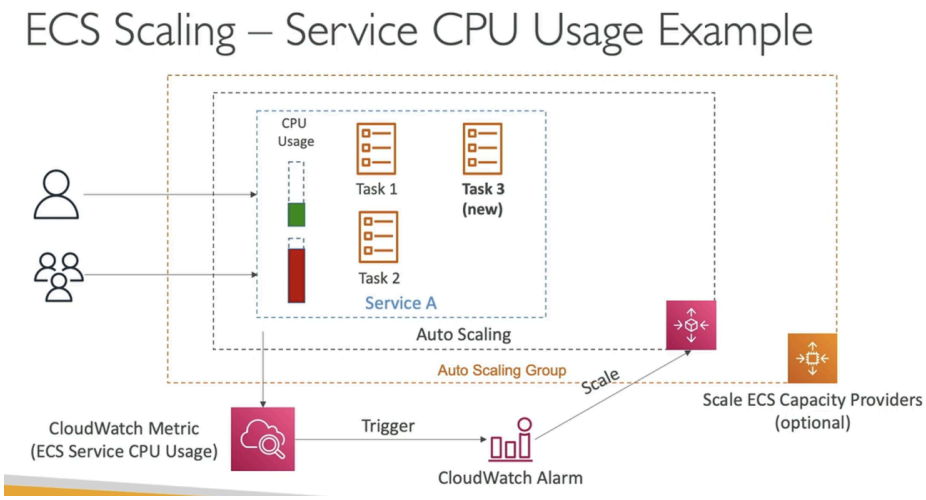
- Mount EFS file system onto ECS tasks
- Works for both EC2 and Fargate
- Tasks running in any AZ will share same data in EFS
- Fargate + EFS = serverless
- Use Case: persistent multi-AZ shared storage for containers
- S3 cannot be mounted as a file system



ECS Service Auto Scaling

- Automatically increase / decrease ECS tasks via Application Auto Scaling
    - Average CPU, memory utilization, or ALB request count
- Target Tracking: scale based on specific CloudWatch metric
- Step scaling: scale based on specific CloudWatch alarm
- Scheduled Scaling: scale based on time/date
- ECS Service Auto Scaling (task level) != EC2 Auto Scaling (EC2 instance level)
- Fargate Auto scaling easier to set up

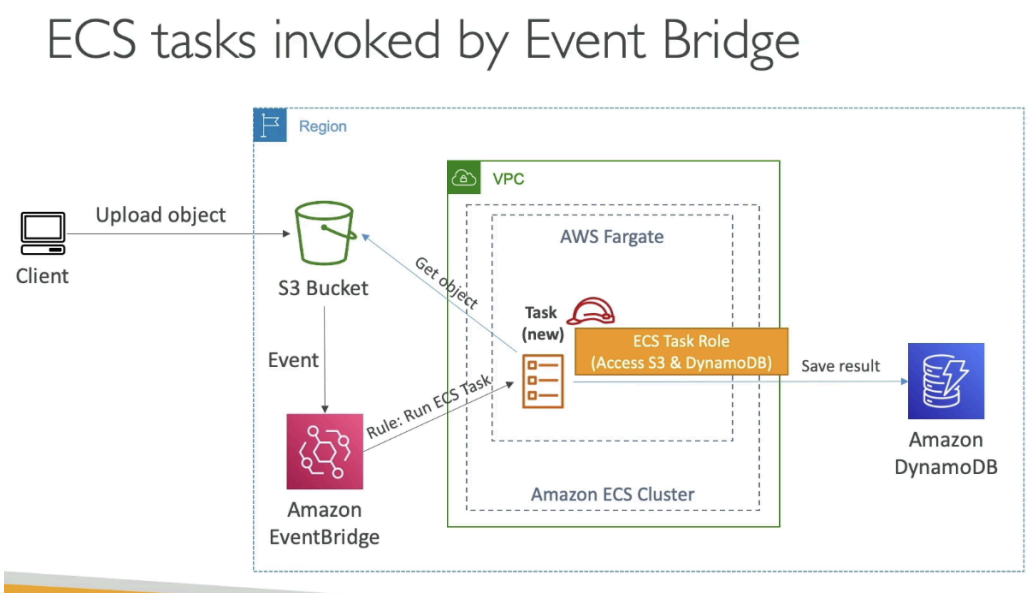
Auto Scaling EC2 Instances for EC2 Launch Type



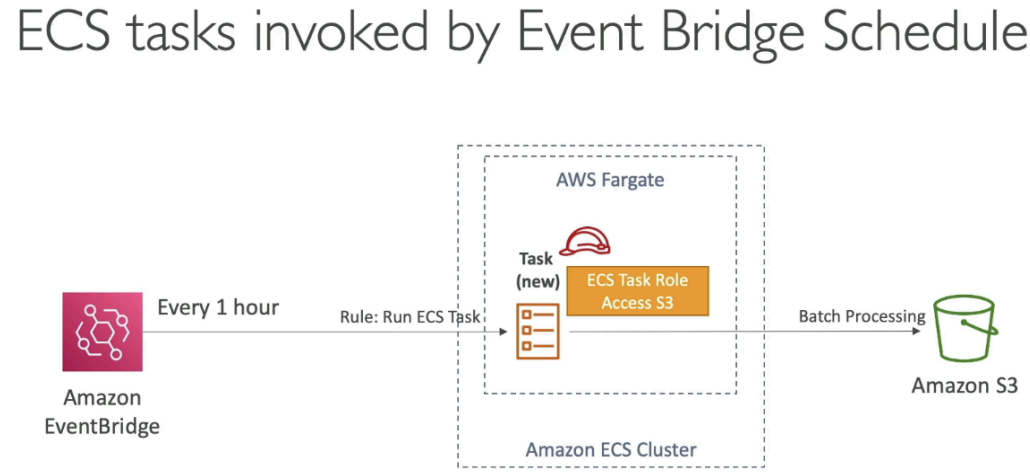
- Auto Scaling Group Scaling
    - Scale ASG based on utilization
- ECS Cluster Capacity Provider
    - Used to automatically provision and scale infrastructure for ECS tasks
    - Paired with ASG to add EC2 instances when missing capacity

ECS Solutions Architecture

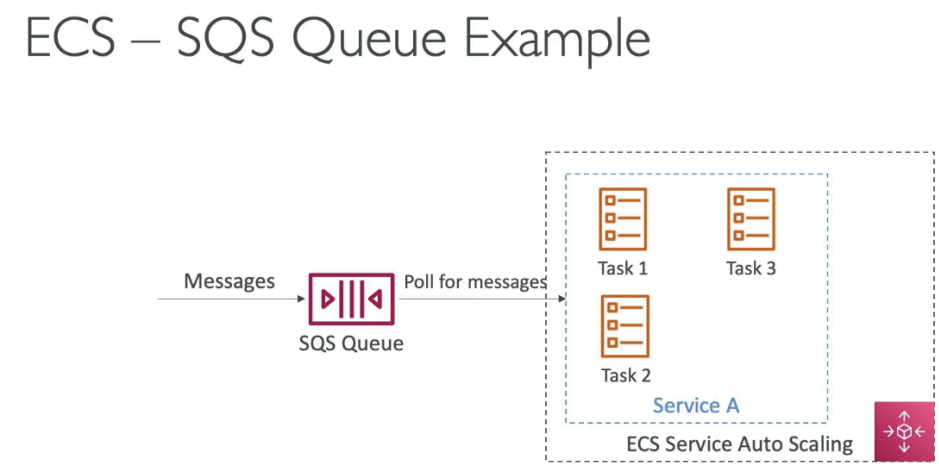
ECS Tasks invoked by Event Bridge



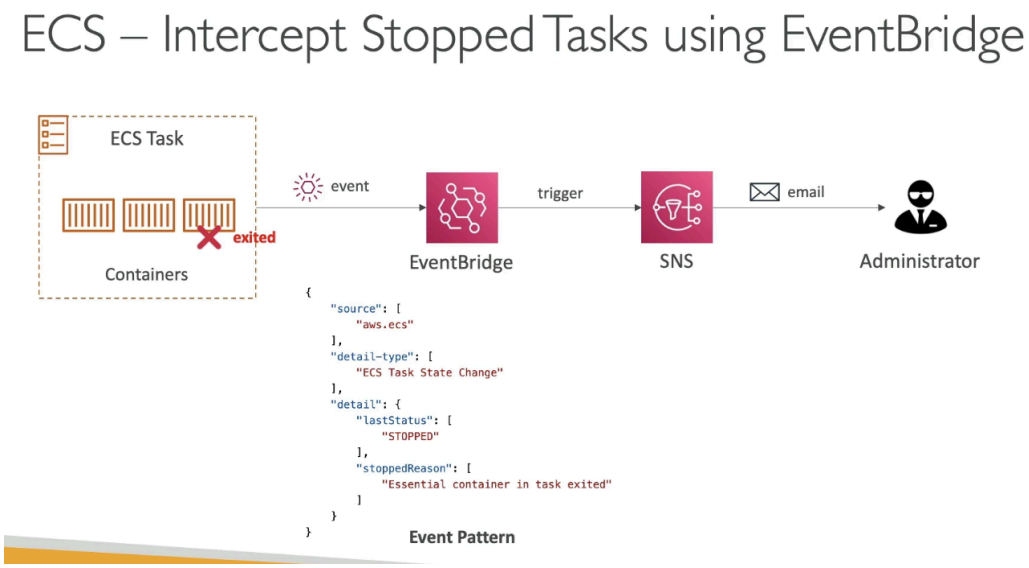
ECS Tasks Invoked by Event Bridge Schedule



ECS- SQS Queue Example



ECS – Intercept Stopped Tasks using EventBridge



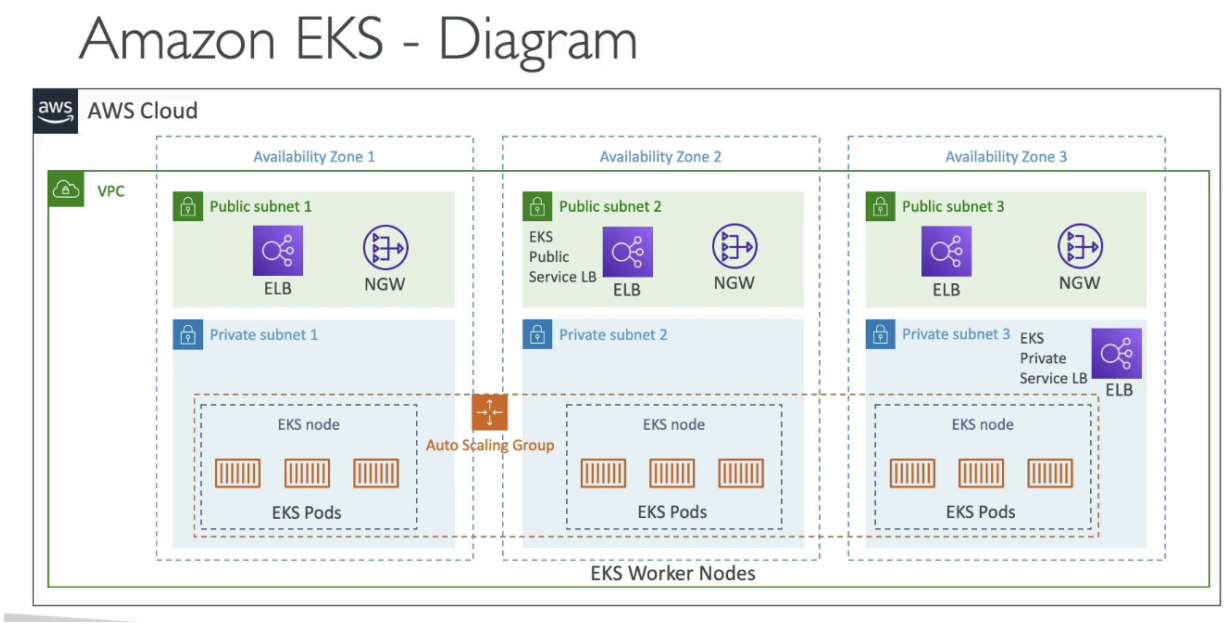
ECR

- Store and manage Docker images (public or private) on AWS
- Fully integrated with ECS, backed by S3
    - IAM Role to EC2 instance to pull images
- Supports image vulnerability scanning, versioning, image tags, image lifecycle

Amazon EKS

- Launches managed Kubernetes clusters on AWS

    - K8 is for automatic deployment, scaling and management of containerized (Docker) application
    - Alternative to ECS
    - Cloud-agnostic: can use in any cloud
- EKS supports EC2 if you want to deploy worker nodes or Fargate for serverless containers



EKS Node Types

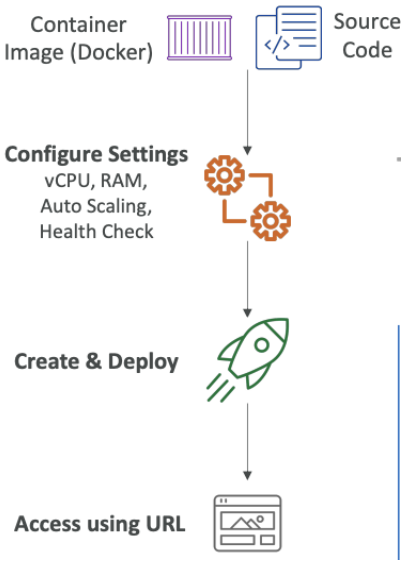
- Managed Node Groups
    - Creates and manages Nodes (EC2 instances) for you
    - Part of ASG managed by EKS
    - Supports on demand or spot instances
- Self Managed Nodes
    - Nodes created by you and registered to EKS cluster and managed by ASG
    - Can use prebuilt AMI – EKS Optimized AMI
    - Supports on demand or spot instances
- AWS Fargate
    - No maintenance, no managed nodes

EKS – Data Volumes

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

AWS App Runner

- Fully managed service to deploy web applications and APIs at scale
    - No infrastructure required
    - Start with source code or container image and automatically builds and deploys web app
- Automatic scaling, highly available, LB, encryption, VPC access support
- Connect to DB, cache, queue
- Use case: web apps, APIs, microservices, rapid production deployments



Section 19: Serverless Overviews from Solution Architect Perspective

Serverless Intro

- No server management / provisionment
- Lambda, DynamoDB, Cognito, API GW, S3, SNS / SQS, Kinesis Data Firehose, Aurora Serverless, Step Functions, Fargate

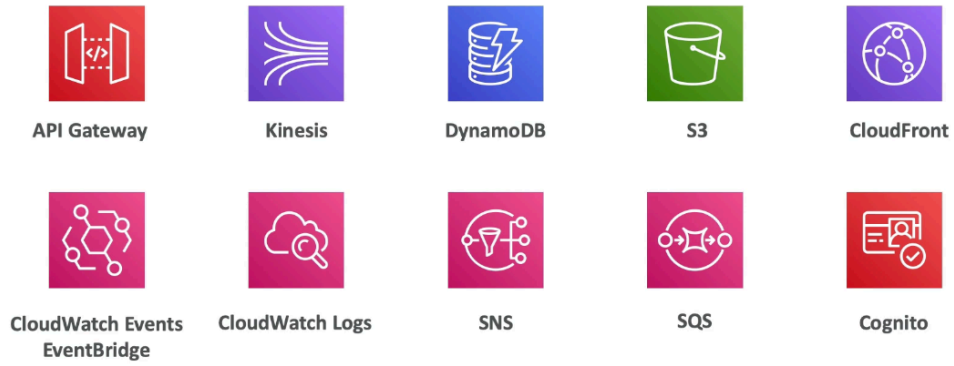
Lambda Overview

- Virtual functions, limited by time (short executions), running on demand, automated scaling
- Max timeout of 15 minutes

Benefits:

- Easy pricing based on pay per request and compute time + generous free tier
- Integration with AWS services, programming languages, monitoring
    - Custom runtime API for any language, community supported and Lambda container image
- Increasing RAM, increases CPU and network functionality

AWS Lambda Integrations
Main ones

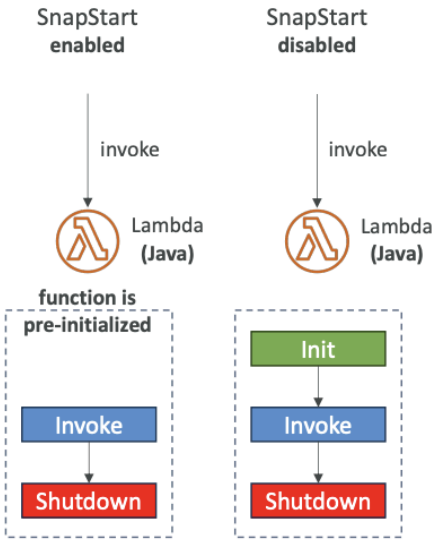


Lambda Limits

- Execution
  - Memory allocation: 128 MB to 10 GB
  - More RAM, more vCPU
  - Max execution time: 15 min, 900 seconds
  - 4KB for env vars
  - /temp space for 512 MB to 10 GB
  - 1000 concurrency executions
- Deployments
  - Zip size: 50 MB
  - Uncompressed deployment (code + dependencies): 250 MB
  - /temp to load files at startup
  - 4 KB for env vars

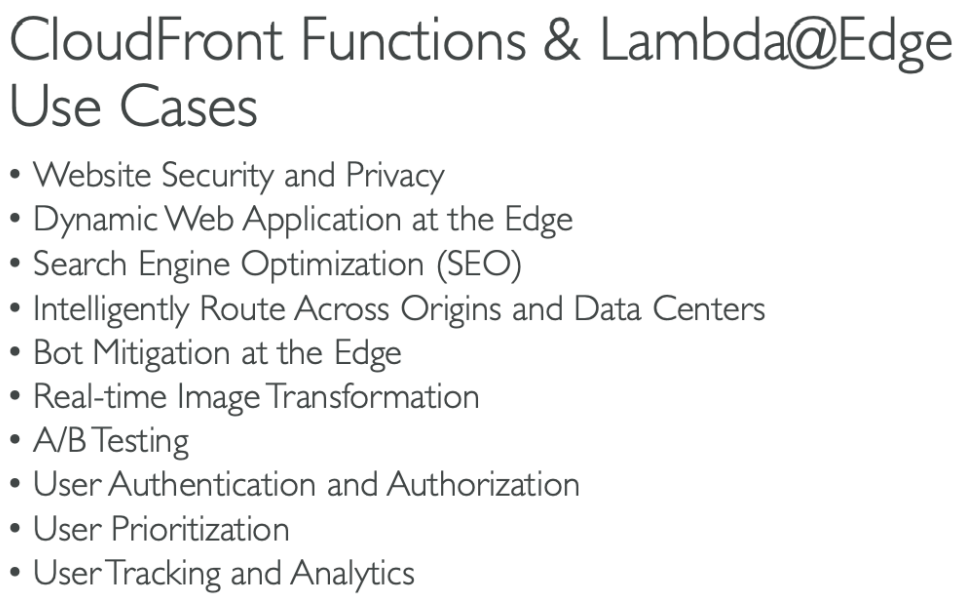
Lambda SnapStart

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



Lambda Invocation
Lifecycle Phases

Customization @ Edge



- 2 types CloudFront Functions & Lambda @ Edge
- Edge Functions:
    - Code to attach to CloudFront distributions; runs close to users to minimize latency
    - Serverless, deployed globally, pay for what you use
    - Customize CDN content
- Use cases:
    - Security + privacy, dynamic web app at the edge, SEO, AB testing, etc…
- CloudFront Functions:
    - Lightweight functions in JS to modify viewer request/response
        - Viewer Request: after CF receives a request from viewer
        - Viewer Response: before CF forwards response to viewer
    - High scale latency sensitive CDN customizations (millions request/second)
        - High performance, high scale
    - Native feature of CloudFront (managed within CF)
- Lambda @ Edge
    - Node or Python, scales to 1000s request / second
    - Change CF requests and responses:
        - Viewer request: after CF receives a request from a viewer
        - Origin request: before CF forwards response from origin
        - Origin response: after CF receives response from origin
        - Viewer response: before CF forwards response to viewer
    - Author functions in 1 AWS region (us east 1), then CF replicates to other locations

CloudFront Functions vs Lambda @ Edge

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, 1/6th price of @Edge	No free tier, charged per request & duration

- Major differences: high scale for CF functions, Lambda @ Edge triggers from both viewer and origin, max execution time lower for CF functions,

Uses Cases

CloudFront Functions vs. Lambda@Edge - Use Cases

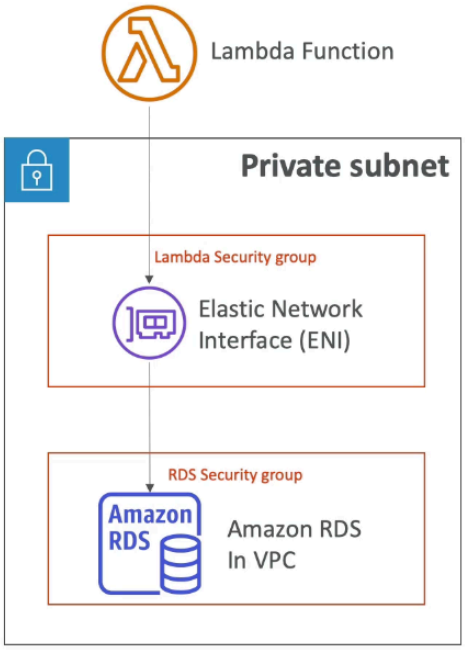
CloudFront Functions

- 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

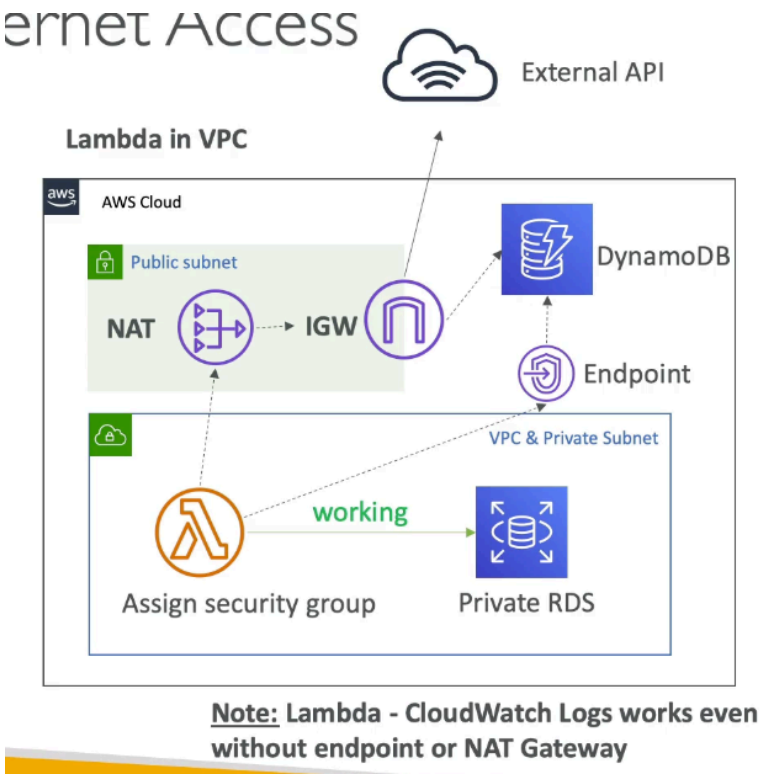
Lambda in VPC



- By default lambda functions are launched outside VPC and cannot access resources in VPC
- Must define VPC ID, subnets and SG. Lambda will create an ENI (Elastic Network interface) in subnets via AWSLambdaVPCAccessExecutionRole

Internet Access

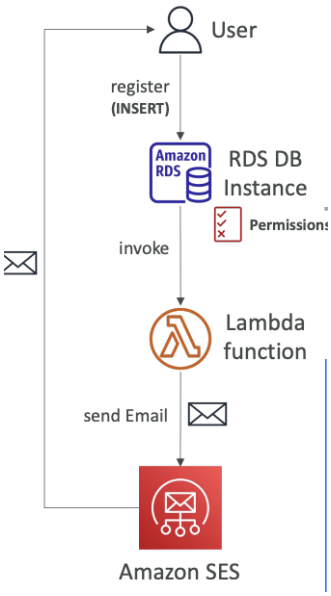
- Lambda function in VPC has no internet access by default
    - Deploying a lambda function in a public subnet does not give internet access or public IP
- Must deploy lambda function in private subnet and give it NAT Gateway / Instance access
- VPC endpoints are used to privately access AWS services without a NAT



Note: Lambda - CloudWatch Logs works even without endpoint or NAT Gateway

Lambda with RDS Proxy

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

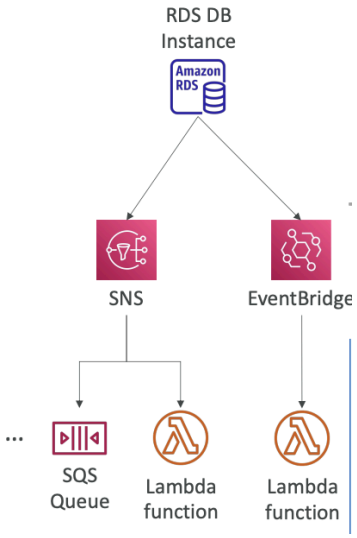


Invoking Lambda from RDS & Aurora

- Invoke lambda functions from within DB instance to allow processing data events from DB
    - Supported for RDS for PostgresQL and Aurora MySQL
- Must allow outbound traffic to lambda function from DB instance
- DB instance must have permissions to invoke lambda function

RDS Event Notifications

- Notifications that tell info about DB instance
    - No info about data itself
    - Info: DB instance, DB snapshot, parameter group, SG, RDS proxy, custom engine version
    - Near real time (up to 5 min)
- Send notifications to SNS or EventBridge



DynamoDB Overview

- Traditional DB use RDBMS DB with SQL, but have strong requirements on how data is structured and vertical/horizontal scaling methods
- NoSQL are non-relational and distributed with no query joints, meaning all data needs to be in 1 row
    - Scales horizontally

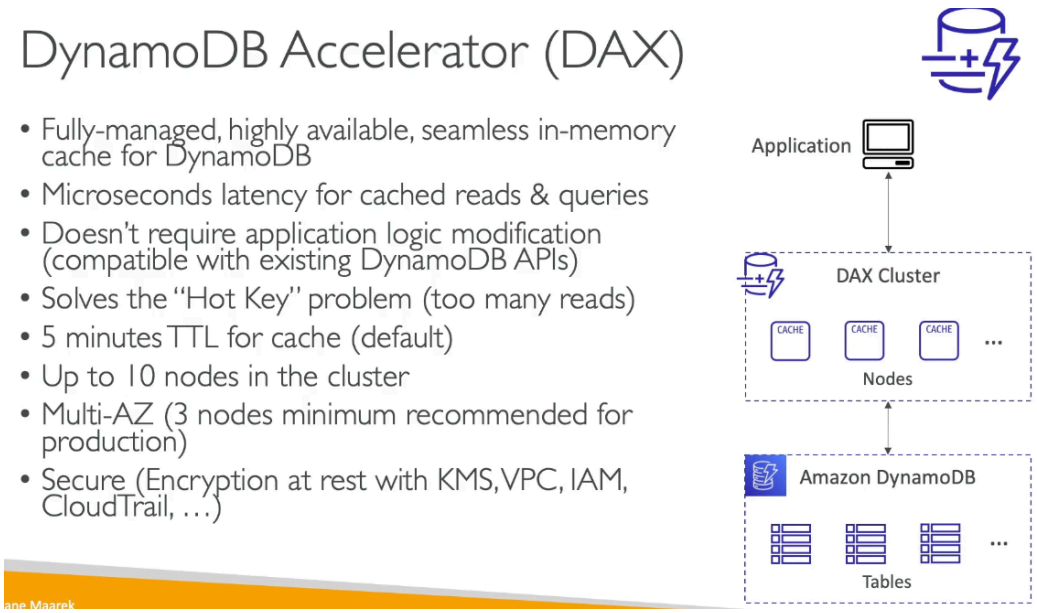
- DynamoDB fully managed, NoSQL (not like RDS), highly available multi AZ, fast performance, integration with IAM, auto scaling, Access classes
    - No maintenance or patching, always available
- Made of tables with each table having a primary key (must be decided at creation time)
- Infinite number of items or rows with attributes (can be added over time or null) with item size up to 400 KB
    - Supported types: Scalar, document, set types
- Scales to massive workloads, distributed DB
    - Millions of requests / seconds, 100s TB storage
    - Single digit ms speed
- Standard & Infrequent (IA) table class

Dynamo WCU & RCU – Throughput

- Control table's capacity (read / write throughput)
- Switch between modes every 24 hours

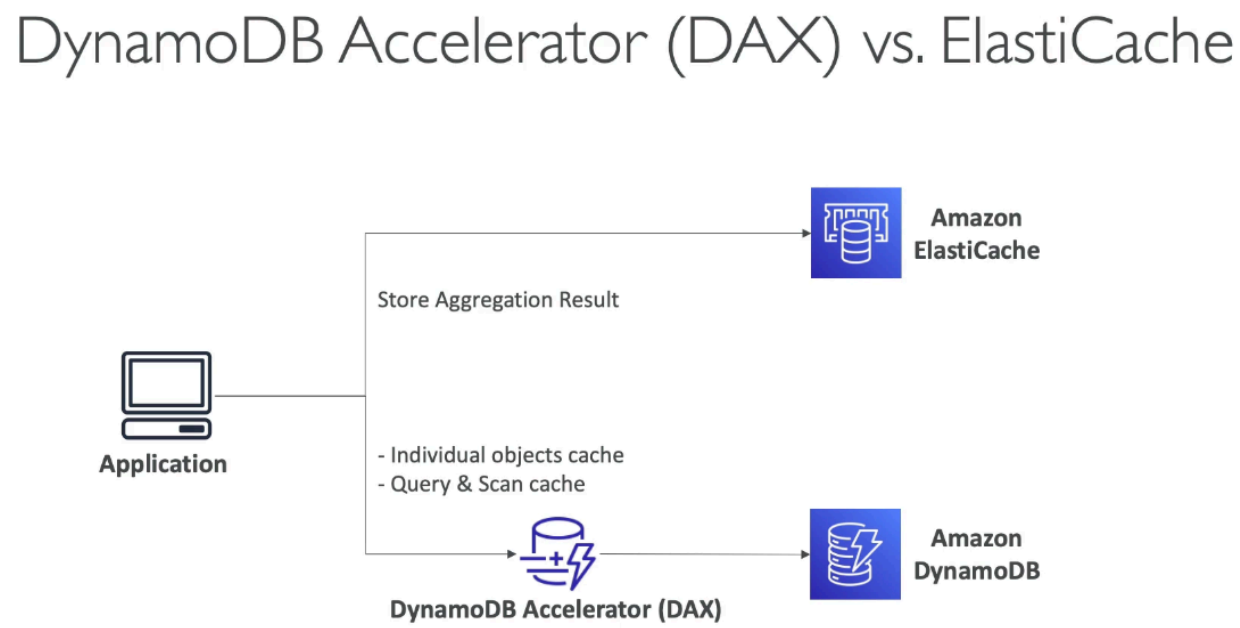
1. Provisioned Mode (default)
    a. Specify number of read/writes per second, planning capacity beforehand
    b. Pay for provisioned read & write capacity units
2. On Demand Mode
    a. Read/writes auto scaled without capacity planning, pay for what you use but more expensive

DynamoDB Accelerator (DAX)

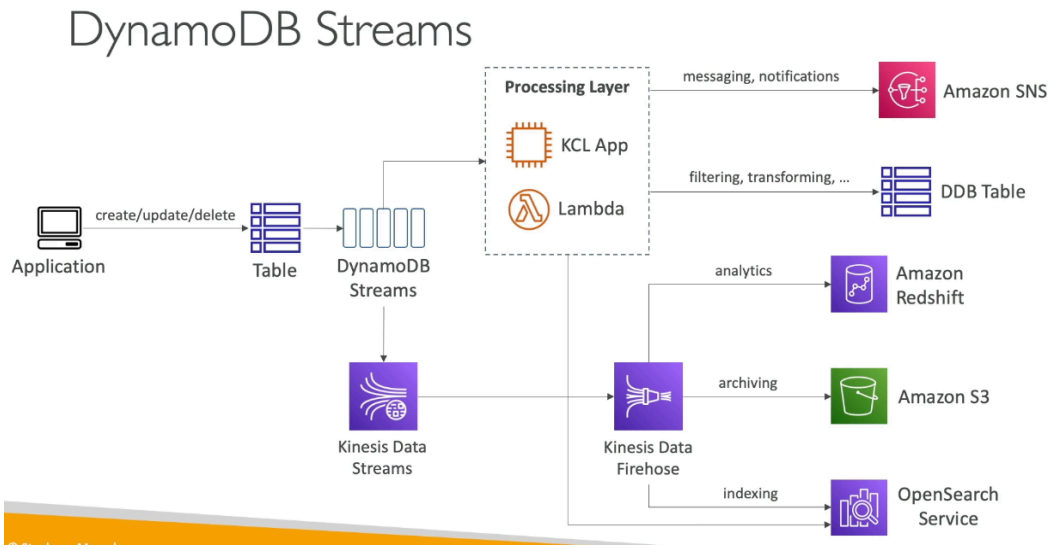


- Fully managed, highly available, low latency, in memory cache, no need to update application logic with multi AZ support and encryption
- Solves “hotkey” problem (too many reads)
- 5 min TTL (default), up to 10 nodes in cluster
    - T type: baseline + bursting for low throughput
    - R type: always ready

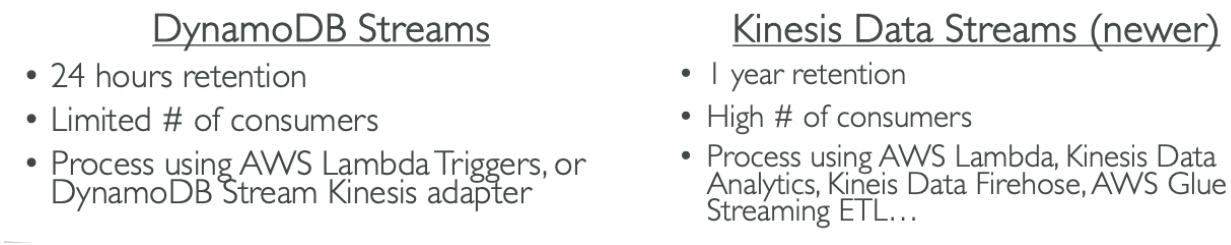
DAX vs ElastiCache



DB Stream Processing



- Ordered stream of item level modifications in a table
    - Records are not retroactively populated in a stream after enabling

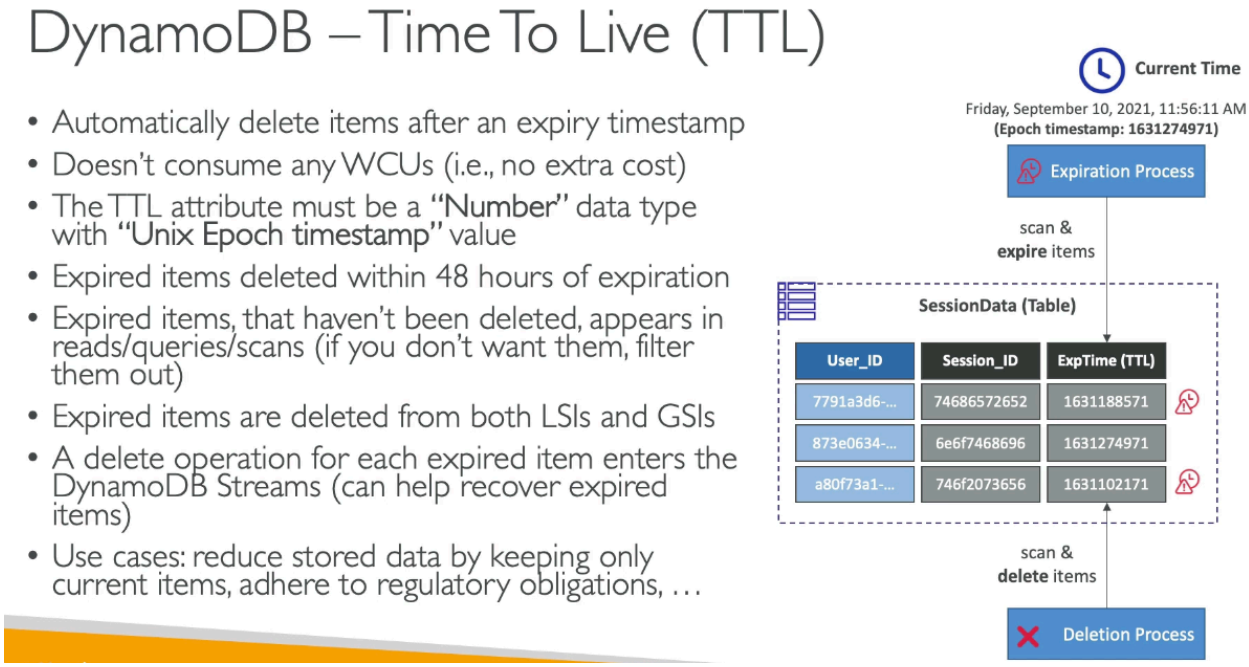


- Uses cases: react to changes in real time, analytics, send data to other services, etc…

DynamoDB Global Tables

- Make table accessible with low latency in multiple regions
- Active active replication (2 way replication)
- Applications can read / write to table in any region
    - Must enable DynamoDB streams as prerequisite

TTL



- Automatically delete items within 48 hours after TTL expires, doesn't consume WCUs
- Expired items that haven't been deleted appears in reads/queries/scans, must filter to not see them

- Expired items are deleted from both LSIs and GSIs
- Delete operation for each expired item enters Streams

Backups for Disaster Recovery

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

Integration with S3

- Export to S3 (must enable point in time recovery)
    - Works for any point in time within 35 days
    - Doesn't affect read capacity of 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 rom S3
    - Import CSV, DynamoDB JSON or ION format
    - Doesn't consume write capacity
    - Creates new table
    - Import errors logged in CloudWatch

API Gateway Overview

- Lambda + GW = no infrastructure
- Support WebSocket, handle API versioning, different environments, security
    - User Auth via IAM, Cognito, custom authorizer
- Create API keys, handle request throttling
- Transform and validate requests and responses
- Generate SDK and API specifications
- Cache API responses

Endpoint Types

1. Edge Optimized (default): for global clients where requests routed through CF Edge location, but API GW still in 1 region

2. Regional: clients within same region, could manually combine with CloudFront
3. Private: accessed from VPC using VPC endpoint ENI using resource policy

API GW Integrations

- Lambda
    - Invoke, easy way to expose REST API backed by Lambda
- HTTP
    - Expose any HTTP endpoints in backend
        - HTTP API, ALB, etc… for added features
- AWS Service
    - Expose any AWS API
        - Example: start Step Function workflow, SQS…

API GW Security

- User authentication
    - IAM, Cognito, custom authorizer
- Custom Domain Name HTTPS security through ACM
    - For edge optimized endpoint certificate must be in us east 1
    - For regional endpoint, certificate must be in API GW region
    - Must setup CNAME or A record in Route 53

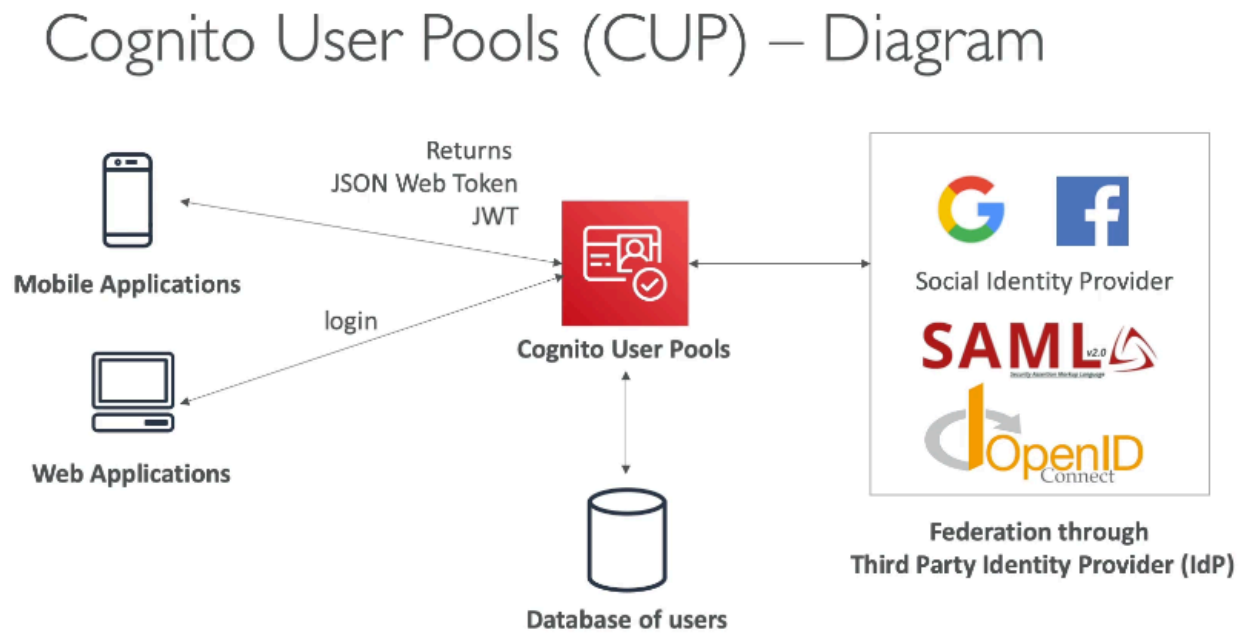
AWS Step Functions

- Serverless model workflows in JSON as state machines (1 per workflow)
    - Order fulfillment, data processing, web apps, any workflow
    - Visualization of the workflow, execution, and history
- Mostly with lambda functions, but also integrates with:
    - EC2, ECS, on premise…
- Features: sequence, parallel, conditions, timeouts, error handling…

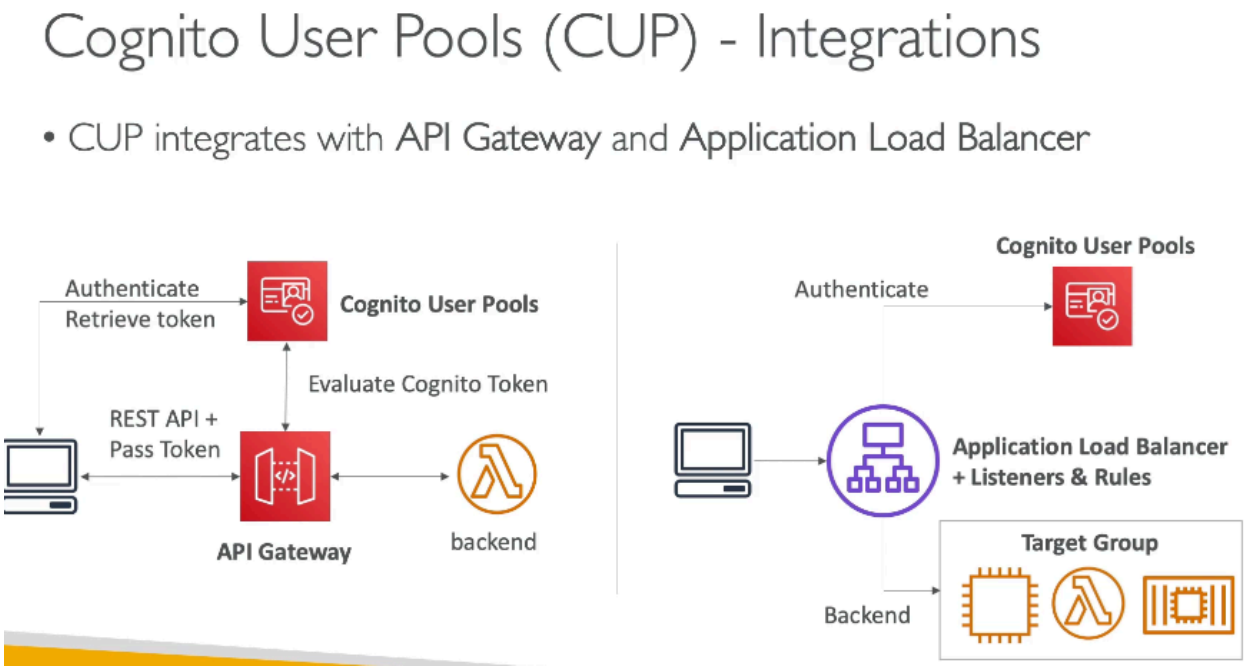
Cognito Overview

- Give users an identity to interact with application
    - Provides a Cognito Hosted UI for users to sign in and sign up
- Cognito User Pools:
    - Sign in functionality for users, integration with API GW & ALB
- Cognito Identity Pools
    - Temporary AWS credentials to access AWS resources directly
    - Integrate with Cognito User pools as identity provider
- Cognito vs IAM: “hundreds of users”, “mobile users”, authenticate with SAML

Cognito User Pools

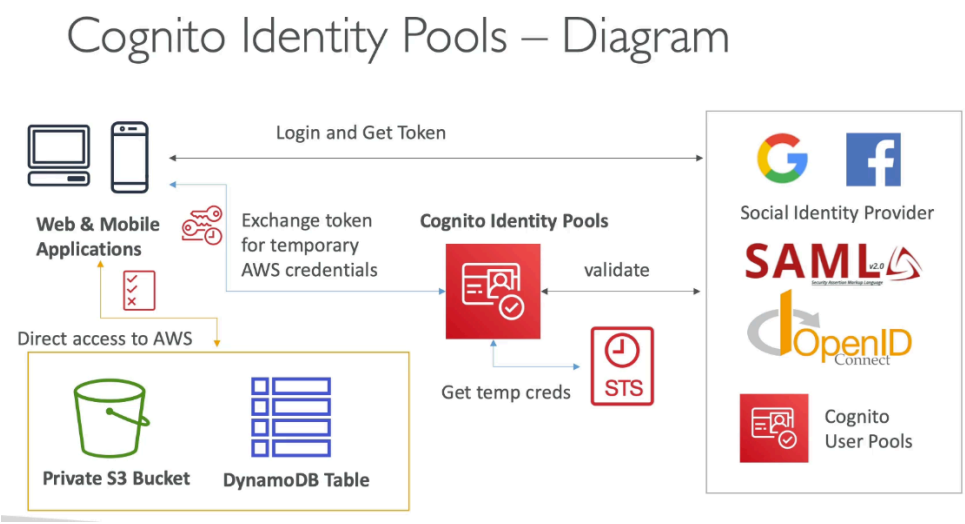


- Serverless DB of users for web / mobile apps
- Simple login (username, password) with password reset, email/phone verification, MFA, 3rd party federated identities (Google, FB, etc…)
    - Can block users if credentials are compromised elsewhere
    - Login sends back as JWT

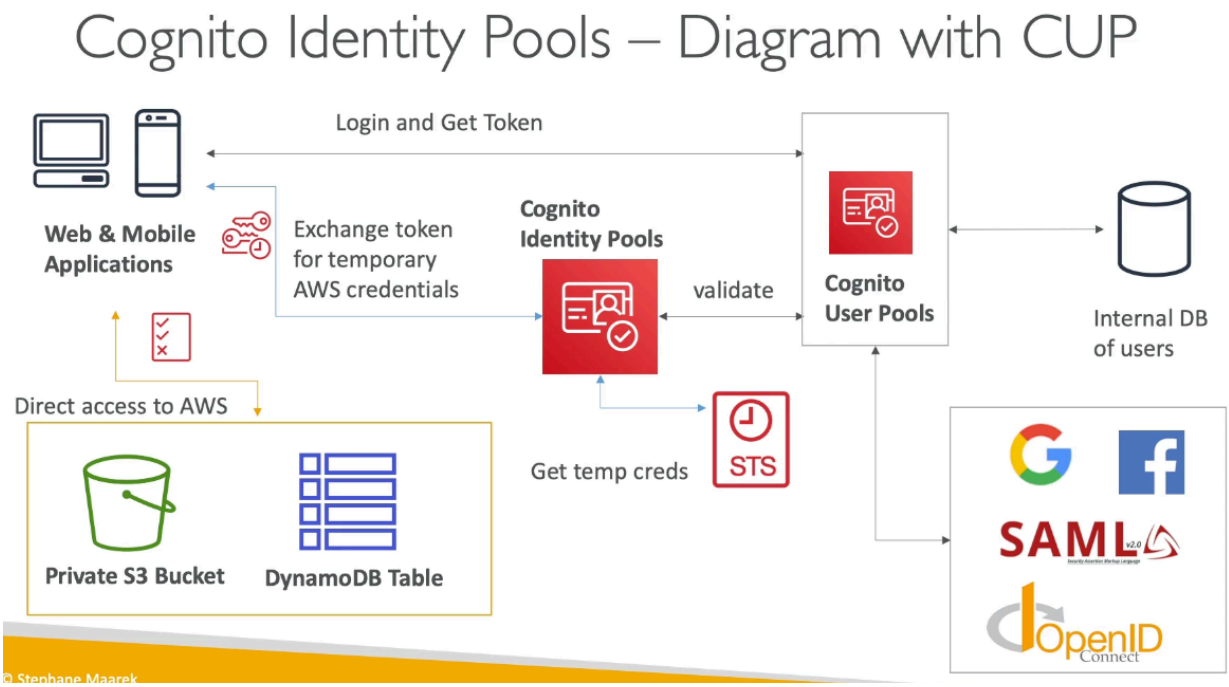


- Integrates with API GW and ALB

Cognito Identity Pools

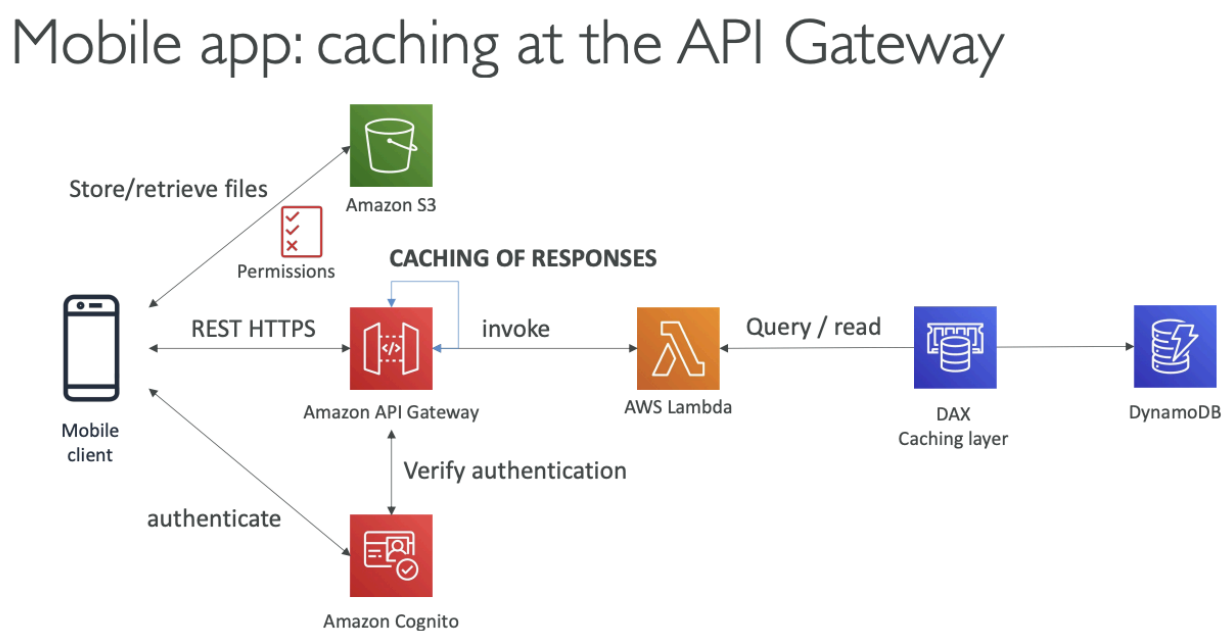


- Get identities for outside users for temporary AWS credentials
    - Identity pool can be public providers, Amazon Cognito, OpenID, custom login
        - Cognito Identity Pools allow for unauthenticated guest access
- Users can access AWS Services directly or via API GW
    - IAM policies applied to credentials are defined in Cognito and can be customized for fine grained control



Section 20: Serverless Solution Architecture Discussions

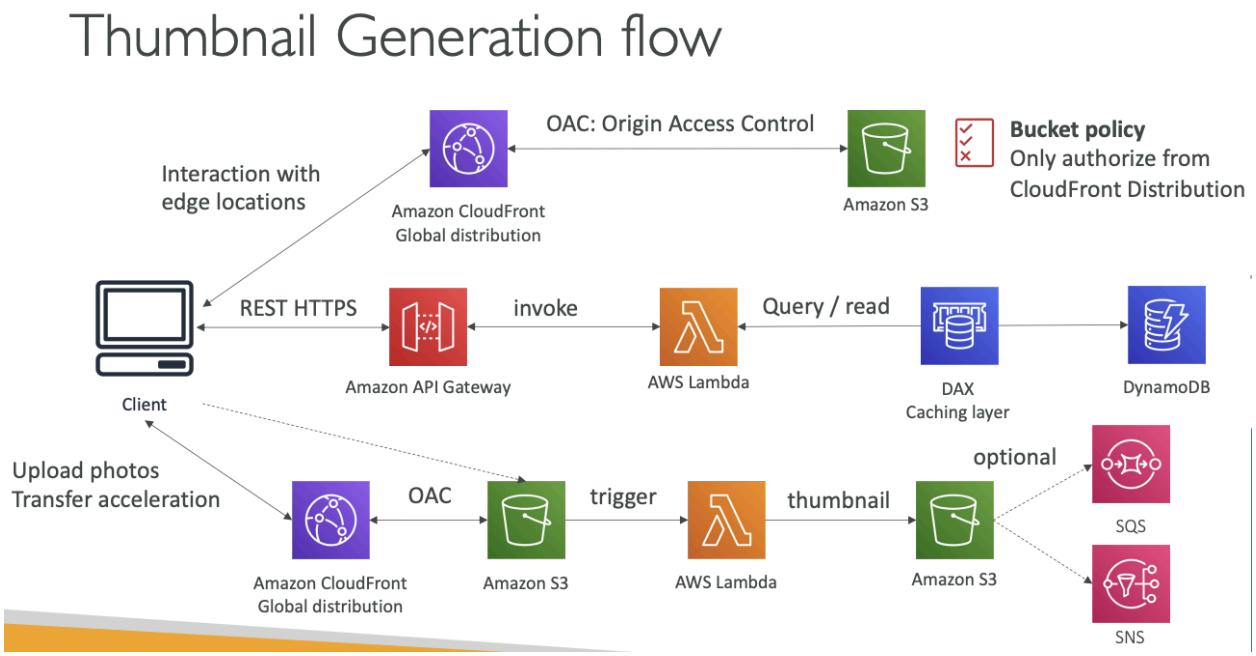
Mobile Application



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

- For REST API, use API Gateway + Lambda for a serverless architecture. To authenticate to access S3, use Cognito Identity Pool to retrieve files from S3.
    - With increased high read throughput of static data, DAX can be used alongside DynamoDB as a serverless DB. API GW can also cache some responses

Serverless Website

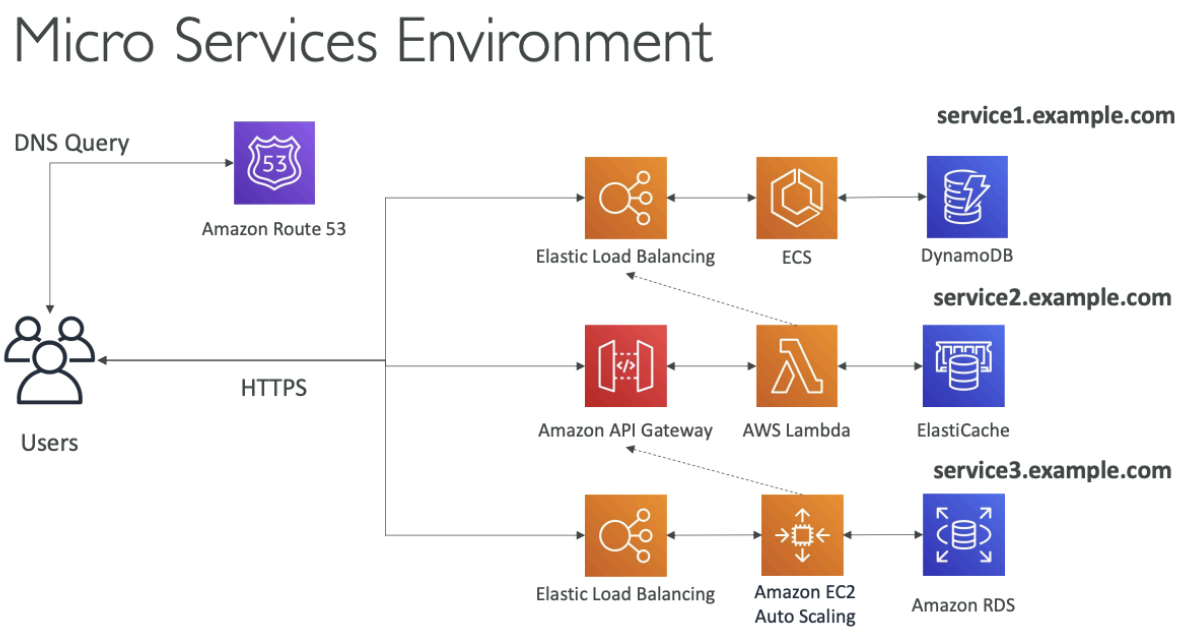


AWS Hosted Website Summary

- 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

- To serve clients globally, use CloudFront to access S3. To securely access, use Origin Access Control (OAC) to only allow CloudFront to access S3. REST API needs API GW to invoke lambda function and a serverless DB like DynamoDB. To send welcome emails, enable Dynamo streams to invoke a Lambda function to send emails via SES.
    - If users upload images for thumbnails, use CloudFront + OAC to S3 again and have a Lambda function triggered to create the thumbnail to S3.

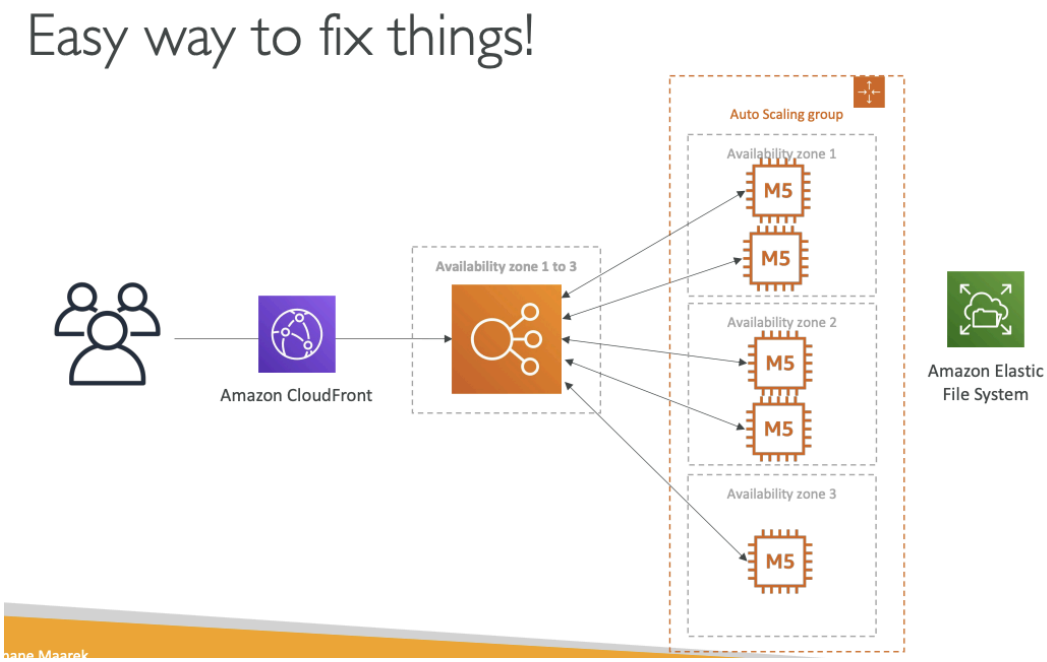
Microservices Architecture



- Design microservices as you want
- Synchronous patterns: API GW, LB
- Asynchronous pattern: SQS, Kinesis, SNS, Lambda triggers (S3)
- Challenges:
    - Repeated overhead for creating each microservices
    - Issues with optimizing server utilization
    - Complexity of running multiple versions of multiple microservices simultaneously
    - Proliferation of client side code requirements to integrate with many separate services
- Can be solved by serverless patterns:
    - API GW, Lambda scale automatically and pay per usage
    - Easily clone API, reproduce environments

Software Updates Offloading

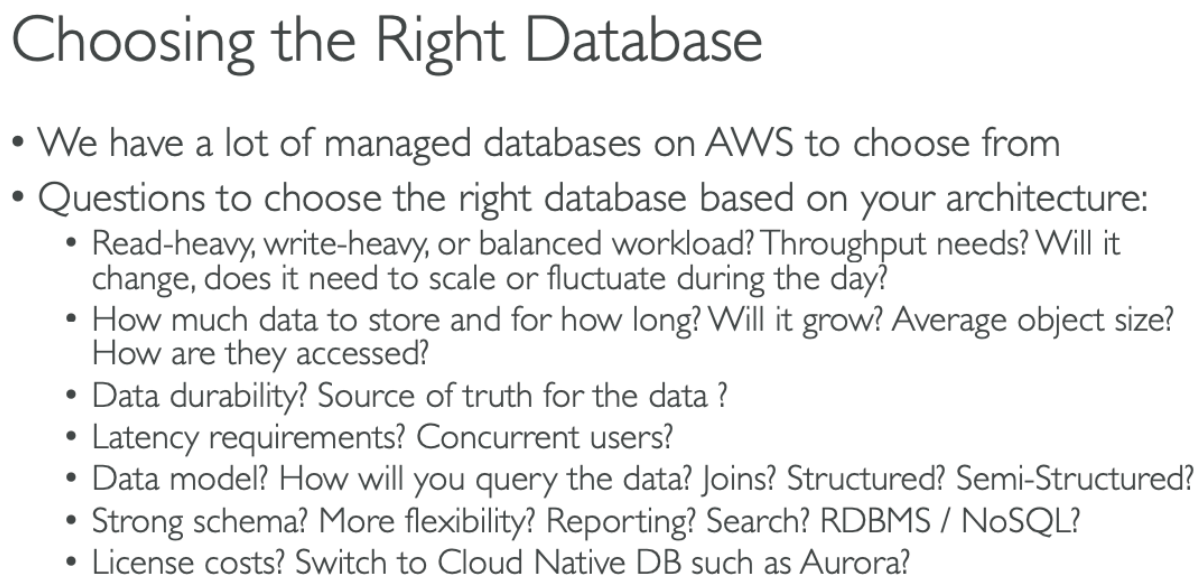
- When a new software update is out, lots of requests are made to update and content is distributed in mass over the network, thus very costly. How to optimize cost?



- Use CloudFront because there are no changes to architecture and will cache software update files at the edge
    - Software files not dynamic, CloudFront auto scales

Section 21: Databases in AWS

Choosing the right DB



Database Types

- RDBMS (SQL): RDS / Aurora, great for joins
- NoSQL: no joins; Dynamo (JSON), ElastiCache (key value), Neptune (graphs), DocumentDB (Mongo), Keyspaces (Apache Cassandra)
- Object Store: S3, Glacier
- Data Warehouse: SQL Analytics; Redshift, Athena, EMR
- Search: OpenSearch (JSON), free text, unstructured searches
- Graphs: Neptune - displays relationships between data
- Ledger: Amazon Quantum Ledger DB
- Time series: Amazon Timestream

RDS Summary

- Managed PostgresQL / MySQL / Oracle / SQL Server / DB2 / MariaDB / Custom
- Provisioned RDS instance size and EBS volume type & size
    - Auto scaling for storage
- Support read replicas and multi AZ
- Security via IAM, SG, KMS, SSL in transit
    - Support for IAM Authentication, Secrets Manager
- Automated backup with point in time (35 days)
    - Manual snapshot for long term storage
- Managed and scheduled maintenance (downtime)
- RDS Custom for access to and customize underlying instance (Oracle & SQL server)
- Use case: relational DB, perform SQL queries, transactions

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
- Aurora Serverless
    - For unpredictable / intermittent workloads, no capacity planning
- Aurora Global:
    - Up to 16 DB Read Instances in each region, < 1 second storage replication
- Aurora Machine Learning:
    - 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

ElastiCache

- Managed Redis / Memcached (similar as RDS, but for cache)
    - In memory data store with sub ms latency
- Redis clustering and multi AZ, read replicas (sharding)
- Security via IAM, SG, KMS, Redis Auth
- Backup / snapshot / point in time restore
- Managed and scheduled maintenance
- Some application code changes needed
- Use case: key / value store, frequent read, cache DB queries, session data, cannot use SQL

DynamoDB

- NoSQL AWS managed, serverless, millisecond latency
- Capacity mode:
    - Provisioned with optional auto scaling
    - On demand
- Can replace ElastiCache as key value store (storing session data, TTL)
- Highly available, multi AZ by default, read / write decoupled, transaction capability
- DAX cluster for read cache, low read latency
- Security, authentication, authorization done via IAM
- Event processing: Dynamo Streams with Lambda or Kinesis Data Stream
- Global tables: active active setup
- Automated backup up to 35 days with point in time (restore to new table) or on demand backup
- Export to S3 without using RCU within point in time window or import to S3 without using WCU
- Great to rapidly evolve schemas
- Use case: serverless DB, distributed serverless cache

S3

- Key value store for objects
    - Large objects, not small
- Serverless, infinite scaling, max size 5 TB, versioning
- Tiers of S3 with lifecycle policies
- Features:
    - Versioning, encryption, replication, MFA delete, access logs…
- Security:
    - IAM, bucket policy, ACL, access points, object lambda, CORS, object / vault lock

- Encryption:
    - SSE S3, SSE KMS, SSE C, client side, TLS, default encryption
- Batch operations via S3 Batch, list files using S3 Inventory
- Performance
    - Multi part upload, S3 transfer acceleration, S3 select
- Automation:
    - S3 Event Notifications
- Use case: static files, key value store for big files, website hosting

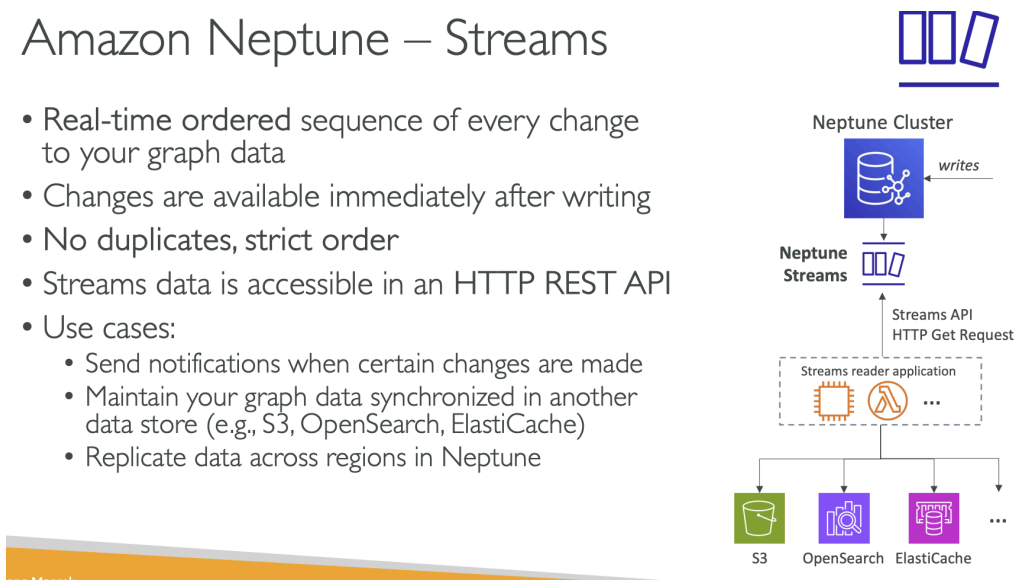
DocumentDB

- AWS version of MongoDB (NoSQL)
    - Used to store, query, index JSON data
        - Auto grows in increments of 10GB
        - Auto scales to workloads
    - Fully managed, highly available with replication across 3 AZ
- Similar deployment concepts as Aurora

Amazon Neptune

- Fully managed graph DB (like social network)
    - Highly available across 3 AZ, up to 15 read replicas
- Build and run apps working with highly connected datasets
    - Optimized for complex and hard queries
    - Store billions of relations and query the graph with ms latency
- Use cases: knowledge graphs, fraud detection, recommendations, social networking

Neptune Streams



- Real time ordered sequence of every change to graph data
    - Changes immediately after writing
- No duplicates, strict ordering
- Stream data accessible via HTTP REST API
- Use cases:
    - Send notifications when certain changes made, maintain graph data synchronized in another data store
    - Replicate data across regions in Neptune

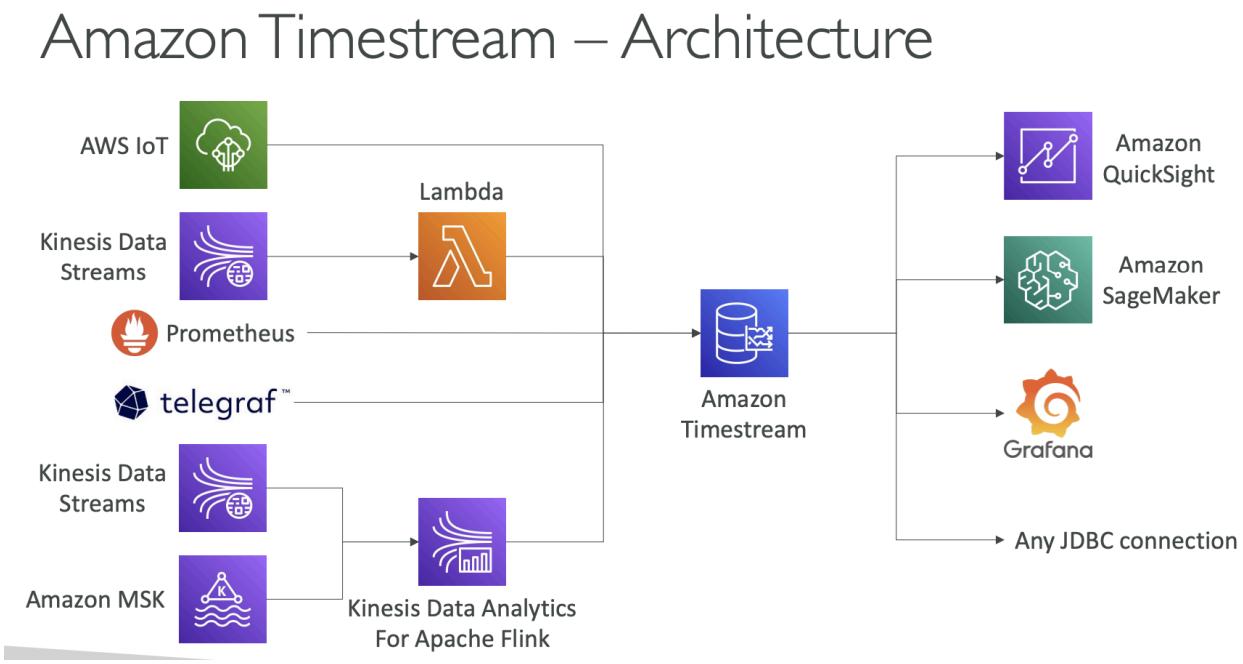
Amazon Keyspaces (for Apache Cassandra)

- Cassandra is NoSQL, Keyspaces is AWS managed, serverless, scalable, highly available
- Automatically scales tables up / down based on traffic
    - Tables replicated 3 times across multiple AZ
- Use Cassandra Query Language (CQL)
- Single digit ms latency at any scale, 1000s of requests / second
- Capacity: on demand or provisioned with auto scaling
- Encryption, backup, point in time up to 35 days
- Use cases: Apache Cassandra

Amazon QLDB

- Quantum Ledger DB
    - Ledge is a book for recording financial transaction
    - Fully managed, serverless, highly available, replication across 3 AZ
- Used to review history of all changes made to application data over time
- Immutable system: no entry can be removed or modified, cryptographically verifiable
    - Journal behind the scenes to have a sequence number any time revision made
- 2-3x better performance than others, use SQL
- Difference with Amazon Managed Blockchain: no decentralization component

Amazon Timestream



- Fully managed, fast, scalable, serverless time series database
    - Automatically scales up/down to adjust capacity
    - 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 case: data in relation to time

Section 22: Data and Analytics

Amazon Athena

- Serverless query service to analyze data stored in S3 via SQL to query files (business analytics, reporting, etc…)
    - Supports CSV, JSON…
    - Pricing: $5 per TB of data scanned
- Commonly used with QuickSight for reporting / dashboards
- Analyze data in S3 using serverless SQL, use Athena

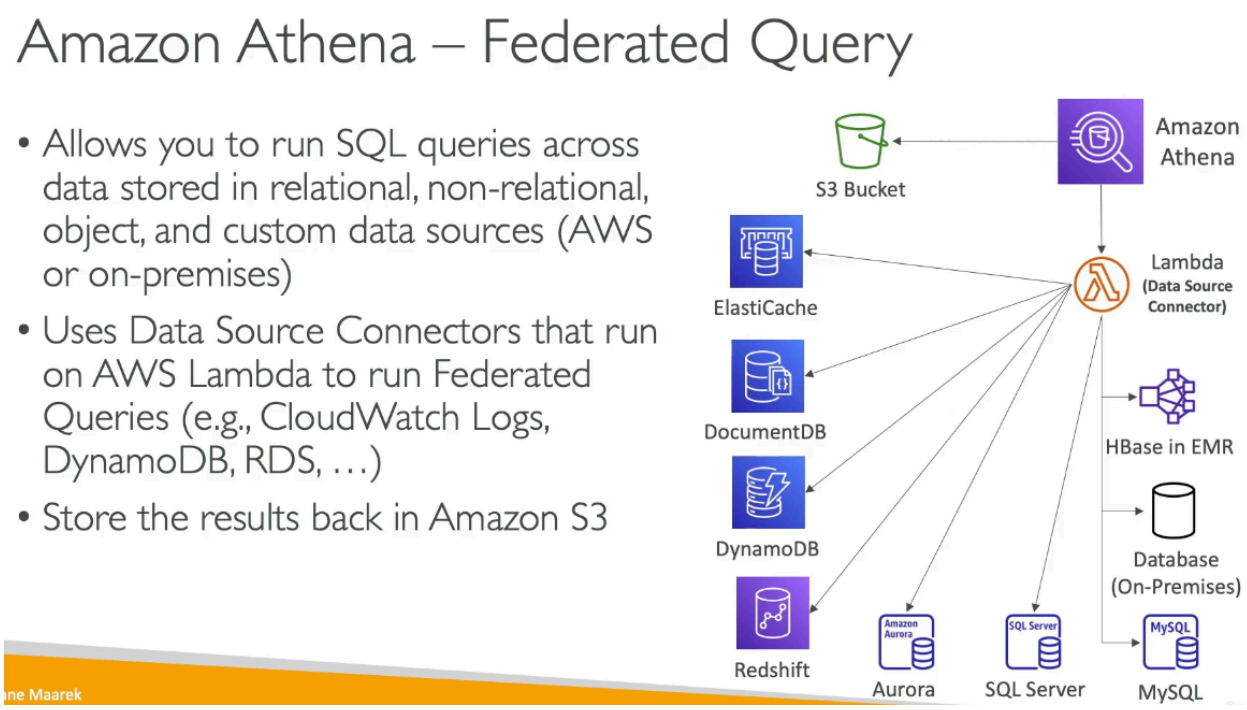
Performance Improvement

- Columnar data for cost savings (less scan) to only scan the columns you need
    - Apache Parquet or OCR is recommended
    - Huge performance improvement
    - Use Glue to convert data to Parquet or ORC
- Compress data for smaller retrievals
- Partition datasets in S3 for easy querying on virtual columns
    - s3://bucket//pathToTable for direct access

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

Federated Query

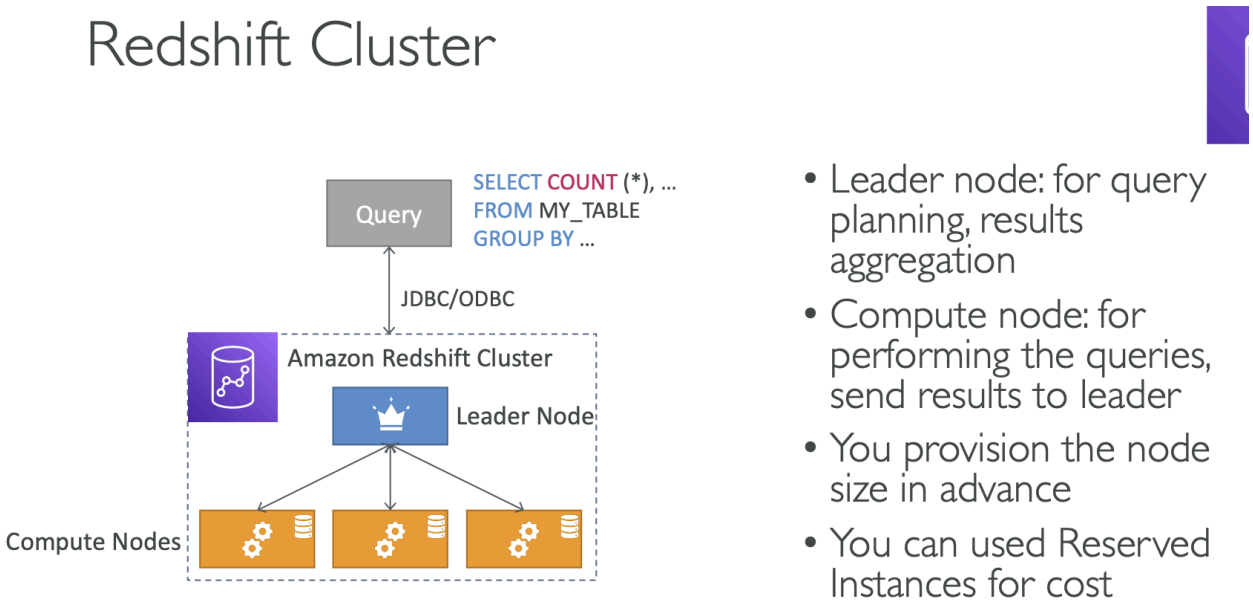


- Allows to run SQL queries across data stored in relational, non relational, object, and custom data sources (AWS or on premise)
- Uses data source connectors that run on Lambda to run Federated Queries in other services (CW logs, DynamoDB, RDS…) and stores results back to S3

Amazon Redshift

- Online analytical processing for analytics and data warehousing
    - SQL interface for queries
    - BI tools like Quicksight or Tableau integrate with it
- Columnar storage of data (instead of row) & parallel query engine
    - Pay as you go based on instances provisioned
- vs Athena: faster queries / joins / aggregations thanks to indexes

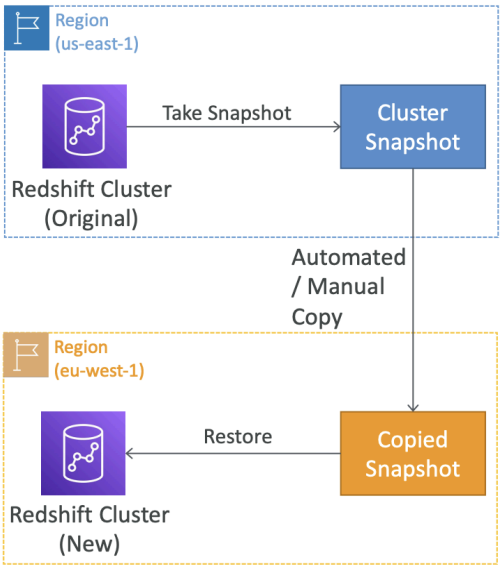
Redshift Cluster



- Leader node: for query planning, results aggregation
- Compute node: perform queries, send results to leader
- Must provision node size in advance, can used reserved instances for cost savings

Snapshots & DR

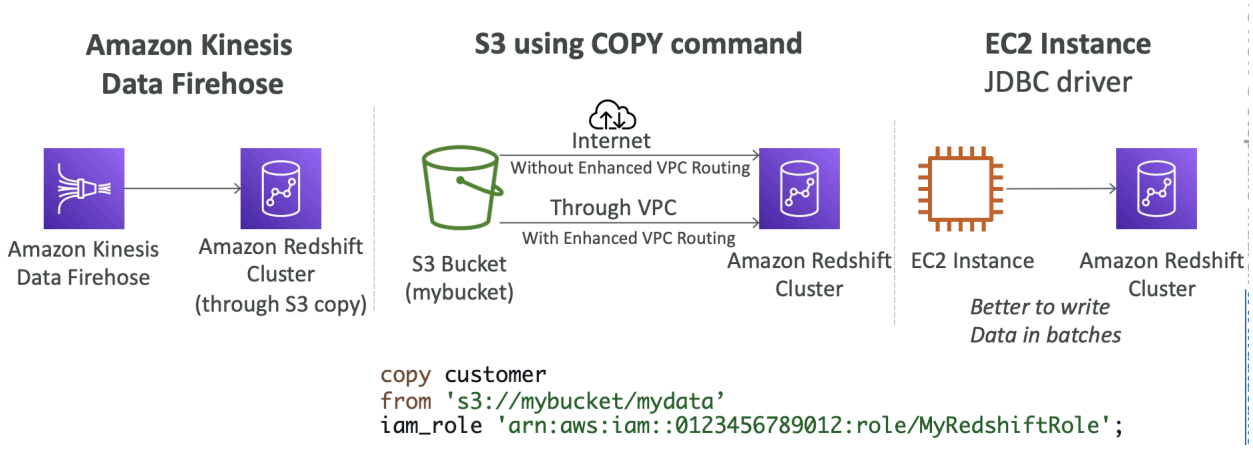
- Redshift has multi AZ for some cluster types
    - Snapshots are point in time backups of a cluster, stored internally in S3
    - Snapshots are incremental (only what has changed is saved)
    - Restore snapshot into new cluster



- Automated: every 8 hours, every 5 GB, or on schedule. Set retention
- Manual: snapshot retained until deleted

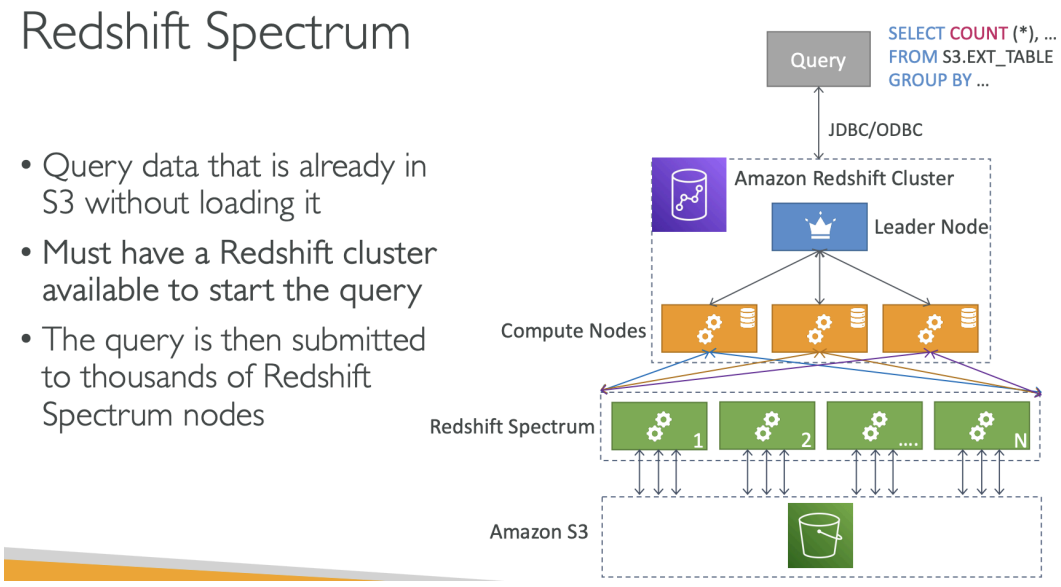
- Can configure Redshift to auto copy snapshots of a cluster to another region

Loading Data into Redshift



- Large inserts better

Redshift Spectrum

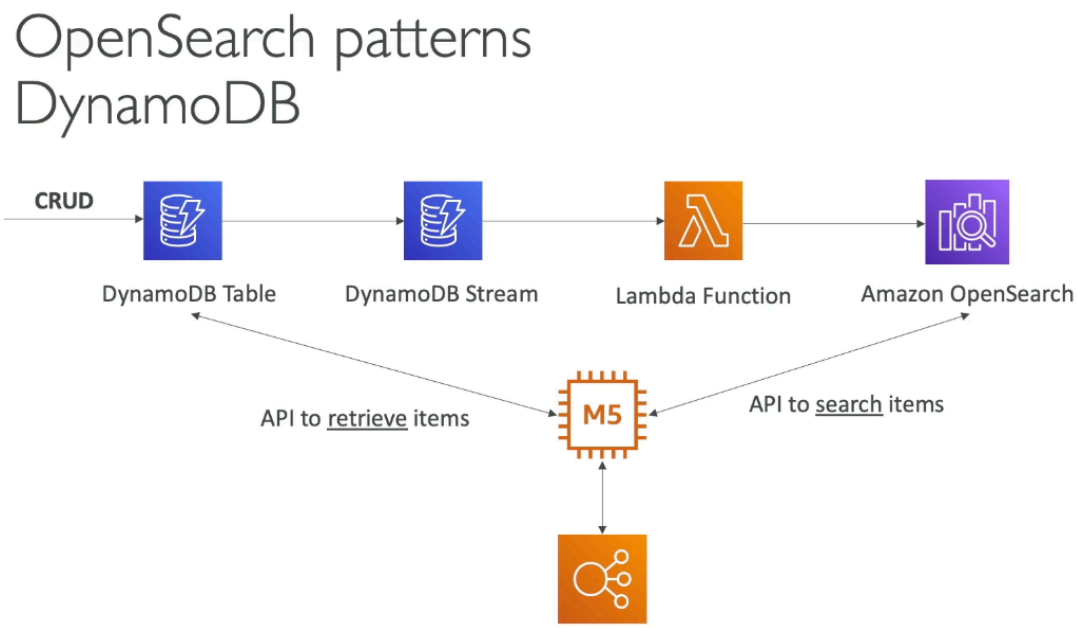


- Query data in S3 without loading
- Must have Redshift cluster available to start query
  - Query is submitted to thousands of Redshift Spectrum nodes

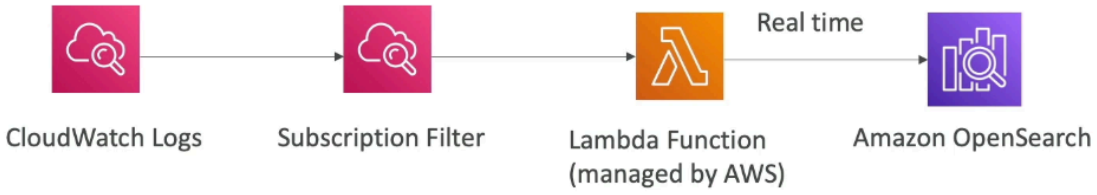
Amazon OpenSearch

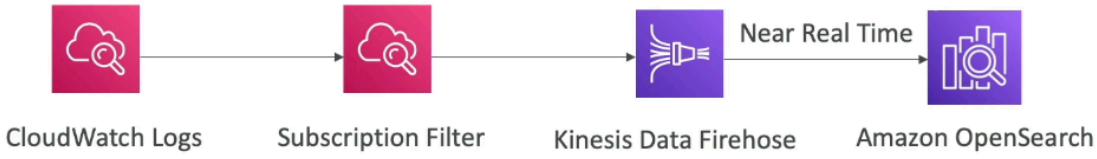
- In DynamoDB, queries only exist by primary key or indexes, but with OpenSearch you can search any field, even partial matches and comes with OpenSearch Dashboards for visualization
    - Common to use OpenSearch as a complement to another DB
    - Ingestion of data from Kinesis Firehose, AWS IoT, CloudWatch logs
- 2 modes: managed cluster or serverless
- No native support for SQL
- Security through Cognito, IAM, KMS, TLS

Architecture Patterns

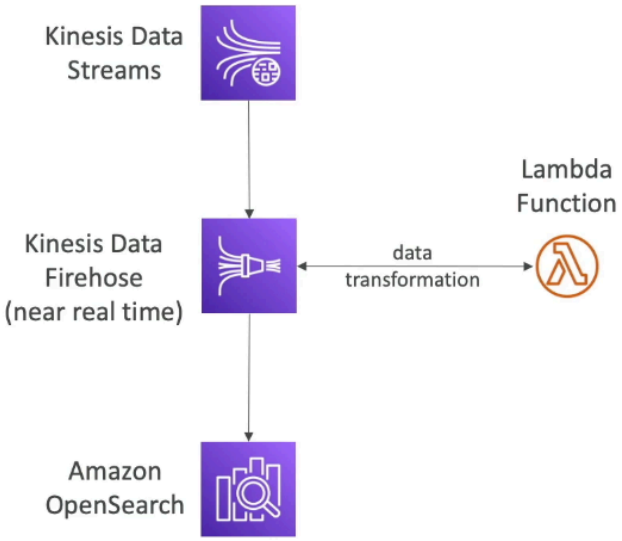


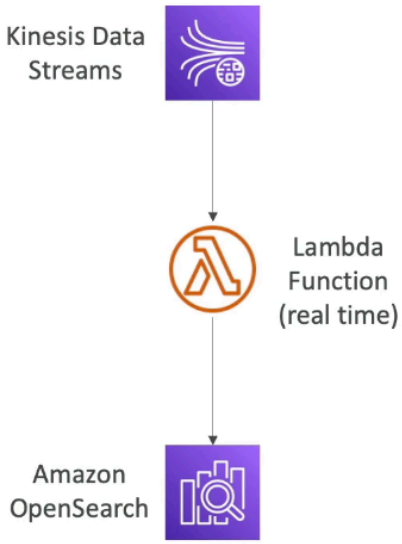
OpenSearch patterns
CloudWatch Logs





OpenSearch patterns
Kinesis Data Streams & Kinesis Data Firehose





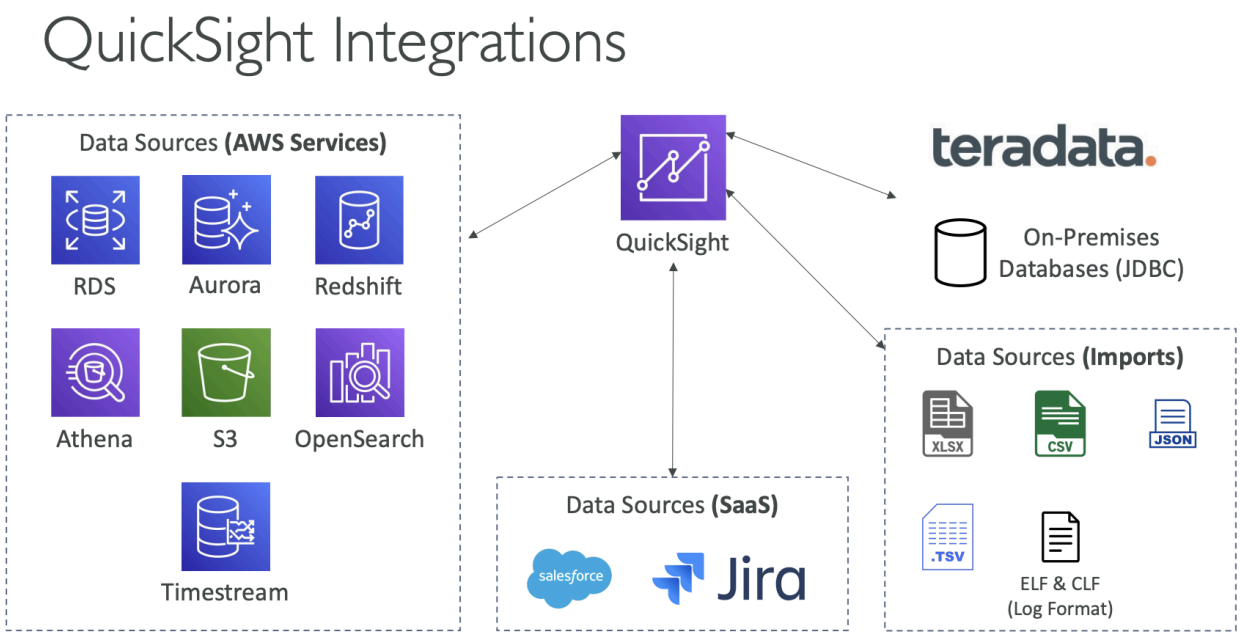
Amazon EMR

- Create Hadoop clusters (big data) to analyze and process vast amount of data
- Clusters made with hundreds of EC2 instances
    - EMR bundles with Apache Spark, Flink…
- Takes care of all provisioning and configuration, auto scaling and integrated with Spot instances
- Use cases: data processing, ML, web indexing, big data

EMR Node Types & Purchasing

- Master node: manage cluster, coordinate, manage health – long running
- Core node: run task and store data – long running
- Task node (optional): just to run tasks – usually spot
- Purchasing:
    - On demand: reliable, predictable, no termination
    - Reserved (1 year min): cost savings (EMR will automatically use if available)
        - Master and core nodes good options for this
    - Spot Instance: cheaper, can be terminated, less reliable
- Can have long running cluster or transient (temp) cluster deployment modes

Amazon QuickSight



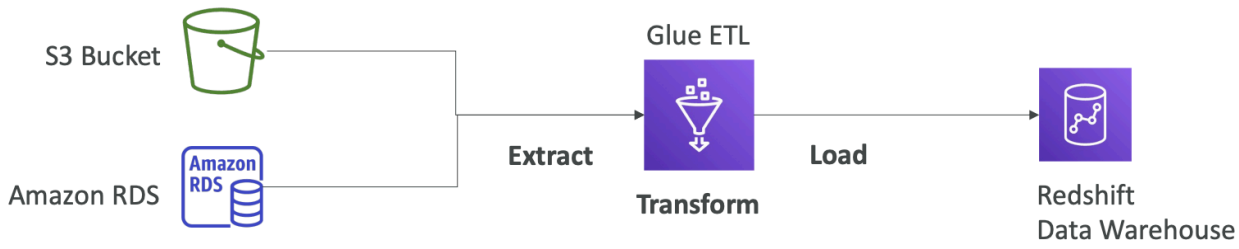
- Serverless ML powered BI to create dashboards
    - Fast, auto scale, embeddable with per session pricing
- Use case: business analytics, visualizations, insights…

- Integrated with RDS, Aurora, Athena, Redshift, S3…
- In memory computation using SPICE engine if data imported into QuickSight
- Enterprise edition:
    - Can set up column level security (CLS) to prevent some columns to be displayed to some users

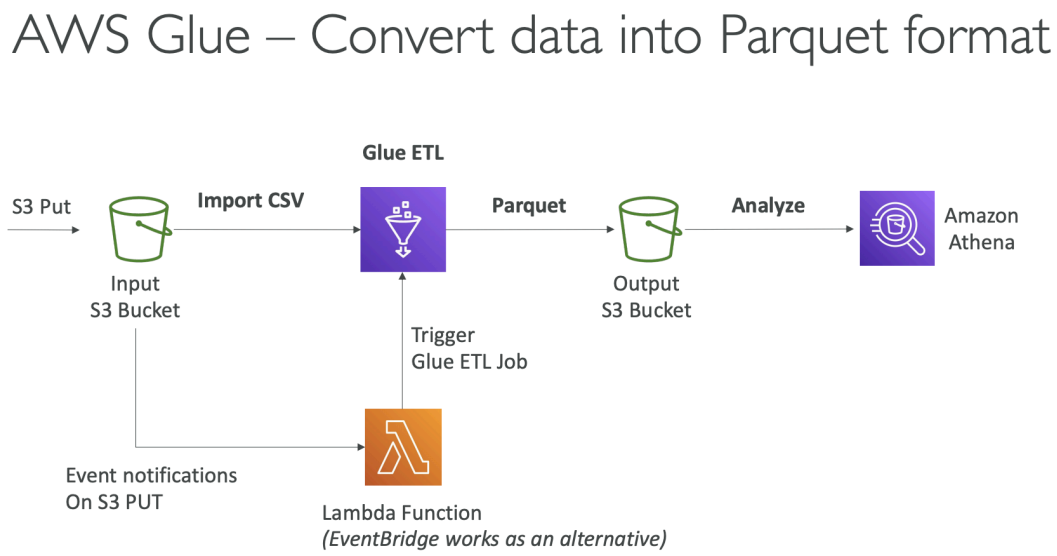
Dashboards & Analytics

- Define users (standard version) and Groups (enterprise)
    - Users & groups only exist within QuickSight
- Dashboard:
    - Read only snapshot of analysis that can be shared
    - Preserves the configuration of analysis (filtering, parameters, controls…)
- Share analysis or dashboard with users or groups by publishing
    - Users who see the dashboard can also see underlying data

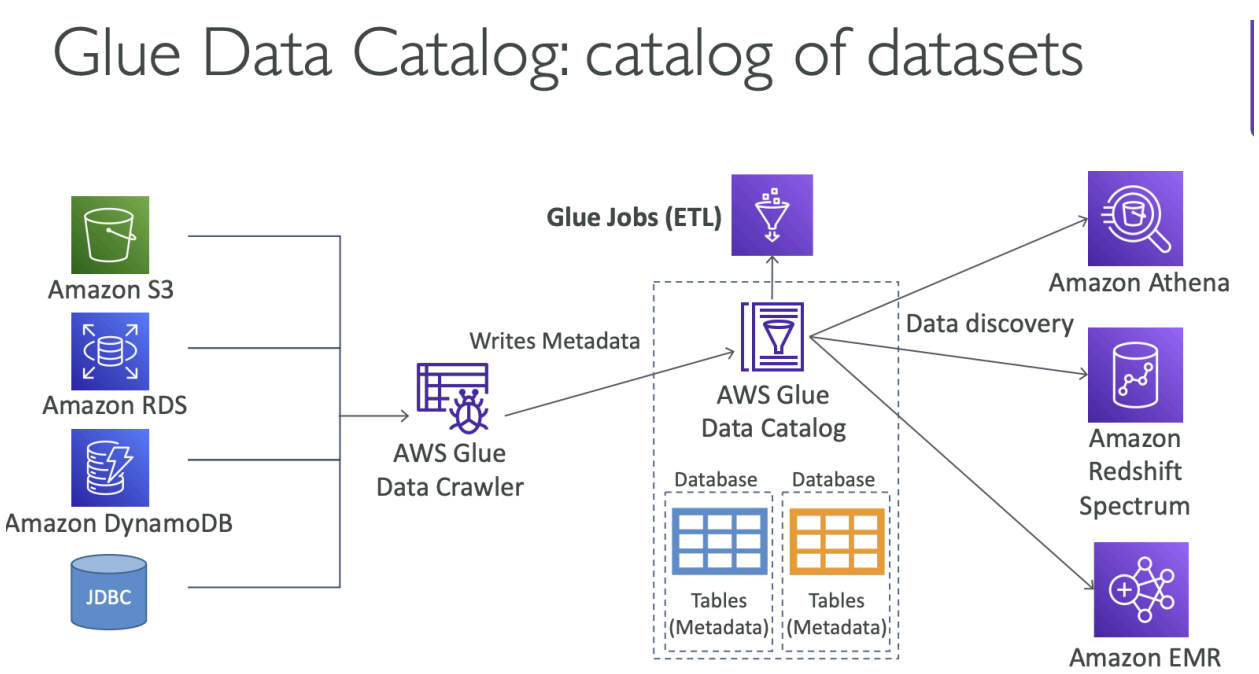
AWS Glue



- Fully serverless, managed extract, transform, and load (ETL) service
    - Prepare and transform data for analytics



Glue Data Catalog

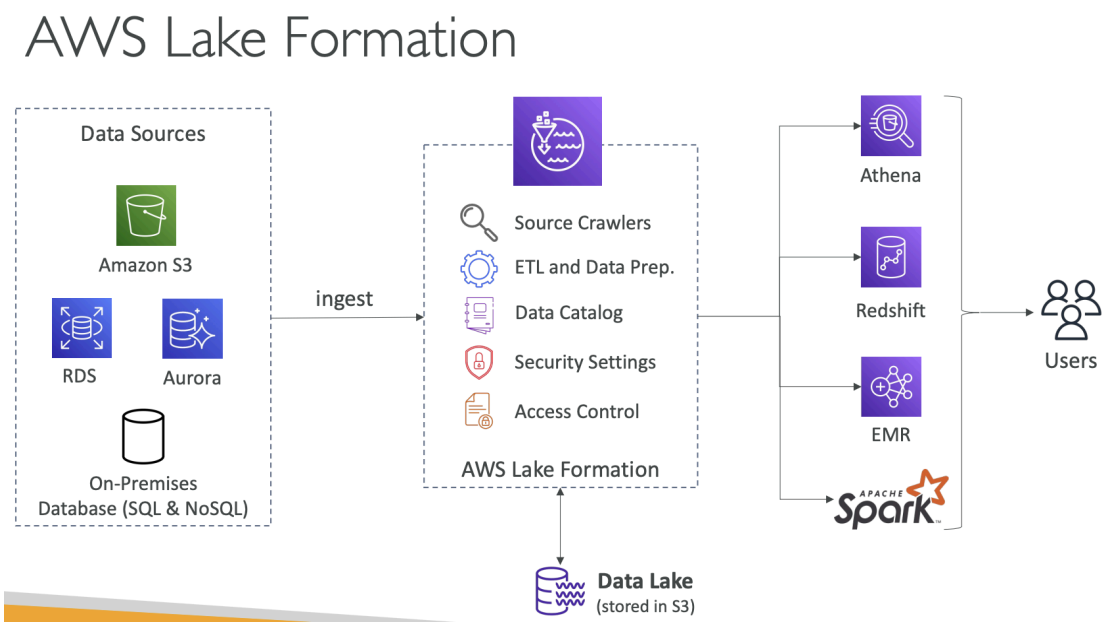


Glue High Level

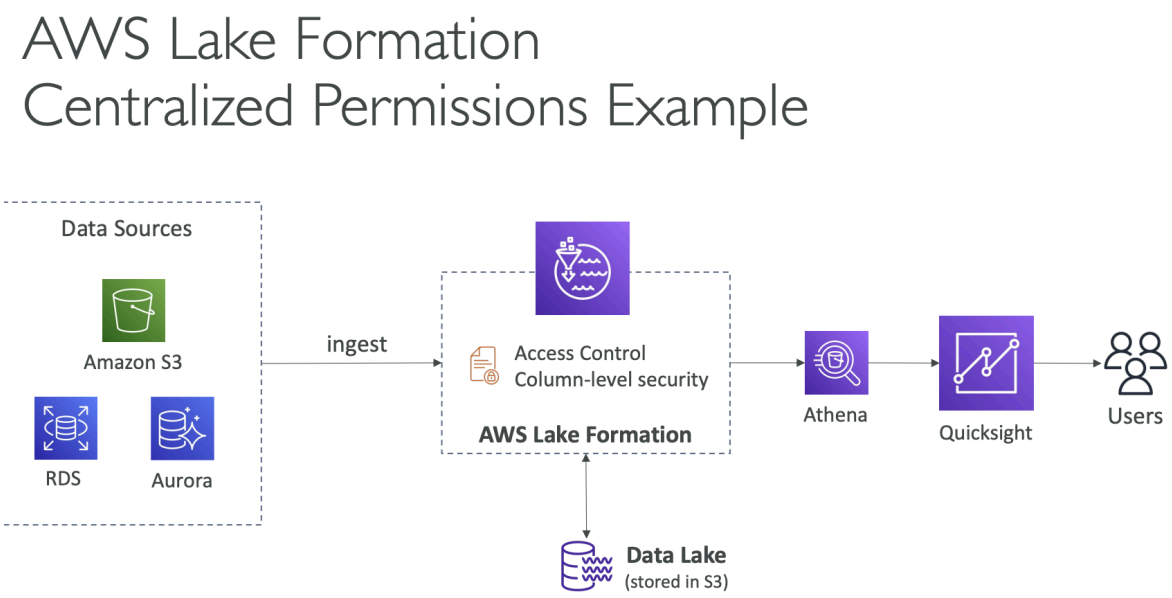
Glue – things to know at a high-level

- 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 LakeFormation

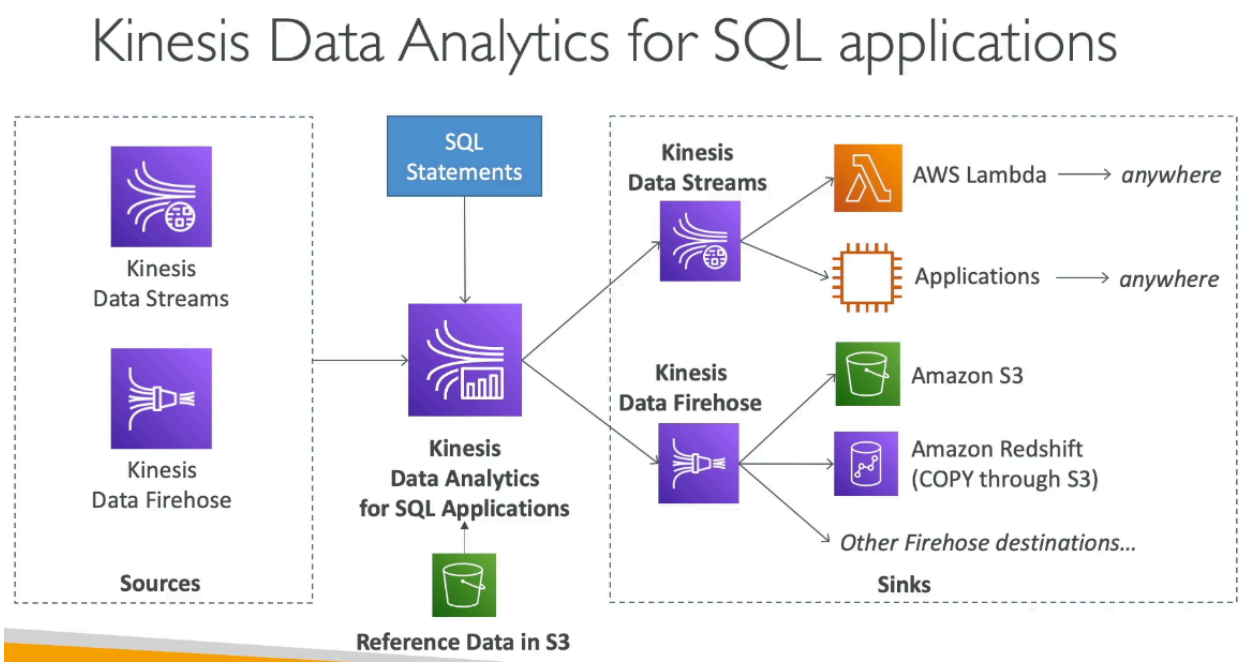


- Data lake = central place to have all your data for analytics purposes
- Fully managed service that makes it easy to setup a data lake in days
    - Discover, cleanse, transform, and ingest data into Data lake
    - Automates complex manual steps and deduplicate
- Combine structured and unstructured data in data lake
- Out of box source blueprints: S3, RDS, relational & NoSQL DB…
- Fine grained access control for applications (row and column level)
- Built on top of AWS Glue



Kinesis Data Analytics

For SQL Applications



Kinesis Data Analytics (SQL application)

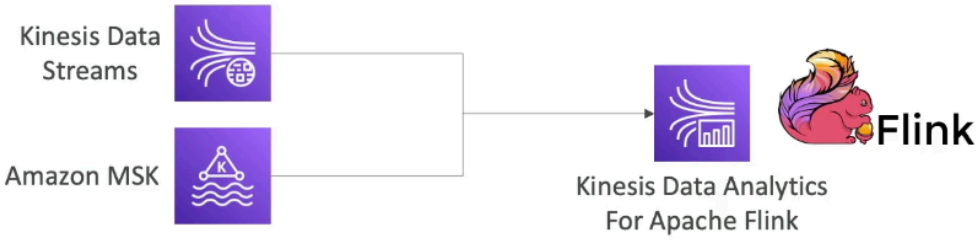
- Real-time analytics on **Kinesis Data Streams & Firehose** using SQL
- Add reference data from Amazon S3 to enrich streaming data
- Fully managed, no servers to provision
- Automatic scaling
- Pay for actual consumption rate
- Output:
  - Kinesis Data Streams: create streams out of the real-time analytics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

- Real time analytics on Kinesis Data Streams or Firehose using SQL
- Fully managed, no servers, automatic scaling, pay for consumption rate

For Apache Flink

Kinesis Data Analytics for Apache Flink

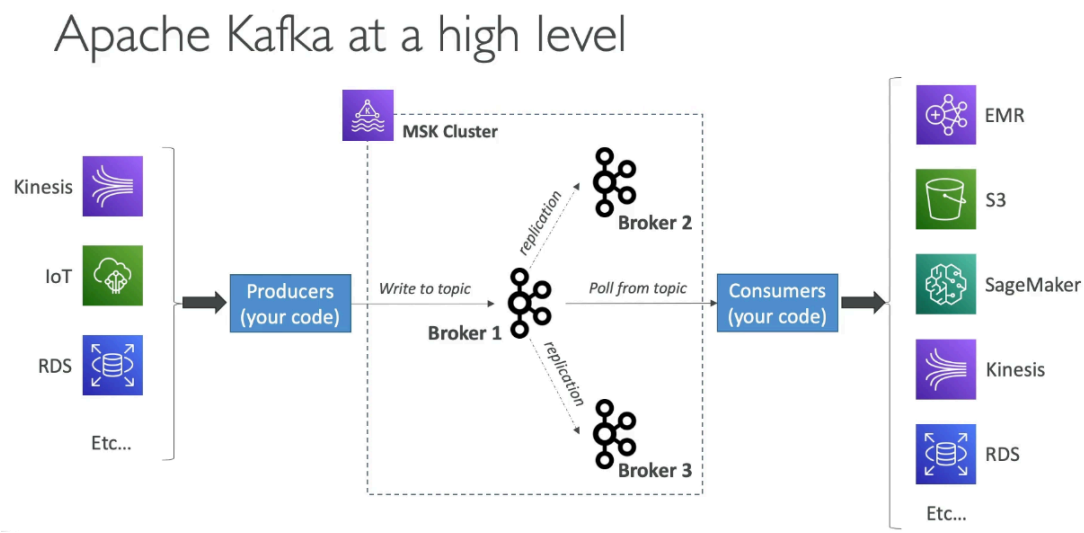
- Use Flink (Java, Scala or SQL) to process and analyze streaming data



- 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)

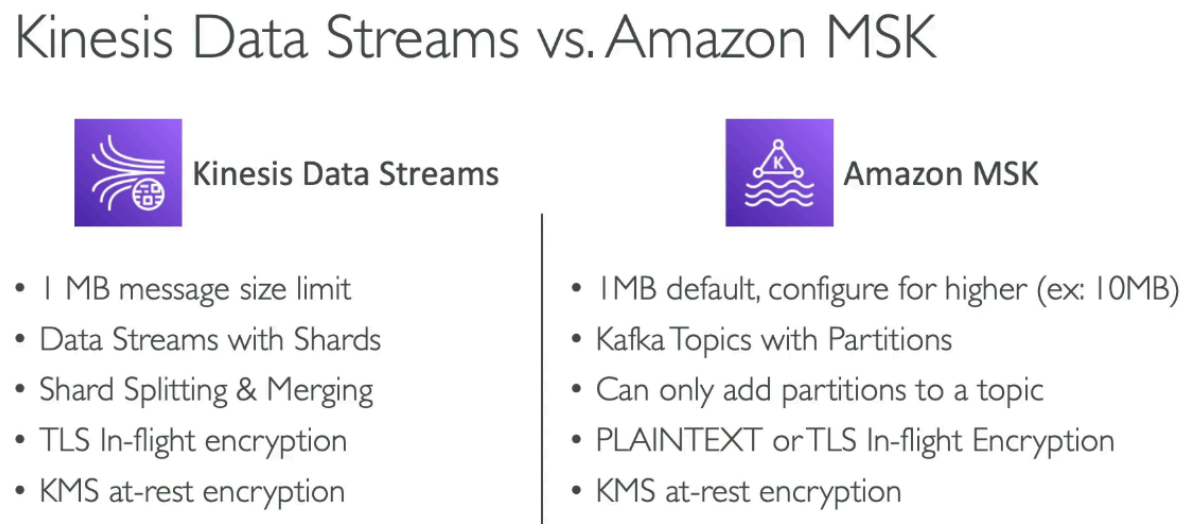
- Java, Scala, SQL to process streaming data
  - Flinks are special apps written as code and can be ran
    - Provisioning of compute resources, parallel computation, automatic scaling
    - Application backups
    - Does not read from Firehose, use SQL instead
- 2 main data sources: Kinesis Data Streams or Amazon MSK

Amazon MSK (Managed Streaming for Apache Kafka)

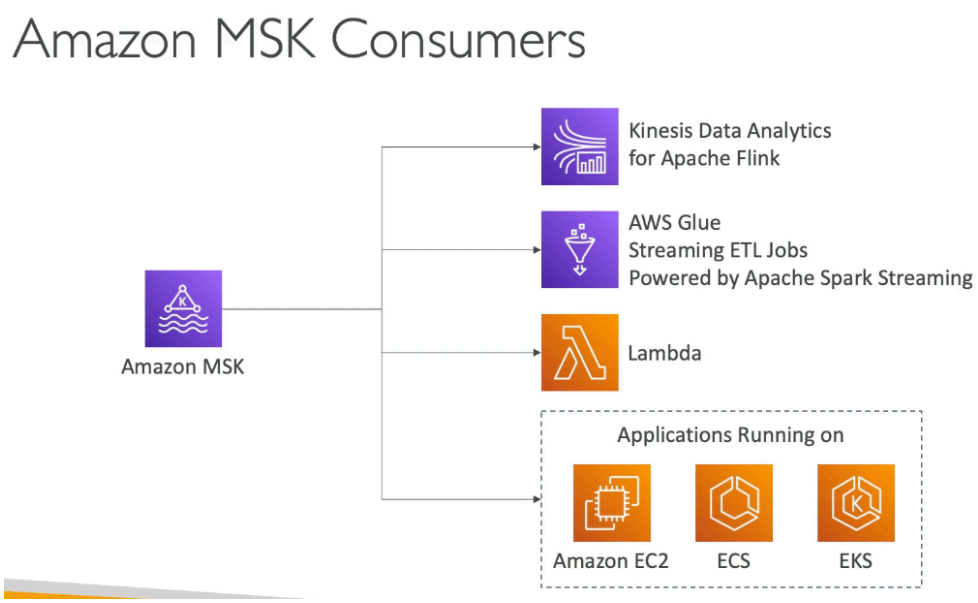


- Alternative to Kinesis, fully managed Apache Kafka on AWS
  - Create, update, delete clusters
  - MSK creates / manages Kafka brokers nodes & Zookeeper nodes for you
  - Deploy MSK cluster in VPC, multi AZ; auto recovery from common failures
  - Data stored on EBS volumes for as long as you want
- MSK Serverless
  - Run Kafka on MSK without managing capacity, where MSK auto provisions resources and scales compute / storage

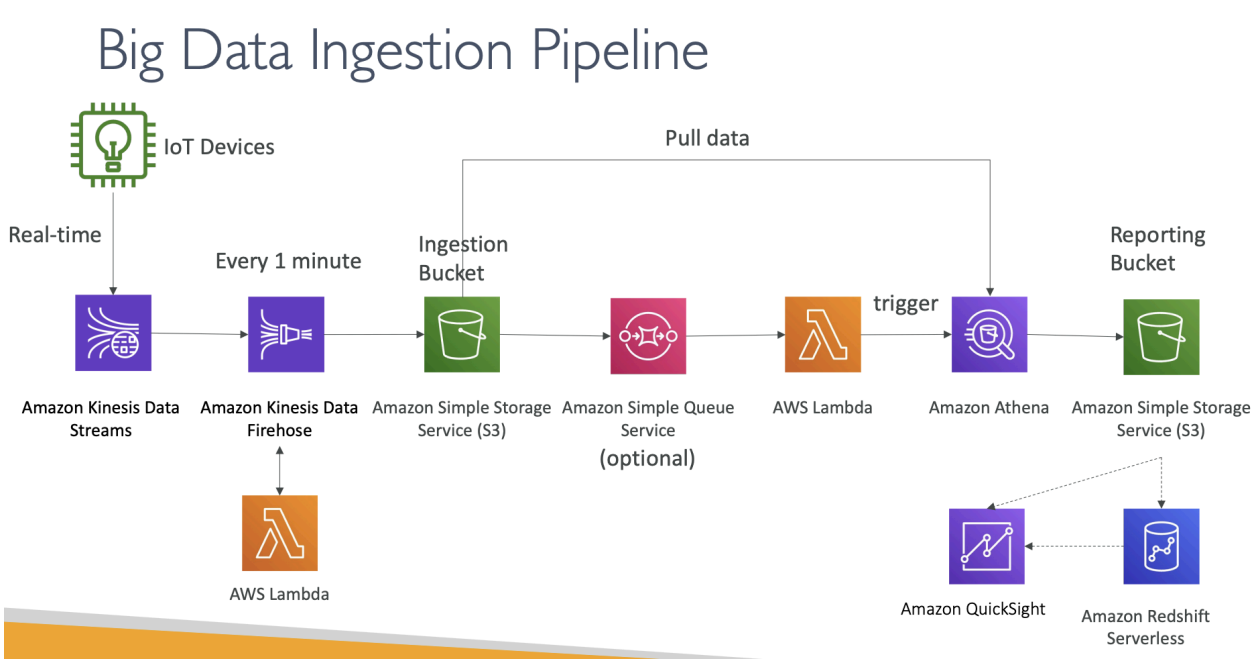
Kinesis Data Stream vs MSK



MSK Consumers



Big Data Ingestion Pipeline



Section 23: Machine Learning

Rekognition Overview

- Find objects, people, text. Scenes in images and videos using ML
    - Facial analysis and search for user verification
    - Create DB of faces or compare against
- Use case: labeling, content moderation, text / face detection / verification, pathing

Content Moderation

- Detecting unwanted content in social media, online…
    - Flag sensitive content for review in Amazon Augmented AI
- Set minimum confidence threshold for items that are flagged
    - Lower means more content shown

Amazon Transcribe

- Speech to text, using deep learning process called automatic speech recognition (ASR) to convert speech to text
    - Auto removes personally identifiable information using Redaction
    - Supports Automatic Language Detection for multi lingual audio
- Use cases: transcribe, closed captioning, generate metadata for media assets to create a fully searchable archive

Amazon Polly

- Text to speech via deep learning

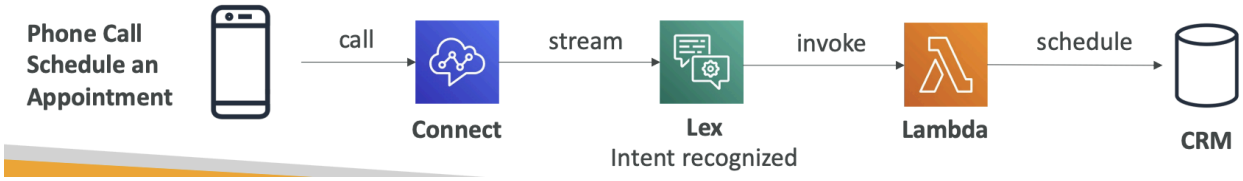
Lexicon & SSML

- Customize pronunciation of words with pronunciation lexicons
    - Upload lexicons and use them in SynthesizeSpeech operation
- Generate speech from plain text or from documents marked up with Speech Synthesis Markup Language (SSML) – more customization
    - Emphasize words or phrases, phonetic pronunciation…

Amazon Translate

- Language translation
    - Can localize content - such as websites and applications - for international users, and to easily translate large volumes of text efficiently

Amazon Lex + Connect



- Lex:
  - Automatic Speech Recognition (ASR) to convert speech to text
  - Natural language understanding
    - Chatbots, call center bots…
- Connect
  - Receive calls, create contact flows, cloud-based virtual contact center
  - Can integrate with other CRM systems or AWS
  - No upfront payments, very cheap

Amazon Comprehend

- NLP, fully managed and serverless
  - ML to find insights and relationships in text
    - Sentiment analysis, extract phrases…
    - Automatically organizes a collection of text files by topic
    - Use cases: sentiment analysis, create and group articles by topic

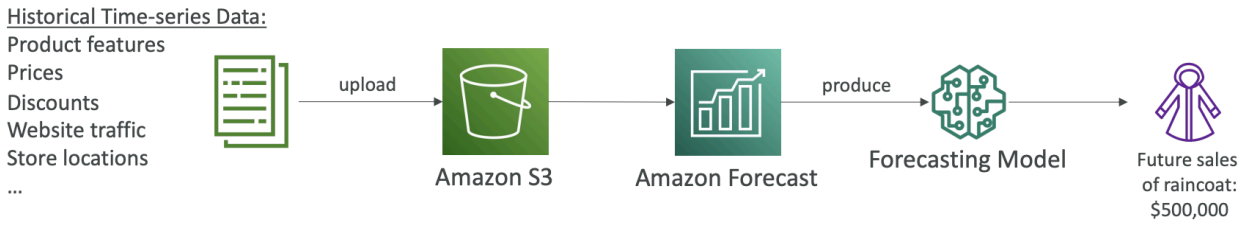
Comprehend Medical

- Detects and returns useful information in unstructured clinical text: doctor notes, test results, etc…
  - Uses NLP to detect protected health info – DetectPHI API
- Store documents in 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

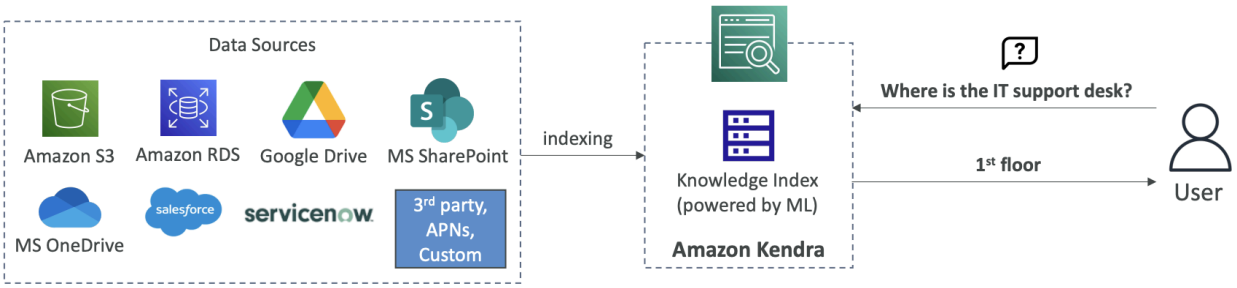
- Fully managed service to build ML models

Amazon Forecast



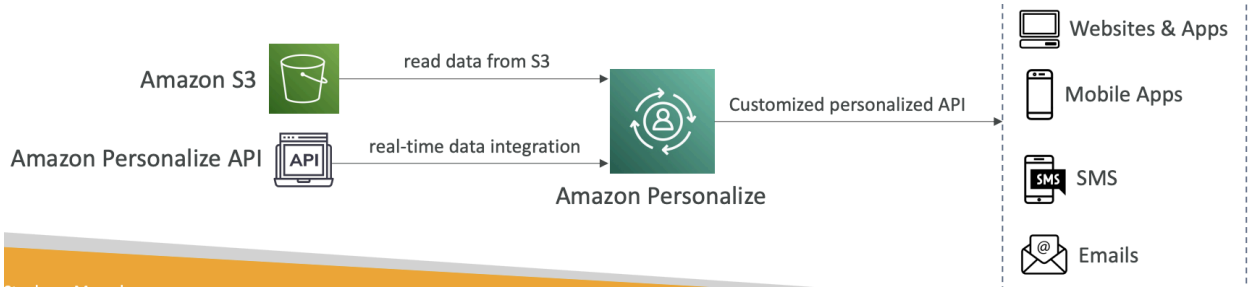
- Fully managed ML to deliver accurate forecasts
    - Reduce forecasting time from months to hours

Amazon Kendra



- Fully managed document search service via ML
    - Extract answers from documents (text, pdf, HTML, PPT, Word…)
    - NLP search capabilities
- Incremental learning: learn from user interactions / feedback to promote preferred results
    - Can manually fine tune search results

Amazon Personalize



- Fully managed ML service to build apps with real time personalized recommendations
    - Example: personalized product recommendations / re-ranking, customized direct marketing

- Integrates into existing websites, applications, SMS, email marketing systems…
    - Implement in days (no need to build, train, deploy ML solutions)

Amazon Textract

- Automatically extract text, handwriting, and data from scanned documents via AI and ML
    - Extra data from forms and tables, read and process any type of documents
    - Use case: financial services, healthcare…

AWS ML Summary

AWS Machine Learning - Summary

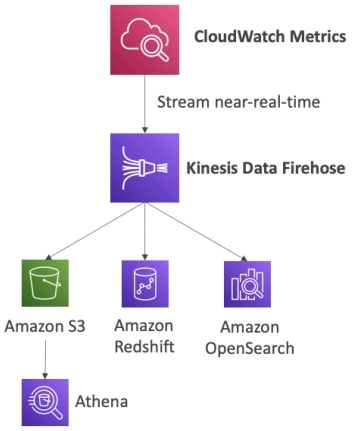
- Rekognition: face detection, labeling, celebrity recognition
- Transcribe: audio to text (ex: subtitles)
- Polly: text to audio
- Translate: translations
- Lex: build conversational bots – chatbots
- Connect: cloud contact center
- Comprehend: natural language processing
- SageMaker: machine learning 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

Section 24: AWS Monitoring & Audit: CloudWatch, CloudTrail & Config

CloudWatch Metrics

- Metrics for every service, can create CloudWatch dashboards
- Metric is a variable to monitor (CPU Utilization, 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

CloudWatch Metric Streams



- Continually stream CW metrics with near real time delivery and low latency
    - Kinesis Firehose, 3rd party…
- Option to filter metrics to only stream subset

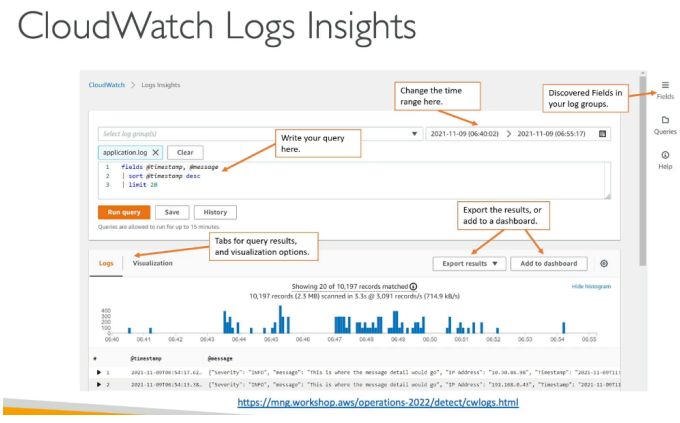
CloudWatch Logs

- Log group: name representing the application
- Log stream: instances within app / log files / containers
- Log expiration policies (never, 10 years)
- Can be sent to S3, Kinesis, Lambda, OpenSearch
- Encrypted by default with KMS based encryption possible

Log Sources

- SDK, CloudWatch Unified Agent, Beanstalk ECS, Lambda, VPC, API GW, CloudTrail, Route 53

Logs Insights

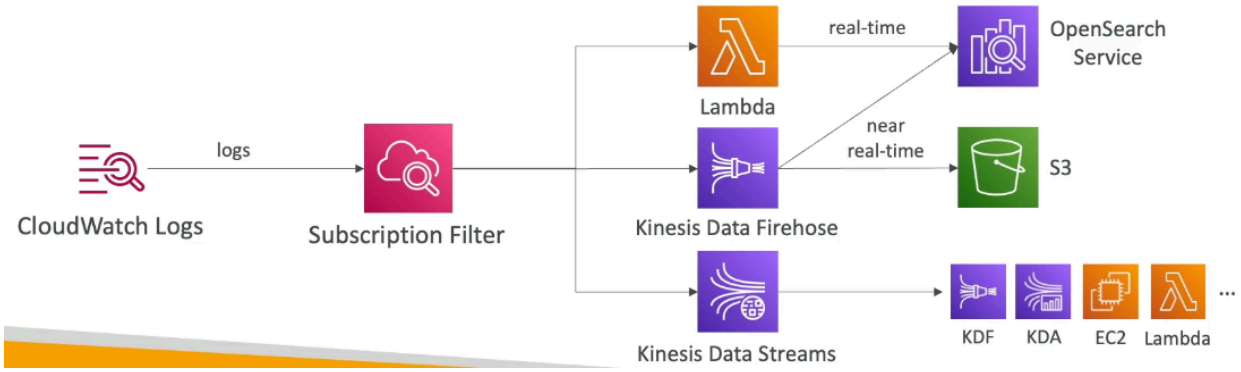


- Can query logs to search and analyze; all fields automatically found, not real time
- Can save queries to CW dashboards and query multiple log groups in different AWS accounts

S3 Export

- Log data can take up to 12 hours to be available for export via CreateExportTask API call, thus not real time

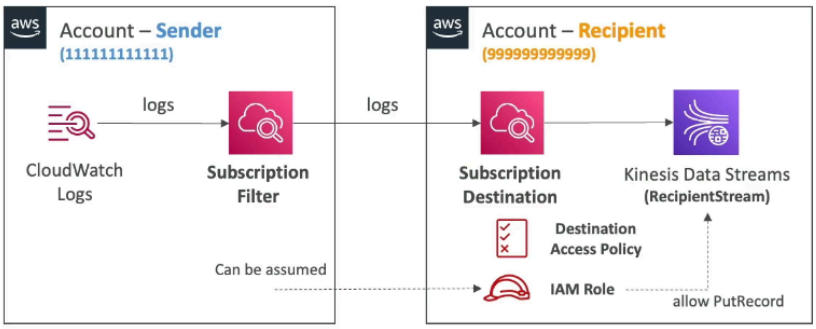
Log Subscriptions



- Real time log events to send to Kinesis or Lambda
- Subscription filter to filter logs delivered to destinations

CloudWatch Logs Subscriptions

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



IAM Role (Cross-Account)

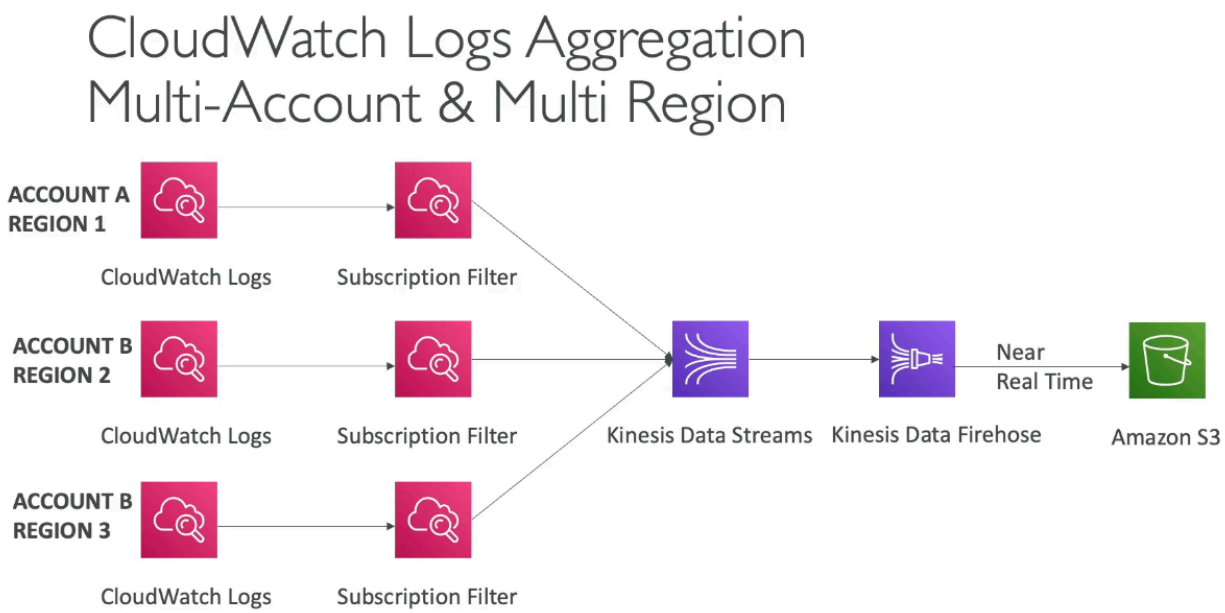
```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:us-east-1:
999999999999:stream/RecipientStream"
    }
  ]
}
```

Destination Access Policy

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

- Cross Account Subscription: sends log events to different AWS account
    - The sender needs a Destination Access Policy to send logs via filter to another account.

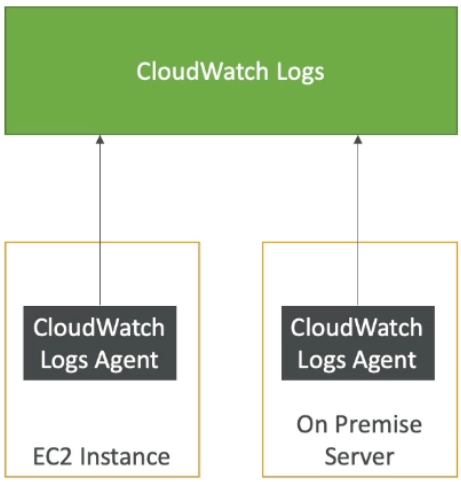
Logs Aggregation



- Can send logs from different accounts in different regions via subscription filter to 1 account (to Kinesis)

CloudWatch Agent & CloudWatch Logs Agent

CW Logs for EC2



- By default no logs go from EC2 to CloudWatch. A CW agent on EC2 will push log files (need IAM permissions); can be on premise too

CW Logs Agent & Unified Agent

- For VMs
- CW Logs Agent
    - Older and can only send to CW logs
- Unified Agent
    - Collect additional system metrics and send to CW
        - CPU, Disk metrics, RAM, etc…
        - More granularity
    - Centralized configuration using SSM parameter store

CloudWatch Alarms

- Trigger notifications for any metric based on sampling, %, max, etc…
- Alarm States: ok, insufficient data, alarm for a period of time (can be high resolution in seconds)
    - High resolution alarm can be 10 or 30 second intervals, where regular alarms are 1 minute multiples
- Alarm Targets
    - Stop, terminate, reboot, etc… EC2 instance, trigger auto scaling, SNS notification
- Alarms can be created based on Log metric filters
- Test alarms via CLI set-alarm-state call

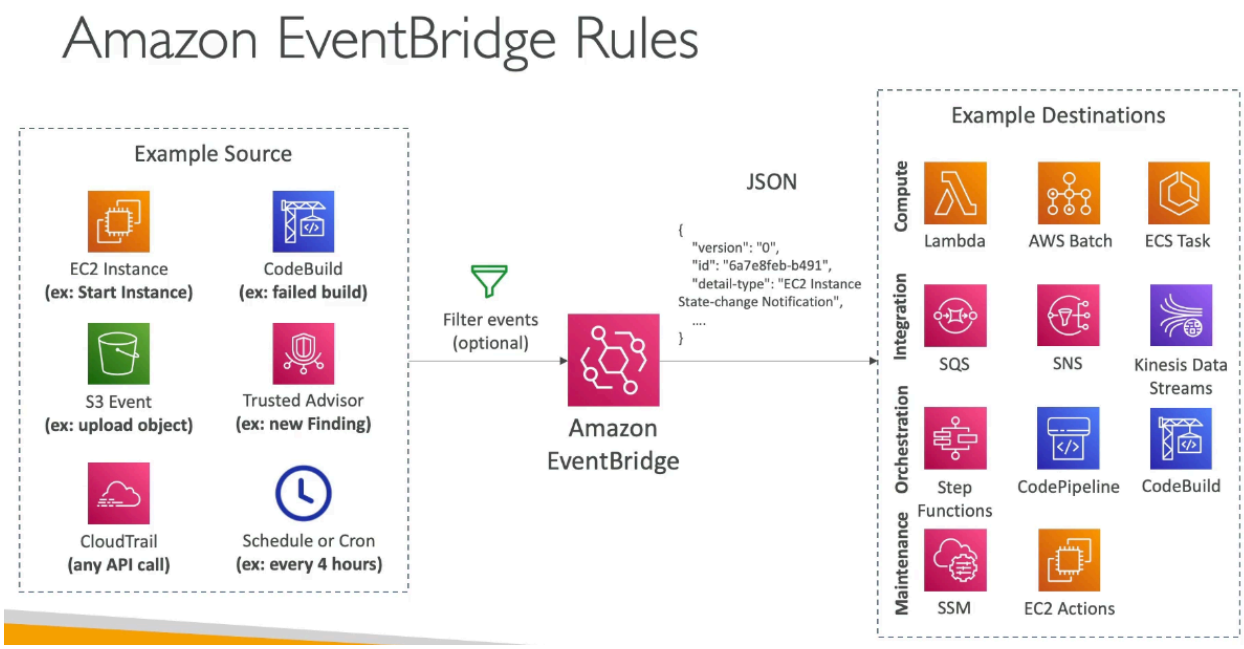
Composite Alarms

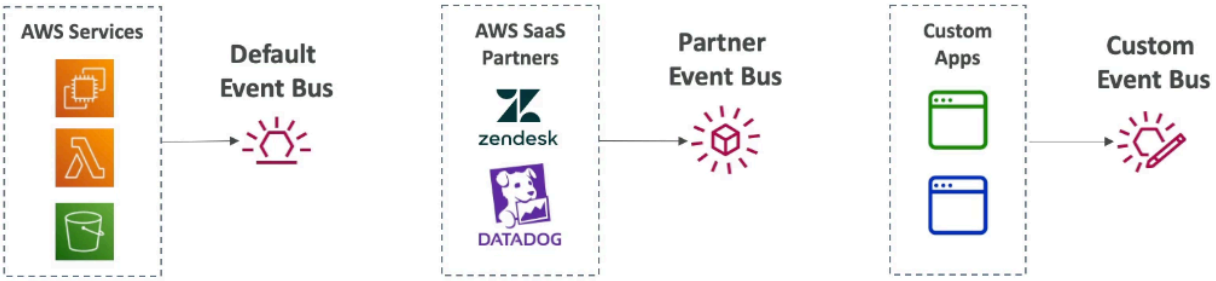
- Alarms are single metric, but composite alarms monitor states of multiple other alarms via AND and OR conditions
    - Reduces alarm noise

EC2 Instance Recovery

- Status check via instance status or system status and can recover
    - Recovery will have same private, public, elastic IP, metadata, placement group

EventBridge





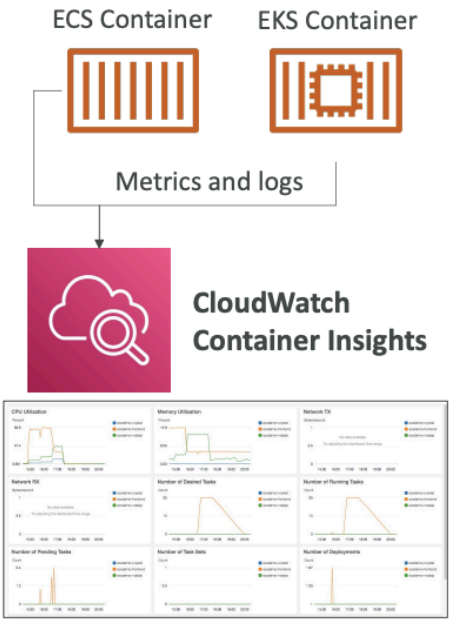
- Default event bus or partner event bus with either AWS services or 3rd party
    - Custom event bus for custom rules
    - Event buses accessed by other AWS accounts using resource based policies
- Schedule cron jobs (scheduled scripts), event pattern (event rules to react to a service doing something), trigger lambda functions, send SNS/SQS messages
    - Archive events (all/filter) sent to event bus (indefinitely or set period) to replay archived events

Scheme Registry

- EventBridge can analyze events in your bus and infer the schema
- Schema registry allows you to generate code for ap to know in advance how data is structured in event bus
    - Can be versioned

Resource Based Policy

- Manage permissions for specific event bus, for example allow events from another AWS account or region → typically to aggregate all events from AWS Organization into 1 account



CloudWatch Container Insights

- Collect, aggregate, summarize metrics and logs from containers
    - ECS, EKS, K8 on EC2, Fargate
- In EKS and Kubernetes, CloudWatch Insights uses containerized version of CW Agent to discover containers

CloudWatch Lambda Insights

- Monitoring and troubleshooting for 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 provided as lambda layer

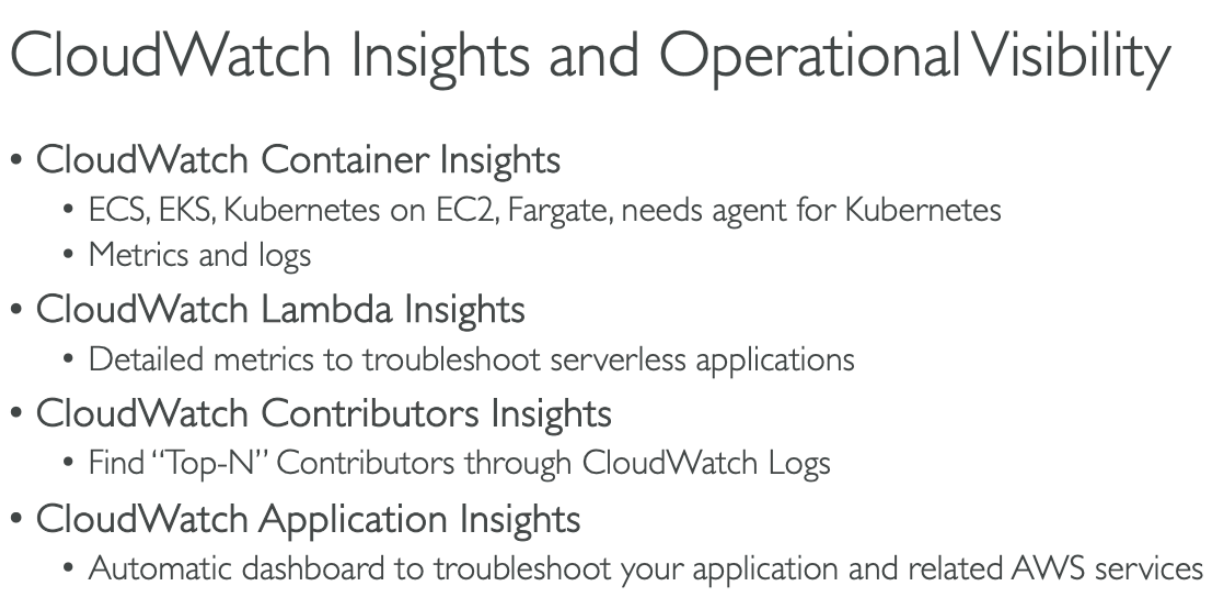
CloudWatch Contributor Insights

- Analyze log data and create time series that display contributor data
    - See metrics about top N contributors
    - Total number of unique contributors and usage
- Find top talkers and understand who or what is impacting system performance
- Works for any AWS generated logs
- Build rules from scratch or use sample rules that AWS has created – leverages your CloudWatch Logs
    - CloudWatch also provides built-in rules that you can use to analyze metrics from other AWS services

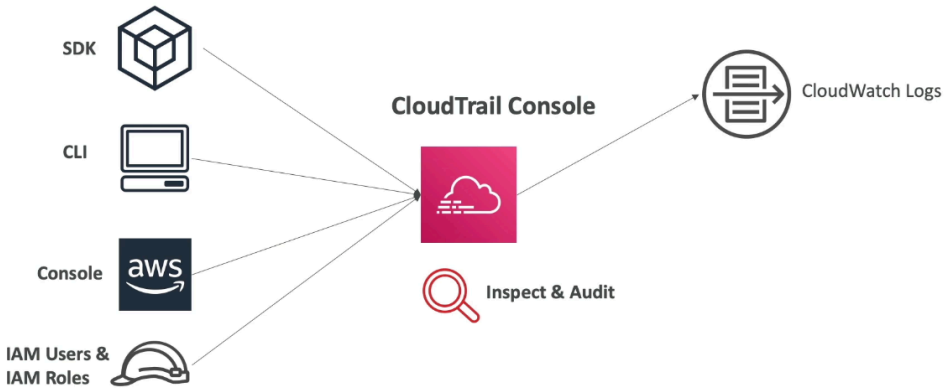
CloudWatch Application Insights

- Provide automated dashboard to show potential problems with monitored applications to help isolate ongoing issues
    - 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 EventBridge and SSM OpsCenter

CloudWatch Insights and Operational Visibility Summary

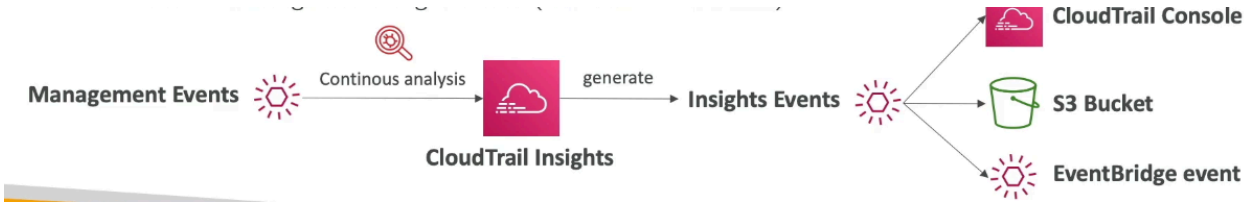


CloudTrail



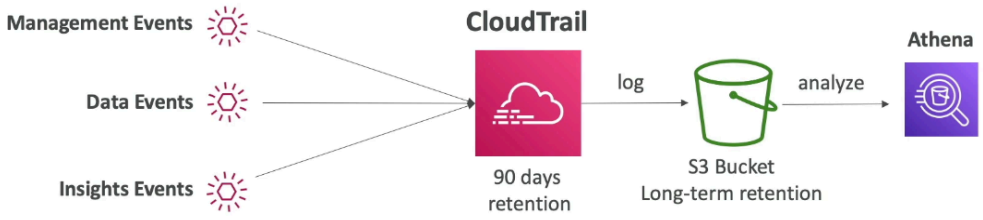
- Governance, compliance for AWS account (enabled by default)
- Gets history of events / API calls made within AWS account where CloudTrail can be sent into CloudWatch or S3
    - Trail applies to all regions by default or single region

CloudTrail Events



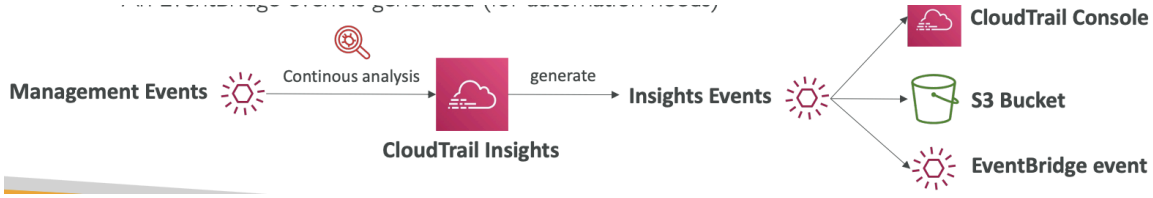
1. Management events: operations performed on resources
    a. Configuring security, logging, etc…
    b. Default trails configured to log management events
    c. Can separate read (no modification) and write events (modify resources)
2. Data events
    a. By default data events not logged (high volume operations)
        i. S3 object level activity, can separate read and write events
        ii. AWS Lambda execution activity (invoke API)
3. CloudTrail Insights Events:
    a. Enable to detect unusual activity in account
        i. Inaccurate resource provisioning, hitting service limits, IAM actions, etc…
    b. Analyzes normal management events for a baseline and continuously analyze write events for unusual patterns
        i. Anomalies show in CloudTrail and event sent to S3 and EventBridge

Retention



- Stored for 90 days by default, but to keep beyond use S3 and use Athena

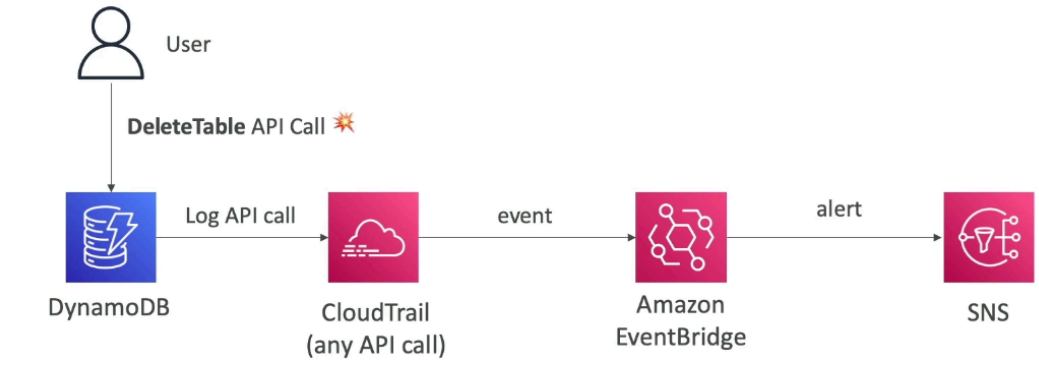
CloudTrail Insights

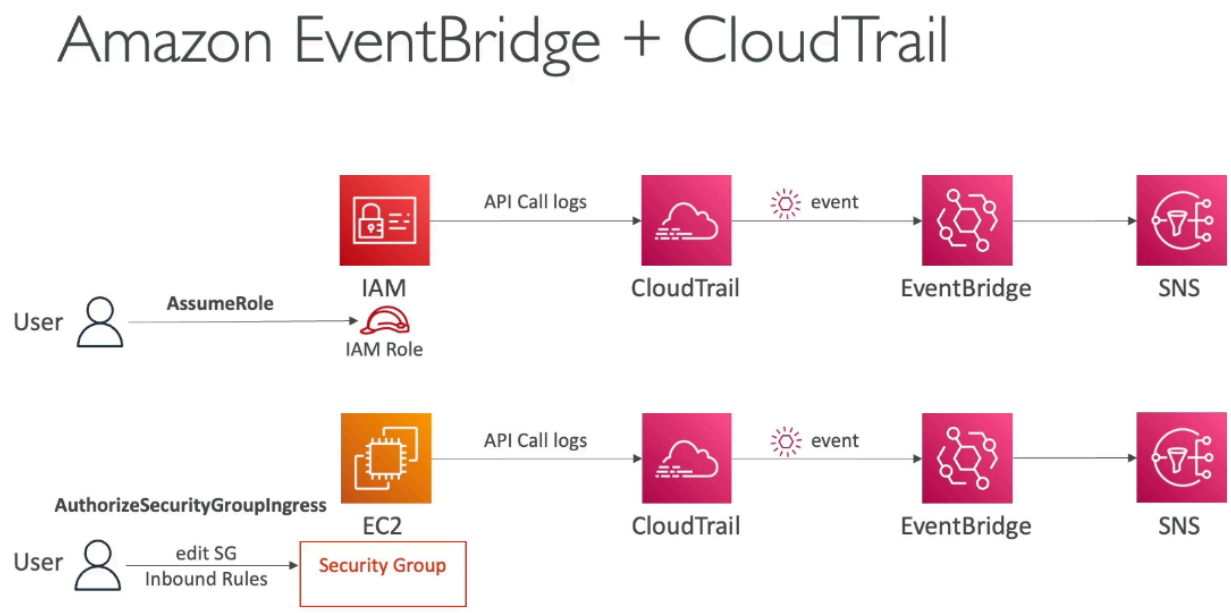


- Enable to detect unusual activity in account
    - Inaccurate resource provisioning, hitting service limits, bursts of IAM actions…
- Analyze normal management events as a baseline and analyze write events to detect unusual patterns
    - Anomalies appear in CloudTrail console
    - Event sent to S3 and EventBridge event is generated

CloudTrail + EventBridge

- Intercept API calls

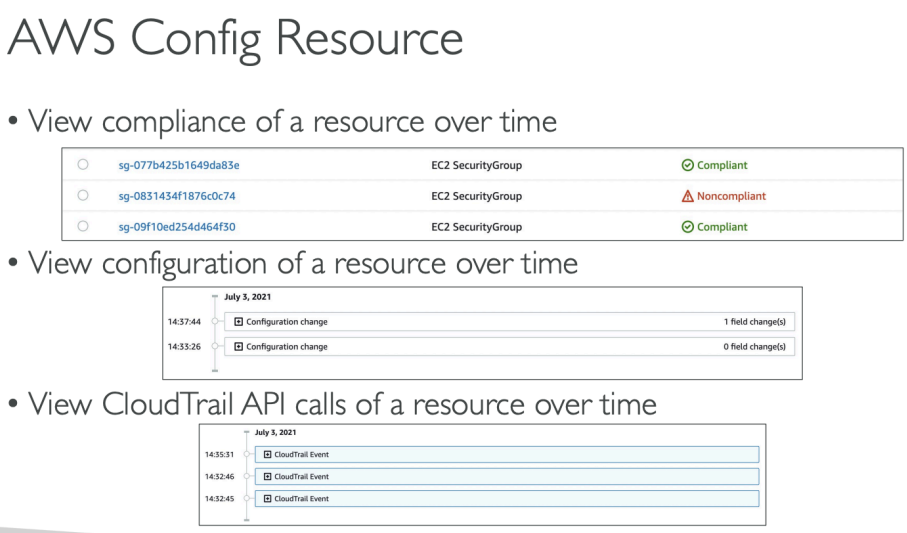




AWS Config

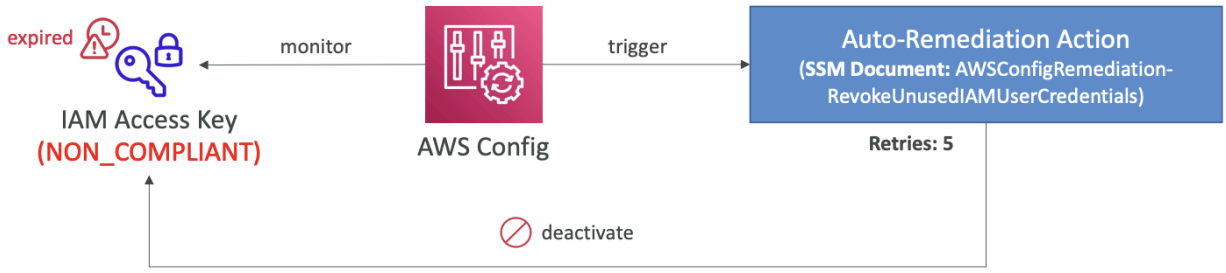
- Helps with auditing and recording compliance of AWS resources
    - Helps record configuration changes over time
        - Any buckets with public access? Has ALB configuration changed over time?
- Receive alerts (SNS) for any changes
- Per region service, can be aggregated across regions and accounts
- Can store data in S3

Config Rules



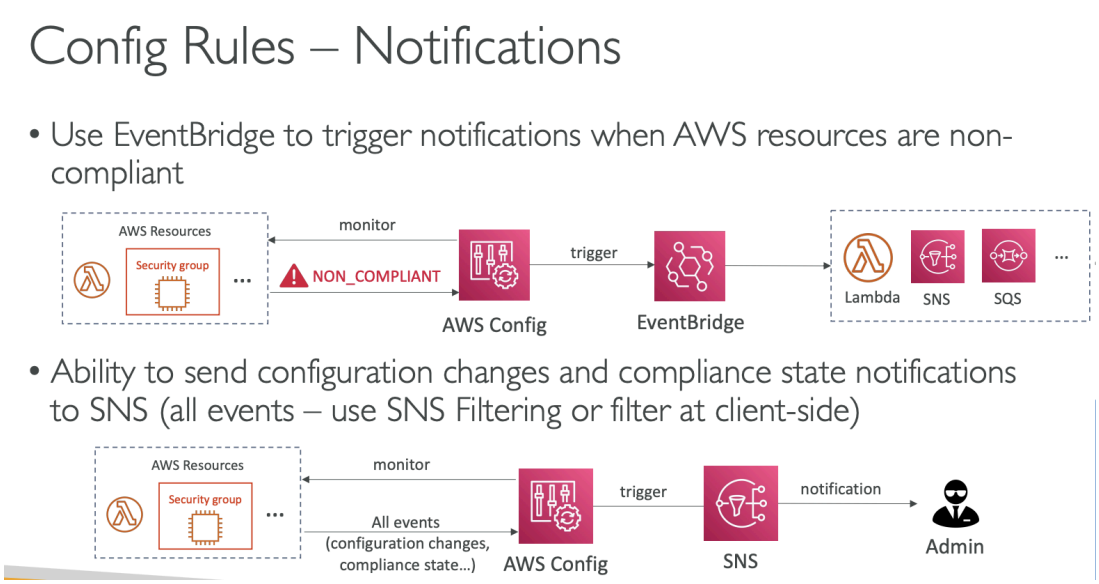
- AWS Managed rules
- Custom rules (defined in Lambda)
- Rules can be evaluated / triggered
    - For each config change
    - And / or at regular time intervals
- Config rules do not prevent actions from happening (no deny)
    - Just gives overview and compliance of resources
- Pricing: no free tier

Remediations



- Automate remediation of non-compliant resources using SSM Automation Documents
    - Use AWS managed Automation Documents or custom
        - Can create custom that invokes Lambda function
    - Remediation retries if resource still non-compliant after auto remediation

Notifications



- Use EventBridge for notifications and can use SNS

CloudWatch vs CloudTrail vs Config

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

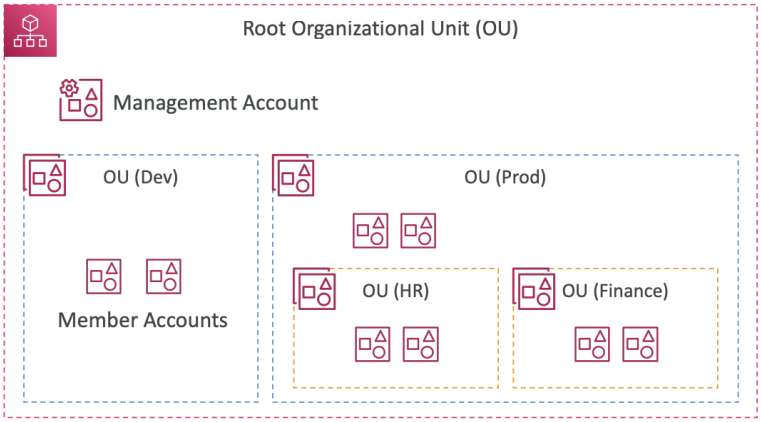
For an Elastic Load Balancer

- 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

Section 25: IAM – Advanced

AWS Organizations

AWS Organizations



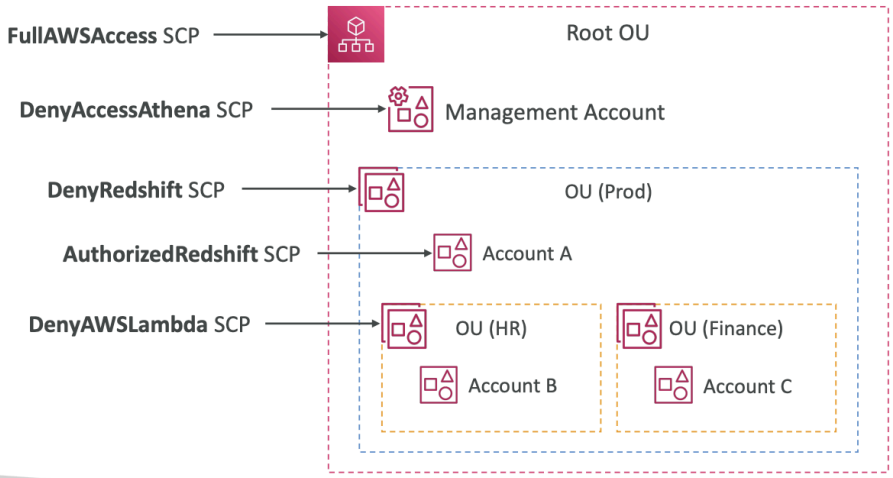
- Global service to allow management of multiple AWS accounts
    - Main account is management account, other accounts are member accounts
        - Members can be part of only 1 organization
- Consolidated Billing across all accounts – single payment with pricing benefits from aggregated usage
- Shared reserve instances and savings plans discounts across accounts
- API available to automate account creation

Advantages

- Multi account vs one account multi VPC
- Tagging standards for billing
- 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

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": "*"
        }
    ]
}
```

- Service Control Policies (SCP)
- IAM policies applied to OU or accounts to restrict users and roles
  - Do not apply to management account
- Must have explicit allow (deny all by default)

IAM Conditions

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", "203.0.113.0/2
                }
            }
        }
    ]
}
```

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

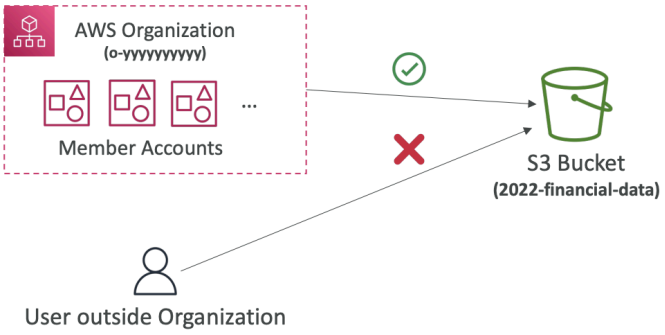
- s3:GetObject, s3:PutObject, s3:DeleteObject applies to arn:awn:s3:::test/*
- => object level permission

```
{
    "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

Resource Policies & aws:PrincipalOrgID

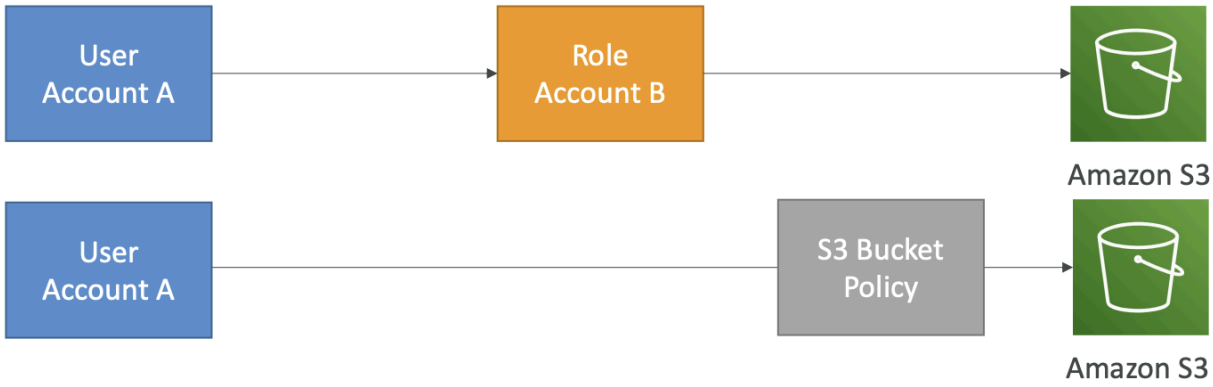
- aws:PrincipalOrgID can be used in any resource policies to restrict access to accounts that are member of an AWS Organization




```
{
    "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"]
                }
            }
        }
    ]
}
```

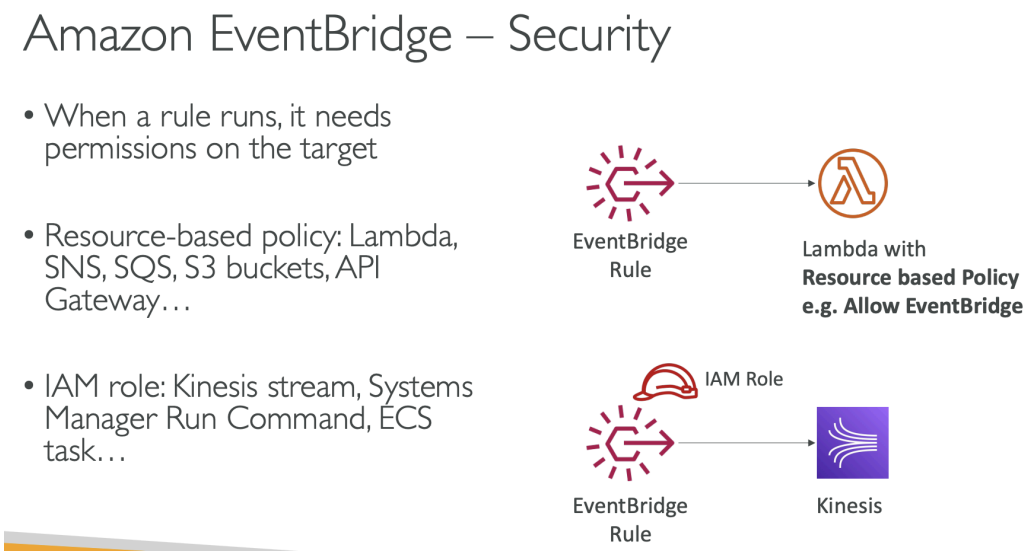
- aws:PrincipalOrgID can be used in any resource policies to restrict access to accounts that are member of AWS org

IAM Roles vs Resource Based Policies



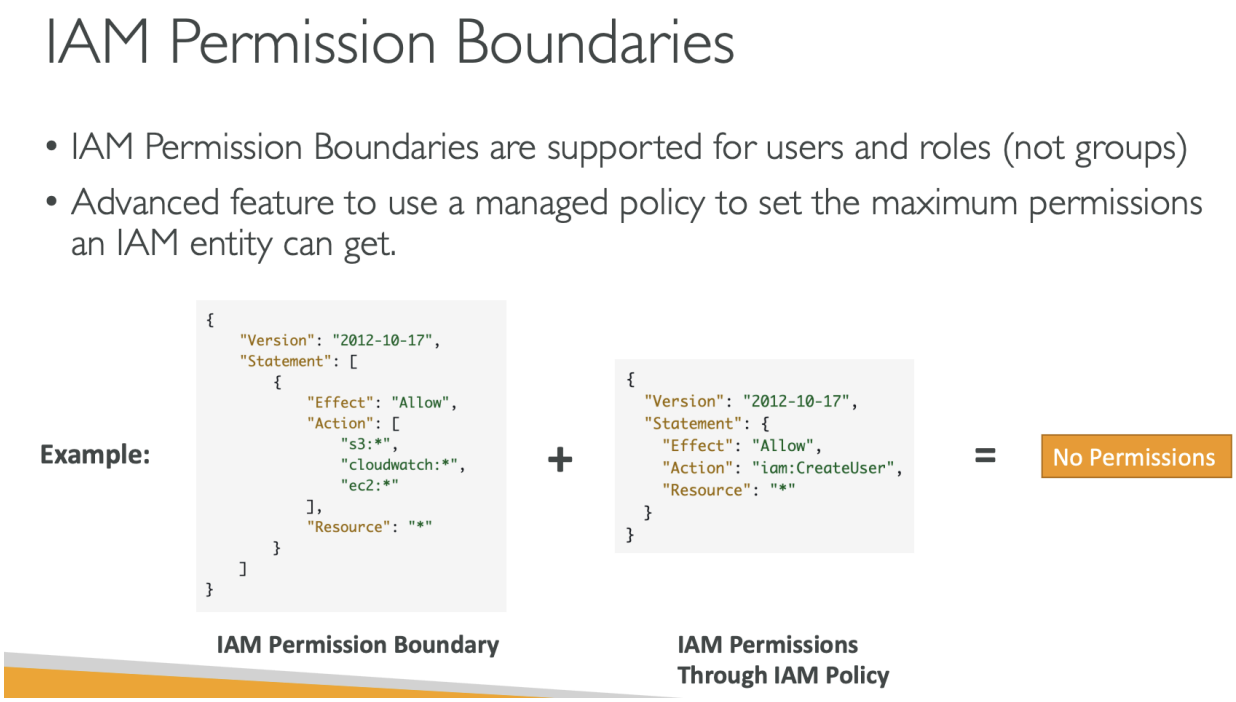
- Cross account:
  - Attach resource based policy to resource or use a role as proxy
- When you assume a role (user, application or service), you give up original permissions and take permissions assigned to the role
- When use a resource based policy, the principal does not give up permissions
  - Example: User in account A needs to scan a DynamoDB table in Account A and dump it in an S3 bucket in Account B

EventBridge – Security



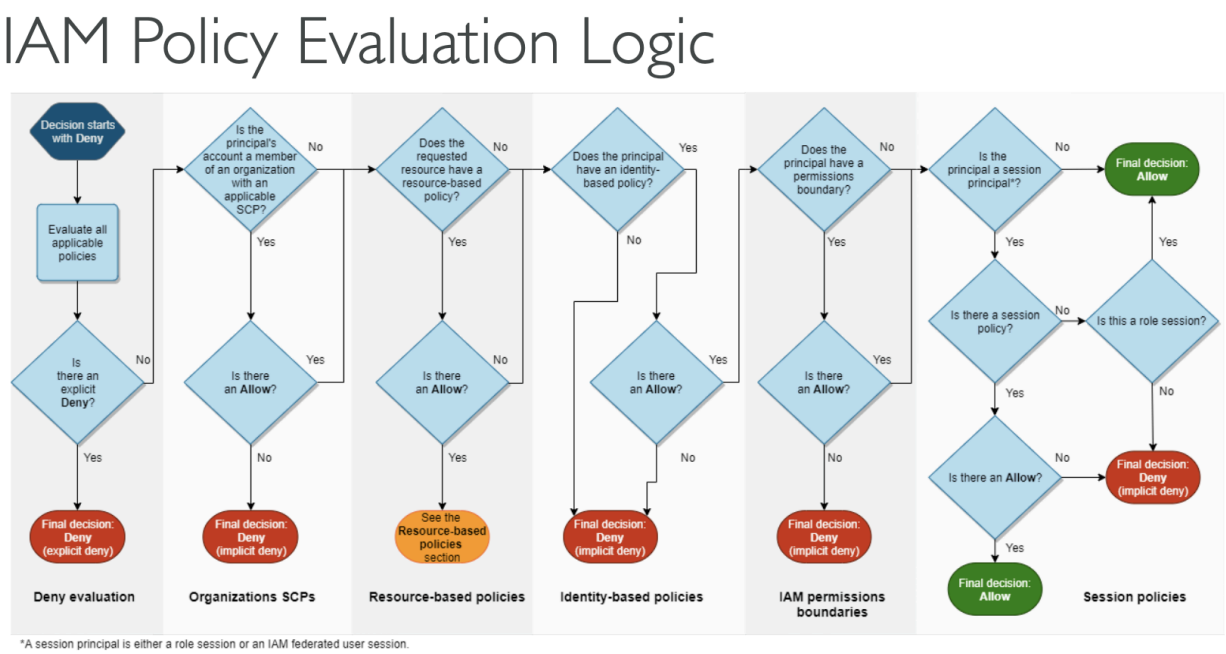
- When a rule runs, it needs permissions on target
- Resource based policy: lambda, S3, API GW…
- IAM Role: Kinesis, ECS task…

IAM Permission Boundaries

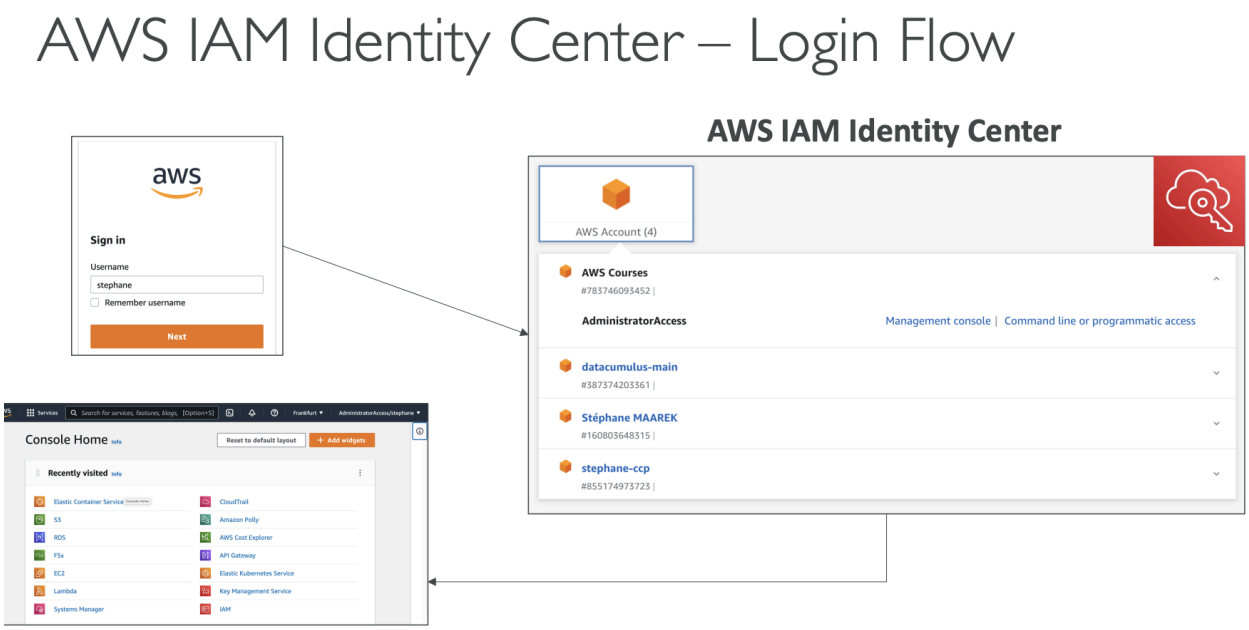


- Supported for users and roles (not groups)
    - Can be used in combinations of AWS Org SCP
- Advanced feature to use a managed policy to set the maximum permissions an IAM entity can get
    - In example, user cannot create user because the max permissions are in the permissions boundary
- Use cases:
    - 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



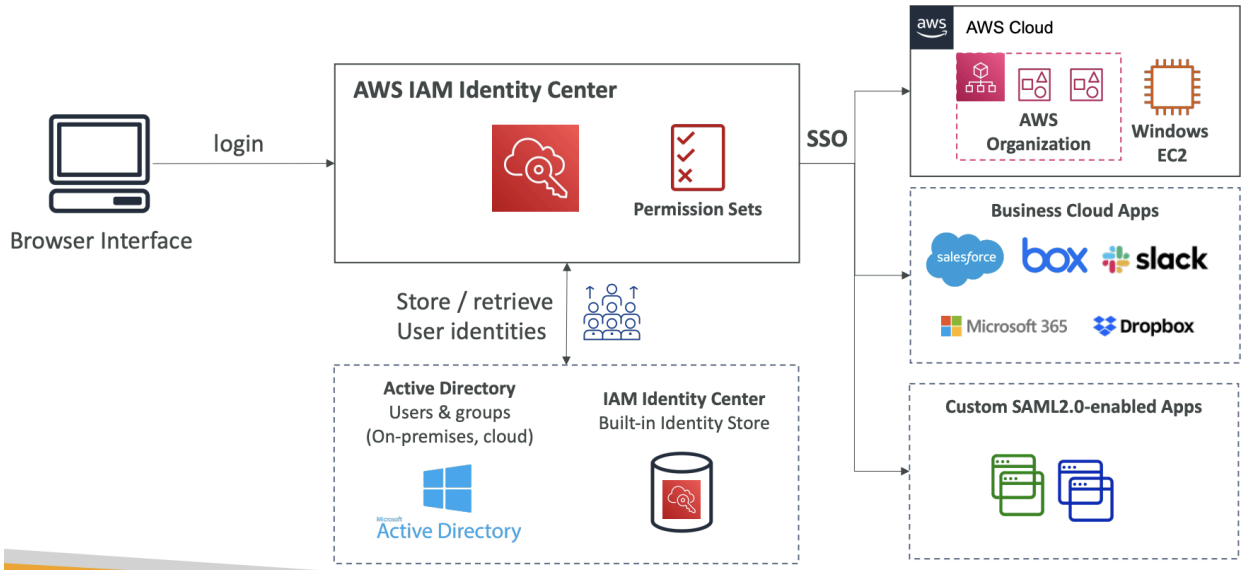
AWS IAM Identity Center



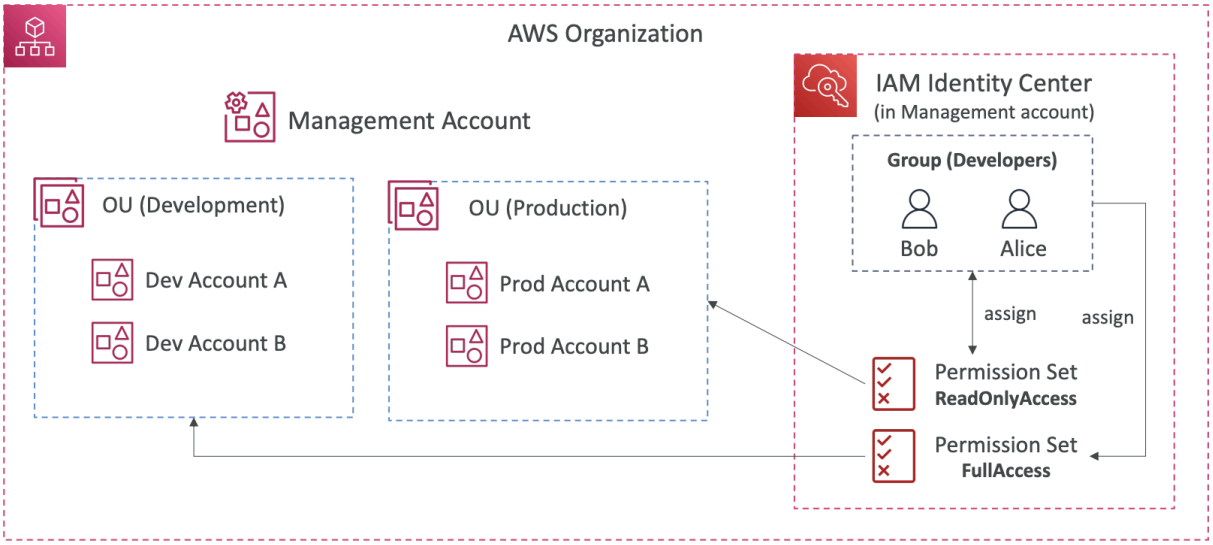
- 1 login SSO for:
  - AWS accounts in AWS Orgs
  - Business cloud applications

    - SAML 2.0 applications
    - EC2 Windows instances
- Identity Provider
    - Built in identity store in IAM Identity Center
    - 3rd party: Active Directory, Okta…

AWS IAM Identity Center



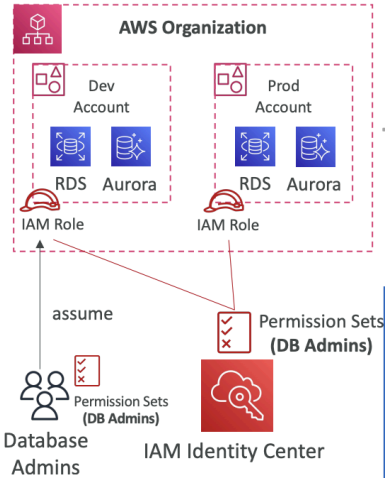
IAM Identity Center



Fine Grained Permissions and Assignments

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

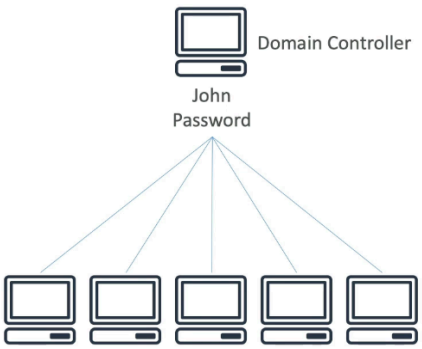


- Multi Account Permissions
  - Managed access across AWS accounts in AWS Org
  - Permission sets – collection of 1+ IAM policies assigned to users and groups to define AWS access
    - Automatically create IAM role for users
- Application Assignments
  - SSO access to many SAML 2.0 business apps
  - Provide required URLs, certificates, metadata
- Attribute based access control (ABAC)
  - Fine grained permissions based on users' attributes stored in IAM Identity Center Identity Store
    - Ex: cost center, title…
  - Use case: define permissions once, then modify AWS access by changing the attributes

AWS Directory Services

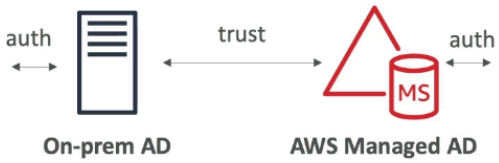
Microsoft Active Directory (AD)

- DB of objects: user accounts, computers, printers, etc… with centralized security management
  - Objects are organized in trees and a group of trees is a forest



AWS Managed Microsoft AD

- Create own AD in AWS, manage users locally, supports MFA
- Establish trust connections with on premise AD



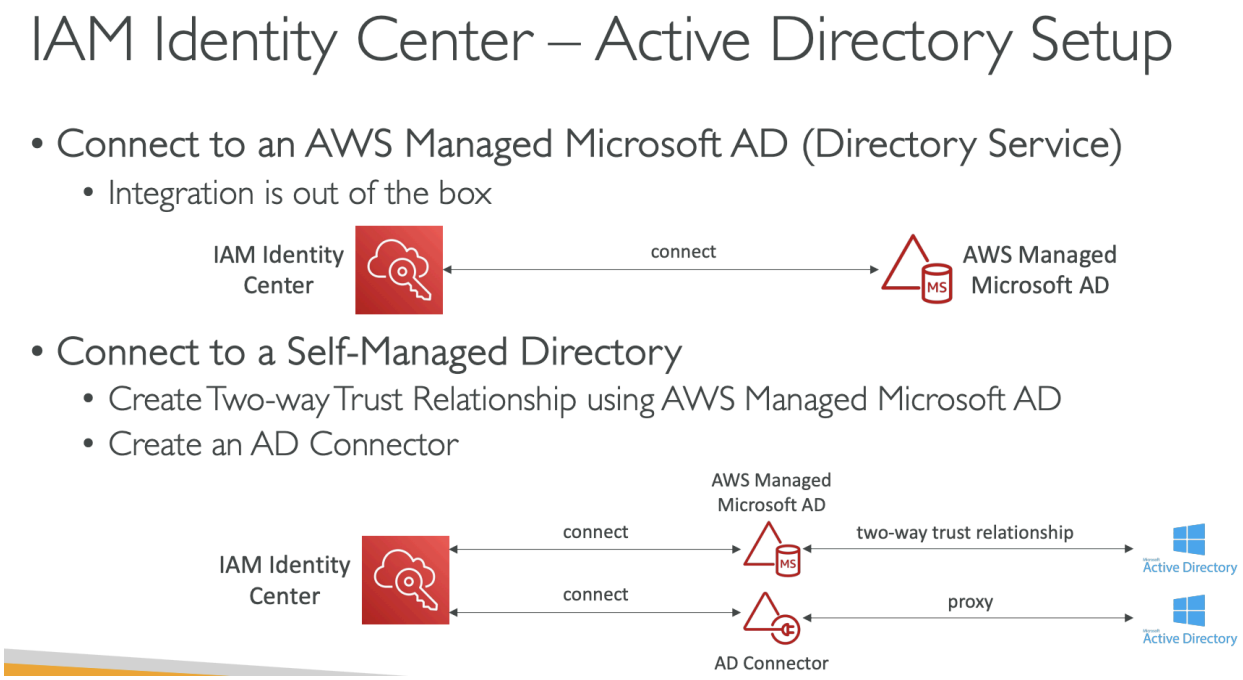
AD Connector

- Directory Gateway (proxy) to redirect to on premise AD, supports MFA
  - Users are managed on on premise AD

Simple AD

- AD compatible managed directory on AWS, cannot be joined with on premise AD

Active Directory Setup

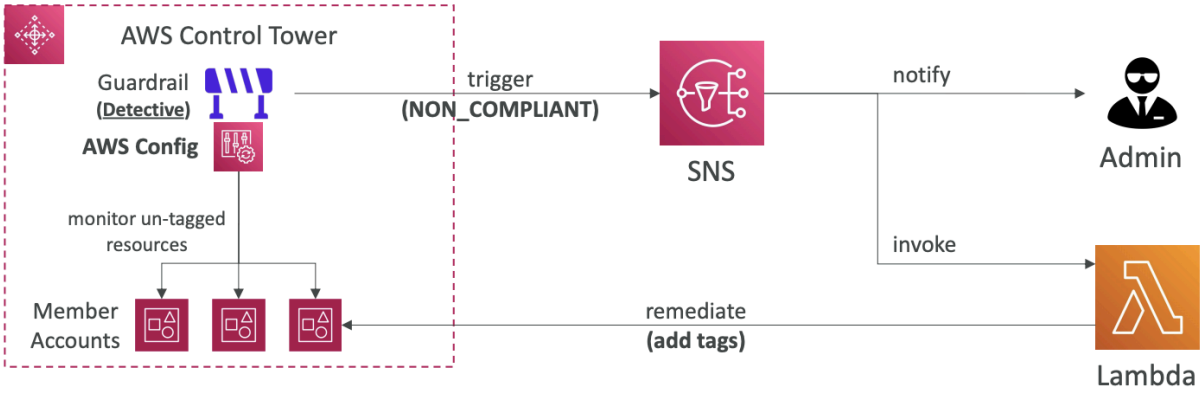


- Connect to AWS Managed Microsoft AD
  - Integration out of the box
- Connect to self managed directory
  - Create 2 way trust relationship using AWS Managed Microsoft AD
  - Create AD connector

AWS Control Tower

- Easy way to set up and govern a secure and compliant multi account AWS environment
    - Use AWS Organizations to create accounts
- Benefits:
    - Automate environment set up
    - Automate ongoing policy management using guardrails
    - Detect policy violations and remediate
    - Monitor compliance through interactive dashboard

Guardrails

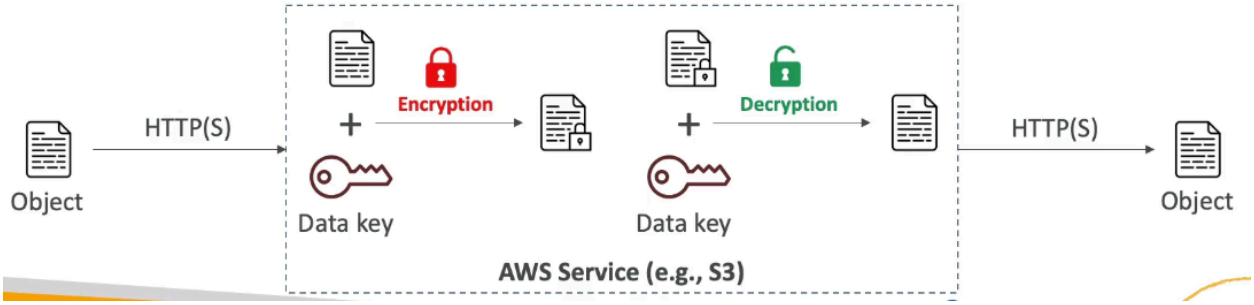


- Provides ongoing governance for your Control Tower environment (AWS Accounts)
- Preventive Guardrail – using SCPs
    - Ex: restrict regions across all accounts
- Detective Guardrail – using AWS Config
    - Ex: identity untagged resources

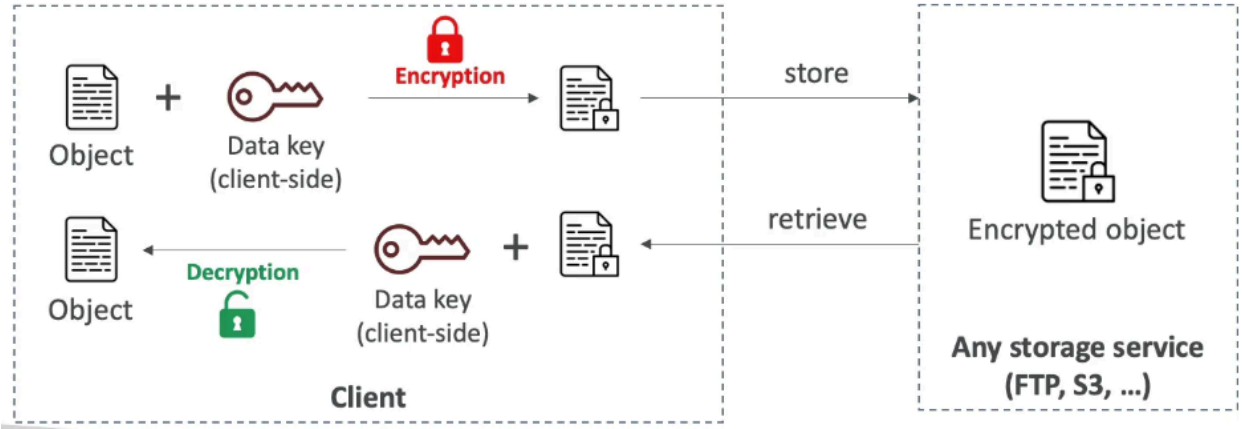
Section 26: AWS Security & Encryption: KMS, SSM Parameter Store, Shield, WAF

Encryption 101

- Encryption in flight (TLS / SSL for HTTPS encryption)
    - Data encrypted before sending and decrypted after receiving → only target server can receive it; ensures no middle man attack occurs



- Server Side encryption at rest
  - Data is encrypted after received by the server and decrypted before being sent
  - Stored in encrypted form via (data) key where the encryption / decryption keys must be managed somewhere and the server needs access to it



- Client side encryption
  - Data is encrypted by the client and never decrypted by the server (server not trusted and cannot decrypt data)
  - Data will be decrypted by a receiving client
  - Could leverage envelope encryption

KMS

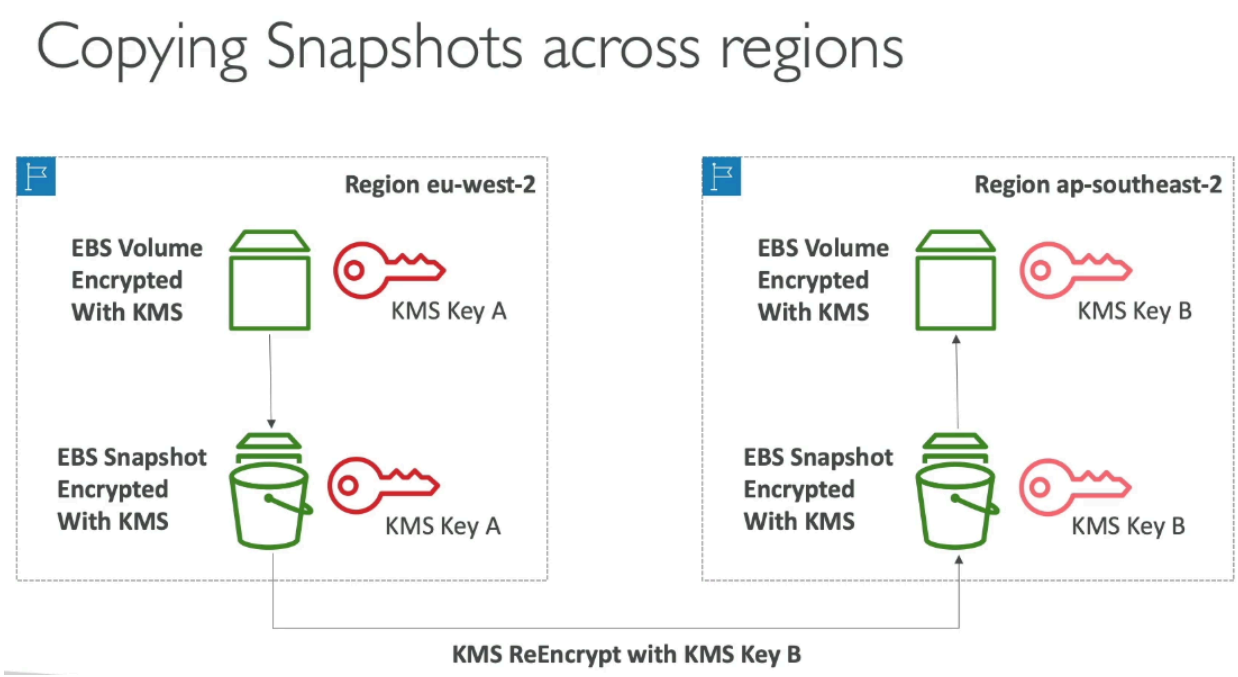
- AWS managed encryption keys, fully integrated with IAM for authorization and easy ways to control access to data; can audit with CloudTrail and integration with AWS services
  - Scoped per region, the same KMS key cannot be in the same region
- KMS key types:
  - AWS owned → default free key for SSE-S3, SSE-SQS, SSE-DDB
  - AWS managed key → free for aws/service name
  - Customer managed key created in KMS → $1 /month
  - Customer managed imported → $1 /month
  - + pay for API call to KMS

- Automatic Key rotation
    - AWS managed: auto rotate every 1 year
    - Customer managed KMS: enabled feature for automatic rotation 1 year
    - Imported: only manual rotation using alias

Key Types

- Symmetric (AES-256)
    - Single encryption key for encryption / decryption that all AWS services integrate with
    - Never get access to KMS key unencrypted
- Asymmetric (RSA & ECC Key Pairs)
    - Public (encrypt) and private (decrypt) key pair for sign / verify
        - Public key is downloadable, but can't access private key unencrypted
    - Used for encryption outside of AWS by users who can't use KMS

Copying Snapshots across Regions



Key Policies

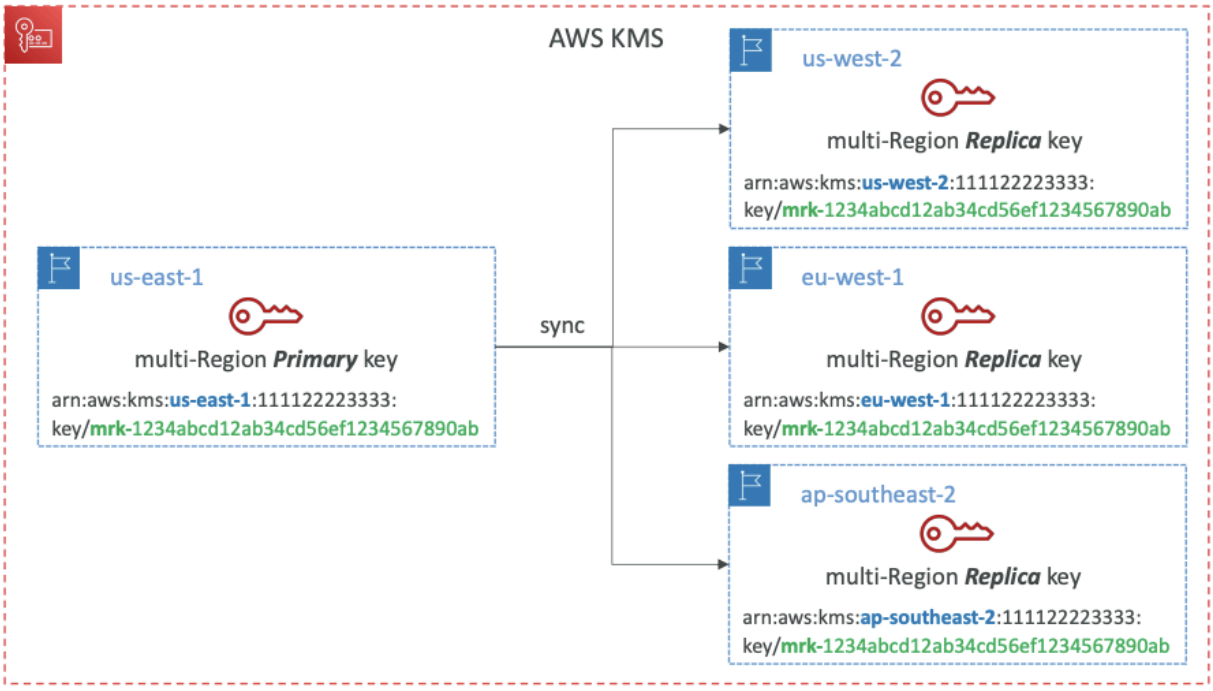
- Control access to KMS keys similar to S3 bucket policy
    - Difference: cannot control access without them
        - If there is no policy, no one can access the KMS key
- Default KMS policy:
    - Created if you don't provide a key policy
    - Complete access to the key to root user = entire AWS account

- Custom Key Policy:
    - Define users, roles access to key and define who can administer the key
    - Useful for cross account access of KMS key

Copying Snapshots across Accounts

1. Create snapshot, encrypted with own KMS key (customer managed)
2. Attach KMS key policy to authorize cross account access
3. Share encrypted snapshot
4. (in target) create a copy of the snapshot, encrypt it with a different customer managed key in account
5. Create a volume from snapshot

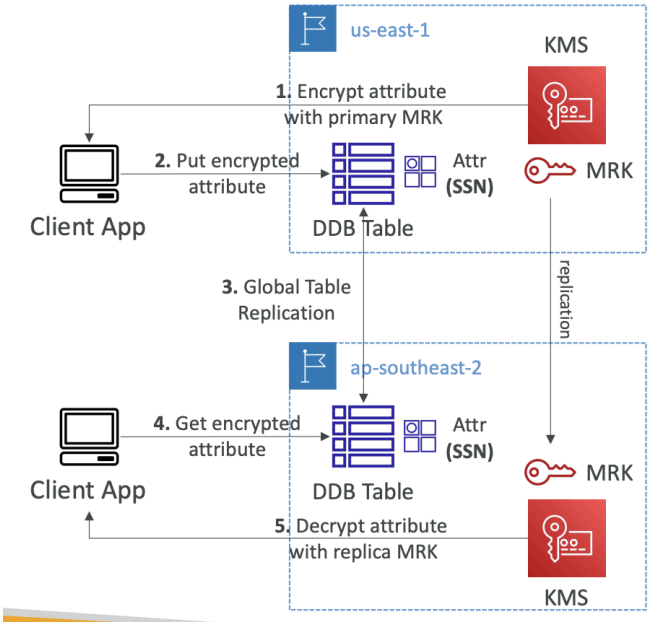
Multi Region Keys



- Identical KMS keys in different regions that can be used interchangeably
- Multi region keys have same key ID, key material, automatic rotation
    - Encrypt in 1 region and decrypt in another
    - No need to re-encrypt or make cross region API calls
- KMS multi region are NOT global (primary + replicas)
    - Each key is managed independently
- Use case: global client side encryption, encryption on global DynamoDB, global Aurora

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

- Can encrypt specific attributes client side in DynamoDB using 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

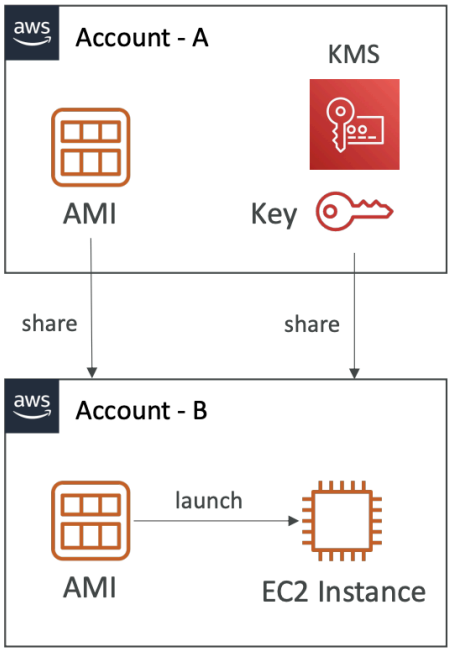
- Encrypt specific attributes client side with AWS Encryption SDK
  - Combine with Aurora Global to have client side encrypted data replicated to other regions
  - If using multi region key, replicated in the same region as Global Aurora DB, then clients can use API calls to KMS in their region to decrypt data client side

S3 Replication Encryption Considerations

- Unencrypted objects and object encrypted with SSE-S3 are replicated by default
- Objects with SSE-C can be replicated
- With SSE-KMS, must enable option
  - Specify which KMS key
  - Adapt KMS key policy for target key
  - IAM role with kms:Decrypt for source KMS key and kms:Encrypt for the target KMS key
    - May get KMS throttling errors
- Can use multi region KMS key, but treated as independent keys by S3
  - Object will still be decrypted and encrypted by the same key even though the key is multi region

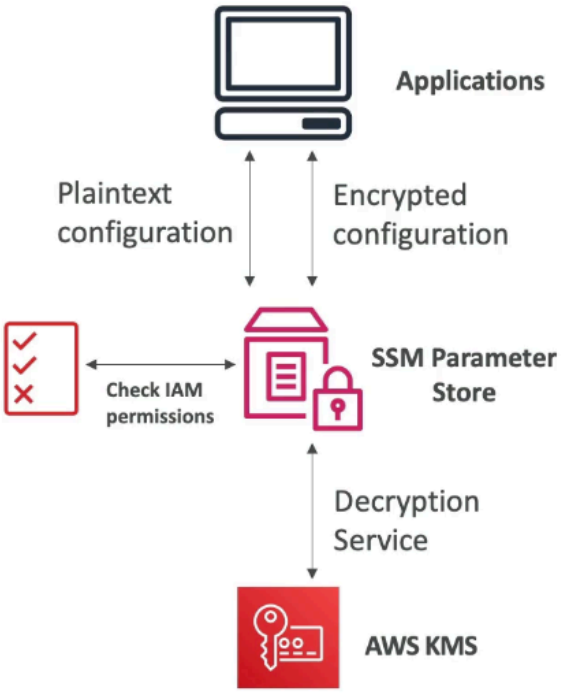
AMI Sharing Process Encrypted via KMS

- AMI in source account encrypted with KMS key from source account
- Must modify the image attribute to add launch permission which corresponds to the specified target AWS account
- Share KMS keys used to encrypt snapshot that the AMI references with the target account / IAM role
- IAM role / user in target account must have permissions to DescribeKey, ReEncrypted, CreateGrant, Decrypt
- When launching EC2 instance from AMI, optionally the target account can re-encrypt AMI with a key of its own



SSM Parameter Store

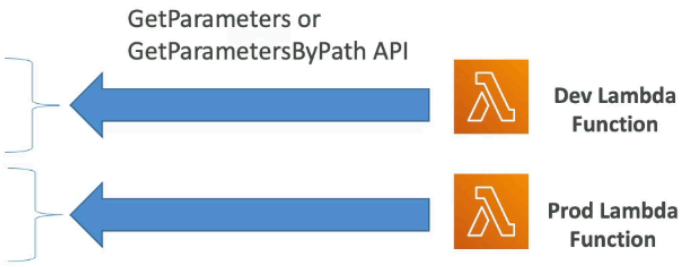
- Secure storage for configuration and secrets with optional encryption with KMS
- Serverless, scalable, durable, easy SDK with version tracking of configurations / secrets
- Security via IAM and EventBridge notifications; integration with CloudFormation



Parameter Store Hierarchy

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)



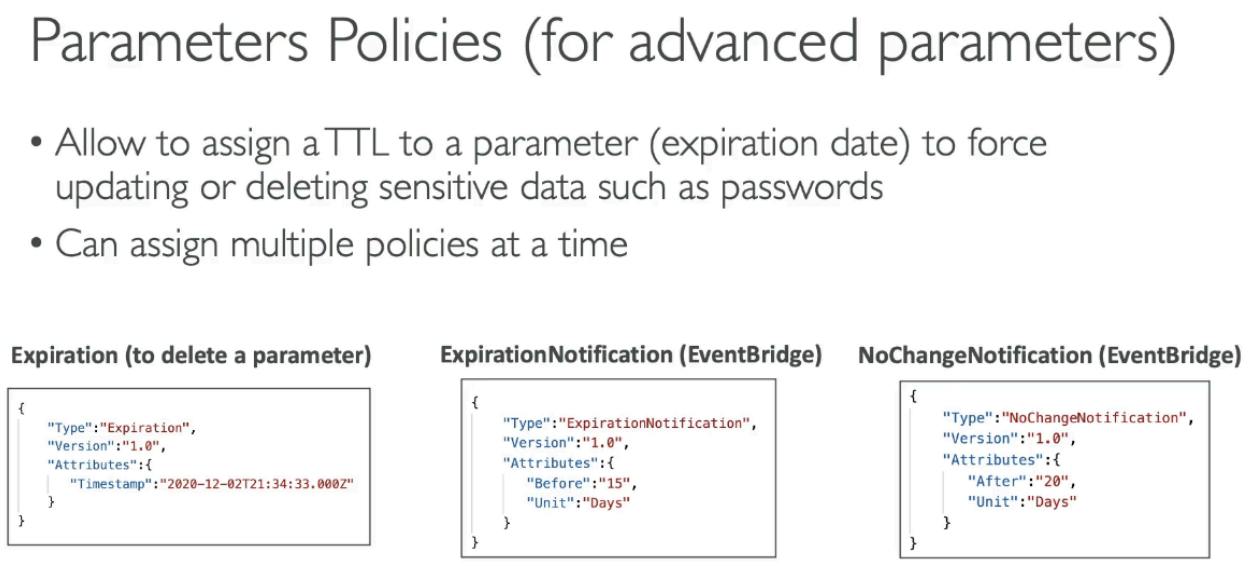
- Can have folders for how far down passwords are and can get secrets manager values

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

- Biggest difference between standard and advanced parameter tiers is parameter policies

Parameters Policies (advanced parameters only)

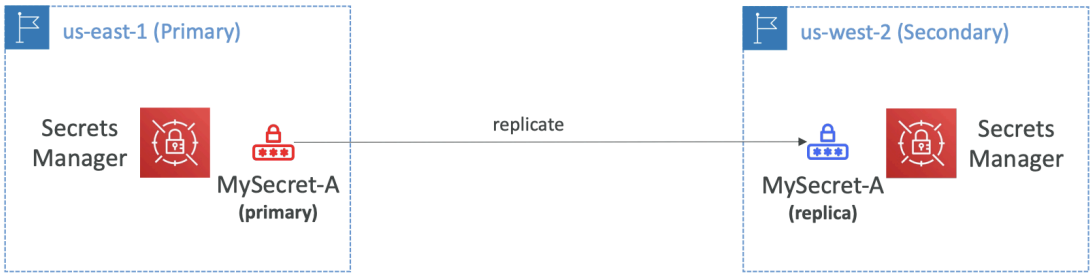


- Allow to assign TTL to a parameter to force update or delete sensitive data
- Can assign multiple policies at a time

Secrets Manager

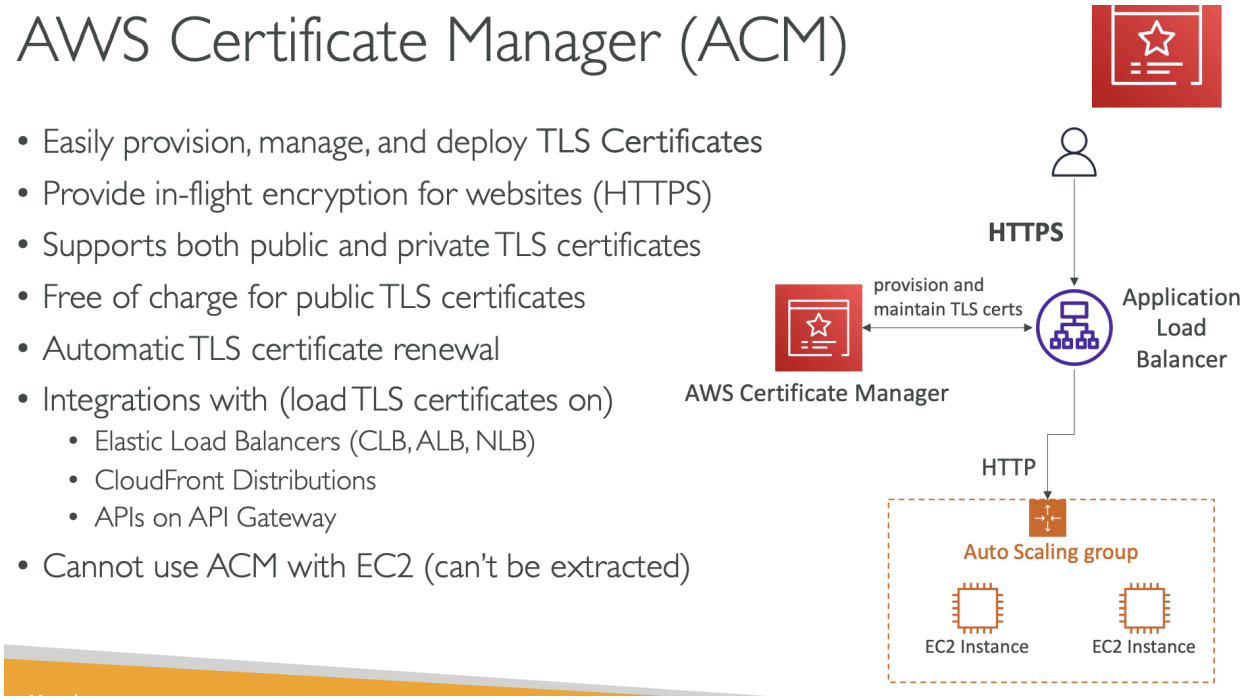
- Capability to force rotation of secrets every X days and automate generation of secrets on rotation (uses lambda)
    - Can encrypt with RDS
- Integration with RDS
    - Mostly meant for RDS

Multi Region Secrets



- Replicate secrets across multiple regions and secrets manager keeps read replicas in sync with primary secret
    - Can promote read replica secret to standalone secret
    - For: multi region apps, disaster recovery, multi region DB…

AWS Certificate Manager (ACM)



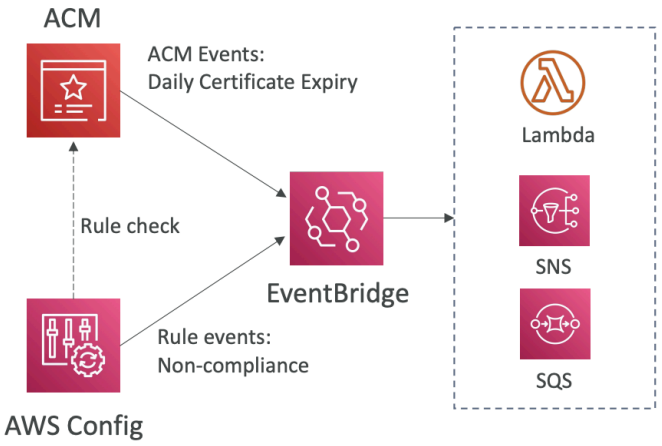
- Easily provision, manage, deploy SSL / TLS Certificates for in flight encryption for websites (HTTPS)
- Supports both public and private TLS certificates, free of charge for public TLS certificates and automatic TLS certificate renewal
- Integrations with (load TLS certificates on) ELB, CloudFront distributions, APIs on API GW
    - Cannot use public certificate on EC2 instances as they cannot be extracted
- Anytime for in flight encryption and generate certificates think ACM

Requesting Public Certificates

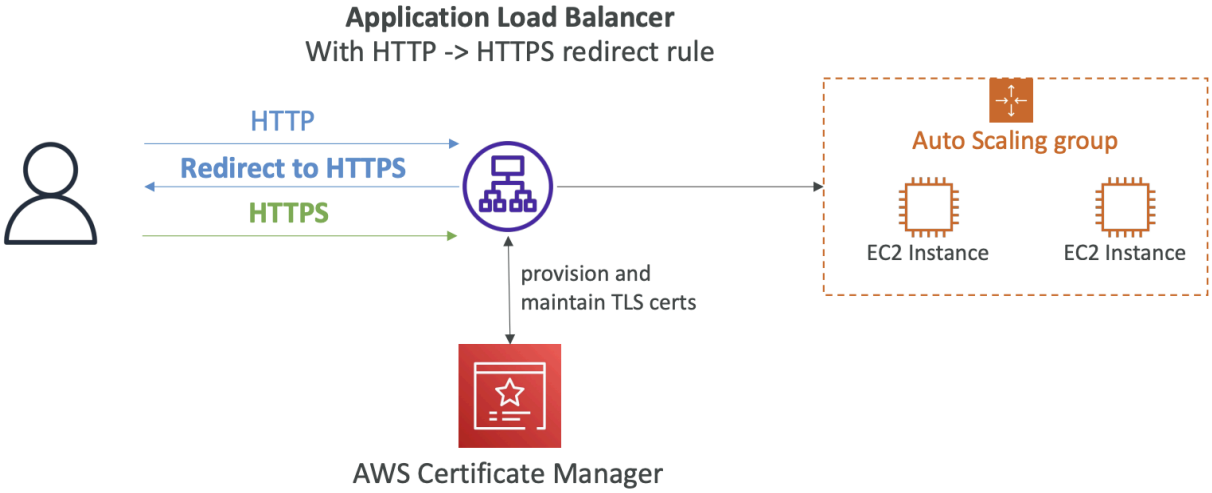
1. List domain names to be included in certificate
    a. Fully qualified domain name (FDDN)
    b. Wildcard domain
2. Select validation method: DNS validation, Email validation
    a. DNS validation preferred for automation purposes
    b. Email validation will send emails to contact addresses in the WHOIS database
    c. DNS validation will leverage a CNAME record to DNS config
3. Takes a few hours to get verified and public certificate will be enrolled for automatic renewal
    a. ACM auto renews ACM generated certificates 60 days before expire

Importing Public Certificates

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



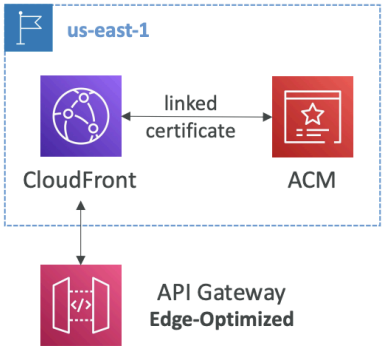
Integration with ALB

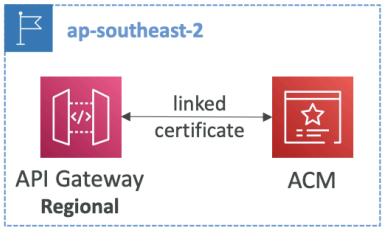


API Gateway – Endpoint Types

1. Edge Optimized (default): for global clients where requests routed through CF Edge location, but API GW still in 1 region
2. Regional: clients within same region, could manually combine with CloudFront
3. Private: accessed from VPC using VPC endpoint ENI using resource policy

Integration with API Gateway



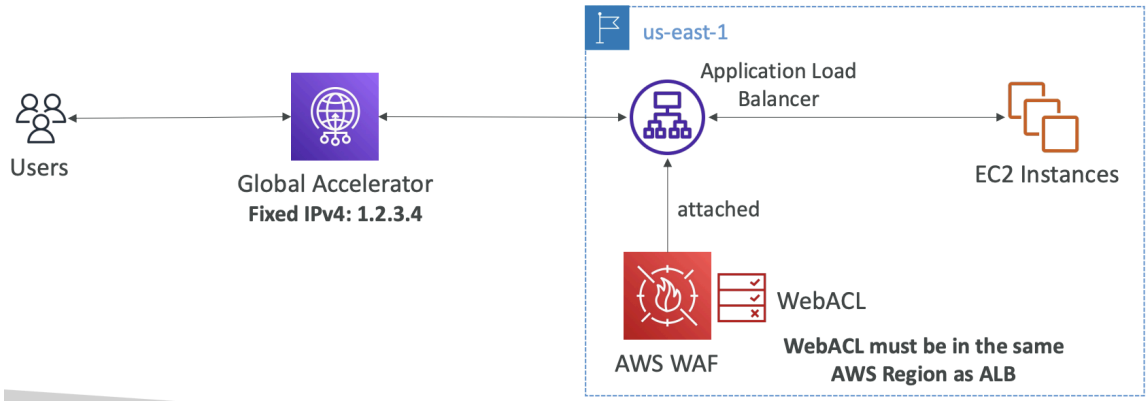


- Create Custom Domain Name in API Gateway
- Edge Optimized (default): for global clients
  - Requests routed through CloudFront Edge Locations (improve latency)
  - API Gateway lives in 1 region
  - TLS certificate must be in same region as CloudFront in us-east-1
  - Setup CNAME or A record in Route 53
- Regional: clients in same region
  - TLS certificate must be imported in API Gateway in same region as API stage

Web Application Firewall (WAF)

- Protects web apps from common web exploits (layer 7 HTTP)
- Deploy on: ALB, API GW, CloudFront, AppSync, Cognito User Pool
- Define Web ACL (Access Control List) rules:
  - IP Set: up to 10,000 IP, use multiple rules for more IP
  - HTTP headers, body, URI strings protects from common attack – SQL injection, Cross Site Scripting (XSS)
  - Size constraints, geo match (block countries)
  - Rate based rules (count event occurrences) – DDoS protection
- Web ACL are regional except CloudFront
- Rule group is a reusable set of rules that can be added to to web ACL

Fixed IP using WAF with ALB



- WAF does not support NLB (layer 4)
- Use global accelerator for fixed IP and WAF on ALB

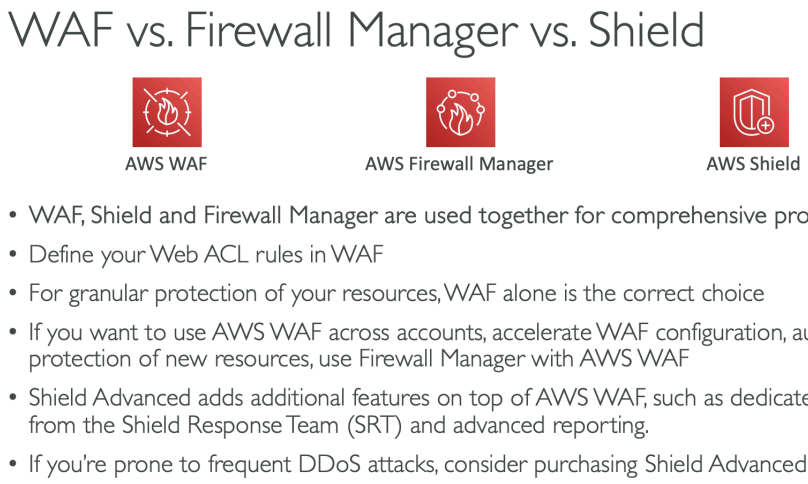
AWS Shield

- Protect from DDoS (many requests at same time)
- AWS Shield Standard
    - Free, protection from attacks like SYN / UDP Floods, Reflection attacks and other layer 3-4 attacks
- AWS Shield Advanced
    - DDoS mitigation service
    - Protect against more sophisticated attack on EC2, ELB, CloudFront, Global Accelerator, Route 53
    - 24/7 access to AWS DDoS response team
    - Protect against higher fees during usage spikes due to DDoS
    - Automatic application layer DDoS mitigation automatically created and deploys WAF rules to mitigate layer 7 attacks

AWS Firewall Manager

- Manage firewall rules in all accounts of AWS Org
- Security policy: common set of security rules
    - WAF rules (ALB, API GW, CloudFront)
    - Shield Advanced rules (ALB, CLB, NLB, Elastic IP, CloudFront)
    - SG for EC2, ALB, ENI in VPC
    - AWS Network Firewall (VPC level)
    - Route 53 resolver DNS firewall
- Policies created at region level
- Rules applied to new resources as they are created across all and future accounts in organization

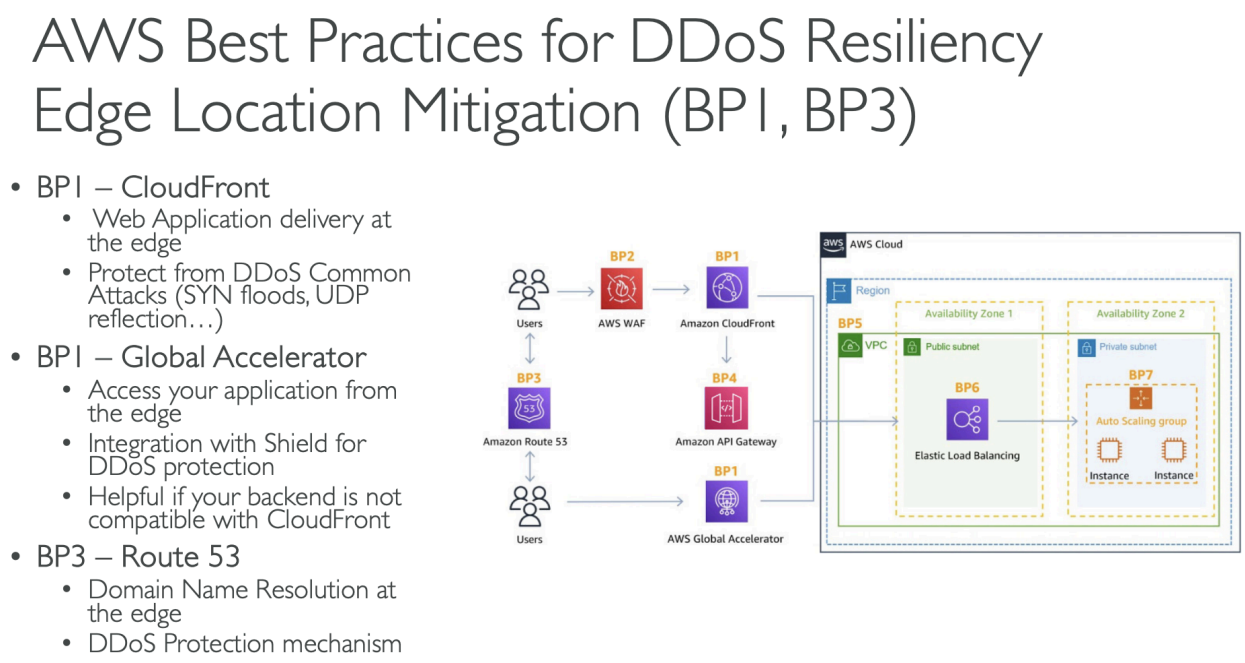
WAF vs Firewall Manager vs Shield



- Used together for comprehensive protection
- Define Web ACL rules in WAF
- For granular protection of resources, WAF alone
- Using WAF across accounts, accelerate WAF configuration, automate protection of new resources, use Firewall Manager with WAF
- Shield Advanced adds features on top of WAF, protects against DDoS

Best Practices for DDoS Resiliency

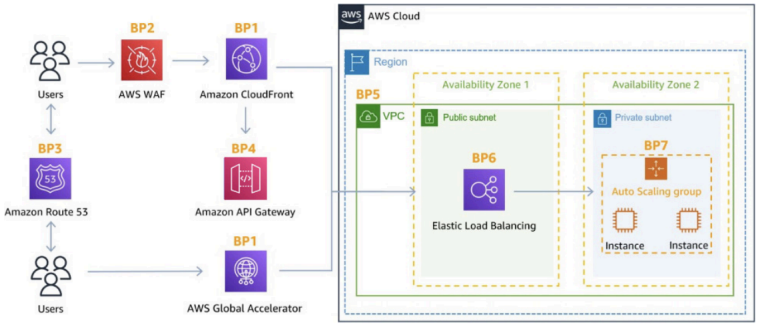
Edge Location Mitigation (BP1, BP3)



- Best Practice (BP) 1 – CloudFront
    - Web application delivery at edge
    - Protect from DDoS common attacks
- BP1 – Global Accelerator
    - Access app from edge
    - Integration with Shield for DDoS protection
    - Helpful if backend is not compatible with CloudFront
- BP3 – Route 53
    - Domain name resolution at the edge
    - DDoS protection

AWS Best Practices for DDoS Resiliency
Best pratices for DDoS mitigation

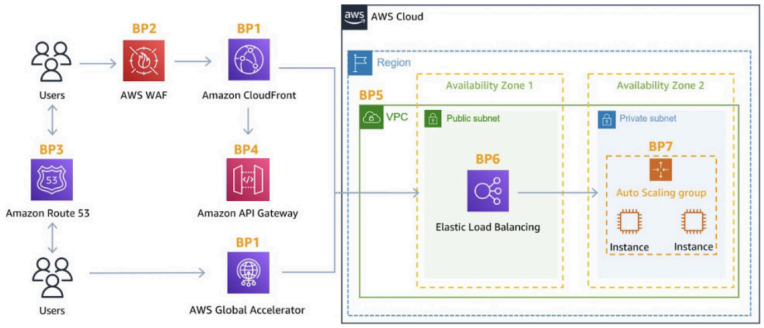
- Infrastructure layer defense (BP1, BP3, BP6)
  - Protect Amazon EC2 against high traffic
  - That includes using Global Accelerator, Route 53, CloudFront, Elastic Load Balancing
- Amazon EC2 with Auto Scaling (BP7)
  - Helps scale in case of sudden traffic surges including a flash crowd or a DDoS attack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



- Infrastructure Layer Defense (BP1, 3, 6)
  - Protect EC2 instances against high traffic
    - Using Global Accelerator, Route 53, CloudFront, ELB
- BP7 – EC2 + Auto scaling
  - Helps scale in traffic surges like DDoS
- BP6 – ELB
  - Scales traffic increases and distribute across 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

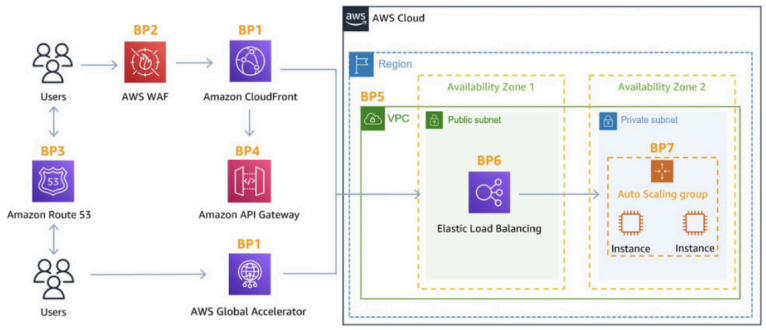


- BP 1, 2 – Detect and filter malicious web requests

        - CloudFront cache static content and serve from edge locations, protecting backend
        - WAF on top of CloudFront and ALB to filter and block requests based on request signatures
        - WAF rate based rules to auto block IPs of bad actors
        - WAF managed rules to block attacks based on IP reputation or anonymous IP
        - CloudFront can block specific geographies
    - Shield Advanced (BP1, 2, 6)
        - Auto application layer DDoS mitigation automatically creates, evaluates and deploys WAF rules to mitigate layer 7 attacks

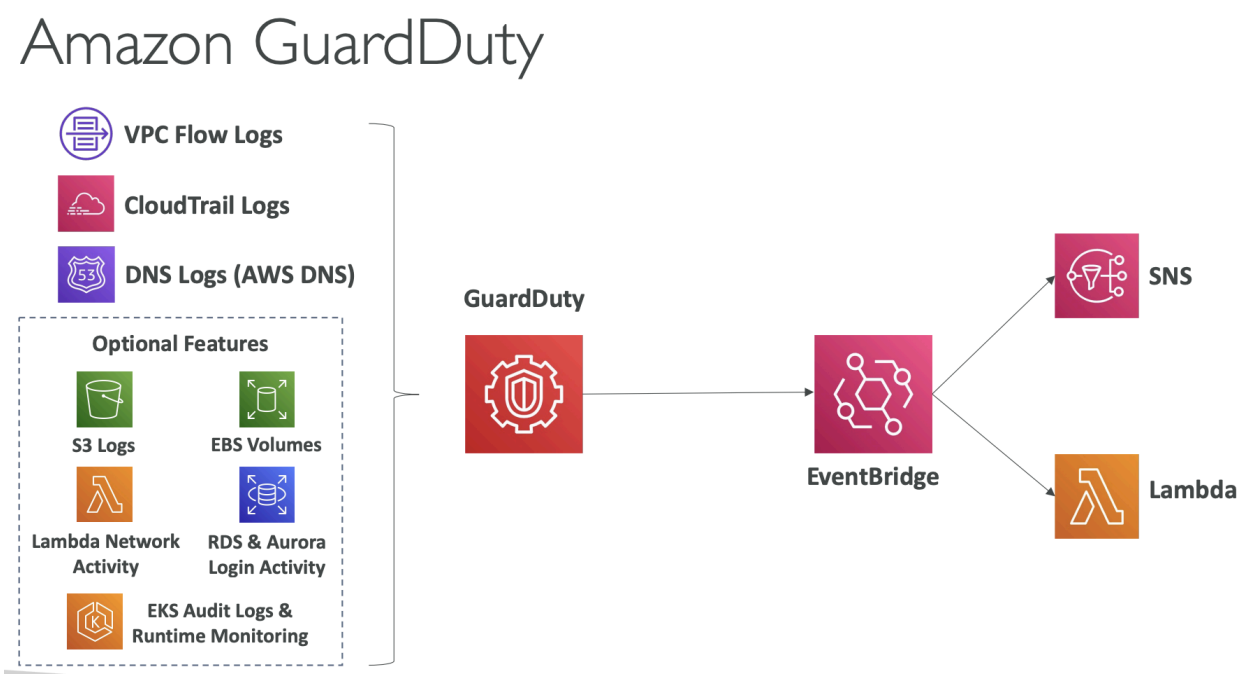
AWS Best Practices for DDoS Resiliency
Attack surface reduction

- 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



- Obfuscating AWS resources (BP1, 4, 6)
    - CloudFront, API GW, ELB to hide backend resources
- SG and Network ACLs (BP5)
    - SG and NACLs to filter traffic based on IP at the subnet or ENI level
    - Elastic IP protected by Shield Advanced
- Protecting API endpoints (BP4)
    - Hide EC2, lambda
    - Edge optimized mode or CloudFront + regional mode (more control for DDoS)
    - WAF + API GW: burst limits, header filtering, API keys

Amazon GuardDuty

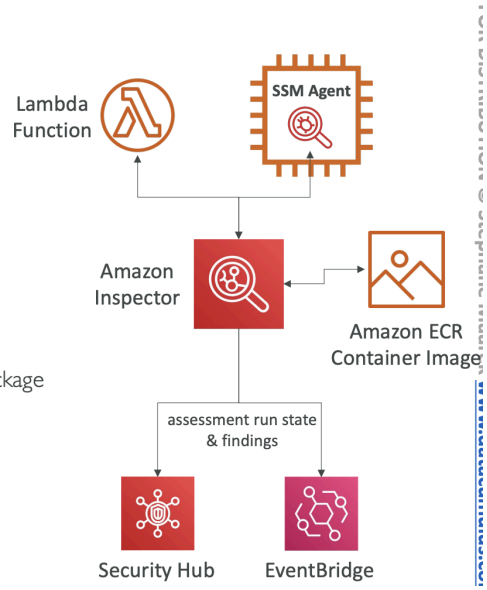


- Intelligent threat discovery via ML for anomaly detection and 3rd party data
    - 1 click, no software install
- Input data includes:
    - CloudTrail Event Logs – unusual API calls, unauthorized deployments…
        - Management events: create VPC subnet, create trail…
        - S3 Data events: get object, delete object…
    - VPC Flow Logs – unusual internal traffic, 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…
- EventBridge rules to be notified via Lambda or SNS
- Can protect against cryptocurrency attacks

Amazon Inspector

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



- Automated security assessments
  - Only for running EC2 instances, container images, and Lambda
- For EC2 instances:
  - Leverage AWS AWS System Manager (SSM) agent
  - Analyze against unintended network accessibility
  - Analyze running OS against known vulnerabilities
- For container images to ECR:
  - Assess container images as they are pushed
- For Lambda:
  - Identified software vulnerabilities in function code and package dependencies
  - Assessment of functions as they are deployed
- Reporting & integration with AWS Security Hub
- Sending findings to EventBridge

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

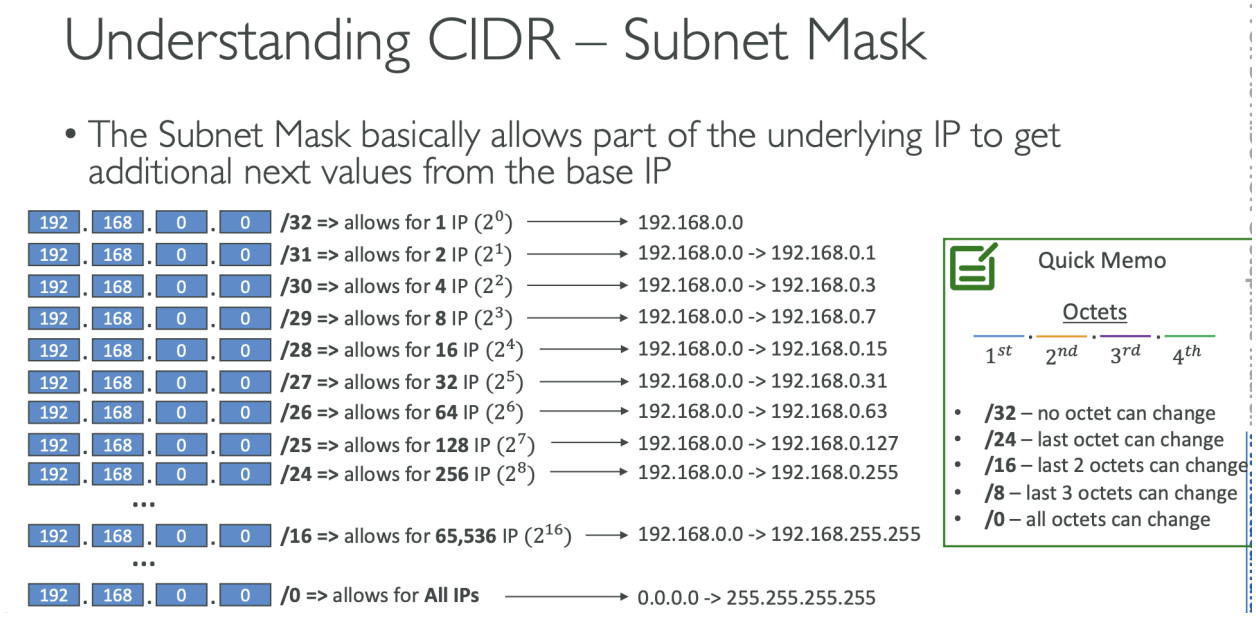
Amazon Macie

- Fully managed data security that uses ML and pattern matching to discover and protect sensitive data in AWS → alerts

Section 27: Networking – VPC

CIDR – IPv4

- Classless Inter-Domain Routing – method for allocating IP
  - Define IP ranges
    - xxx/32 → 1 IP
    - 0.0.0/0 → all IPs
    - 192.168.0.0/26 → 192.168.0.0 – 192.168.0.63 (64 IPs)
- Used in SG rules and AWS Networking
- 2 components
  - Base IP
    - Represents an IP contained in range
  - Subnet mask



- Allows part of underlying IP to get additional next values from base IP
- Defined how many bits can change in IP
    - /0, /24, /32
- Can take 2 forms
    - /8 → 255.0.0.0
    - /16 → 255.255.0.0
    - /32 → 255.255.255.255

Understanding CIDR – Little Exercise

- 192.168.0.0/24 = … ?
    - 192.168.0.0 – 192.168.0.255 (256 IPs)
- 192.168.0.0/16 = … ?
    - 192.168.0.0 – 192.168.255.255 (65,536 IPs)
- 134.56.78.123/32 = … ?
    - 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)

- Internet Assigned Numbers Authority (IANA) established certina blocks of IPv4 addresses for private and public addresses
    - Private IP
        - 10.0.0.0 → 10.255.255.255 (10.0.0.0/8) → 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) → home networks
    - Rest are public IPs

Default VPC Overview

- All new accounts have default VPC
- New EC2 instances are launched into default VPC if no subnet is specified
- Default VPC has internet connectivity and all EC2 instances have public IPv4 address
- Get public and private IPv4 DNS names

VPC in AWS – IPv4

- Can have multiple VPC in region (soft limit of 5 per region)
- Max CIDR per VPC is 5; for each CIDR:
    - Min size: /28 (16 IP)
    - Max size /16 (65536 IP)
- Because VPC is private, only private IPv4 ranges allowed
- VPC CIDR should not overlap with other networks (VPCs or corporate networks…)

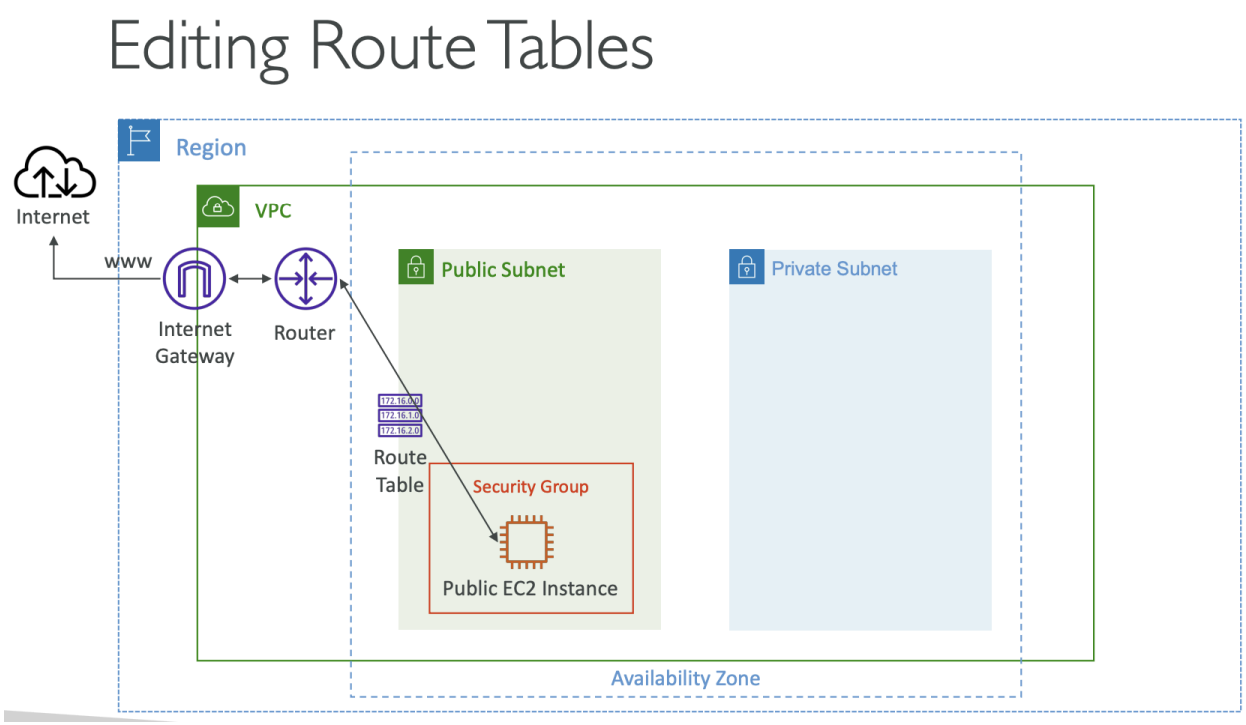
VPC – Subnet (IPv4)

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)

- AWS reserves 5 IP addresses (first 4, last 1) in each subnet
    - Are not available for use and can't be assigned to EC2 instance
- If need 29 IP addresses for EC2 instances, can't use /27 because 2^5 IP addresses, so must use /26

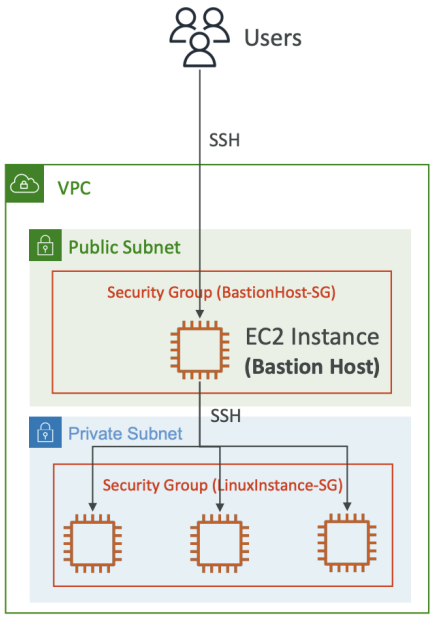
Internet Gateway



- Allows resources in VPC to connect to internet
    - Do not allow internet on its own, need route table
- Scales horizontally and highly available / redundant
- Must be created separately from VPC
    - 1 VPC can only be attached to 1 IGW and vice versa

Bastion Hosts

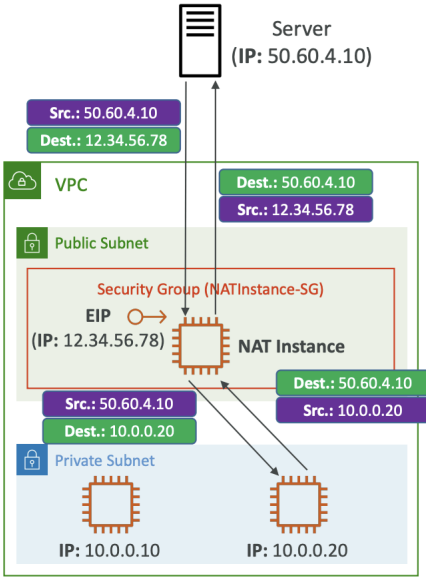
- Used to access private subnet EC2 instances
- Bastion host in the public subnet that then connects to private subnets
    - Bastion host SG must allow inbound from target on port 22 from restricted CIDR
    - SG of EC2 instances must allow SG of Bastion Host or private IP of bastion host



Nat Instance (outdated, still at exam)

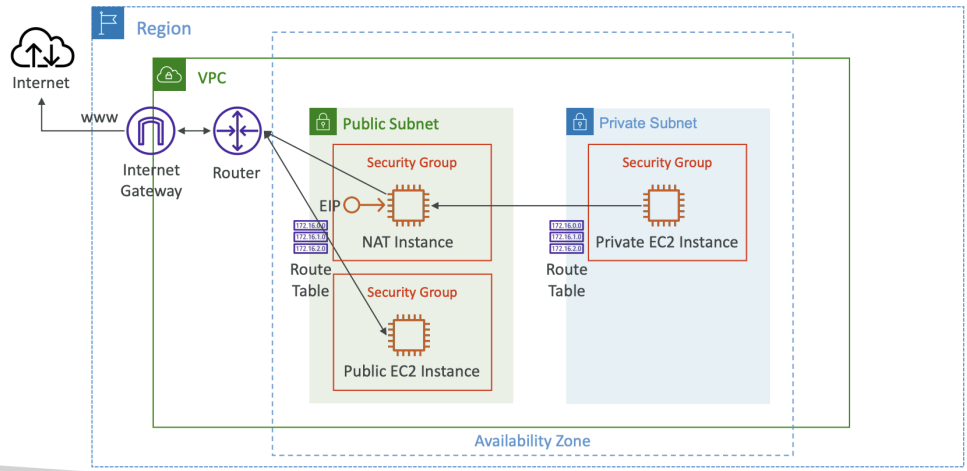
NAT Instance (outdated, but still at the exam)

- 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



- Network Address Translation
- Allow EC2 instance in private subnets to connect to internet
- Must be launched in public subnet
- Must disable EC2 setting: source / destination check
  - The source / destination at each stage will get rewritten
- Must have Elastic IP attached
- Route table must be configured to route traffic from private subnets to NAT instance

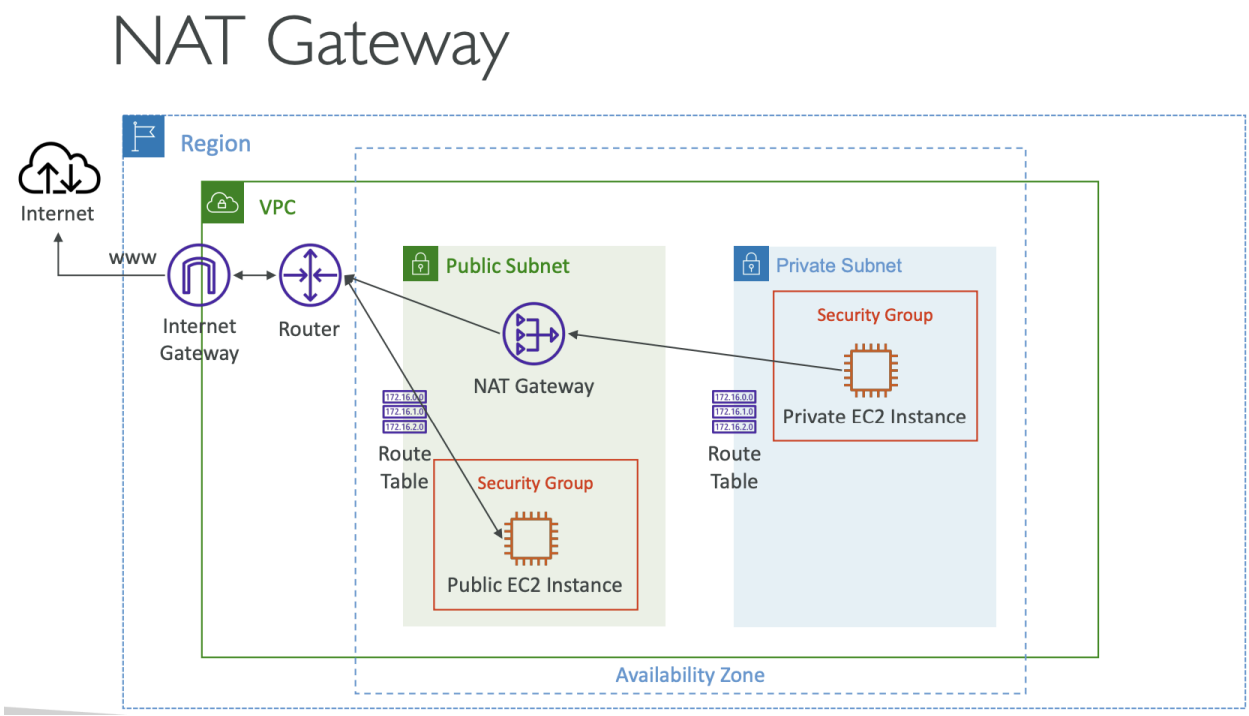
NAT Instance



Comments

- Not highly available / resilient setup out of the box
    - Need to create ASG in multi AZ + resilient user data script
- Internet traffic bandwidth depends on EC2 instance type
- Must manage SG & rules
    - Inbound
        - Allow HTTP / S traffic coming from private subnets
        - Allow SSH from home network
    - Outbound
        - Allow HTTP / S traffic to internet

NAT Gateway

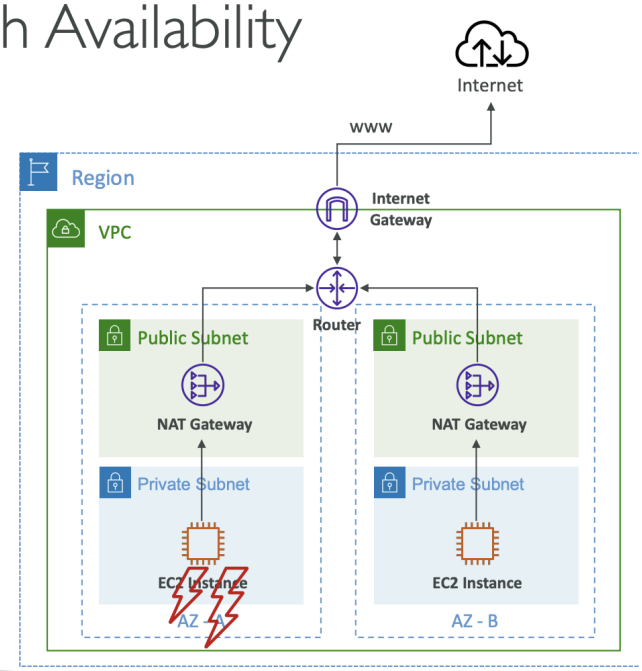


- AWS managed, higher bandwidth, high availability, no administration
    - 5 Gbps of bandwidth with auto scaling up to 100 Gbps
- Pay per hour for usage and bandwidth
- Created in specific AZ, uses Elastic IP
- Can’t be used by EC2 instances in same subnet (only other subnets)
- Requires IGW (private subnet → NAT GW → IGW)
- No SG to manage

NAT with High Availability

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



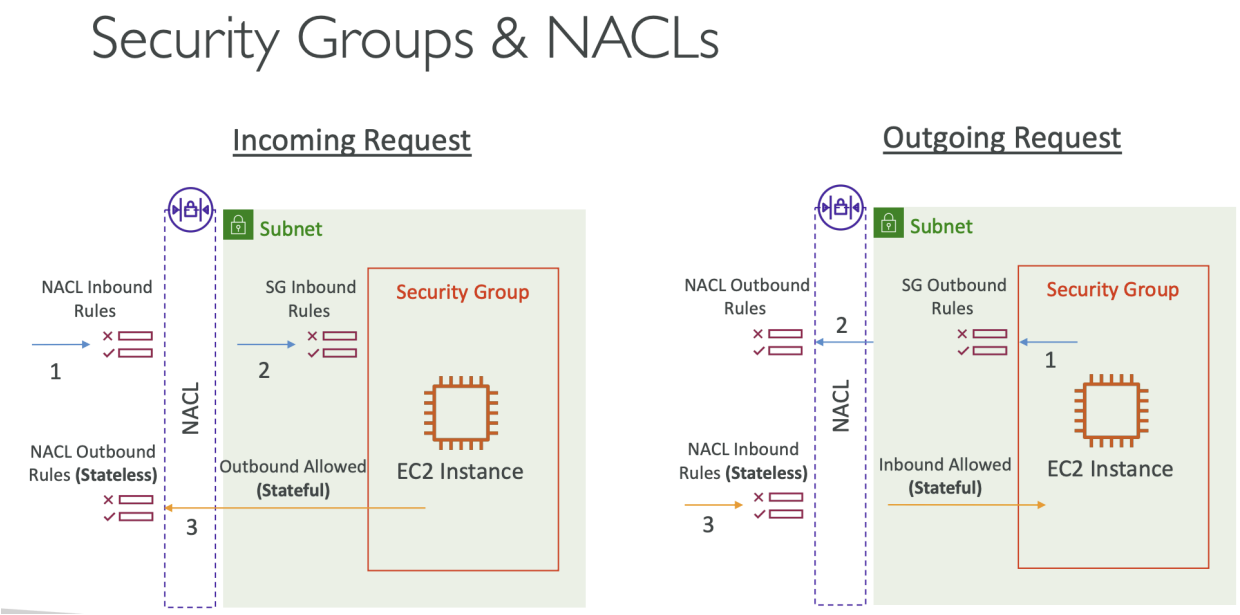
- Resilient within single AZ
  - Must create multiple NAT GW in multiple AZ for fault tolerance
  - No cross AZ failover needed because if an AZ goes down it doesn't need NAT

NAT Gateway vs NAT Instance

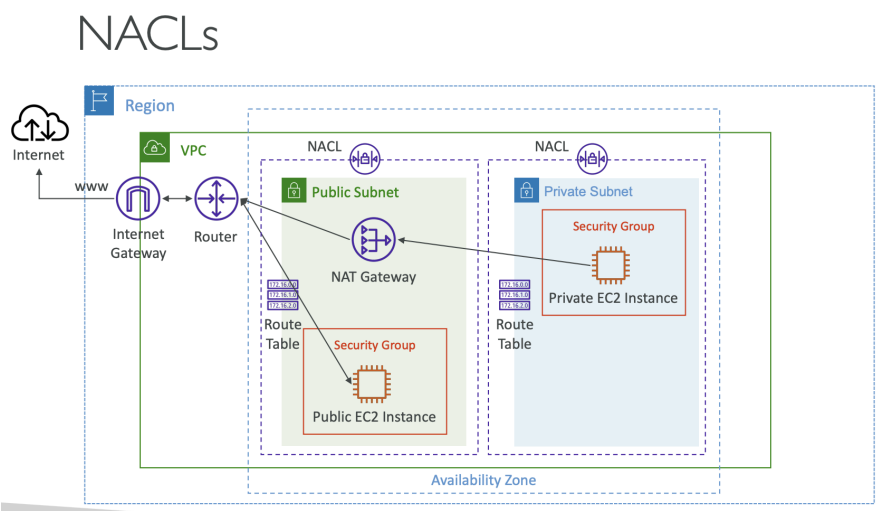
NAT Gateway vs. NAT Instance

	NAT Gateway	NAT Instance
Availability	Highly available within AZ (create in another AZ)	Use a script to manage failover between instances
Bandwidth	Up to 100 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?	✗	✓

Security Group & NACLs



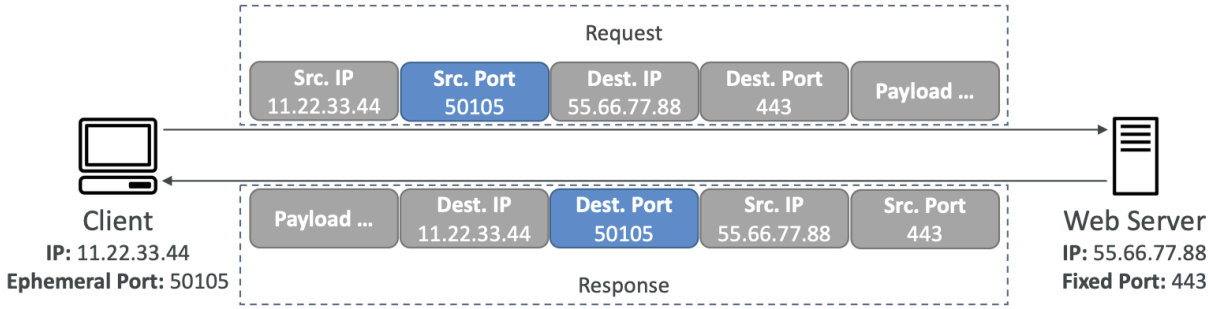
- SG is stateful meaning whatever is accepted can go out. NACL is stateless where in and out must be accepted
- NACL is like firewall that control traffic to / from subnet
    - Great at blocking specific IP at the subnet level
- 1 NACL per subnet, new subnets are assigned default NACL
- Define NACL rules:
    - Rules have a number, higher precedence = lower number
    - First rule matched will be the decision, where last rule is * and denies if no rule match
    - Add rules by increment of 100
- Newly created NACL deny everything



Default NACL

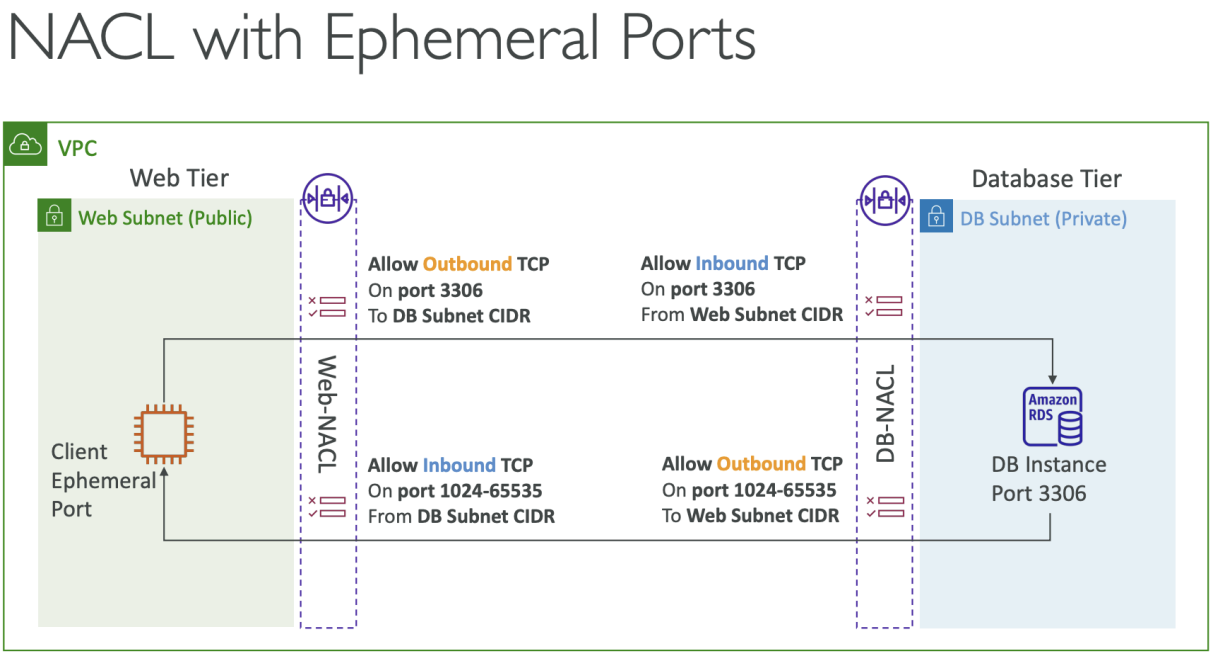
- Accepts everything inbound / outbound with the subnets it's associated with
    - Do not modify default, create custom NACL

Ephemeral Ports

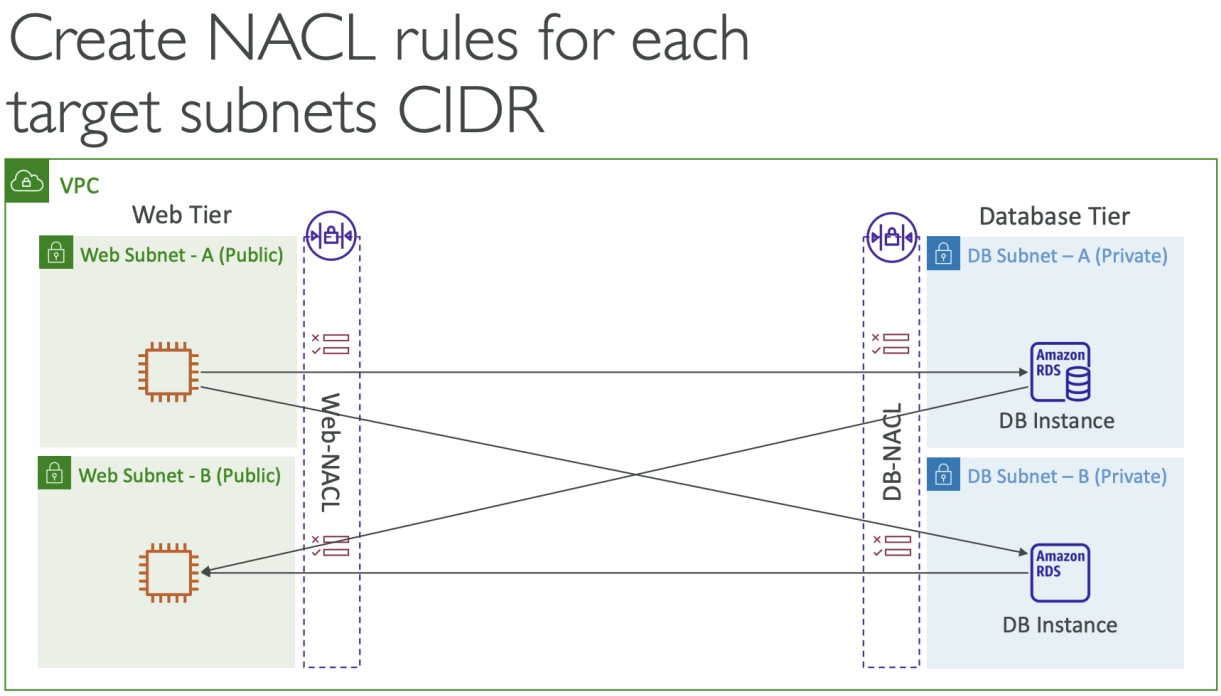


- For any 2 endpoints to establish a connection, they must use ports
- Clients connect to a defined port and expect a response on an ephemeral port
    - Different OS use different port ranges

NACL with Ephemeral Ports



Create NACL rules for each target subnets CIDR

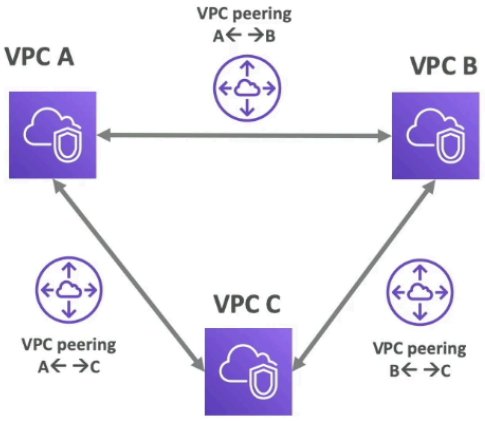


Security Group vs NACLs

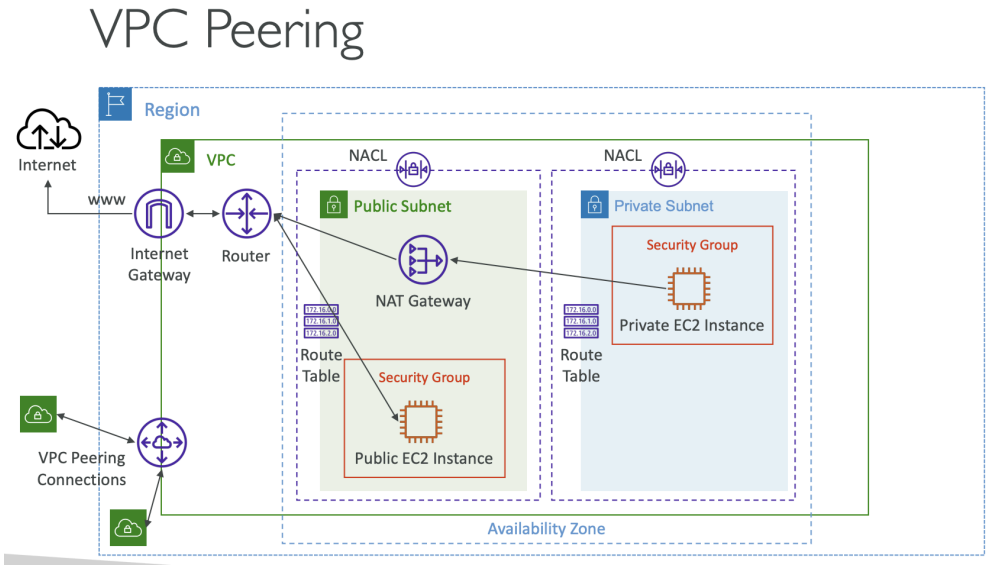
Security Group	NACL
Operates at the instance level	Operates at the subnet level
Supports allow rules only	Supports allow rules and deny rules
Stateful: return traffic is automatically allowed, regardless of any rules	**Stateless:** return traffic must be explicitly allowed by rules (think of ephemeral 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

VPC Peering

- Privately connect 2 VPCs using AWS network to make them behave as if they were in the same network
- Must not have overlapping CIDRs
- Not transitive (must be established for each VPC that need to communicate)
- Must update route tables in each VPC's subnets to ensure EC2 instances can communicate with each other

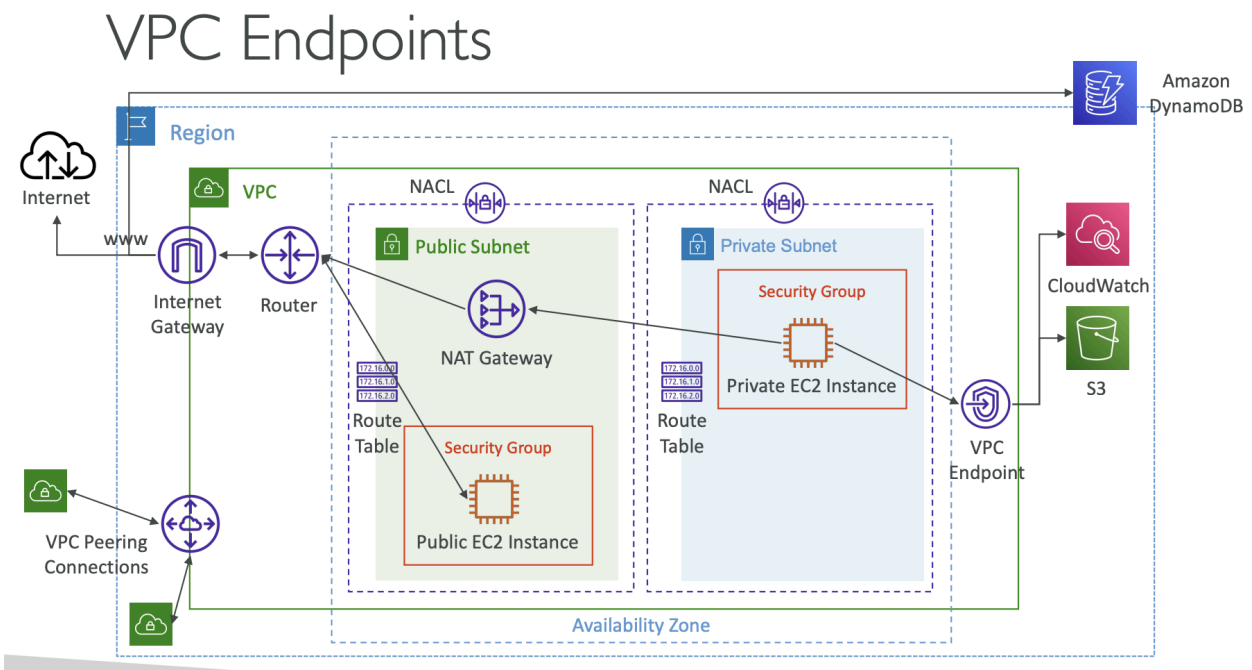


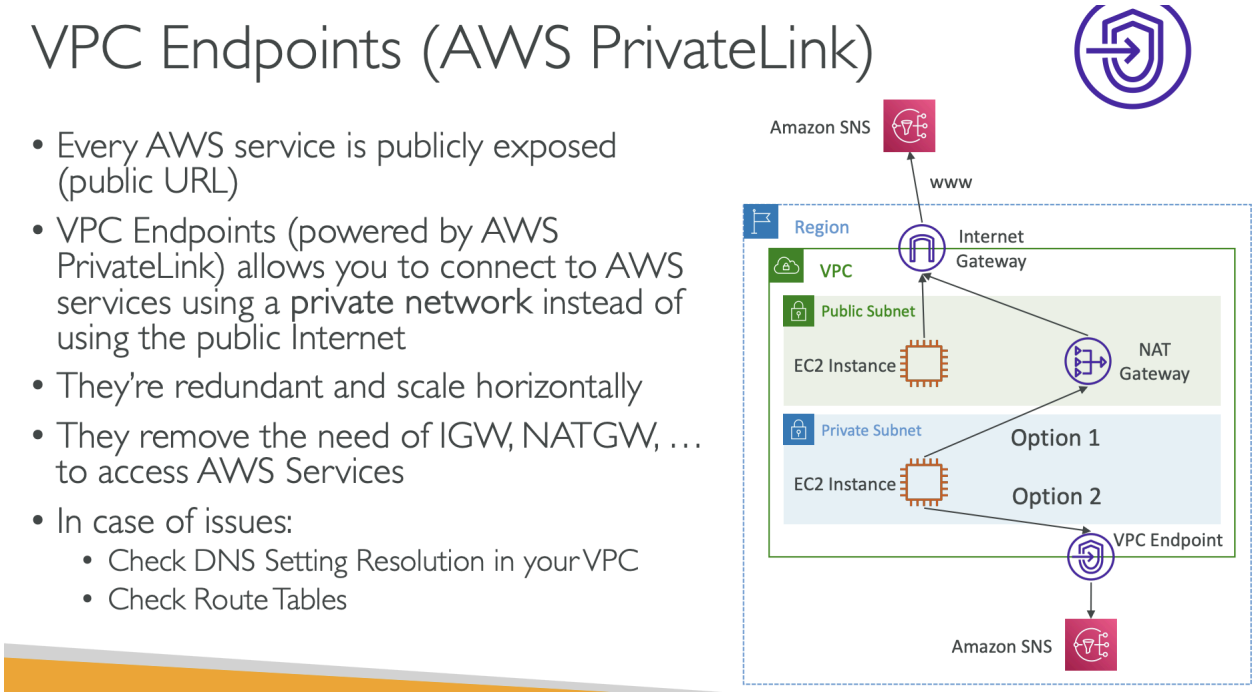
VPC Peering – Good to know



- Can create VPC peering connection between VPC in different accounts / regions
- Can reference a SG in a peered VPC
    - Works cross account in same region

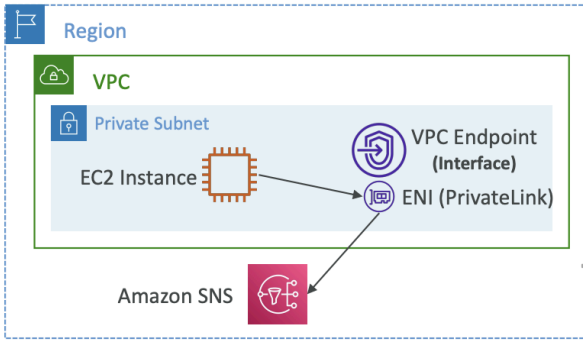
VPC Endpoints (AWS PrivateLink)





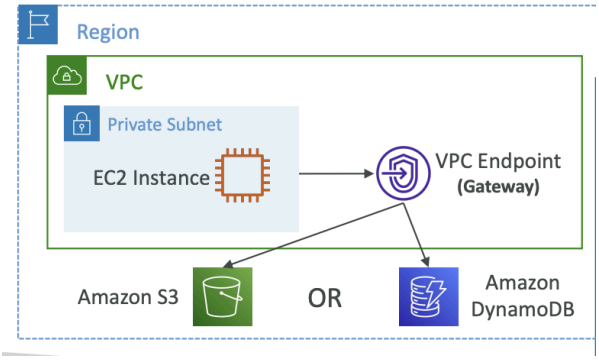
- Every AWS service is publicly exposed (public URL)
- VPC endpoints allows connection to AWS services using private internet
    - Redundant and scale horizontally
    - Removes the need for IGW, NATGW…

- In case of issues:
    - Check DNS setting resolution in VPC
    - Check route table



Interface Endpoint (powered by PrivateLink)

- Provisions ENI (private IP) as entry point (must attach to SG)
- Supports most AWS services
- $ per hour + $ per GB data processed



Gateway Endpoints

- Provisions a gateway and must be used as a target in a route table (no SG used)
- Supports S3 and DynamoDB
- Free

Gateway or Interface Endpoint for S3?

- Gateway preferred at exam, cost is free
- Interface Endpoint is preferred access is required from on premise, a different VPC or region

VPC Flow Logs

- Capture info about IP traffic going into interfaces:
    - VPC flow logs
    - Subnet flow logs
    - Elastic Network Interface flow logs
- Helps monitor / troubleshoot connectivity issues
- Captures network info from AWS managed interfaces and can be sent to other AWS resources (CloudWatch, S3)

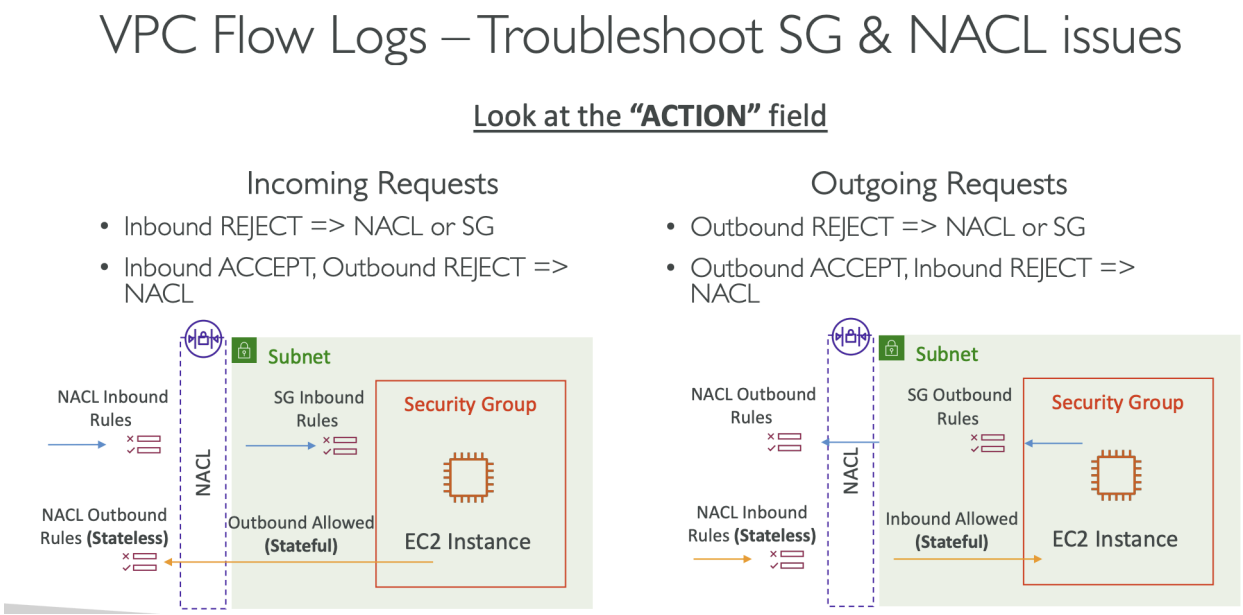
VPC Flow Log Syntax

- srcaddr & dstaddr – help identify problematic IP
- srcport & dstport – help identity 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

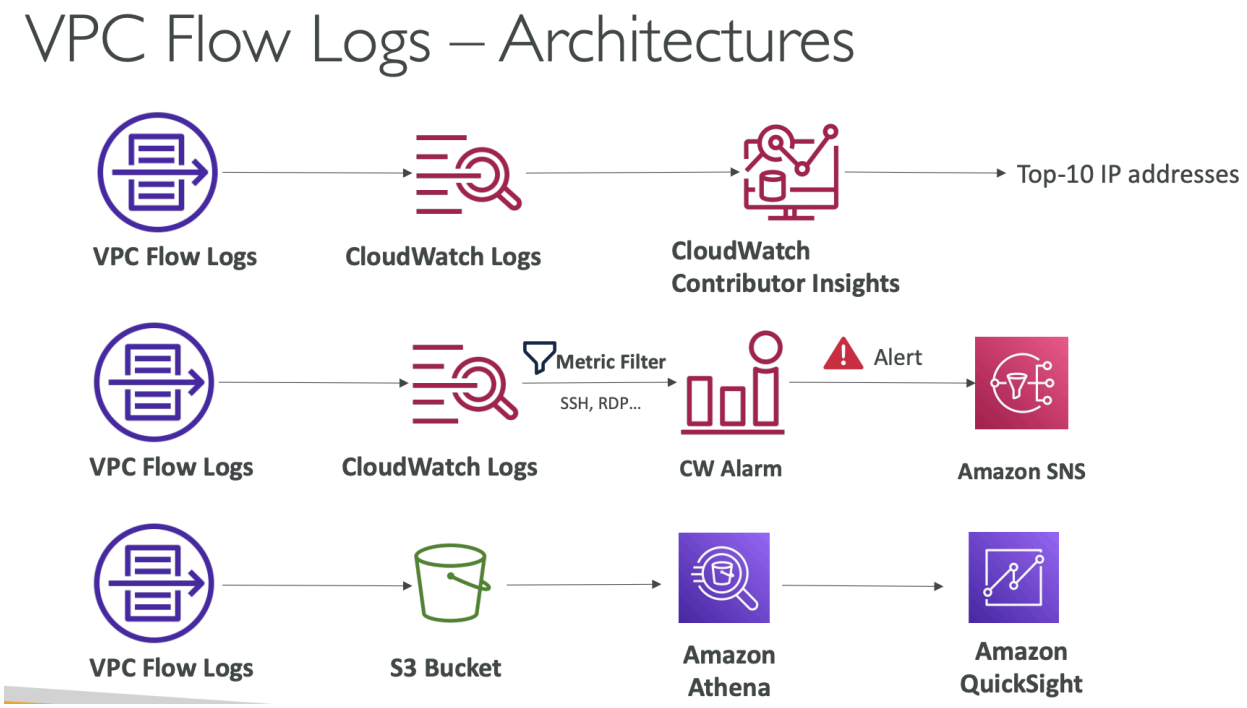
- Query using Athena on S3 or CloudWatch Logs Insights

Troubleshoot SG & NACL Issues

- Look at action field

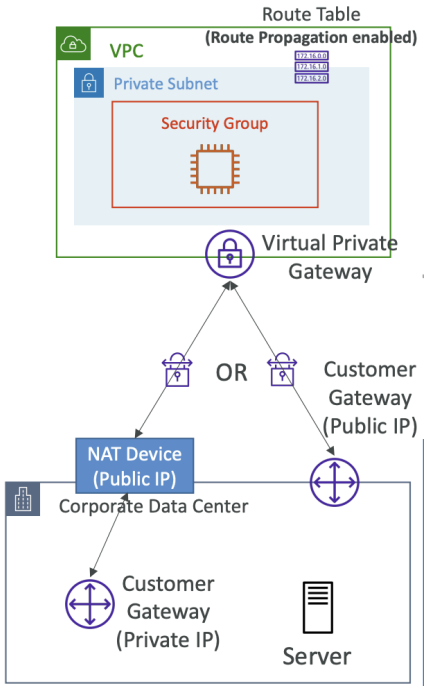


Architectures



AWS Site to Site VPN

- Virtual Private Gateway (VPW)
  - VPN concentrator on AWS side of VPN connection
  - VGW created and attached to 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 or physical device on customer side of VPN connection

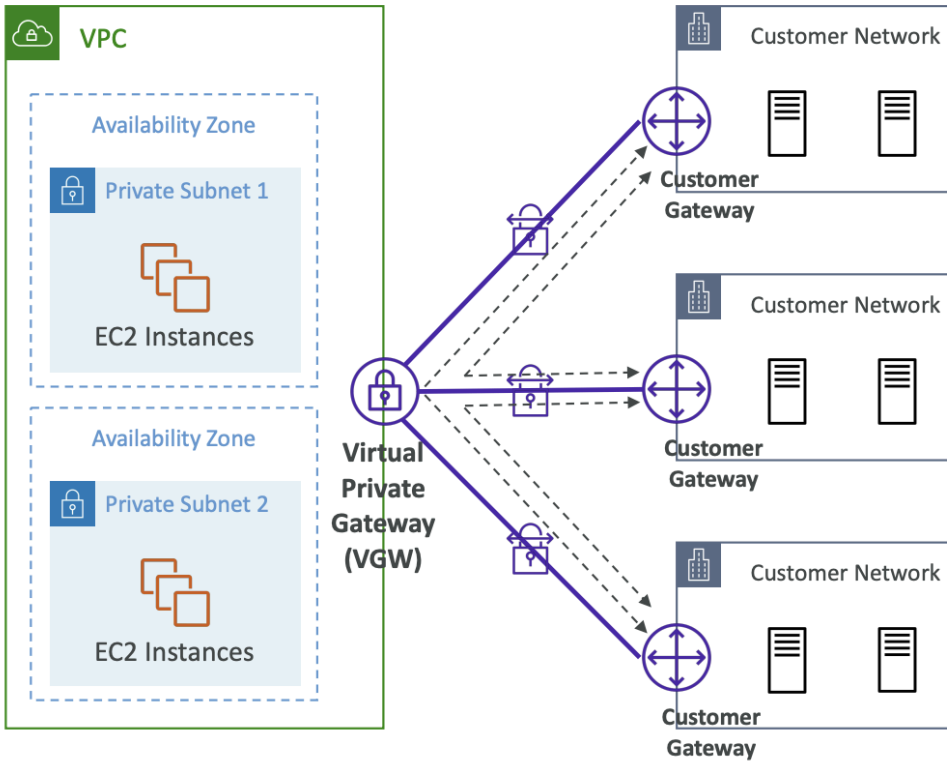


Site to Site VPN Connections

- Customer Gateway Device (on premise)
  - What IP to use?
    - Public internet routable IP for customer gateway device
    - If it's behind a NAT device that has NAT traversal (NAT-T), use the public IP of the NAT device
      - Private IP connection

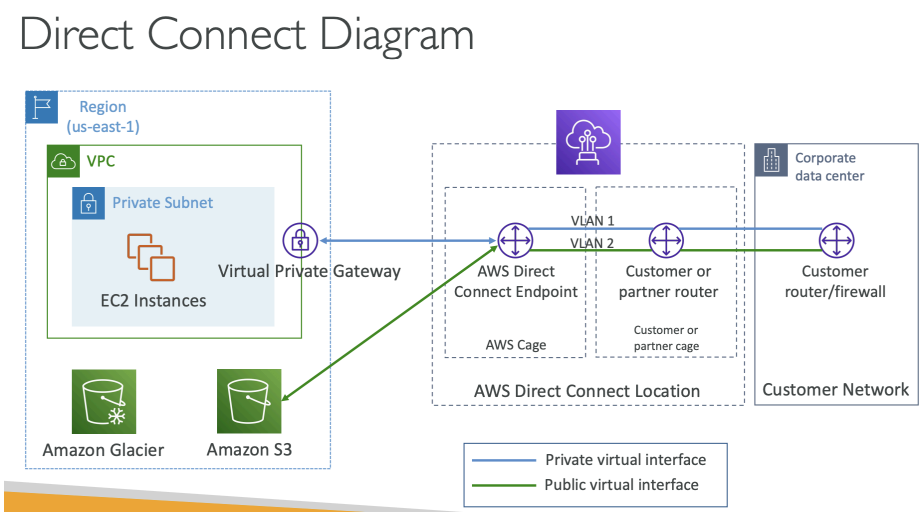
- Enable Route Propagation for VPG in route table that is associated with your subnets
- If you need to ping EC2 instances from on premise, make sure to add ICMP protocol on the inbound of SG

AWS VPN CloudHub



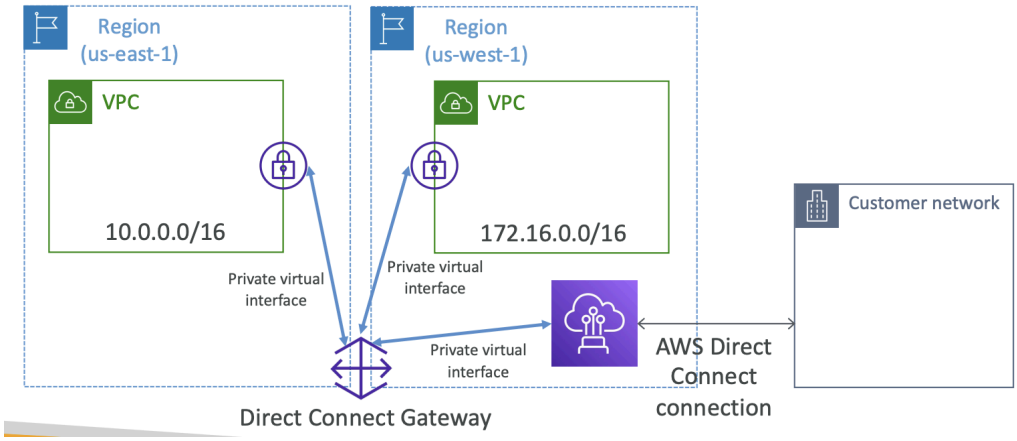
- Provides secure communication between multiple on premise sites if you have multiple VPN connections
    - VPC connection via public internet
    - Connect multiple VPN connections on same VGW, set up dynamic routing and configure route tables
- Low cost hub and spoke model for primary or secondary network connectivity between different locations (VPN only)

Direct Connect (DX)

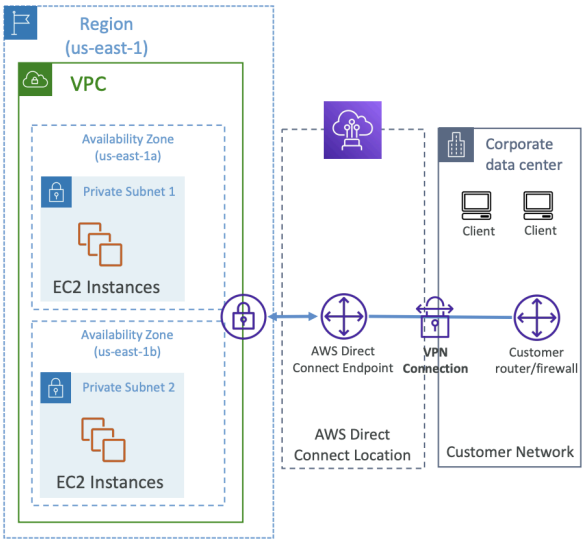


- Private dedicated connection from remote network to VPC; supports IPv4 and v6
    - Must be set up between on premise and AWS Direct Connect locations, 1+ month to set up
    - Access public and private resources on same connection
- Need to set up a Virtual Private Gateway on VPC
- Dedicated Connection:
    - 1, 10, 100 Gbps capacity + physical ethernet port
- Hosted Connection:
    - 50, 500 Mbps, 10 Gbps
    - Connection requests made via AWS Direct Connect Partners
    - Capacity can be added or removed on demand
- Use cases: increased bandwidth, consistent network, hybrid environments

Direct Connect Gateway



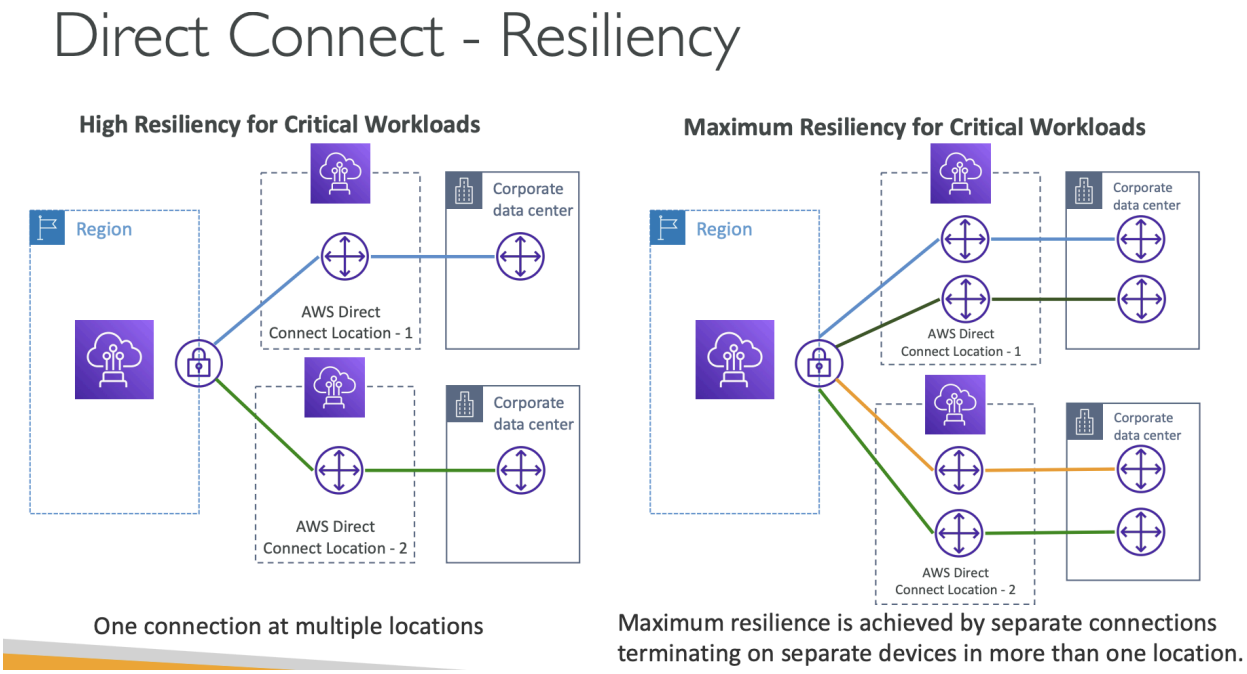
- Direct Connect to 1+ VPC in many different regions (same account)



DX Encryption

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

DX – Resiliency

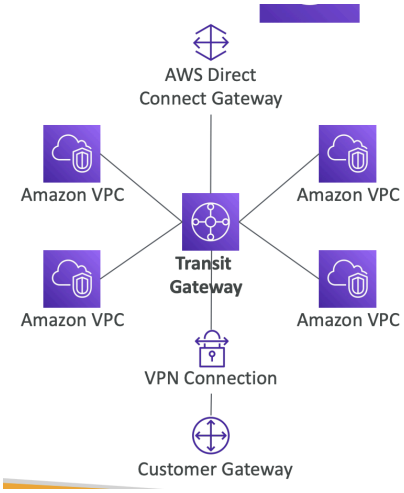


- High Resiliency for Critical Workloads
    - One connection at multiple locations
- Maximum resiliency for critical workloads
    - Separate connections terminated on separate devices in more than 1 location

Site to Site VPN Connection as Backup

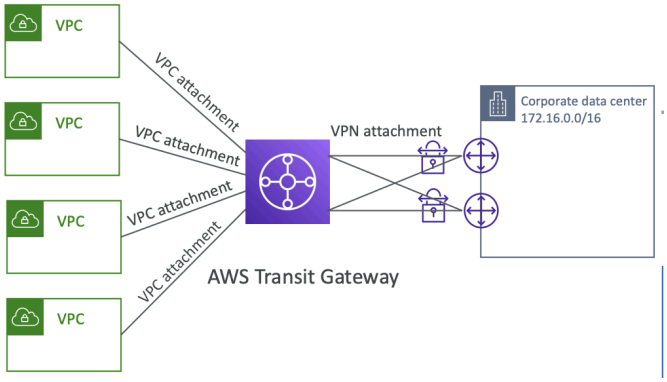
- In case Direct Connect fails, can set a backup Direct Connect connection (expensive) or site to site VPN connection

Transit Gateway



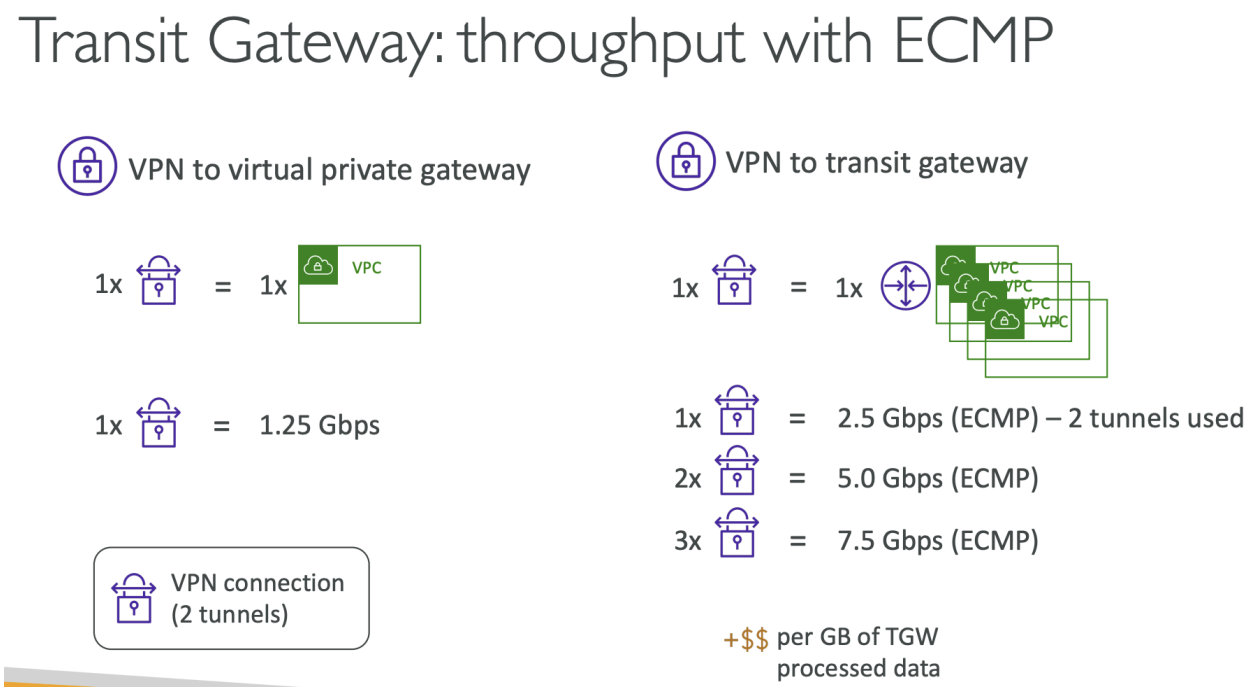
- Transitive peering between thousands of VPC and on premise, hub and spoke (star) connection
  - Works with Direct Connect Gateway, VPN connections
  - Supports IP multicast
- Regional resource, but can work cross region
  - Share cross account using Resource Access Manager (RAM)
  - Can peer Transit Gateway across regions
- Route tables: Limit which VPC can talk with other VPC

Transit Gateway: Site to Site VPN ECMP

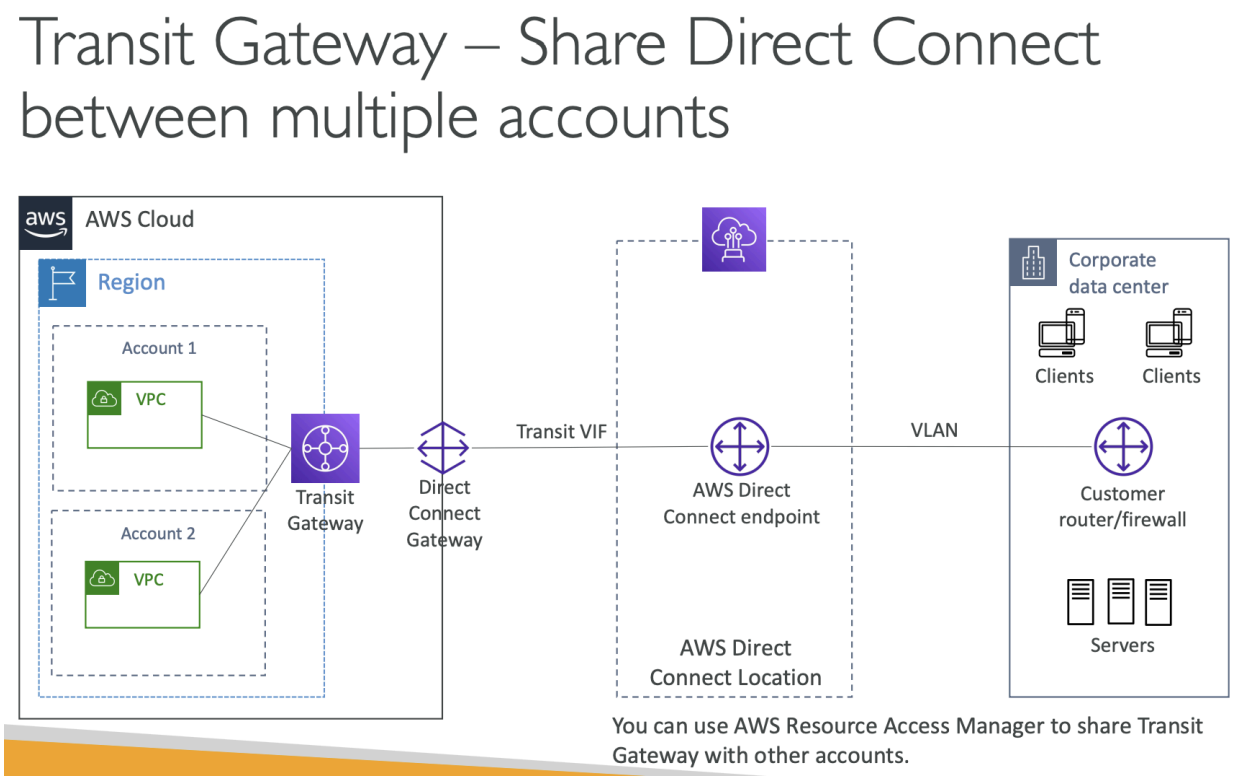


- Equal cost multi path routing
  - Routing strategy to allow to forward a packet over multiple best path
- Use case: create multiple site to site VPN connections to increase bandwidth of connection to AWS

Throughput with ECMP

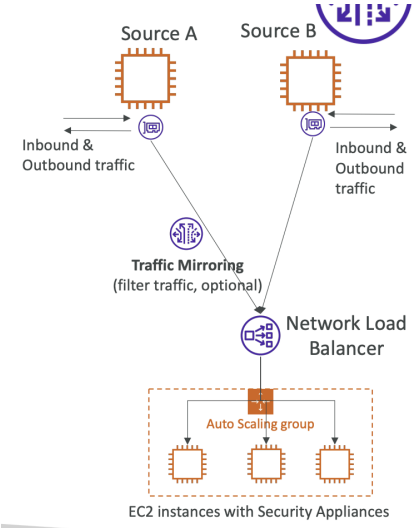


Share Direct Connect between multiple accounts



VPC Traffic Mirroring

- Capture and inspect network traffic in VPC
- Route traffic to security appliances that you manage
- Capture traffic:
    - From (source): ENI
        - Captures all packets or filter
    - To (targets): ENI or NLB
- Source and target can be in same VPC or with VPC peering
- Use case: content inspection, threat monitoring, troubleshooting



IPv6 in VPC

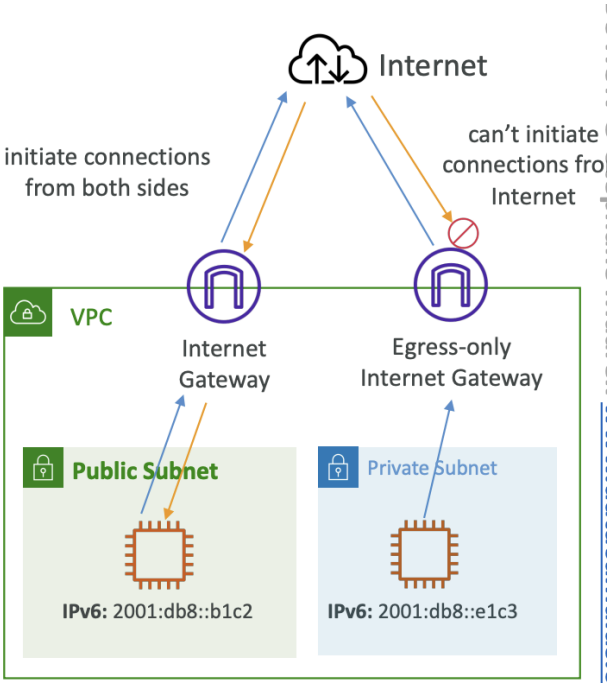
- Every IPv6 address in AWS is public and internet routable (no private range)
- IPv4 cannot be disabled for VPC and subnets, can enable IPv6 to operate in dual stack mode
    - EC2 instances will at least get internal IPv4 and public IPv6
    - Can communicate using either to internet via Internet Gateway

IPv6 Troubleshooting

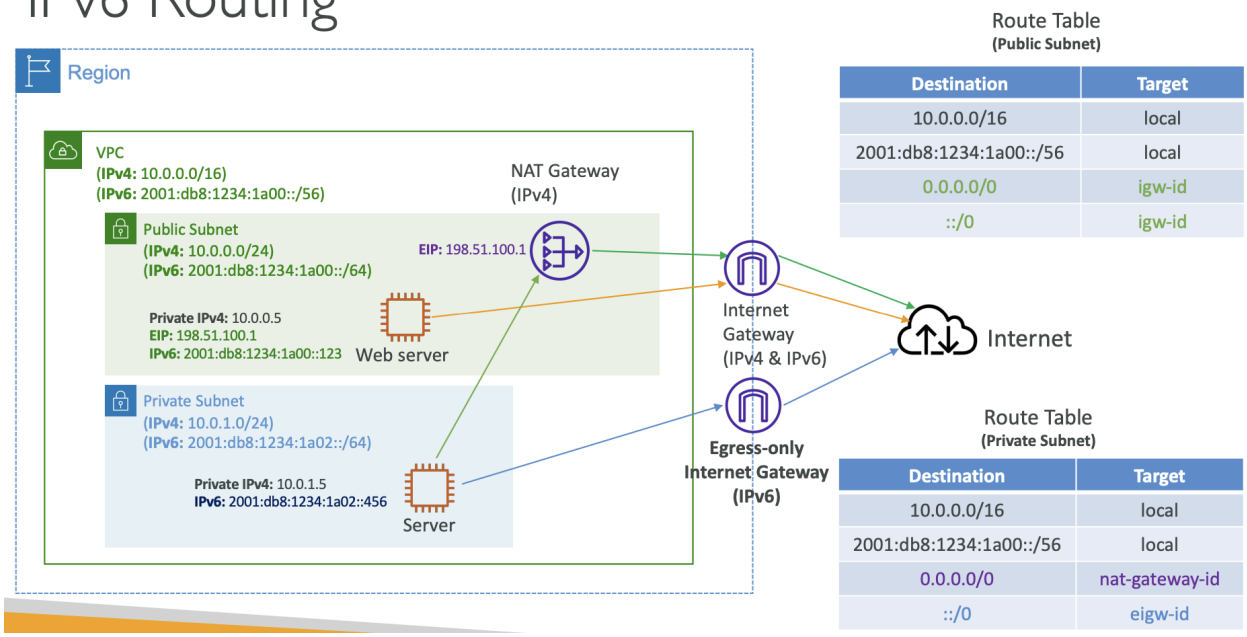
- If you cannot launch an EC2 instance in your subnet with IPv6 enabled VPC, it's because there are no available IPv4 in subnet
    - Solution: create new IPv4 CIDR in subnet

Egress Only Internet Gateway

- Used only for IPv6 traffic, similar to NAT Gateway for IPv6
- Allows instances in VPC outbound connections over IPv6 while preventing internet to initiate IPv6 connection to instances
- Must update route tables



IPv6 Routing



VPC Summary

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, when the target is an IPv4 address

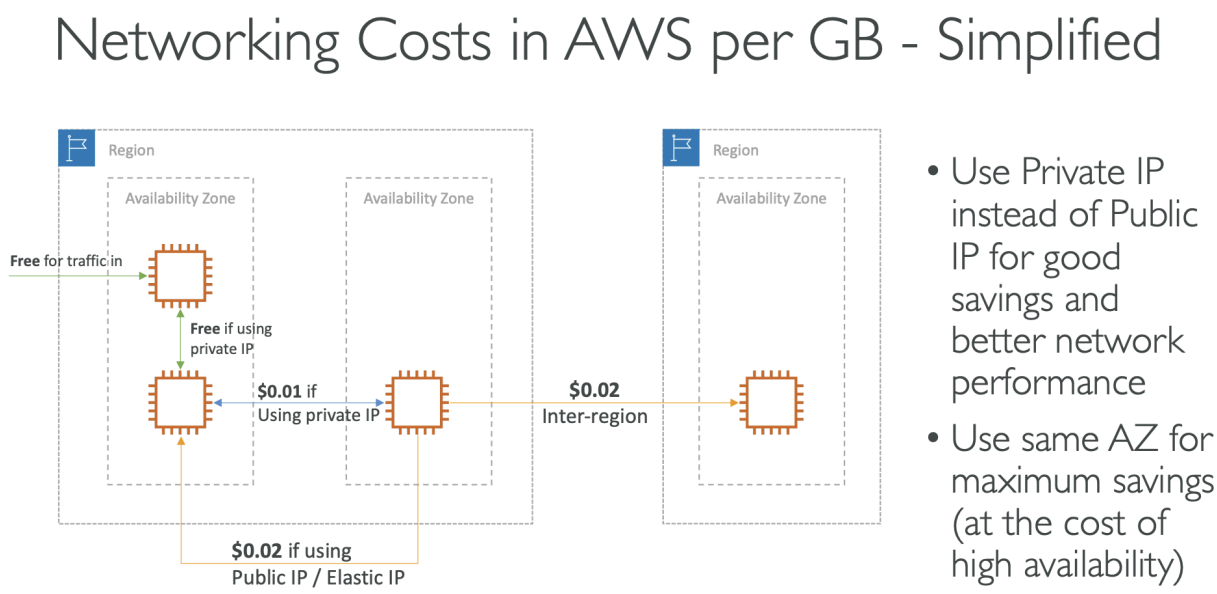
VPC Section Summary (2/3)

- 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 targets

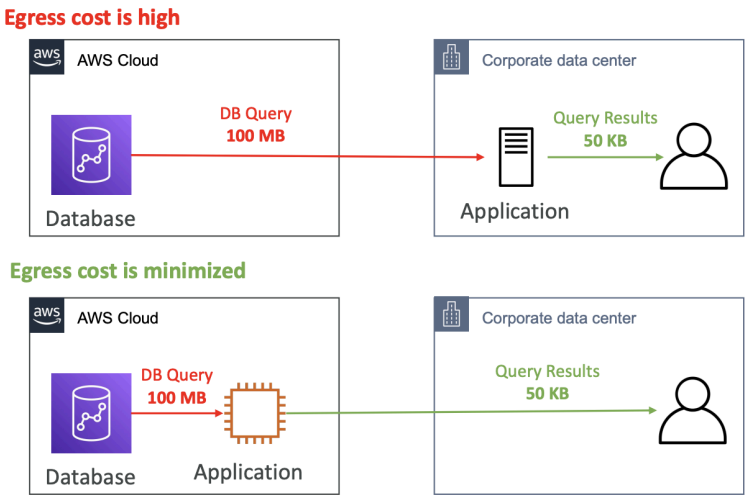
Network Costs in AWS



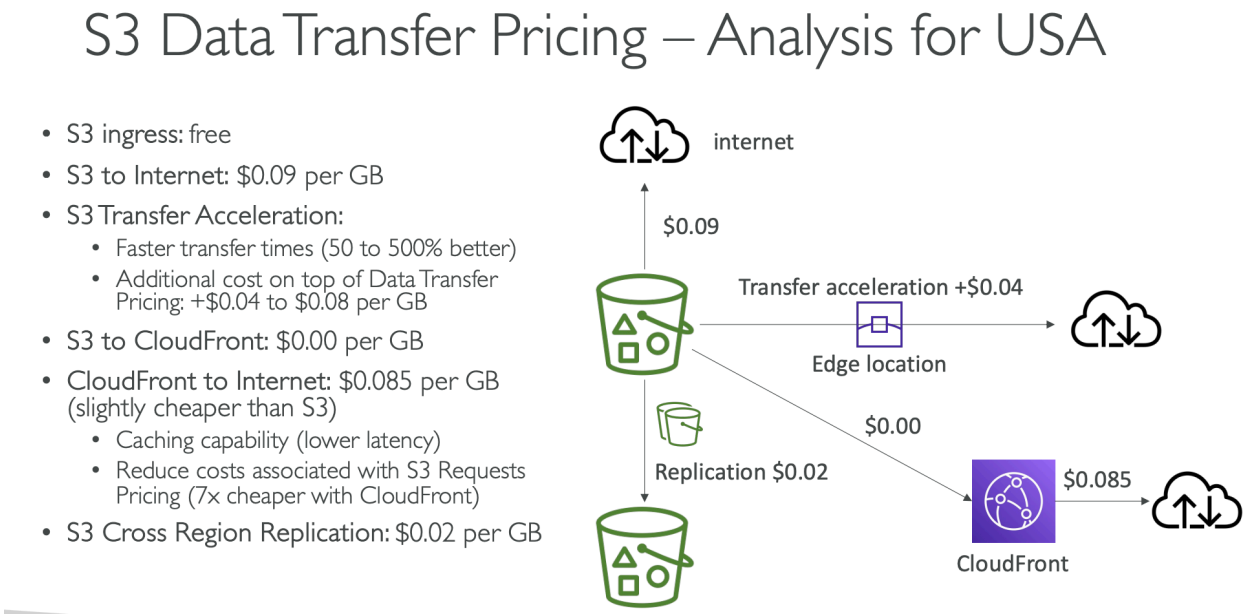
- Use private IP for savings and performance and use same AZ (cost of high availability)

Minimize Egress traffic network cost

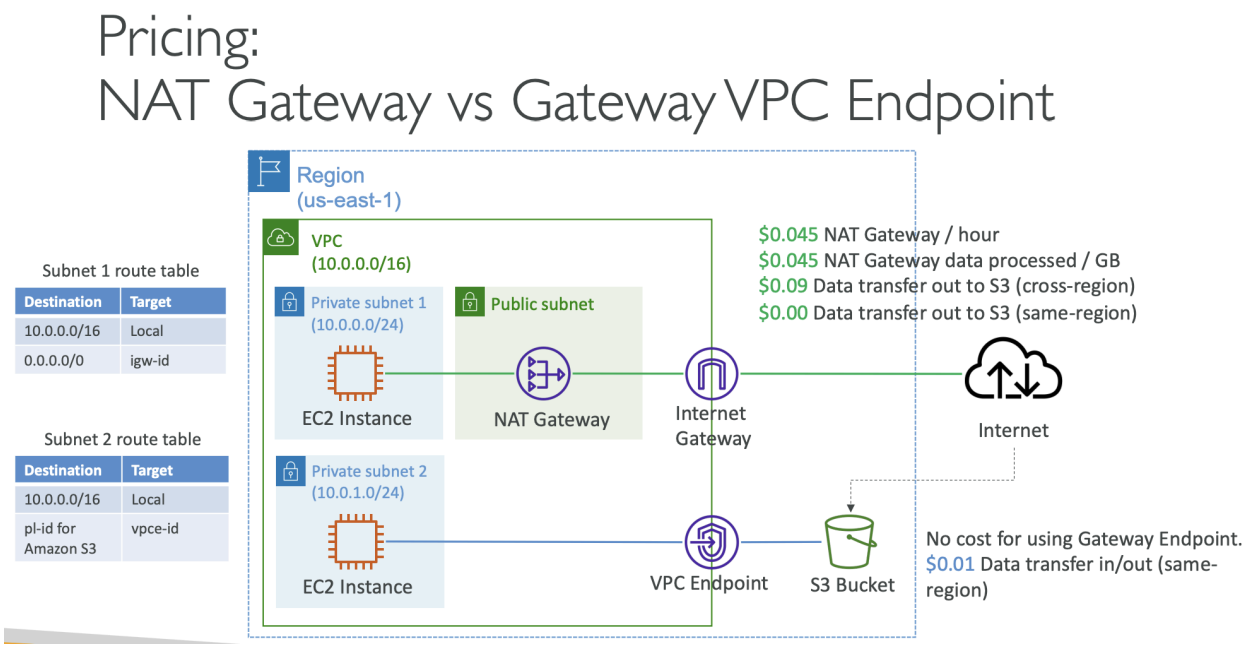
- Egress: outbound traffic (AWS to outside)
- Ingress: inbound (outside to AWS)
  - Keep traffic in AWS for cost saving
- Direct connection location that are co-located in same AWS region result in lower cost for egress network



S3 Data Transfer Pricing



- S3 ingress, to CloudFront = free
- S3 to internet, transfer acceleration, CRR = cost



AWS Network Firewall

- Protect entire VPC from layer 3 to 7
- Any direction, inspecting:

        - VPC to VPC traffic
        - In / outbound traffic
        - To / from Direct Connect & Site to Site VPN
- Internally uses Gateway LB
    - Rules centrally managed cross account by AWS Firewall Manager to apply to many VPCs

Fine Grained Controls

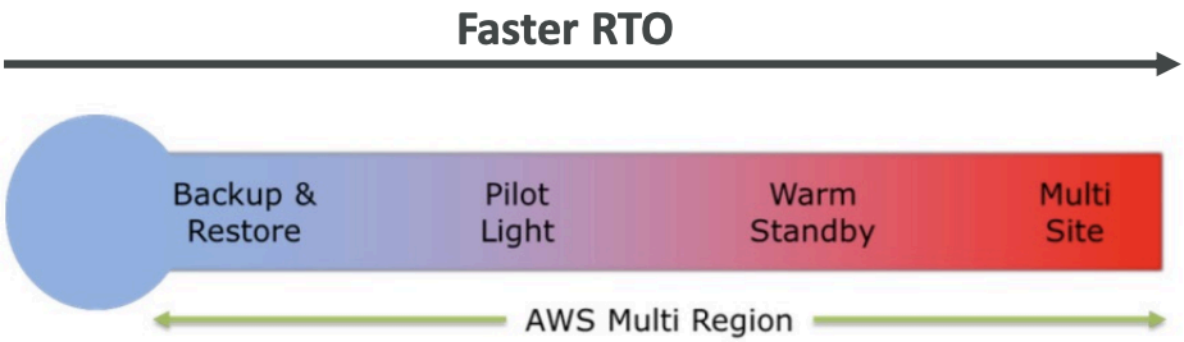
- Supports many rules
    - IP & port, protocol, domain level, general pattern matching
- Traffic filtering: allow, drop, alert for traffic that matches rules
- Active flow inspection to protect against network threats with intrusion prevention capabilities
- Sends logs to S3, CloudWatch Logs, Kinesis Firehose

Section 28: Disaster Recovery & Migrations

Disaster Recovery Overview

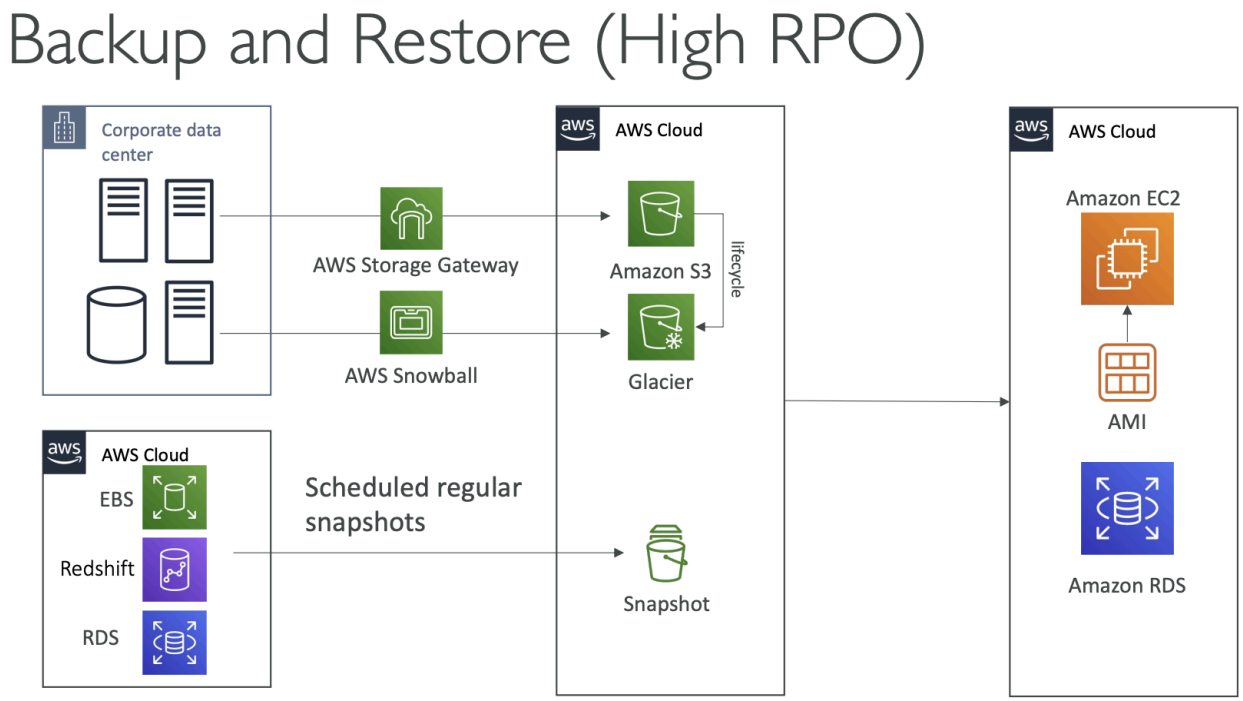
- RPO: recovery point objective
    - How much data loss can be accepted?
- RTO: recovery time objective
    - Downtime application has

Disaster Recovery Strategies



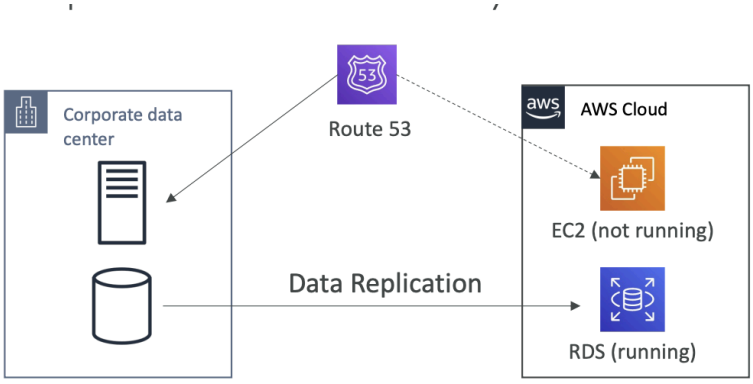
- Backup and restore
- Pilot light
- Warm standby
- Hot / multi sight approach

Backup and Restore (High RPO)



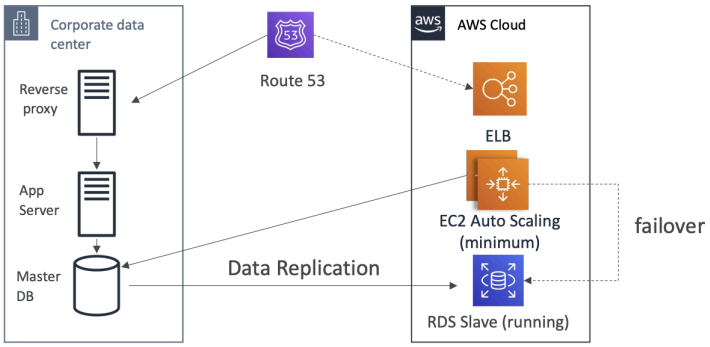
- Inexpensive, high RPO

Pilot Light



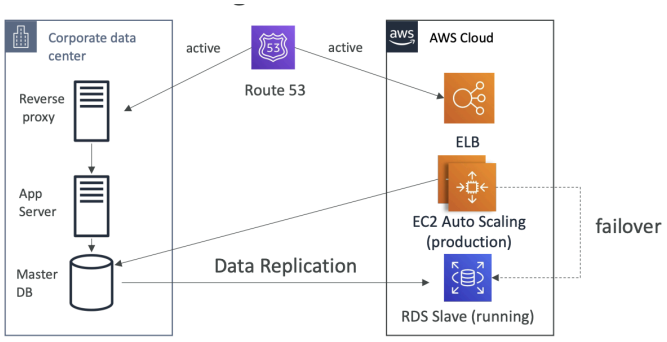
- Small version of app always running on the cloud, similar to backup and restore
- Useful for critical core and faster than backup and restore as critical core is up

Warm Standby



- Full system is running at minimum size and scaled to production load on disaster

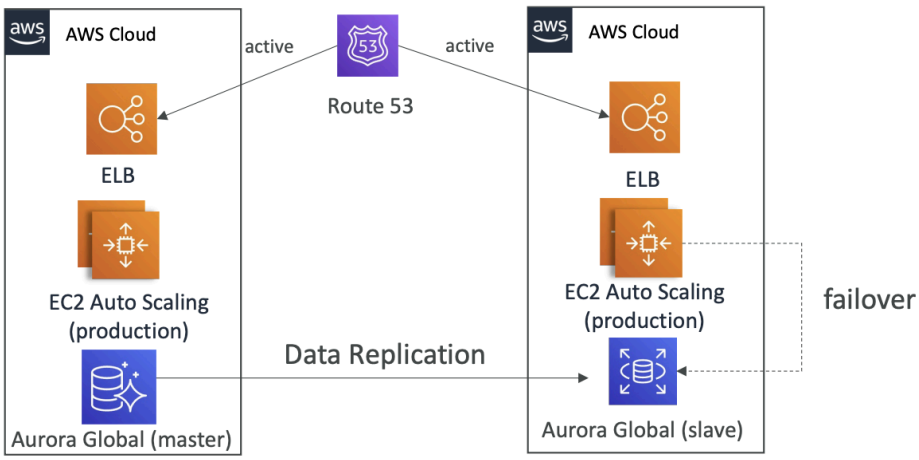
Multi Site / Hot Approach



- Very low RTO, very expensive with full production scale running AWS and on premise

All AWS Multi Region

All AWS Multi Region

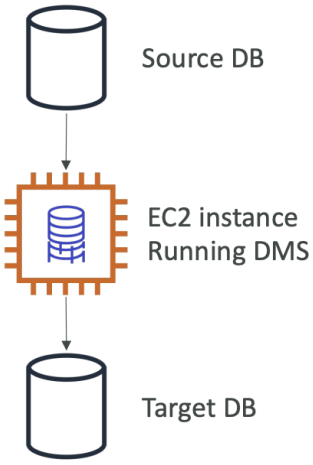


Disaster Recovery Tips

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

Database Migration Service (DMS)



- Quickly and securely migrate DB to AWS, resilient, self healing
- Source DB remains available during migration
- Supports:
  - Homogeneous migrations (ex: Oracle to Oracle)
  - Heterogeneous migrations: (ex: Microsoft SQL Server to Aurora)
- Continuous data replication using CDC (change data capture)
- Must create EC2 instance to perform the replication tasks

DMS Sources and Targets

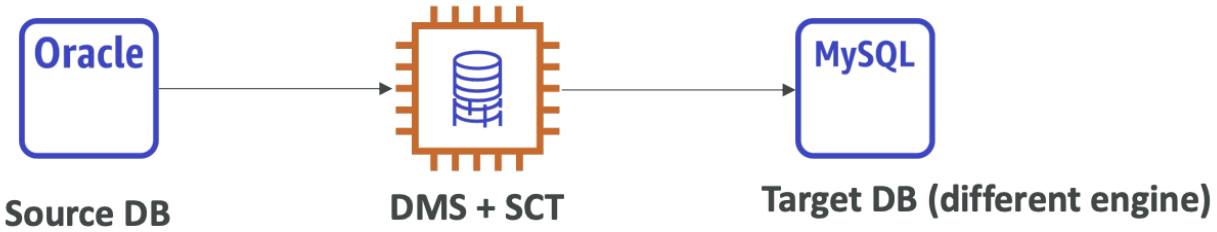
SOURCES:

- 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:

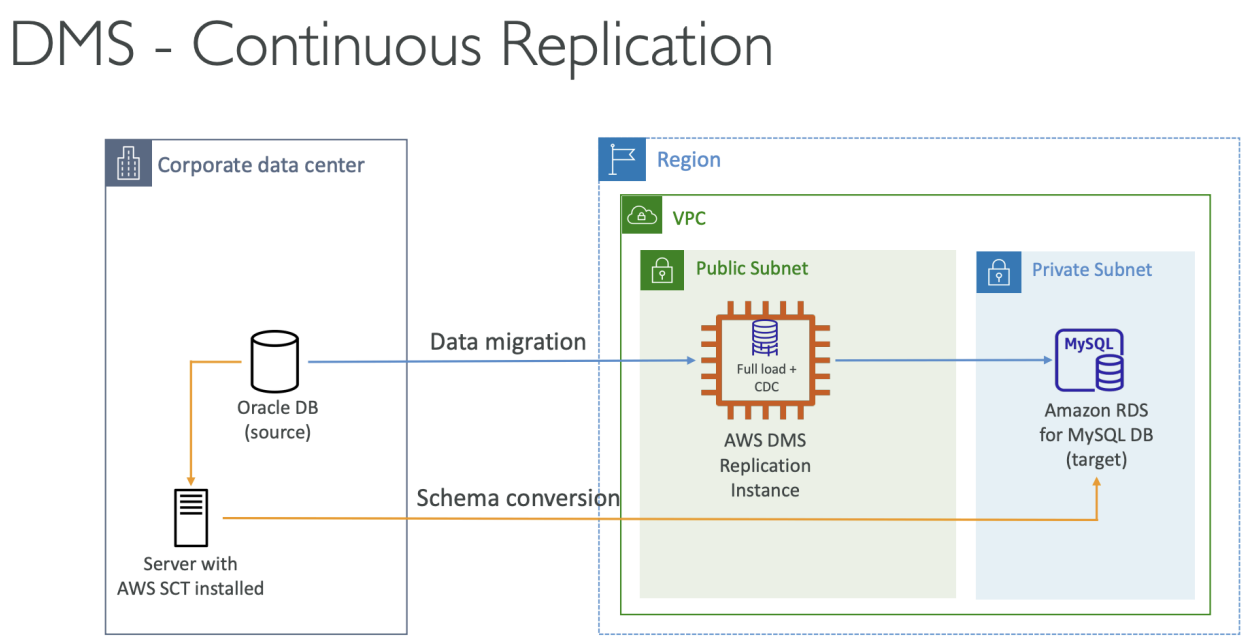
- On-Premises and EC2 instances databases: Oracle, MS SQL Server, MySQL, MariaDB, PostgreSQL, SAP
- Amazon RDS
- Redshift, DynamoDB, S3
- OpenSearch Service
- Kinesis Data Streams
- Apache Kafka
- DocumentDB & Amazon Neptune
- Redis & Babelfish

AWS Schema Conversion Tool (SCT)

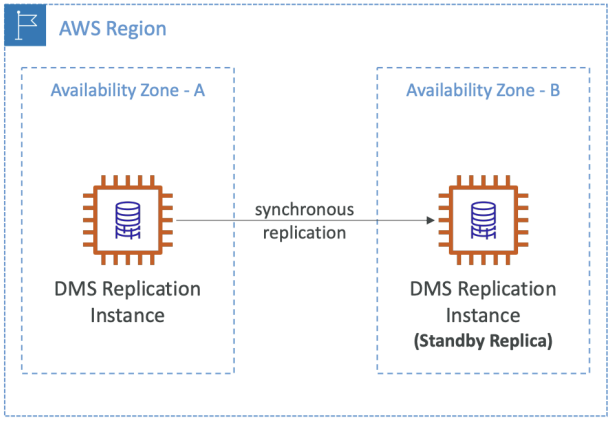


- Convert DB schema from one engine to another
  - Do not need if migrating the same DB engine
  - Ex: Oracle to Aurora
- Prefer compute intensive instances to optimize data conversions

DMS Continuous Replications



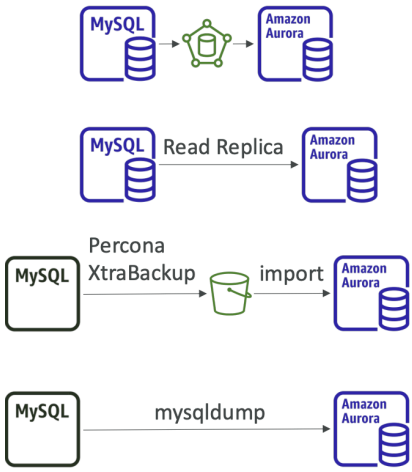
AWS DMS – Multi AZ Deployment



- When multi AZ enabled, DMS provisions and maintains a synchronously stand replica in different AZ
    - Advantages: data redundancy, minimize latency, eliminate I/O freezes

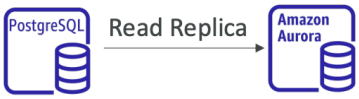
RDS & Aurora MySQL Migrations

- RDS to Aurora MySQL
    - Option 1: DB Snapshots from RDS MySQL restored as MySQL Aurora DB
        - Potential downtime
    - 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 file backup in S3
        - Create Aurora MySQL DB from S3
    - Option 2:
        - Create Aurora MySQL DB and use mysqldump utility to migrate MySQL into Aurora (slower than S3)
- Use DMS if both DB up and running

RDS & Aurora PostgreSQL Migrations

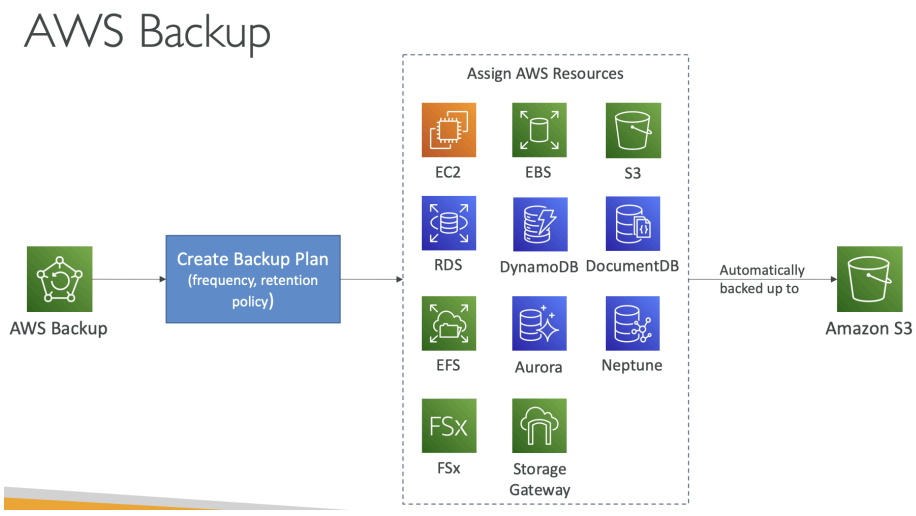


- RDS PostgreSQL to Aurora PostgreSQL \
    - Option 1: DB Snapshots from RDS PostgreSQL restored as PostgreSQL Aurora DB
    - Option 2: Create Aurora read replica from RDS PostgreSQL and when replication lag is 0, promote as own DB cluster
- External PostgreSQL to Aurora PostgreSQL
    - Create backup into S3 and import via aws_s3 Aurora extension
- Use DMS if both DB 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 that centrally manage and automate backups across AWS services
    - No need to create custom scripts and manual processes
- Supports many AWS services, cross region backups, cross account backups
- Supports point in time recovery with on demand and scheduled backups
    - Tag based backup policies
- Can create backup policy known as backup plans
    - Backup frequency
    - Backup window
    - Transition to cold storage
    - Retention period

AWS Backup Vault Lock

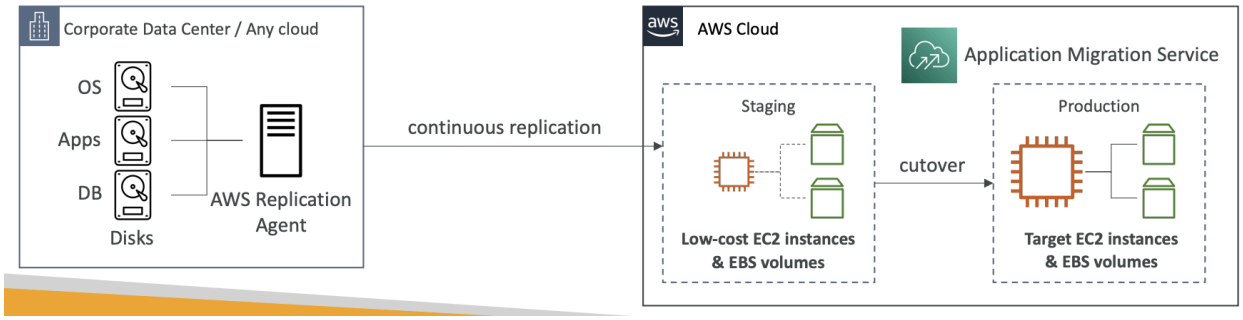
- Enforce WORM (write once read many) state for all backups that you store in AWS Backup Vault
    - Additional layer of defense to protect backups against accidental or malicious delete or updates that shorten or later retention periods
    - Root user cannot delete backup when enabled

AWS Application Discovery Service

- Plan migration projects by gathering info about on premise centers
    - Server utilization data and dependency mapping important for migrations
- Agentless Discovery (AWS Agentless Discovery Connector)
    - VM inventory, configuration, and performance history such as CPU, memory…
- Agent Based Discovery (AWS Application Discovery Agent)

    - System configuration, performance, details of network connections between systems
- Resulting data can be viewed in AWS Migration Hub

AWS Application Migration Service (MGN)

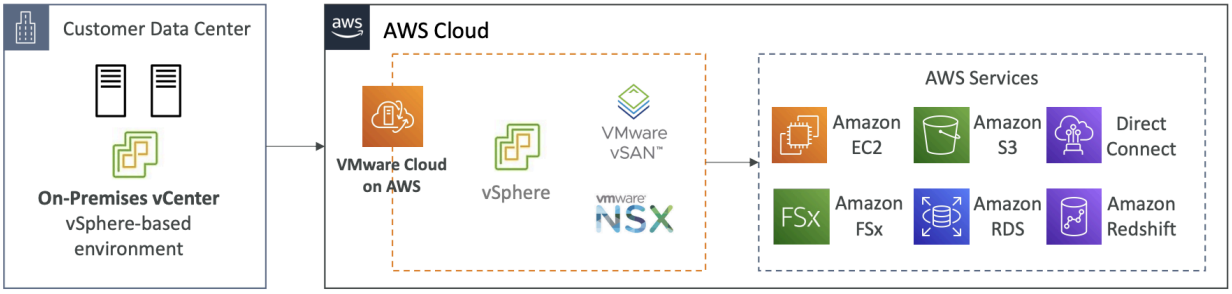


- Lift and shift (rehost) solution which simplify migrating apps to AWS
    - Converts physical, virtual, and cloud based servers to run natively on AWS
        - Supports many platforms, OS, and DB with minimal downtime, reduced cost

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(TB)*1000(GB)*1000(MB)*8(Mb)/100 Mbps = 16,000,000s = 185d
- Over direct connect 1 Gbps:
    - Long for the one-time setup (over a month)
    - Will take 200(TB)*1000(GB)*8(Gb)/1 Gbps = 1,600,000s = 18.5d
- 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

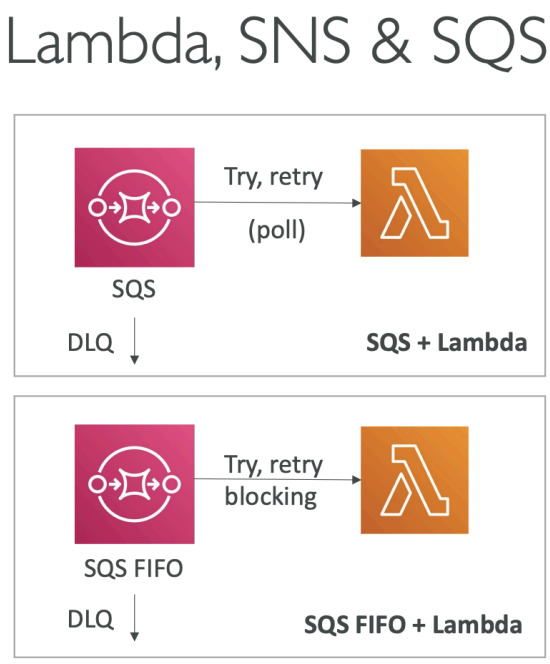
VMware Cloud on AWS

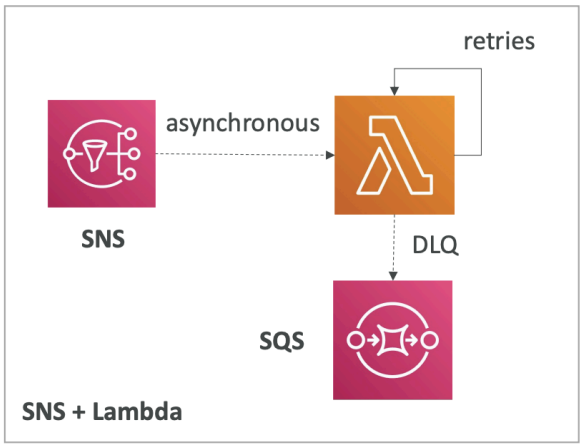


- Migrates VMware vSphere based applications and workloads to AWS
    - Can run production workloads across multiple cloud environments with disaster recovery and AWS services

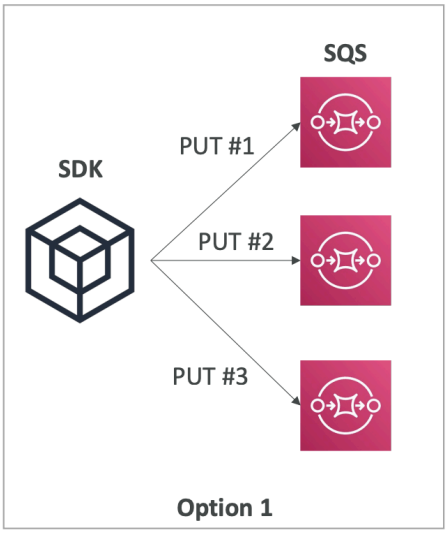
Section 29: More Solution Architectures

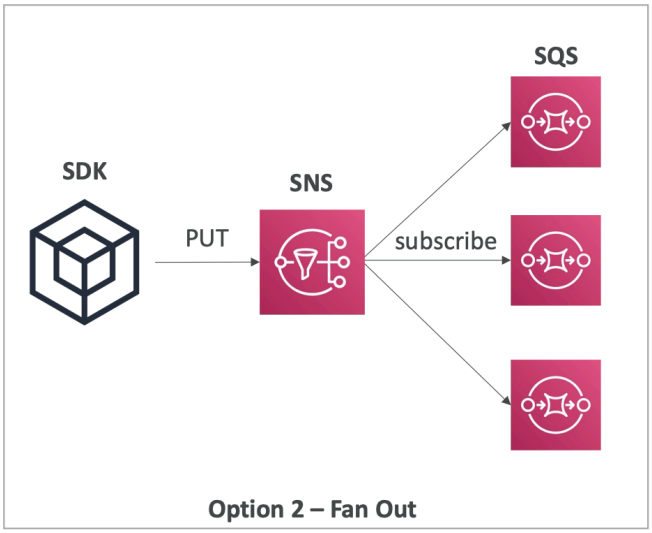
Event based Processing





Fan Out Pattern: deliver to multiple SQS

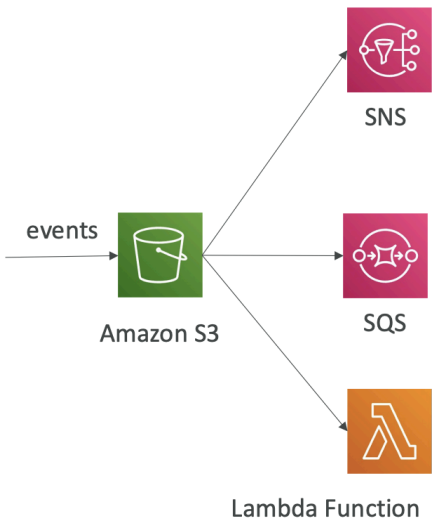




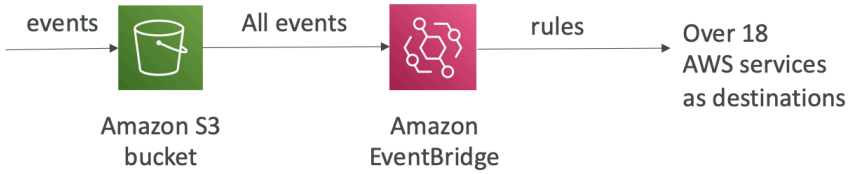
S3 Event Notifications

- 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

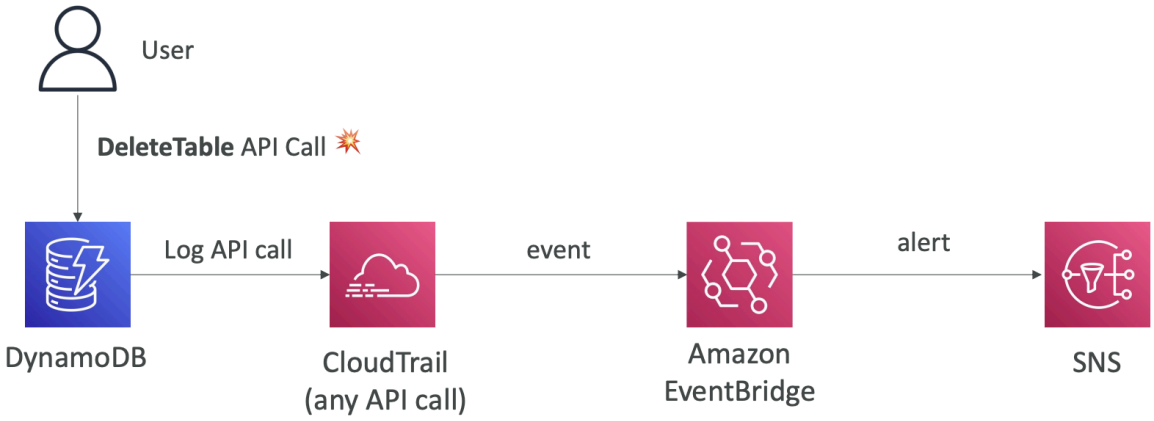


S3 Event Notifications with Amazon EventBridge

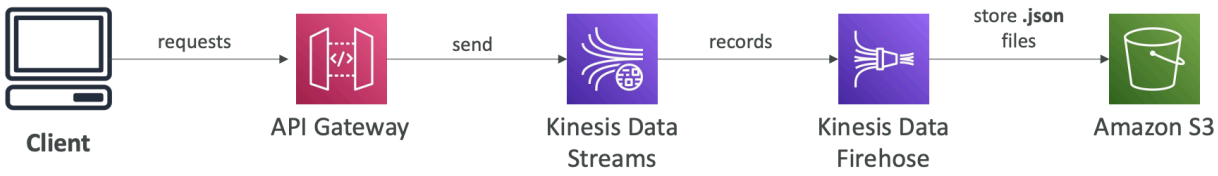


- Advanced filtering options with JSON rules (metadata, object size, name...)
- Multiple Destinations – ex Step Functions, Kinesis Streams / Firehose…
- EventBridge Capabilities – Archive, Replay Events, Reliable delivery

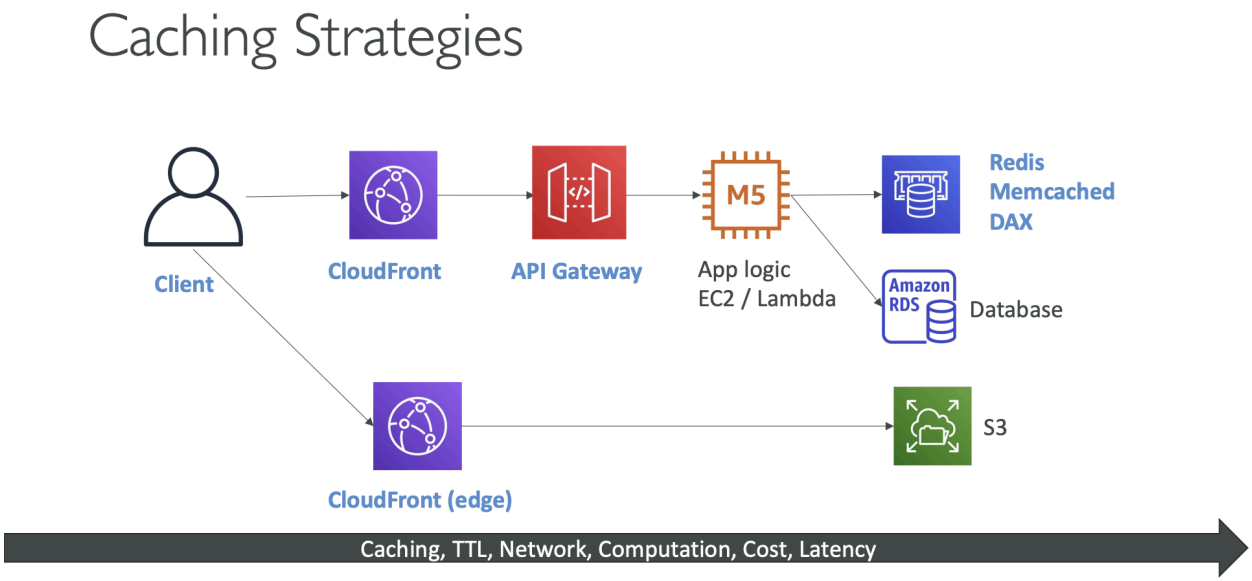
Amazon EventBridge – Intercept API Calls



API Gateway – AWS Service Integration
Kinesis Data Streams example

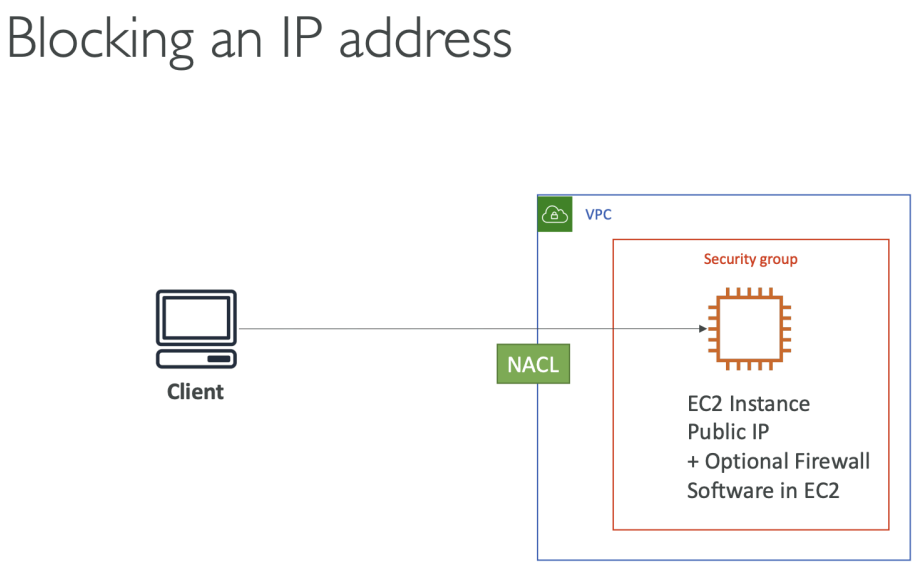


Caching Strategies

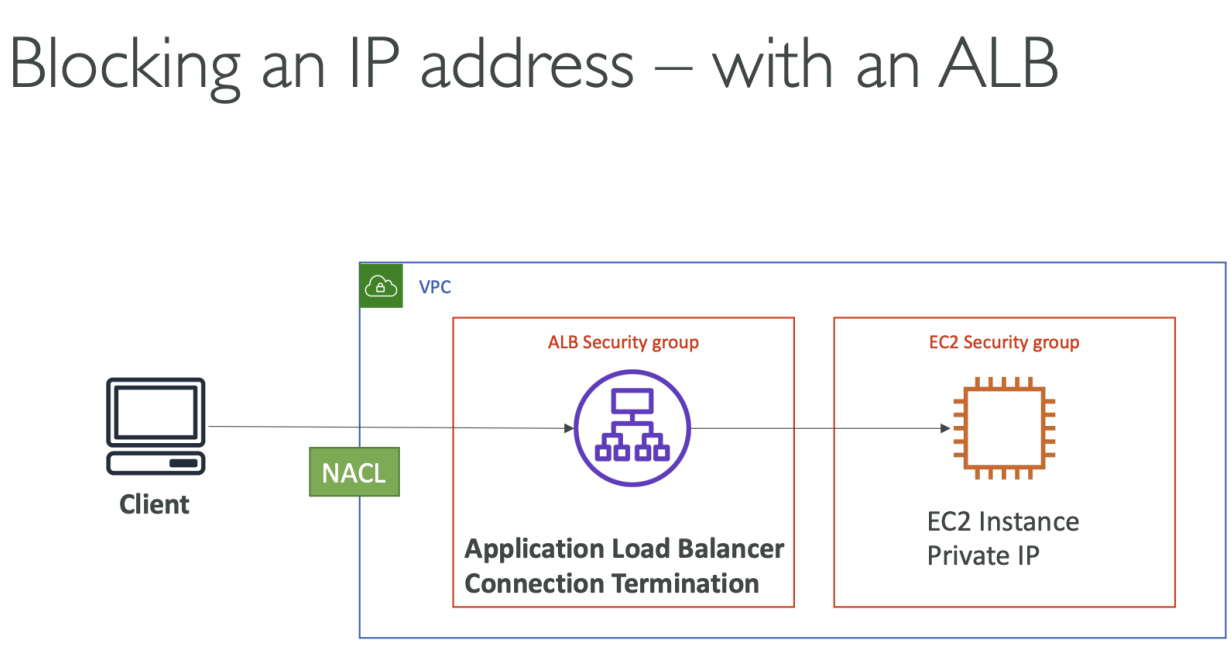


- CloudFront is cached at the edge with the potential of stale data, but has fast return to user
- API GW is regional, thus more lag between client and GW
- App logic cache via ElastiCache or DAX allows high frequent reads to be returned faster

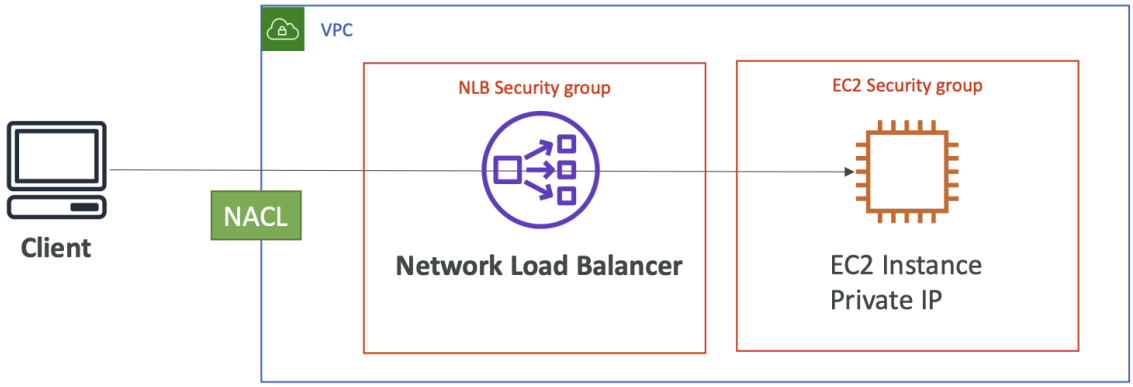
Blocking IP in AWS



- NACL can deny, but SG can only allow. SG can allow a subset of IP, but is not as effective. Since requests reach the instance, there will be a CPI cost to process the request

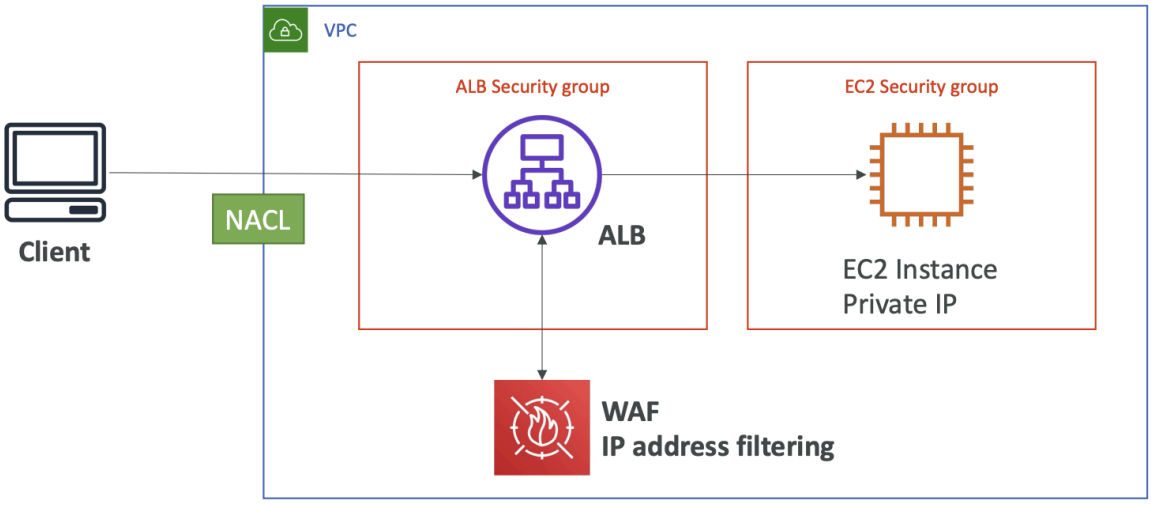


Blocking an IP address – with an NLB



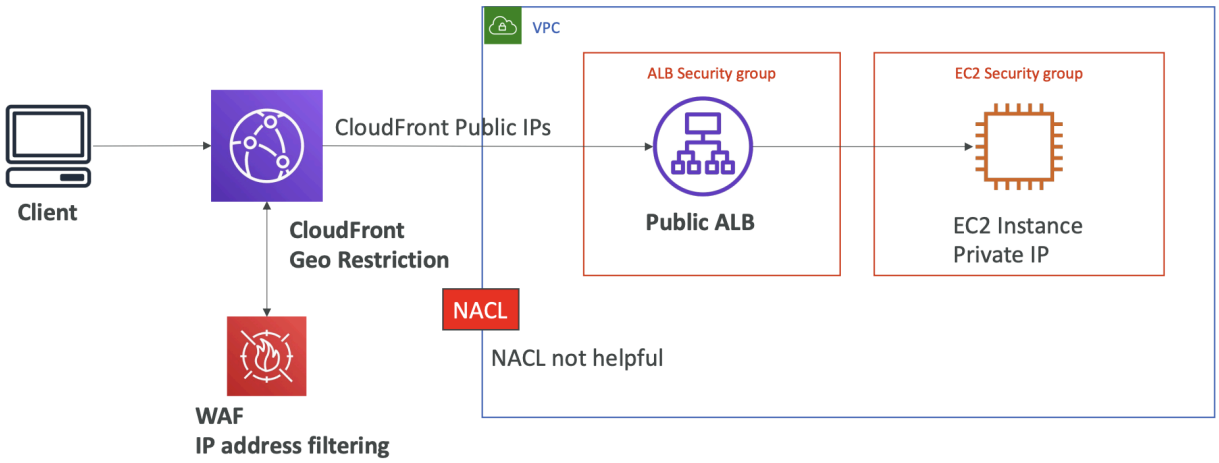
- LB will terminate the client connection and create a new connection from ALB to instance. Instance SG must allow ALB SG as source

Blocking an IP address – ALB + WAF



- WAF can have address filtering that is a service installed on ALB

Blocking an IP address – ALB, CloudFront WAF



- Using CloudFront, the ALB will only see CloudFront IP so NACL on ALB is not helpful. WAF should be on CloudFront instead to see client IPs

High Performance Computing on AWS

- Can create high number of resources and speed up time to results by adding more resources; pay for only what you use

Data Management & Transfer

Data Management & Transfer

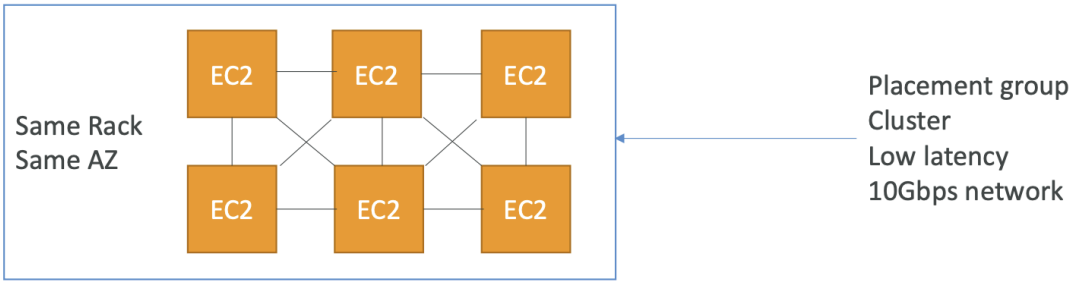
- AWS Direct Connect:
  - Move GB/s of data to the cloud, over a private secure network
- Snowball & Snowmobile
  - Move PB of data to the cloud
- AWS DataSync
  - Move large amount of data between on-premises and S3, EFS, FSx for Windows

Compute and Networking

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

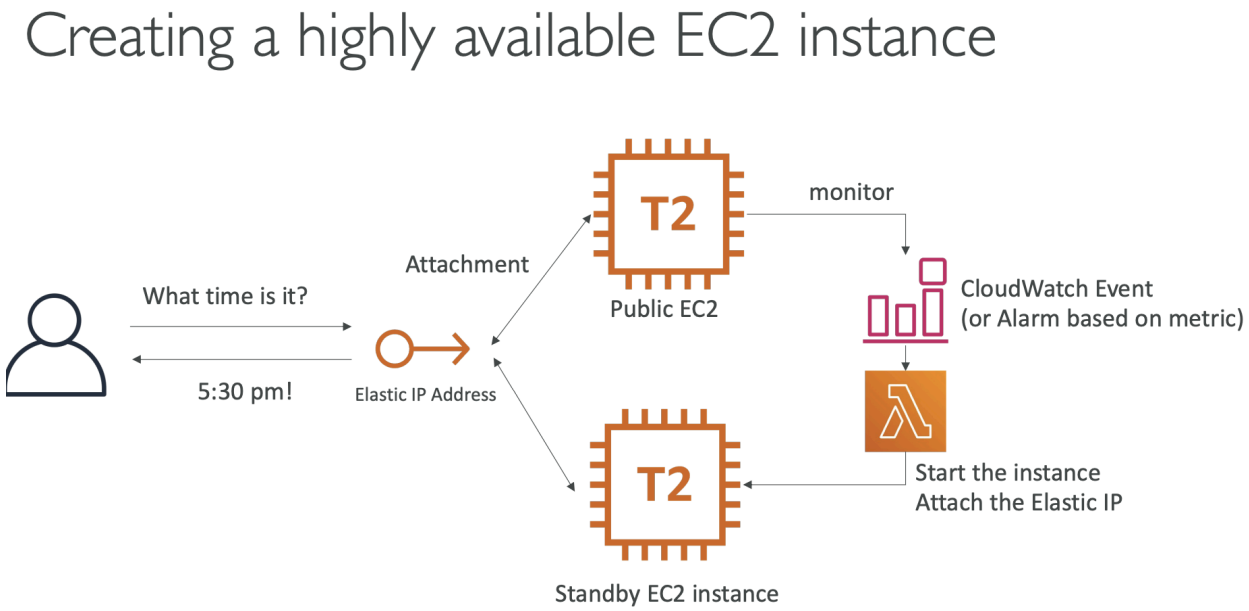
- Network storage:
  - **Amazon S3:** large blob, not a file system
  - **Amazon EFS:** scale IOPS based on total size, or use provisioned IOPS
  - **Amazon FSx for Lustre:**
    - HPC optimized distributed file system, millions of IOPS
    - Backed by S3

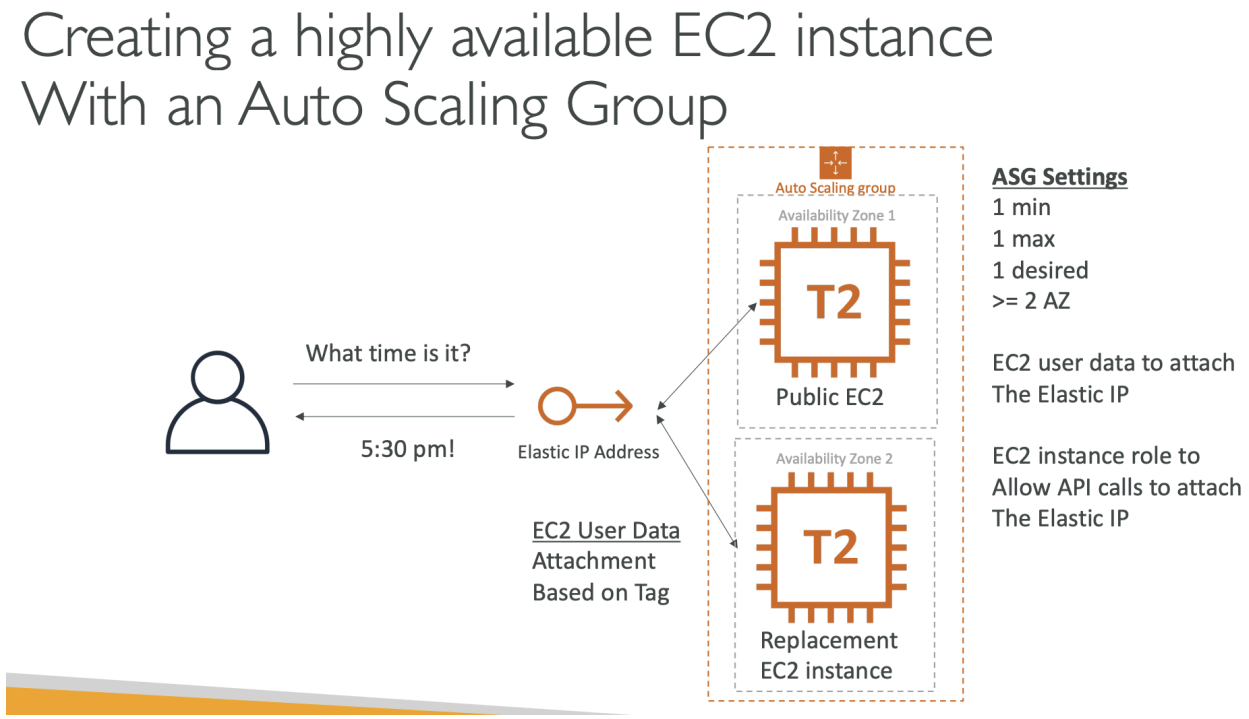
Automation and Orchestration

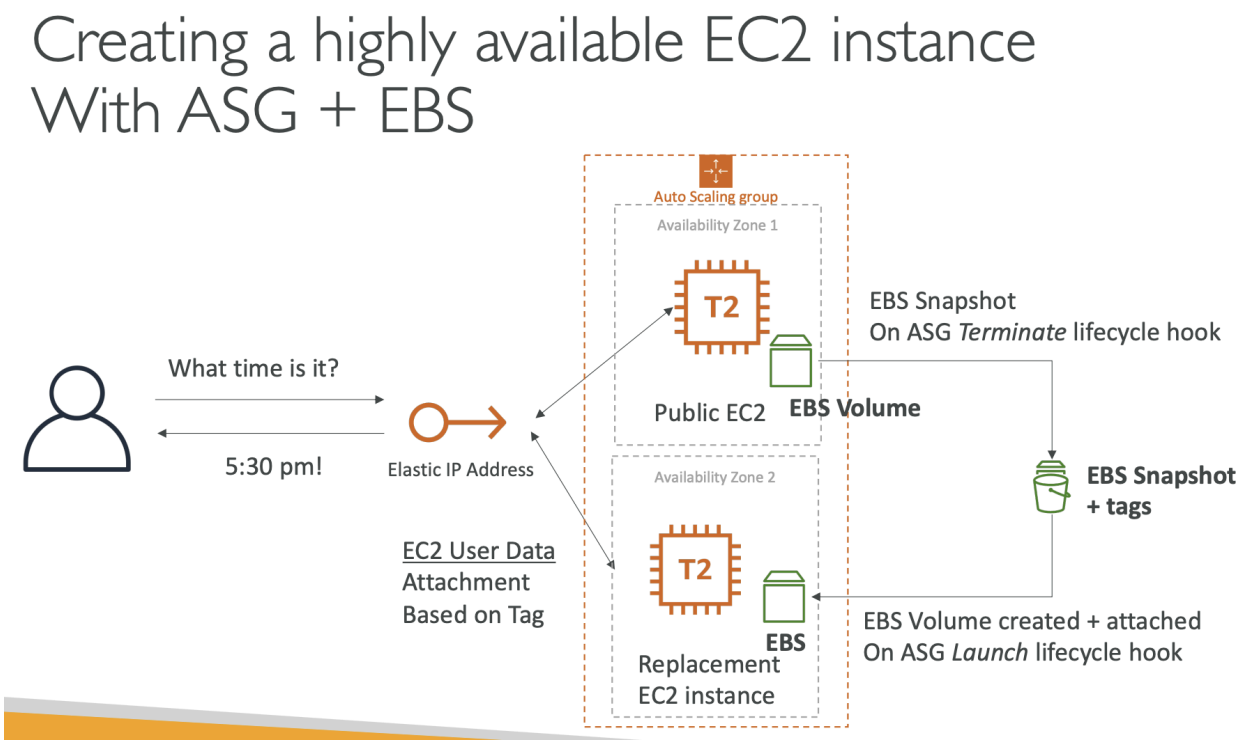
- AWS Batch
  - **AWS Batch** supports multi-node parallel jobs, which enables you to run single jobs that span multiple EC2 instances.
  - Easily schedule jobs and launch EC2 instances accordingly

- 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)

EC2 Instance High Availability







Section 30: Other Services

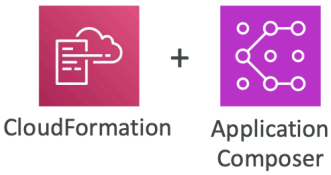
CloudFormation Intro

- Declarative way to outline AWS infrastructure via code, created in the right order, declaratively
- IaaC
    - No resources manually created, version control with code
- Cost: all resources within the stack is tagged with in identifier to see cost
    - Can estimate costs
    - Can create and destroy
- Productivity:
    - Destroy and recreate infrastructure on the fly
- Separation of concern: create many stacks for many apps and many layers
- Don't reinvent the wheel with existing templates and documentation

CF + Application Composer

CloudFormation + Application Composer

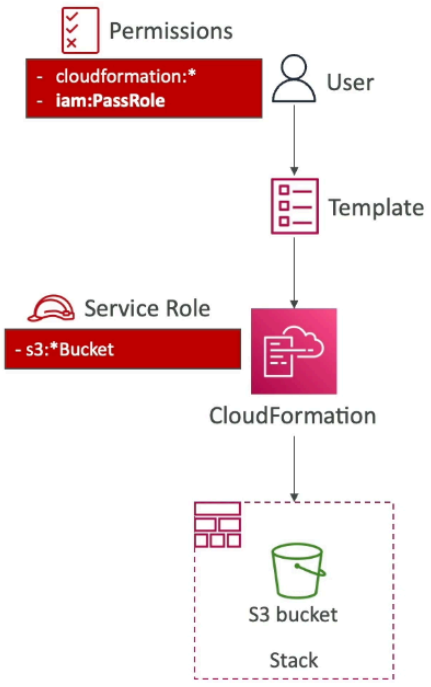
- Example: WordPress CloudFormation Stack
- We can see all the resources
- We can see the relations between the components



CF – Service Roll

CloudFormation – Service Role

- IAM role that allows CloudFormation to create/update/delete stack resources on your behalf
- Give ability to users to create/update/delete the stack resources even if they don't have permissions to work with the resources in the stack
- Use cases:
  - You want to achieve the least privilege principle
  - But you don't want to give the user all the required permissions to create the stack resources
- User must have iam:PassRole permissions



- IAM role that allows CF to create / update / delete stack resources on behalf
  - Gives users ability to update resources even without permissions to work with the resources in the stack

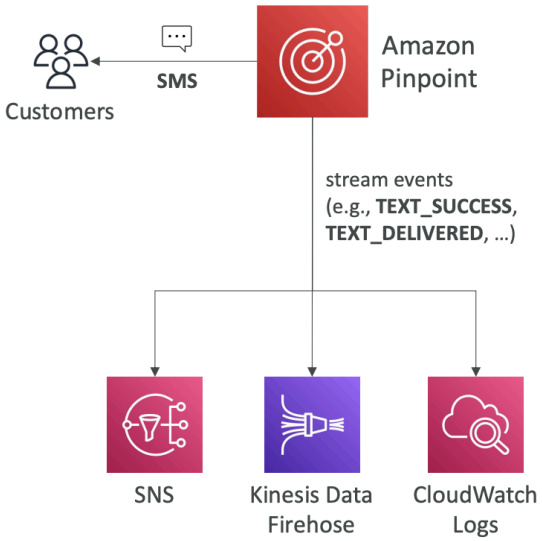
- User must have iam:PassRole Permissions

AWS Simple Email Service (SES)

- Send emails and ability receive email
  - Integrates with S3, SNS, Lambda and IAM for allowing to send emails
  - Allows in/outbound emails
  - Send emails using app via Console or API or SMTP
- Reputation dashboard, performance insights, anti spam feedback
  - Statistics such as email deliveries, bounces, email open…
- Supports DomainKeys Identified Mail (DKIM) and Sender Policy Framework (SPF)
- Flexible IP deployment: shared, dedicated, and customer owned IP
- Use case: transactional, marketing, bulk email communications

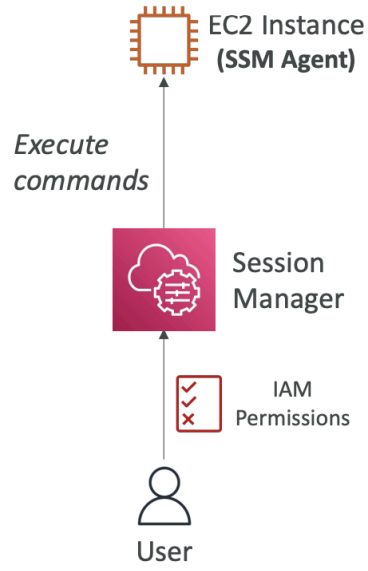
Amazon Pinpoint

- Scalable 2 way (outbound/inbound) marketing communications service that supports email, SMS, push, voice, and in app messaging
  - Can segment and personalize messages
  - Can receive replies
  - Scales to billions of messages
- Use cases: run campaigns by sending marketing, bulk, transactional SMS messages
- vs SNS or SES?
  - SNS & SES you manage each message's audience, content and delivery schedule
  - Pinpoint can create message templates, delivery schedules, highly-targeted segments and full campaigns



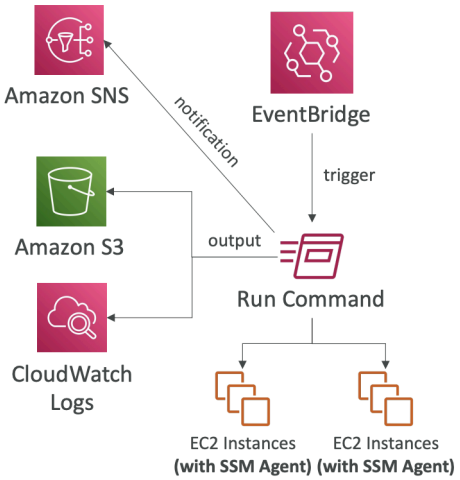
Systems Manager – SSM Session Manager

- Allows to start a secure shell to EC2 and on premise servers without SSH, bastion host, or SSH keys (no port 22 needed, better security)
  - Supports all OS, can send session log data to S3, or CloudWatch Logs



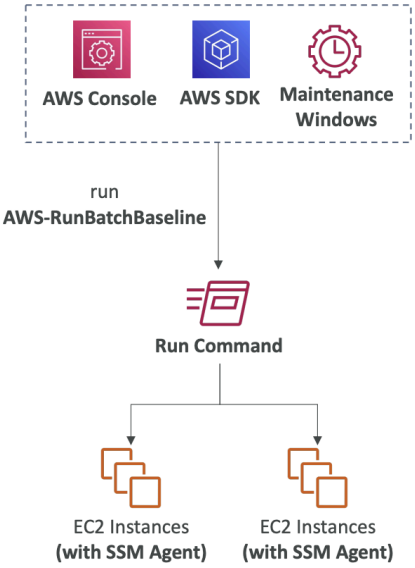
SSM – Run Command

- Execute document (script) or just run a command
  - Run command across multiple instances (using resource groups)
  - No need for SSH
  - Command output sent to S3 or CloudWatch Logs
  - Send notifications to SNS about command status
  - Integrated with IAM & CloudTrail
  - Can be invoked via EventBridge



SSM – Patch Manager

- Automated patching managed instances
  - OS updates, app updates, security updates
  - Supports EC2 instances, on premise; all OS
- Patch on demand or schedule via Maintenance Windows
- Scan instances and generate patch compliance report (missing patches)

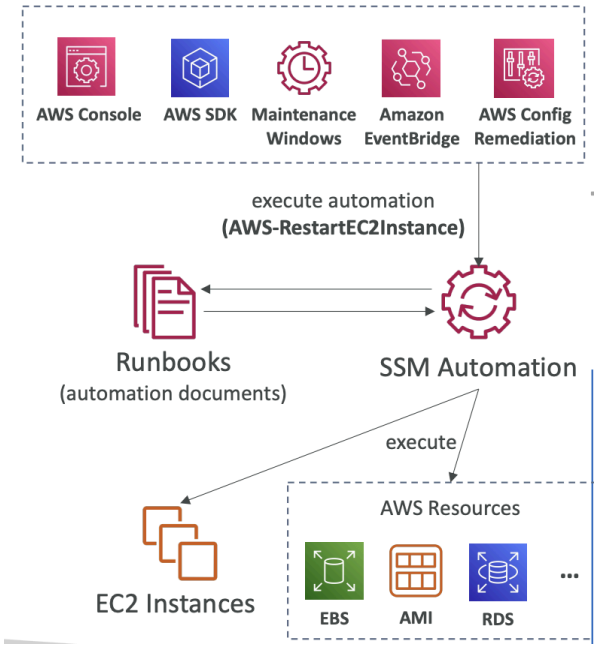


SSM – Maintenance Windows

- Schedule for when to perform actions on instances
  - Ex: OS patching, update drivers, install software…
- Contains:
  - Schedule, duration, set of registered instances, set of registered tasks

SSM – Automation

- Simplified common maintenance and deployment tasks of EC2 instances and other AWS resources
    - Ex: restart instances, create AMI, EBS snapshot
- Automation Runbook: SSM Documents for predefined actions on EC2 instances or AWS resources



Cost Explorer

- Visualize, understand and manage AWS costs and usage over time
- Create custom reports that analyze cost and usage data
    - Analyze data at high level: total costs and usage across all accounts
    - Hourly, monthly, resource level granularity
- Choose optimal savings plan
- Forecast usage up to 12 months based on previous usage

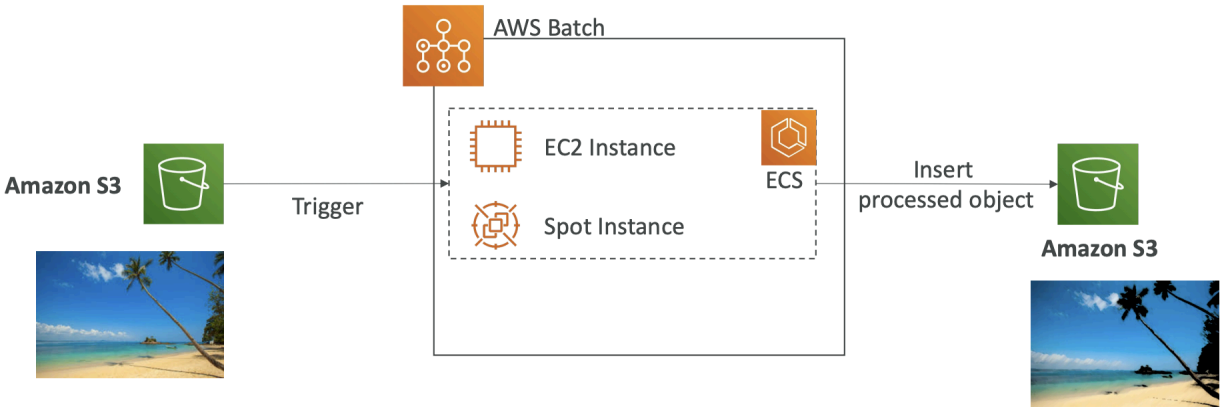
AWS Cost Anomaly Detection

- Continuously monitor cost and usage using ML to detect unusual spends
    - Learns unique historical patterns to detect one time cost spike and / or continuous cost increases (no threshold needed)
- Monitor AWS services, member accounts, cost allocation tags, cost categories
- Sends anomaly detection report with root cause analysis
    - Notified with individual alerts or daily / weekly summary (SNS)

AWS Batch

- Fully managed batch processing at any scale; can efficiently run 100,000s of computing batch jobs on AWS
- Batch is a job with a start and end (not continuous)
    - Batch will dynamically launch EC2 instances or Spot Instances
    - AWS Batch provisions right amount of compute / memory
    - Just submit or schedule batch jobs and AWS Batch does the rest
- Batch jobs defined as docker images and run on ECS
- Helpful for cost optimizations and focusing less on infrastructure

AWS Batch – Simplified Example



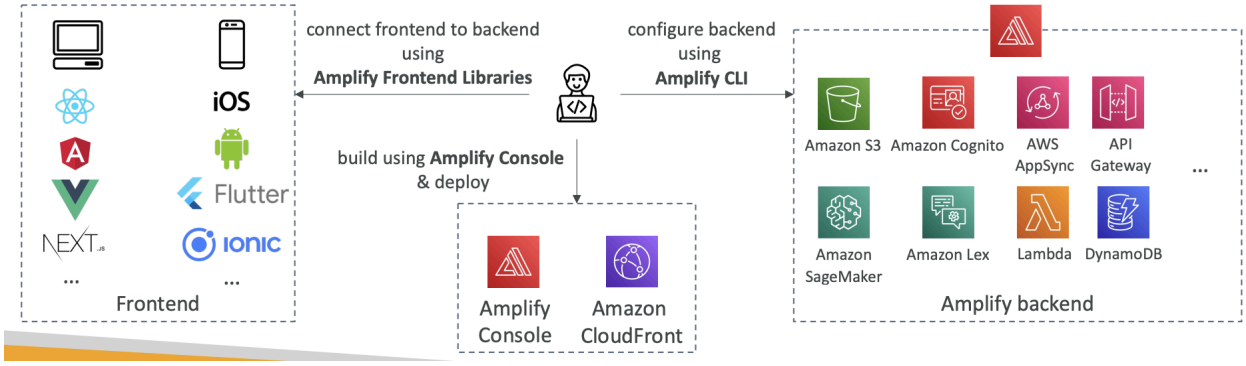
Batch vs Lambda

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

Amazon AppFlow

- Fully managed integration service that enables you to securely transfer data between SaaS apps and AWS
  - Source: Salesforce, SAP, Slack, ServiceNow…
  - Destination: S3, Redshift, or non-AWS like Snowflake and Salesforce
  - Frequency: scheduled, response to events, or on demand
  - Data transformation capabilities like filtering and validation
  - Encrypted over public internet or privately over AWS PrivateLink
  - No time writing integrations and leverage APIs

AWS Amplify



- Create mobile and web apps → Elastic beanstalk for mobile and web apps powered by CloudFormation
- Relies on DynamoDB, AppSync, Cognito, S3 as backend with any frontend library

Section 31: Whitepapers and Architectures

Well Architected Framework

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

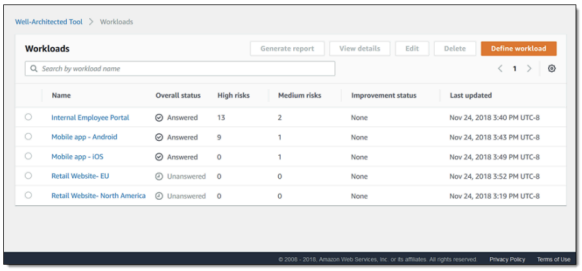
- 1) Operational Excellence
- 2) Security
- 3) Reliability
- 4) Performance Efficiency
- 5) Cost Optimization
- 6) Sustainability

- They are not something to balance, or trade-offs, they're a synergy

AWS Well Architected Tool

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



AWS Trusted Advisor

Trusted Advisor

- No need to install anything – high level AWS account assessment
- Analyze your AWS accounts and provides recommendation on 6 categories:
  - Cost optimization
  - Performance
  - Security
  - Fault tolerance
  - Service limits
  - Operational Excellence
- Business & Enterprise Support plan
  - Full Set of Checks
  - Programmatic Access using AWS Support API