

# AWS Certified SysOps Administrator Associate

By Stéphane Maarek



COURSE →



EXTRA PRACTICE EXAMS →

# Disclaimer: These slides are copyrighted and strictly for personal use only

- This document is reserved for people enrolled into the [AWS Certified SysOps Administrator Associate course by Stephane Maarek](#).
- **Please do not share this document**, it is intended for personal use and exam preparation only, thank you.
- If you've obtained these slides for free on a website that is not the course's website, please reach out to [piracy@datacumulus.com](mailto:piracy@datacumulus.com). Thanks!
- Best of luck for the exam and happy learning!

# AWS Certified SysOps Administrator Associate Course SOA-C02

# Welcome!

- We're going to prepare for the SysOps exam **SOA-C02**
- It's a challenging certification, this course will be long and interesting 😊
- **How the course is structured:**
  - If some base knowledge is needed: videos are imported from
    - Certified Cloud Practitioner, ex: *[CCP] EC2 Instances Launch Types*
    - Certified Solutions Architect Associate, ex: *[SAA] EC2 Hibernate Hands On*
    - Certified Developer Associate, ex: *[DVA] CloudFormation Drift*
  - SysOps-specific videos (not marked with [...])

# AWS Certified SysOps Exam Labs

- Starting with SOA-C02, the exam will contain 3 labs
- Each exam lab consists of several different tasks
- AWS recommends to allocate 20 minutes for each exam lab
- First, you'll need to answer the multiple-choice/multiple-responses questions (you can't go back once you have completed this section)
- Second, you'll need to answer the exam labs
- You must complete all the work on the exam lab, before you move to the next one (you won't be able to go back to a lab once you have completed it)

# AWS Certified SysOps Exam Sample Lab

The screenshot displays the AWS Management Console interface for a sample lab. The main content area is titled "AWS Management Console" and features several sections:

- AWS services:** A search bar for finding services, with an example: "Example: Relational Database Service, database, RDS".
- Build a solution:** A grid of six solution cards, each with an icon, title, and duration:
  - Launch a virtual machine (With EC2, 2-3 minutes)
  - Build a web app (With Elastic Beanstalk, 6 minutes)
  - Build using virtual servers (With Lightsail, 1-2 minutes)
  - Register a domain (With Route 53, 3 minutes)
  - Connect an IoT device (With AWS IoT, 5 minutes)
  - Start migrating to AWS (With CloudEndure Migration, 1-2 minutes)
- Stay connected to your AWS resources on-the-go:** A section for downloading the AWS Console Mobile App.
- Explore AWS:** Promotional cards for Amazon S3 Glacier, Free Digital Training, and Amazon Elasticsearch Service.

On the right side, a sidebar contains the lab instructions:

- Requirements:**
  - Use the **us-east-2** Region for all resources.
  - Unless specified in the instructions below, use default configuration settings.
- Scenario:** "A company is deploying a new web application that requires a database. Configure a highly available Amazon RDS for MySQL DB instance (v6.0)."
- Steps:**
  - Create a custom DB parameter group and set the **event\_scheduler** parameter to true and use this parameter group during DB instance creation.
  - Create a custom KMS key and use this key during DB instance creation.
  - Create a VPC Security Group that allows TCP/3306 from the CIDR range 192.168.1.0/24 and use this security group during DB instance creation.
  - Create and Launch the RDS DB instance.
  - After launch, take a manual RDS DB snapshot.
  - Provide the DB snapshot ARN:
- Important:** Click the **Next** button to complete this lab and continue to the next lab. Once you click the **Next** button, you will **NOT** be able to return to this lab.

# How to prepare for the labs?

Follow along with me and practice in the course

# About me

- I'm Stephane!
- 9x AWS Certified
- Worked with AWS many years: built websites, apps, streaming platforms
- Veteran Instructor on AWS (Certifications, CloudFormation, Lambda, EC2...)
- You can find me on
  - GitHub: <https://github.com/simplesteph>
  - LinkedIn: <https://www.linkedin.com/in/stephanemaarek>
  - Medium: <https://medium.com/@stephane.maarek>
  - Twitter: <https://twitter.com/stephanemaarek>
  - Instagram: <https://instagram.com/stephanemaarek>



- ★ 4.7 Instructor Rating
- 🏆 289,838 Reviews
- 👤 904,779 Students
- ▶ 41 Courses



# EC2 for SysOps

Rocking EC2 from a SysOps perspective

# EC2 Changing Instance Type



- This only works for EBS backed instances
- Stop the instance
- Instance Settings => Change Instance Type
- Start Instance

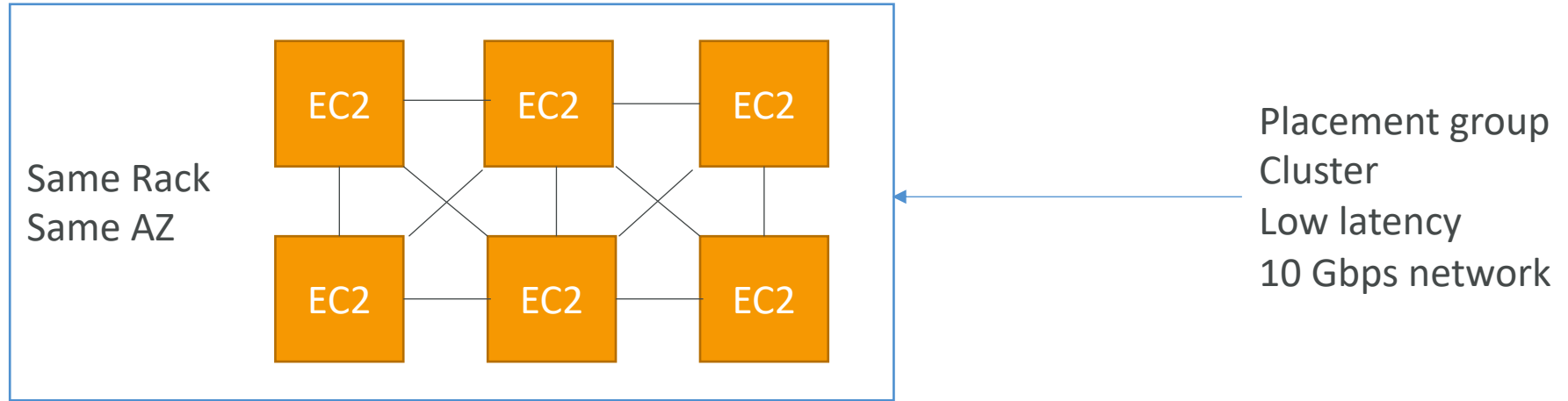
# Enhanced Networking

- **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
  - Works for newer generation EC2 Instances
- **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

# Placement Groups

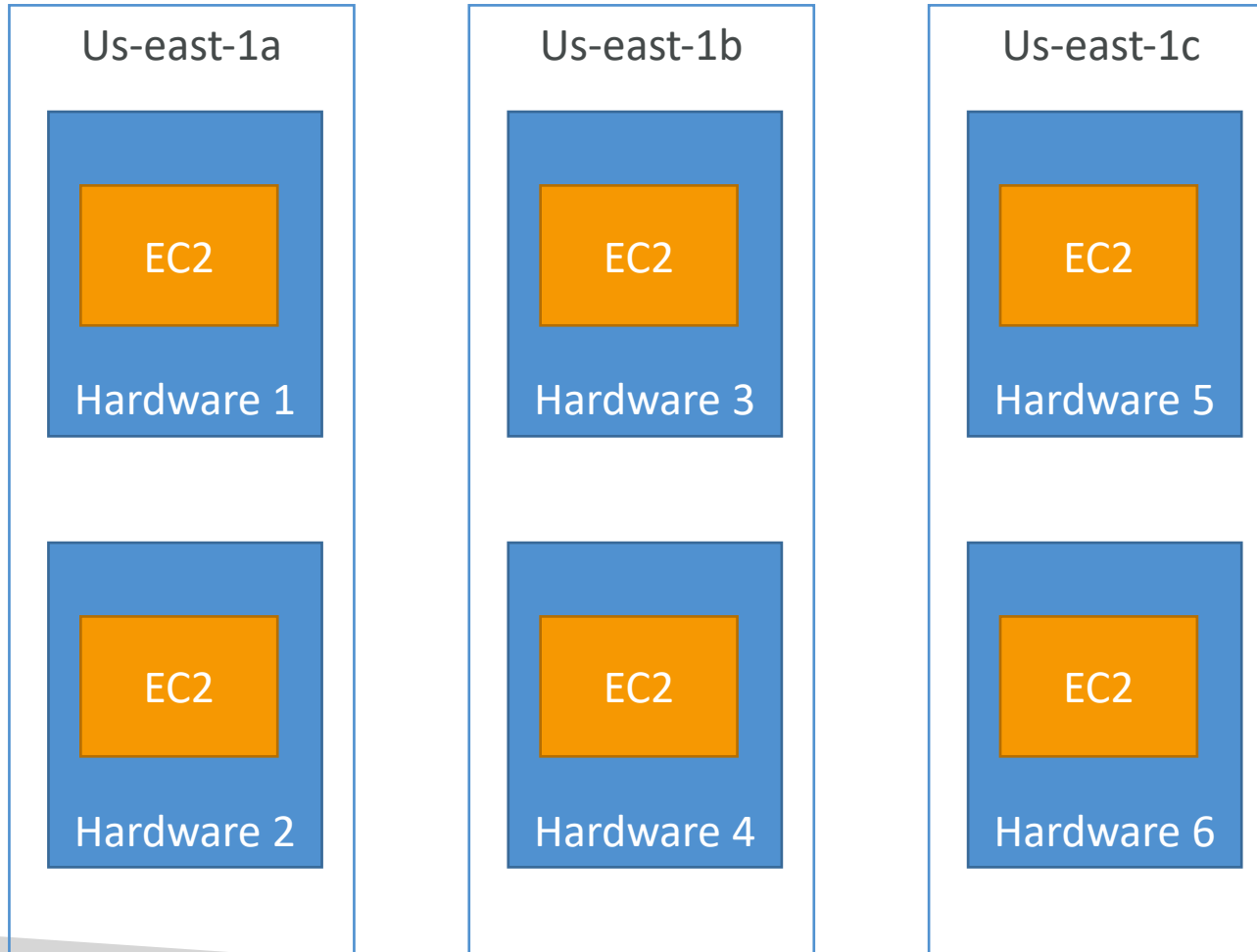
- Sometimes you want control over the EC2 Instance placement strategy
- That strategy can be defined using placement groups
- When you create a placement group, you specify one of the following strategies for the group:
  - *Cluster*—clusters instances into a low-latency group in a single Availability Zone
  - *Spread*—spreads instances across underlying hardware (max 7 instances per group per AZ) – critical applications
  - *Partition*—spreads instances across many different partitions (which rely on different sets of racks) within an AZ. Scales to 100s of EC2 instances per group (Hadoop, Cassandra, Kafka)

# Placement Groups Cluster



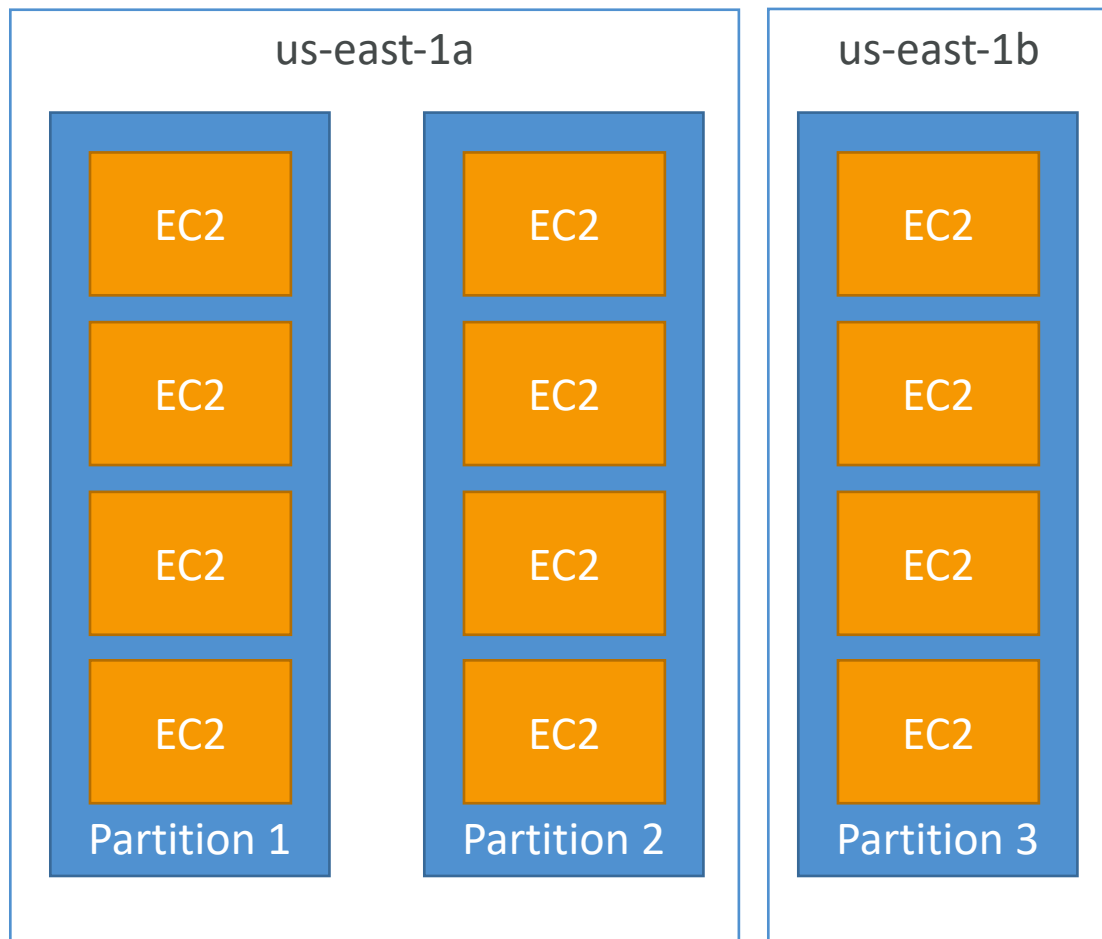
- Pros: Great network (10 Gbps bandwidth between instances with Enhanced Networking enabled - recommended)
- Cons: If the rack fails, all instances fails at the same time
- Use case:
  - Big Data job that needs to complete fast
  - Application that needs extremely low latency and high network throughput

# 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

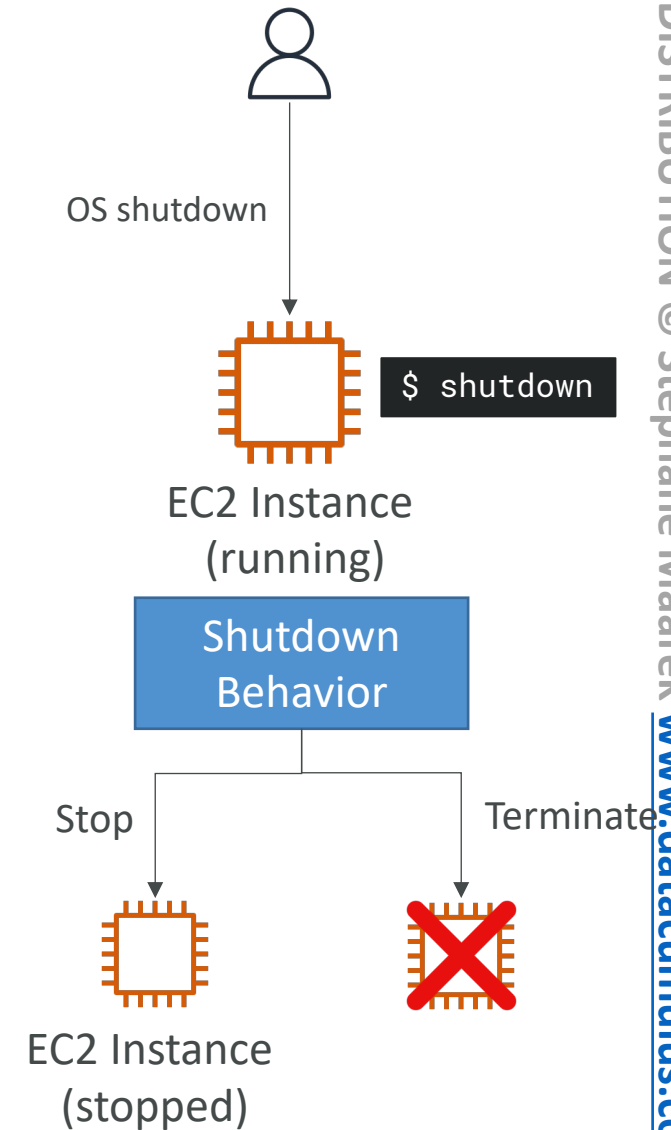
# Placements Groups Partition



- Up to 7 partitions per AZ
- Can span across multiple AZs in the same region
- Up to 100s of EC2 instances
- The instances in a partition do not share racks with the instances in the other partitions
- A partition failure can affect many EC2 but won't affect other partitions
- EC2 instances get access to the partition information as metadata
- Use cases: HDFS, HBase, Cassandra, Kafka

# Shutdown Behavior

- **Shutdown Behavior:** How should the instance react when shutdown is done using the OS?
  - Stop (default)
  - Terminate
- This is not applicable when shutting down from AWS console.
- CLI Attribute: `InstanceInitiatedShutdownBehavior`





# Termination Protection

- **Enable termination protection:**  
To protect against accidental termination in AWS Console or CLI
- **Exam Tip:**
  - We have an instance where shutdown behavior = terminate and enable terminate protection is ticked
  - We shutdown the instance from the OS, what will happen ?
  - The instance will still be terminated!

# EC2 Launch Troubleshooting

- **# InstanceLimitExceeded:** if you get this error, it means that you have reached your limit of max number of **vCPUs** per region
- On-Demand instance limits are set on a per-region basis
- Example: If you run on-demand (A, C, D, H, I, M, R, T, Z) instance types you'll have 64 vCPUs (default)
- Resolution: Either launch the instance in a different region or request AWS to increase your limit of the region
- NOTE: vCPU-based limits only apply to running On-Demand instances and Spot instances

# EC2 Launch Troubleshooting

- **# InsufficientInstanceCapacity** : if you get this error, it means AWS does not have that enough On-Demand capacity in the particular AZ where the instance is launched.
- Resolution :
  - Wait for few mins before requesting again.
  - If requesting more than 1 requests, break down the requests. If you need 5 instances, rather than a single request of 5, request one by one.
  - If urgent, **submit a request for a different instance type now**, which can be resized later.
  - Also, can request the EC2 instance in a different AZ

# EC2 Launch Troubleshooting

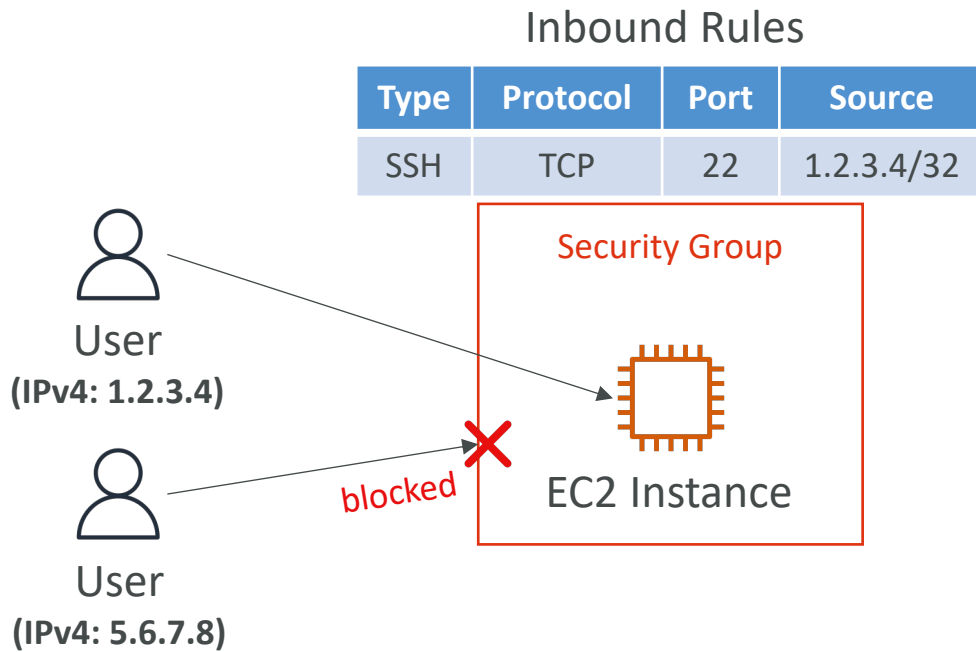
- **# Instance Terminates Immediately** (goes from pending to terminated)
  - You've reached your EBS volume limit.
  - An EBS snapshot is corrupt.
  - The root EBS volume is encrypted and you do not have permissions to access the KMS key for decryption.
  - The instance store-backed AMI that you used to launch the instance is missing a required part (an image.part.xx file).
- **To find the exact reason**, check out the EC2 console of AWS - instances - Description tab, note the reason next to the State transition reason label.

# EC2 SSH troubleshooting

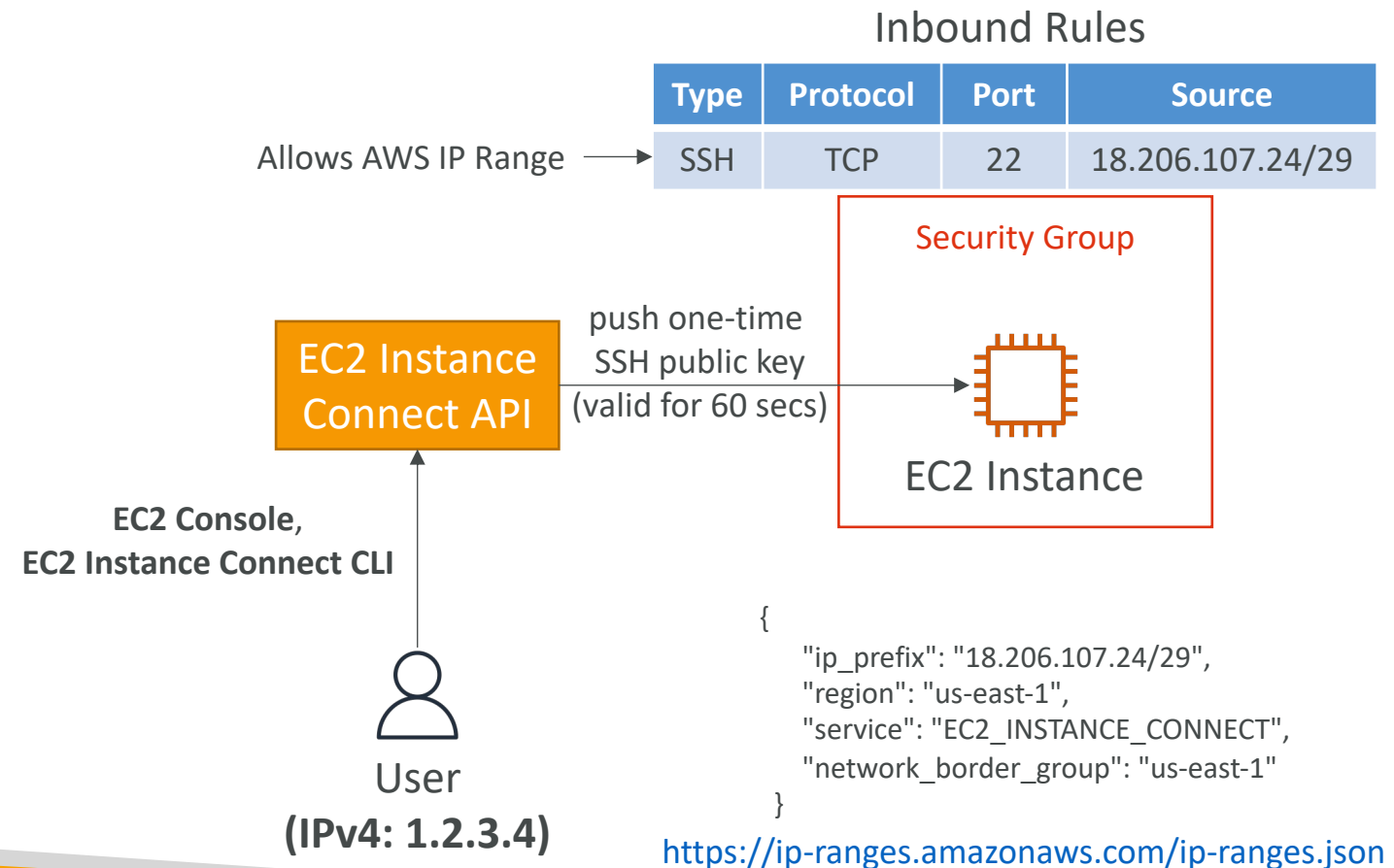
- Make sure the private key (pem file) on your linux machine has 400 permissions, else you will get **“Unprotected private key file”** error
- Make sure the username for the OS is given correctly when logging via SSH, else you will get **“Host key not found”, “Permission denied”, or “Connection closed by [instance] port 22”** error
- Possible reasons for **“Connection timed out”** to EC2 instance via SSH:
  - SG is not configured correctly
  - NACL is not configured correctly
  - Check the route table for the subnet (routes traffic destined outside VPC to IGW)
  - Instance doesn't have a public IPv4
  - CPU load of the instance is high

# SSH vs. EC2 Instance Connect

## Connect using SSH



## Connect using EC2 Instance Connect



# EC2 Instances Purchasing Options

- **On-Demand Instances** – short workload, predictable pricing, pay by second
- **Reserved (1 & 3 years)**
  - **Reserved Instances** – long workloads
  - **Convertible Reserved Instances** – long workloads with flexible instances
- **Savings Plans (1 & 3 years)** – commitment to an amount of usage, long workload
- **Spot Instances** – short workloads, cheap, can lose instances (less reliable)
- **Dedicated Hosts** – book an entire physical server, control instance placement
- **Dedicated Instances** – no other customers will share your hardware
- **Capacity Reservations** – reserve capacity in a specific AZ for any duration

# EC2 On Demand

- Pay for what you use:
  - Linux or Windows - billing per second, after the first minute
  - All other operating systems - billing per hour
- Has the highest cost but 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



# EC2 Reserved Instances

- Up to **72%** discount compared to On-demand
- You reserve a specific instance attributes (**Instance Type, Region, Tenancy, OS**)
- **Reservation Period** – 1 year (+discount) or 3 years (+++discount)
- **Payment Options** – No Upfront (+), Partial Upfront (++), All Upfront (+++)
- **Reserved Instance's Scope** – Regional or Zonal (reserve capacity in an AZ)
- Recommended for steady-state usage applications (think database)
- You can buy and sell in the Reserved Instance Marketplace
  
- **Convertible Reserved Instance**
  - Can change the EC2 instance type, instance family, OS, scope and tenancy
  - Up to **66%** discount

**Note:** the % discounts are different from the video as AWS change them over time – the exact numbers are not needed for the exam. This is just for illustrative purposes 😊

# EC2 Savings Plans

- Get a discount based on long-term usage (up to 72% - same as RIs)
- Commit to a certain type of usage (\$10/hour for 1 or 3 years)
- Usage beyond EC2 Savings Plans is billed at the On-Demand price
  
- Locked to a specific instance family & AWS region (e.g., M5 in us-east-1)
- Flexible across:
  - Instance Size (e.g., m5.xlarge, m5.2xlarge)
  - OS (e.g., Linux, Windows)
  - Tenancy (Host, Dedicated, Default)

# EC2 Spot Instances



- Can get a **discount of up to 90%** compared to On-demand
- Instances that you can “lose” at any point of time if your max price is less than the current spot price
- The **MOST cost-efficient** instances in AWS
- Useful for workloads that are resilient to failure
  - Batch jobs
  - Data analysis
  - Image processing
  - Any **distributed** workloads
  - Workloads with a flexible start and end time
- Not suitable for critical jobs or databases

# EC2 Dedicated Hosts

- A physical server with EC2 instance capacity fully dedicated to your use
- Allows you address **compliance requirements** and **use your existing server-bound software licenses** (per-socket, per-core, per-VM software licenses)
- Purchasing Options:
  - **On-demand** – pay per second for active Dedicated Host
  - **Reserved** - 1 or 3 years (No Upfront, Partial Upfront, All Upfront)
- The most expensive option
- Useful for software that have complicated licensing model (BYOL – Bring Your Own License)
- Or for companies that have strong regulatory or compliance needs

# EC2 Dedicated Instances

- Instances run on hardware that's dedicated to you
- May share hardware with other instances in same account
- No control over instance placement (can move hardware after Stop / Start)

Characteristic	Dedicated Instances	Dedicated Hosts
Enables the use of dedicated physical servers	X	X
Per instance billing (subject to a \$2 per region fee)	X	
Per host billing		X
Visibility of sockets, cores, host ID		X
Affinity between a host and instance		X
Targeted instance placement		X
Automatic instance placement	X	X
Add capacity using an allocation request		X

# EC2 Capacity Reservations

- Reserve **On-Demand** instances capacity in a specific AZ for any duration
- You always have access to EC2 capacity when you need it
- **No time commitment** (create/cancel anytime), **no billing discounts**
- Combine with Regional Reserved Instances and Savings Plans to benefit from billing discounts
- You're charged at On-Demand rate whether you run instances or not
- Suitable for short-term, uninterrupted workloads that needs to be in a 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

# Price Comparison

## Example – m4.large – us-east-1

Price Type	Price (per hour)
On-Demand	\$0.10
Spot Instance (Spot Price)	\$0.038 - \$0.039 (up to 61% off)
Reserved Instance (1 year)	\$0.062 (No Upfront) - \$0.058 (All Upfront)
Reserved Instance (3 years)	\$0.043 (No Upfront) - \$0.037 (All Upfront)
EC2 Savings Plan (1 year)	\$0.062 (No Upfront) - \$0.058 (All Upfront)
Reserved <b>Convertible</b> Instance (1 year)	\$0.071 (No Upfront) - \$0.066 (All Upfront)
Dedicated Host	On-Demand Price
Dedicated Host Reservation	Up to 70% off
Capacity Reservations	On-Demand Price

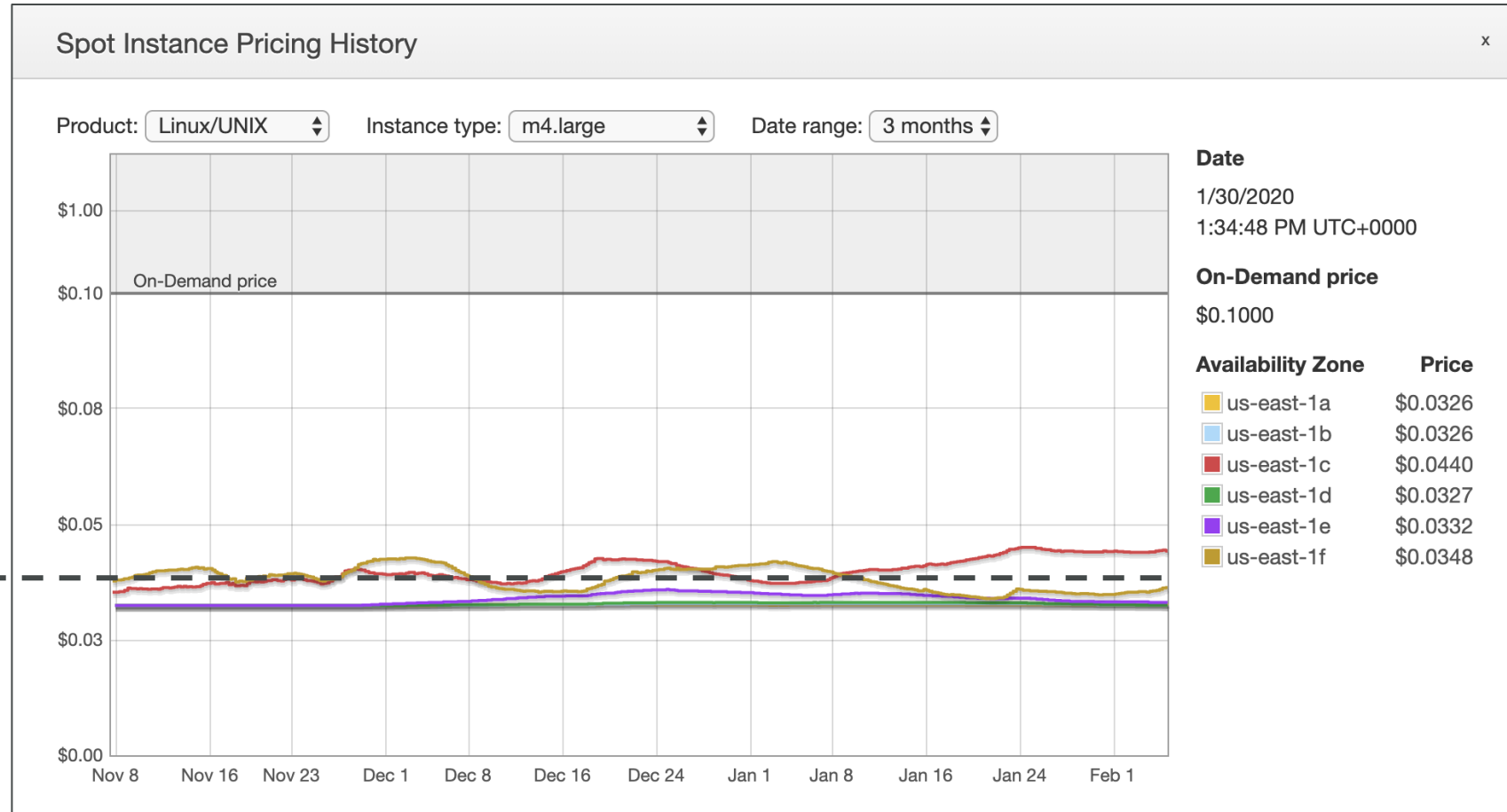




# EC2 Spot Instance Requests

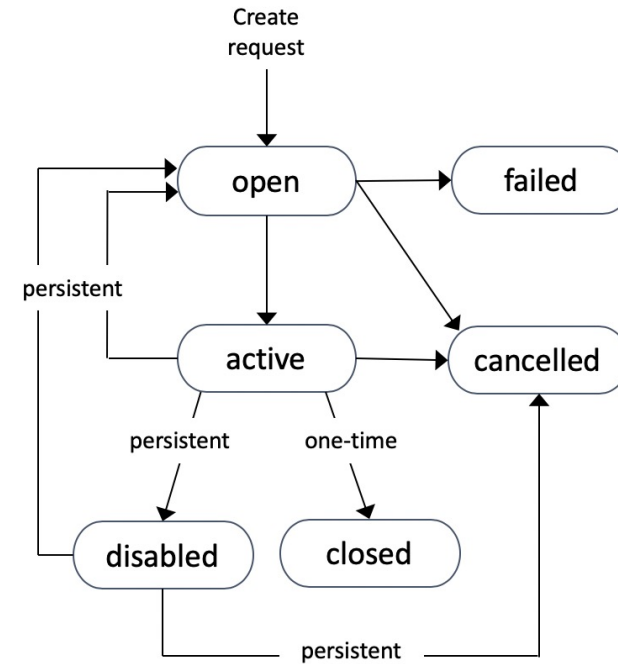
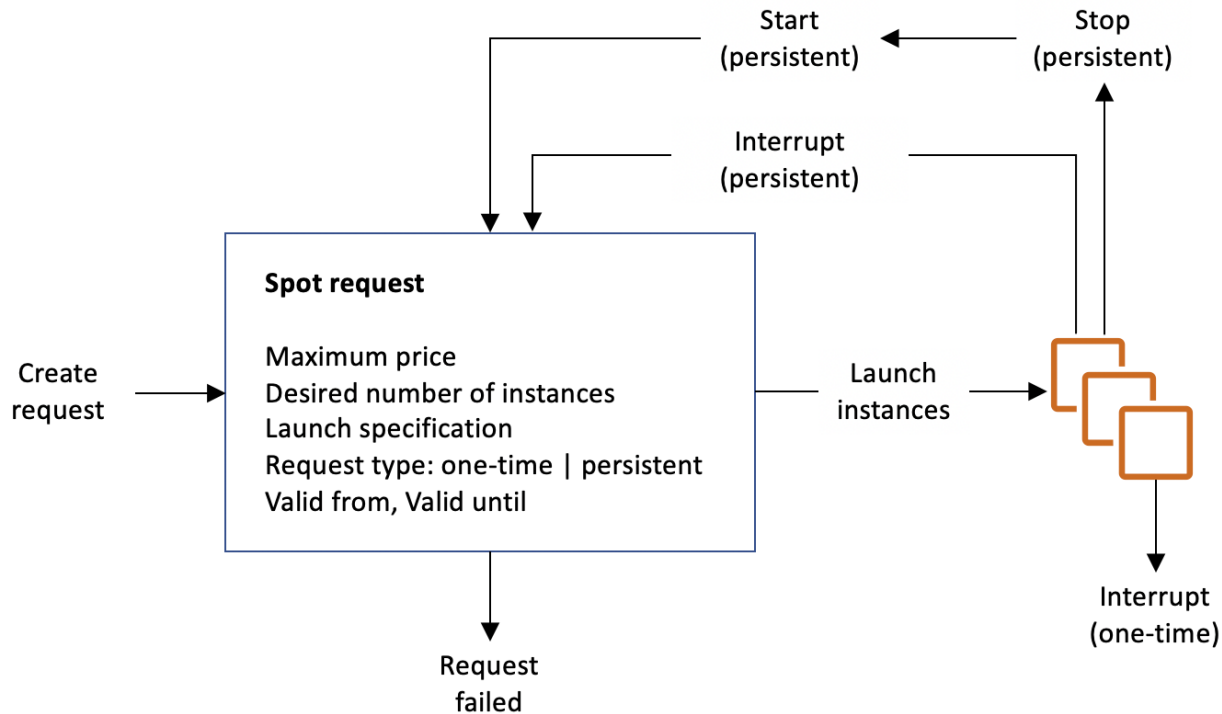
- Can get a discount of up to 90% compared to On-demand
- Define **max spot price** and get the instance while **current spot price < max**
  - The hourly spot price varies based on offer and capacity
  - If the current spot price > your max price you can choose to **stop** or **terminate** your instance with a 2 minutes grace period.
- Other strategy: **Spot Block**
  - “block” spot instance during a specified time frame (1 to 6 hours) without interruptions
  - In rare situations, the instance may be reclaimed
- Used for batch jobs, data analysis, or workloads that are resilient to failures.
- Not great for critical jobs or databases

# EC2 Spot Instances Pricing



<https://console.aws.amazon.com/ec2sp/v1/spot/home?region=us-east-1#>

# How to terminate Spot Instances?



You can only cancel Spot Instance requests that are **open, active, or disabled**.  
Cancelling a Spot Request does not terminate instances  
You must first cancel a Spot Request, and then terminate the associated Spot Instances

# Spot Fleets

- Spot Fleets = set of Spot Instances + (optional) On-Demand Instances
- The Spot Fleet will try to meet the target capacity with price constraints
  - Define possible launch pools: instance type (m5.large), OS, Availability Zone
  - Can have multiple launch pools, so that the fleet can choose
  - Spot Fleet stops launching instances when reaching capacity or max cost
- Strategies to allocate Spot Instances:
  - **lowestPrice:** from the pool with the lowest price (cost optimization, short workload)
  - **diversified:** distributed across all pools (great for availability, long workloads)
  - **capacityOptimized:** pool with the optimal capacity for the number of instances
- Spot Fleets allow us to automatically request Spot Instances with the lowest price

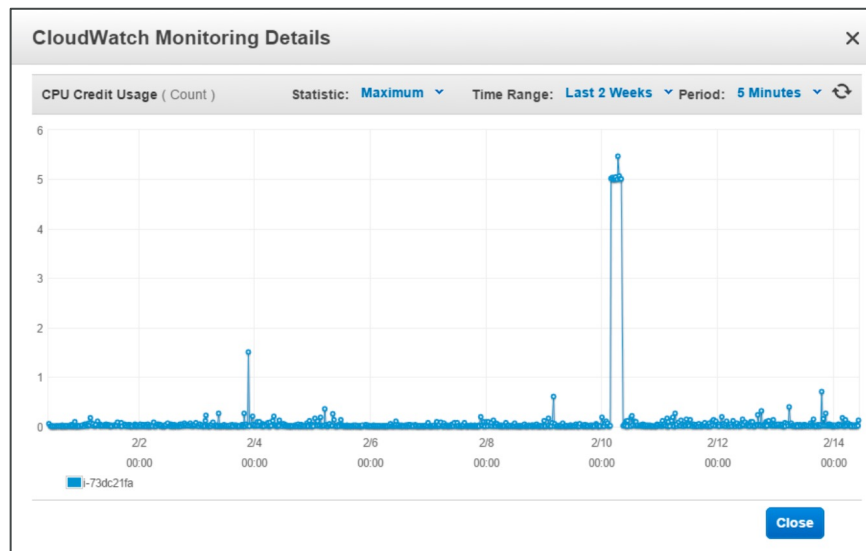
# Burstable Instances (T2/T3)

- AWS has the concept of burstable instances (T2/T3 machines)
- Burst means that overall, the instance has OK CPU performance.
- When the machine needs to process something unexpected (a spike in load for example), it can burst, and CPU can be VERY good.
- If the machine bursts, it utilizes “burst credits”
- If all the credits are gone, the CPU becomes BAD
- If the machine stops bursting, credits are accumulated over time

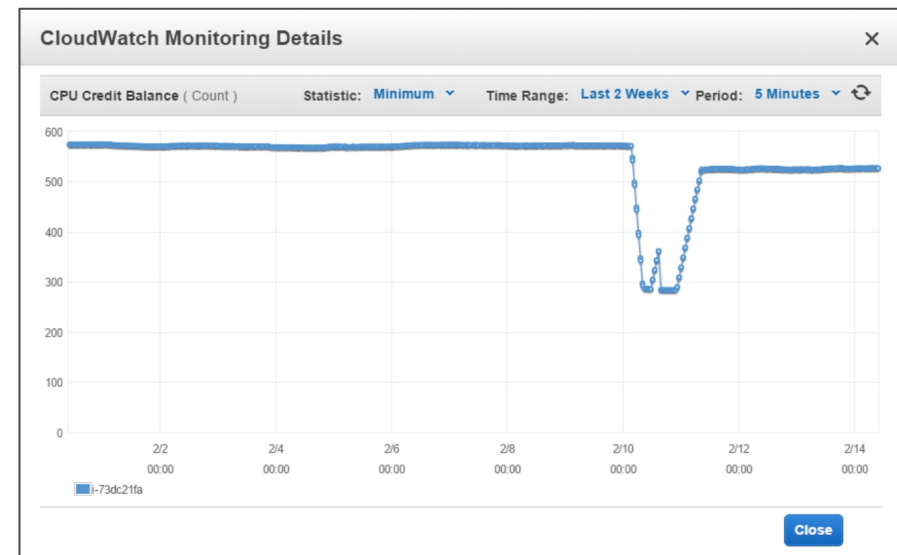
# Burstable Instances (T2/T3)

- Burstable instances can be amazing to handle unexpected traffic and getting the insurance that it will be handled correctly
- If your instance consistently runs low on credit, you need to move to a different kind of non-burstable instance

Credit usage



Credit balance

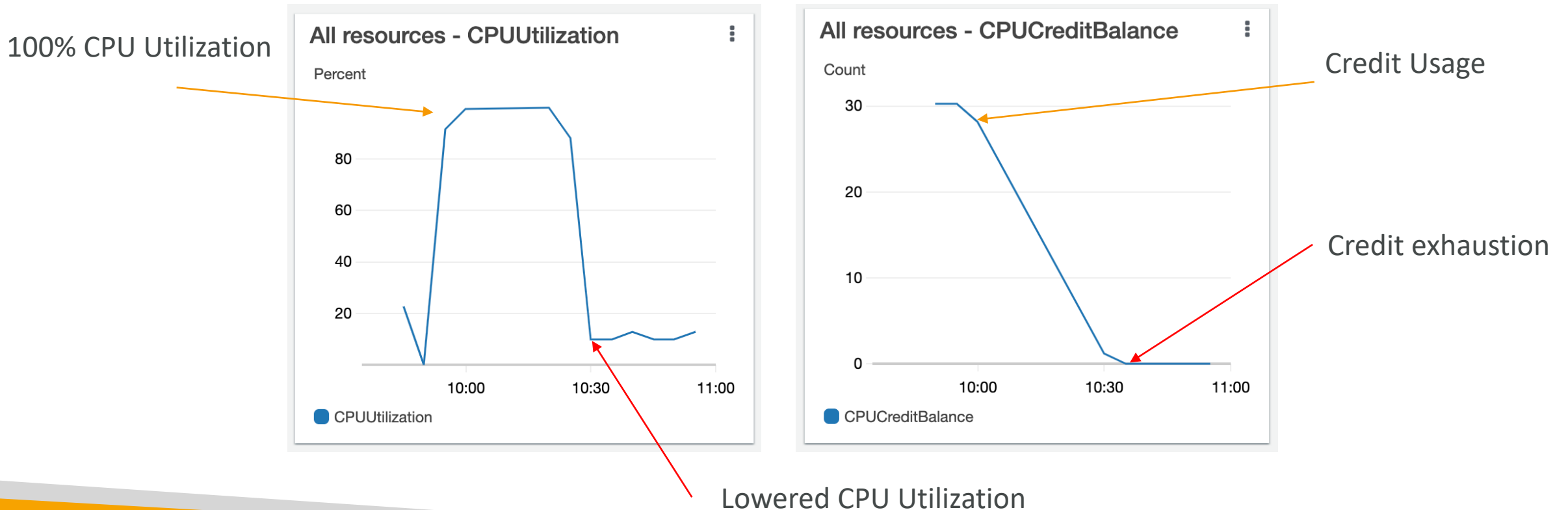


# CPU Credits

Instance type	Launch credits	vCPUs	CPU credits earned per hour	Maximum earned CPU credit balance	vCPUs	Baseline performance (% CPU utilization)
t2.nano	30	1	3	72	1	5%
t2.micro	30	1	6	144	1	10%
t2.small	30	1	12	288	1	20%
t2.medium	60	2	24	576	2	40% (of 200% max)*
t2.large	60	2	36	864	2	60% (of 200% max)*
t2.xlarge	120	4	54	1296	4	90% (of 400% max)*
t2.2xlarge	240	8	81	1944	8	135% (of 800% max)*

# What happens when credit are exhausted?

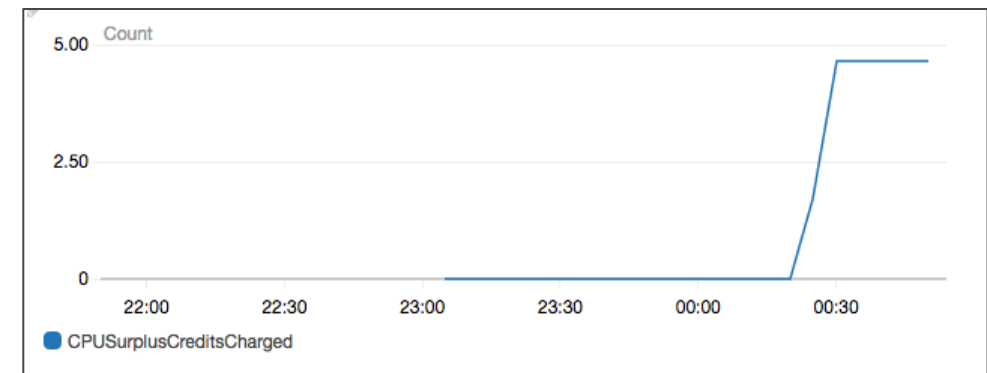
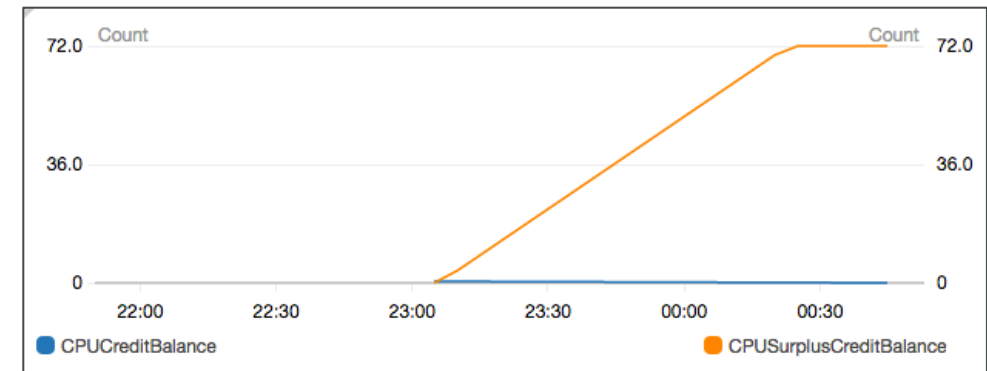
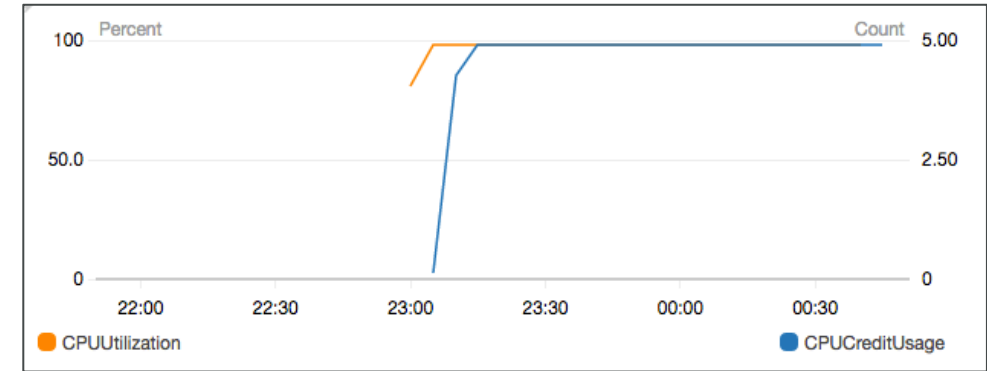
- Experiment: run a CPU stress command (to peak at 100%)
- After the credits are exhausted, the measured CPU utilization drops





# T2/T3 Unlimited

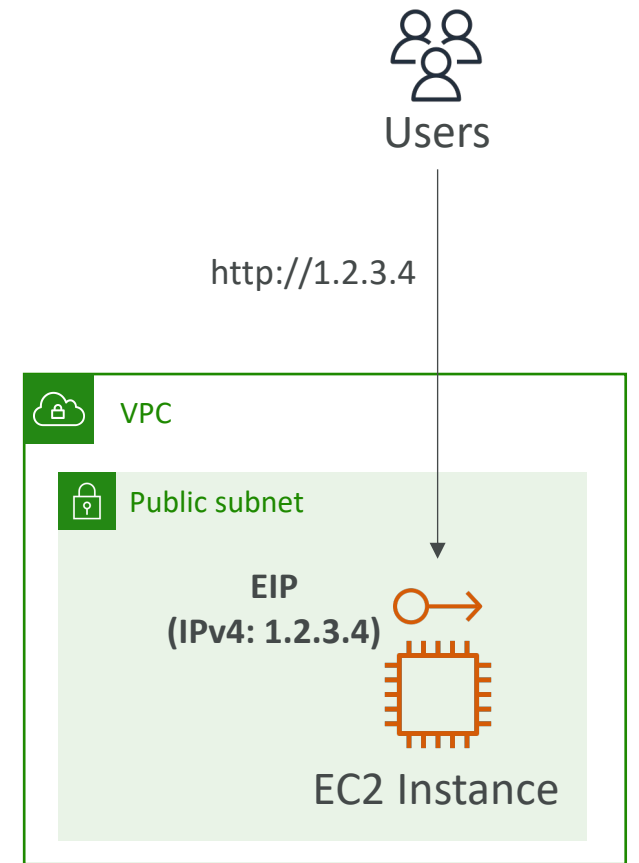
- It is possible to have an “unlimited burst credit balance”
- You pay extra money if you go over your credit balance, but you don't lose in performance
- If average CPU usage over a 24-hour period exceeds the baseline, the instance is billed for additional usage per vCPU/hour
- Be careful, costs could go high if you're not monitoring the CPU health of your instances



<https://aws.amazon.com/blogs/aws/new-t2-unlimited-going-beyond-the-burst-with-high-performance/>

# Elastic IPs

- When you stop and then start an EC2 instance, it changes its public IP
- If you need to have a fixed public IP, you need an Elastic IP
- An Elastic IP is a public IPv4 you own as long as you don't delete it
- You can attach it to one instance at a time
- You can remap it across instances
- You don't pay for the Elastic IP if it's attached to a server
- You pay for the Elastic IP if it's not attached to a server



# Elastic IPs

- With an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.
- You can only have 5 Elastic IP in your account (you can ask AWS to increase that).
- How you can avoid using Elastic IP:
  - Always think if other alternatives are available to you
  - You could use a random public IP and register a DNS name to it
  - Or use a Load Balancer with a static hostname

# CloudWatch Metrics for EC2

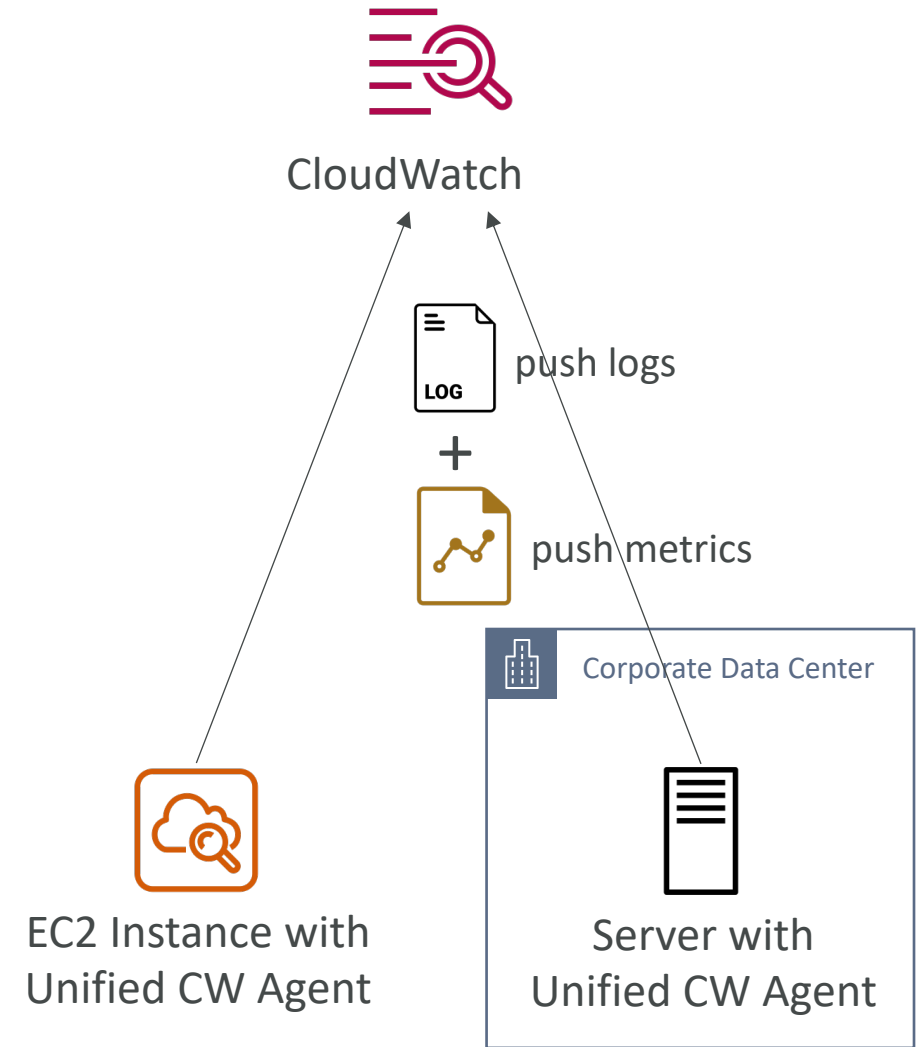
- **AWS Provided metrics (AWS pushes them):**
  - Basic Monitoring (default): metrics are collected at a 5 minute interval
  - Detailed Monitoring (paid): metrics are collected at a 1 minute interval
  - Includes CPU, Network, Disk and Status Check Metrics
- **Custom metric (yours to push):**
  - Basic Resolution: 1 minute resolution
  - High Resolution: all the way to 1 second resolution
  - Include RAM, application level metrics
  - Make sure the IAM permissions on the EC2 instance role are correct !

# EC2 included metrics

- CPU: CPU Utilization + Credit Usage / Balance
- Network: Network In / Out
- Status Check:
  - Instance status = check the EC2 VM
  - System status = check the underlying hardware
- Disk: Read / Write for Ops / Bytes (only for instance store)
  
- RAM is NOT included in the AWS EC2 metrics

# Unified CloudWatch Agent

- For virtual servers (EC2 instances, on-premises servers, ...)
- Collect additional system-level metrics such as RAM, processes, used disk space, etc.
- Collect logs to send to CloudWatch Logs
  - No logs from inside your EC2 instance will be sent to CloudWatch Logs without using an agent
- Centralized configuration using SSM Parameter Store
- Make sure IAM permissions are correct
- Default namespace for metrics collected by the Unified CloudWatch agent is **CWAgent** (can be configured/changed)

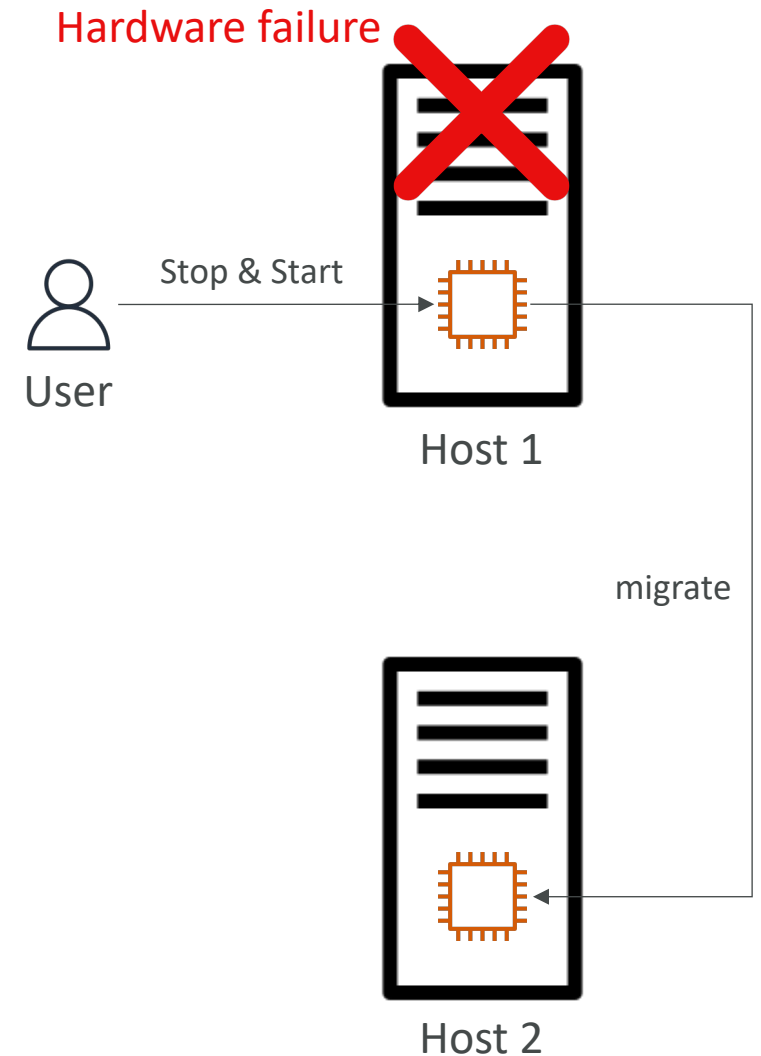


# Unified CloudWatch Agent – procstat Plugin

- Collect metrics and monitor system utilization of individual processes
- Supports both Linux and Windows servers
- Example: amount of time the process uses CPU, amount of memory the process uses, ...
- Select which processes to monitor by
  - **pid\_file**: name of process identification number (PID) files they create
  - **exe**: process name that match string you specify (RegEx)
  - **pattern**: command lines used to start the processes (RegEx)
- Metrics collected by procstat plugin begins with “**procstat**” prefix (e.g., `procstat_cpu_time`, `procstat_cpu_usage`, ...)

# Status Checks

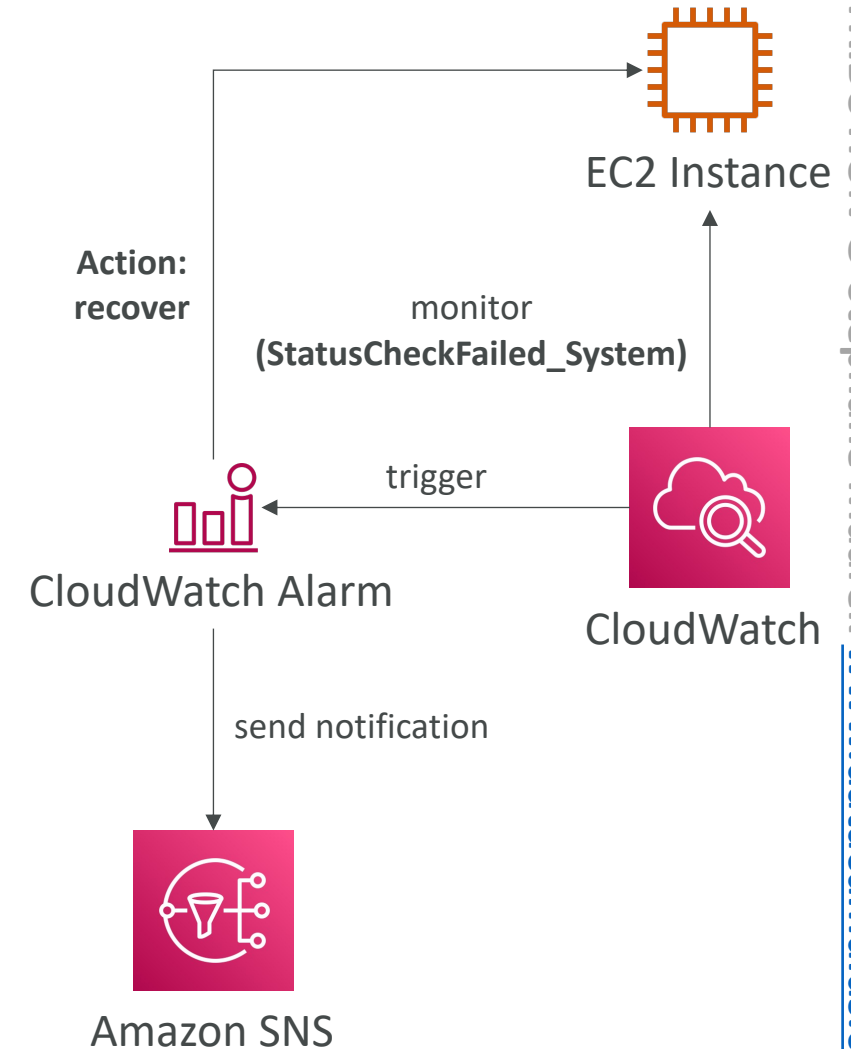
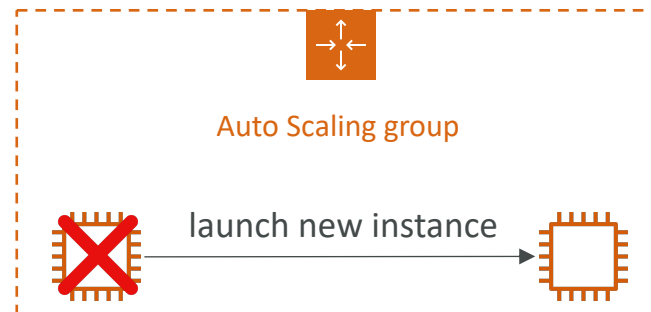
- Automated checks to identify hardware and software issues
- **System Status Checks**
  - Monitors problems with AWS systems (software/hardware issues on the physical host, loss of system power, ...)
  - Check **Personal Health Dashboard** for any scheduled critical maintenance by AWS to your instance's host
  - Resolution: stop and start the instance (instance migrated to a new host)
- **Instance Status Checks**
  - Monitors software/network configuration of your instance (invalid network configuration, exhausted memory, ...)
  - Resolution: reboot the instance or change instance configuration





# Status Checks - CW Metrics & Recovery

- CloudWatch Metrics (1 minute interval)
  - StatusCheckFailed\_System
  - StatusCheckFailed\_Instance
  - StatusCheckFailed (for both)
- Option 1: **CloudWatch Alarm**
  - Recover EC2 instance with the same private/public IP, EIP, metadata, and Placement Group
  - Send notifications using SNS
- Option 2: **Auto Scaling Group**
  - Set min/max/desired 1 to recover an instance but won't keep the same private and elastic IP

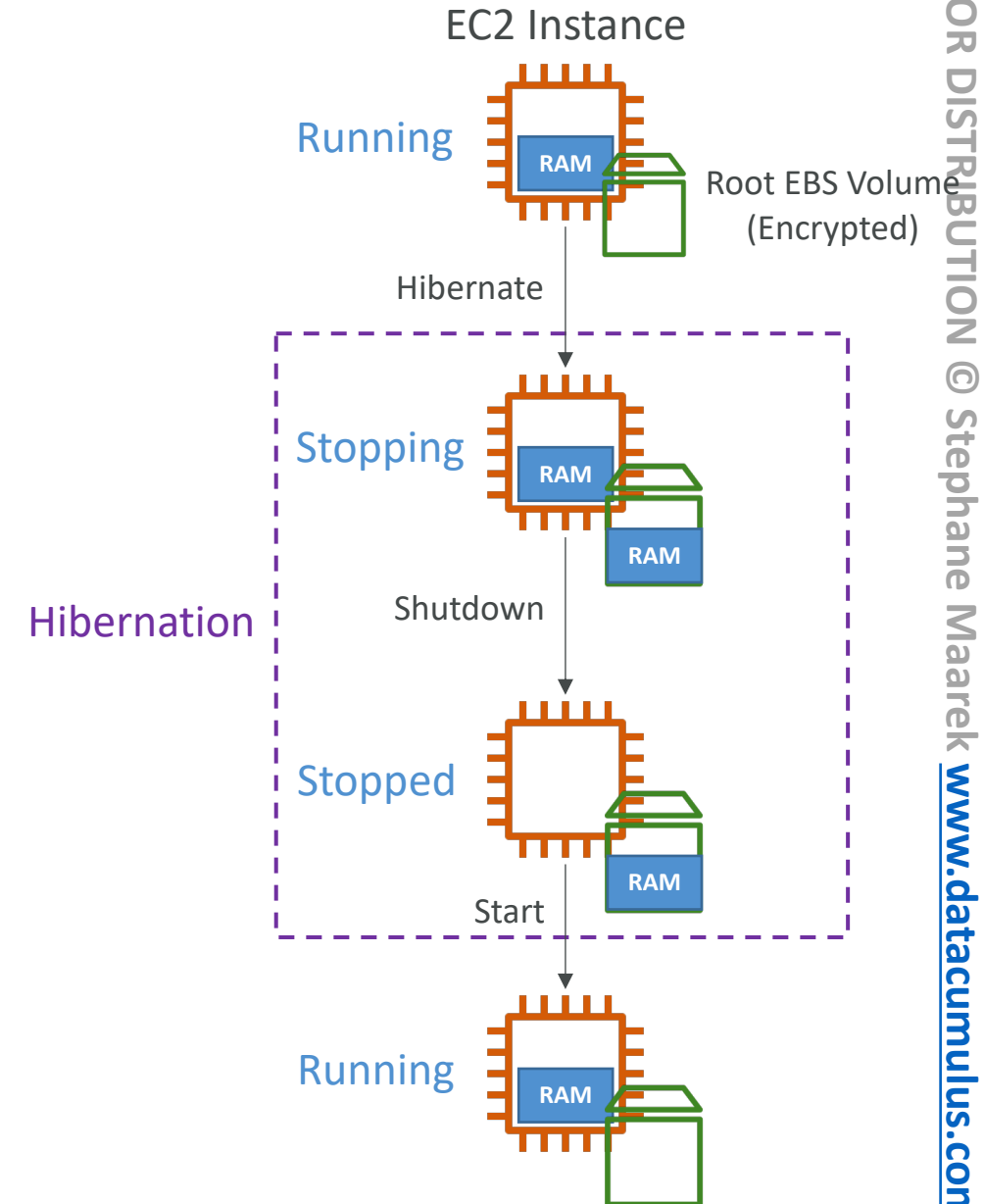


# EC2 Hibernate

- We know we can stop, terminate instances
  - **Stop** – s the data on disk (EBS) is kept intact in the next start
  - **Terminate** – any EBS volumes (root) also set-up to be destroyed is lost
- On start, the following happens:
  - First start: the OS boots & the EC2 User Data script is run
  - Following starts: the OS boots up
  - Then your application starts, caches get warmed up, and that can take time!

# EC2 Hibernate

- Introducing **EC2 Hibernate**:
  - The in-memory (RAM) state is preserved
  - The instance boot is much faster!  
(the OS is not stopped / restarted)
  - Under the hood: the RAM state is written to a file in the root EBS volume
  - The root EBS volume must be encrypted
- **Use cases:**
  - Long-running processing
  - Saving the RAM state
  - Services that take time to initialize



# EC2 Hibernate – Good to know

- **Supported Instance Families** – C3, C4, C5, I3, M3, M4, R3, R4, T2, T3, ...
- **Instance RAM Size** – must be less than 150 GB.
- **Instance Size** – not supported for bare metal instances.
- **AMI** – Amazon Linux 2, Linux AMI, Ubuntu, RHEL, CentOS & Windows...
- **Root Volume** – must be EBS, encrypted, not instance store, and large
- Available for **On-Demand**, **Reserved** and **Spot** Instances
  
- An instance can **NOT** be hibernated more than 60 days

# AMI Section

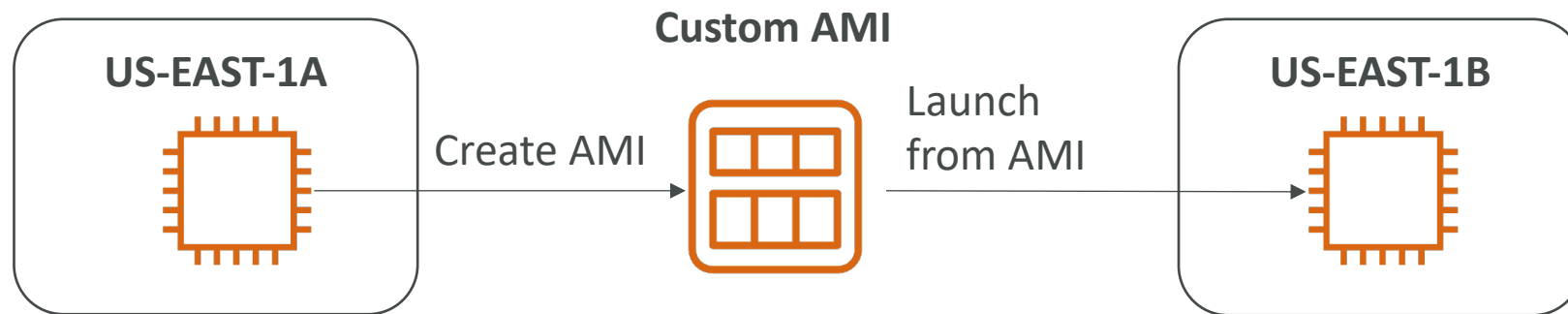


# AMI Overview

- AMI = Amazon Machine Image
- AMI are a **customization** of an EC2 instance
  - You add your own software, configuration, operating system, monitoring...
  - Faster boot / configuration time because all your software is pre-packaged
- AMI are built for a **specific region** (and can be copied across regions)
- You can launch EC2 instances from:
  - A **Public AMI**: AWS provided
  - **Your own AMI**: you make and maintain them yourself
  - An **AWS Marketplace AMI**: an AMI someone else made (and potentially sells)

# AMI Process (from an EC2 instance)

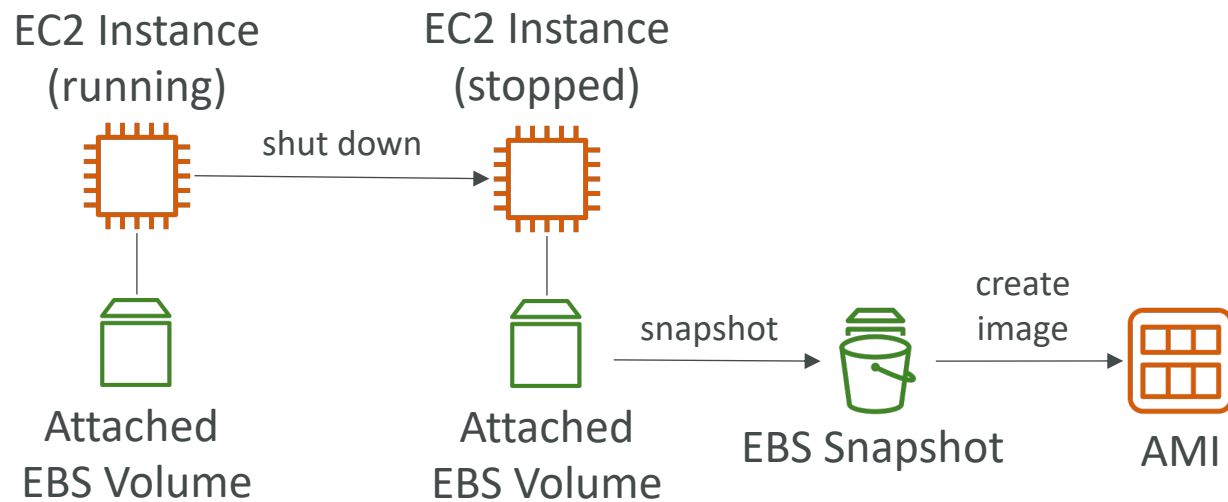
- Start an EC2 instance and customize it
- Stop the instance (for data integrity)
- Build an AMI – this will also create EBS snapshots
- Launch instances from other AMIs



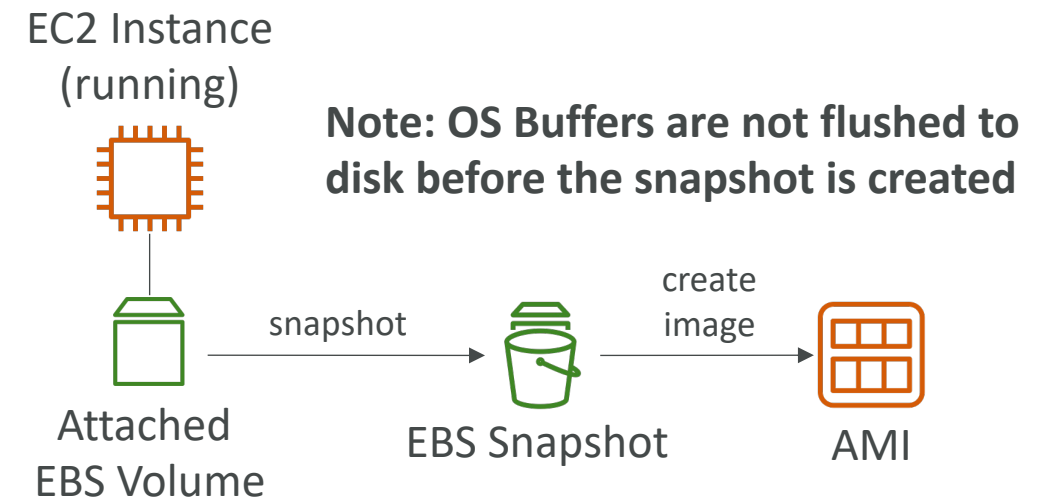
# AMI No-Reboot Option

- Enables you to create an AMI without shutting down your instance
- By default, it's not selected (AWS will shut down the instance before creating an AMI to maintain the file system integrity)

## With No-Reboot **Disabled** (default)



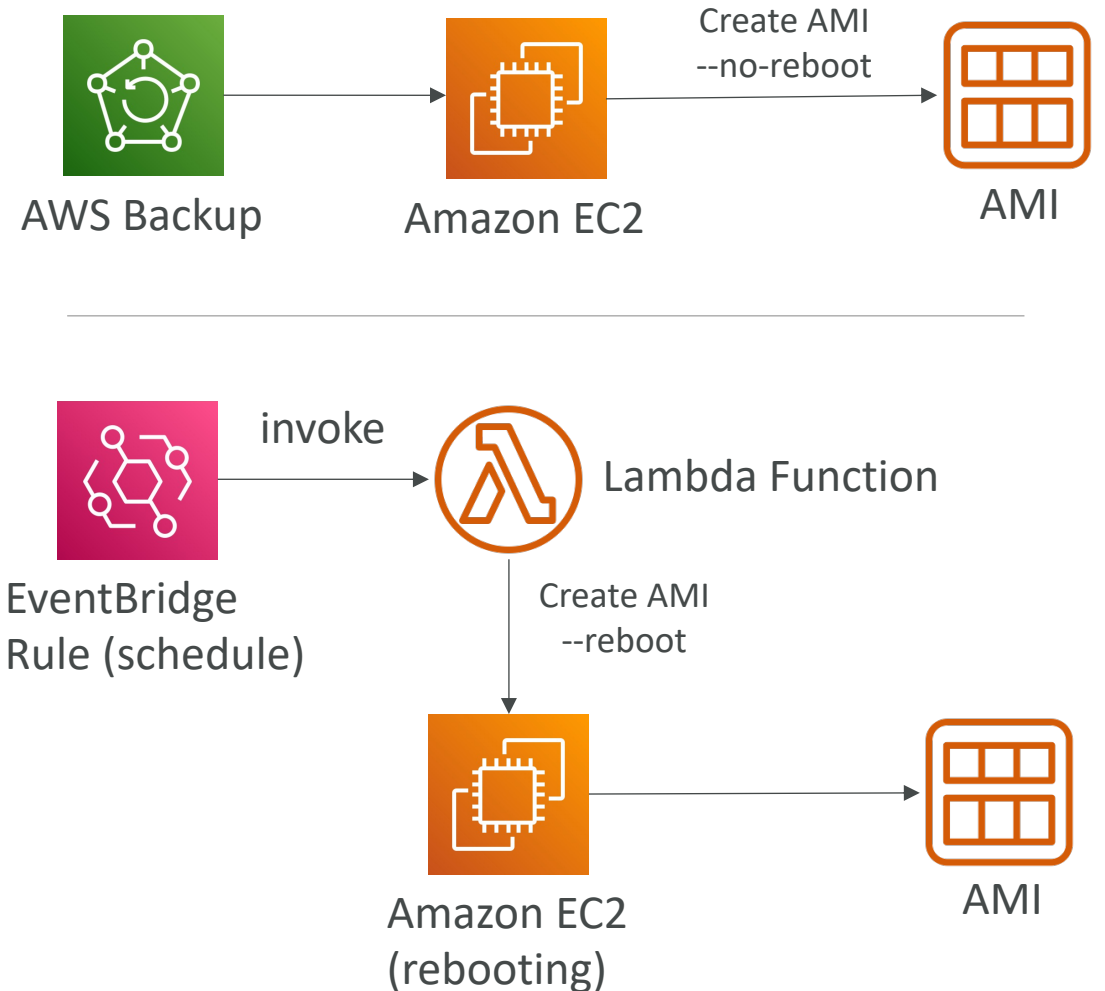
## With No-Reboot **Enabled**



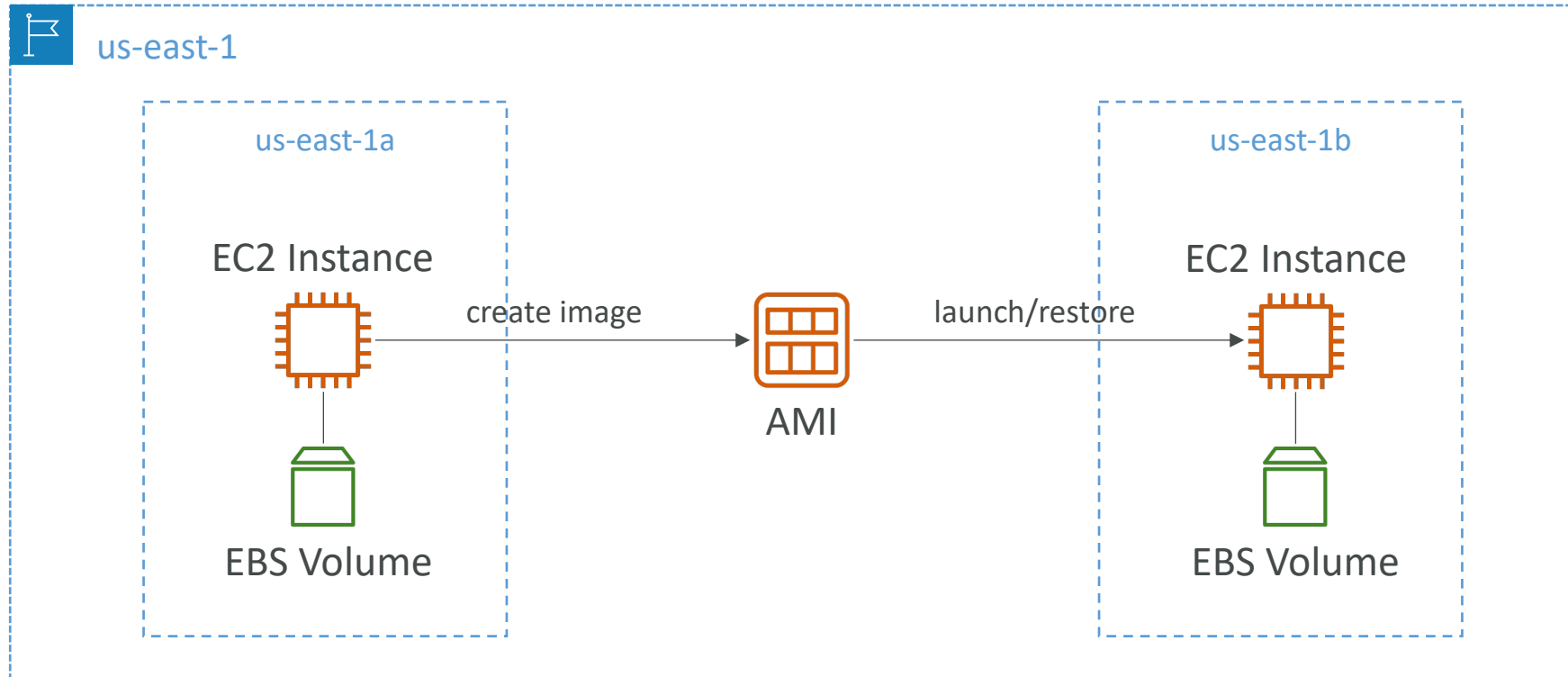


# AWS Backup Plans to create AMI

- AWS Backup doesn't reboot the instances while taking EBS snapshots (no-reboot behavior)
  - This won't help you to create an AMI that guarantees file system integrity since you need to reboot the instance
  - To maintain integrity you need to provide the reboot parameter while taking images (EventBridge + Lambda + CreateImage API with reboot)

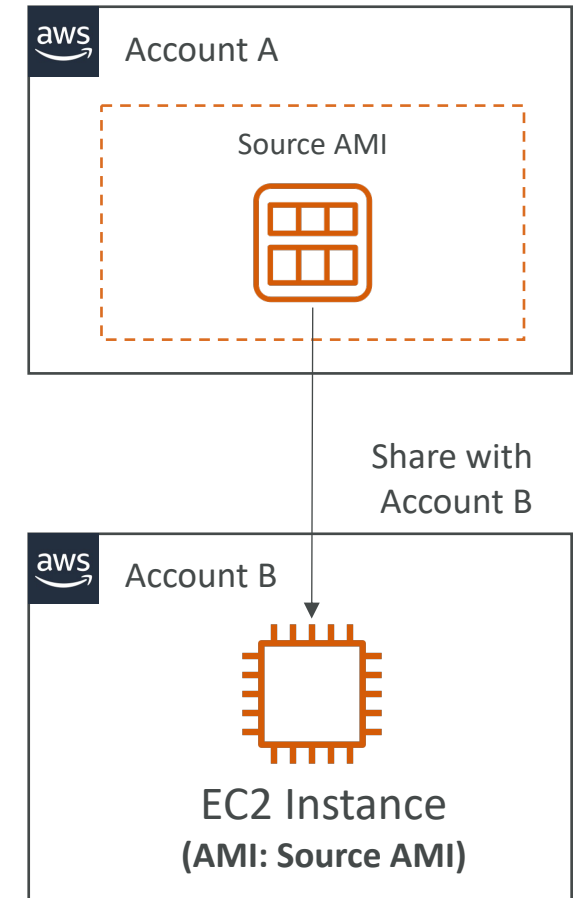


# EC2 Instance Migration between AZ

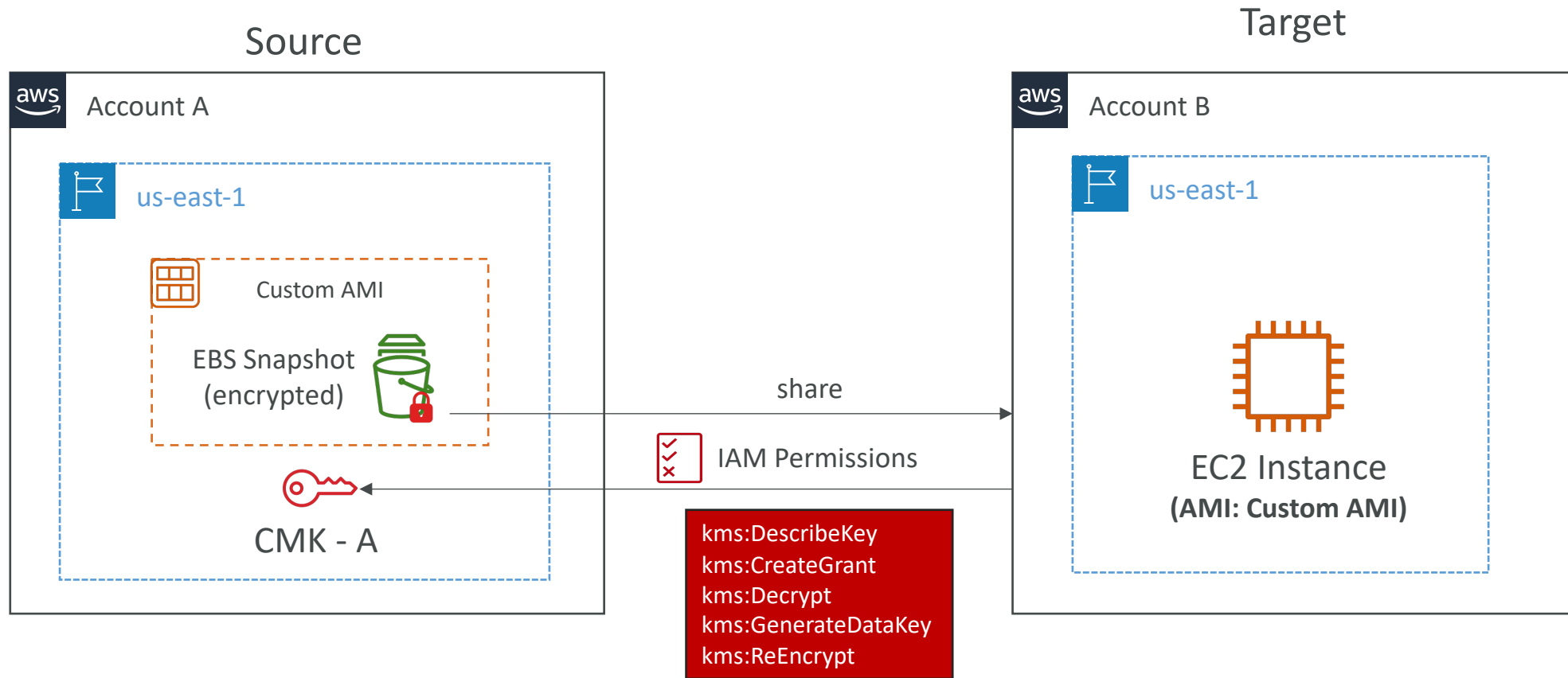


# Cross-Account AMI Sharing

- You can share an AMI with another AWS account
- Sharing an AMI does not affect the ownership of the AMI
- You can only share AMIs that have unencrypted volumes and volumes that are encrypted with a customer managed key
- If you share an AMI with encrypted volumes, you must also share any customer managed keys used to encrypt them.

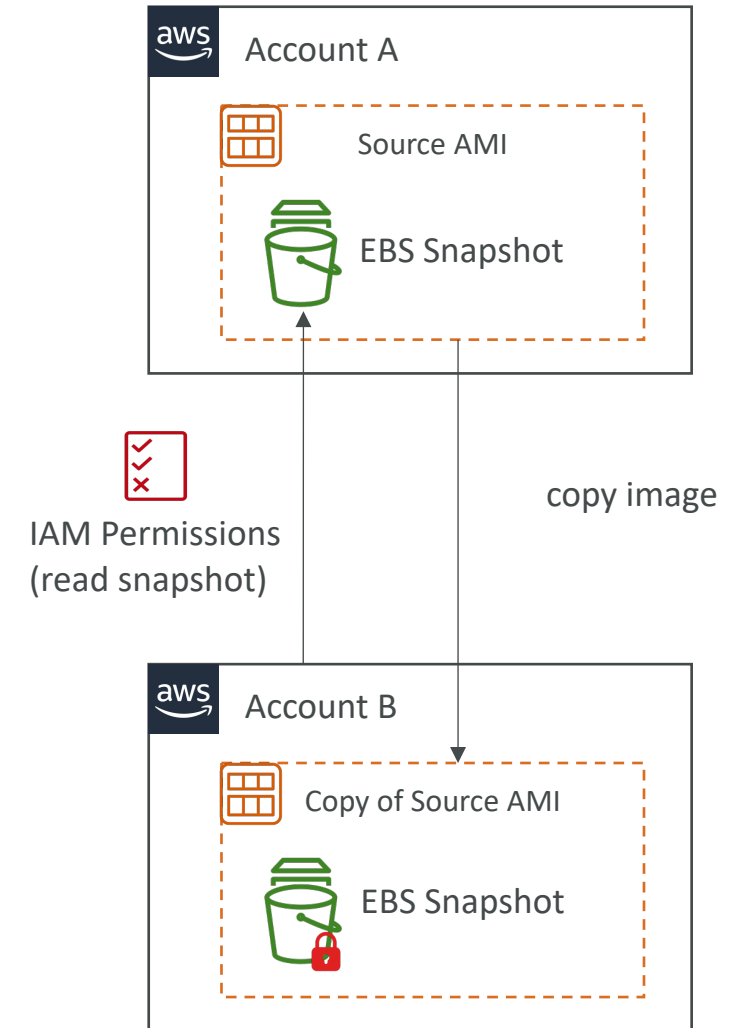


# AMI Sharing with KMS Encryption



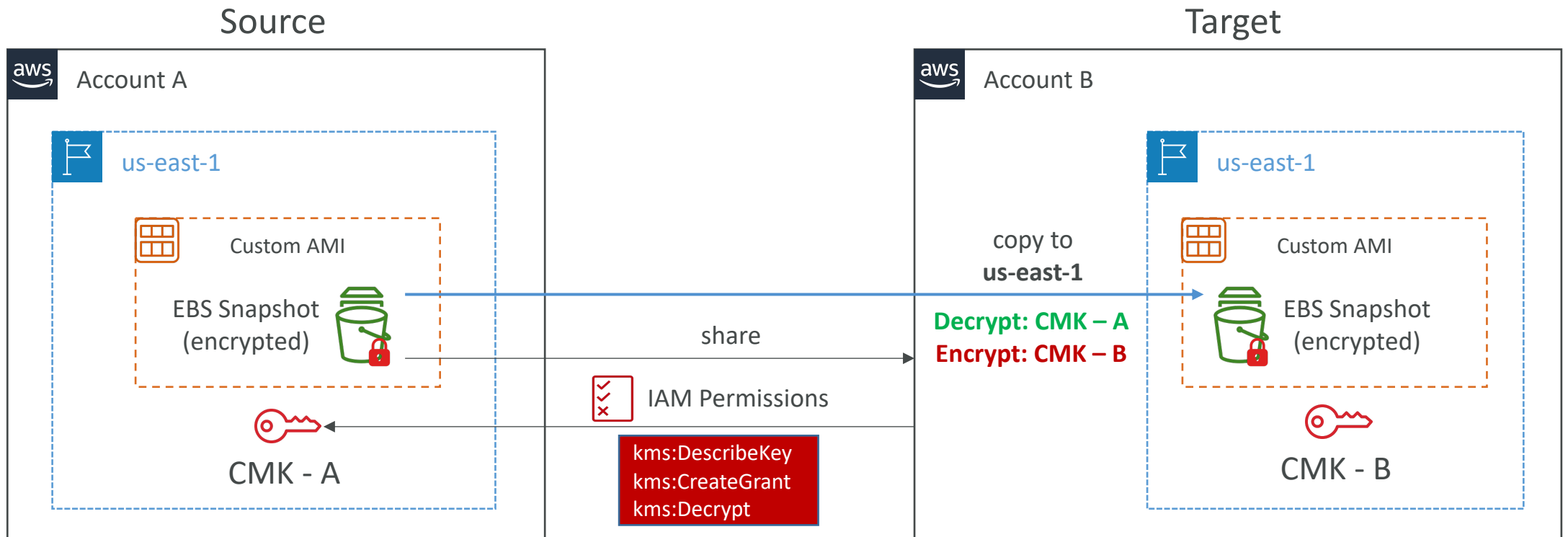
# Cross-Account AMI Copy

- If you copy an AMI that has been shared with your account, you are the owner of the target AMI in your account
- The owner of the source AMI must grant you read permissions for the storage that backs the AMI (EBS Snapshot)
- If the shared AMI has encrypted snapshots, the owner must share the key or keys with you as well
- Can encrypt the AMI with your own CMK while copying



# AMI Copy with KMS Encryption

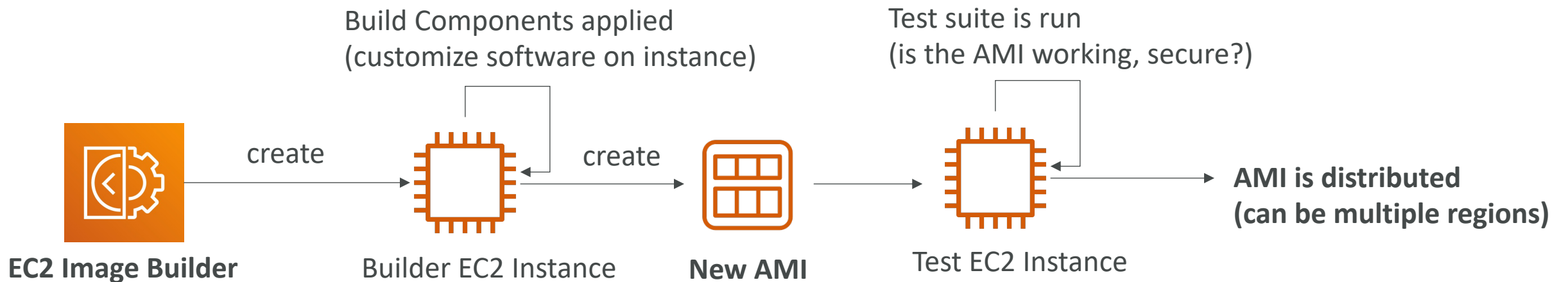
## Cross-Region / Cross-Account Encrypted AMI Copy



# EC2 Image Builder

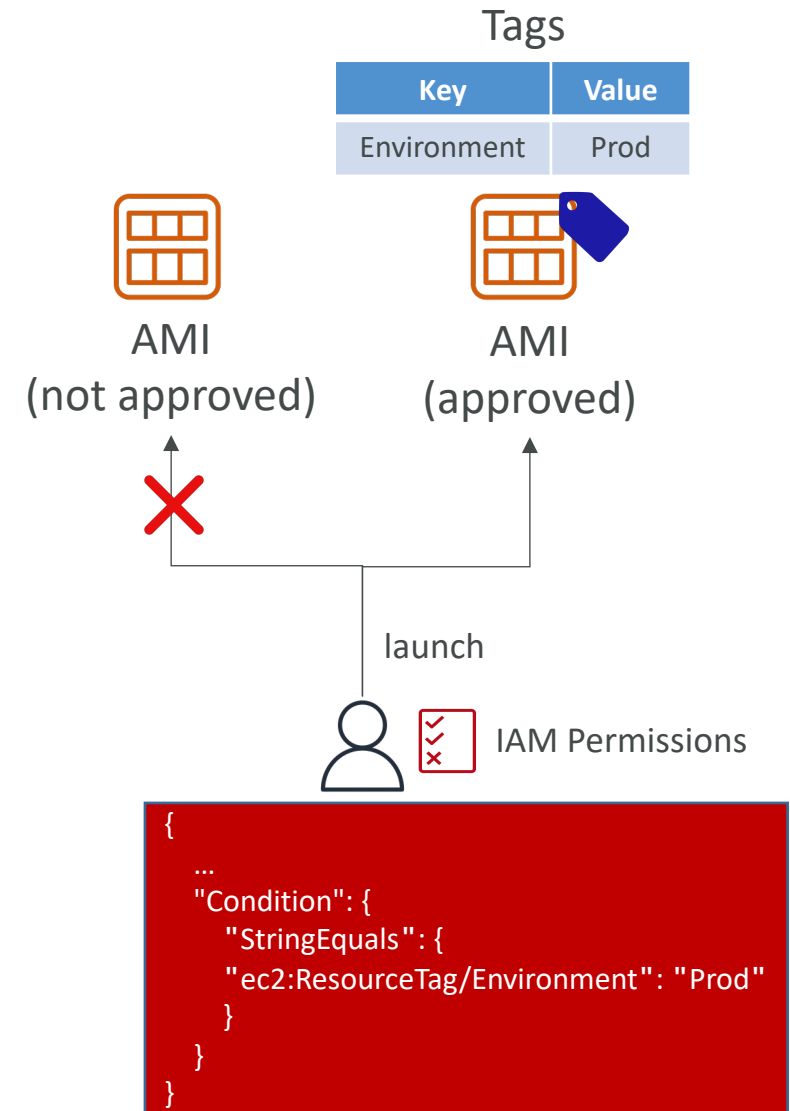
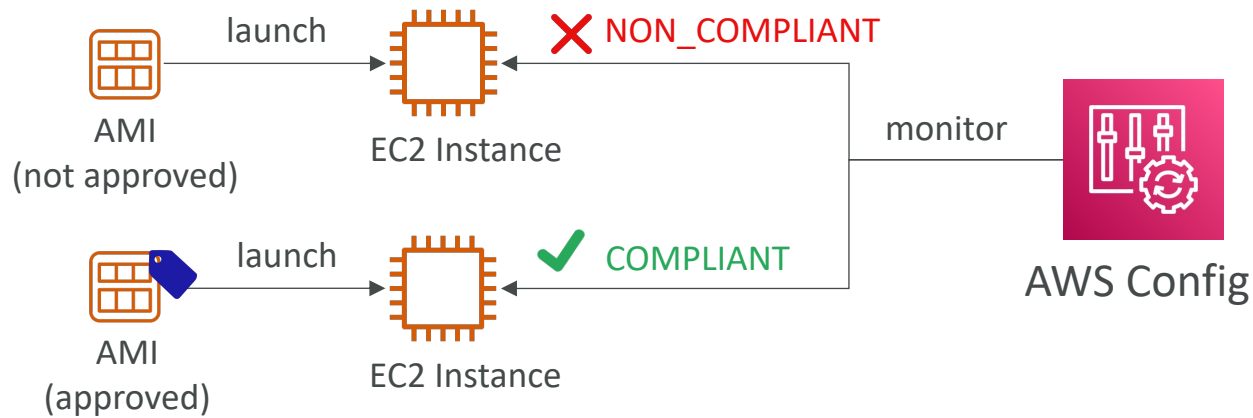


- Used to automate the creation of Virtual Machines or container images
- => Automate the creation, maintain, validate and test **EC2 AMIs**
- Can be run on a schedule (weekly, whenever packages are updated, etc...)
- Free service (only pay for the underlying resources)



# AMI in Production

- You can force users to only launch EC2 instances from pre-approved AMIs (AMIs tagged with specific tags) using IAM policies
- Combine with AWS Config to find non-compliant EC2 instance (instances launched with non-approved AMIs)





# Management of EC2 at scale

Systems Manager & OpsWorks

# AWS Systems Manager Overview



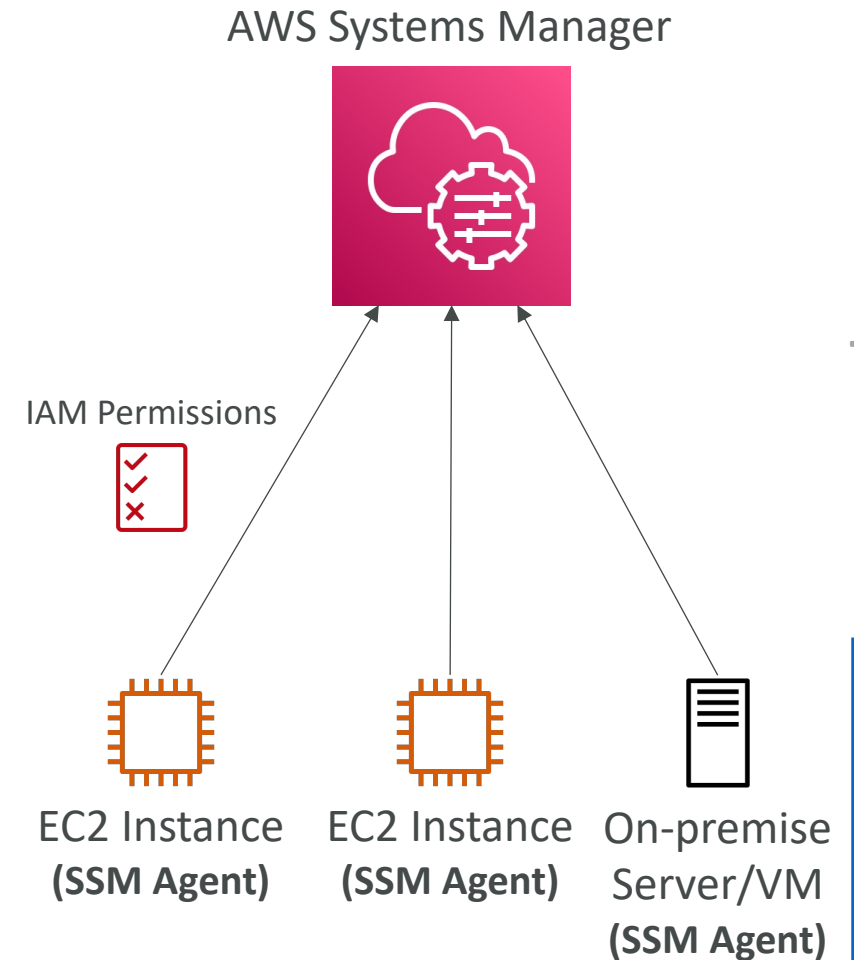
- Helps you manage your **EC2** and **On-Premises** systems at scale
- Get operational insights about the state of your infrastructure
- Easily detect problems
- **Patching automation for enhanced compliance**
- Works for both Windows and Linux OS
- Integrated with CloudWatch metrics / dashboards
- Integrated with AWS Config
- Free service

# AWS Systems Manager Features

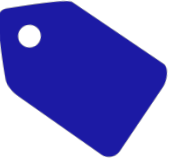
- **Resource Groups**
- Operations Management
  - Explorer
  - OpsCenter
  - CloudWatch Dashboard
  - PHD
  - Incident Manager
- Shared Resources
  - **Documents**
- Change Management
  - Change Manager
  - **Automation**
  - Change Calendar
  - **Maintenance Windows**
- Application Management
  - Application Manager
  - AppConfig
  - **Parameter Store**
- Node Management
  - Fleet Manager
  - Compliance
  - **Inventory**
  - Hybrid Activations
  - **Session Manager**
  - **Run Command**
  - **State Manager**
  - **Patch Manager**
  - Distributer

# How Systems Manager works

- We need to install the SSM agent onto the systems we control
- Installed by default on Amazon Linux 2 AMI & some Ubuntu AMI
- If an instance can't be controlled with SSM, it's probably an issue with the SSM agent!
- Make sure the EC2 instances have a proper IAM role to allow SSM actions



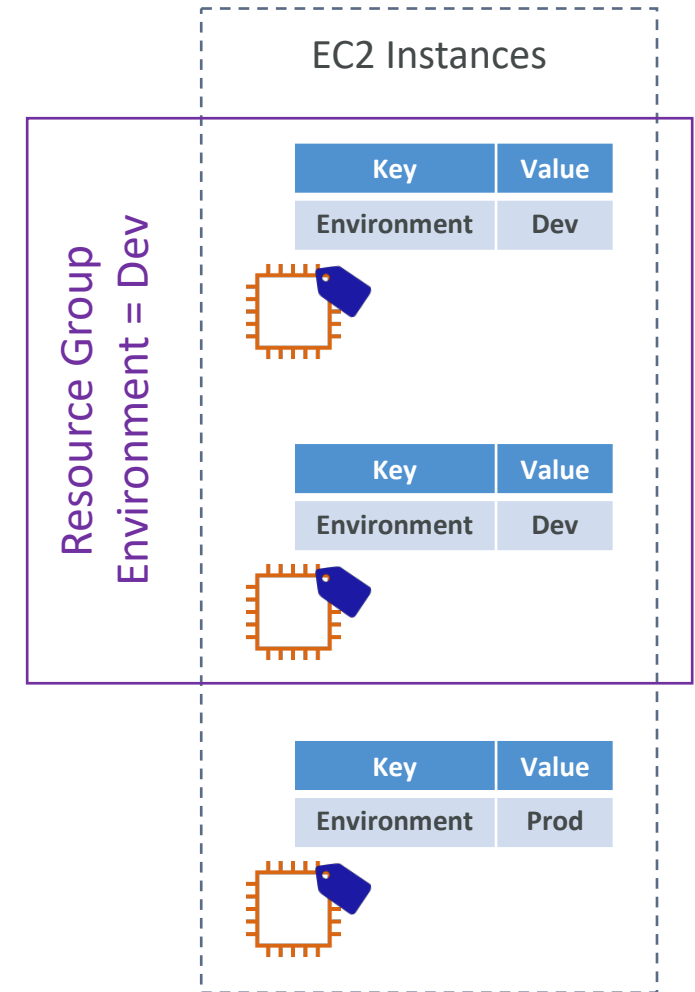
# AWS Tags



- You can add text key-value pairs called Tags to many AWS resources
- Commonly used in EC2
- Free naming, common tags are Name, Environment, Team ...
- They're used for
  - Resource grouping
  - Automation
  - Cost allocation
- Better to have too many tags than too few!

# Resource Groups

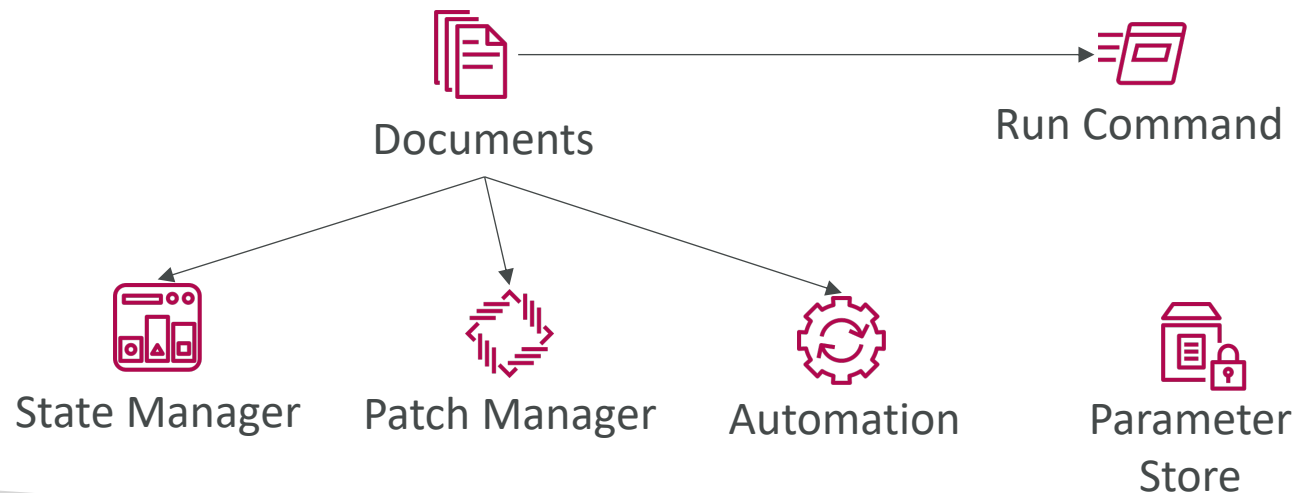
- Create, view or manage logical group of resources thanks to **tags**
- Allows creation of logical groups of resources such as
  - Applications
  - Different layers of an application stack
  - Production versus development environments
- Regional service
- Works with EC2, S3, DynamoDB, Lambda, etc...



# SSM – Documents

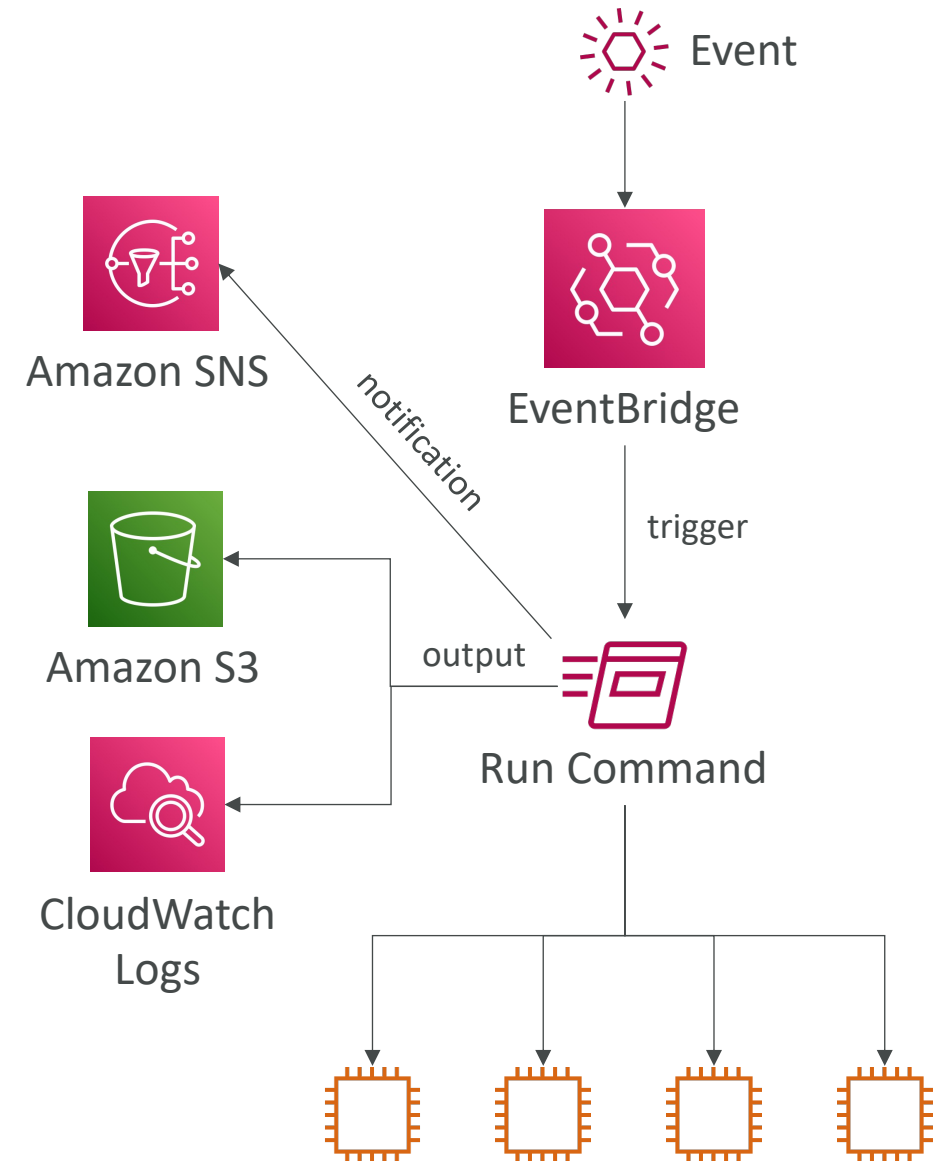
- Documents can be in JSON or YAML
- You define parameters
- You define actions
- Many documents already exist in AWS

```
---
schemaVersion: '2.2'
description: State Manager Bootstrap Example
parameters: {}
mainSteps:
  - action: aws:runShellScript
    name: configureServer
    inputs:
      runCommand:
        - sudo yum install -y httpd
        - sudo yum --enablerepo=epel install -y clamav
```



# SSM – Run Command

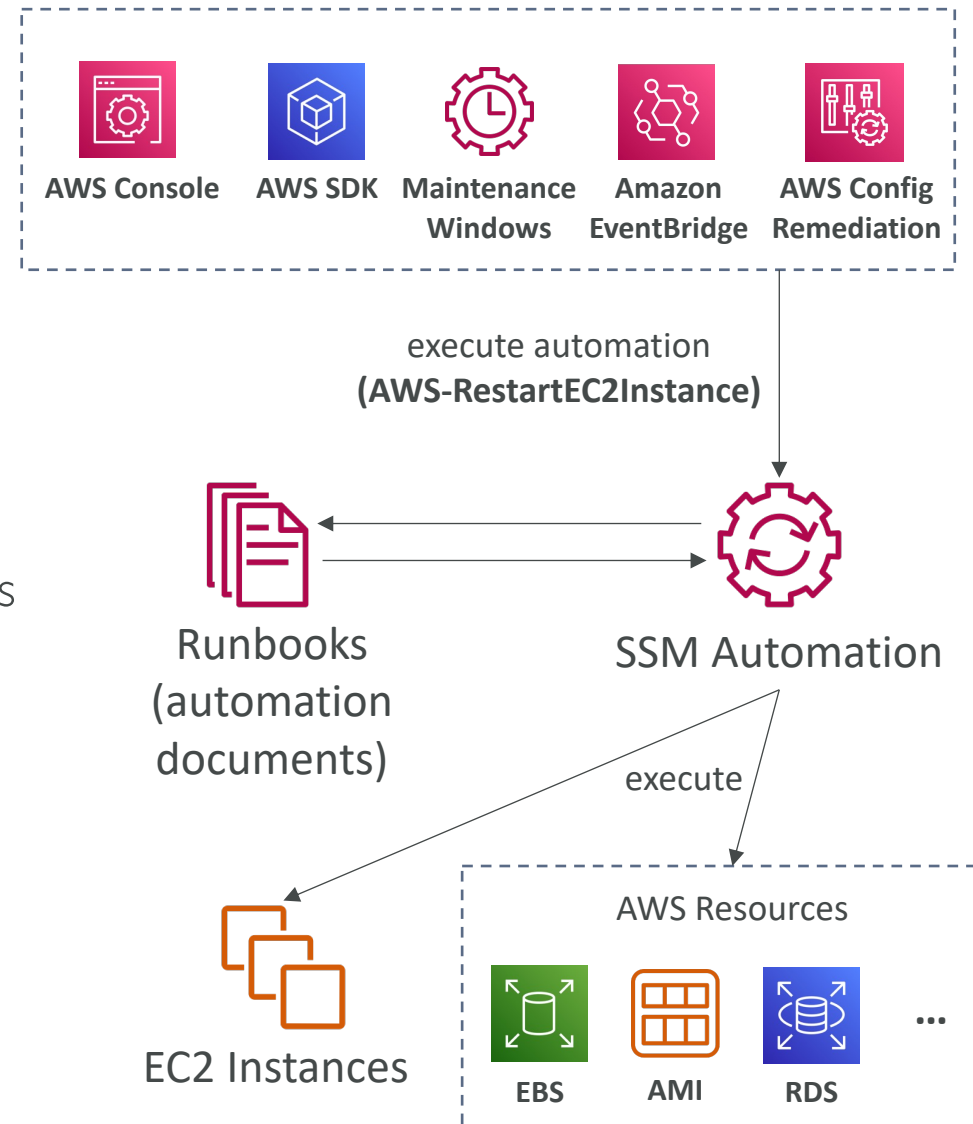
- Execute a document (= script) or just run a command
- Run command across multiple instances (using resource groups)
- Rate Control / Error Control
- Integrated with IAM & CloudTrail
- No need for SSH
- Command Output can be shown in the Console, sent to S3 bucket or CloudWatch Logs
- Send notifications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





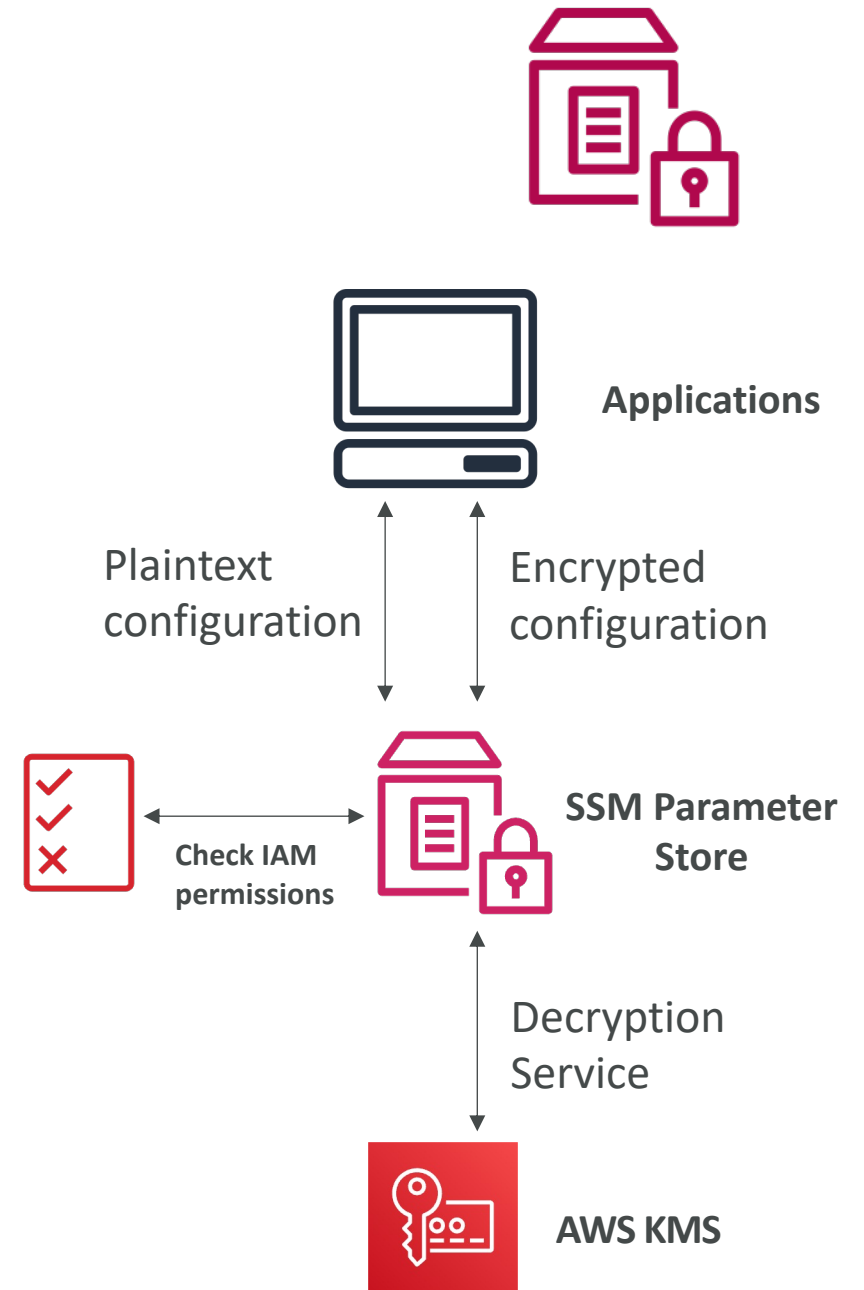
# SSM - Automation

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



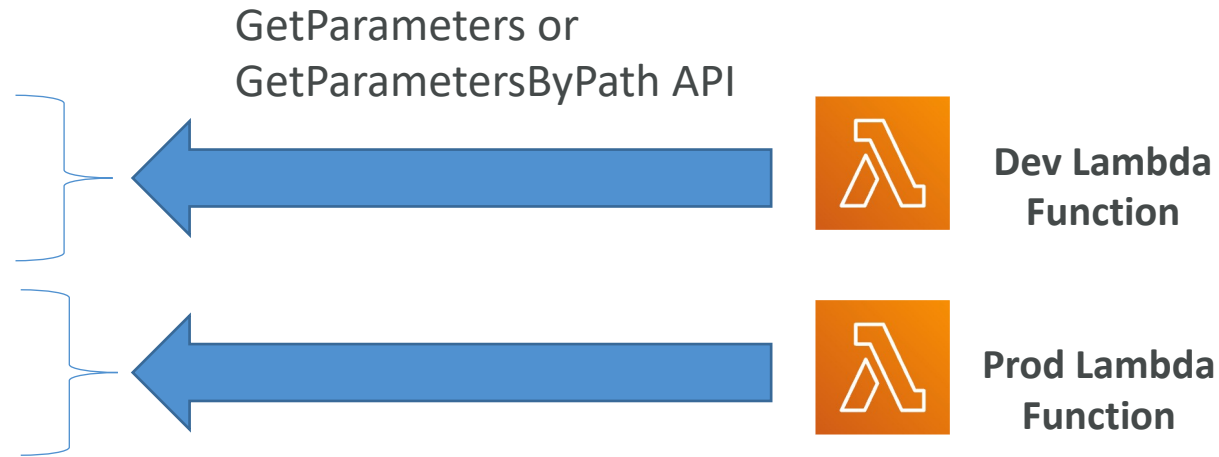
# SSM Parameter Store

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



# SSM Parameter Store Hierarchy

- /my-department/
  - my-app/
    - dev/
      - db-url
      - db-password
    - prod/
      - db-url
      - db-password
  - other-app/
- /other-department/
- */aws/reference/secretsmanager/secret\_ID\_in\_Secrets\_Manager*
- */aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86\_64-gp2 (public)*



# Standard and advanced parameter tiers

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

# Parameters Policies (for advanced parameters)

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

## Expiration (to delete a parameter)

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

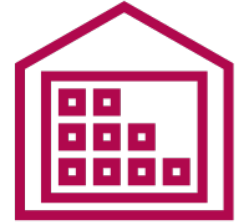
## ExpirationNotification (EventBridge)

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

## NoChangeNotification (EventBridge)

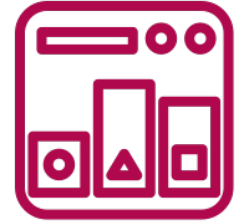
```
{
  "Type": "NoChangeNotification",
  "Version": "1.0",
  "Attributes": {
    "After": "20",
    "Unit": "Days"
  }
}
```

# SSM – Inventory



- Collect metadata from your managed instances (EC2/On-premises)
- Metadata includes installed software, OS drivers, configurations, installed updates, running services ...
- View data in AWS Console or store in S3 and query and analyze using Athena and QuickSight
- Specify metadata collection interval (minutes, hours, days)
- Query data from multiple AWS accounts and regions
- Create Custom Inventory for your custom metadata (e.g., rack location of each managed instance)

# SSM – State Manager



- Automate the process of keeping your managed instances (EC2/On-premises) in a state that you define
- Use cases: bootstrap instances with software, patch OS/software updates on a schedule ...
- **State Manager Association:**
  - Defines the state that you want to maintain to your managed instances
  - Example: port 22 must be closed, antivirus must be installed ...
  - Specify a schedule when this configuration is applied
- Uses SSM Documents to create an Association (e.g., SSM Document to configure CW Agent)

# SSM – Patch Manager



- Automates the process of patching managed instances
- OS updates, applications updates, security updates, ...
- Supports both EC2 instances and on-premises servers
- Supports Linux, macOS, and Windows
- Patch on-demand or on a schedule using **Maintenance Windows**
- Scan instances and generate patch compliance report (missing patches)
- Patch compliance report can be sent to S3



# SSM – Patch Manager



- **Patch Baseline**
  - Defines which patches should and shouldn't be installed on your instances
  - Ability to create custom Patch Baselines (specify approved/rejected patches)
  - Patches can be auto-approved within days of their release
  - By default, install only critical patches and patches related to security
- **Patch Group**
  - Associate a set of instances with a specific Patch Baseline
  - Example: create Patch Groups for different environments (dev, test, prod)
  - Instances should be defined with the tag key **Patch Group**
  - An instance can only be in one Patch Group
  - Patch Group can be registered with only one Patch Baseline

# SSM – Patch Manager



AWS Console



AWS SDK

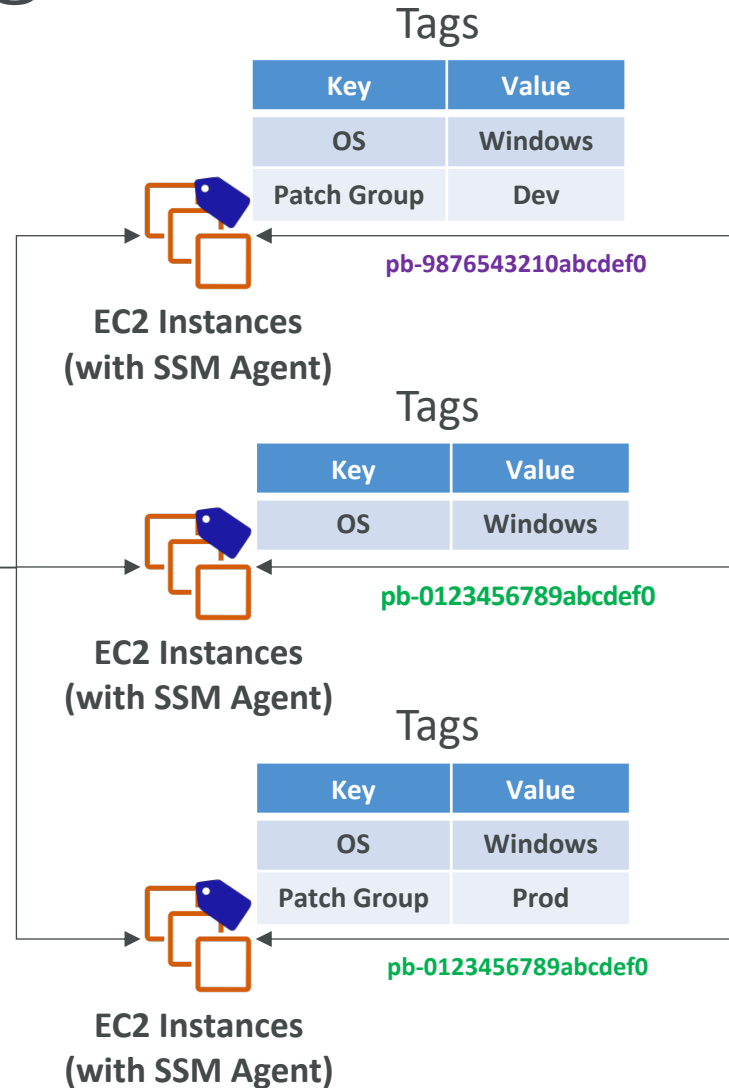


Maintenance Windows

run Document  
**AWS-RunPatchBaseline**

Run Command

When using **Maintenance Windows**, use **Rate Control** to specify the max. number/percentage of targets allowed to run in parallel



SSM Agent query for Batch Baselines



Patch Baselines

Patch Baseline ID	Patch Group	Default
pb-0123456789abcdef0	Default	Yes
pb-9876543210abcdef0	Dev	No

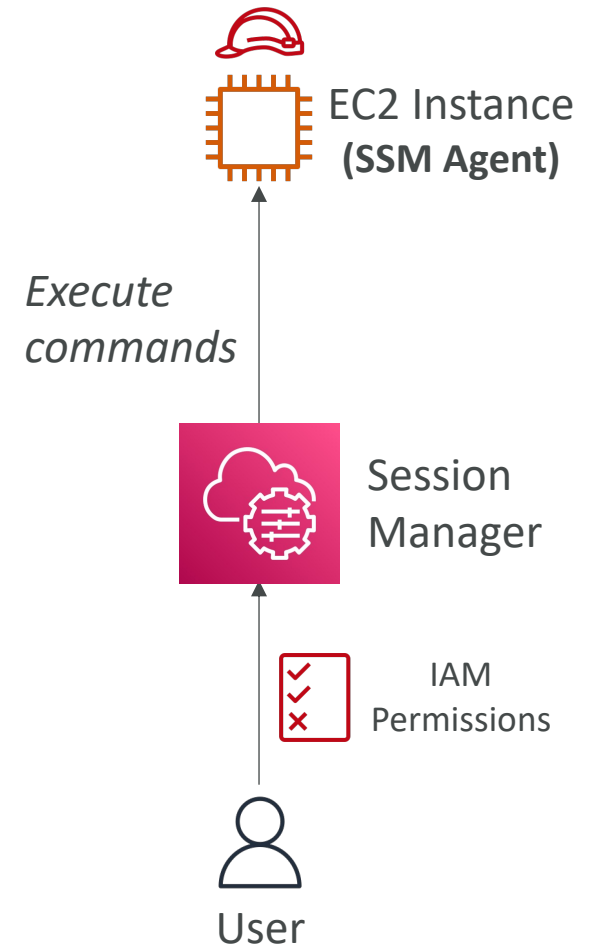


# SSM – Maintenance Windows

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

# SSM – Session Manager

- Allows you to start a secure shell on your EC2 and on-premises servers
- Access through AWS Console, AWS CLI, or Session Manager SDK
- **Does not need SSH access, bastion hosts, or SSH keys**
- Supports Linux, macOS, and Windows
- Log connections to your instances and executed commands
- Session log data can be sent to S3 or CloudWatch Logs
- CloudTrail can intercept **StartSession** events



# SSM – Session Manager

- IAM Permissions
  - Control which users/groups can access Session Manager and which instances
  - Use tags to restrict access to only specific EC2 instances
  - Access SSM + write to S3 + write to CloudWatch
- Optionally, you can restrict commands a user can run in a session

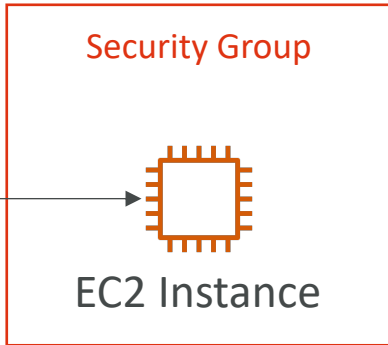
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ssm:StartSession",
      "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
      "Condition": {
        "StringLike": {
          "ssm:resourceTag/Environment": ["Dev"]
        }
      }
    }
  ]
}
```

# SSH vs. SSM Session Manager

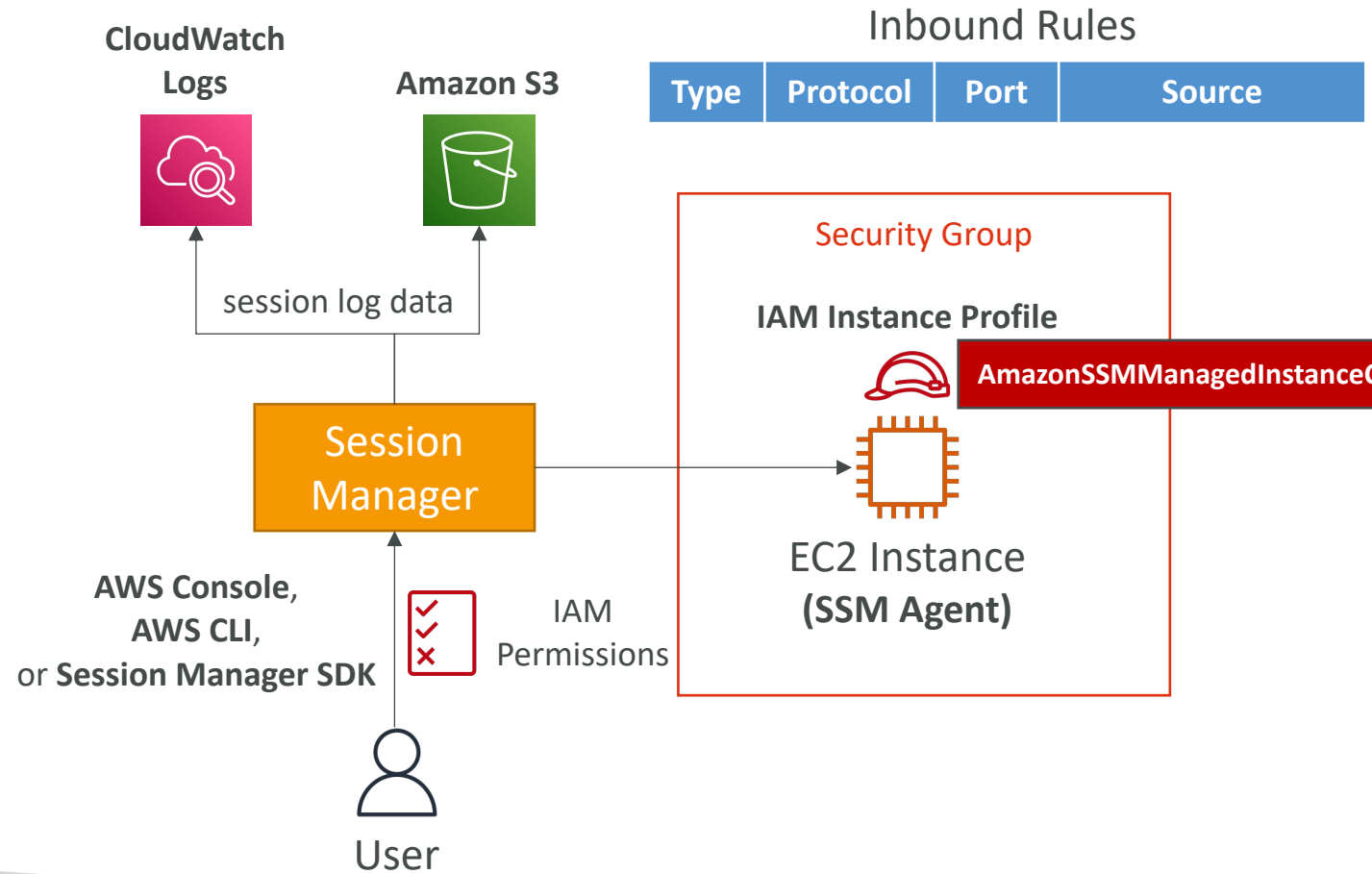
## Connect using SSH

Inbound Rules

Type	Protocol	Port	Source
SSH	TCP	22	1.2.3.4/32

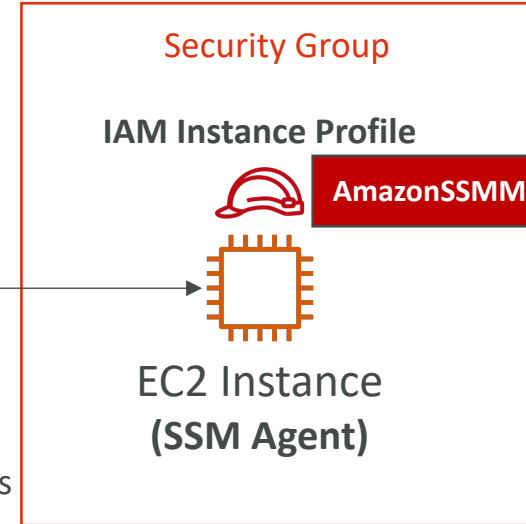


## Connect using SSM Session Manager

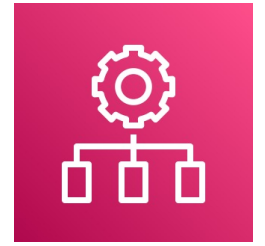


Inbound Rules

Type	Protocol	Port	Source
------	----------	------	--------



# AWS OpsWorks



- Chef & Puppet help you perform server configuration automatically, or repetitive actions
- They work great with EC2 & On-Premises VM
- AWS OpsWorks = Managed Chef & Puppet
- It's an alternative to AWS SSM
  
- No hands on here, no knowledge of chef and puppet needed
- In the exam: Chef & Puppet needed => AWS OpsWorks



# Quick word on Chef / Puppet

- They help with managing configuration as code
- Helps in having consistent deployments
- Works with Linux / Windows
- Can automate user accounts, cron, ntp, packages, services...
- They leverage "Recipes" or "Manifests"
- Chef / Puppet have similarities with SSM / Beanstalk / CloudFormation but they're open-source tools that work cross-cloud



# EC2 High Availability and Scalability

Load Balancer and Auto Scaling Groups

# Scalability and High Availability Section

- Load Balancers:
  - Troubleshooting
  - Advanced options and logging
  - CloudWatch integrations
- Auto Scaling
  - Troubleshooting
  - Advanced options and logging
  - CloudWatch integrations

# Scalability & High Availability

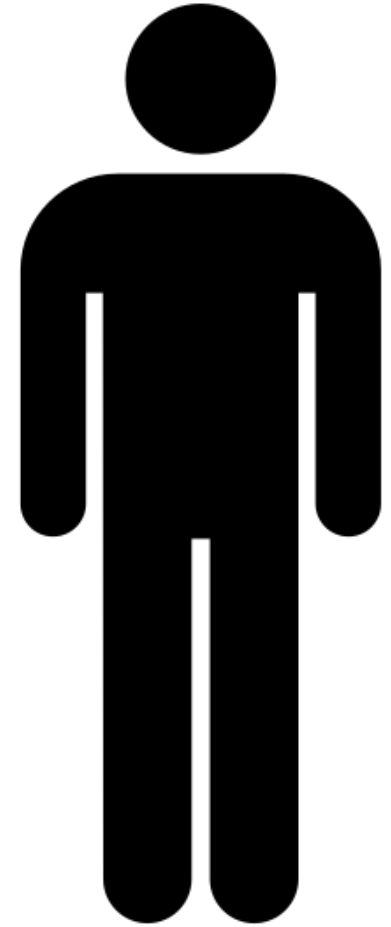
- Scalability means that an application / system can handle greater loads by adapting.
- There are two kinds of scalability:
  - Vertical Scalability
  - Horizontal Scalability (= elasticity)
- Scalability is linked but different to High Availability
- Let's deep dive into the distinction, using a call center as an example

# Vertical Scalability

- Vertically scalability means increasing the size of the instance
- For example, your application runs on a t2.micro
- Scaling that application vertically means running it on a t2.large
- Vertical scalability is very common for non distributed systems, such as a database.
- RDS, ElastiCache are services that can scale vertically.
- There's usually a limit to how much you can vertically scale (hardware limit)



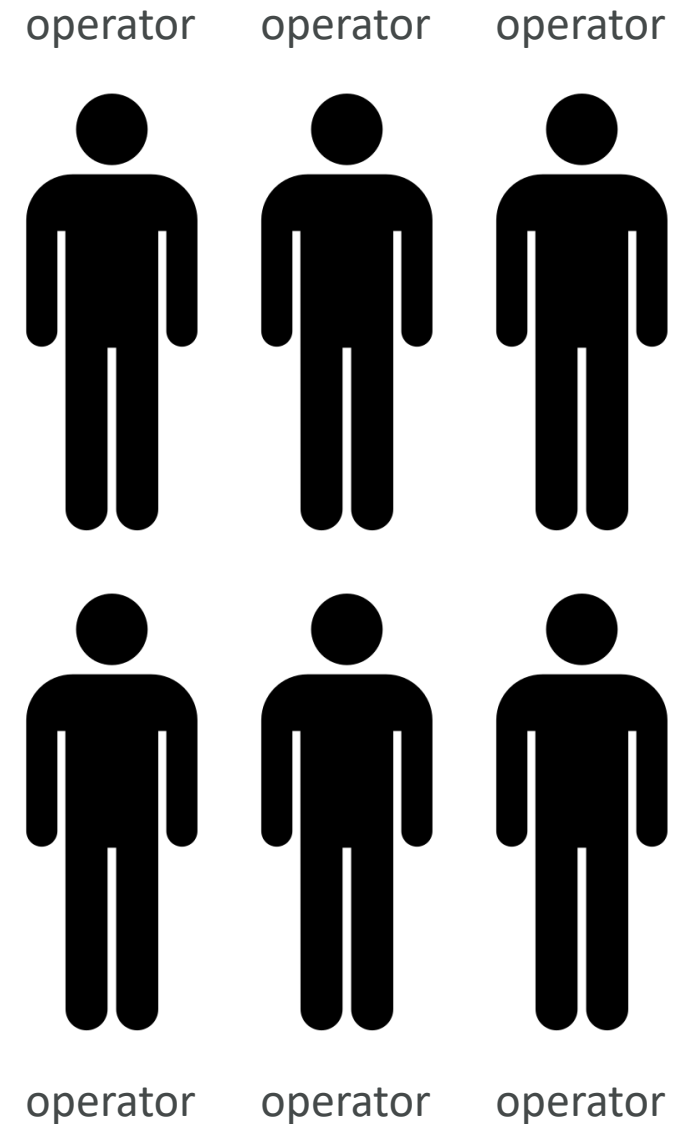
junior operator



senior operator

# Horizontal Scalability

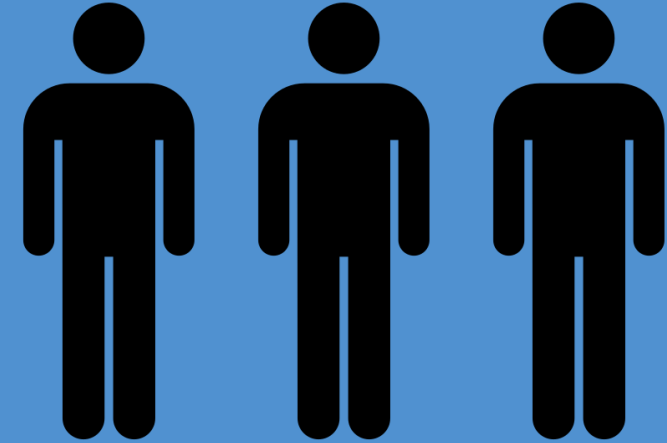
- Horizontal Scalability means increasing the number of instances / systems for your application
- Horizontal scaling implies distributed systems.
- This is very common for web applications / modern applications
- It's easy to horizontally scale thanks the cloud offerings such as Amazon EC2



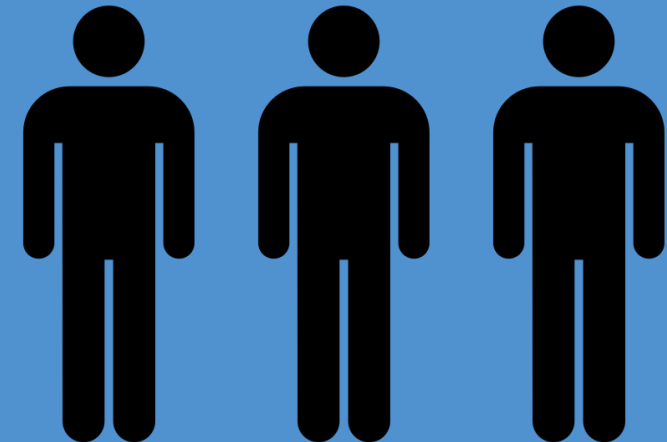
# High Availability

- High Availability usually goes hand in hand with horizontal scaling
- High availability means running your application / system in at least 2 data centers (== Availability Zones)
- The goal of high availability is to survive a data center loss
  
- The high availability can be passive (for RDS Multi AZ for example)
- The high availability can be active (for horizontal scaling)

first building in New York



second building in San Francisco

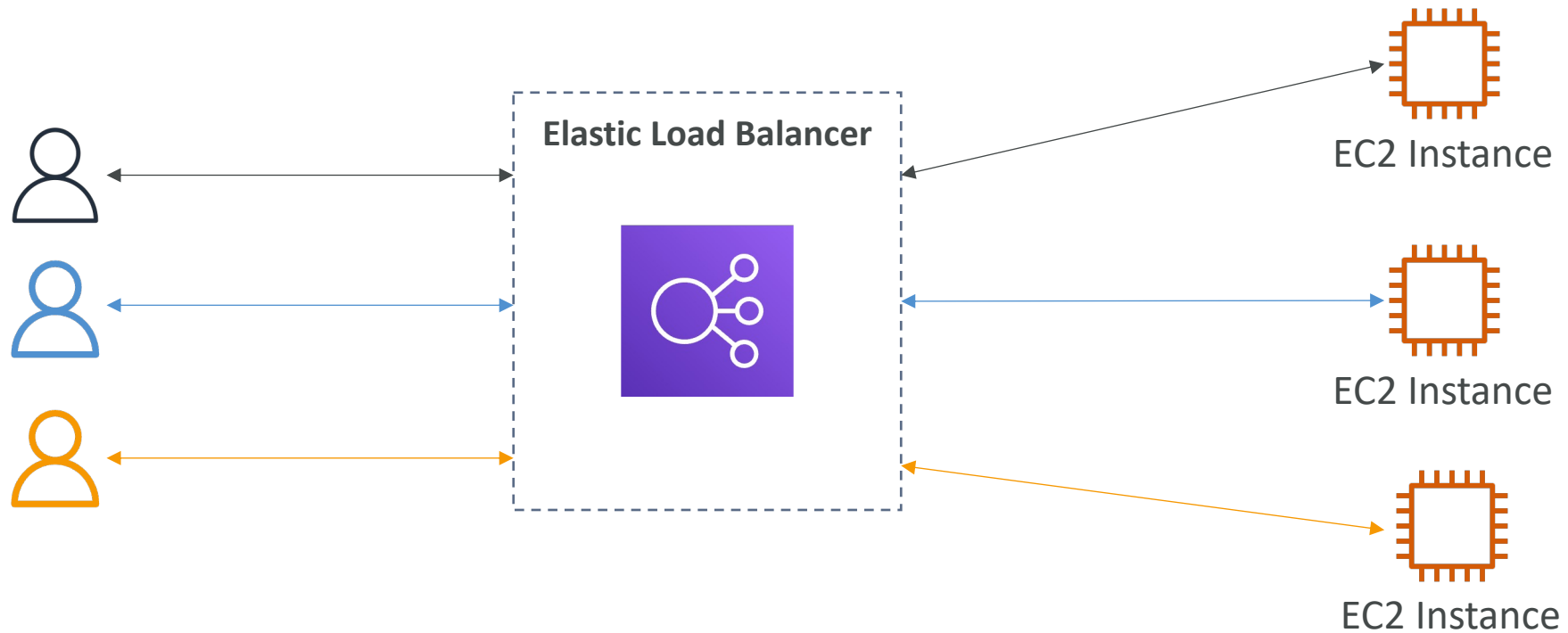


# High Availability & Scalability For EC2

- Vertical Scaling: Increase instance size (= scale up / down)
  - From: t2.nano - 0.5G of RAM, 1 vCPU
  - To: u-12tb1.metal – 12.3 TB of RAM, 448 vCPUs
- Horizontal Scaling: Increase number of instances (= scale out / in)
  - Auto Scaling Group
  - Load Balancer
- High Availability: Run instances for the same application across multi-AZ
  - Auto Scaling Group multi-AZ
  - Load Balancer multi-AZ

# What is load balancing?

- Load Balancers are servers that forward traffic to multiple servers (e.g., EC2 instances) downstream





# Why use a load balancer?

- Spread load across multiple downstream instances
- Expose a single point of access (DNS) to your application
- Seamlessly handle failures of downstream instances
- Do regular health checks to your instances
- Provide SSL termination (HTTPS) for your websites
- Enforce stickiness with cookies
- High availability across zones
- Separate public traffic from private traffic

# Why use an Elastic Load Balancer?

- An Elastic Load Balancer is a **managed load balancer**
  - AWS guarantees that it will be working
  - AWS takes care of upgrades, maintenance, high availability
  - AWS provides only a few configuration knobs
- It costs less to setup your own load balancer but it will be a lot more effort on your end
- It is integrated with many AWS offerings / services
  - EC2, EC2 Auto Scaling Groups, Amazon ECS
  - AWS Certificate Manager (ACM), CloudWatch
  - Route 53, AWS WAF, AWS Global Accelerator

# Health Checks

- Health Checks are crucial for Load Balancers
- They enable the load balancer to know if instances it forwards traffic to are available to reply to requests
- The health check is done on a port and a route (/health is common)
- If the response is not 200 (OK), then the instance is unhealthy





# Types of load balancer on AWS

- AWS has **4 kinds of managed Load Balancers**
- **Classic Load Balancer** (v1 - old generation) – 2009 – CLB
  - HTTP, HTTPS, TCP, SSL (secure TCP)
- **Application Load Balancer** (v2 - new generation) – 2016 – ALB
  - HTTP, HTTPS, WebSocket
- **Network Load Balancer** (v2 - new generation) – 2017 – NLB
  - TCP, TLS (secure TCP), UDP
- **Gateway Load Balancer** – 2020 – GWLB
  - Operates at layer 3 (Network layer) – IP Protocol
- Overall, it is recommended to use the newer generation load balancers as they provide more features
- Some load balancers can be setup as **internal** (private) or **external** (public) ELBs

# Load Balancer Security Groups



## Load Balancer Security Group:

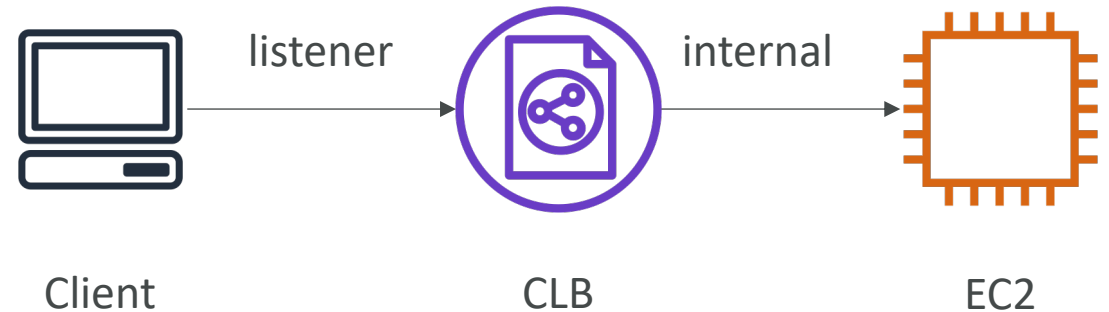
Type <a href="#">i</a>	Protocol <a href="#">i</a>	Port Range <a href="#">i</a>	Source <a href="#">i</a>	Description <a href="#">i</a>
HTTP	TCP	80	0.0.0.0/0	Allow HTTP from an...
HTTPS	TCP	443	0.0.0.0/0	Allow HTTPS from a...

## Application Security Group: Allow traffic only from Load Balancer

Type <a href="#">i</a>	Protocol <a href="#">i</a>	Port Range <a href="#">i</a>	Source <a href="#">i</a>	Description <a href="#">i</a>
HTTP	TCP	80	sg-054b5ff5ea02f2b6e (load-b	Allow Traffic only...

# Classic Load Balancers (v1)

- Supports TCP (Layer 4), HTTP & HTTPS (Layer 7)
- Health checks are TCP or HTTP based
- Fixed hostname  
XXX.region.elb.amazonaws.com





# Application Load Balancer (v2)

- Application load balancers is Layer 7 (HTTP)
- Load balancing to multiple HTTP applications across machines (target groups)
- Load balancing to multiple applications on the same machine (ex: containers)
- Support for HTTP/2 and WebSocket
- Support redirects (from HTTP to HTTPS for example)

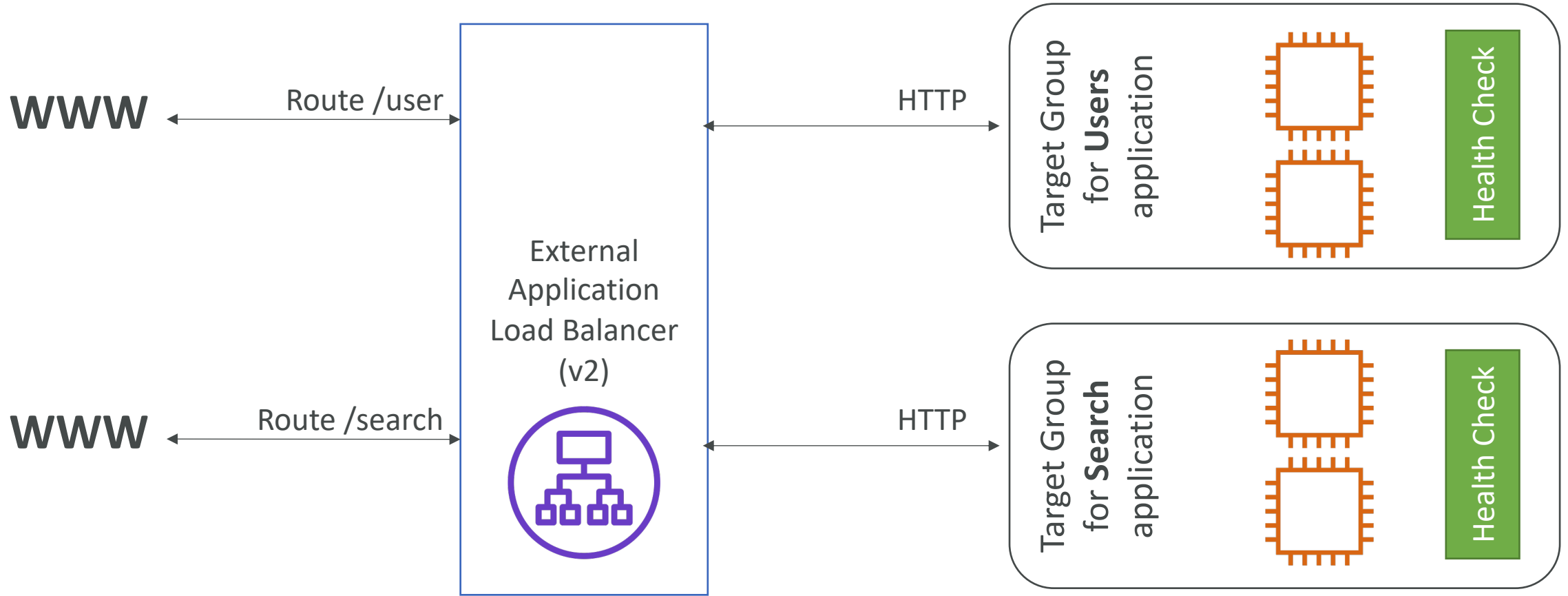


# Application Load Balancer (v2)

- Routing tables to different target groups:
  - Routing based on path in URL (example.com/[users](#) & example.com/[posts](#))
  - Routing based on hostname in URL ([one.example.com](#) & [other.example.com](#))
  - Routing based on Query String, Headers (example.com/users?[id=123&order=false](#))
- ALB are a great fit for micro services & container-based application (example: Docker & Amazon ECS)
- Has a port mapping feature to redirect to a dynamic port in ECS
- In comparison, we'd need multiple Classic Load Balancer per application



# Application Load Balancer (v2) HTTP Based Traffic

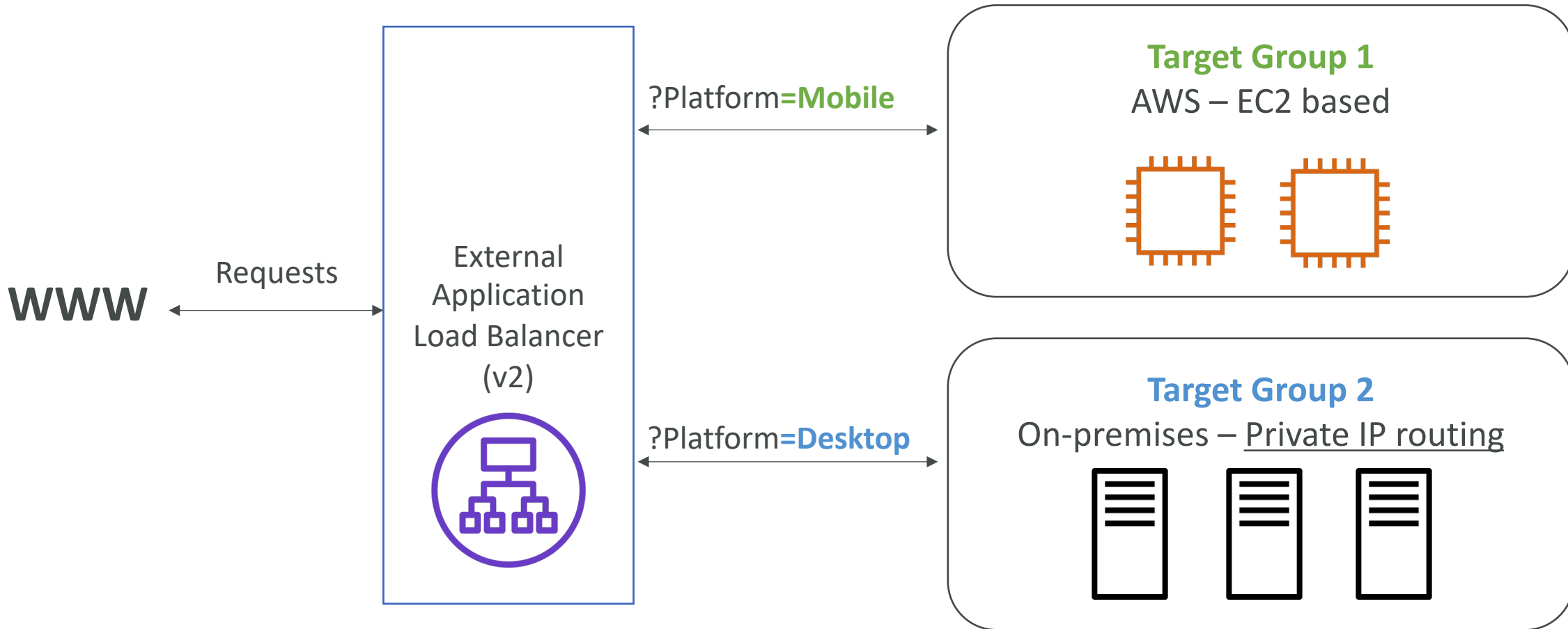


# Application Load Balancer (v2)

## Target Groups

- EC2 instances (can be managed by an Auto Scaling Group) – HTTP
  - ECS tasks (managed by ECS itself) – HTTP
  - Lambda functions – HTTP request is translated into a JSON event
  - IP Addresses – must be private IPs
- 
- ALB can route to multiple target groups
  - Health checks are at the target group level

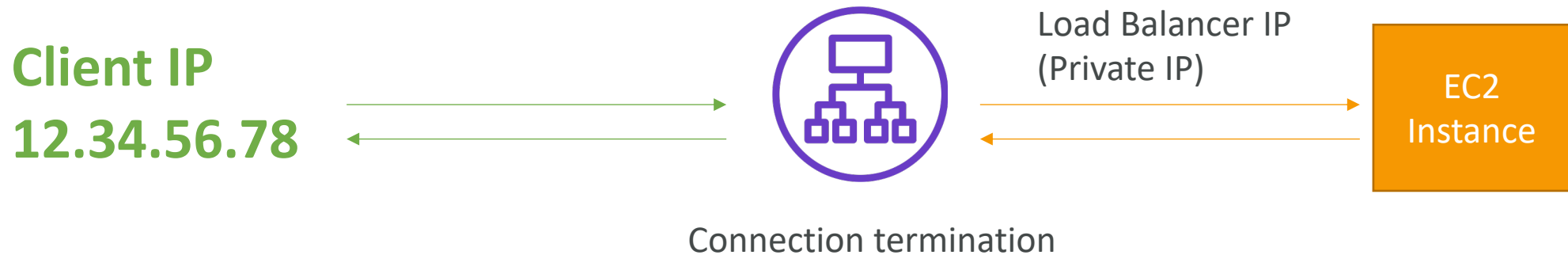
# Application Load Balancer (v2) Query Strings/Parameters Routing

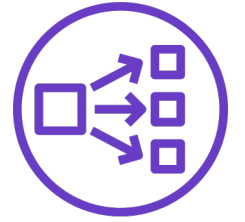


# Application Load Balancer (v2)

## Good to Know

- Fixed hostname (XXX.region.elb.amazonaws.com)
- The application servers don't see the IP of the client directly
  - The true IP of the client is inserted in the header **X-Forwarded-For**
  - We can also get Port (X-Forwarded-Port) and proto (X-Forwarded-Proto)

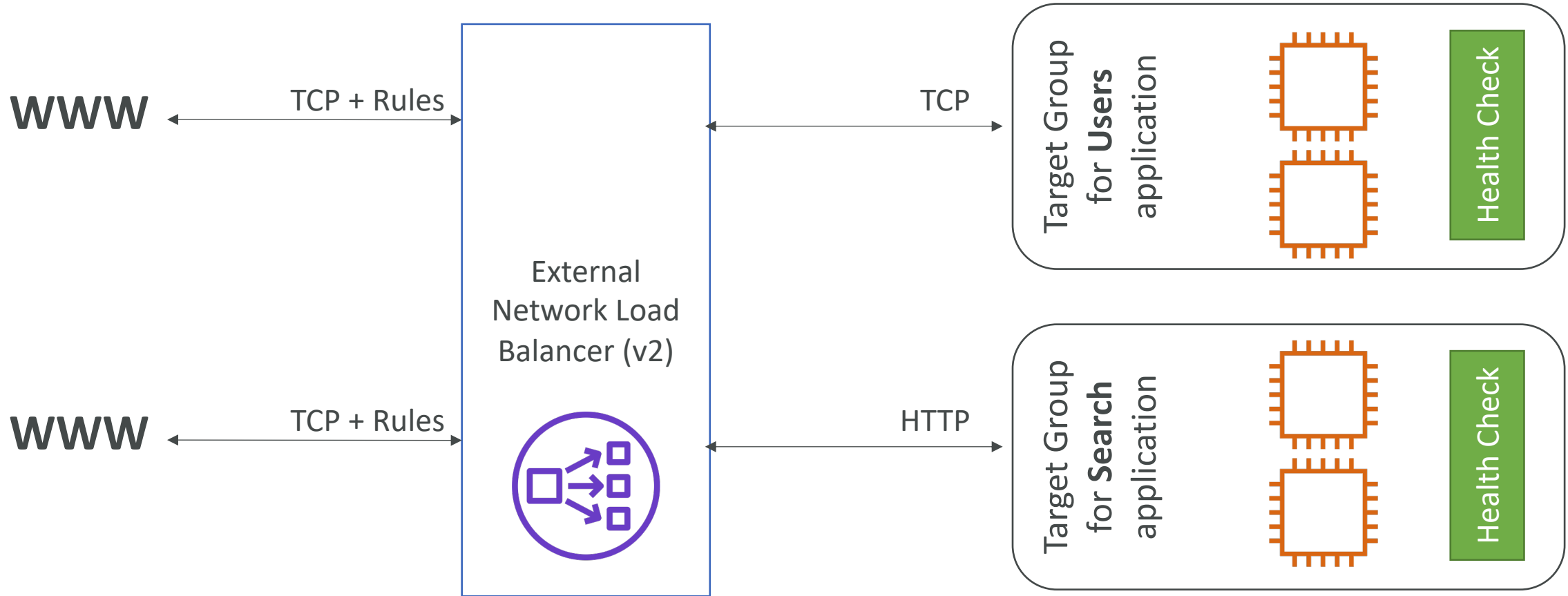




# Network Load Balancer (v2)

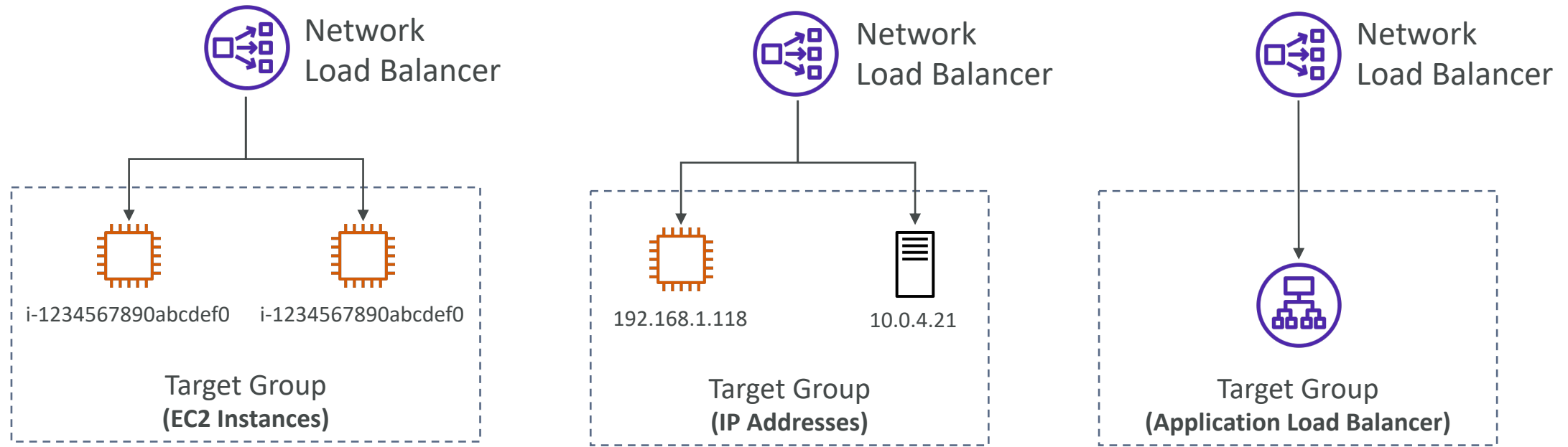
- Network load balancers (Layer 4) allow to:
  - Forward TCP & UDP traffic to your instances
  - Handle millions of request per seconds
  - Less latency ~ 100 ms (vs 400 ms for ALB)
- NLB has one static IP per AZ, and supports assigning Elastic IP (helpful for whitelisting specific IP)
- NLB are used for extreme performance, TCP or UDP traffic
- Not included in the AWS free tier

# Network Load Balancer (v2) TCP (Layer 4) Based Traffic



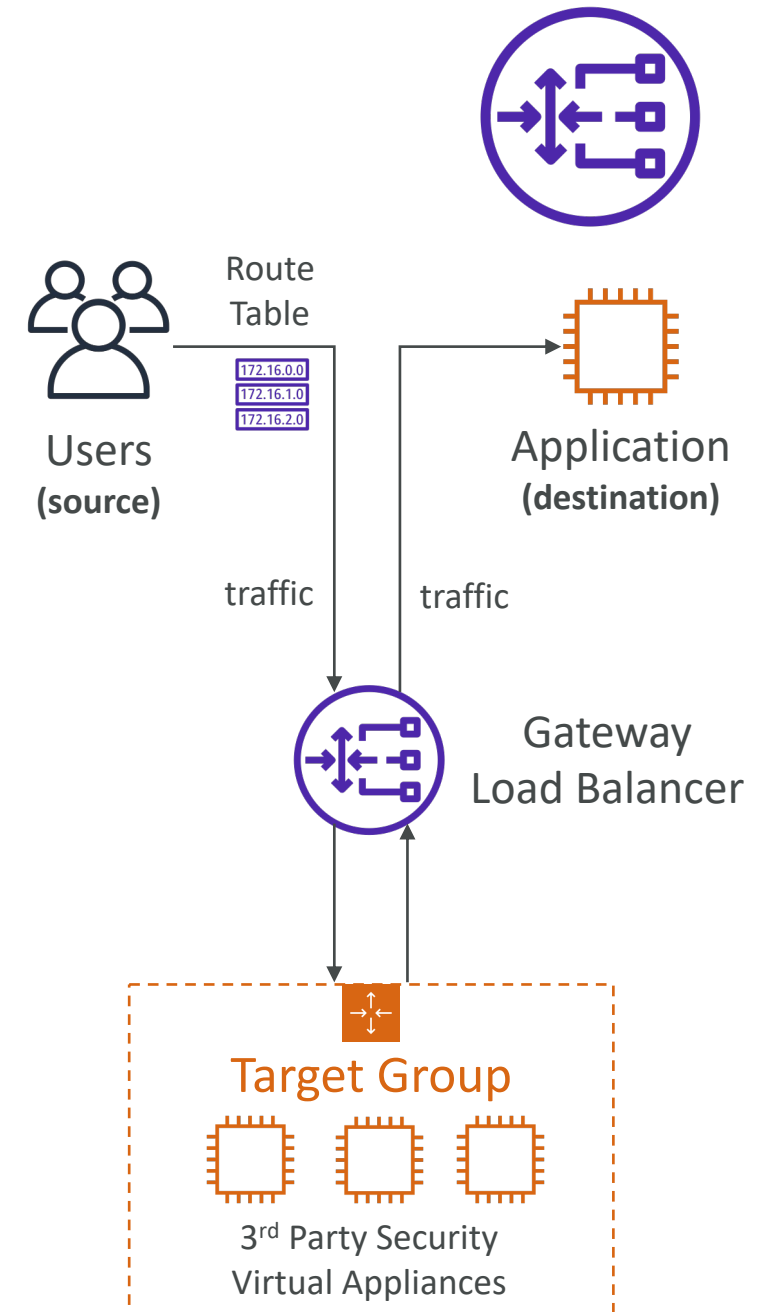
# Network Load Balancer – Target Groups

- EC2 instances
- IP Addresses – must be private IPs
- Application Load Balancer
- Health Checks support the TCP, HTTP and HTTPS Protocols



# Gateway Load Balancer

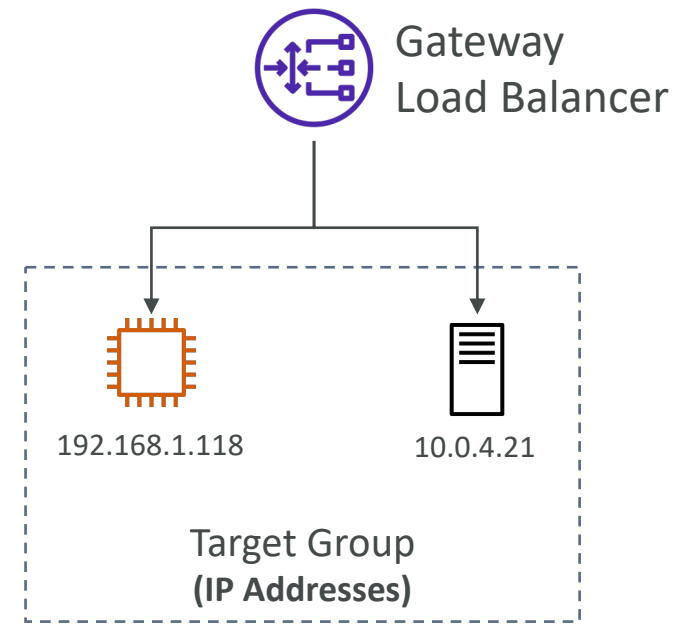
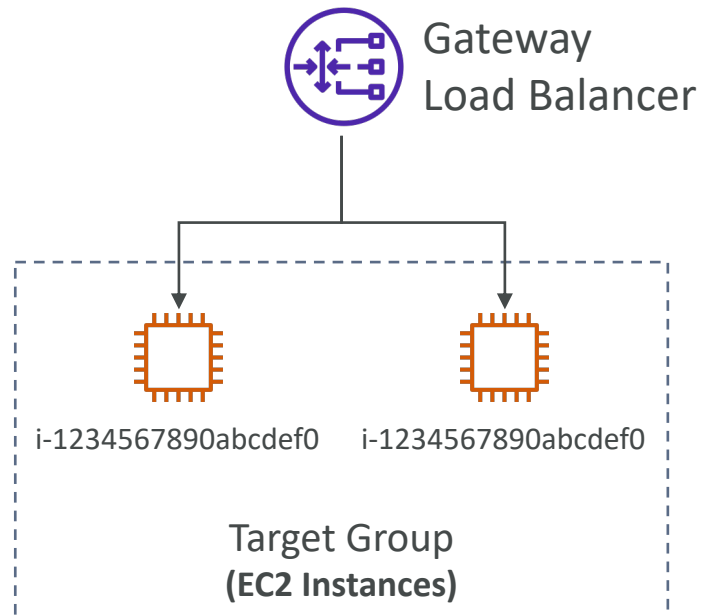
- Deploy, scale, and manage a fleet of 3<sup>rd</sup> party network virtual appliances in AWS
- Example: Firewalls, Intrusion Detection and Prevention Systems, Deep Packet Inspection Systems, payload manipulation, ...
- Operates at Layer 3 (Network Layer) – IP Packets
- Combines the following functions:
  - **Transparent Network Gateway** – single entry/exit for all traffic
  - **Load Balancer** – distributes traffic to your virtual appliances
- Uses the **GENEVE** protocol on port **6081**





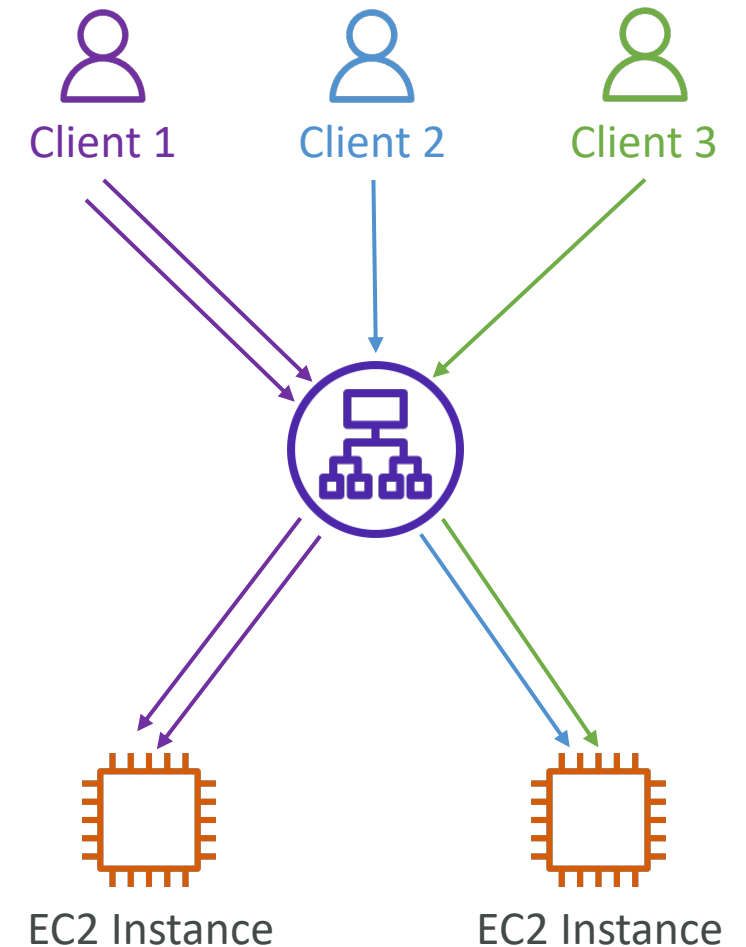
# Gateway Load Balancer – Target Groups

- EC2 instances
- IP Addresses – must be private IPs



# Sticky Sessions (Session Affinity)

- It is possible to implement stickiness so that the same client is always redirected to the same instance behind a load balancer
- This works for Classic Load Balancers & Application Load Balancers
- The “cookie” used for stickiness has an expiration date you control
- Use case: make sure the user doesn't lose his session data
- Enabling stickiness may bring imbalance to the load over the backend EC2 instances



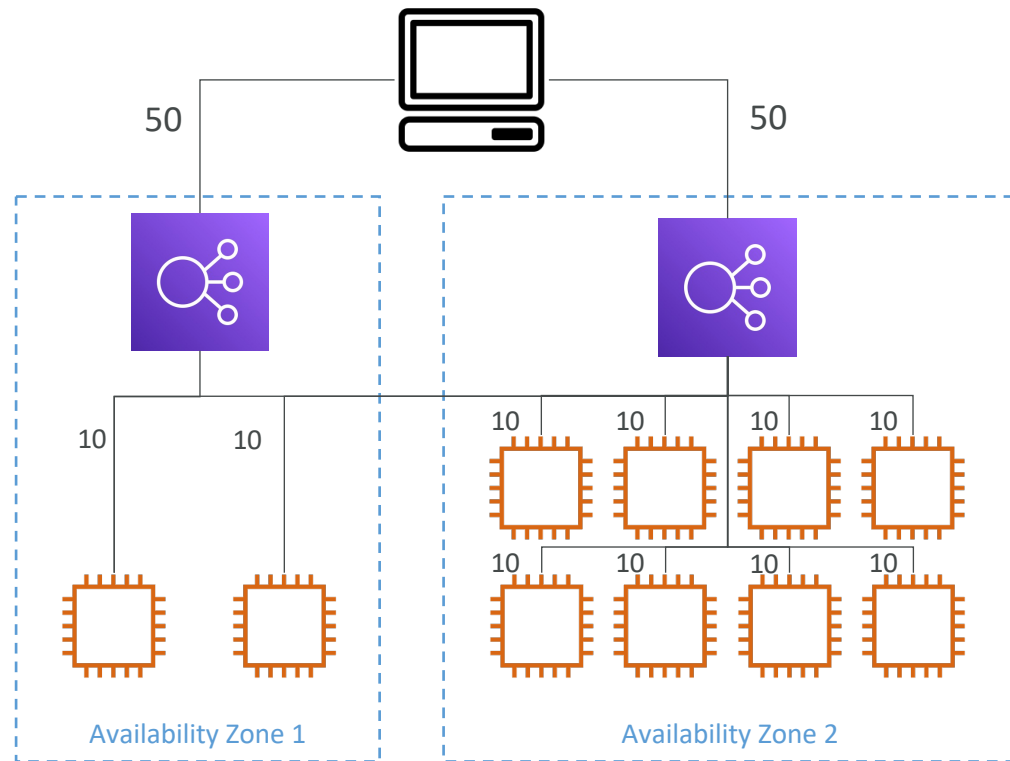
# Sticky Sessions – Cookie Names

- Application-based Cookies
  - Custom cookie
    - Generated by the target
    - Can include any custom attributes required by the application
    - Cookie name must be specified individually for each target group
    - Don't use **AWSALB**, **AWSALBAPP**, or **AWSALBTG** (reserved for use by the ELB)
  - Application cookie
    - Generated by the load balancer
    - Cookie name is **AWSALBAPP**
- Duration-based Cookies
  - Cookie generated by the load balancer
  - Cookie name is **AWSALB** for ALB, **AWSELB** for CLB

# Cross-Zone Load Balancing

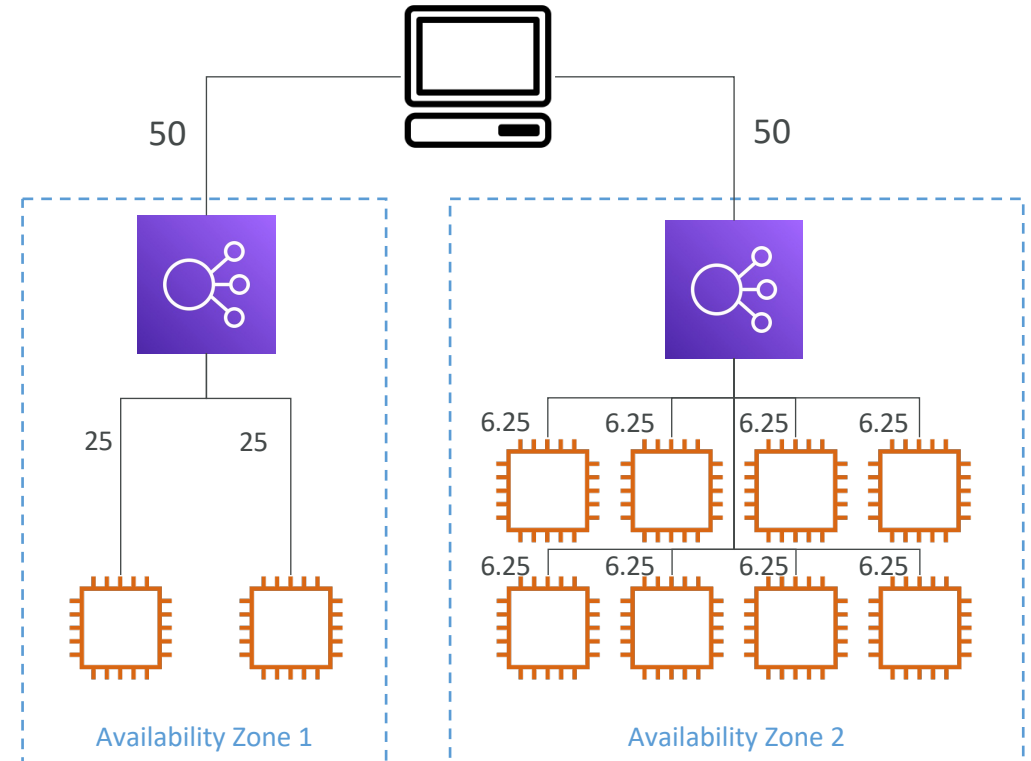
## With Cross Zone Load Balancing:

each load balancer instance distributes evenly across all registered instances in all AZ



## Without Cross Zone Load Balancing:

Requests are distributed in the instances of the node of the Elastic Load Balancer



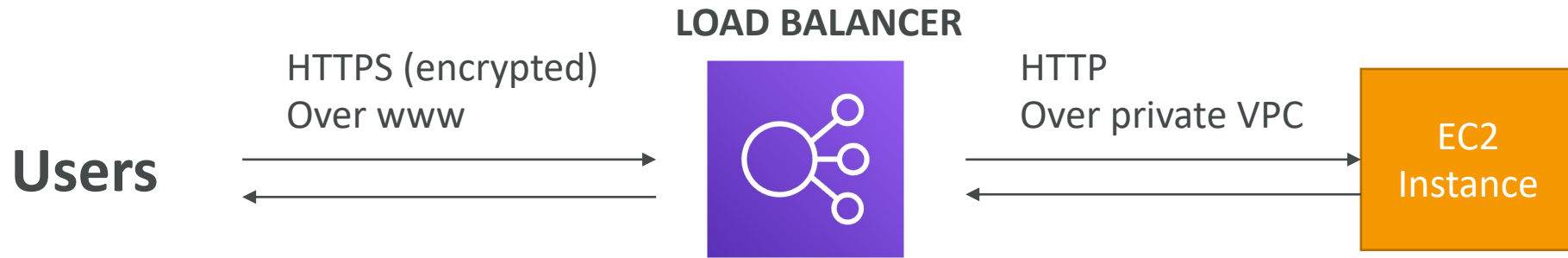
# Cross-Zone Load Balancing

- **Application Load Balancer**
  - Enabled by default (can be disabled at the Target Group level)
  - No charges for inter AZ data
- **Network Load Balancer & Gateway Load Balancer**
  - Disabled by default
  - You pay charges (\$) for inter AZ data if enabled
- **Classic Load Balancer**
  - Disabled by default
  - No charges for inter AZ data if enabled

# SSL/TLS - Basics

- An SSL Certificate allows traffic between your clients and your load balancer to be encrypted in transit (in-flight encryption)
- **SSL** refers to Secure Sockets Layer, used to encrypt connections
- **TLS** refers to Transport Layer Security, which is a newer version
- Nowadays, **TLS certificates are mainly used**, but people still refer as SSL
- Public SSL certificates are issued by Certificate Authorities (CA)
- Comodo, Symantec, GoDaddy, GlobalSign, Digicert, Letsencrypt, etc...
- SSL certificates have an expiration date (you set) and must be renewed

# Load Balancer - SSL Certificates



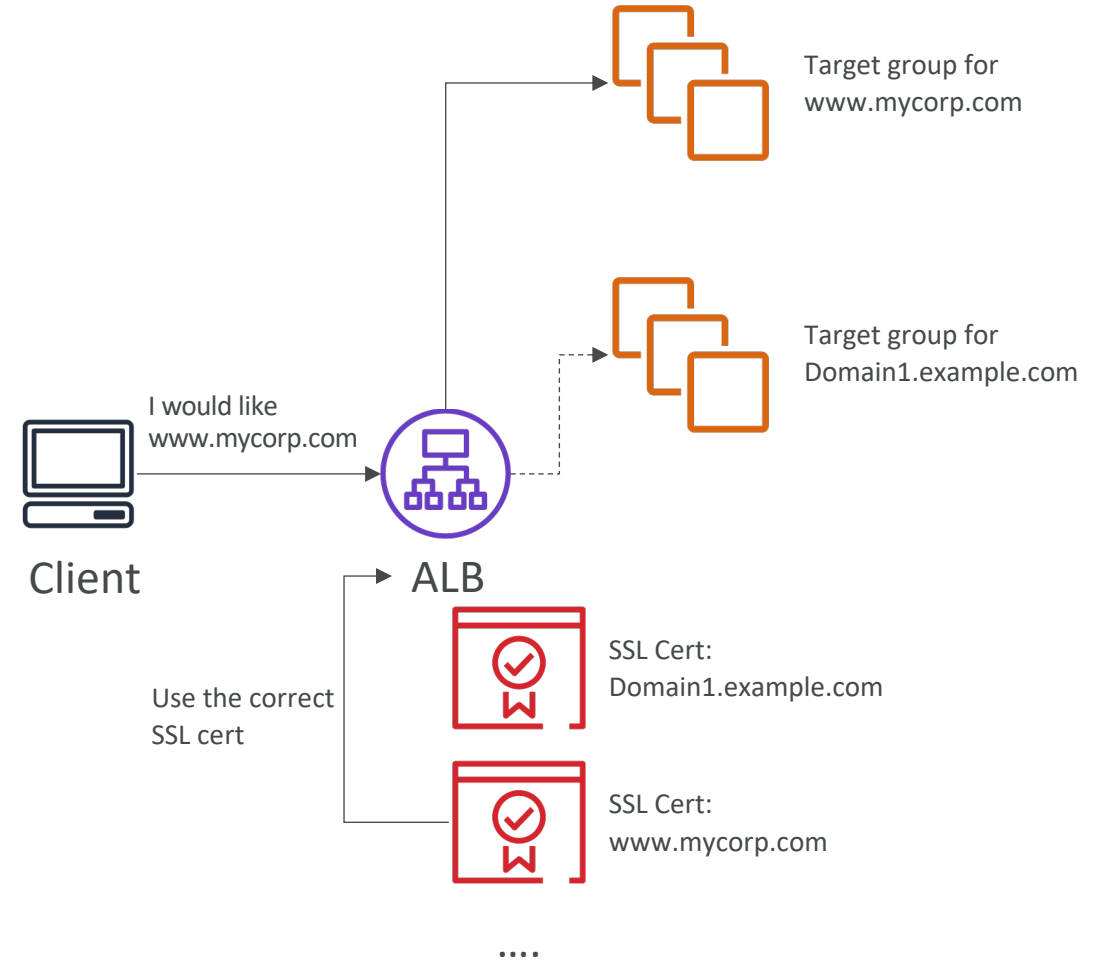
- The load balancer uses an X.509 certificate (SSL/TLS server certificate)
- You can manage certificates using ACM (AWS Certificate Manager)
- You can create upload your own certificates alternatively
- HTTPS listener:
  - You must specify a default certificate
  - You can add an optional list of certs to support multiple domains
  - **Clients can use SNI (Server Name Indication) to specify the hostname they reach**
  - Ability to specify a security policy to support older versions of SSL / TLS (legacy clients)

# SSL – Server Name Indication (SNI)

- SNI solves the problem of loading **multiple SSL certificates onto one web server** (to serve multiple websites)
- It's a “newer” protocol, and requires the client to **indicate** the hostname of the target server in the initial SSL handshake
- The server will then find the correct certificate, or return the default one

## Note:

- Only works for ALB & NLB (newer generation), CloudFront
- Does not work for CLB (older gen)



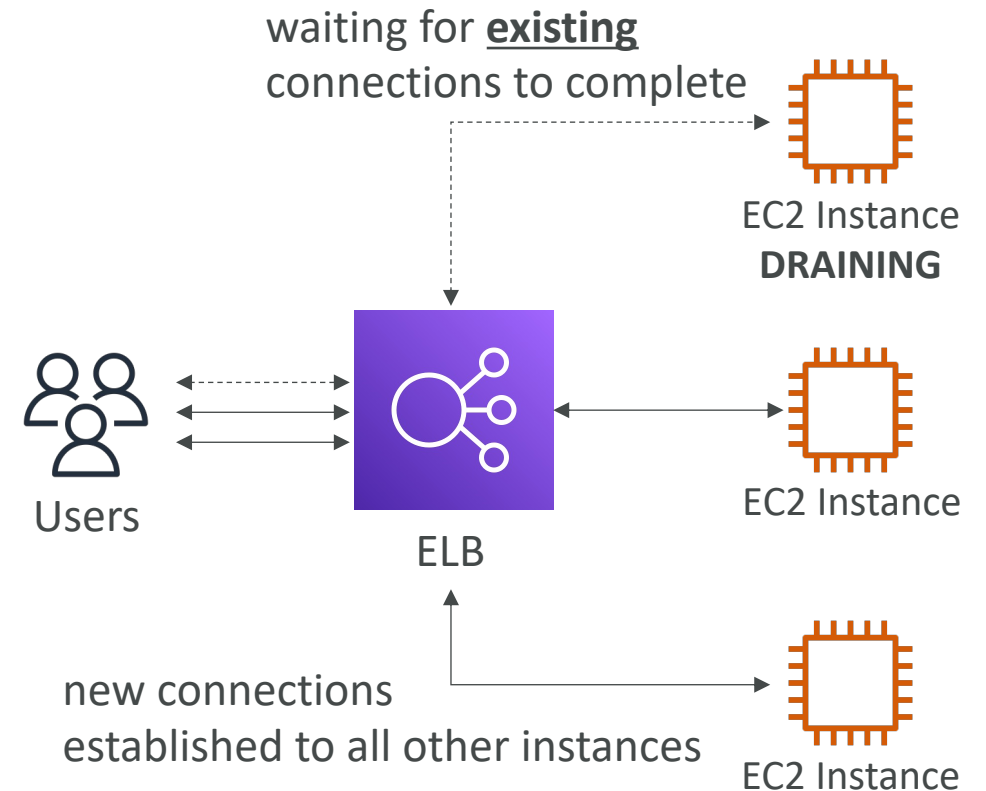


# Elastic Load Balancers – SSL Certificates

- **Classic Load Balancer (v1)**
  - Support only one SSL certificate
  - Must use multiple CLB for multiple hostname with multiple SSL certificates
- **Application Load Balancer (v2)**
  - Supports multiple listeners with multiple SSL certificates
  - Uses Server Name Indication (SNI) to make it work
- **Network Load Balancer (v2)**
  - Supports multiple listeners with multiple SSL certificates
  - Uses Server Name Indication (SNI) to make it work

# Connection Draining

- Feature naming
  - Connection Draining – for CLB
  - Deregistration Delay – for ALB & NLB
- Time to complete “in-flight requests” while the instance is de-registering or unhealthy
- Stops sending new requests to the EC2 instance which is de-registering
- Between 1 to 3600 seconds (default: 300 seconds)
- Can be disabled (set value to 0)
- Set to a low value if your requests are short



# ELB Health Checks

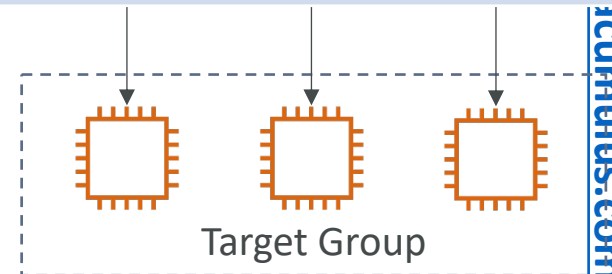
- Target Health Status
  - **Initial:** registering the target
  - **Healthy**
  - **Unhealthy**
  - **Unused:** target is not registered
  - **Draining:** de-registering the target
  - **Unavailable:** health checks disabled
- If a target group contains only unhealthy targets, ELB routes requests across its unhealthy targets

Application Load Balancer



Health Check

Setting	Value	Description
HealthCheckProtocol	HTTP	Protocol used to perform health checks
HealthCheckPort	80	Port used to perform health checks
HealthCheckPath	/	Destination for health checks on targets
HealthCheckTimeoutSeconds	5	Consider the health check failed if no response after 5 seconds
HealthCheckIntervalSeconds	30	Send health check every 30 seconds
HealthyThresholdCount	3	Consider the target healthy after 3 successful health checks
UnhealthyThresholdCount	5	Consider the target unhealthy after 5 failed health checks

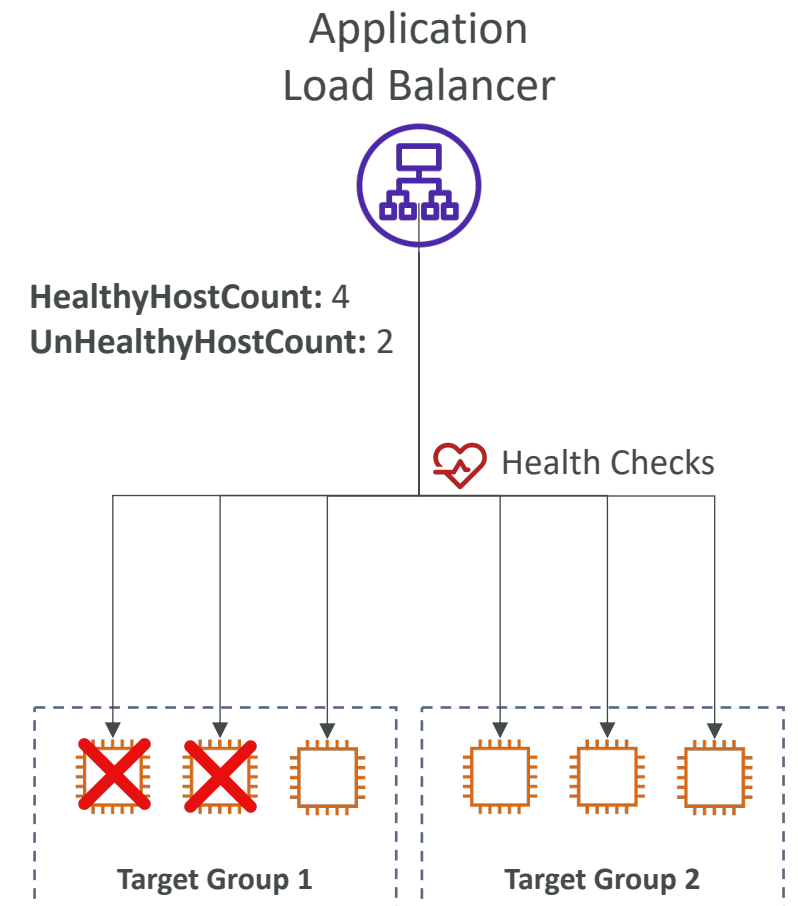


# Load Balancer Error codes

- Successful request : Code 200.
- Unsuccessful at client side : 4XX code.
  - Error 400 : Bad Request
  - Error 401 : Unauthorized
  - Error 403 : Forbidden
  - Error 460 : Client closed connection.
  - Error 463 : X-Forwarded For header with >30 IP (Similar to malformed request).
- Unsuccessful at server side : 5xx code.
  - An error 500 / Internal server error would mean some error on the ELB itself.
  - Error 502 : Bad Gateway
  - **An error 503 / Service Unavailable**
  - Error 504 / Gateway timeout : probably an issue within the server.
  - Error 561 : Unauthorized

# Load Balancers Monitoring

- All Load Balancer metrics are directly pushed to CloudWatch metrics
- BackendConnectionErrors
- HealthyHostCount / UnHealthyHostCount
- HTTPCode\_Backend\_2XX: Successful request.
- HTTPCode\_Backend\_3XX, redirected request
- HTTPCode\_ELB\_4XX: Client error codes
- HTTPCode\_ELB\_5XX: Server error codes generated by the load balancer.
- Latency
- RequestCount
- RequestCountPerTarget
- SurgeQueueLength: The total number of requests (HTTP listener) or connections (TCP listener) that are pending routing to a healthy instance. Help to scale out ASG. Max value is 1024
- SpilloverCount: The total number of requests that were rejected because the surge queue is full.



# Load Balancer troubleshooting using metrics

- **HTTP 400: BAD\_REQUEST** => The client sent a malformed request that does not meet HTTP specifications.
- **HTTP 503: Service Unavailable** => Ensure that you have healthy instances in every Availability Zone that your load balancer is configured to respond in. Look for HealthyHostCount in CloudWatch
- **HTTP 504: Gateway Timeout** => Check if keep-alive settings on your EC2 instances are enabled and make sure that the keep-alive timeout is greater than the idle timeout settings of load balancer.
- Set alarms & look at the documentation for troubleshooting:  
<https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/ts-elb-error-message.html>

# Load Balancers Access Logs

- Access logs from Load Balancers can be stored in S3 and contain:
  - Time
  - Client IP address
  - Latencies
  - Request paths
  - Server response
  - Trace Id
- Only pay for the S3 storage
- Helpful for compliance reason
- Helpful for keeping access data even after ELB or EC2 instances are terminated
- Access Logs are already encrypted

# Application Load Balancer Request Tracing

- Request tracing – Each HTTP request has an added custom header 'X-Amzn-Trace-Id'

- Example:

X-Amzn-Trace-Id: Root=1-67891233-abcdef012345678912345678

- This is useful in logs / distributed tracing platform to track a single request
- Application Load Balancer is not (yet) integrated with X-Ray

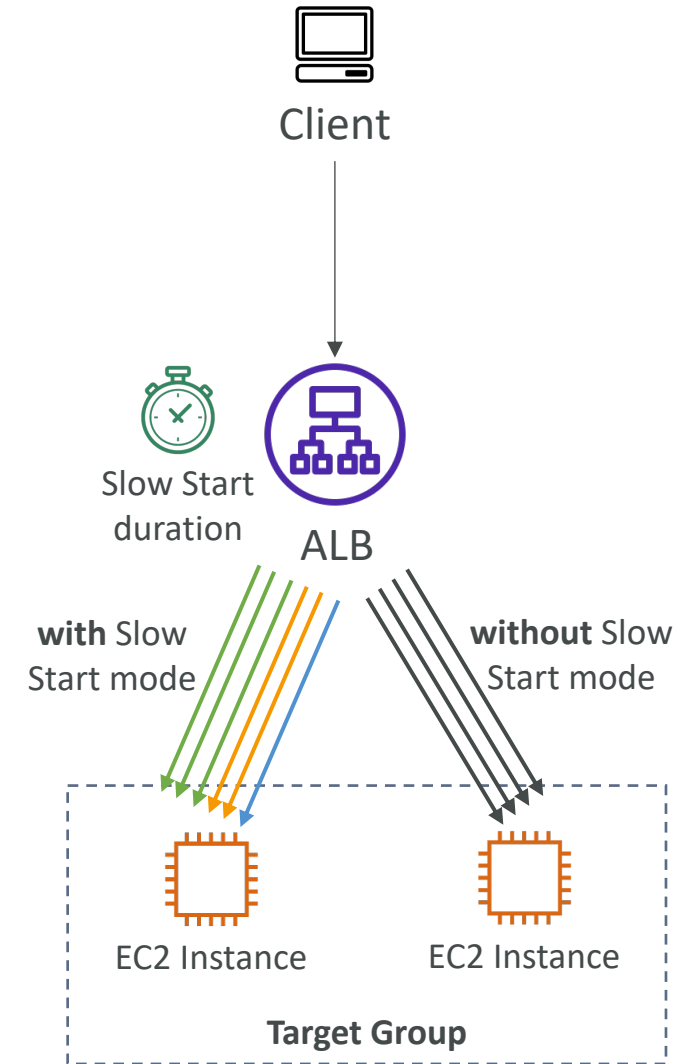


# Target Groups Settings

- `deregistration_delay.timeout_seconds`: time the load balancer waits before deregistering a target
- `slow_start.duration_seconds`: (see next slide)
- `load_balancing.algorithm.type`: how the load balancer selects targets when routing requests (Round Robin, Least Outstanding Requests)
- `stickiness.enabled`
- `stickiness.type`: application-based or duration-based cookie
- `stickiness.app_cookie.cookie_name`: name of the application cookie
- `stickiness.app_cookie.duration_seconds`: application-based cookie expiration period
- `stickiness.lb_cookie.duration_seconds`: duration-based cookie expiration period

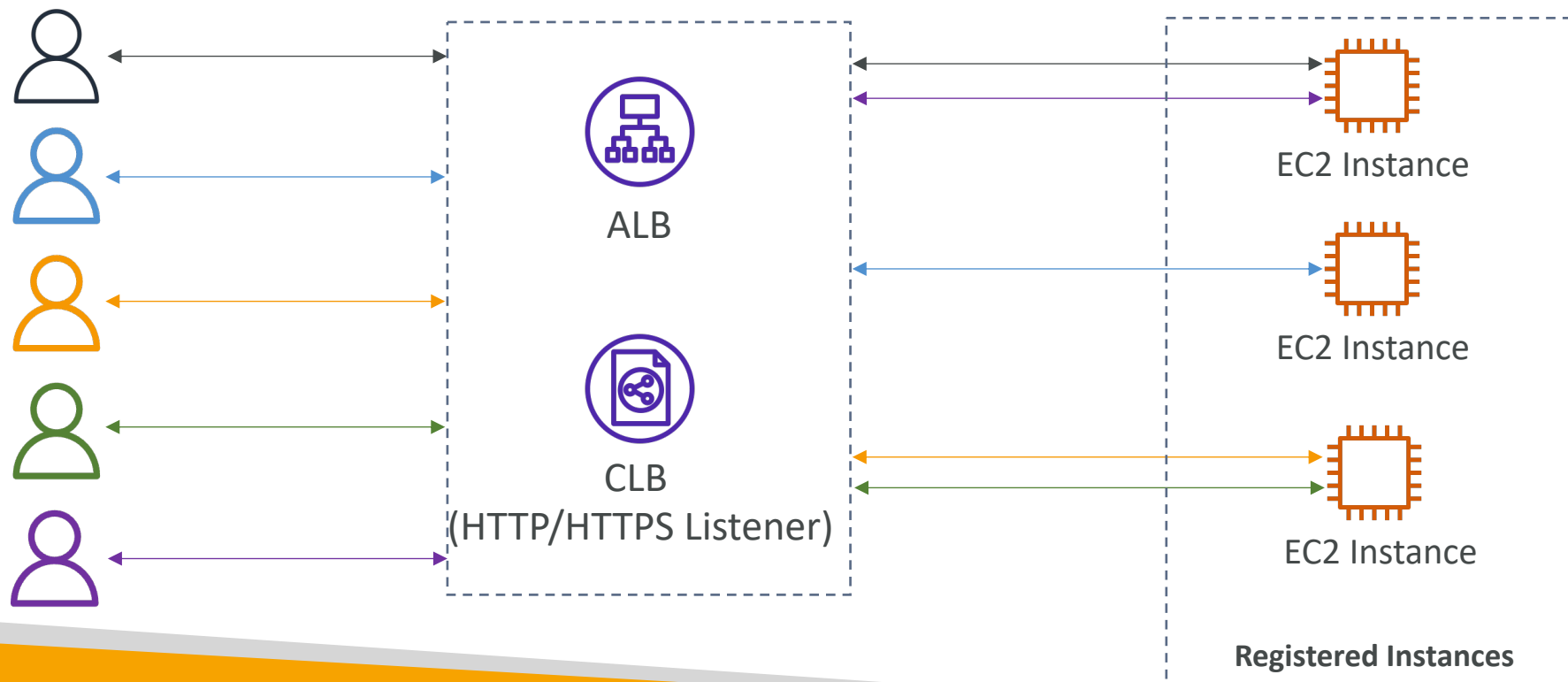
# Slow Start Mode

- By default, a target receives its full share of requests once it's registered with the target group
- **Slow Start Mode** gives healthy targets time to warm-up before the load balancer sends them a full share of requests
- The load balancer linearly increases the number of requests that it sends to the target
- A target exits Slow Start Mode when:
  - The duration period elapses
  - The target becomes unhealthy
- To disable, set **Slow start duration** value to 0



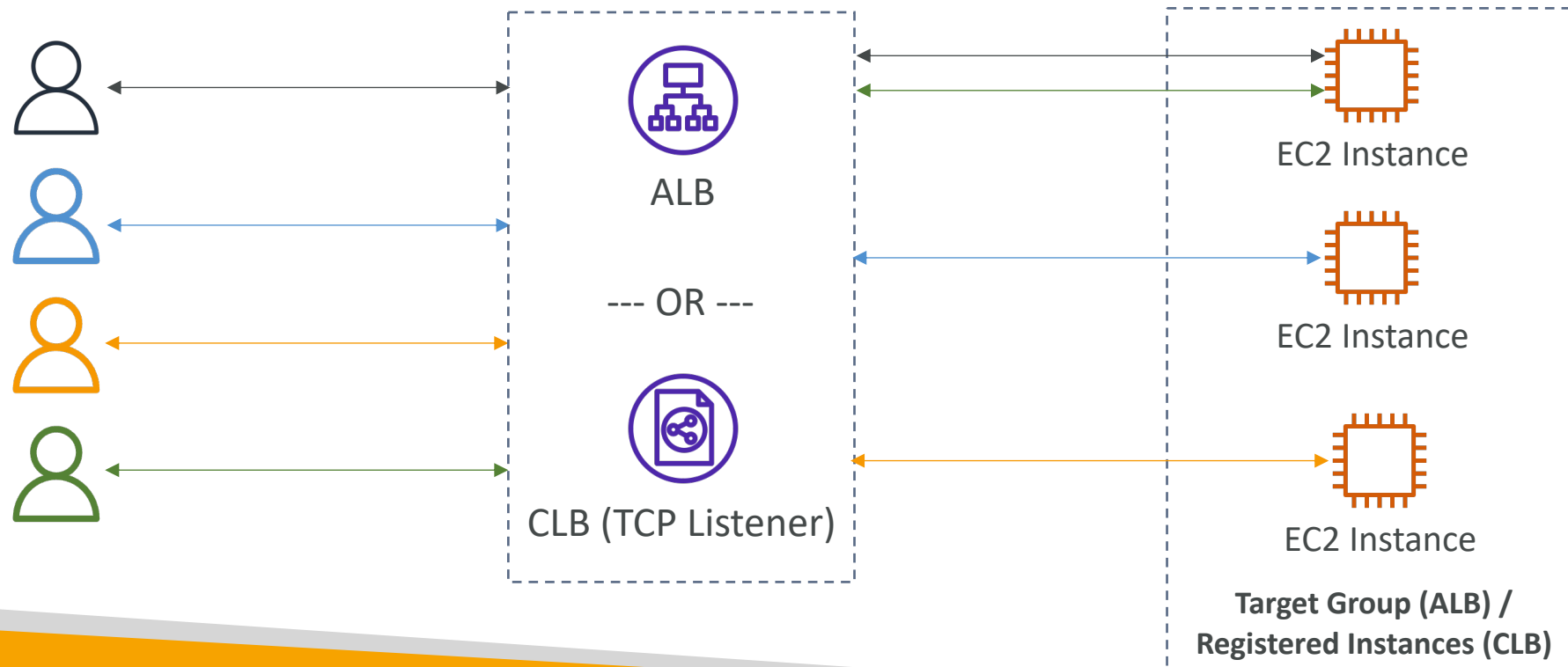
# Request Routing Algorithms – Least Outstanding Requests

- The next instance to receive the request is the instance that has the lowest number of pending/unfinished requests
- Works with **Application Load Balancer** and **Classic Load Balancer (HTTP/HTTPS)**



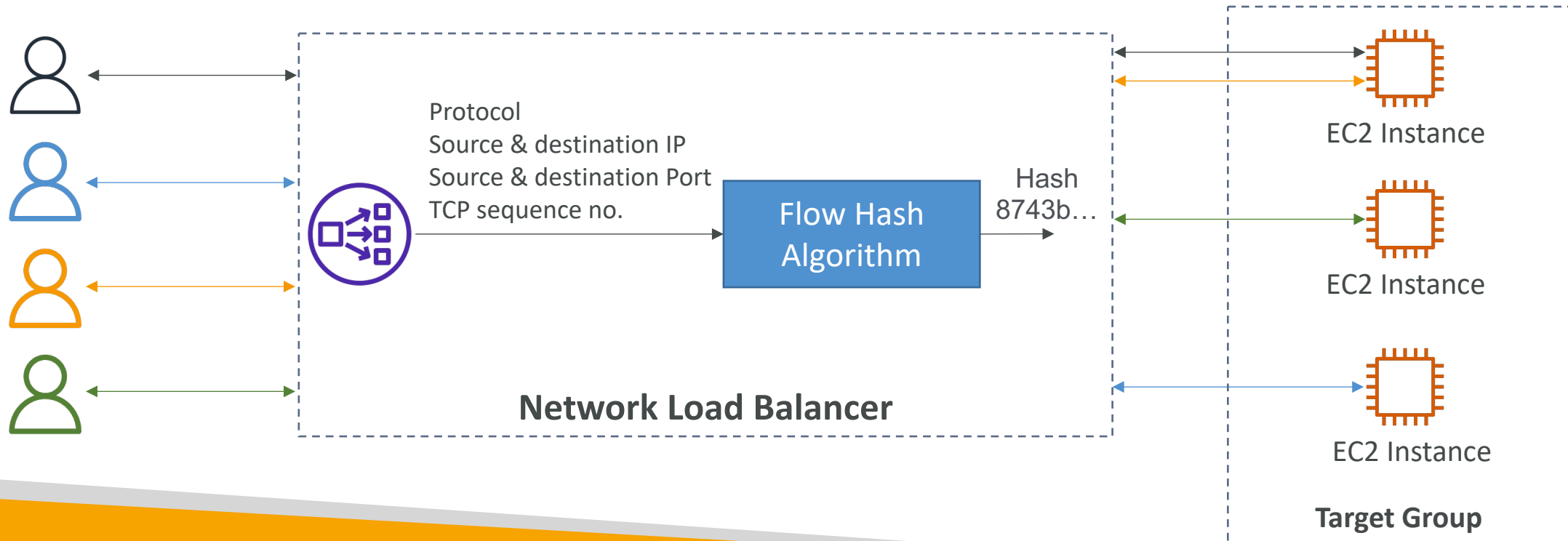
# Request Routing Algorithms – Round Robin

- Equally choose the targets from the target group
- Works with **Application Load Balancer** and **Classic Load Balancer (TCP)**



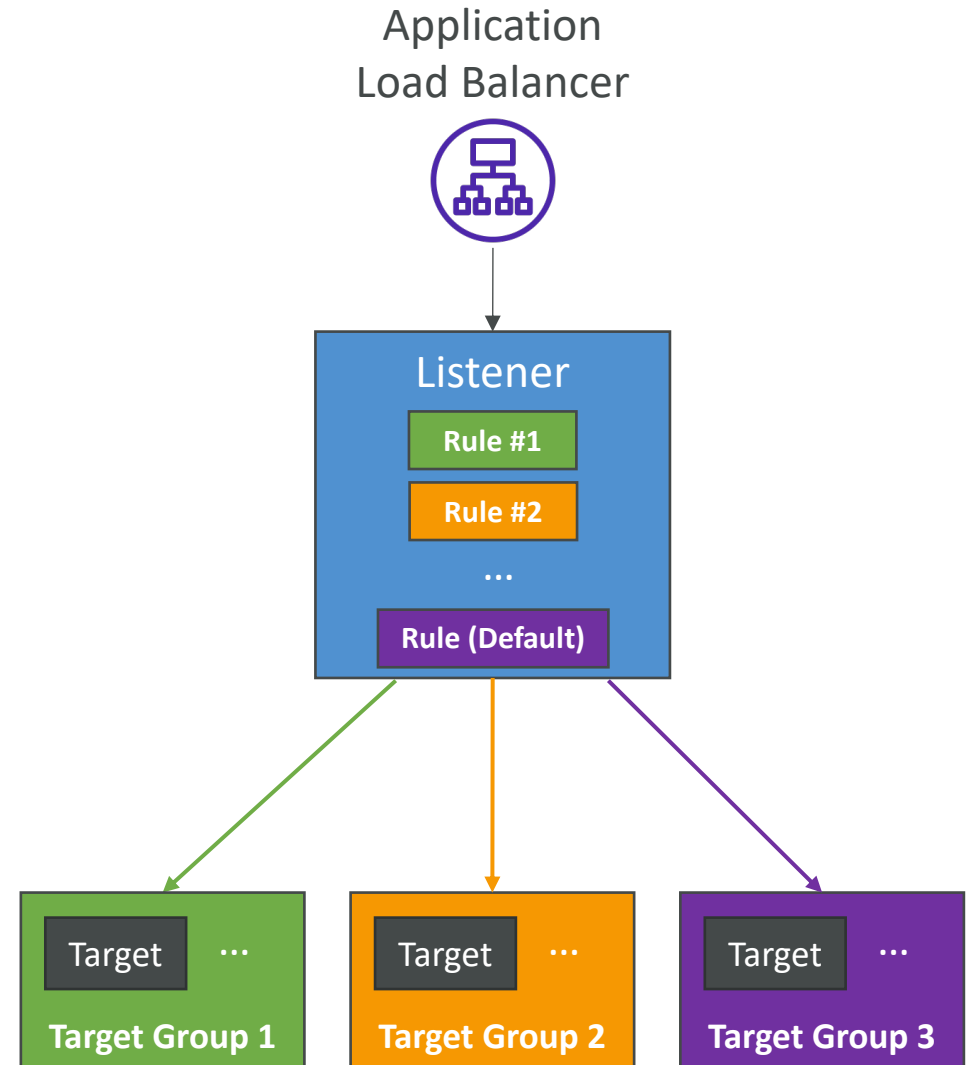
# Request Routing Algorithms – Flow Hash

- Selects a target based on the protocol, source/destination IP address, source/destination port, and TCP sequence number
- Each TCP/UDP connection is routed to a single target for the life of the connection
- Works with **Network Load Balancer**



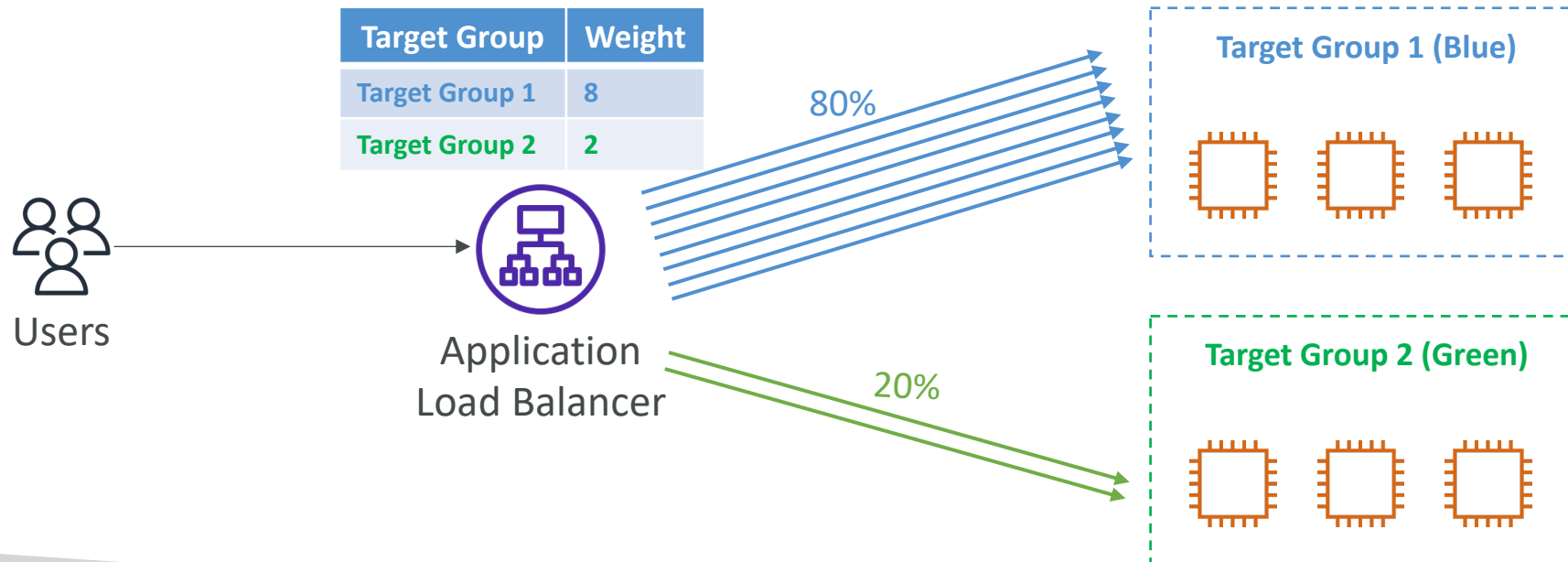
# ALB – Listener Rules

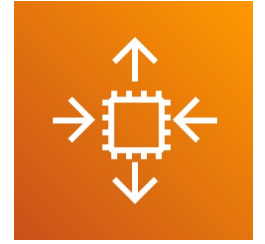
- Processed in order (with Default Rule)
- Supported Actions (forward, redirect, fixed-response)
- Rule Conditions:
  - host-header
  - http-request-method
  - path-pattern
  - source-ip
  - http-header
  - query-string



# Target Group Weighting

- Specify weight for each Target Group on a single Rule
- Example: multiple versions of your app, blue/green deployment
- Allows you to control the distribution of the traffic to your applications



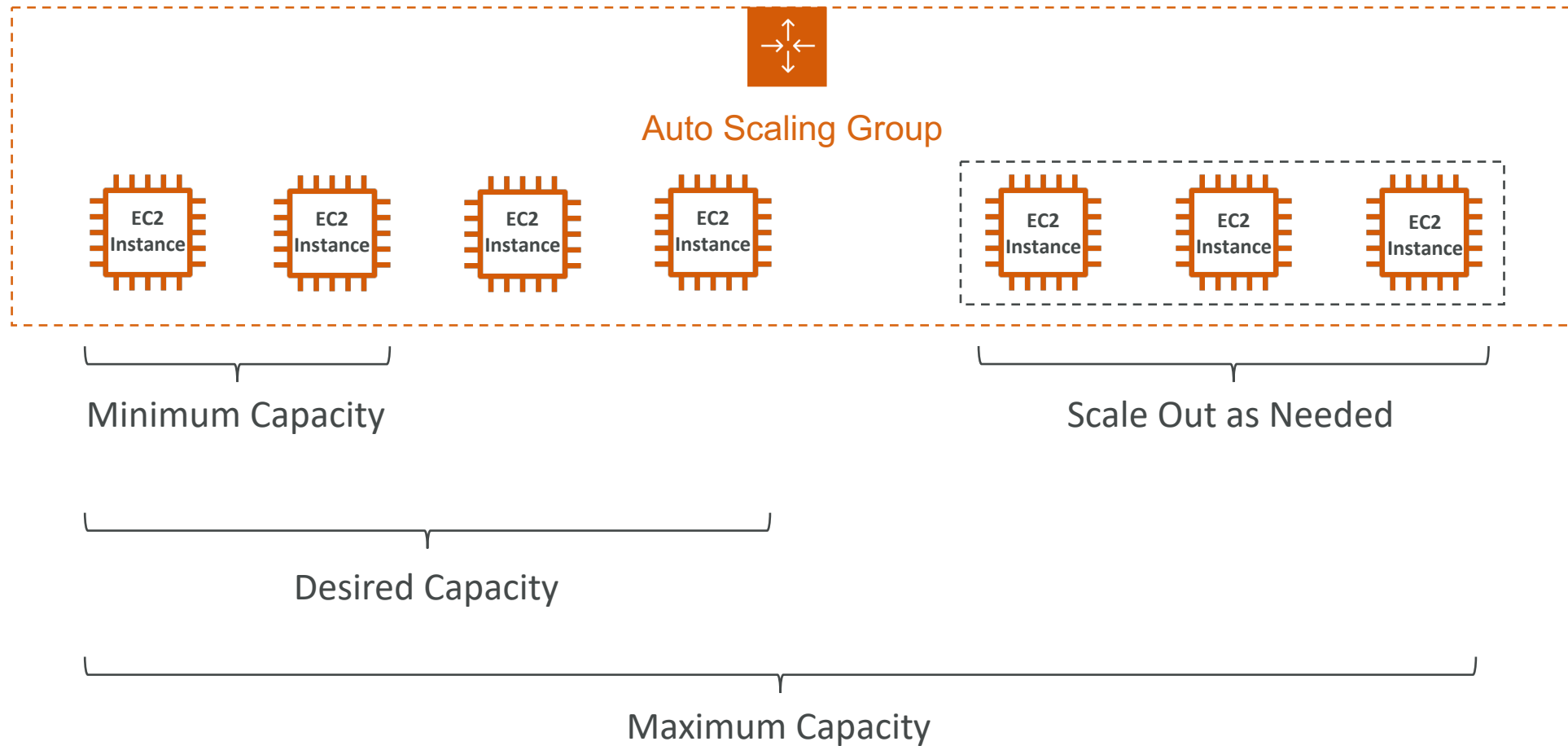


# What's an Auto Scaling Group?

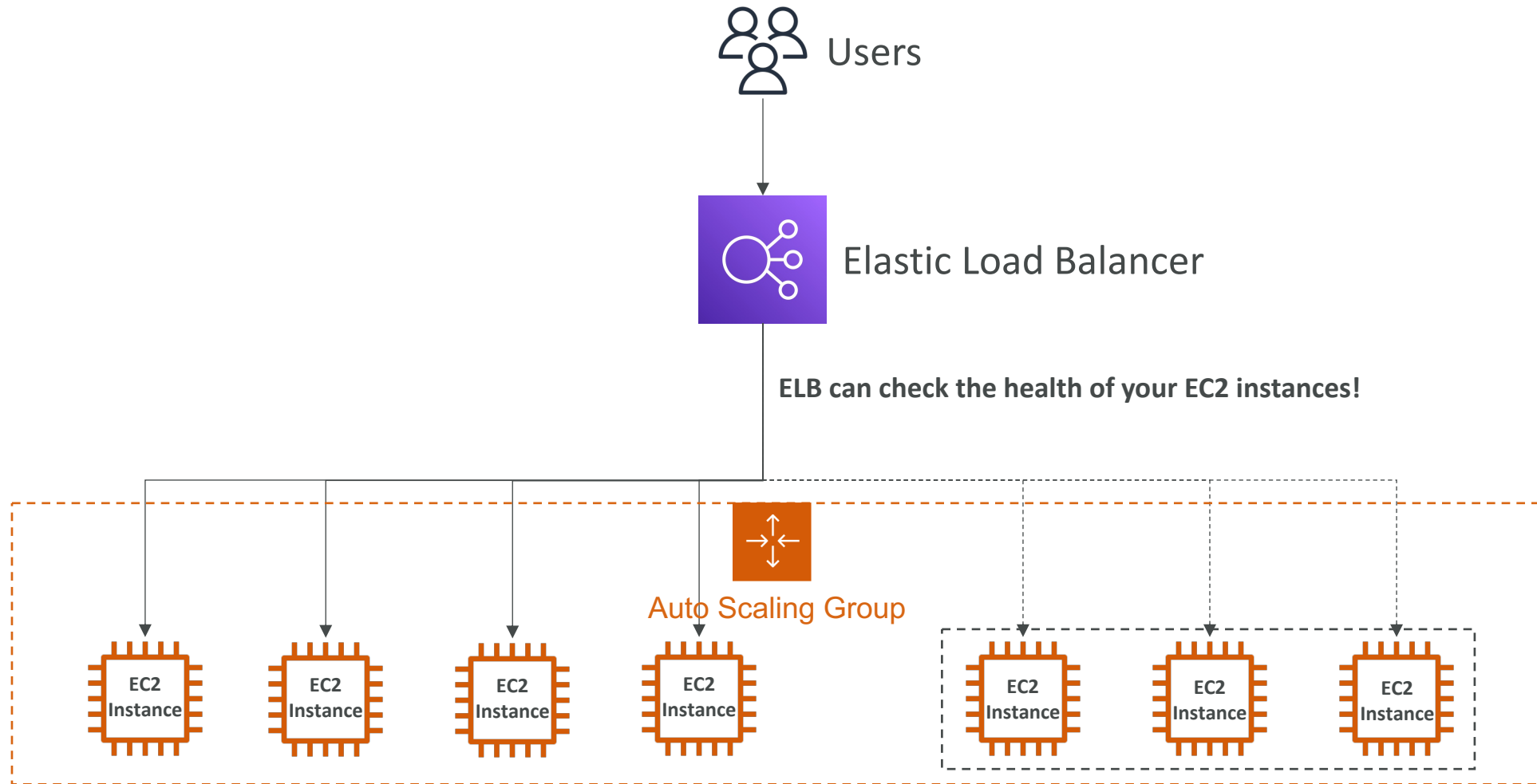
- In real-life, the load on your websites and application can change
- In the cloud, you can create and get rid of servers very quickly
- The goal of an Auto Scaling Group (ASG) is to:
  - Scale out (add EC2 instances) to match an increased load
  - Scale in (remove EC2 instances) to match a decreased load
  - Ensure we have a minimum and a maximum number of EC2 instances running
  - Automatically register new instances to a load balancer
  - Re-create an EC2 instance in case a previous one is terminated (ex: if unhealthy)
- ASG are free (you only pay for the underlying EC2 instances)



# Auto Scaling Group in AWS

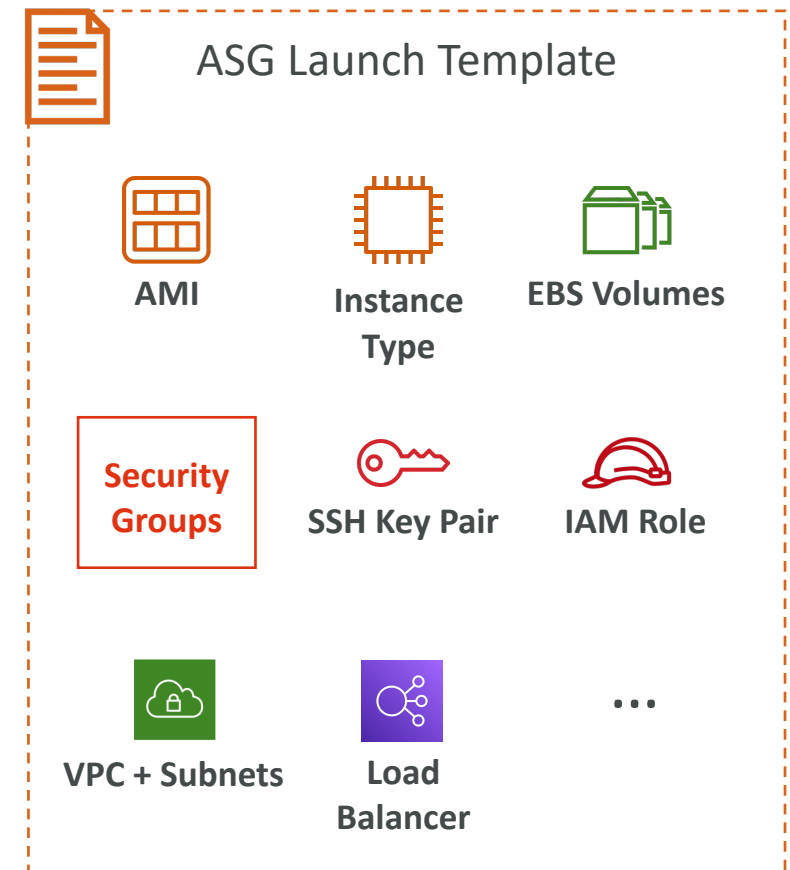


# Auto Scaling Group in AWS With Load Balancer



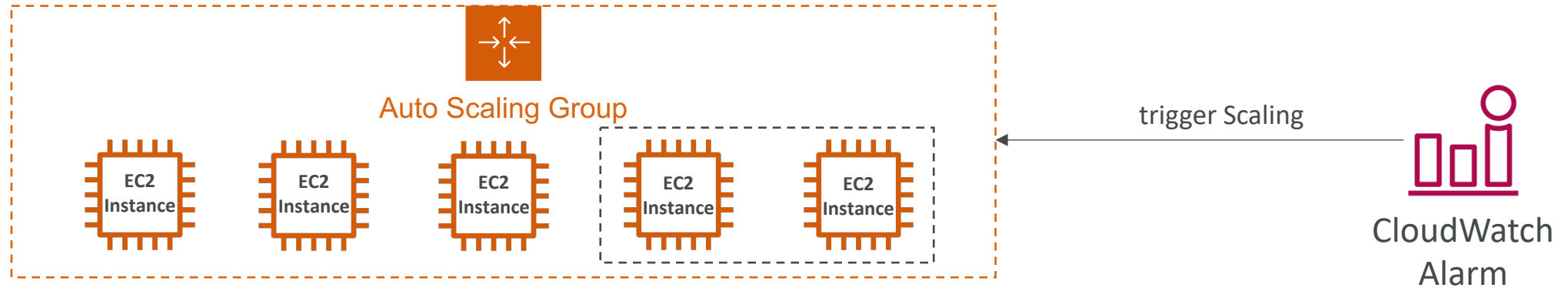
# Auto Scaling Group Attributes

- A **Launch Template** (older “Launch Configurations” are deprecated)
  - AMI + Instance Type
  - EC2 User Data
  - EBS Volumes
  - Security Groups
  - SSH Key Pair
  - IAM Roles for your EC2 Instances
  - Network + Subnets Information
  - Load Balancer Information
- Min Size / Max Size / Initial Capacity
- Scaling Policies



# Auto Scaling - CloudWatch Alarms & Scaling

- It is possible to scale an ASG based on CloudWatch alarms
- An alarm monitors a metric (such as **Average CPU**, or a **custom metric**)
- Metrics such as Average CPU are computed for the overall ASG instances
- Based on the alarm:
  - We can create scale-out policies (increase the number of instances)
  - We can create scale-in policies (decrease the number of instances)

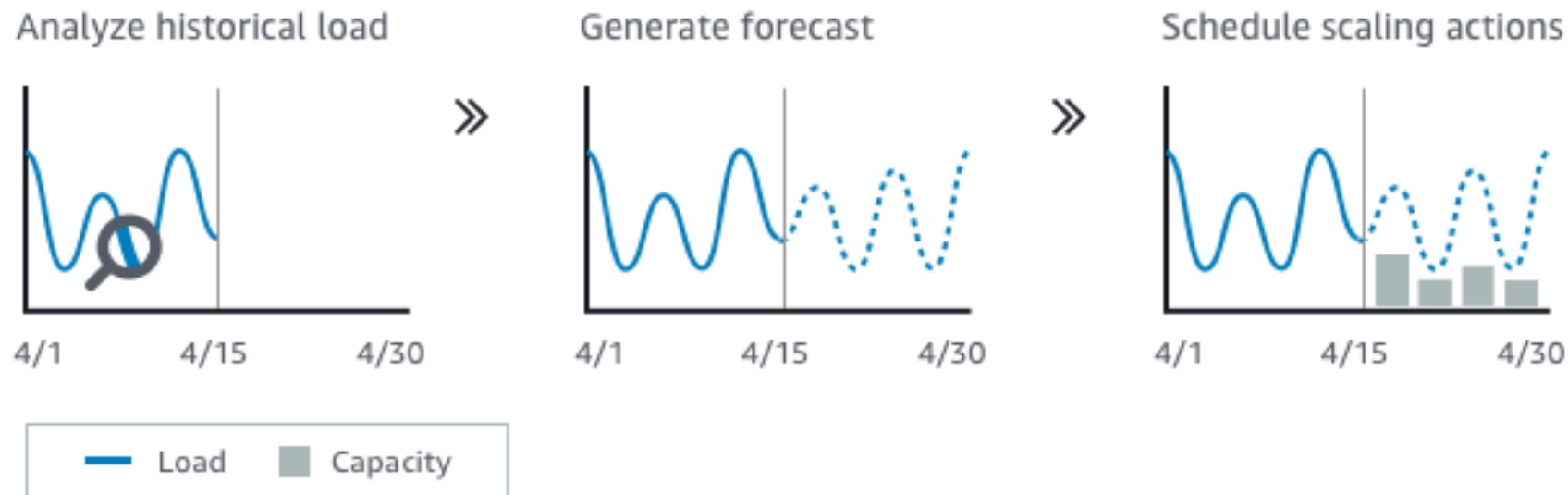


# Auto Scaling Groups – Dynamic Scaling Policies

- **Target Tracking Scaling**
  - Most simple and easy to set-up
  - Example: I want the average ASG CPU to stay at around 40%
- **Simple / Step Scaling**
  - When a CloudWatch alarm is triggered (example CPU > 70%), then add 2 units
  - When a CloudWatch alarm is triggered (example CPU < 30%), then remove 1
- **Scheduled Actions**
  - Anticipate a scaling based on known usage patterns
  - Example: increase the min capacity to 10 at 5 pm on Fridays

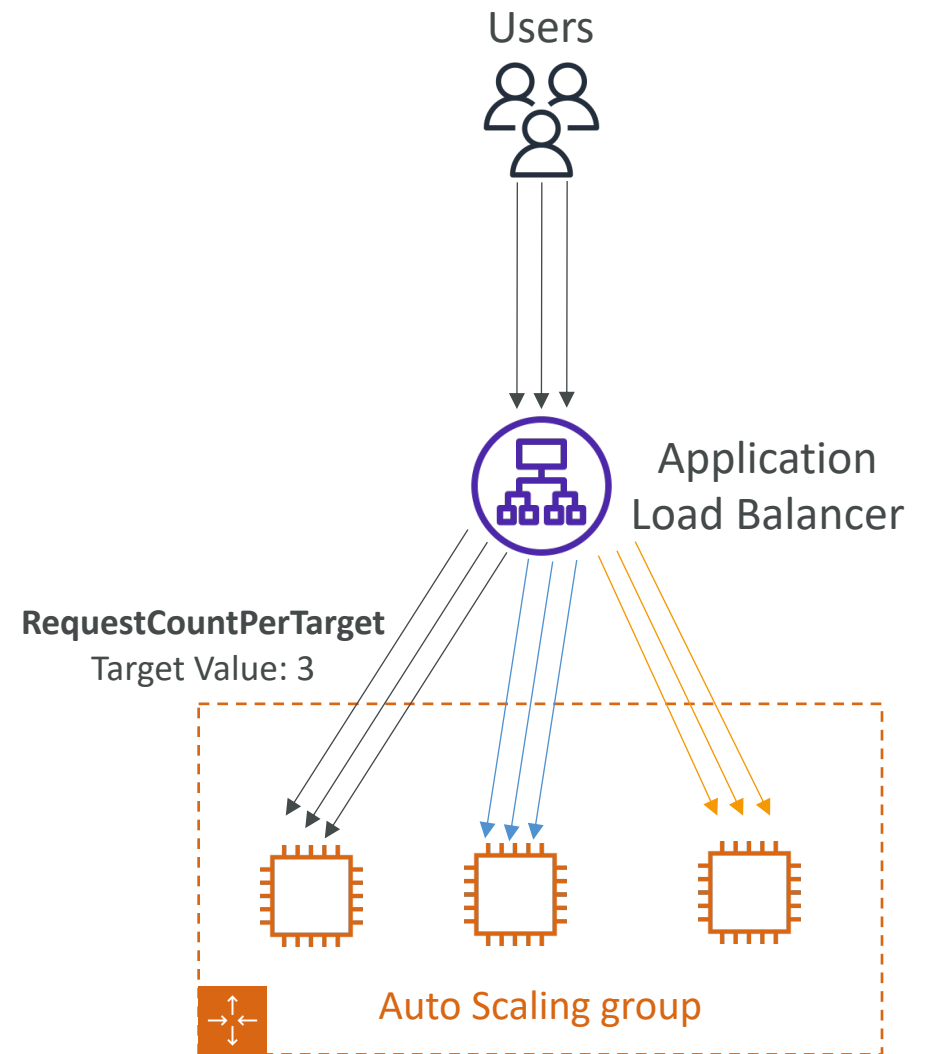
# Auto Scaling Groups – Predictive Scaling

- **Predictive scaling:** continuously forecast load and schedule scaling ahead



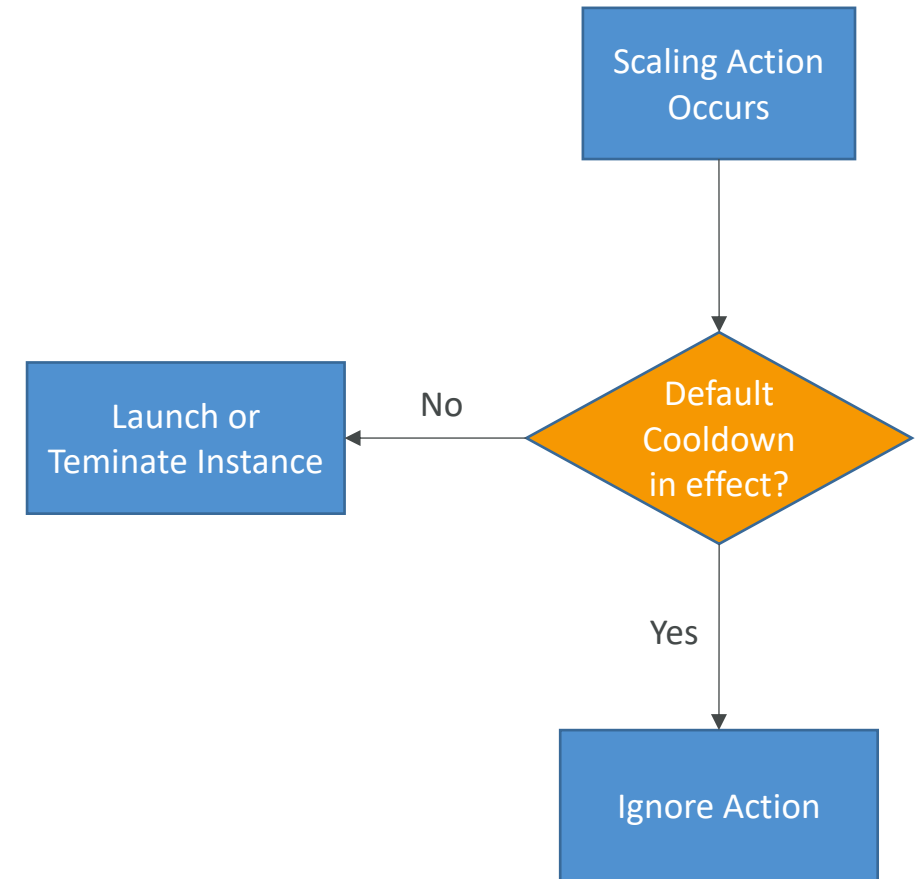
# Good metrics to scale on

- **CPUUtilization:** Average CPU utilization across your instances
- **RequestCountPerTarget:** to make sure the number of requests per EC2 instances is stable
- **Average Network In / Out** (if you're application is network bound)
- **Any custom metric** (that you push using CloudWatch)



# Auto Scaling Groups - Scaling Cooldowns

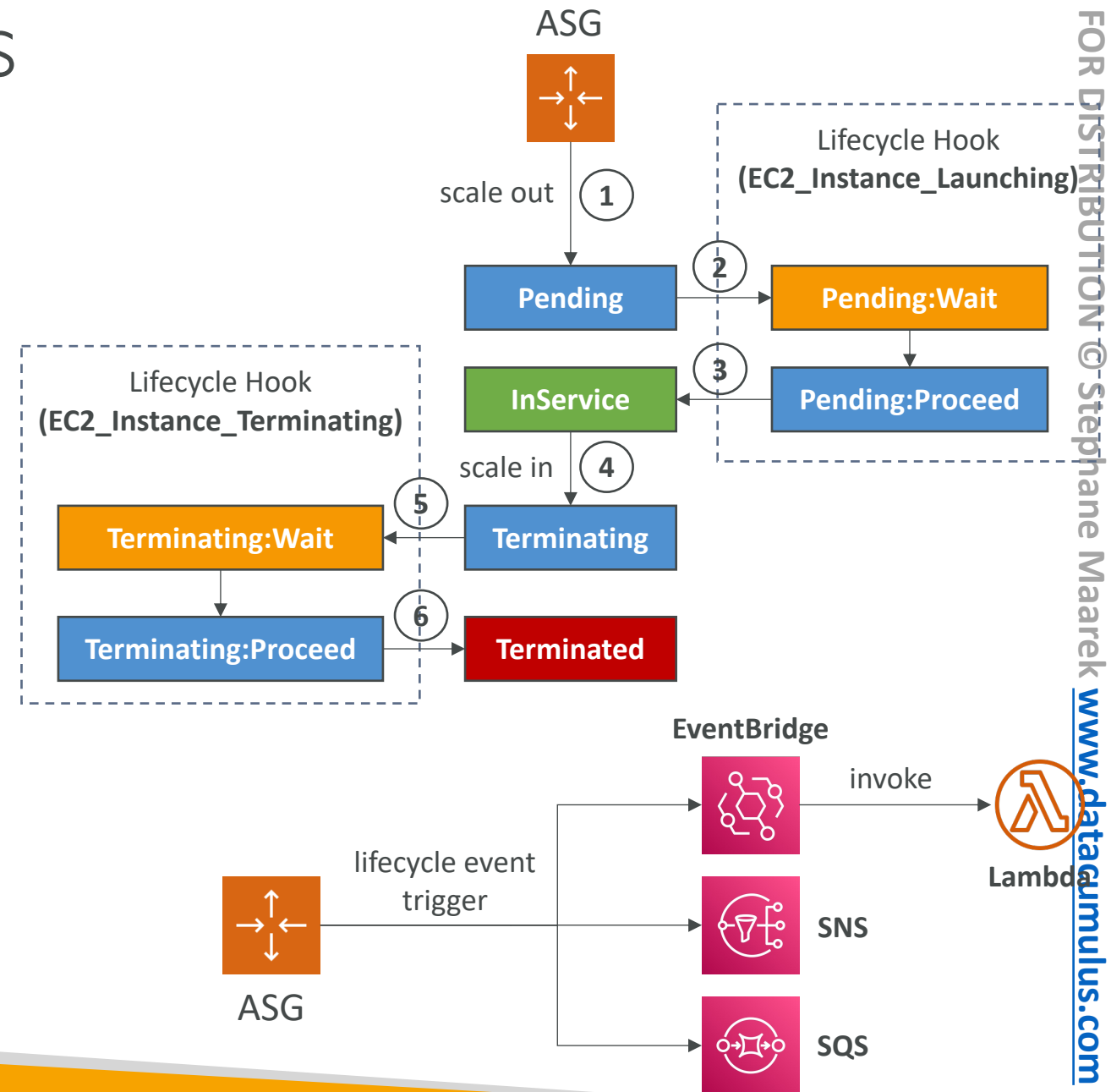
- After a scaling activity happens, you are in the **cooldown period** (default 300 seconds)
- During the cooldown period, the ASG will not launch or terminate additional instances (to allow for metrics to stabilize)
- Advice: Use a ready-to-use AMI to reduce configuration time in order to be serving request faster and reduce the cooldown period





# ASG – Lifecycle Hooks

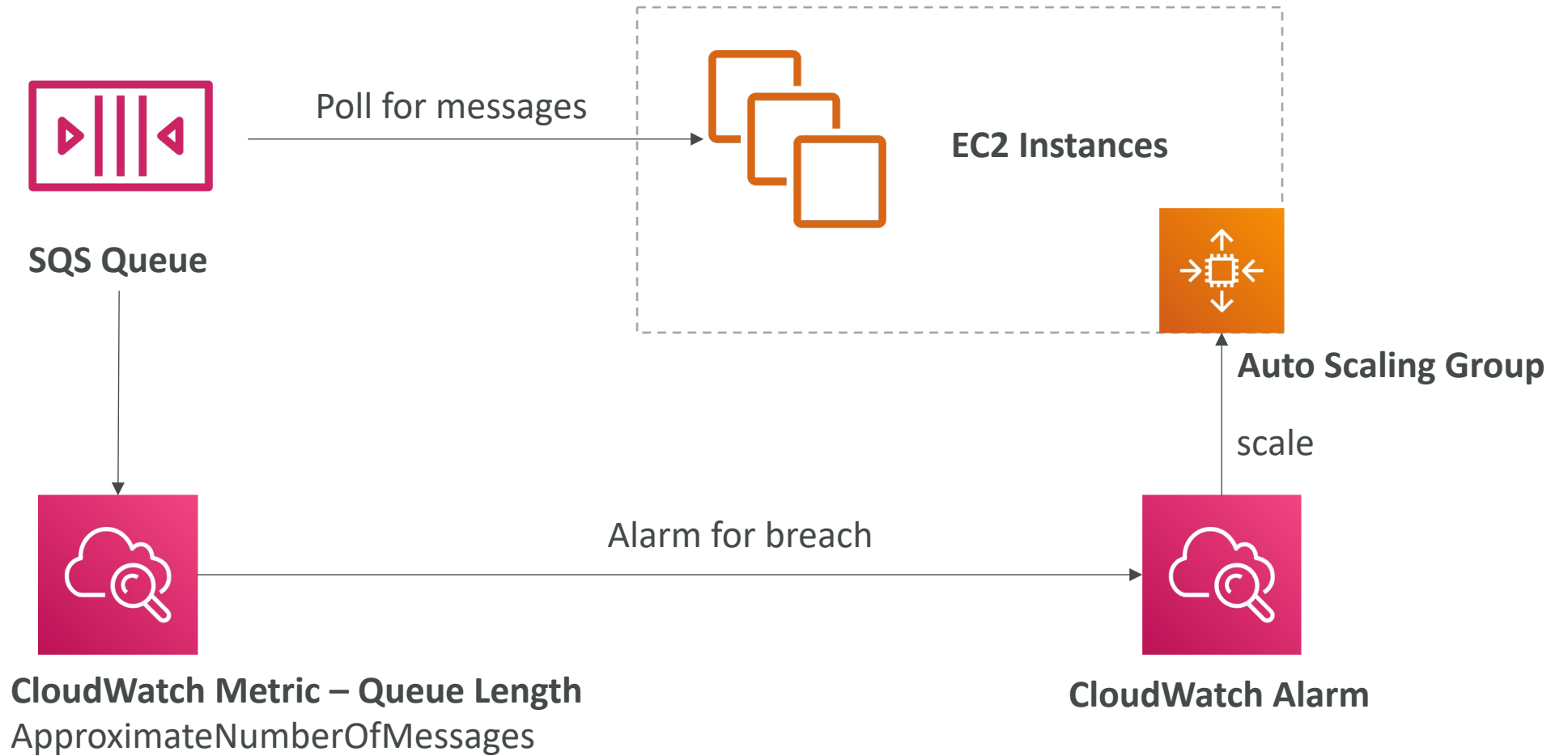
- By default, as soon as an instance is launched in an ASG it's in service
- You can perform extra steps before the instance goes in service (Pending state)
  - Define a script to run on the instances as they start
- You can perform some actions before the instance is terminated (Terminating state)
  - Pause the instances before they're terminated for troubleshooting
- Use cases: cleanup, log extraction, special health checks
- Integration with EventBridge, SNS, and SQS



# Launch Configuration vs. Launch Template

- Both:
  - ID of the Amazon Machine Image (AMI), the instance type, a key pair, security groups, and the other parameters that you use to launch EC2 instances (tags, EC2 user-data...)
  - You can't edit both Launch Configurations and Launch Templates
- Launch Configuration (legacy):
  - Must be re-created every time
- Launch Template (newer):
  - Can have multiple versions
  - Create parameters subsets (partial configuration for re-use and inheritance)
  - Provision using both On-Demand and Spot instances (or a mix)
  - Supports Placement Groups, Capacity Reservations, Dedicated hosts, multiple instance types
  - Can use T2 unlimited burst feature
  - Recommended by AWS going forward

# SQS with Auto Scaling Group (ASG)



# ASG Health Checks

- To make sure you have high availability, means you have least 2 instances running across 2 AZ in your ASG (must configure multi-AZ ASG)
- Health checks available:
  - EC2 Status Checks
  - ELB Health Checks
  - Custom Health Checks: send instance's health to ASG using AWS CLI or AWS SDK
- ASG will launch a new instance after terminating an unhealthy one
- ASG will not reboot unhealthy hosts for you
- Good to know CLI:
  - set-instance-health (use with Custom Health Checks)
  - terminate-instance-in-auto-scaling-group

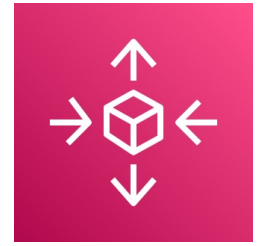
# Troubleshooting ASG issues

- <number of instances> instance(s) are already running. Launching EC2 instance failed.
  - The Auto Scaling group has reached the limit set by the MaximumCapacity parameter. Update your Auto Scaling group by providing a new value for the maximum capacity.
- Launching EC2 instances is failing:
  - The security group does not exist. SG might have been deleted
  - The key pair does not exist. The key pair might have been deleted
- If the ASG fails to launch an instance for over 24 hours, it will automatically suspend the processes (administration suspension)

# CloudWatch Metrics for ASG

- Metrics are collected every 1 minute
- ASG-level metrics: (opt-in)
  - GroupMinSize, GroupMaxSize, GroupDesiredCapacity
  - GroupInServiceInstances, GroupPendingInstances, GroupStandbyInstances
  - GroupTerminatingInstances, GroupTotalInstances
  - You should enable metric collection to see these metrics
- EC2-level metrics (enabled): CPU Utilization, etc...
  - Basic monitoring: 5 minutes granularity
  - Detailed monitoring: 1 minute granularity

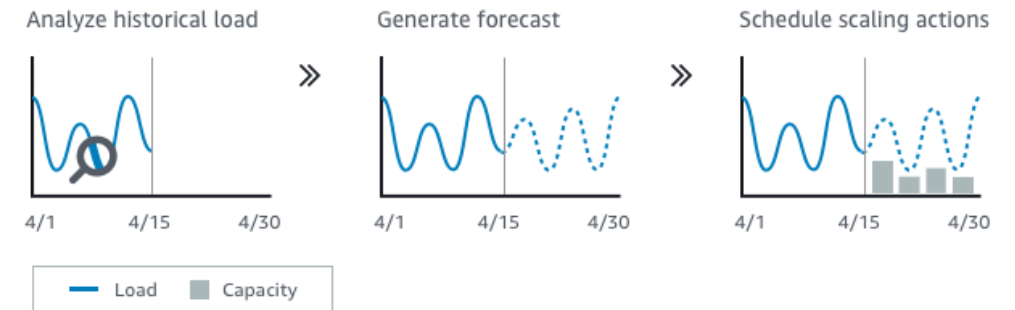
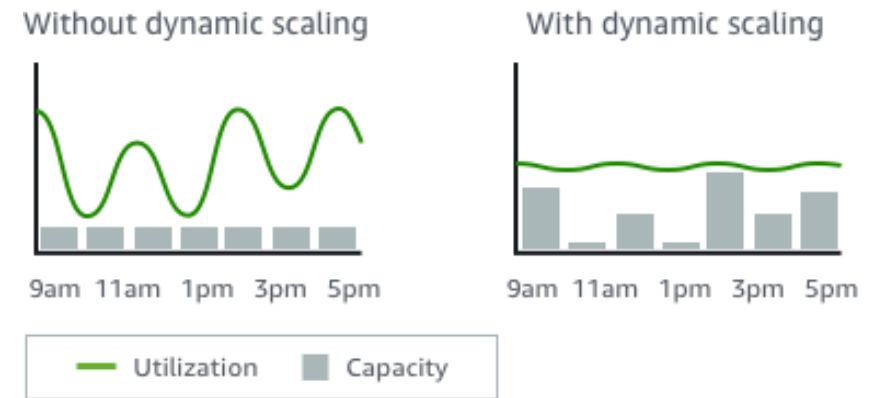
# AWS Auto Scaling



- Backbone service of auto scaling for scalable resources in AWS:
- **Amazon EC2 Auto Scaling groups:** Launch or terminate EC2 instances
- **Amazon EC2 Spot Fleet requests:** Launch or terminate instances from a Spot Fleet request, or automatically replace instances that get interrupted for price or capacity reasons.
- **Amazon ECS:** Adjust the ECS service desired count up or down
- **Amazon DynamoDB (table or global secondary index):** WCU & RCU
- **Amazon Aurora:** Dynamic Read Replicas Auto Scaling

# AWS Auto Scaling – Scaling Plans

- **Dynamic scaling:** creates a target tracking scaling policy
  - Optimize for availability => 40% of resource utilization
  - Balance availability and cost => 50% of resource utilization
  - Optimize for cost => 70% of resource utilization
  - Custom => choose own metric and target value
  - Options: Disable scale-in, cooldown period, warmup time (for ASG)
- **Predictive scaling:** continuously forecast load and schedule scaling ahead

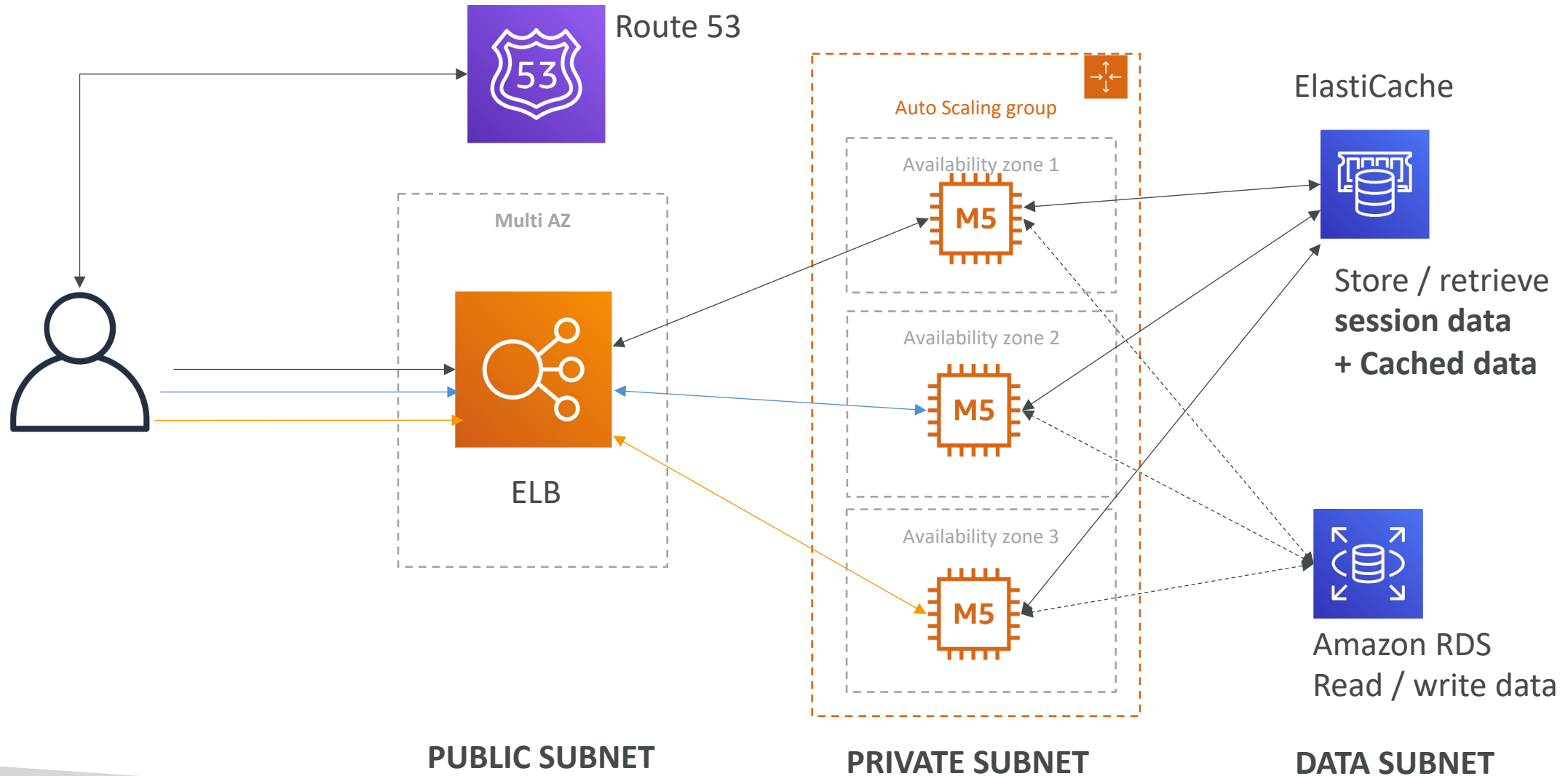


<https://docs.aws.amazon.com/autoscaling/plans/userguide/how-it-works.html>



# Elastic Beanstalk

# Typical architecture: Web App 3-tier



# Developer problems on AWS

- Managing infrastructure
  - Deploying Code
  - Configuring all the databases, load balancers, etc
  - Scaling concerns
- 
- Most web apps have the same architecture (ALB + ASG)
  - All the developers want is for their code to run!
  - Possibly, consistently across different applications and environments

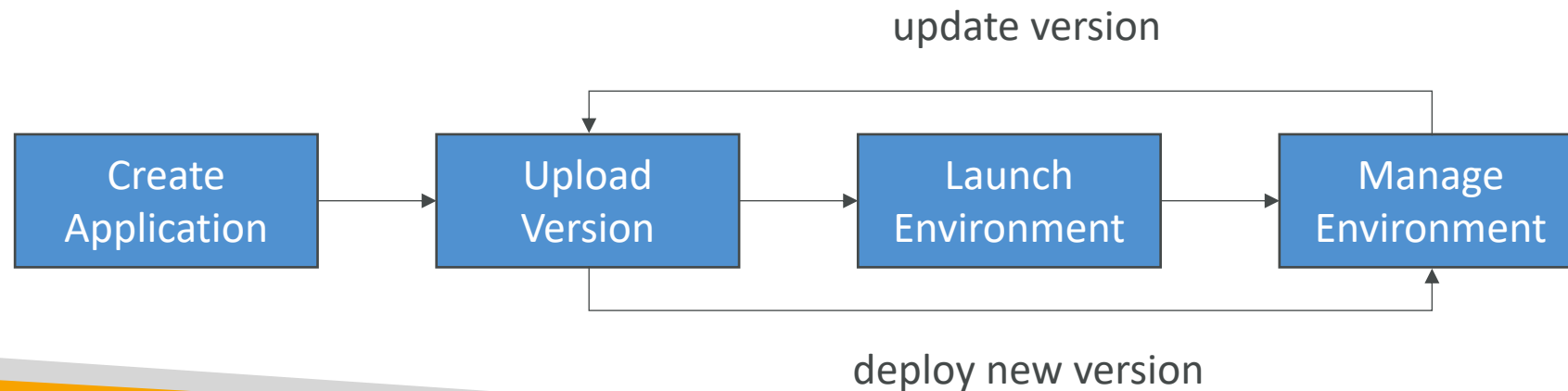
# Elastic Beanstalk – Overview



- Elastic Beanstalk is a developer centric view of deploying an application on AWS
- It uses all the component's we've seen before: EC2, ASG, ELB, RDS, ...
- Managed service
  - Automatically handles capacity provisioning, load balancing, scaling, application health monitoring, instance configuration, ...
  - Just the application code is the responsibility of the developer
- We still have full control over the configuration
- Beanstalk is free but you pay for the underlying instances

# Elastic Beanstalk – Components

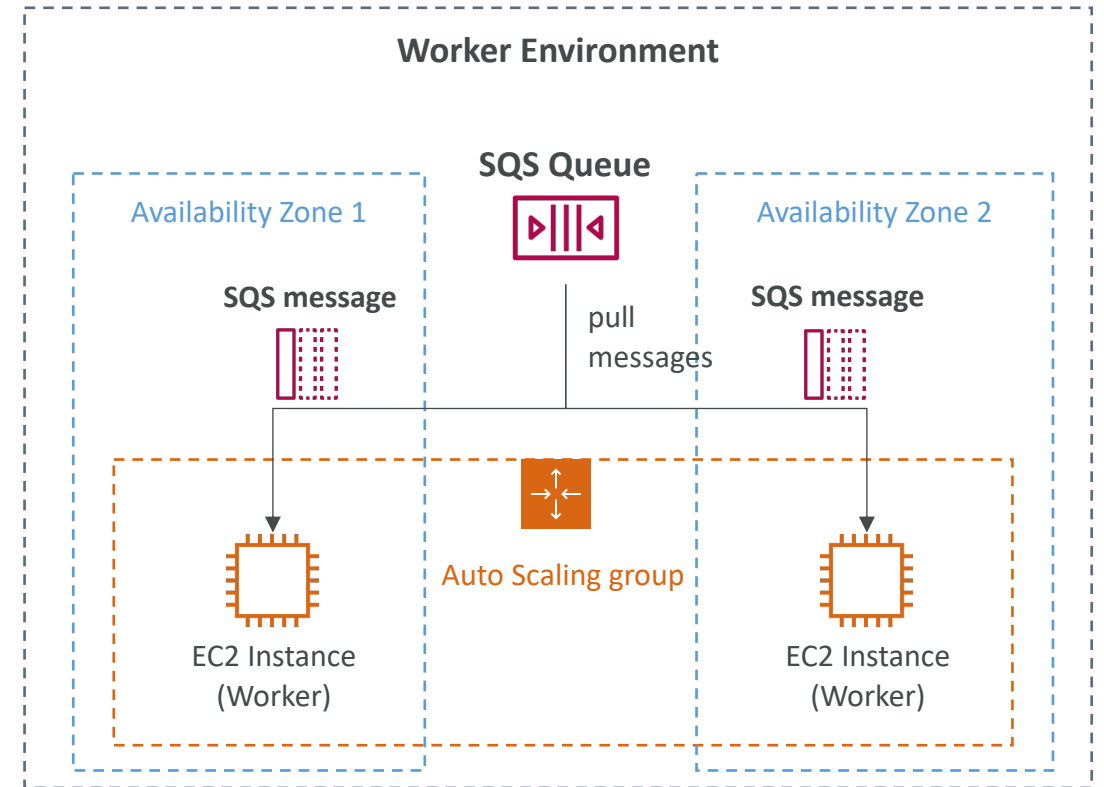
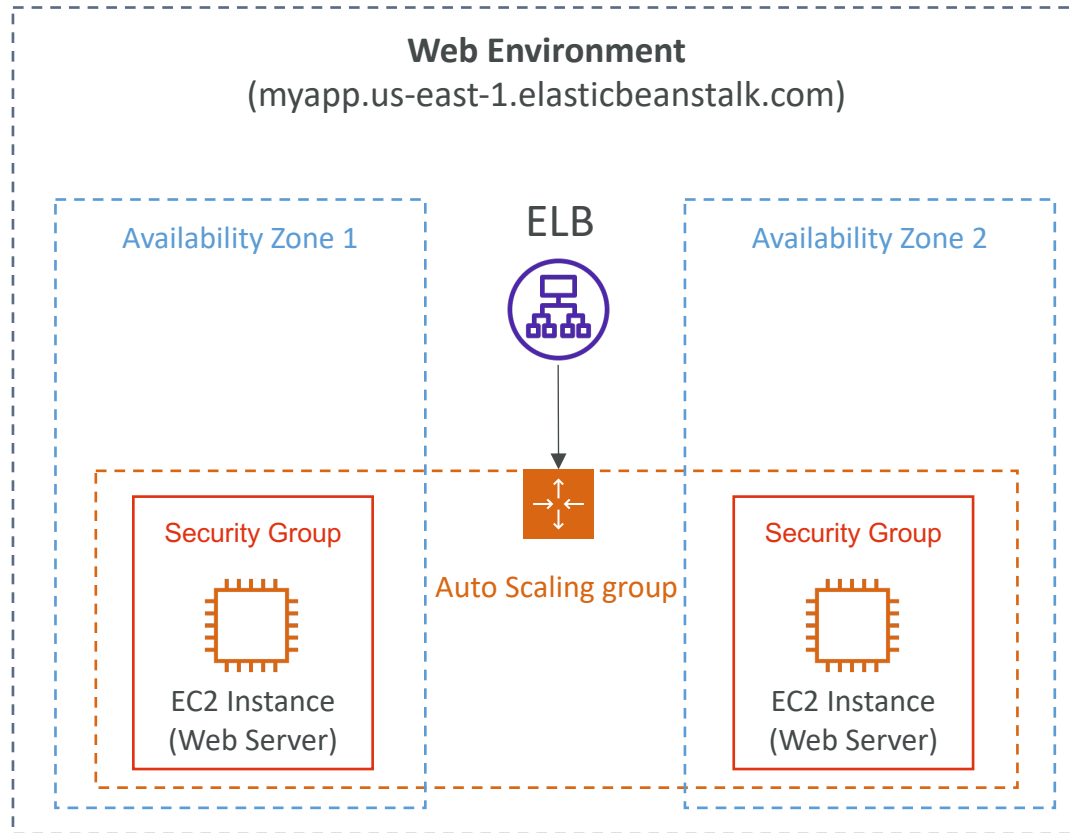
- **Application:** collection of Elastic Beanstalk components (environments, versions, configurations, ...)
- **Application Version:** an iteration of your application code
- **Environment**
  - Collection of AWS resources running an application version (only one application version at a time)
  - **Tiers:** Web Server Environment Tier & Worker Environment Tier
  - You can create multiple environments (dev, test, prod, ...)



# Elastic Beanstalk – Supported Platforms

- Go
- Java SE
- Java with Tomcat
- .NET Core on Linux
- .NET on Windows Server
- Node.js
- PHP
- Python
- Ruby
- Packer Builder
- Single Container Docker
- Multi-container Docker
- Preconfigured Docker
- If not supported, you can write your custom platform (advanced)

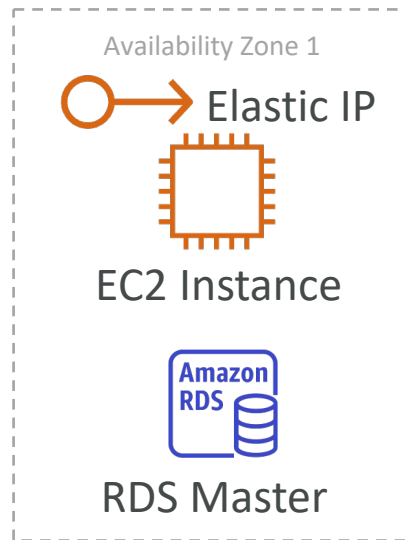
# Web Server Tier vs. Worker Tier



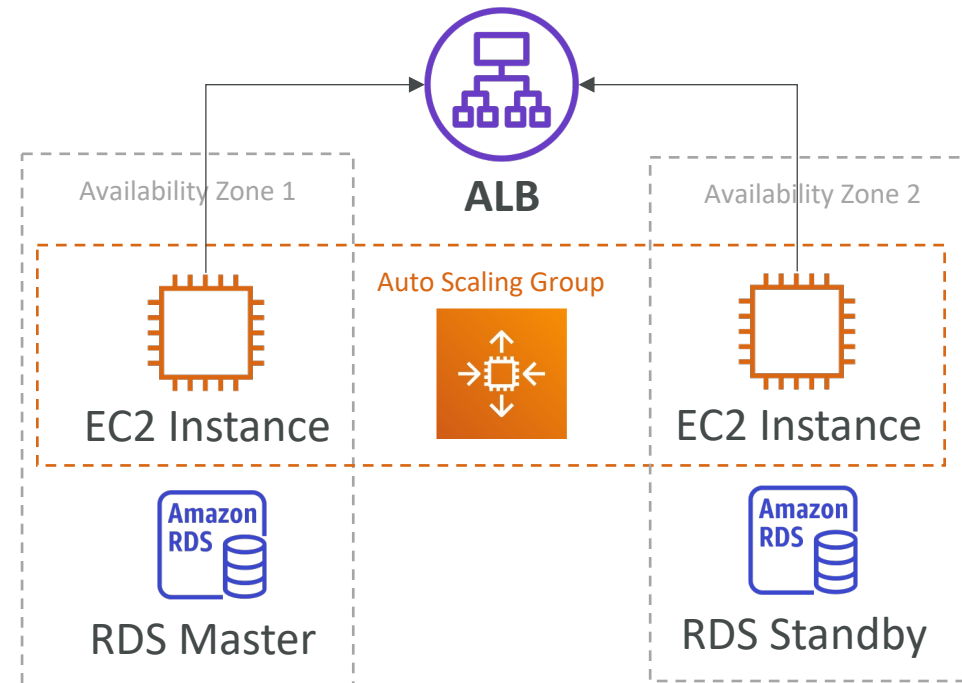
- Scale based on the number of SQS messages
- Can push messages to SQS queue from another Web Server Tier

# Elastic Beanstalk Deployment Modes

**Single Instance**  
Great for dev



**High Availability with Load Balancer**  
Great for prod





# AWS CloudFormation

Managing your infrastructure as code

# Infrastructure as Code

- Currently, we have been doing a lot of manual work
- All this manual work will be very tough to reproduce:
  - In another region
  - in another AWS account
  - Within the same region if everything was deleted
- Wouldn't it be great, if all our infrastructure was... code?
- That code would be deployed and create / update / delete our infrastructure

# What is CloudFormation



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

# Benefits of AWS CloudFormation (1/2)

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

# Benefits of AWS CloudFormation (2/2)

- Productivity
  - Ability to destroy and re-create an infrastructure on the cloud on the fly
  - Automated generation of Diagram for your templates!
  - Declarative programming (no need to figure out ordering and orchestration)
- Separation of concern: create many stacks for many apps, and many layers. Ex:
  - VPC stacks
  - Network stacks
  - App stacks
- Don't re-invent the wheel
  - Leverage existing templates on the web!
  - Leverage the documentation

# How CloudFormation Works

- Templates have to be uploaded in S3 and then referenced in CloudFormation
- To update a template, we can't edit previous ones. We have to re-upload a new version of the template to AWS
- Stacks are identified by a name
- Deleting a stack deletes every single artifact that was created by CloudFormation.

# Deploying CloudFormation templates

- Manual way:
  - Editing templates in the CloudFormation Designer
  - Using the console to input parameters, etc
- Automated way:
  - Editing templates in a YAML file
  - Using the AWS CLI (Command Line Interface) to deploy the templates
  - Recommended way when you fully want to automate your flow

# CloudFormation Building Blocks

Templates components (one course section for each):

1. **Resources:** your AWS resources declared in the template (MANDATORY)
2. Parameters: the dynamic inputs for your template
3. Mappings: the static variables for your template
4. Outputs: References to what has been created
5. Conditionals: List of conditions to perform resource creation
6. Metadata

Templates helpers:

1. References
2. Functions



# Note:

## This is an introduction to CloudFormation

- It can take over 3 hours to properly learn and master CloudFormation
- This section is meant so you get a good idea of how it works
- We'll be slightly less hands-on than in other sections
  
- We'll learn everything we need to answer questions for the exam
- The exam does not require you to actually write CloudFormation
- The exam expects you to understand how to read CloudFormation

# Introductory Example

- We're going to create a simple EC2 instance.
  - Then we're going to create to add an Elastic IP to it
  - And we're going to add two security groups to it
  - For now, forget about the code syntax.
  - We'll look at the structure of the files later on
- 
- We'll see how in no-time, we are able to get started with CloudFormation!



Amazon EC2

# YAML Crash Course

```
1  invoice:      34843
2  date   :      2001-01-23
3  bill-to:
4    given  :      Chris
5    family :      Dumars
6    address:
7      lines: |
8          458 Walkman Dr.
9          Suite #292
10     city   :      Royal Oak
11     state  :      MI
12     postal :      48046
13  product:
14    - sku    :      BL394D
15      quantity :      4
16      description :      Basketball
17      price    :      450.00
18    - sku    :      BL4438H
19      quantity :      1
20      description :      Super Hoop
21      price    :      2392.00
```

- YAML and JSON are the languages you can use for CloudFormation.
- JSON is horrible for CF
- YAML is great in so many ways
- Let's learn a bit about it!
  
- Key value Pairs
- Nested objects
- Support Arrays
- Multi line strings
- Can include comments!

# What are resources?

- Resources are the core of your CloudFormation template (MANDATORY)
- They represent the different AWS Components that will be created and configured
- Resources are declared and can reference each other
  
- AWS figures out creation, updates and deletes of resources for us
- There are over 224 types of resources (!)
- Resource types identifiers are of the form:

**AWS :: aws-product-name :: data-type-name**

# How do I find resources documentation?

- I can't teach you all of the 224 resources, but I can teach you how to learn how to use them.
- All the resources can be found here:  
<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-template-resource-type-ref.html>
- Then, we just read the docs 😊
- Example here (for an EC2 instance):  
<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-instance.html>

# Analysis of CloudFormation Template

- Going back to the example of the introductory section, let's learn why it was written this way.
- Relevant documentation can be found here:
  - <http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-instance.html>
  - <http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-security-group.html>
  - <http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-eip.html>

# FAQ for resources

- Can I create a dynamic amount of resources?
  - No, you can't. Everything in the CloudFormation template has to be declared. You can't perform code generation there
- Is every AWS Service supported?
  - Almost. Only a select few niches are not there yet
  - You can work around that using AWS Lambda Custom Resources

# What are parameters?

- Parameters are a way to provide inputs to your AWS CloudFormation template
- They're important to know about if:
  - You want to reuse your templates across the company
  - Some inputs can not be determined ahead of time
- Parameters are extremely powerful, controlled, and can prevent errors from happening in your templates thanks to types.



# When should you use a parameter?

- Ask yourself this:
  - Is this CloudFormation resource configuration likely to change in the future?
  - If so, make it a parameter.
- You won't have to re-upload a template to change its content 😊

```
Parameters:
```

```
  SecurityGroupDescription:
```

```
    Description: Security Group Description  
    (Simple parameter)
```

```
    Type: String
```

# Parameters Settings

Parameters can be controlled by all these settings:

- **Type:**
  - String
  - Number
  - CommaDelimitedList
  - List<Type>
  - AWS Parameter (to help catch invalid values – match against existing values in the AWS Account)
- **Description**
- **Constraints**
  - **ConstraintDescription (String)**
  - **Min/MaxLength**
  - **Min/MaxValue**
  - **Defaults**
  - **AllowedValues (array)**
  - **AllowedPattern (regexp)**
  - **NoEcho (Boolean)**

# How to Reference a Parameter

- The `Fn::Ref` function can be leveraged to reference parameters
- Parameters can be used anywhere in a template.
- The shorthand for this in YAML is `!Ref`
- The function can also reference other elements within the template

```
DbSubnet1:  
  Type: AWS::EC2::Subnet  
  Properties:  
    VpcId: !Ref MyVPC
```

# Concept: Pseudo Parameters

- AWS offers us pseudo parameters in any CloudFormation template.
- These can be used at any time and are enabled by default

Reference Value	Example Return Value
<code>AWS::AccountId</code>	1234567890
<code>AWS::NotificationARNs</code>	[arn:aws:sns:us-east-1:123456789012:MyTopic]
<code>AWS::NoValue</code>	Does not return a value.
<code>AWS::Region</code>	us-east-2
<code>AWS::StackId</code>	arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/1c2fa620-982a-11e3-aff7-50e2416294e0
<code>AWS::StackName</code>	MyStack

# What are mappings?

- Mappings are fixed variables within your CloudFormation Template.
- They're very handy to differentiate between different environments (dev vs prod), regions (AWS regions), AMI types, etc
- All the values are hardcoded within the template
- Example:

```
Mappings:  
  Mapping01:  
    Key01:  
      Name: Value01  
    Key02:  
      Name: Value02  
    Key03:  
      Name: Value03
```

```
RegionMap:  
  us-east-1:  
    "32": "ami-6411e20d"  
    "64": "ami-7a11e213"  
  us-west-1:  
    "32": "ami-c9c7978c"  
    "64": "ami-cfc7978a"  
  eu-west-1:  
    "32": "ami-37c2f643"  
    "64": "ami-31c2f645"
```

# When would you use mappings vs parameters ?

- Mappings are great when you know in advance all the values that can be taken and that they can be deduced from variables such as
  - Region
  - Availability Zone
  - AWS Account
  - Environment (dev vs prod)
  - Etc...
- They allow safer control over the template.
- Use parameters when the values are really user specific

# Fn::FindInMap

## Accessing Mapping Values

- We use **Fn::FindInMap** to return a named value from a specific key
- **!FindInMap [ MapName, TopLevelKey, SecondLevelKey ]**

```
AWSTemplateFormatVersion: "2010-09-09"
Mappings:
  RegionMap:
    us-east-1:
      "32": "ami-6411e20d"
      "64": "ami-7a11e213"
    us-west-1:
      "32": "ami-c9c7978c"
      "64": "ami-cfc7978a"
    eu-west-1:
      "32": "ami-37c2f643"
      "64": "ami-31c2f645"
    ap-southeast-1:
      "32": "ami-66f28c34"
      "64": "ami-60f28c32"
    ap-northeast-1:
      "32": "ami-9c03a89d"
      "64": "ami-a003a8a1"
Resources:
  myEC2Instance:
    Type: "AWS::EC2::Instance"
    Properties:
      ImageId: !FindInMap [RegionMap, !Ref "AWS::Region", 32]
      InstanceType: m1.small
```

# What are outputs?

- The Outputs section declares *optional* outputs values that we can import into other stacks (if you export them first)!
- You can also view the outputs in the AWS Console or in using the AWS CLI
- They're very useful for example if you define a network CloudFormation, and output the variables such as VPC ID and your Subnet IDs
- It's the best way to perform some collaboration cross stack, as you let expert handle their own part of the stack
- You can't delete a CloudFormation Stack if its outputs are being referenced by another CloudFormation stack



# Outputs Example

- Creating a SSH Security Group as part of one template
- We create an output that references that security group

## Outputs:

### StackSSHSecurityGroup:

**Description:** The SSH Security Group for our Company

**Value:** !Ref MyCompanyWideSSHSecurityGroup

**Export:**

**Name:** SSHSecurityGroup

# Cross Stack Reference

- We then create a second template that leverages that security group
- For this, we use the **Fn::ImportValue** function
- You can't delete the underlying stack until all the references are deleted too.

```
Resources:
  MySecureInstance:
    Type: AWS::EC2::Instance
    Properties:
      AvailabilityZone: us-east-1a
      ImageId: ami-a4c7edb2
      InstanceType: t2.micro
      SecurityGroups:
        - !ImportValue SSHSecurityGroup
```

# What are conditions used for?

- Conditions are used to control the creation of resources or outputs based on a condition.
- Conditions can be whatever you want them to be, but common ones are:
  - Environment (dev / test / prod)
  - AWS Region
  - Any parameter value
- Each condition can reference another condition, parameter value or mapping

# How to define a condition?

## Conditions:

```
CreateProdResources: !Equals [ !Ref EnvType, prod ]
```

- The logical ID is for you to choose. It's how you name condition
- The intrinsic function (logical) can be any of the following:
  - Fn::And
  - Fn::Equals
  - Fn::If
  - Fn::Not
  - Fn::Or

# Using a Condition

- Conditions can be applied to resources / outputs / etc...

```
Resources:  
  MountPoint:  
    Type: "AWS::EC2::VolumeAttachment"  
    Condition: CreateProdResources
```

# CloudFormation Must Know Intrinsic Functions

- Ref
- Fn::GetAtt
- Fn::FindInMap
- Fn::ImportValue
- Fn::Join
- Fn::Sub
- Condition Functions (Fn::If, Fn::Not, Fn::Equals, etc...)

# Fn::Ref

- The `Fn::Ref` function can be leveraged to reference
  - `Parameters` => returns the value of the parameter
  - `Resources` => returns the physical ID of the underlying resource (ex: EC2 ID)
- The shorthand for this in YAML is `!Ref`

```
DbSubnet1:  
  Type: AWS::EC2::Subnet  
  Properties:  
    VpcId: !Ref MyVPC
```

# Fn::GetAtt

- Attributes are attached to any resources you create
- To know the attributes of your resources, the best place to look at is the documentation.
- For example: the AZ of an EC2 machine!

## Resources:

### EC2Instance:

Type: "AWS::EC2::Instance"

### Properties:

ImageId: ami-1234567

InstanceType: t2.micro

## NewVolume:

Type: "AWS::EC2::Volume"

Condition: CreateProdResources

### Properties:

Size: 100

AvailabilityZone:

!GetAtt EC2Instance.AvailabilityZone



# Fn::FindInMap

## Accessing Mapping Values

- We use **Fn::FindInMap** to return a named value from a specific key
- **!FindInMap [ MapName, TopLevelKey, SecondLevelKey ]**

```
AWSTemplateFormatVersion: "2010-09-09"
Mappings:
  RegionMap:
    us-east-1:
      "32": "ami-6411e20d"
      "64": "ami-7a11e213"
    us-west-1:
      "32": "ami-c9c7978c"
      "64": "ami-cfc7978a"
    eu-west-1:
      "32": "ami-37c2f643"
      "64": "ami-31c2f645"
    ap-southeast-1:
      "32": "ami-66f28c34"
      "64": "ami-60f28c32"
    ap-northeast-1:
      "32": "ami-9c03a89d"
      "64": "ami-a003a8a1"
Resources:
  myEC2Instance:
    Type: "AWS::EC2::Instance"
    Properties:
      ImageId: !FindInMap [RegionMap, !Ref "AWS::Region", 32]
      InstanceType: m1.small
```

# Fn::ImportValue

- Import values that are exported in other templates
- For this, we use the **Fn::ImportValue** function

```
Resources:
  MySecureInstance:
    Type: AWS::EC2::Instance
    Properties:
      AvailabilityZone: us-east-1a
      ImageId: ami-a4c7edb2
      InstanceType: t2.micro
      SecurityGroups:
        - !ImportValue SSHSecurityGroup
```

# Fn::Join

- Join values with a delimiter

```
!Join [ delimiter, [ comma-delimited list of values ] ]
```

- This creates “a:b:c”

```
!Join [ ":", [ a, b, c ] ]
```

# Function `Fn::Sub`

- `Fn::Sub`, or `!Sub` as a shorthand, is used to substitute variables from a text. It's a very handy function that will allow you to fully customize your templates.
- For example, you can combine `Fn::Sub` with References or AWS Pseudo variables!
- `String` must contain `${VariableName}` and will substitute them

```
!Sub  
- String  
- { Var1Name: Var1Value, Var2Name: Var2Value }
```

```
!Sub String
```

# Condition Functions

## Conditions:

```
CreateProdResources: !Equals [ !Ref EnvType, prod ]
```

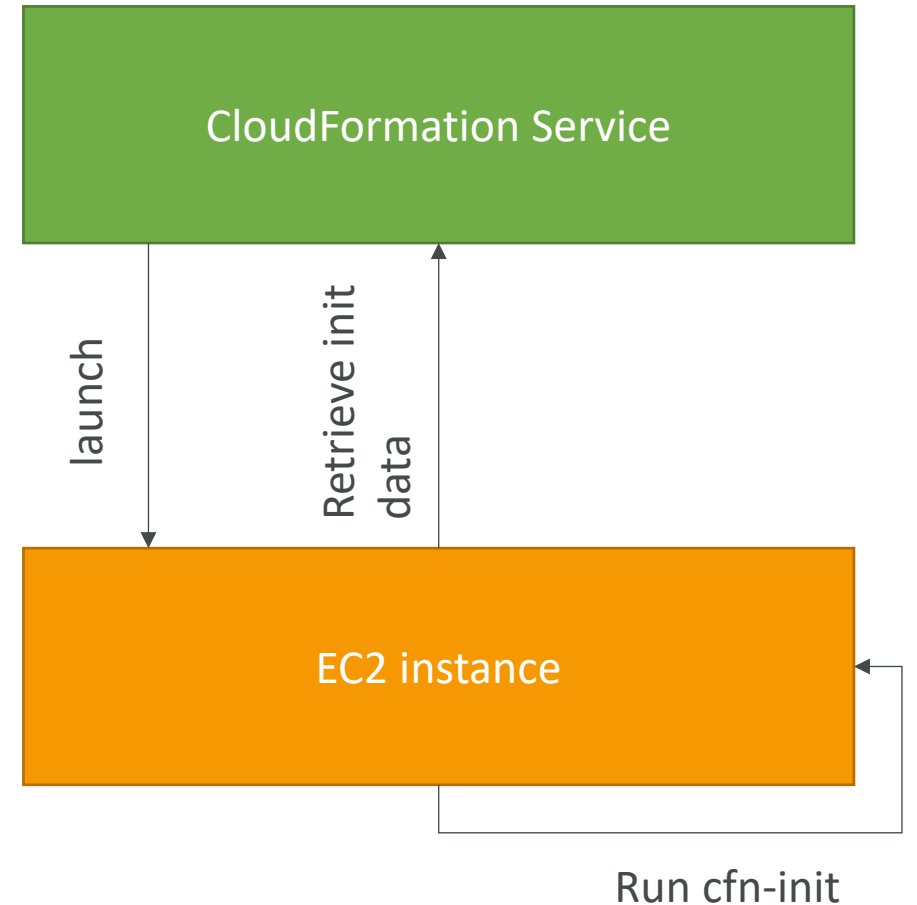
- The logical ID is for you to choose. It's how you name condition
- The intrinsic function (logical) can be any of the following:
  - Fn::And
  - Fn::Equals
  - Fn::If
  - Fn::Not
  - Fn::Or

# User Data in EC2 for CloudFormation

- We can have user data at EC2 instance launch through the console
- We can also include it in CloudFormation
- The important thing to pass is the entire script through the function `Fn::Base64`
- Good to know: user data script log is in `/var/log/cloud-init-output.log`
- Let's see how to do this in CloudFormation

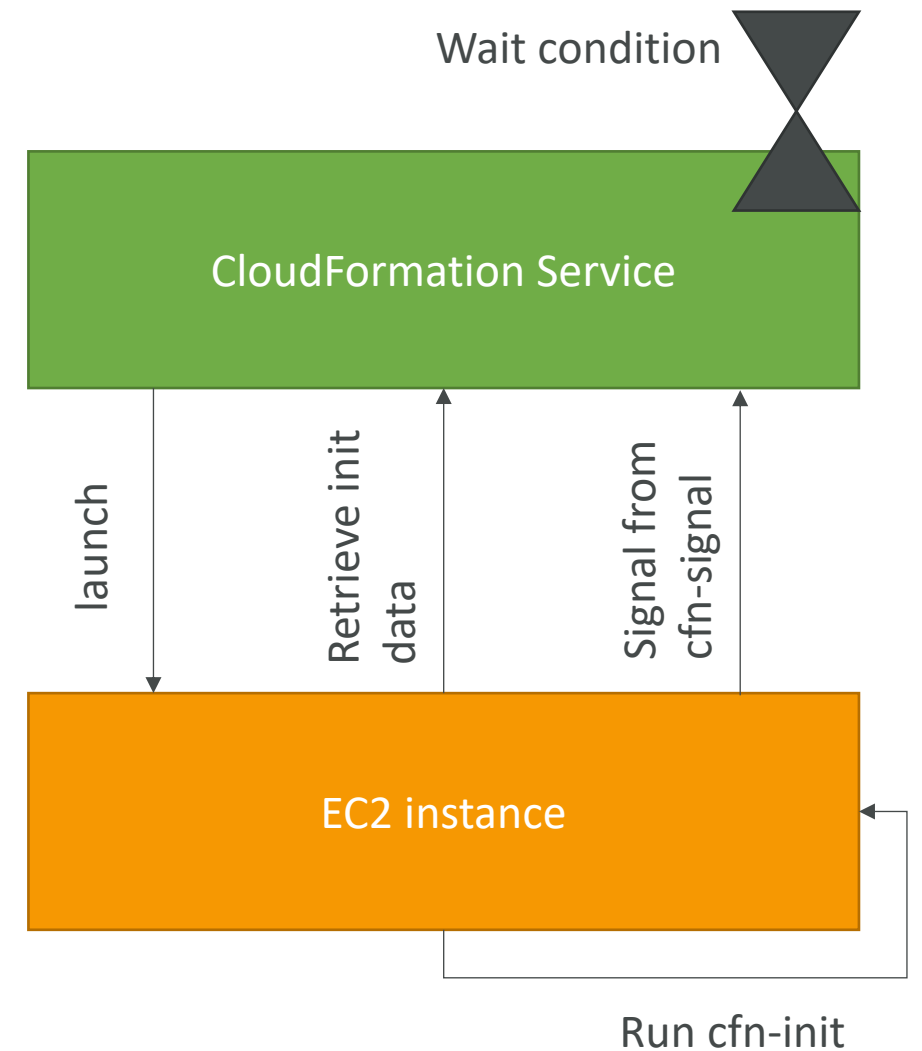
# cfn-init

- AWS::CloudFormation::Init must be in the Metadata of a resource
- With the cfn-init script, it helps make complex EC2 configurations readable
- The EC2 instance will query the CloudFormation service to get init data
- Logs go to `/var/log/cfn-init.log`
- Let's see how it works through a sample CloudFormation



# cfn-signal & wait conditions

- We still don't know how to tell CloudFormation that the EC2 instance got properly configured after a **cfn-init**
- For this, we can use the **cfn-signal** script!
  - We run cfn-signal right after cfn-init
  - Tell CloudFormation service to keep on going or fail
- We need to define **WaitCondition**:
  - Block the template until it receives a signal from cfn-signal
  - We attach a **CreationPolicy** (also works on EC2, ASG)





# Wait Condition Didn't Receive the Required Number of Signals from an Amazon EC2 Instance

- Ensure that the AMI you're using has the AWS CloudFormation helper scripts installed. If the AMI doesn't include the helper scripts, you can also download them to your instance.
- Verify that the `cfn-init` & `cfn-signal` command was successfully run on the instance. You can view logs, such as `/var/log/cloud-init.log` or `/var/log/cfn-init.log`, to help you debug the instance launch.
- You can retrieve the logs by logging in to your instance, but you must disable rollback on failure or else AWS CloudFormation deletes the instance after your stack fails to create.
- Verify that the instance has a connection to the Internet. If the instance is in a VPC, the instance should be able to connect to the Internet through a NAT device if it's in a private subnet or through an Internet gateway if it's in a public subnet.
- For example, run: `curl -I https://aws.amazon.com`

# Rollbacks on failures

- Stack Creation Fails: (CreateStack API)
  - Default: everything rolls back (gets deleted). We can look at the log `OnFailure=ROLLBACK`
  - Troubleshoot: Option to disable rollback and manually troubleshoot `OnFailure=DO_NOTHING`
  - Delete: get rid of the stack entirely, do not keep anything `OnFailure=DELETE`
- Stack Update Fails: (UpdateStack API)
  - The stack automatically rolls back to the previous known working state
  - Ability to see in the log what happened and error messages

# Stack creation failures

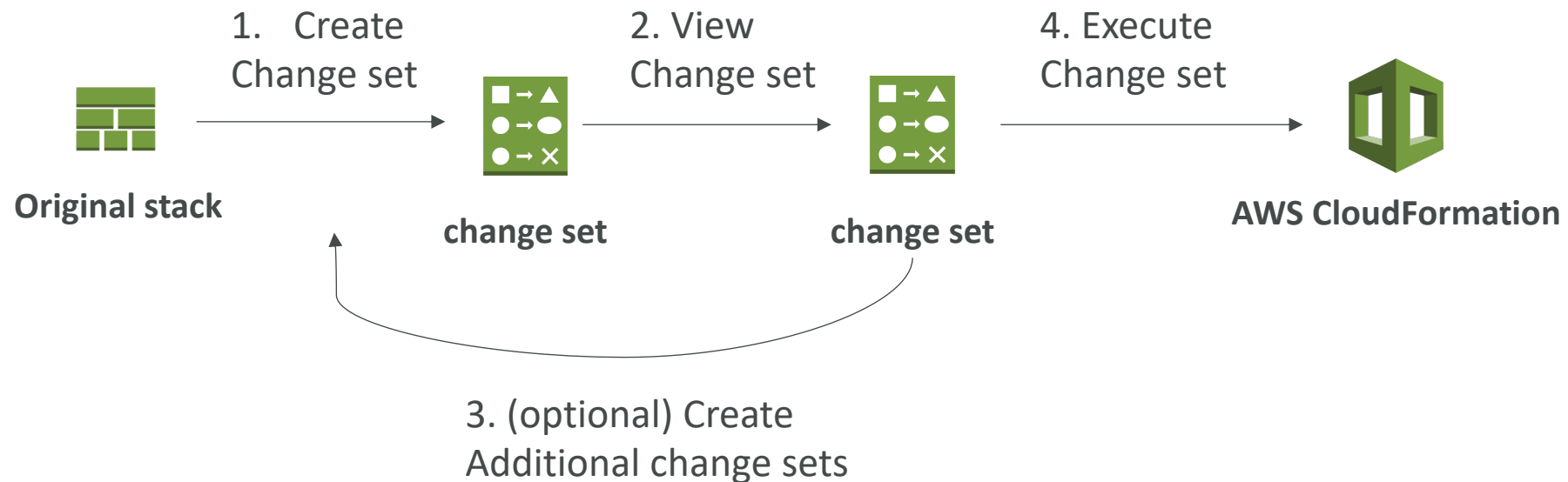
- If a CloudFormation stack creation fails, you will get the status `ROLLBACK_COMPLETE`
- This means:
  - 1. CloudFormation tried to create some resources
  - 2. One resource creation failed
  - 3. CloudFormation rolled back the resources (`ROLLBACK`, `DO_NOTHING`)
  - 4. The stack is in failed created `ROLLBACK_COMPLETE` state
- To resolve the error, there's only one way:  
Delete the failed stack and create a new stack
- You can't update, validate or change-set on a create failed stack

# Nested stacks

- Nested stacks are stacks as part of other stacks
- They allow you to isolate repeated patterns / common components in separate stacks and call them from other stacks
- Example:
  - Load Balancer configuration that is re-used
  - Security Group that is re-used
- Nested stacks are considered best practice
- To update a nested stack, always update the parent (root stack)

# ChangeSets

- When you update a stack, you need to know what changes before it happens for greater confidence
- ChangeSets won't say if the update will be successful



From: <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-updating-stacks-changesets.html>

# CloudFormation Drift

- CloudFormation allows you to create infrastructure
- But it doesn't protect you against manual configuration changes
- How do we know if our resources have **drifted**?
- We can use CloudFormation drift!
- Not all resources are supported yet:  
<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-stack-drift-resource-list.html>

# Retaining Data on Deletes

- You can put a `DeletionPolicy` on any resource to control what happens when the CloudFormation template is deleted
- **DeletionPolicy=Retain:**
  - Specify on resources to preserve / backup in case of CloudFormation deletes
  - To keep a resource, specify **Retain** (works for any resource / nested stack)
- **DeletionPolicy=Snapshot:**
  - EBS Volume, ElastiCache Cluster, ElastiCache ReplicationGroup
  - RDS DBInstance, RDS DBCluster, Redshift Cluster
- **DeletePolicy=Delete (default behavior):**
  - Note: for `AWS::RDS::DBCluster` resources, the default policy is Snapshot
  - Note: to delete an S3 bucket, you need to first empty the bucket of its content

# Termination Protection on Stacks

- To prevent accidental deletes of CloudFormation templates, use TerminationProtection
- Let's see this quickly!

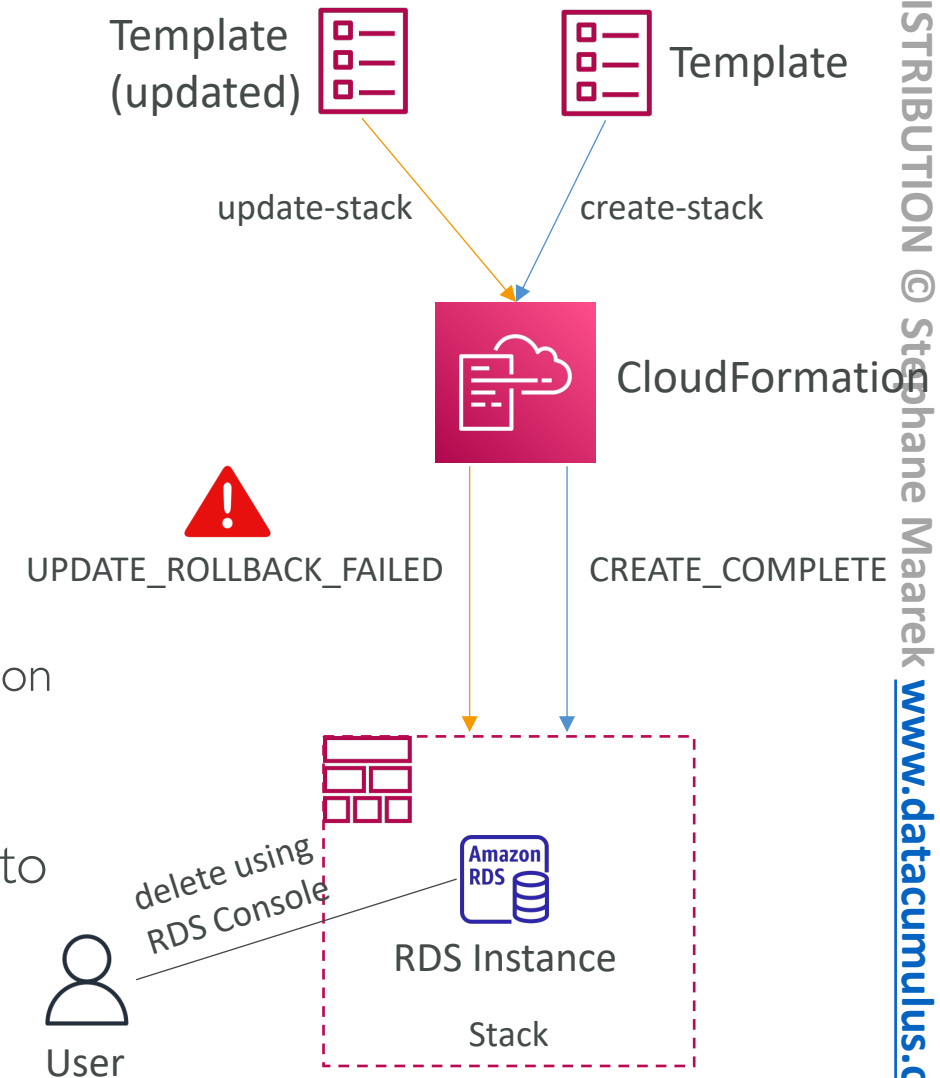


# CloudFormation StackSets

- Create, update, or delete stacks across **multiple accounts and regions** with a single operation
- Administrator account to create StackSets
- Trusted accounts to create, update, delete stack instances from StackSets
- When you update a stack set, *all* associated stack instances are updated throughout all accounts and regions.
- Ability to set a maximum concurrent actions on targets (# or %)
- Ability to set failure tolerance (# or %)

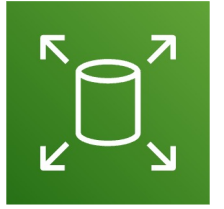
# Continue Rolling Back an Update

- A stack goes into the UPDATE\_ROLLBACK\_FAILED state when CloudFormation can't roll back all changes during an update
- A resource can't return to its original state, causing the rollback to fail
- Example: roll back to an old database instance that was deleted outside CloudFormation
- Solutions:
  - Fix the errors manually outside of CloudFormation and then continue update rollback the stack
  - Skip the resources that can't rollback successfully (CloudFormation will mark the failed resources as UPDATE\_COMPLETE)
- You can't update a stack in this state
- For nested stacks, rolling back the parent stack will attempt to roll back all the child stacks as well



# EC2 Storage and Data Management

EBS, Instance Store & EFS



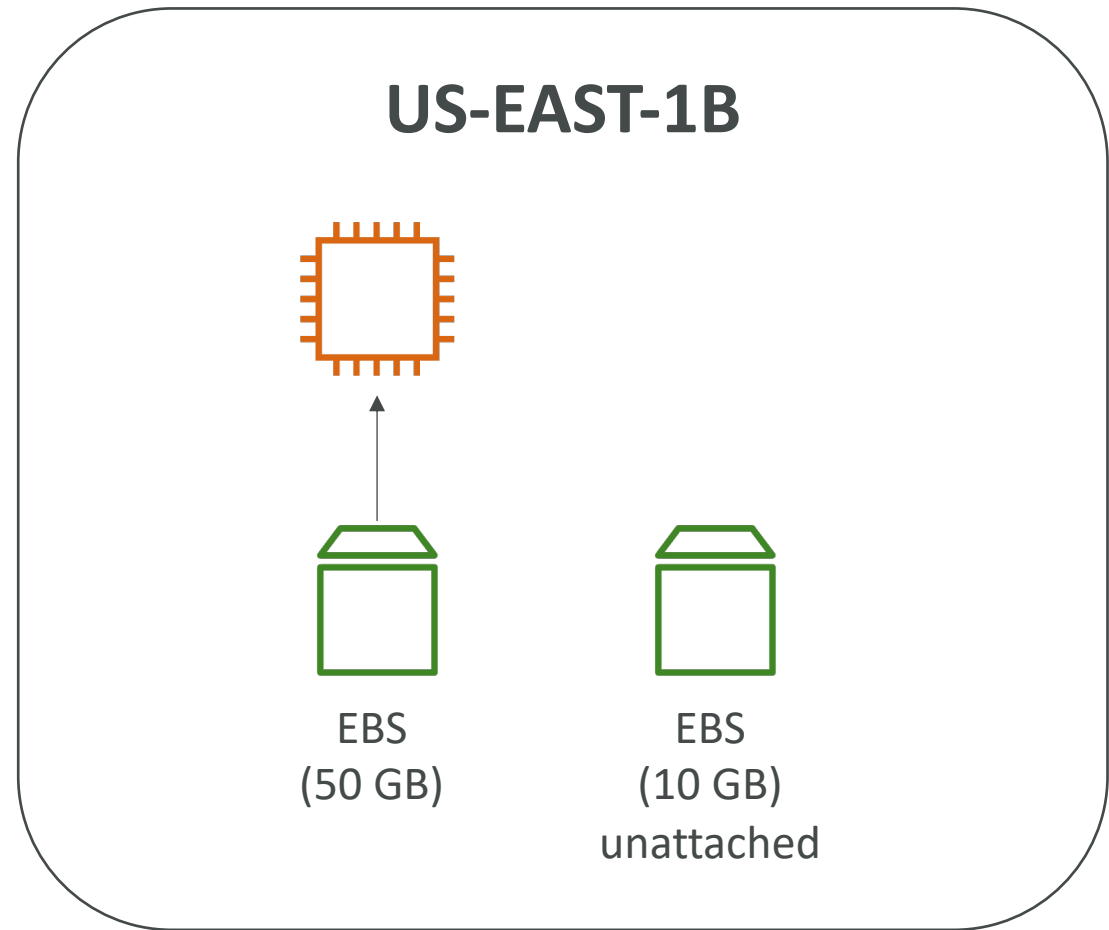
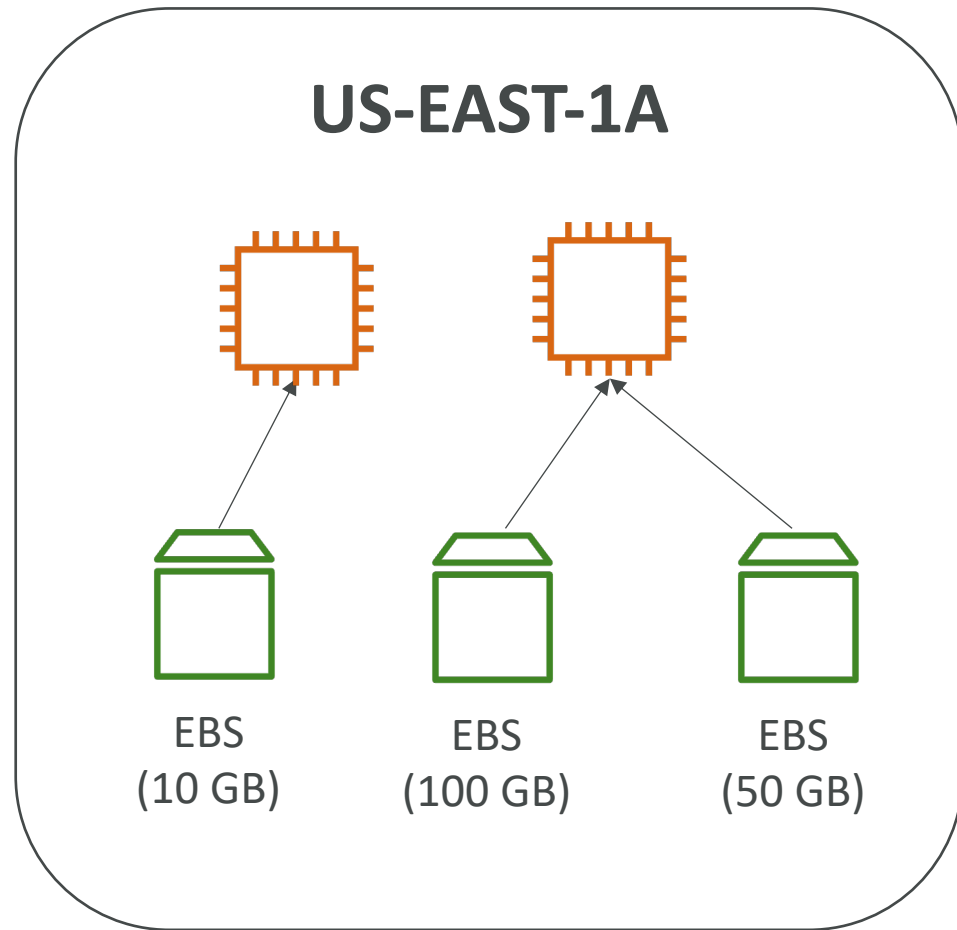
# What's an EBS Volume?

- An **EBS (Elastic Block Store) Volume** is a **network** drive you can attach to your instances while they run
- It allows your instances to persist data, even after their termination
- **They can only be mounted to one instance at a time** (at the CCP level)
- They are bound to a **specific availability zone**
  
- Analogy: Think of them as a “network USB stick”
- Free tier: 30 GB of free EBS storage of type General Purpose (SSD) or Magnetic per month

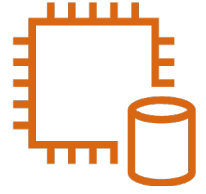
# EBS Volume

- It's a network drive (i.e. not a physical drive)
  - It uses the network to communicate the instance, which means there might be a bit of latency
  - It can be detached from an EC2 instance and attached to another one quickly
- It's locked to an Availability Zone (AZ)
  - An EBS Volume in us-east-1a cannot be attached to us-east-1b
  - To move a volume across, you first need to snapshot it
- Have a provisioned capacity (size in GBs, and IOPS)
  - You get billed for all the provisioned capacity
  - You can increase the capacity of the drive over time

# EBS Volume - Example



# EC2 Instance Store



- EBS volumes are **network drives** with good but “limited” performance
- If you need a high-performance hardware disk, use EC2 Instance Store
- Better I/O performance
- EC2 Instance Store lose their storage if they're stopped (ephemeral)
- Good for buffer / cache / scratch data / temporary content
- Risk of data loss if hardware fails
- Backups and Replication are your responsibility

# Local EC2 Instance Store

Very high IOPS

Instance Size	100% Random Read IOPS	Write IOPS
i3.large *	100,125	35,000
i3.xlarge *	206,250	70,000
i3.2xlarge	412,500	180,000
i3.4xlarge	825,000	360,000
i3.8xlarge	1.65 million	720,000
i3.16xlarge	3.3 million	1.4 million
i3.metal	3.3 million	1.4 million
i3en.large *	42,500	32,500
i3en.xlarge *	85,000	65,000
i3en.2xlarge *	170,000	130,000
i3en.3xlarge	250,000	200,000
i3en.6xlarge	500,000	400,000
i3en.12xlarge	1 million	800,000
i3en.24xlarge	2 million	1.6 million
i3en.metal	2 million	1.6 million



# EBS Volume Types

- EBS Volumes come in 6 types
  - **gp2 / gp3 (SSD)**: General purpose SSD volume that balances price and performance for a wide variety of workloads
  - **io1 / io2 (SSD)**: Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads
  - **st1 (HDD)**: Low cost HDD volume designed for frequently accessed, throughput-intensive workloads
  - **sc1 (HDD)**: Lowest cost HDD volume designed for less frequently accessed workloads
- EBS Volumes are characterized in Size | Throughput | IOPS (I/O Ops Per Sec)
- When in doubt always consult the AWS documentation – it's good!
- Only gp2/gp3 and io1/io2 can be used as boot volumes

# 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

# 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/io2 (4 GiB - 16 TiB):
  - Max PIOPS: 64,000 for Nitro EC2 instances & 32,000 for other
  - Can increase PIOPS independently from storage size
  - io2 have more durability and more IOPS per GiB (at the same price as io1)
- 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

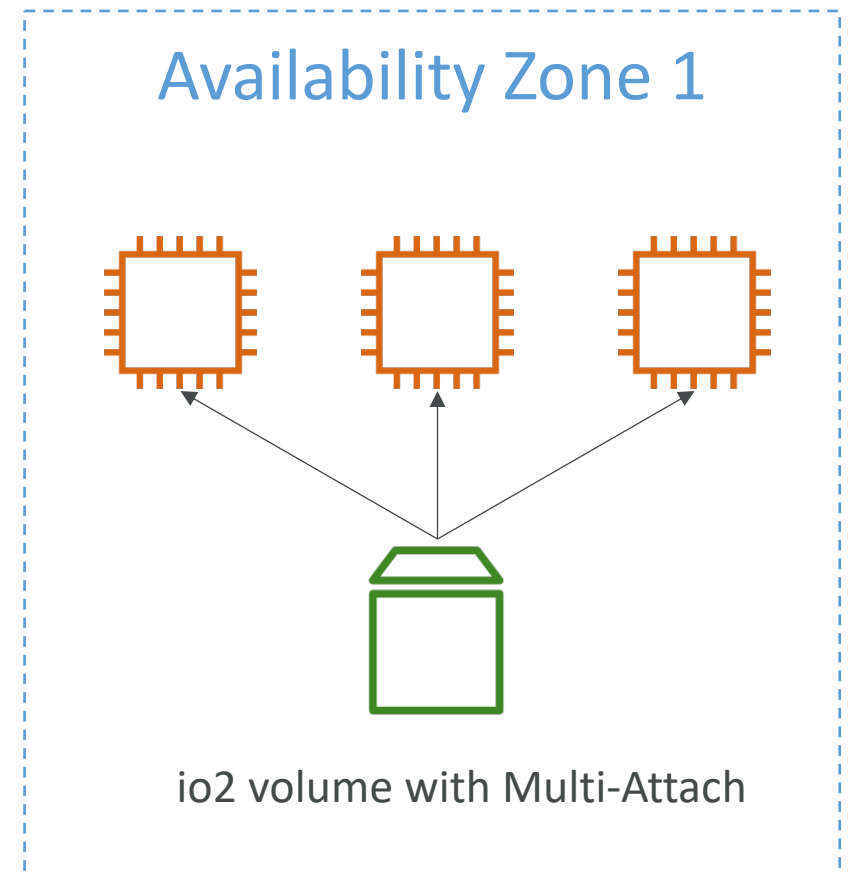
	General Purpose SSD		Provisioned IOPS SSD		
Volume type	gp3	gp2	io2 Block Express ‡	io2	io1
<b>Durability</b>	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.999% durability (0.001% annual failure rate)		99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)
<b>Use cases</b>	<ul style="list-style-type: none"> <li>Low-latency interactive apps</li> <li>Development and test environments</li> </ul>		Workloads that require sub-millisecond latency, and sustained IOPS performance or more than 64,000 IOPS or 1,000 MiB/s of throughput	<ul style="list-style-type: none"> <li>Workloads that require sustained IOPS performance or more than 16,000 IOPS</li> <li>I/O-intensive database workloads</li> </ul>	
<b>Volume size</b>	1 GiB - 16 TiB		4 GiB - 64 TiB	4 GiB - 16 TiB	
<b>Max IOPS per volume (16 KiB I/O)</b>	16,000		256,000	64,000 †	

	Throughput Optimized HDD	Cold HDD
<b>Volume type</b>	st1	sc1
<b>Durability</b>	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)
<b>Use cases</b>	<ul style="list-style-type: none"> <li>Big data</li> <li>Data warehouses</li> <li>Log processing</li> </ul>	<ul style="list-style-type: none"> <li>Throughput-oriented storage for data that is infrequently accessed</li> <li>Scenarios where the lowest storage cost is important</li> </ul>
<b>Volume size</b>	125 GiB - 16 TiB	125 GiB - 16 TiB
<b>Max IOPS per volume (1 MiB I/O)</b>	500	250
<b>Max throughput per volume</b>	500 MiB/s	250 MiB/s
<b>Amazon EBS Multi-attach</b>	Not supported	Not supported
<b>Boot volume</b>	Not supported	Not supported

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html#solid-state-drives>

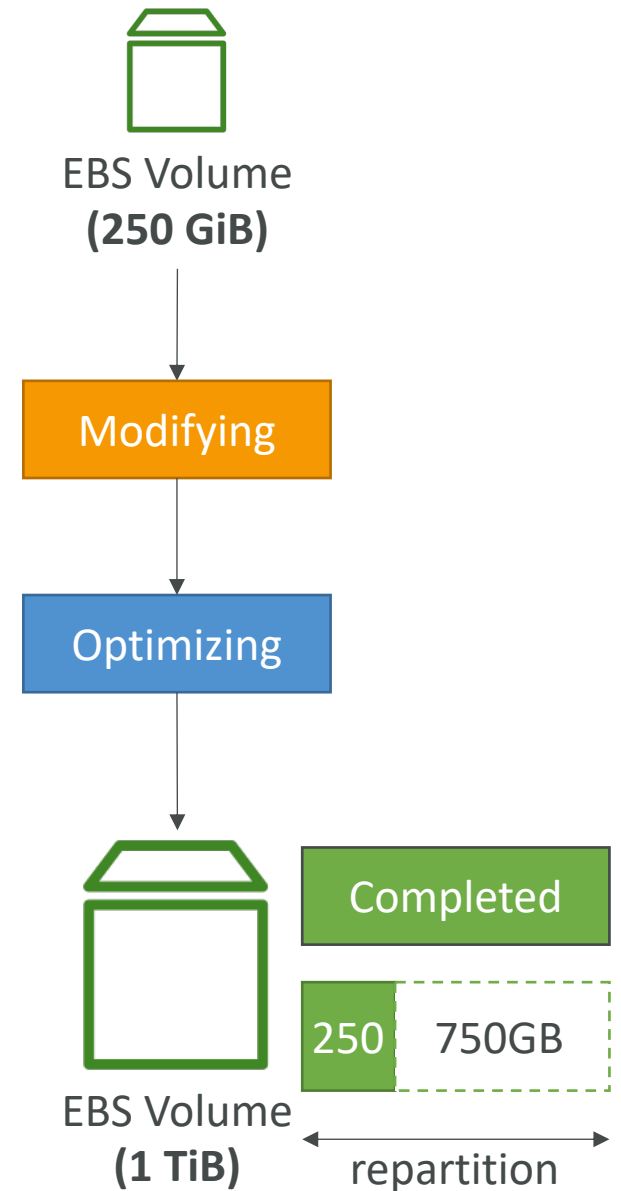
# EBS Multi-Attach – io1/io2 family

- Attach the same EBS volume to multiple EC2 instances in the same AZ
- Each instance has full read & write permissions to the high-performance volume
- Use case:
  - Achieve **higher application availability** in clustered Linux applications (ex: Teradata)
  - Applications must manage concurrent write operations
- Up to 16 EC2 Instances at a time
- Must use a file system that's cluster-aware (not XFS, EX4, etc...)



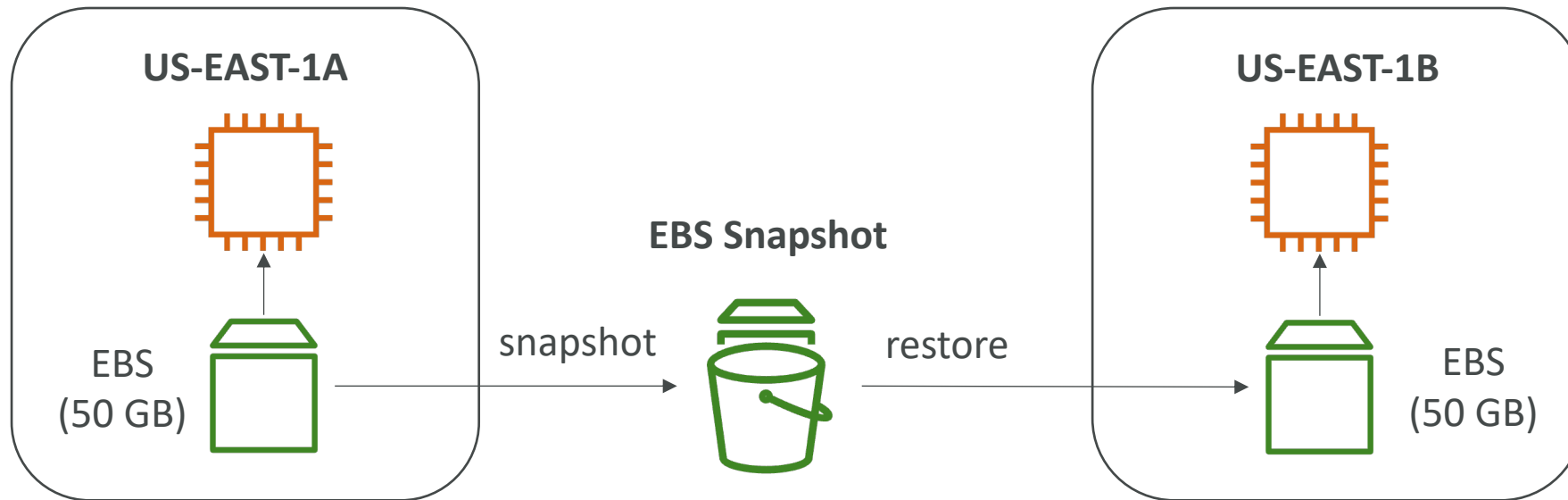
# EBS Volume Resizing

- You can only increase the EBS volumes:
  - Size (any volume type)
  - IOPS (only in IO1)
- After resizing an EBS volume, you need to repartition your drive
- After increasing the size, it's possible for the volume to be in a long time in the "optimisation" phase. The volume is still usable
- You can't decrease the size of your EBS volume (create another smaller volume then migrate data)



# EBS Snapshots

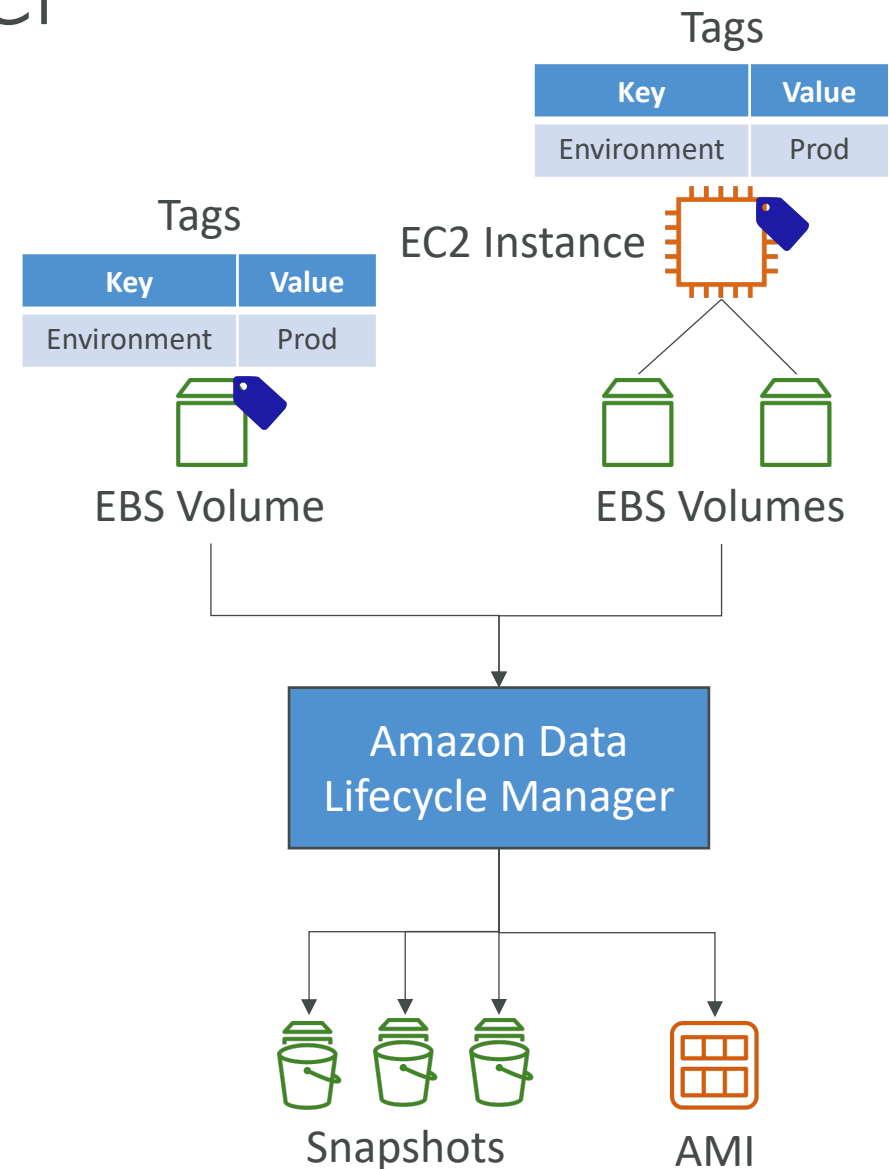
- Make a backup (snapshot) of your EBS volume at a point in time
- Not necessary to detach volume to do snapshot, but recommended
- Can copy snapshots across AZ or Region





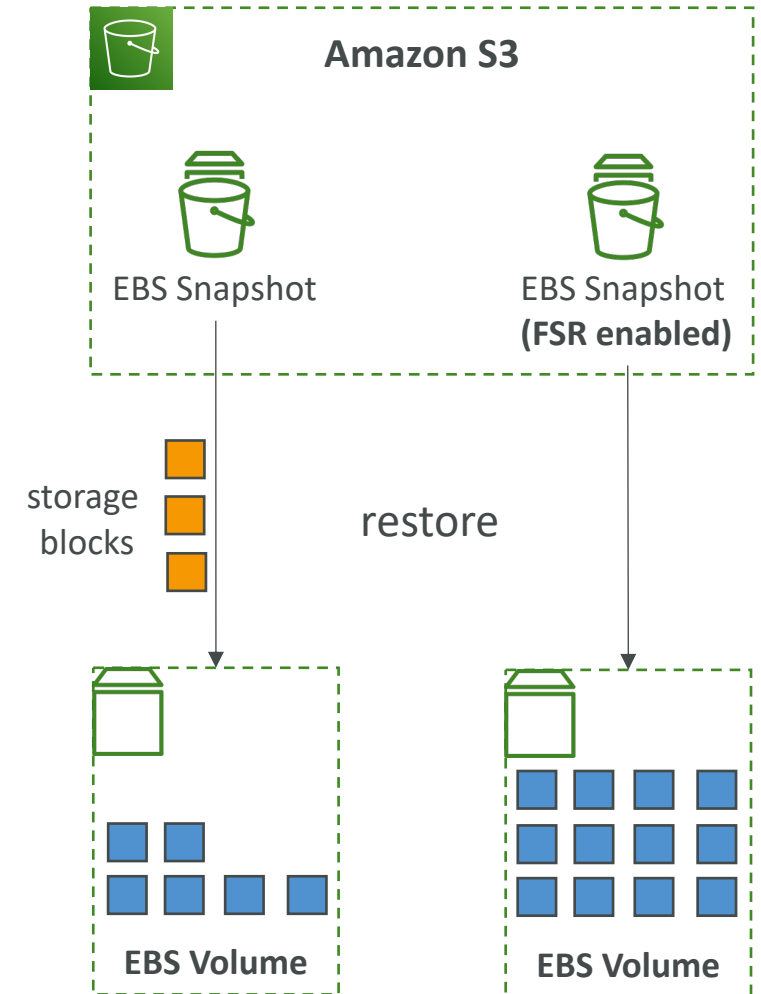
# Amazon Data Lifecycle Manager

- Automate the creation, retention, and deletion of EBS snapshots and EBS-backed AMIs
- Schedule backups, cross-account snapshot copies, delete outdated backups, ...
- Uses resource tags to identify the resources (EC2 instances, EBS volumes)
- Can't be used to manage snapshots/AMIs created outside DLM
- Can't be used to manage instance-store backed AMIs



# EBS Snapshots – Fast Snapshot Restore (FSR)

- EBS Snapshots stored in S3
- By default, there's a latency of I/O operations the first time each block is accessed (block must be pulled from S3)
- **Solution:** force the initialization of the entire volume (using the *dd* or *fio* command), or you can enable FSR
- FSR helps you to create a volume from a snapshot that is fully initialized at creation (no I/O latency)
- Enabled for a snapshot in a particular AZ (billed per minute – very expensive \$\$\$)
- Can be enabled on snapshots created by Data Lifecycle Manager



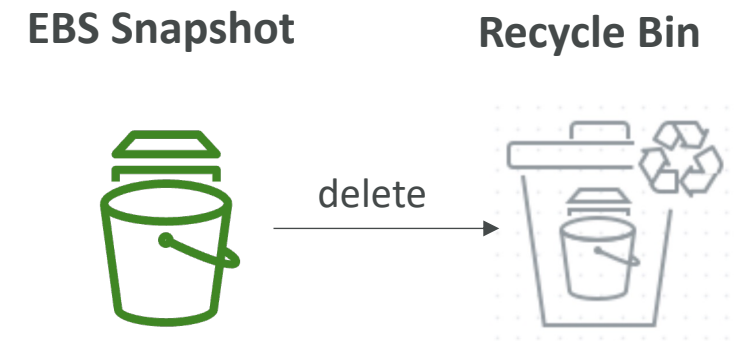
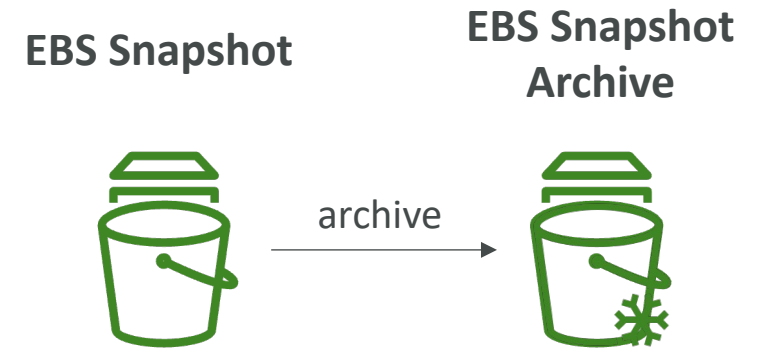
# EBS Snapshots Features

- EBS Snapshot Archive

- Move a Snapshot to an "archive tier" that is 75% cheaper
- Takes within 24 to 72 hours for restoring the archive

- Recycle Bin for EBS Snapshots

- Setup rules to retain deleted snapshots so you can recover them after an accidental deletion
- Specify retention (from 1 day to 1 year)



# EBS Migration

- EBS Volumes are only locked to a specific AZ
- To migrate it to a different AZ (or region):
  - Snapshot the volume
  - (optional) Copy the volume to a different region
  - Create a volume from the snapshot in the AZ of your choice
- Let's practice!

# EBS Encryption

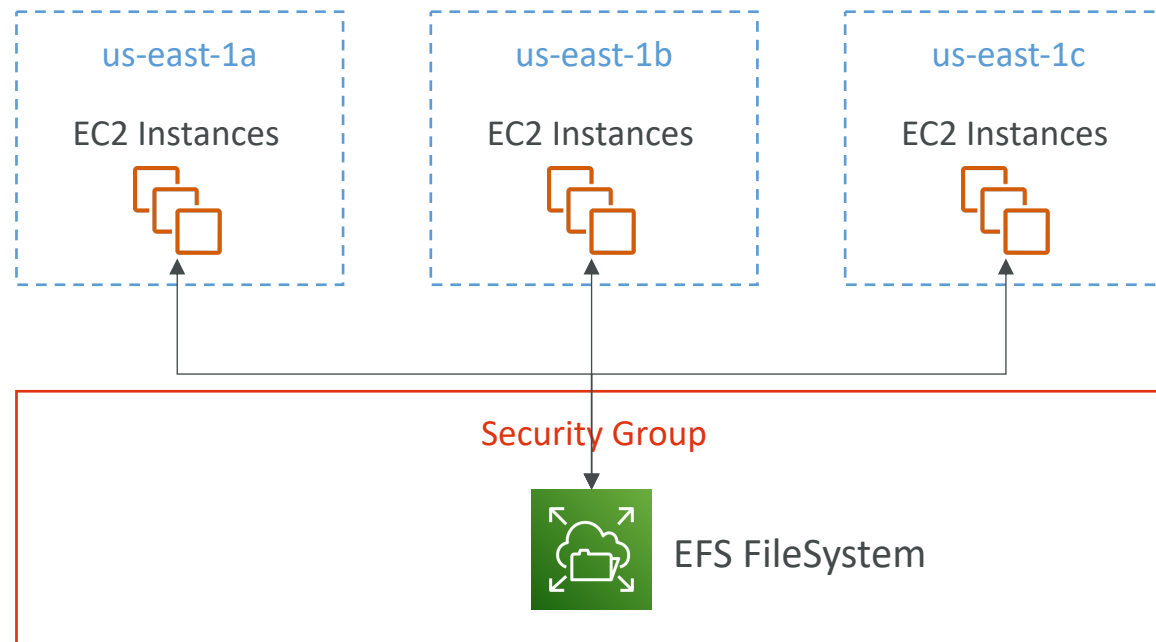
- When you create an encrypted EBS volume, you get the following:
  - Data at rest is encrypted inside the volume
  - All the data in flight moving between the instance and the volume is encrypted
  - All snapshots are encrypted
  - All volumes created from the snapshot
- Encryption and decryption are handled transparently (you have nothing to do)
- Encryption has a minimal impact on latency
- EBS Encryption leverages keys from KMS (AES-256)
- Copying an unencrypted snapshot allows encryption
- Snapshots of encrypted volumes are encrypted

# Encryption: encrypt an unencrypted EBS volume

- Create an EBS snapshot of the volume
- Encrypt the EBS snapshot ( using copy )
- Create new ebs volume from the snapshot ( the volume will also be encrypted )
- Now you can attach the encrypted volume to the original instance

# EFS – Elastic File System

- Managed NFS (network file system) that can be mounted on many EC2
- EFS works with EC2 instances in multi-AZ
- Highly available, scalable, expensive (3x gp2), pay per use



# EFS – Elastic File System

- Use cases: content management, web serving, data sharing, Wordpress
- Uses NFSv4.1 protocol
- Uses security group to control access to EFS
- **Compatible with Linux based AMI (not Windows)**
- Encryption at rest using KMS
  
- POSIX file system (~Linux) that has a standard file API
- File system scales automatically, pay-per-use, no capacity planning!

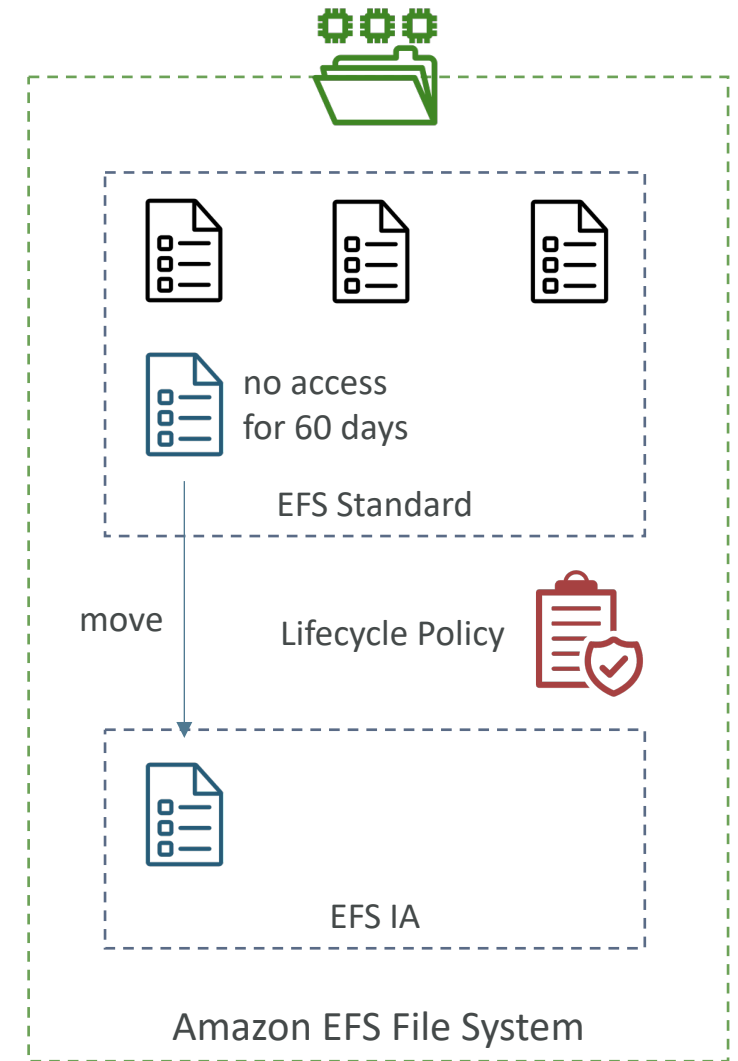


# EFS – Performance & Storage Classes

- **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 1 GiB/s for writes
    - Used for unpredictable workloads

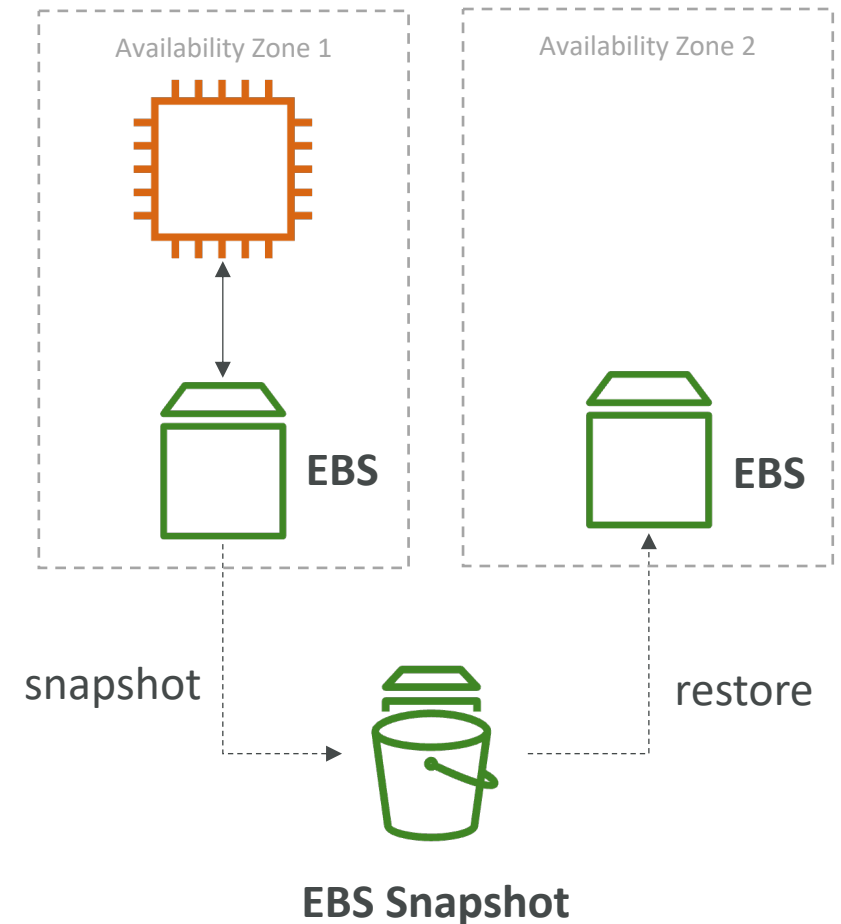
# EFS – Storage Classes

- 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



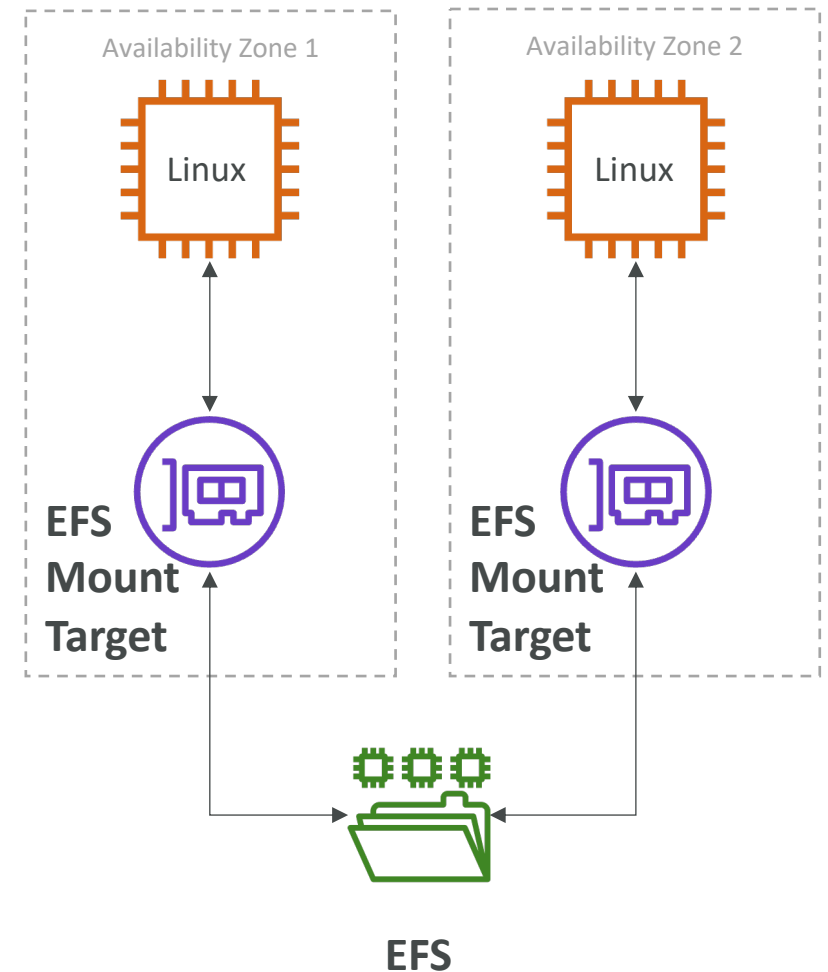
# EBS vs EFS – Elastic Block Storage

- EBS volumes...
  - one instance (except multi-attach io1/io2)
  - are locked at the Availability Zone (AZ) level
  - gp2: IO increases if the disk size increases
  - io1: can increase IO independently
- To migrate an EBS volume across AZ
  - Take a snapshot
  - Restore the snapshot to another AZ
  - EBS backups use IO and you shouldn't run them while your application is handling a lot of traffic
- Root EBS Volumes of instances get terminated by default if the EC2 instance gets terminated. (you can disable that)



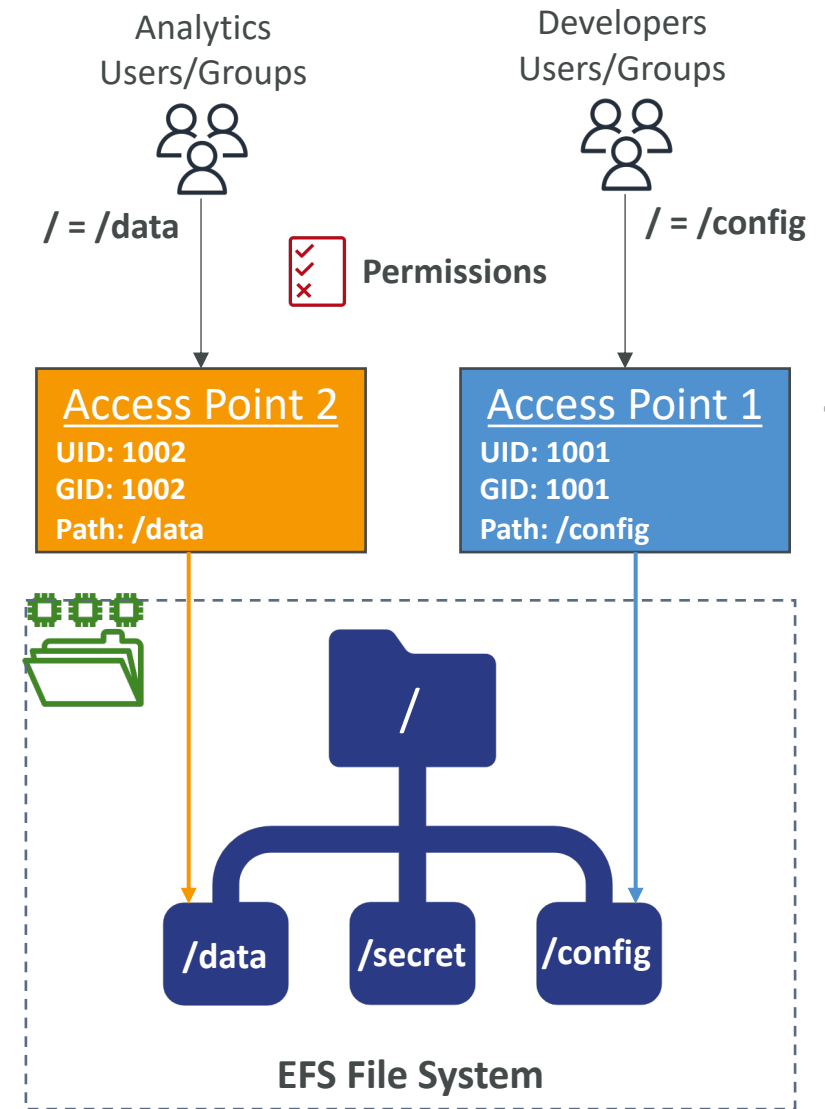
# EBS vs EFS – Elastic File System

- Mounting 100s of instances across AZ
- EFS share website files (WordPress)
- Only for Linux Instances (POSIX)
- EFS has a higher price point than EBS
- Can leverage EFS-IA for cost savings
- Remember: EFS vs EBS vs Instance Store



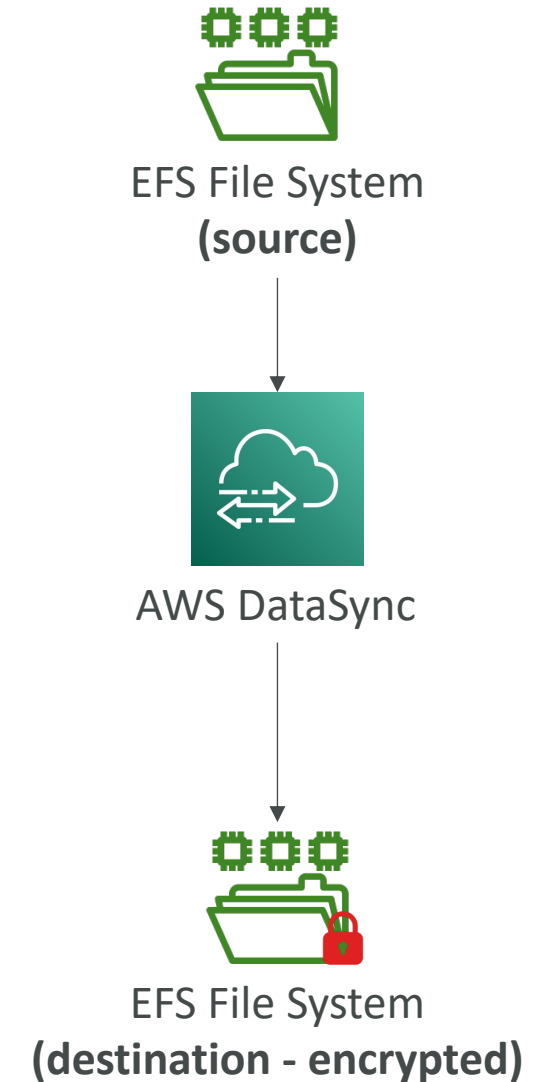
# EFS – Access Points

- Easily manage applications access to NFS environments
- Enforce a POSIX user and group to use when accessing the file system
- Restrict access to a directory within the file system and optionally specify a different root directory
- Can restrict access from NFS clients using IAM policies



# EFS - Operations

- Operations that can be done in place:
  - Lifecycle Policy (enable IA or change IA settings)
  - Throughput Mode and Provisioned Throughput Numbers
  - EFS Access Points
- Operations that require a migration using DataSync (replicates all file attributes and metadata)
  - Migration to encrypted EFS
  - Performance Mode (e.g. Max IO)



# EFS – CloudWatch Metrics

- **PercentIOLimit**
  - How close the file system reaching the I/O limit (General Purpose)
  - If at 100%, move to Max I/O (migration)
- **BurstCreditBalance**
  - The number of burst credits the file system can use to achieve higher throughput levels
- **StorageBytes**
  - File system's size in bytes (15 minutes interval)
  - Dimensions: Standard, IA, Total (Standard + IA)

Throughput utilization (%)



Burst Credit balance



# Amazon S3 Section



# Section introduction

- Amazon S3 is one of the main building blocks of AWS
- It's advertised as "infinitely scaling" storage
- Many websites use Amazon S3 as a backbone
- Many AWS services use Amazon S3 as an integration as well
- We'll have a step-by-step approach to S3



# Amazon S3 Use cases

- Backup and storage
- Disaster Recovery
- Archive
- Hybrid Cloud storage
- Application hosting
- Media hosting
- Data lakes & big data analytics
- Software delivery
- Static website



Nasdaq stores 7 years of data into S3 Glacier



Sysco runs analytics on its data and gain business insights

# Amazon S3 - Buckets

- Amazon S3 allows people to store objects (files) in “buckets” (directories)
- Buckets must have a **globally unique name** (across all regions all accounts)
- Buckets are defined at the region level
- S3 looks like a global service but buckets are created in a region
- Naming convention
  - No uppercase, No underscore
  - 3-63 characters long
  - Not an IP
  - Must start with lowercase letter or number
  - Must NOT start with the prefix **xn--**
  - Must NOT end with the suffix **-s3alias**



S3 Bucket

# Amazon S3 - Objects

- Objects (files) have a Key
- The **key** is the **FULL** path:
  - `s3://my-bucket/my_file.txt`
  - `s3://my-bucket/my_folder1/another_folder/my_file.txt`
- The key is composed of **prefix** + **object name**
  - `s3://my-bucket/my_folder1/another_folder/my_file.txt`
- There's no concept of "directories" within buckets (although the UI will trick you to think otherwise)
- Just keys with very long names that contain slashes ("/")



Object



S3 Bucket  
with Objects

# Amazon S3 – Objects (cont.)



- Object values are the content of the body:
  - Max. Object Size is 5TB (5000GB)
  - If uploading more than 5GB, must use “multi-part upload”
- Metadata (list of text key / value pairs – system or user metadata)
- Tags (Unicode key / value pair – up to 10) – useful for security / lifecycle
- Version ID (if versioning is enabled)

# Amazon S3 – Security

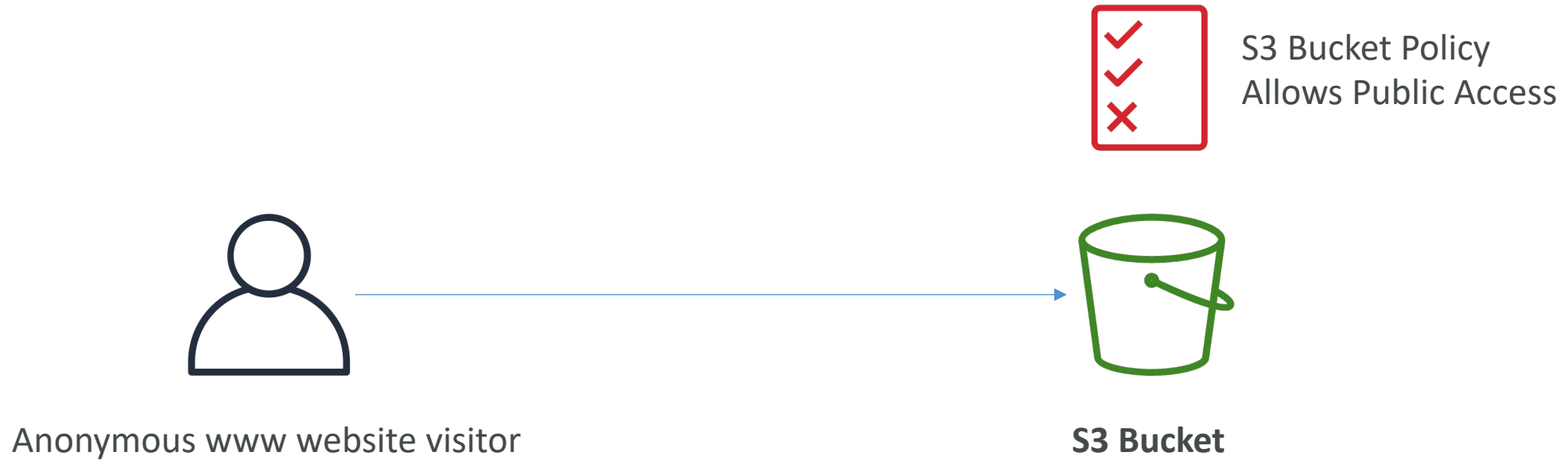
- User-Based
  - IAM Policies – which API calls should be allowed for a specific user from IAM
- Resource-Based
  - Bucket Policies – bucket wide rules from the S3 console - allows cross account
  - Object Access Control List (ACL) – finer grain (can be disabled)
  - Bucket Access Control List (ACL) – less common (can be disabled)
- **Note:** an IAM principal can access an S3 object if
  - The user IAM permissions ALLOW it OR the resource policy ALLOWS it
  - AND there's no explicit DENY
- **Encryption:** encrypt objects in Amazon S3 using encryption keys

# S3 Bucket Policies

- JSON based policies
  - Resources: buckets and objects
  - Effect: Allow / Deny
  - Actions: Set of API to Allow or Deny
  - Principal: The account or user to apply the policy to
- Use S3 bucket for policy to:
  - Grant public access to the bucket
  - Force objects to be encrypted at upload
  - Grant access to another account (Cross Account)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicRead",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::examplebucket/*"
      ]
    }
  ]
}
```

# Example: Public Access - Use Bucket Policy

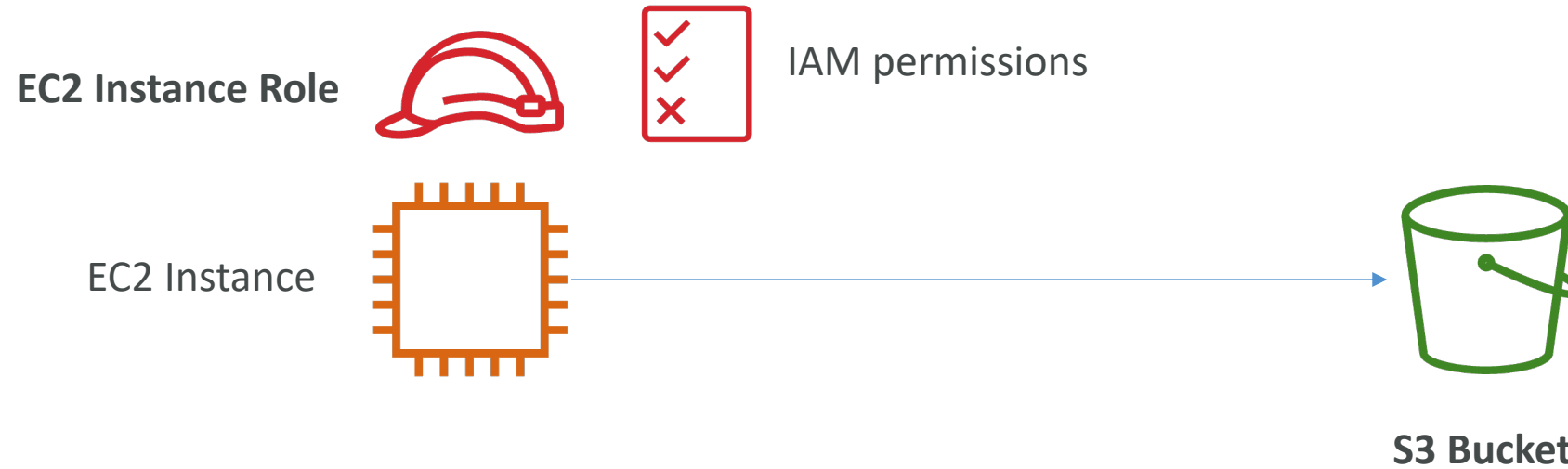




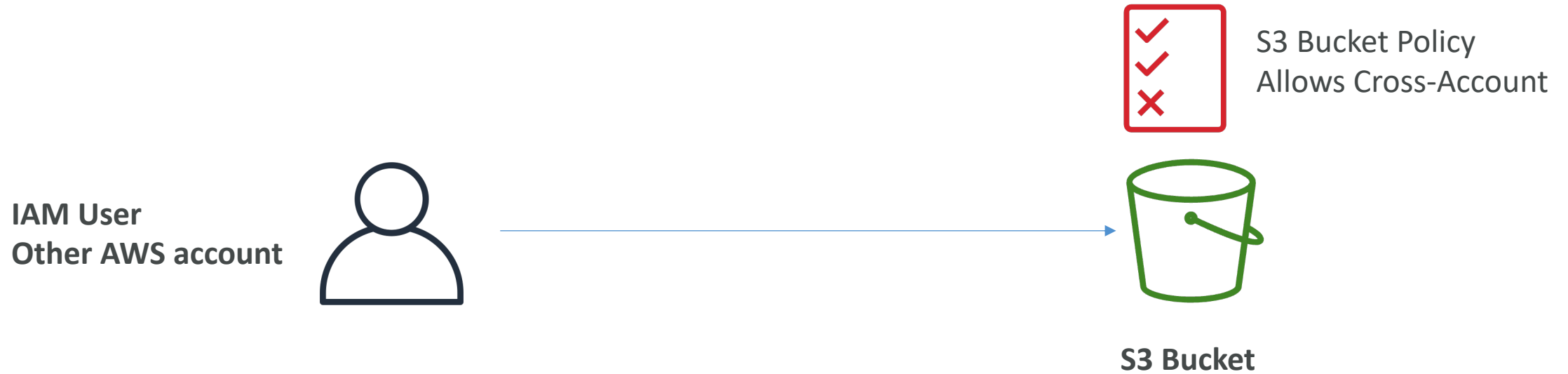
# Example: User Access to S3 – IAM permissions



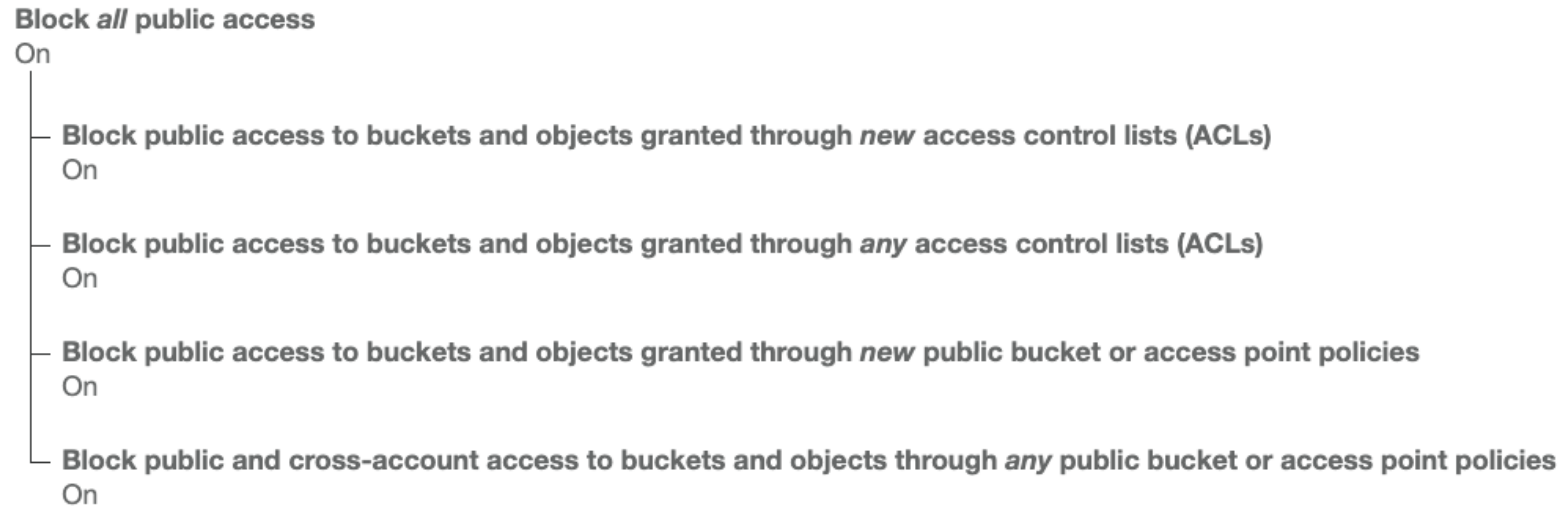
# Example: EC2 instance access - Use IAM Roles



# Advanced: Cross-Account Access – Use Bucket Policy



# Bucket settings for Block Public Access



- These settings were created to prevent company data leaks
- If you know your bucket should never be public, leave these on
- Can be set at the account level

# S3 Bucket Policies

- Use S3 bucket for policy to:
  - Grant public access to the bucket
  - Force objects to be encrypted at upload
  - Grant access to another account (Cross Account)
- Optional Conditions on:
  - Public IP or Elastic IP (not on Private IP)
  - Source VPC or Source VPC Endpoint – only works with VPC Endpoints
  - CloudFront Origin Identity
  - MFA
- Examples here: <https://docs.aws.amazon.com/AmazonS3/latest/dev/example-bucket-policies.html>

# Bucket Policies – Advanced Examples

- Restrict access to only principals from AWS accounts inside an AWS Organization using `aws:PrincipalOrgID` condition key

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::mybucket/*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": ["o-exampleorgid"]
        }
      }
    }
  ]
}
```

- Prevent uploads of unencrypted objects to an S3 bucket using `s3:x-amz-server-side-encryption` condition key

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

# Bucket Policies – Advanced Examples

- Restrict access to specific IP addresses using NotIpAddress condition key

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::mybucket",
        "arn:aws:s3:::mybucket/*"
      ],
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": "54.240.143.0/24"
        }
      }
    }
  ]
}
```

# Bucket Policies – Advanced Examples

- Grant user access to list and download all objects in an S3 bucket

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

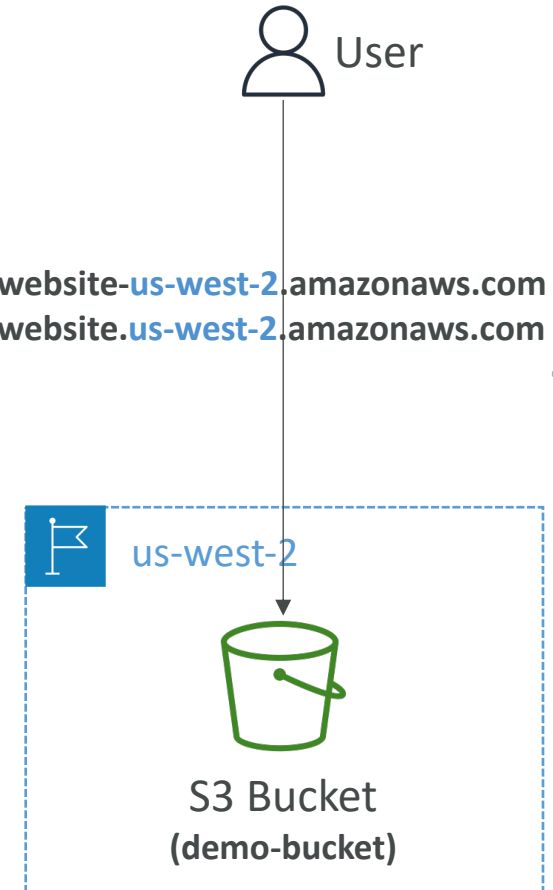
- Restrict access to users authenticated using MFA using **MultiFactorAuthPresent** condition key

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::mybucket/*",
      "Condition": {
        "Bool": {
          "aws:MultiFactorAuthPresent": "true"
        }
      }
    }
  ]
}
```



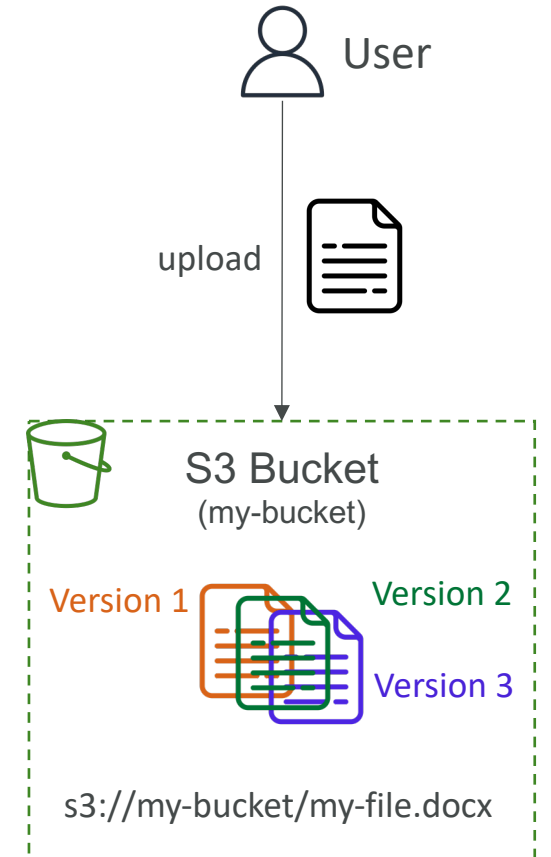
# Amazon S3 – Static Website Hosting

- S3 can host static websites and have them accessible on the Internet
- The website URL will be (depending on the region)
  - <http://demo-bucket.s3-website-us-west-2.amazonaws.com>  
<http://demo-bucket.s3-website-us-west-2.amazonaws.com>
  - <http://bucket-name.s3-website-aws-region.amazonaws.com>  
OR
  - <http://bucket-name.s3-website.aws-region.amazonaws.com>
- If you get a **403 Forbidden** error, make sure the bucket policy allows public reads!



# Amazon S3 - Versioning

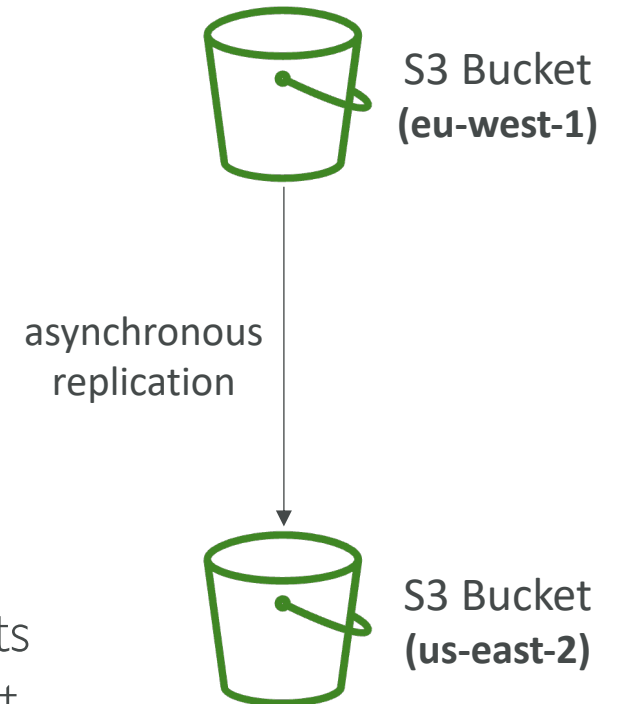
- You can version your files in Amazon S3
- It is enabled at the **bucket level**
- Same key overwrite will change the “version”: 1, 2, 3....
- It is best practice to version your buckets
  - Protect against unintended deletes (ability to restore a version)
  - Easy roll back to previous version
- Notes:
  - Any file that is not versioned prior to enabling versioning will have version “null”
  - Suspending versioning does not delete the previous versions



# Amazon S3 – Replication (CRR & SRR)



- Must enable **Versioning** in source and destination buckets
- **Cross-Region Replication (CRR)**
- **Same-Region Replication (SRR)**
- Buckets can be in different AWS accounts
- Copying is asynchronous
- Must give proper IAM permissions to S3
- Use cases:
  - **CRR** – compliance, lower latency access, replication across accounts
  - **SRR** – log aggregation, live replication between production and test accounts



# Amazon S3 – Replication (Notes)

- After you enable Replication, 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 setting)
  - Deletions with a version ID are not replicated (to avoid malicious deletes)
- There is 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

- 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 (99.9999999999%, 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

# S3 Standard – General Purpose



- 99.99% Availability
  - Used for frequently accessed data
  - Low latency and high throughput
  - Sustain 2 concurrent facility failures
- 
- Use Cases: Big Data analytics, mobile & gaming applications, content distribution...

# S3 Storage Classes – Infrequent Access

- For data that is less frequently accessed, but requires rapid access when needed
- Lower cost than S3 Standard
- 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-premises data, or data you can recreate





# 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





# 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

# S3 Storage Classes Comparison

	Standard	Intelligent-Tiering	Standard-IA	One Zone-IA	Glacier Instant Retrieval	Glacier Flexible Retrieval	Glacier Deep Archive
Durability	99.999999999% == (11 9's)						
Availability	99.99%	99.9%	99.9%	99.5%	99.9%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99%	99.9%	99.9%
Availability Zones	>= 3	>= 3	>= 3	1	>= 3	>= 3	>= 3
Min. Storage Duration Charge	None	None	30 Days	30 Days	90 Days	90 Days	180 Days
Min. Billable Object Size	None	None	128 KB	128 KB	128 KB	40 KB	40 KB
Retrieval Fee	None	None	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved

<https://aws.amazon.com/s3/storage-classes/>

# S3 Storage Classes – Price Comparison

## Example: us-east-1

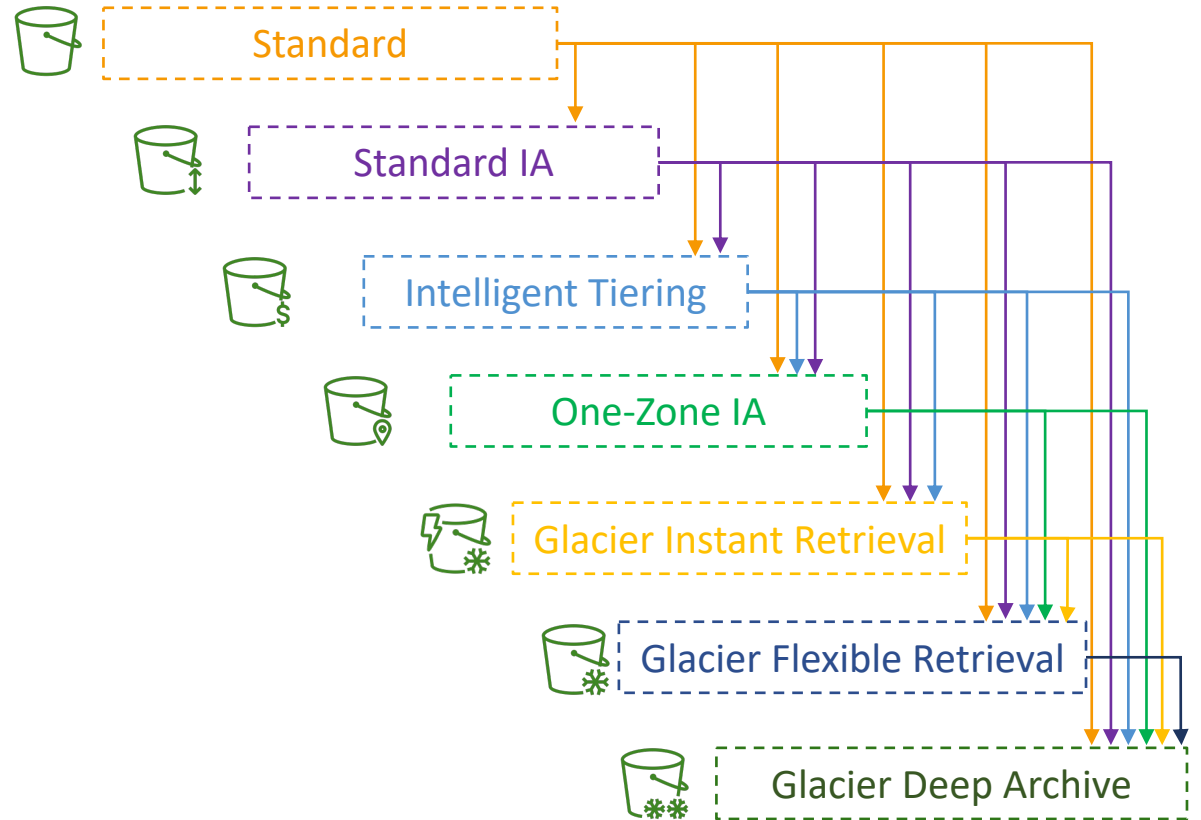
	Standard	Intelligent-Tiering	Standard-IA	One Zone-IA	Glacier Instant Retrieval	Glacier Flexible Retrieval	Glacier Deep Archive
Storage Cost (per GB per month)	\$0.023	\$0.0025 - \$0.023	%0.0125	\$0.01	\$0.004	\$0.0036	\$0.00099
Retrieval Cost (per 1000 request)	<b>GET:</b> \$0.0004 <b>POST:</b> \$0.005	<b>GET:</b> \$0.0004 <b>POST:</b> \$0.005	<b>GET:</b> \$0.001 <b>POST:</b> \$0.01	<b>GET:</b> \$0.001 <b>POST:</b> \$0.01	<b>GET:</b> \$0.01 <b>POST:</b> \$0.02	<b>GET:</b> \$0.0004 <b>POST:</b> \$0.03  <b>Expedited:</b> \$10 <b>Standard:</b> \$0.05 <b>Bulk:</b> free	<b>GET:</b> \$0.0004 <b>POST:</b> \$0.05  <b>Standard:</b> \$0.10 <b>Bulk:</b> \$0.025
Retrieval Time	Instantaneous					<b>Expedited</b> (1 – 5 mins) <b>Standard</b> (3 – 5 hours) Bulk (5 – 12 hours)	<b>Standard</b> (12 hours) <b>Bulk</b> (48 hours)
Monitoring Cost (per 1000 objects)		\$0.0025					

<https://aws.amazon.com/s3/pricing/>

# Advanced S3

# Amazon S3 – Moving between Storage Classes

- You can transition objects between storage classes
- For infrequently accessed object, move them to **Standard IA**
- For archive objects that you don't need fast access to, move them to **Glacier** or **Glacier Deep Archive**
- Moving objects can be automated using a **Lifecycle Rules**



# Amazon S3 – Lifecycle Rules



- **Transition Actions** – configure objects to transition to another storage class
  - Move objects to Standard IA class 60 days after creation
  - Move to Glacier for archiving after 6 months
- **Expiration actions** – configure objects to expire (delete) after some time
  - Access log files can be set to delete after a 365 days
  - Can be used to delete old versions of files (if versioning is enabled)
  - Can be used to delete incomplete Multi-Part uploads
- Rules can be created for a certain prefix (example: `s3://mybucket/mp3/*`)
- Rules can be created for certain objects Tags (example: `Department: Finance`)

# Amazon S3 – Lifecycle Rules (Scenario 1)

- Your application on EC2 creates images thumbnails after profile photos are uploaded to Amazon S3. These thumbnails can be easily recreated, and only need to be kept for 60 days. The source images should be able to be immediately retrieved for these 60 days, and afterwards, the user can wait up to 6 hours. How would you design this?
- S3 source images can be on **Standard**, with a lifecycle configuration to transition them to **Glacier** after 60 days
- S3 thumbnails can be on **One-Zone IA**, with a lifecycle configuration to expire them (delete them) after 60 days

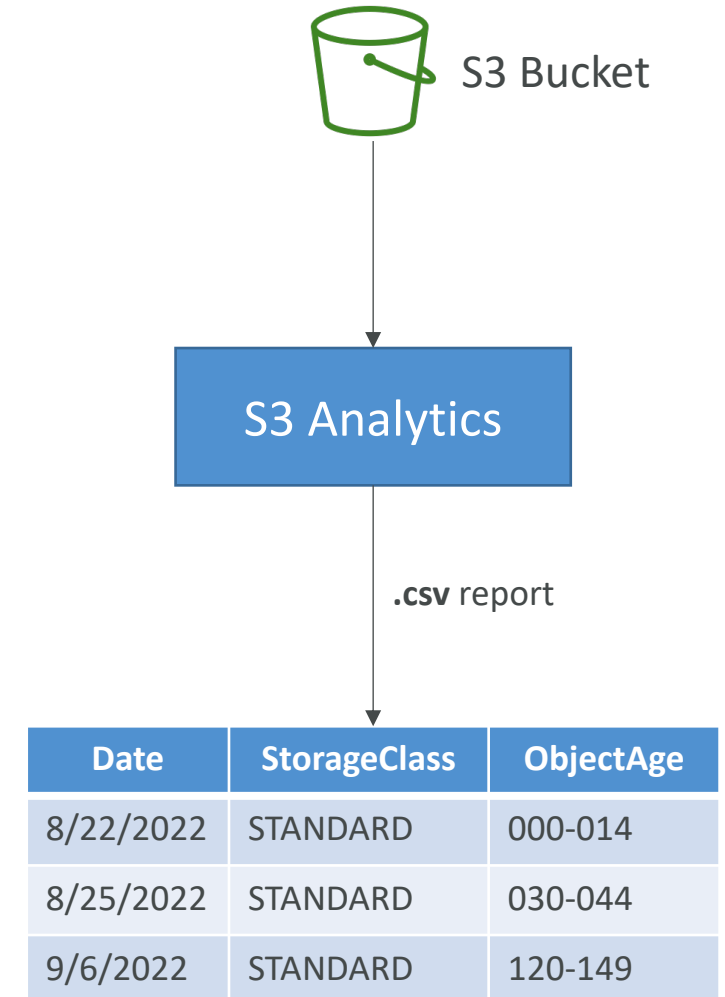


# Amazon S3 – Lifecycle Rules (Scenario 2)

- A rule in your company states that you should be able to recover your deleted S3 objects immediately for 30 days, although this may happen rarely. After this time, and for up to 365 days, deleted objects should be recoverable within 48 hours.
- **Enable S3 Versioning** in order to have object versions, so that “deleted objects” are in fact hidden by a “delete marker” and can be recovered
- Transition the “noncurrent versions” of the object to **Standard IA**
- Transition afterwards the “noncurrent versions” to **Glacier Deep Archive**

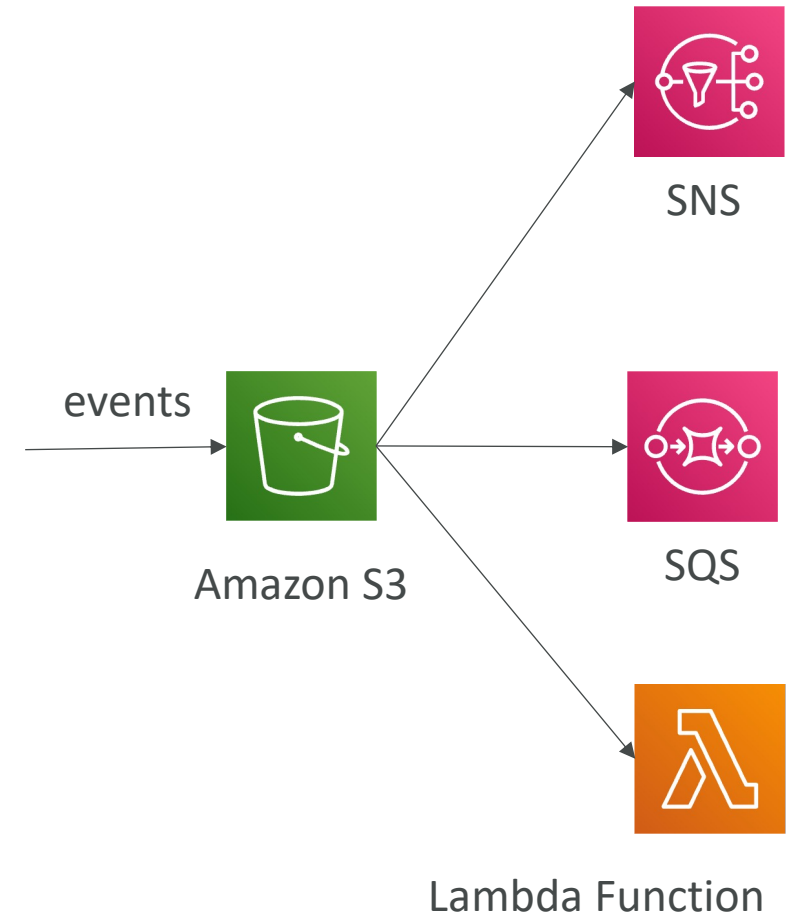
# Amazon S3 Analytics – Storage Class Analysis

- Help you 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 is updated daily
- 24 to 48 hours to start seeing data analysis
- Good first step to put together Lifecycle Rules (or improve them)!

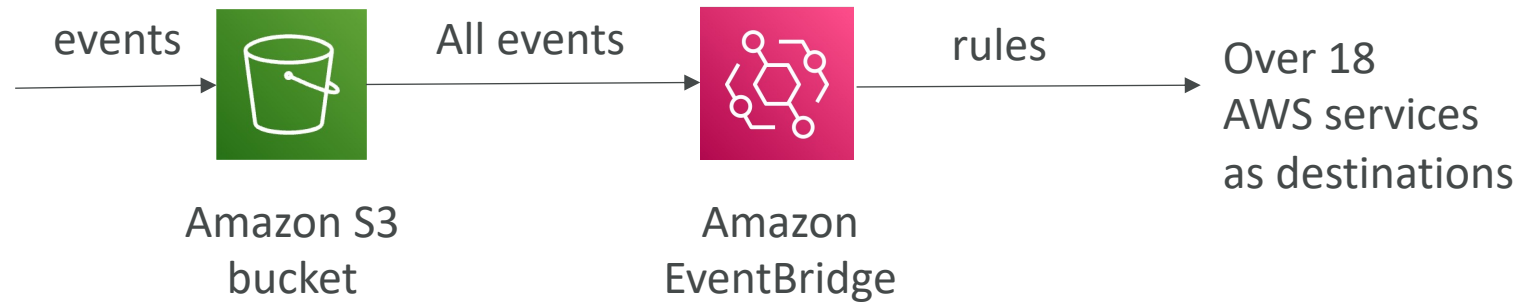


# S3 Event Notifications

- 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

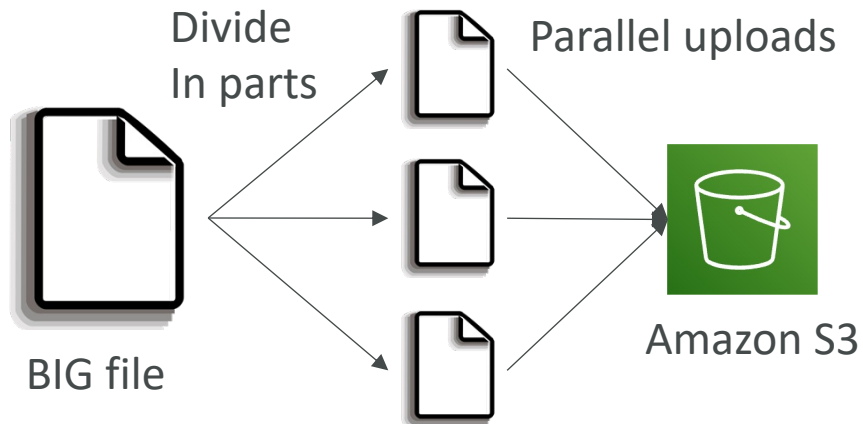
# S3 – Baseline Performance

- Amazon S3 automatically scales to high request rates, latency 100-200 ms
- Your application can achieve at least **3,500 PUT/COPY/POST/DELETE** or **5,500 GET/HEAD** requests per second per prefix in a bucket.
- There are no limits to the number of prefixes in a bucket.
- Example (object path => prefix):
  - bucket/folder1/sub1/file => /folder1/sub1/
  - bucket/folder1/sub2/file => /folder1/sub2/
  - bucket/1/file => /1/
  - bucket/2/file => /2/
- If you spread reads across all four prefixes evenly, you can achieve 22,000 requests per second for GET and HEAD

# S3 Performance

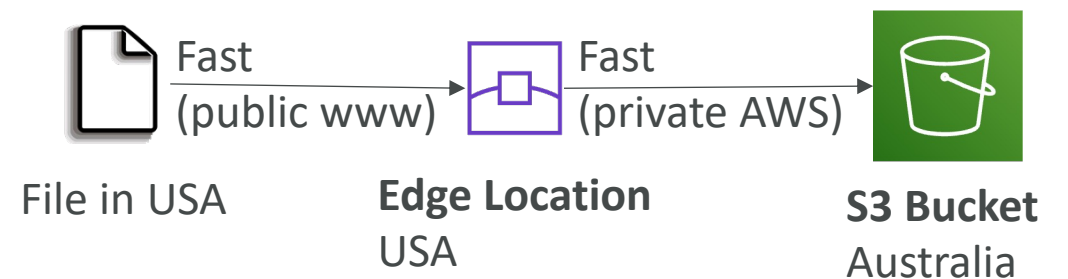
- Multi-Part upload:

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



- S3 Transfer Acceleration

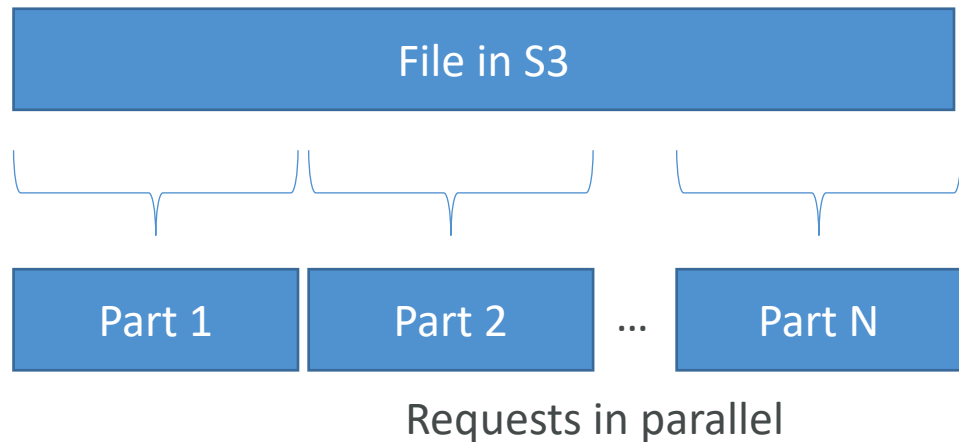
- Increase transfer speed by transferring file to an AWS edge location which will forward the data to the S3 bucket in the target region
- Compatible with multi-part upload



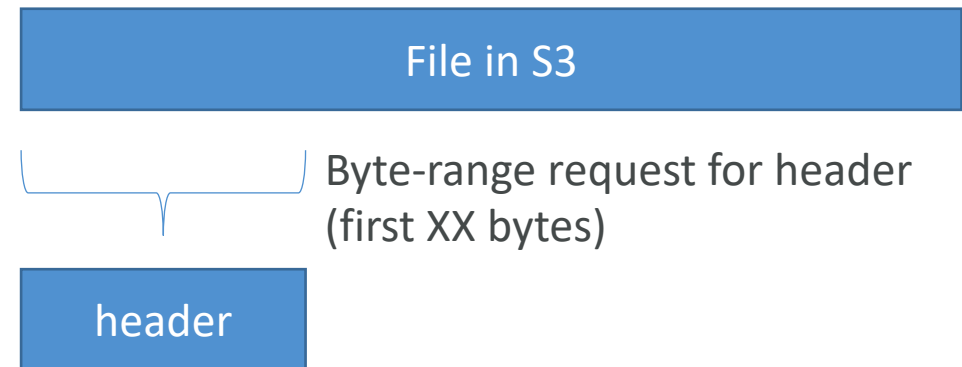
# S3 Performance – S3 Byte-Range Fetches

- Parallelize GETs by requesting specific byte ranges
- Better resilience in case of failures

Can be used to speed up downloads

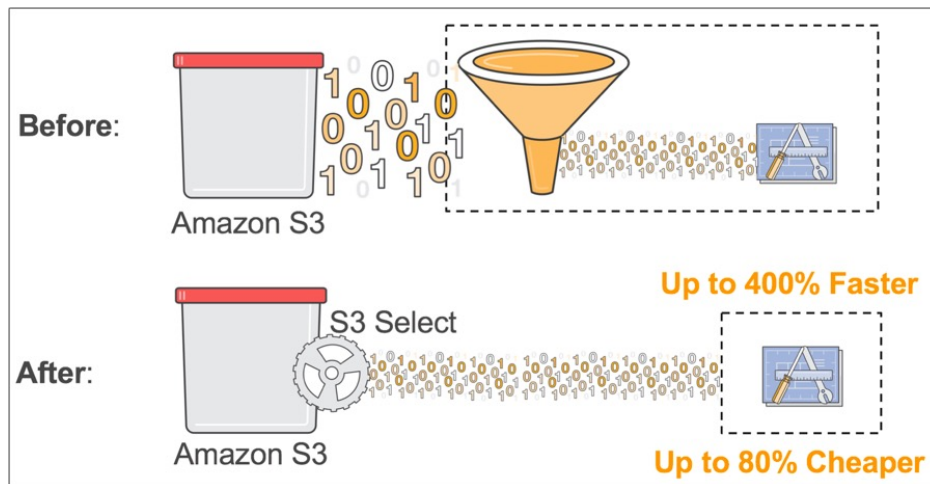


Can be used to retrieve only partial data (for example the head of a file)



# S3 Select & Glacier Select

- Retrieve less data using SQL by performing **server-side filtering**
- Can filter by rows & columns (simple SQL statements)
- Less network transfer, less CPU cost client-side



<https://aws.amazon.com/blogs/aws/s3-glacier-select/>

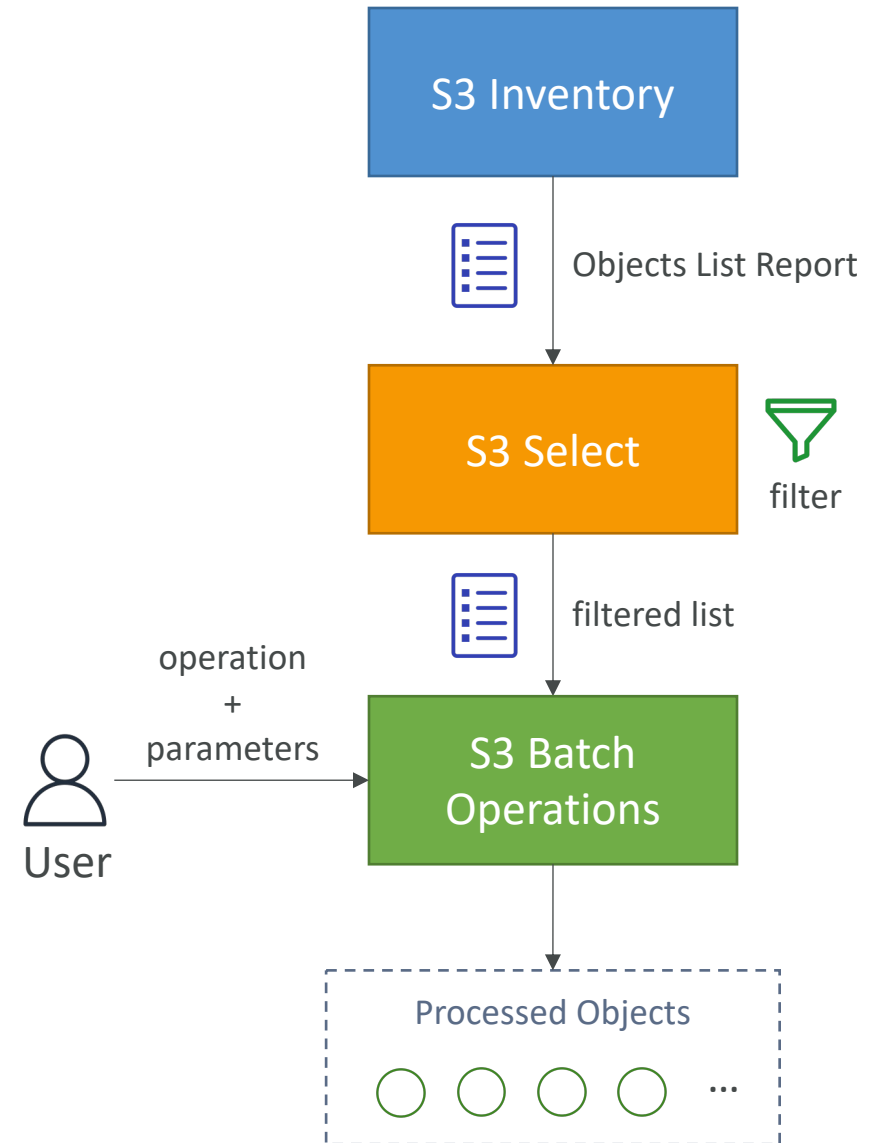


Server-side filtering



# S3 Batch Operations

- Perform bulk operations on existing S3 objects with a single request, example:
  - Modify object metadata & properties
  - Copy objects between S3 buckets
  - **Encrypt un-encrypted objects**
  - Modify ACLs, tags
  - Restore objects from S3 Glacier
  - Invoke Lambda function to perform custom action on each object
- A job consists of a list of objects, the action to perform, and optional parameters
- S3 Batch Operations manages retries, tracks progress, sends completion notifications, generate reports ...
- You can use S3 Inventory to get object list and use S3 Select to filter your objects



# S3 Inventory

- List objects and their corresponding metadata (alternative to S3 List API operation)
- Usage examples:
  - Audit and report on the replication and encryption status of your objects
  - Get the number of objects in an S3 bucket
  - Identify the total storage of previous object versions
- Generate daily or weekly reports
- Output files: CSV, ORC, or Apache Parquet
- You can query all the data using Amazon Athena, Redshift, Presto, Hive, Spark...
- You can filter generated report using S3 Select
- Use cases: Business, Compliance, Regulatory needs, ...



# Amazon S3 Glacier

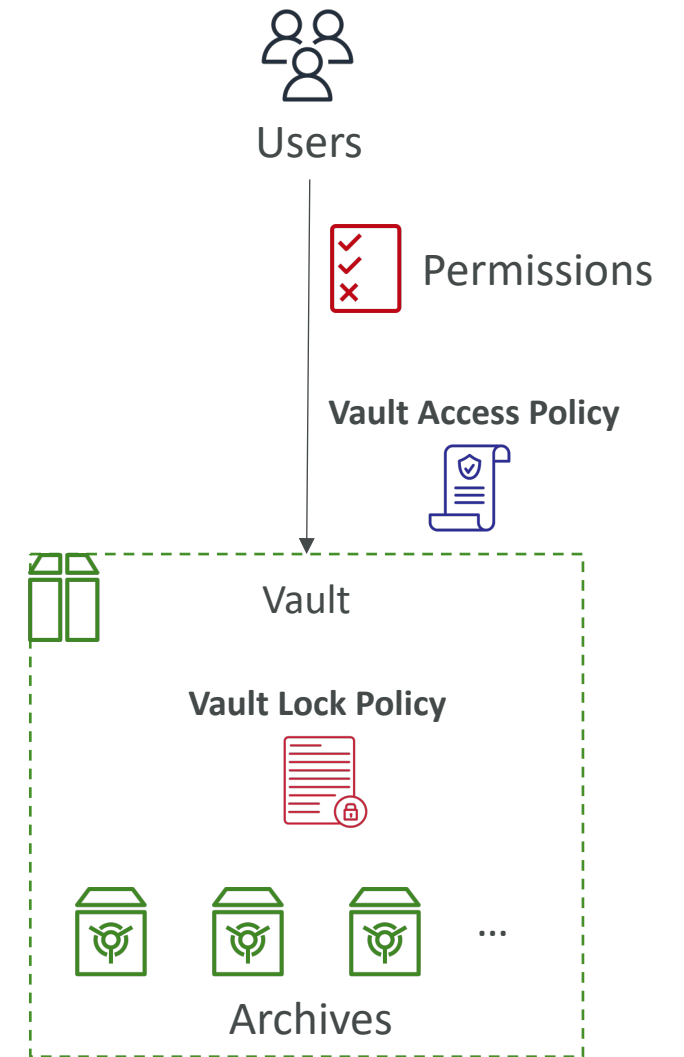
- Low-cost object storage meant for archiving / backup
- Data is retained for the longer term (10s of years)
- Alternative to on-premises magnetic tape storage
- Average annual durability is 99.999999999%
- Cost per storage per month (\$0.004 / GB – Standard | \$0.00099 / GB Deep Archive)
  
- Each item in Glacier is called “**Archive**” (up to 40TB)
- Archives are stored in “**Vaults**”
- By default, data encrypted at rest using AES-256 – keys managed by AWS
  
- Exam tip: archival from S3 after XXX days => use Glacier

# Amazon S3 Glacier Operations

- **Vault Operations:**
  - **Create & Delete** – delete only when there's no archives in it
  - **Retrieving Metadata** – creation date, number of archives, total size of all archives, ...
  - **Download Inventory** – list of archives in the vault (archive ID, creation date, size, ...)
- **Glacier Operations:**
  - **Upload** - single operation or by parts (MultiPart upload) for larger archives
  - **Download** - first initiate a retrieval job for the archive, Glacier then prepares it for download. User then has a limited time to download the data from staging server. (optionally, specify a range or portion of bytes to retrieve)
  - **Delete** - use Glacier Rest API or AWS SDKs by specifying archive ID
- Restore links have an expiry date
- **Retrieval Options:**
  - **Expedited (1 to 5 minutes retrieval)** – \$0.03 per GB and \$10 per 1000 requests
  - **Standard (3 to 5 hours)** - \$0.01 per GB and 0.03 per 1000 requests
  - **Bulk (5 to 12 hours)** - \$0.0025 per GB and \$0.025 per 1000 requests

# Amazon S3 Glacier - Vault Policies & Vault Lock

- Each Vault has:
  - ONE vault access policy
  - ONE vault lock policy
- Vault Policies are written in JSON
- Vault Access Policy is like a bucket policy (restrict user / account permissions)
- Vault Lock Policy is a policy you lock, for regulatory and compliance requirements.
  - The policy is immutable, it can never be changed (that's why it's call **LOCK**)
  - Example 1: forbid deleting an archive if less than 1 year old
  - Example 2: implement WORM policy (write once read many)



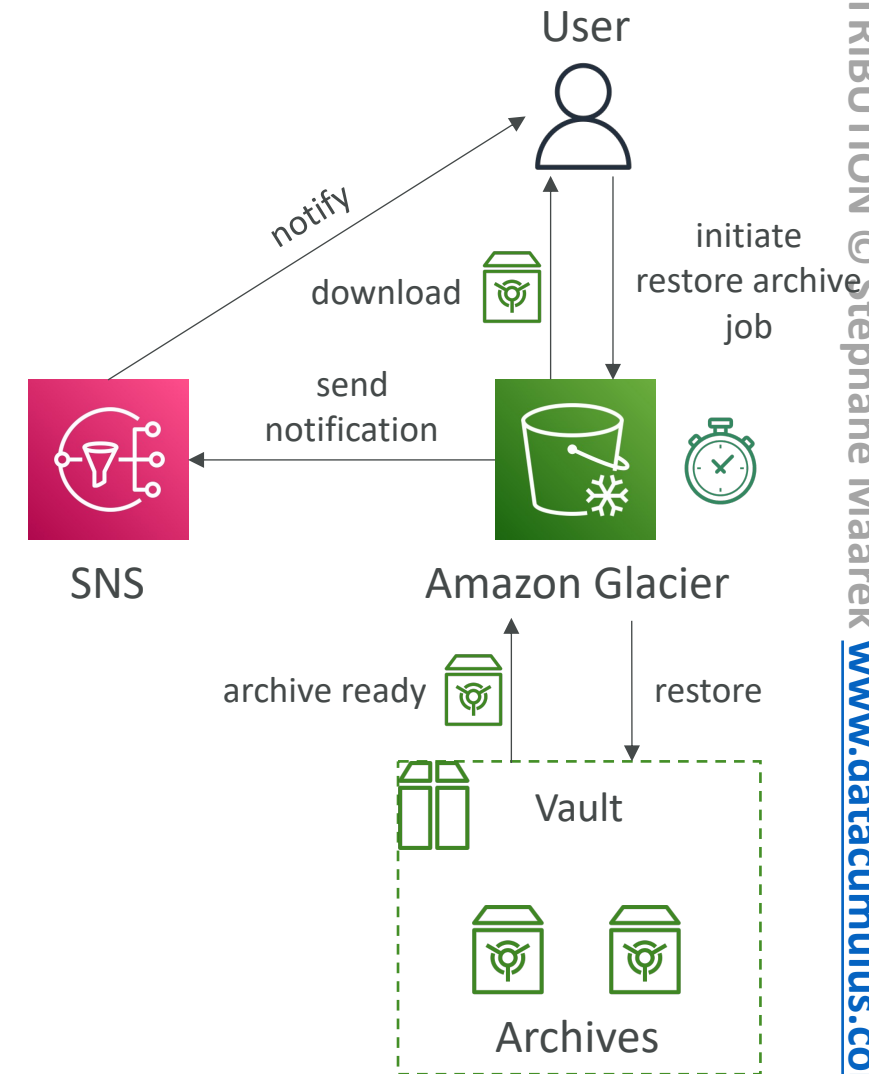
# Glacier – Notifications for Restore Operations

- Vault Notification Configuration

- Configure a vault so that when a job completes, a message is sent to SNS
- Optionally, specify an SNS topic when you initiate a job

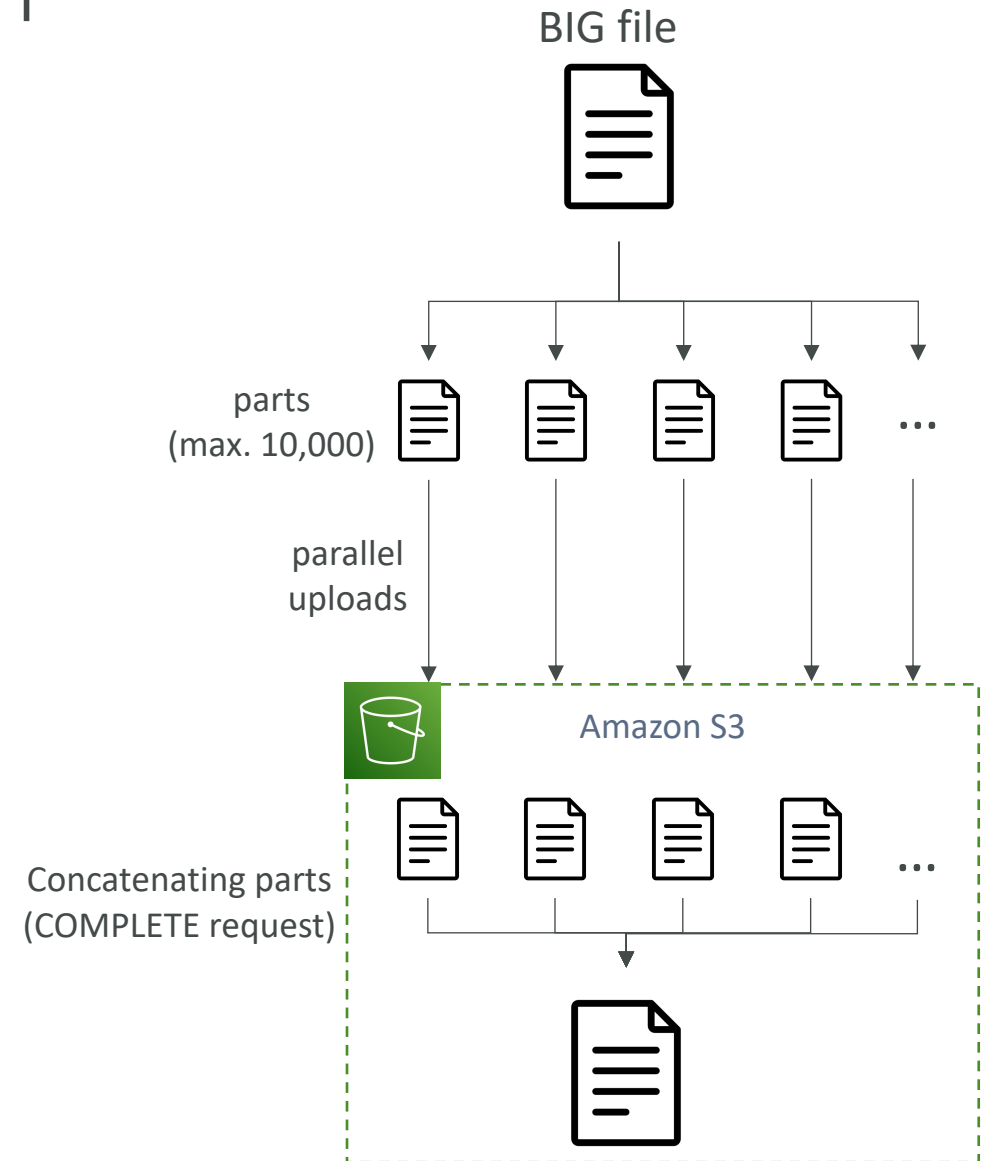
- S3 Event Notifications

- S3 supports the restoration of objects archived to S3 Glacier storage classes
- `s3:ObjectRestore:Post` => notify when object restoration initiated
- `s3:ObjectRestore:Completed` => notify when object restoration completed



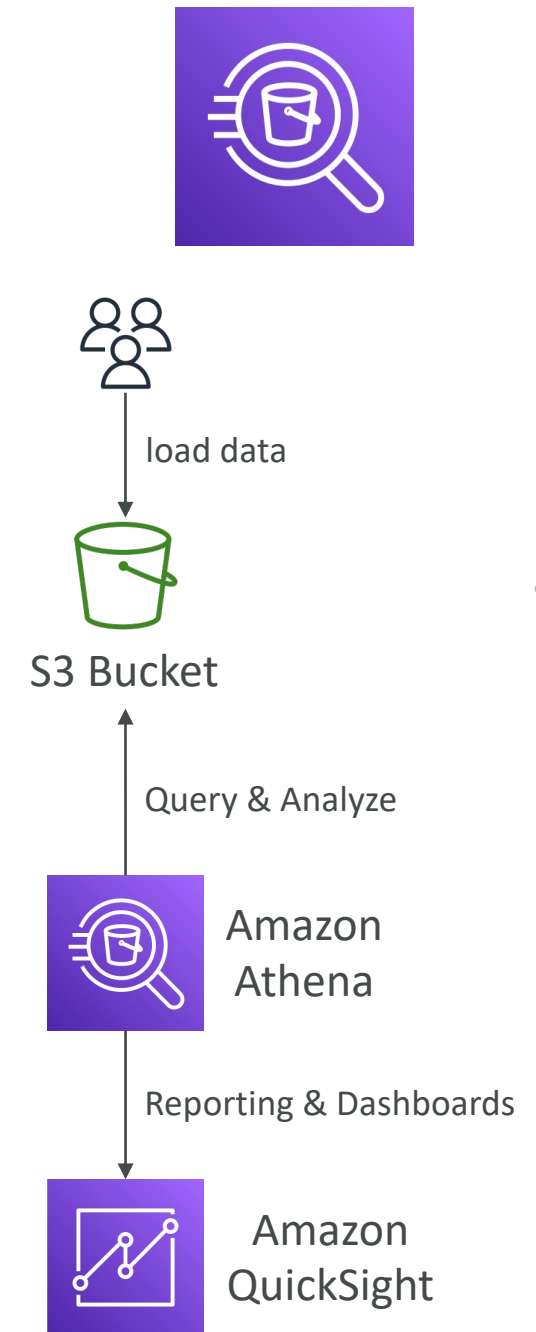
# S3 Multi Part Upload – Deep Dive

- Upload large objects in parts (any order)
- Recommended for files > 100MB, must use for files > 5GB
- Can help parallelize uploads (speed up transfers)
- Max. parts: 10,000
- **Failures:** restart uploading **ONLY** failed parts (improves performance)
- Use **Lifecycle Policy** to automate old parts **deletion** of unfinished upload after x days (e.g., network outage)
- Upload using AWS CLI or AWS SDK



# Amazon Athena

- **Serverless** query service to analyze data stored in Amazon S3
- Uses standard SQL language to query the files (built on Presto)
- Supports CSV, JSON, ORC, Avro, and Parquet
- Pricing: \$5.00 per TB of data scanned
- Commonly used with Amazon Quicksight for reporting/dashboards
- **Use cases:** Business intelligence / analytics / reporting, analyze & query VPC Flow Logs, ELB Logs, **CloudTrail trails**, etc...
- **Exam Tip:** analyze data in S3 using serverless SQL, use Athena





# Amazon Athena – Performance Improvement

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

# Amazon S3 Security

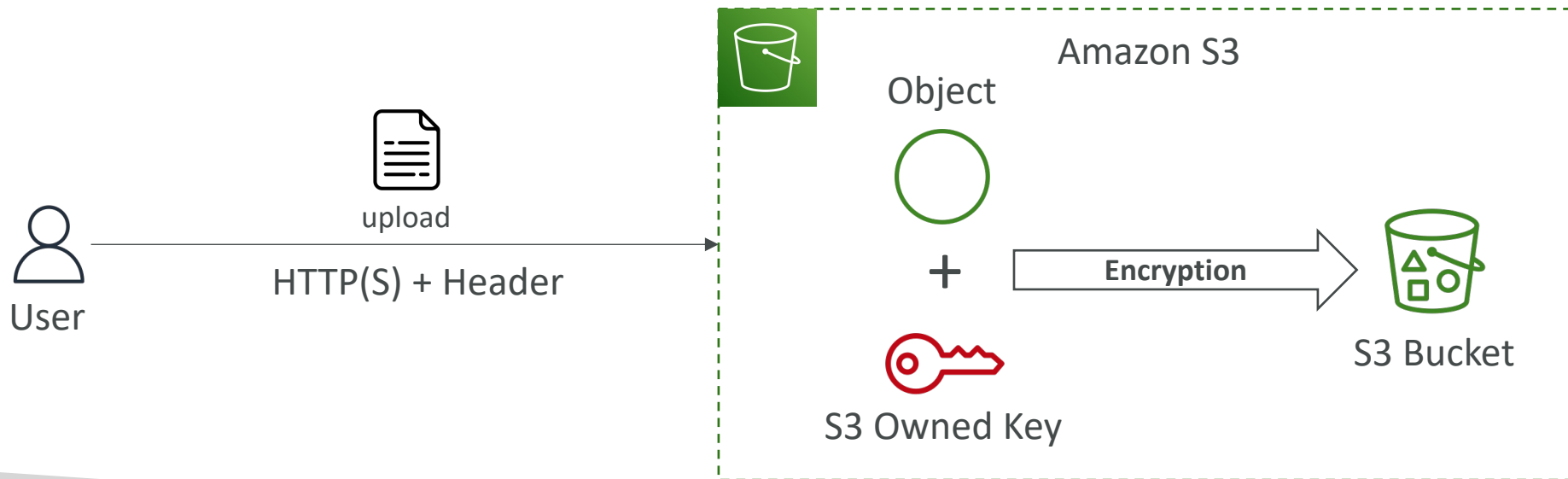


# Amazon S3 – Object Encryption

- You can encrypt objects in S3 buckets using one of 4 methods
- **Server-Side Encryption (SSE)**
  - **Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3) – Enabled by Default**
    - Encrypts S3 objects using keys handled, managed, and owned by AWS
  - **Server-Side Encryption with KMS Keys stored in AWS KMS (SSE-KMS)**
    - Leverage AWS Key Management Service (AWS KMS) to manage encryption keys
  - **Server-Side Encryption with Customer-Provided Keys (SSE-C)**
    - When you want to manage your own encryption keys
- **Client-Side Encryption**
- It's important to understand which ones are for which situation for the exam

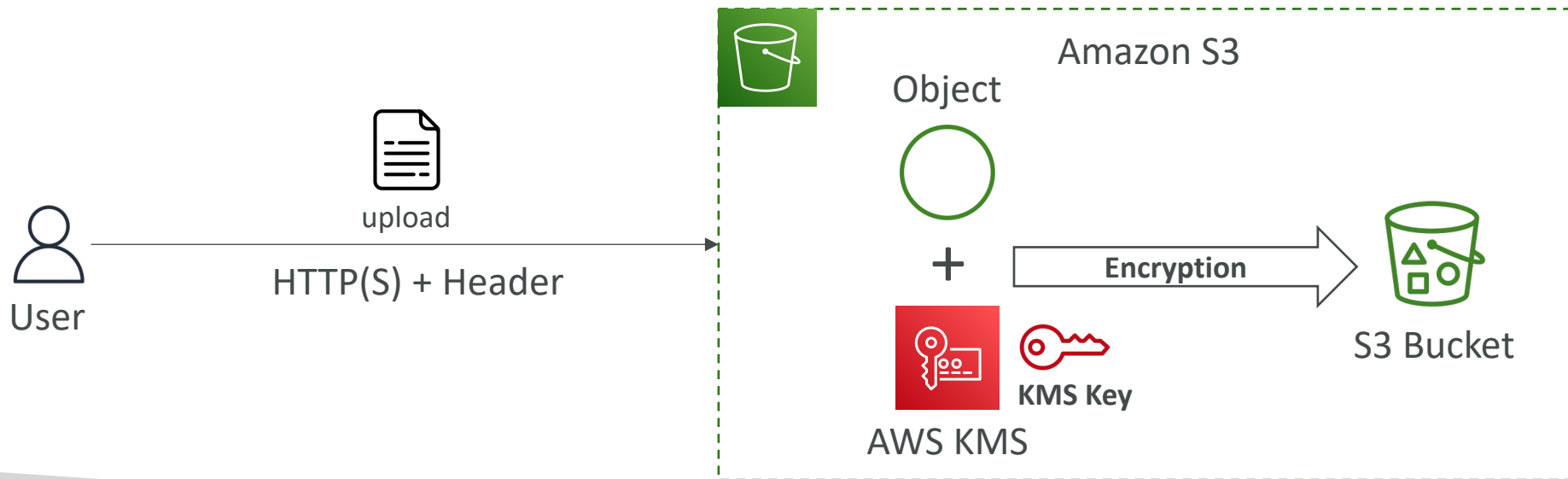
# Amazon S3 Encryption – SSE-S3

- Encryption using keys handled, managed, and owned by AWS
- Object is encrypted server-side
- Encryption type is **AES-256**
- Must set header "x-amz-server-side-encryption": "AES256"
- Enabled by default for new buckets & new objects



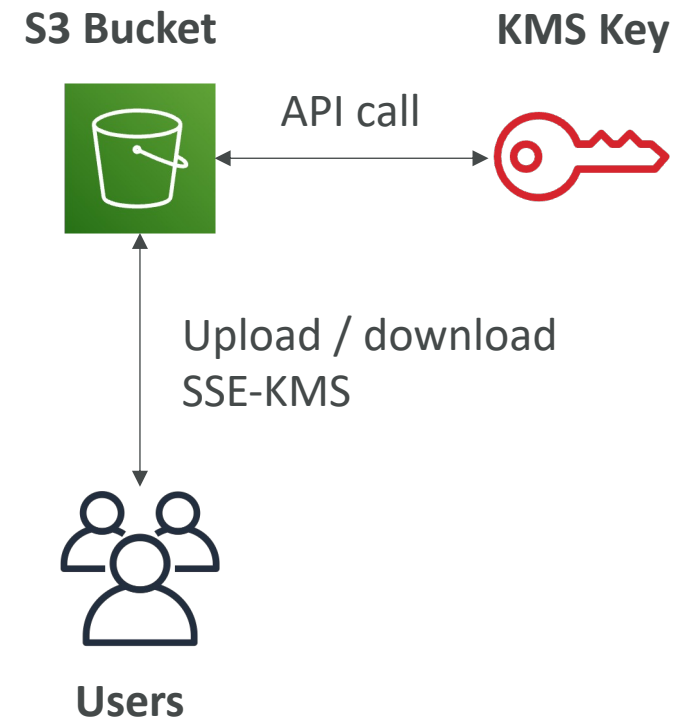
# Amazon S3 Encryption – SSE-KMS

- Encryption using keys handled and managed by AWS KMS (Key Management Service)
- KMS advantages: user control + audit key usage using CloudTrail
- Object is encrypted server side
- Must set header "x-amz-server-side-encryption": "aws:kms"



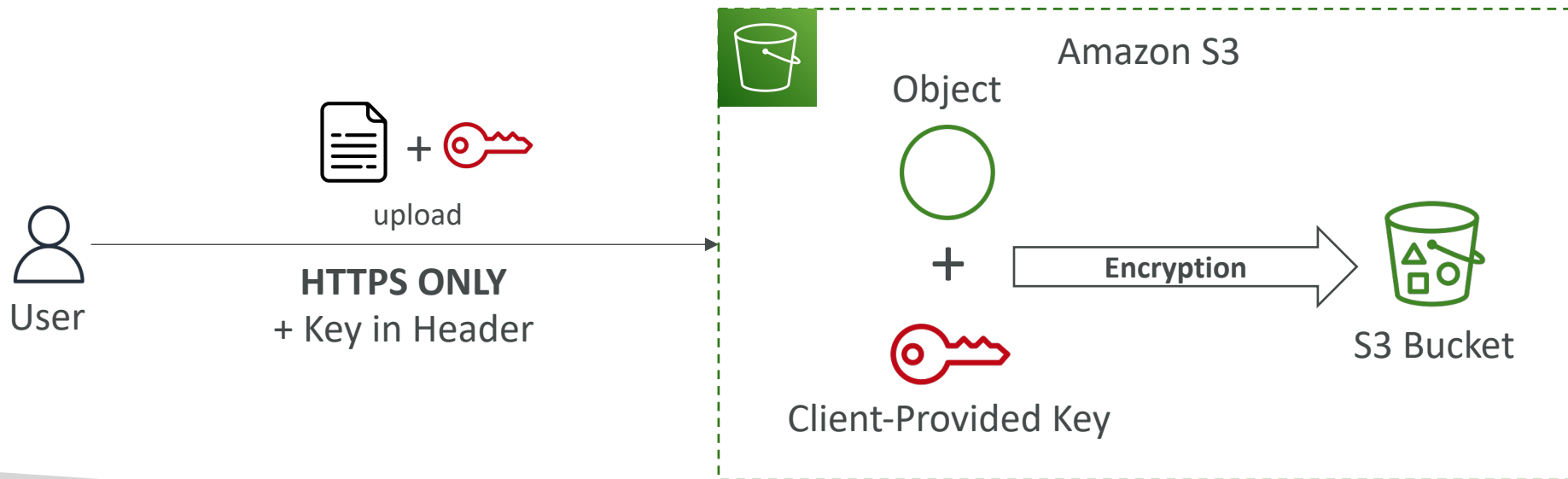
# SSE-KMS Limitation

- If you use SSE-KMS, you may be impacted by the KMS limits
- When you upload, it calls the **GenerateDataKey** KMS API
- When you download, it calls the **Decrypt** KMS API
- Count towards the KMS quota per second (5500, 10000, 30000 req/s based on region)
- You can request a quota increase using the Service Quotas Console



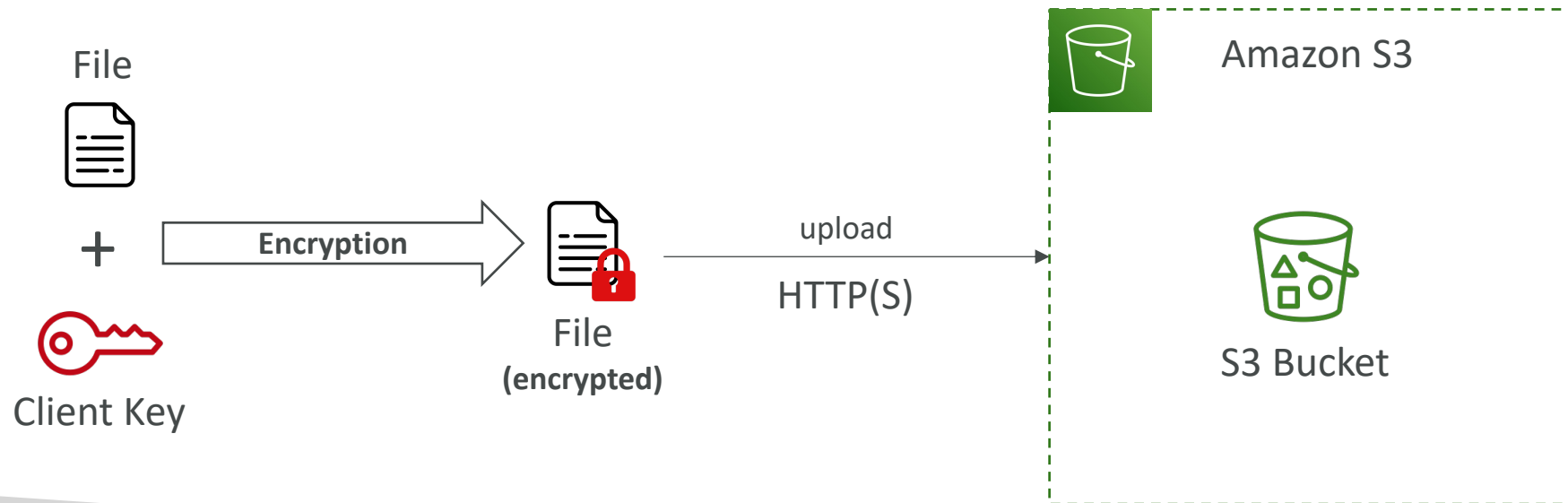
# Amazon S3 Encryption – SSE-C

- Server-Side Encryption using keys fully managed by the customer outside of AWS
- Amazon S3 does **NOT** store the encryption key you provide
- **HTTPS** must be used
- Encryption key must be provided in HTTP headers, for every HTTP request made



# Amazon S3 Encryption – Client-Side Encryption

- Use client libraries such as **Amazon S3 Client-Side Encryption Library**
- Clients must encrypt data themselves before sending to Amazon S3
- Clients must decrypt data themselves when retrieving from Amazon S3
- Customer fully manages the keys and encryption cycle





# Amazon S3 – Encryption in transit (SSL/TLS)

- Encryption in flight is also called SSL/TLS
- Amazon S3 exposes two endpoints:
  - HTTP Endpoint – non encrypted
  - HTTPS Endpoint – encryption in flight
- HTTPS is recommended
- HTTPS is mandatory for SSE-C
- Most clients would use the HTTPS endpoint by default



# 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"
        }
      }
    }
  ]
}
```

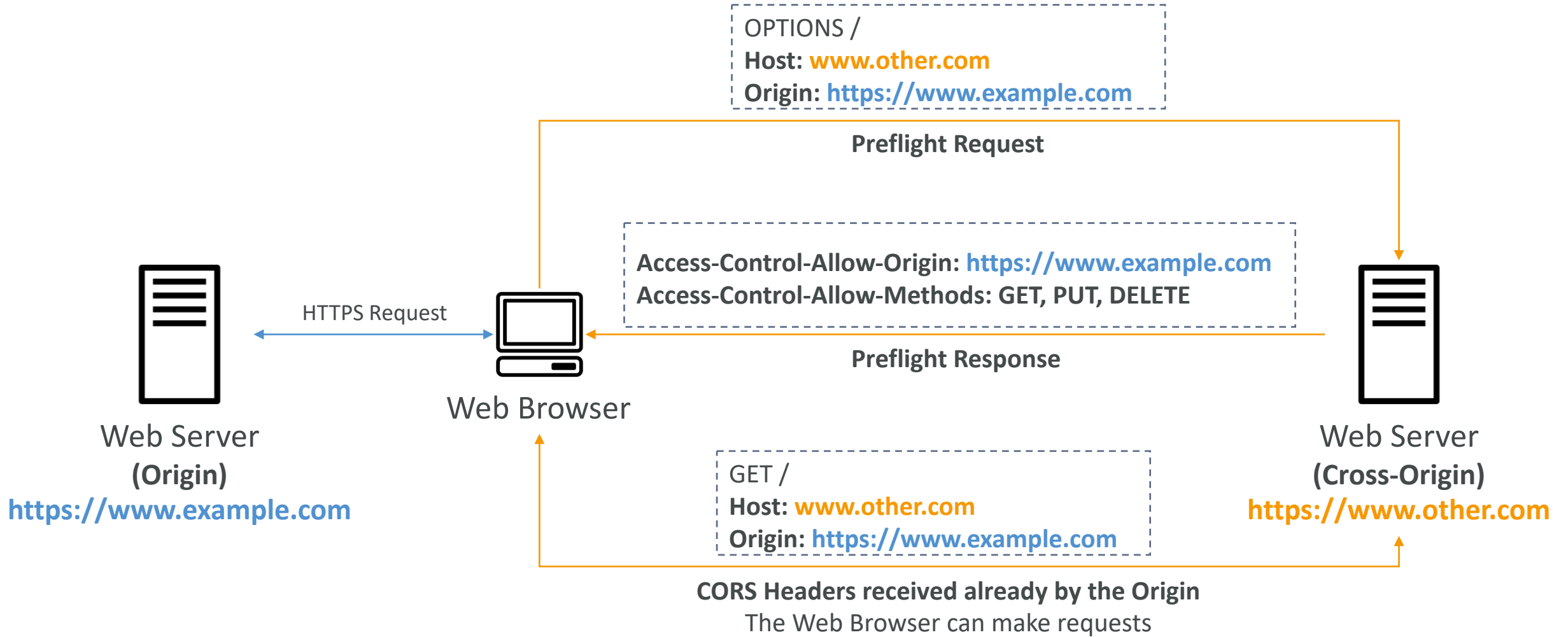
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "s3:PutObject",
      "Principal": "*",
      "Resource": "arn:aws:s3:::my-bucket/*",
      "Condition": {
        "Null": {
          "s3:x-amz-server-side-encryption-customer-algorithm": "true"
        }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

- Note: Bucket Policies are evaluated before “Default Encryption”

# What is CORS?

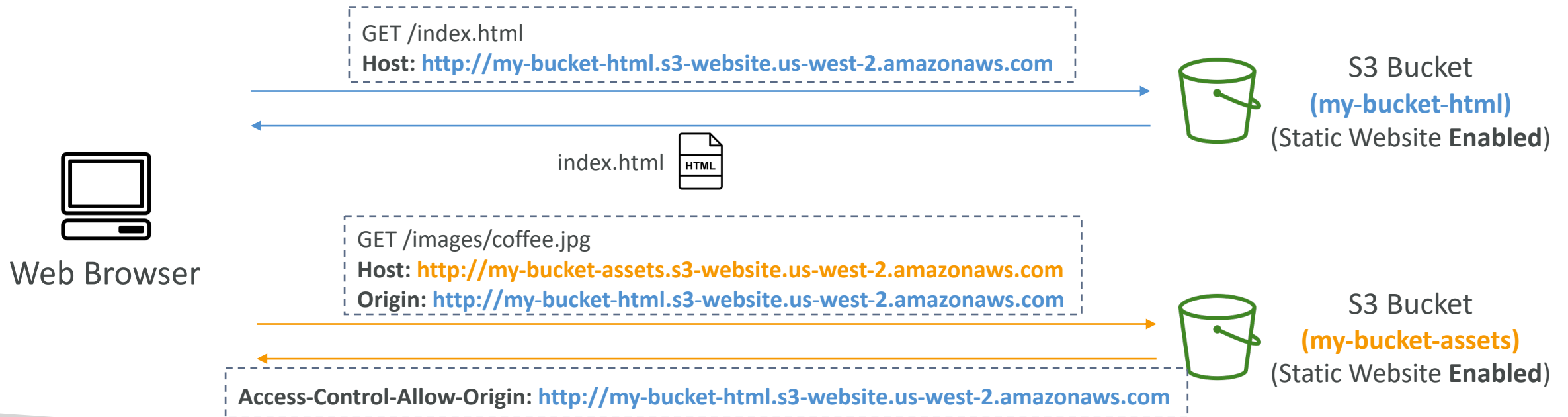
- Cross-Origin Resource Sharing (CORS)
- Origin = scheme (protocol) + host (domain) + port
  - example: <https://www.example.com> (implied port is 443 for HTTPS, 80 for HTTP)
- **Web Browser** based mechanism to allow requests to other origins while visiting the main origin
- Same origin: <http://example.com/app1> & <http://example.com/app2>
- Different origins: <http://www.example.com> & <http://other.example.com>
- The requests won't be fulfilled unless the other origin allows for the requests, using **CORS Headers** (example: **Access-Control-Allow-Origin**)

# What is CORS?



# Amazon S3 – CORS

- If a client makes a cross-origin request on our S3 bucket, we need to enable the correct CORS headers
- It's a popular exam question
- You can allow for a specific origin or for \* (all origins)



# Amazon S3 – MFA Delete

- **MFA (Multi-Factor Authentication)** – force users to generate a code on a device (usually a mobile phone or hardware) before doing important operations on S3
- MFA will be required to:
  - Permanently delete an object version
  - Suspend Versioning on the bucket
- MFA won't be required to:
  - Enable Versioning
  - List deleted versions
- To use MFA Delete, **Versioning must be enabled** on the bucket
- **Only the bucket owner (root account) can enable/disable MFA Delete**



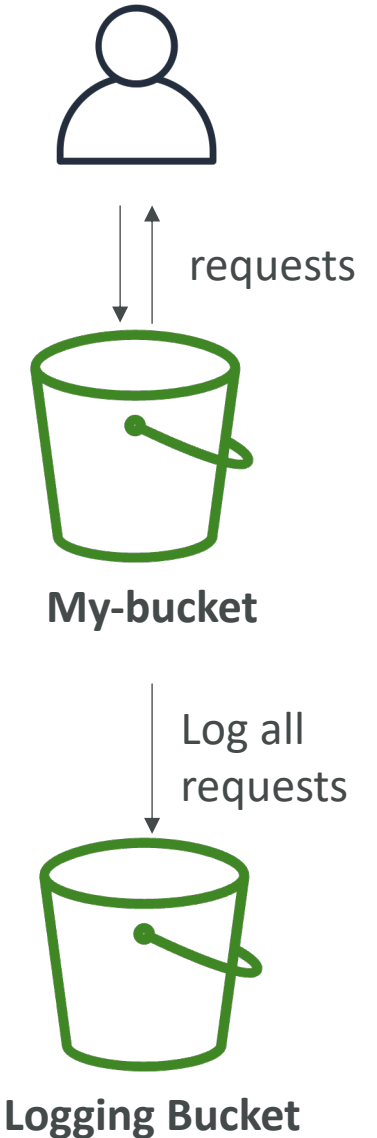
Google Authenticator



MFA Hardware Device

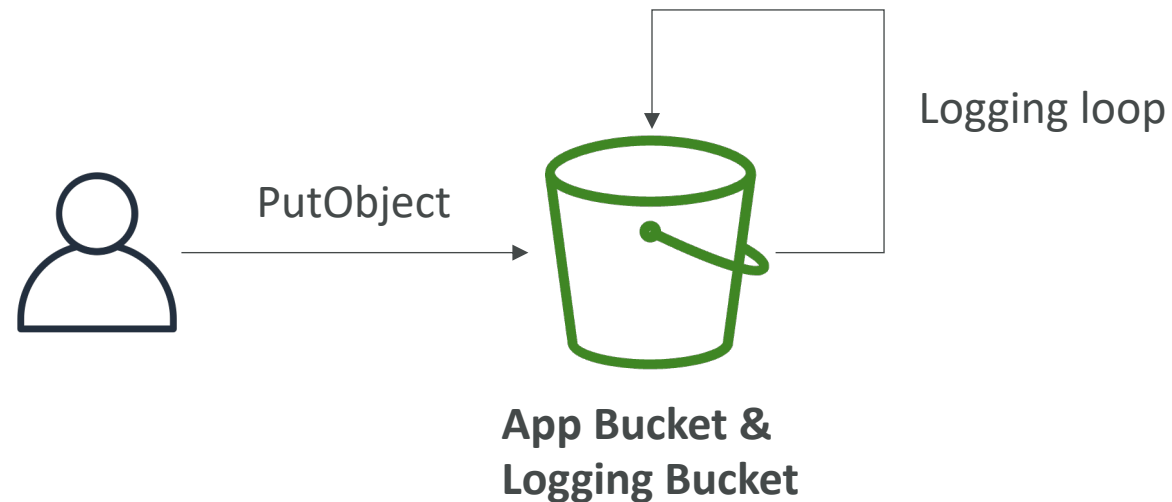
# S3 Access Logs

- For audit purpose, you may want to log all access to S3 buckets
- Any request made to S3, from any account, authorized or denied, will be logged into another S3 bucket
- That data can be analyzed using data analysis tools...
- The target logging bucket must be in the same AWS region
- The log format is at:  
<https://docs.aws.amazon.com/AmazonS3/latest/dev/LogFormat.html>



# S3 Access Logs: Warning

- Do not set your logging bucket to be the monitored bucket
- It will create a logging loop, and your bucket will grow exponentially

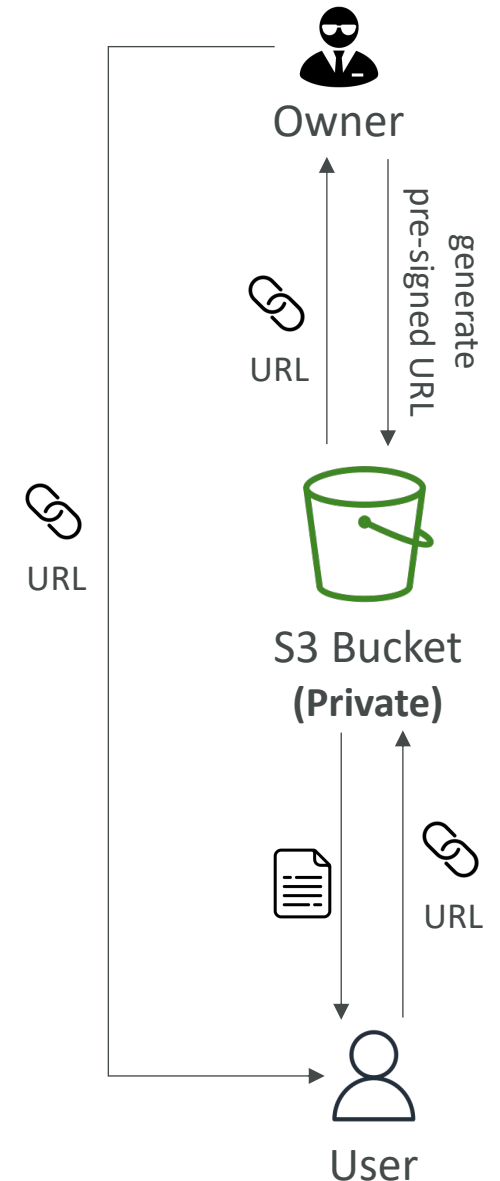


Do not try this at home 😊



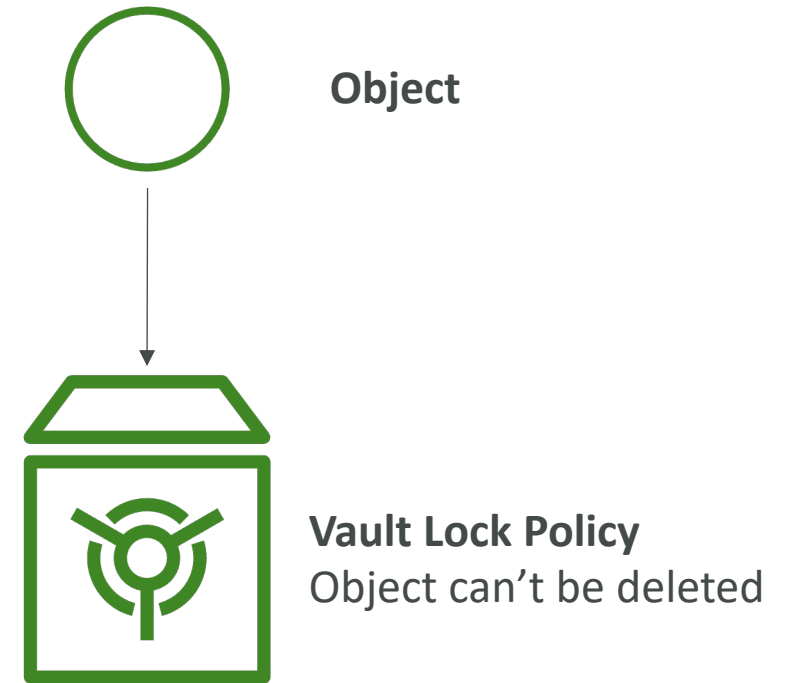
# Amazon S3 – Pre-Signed URLs

- Generate pre-signed URLs using the **S3 Console**, **AWS CLI** or **SDK**
- **URL Expiration**
  - **S3 Console** – 1 min up to 720 mins (12 hours)
  - **AWS CLI** – configure expiration with `--expires-in` parameter in seconds (default 3600 secs, max. 604800 secs ~ 168 hours)
- Users given a pre-signed URL inherit the permissions of the user that generated the URL for GET / PUT
- **Examples:**
  - Allow only logged-in users to download a premium video from your S3 bucket
  - Allow an ever-changing list of users to download files by generating URLs dynamically
  - Allow temporarily a user to upload a file to a precise location in your S3 bucket



# S3 Glacier Vault Lock

- Adopt a WORM (Write Once Read Many) model
- Create a Vault Lock Policy
- Lock the policy for future edits (can no longer be changed or deleted)
- Helpful for compliance and data retention



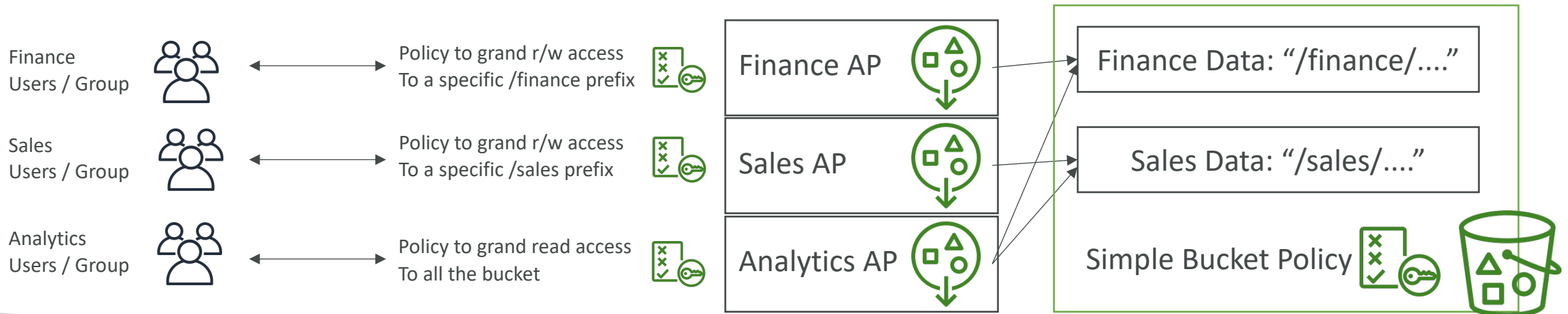
# S3 Object Lock (versioning must be enabled)

- Adopt a WORM (Write Once Read Many) model
- Block an object version deletion for a specified amount of time
- **Retention mode - Compliance:**
  - Object versions can't be overwritten or deleted by any user, including the root user
  - Objects retention modes can't be changed, and retention periods can't be shortened
- **Retention mode - Governance:**
  - Most users can't overwrite or delete an object version or alter its lock settings
  - Some users have special permissions to change the retention or delete the object
- **Retention Period:** protect the object for a fixed period, it can be extended
- **Legal Hold:**
  - protect the object indefinitely, independent from retention period
  - can be freely placed and removed using the *s3:PutObjectLegalHold* IAM permission

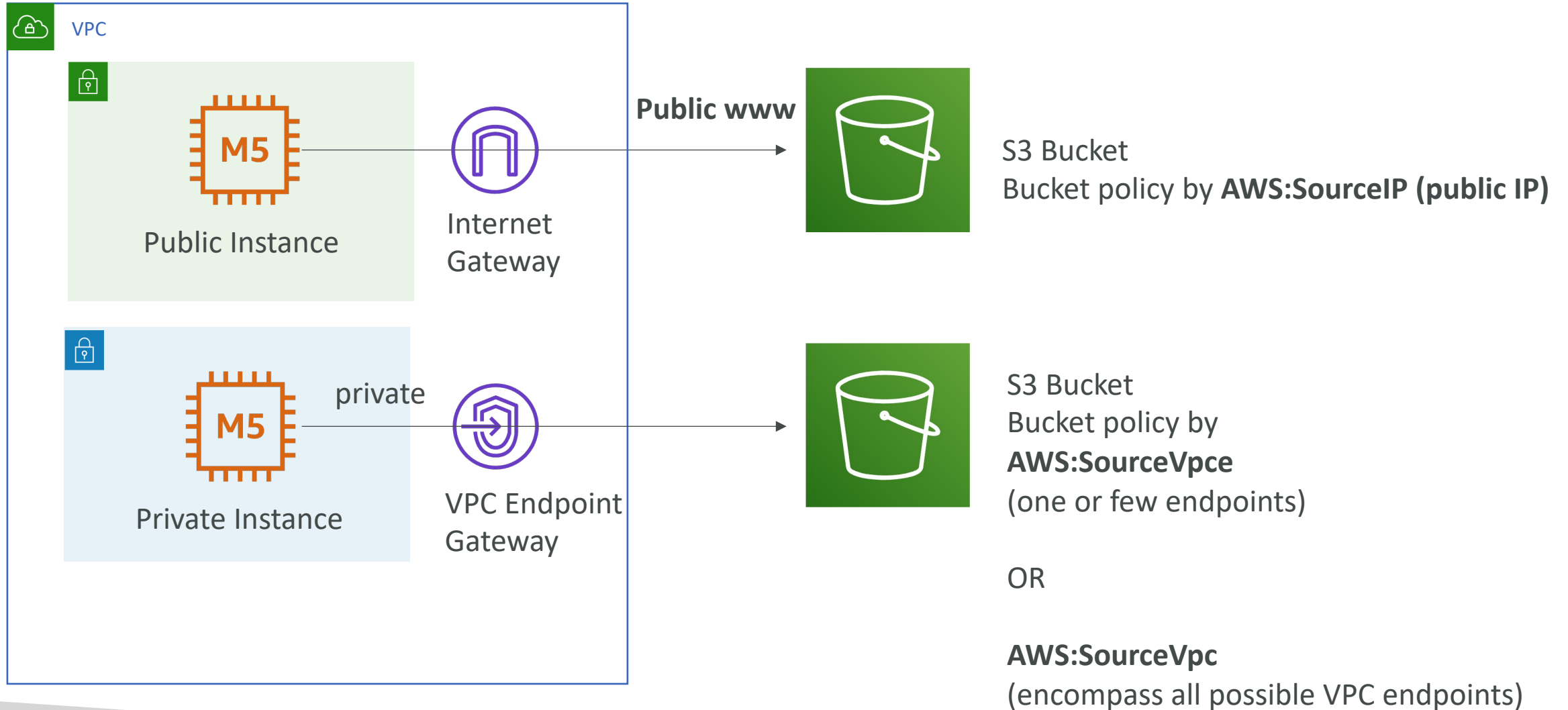
# S3 – Access Points



- Each Access Point gets its own DNS and policy to limit who can access it
  - A specific IAM user / group
  - One policy per Access Point => Easier to manage than complex bucket policies
- Can restrict to traffic from a specific VPC
- Access points are linked to a specific bucket (unique name per acct/region)



# VPC Endpoint Gateway for S3

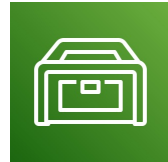


# Advanced Storage Solutions

# AWS Snow Family

- Highly-secure, portable devices to collect and process data at the edge, and migrate data into and out of AWS

- Data migration:



Snowcone



Snowball Edge



Snowmobile

- Edge computing:



Snowcone



Snowball Edge

# Data Migrations with AWS Snow Family

	Time to Transfer		
	100 Mbps	1Gbps	10Gbps
10 TB	12 days	30 hours	3 hours
100 TB	124 days	12 days	30 hours
1 PB	3 years	124 days	12 days

## Challenges:

- Limited connectivity
- Limited bandwidth
- High network cost
- Shared bandwidth (can't maximize the line)
- Connection stability

AWS Snow Family: offline devices to perform data migrations

If it takes more than a week to transfer over the network, use Snowball devices!

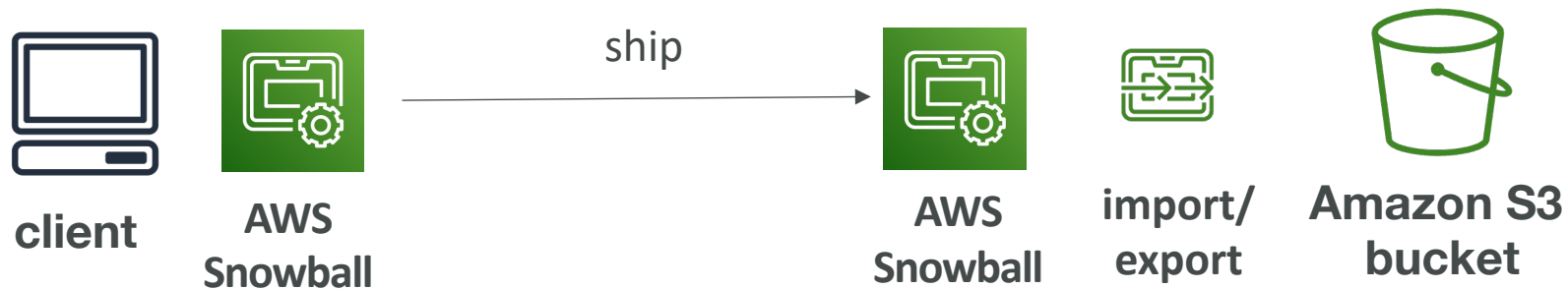


# Diagrams

- Direct upload to S3:



- With Snow Family:



# Snowball Edge (for data transfers)



- Physical data transport solution: move TBs or PBs of data in or out of AWS
- Alternative to moving data over the network (and paying network fees)
- Pay per data transfer job
- Provide block storage and Amazon S3-compatible object storage
- **Snowball Edge Storage Optimized**
  - 80 TB of HDD capacity for block volume and S3 compatible object storage
- **Snowball Edge Compute Optimized**
  - 42 TB of HDD capacity for block volume and S3 compatible object storage
- Use cases: large data cloud migrations, DC decommission, disaster recovery



# AWS Snowcone & Snowcone SSD



- Small, portable computing, anywhere, rugged & secure, withstands harsh environments
- Light (4.5 pounds, 2.1 kg)
- Device used for edge computing, storage, and data transfer
- Snowcone – 8 TB of HDD Storage
- Snowcone SSD – 14 TB of SSD Storage
- Use Snowcone where Snowball does not fit (space-constrained environment)
- Must provide your own battery / cables
  
- Can be sent back to AWS offline, or connect it to internet and use **AWS DataSync** to send data



# AWS Snowmobile



- Transfer exabytes of data (1 EB = 1,000 PB = 1,000,000 TBs)
- Each Snowmobile has 100 PB of capacity (use multiple in parallel)
- High security: temperature controlled, GPS, 24/7 video surveillance
- Better than Snowball if you transfer more than 10 PB

# 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 usable	< 100 PB
Migration Size	Up to 24 TB, online and offline	Up to petabytes, offline	Up to exabytes, offline
DataSync agent	Pre-installed		
Storage Clustering		Up to 15 nodes	

# Snow Family – Usage Process

1. Request Snowball devices from the AWS console for delivery
2. Install the snowball client / AWS OpsHub on your servers
3. Connect the snowball to your servers and copy files using the client
4. Ship back the device when you're done (goes to the right AWS facility)
5. Data will be loaded into an S3 bucket
6. Snowball is completely wiped

# What is Edge Computing?

- Process data while it's being created on an **edge location**
  - A truck on the road, a ship on the sea, a mining station underground...



- These locations may have
  - Limited / no internet access
  - Limited / no easy access to computing power
- We setup a **Snowball Edge / Snowcone** device to do edge computing
- Use cases of Edge Computing:
  - Preprocess data
  - Machine learning at the edge
  - Transcoding media streams
- Eventually (if need be) we can ship back the device to AWS (for transferring data for example)

# Snow Family – Edge Computing

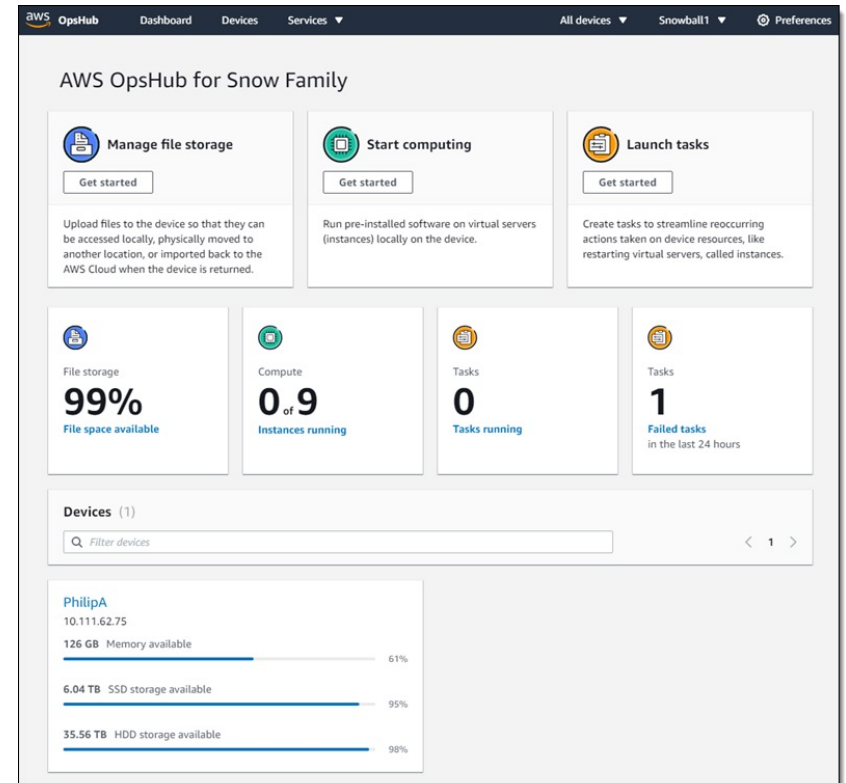
- **Snowcone & Snowcone SSD (smaller)**
  - 2 CPUs, 4 GB of memory, wired or wireless access
  - USB-C power using a cord or the optional battery
- **Snowball Edge – Compute Optimized**
  - 52 vCPUs, 208 GiB of RAM
  - Optional GPU (useful for video processing or machine learning)
  - 42 TB usable storage
- **Snowball Edge – Storage Optimized**
  - Up to 40 vCPUs, 80 GiB of RAM
  - Object storage clustering available
- All: Can run EC2 Instances & AWS Lambda functions (using AWS IoT Greengrass)
- Long-term deployment options: 1 and 3 years discounted pricing





# 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/>

# Amazon FSx – Overview



- Launch 3rd party high-performance file systems on AWS
- Fully managed service



FSx for Lustre



FSx for Windows  
File Server



FSx for  
NetApp ONTAP



FSx for  
OpenZFS

# 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

# 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)**

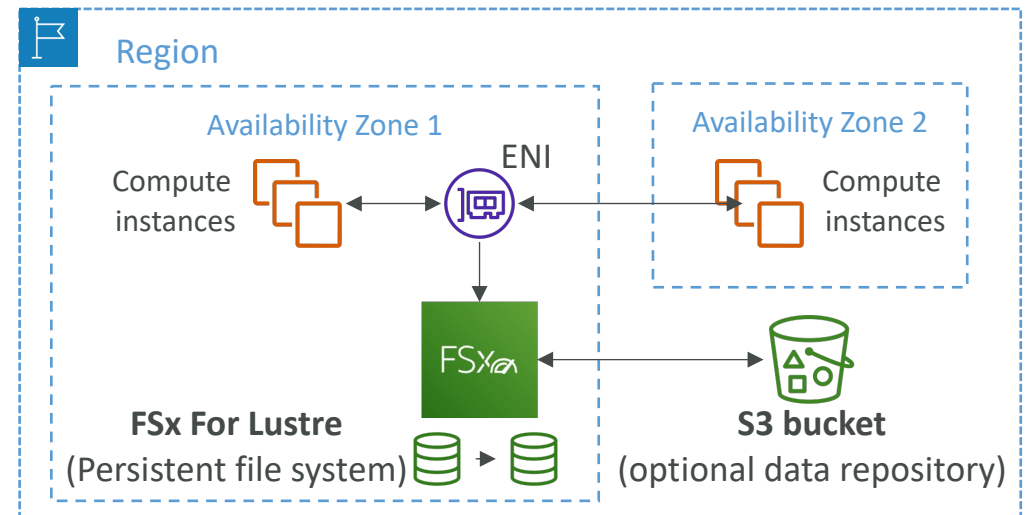
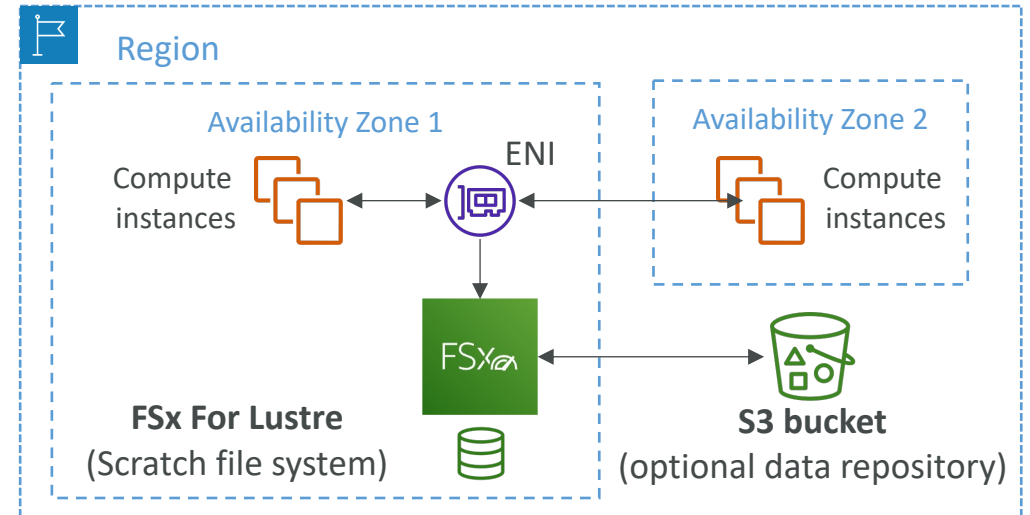
# FSx Lustre - File System Deployment Options

- **Scratch File System**

- Temporary storage
- Data is not replicated (doesn't persist if file server fails)
- High burst (6x faster, 200MBps per TiB)
- Usage: short-term processing, optimize costs

- **Persistent File System**

- Long-term storage
- Data is replicated within same AZ
- Replace failed files within minutes
- Usage: long-term processing, sensitive data



# Amazon FSx for NetApp ONTAP

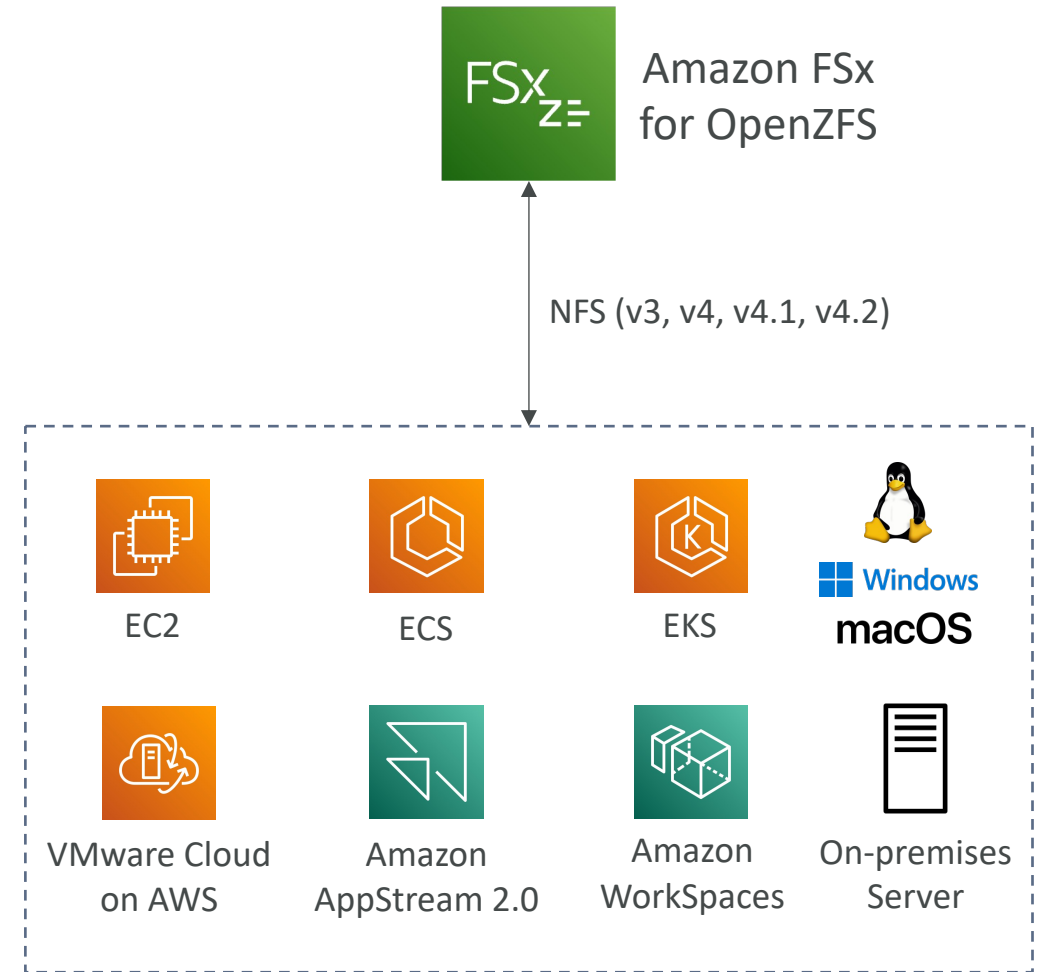


- Managed NetApp ONTAP on AWS
- File System compatible with NFS, SMB, iSCSI protocol
- Move workloads running on ONTAP or NAS to AWS
- Works with:
  - Linux
  - Windows
  - MacOS
  - VMware Cloud on AWS
  - Amazon Workspaces & AppStream 2.0
  - Amazon EC2, ECS and EKS
- Storage shrinks or grows automatically
- Snapshots, replication, low-cost, compression and data de-duplication
- Point-in-time instantaneous cloning (helpful for testing new workloads)



# Amazon FSx for OpenZFS

- Managed OpenZFS file system on AWS
- File System compatible with NFS (v3, v4, v4.1, v4.2)
- Move workloads running on ZFS to AWS
- Works with:
  - Linux
  - Windows
  - MacOS
  - VMware Cloud on AWS
  - Amazon Workspaces & AppStream 2.0
  - Amazon EC2, ECS and EKS
- Up to 1,000,000 IOPS with < 0.5ms latency
- Snapshots, compression and low-cost
- Point-in-time instantaneous cloning (helpful for testing new workloads)

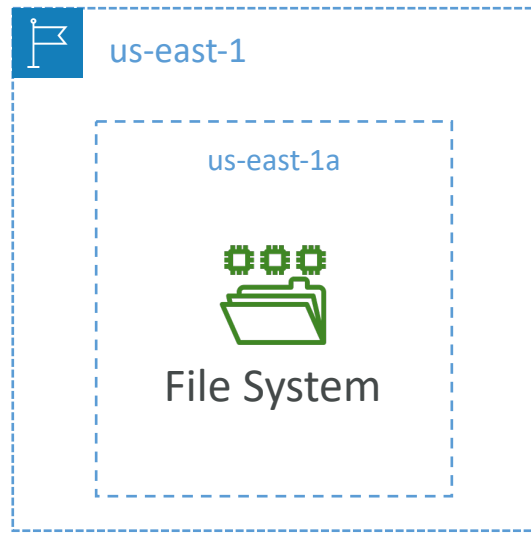


# FSx for SysOps



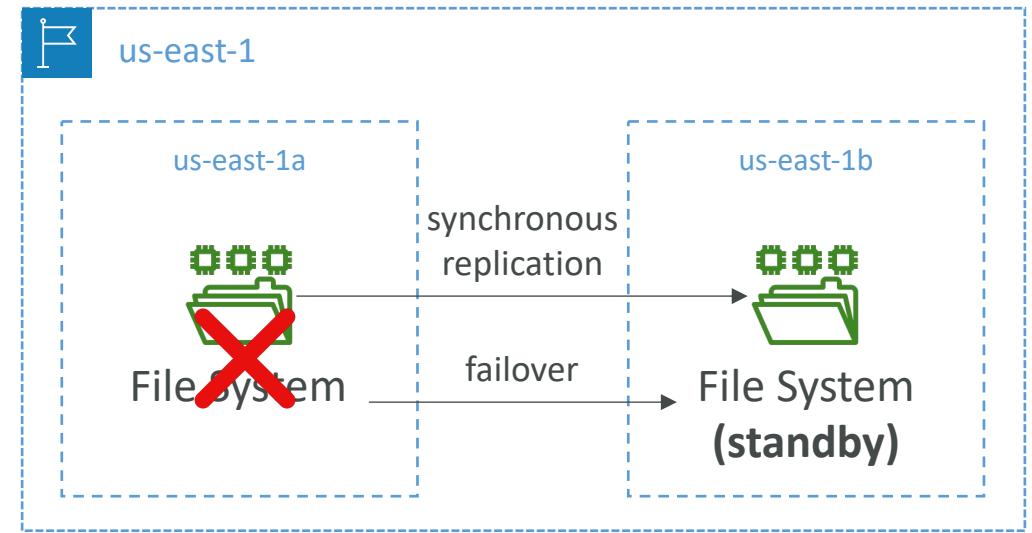
- FSx for Windows – Single-AZ

- Automatically replicates data within an AZ
- Two generations: Single-AZ 1 (SSD), Single-AZ 2 (SSD & HDD)



- FSx for Windows – Multi-AZ

- Automatically replicates data across AZs (synchronous)
- Standby file server in a different AZ (automatic failover)



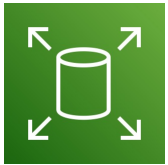


# Hybrid Cloud for Storage

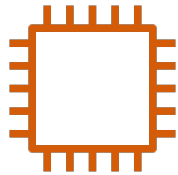
- AWS is pushing for "hybrid cloud"
  - Part of your infrastructure is on the cloud
  - Part of your infrastructure is on-premises
- This can be due to
  - Long cloud migrations
  - Security requirements
  - Compliance requirements
  - IT strategy
- S3 is a proprietary storage technology (unlike EFS / NFS), so how do you expose the S3 data on-premises?
- AWS Storage Gateway!

# AWS Storage Cloud Native Options

## Block



Amazon EBS



EC2 Instance Store

## File



Amazon EFS



Amazon FSx

## Object



Amazon S3

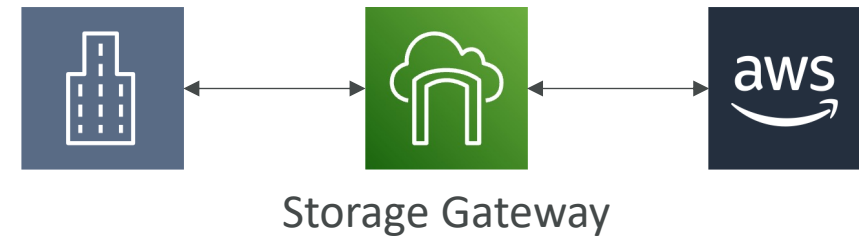


Amazon Glacier

# AWS Storage Gateway

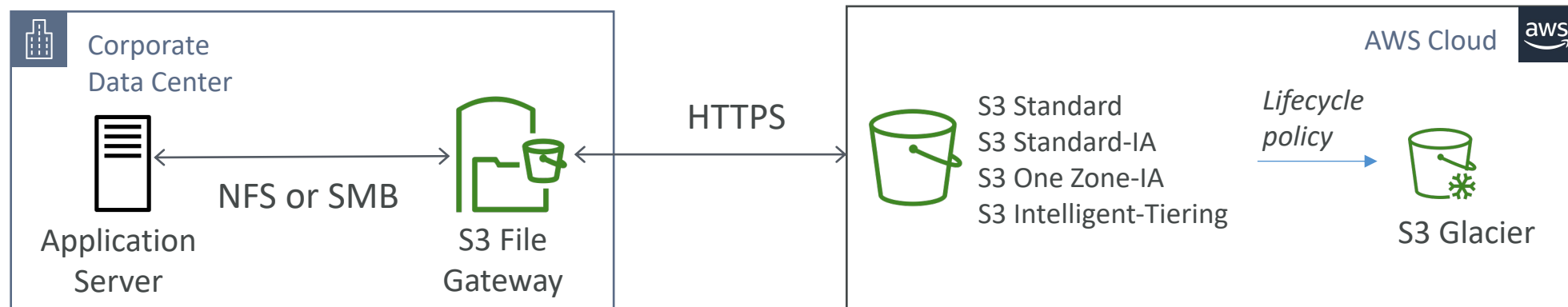


- Bridge between on-premises data and cloud data
- Use cases:
  - disaster recovery
  - backup & restore
  - tiered storage
  - on-premises cache & low-latency files access
- Types of Storage Gateway:
  - S3 File Gateway
  - FSx File Gateway
  - Volume Gateway
  - Tape Gateway



# Amazon S3 File Gateway

- Configured S3 buckets are accessible using the NFS and SMB protocol
- Most recently used data is cached in the file gateway
- Supports S3 Standard, S3 Standard IA, S3 One Zone A, S3 Intelligent Tiering
- Transition to S3 Glacier using a Lifecycle Policy
- Bucket access using IAM roles for each File Gateway
- SMB Protocol has integration with Active Directory (AD) for user authentication



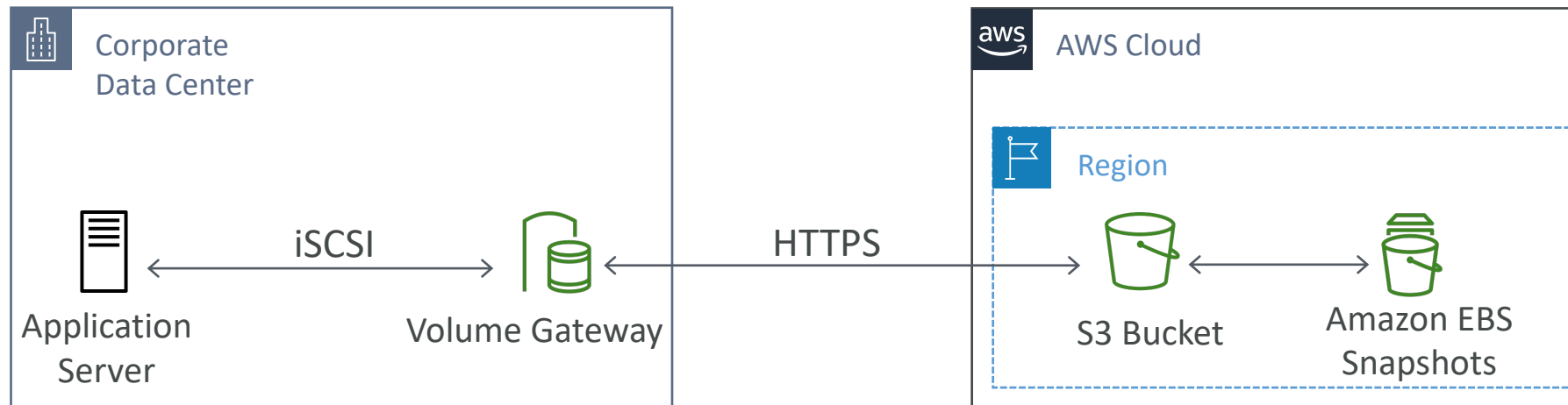
# Amazon FSx File Gateway

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



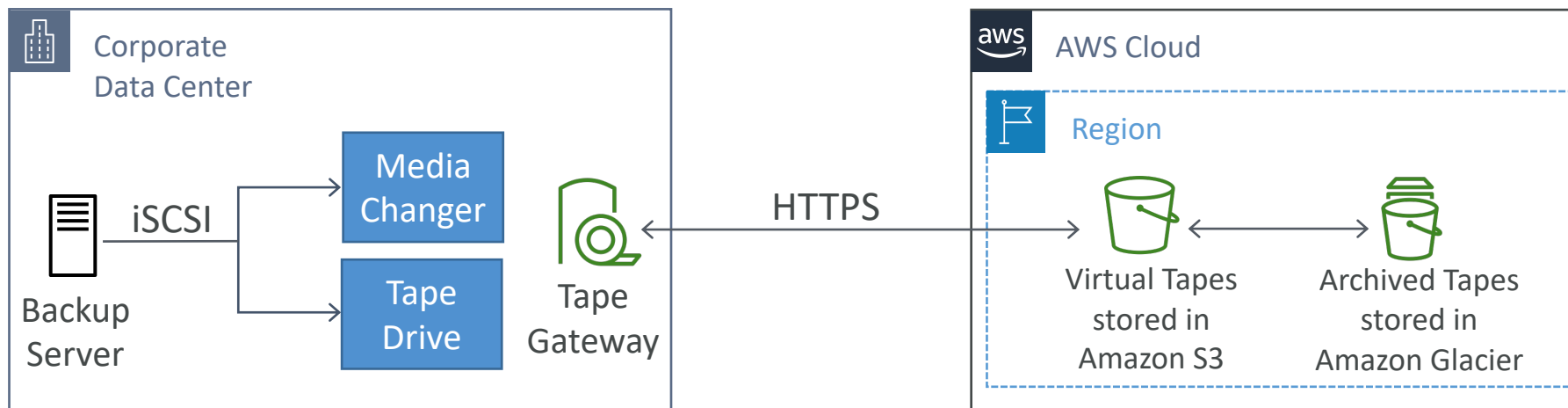
# Volume Gateway

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



# Tape Gateway

- Some companies have backup processes using physical tapes (!)
- With Tape Gateway, companies use the same processes but, in the cloud
- Virtual Tape Library (VTL) backed by Amazon S3 and Glacier
- Back up data using existing tape-based processes (and iSCSI interface)
- Works with leading backup software vendors



# Storage Gateway – Hardware appliance

- Using Storage Gateway means you need on-premises virtualization
- Otherwise, you can use a **Storage Gateway Hardware Appliance**
- You can buy it on amazon.com
- Works with File Gateway, Volume Gateway, Tape Gateway
- Has the required CPU, memory, network, SSD cache resources
- Helpful for daily NFS backups in small data centers

## Select host platform

- VMware ESXi
- Microsoft Hyper-V 2012R2/2016
- Linux KVM
- Amazon EC2
- Hardware Appliance**

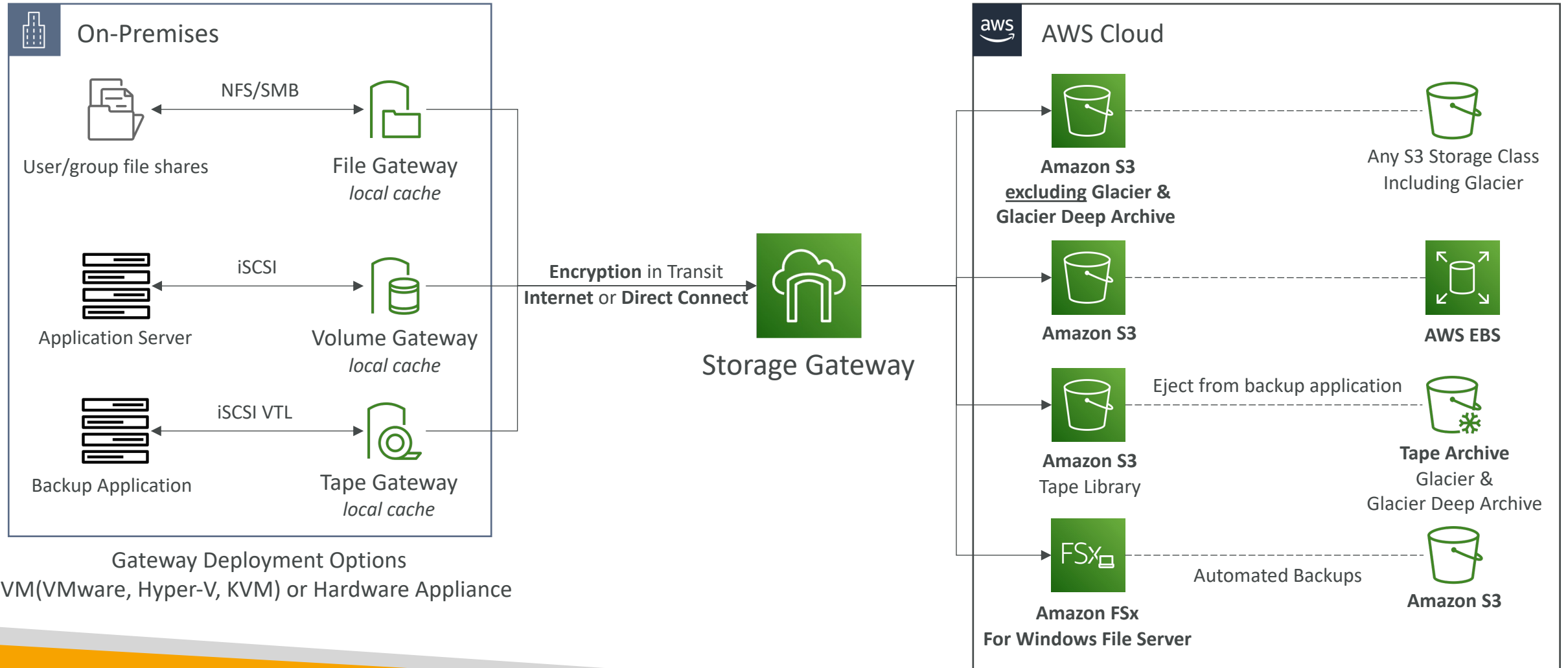
[Buy on Amazon](#)

[Activate Appliance](#)





# AWS Storage Gateway



# Amazon FSx File Gateway

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



# Storage Gateway – SysOps

- File Gateway is POSIX compliant (Linux file system)
  - POSIX metadata ownership, permissions, and timestamps stored in the object's metadata in S3
- Reboot Storage Gateway VM: (e.g., maintenance)
  - **File Gateway:** simply restart the Storage Gateway VM
  - **Volume and Tape Gateway:**
    - Stop Storage Gateway Service (AWS Console, VM local Console, Storage Gateway API)
    - Reboot the Storage Gateway VM
    - Start Storage Gateway Service (AWS Console, VM local Console, Storage Gateway API)

# Storage Gateway – Activations

- Two ways to get Activation Key:
  - Using the Gateway VM CLI
  - Make a web request to the Gateway VM (Port 80)
- Troubleshooting Activation Failures:
  - Make sure the Gateway VM has port 80 opened
  - Check that the Gateway VM has the correct time and synchronizing its time automatically to a Network Time Protocol (NTP) server

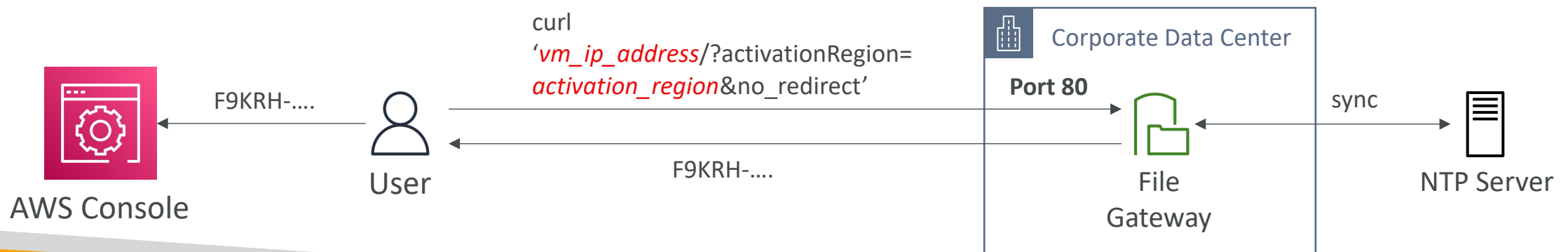
```
AWS Appliance Activation - Configuration
#####
## Currently connected network adapters:
##
## eth0:
#####

1: HTTP/SOCKS Proxy Configuration
2: Network Configuration
3: Test Network Connectivity
4: View System Resource Check (0 Errors)
5: License Information
6: Command Prompt

0: Get activation key

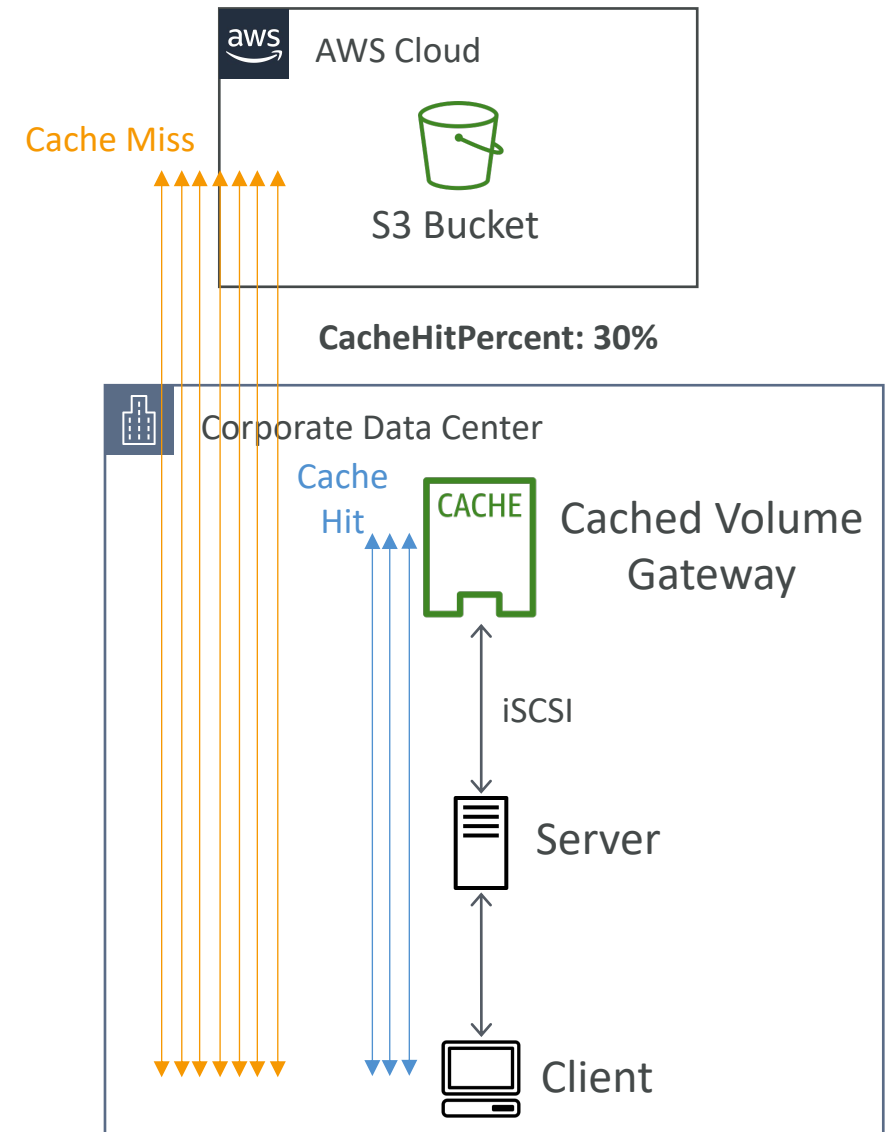
Press "x" to exit session

Enter command: █
```



# Storage Gateway – Volume Gateway Cache

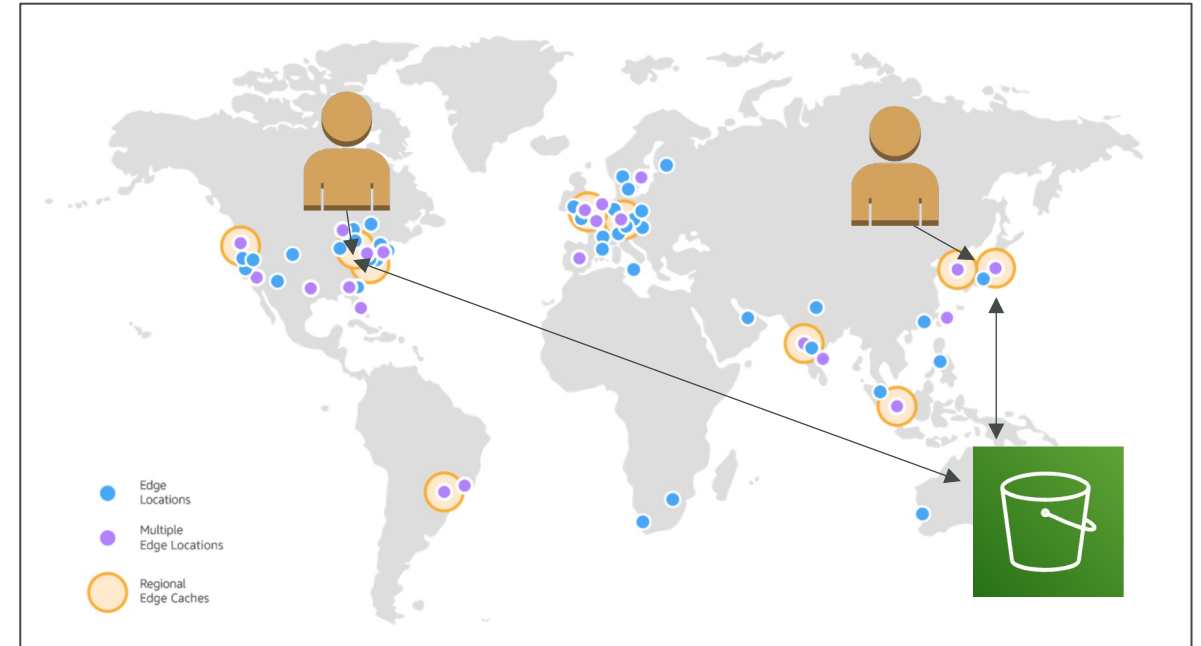
- Cached mode: only the most recent data is stored
- Looking at cache efficiency
  - Look at the CacheHitPercent metric (you want it to be high)
  - Look at the CachePercentUsed (you don't want it to be too high)
- Create a larger cache disk
  - Use the cached volume to clone a new volume of a larger size
  - Select the new disk as the cached volume



# CloudFront

# Amazon CloudFront

- Content Delivery Network (CDN)
- Improves read performance, content is cached at the edge
- Improves users experience
- 216 Point of Presence globally (edge locations)
- DDoS protection (because worldwide), integration with Shield, AWS Web Application Firewall



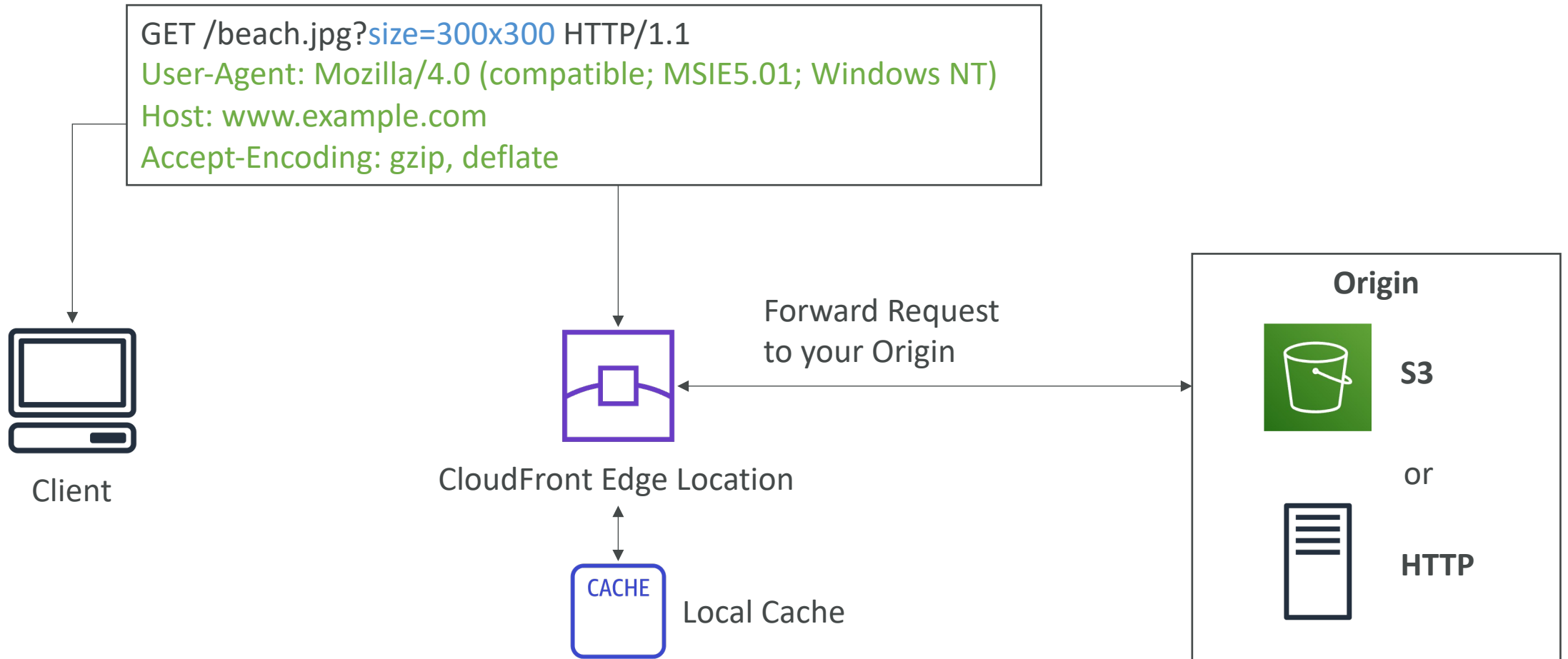
Source: <https://aws.amazon.com/cloudfront/features/?nc=sn&loc=2>

# CloudFront – Origins

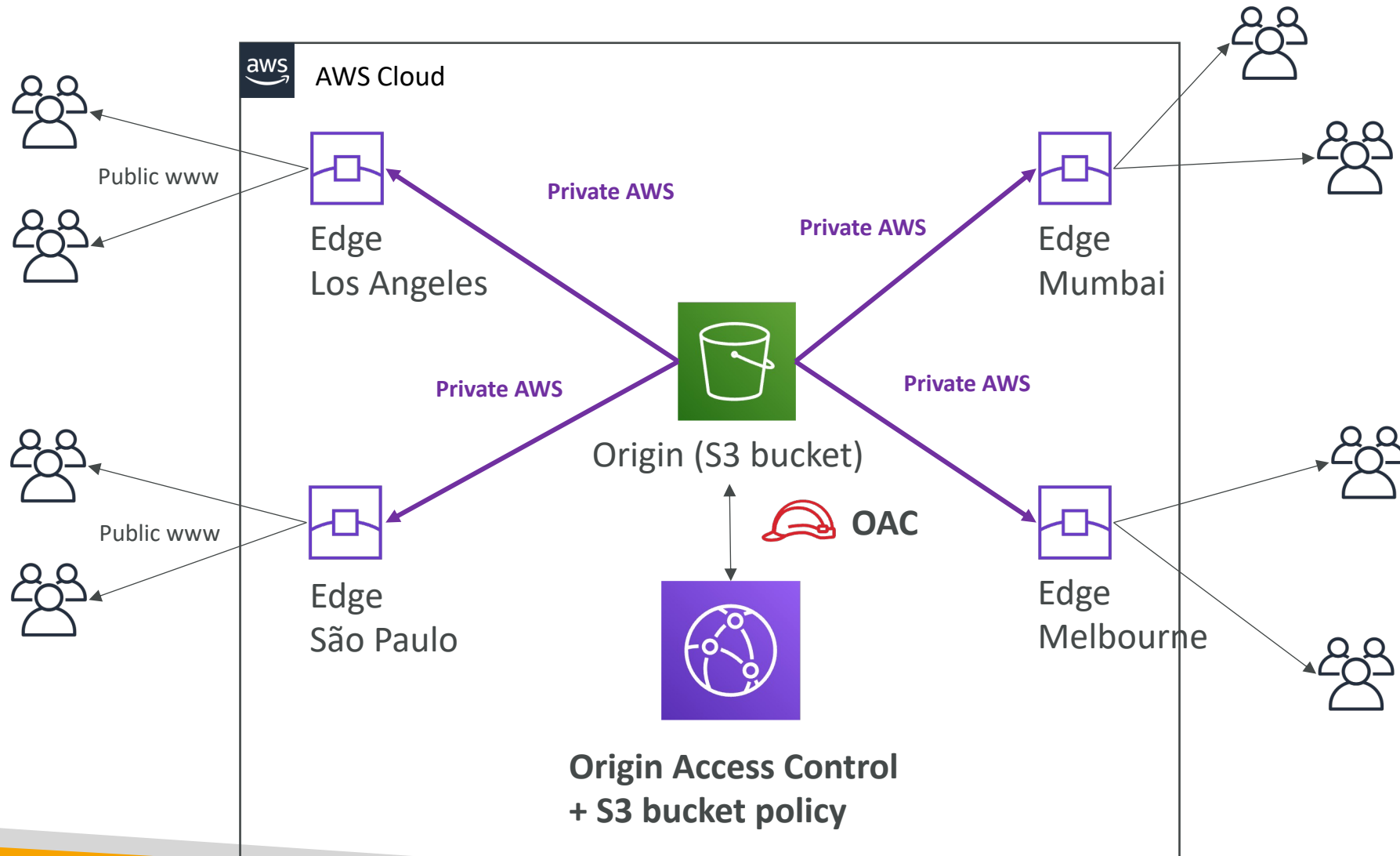
- **S3 bucket**
  - For distributing files and caching them at the edge
  - Enhanced security with CloudFront **Origin Access Control (OAC)**
  - OAC is replacing Origin Access Identity (OAI)
  - CloudFront can be used as an ingress (to upload files to S3)
- **Custom Origin (HTTP)**
  - Application Load Balancer
  - EC2 instance
  - S3 website (must first enable the bucket as a static S3 website)
  - Any HTTP backend you want



# CloudFront at a high level



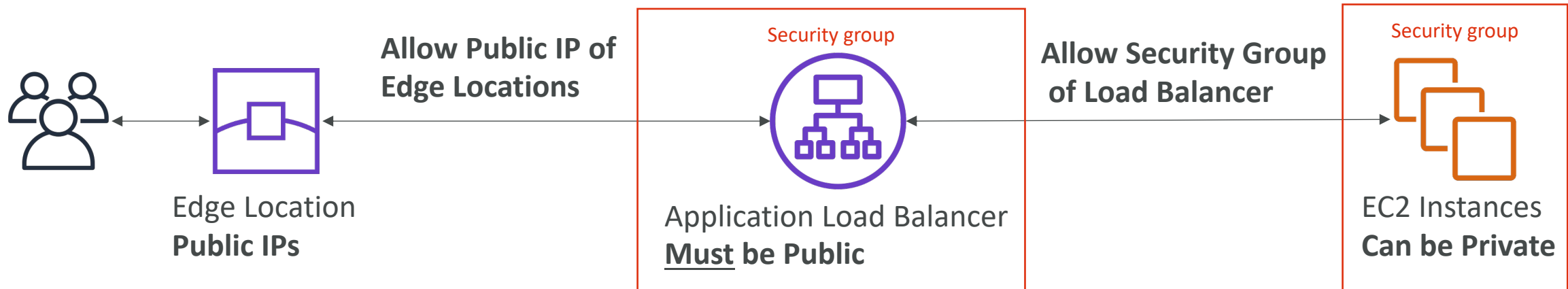
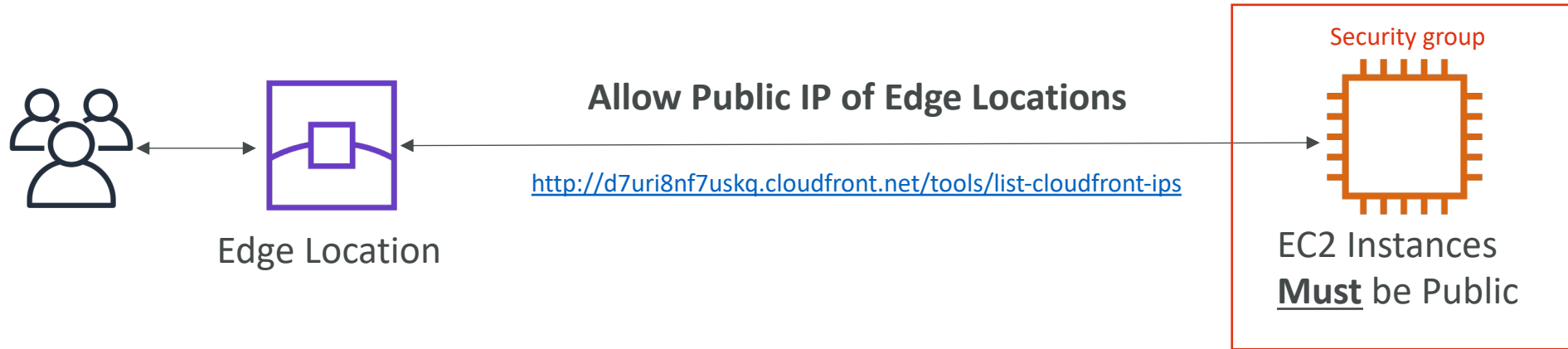
# CloudFront – S3 as an Origin



# CloudFront vs S3 Cross Region Replication

- CloudFront:
  - Global Edge network
  - Files are cached for a TTL (maybe a day)
  - Great for static content that must be available everywhere
- S3 Cross Region Replication:
  - Must be setup for each region you want replication to happen
  - Files are updated in near real-time
  - Read only
  - Great for dynamic content that needs to be available at low-latency in few regions

# CloudFront – ALB or EC2 as an origin

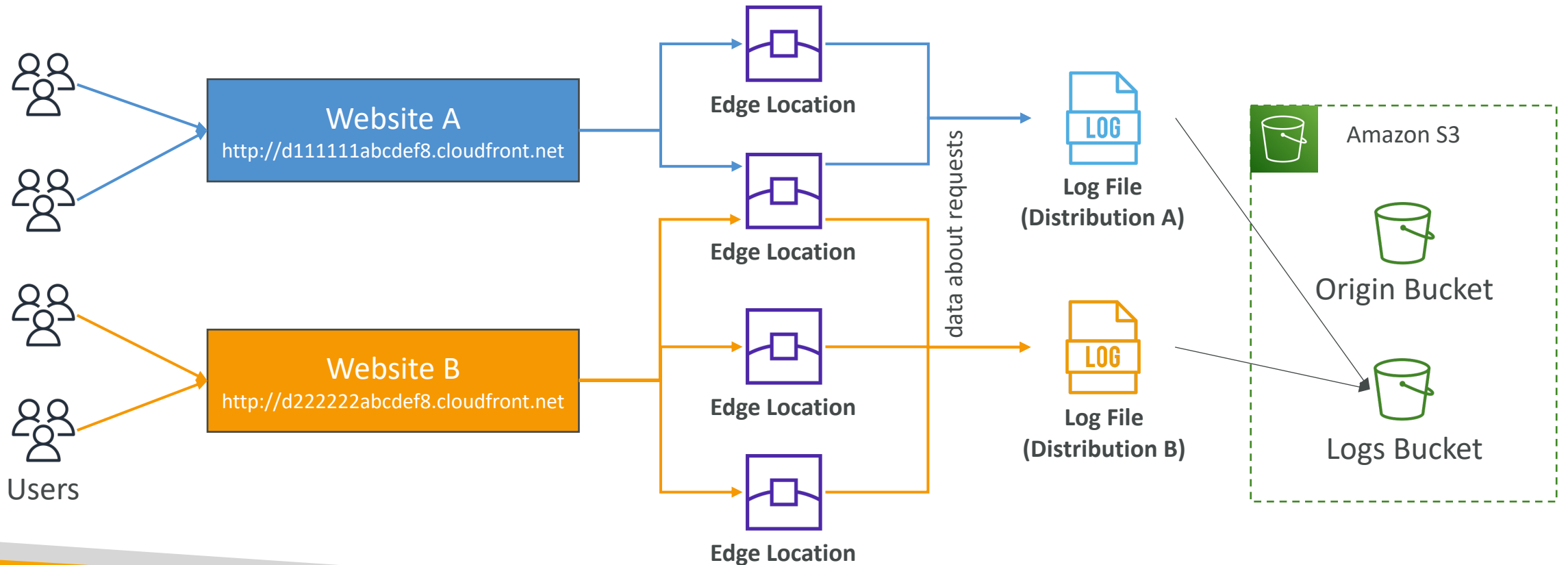


# CloudFront Geo Restriction

- You can restrict who can access your distribution
  - **Allowlist:** Allow your users to access your content only if they're in one of the countries on a list of approved countries.
  - **Blocklist:** Prevent your users from accessing your content if they're in one of the countries on a list of banned countries.
- The “country” is determined using a 3<sup>rd</sup> party Geo-IP database
- Use case: Copyright Laws to control access to content

# CloudFront Access Logs

- Logs every request made to CloudFront into a logging S3 bucket



# CloudFront Reports

- It's possible to generate reports on:
  - Cache Statistics Report
  - Popular Objects Report
  - Top Referrers Report
  - Usage Reports
  - Viewers Report
- These reports are based on the data from the Access Logs.

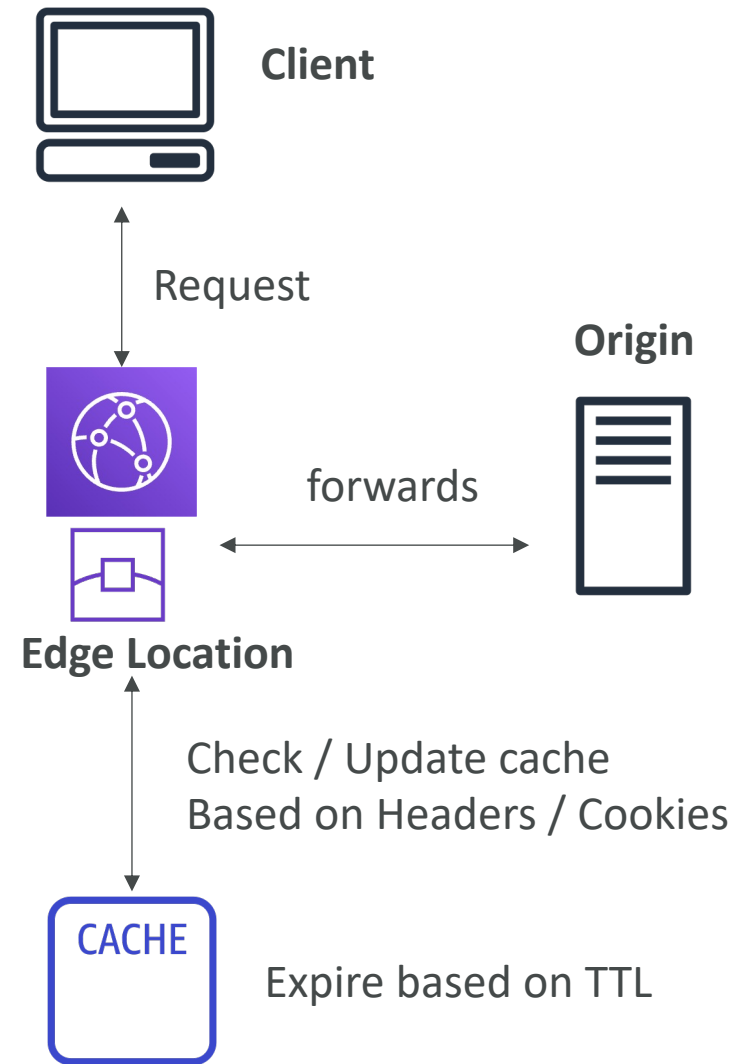
# CloudFront troubleshooting

- CloudFront caches HTTP 4xx and 5xx status codes returned by S3 ( or the origin server)
- 4xx error code indicates that user doesn't have access to the underlying bucket (403) or the object user is requesting is not found (404)
- 5xx error codes indicates Gateway issues



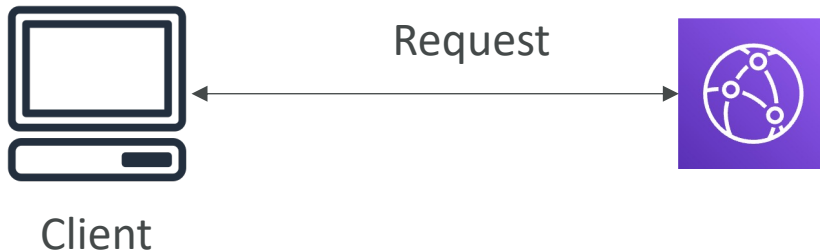
# CloudFront Caching

- Cache based on
  - Headers
  - Session Cookies
  - Query String Parameters
- The cache lives at each CloudFront **Edge Location**
- You want to maximize the cache hit rate to minimize requests on the origin
- Control the TTL (0 seconds to 1 year), can be set by the origin using the Cache-Control header, Expires header...
- You can invalidate part of the cache using the **CreateInvalidation API**



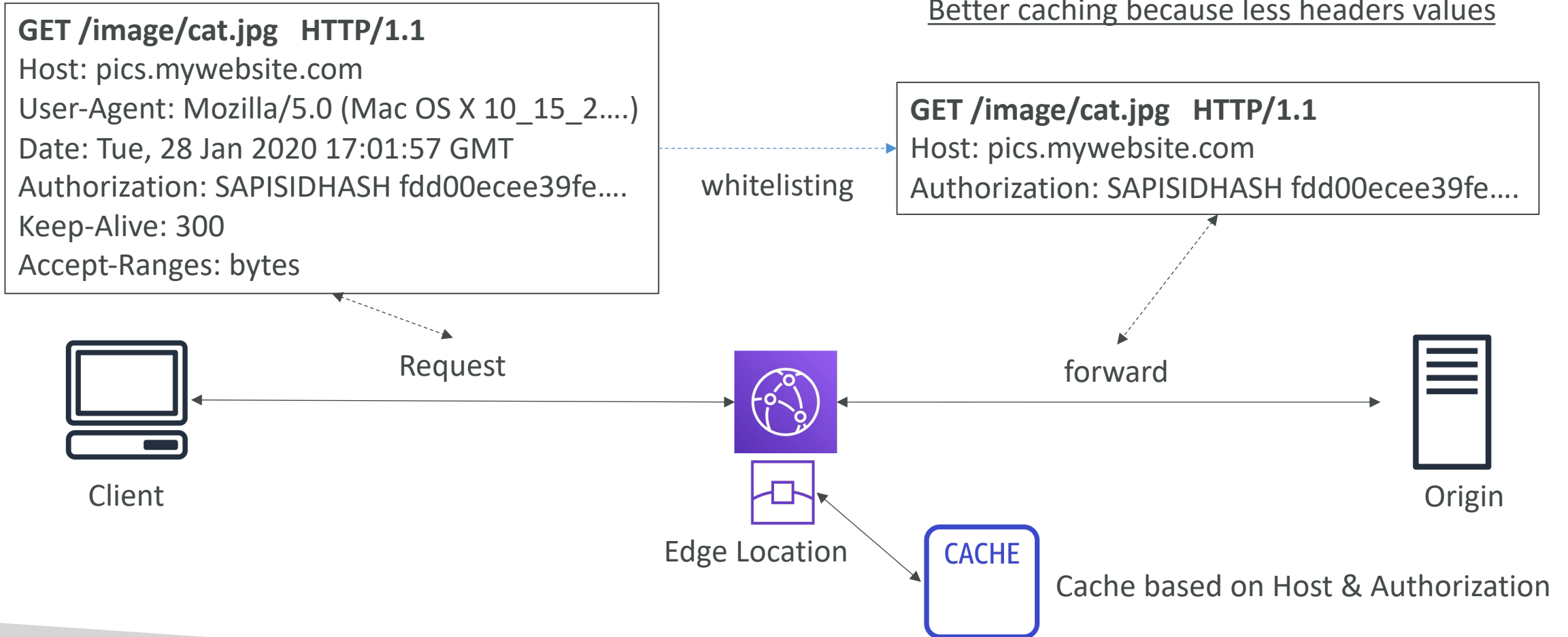
# CloudFront Cache Behaviour for Headers

```
GET /image/cat.jpg HTTP/1.1
Host: pics.mywebsite.com
User-Agent: Mozilla/5.0 (Mac OS X 10_15_2....)
Date: Tue, 28 Jan 2020 17:01:57 GMT
Authorization: Sapisidhash fdd00ecee39fe....
Keep-Alive: 300
Accept-Ranges: bytes
```



1. Forward all headers to your origin
  - => no caching, every request to origin
  - => TTL must be set to 0
2. Forward a whitelist of headers
  - caching based on values in all the specified headers
3. (None) == Forward only the default headers
  - no caching based on request headers
  - Best caching performance

# CloudFront Caching – Whitelist Headers



# CloudFront Origin Headers vs Cache Behavior

- Origin Custom Headers:

- Origin-level setting
- Set a **constant** header / header value for all requests to origin

Origin Custom Headers	Header Name	Value	<b>i</b>
	<input type="text" value="Constant Header"/>	<input type="text" value="Constant Header Value"/>	
	<input type="text"/>	<input type="text"/>	

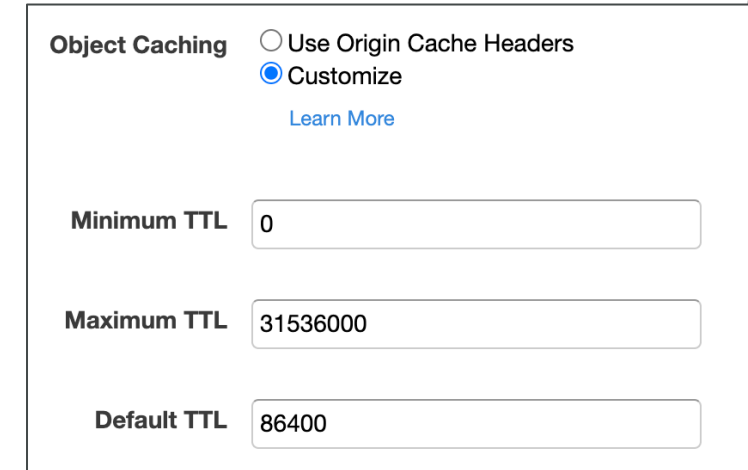
- Behavior setting:

- Cache-related settings
- Contains the whitelist of headers to forward

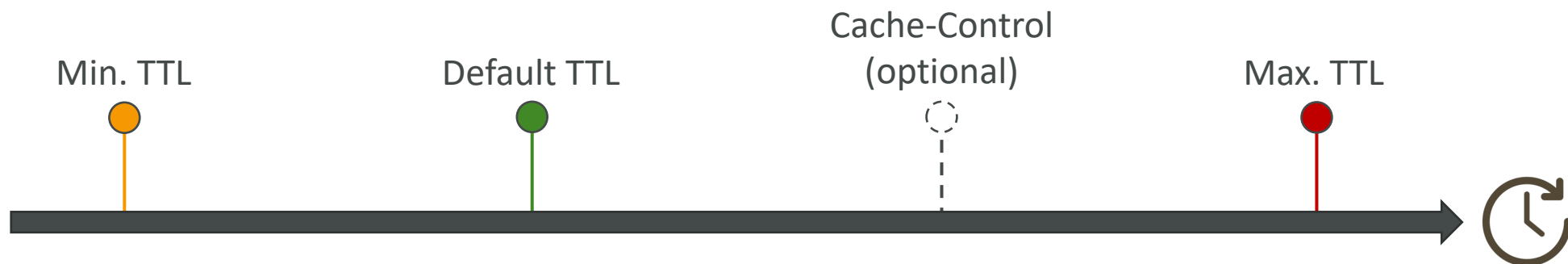
Whitelist Headers	<b>i</b>
<input type="text" value="Filter headers or enter a custom header"/>	<b>2 header(s) whitelisted</b>
<ul style="list-style-type: none"><li>Accept</li><li>Accept-Charset</li><li>Accept-Datetime</li><li>Accept-Encoding</li><li>Accept-Language</li><li>Access-Control-Request-Headers</li></ul>	<input type="button" value="Add Custom &gt;&gt;"/> <input type="button" value="Add &gt;&gt;"/> <input type="button" value="&lt;&lt; Remove"/>
	<ul style="list-style-type: none"><li>CloudFront-Is-Desktop-Viewer</li><li>CloudFront-Is-Mobile-Viewer</li></ul>

# CloudFront Caching TTL

- “Cache-Control: max-age” is preferred to “Expires” header
- If the origin always sends back the header Cache-Control , then you can set the TTL to be controlled only by that header
- In case you want to set min/max boundaries, you choose “customize” for the Object Caching setting
- In case the Cache-Control header is missing, it will default to “default value”

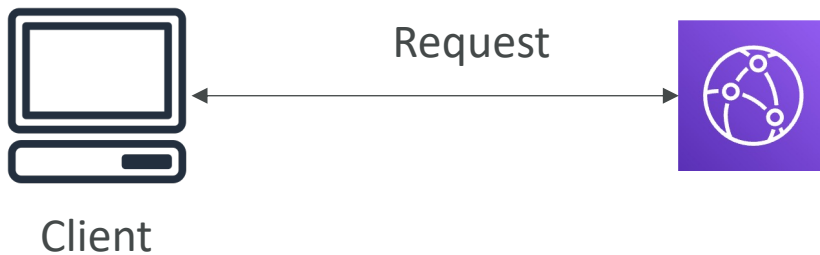


The screenshot shows the 'Object Caching' configuration interface. It features three radio buttons: 'Use Origin Cache Headers' (unselected), 'Customize' (selected), and a 'Learn More' link. Below are three input fields: 'Minimum TTL' with the value '0', 'Maximum TTL' with the value '31536000', and 'Default TTL' with the value '86400'.



# CloudFront Cache Behaviour for Cookies

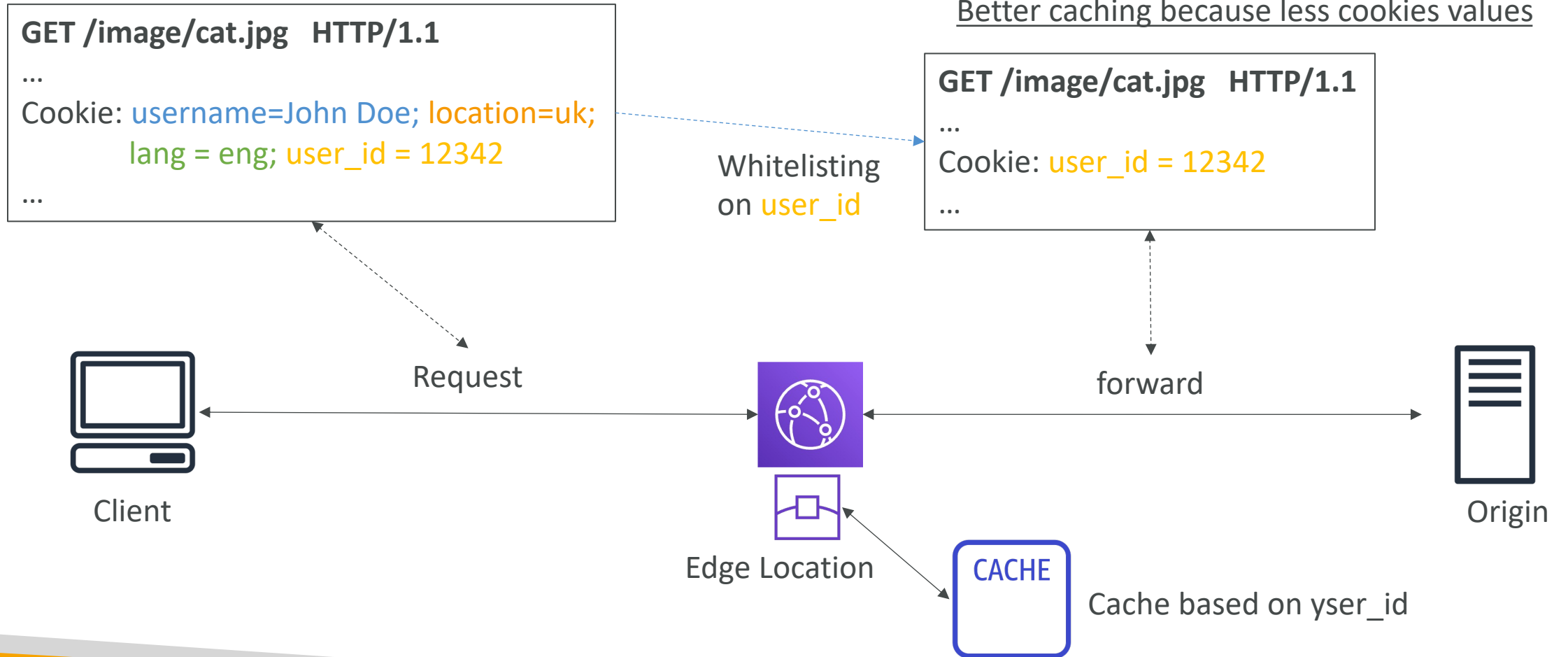
```
GET /image/cat.jpg HTTP/1.1
...
Cookie: username=John Doe; location=uk;
       lang = eng; user_id = 12342
...
```



Cookies is a specific request header

1. Default: do not process the cookies
  - Caching is not based on cookies
  - Cookies are not forwarded
2. Forward a whitelist of cookies
  - caching based on values in all the specified cookies
3. Forward all cookies
  - Worst caching performance

# CloudFront Caching – Cookies



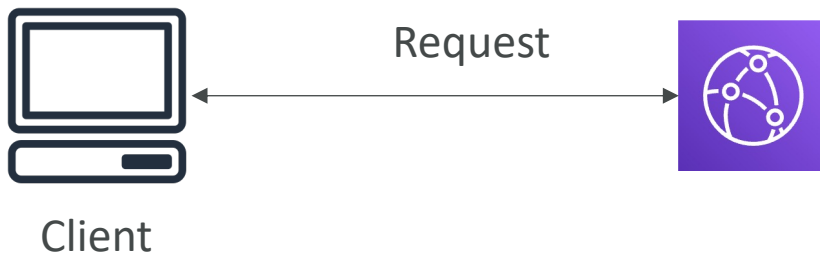
# CloudFront Cache Behaviour for Query Strings

Query Strings Parameters are in the URL

```
GET /image/cat.jpg?border=red&size=large HTTP/1.1
```

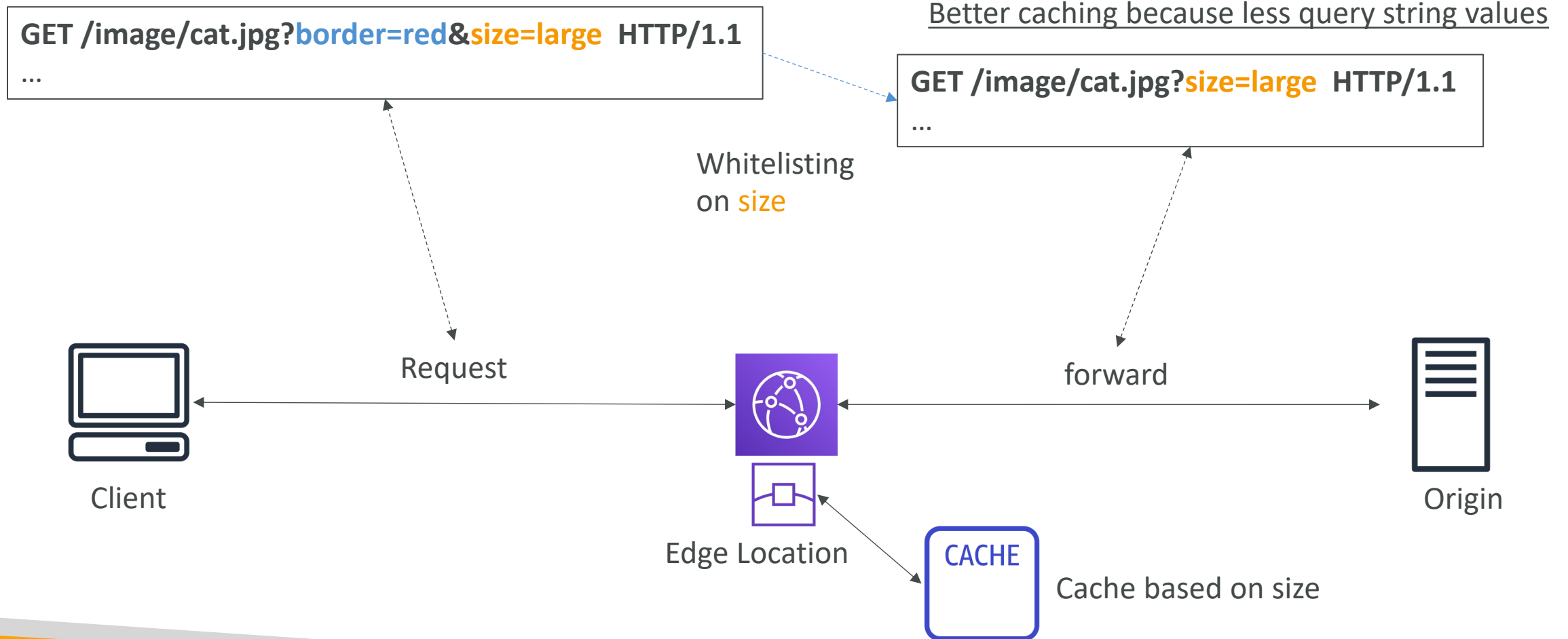
...

1. Default: do not process the query strings
  - Caching is not based on query strings
  - Parameters are not forwarded
2. Forward a whitelist of query strings
  - Caching based on the parameter whitelist
3. Forward all query strings
  - Caching based on all parameters

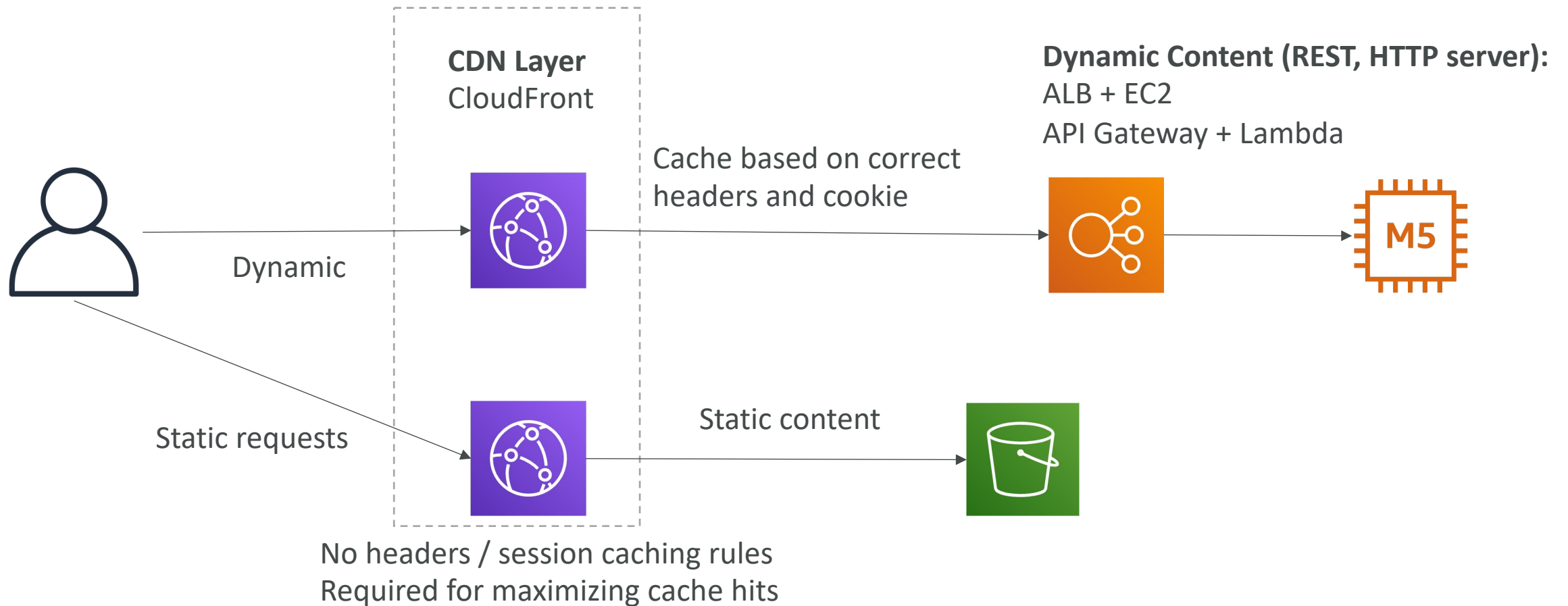




# CloudFront Caching – Query String Parameters



# CloudFront – Maximize cache hits by separating static and dynamic distributions

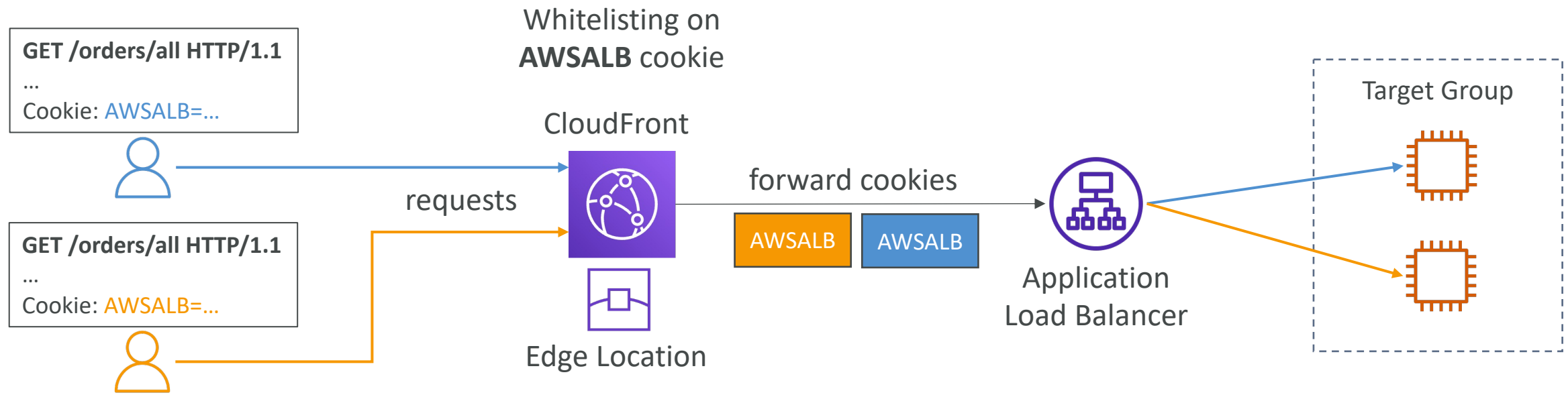


# CloudFront – Increasing Cache Ratio

- Monitor the CloudWatch metric [CacheHitRate](#)
- Specify how long to cache your objects: Cache-Control max-age header
- Specify none or the minimally required **headers**
- Specify none or the minimally required **cookies**
- Specify none or the minimally required **query string parameters**
- Separate static and dynamic distributions (two origins)

# CloudFront with ALB sticky sessions

- Must forward / whitelist the cookie that controls the session affinity to the origin to allow the session affinity to work
- Set a TTL to a value lesser than when the authentication cookie expires



# Databases

# Amazon RDS Overview



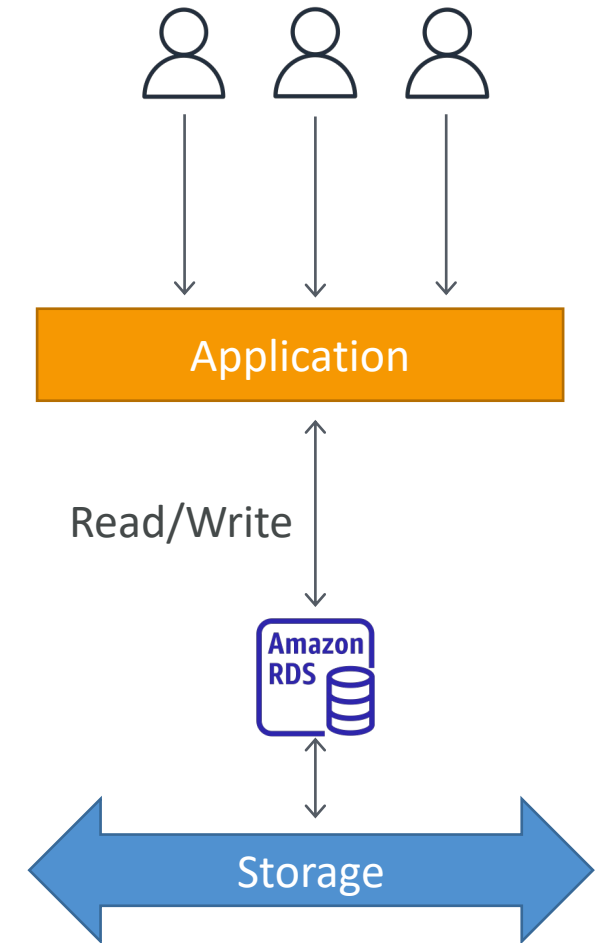
- RDS stands for Relational Database Service
- It's a managed DB service for DB use SQL as a query language.
- It allows you to create databases in the cloud that are managed by AWS
  - Postgres
  - MySQL
  - MariaDB
  - Oracle
  - Microsoft SQL Server
  - Aurora (AWS Proprietary database)

# Advantage over using RDS versus deploying DB on EC2

- RDS is a managed service:
  - Automated provisioning, OS patching
  - Continuous backups and restore to specific timestamp (Point in Time Restore)!
  - Monitoring dashboards
  - Read replicas for improved read performance
  - Multi AZ setup for DR (Disaster Recovery)
  - Maintenance windows for upgrades
  - Scaling capability (vertical and horizontal)
  - Storage backed by EBS (gp2 or io1)
- BUT you can't SSH into your instances

# RDS – Storage Auto Scaling

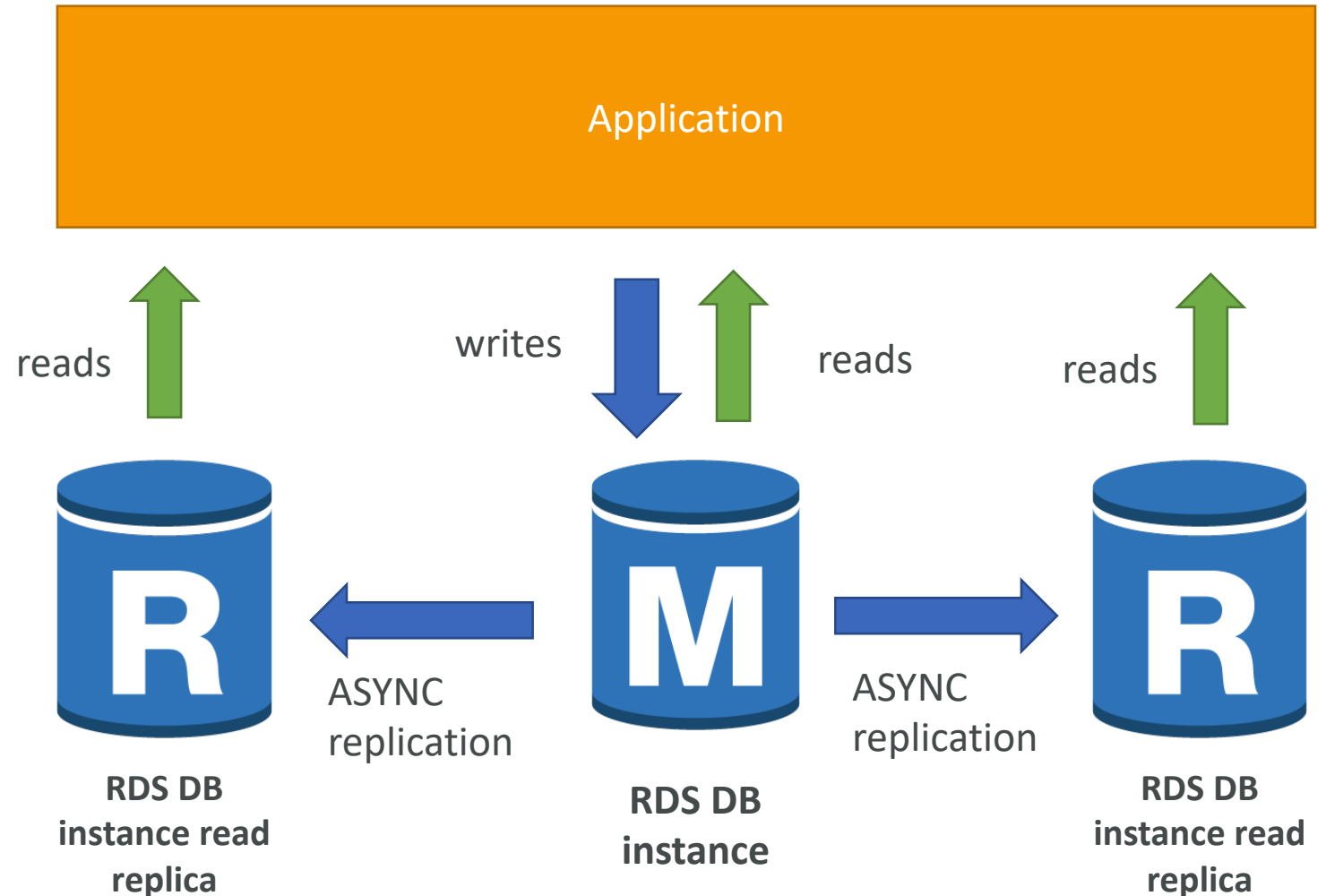
- Helps you increase storage on your RDS DB instance dynamically
- When RDS detects you are running out of free database storage, it scales automatically
- Avoid manually scaling your database storage
- You have to set **Maximum Storage Threshold** (maximum limit for DB storage)
- Automatically modify storage if:
  - Free storage is less than 10% of allocated storage
  - Low-storage lasts at least 5 minutes
  - 6 hours have passed since last modification
- Useful for applications with **unpredictable workloads**
- Supports all RDS database engines (MariaDB, MySQL, PostgreSQL, SQL Server, Oracle)





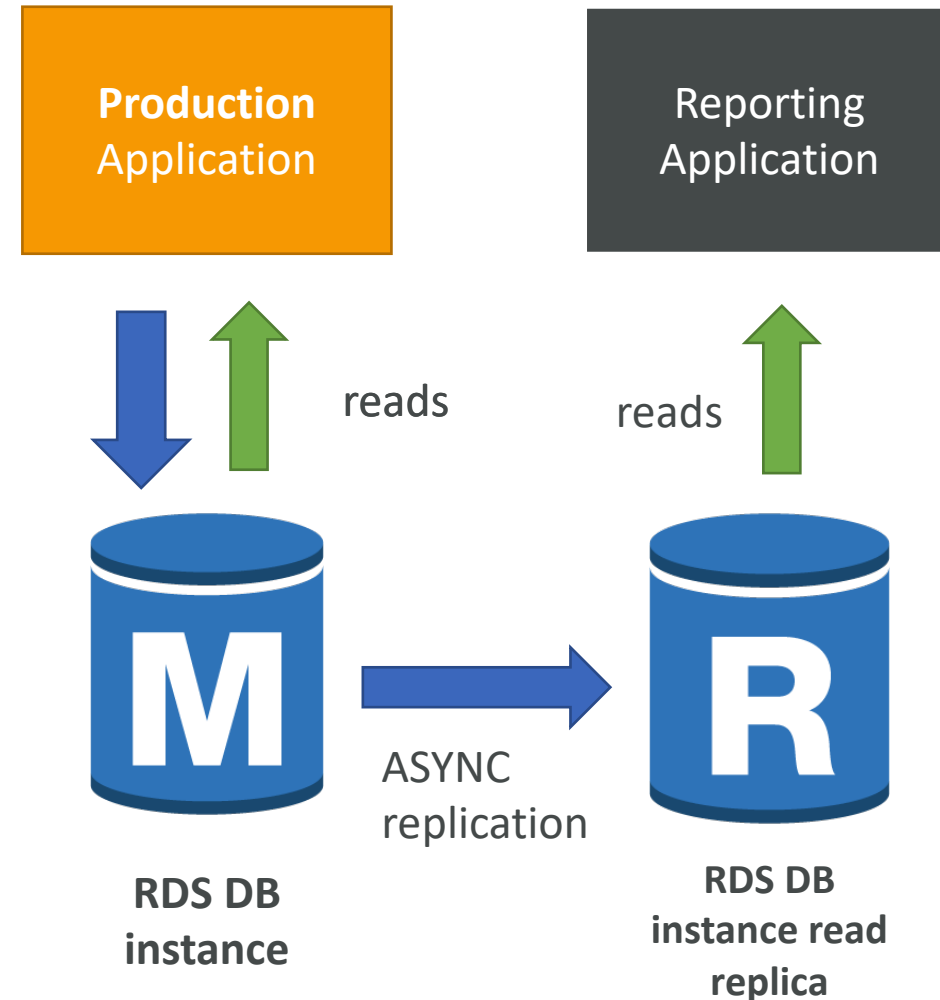
# RDS Read Replicas for read scalability

- Up to 5 Read Replicas
- Within AZ, Cross AZ or Cross Region
- Replication is **ASYNC**, so reads are eventually consistent
- Replicas can be promoted to their own DB
- Applications must update the connection string to leverage read replicas



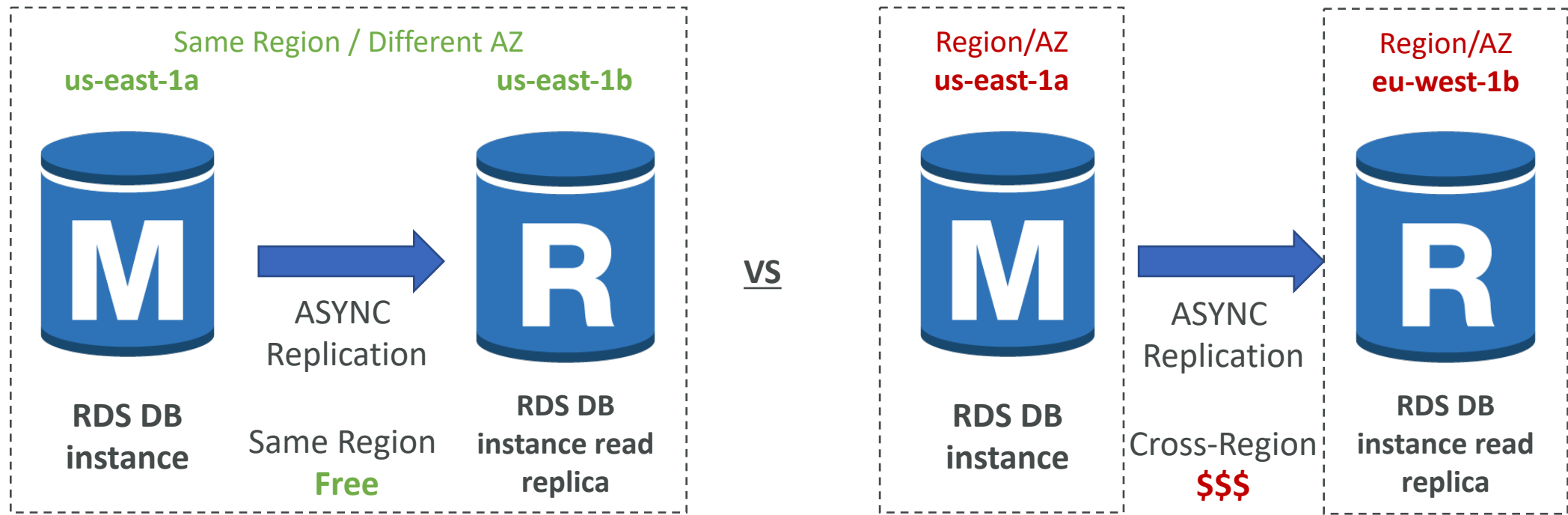
# RDS Read Replicas – Use Cases

- You have a production database that is taking on normal load
- You want to run a reporting application to run some analytics
- You create a Read Replica to run the new workload there
- The production application is unaffected
- Read replicas are used for SELECT (=read) only kind of statements (not INSERT, UPDATE, DELETE)



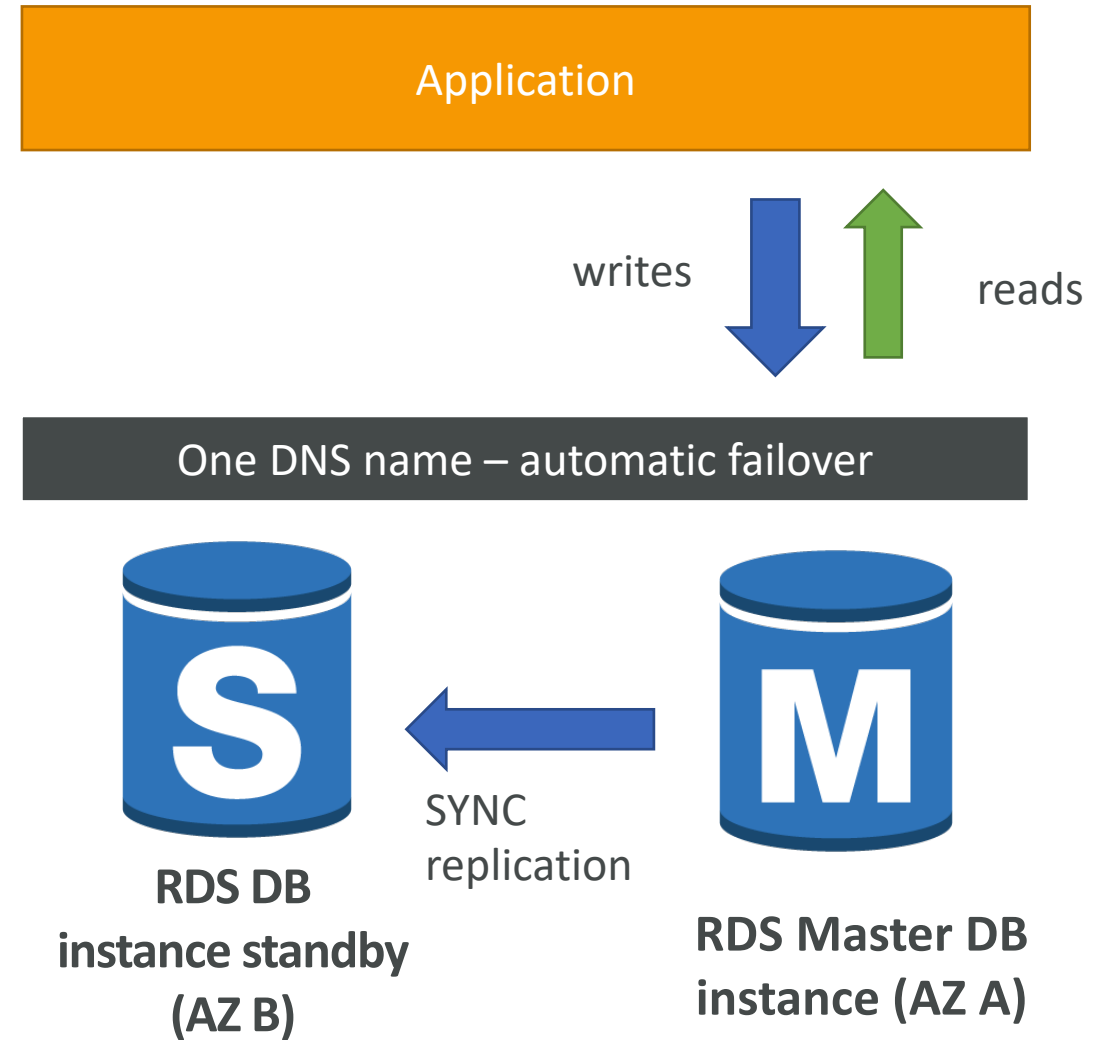
# RDS Read Replicas – Network Cost

- In AWS there's a network cost when data goes from one AZ to another
- For RDS Read Replicas within the same region, you don't pay that fee



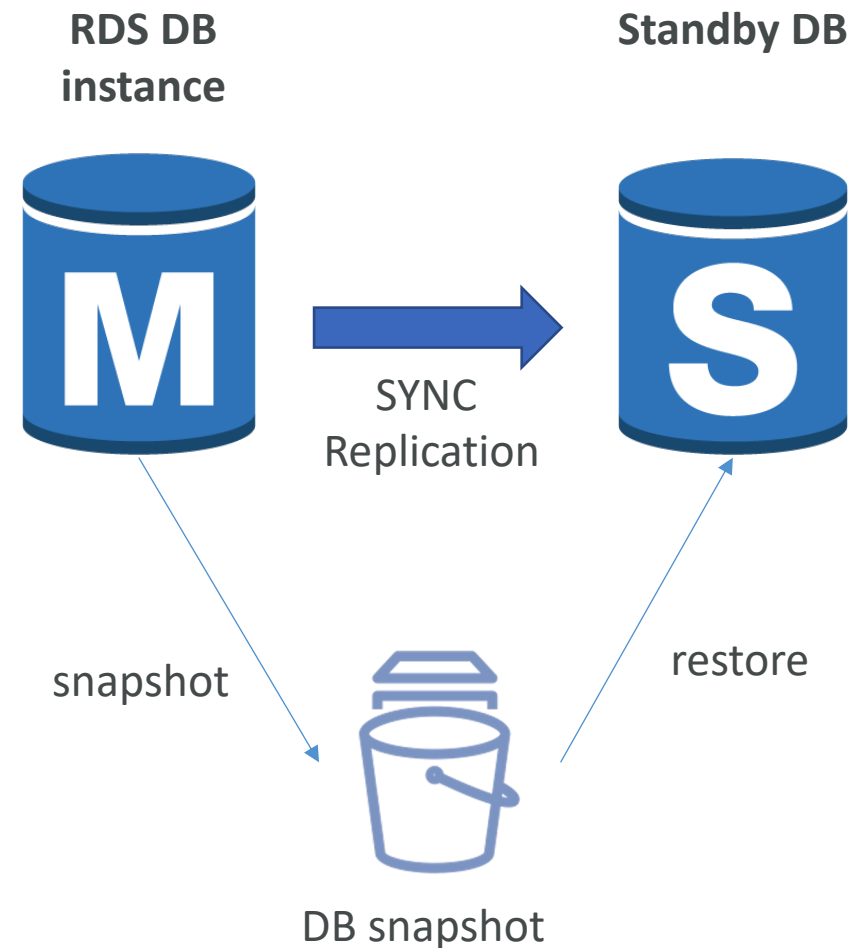
# RDS Multi AZ (Disaster Recovery)

- SYNC replication
- One DNS name – automatic app failover to standby
- Increase **availability**
- Failover in case of loss of AZ, loss of network, instance or storage failure
- No manual intervention in apps
- Not used for scaling
- Note: The Read Replicas be setup as Multi AZ for Disaster Recovery (DR)



# RDS – From Single-AZ to Multi-AZ

- Zero downtime operation (no need to stop the DB)
- Just click on “modify” for the database
- The following happens internally:
  - A snapshot is taken
  - A new DB is restored from the snapshot in a new AZ
  - Synchronization is established between the two databases

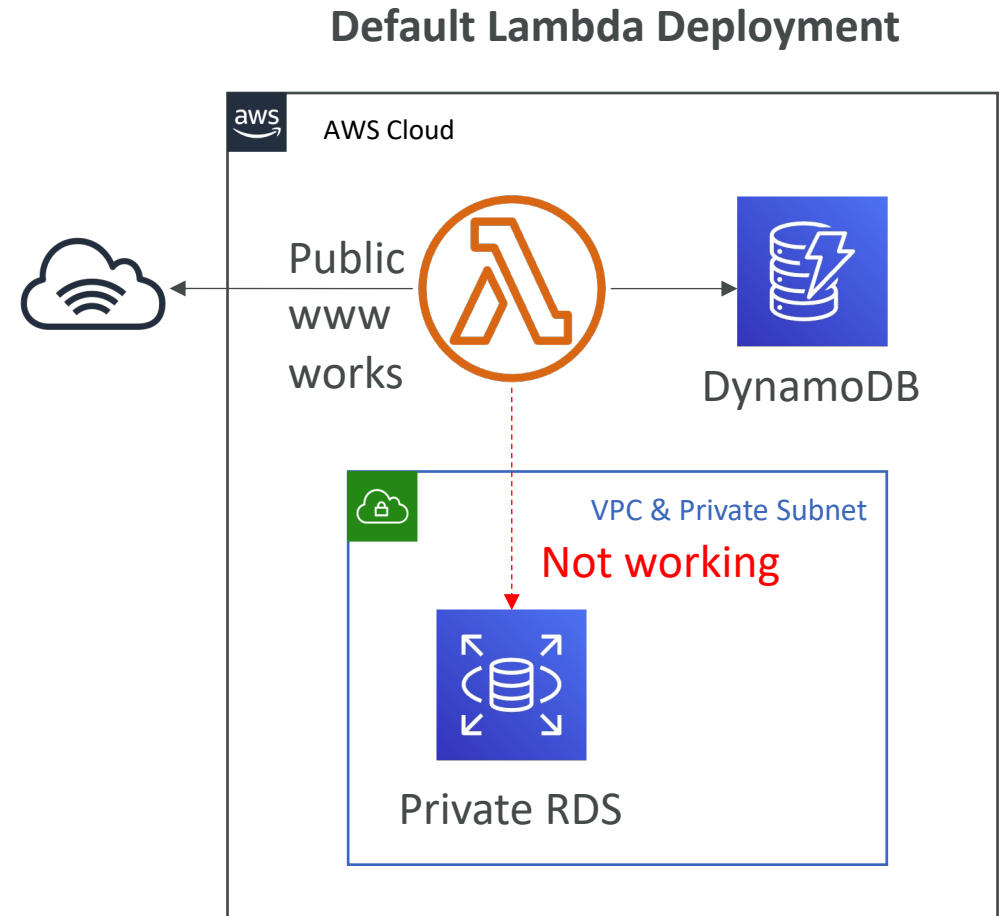


# RDS Multi AZ – Failover Conditions

- The primary DB instance
  - Failed
  - OS is undergoing software patches
  - Unreachable due to loss of network connectivity
  - Modified (e.g., DB instance type changed)
  - Busy and unresponsive
  - Underlying storage failure
- An Availability Zone outage
- A manual failover of the DB instance was initiated using **Reboot with failover**

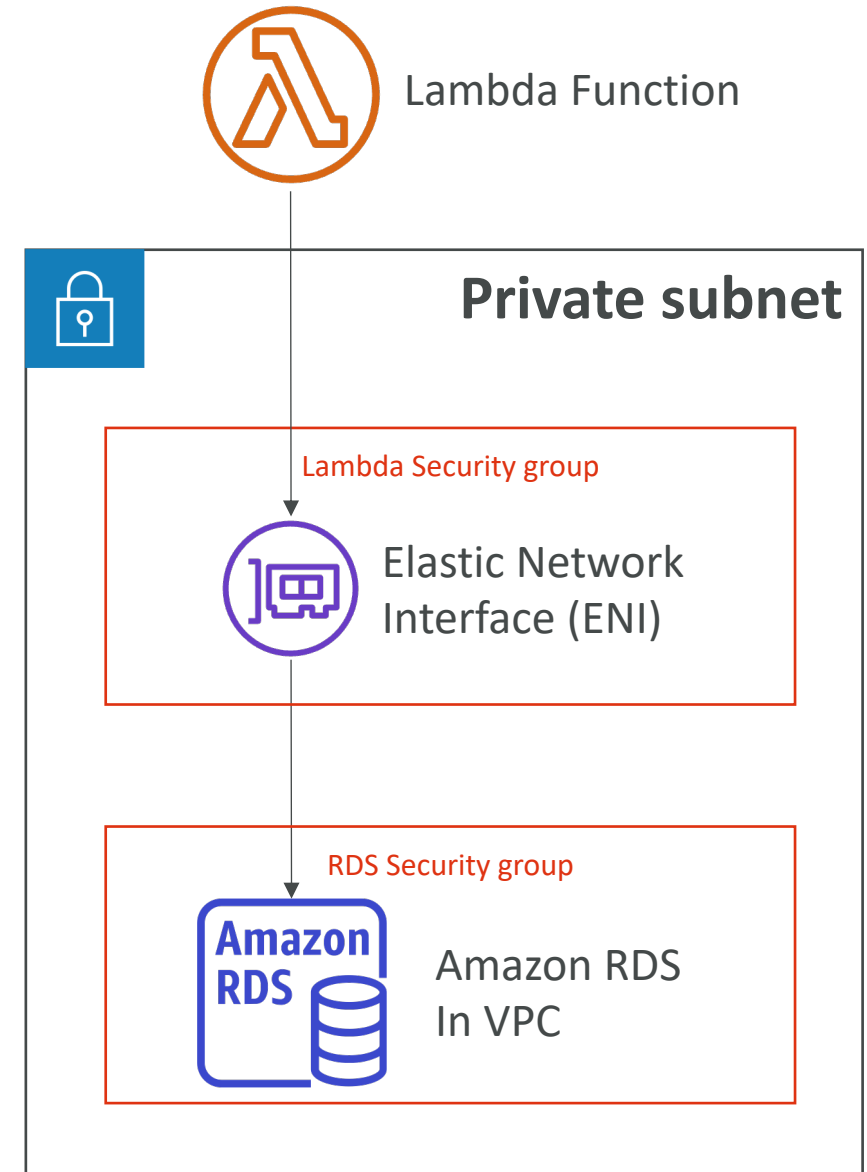
# Lambda by default

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



# Lambda in VPC

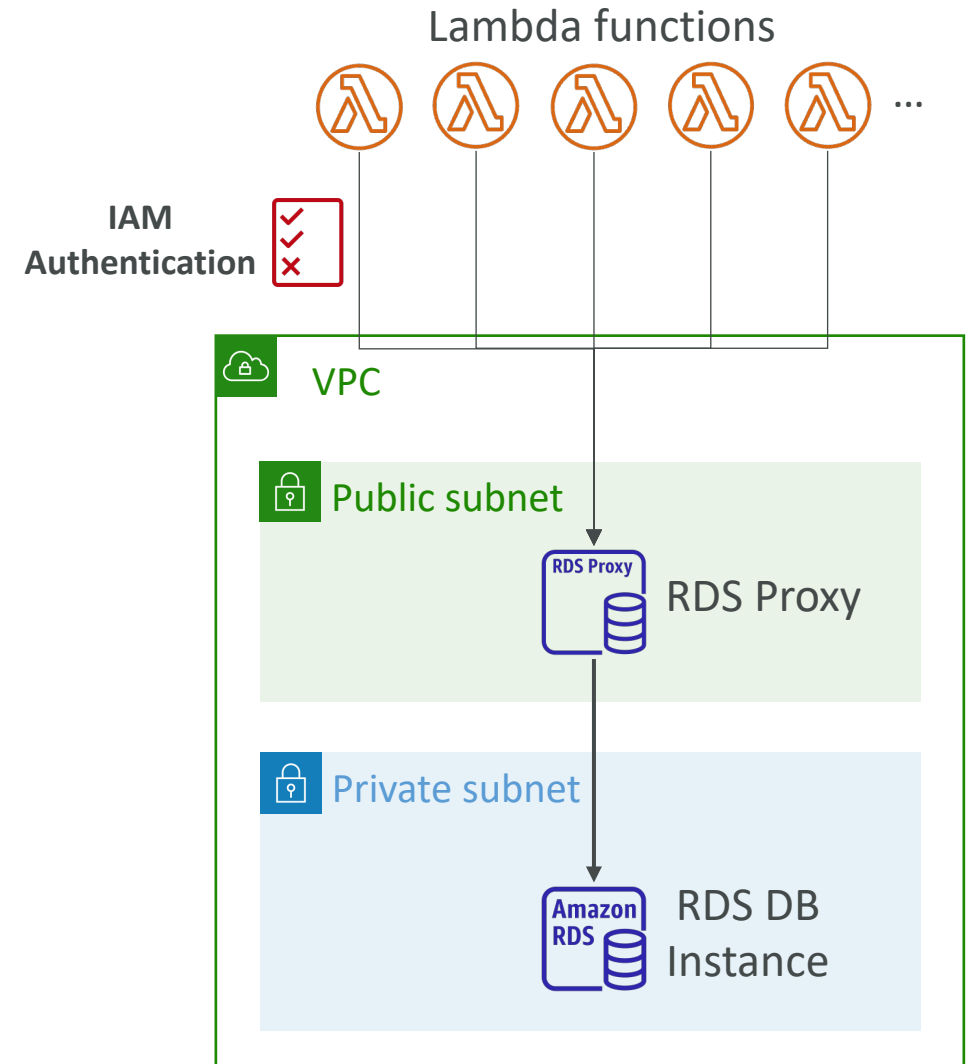
- You must define the VPC ID, the Subnets and the Security Groups
- Lambda will create an ENI (Elastic Network Interface) in your subnets
- `AWSLambdaVPCAccessExecutionRole`





# RDS Proxy for AWS Lambda

- When using Lambda functions with RDS, it opens and maintains a database connection
- This can result in a “TooManyConnections” exception
- With **RDS Proxy**, you no longer need code that handles cleaning up idle connections and managing connection pools
- Supports IAM authentication or DB authentication, auto-scaling
- The Lambda function must have connectivity to the Proxy (public proxy => public Lambda, private proxy => Lambda in VPC)



# DB Parameter Groups

- You can configure the DB engine using Parameter Groups
- Dynamic parameters are applied immediately
- Static parameters are applied after instance reboot
- You can modify parameter group associated with a DB (must reboot)
- See documentation for list of parameters for a DB technology
- **Must-know parameter:**
  - PostgreSQL / SQL Server: `rds.force_ssl=1` => force SSL connections
  - Reminder: for SSL on MySQL / MariaDB, you must run:  
`GRANT SELECT ON mydatabase.* TO 'myuser'@'%' IDENTIFIED BY '...' REQUIRE SSL;`

# RDS Backup vs. Snapshots

## Backups

- Backups are “continuous” and allow point in time recovery
- Backups happen during maintenance windows
- When you delete a DB instance, you can retain automated backups
- Backups have a retention period you set between 0 and 35 days
- To disable backups, set retention period to 0

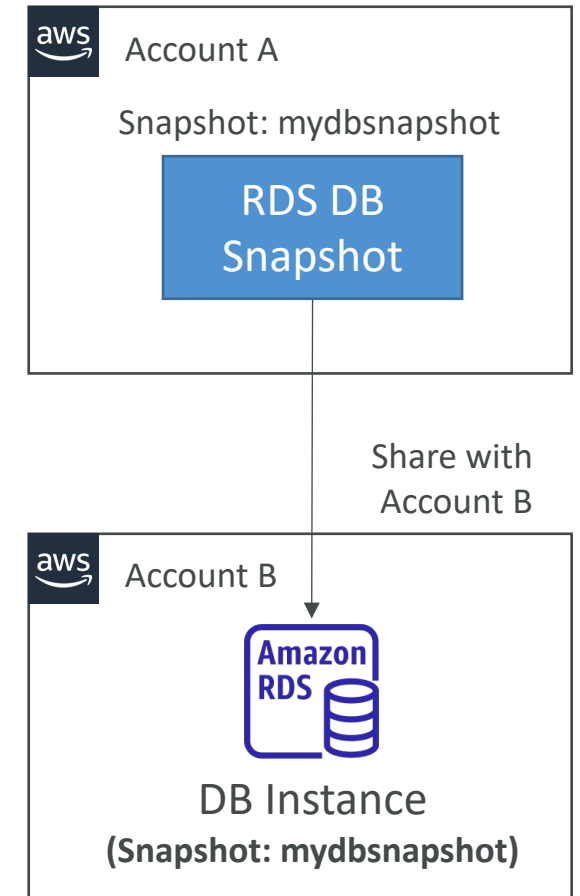
## Snapshots

- Snapshots takes IO operations and can stop the database from seconds to minutes
- Snapshots taken on Multi AZ DB don't impact the master – just the standby
- Snapshots are incremental after the first snapshot (which is full)
- You can copy & share DB Snapshots
- **Manual** Snapshots don't expire
- You can take a “final snapshot” when you delete your DB

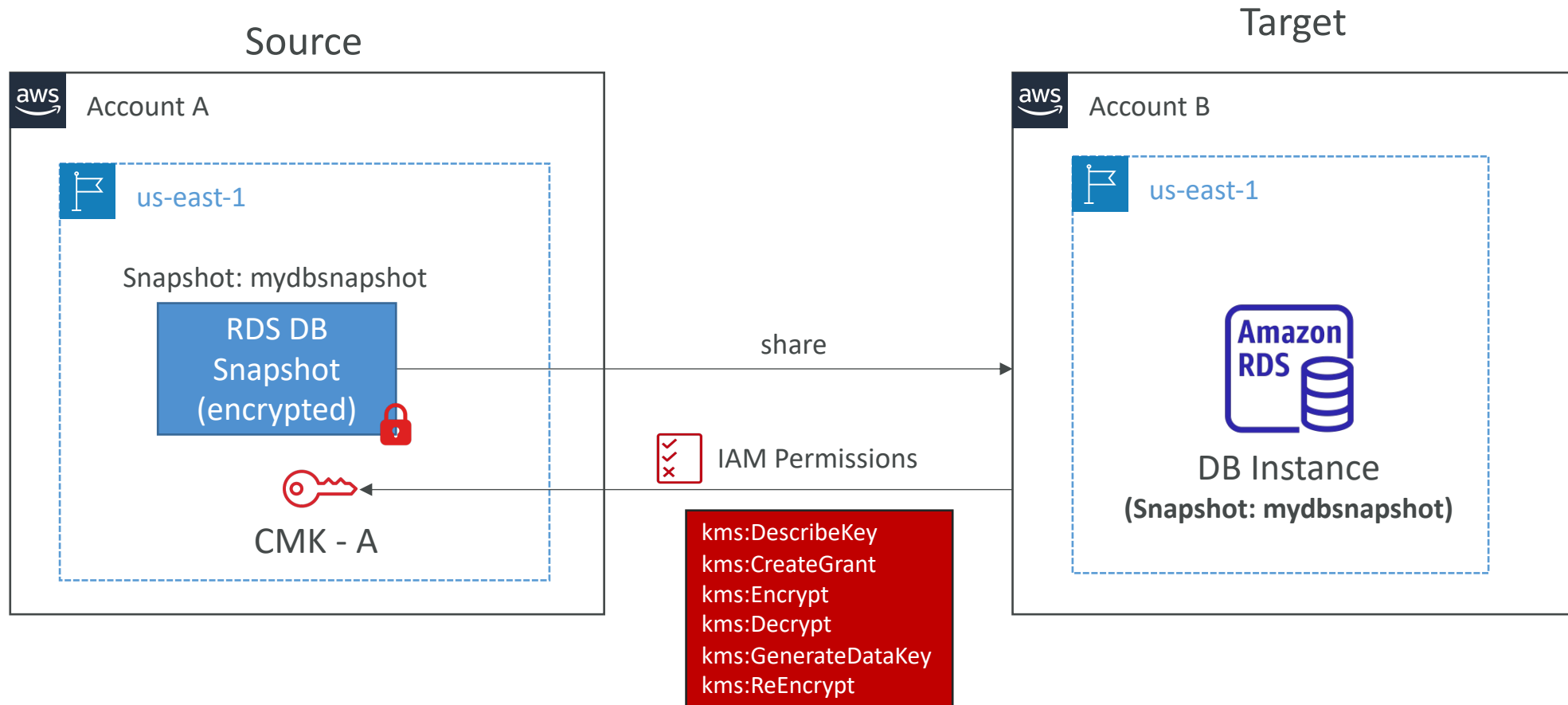
Restoring from Automated Backups or DB Snapshots creates a new DB Instance

# RDS Snapshots Sharing

- **Manual snapshots:** can be shared with AWS accounts
- **Automated snapshots:** can't be shared, copy first
- You can only share unencrypted snapshots and snapshots encrypted with a customer managed key
- If you share an encrypted snapshots, you must also share any customer managed keys used to encrypt them

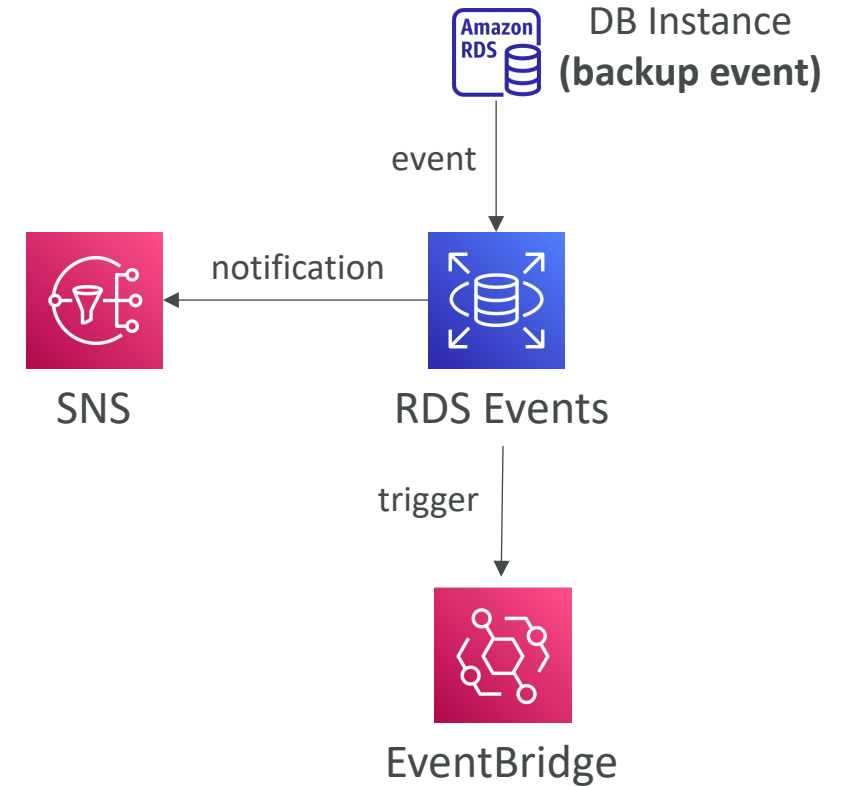


# RDS Snapshots Sharing with KMS Encryption

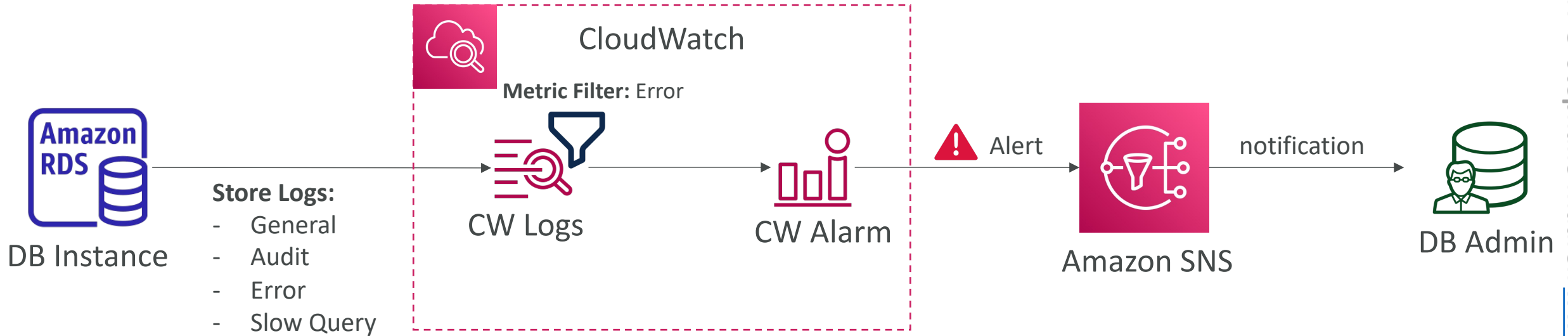


# RDS Events & Event Subscriptions

- RDS keeps record of events related to:
  - DB instances
  - Snapshots
  - Parameter groups, security groups ...
- Example: DB state changed from pending to running
- **RDS Event Subscriptions**
  - Subscribe to events to be notified when an event occurs using SNS
  - Specify the Event Source (instances, SGs, ...) and the Event Category (creation, failover, ...)
- RDS delivers events to EventBridge



# RDS Database Log Files



# RDS with CloudWatch

- CloudWatch metrics associated with RDS (gathered from the hypervisor):
  - DatabaseConnections
  - SwapUsage
  - ReadIOPS / WriteIOPS
  - ReadLatency / WriteLatency
  - ReadThroughPut / WriteThroughPut
  - DiskQueueDepth
  - FreeStorageSpace
- Enhanced Monitoring (gathered from an agent on the DB instance)
  - Useful when you need to see how different processes or threads use the CPU
  - Access to over 50 new CPU, memory, file system, and disk I/O metrics

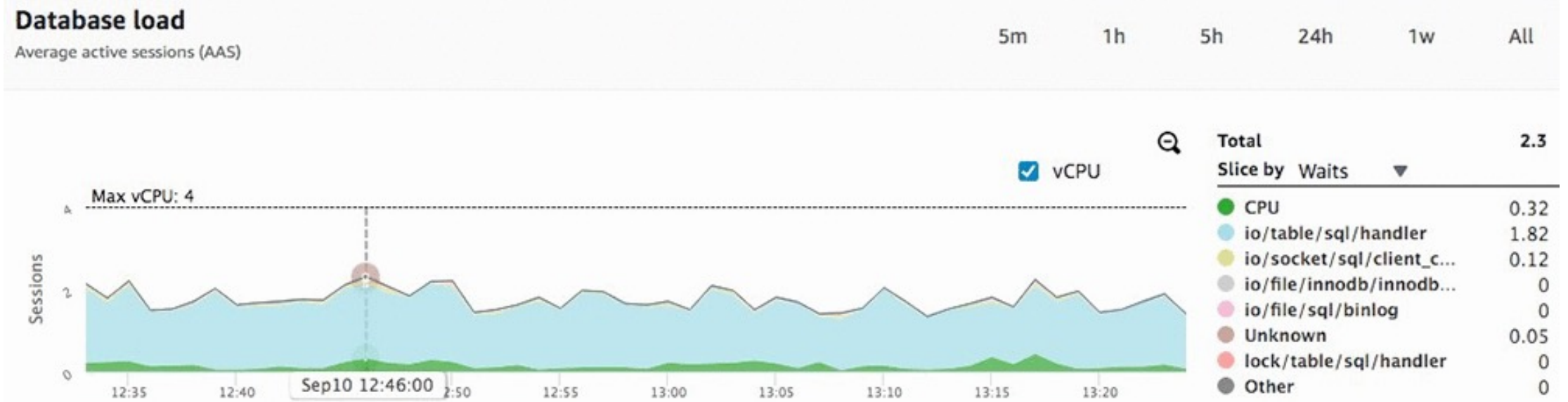


# RDS Performance Insights

- Visualize your database performance and analyze any issues that affect it
- With the Performance Insights dashboard, you can visualize the database load and filter the load:
  - By Waits => find the resource that is the bottleneck (CPU, IO, lock, etc...)
  - By SQL statements => find the SQL statement that is the problem
  - By Hosts => find the server that is using the most our DB
  - By Users => find the user that is using the most our DB
- DBLoad = the number of active sessions for the DB engine
- You can view the SQL queries that are putting load on your database

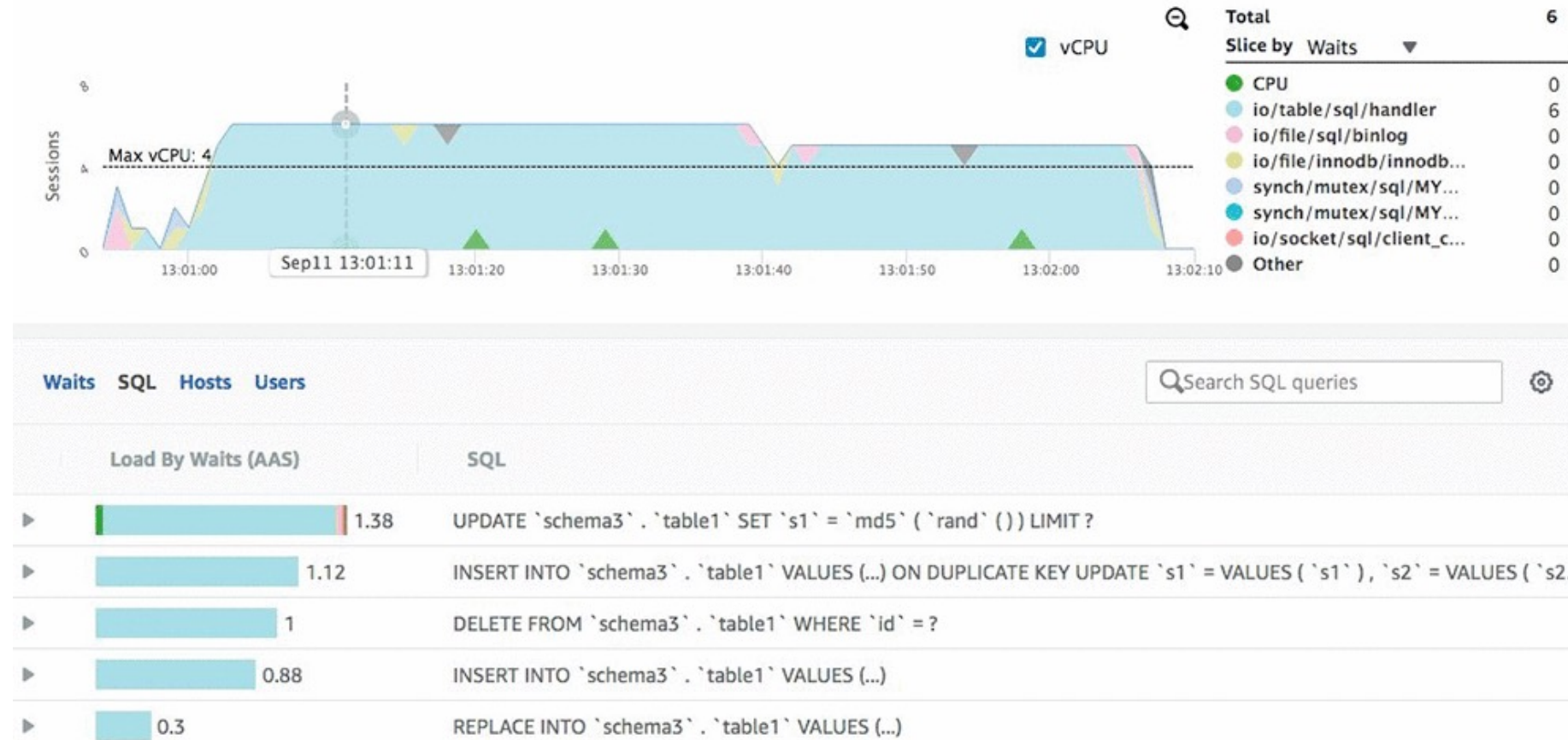
# Performance Insights Screenshots

## DB Waits



From: <https://aws.amazon.com/blogs/database/tuning-amazon-rds-for-mysql-with-performance-insights>

# Performance Insights Screenshots SQL



From: <https://aws.amazon.com/blogs/database/tuning-amazon-rds-for-mysql-with-performance-insights>

# Performance Insights Screenshots Users



From: <https://aws.amazon.com/blogs/database/tuning-amazon-rds-for-mysql-with-performance-insights/>

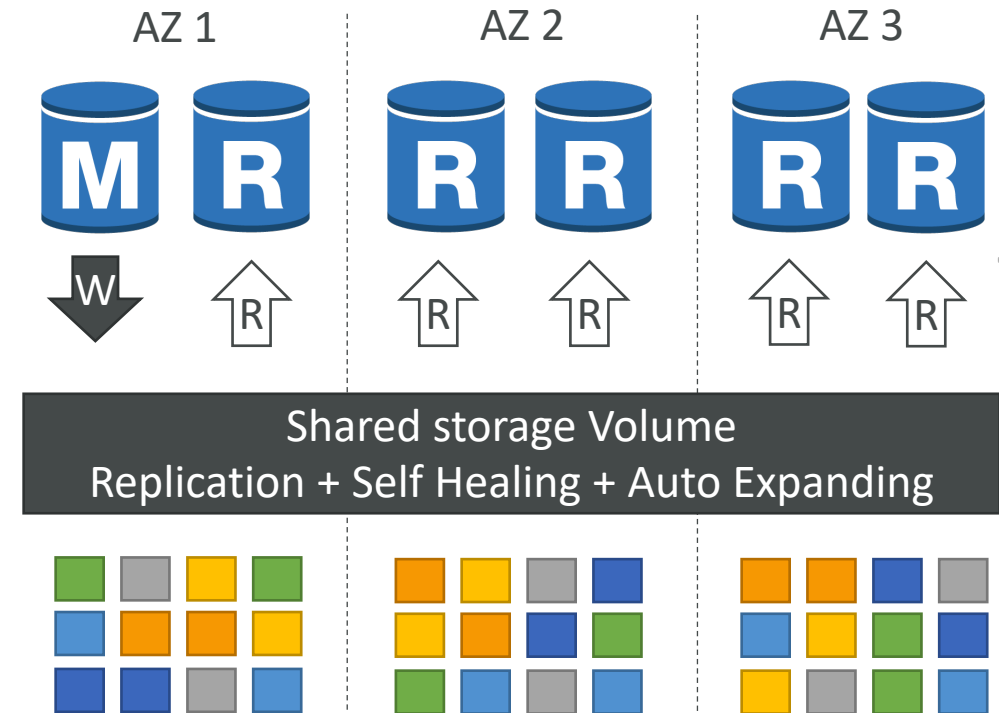


# Amazon Aurora

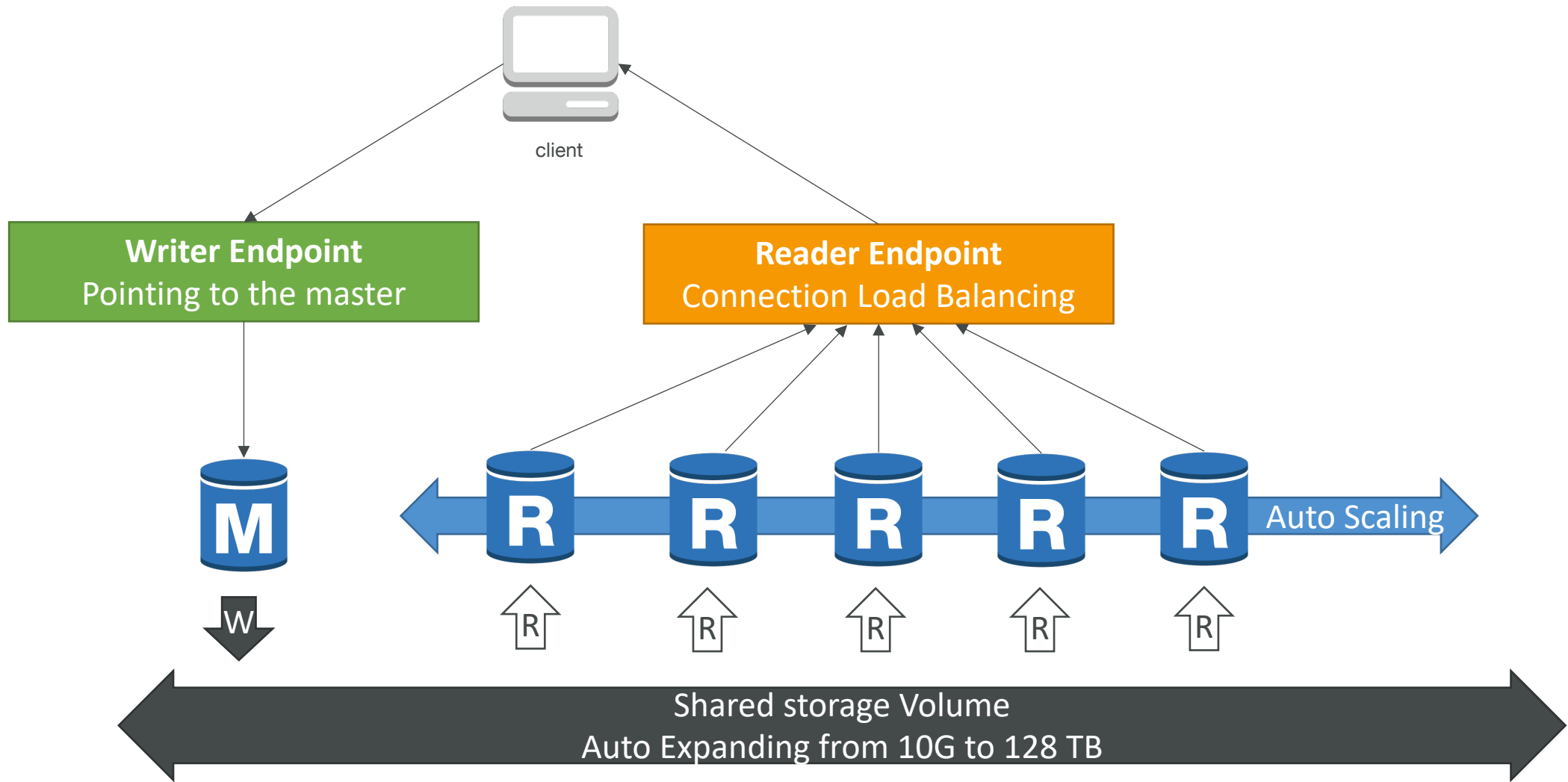
- Aurora is a proprietary technology from AWS (not open sourced)
- Postgres and MySQL are both supported as Aurora DB (that means your drivers will work as if Aurora was a Postgres or MySQL database)
- Aurora is “AWS cloud optimized” and claims 5x performance improvement over MySQL on RDS, over 3x the performance of Postgres on RDS
- Aurora storage automatically grows in increments of 10GB, up to 128 TB.
- Aurora can have 15 replicas while MySQL has 5, and the replication process is faster (sub 10 ms replica lag)
- Failover in Aurora is instantaneous. It’s HA (High Availability) native.
- Aurora costs more than RDS (20% more) – but is more efficient

# Aurora High Availability and Read Scaling

- 6 copies of your data across 3 AZ:
  - 4 copies out of 6 needed for writes
  - 3 copies out of 6 need for reads
  - Self healing with peer-to-peer replication
  - Storage is striped across 100s of volumes
- One Aurora Instance takes writes (master)
- Automated failover for master in less than 30 seconds
- Master + up to 15 Aurora Read Replicas serve reads
- Support for Cross Region Replication



# Aurora DB Cluster



# Features of Aurora

- Automatic fail-over
- Backup and Recovery
- Isolation and security
- Industry compliance
- Push-button scaling
- Automated Patching with Zero Downtime
- Advanced Monitoring
- Routine Maintenance
- Backtrack: restore data at any point of time without using backups



# Backups, Backtracking & Restores in Aurora

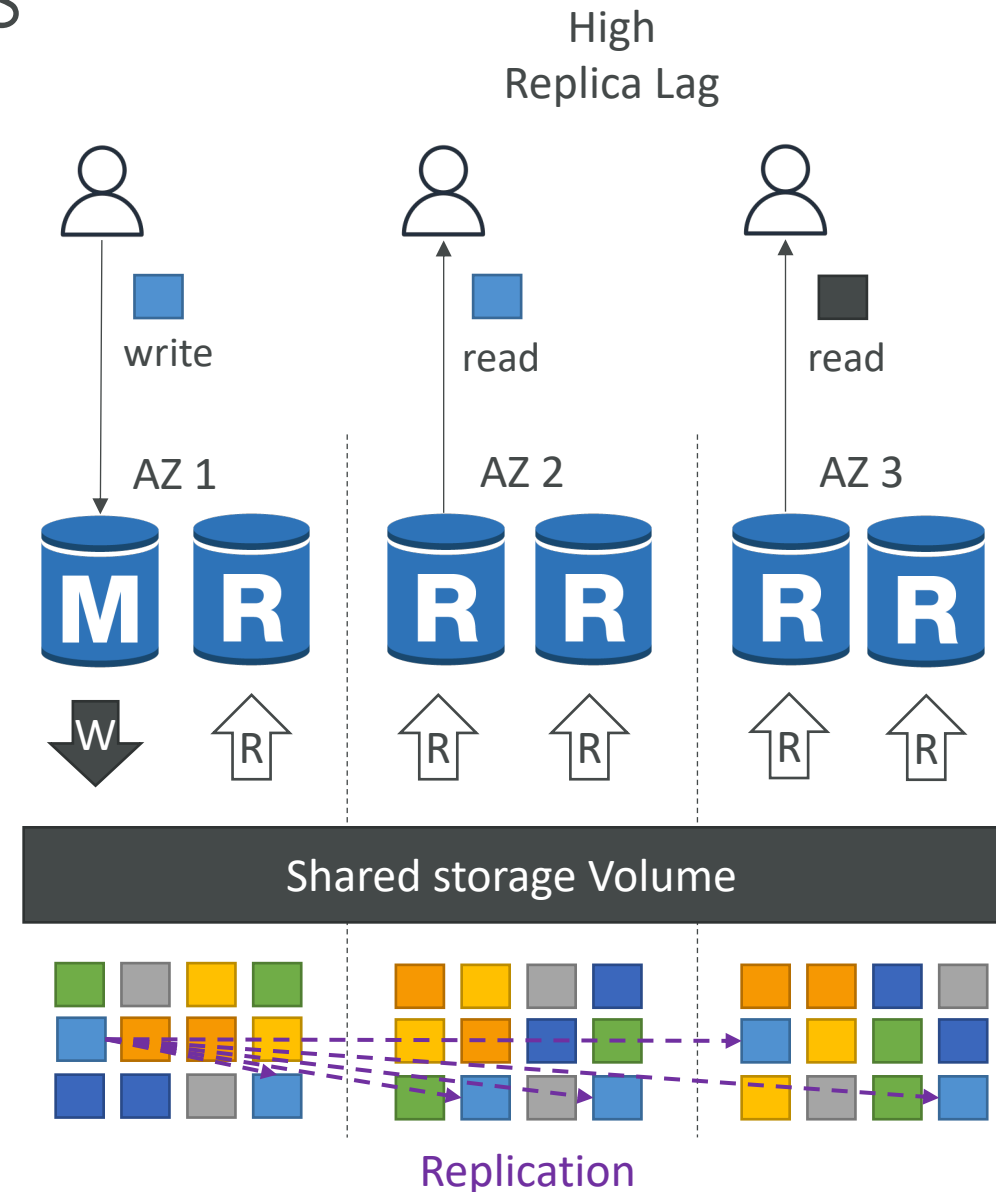
- **Automatic Backups**
  - Retention period 1-35 days (can't be disabled)
  - PITR, restore your DB cluster within 5 minutes of the current time
  - Restore to a new DB cluster
- **Aurora Backtracking**
  - Rewind the DB cluster back and forth in time (up to 72 hours)
  - Doesn't create a new DB cluster (in-place restore)
  - Supports Aurora MySQL only
- **Aurora Database Cloning**
  - Creates a new DB cluster that uses the same DB cluster volume as the original cluster
  - Uses *copy-on-write* protocol (use the original/single copy of the data and allocate storage only when changes made to the data)
  - Example: create a test environment using your production data

# Aurora for SysOps

- You can associate a priority tier (0-15) on each Read Replica
  - Controls the failover priority
  - RDS will promote the Read Replica with the highest priority (lowest tier)
  - If replicas have the same priority, RDS promotes the largest in size
  - If replicas have the same priority and size, RDS promotes arbitrary replica
- You can migrate an RDS MySQL snapshot to Aurora MySQL Cluster

# Aurora: CloudWatch metrics

- **AuroraReplicaLag:** amount of lag when replicating updates from the primary instance
- **AuroraReplicaLagMaximum:** max. amount of lag across all DB instances in the cluster
- **AuroraReplicaLagMinimum:** min. amount of lag across all DB instances in the cluster
- If replica lag is high, that means the users will have a different experience based on which replica they get the data from (eventual consistency)
- **DatabaseConnections:** current number of connections to a DB instance
- **InsertLatency:** average duration of insert operations



# RDS & Aurora Security

- **At-rest encryption:**
  - Database master & replicas encryption using AWS KMS – must be defined as launch time
  - If the master is not encrypted, the read replicas cannot be encrypted
  - To encrypt an un-encrypted database, go through a DB snapshot & restore as encrypted
- **In-flight encryption:** TLS-ready by default, use the AWS TLS root certificates client-side
- **IAM Authentication:** IAM roles to connect to your database (instead of username/pw)
- **Security Groups:** Control Network access to your RDS / Aurora DB
- **No SSH available** except on RDS Custom
- **Audit Logs can be enabled** and sent to CloudWatch Logs for longer retention



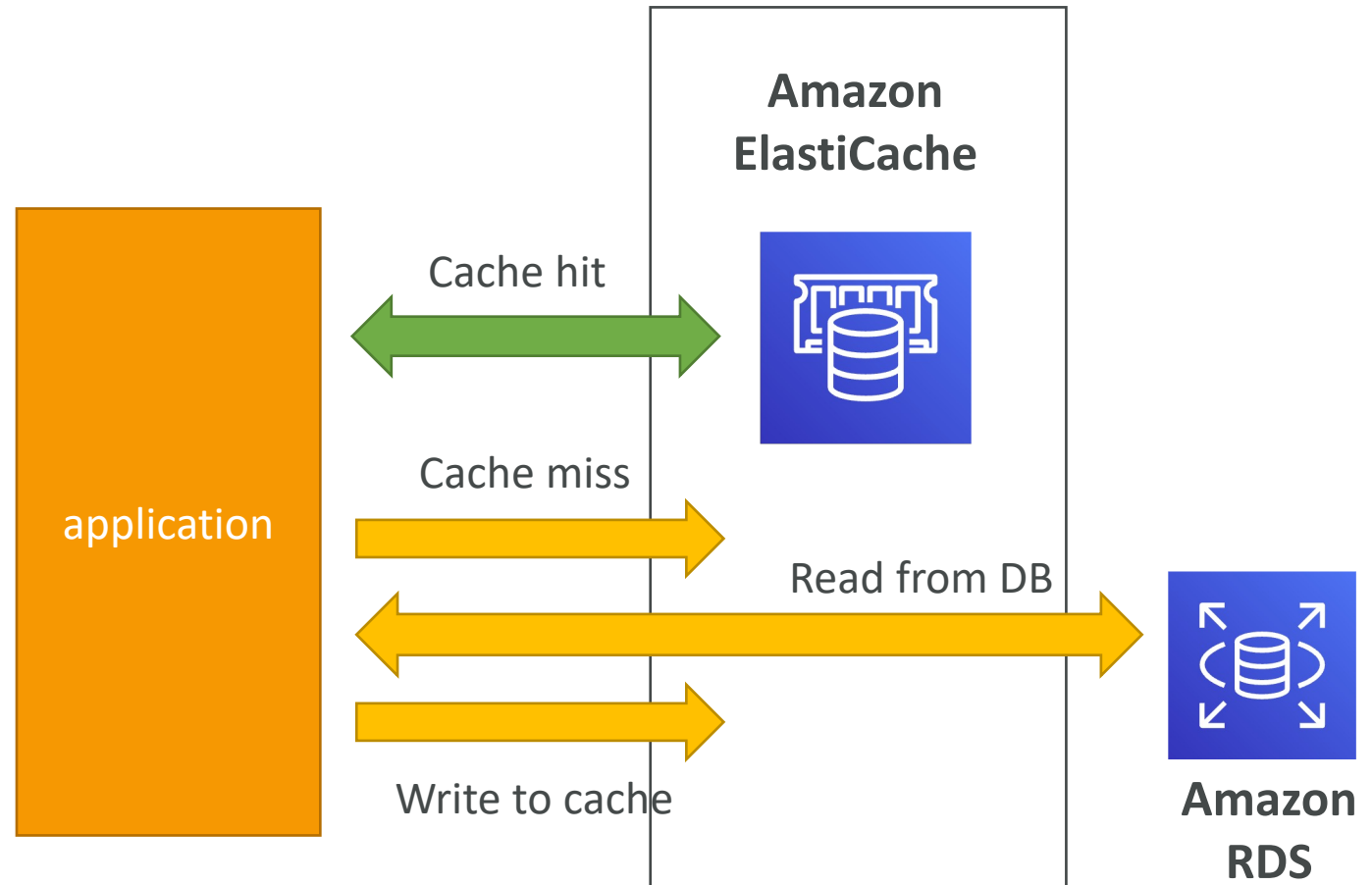
# Amazon ElastiCache Overview

- The same way RDS is to get managed Relational Databases...
- ElastiCache is to get managed Redis or Memcached
- Caches are in-memory databases with really high performance, low latency
- Helps reduce load off of databases for read intensive workloads
- Helps make your application stateless
- AWS takes care of OS maintenance / patching, optimizations, setup, configuration, monitoring, failure recovery and backups
- Using ElastiCache involves heavy application code changes

# ElastiCache

## Solution Architecture - DB Cache

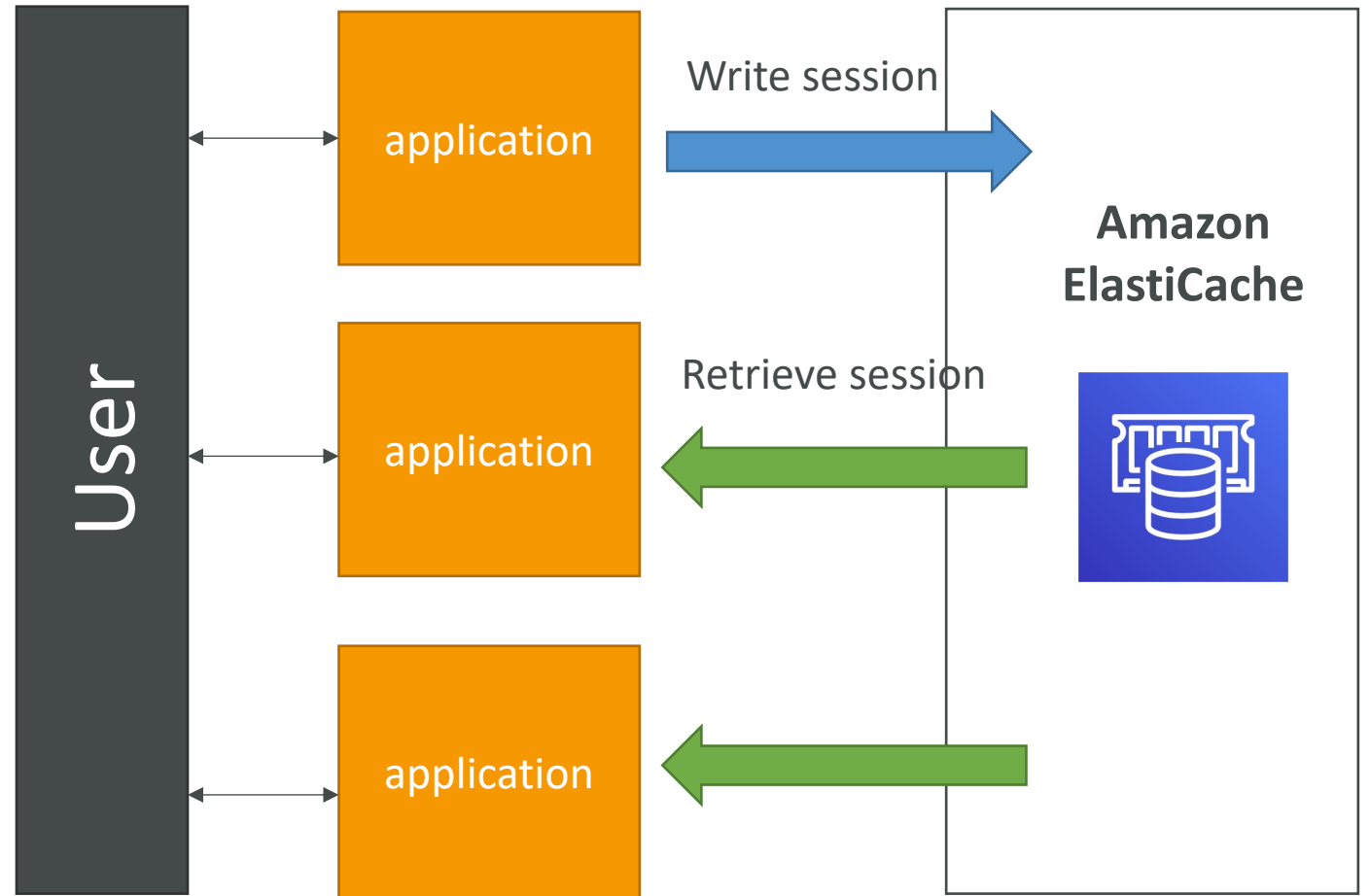
- Applications queries ElastiCache, if not available, get from RDS and store in ElastiCache.
- Helps relieve load in RDS
- Cache must have an invalidation strategy to make sure only the most current data is used in there.



# 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



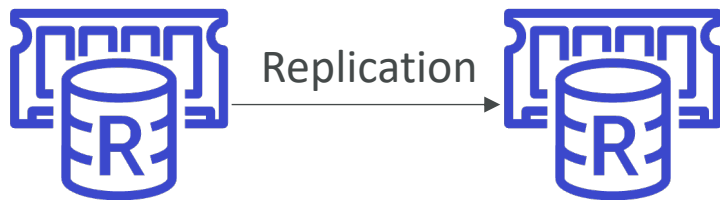
# ElastiCache – Redis vs Memcached

## REDIS

- Multi AZ with Auto-Failover
- Read Replicas to scale reads and have high availability
- Data Durability using AOF persistence
- Backup and restore features

## MEMCACHED

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

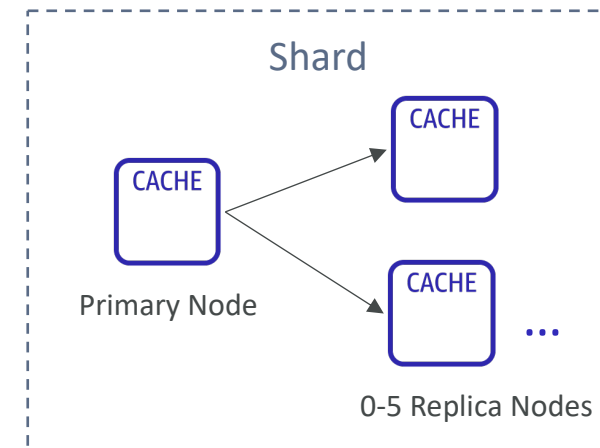




# ElastiCache Replication: Cluster Mode Disabled

- One primary node, up to 5 replicas
- Asynchronous Replication
- The primary node is used for read/write
- The other nodes are read-only
- One shard, all nodes have all the data
- Guard against data loss if node failure
- Multi-AZ enabled by default for failover
- Helpful to scale read performance

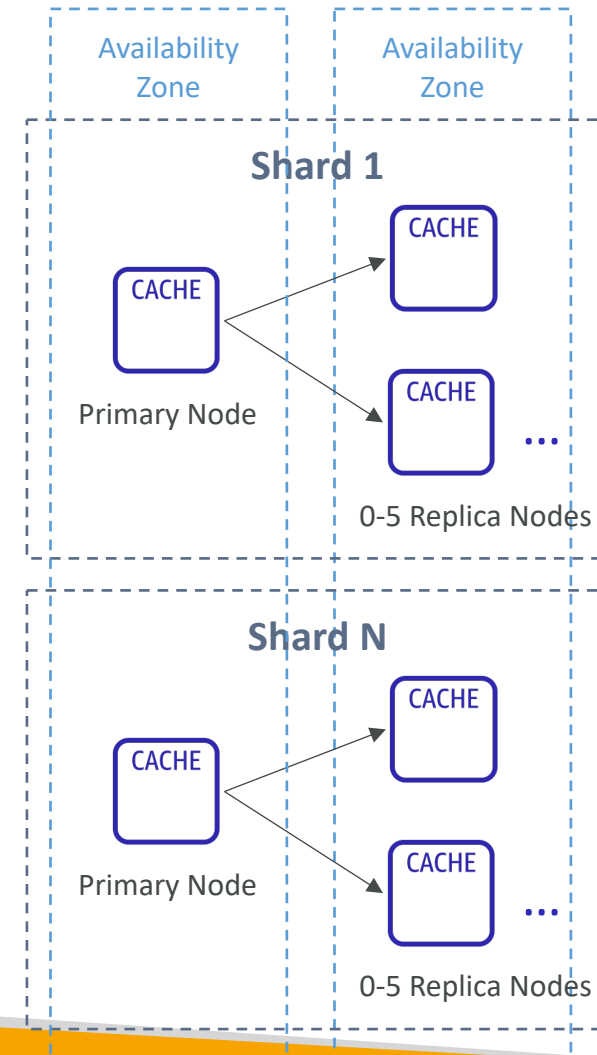
Redis (cluster mode disabled) Cluster with Replication



# ElastiCache Replication: Cluster Mode Enabled

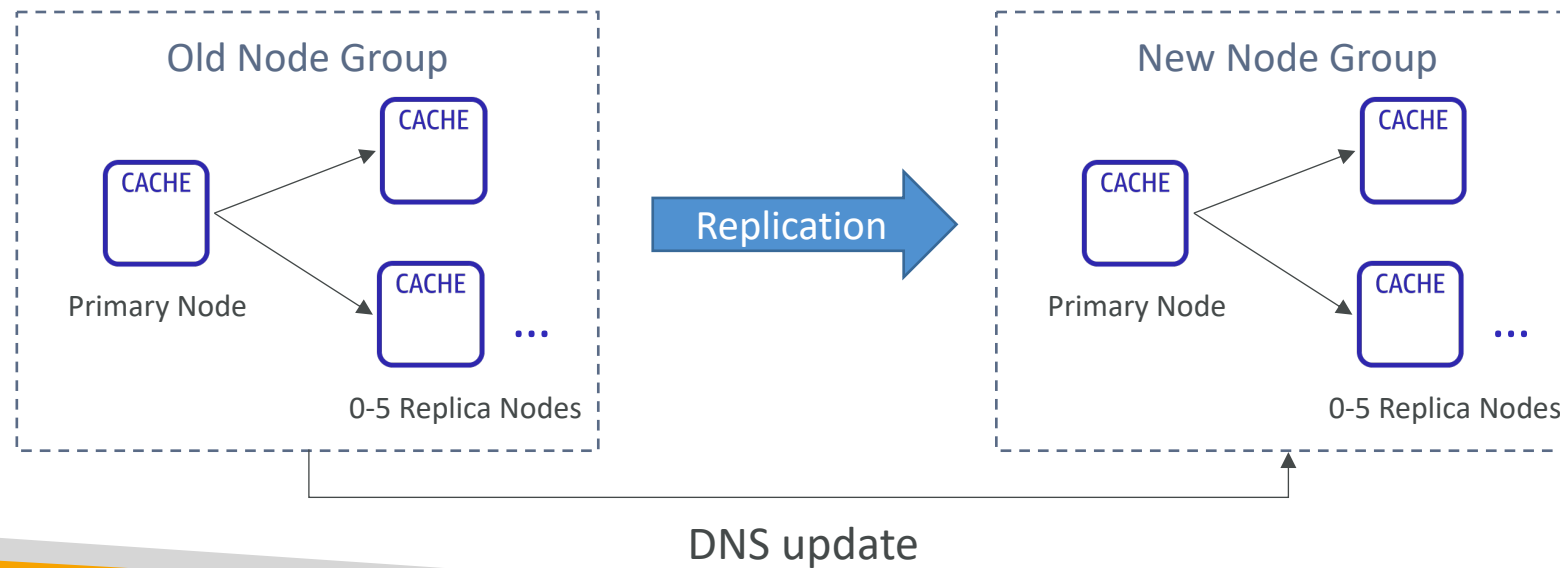
Redis (cluster mode enabled) Cluster  
with Replication

- Data is partitioned across shards (helpful to scale writes)
- Each shard has a primary and up to 5 replica nodes (same concept as before)
- Multi-AZ capability
- Up to 500 nodes per cluster:
  - 500 shards with single master
  - 250 shards with 1 master and 1 replica
  - ...
  - 83 shards with one master and 5 replicas



# Redis Scaling – Cluster Mode Disabled

- **Horizontal:**
  - Scale out/in by adding/removing read replicas (max. 5 replicas)
- **Vertical:**
  - Scale up/down to larger/smaller node type
  - ElastiCache will internally create a new node group, then data replication and DNS update



# Redis Scaling – Cluster Mode Enabled

- Two Modes:
  - **Online Scaling:** continue serving requests during the scaling process (no downtime, some degradation in performance)
  - **Offline Scaling:** unable to serve requests during the scaling process (backup and restore). Additional configurations supported (change node type, upgrade engine version, ...)
- **Horizontal:** (Resharding and Shard Rebalancing)
  - **Resharding:** scale out/in by adding/removing shards
  - **Shard Rebalancing:** equally distribute the keyspaces among the shards as possible
  - Supports **Online** and **Offline Scaling**
- **Vertical:** (change read/write capacity)
  - Scale up/down to larger/smaller node type
  - Supports **Online Scaling**

# Redis Metrics to Monitor

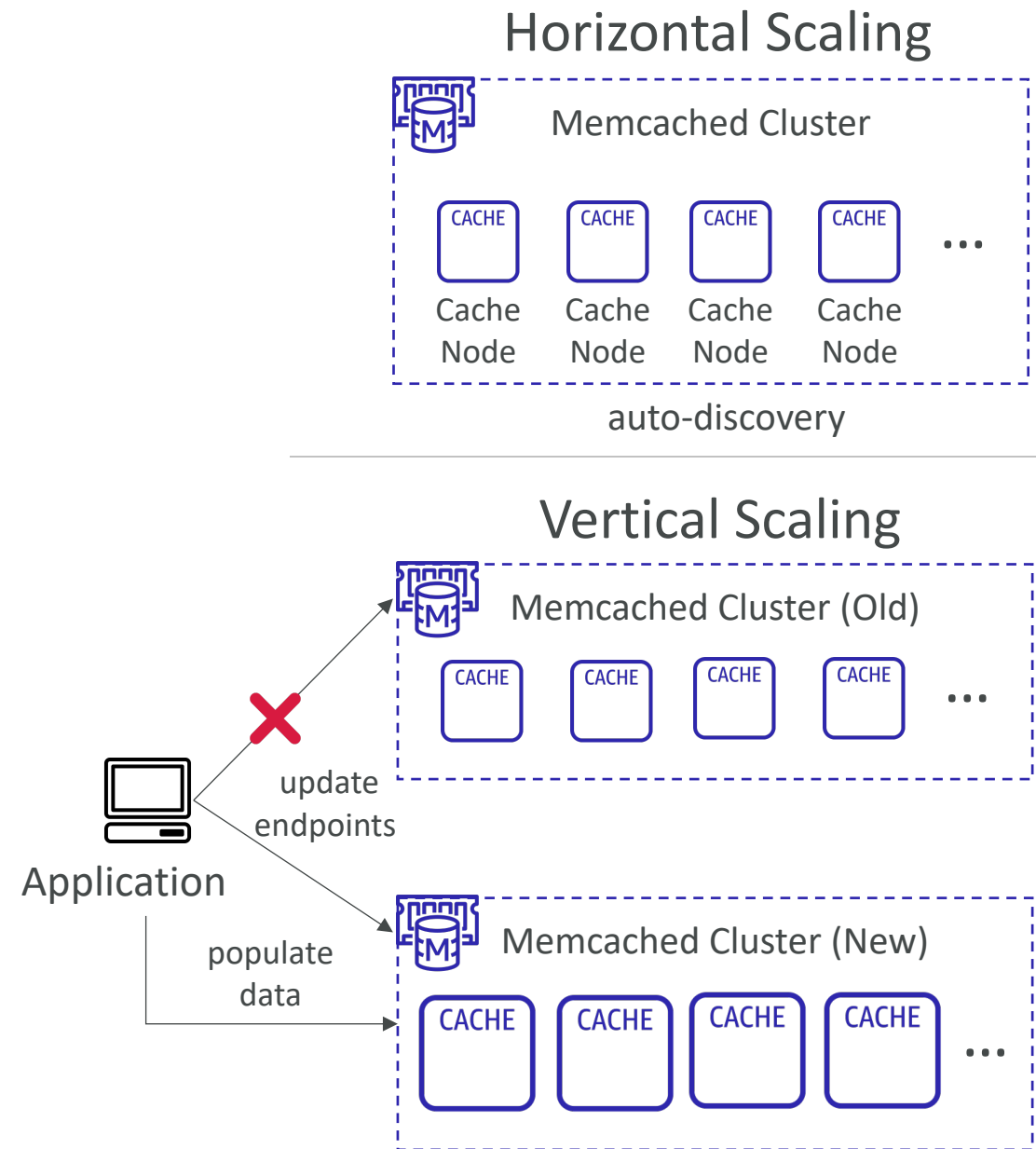
- **Evictions:** the number of non-expired items the cache evicted to allow space for new writes (memory is overfilled). Solution:
  - Choose an eviction policy to evict expired items (e.g., evict least recently used (LRU) items)
  - Scale up to larger node type (more memory) or scale out by adding more nodes
- **CPU Utilization:** monitor CPU utilization for the entire host
  - Solution: scale up to larger node type (more memory) or scale out by adding more nodes
- **Swap Usage:** should not exceed 50 MB
  - Solution: verify that you have configured enough reserved memory

# Redis Metrics to Monitor

- **CurrConnections:** the number of concurrent and active connections
  - Solution: investigate application behavior to address the issue
- **DatabaseMemoryUsagePercentage:** the percentage of memory utilization
- **NetworkBytesIn/Out & NetworkPacketsIn/Out**
- **ReplicationBytes:** the volume of data being replicated
- **ReplicationLag:** how far behind the replica is from the primary node

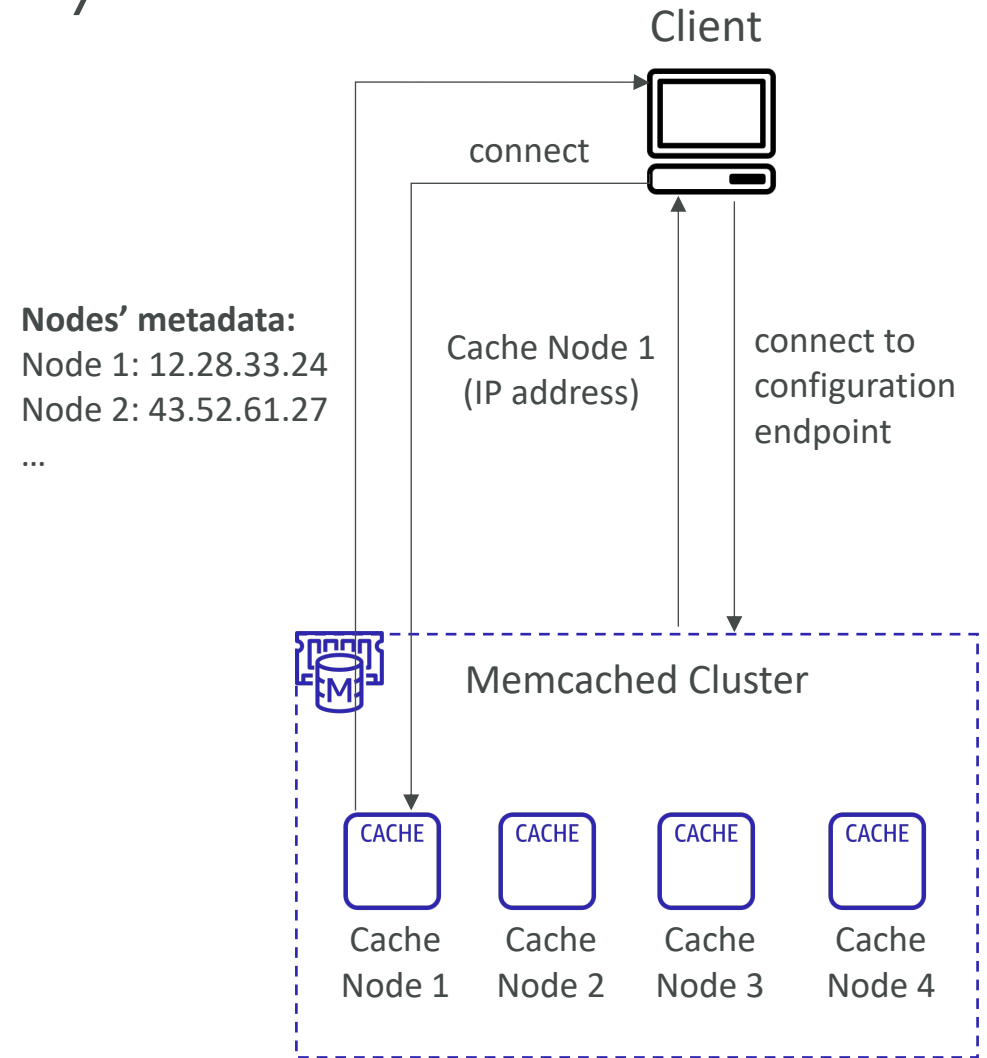
# Memcached Scaling

- Memcached Clusters can have 1-40 nodes (soft limit)
- **Horizontal:**
  - Add/remove nodes from the cluster
  - [Auto-discovery](#) allow your app to find nodes
- **Vertical:**
  - Scale up/down to larger/smaller node type
  - Scale up process:
    - Create a new cluster with the new node type
    - Update your application to use the new cluster's endpoints
    - Delete the old cluster
  - Memcached clusters/nodes start out empty



# Memcached Auto Discovery

- Typically you need to manually connect to individual cache nodes in the cluster using its DNS endpoints
- Auto Discovery automatically identifies all of the nodes
- All the cache nodes in the cluster maintain a list of metadata about all other nodes
- This is seamless from a client perspective





# Memcached Metrics to Monitor

- **Evictions:** the number of non-expired items the cache evicted to allow space for new writes (memory is overfilled). Solution:
  - Choose an eviction policy to evict expired items (e.g., LRU items)
  - Scale up to larger node type (more memory) or scale out by adding more nodes
- **CPUUtilization**
  - Solution: scale up to larger node type or scale out by adding more nodes
- **SwapUsage:** should not exceed 50 MB
- **CurrConnections:** the number of concurrent and active connections
  - Solution: investigate application behavior to address the issue
- **FreeableMemory:** amount of free memory on the host

# AWS Monitoring, Audit and Performance

CloudWatch, CloudTrail & AWS Config



# AWS CloudWatch Metrics

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

# EC2 Detailed monitoring

- EC2 instance metrics have metrics “every 5 minutes”
- With detailed monitoring (for a cost), you get data “every 1 minute”
- Use detailed monitoring if you want to scale faster for your ASG!
- The AWS Free Tier allows us to have 10 detailed monitoring metrics
- Note: EC2 Memory usage is by default not pushed (must be pushed from inside the instance as a custom metric)

# CloudWatch Custom Metrics

- Possibility to define and send your own custom metrics to CloudWatch
- Example: memory (RAM) usage, disk space, number of logged in users ...
- Use API call **PutMetricData**
- Ability to use dimensions (attributes) to segment metrics
  - Instance.id
  - Environment.name
- Metric resolution (**StorageResolution** API parameter – two possible value):
  - Standard: 1 minute (60 seconds)
  - High Resolution: 1/5/10/30 second(s) – Higher cost
- **Important:** Accepts metric data points two weeks in the past and two hours in the future (make sure to configure your EC2 instance time correctly)

# CloudWatch Dashboards

- Great way to setup custom dashboards for quick access to key metrics and alarms
- Dashboards are global
- Dashboards can include graphs from different AWS accounts and regions
- You can change the time zone & time range of the dashboards
- You can setup automatic refresh (10s, 1m, 2m, 5m, 15m)
- Dashboards can be shared with people who don't have an AWS account (public, email address, 3<sup>rd</sup> party SSO provider through Amazon Cognito)
- Pricing:
  - 3 dashboards (up to 50 metrics) for free
  - \$3/dashboard/month afterwards

# CloudWatch Logs



- **Log groups:** arbitrary name, usually representing an application
- **Log stream:** instances within application / log files / containers
- Can define log expiration policies (never expire, 30 days, etc..)
- **CloudWatch Logs can send logs to:**
  - Amazon S3 (exports)
  - Kinesis Data Streams
  - Kinesis Data Firehose
  - AWS Lambda
  - OpenSearch

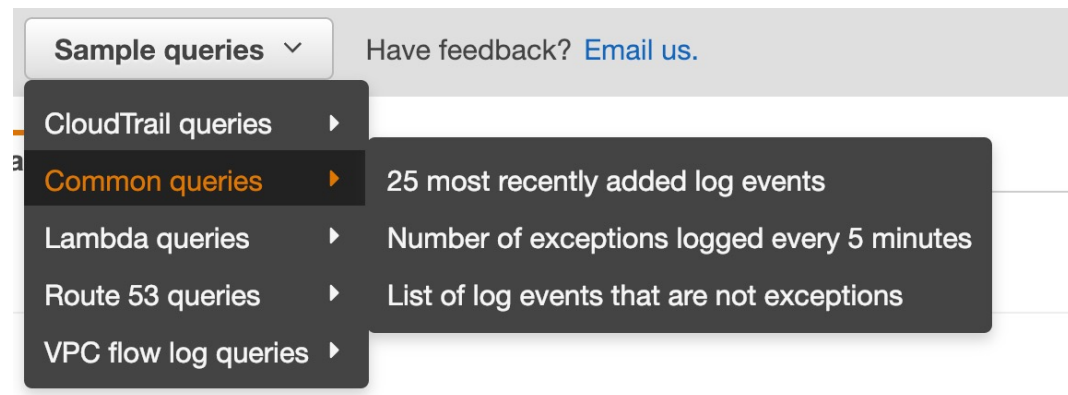
# CloudWatch Logs - Sources

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



# CloudWatch Logs Metric Filter & Insights

- CloudWatch Logs can use filter expressions
  - For example, find a specific IP inside of a log
  - Or count occurrences of “ERROR” in your logs
- Metric filters can be used to trigger CloudWatch alarms
- CloudWatch Logs Insights can be used to query logs and add queries to CloudWatch Dashboards

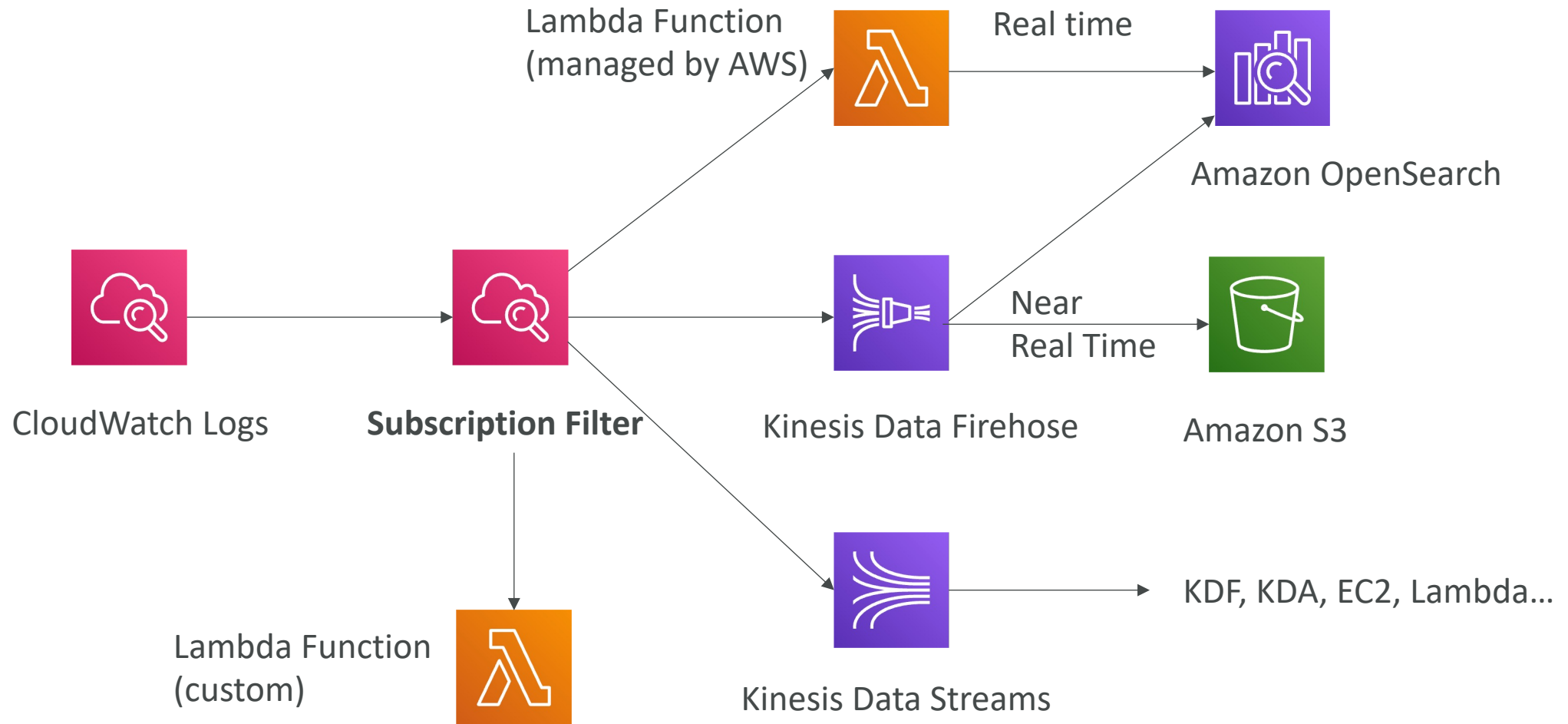


# CloudWatch Logs – S3 Export

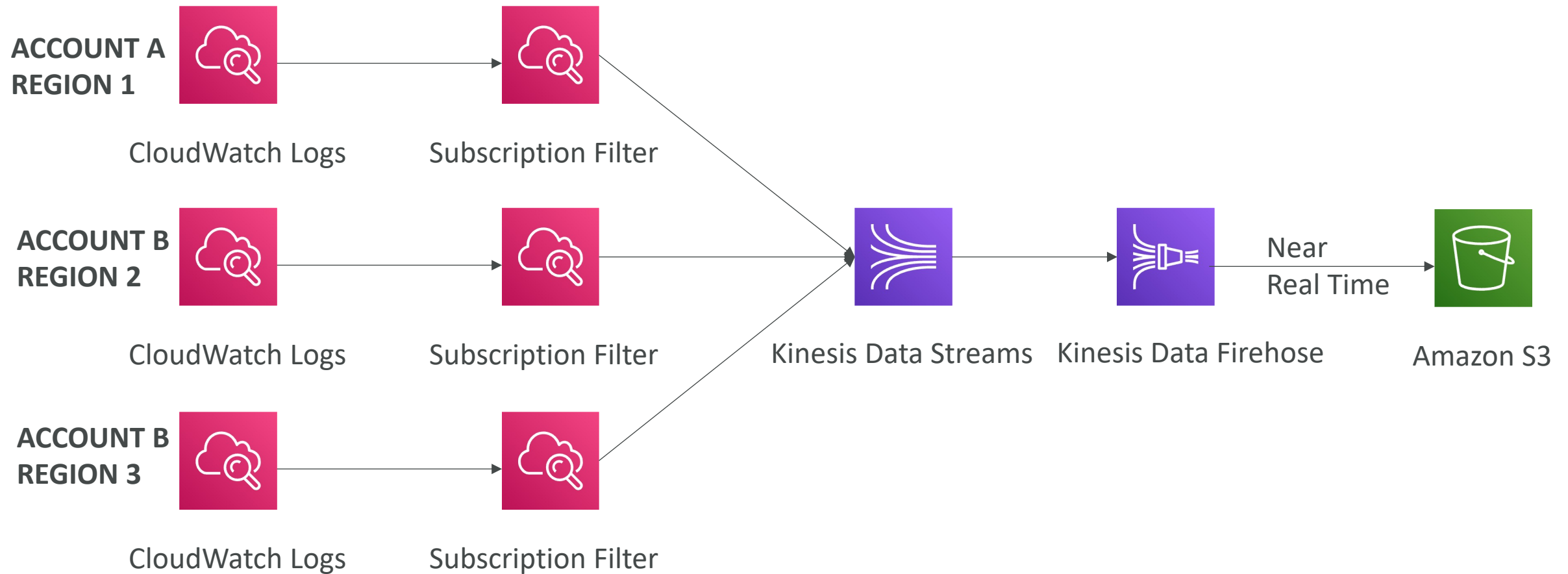


- Log data can take **up to 12 hours** to become available for export
- The API call is **CreateExportTask**
- Not near-real time or real-time... use Logs Subscriptions instead

# CloudWatch Logs Subscriptions



# CloudWatch Logs Aggregation Multi-Account & Multi Region



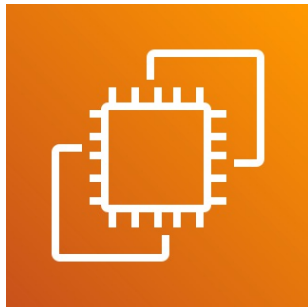
# CloudWatch Alarms

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

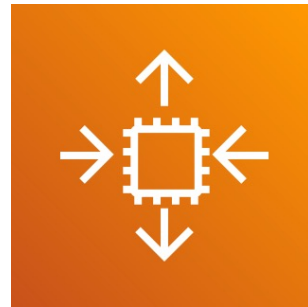


# CloudWatch Alarm Targets

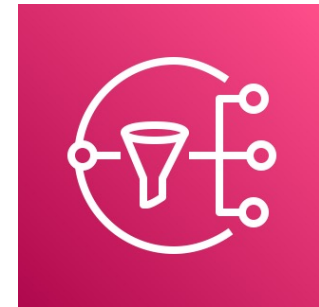
- Stop, Terminate, Reboot, or Recover an EC2 Instance
- Trigger Auto Scaling Action
- Send notification to SNS (from which you can do pretty much anything)



Amazon EC2



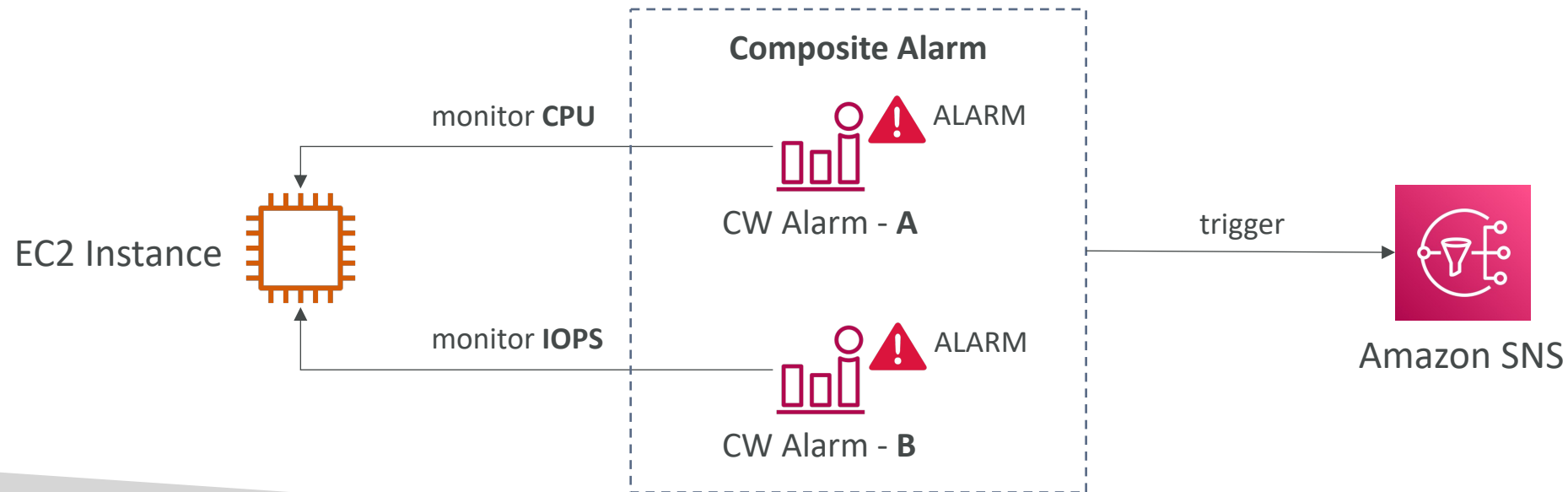
EC2 Auto Scaling



Amazon SNS

# CloudWatch Alarms – Composite Alarms

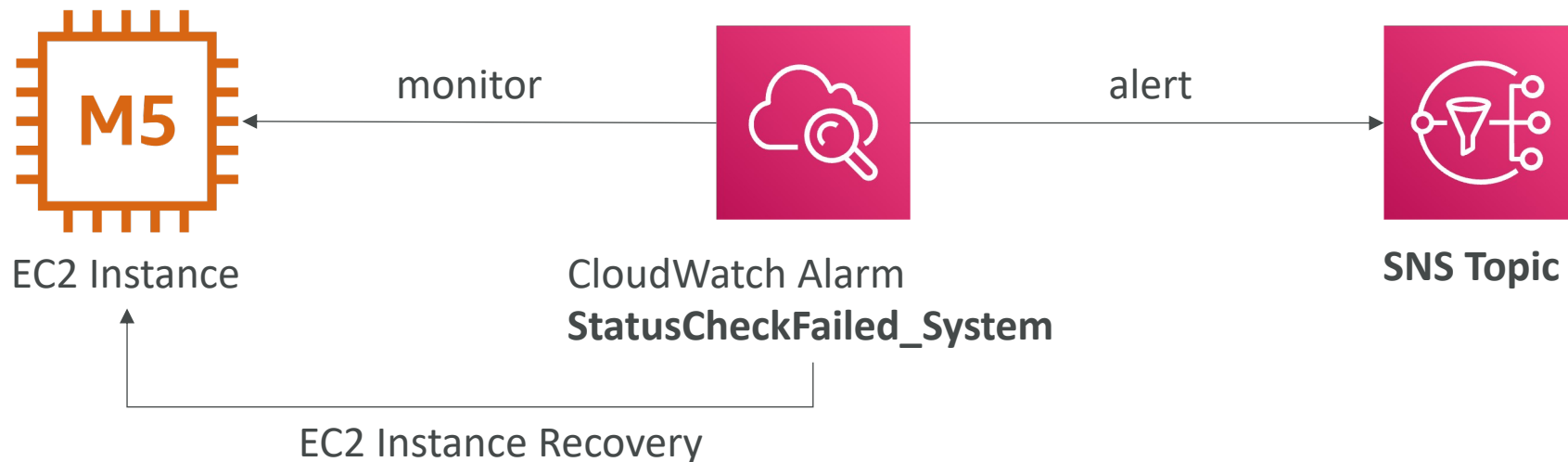
- CloudWatch Alarms are on a single metric
- Composite Alarms are monitoring the states of multiple other alarms
- AND and OR conditions
- Helpful to reduce “alarm noise” by creating complex composite alarms



# EC2 Instance Recovery

- Status Check:

- Instance status = check the EC2 VM
- System status = check the underlying hardware

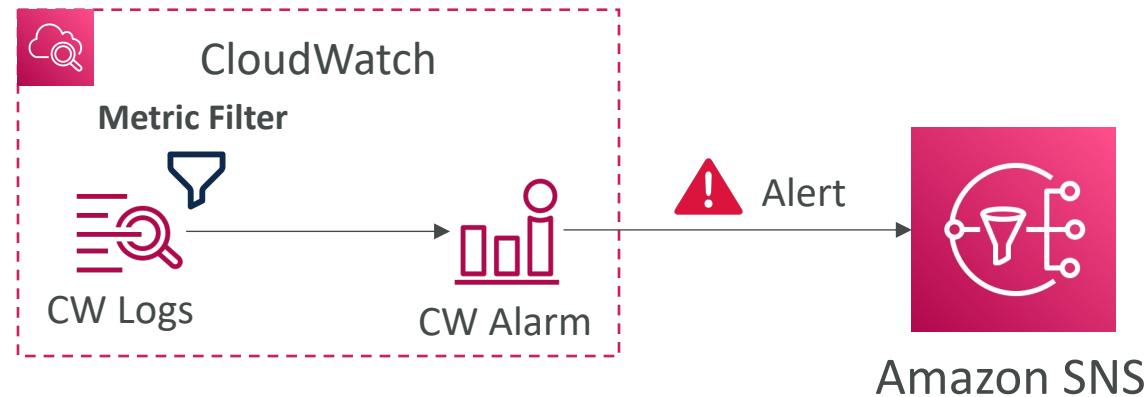


- **Recovery:** Same Private, Public, Elastic IP, metadata, placement group



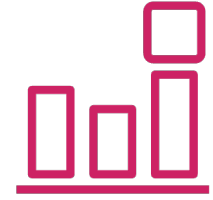
# CloudWatch Alarm: good to know

- Alarms can be created based on CloudWatch Logs Metrics Filters



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

# CloudWatch Events



- Event Pattern: Intercept events from AWS services (Sources)
  - Example sources: EC2 Instance Start, CodeBuild Failure, S3, Trusted Advisor
  - Can intercept any API call with CloudTrail integration
- Schedule or Cron (example: create an event every 4 hours)
  
- A JSON payload is created from the event and passed to a target...
  - **Compute:** Lambda, Batch, ECS task
  - **Integration:** SQS, SNS, Kinesis Data Streams, Kinesis Data Firehose
  - **Orchestration:** Step Functions, CodePipeline, CodeBuild
  - **Maintenance:** SSM, EC2 Actions



# Amazon EventBridge

- EventBridge is the next evolution of CloudWatch Events
- **Default Event Bus** – generated by AWS services (CloudWatch Events)
- **Partner Event Bus** – receive events from SaaS service or applications (Zendesk, DataDog, Segment, Auth0...)
- **Custom Event Buses** – for your own applications
- Event buses can be accessed by other AWS accounts
- You can **archive events** (all/filter) sent to an event bus (indefinitely or set period)
- Ability to **replay archived events**
  
- **Rules:** how to process the events (like CloudWatch Events)

# Amazon EventBridge – Schema Registry

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

aws.codepipeline@CodePipelineActionExecuti...

### Schema details

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

### Version 1 Created on Dec 1, 2019, 12:11 AM GMT

Action ▾ [Download code bindings](#)

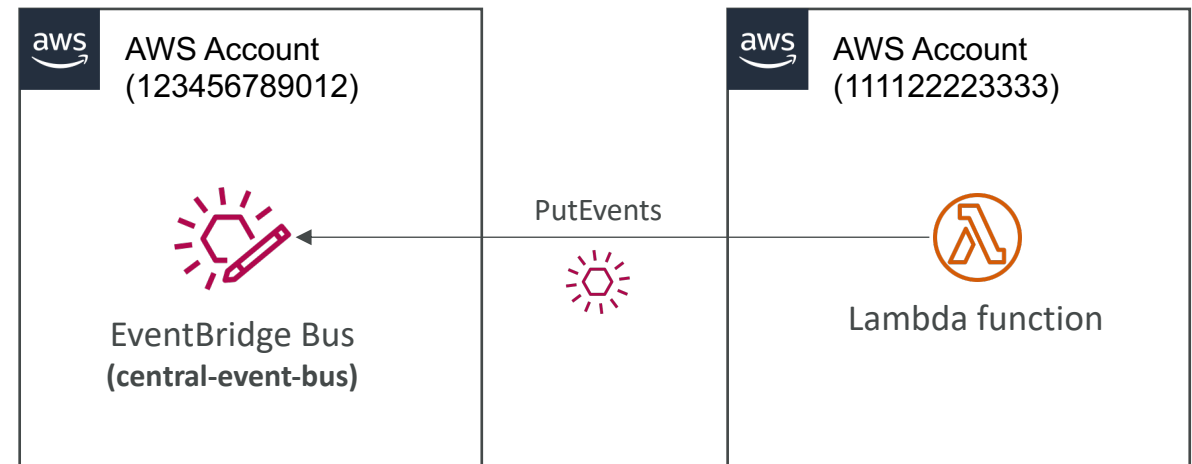
```
1 {
2   "openapi": "3.0.0",
3   "info": {
4     "version": "1.0.0",
5     "title": "CodePipelineActionExecutionStateChange"
6   },
7   "paths": {},
8   "components": {
9     "schemas": {
10      "ANSEvent": {
```

# Amazon EventBridge – Resource-based Policy

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

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

**Allow events from another AWS account**



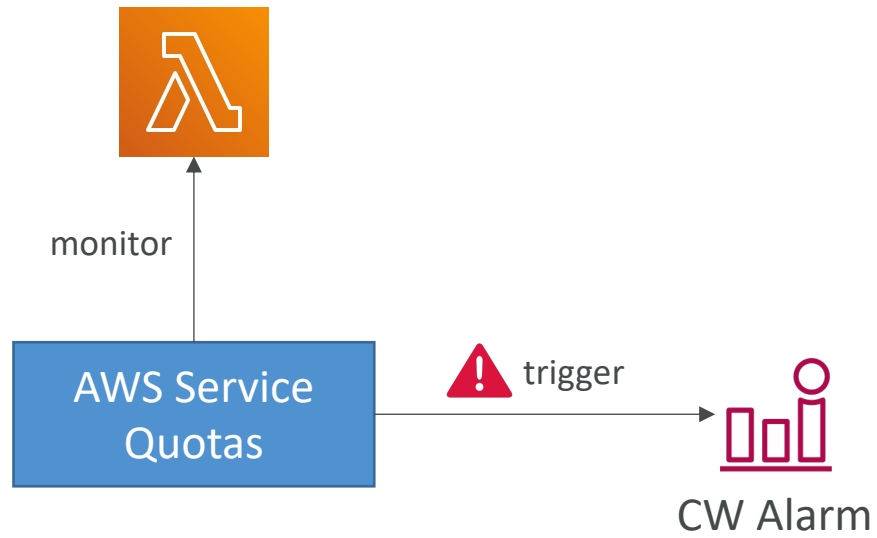
# Amazon EventBridge vs CloudWatch Events

- Amazon EventBridge builds upon and extends CloudWatch Events.
- It uses the same service API and endpoint, and the same underlying service infrastructure.
- EventBridge allows extension to add event buses for your custom applications and your third-party SaaS apps.
- Event Bridge has the Schema Registry capability
- EventBridge has a different name to mark the new capabilities
- Over time, the CloudWatch Events name will be replaced with EventBridge.

# Service Quotas CloudWatch Alarms

- Notify you when you're close to a service quota value threshold
- Create CloudWatch Alarms on the Service Quotas console
- Example: Lambda concurrent executions
- Helps you know if you need to request a quota increase or shutdown resources before limit is reached

## AWS Lambda Quota



### Create a CloudWatch alarm: Concurrent executions

Description  
The maximum number of events that functions can process simultaneously in the current Region.

Alarm threshold  
This alarm will notify you based on the threshold you choose.

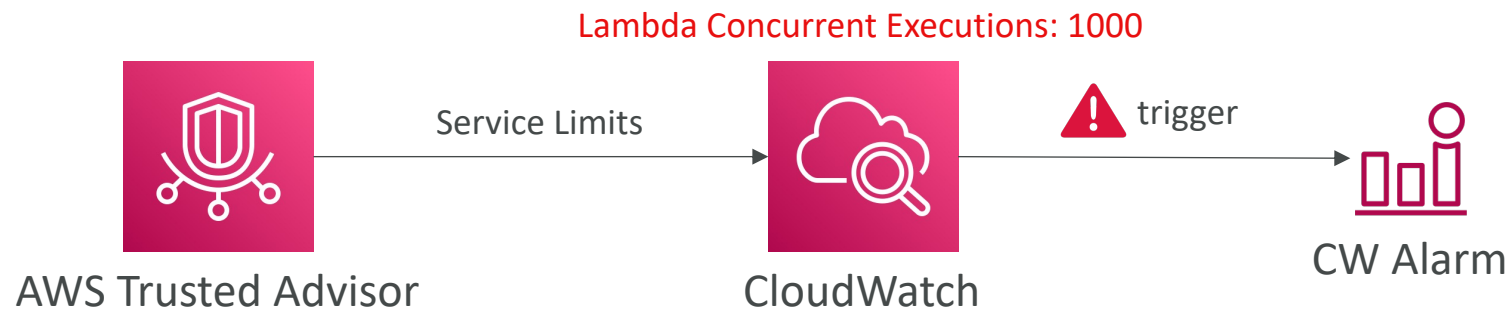
Alarm name  
  
Required. Alarm names must be unique within an AWS account.

Region  
US East (N. Virginia) us-east-1

Pricing  
Using CloudWatch can incur costs. [CloudWatch pricing](#)

# Alternative: Trusted Advisor + CW Alarms

- Limited number of Service Limits checks in Trusted Advisor (~50)
- Trusted Advisor publishes its check results to CloudWatch
- You can create CloudWatch Alarms on service quota usage (Service Limits)



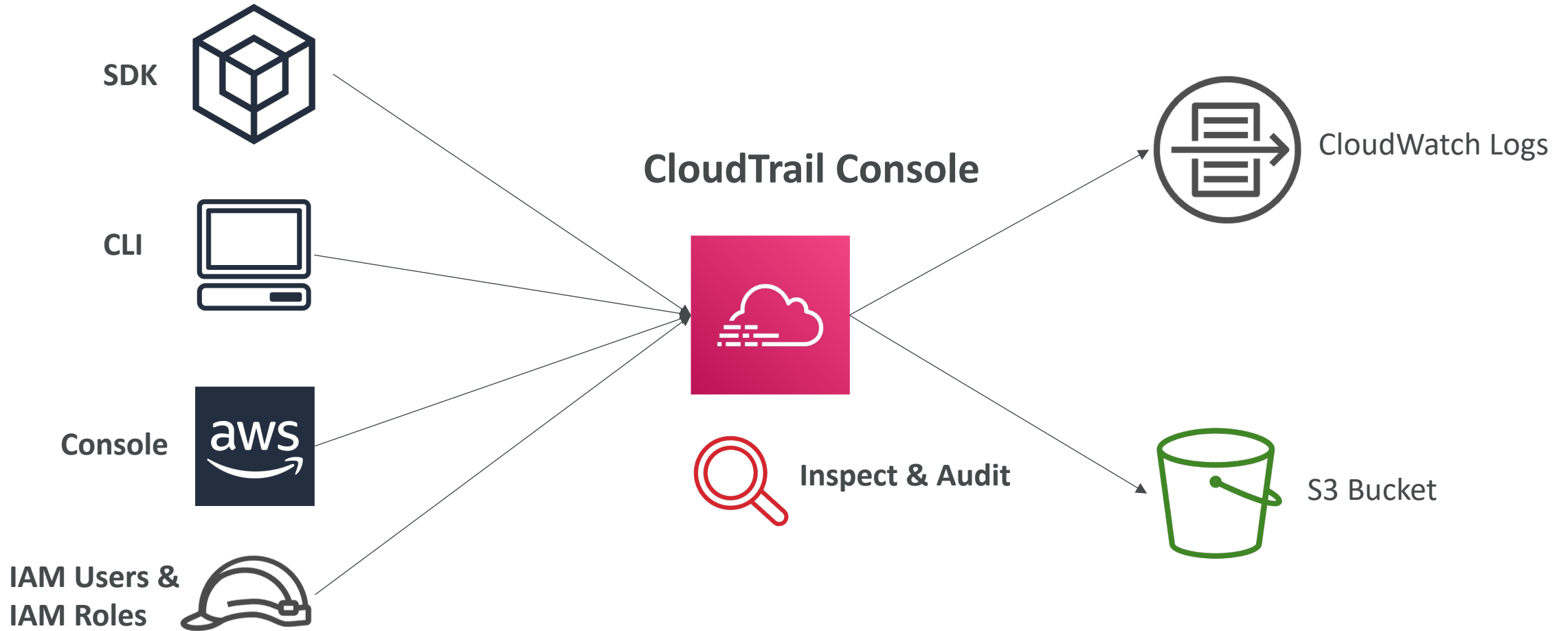


# AWS CloudTrail



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

# CloudTrail Diagram



# CloudTrail Events

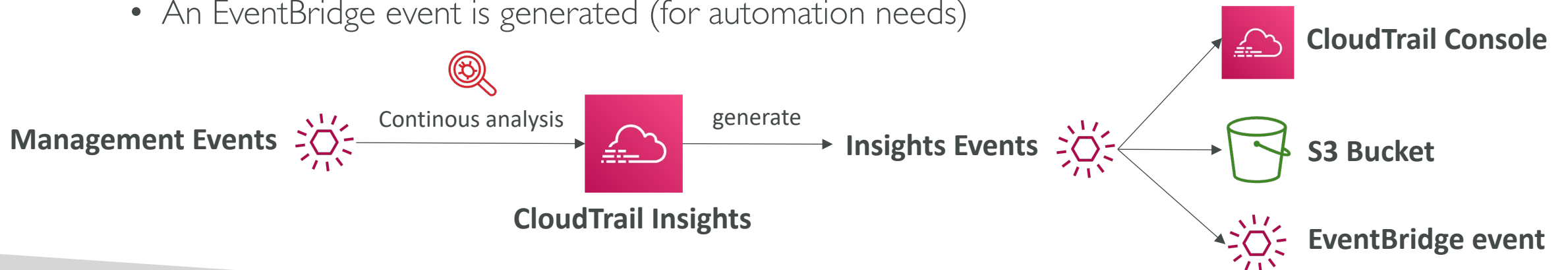


- **Management Events:**
  - Operations that are performed on resources in your AWS account
  - Examples:
    - Configuring security (IAM **AttachRolePolicy**)
    - Configuring rules for routing data (Amazon EC2 **CreateSubnet**)
    - Setting up logging (AWS CloudTrail **CreateTrail**)
  - **By default,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

# CloudTrail Insights

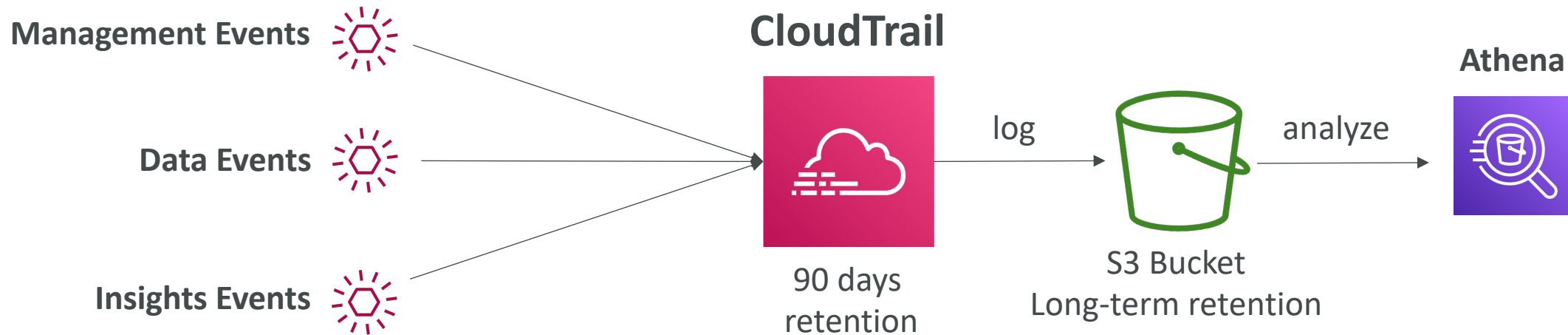


- Enable **CloudTrail Insights** to **detect unusual activity** in your account:
  - inaccurate resource provisioning
  - hitting service limits
  - Bursts of AWS IAM actions
  - Gaps in periodic maintenance activity
- CloudTrail Insights analyzes normal management events to create a baseline
- And then **continuously analyzes write events** to **detect unusual patterns**
  - Anomalies appear in the CloudTrail console
  - Event is sent to Amazon S3
  - An EventBridge event is generated (for automation needs)



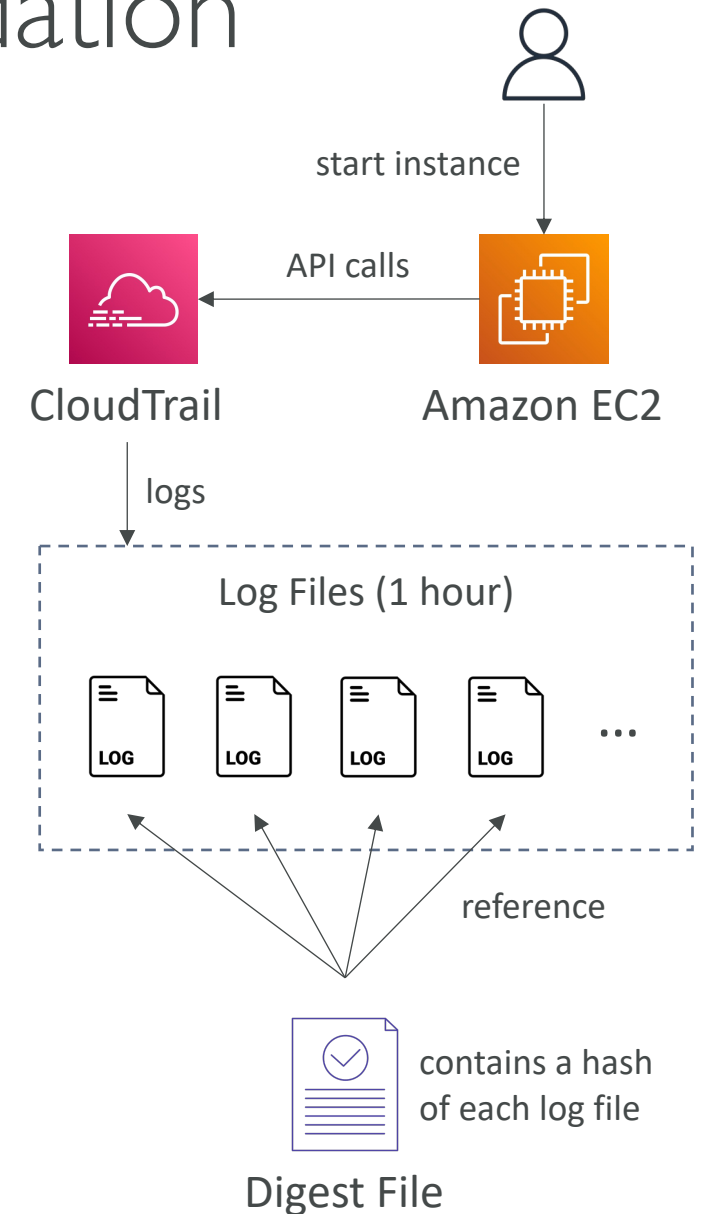
# CloudTrail Events Retention

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

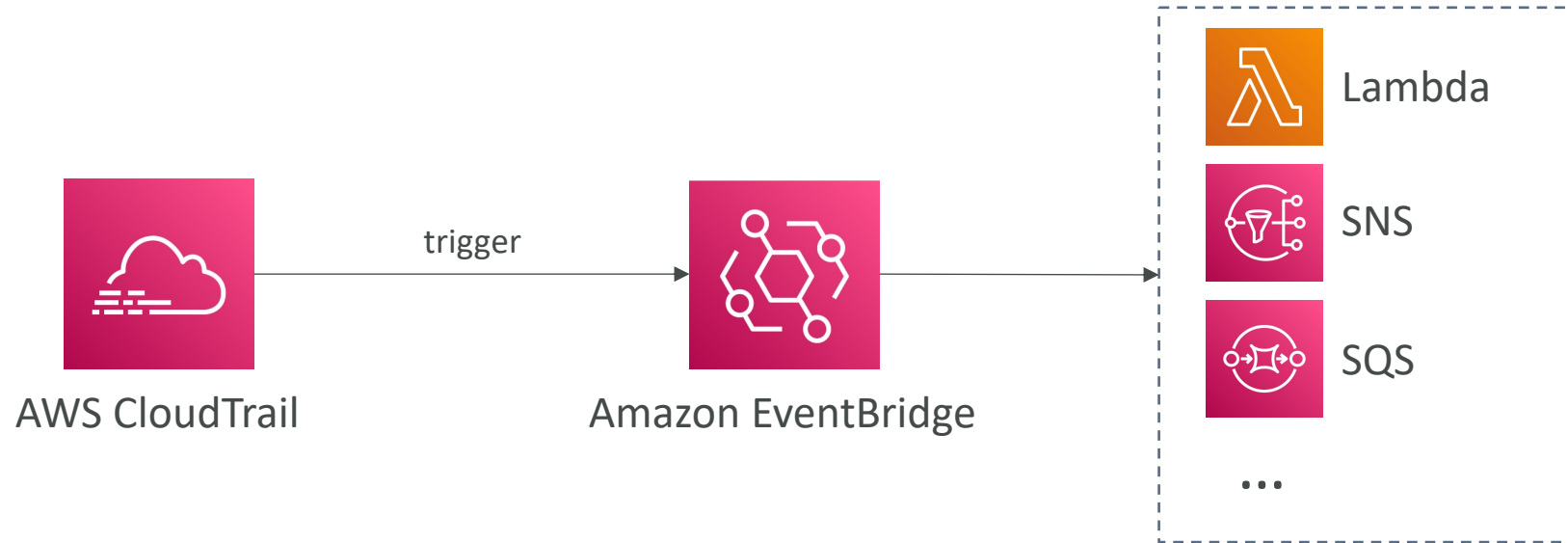


# CloudTrail – Log File Integrity Validation

- **Digest Files:**
  - References the log files for the last hour and contains a hash of each
  - Stored in the same S3 bucket as log files (different folder)
- Helps you determine whether a log file was modified/deleted after CloudTrail delivered it
- Hashing using SHA-256, Digital Signing using SHA-256 with RSA
- Protect the S3 bucket using bucket policy, versioning, MFA Delete protection, encryption, object lock
- Protect CloudTrail using IAM



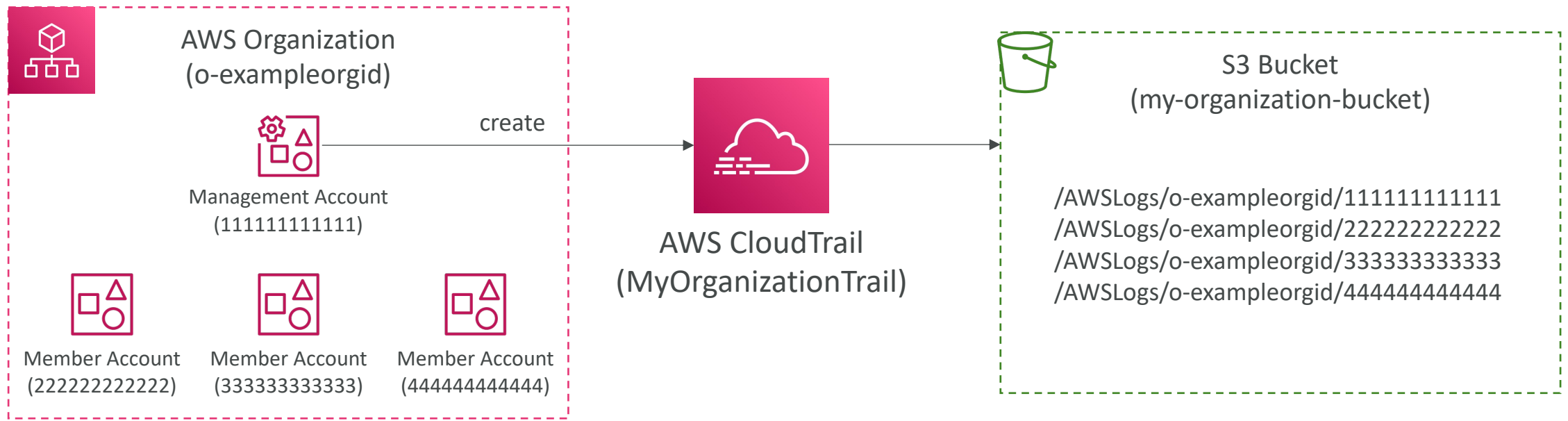
# CloudTrail – Integration with EventBridge



- Used to react to any API call being made in your account
- CloudTrail is not “real-time”:
  - Delivers an event within 15 minutes of an API call
  - Delivers log files to an S3 bucket every 5 minutes

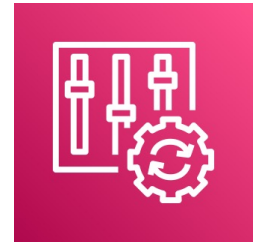
# CloudTrail – Organizations Trails

- A trail that will log all events for all AWS accounts in an AWS Organization
- Log events for management and member accounts
- Trail with the same name will be created in every AWS account (IAM permissions)
- Member accounts can't remove or modify the organization trail (view only)





# AWS Config



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

# Config Rules

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

# AWS Config Resource

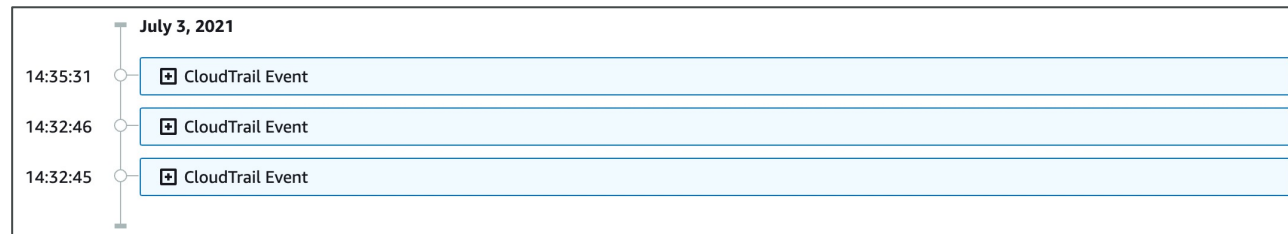
- View compliance of a resource over time

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

- View configuration of a resource over time

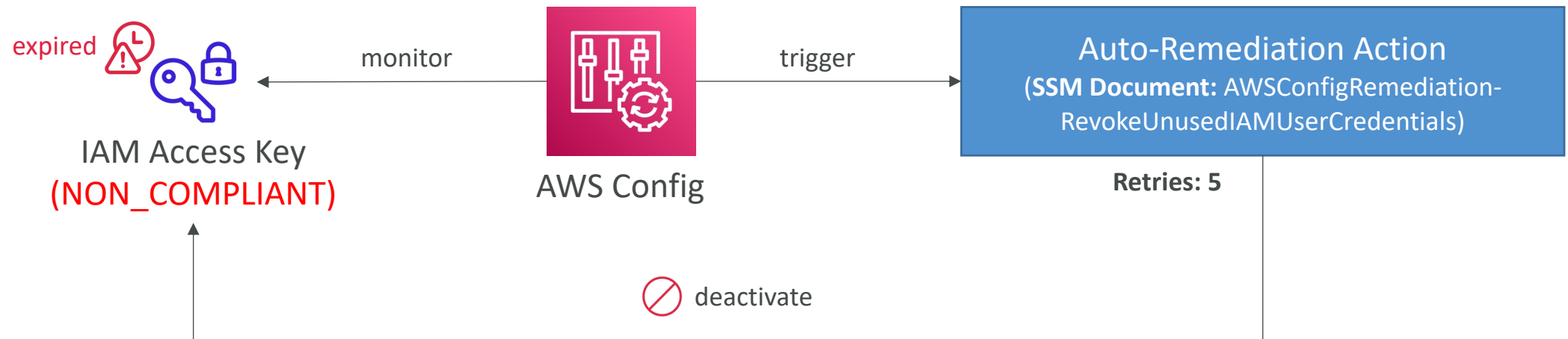


- View CloudTrail API calls of a resource over time



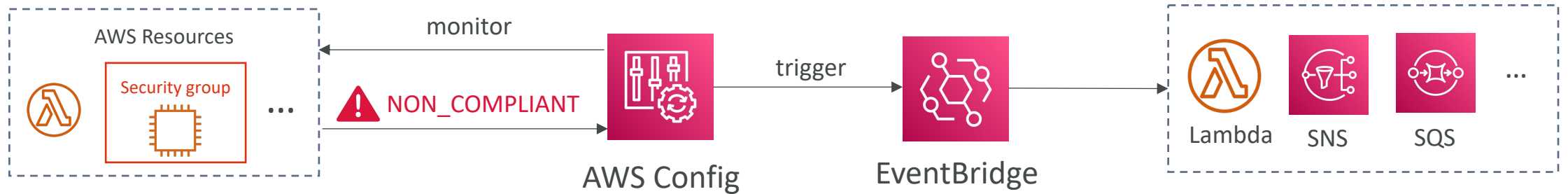
# Config Rules – Remediations

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

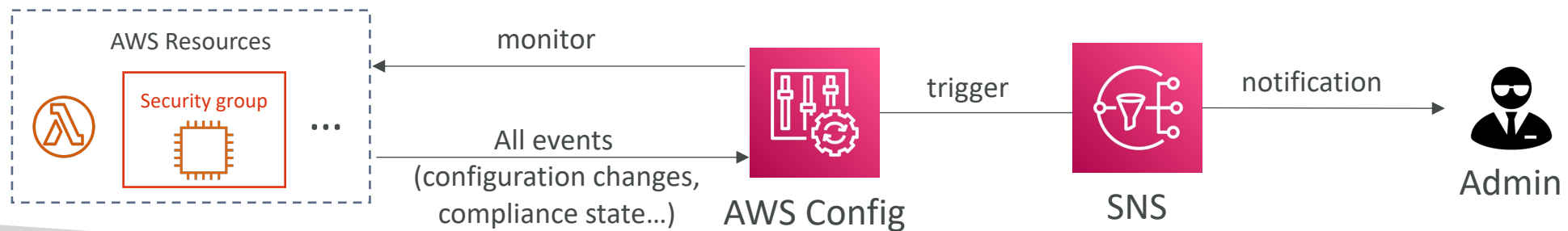


# Config Rules – Notifications

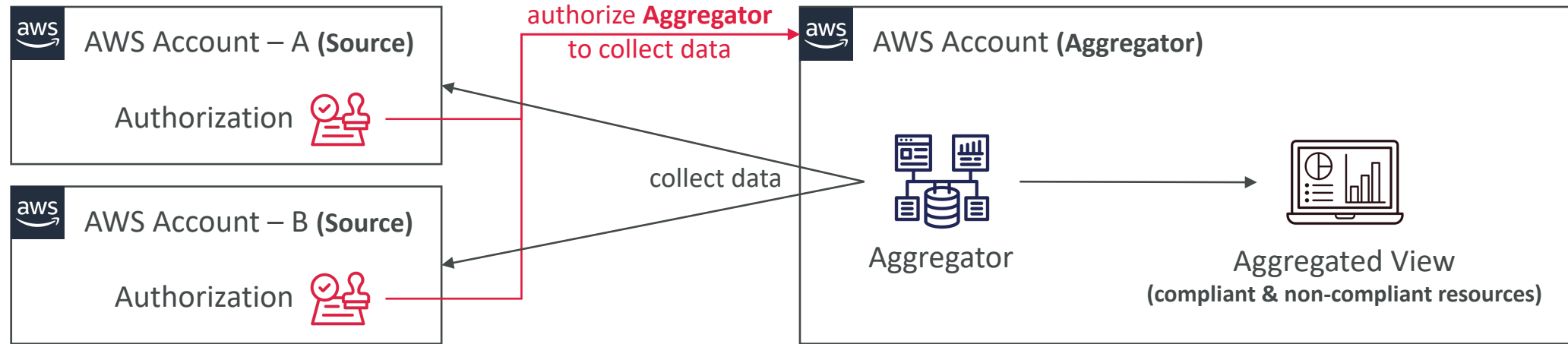
- Use EventBridge to trigger notifications when AWS resources are non-compliant



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



# AWS Config – Aggregators



- The aggregator is created in one central aggregator account
- Aggregates rules, resources, etc... across multiple accounts & regions
- If using AWS Organizations, no need for individual Authorization
- Rules are created in each individual source AWS account
- Can deploy rules to multiple target accounts using CloudFormation StackSets

# 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 a % 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



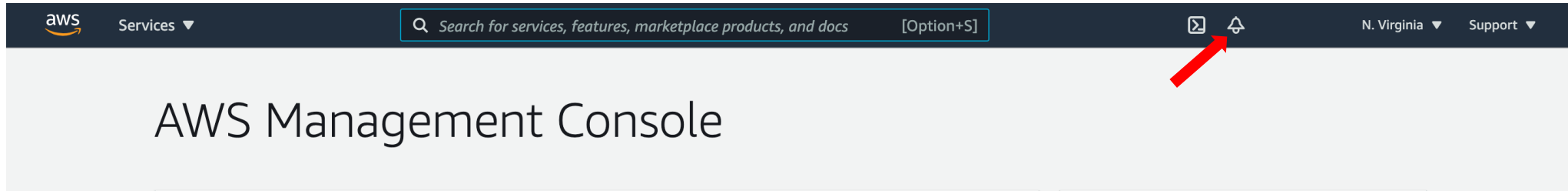
# AWS Account Management

Health Dashboards, AWS Organizations and Billing Console

# AWS Status - Service Health Dashboard

- Shows all regions, all services health
- Shows historical information for each day
- Has an RSS feed you can subscribe to
- <https://status.aws.amazon.com/>

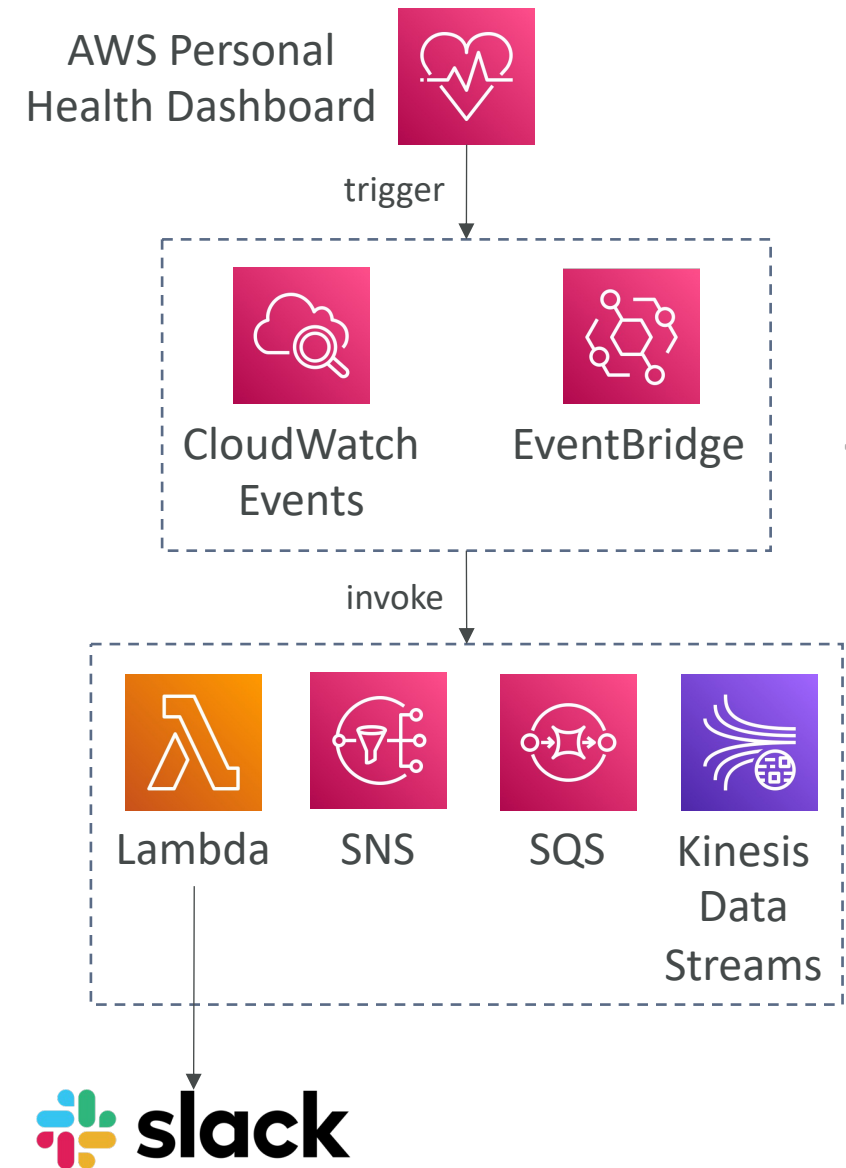
# AWS Personal Health Dashboard



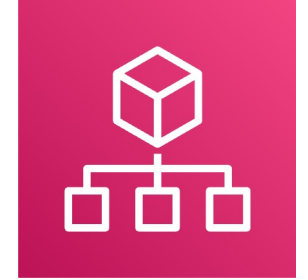
- Global service
- Show how AWS outages directly impact you
- Show the impact on your resources
- List issues and actions you can do to remediate them
- Will show **maintenance** events from AWS
- Programmatically accessible through the **AWS Health API**
- Aggregations across multiple accounts of an **AWS Organization**
- <https://phd.aws.amazon.com/>

# Health Event Notifications

- Use EventBridge (CloudWatch Events) to react to changes for AWS Health events in your AWS account
- Example: receive email notifications when EC2 instances in your AWS account are scheduled for updates
- Can't be used to return public events from the Service Health Dashboard
- Use cases: send notifications, capture event information, take corrective action, ...

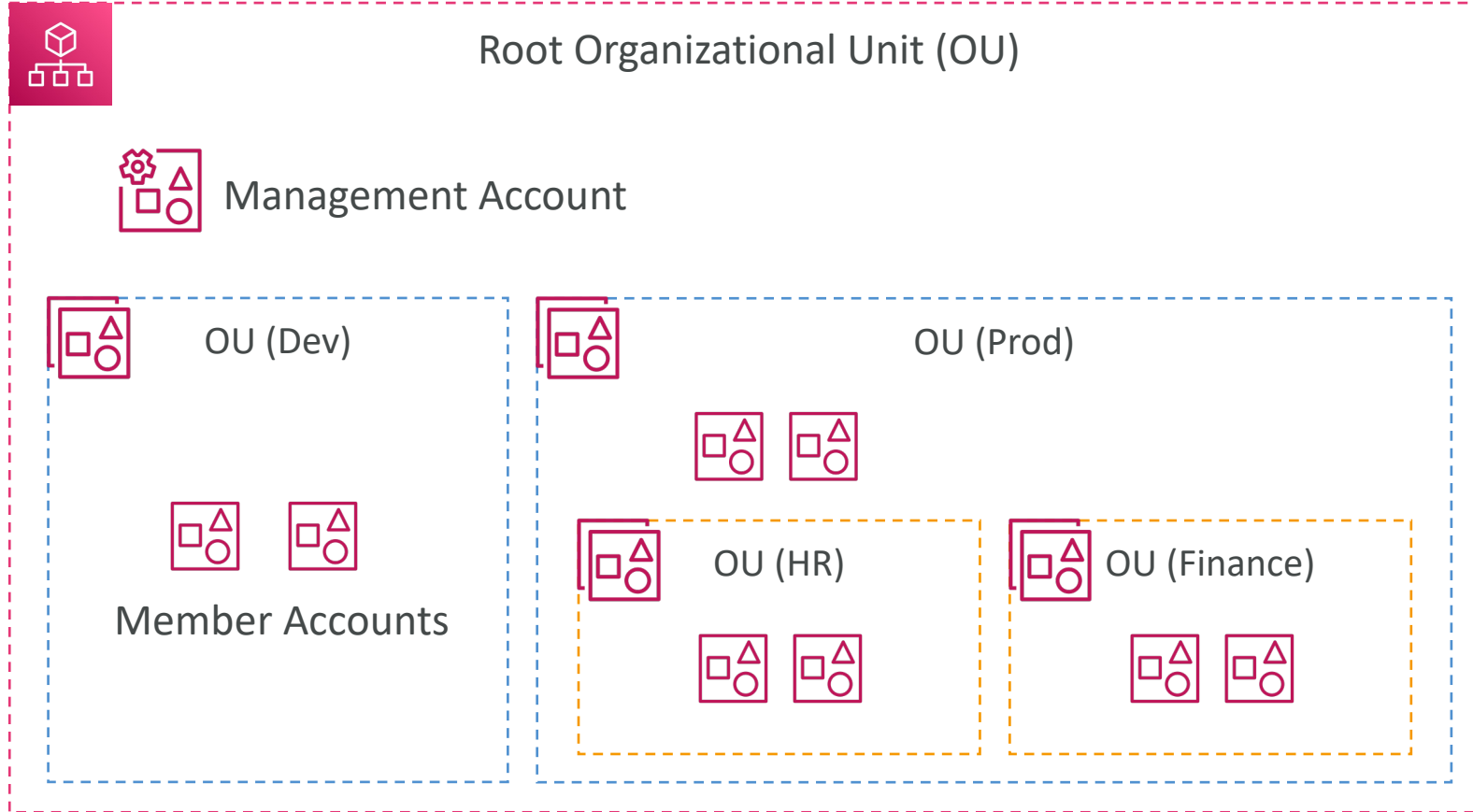


# AWS Organizations



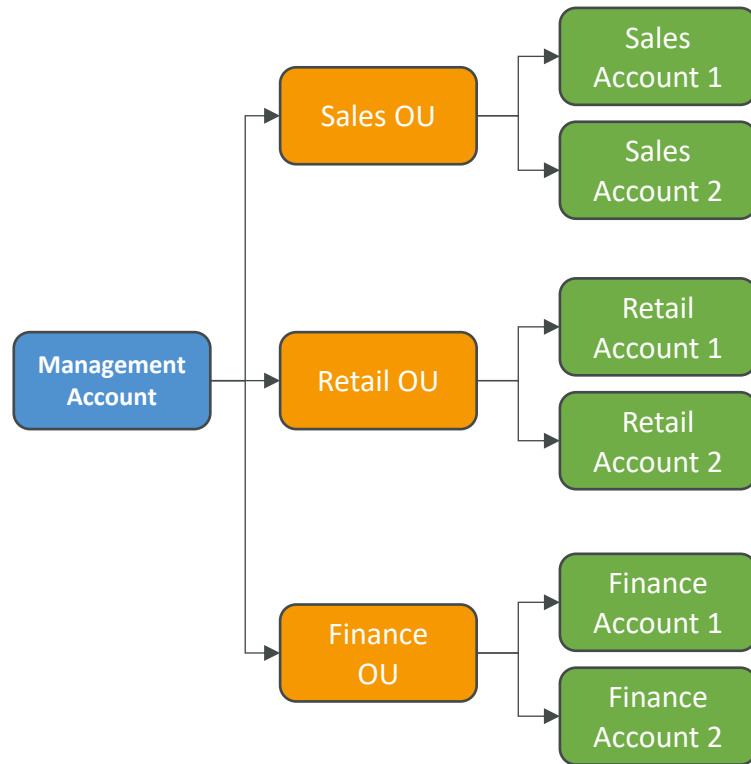
- Global service
- Allows to manage multiple AWS accounts
- The main account is the management account
- Other accounts are member accounts
- Member accounts can only be part of one organization
- Consolidated Billing across all accounts - single payment method
- Pricing benefits from aggregated usage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

# AWS Organizations

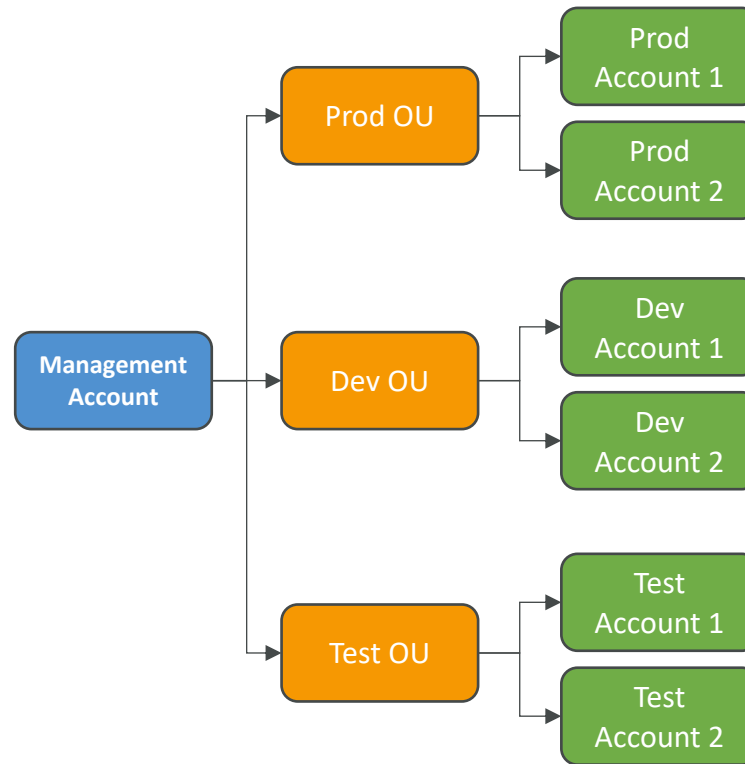


# Organizational Units (OU) - Examples

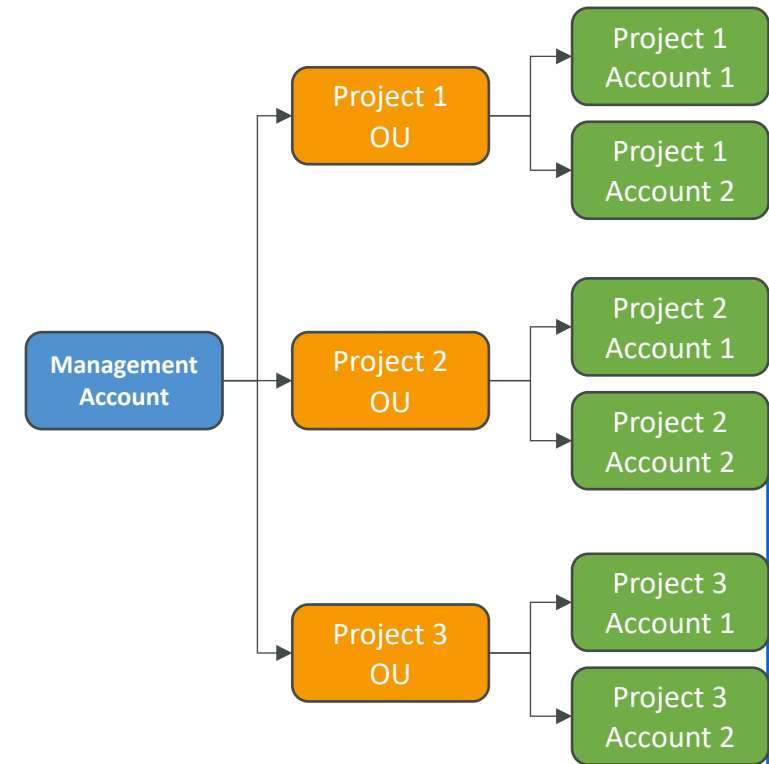
## Business Unit



## Environmental Lifecycle



## Project-Based

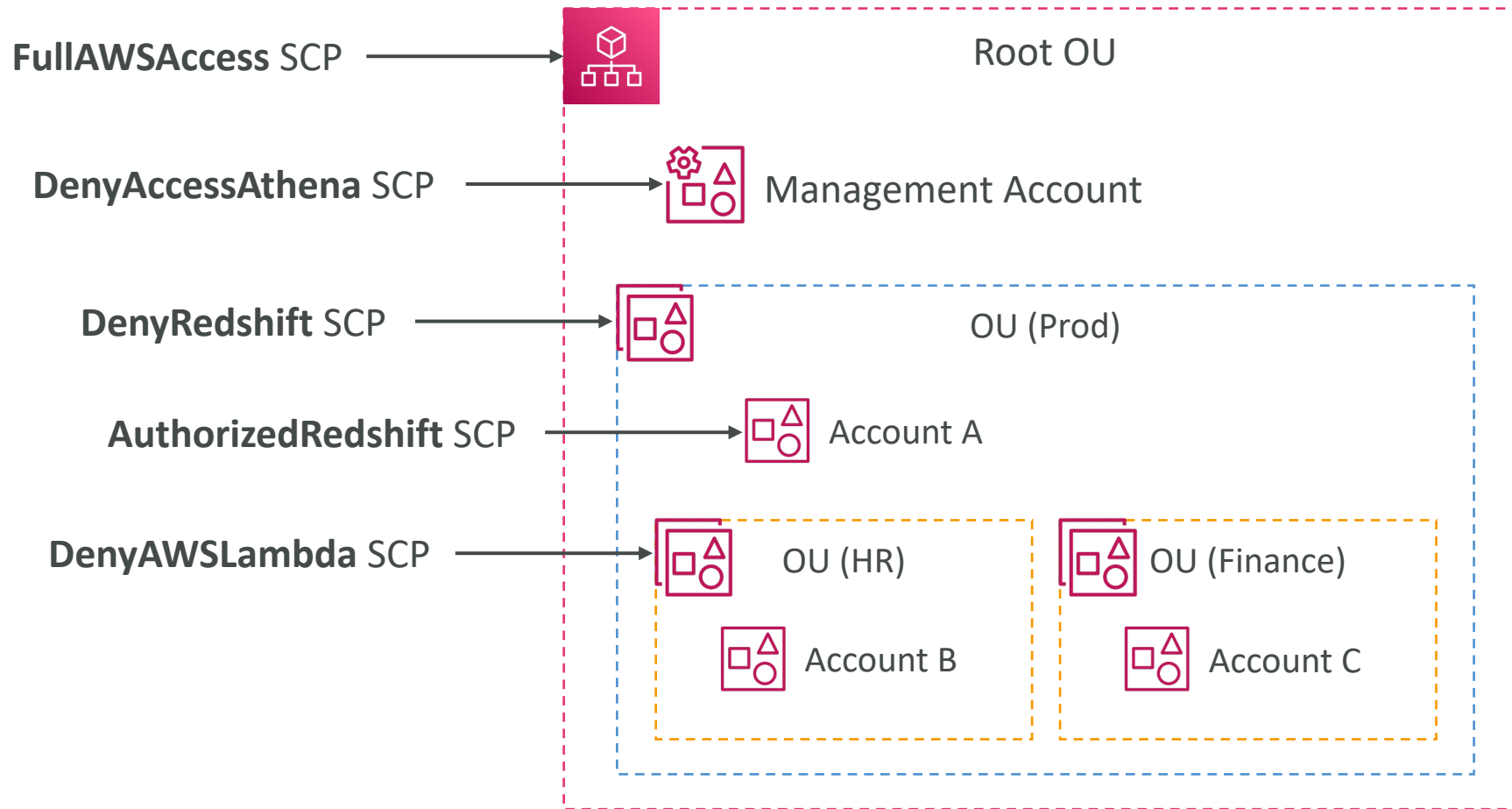


# AWS Organizations

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



# SCP Hierarchy



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

# SCP Examples

## Blocklist and Allowlist strategies

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

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

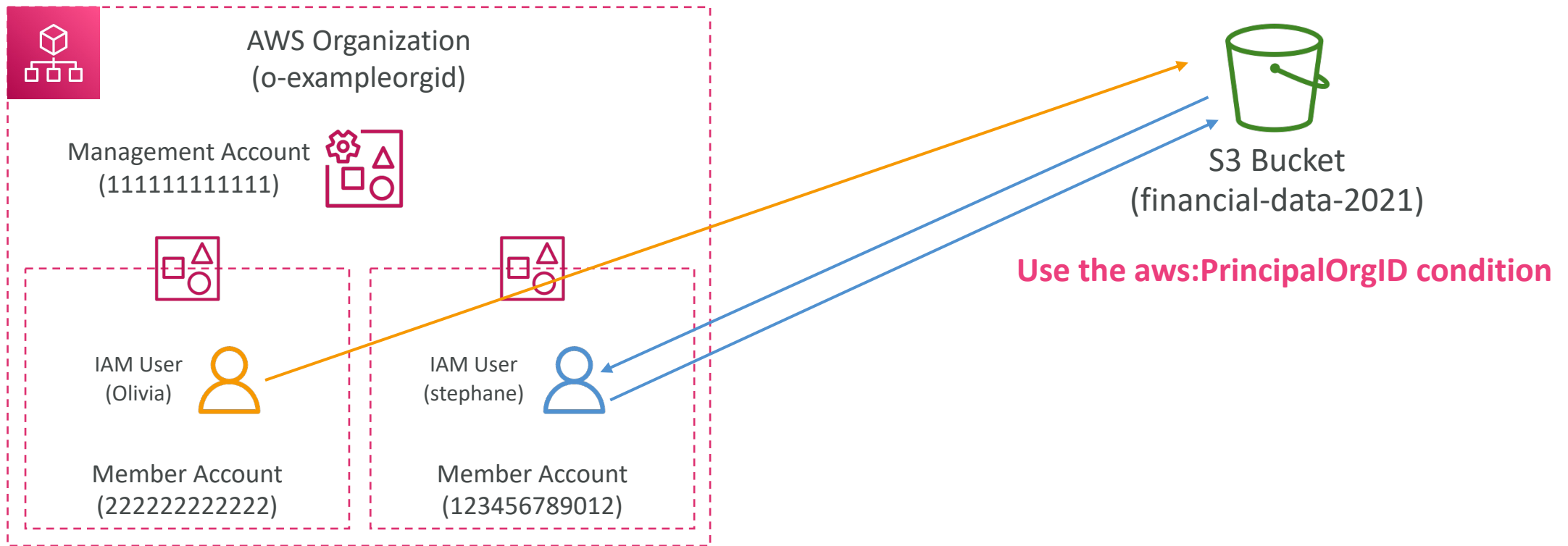
More examples: [https://docs.aws.amazon.com/organizations/latest/userguide/orgs\\_manage\\_policies\\_example-scps.html](https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_example-scps.html)

# AWS Organizations – Reserved Instances

- For billing purposes, the consolidated billing feature of AWS Organizations treats all the accounts in the organization as one account.
- This means that **all accounts** in the organization can receive the hourly cost benefit of Reserved Instances that are purchased **by any other account**.
- **The payer account (master account) of an organization** can turn off Reserved Instance (RI) discount and Savings Plans discount sharing for any accounts in that organization, including the payer account
- This means that RIs and Savings Plans discounts aren't shared between any accounts that have sharing turned off.
- To share an RI or Savings Plans discount with an account, both accounts must have sharing turned on.

# AWS Organizations – IAM Policies

- Use `aws:PrincipalOrgID` condition key in your resource-based policies to restrict access to IAM principals from accounts in an AWS Organization



# AWS Organizations – Tag Policies

- Helps you standardize tags across resources in an AWS Organization
- Ensure consistent tags, audit tagged resources, maintain proper resources categorization, ...
- You define tag keys and their allowed values
- Helps with AWS Cost Allocation Tags and Attribute-based Access Control
- Prevent any non-compliant tagging operations on specified services and resources (has no effect on resources without tags)
- Generate a report that lists all tagged/non-compliant resources
- Use CloudWatch Events to monitor non-compliant tags

```
{
  "tags": {
    "costcenter": {
      "tag_key": {
        "@@assign": "CostCenter"
      },
      "tag_value": {
        "@@assign": ["100", "200"]
      },
      "enforced_for": {
        "@@assign": ["secretsmanager:*"]
      }
    }
  }
}
```

# AWS Control Tower



- Easy way to **set up and govern a secure and compliant multi-account AWS environment** based on best practices
- Benefits:
  - Automate the set up of your environment in a few clicks
  - Automate ongoing policy management using guardrails
  - Detect policy violations and remediate them
  - Monitor compliance through an interactive dashboard
- AWS Control Tower runs on top of AWS Organizations:
  - It automatically sets up AWS Organizations to organize accounts and implement SCPs (Service Control Policies)

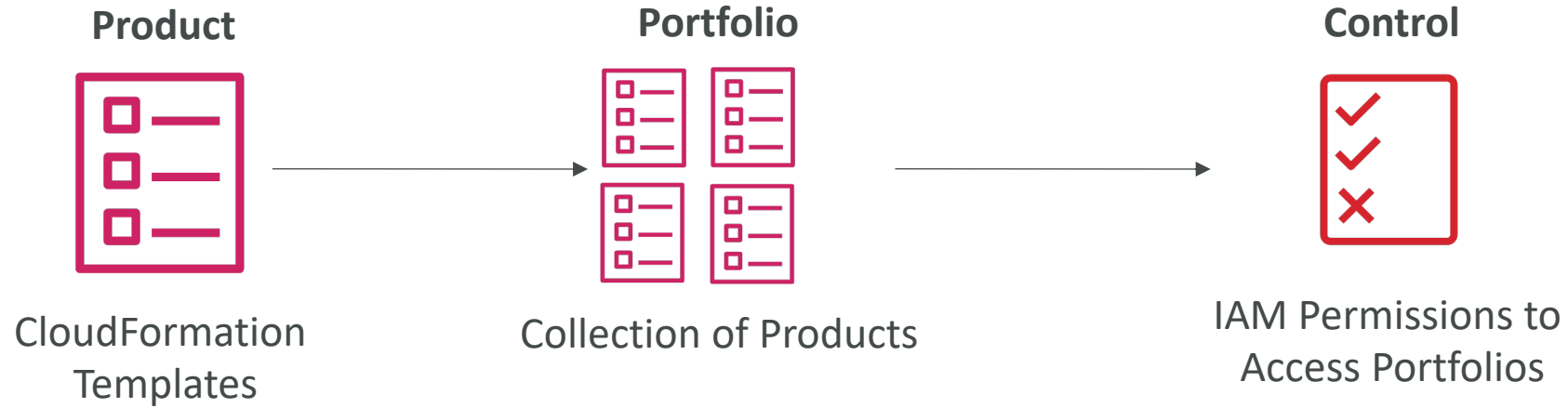
# AWS Service Catalog



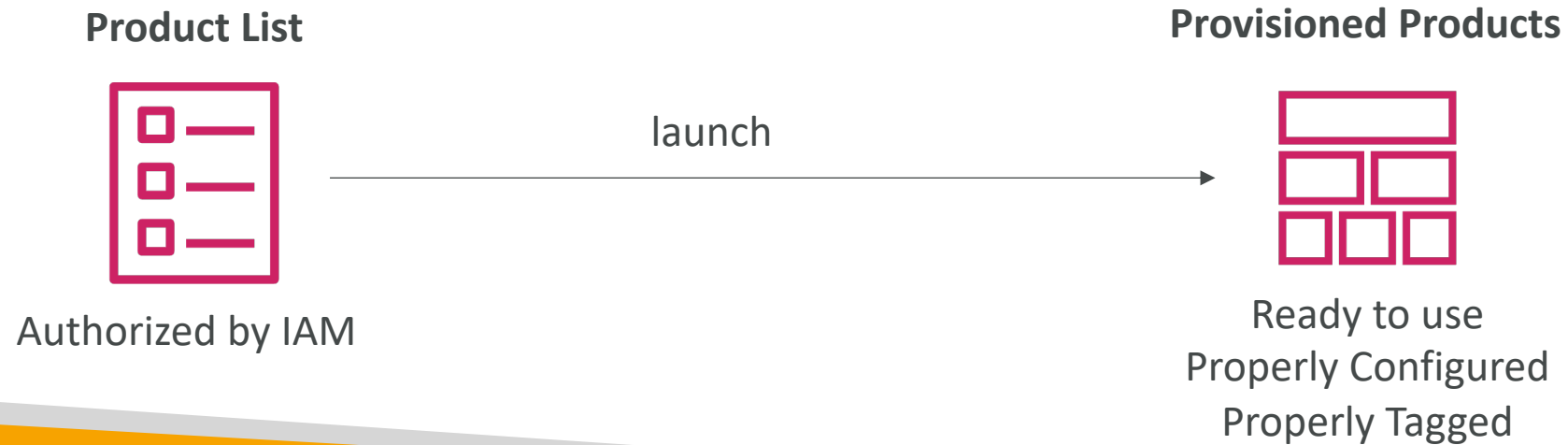
- Users that are new to AWS have too many options, and may create stacks that are not compliant / in line with the rest of the organization
- Some users just want a quick **self-service portal** to launch a set of **authorized products** pre-defined by admins
- Includes: virtual machines, databases, storage options, etc...
- Enter AWS Service Catalog!

# Service Catalog diagram

## ADMIN TASKS



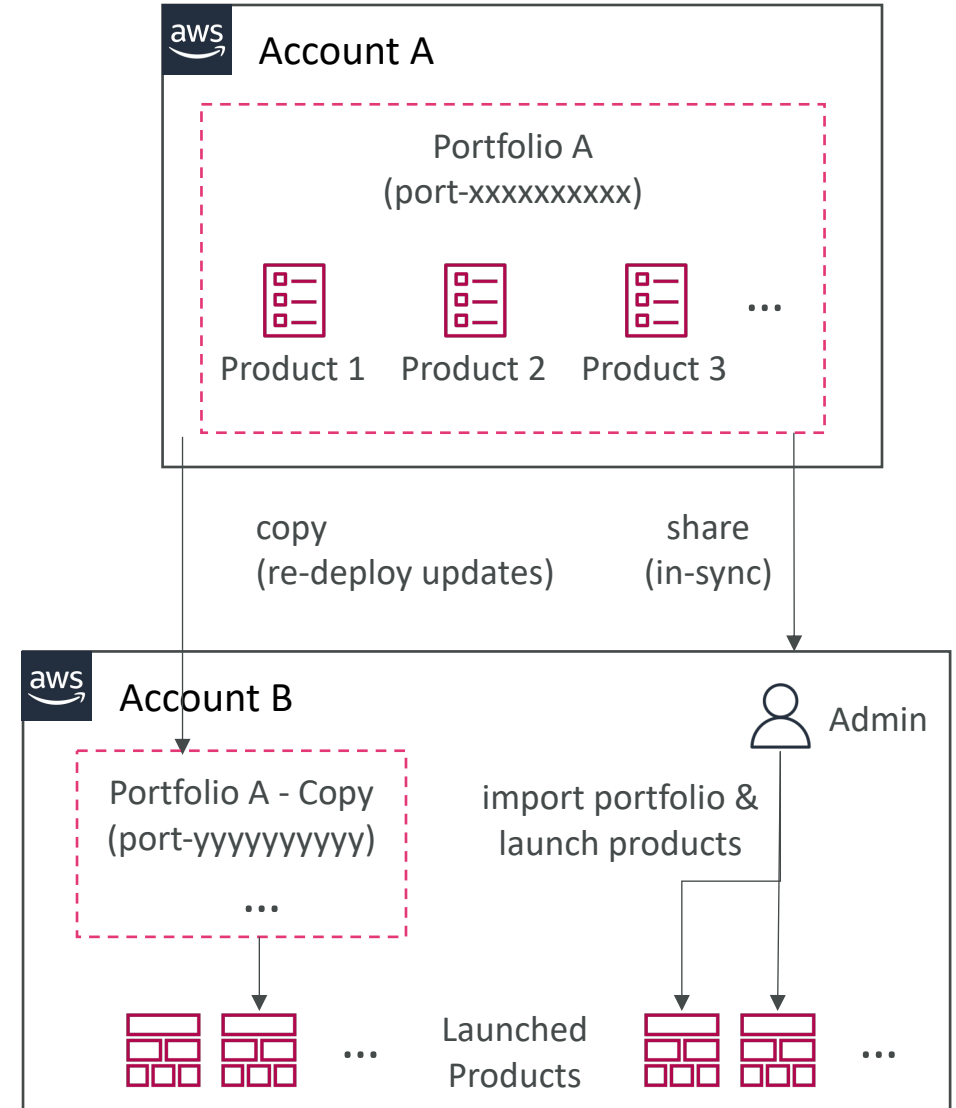
## USER TASKS





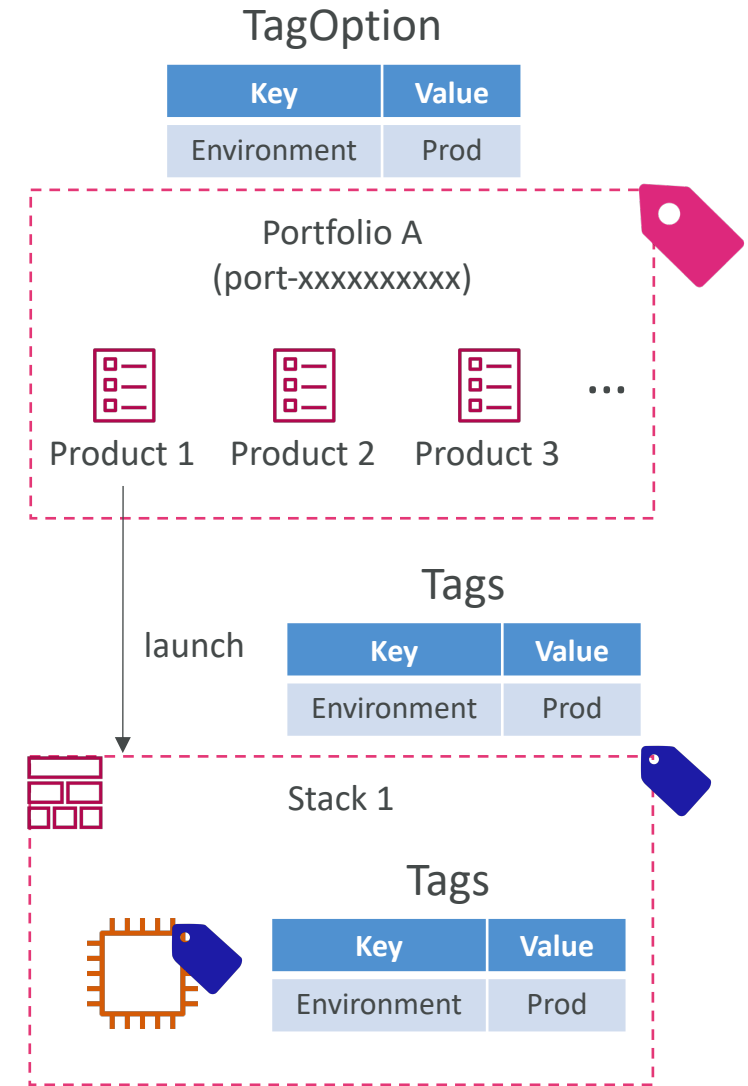
# AWS Service Catalog – Sharing Catalogs

- Share portfolios with individual AWS accounts or AWS Organizations
- Sharing options:
  - Share a reference of the portfolio, then import the shared portfolio in the recipient account (stays in-sync with the original portfolio)
  - Deploy a copy of the portfolio into the recipient account (must re-deploy any updates)
- Ability to add products from the imported portfolio to the local portfolio



# AWS Service Catalog – TagOptions Library

- Easily manage tags on provisioned products
- **TagOption:**
  - Key-value pair managed in AWS Service Catalog
  - Used to create an AWS Tag
- Can be associated with Portfolios and Products
- Use cases: proper resources tagging, defined allowed tags, ...
- Can be shared with other AWS accounts and AWS Organizations



# AWS Billing Alarms



- Billing data metric is stored in CloudWatch us-east-1
- Billing data are for overall worldwide AWS costs
- It's for actual cost, not for project costs
- Let's create a billing alarm together!

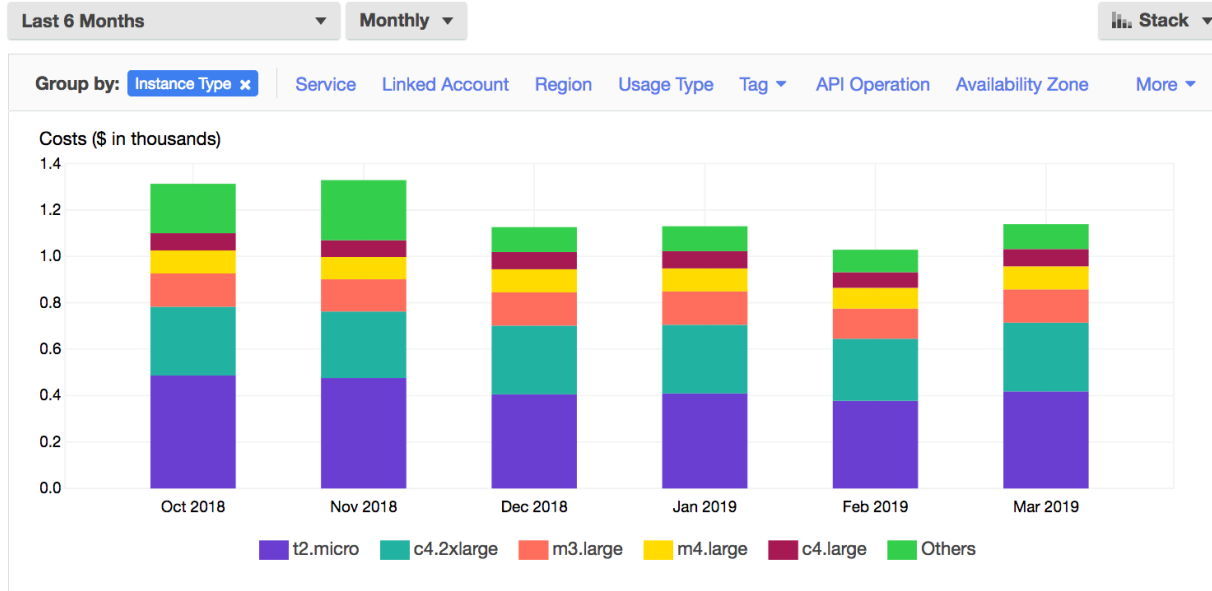
# Cost Explorer



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

# Cost Explorer – Monthly Cost by AWS Service

## Monthly costs by service



Download CSV

Instance Type	Oct 1, 2018	Nov 1, 2018	Dec 1, 2018	Jan 1, 2019
Total cost (\$)	1,312.71	1,328.54	1,125.99	1,129.65
t2.micro (\$)	486.75	475.89	405.63	409.27
c4.2xlarge (\$)	296.11	286.56	296.11	296.11

### FILTERS

CLEAR ALL

- Service Include only
- EC2-Instances x 1
- Linked Account Include all
- Region Include all
- Instance Type Include all
- Usage Type Include all
- Usage Type Group Include all
- Tag Include All
- API Operation Include all
- Charge Type Include all

More filters

### ADVANCED OPTIONS

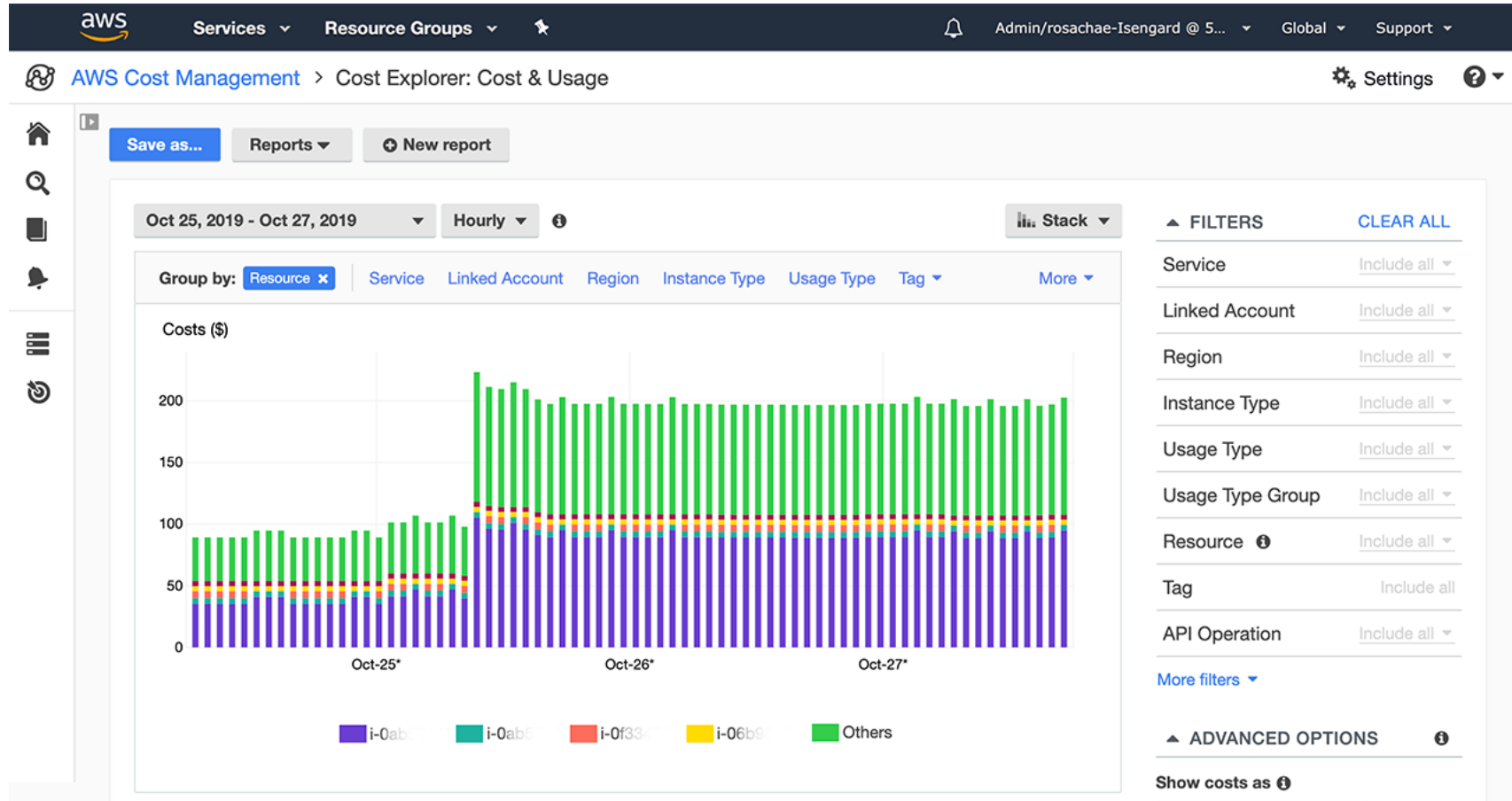
Show costs as

Unblended costs

Include costs related to

Show only untagged resources

# Cost Explorer– Hourly & Resource Level



# Cost Explorer – Savings Plan Alternative to Reserved Instances

Recommendation options

**Savings Plans type**  
 Compute  
 EC2 Instance

**Savings Plans term**  
 1-year  
 3-year

**Payment option**  
 All upfront  
 Partial upfront  
 No upfront

**Based on the past**  
 7 days  
 30 days  
 60 days

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

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

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

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

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

Recommended Compute Savings Plans Download CSV Add selected Savings Plan(s) to cart

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

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

# Cost Explorer – Forecast Usage







# AWS Budgets

- Create budget and **send alarms when costs exceeds the budget**
- 4 types of budgets: Usage, Cost, Reservation, Savings Plans
- For Reserved Instances (RI)
  - Track utilization
  - Supports EC2, ElastiCache, RDS, Redshift
- Up to 5 SNS notifications per budget
- Can filter by: Service, Linked Account, Tag, Purchase Option, Instance Type, Region, Availability Zone, API Operation, etc...
- Same options as AWS Cost Explorer!
- 2 budgets are free, then \$0.02/day/budget

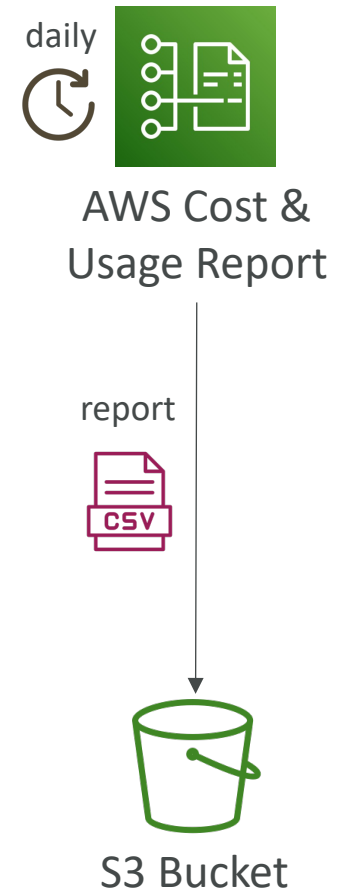
# Cost Allocation Tags

- Use **cost allocation tags** to track your AWS costs on a detailed level
- **AWS generated tags**
  - Automatically applied to the resource you create
  - Starts with Prefix **aws:** (e.g. **aws: createdBy**)
- **User-defined tags**
  - Defined by the user
  - Starts with Prefix **user:**

Total Cost ▾	user:Owner ▾	user:Stack ▾	user:Cost Center ▾	user:Application ▾
0.95	DbAdmin	Test	80432	Widget2
0.01	DbAdmin	Test	80432	Widget2
3.84	DbAdmin	Prod	80432	Widget2
6.00	DbAdmin	Test	78925	Widget1
234.63	SysEng	Prod	78925	Widget1
0.73	DbAdmin	Test	78925	Widget1
0.00	DbAdmin	Prod	80432	Portal
2.47	DbAdmin	Prod	78925	Portal

# Cost and Usage Reports

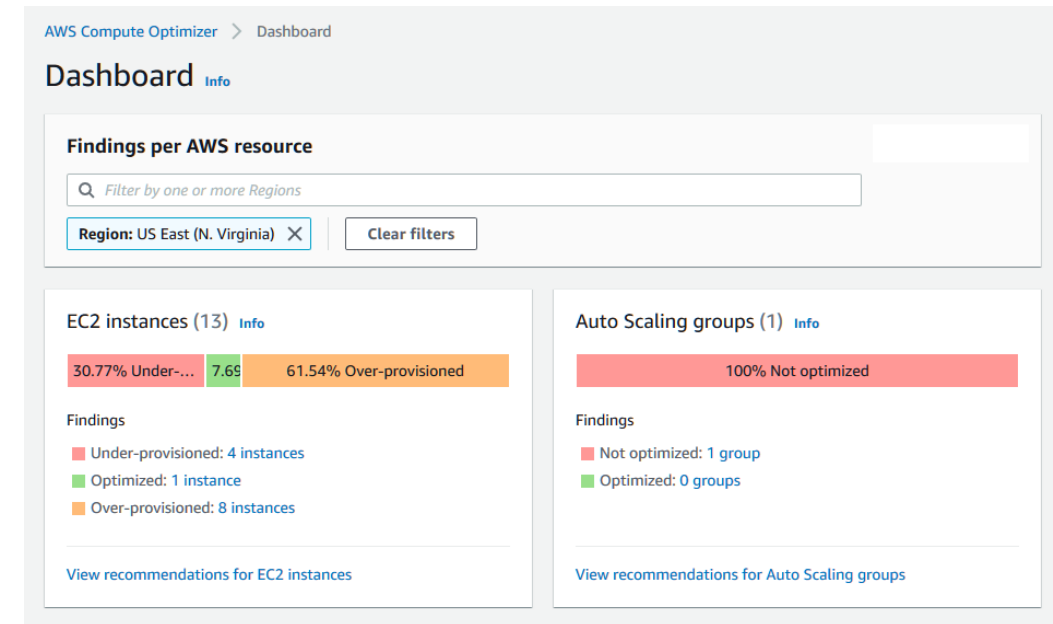
- Dive deeper into your AWS costs and usage
- The AWS Cost & Usage Report contains the most comprehensive set of AWS cost and usage data available
- Includes additional metadata about AWS services, pricing, and reservations (e.g., Amazon EC2 Reserved Instances (RIs))
- The AWS Cost & Usage Report lists AWS usage for each:
  - service category used by an account
  - in hourly or daily line items
  - any tags that you have activated for cost allocation purposes
- Can be configured for daily exports to S3
- Can be integrated with Athena, Redshift or QuickSight



# AWS Compute Optimizer



- Reduce costs and improve performance by recommending optimal AWS resources for your workloads
- Helps you choose optimal configurations and right-size your workloads (over/under provisioned)
- Uses Machine Learning to analyze your resources' configurations and their utilization CloudWatch metrics
- Supported resources
  - EC2 instances
  - EC2 Auto Scaling Groups
  - EBS volumes
  - Lambda functions
- Lower your costs by up to 25%
- Recommendations can be exported to S3



# Cost and Usage Reports

	M	N	O	P	R	S	T
1	lineltem/ProductCode	lineltem/UsageType	lineltem/Operation	lineltem/AvailabilityZone	lineltem/UsageAmount	lineltem/CurrencyCode	lineltem/LineltemDescription
2	AmazonEC2	CW:AlarmMonitorUsage	Unknown		0.00134409	USD	\$0.00 per alarm-month - first 10 alarms
3	AmazonS3	Requests-Tier1	ListAllMyBuckets		2	USD	\$0.00 per request - PUT, COPY, POST, or LIST requests under the monthly global free tier
4	AmazonEC2	CW:AlarmMonitorUsage	Unknown		0.00134409	USD	\$0.00 per alarm-month - first 10 alarms
5	AmazonEC2	APS2-EBS:VolumeUsage_gp2	CreateVolume-Gp2		0.01344086	USD	\$0.00 per GB-month of General Purpose (SSD) provisioned storage under monthly free tier
6	AmazonEC2	APS2-EBS:VolumeUsage_gp2	CreateVolume-Gp2		0.01344086	USD	\$0.00 per GB-month of General Purpose (SSD) provisioned storage under monthly free tier
7	AmazonEC2	USW2-BoxUsage:t2.micro	RunInstances:0002	us-west-2a	1	USD	\$0.00 per Windows t2.micro instance-hour (or partial hour) under monthly free tier
8	AmazonEC2	USW2-USE1-AWS-Out-Bytes	PublicIP-Out		0.00000174	USD	\$0.000 per GB - data transfer out under the monthly global free tier
9	AmazonEC2	USW2-USE1-AWS-In-Bytes	PublicIP-In		0.00000138	USD	\$0.00 per GB - US West (Oregon) data transfer from US East (Northern Virginia)
10	AmazonEC2	USW2-USW1-AWS-In-Bytes	PublicIP-In		0.00000149	USD	\$0.00 per GB - US West (Oregon) data transfer from US West (Northern California)
11	AmazonS3	Requests-Tier1	ListAllMyBuckets		2	USD	\$0.00 per request - PUT, COPY, POST, or LIST requests under the monthly global free tier
12	AmazonEC2	USW2-DataTransfer-Out-Bytes	RunInstances		0.00038144	USD	\$0.000 per GB - data transfer out under the monthly global free tier
13	AmazonEC2	USW2-USW1-AWS-Out-Bytes	PublicIP-Out		0.00000174	USD	\$0.000 per GB - data transfer out under the monthly global free tier
14	AmazonEC2	USW2-DataTransfer-In-Bytes	RunInstances		0.00030951	USD	\$0.000 per GB - data transfer in per month
15	AmazonEC2	USW2-BoxUsage:t2.micro	RunInstances:0002	us-west-2a	1	USD	\$0.00 per Windows t2.micro instance-hour (or partial hour) under monthly free tier
16	AmazonEC2	USW2-USW1-AWS-Out-Bytes	PublicIP-Out		0.00000349	USD	\$0.000 per GB - data transfer out under the monthly global free tier
17	AmazonEC2	USW2-USW1-AWS-In-Bytes	PublicIP-In		0.00000276	USD	\$0.00 per GB - US West (Oregon) data transfer from US West (Northern California)
18	AmazonEC2	APS2-EBS:VolumeUsage_gp2	CreateVolume-Gp2		0.01344086	USD	\$0.00 per GB-month of General Purpose (SSD) provisioned storage under monthly free tier
19	AmazonEC2	CW:AlarmMonitorUsage	Unknown		0.00134409	USD	\$0.00 per alarm-month - first 10 alarms
20	AmazonEC2	USW2-BoxUsage:t2.micro	RunInstances:0002	us-west-2a	1	USD	\$0.00 per Windows t2.micro instance-hour (or partial hour) under monthly free tier
21	AmazonEC2	USW2-DataTransfer-Regional-Bytes	PublicIP-Out		0.00000349	USD	\$0.000 per GB - regional data transfer under the monthly global free tier
22	AmazonEC2	USW2-DataTransfer-In-Bytes	RunInstances		0.00032071	USD	\$0.000 per GB - data transfer in per month
23	AmazonEC2	USW2-DataTransfer-Regional-Bytes	PublicIP-In		0.00000302	USD	\$0.000 per GB - regional data transfer under the monthly global free tier
24	AmazonEC2	USW2-USE1-AWS-Out-Bytes	PublicIP-Out		0.00000174	USD	\$0.000 per GB - data transfer out under the monthly global free tier
25	AmazonEC2	USW2-DataTransfer-Out-Bytes	RunInstances		0.00045736	USD	\$0.000 per GB - data transfer out under the monthly global free tier
26	AmazonEC2	USW2-DataTransfer-In-Bytes	RunInstances		0.00036737	USD	\$0.000 per GB - data transfer in per month
27	AmazonEC2	USW2-APN2-AWS-In-Bytes	PublicIP-In		0.00000005	USD	\$0.00 per GB - US West (Oregon) data transfer from Asia Pacific (Seoul)
28	AmazonEC2	USW2-APN2-AWS-Out-Bytes	PublicIP-Out		0.00000018	USD	\$0.000 per GB - data transfer out under the monthly global free tier
29	AmazonEC2	USW2-USE1-AWS-In-Bytes	PublicIP-In		0.00000153	USD	\$0.00 per GB - US West (Oregon) data transfer from US East (Northern Virginia)
30	AmazonEC2	USW2-DataTransfer-Out-Bytes	RunInstances		0.00039945	USD	\$0.000 per GB - data transfer out under the monthly global free tier
31	AmazonEC2	CW:AlarmMonitorUsage	Unknown		0.00134409	USD	\$0.00 per alarm-month - first 10 alarms

# Disaster Recovery

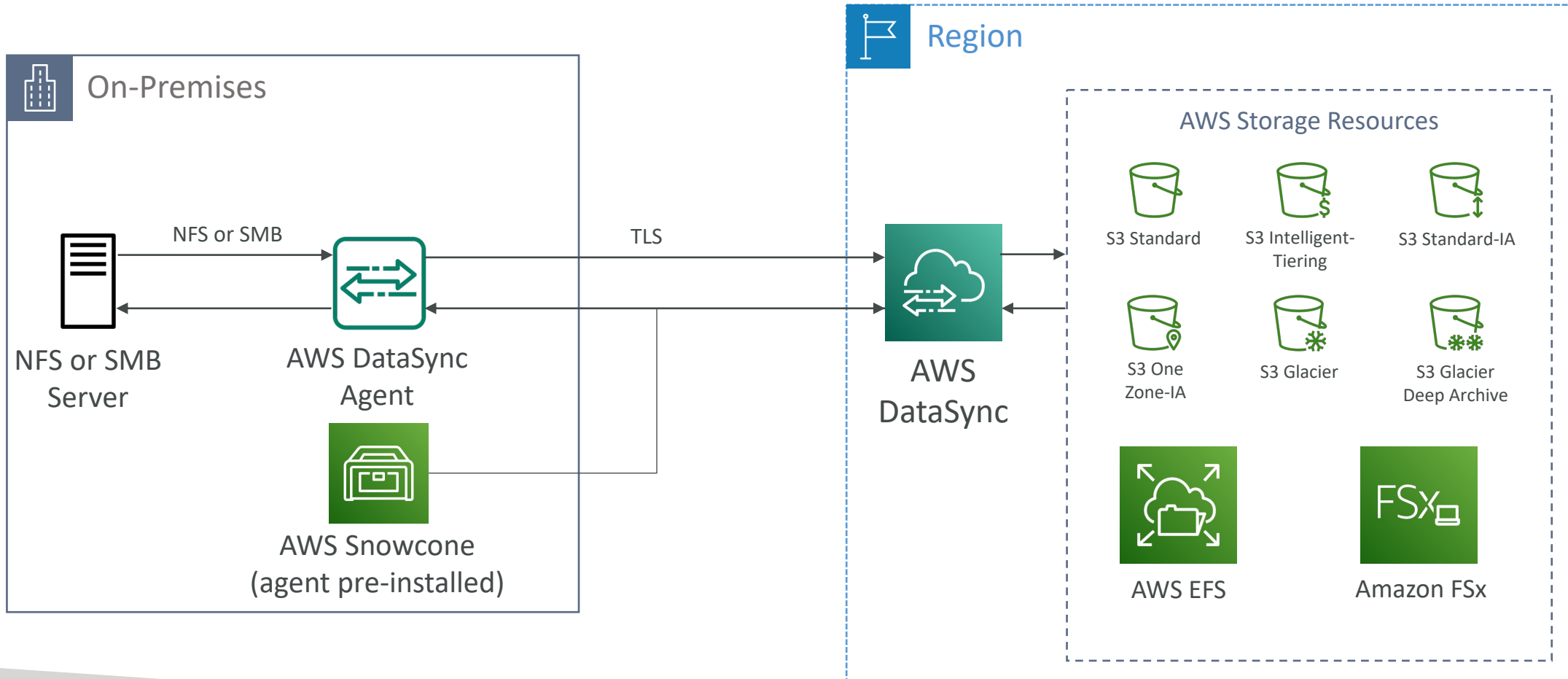


# AWS DataSync

- 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

# AWS DataSync

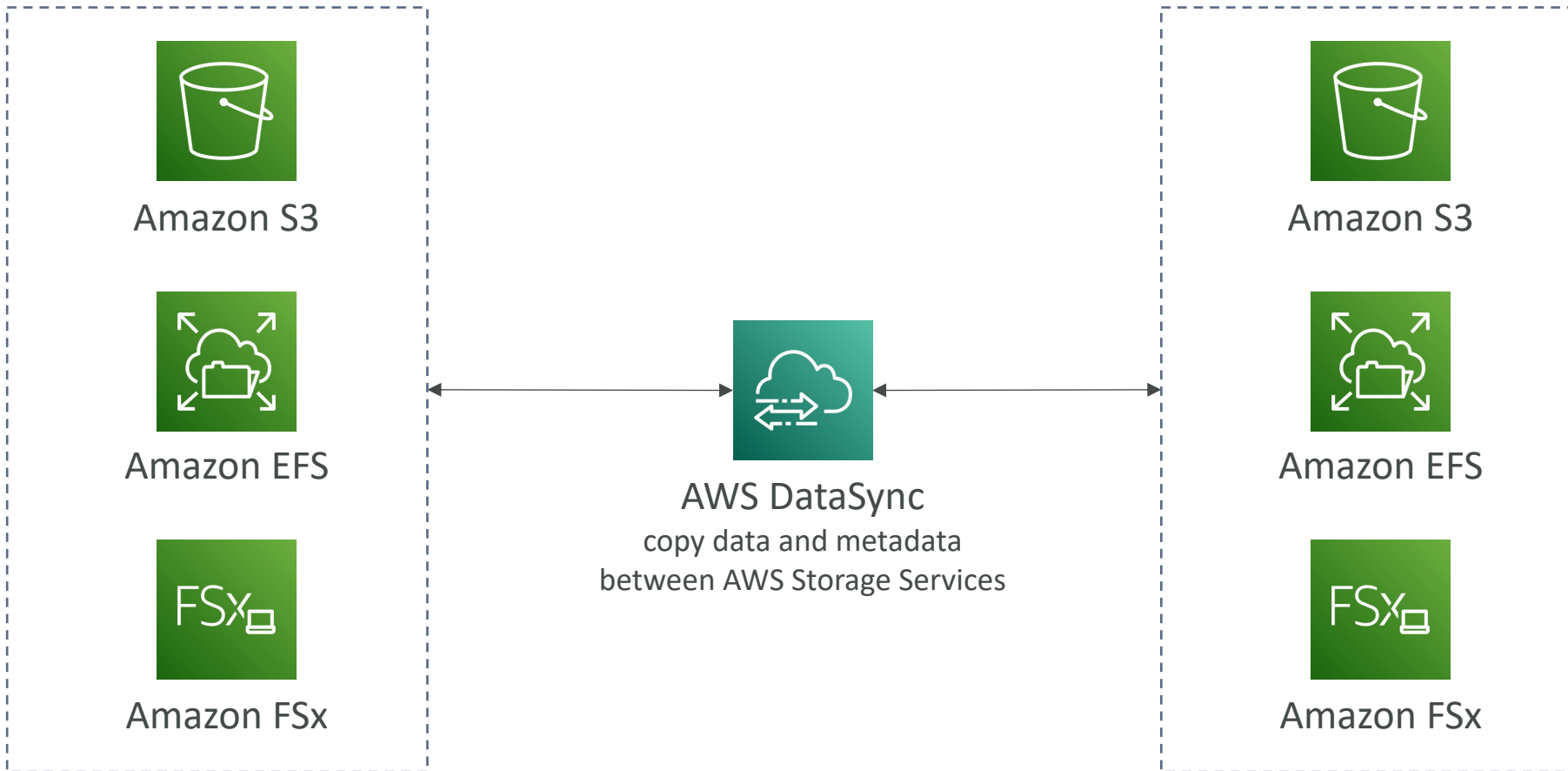
## NFS / SMB to AWS (S3, EFS, FSx...)





# AWS DataSync

## Transfer between AWS storage services



# AWS Backup



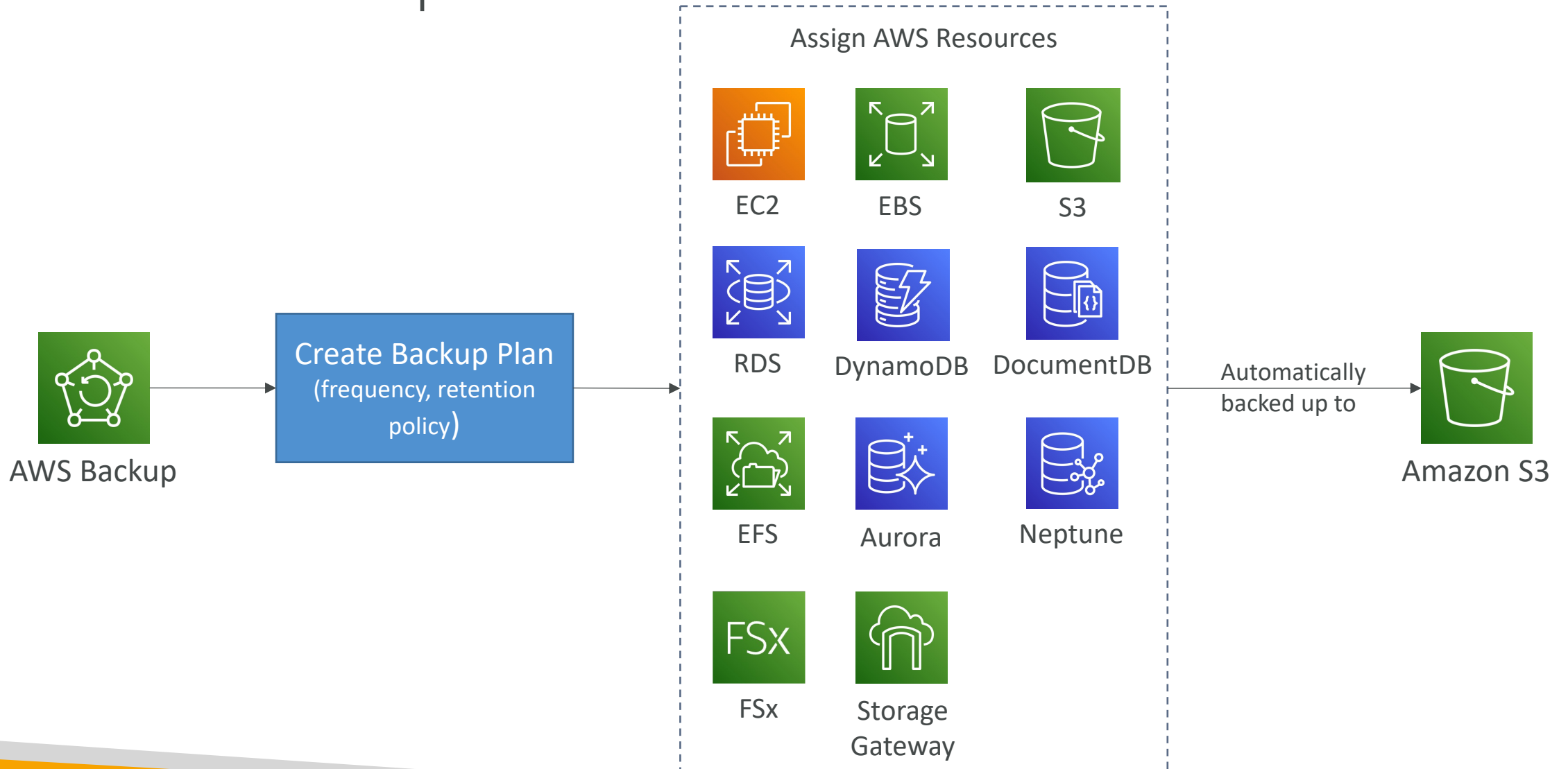
- Fully managed service
- Centrally manage and automate backups across AWS services
- No need to create custom scripts and manual processes
- Supported services:
  - Amazon EC2 / Amazon EBS
  - Amazon S3
  - Amazon RDS (all DBs engines) / Amazon Aurora / Amazon DynamoDB
  - Amazon DocumentDB / Amazon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



# AWS Backup

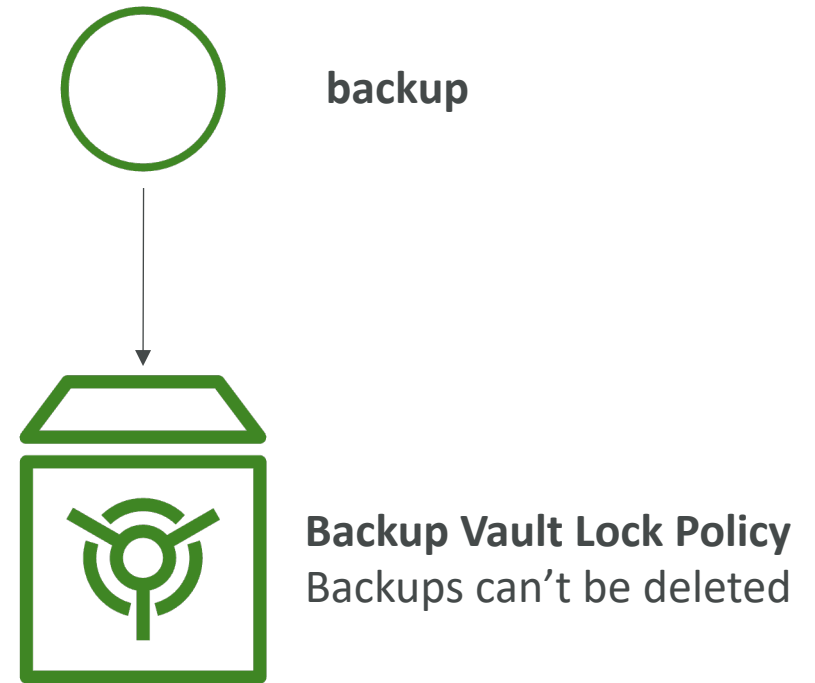
- Supports PITR for supported services
- On-Demand and Scheduled backups
- Tag-based backup policies
- You create backup policies known as **Backup Plans**
  - Backup frequency (every 12 hours, daily, weekly, monthly, cron expression)
  - Backup window
  - Transition to Cold Storage (Never, Days, Weeks,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

# AWS Backup



# AWS Backup Vault Lock

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



# Security and Compliance

# AWS Shared Responsibility Model

- AWS responsibility - Security **of** the Cloud
  - Protecting infrastructure (hardware, software, facilities, and networking) that runs all the AWS services
  - Managed services like S3, DynamoDB, RDS, etc.
- Customer responsibility - Security **in** the Cloud
  - For EC2 instance, customer is responsible for management of the guest OS (including security patches and updates), firewall & network configuration, IAM
  - Encrypting application data
- Shared controls:
  - Patch Management, Configuration Management, Awareness & Training

# Example, for RDS



- AWS responsibility:
  - Manage the underlying EC2 instance, disable SSH access
  - Automated DB patching
  - Automated OS patching
  - Audit the underlying instance and disks & guarantee it functions
- Your responsibility:
  - Check the ports / IP / security group inbound rules in DB's SG
  - In-database user creation and permissions
  - Creating a database with or without public access
  - Ensure parameter groups or DB is configured to only allow SSL connections
  - Database encryption setting

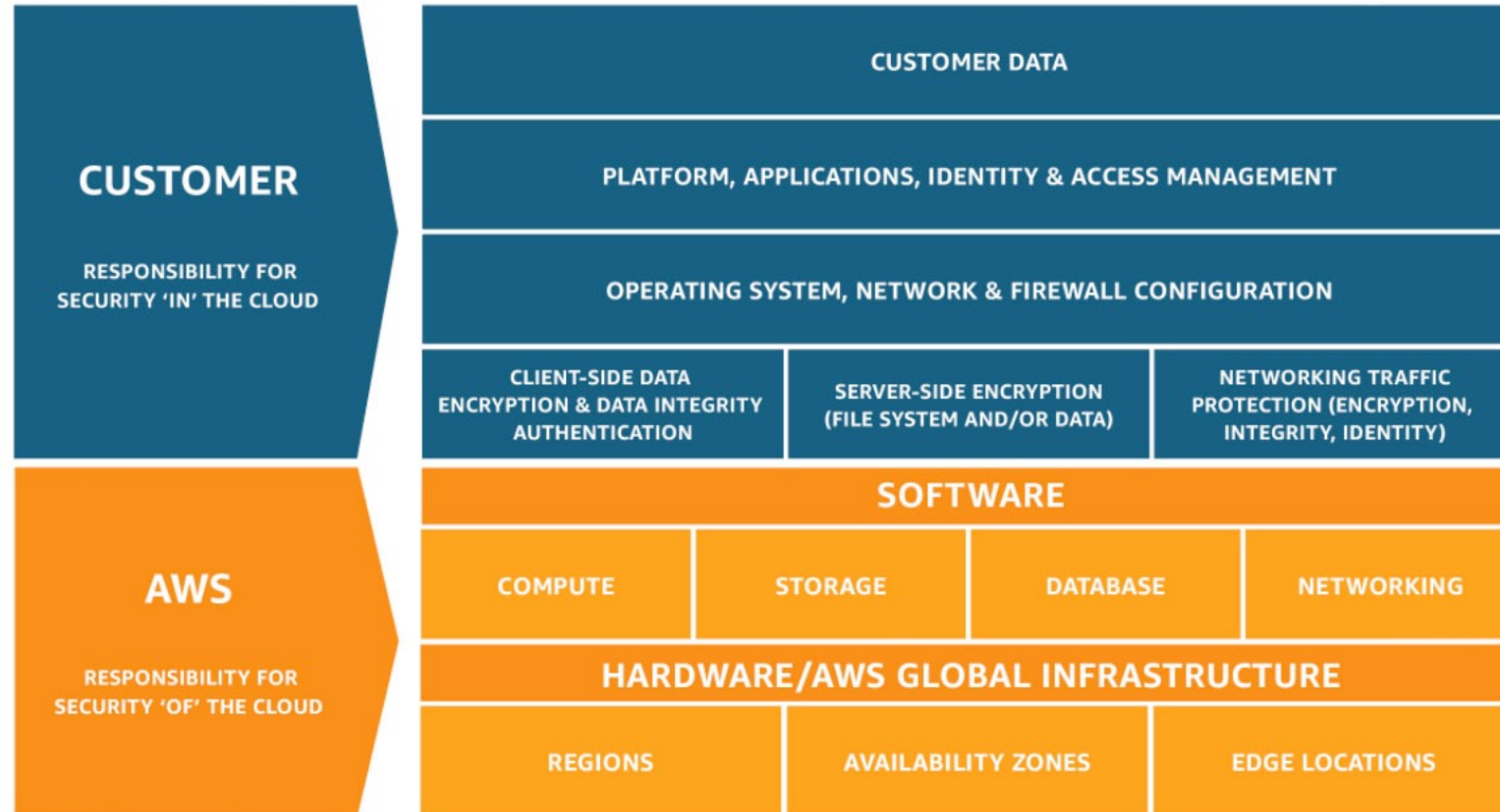


# Example, for S3



- AWS responsibility:
  - Guarantee you get unlimited storage
  - Guarantee you get encryption
  - Ensure separation of the data between different customers
  - Ensure AWS employees can't access your data
- Your responsibility:
  - Bucket configuration
  - Bucket policy / public setting
  - IAM user and roles
  - Enabling encryption

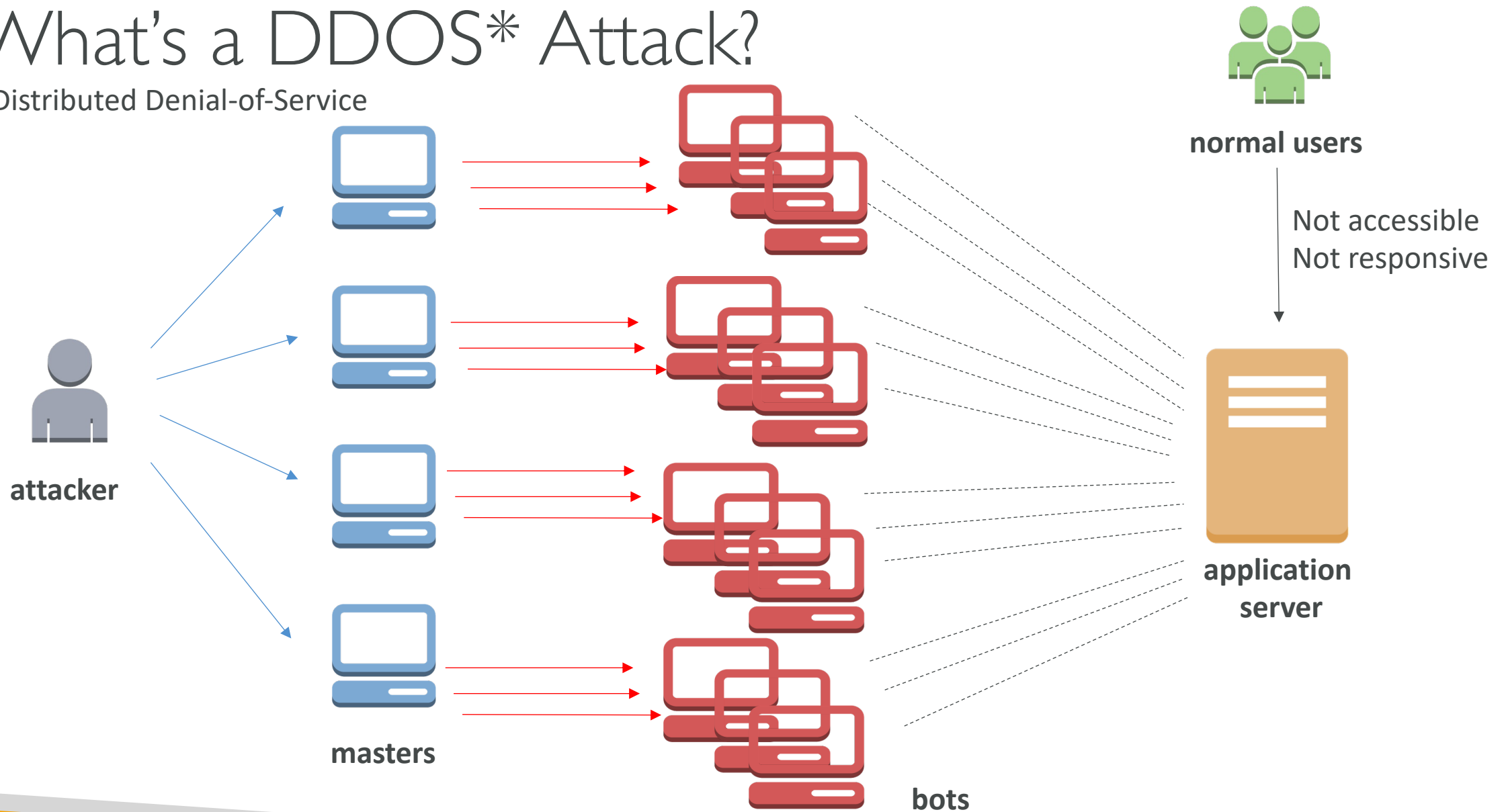
# Shared Responsibility Model diagram



<https://aws.amazon.com/compliance/shared-responsibility-model/>

# What's a DDOS\* Attack?

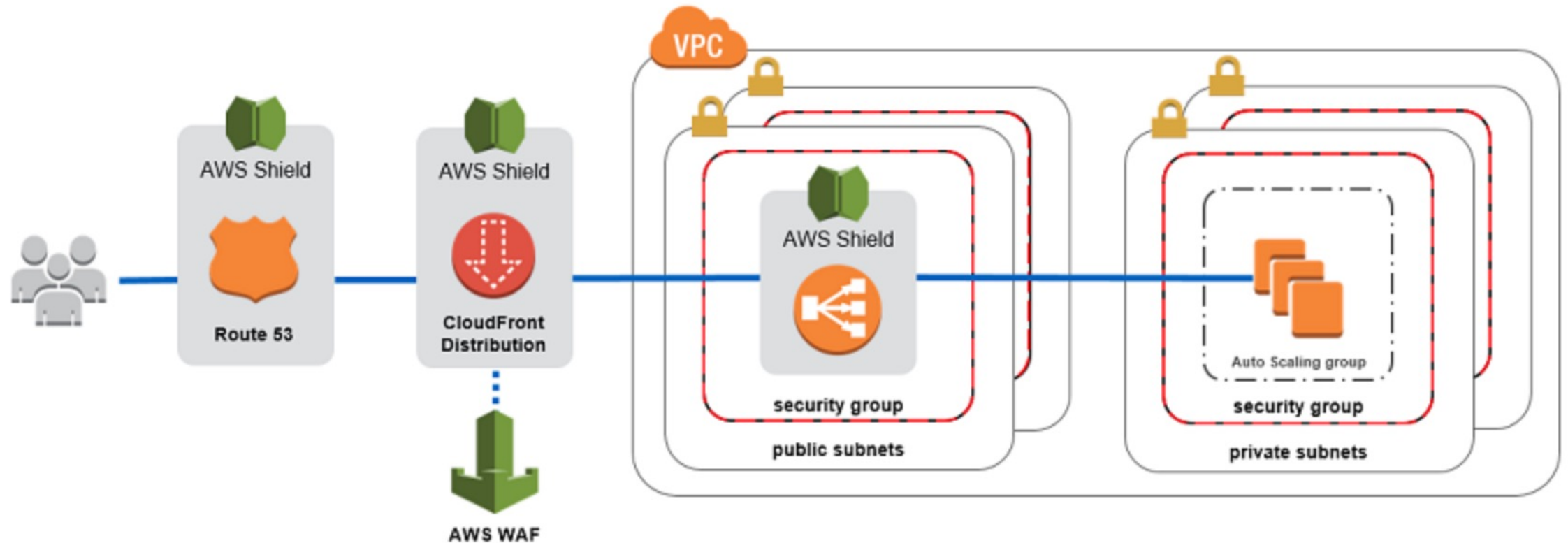
\*Distributed Denial-of-Service



# DDoS Protection on AWS

- **AWS Shield Standard:** protects against DDoS attack for your website and applications, for all customers at no additional costs
- **AWS Shield Advanced:** 24/7 premium DDoS protection
- **AWS WAF:** Filter specific requests based on rules
- **CloudFront and Route 53:**
  - Availability protection using global edge network
  - Combined with AWS Shield, provides attack mitigation at the edge
- Be ready to scale – leverage AWS Auto Scaling

# Sample Reference Architecture



<https://aws.amazon.com/answers/networking/aws-ddos-attack-mitigation/>

# AWS Shield



- **AWS Shield Standard:**
  - Free service that is activated for every AWS customer
  - Provides protection from attacks such as SYN/UDP Floods, Reflection attacks and other layer 3/layer 4 attacks
- **AWS Shield Advanced:**
  - Optional DDoS mitigation service (\$3,000 per month per organization)
  - Protect against more sophisticated attack on [Amazon EC2](#), [Elastic Load Balancing \(ELB\)](#), [Amazon CloudFront](#), [AWS Global Accelerator](#), and [Route 53](#)
  - 24/7 access to AWS DDoS response team (DRP)
  - Protect against higher fees during usage spikes due to DDoS

# AWS WAF – Web Application Firewall



- Protects your web applications from common web exploits (Layer 7)
- Layer 7 is HTTP (vs Layer 4 is TCP)
- Deploy on Application Load Balancer, API Gateway, CloudFront
- Define Web ACL (Web Access Control List):
  - Rules can include: IP addresses, HTTP headers, HTTP body, or URI strings
  - Protects from common attack - SQL injection and Cross-Site Scripting (XSS)
  - Size constraints, geo-match (block countries)
  - Rate-based rules (to count occurrences of events) – for DDoS protection

# Penetration Testing on AWS Cloud



- AWS customers are welcome to carry out security assessments or penetration tests against their AWS infrastructure **without prior approval for 8 services**:
  - Amazon EC2 instances, NAT Gateways, and Elastic Load Balancers
  - Amazon RDS
  - Amazon CloudFront
  - Amazon Aurora
  - Amazon API Gateways
  - AWS Lambda and Lambda Edge functions
  - Amazon Lightsail resources
  - Amazon Elastic Beanstalk environments
- List can increase over time (you won't be tested on that at the exam)



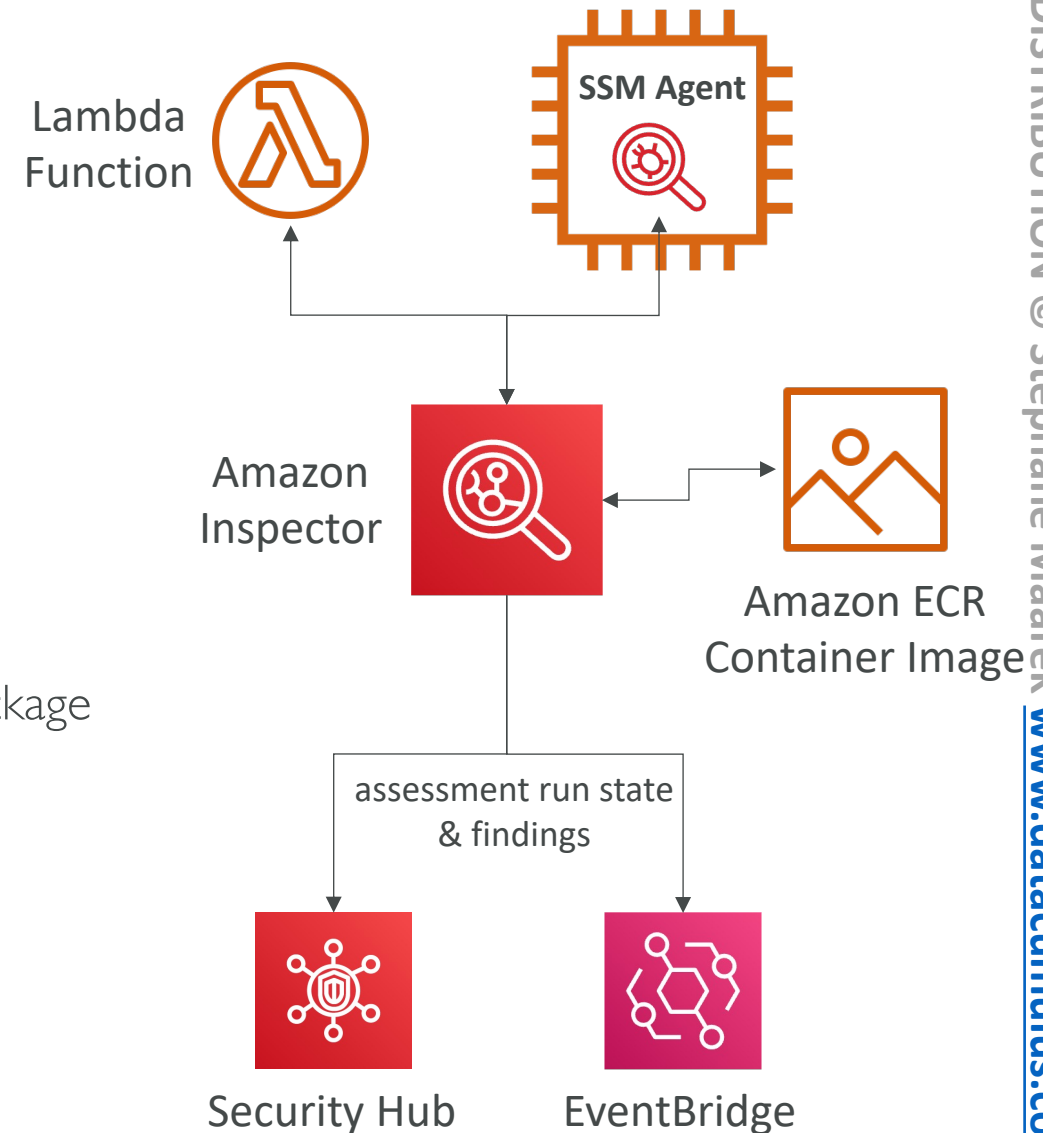
# Penetration Testing on your AWS Cloud



- Prohibited Activities
  - DNS zone walking via Amazon Route 53 Hosted Zones
  - Denial of Service (DoS), Distributed Denial of Service (DDoS), Simulated DoS, Simulated DDoS
  - Port flooding
  - Protocol flooding
  - Request flooding (login request flooding, API request flooding)
- For any other simulated events, contact [aws-security-simulated-event@amazon.com](mailto:aws-security-simulated-event@amazon.com)
- Read more: <https://aws.amazon.com/security/penetration-testing/>

# Amazon Inspector

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



# What does Amazon Inspector evaluate?



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

# Logging in AWS for security and compliance

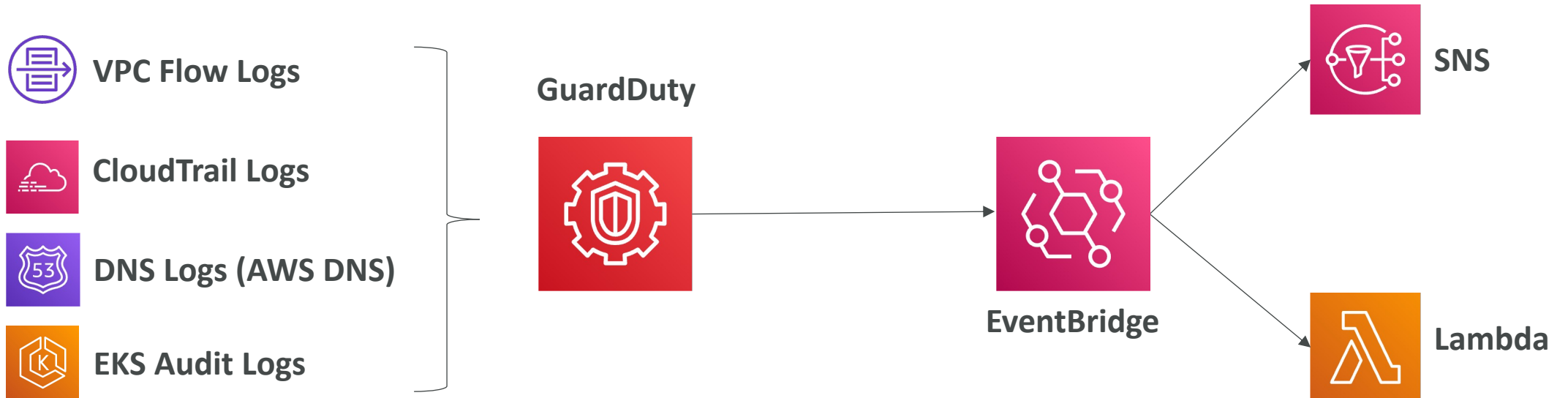
- To help compliance requirements, AWS provides many service-specific security and audit logs
- Service Logs include:
  - CloudTrail trails - trace all API calls
  - Config Rules - for config & compliance over time
  - CloudWatch Logs - for full data retention
  - VPC Flow Logs - IP traffic within your VPC
  - ELB Access Logs - metadata of requests made to your load balancers
  - CloudFront Logs - web distribution access logs
  - WAF Logs - full logging of all requests analyzed by the service
- Logs can be analyzed using AWS Athena if they're stored in S3
- You should encrypt logs in S3, control access using IAM & Bucket Policies, MFA
- Move Logs to Glacier for cost savings
- Read whitepaper if interested at:  
[https://d0.awsstatic.com/whitepapers/compliance/AWS\\_Security\\_at\\_Scale\\_Logging\\_in\\_AWS\\_Whitepaper.pdf](https://d0.awsstatic.com/whitepapers/compliance/AWS_Security_at_Scale_Logging_in_AWS_Whitepaper.pdf)



# Amazon GuardDuty

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

# Amazon GuardDuty



# AWS Macie



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





# Trusted Advisor

- No need to install anything – high level AWS account assessment
- Analyze your AWS accounts and provides recommendation:

## Cost Optimization



## Performance



## Security



## Fault Tolerance



## Service Limits



- Core Checks and recommendations – all customers
- Can enable weekly email notification from the console
- Full Trusted Advisor – Available for **Business & Enterprise** support plans
  - Ability to set CloudWatch alarms when reaching limits
  - Programmatic Access using AWS Support API





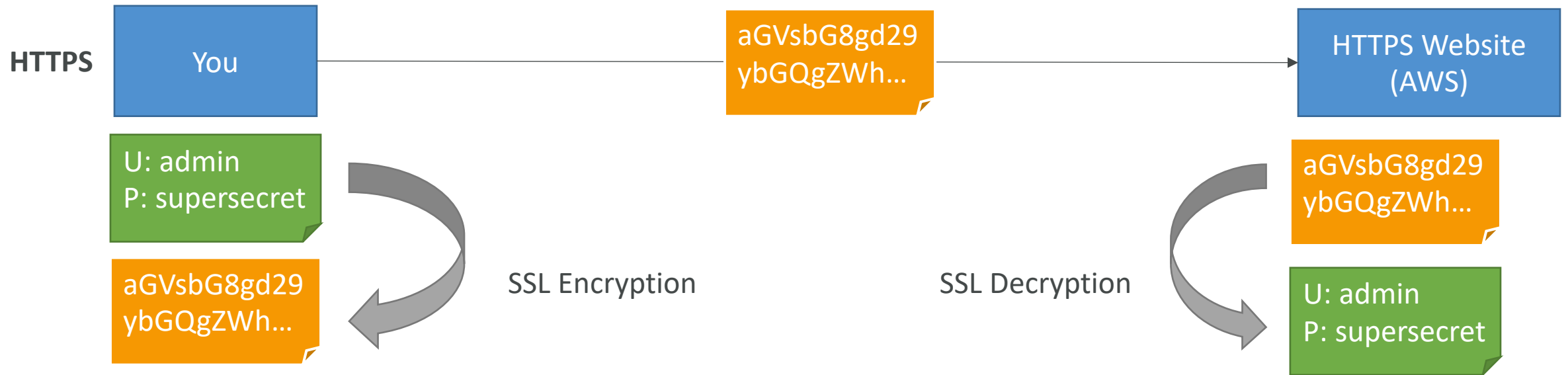
# Trusted Advisor Checks Examples

- Cost Optimization:
  - low utilization EC2 instances, idle load balancers, under-utilized EBS volumes...
  - Reserved instances & savings plans optimizations,
- Performance:
  - High utilization EC2 instances, CloudFront CDN optimizations
  - EC2 to EBS throughput optimizations, Alias records recommendations
- Security:
  - MFA enabled on Root Account, IAM key rotation, exposed Access Keys
  - S3 Bucket Permissions for public access, security groups with unrestricted ports
- Fault Tolerance:
  - EBS snapshots age, Availability Zone Balance
  - ASG Multi-AZ, RDS Multi-AZ, ELB configuration...
- Service Limits

# Why encryption?

## Encryption in flight (SSL)

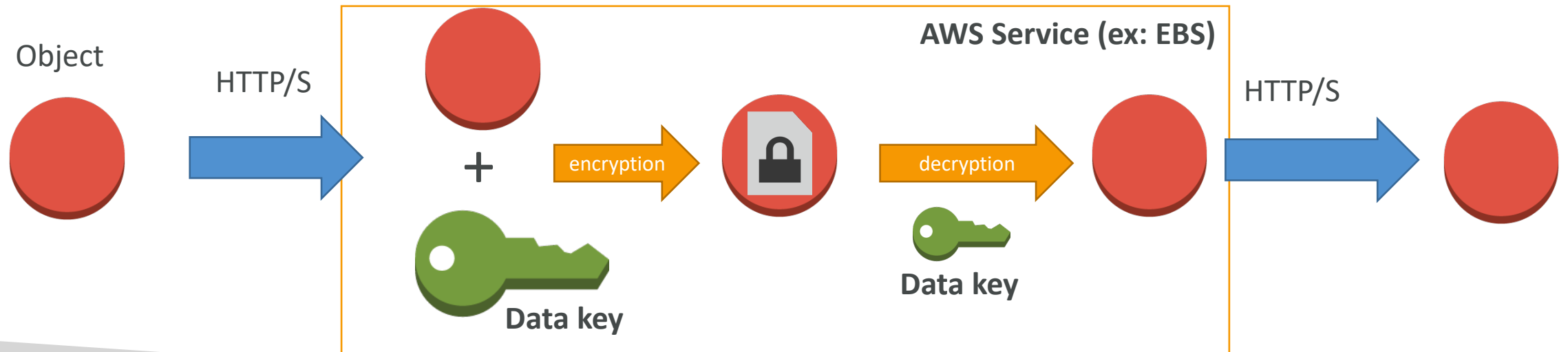
- Data is encrypted before sending and decrypted after receiving
- SSL certificates help with encryption (HTTPS)
- Encryption in flight ensures no MITM (man in the middle attack) can happen



# Why encryption?

## Server side encryption at rest

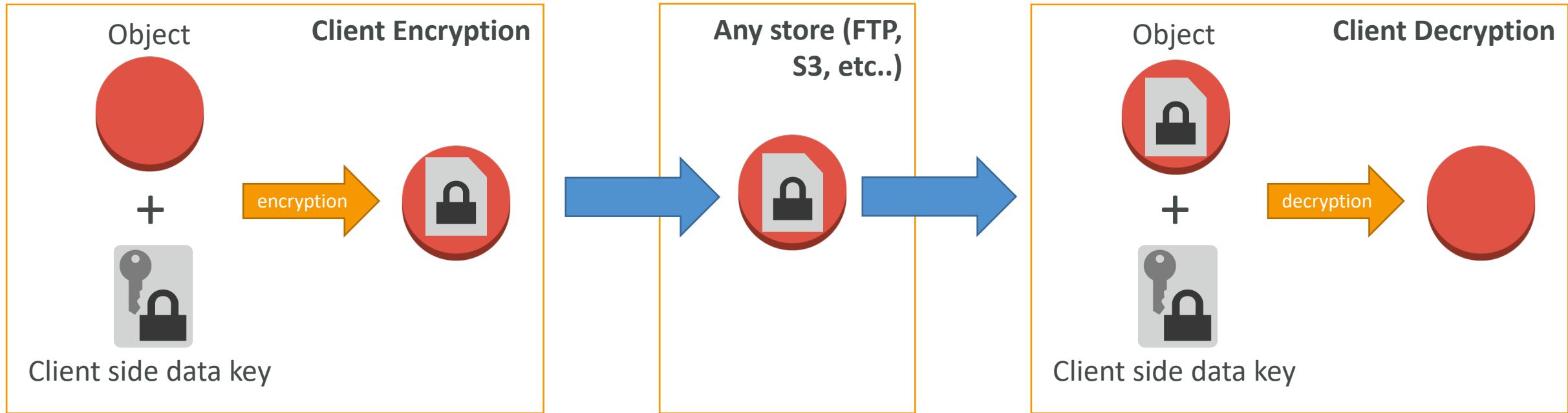
- Data is encrypted after being received by the server
- Data is decrypted before being sent
- It is stored in an encrypted form thanks to a key (usually a data key)
- The encryption / decryption keys must be managed somewhere and the server must have access to it



# Why encryption?

## Client side encryption

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





# AWS KMS (Key Management Service)

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

# KMS Keys Types

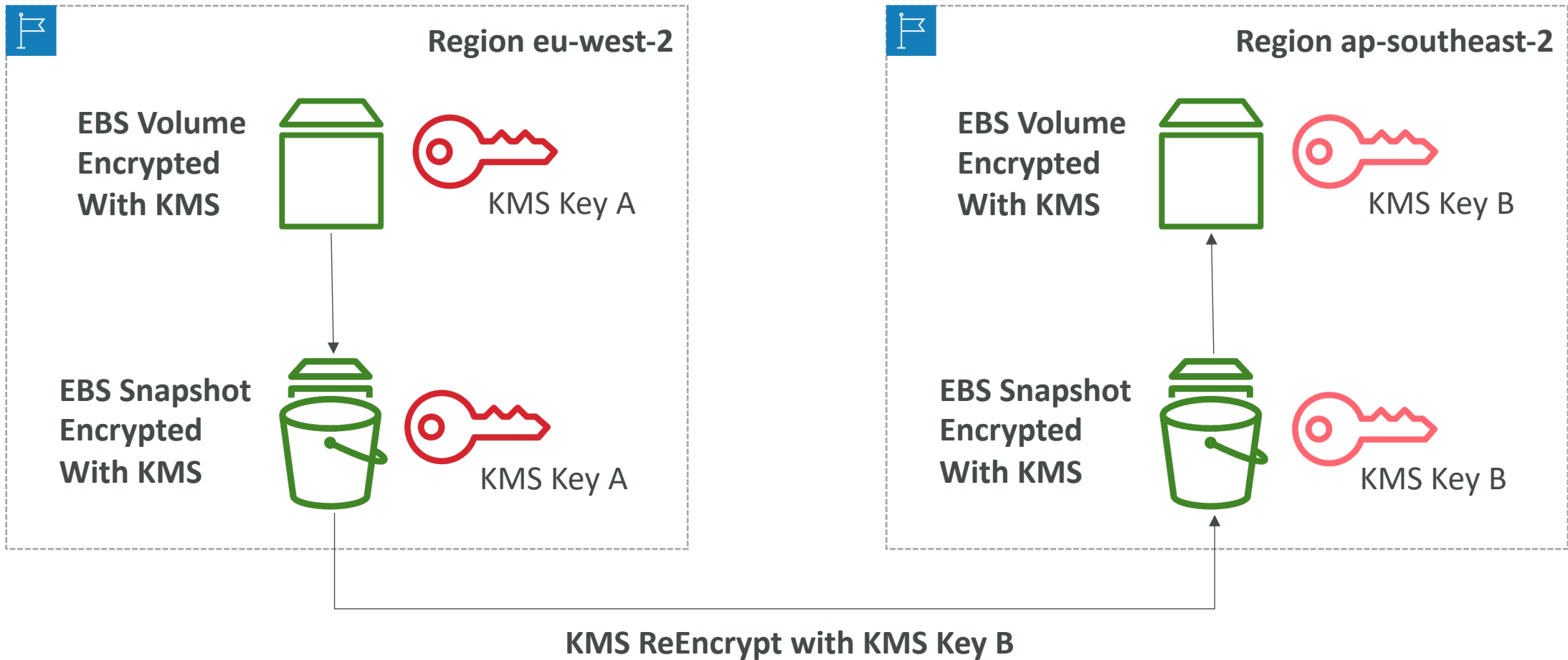
- KMS Keys is the new name of KMS Customer *Master* Key
- Symmetric (AES-256 keys)
  - Single encryption key that is used to Encrypt and Decrypt
  - AWS services that are integrated with KMS use Symmetric CMKs
  - You never get access to the KMS Key unencrypted (must call KMS API to use)
- Asymmetric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

# AWS KMS (Key Management Service)



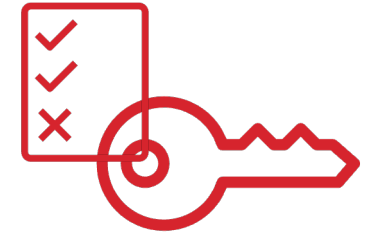
- Three types of KMS Keys:
  - AWS Managed Key: **free** (aws/service-name, example: aws/rds or aws/ebs)
  - Customer Managed Keys (CMK) created in KMS: **\$1 / month**
  - Customer Managed Keys imported (must be 256-bit symmetric key): **\$1 / month**
- + pay for API call to KMS (\$0.03 / 10000 calls)
- Automatic Key rotation:
  - AWS-managed KMS Key: automatic every 1 year
  - Customer-managed KMS Key: (must be enabled) automatic every 1 year
  - Imported KMS Key: only manual rotation possible using alias

# Copying Snapshots across regions





# KMS Key Policies



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

# Copying Snapshots across accounts

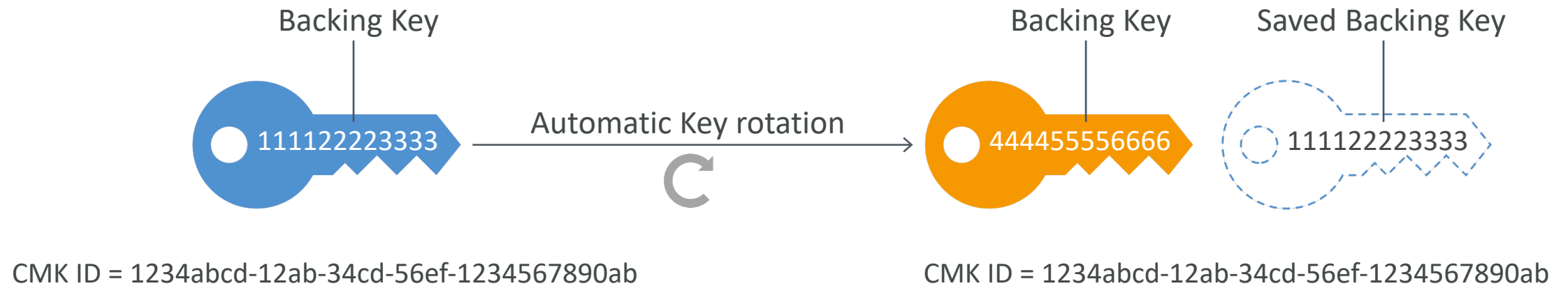
1. Create a Snapshot, encrypted with your own KMS Key (Customer Managed Key)
2. Attach a KMS Key Policy to authorize cross-account access
3. Share the encrypted snapshot
4. (in target) Create a copy of the Snapshot, encrypt it with a CMK in your account
5. Create a volume from the snapshot

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

KMS Key Policy

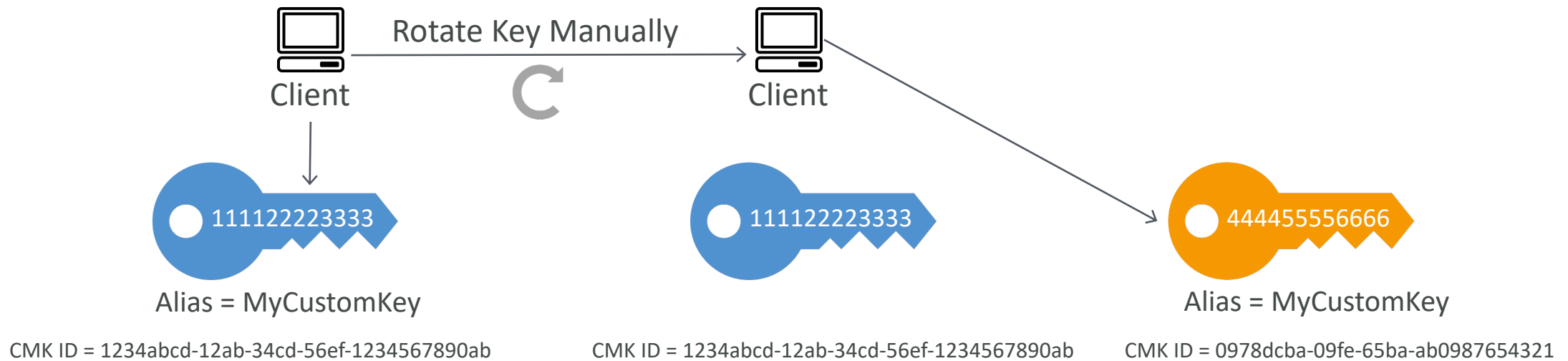
# KMS Automatic Key Rotation

- For Customer-managed CMK (not AWS managed CMK)
- If enabled: automatic key rotation happens every 1 year
- Previous key is kept active so you can decrypt old data
- New Key has the same CMK ID (only the backing key is changed)



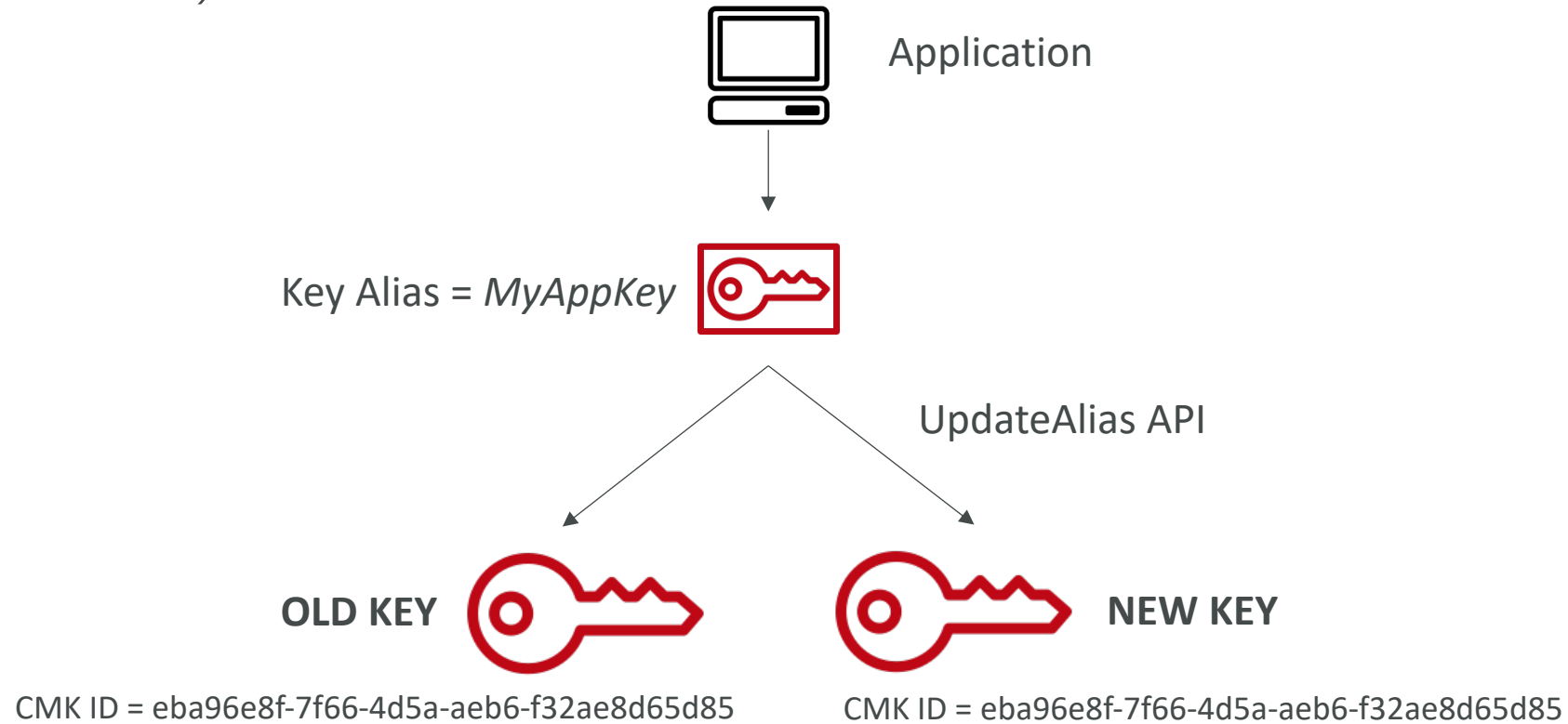
# KMS Manual Key Rotation

- When you want to rotate key every 90 days, 180 days, etc...
- New Key has a different CMK ID
- Keep the previous key active so you can decrypt old data
- Better to use aliases in this case (to hide the change of key for the application)
- Good solution to rotate CMK that are not eligible for automatic rotation (like asymmetric CMK)



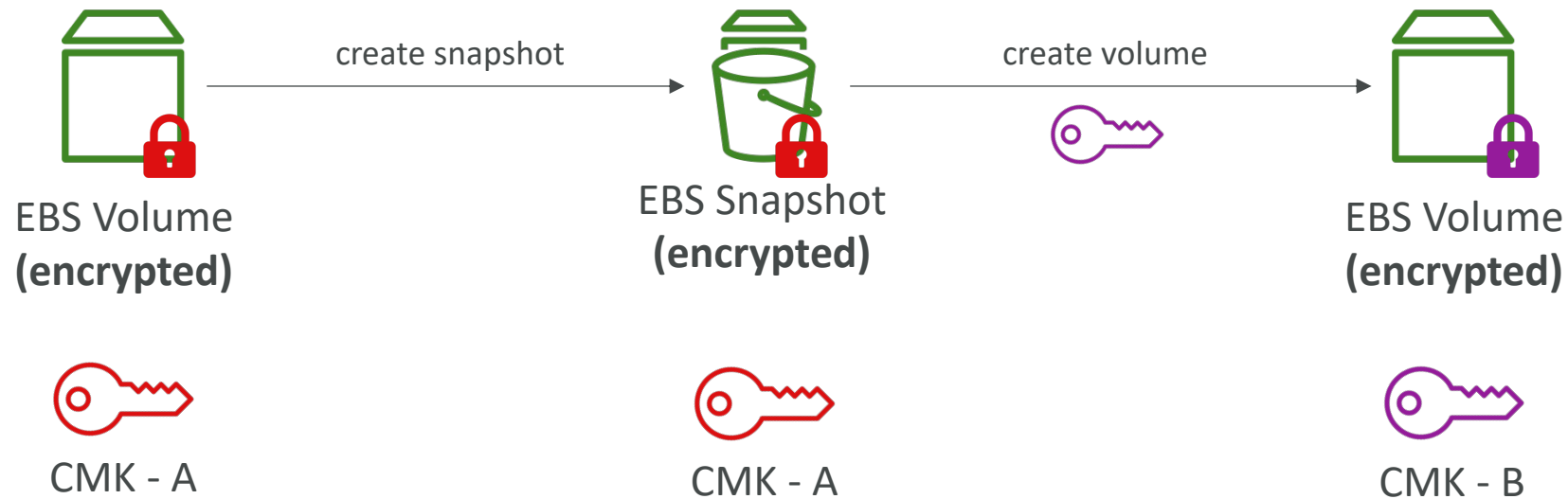
# KMS Alias Updating

- Better to use aliases in this case (to hide the change of key for the application)



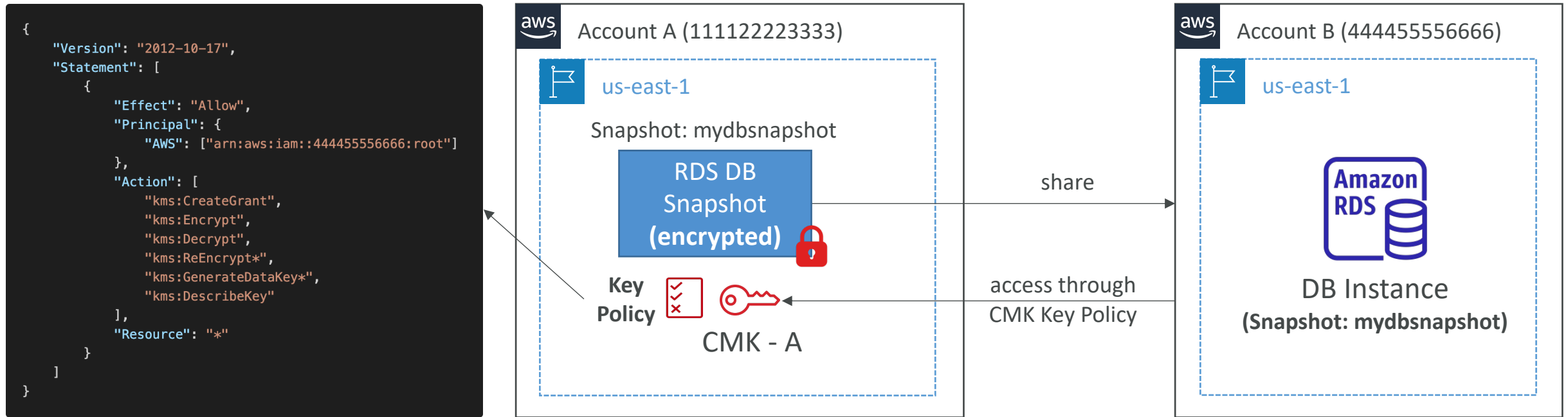
# Changing The KMS Key For An Encrypted EBS Volume

- You can't change the encryption keys used by an EBS volume
- Create an EBS snapshot and create a new EBS volume and specify the new KMS key



# Sharing KMS Encrypted RDS DB Snapshots

- You can share RDS DB snapshots encrypted with KMS CMK with other accounts, but must first share the KMS CMK with the target account using Key Policy



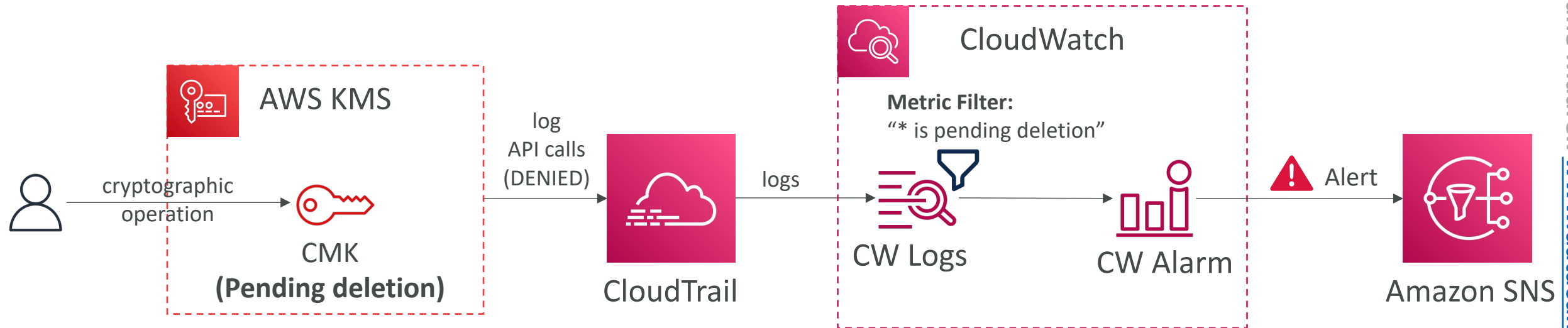
# KMS Key Deletion Considerations

- Schedule CMK for deletion with a waiting period of 7 to 30 days
  - CMK's status is “Pending deletion” during the waiting period
- During the CMK's deletion waiting period:
  - The CMK can't be used for cryptographic operations (e.g., can't decrypt KMS-encrypted objects in S3 – SSE-KMS)
  - The key is not rotated even if planned
- You can cancel the key deletion during the waiting period
- Consider disabling your key instead of deleting it if you're not sure!



# KMS Key Deletion – CloudWatch Alarm

- Use CloudTrail, CloudWatch Logs, CloudWatch Alarms and SNS to be notified when someone tries to use a CMK that's "Pending deletion" in a cryptographic operation (Encrypt, Decrypt, ...)

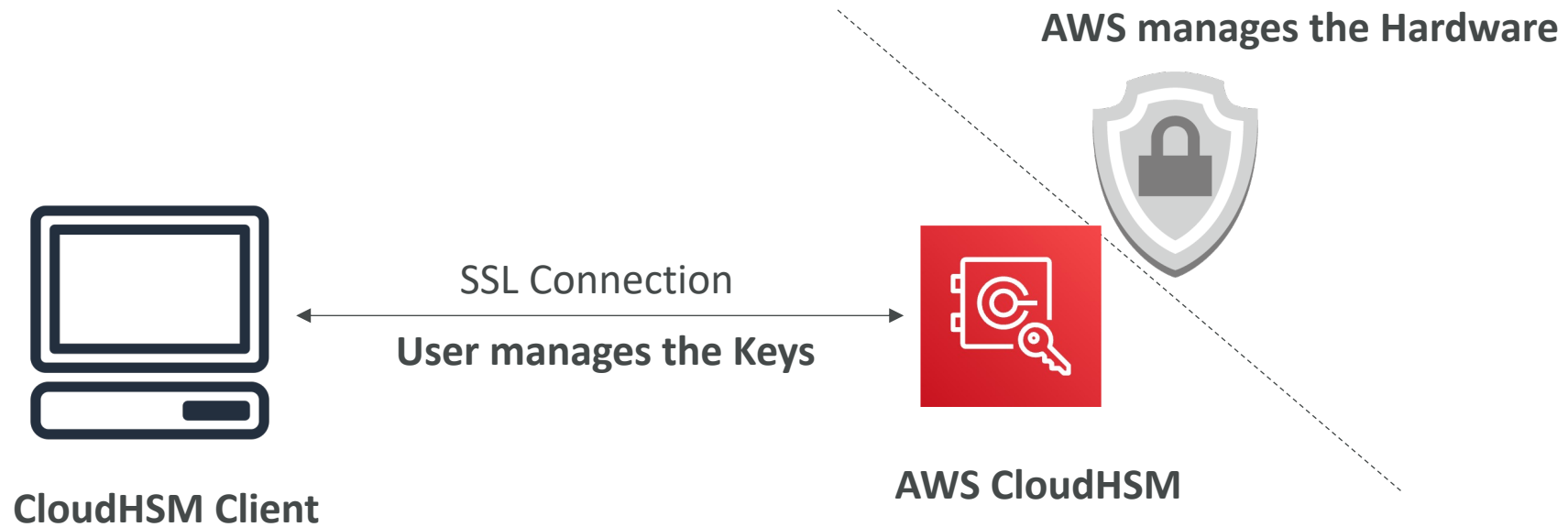


# CloudHSM



- KMS => AWS manages the software for encryption
- CloudHSM => AWS provisions encryption **hardware**
- Dedicated Hardware (HSM = Hardware Security Module)
- You manage your own encryption keys entirely (not AWS)
- HSM device is tamper resistant, FIPS 140-2 Level 3 compliance
- Supports both symmetric and **asymmetric** encryption (SSL/TLS keys)
- No free tier available
- Must use the CloudHSM Client Software
- Redshift supports CloudHSM for database encryption and key management
- Good option to use with SSE-C encryption

# CloudHSM Diagram



## IAM permissions:

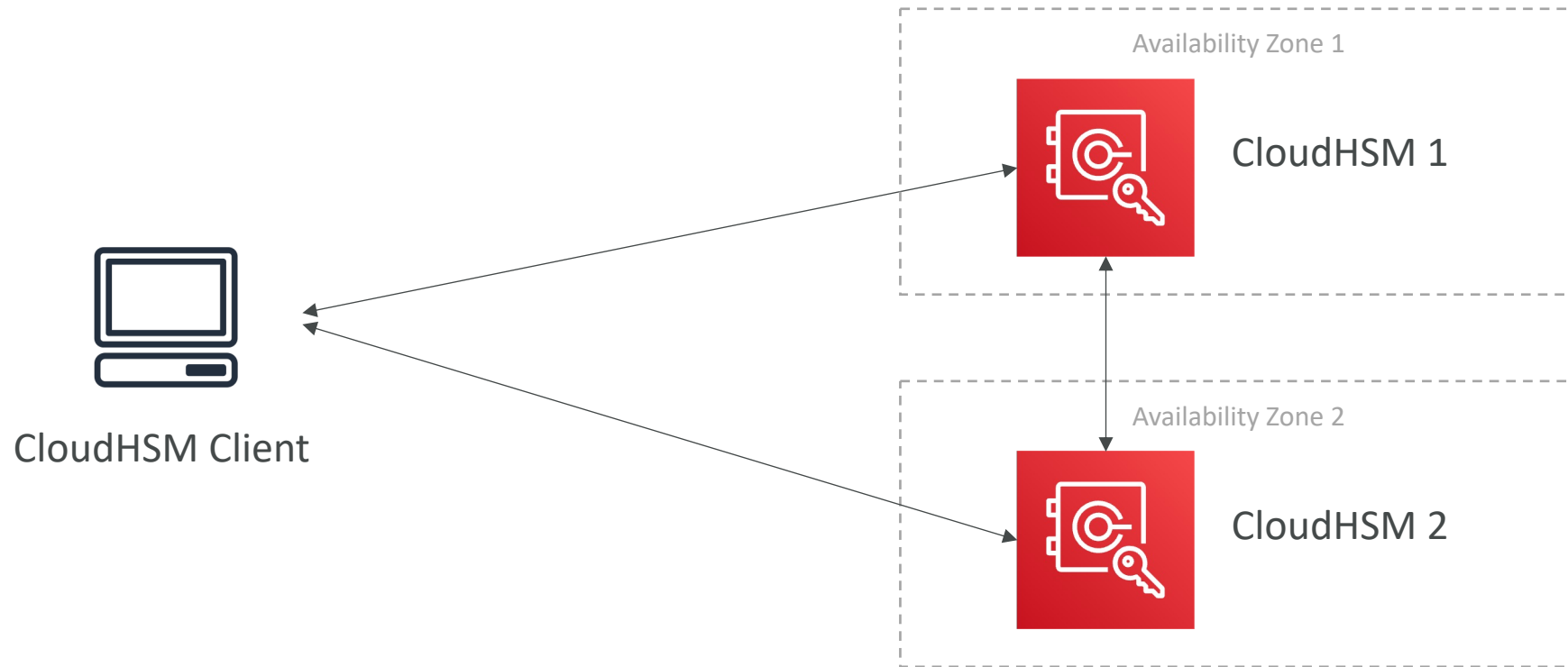
- CRUD an HSM Cluster

## CloudHSM Software:

- Manage the Keys
- Manage the Users

# CloudHSM – High Availability

- CloudHSM clusters are spread across Multi AZ (HA)
- Great for availability and durability



# CloudHSM vs. KMS

Feature	AWS KMS	AWS CloudHSM
Tenancy	Multi-Tenant	Single-Tenant
Standard	FIPS 140-2 Level 2	FIPS 140-2 Level 3
Master Keys	<ul style="list-style-type: none"><li>• AWS Owned CMK</li><li>• AWS Managed CMK</li><li>• Customer Managed CMK</li></ul>	Customer Managed CMK
Key Types	<ul style="list-style-type: none"><li>• Symmetric</li><li>• Asymmetric</li><li>• Digital Signing</li></ul>	<ul style="list-style-type: none"><li>• Symmetric</li><li>• Asymmetric</li><li>• Digital Signing &amp; Hashing</li></ul>
Key Accessibility	Accessible in multiple AWS regions (can't access keys outside the region it's created in)	<ul style="list-style-type: none"><li>• Deployed and managed in a VPC</li><li>• Can be shared across VPCs (VPC Peering)</li></ul>
Cryptographic Acceleration	None	<ul style="list-style-type: none"><li>• SSL/TLS Acceleration</li><li>• Oracle TDE Acceleration</li></ul>
Access & Authentication	AWS IAM	You create users and manage their permissions

# CloudHSM vs. KMS

Feature	AWS KMS	AWS CloudHSM
High Availability	AWS Managed Service	Add multiple HSMs over different AZs
Audit Capability	<ul style="list-style-type: none"><li>• CloudTrail</li><li>• CloudWatch</li></ul>	<ul style="list-style-type: none"><li>• CloudTrail</li><li>• CloudWatch</li><li>• MFA support</li></ul>
Free Tier	Yes	No

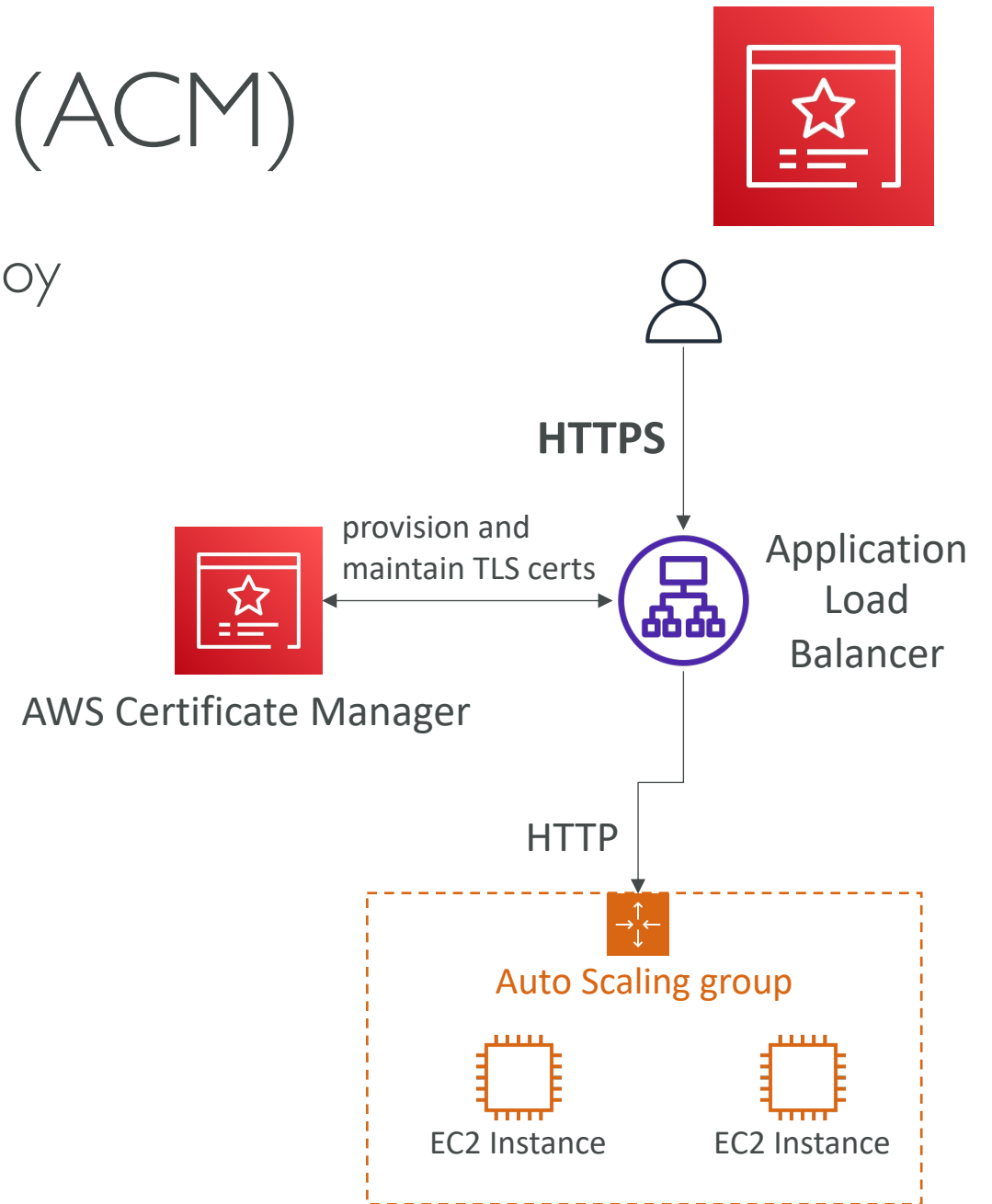
# AWS Artifact (not really a service)



- Portal that provides customers with on-demand access to AWS compliance documentation and AWS agreements
- [Artifact Reports](#) - Allows you to download AWS security and compliance documents from third-party auditors, like AWS ISO certifications, Payment Card Industry (PCI), and System and Organization Control (SOC) reports
- [Artifact Agreements](#) - Allows you to review, accept, and track the status of AWS agreements such as the Business Associate Addendum (BAA) or the Health Insurance Portability and Accountability Act (HIPAA) for an individual account or in your organization
- Can be used to support internal audit or compliance

# AWS Certificate Manager (ACM)

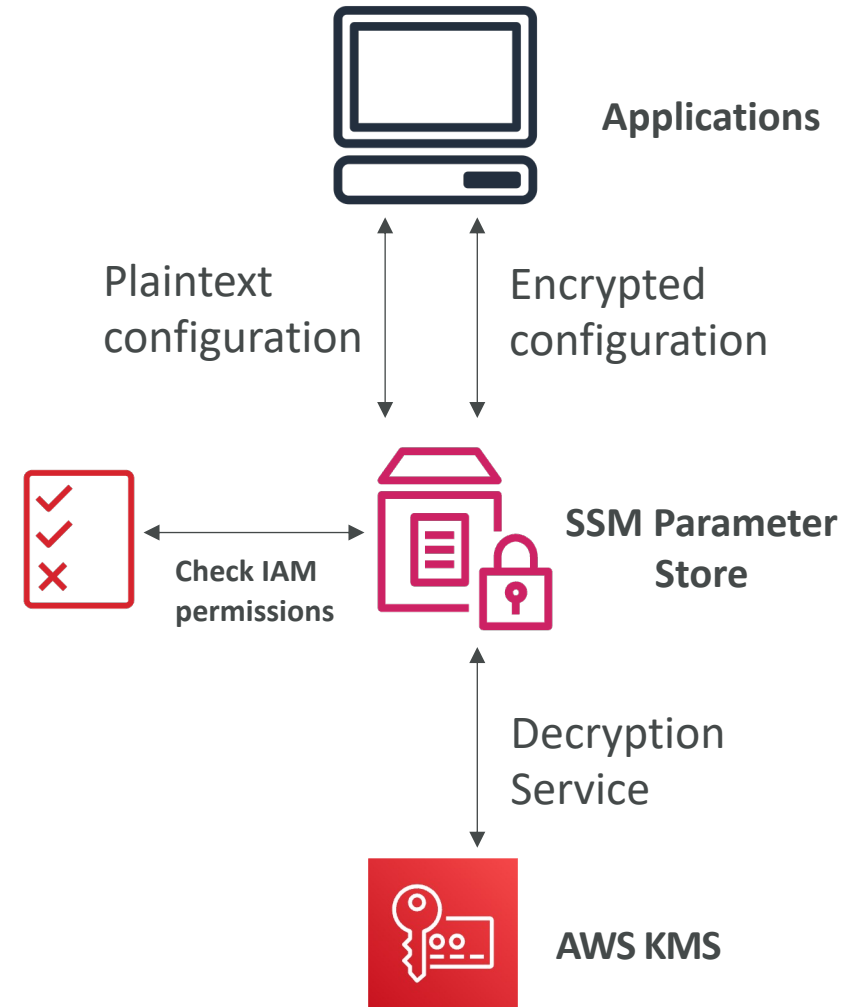
- Let's you easily provision, manage, and deploy **SSL/TLS Certificates**
- Used to provide in-flight encryption for websites (HTTPS)
- Supports both public and private TLS certificates
- Free of charge for public TLS certificates
- Automatic TLS certificate renewal
- Integrations with (load TLS certificates on)
  - Elastic Load Balancers
  - CloudFront Distributions
  - APIs on API Gateway





# SSM Parameter Store

- Secure storage for configuration and secrets
- Optional Seamless Encryption using KMS
- Serverless, scalable, durable, easy SDK
- Version tracking of configurations / secrets
- Configuration management using path & IAM
- Notifications with CloudWatch Events
- Integration with CloudFormation



# SSM Parameter Store Hierarchy

- /my-department/

- my-app/

- dev/

- db-url

- db-password

- prod/

- db-url

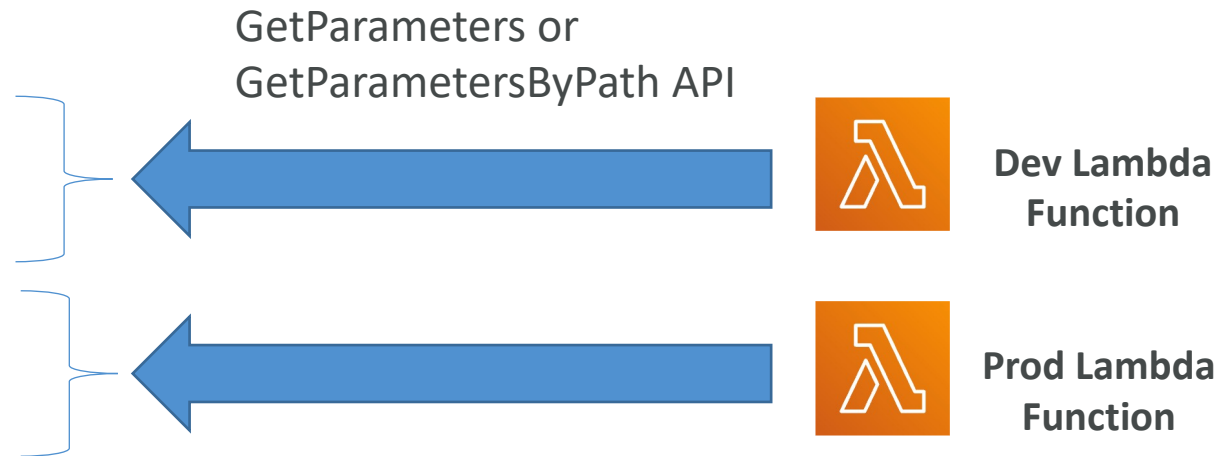
- db-password

- other-app/

- /other-department/

- */aws/reference/secretsmanager/secret\_ID\_in\_Secrets\_Manager*

- */aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86\_64-gp2*



# 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
API Interaction Pricing (higher throughput = up to 1000 Transactions per second)	Standard Throughput: free Higher Throughput: \$0.05 per 10,000 API interactions	Standard Throughput: \$0.05 per 10,000 API interactions Higher Throughput: \$0.05 per 10,000 API interactions

# Parameters Policies (for advanced parameters)

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

## Expiration (to delete a parameter)

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

## ExpirationNotification (CW Events)

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

## NoChangeNotification (CW Events)

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

# AWS Secrets Manager



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

# SSM Parameter Store vs Secrets Manager

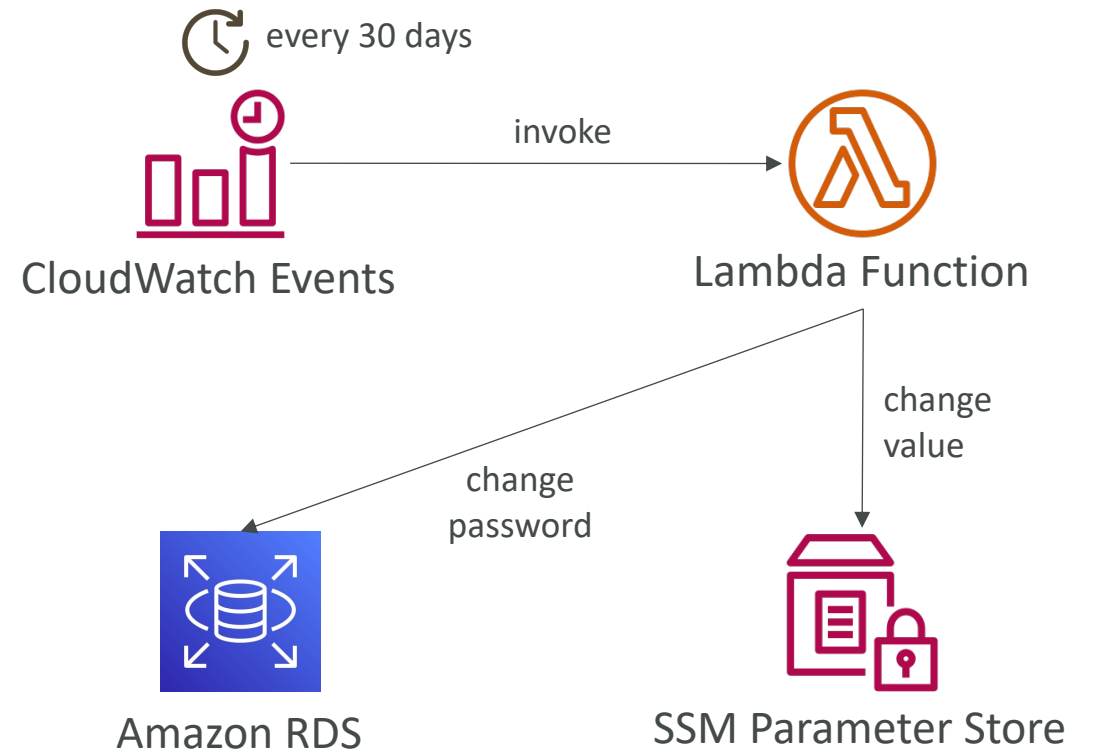
- **Secrets Manager (\$\$\$):**
  - Automatic rotation of secrets with AWS Lambda
  - Lambda function is provided for RDS, Redshift, DocumentDB
  - KMS encryption is mandatory
  - Can integration with CloudFormation
- **SSM Parameter Store (\$):**
  - Simple API
  - No secret rotation (can enable rotation using Lambda triggered by CW Events)
  - KMS encryption is optional
  - Can integration with CloudFormation
  - Can pull a Secrets Manager secret using the SSM Parameter Store API

# SSM Parameter Store vs. Secrets Manager Rotation

## AWS Secrets Manager

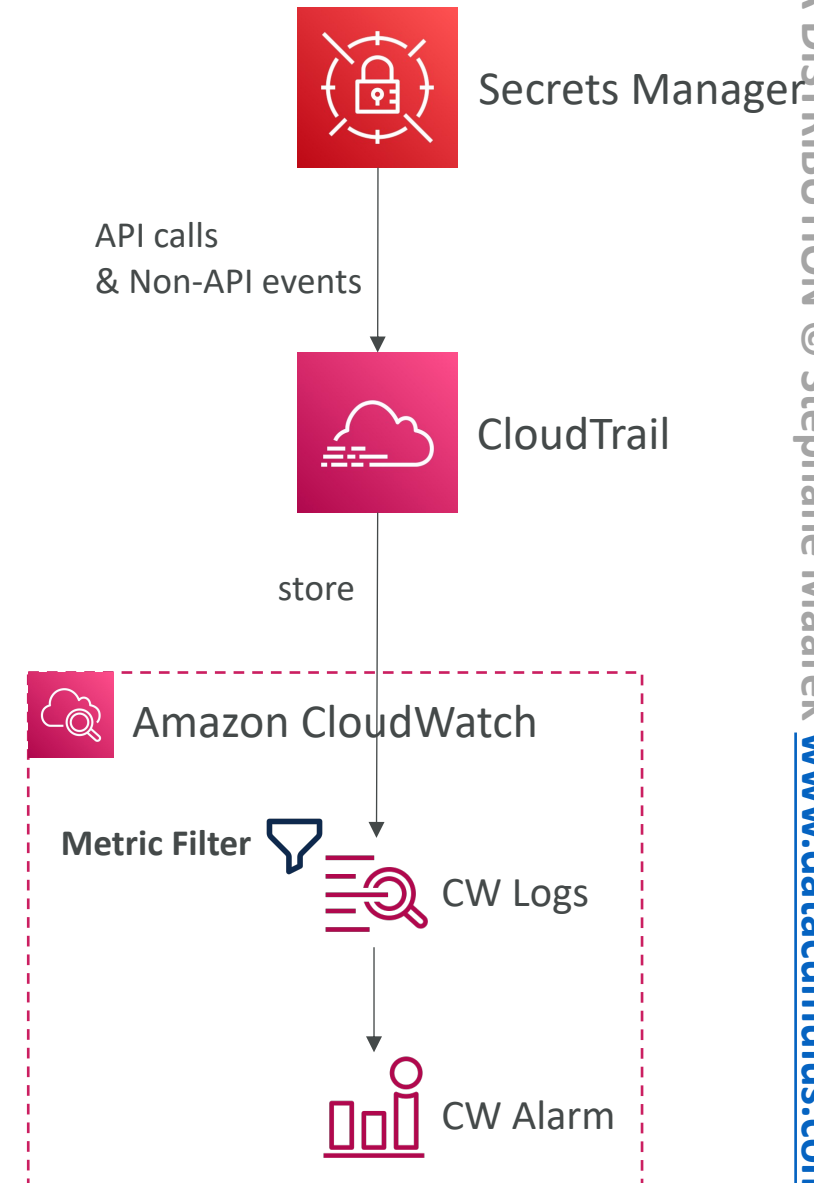


## SSM Parameter Store



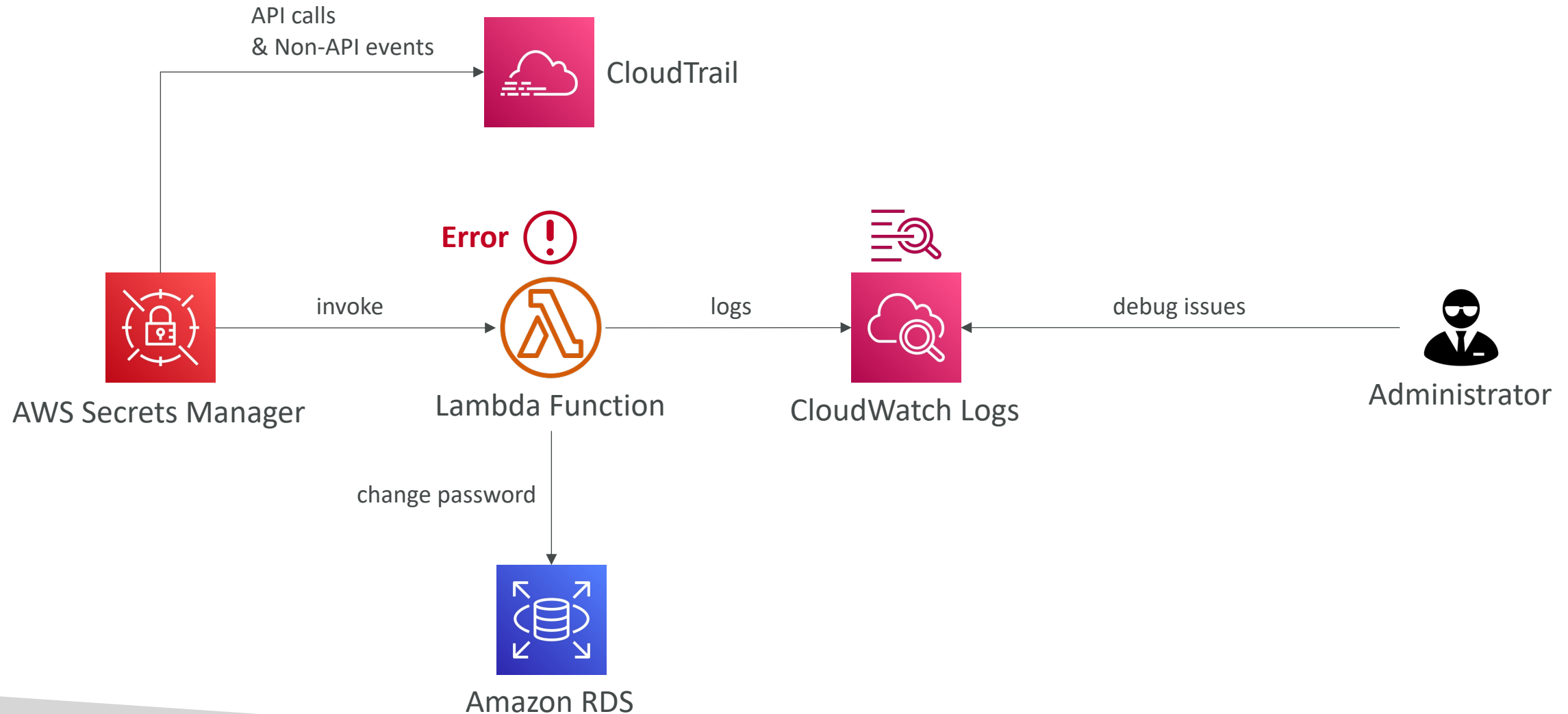
# Secrets Manager – Monitoring

- CloudTrail captures API calls to the Secrets Manager API
- CloudTrail captures other related events that might have a security or compliance impact on your AWS account or might help you troubleshoot operational problems.
- **CloudTrail records these events as non-API service events.**
  - RotationStarted event
  - RotationSucceeded event
  - RotationFailed event
  - RotationAbandoned event – a manual change to a secret instead of automated rotation
  - StartSecretVersionDelete event
  - CancelSecretVersionDelete event
  - EndSecretVersionDelete event
- Combine with CloudWatch Logs and CloudWatch alarms for automations





# Secrets Manager – Troubleshooting Rotation



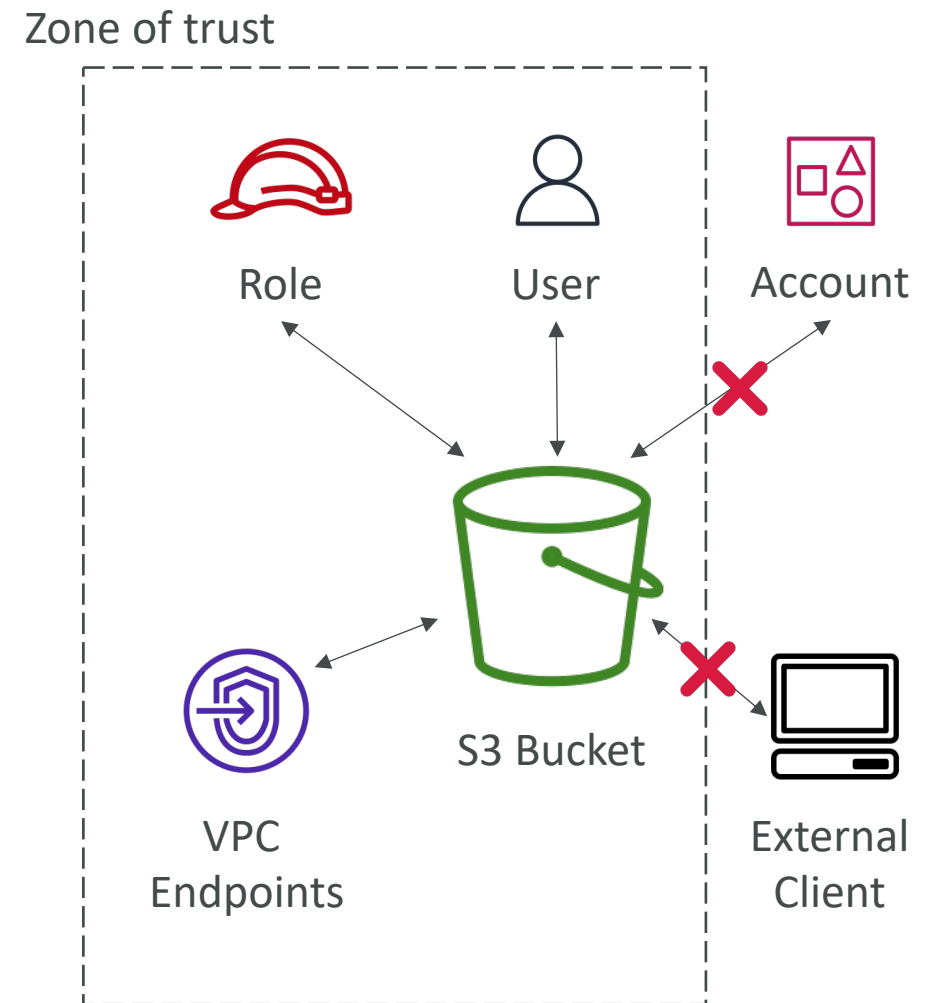
# Identity

# IAM Security Tools

- IAM Credentials Report (account-level)
  - a report that lists all your account's users and the status of their various credentials
- IAM Access Advisor (user-level)
  - Access advisor shows the service permissions granted to a user and when those services were last accessed.
  - You can use this information to revise your policies.

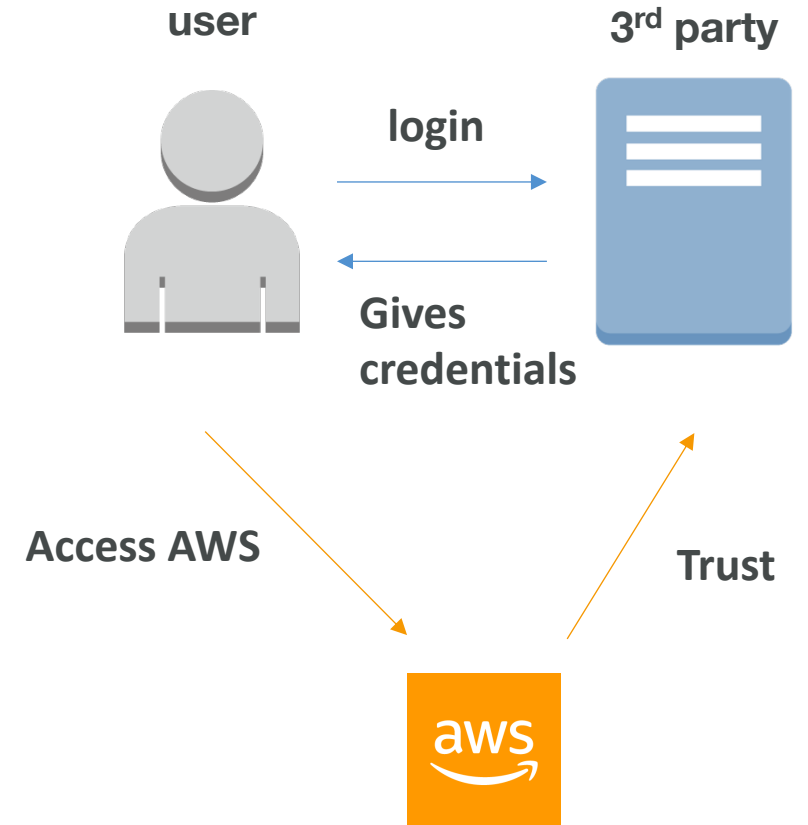
# IAM Access Analyzer

- Find out which resources are shared externally
  - S3 Buckets
  - IAM Roles
  - KMS Keys
  - Lambda Functions and Layers
  - SQS queues
  - Secrets Manager Secrets
- Define **Zone of Trust** = AWS Account or AWS Organization
- Access outside zone of trusts => findings



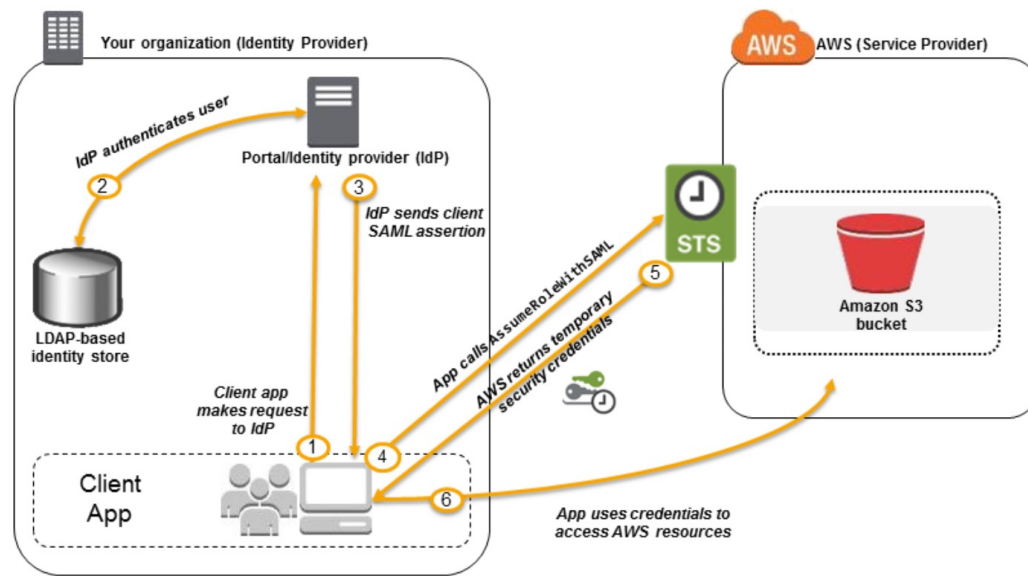
# What's Identity Federation?

- Federation lets users outside of AWS to assume temporary role for accessing AWS resources.
- These users assume identity provided access role.
- Federation assumes a form of 3rd party authentication
  - LDAP
  - Microsoft Active Directory (~= SAML)
  - Single Sign On
  - Open ID
  - Cognito
- Using federation, you don't need to create IAM users (user management is outside of AWS)

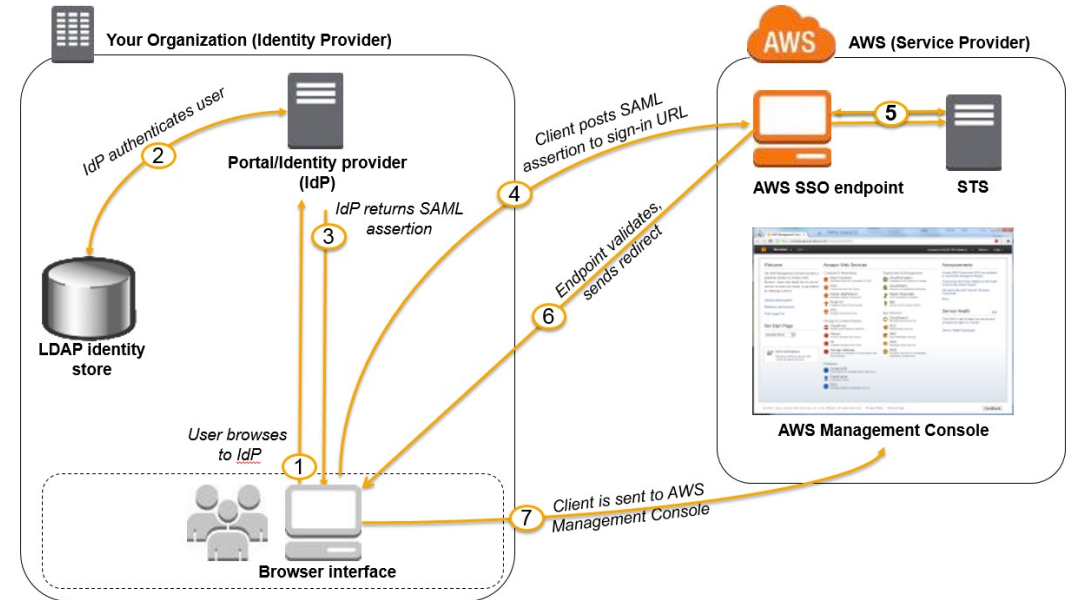


# SAML Federation For Enterprises

- To integrate Active Directory / ADFS with AWS (or any SAML 2.0)
- Provides access to AWS Console or CLI (through temporary creds)
- No need to create an IAM user for each of your employees



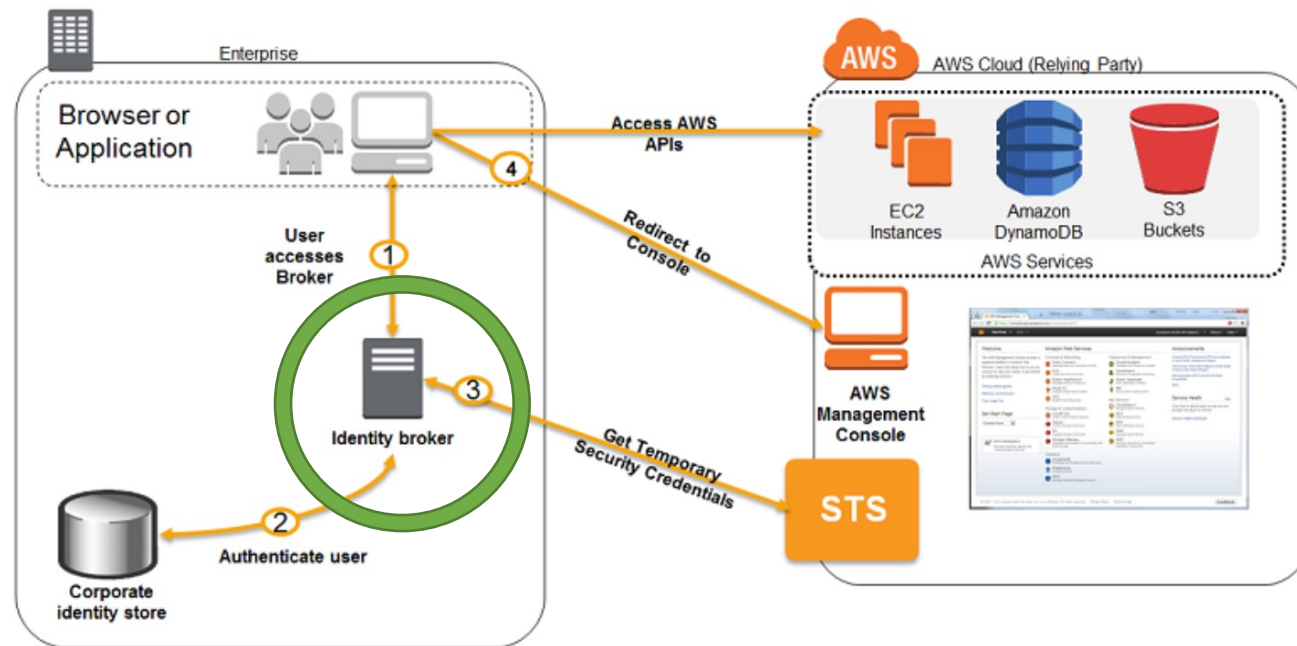
[https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_roles\\_providers\\_saml.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_saml.html)



[https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_roles\\_providers\\_enable-console-saml.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_enable-console-saml.html)

# Custom Identity Broker Application For Enterprises

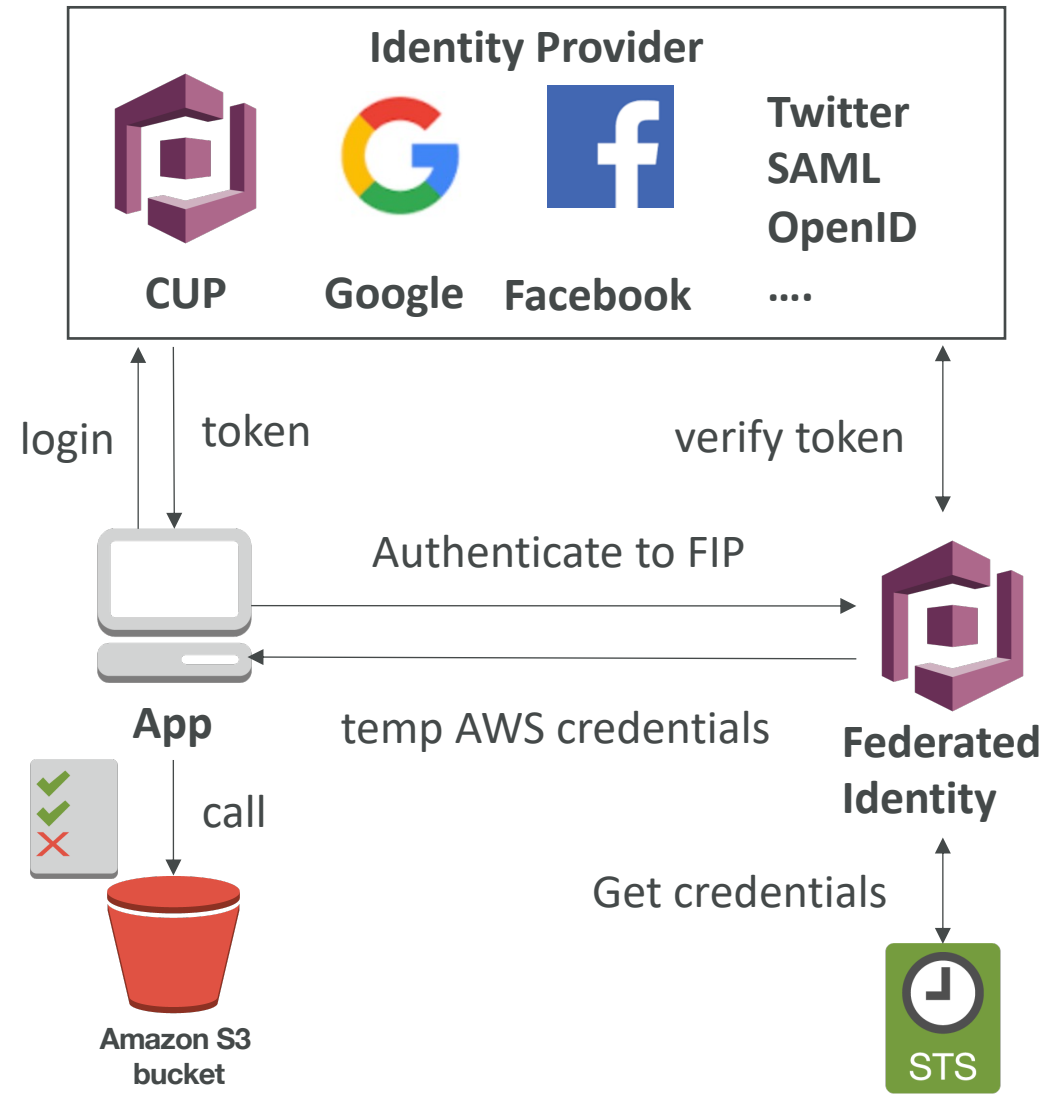
- Use only if identity provider is not compatible with SAML 2.0
- The identity broker must determine the appropriate IAM policy



[https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_roles\\_common-scenarios\\_federated-users.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_common-scenarios_federated-users.html)

# AWS Cognito - Federated Identity Pools For Public Applications

- **Goal:**
  - Provide direct access to AWS Resources from the Client Side
- **How:**
  - Log in to federated identity provider – or remain anonymous
  - Get temporary AWS credentials back from the Federated Identity Pool
  - These credentials come with a pre-defined IAM policy stating their permissions
- **Example:**
  - provide (temporary) access to write to S3 bucket using Facebook Login
- **Note:**
  - Web Identity Federation is an alternative to using Cognito but AWS recommends against it





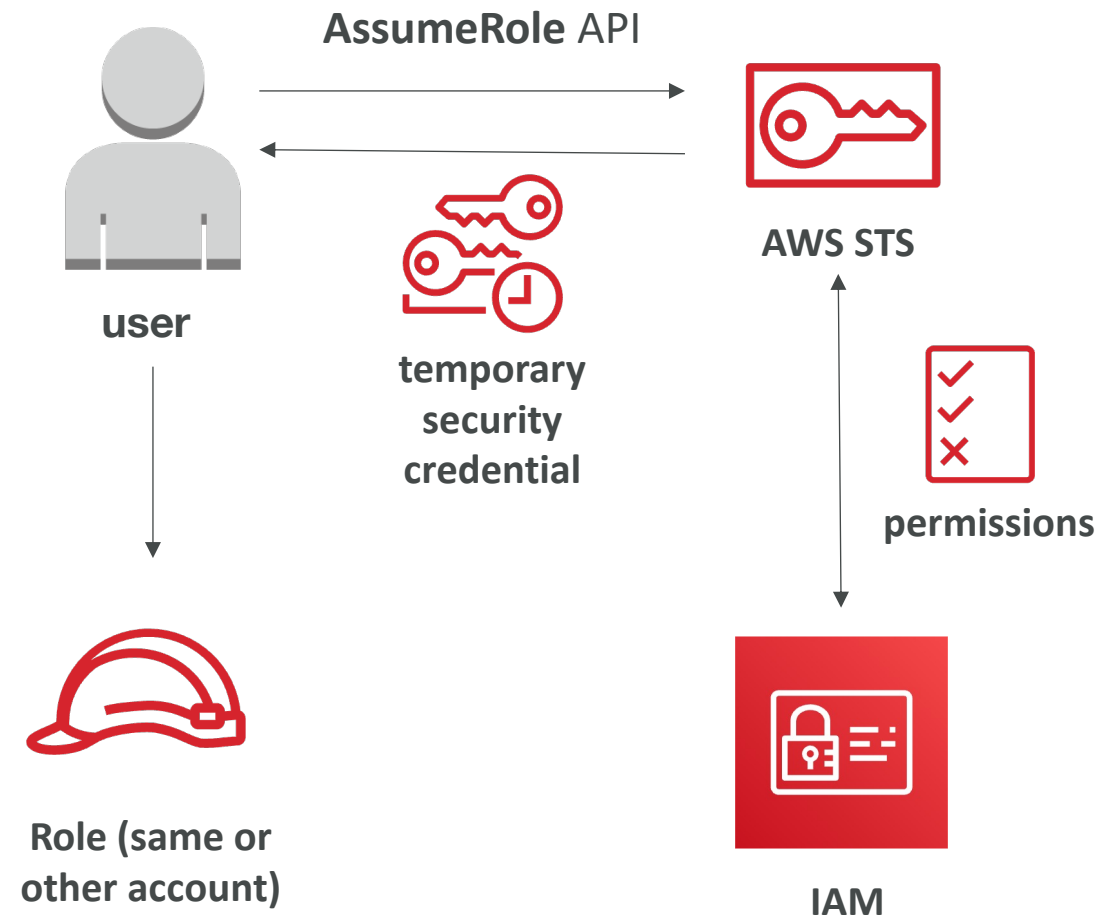


# AWS STS – Security Token Service

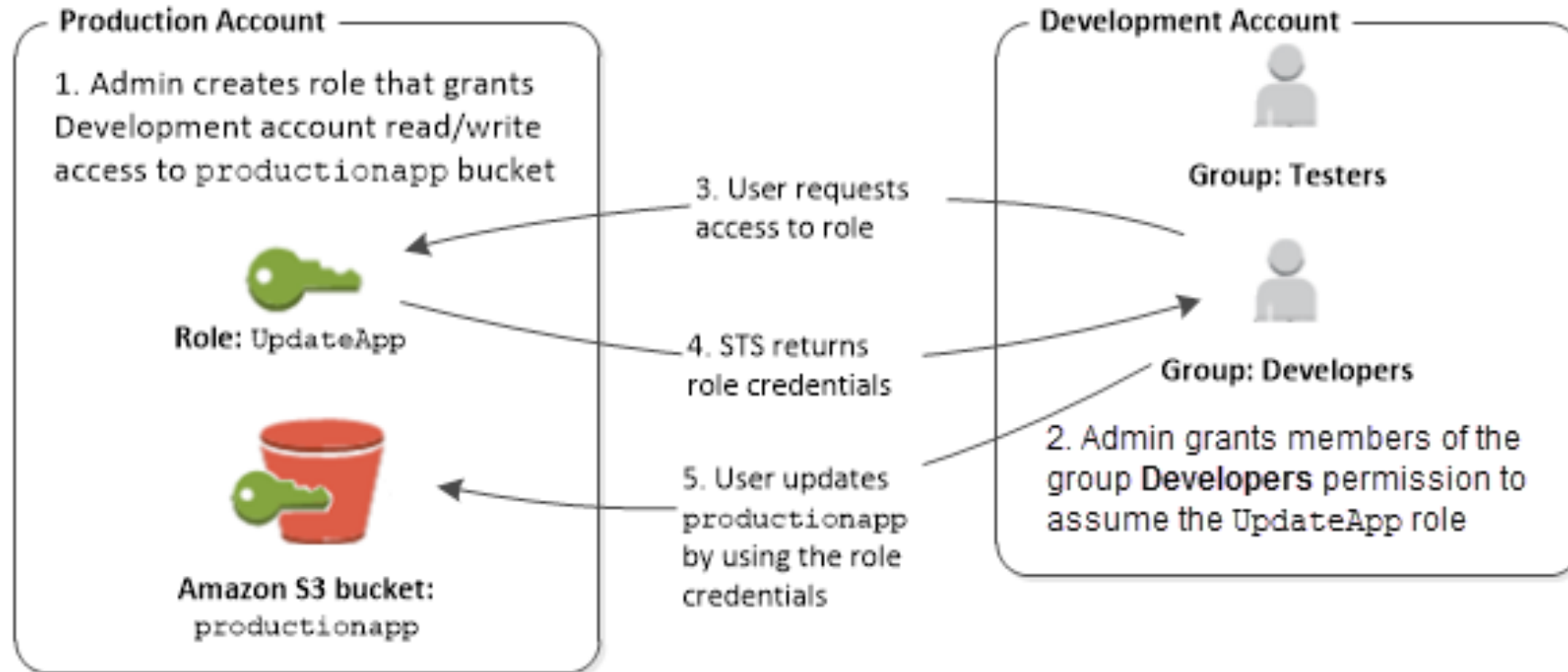
- Allows to grant limited and temporary access to AWS resources.
- Token is valid for up to one hour (must be refreshed)
- **AssumeRole**
  - Within your own account: for enhanced security
  - Cross Account Access: assume role in target account to perform actions there
- **AssumeRoleWithSAML**
  - return credentials for users logged with SAML
- **AssumeRoleWithWebIdentity**
  - return creds for users logged with an IdP (Facebook Login, Google Login, OIDC compatible...)
  - AWS recommends against using this, and using **Cognito** instead
- **GetSessionToken**
  - for MFA, from a user or AWS account root user

# Using STS to Assume a Role

- Define an IAM Role within your account or cross-account
- Define which principals can access this IAM Role
- Use AWS STS (Security Token Service) to retrieve credentials and impersonate the IAM Role you have access to (**AssumeRole API**)
- Temporary credentials can be valid between 15 minutes to 1 hour



# Cross account access with STS

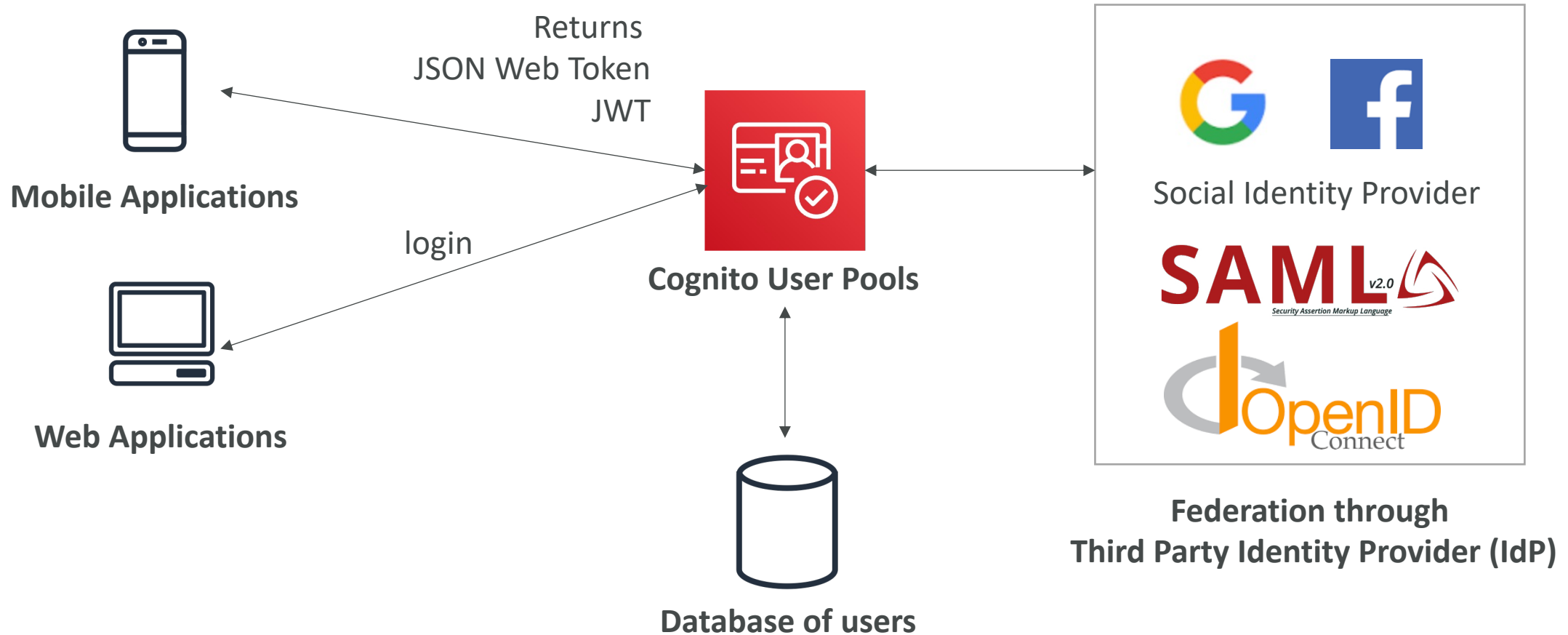


[https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_roles\\_common-scenarios\\_aws-accounts.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_common-scenarios_aws-accounts.html)

# Cognito User Pools (CUP) – User Features

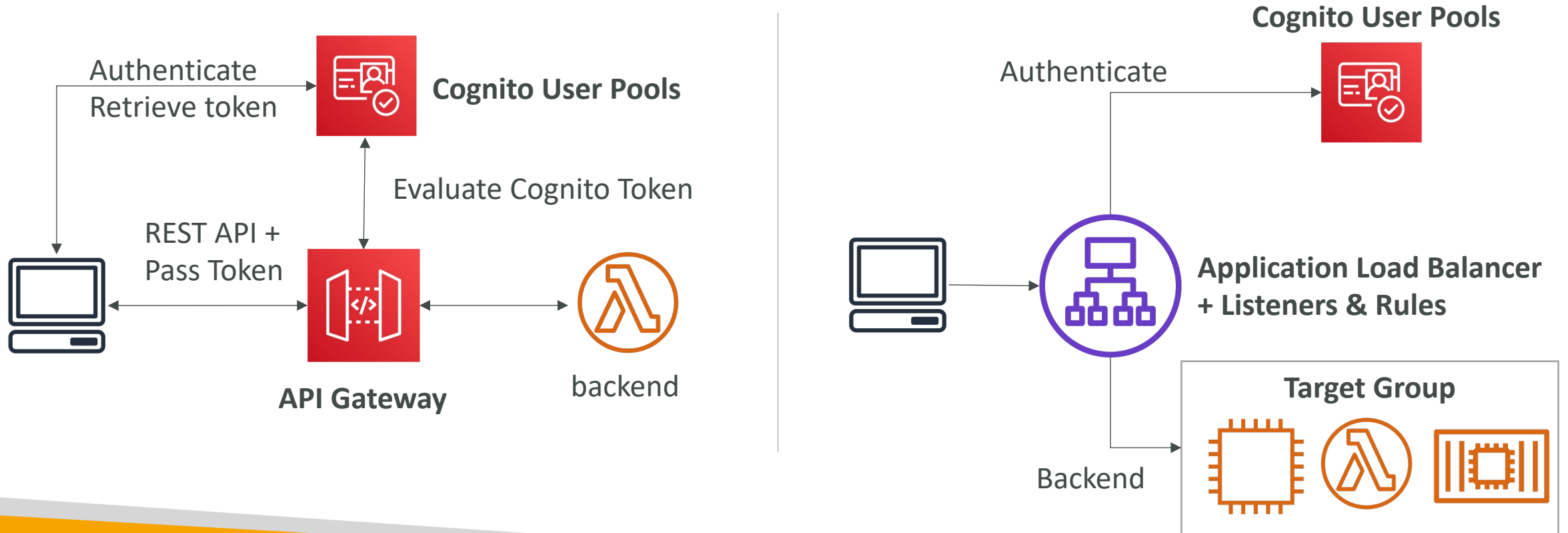
- Create a serverless database of user for your web & mobile apps
- Simple login: Username (or email) / password combination
- Password reset
- Email & Phone Number Verification
- Multi-factor authentication (MFA)
- Federated Identities: users from Facebook, Google, SAML...
- Feature: block users if their credentials are compromised elsewhere
- Login sends back a JSON Web Token (JWT)

# Cognito User Pools (CUP) – Diagram



# Cognito User Pools (CUP) - Integrations

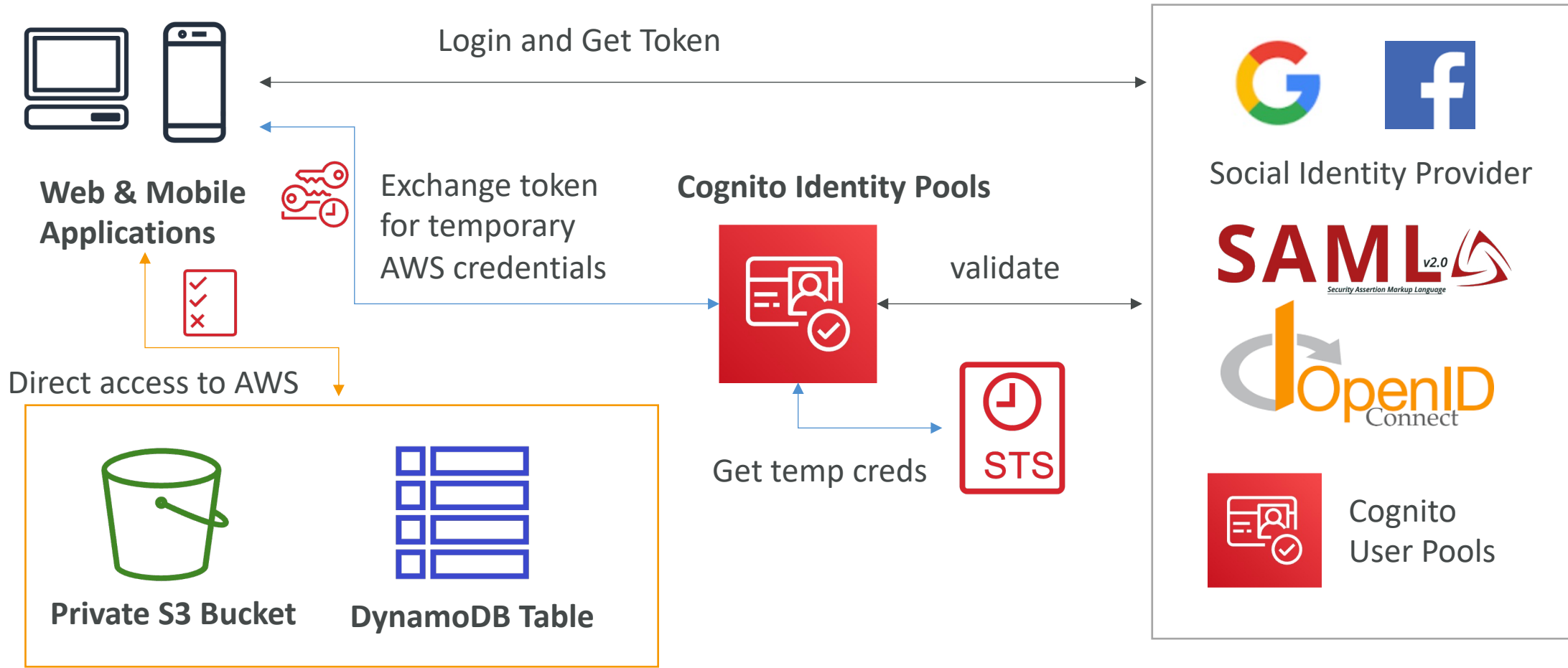
- CUP integrates with **API Gateway** and **Application Load Balancer**



# Cognito Identity Pools (Federated Identities)

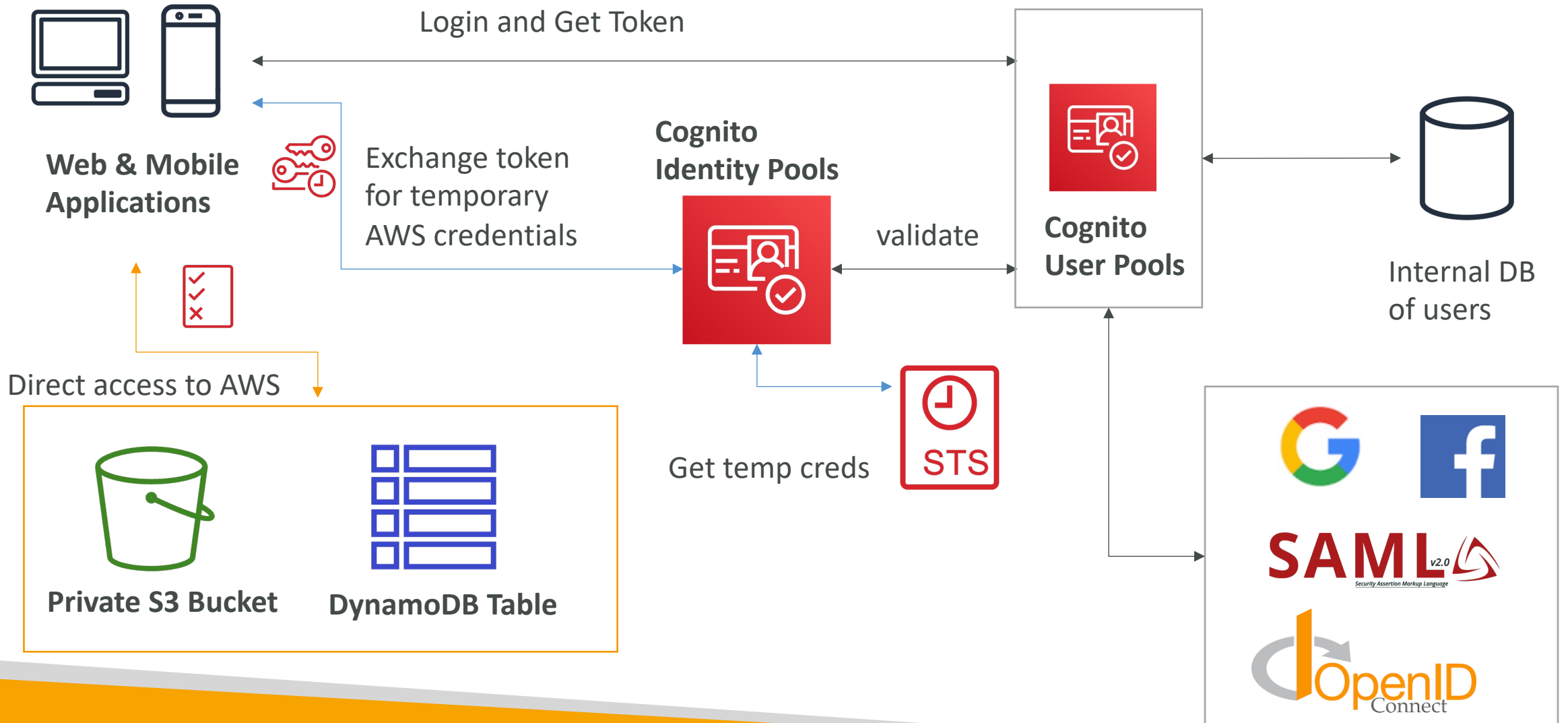
- Get identities for “users” so they obtain temporary AWS credentials
- Your identity pool (e.g identity source) can include:
  - Public Providers (Login with Amazon, Facebook, Google, Apple)
  - Users in an Amazon Cognito user pool
  - OpenID Connect Providers & SAML Identity Providers
  - Developer Authenticated Identities (custom login server)
  - Cognito Identity Pools allow for unauthenticated (guest) access
- Users can then access AWS services directly or through API Gateway
  - The IAM policies applied to the credentials are defined in Cognito
  - They can be customized based on the user\_id for fine grained control

# Cognito Identity Pools – Diagram





# Cognito Identity Pools – Diagram with CUP



# Cognito Identity Pools – IAM Roles

- Default IAM roles for authenticated and guest users
- Define rules to choose the role for each user based on the user's ID
- You can partition your users' access using **policy variables**
- IAM credentials are obtained by Cognito Identity Pools through STS
- The roles must have a “trust” policy of Cognito Identity Pools

# Cognito Identity Pools – Guest User example

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::mybucket/assets/my_picture.jpg"
      ]
    }
  ]
}
```

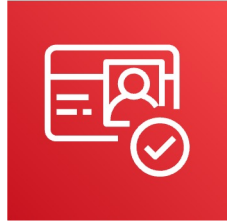
# Cognito Identity Pools – Policy variable on S3

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": ["s3:ListBucket"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::mybucket"],
      "Condition": {"StringLike": {"s3:prefix": ["${cognito-identity.amazonaws.com:sub}/*"]}}
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::mybucket,${cognito-identity.amazonaws.com:sub}/*"]
    }
  ]
}
```

# Cognito Identity Pools – DynamoDB

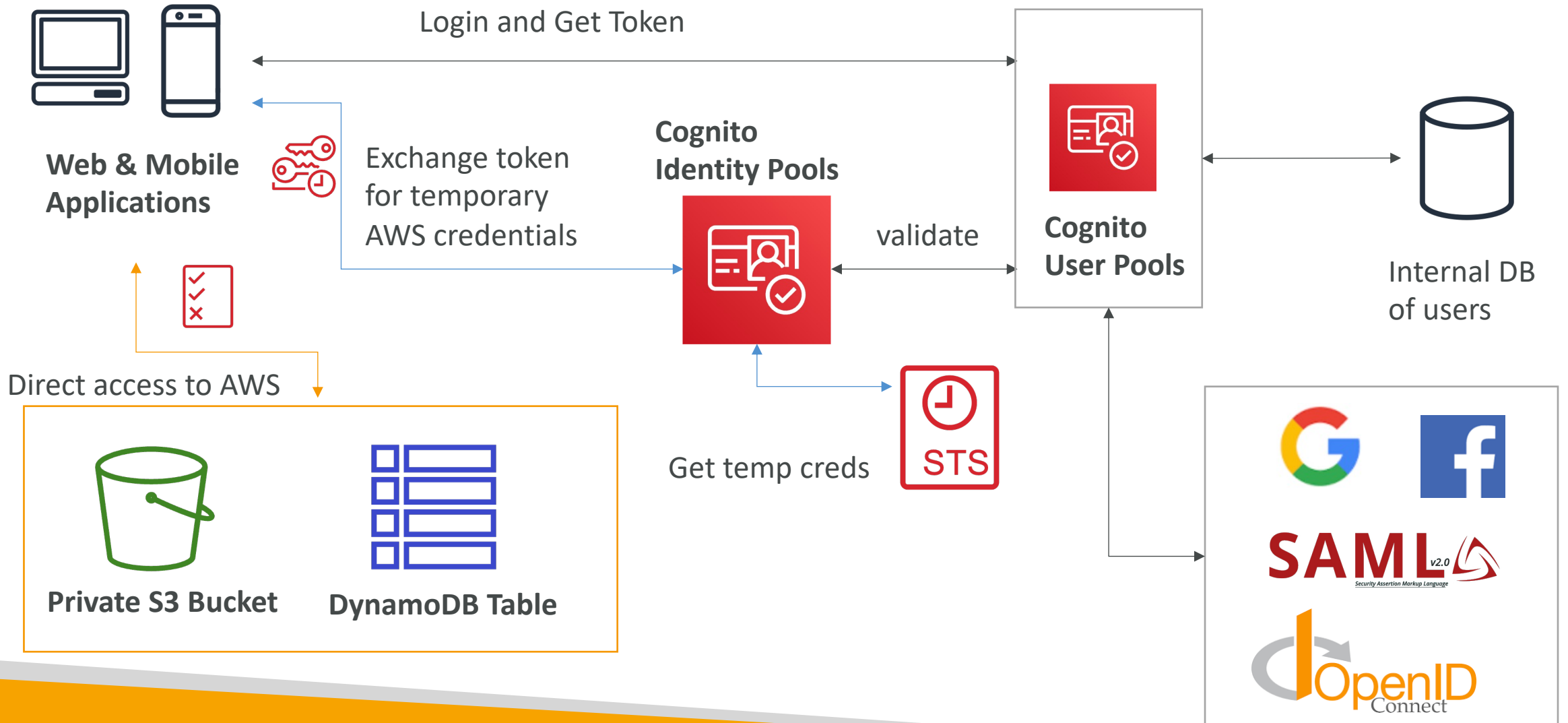
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:GetItem", "dynamodb:BatchGetItem", "dynamodb:Query",
        "dynamodb:PutItem", "dynamodb:UpdateItem", "dynamodb>DeleteItem",
        "dynamodb:BatchWriteItem"
      ],
      "Resource":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

# Cognito User Pools vs Identity Pools



- **Cognito User Pools:**
  - Database of users for your web and mobile application
  - Allows to federate logins through Public Social, OIDC, SAML...
  - Can customize the hosted UI for authentication (including the logo)]
  - Has triggers with AWS Lambda during the authentication flow
- **Cognito Identity Pools:**
  - Obtain AWS credentials for your users
  - Users can login through Public Social, OIDC, SAML & Cognito User Pools
  - Users can be unauthenticated (guests)
  - Users are mapped to IAM roles & policies, can leverage policy variables
- **CUP + CIP = manage user / password + access AWS services**

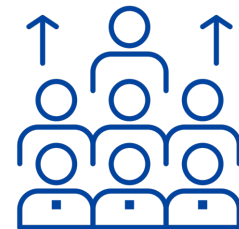
# Cognito Identity Pools – Diagram with CUP



# AWS IAM Identity Center (successor to AWS Single Sign-On)



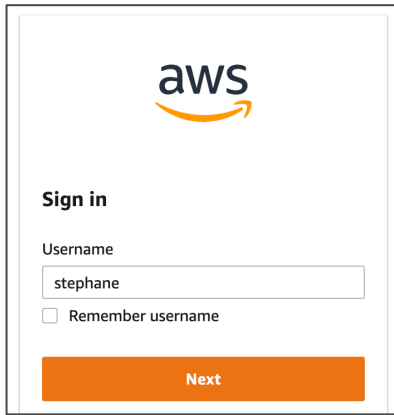
- One login (single sign-on) for all your
  - AWS accounts in AWS Organizations
  - Business cloud applications (e.g., Salesforce, Box, Microsoft 365, ...)
  - SAML2.0-enabled applications
  - EC2 Windows Instances
- Identity providers
  - Built-in identity store in IAM Identity Center
  - 3<sup>rd</sup>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



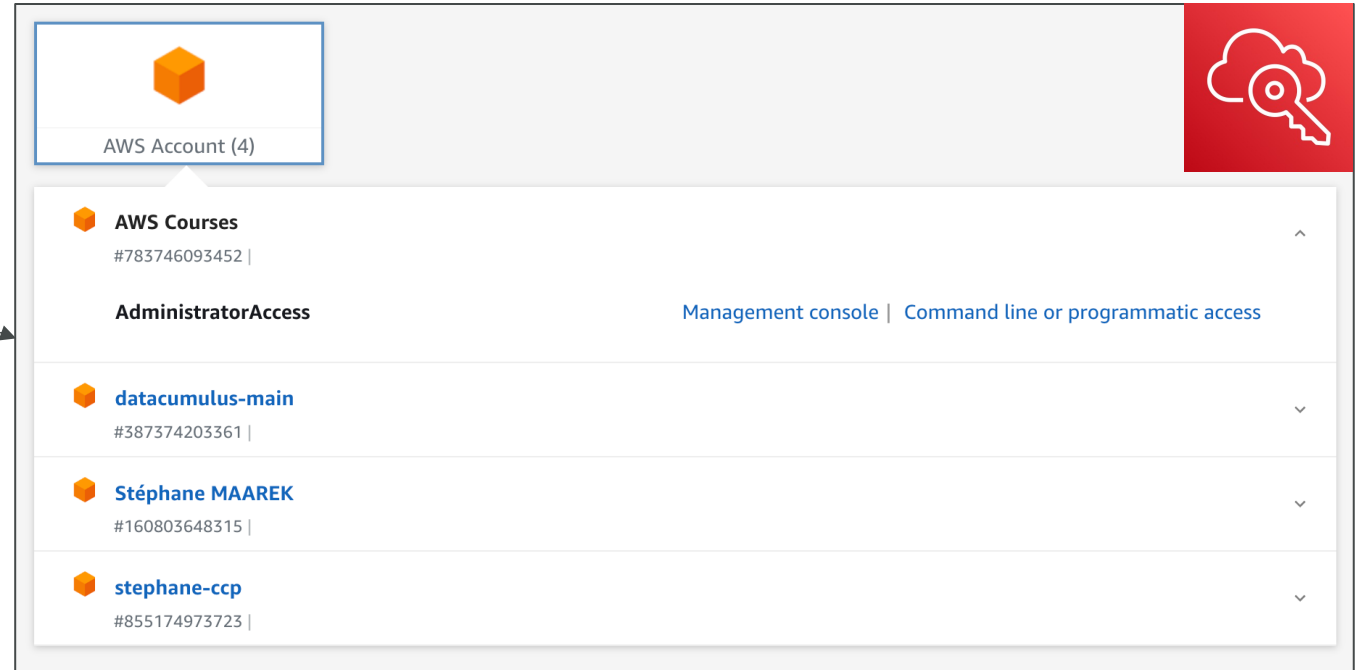


# AWS IAM Identity Center – Login Flow

## AWS IAM Identity Center

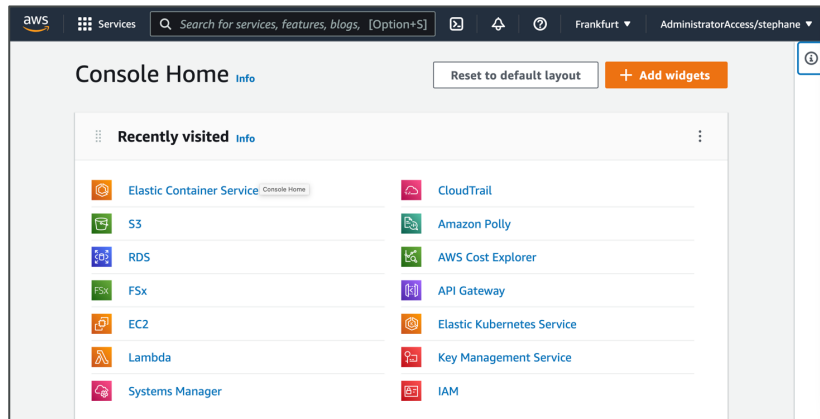


The sign-in form features the AWS logo at the top. Below it, the text "Sign in" is displayed. A "Username" label is followed by a text input field containing the name "stephane". A checkbox labeled "Remember username" is positioned below the input field. At the bottom of the form is a prominent orange "Next" button.



This screen displays a list of AWS accounts under the heading "AWS Account (4)". A red key icon is visible in the top right corner. The list includes:

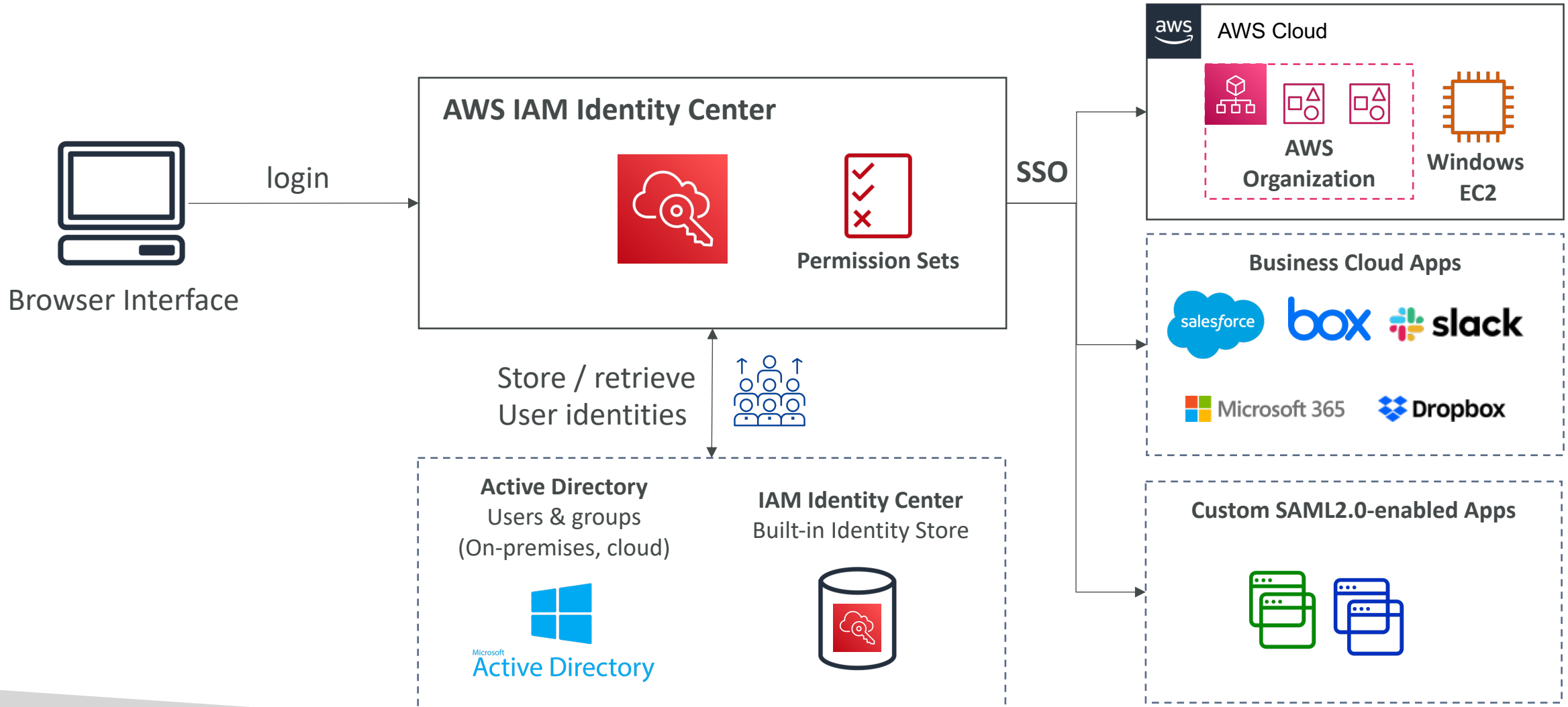
- AWS Courses** (#783746093452) with a link for "Management console" and "Command line or programmatic access".
- datacumulus-main** (#387374203361)
- Stéphane MAAREK** (#160803648315)
- stephane-ccp** (#855174973723)



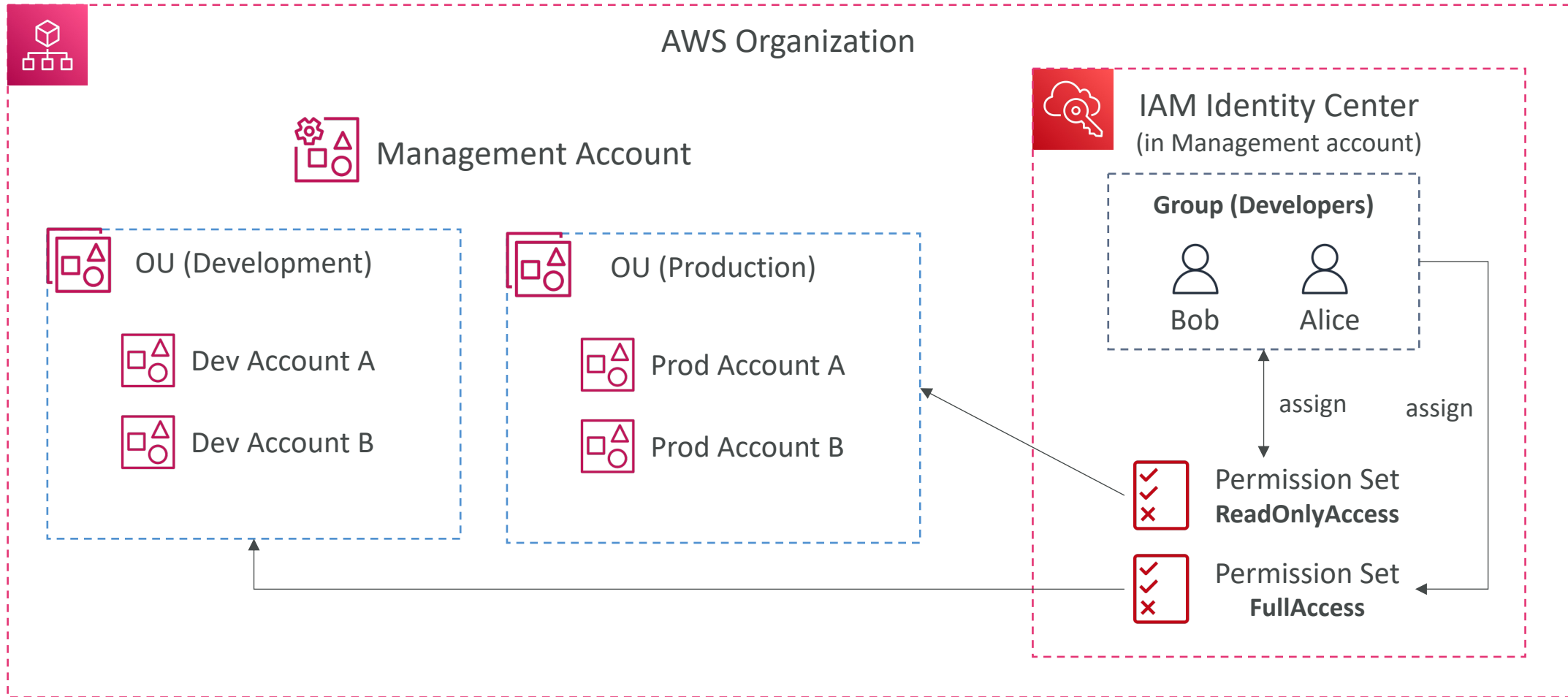
The "Console Home" dashboard shows a search bar at the top with the text "Search for services, features, blogs, [Option+S]". Below the search bar are buttons for "Reset to default layout" and "+ Add widgets". A "Recently visited" section lists various AWS services:

- Elastic Container Service
- S3
- RDS
- FSx
- EC2
- Lambda
- Systems Manager
- CloudTrail
- Amazon Polly
- AWS Cost Explorer
- API Gateway
- Elastic Kubernetes Service
- Key Management Service
- IAM

# AWS IAM Identity Center



# IAM Identity Center

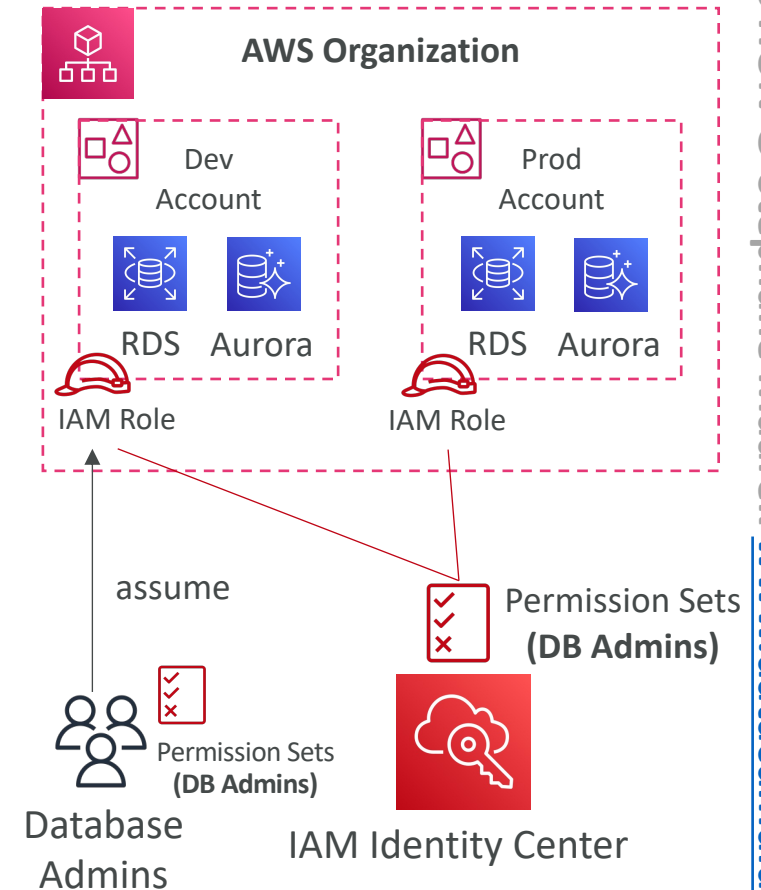


# AWS IAM Identity Center

## Fine-grained Permissions and Assignments

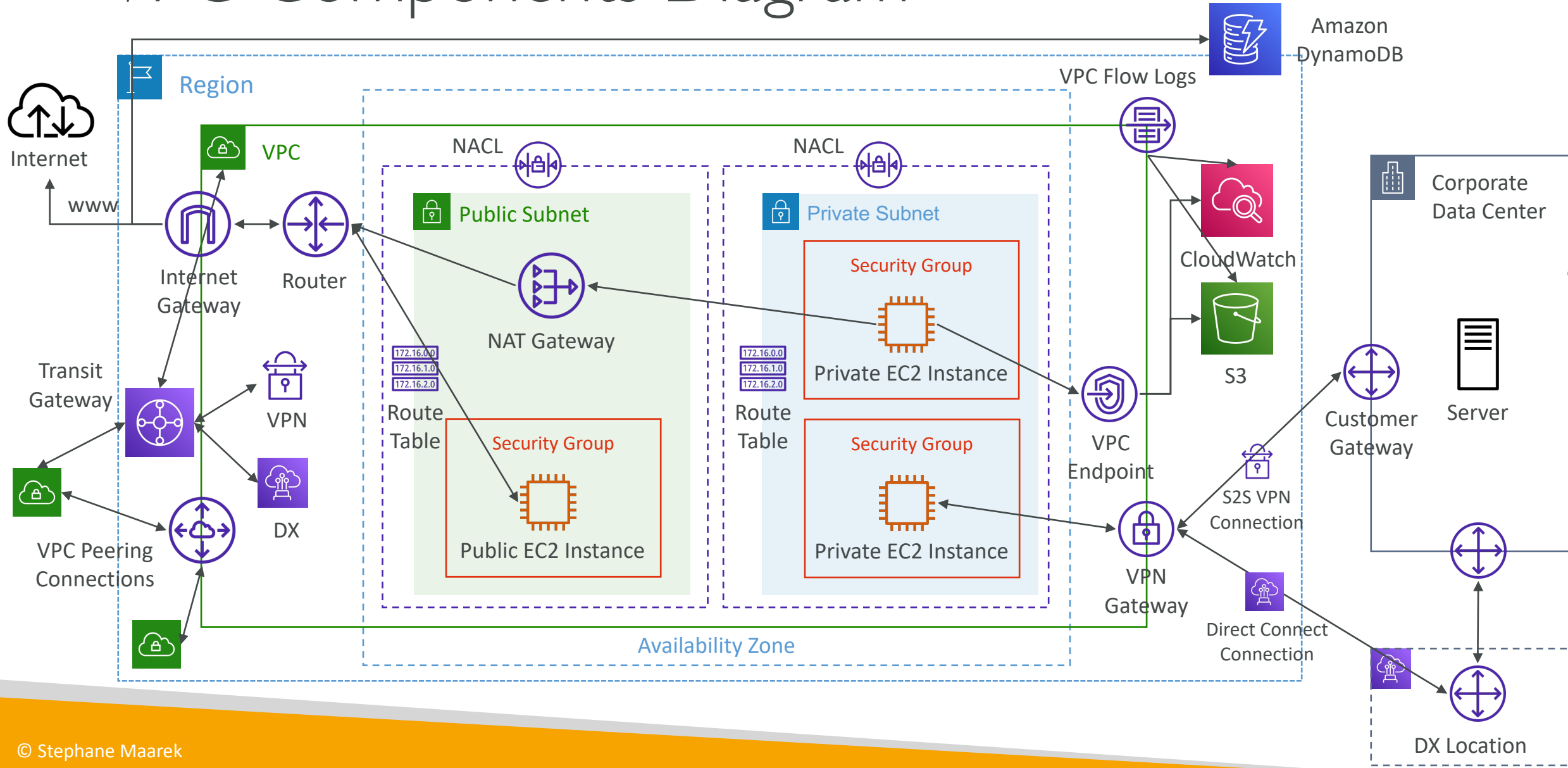


- **Multi-Account Permissions**
  - Manage access across AWS accounts in your AWS Organization
  - Permission Sets – a collection of one or more IAM Policies assigned to users and groups to define AWS access
- **Application Assignments**
  - SSO access to many SAML 2.0 business applications (Salesforce,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



# Virtual Private Cloud (VPC)

# VPC Components Diagram



# Understanding CIDR – IPv4

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

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

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

# Understanding CIDR – IPv4

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



# Understanding CIDR – Subnet Mask

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

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



## Quick Memo

### Octets



- /32** – no octet can change
- /24** – last octet can change
- /16** – last 2 octets can change
- /8** – last 3 octets can change
- /0** – all octets can change

# Understanding CIDR – Little Exercise

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

# Public vs. Private IP (IPv4)

- The Internet Assigned Numbers Authority (IANA) established certain blocks of IPv4 addresses for the use of private (LAN) and public (Internet) addresses
- **Private IP** can only allow certain values:
  - 10.0.0.0 – 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) ← e.g., home networks
- All the rest of the IP addresses on the Internet are Public

# Default VPC Walkthrough

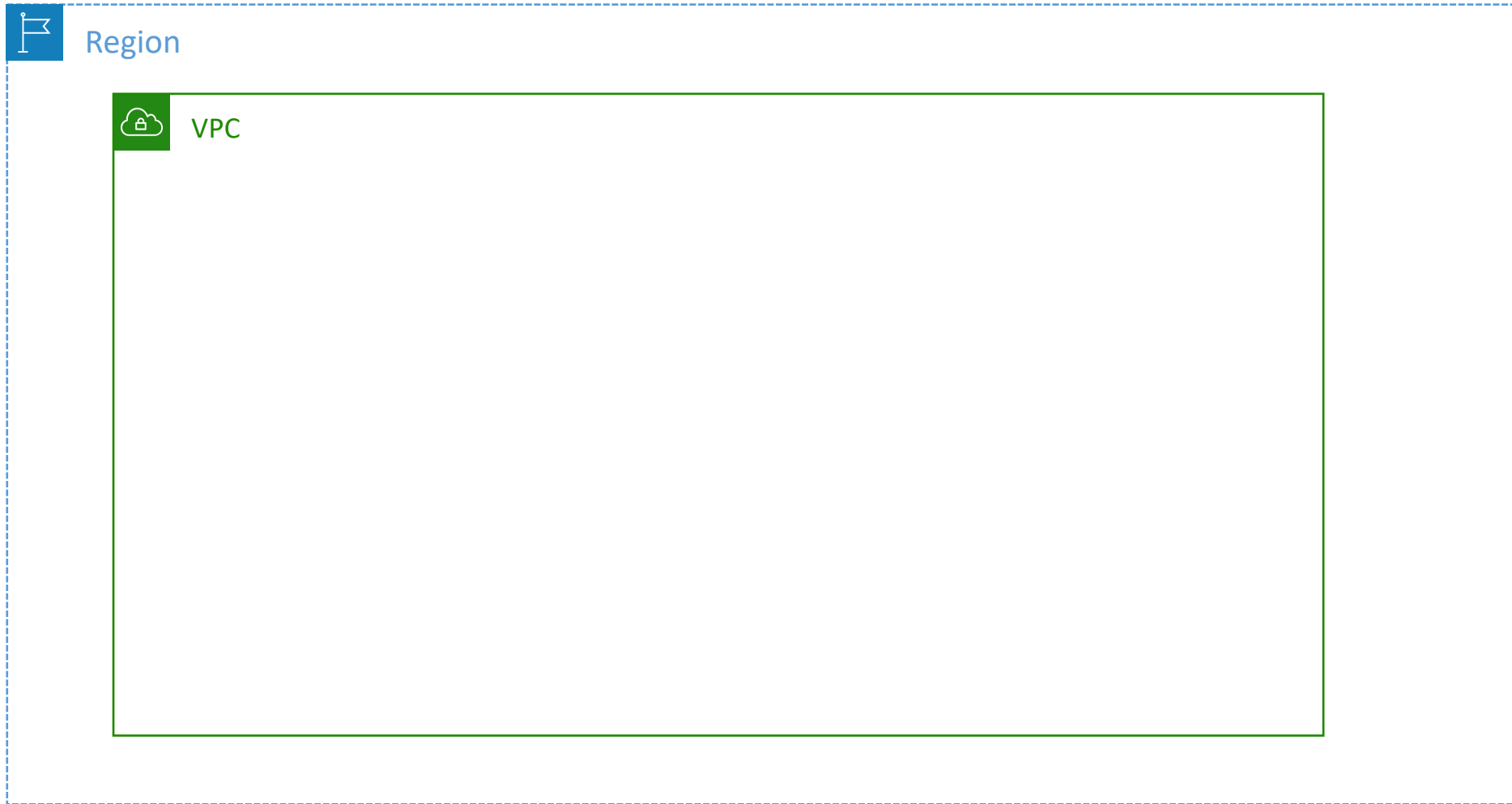
- All new AWS accounts have a default VPC
- New EC2 instances are launched into the default VPC if no subnet is specified
- Default VPC has Internet connectivity and all EC2 instances inside it have public IPv4 addresses
- We also get a public and a private IPv4 DNS names



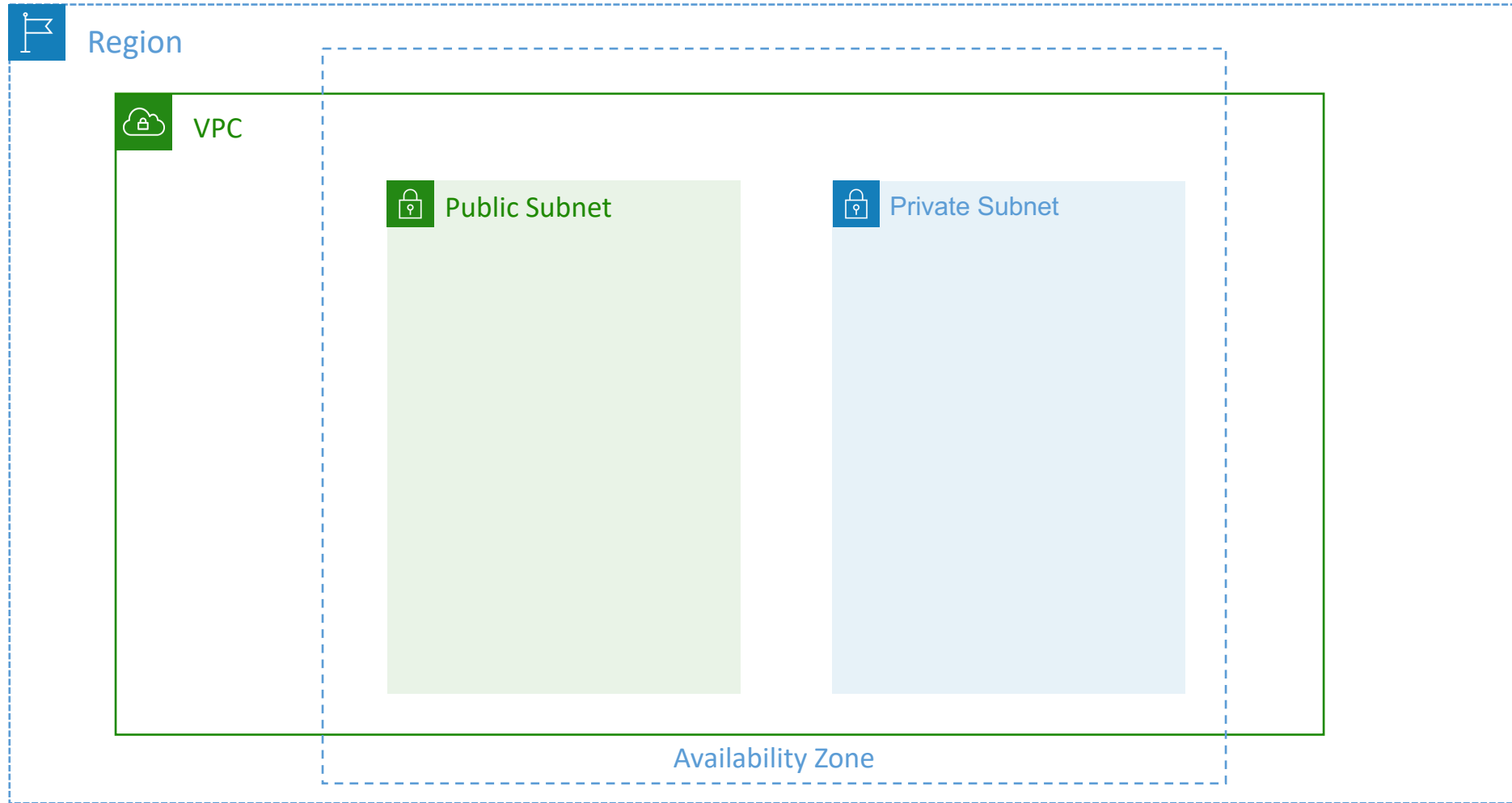
# VPC in AWS – IPv4

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

# State of Hands-on



# Adding Subnets





# VPC – Subnet (IPv4)

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

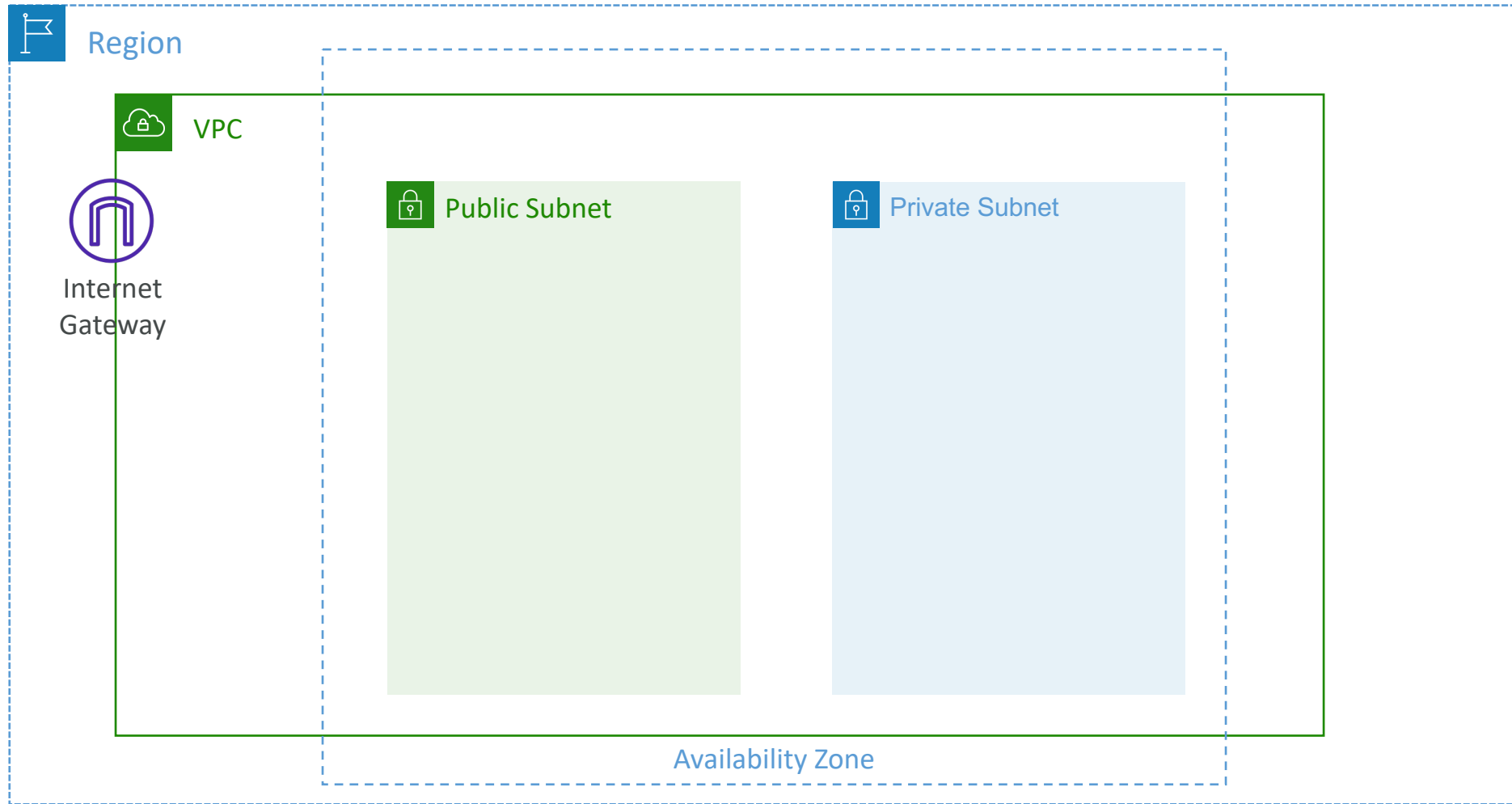




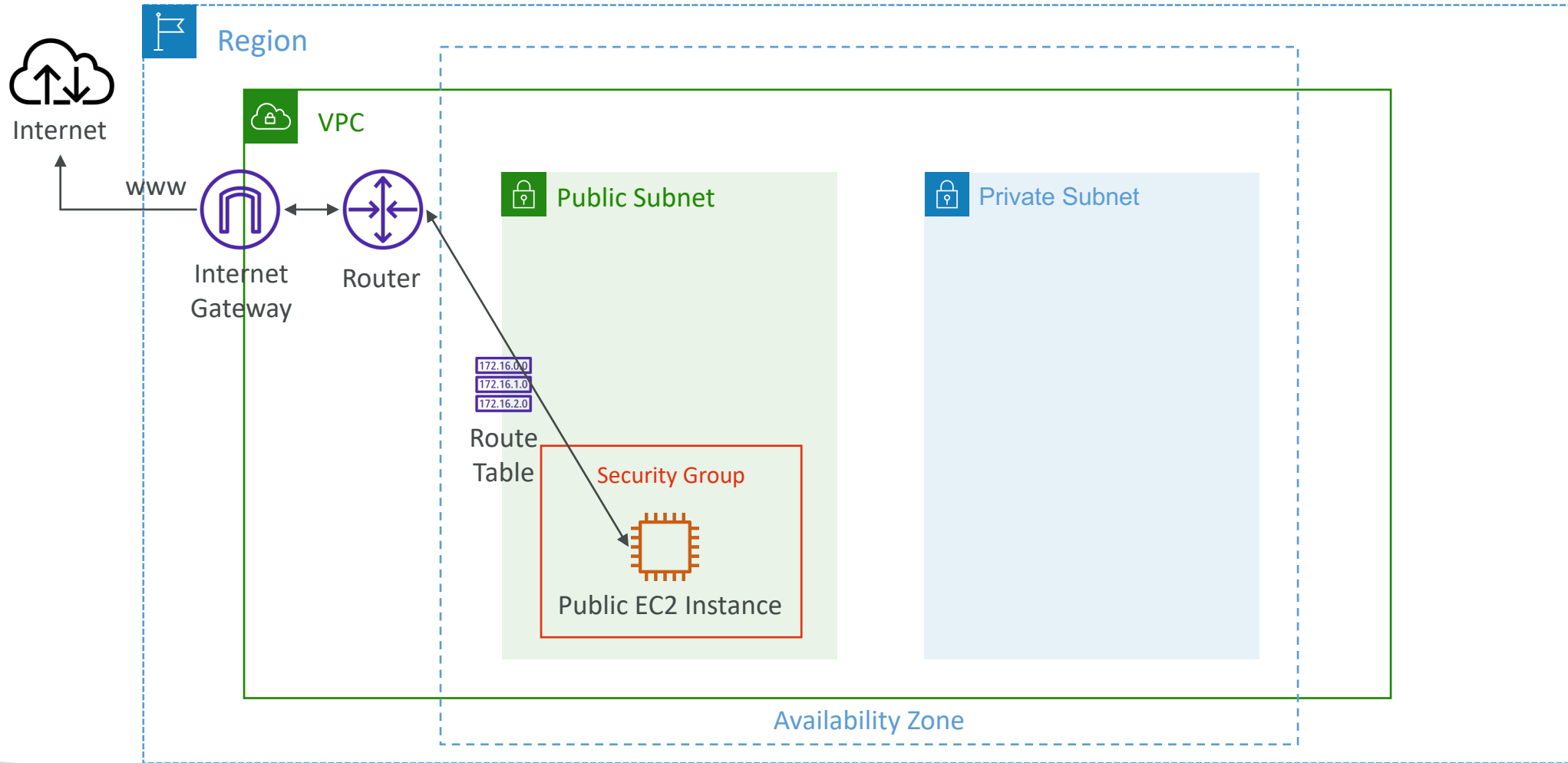
# Internet Gateway (IGW)

- Allows resources (e.g., EC2 instances) in a VPC connect to the Internet
- It scales horizontally and is highly available and redundant
- Must be created separately from a VPC
- One VPC can only be attached to one IGW and vice versa
  
- Internet Gateways on their own do not allow Internet access...
- Route tables must also be edited!

# Adding Internet Gateway

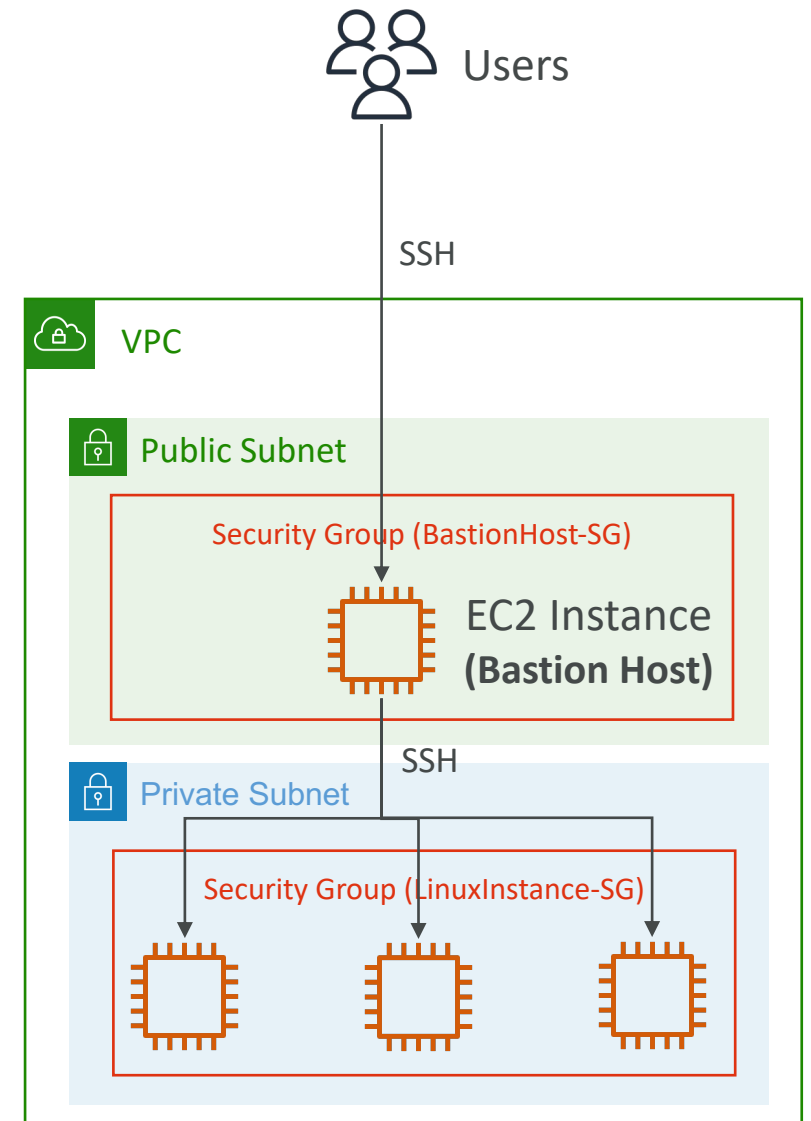


# Editing Route Tables



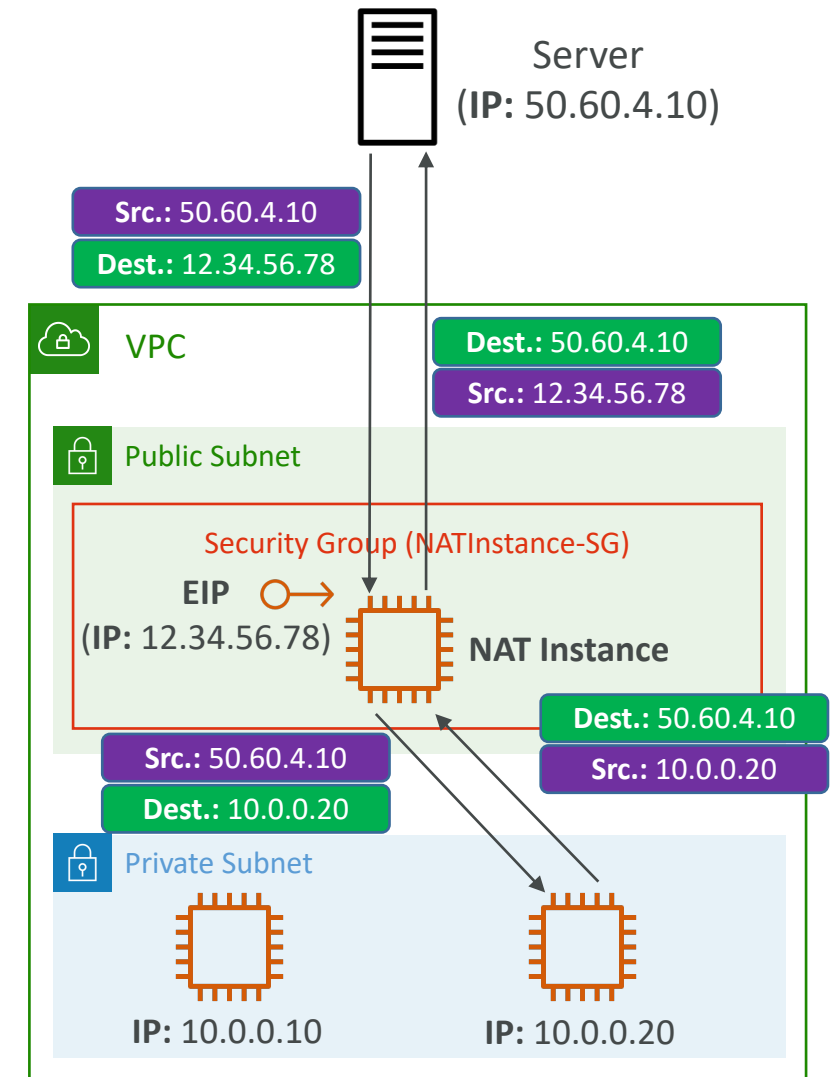
# Bastion Hosts

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

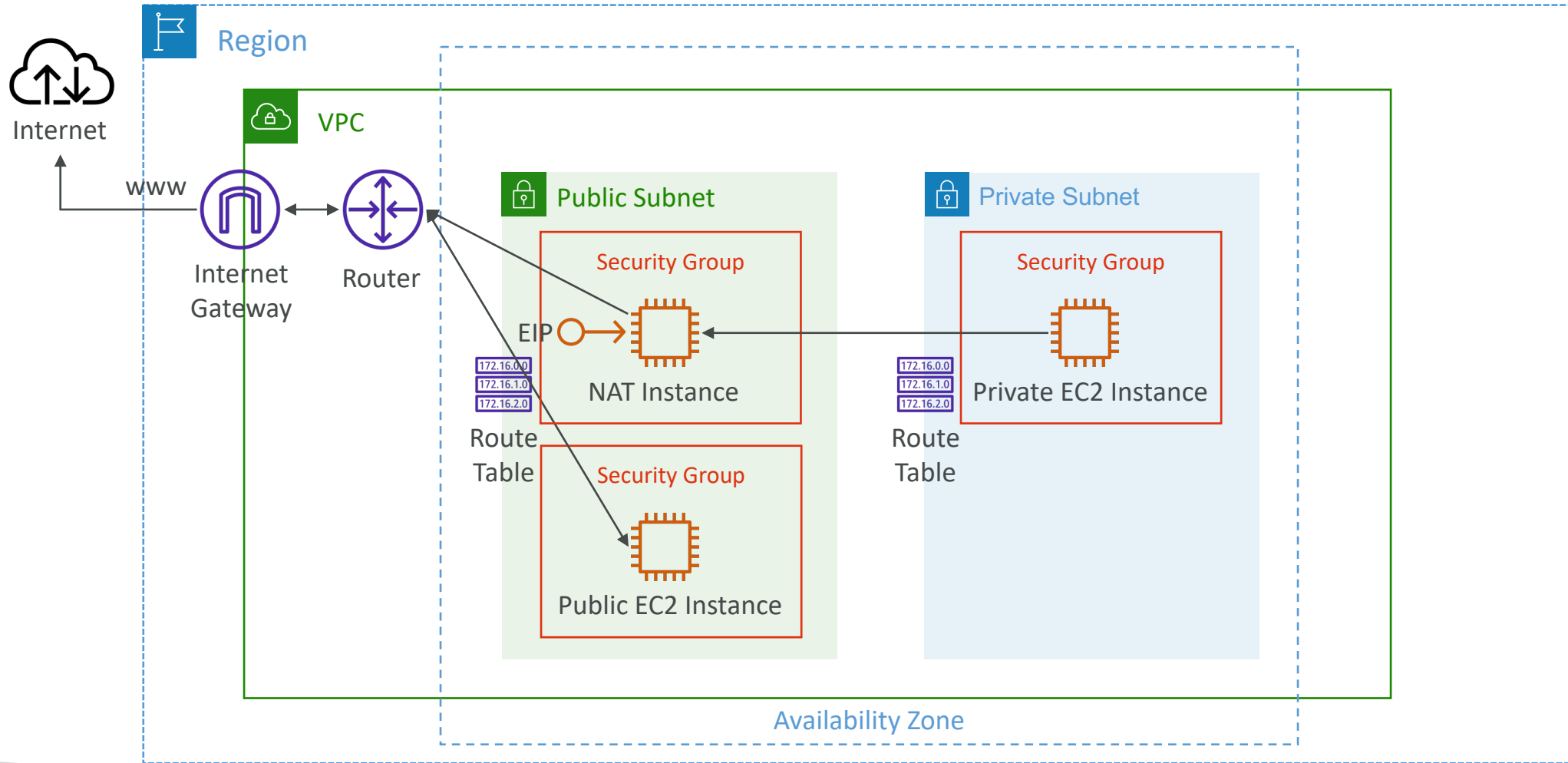


# NAT Instance (outdated, but still at the exam)

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



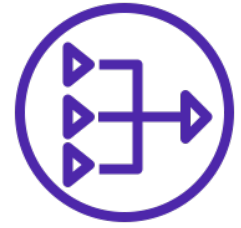
# NAT Instance



# NAT Instance – Comments

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

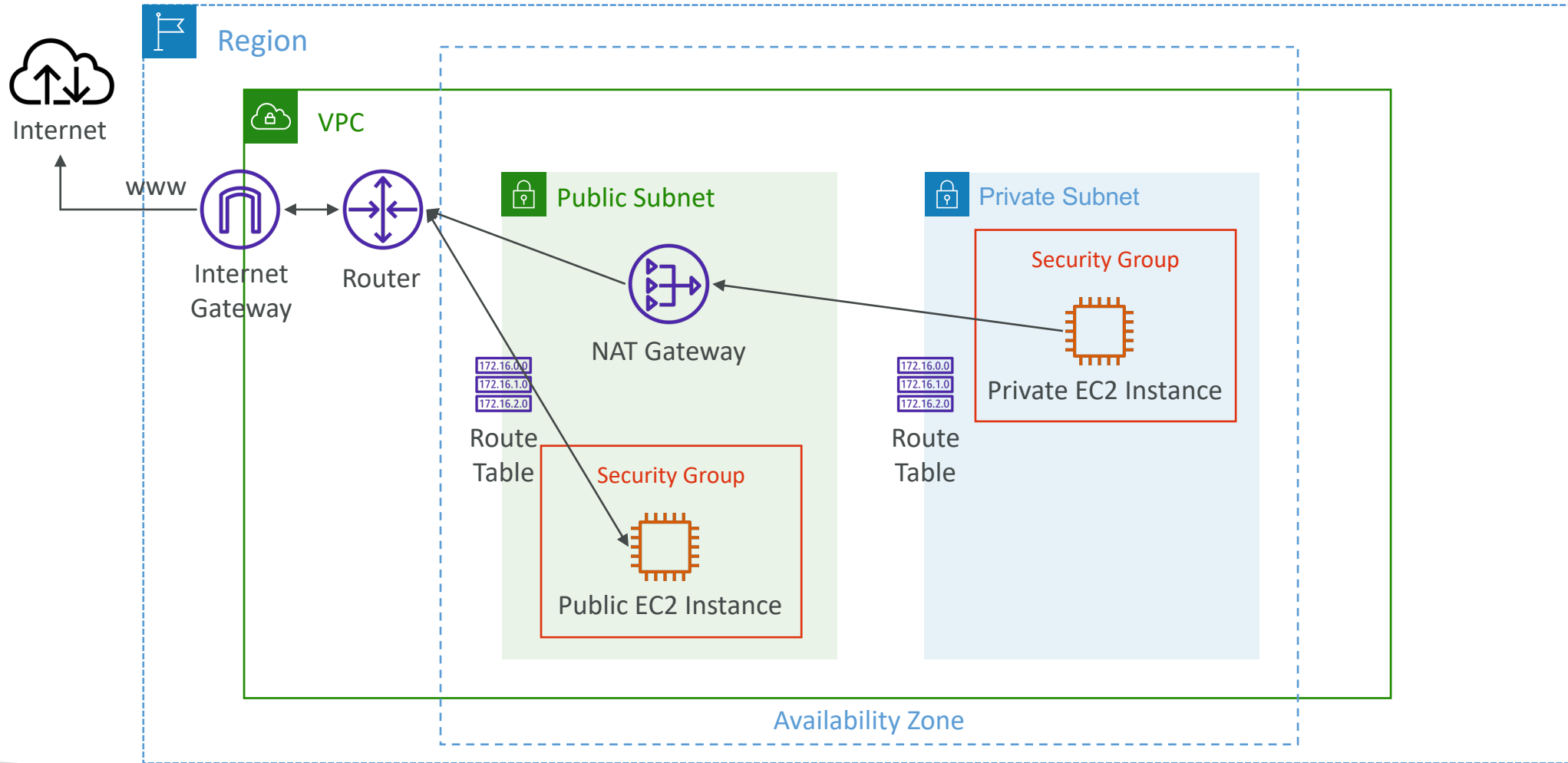
# NAT Gateway



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

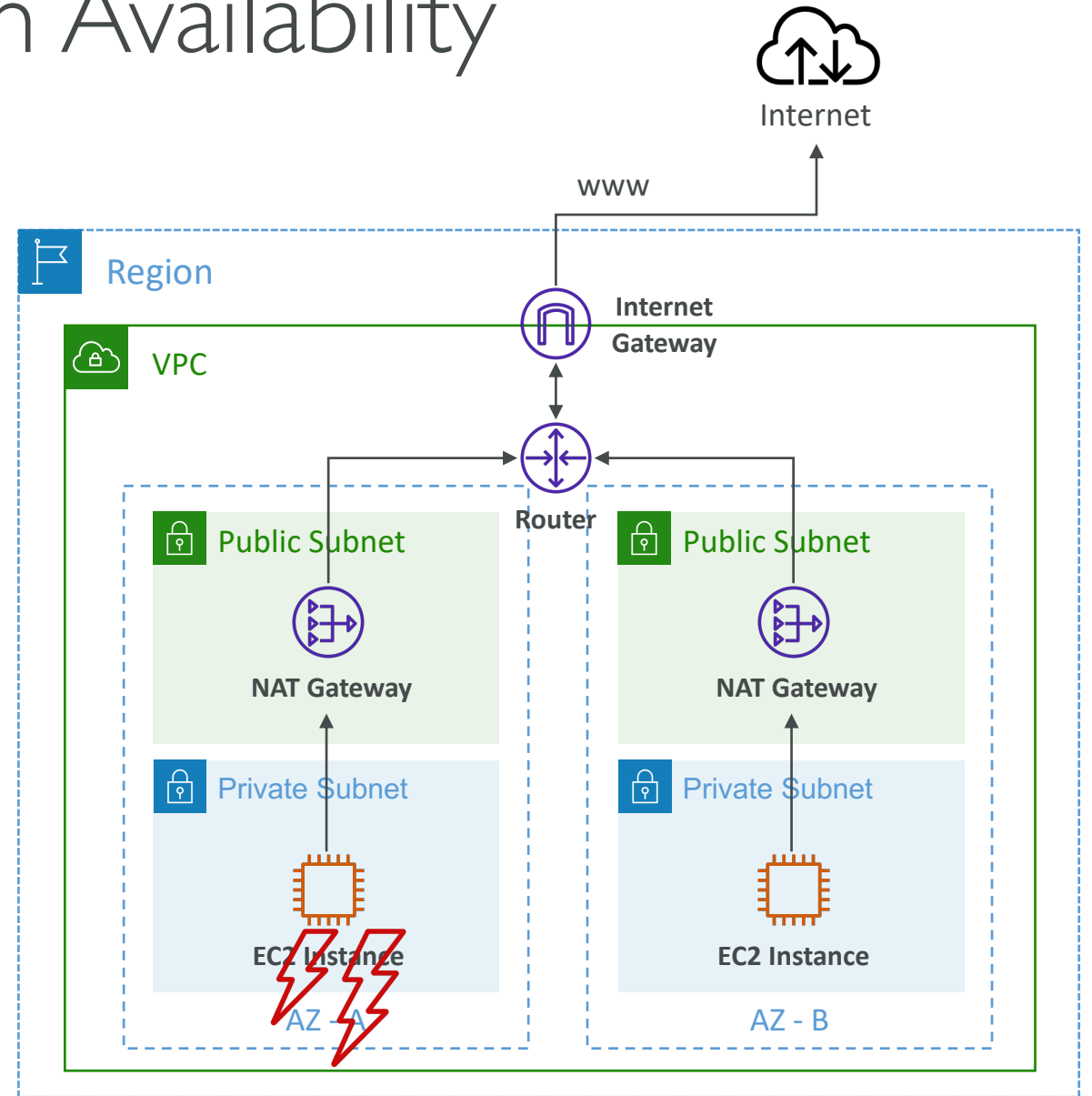


# NAT Gateway



# NAT Gateway with High Availability

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



# NAT Gateway vs. NAT Instance

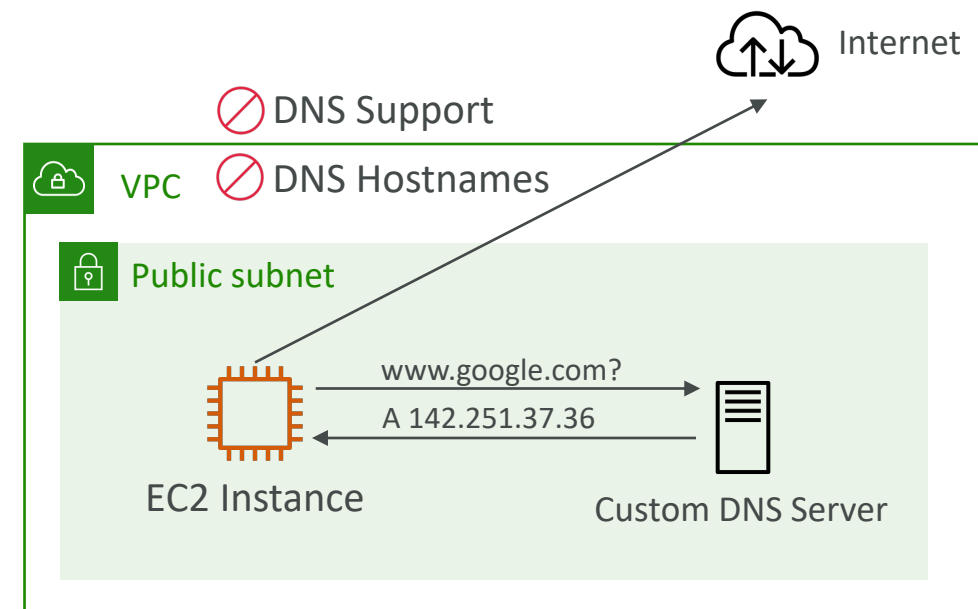
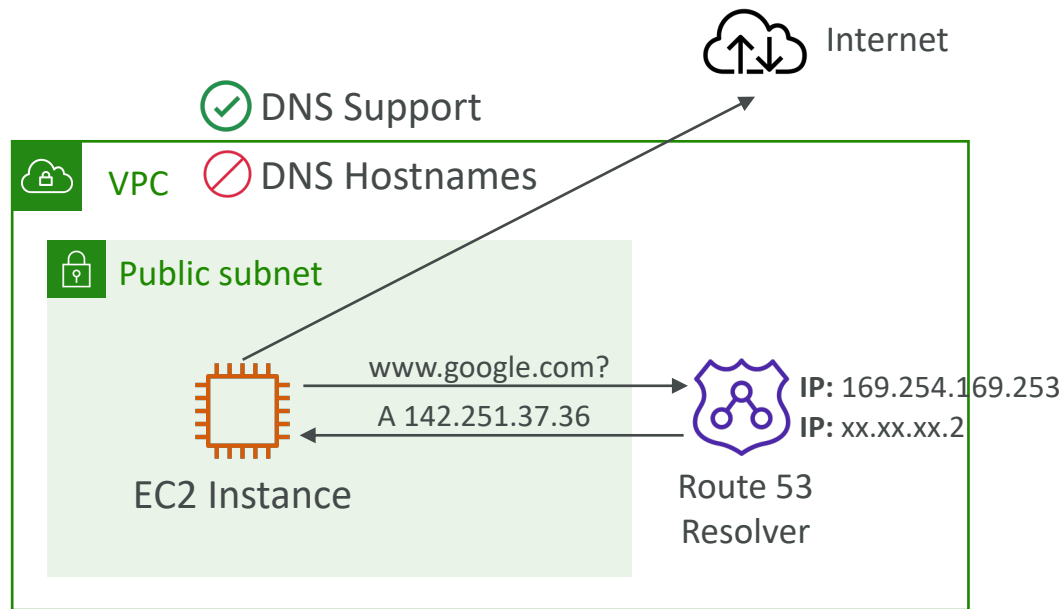
	NAT Gateway	NAT Instance
<b>Availability</b>	Highly available within AZ (create in another AZ)	Use a script to manage failover between instances
<b>Bandwidth</b>	Up to 45 Gbps	Depends on EC2 instance type
<b>Maintenance</b>	Managed by AWS	Managed by you (e.g., software, OS patches, ...)
<b>Cost</b>	Per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

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

# DNS Resolution in VPC

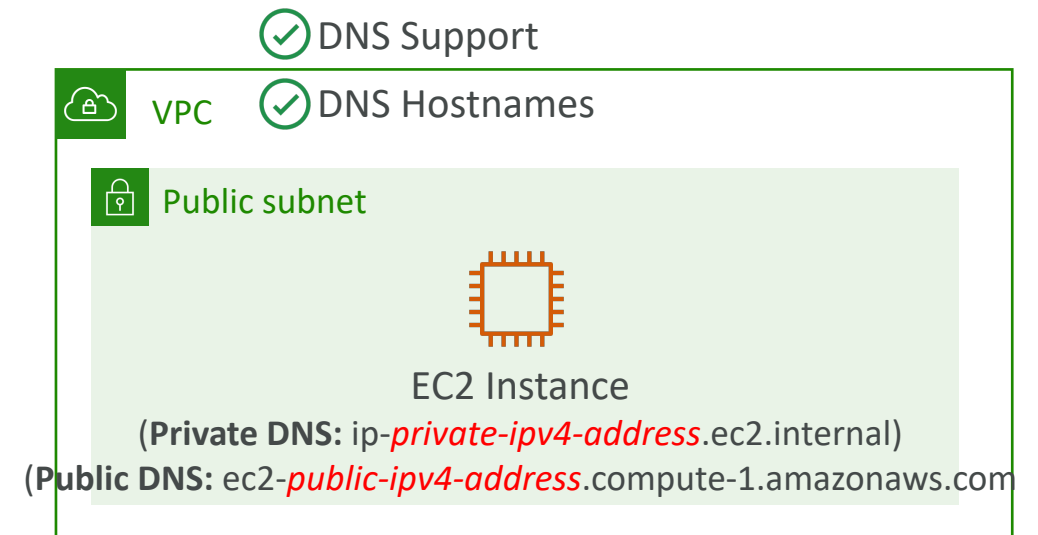
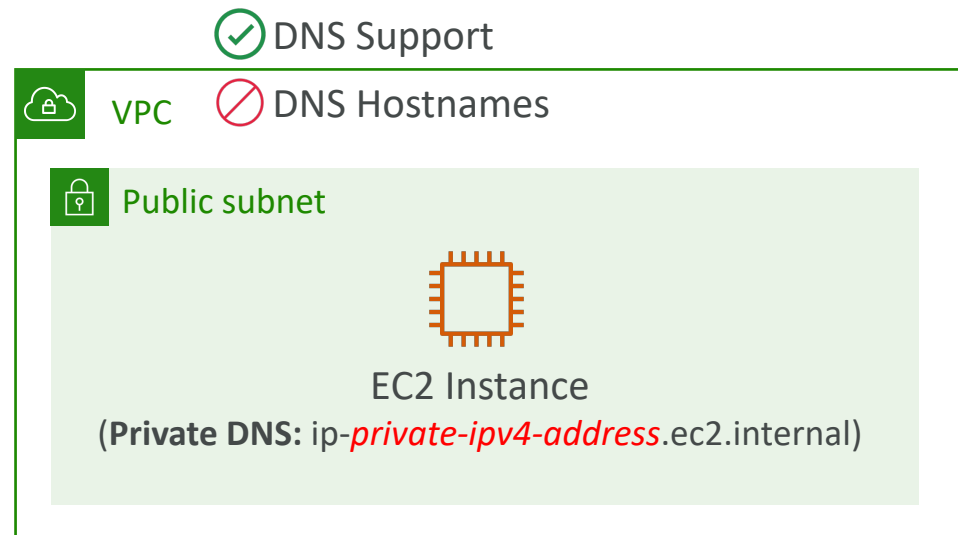
- DNS Resolution (enableDnsSupport)

- Decides if DNS resolution from Route 53 Resolver server is supported for the VPC
- True (default): it queries the Amazon Provider DNS Server at 169.254.169.253 or the reserved IP address at the base of the VPC IPv4 network range plus two (.2)



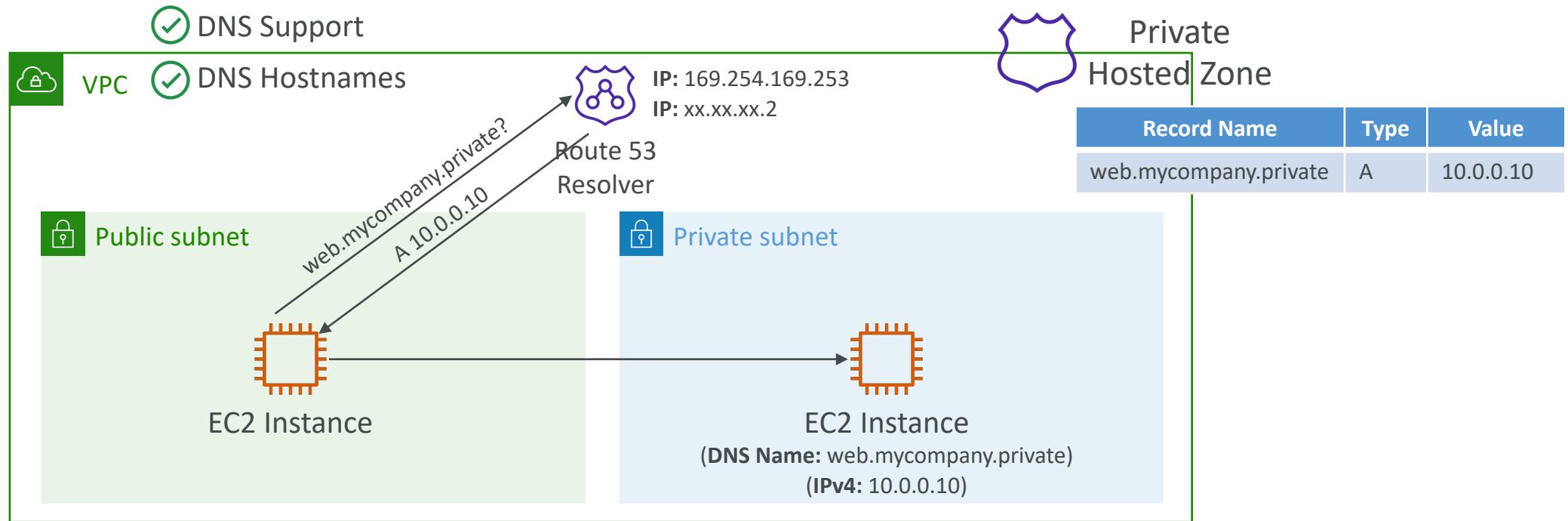
# DNS Resolution in VPC

- DNS Hostnames (`enableDnsHostnames`)
  - By default,
    - True => default VPC
    - False => newly created VPCs
  - Won't do anything unless `enableDnsSupport=true`
  - If True, assigns public hostname to EC2 instance if it has a public IPv4



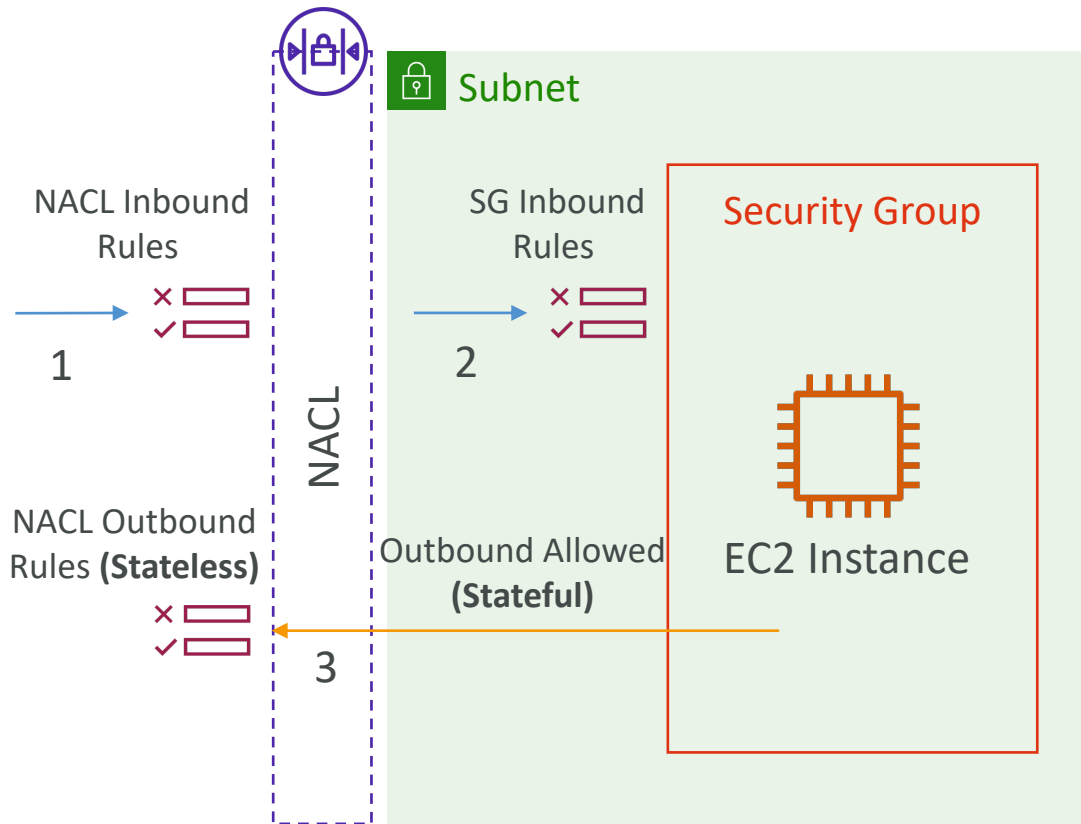
# DNS Resolution in VPC

- If you use custom DNS domain names in a Private Hosted Zone in Route 53, you must set both these attributes (`enableDnsSupport` & `enableDnsHostnames`) to true

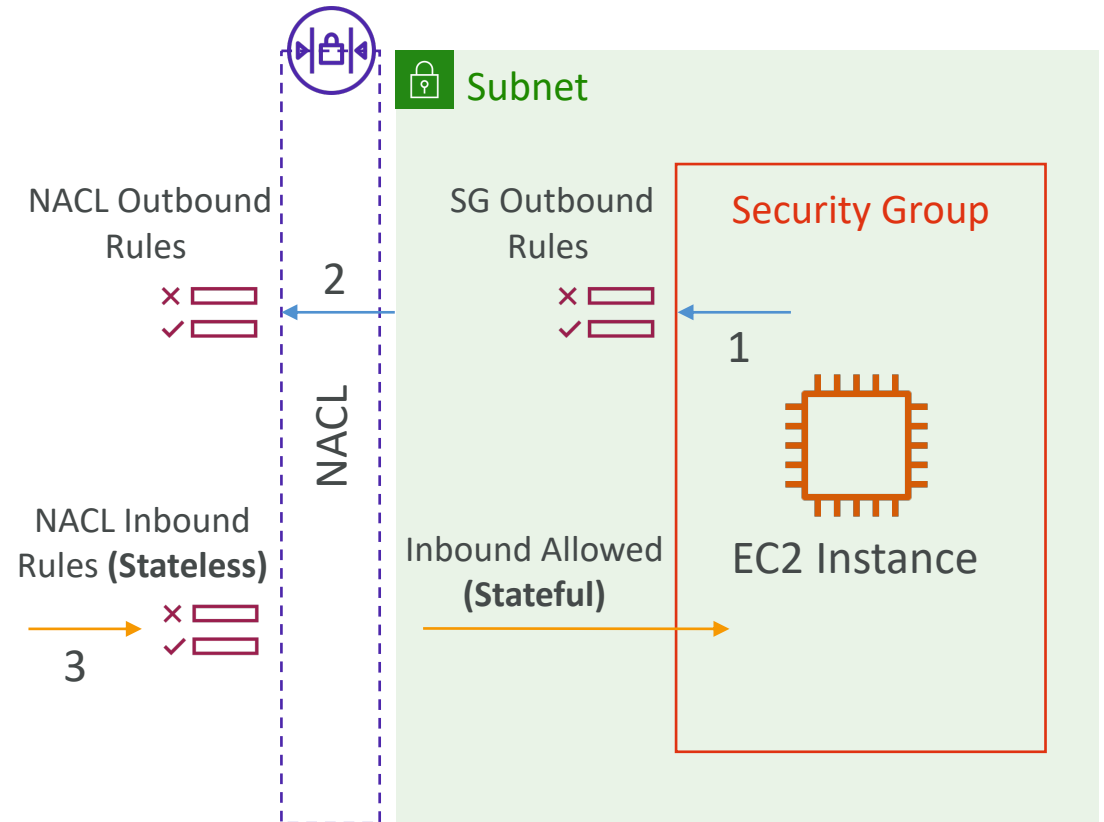


# Security Groups & NACLs

## Incoming Request



## Outgoing Request



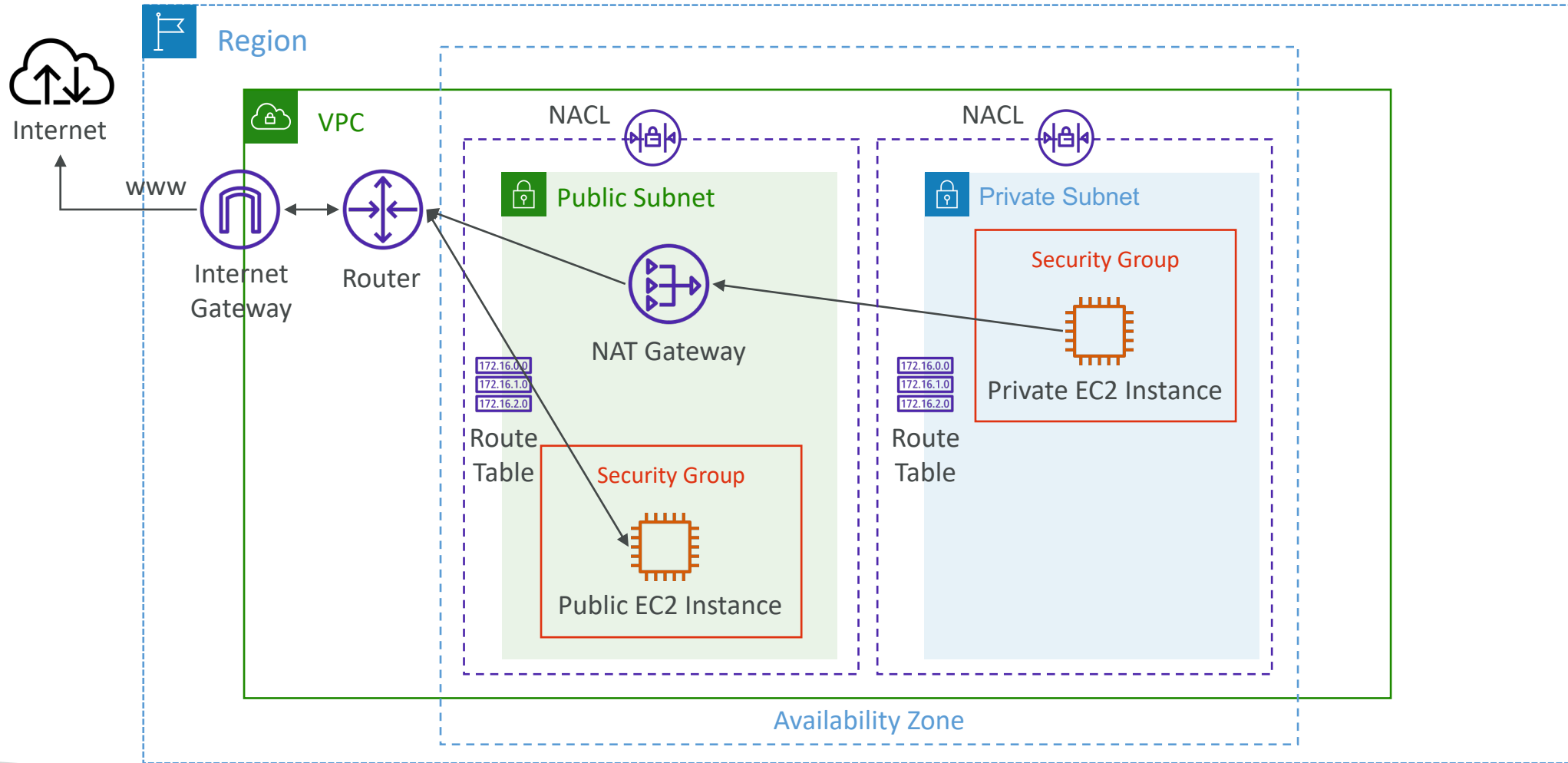
# Network Access Control List (NACL)



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



# NACLs



# Default NACL

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



Default NACL for a VPC that supports IPv4

## Inbound Rules

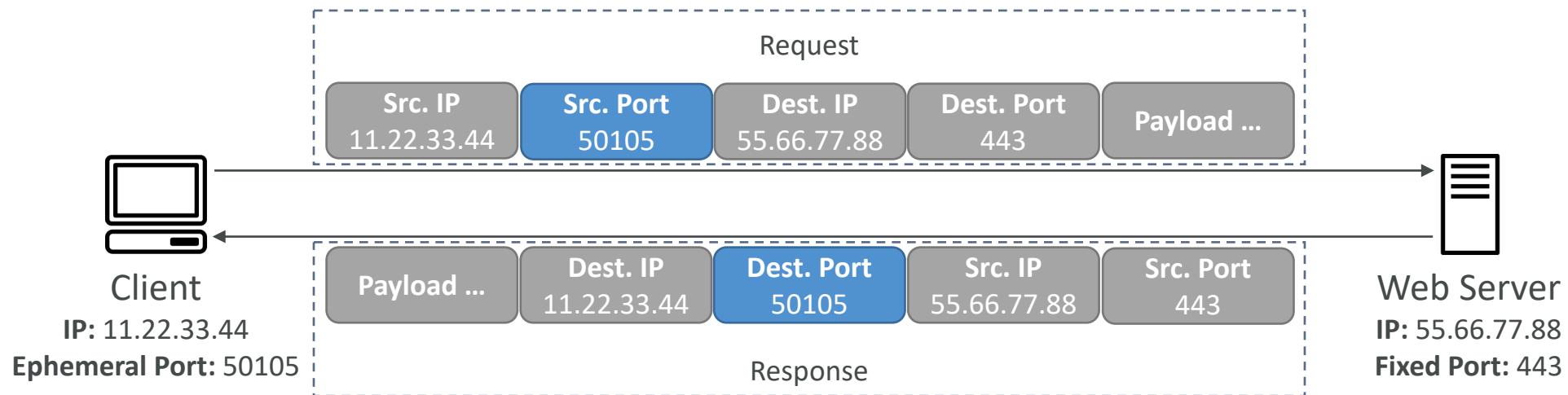
Rule #	Type	Protocol	Port Range	Source	Allow/Deny
100	All IPv4 Traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 Traffic	All	All	0.0.0.0/0	DENY

## Outbound Rules

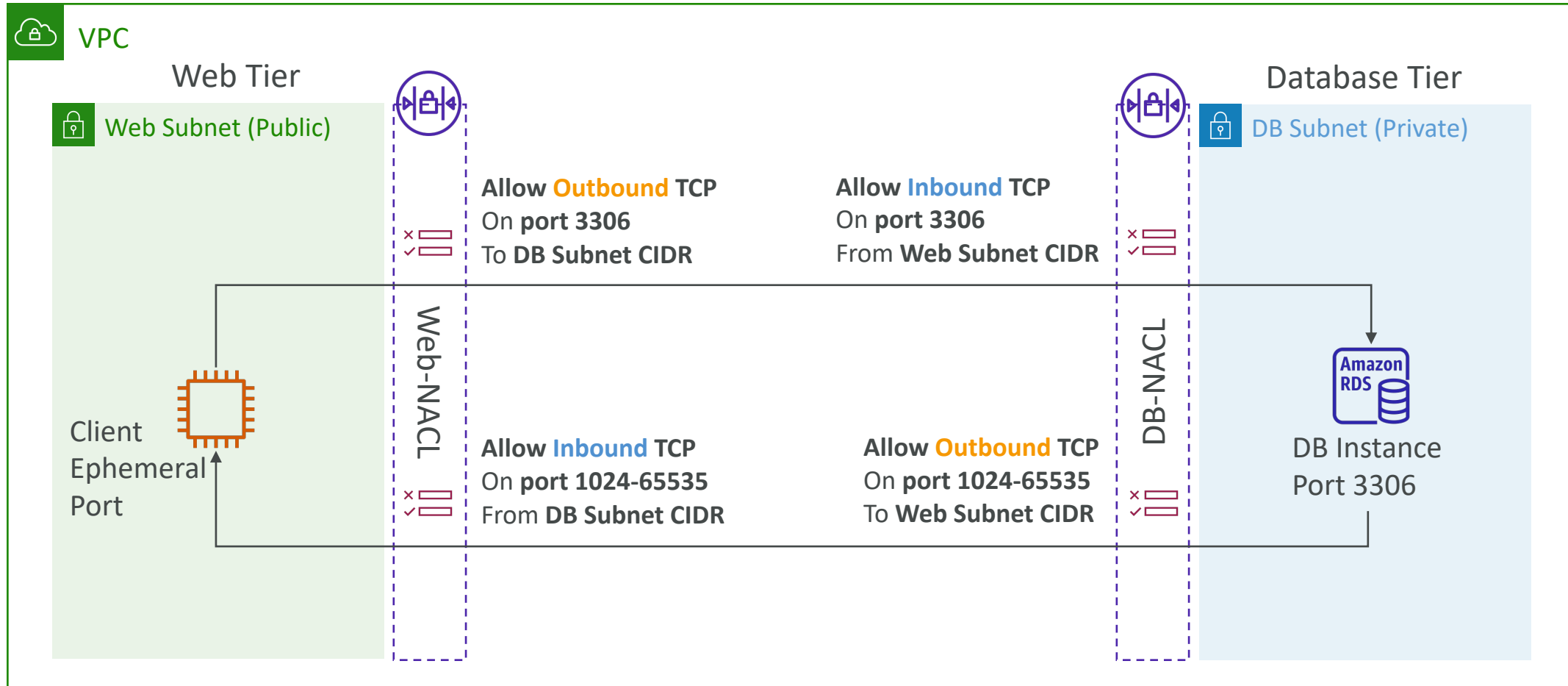
Rule #	Type	Protocol	Port Range	Destination	Allow/Deny
100	All IPv4 Traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 Traffic	All	All	0.0.0.0/0	DENY

# Ephemeral Ports

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

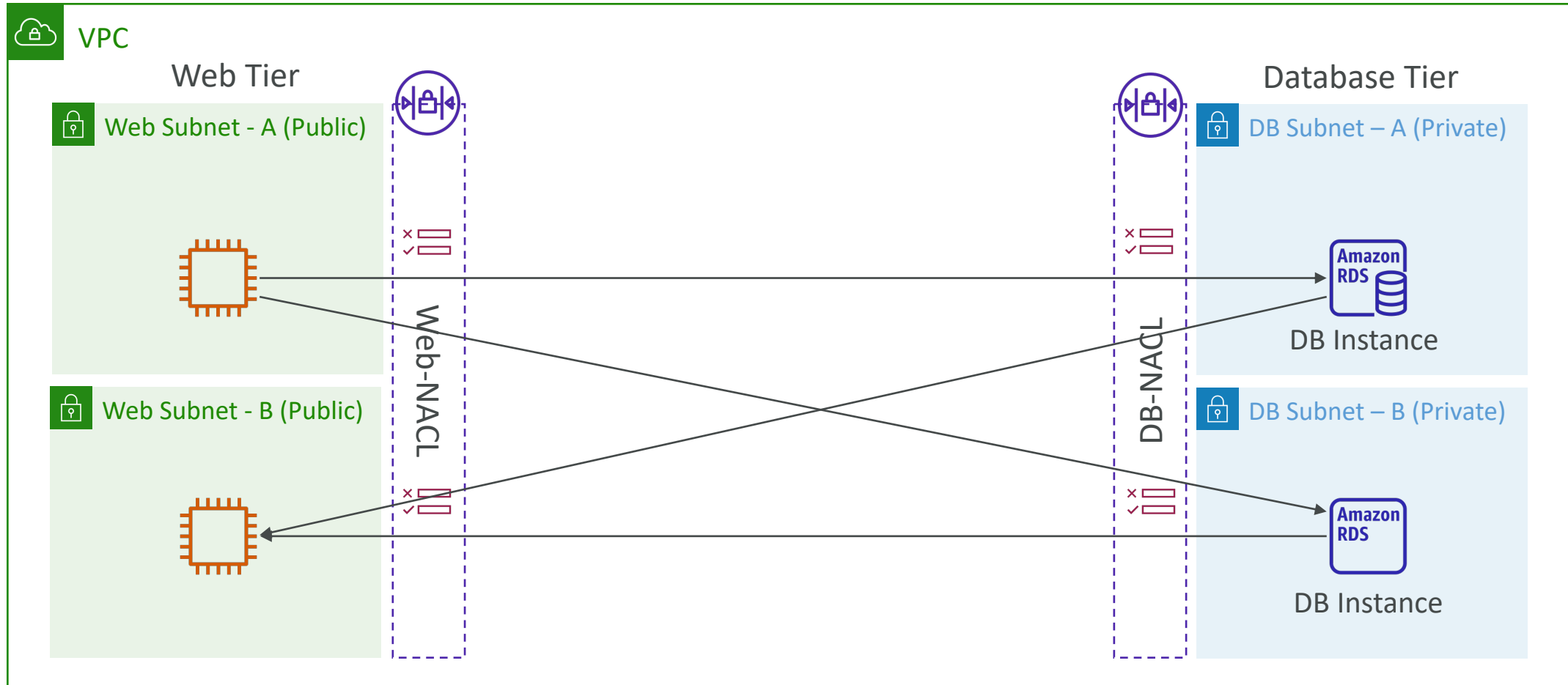


# NACL with Ephemeral Ports



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

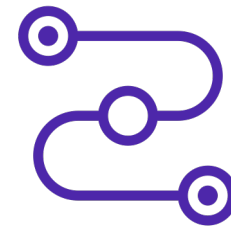
# Create NACL rules for each target subnets CIDR



# Security Group vs. NACLs

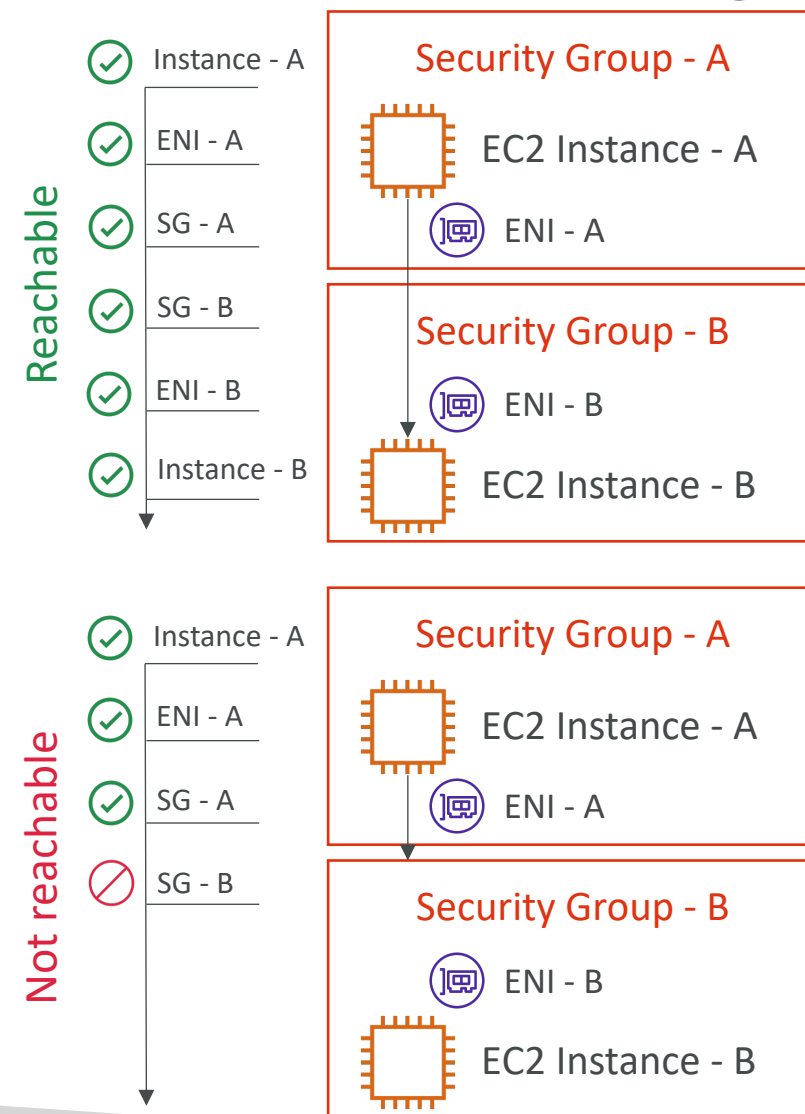
Security Group	NACL
Operates at the instance level	Operates at the subnet level
Supports allow rules only	Supports allow rules and deny rules
<b>Stateful:</b> return traffic is automatically allowed, regardless of any rules	<b>Stateless:</b> return traffic must be explicitly allowed by rules (think of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

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



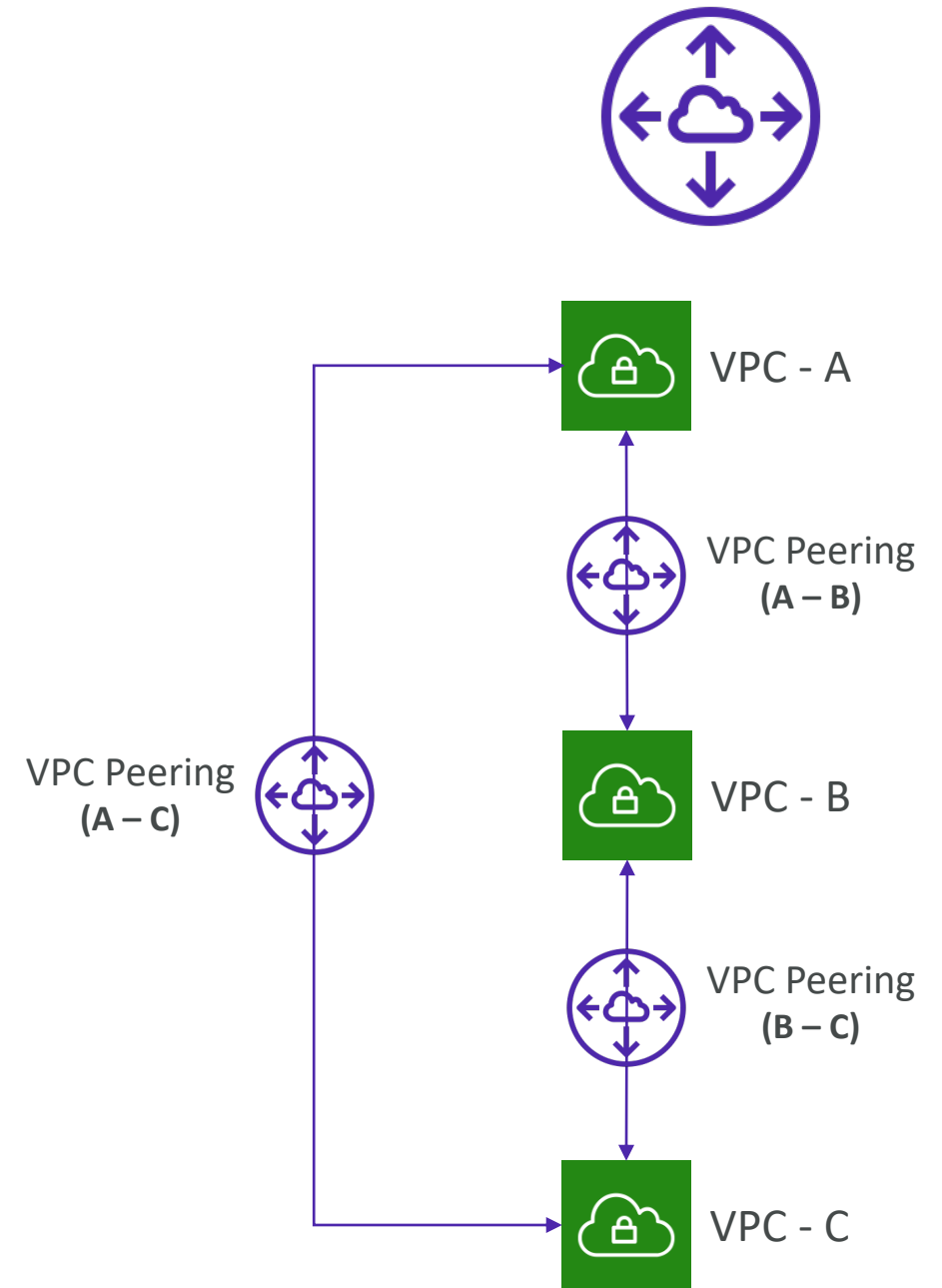
# VPC – Reachability Analyzer

- A network diagnostics tool that troubleshoots network connectivity between two endpoints in your VPC(s)
- It builds a model of the network configuration, then checks the reachability based on these configurations (**it doesn't send packets**)
- When the destination is
  - **Reachable** – it produces hop-by-hop details of the virtual network path
  - **Not reachable** – it identifies the blocking component(s) (e.g., configuration issues in SGs, NACLs, Route Tables, ...)
- Use cases: troubleshoot connectivity issues, ensure network configuration is as intended, ...



# VPC Peering

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





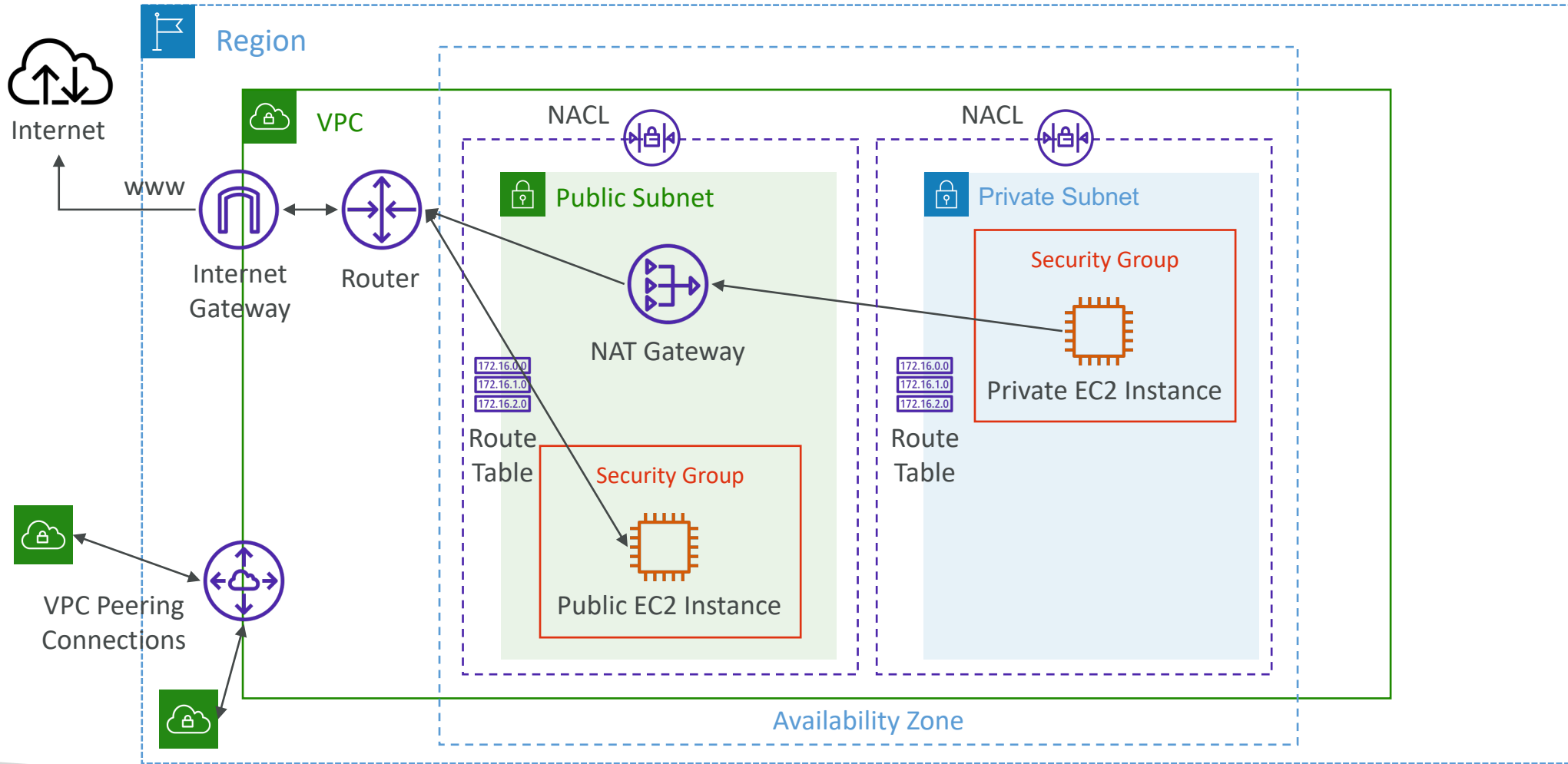
# VPC Peering – Good to know

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

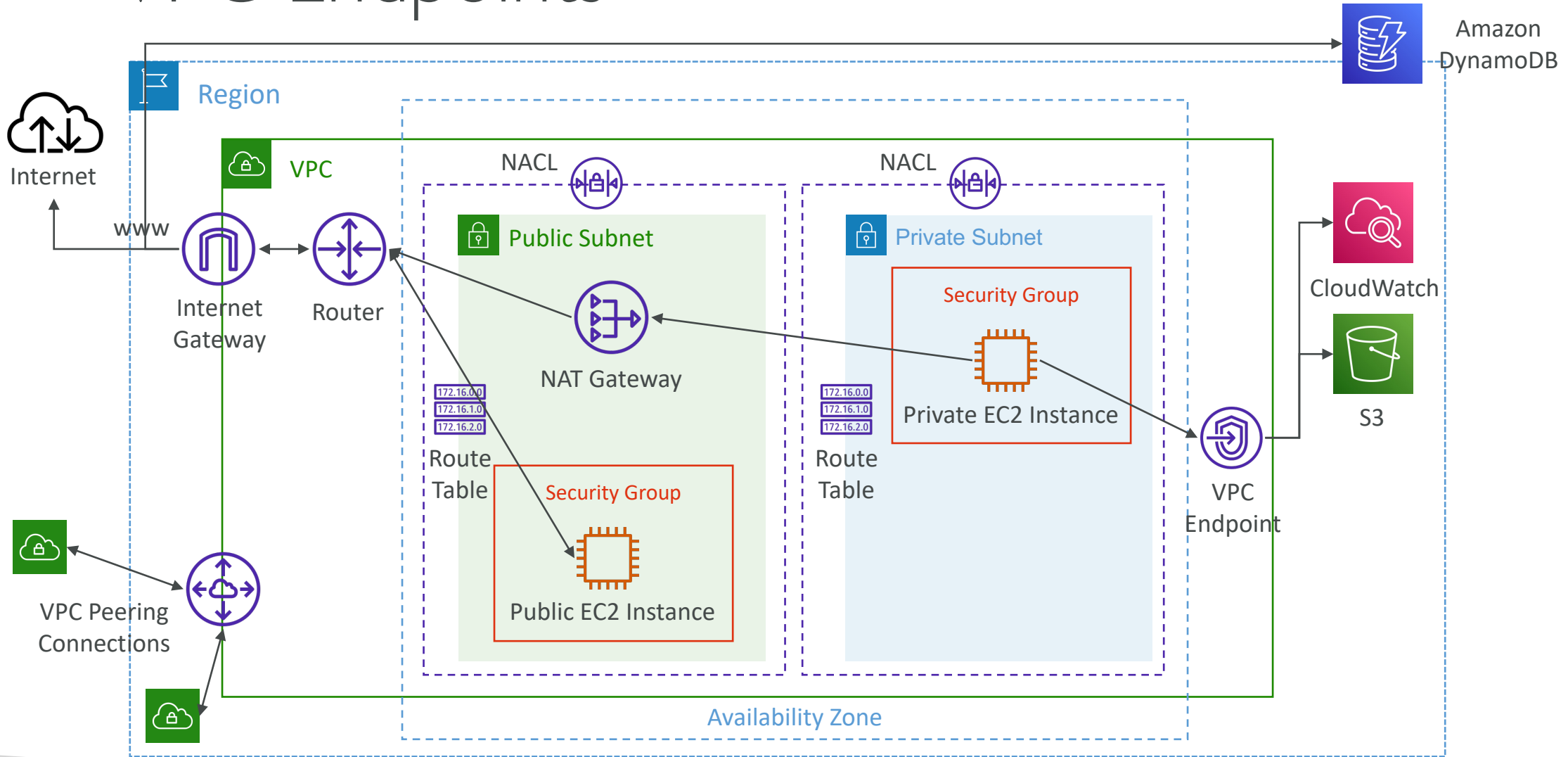
Type ▾	Protocol ▾	Port range ▾	Source ▾
HTTP	TCP	80	sg-04991f9af3473b939 / default
HTTP	TCP	80	<span style="background-color: red; color: black;">XXXXXXXXXX</span> / sg-027ad1f7865d4be76

↑  
Account ID

# VPC Peering



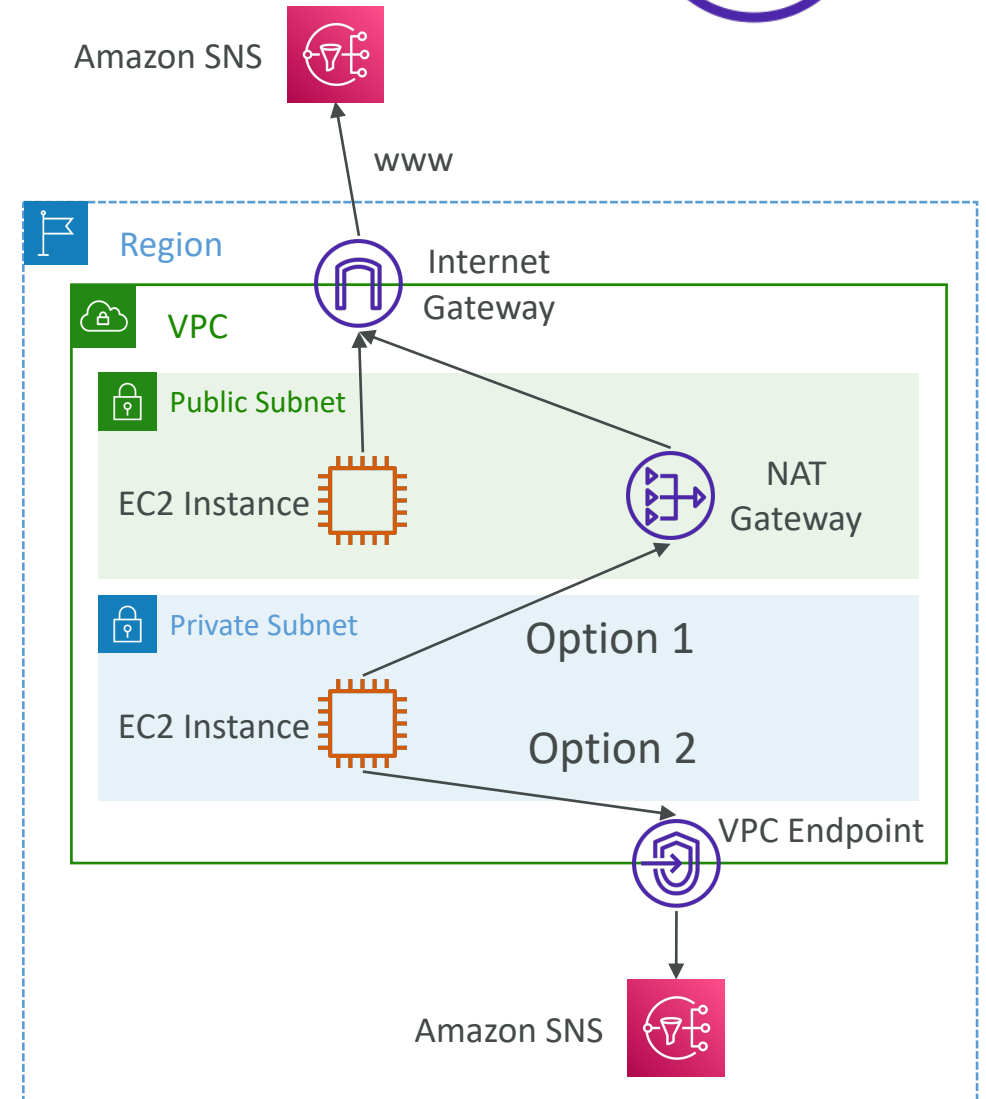
# VPC Endpoints



# VPC Endpoints (AWS PrivateLink)



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



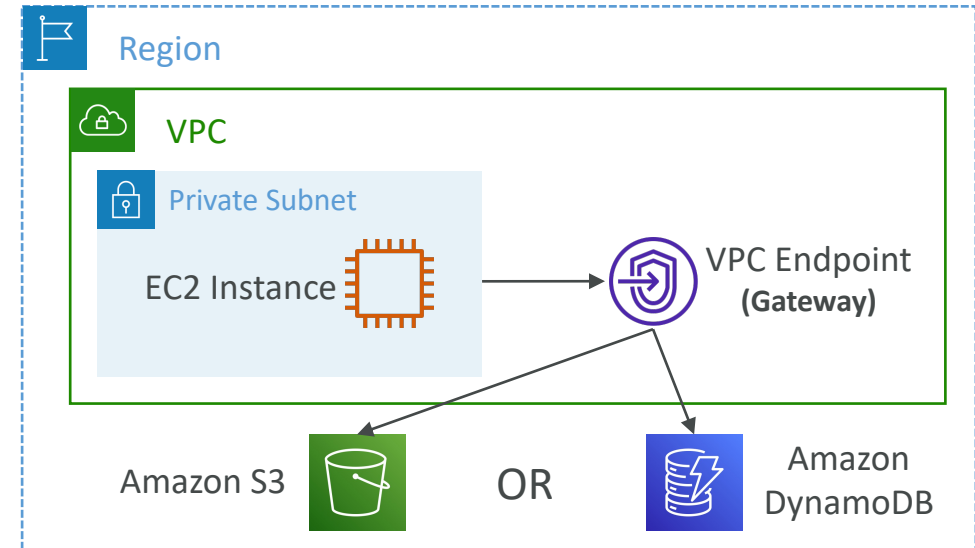
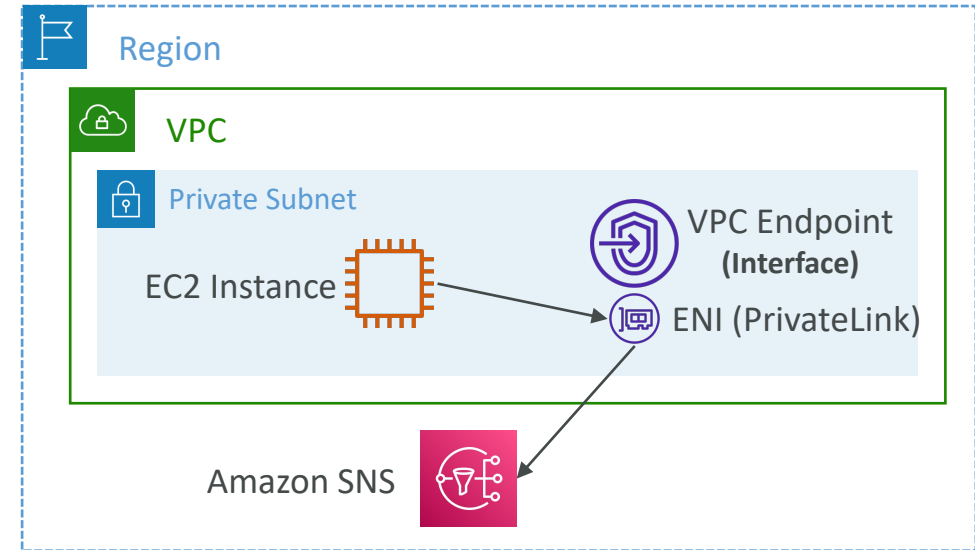
# Types of Endpoints

- Interface Endpoints (powered by PrivateLink)

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

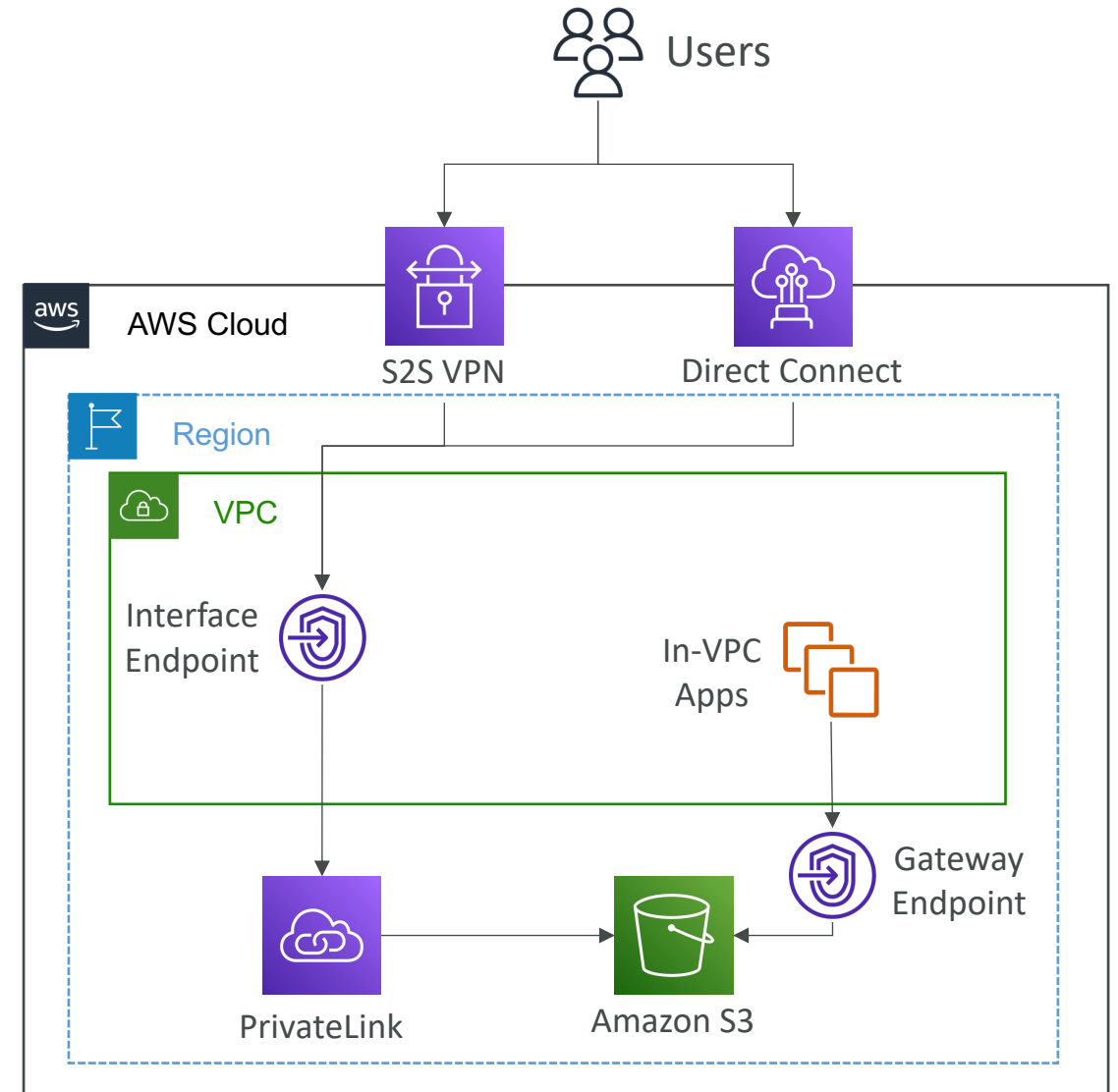
- Gateway Endpoints

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



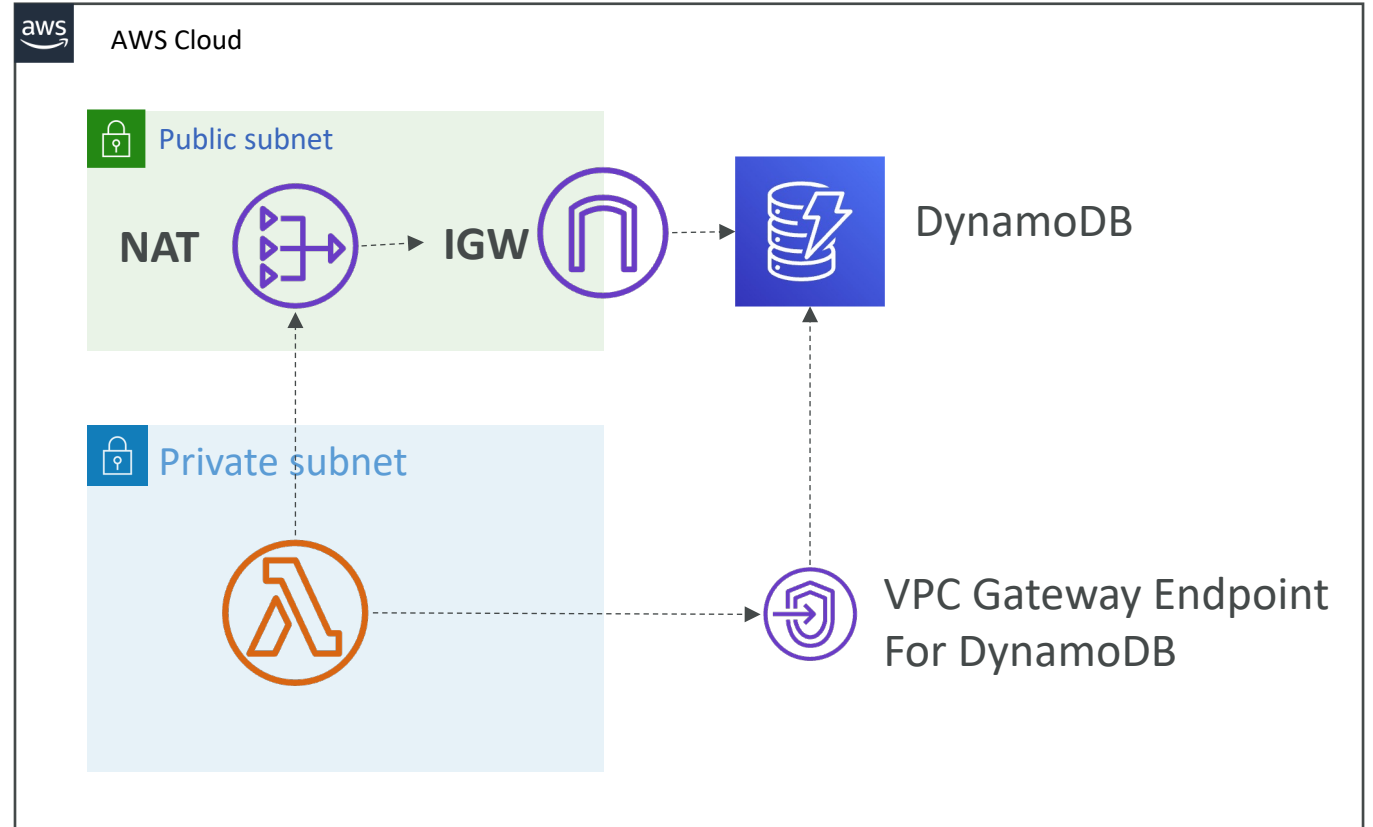
# Gateway or Interface Endpoint for S3?

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



# Lambda in VPC accessing DynamoDB

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



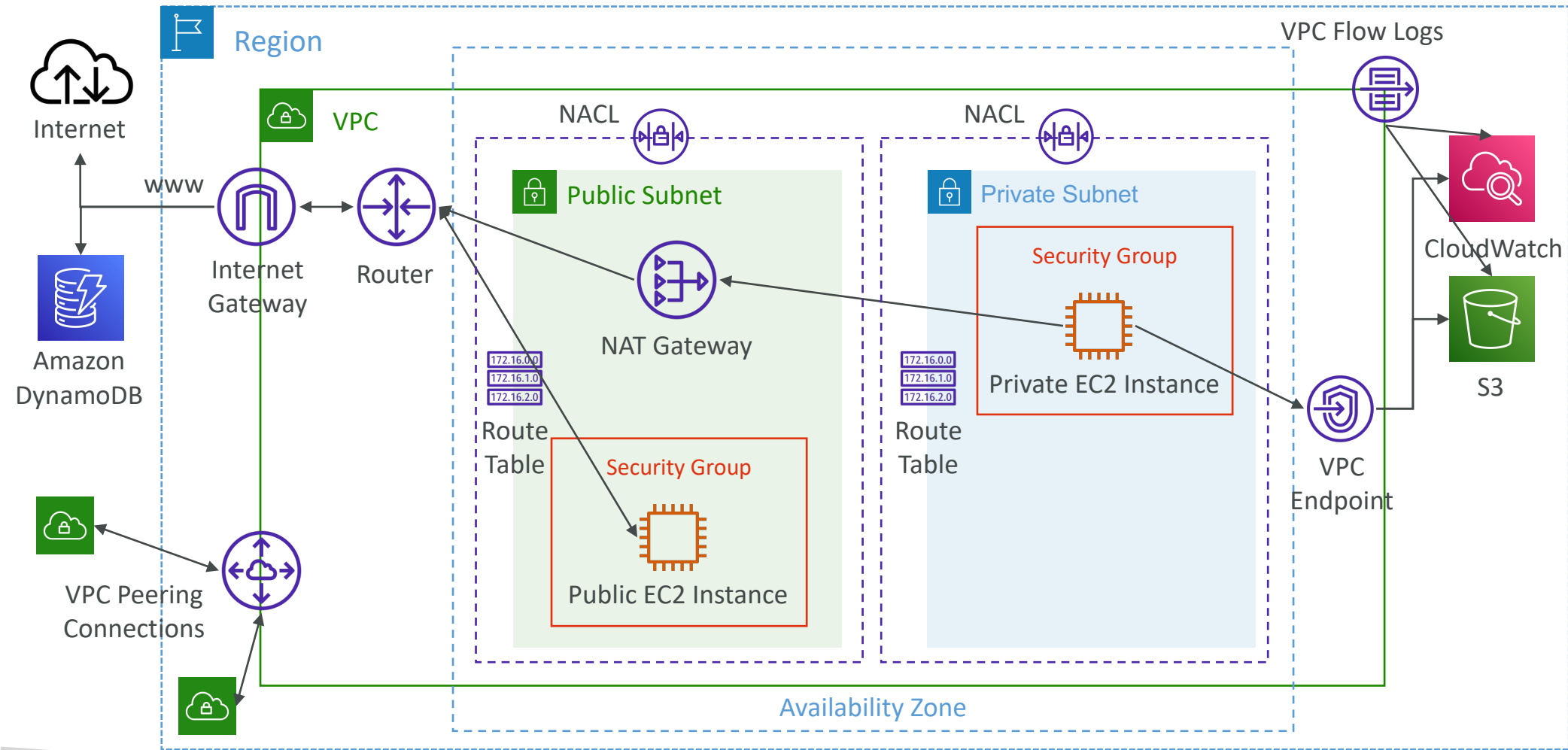


# VPC Flow Logs

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



# VPC Flow Logs



# VPC Flow Logs Syntax

version	account-id	interface-id	srcaddr	dstaddr	srcport	dstport	protocol	bytes	start	end	action	log-status	
2	123456789010	eni-1235b8ca123456789	172.31.16.139	172.31.16.21	20641	22	6	20	4249	1418530010	1418530070	ACCEPT	OK
2	123456789010	eni-1235b8ca123456789	172.31.9.69	172.31.9.12	49761	3389	6	20	4249	1418530010	1418530070	REJECT	OK

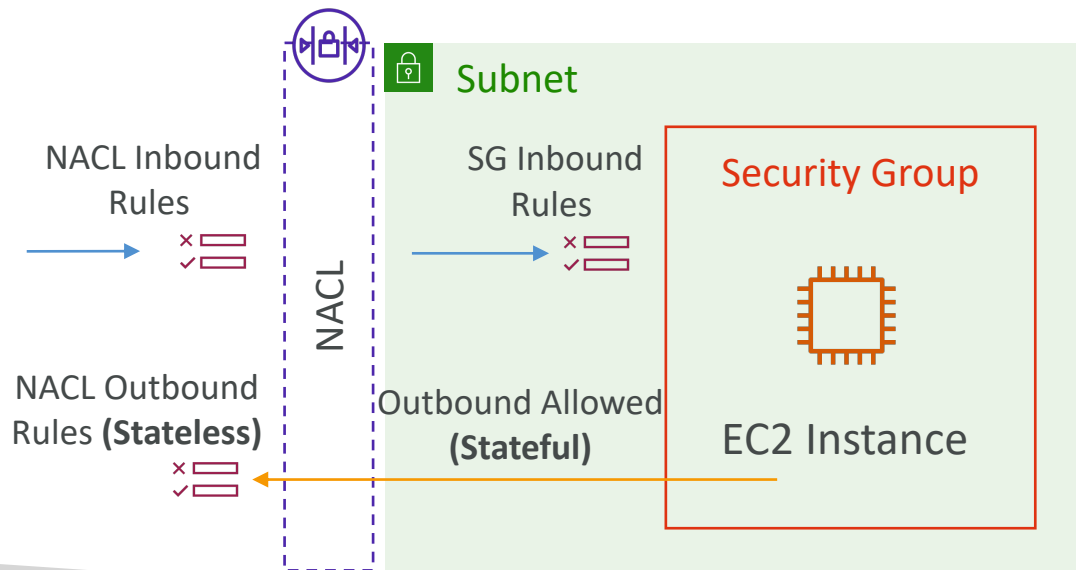
- srcaddr & dstaddr – help identify problematic IP
- srcport & dstport – help identify problematic ports
- **Action** – success or failure of the request due to Security Group / NACL
- Can be used for analytics on usage patterns, or malicious behavior
- Query VPC flow logs using Athena on S3 or CloudWatch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

# VPC Flow Logs – Troubleshoot SG & NACL issues

Look at the “ACTION” field

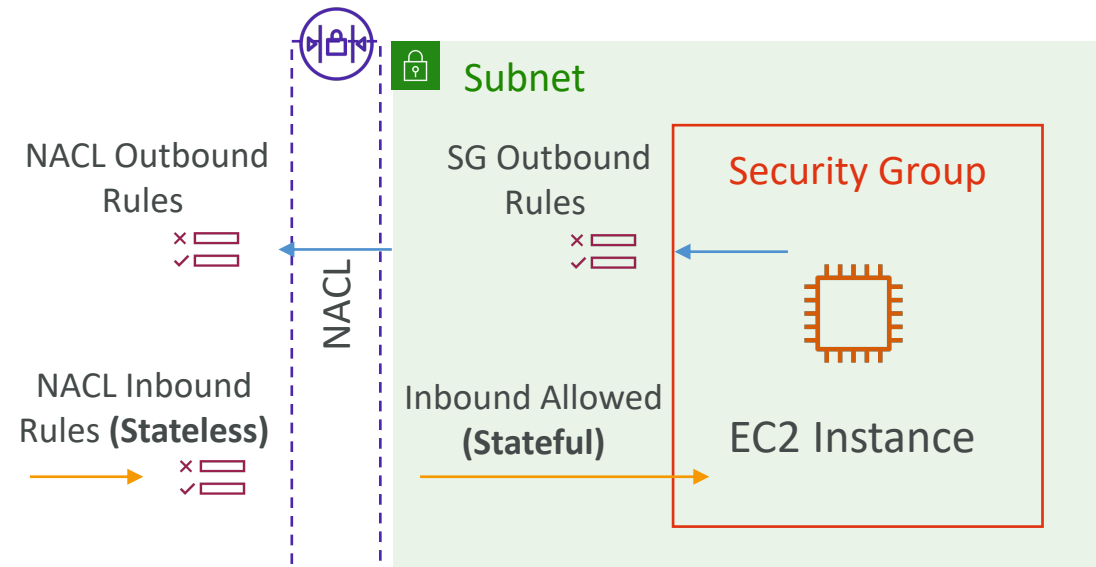
## Incoming Requests

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

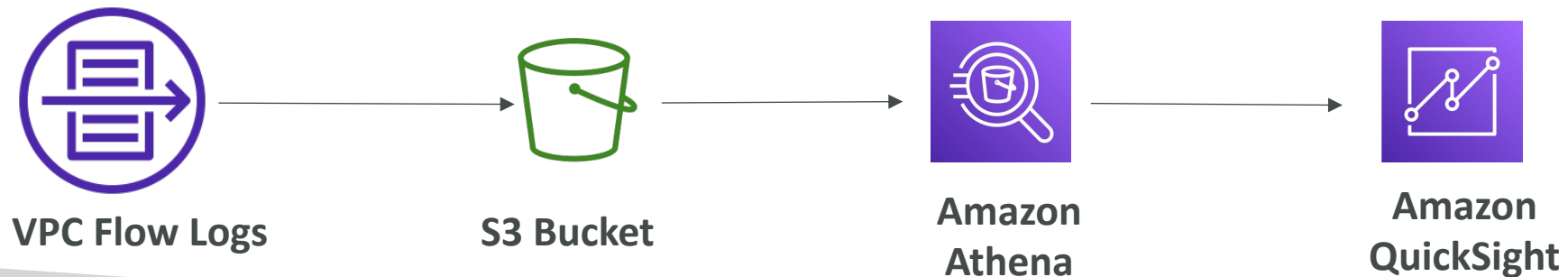
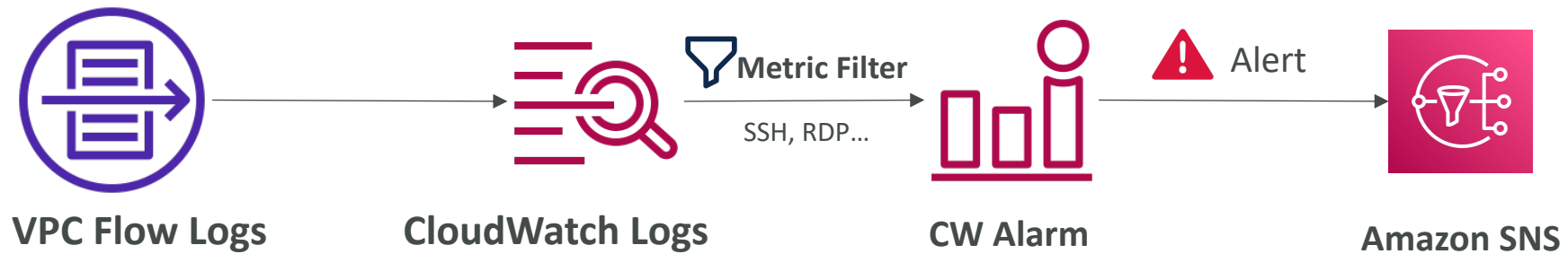
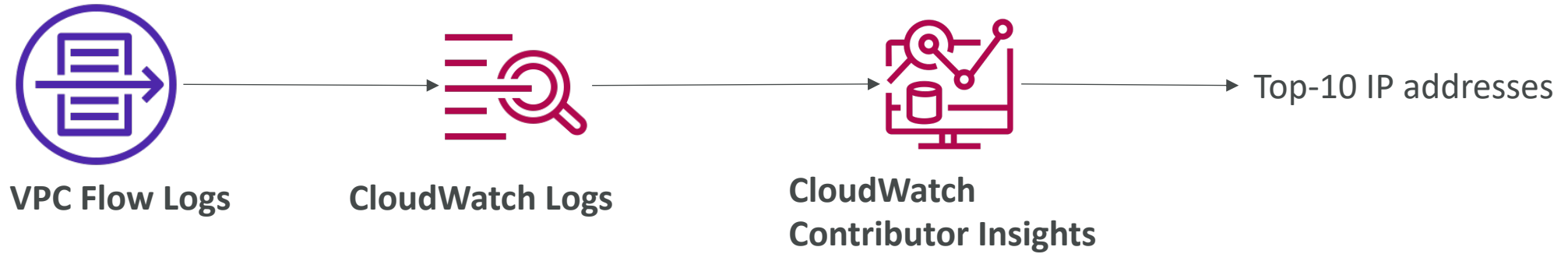


## Outgoing Requests

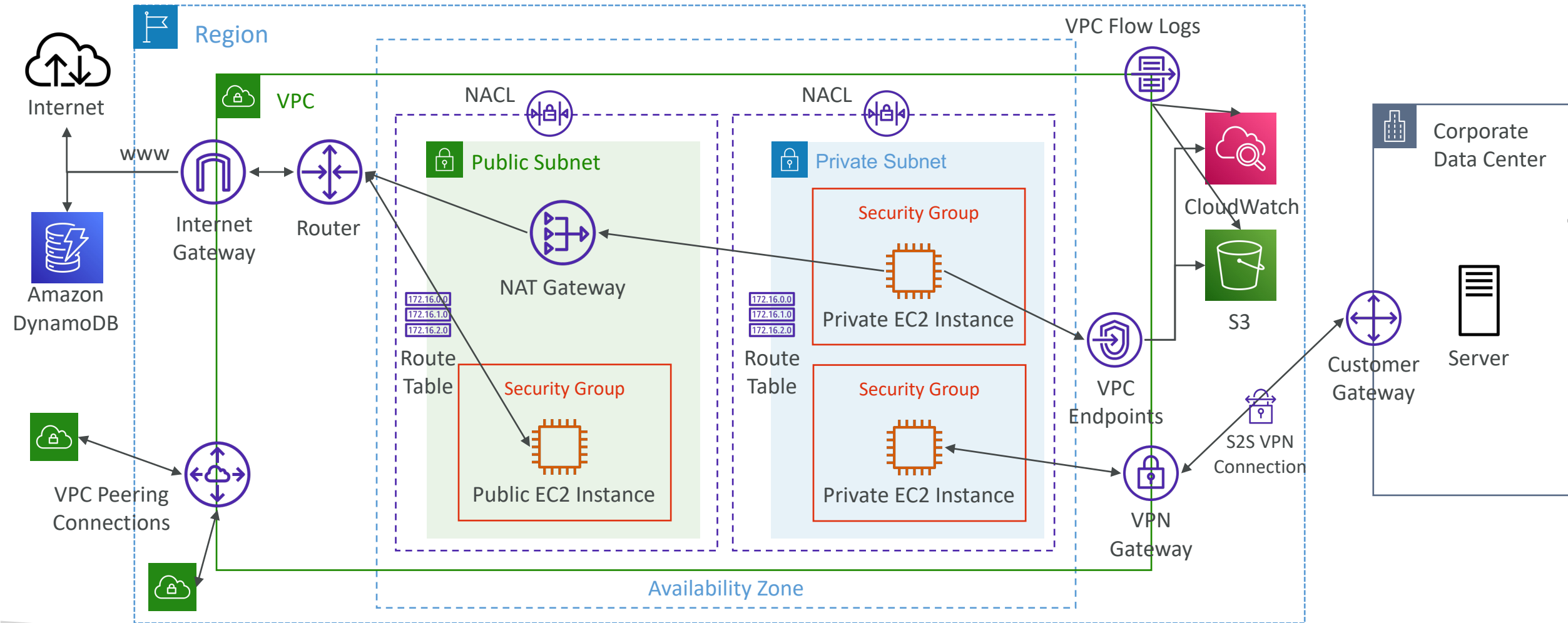
- Outbound REJECT => NACL or SG
- Outbound ACCEPT, Inbound REJECT => NACL



# VPC Flow Logs – Architectures



# AWS Site-to-Site VPN



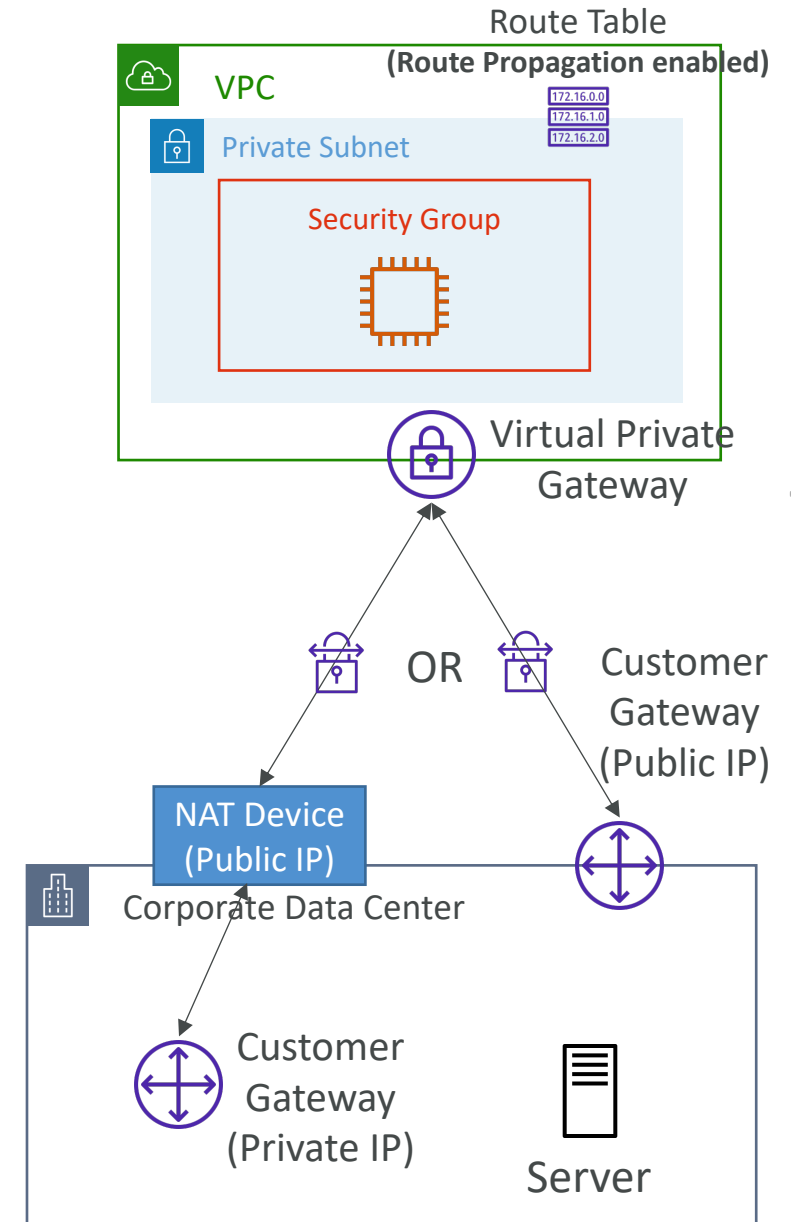


# AWS Site-to-Site VPN

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

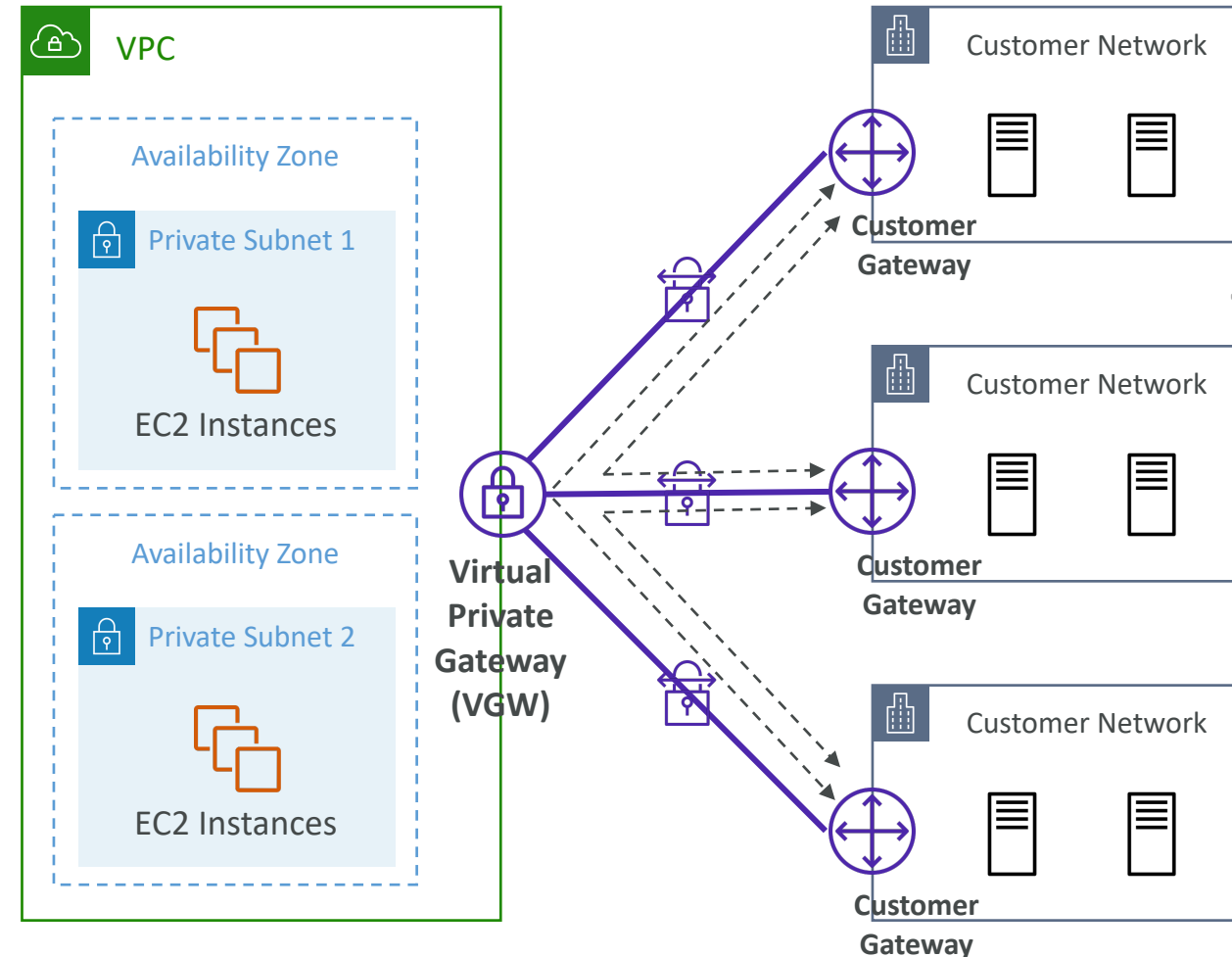
# Site-to-Site VPN Connections

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



# AWS VPN CloudHub

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



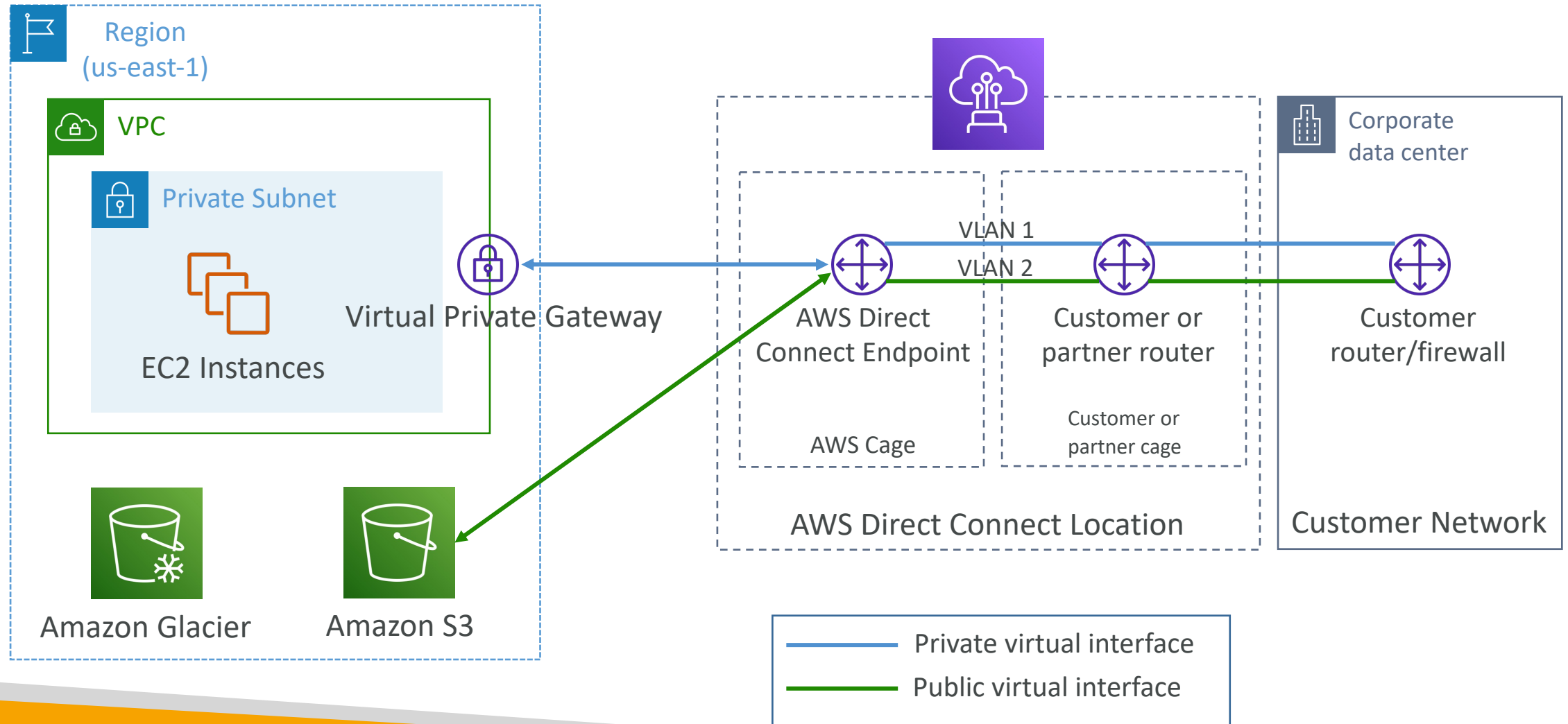


# Direct Connect (DX)



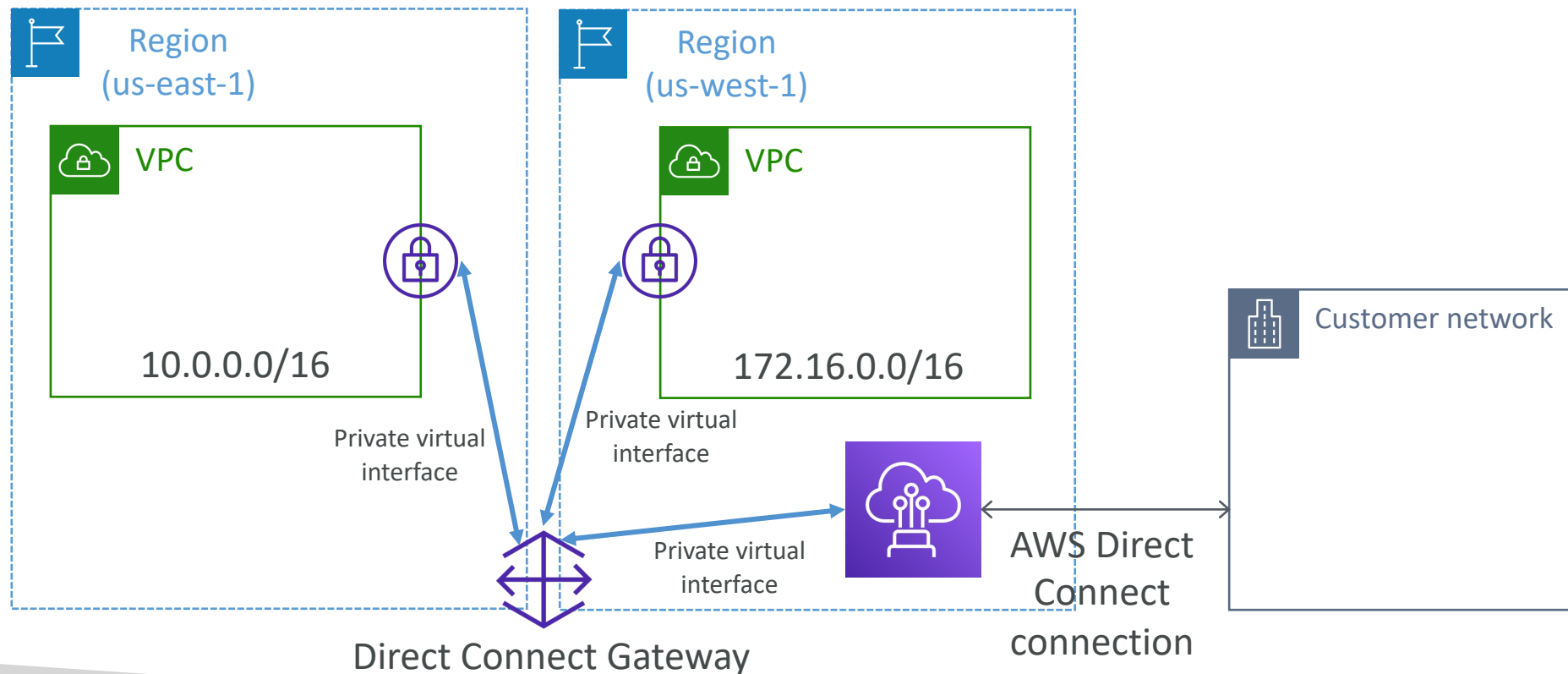
- Provides a dedicated private connection from a remote network to your VPC
- Dedicated connection must be setup between your DC and AWS Direct Connect locations
- You need to setup a Virtual Private Gateway on your VPC
- Access public resources (S3) and private (EC2) on same connection
- Use Cases: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

# Direct Connect Diagram



# Direct Connect Gateway

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

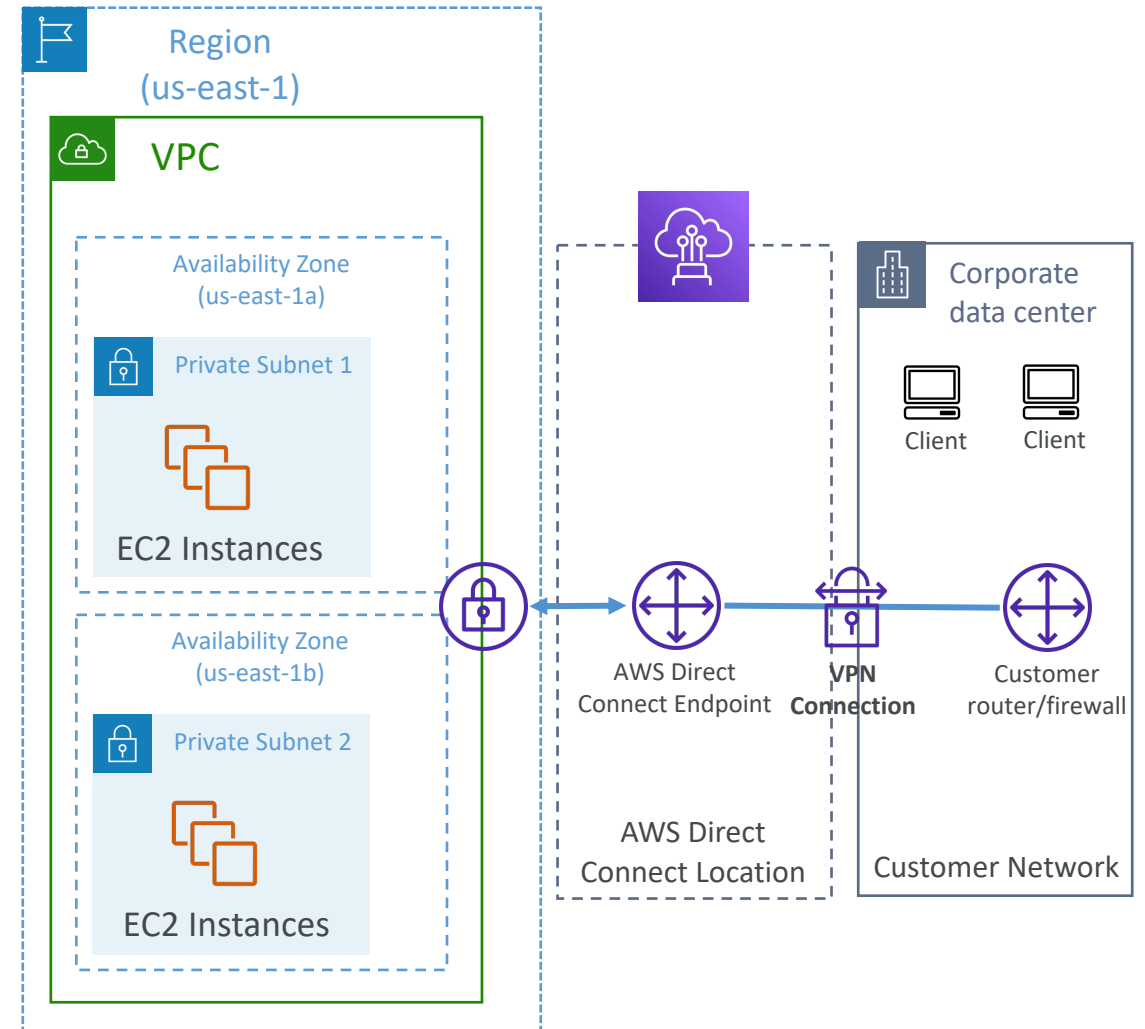


# Direct Connect – Connection Types

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

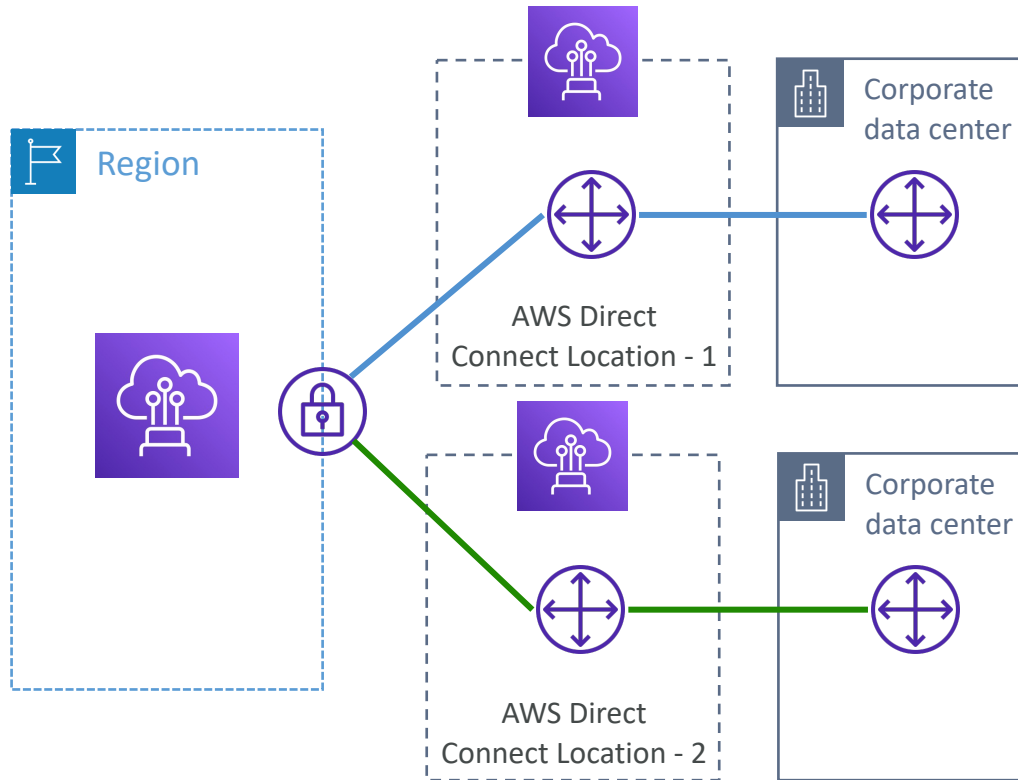
# Direct Connect – Encryption

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



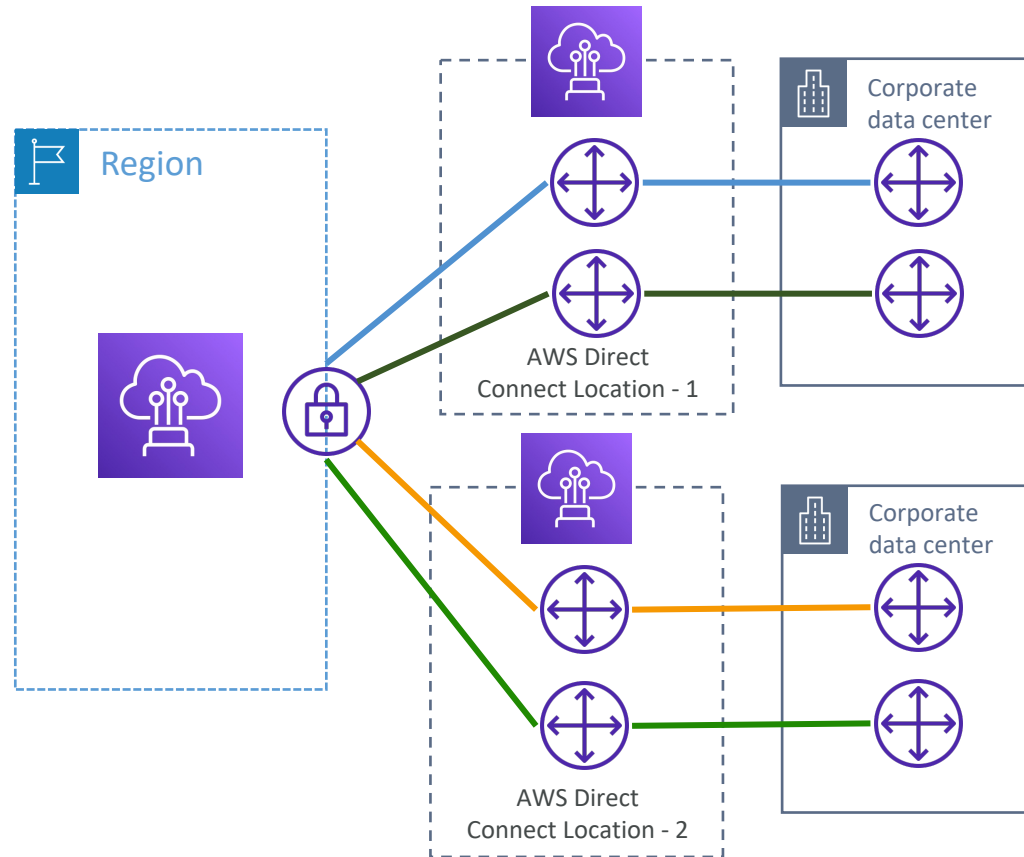
# Direct Connect - Resiliency

## High Resiliency for Critical Workloads



One connection at multiple locations

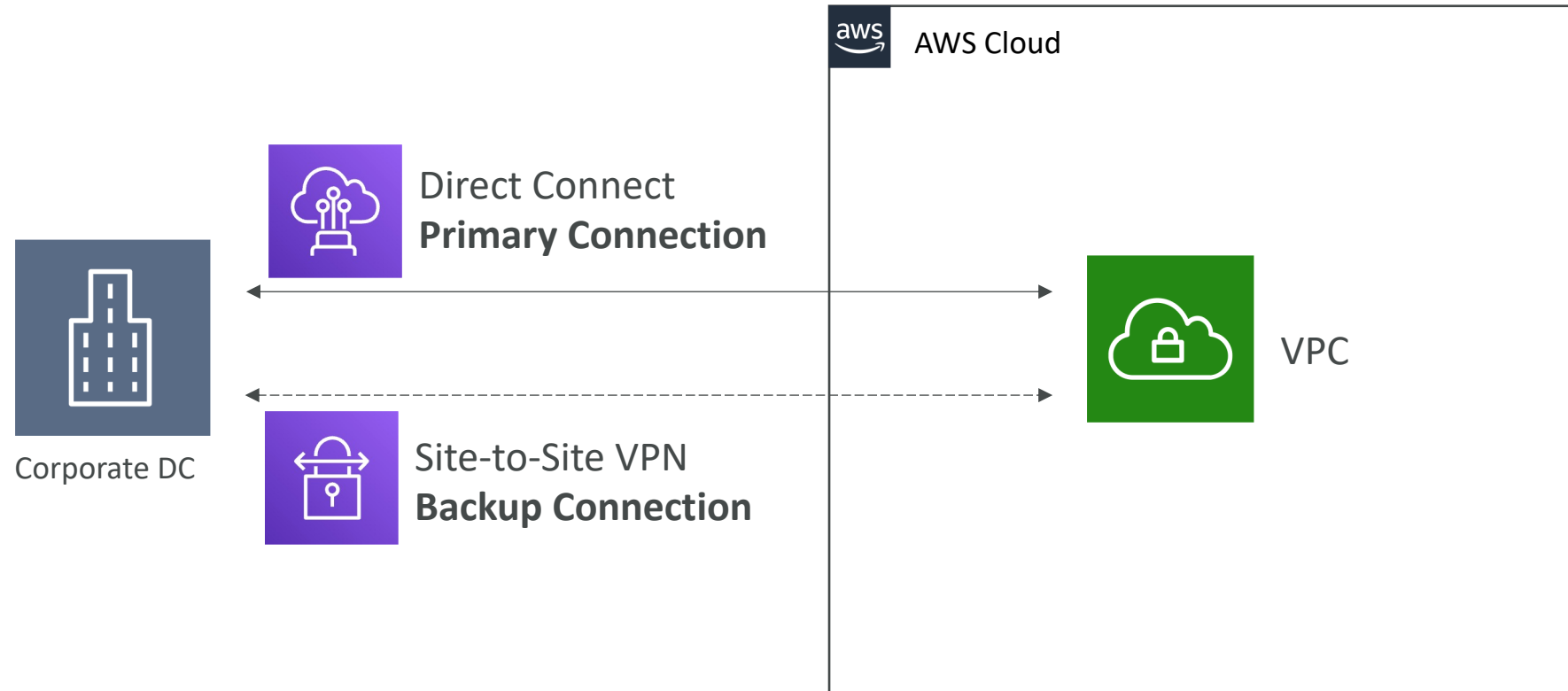
## Maximum Resiliency for Critical Workloads



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

# Site-to-Site VPN connection as a backup

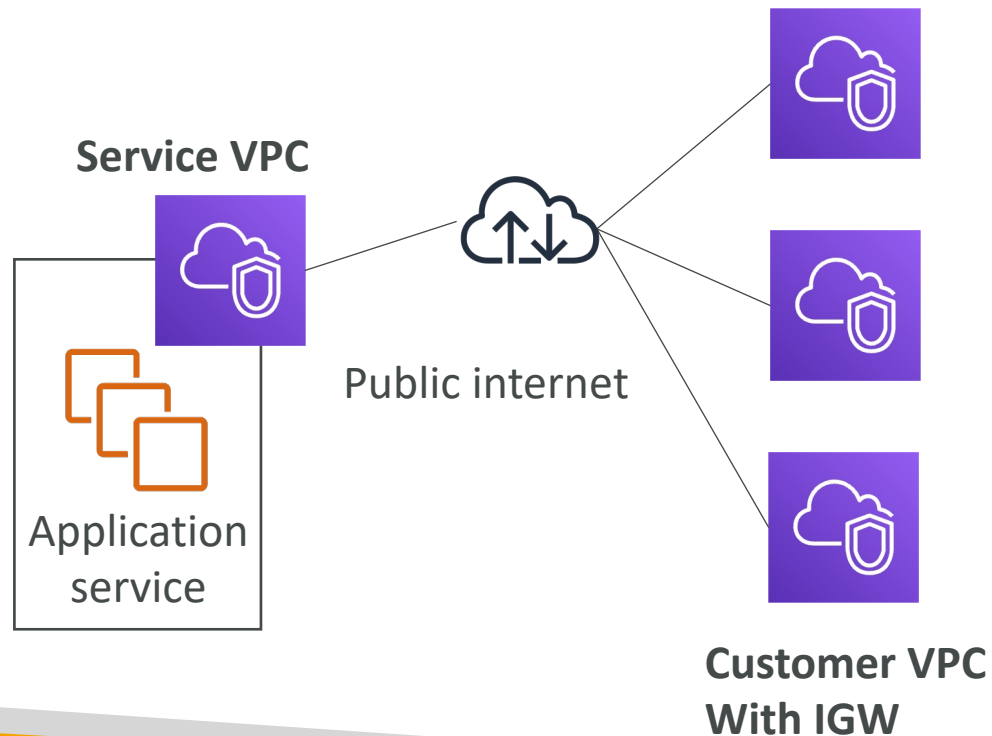
- In case Direct Connect fails, you can set up a backup Direct Connect connection (expensive), or a Site-to-Site VPN connection



# Exposing services in your VPC to other VPC

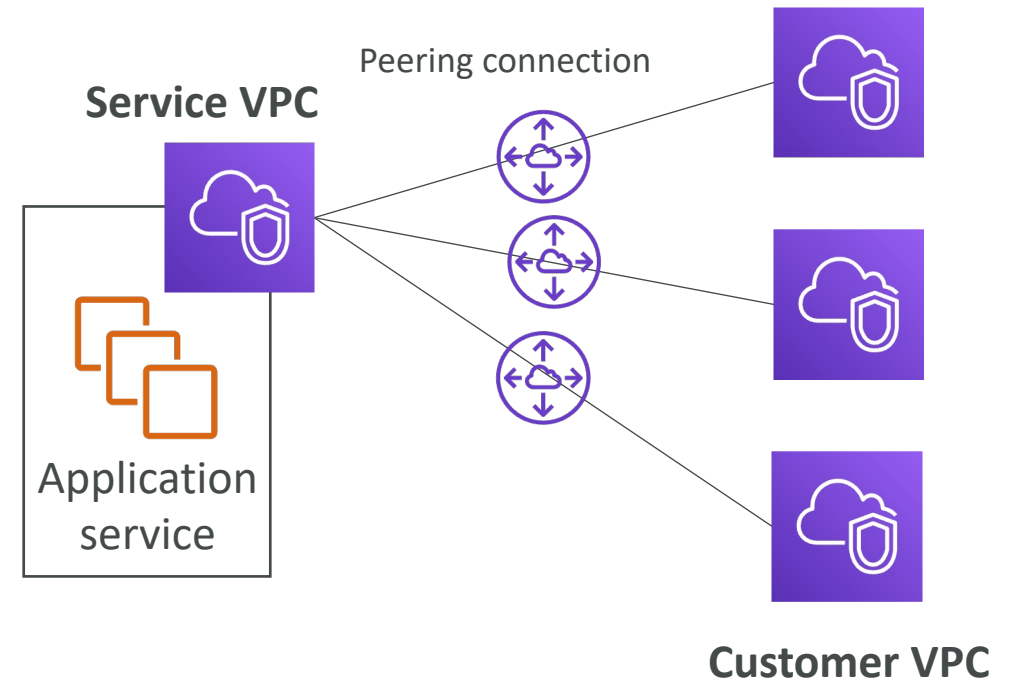
- Option 1: make it public

- Goes through the public www
- Tough to manage access



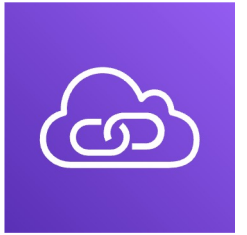
- Option 2: VPC peering

- Must create many peering relations
- Opens the **whole** network

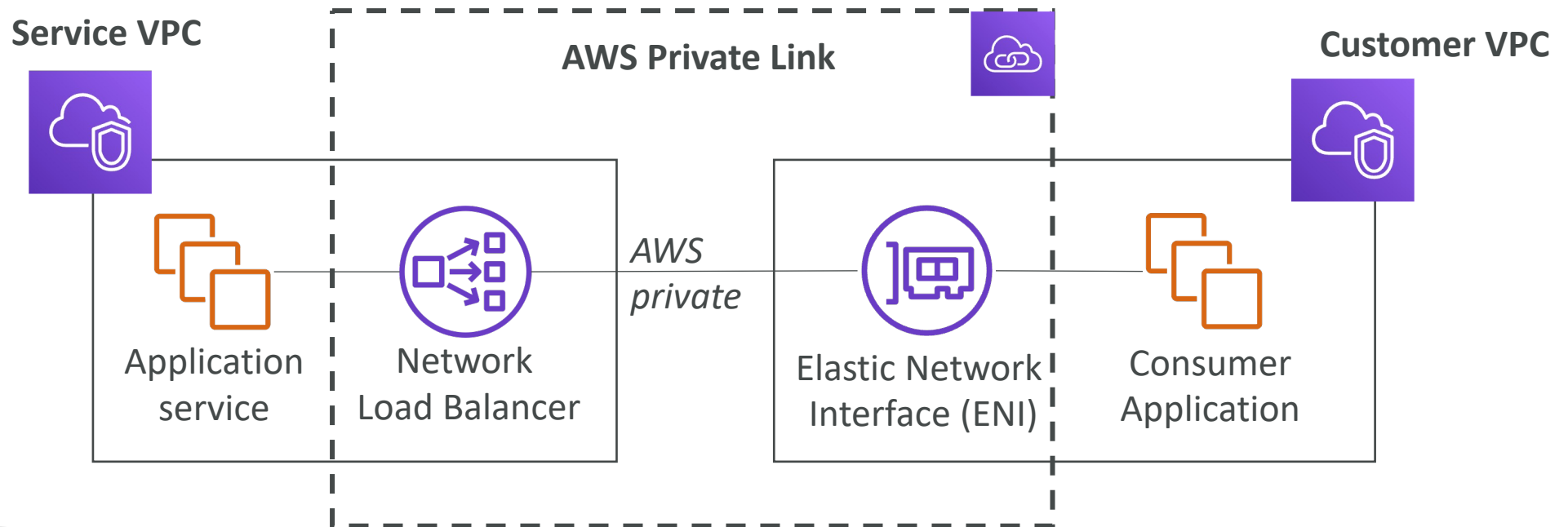




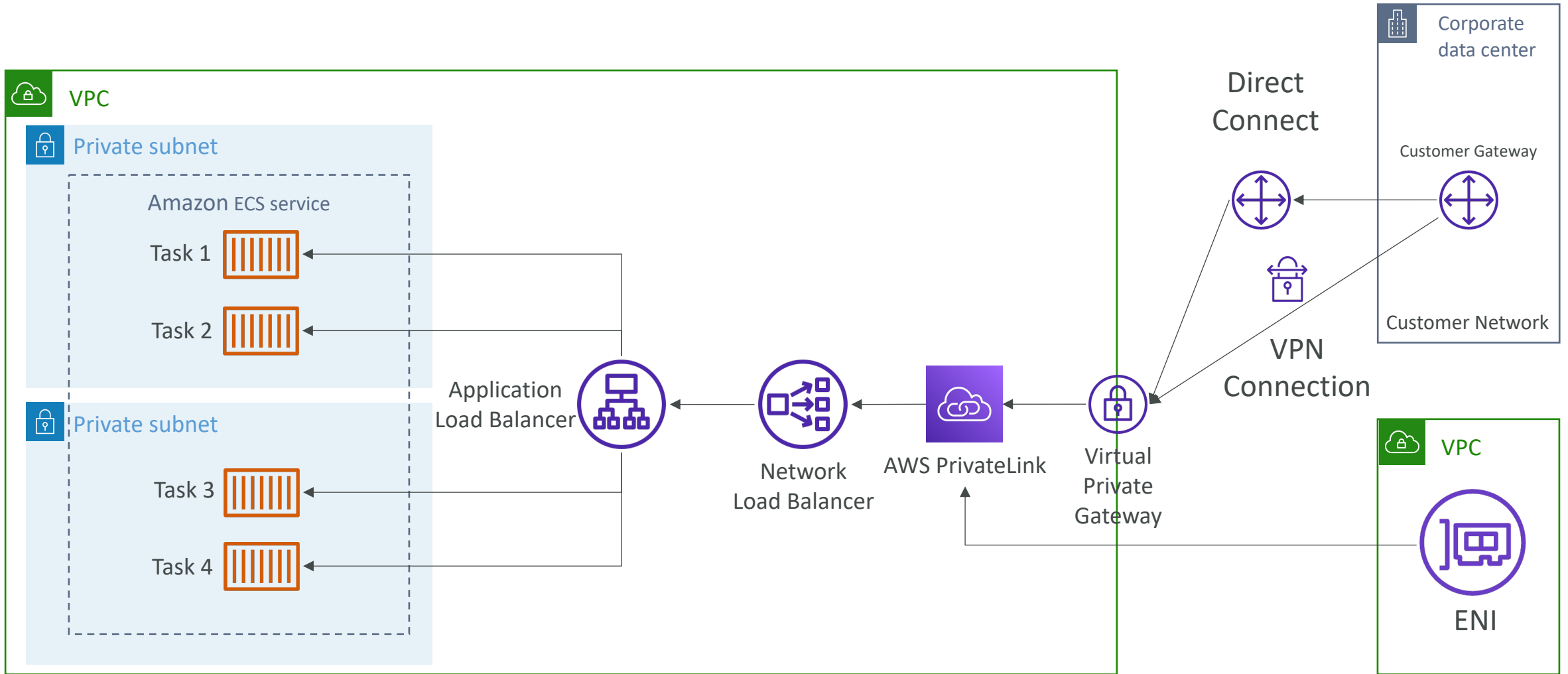
# AWS PrivateLink (VPC Endpoint Services)



- Most secure & scalable way to expose a service to 1000s of VPC (own or other accounts)
- Does not require VPC peering, internet gateway, NAT, route tables...
- Requires a network load balancer (Service VPC) and ENI (Customer VPC) or GWLB
- If the NLB is in multiple AZ, and the ENIs in multiple AZ, the solution is fault tolerant!



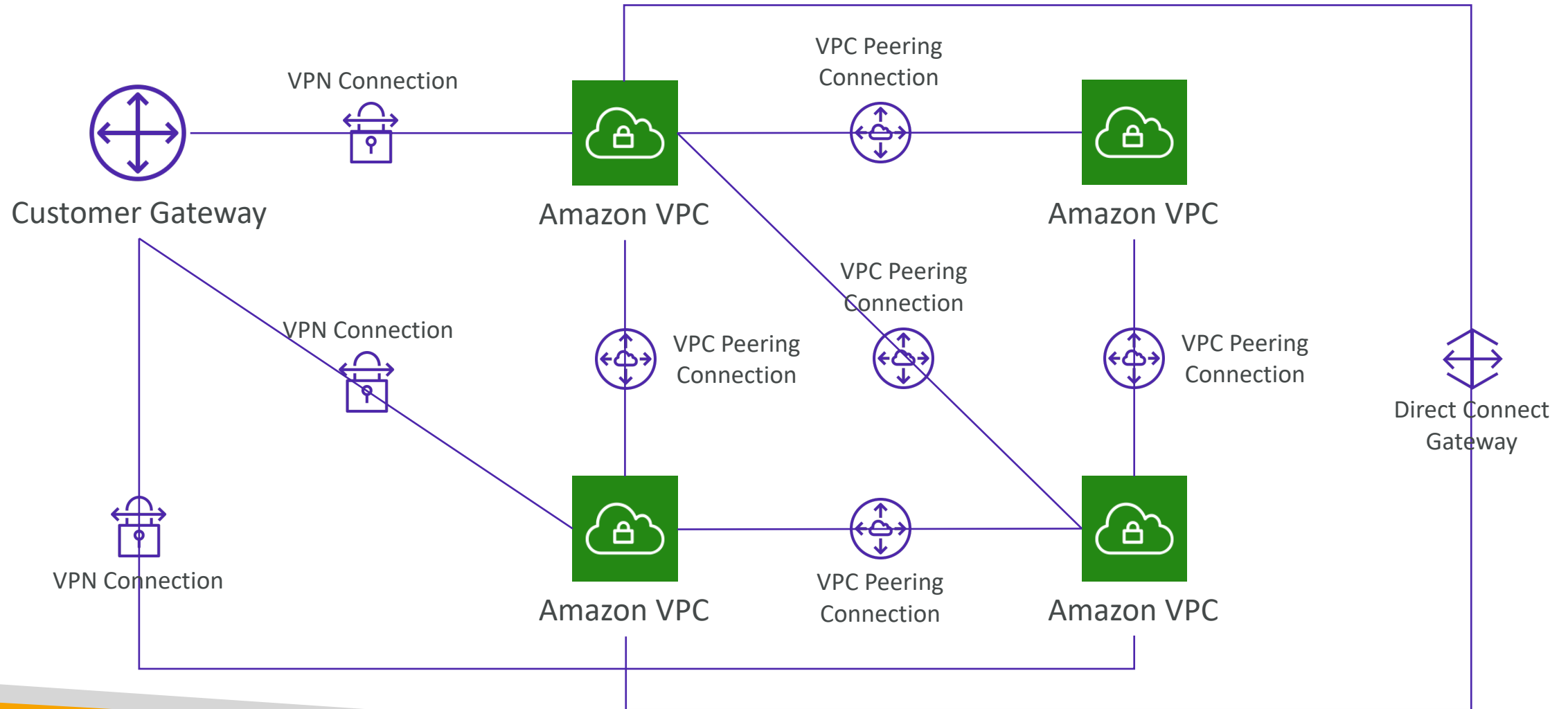
# AWS Private Link & ECS



# EC2-Classic & AWS ClassicLink (deprecated)

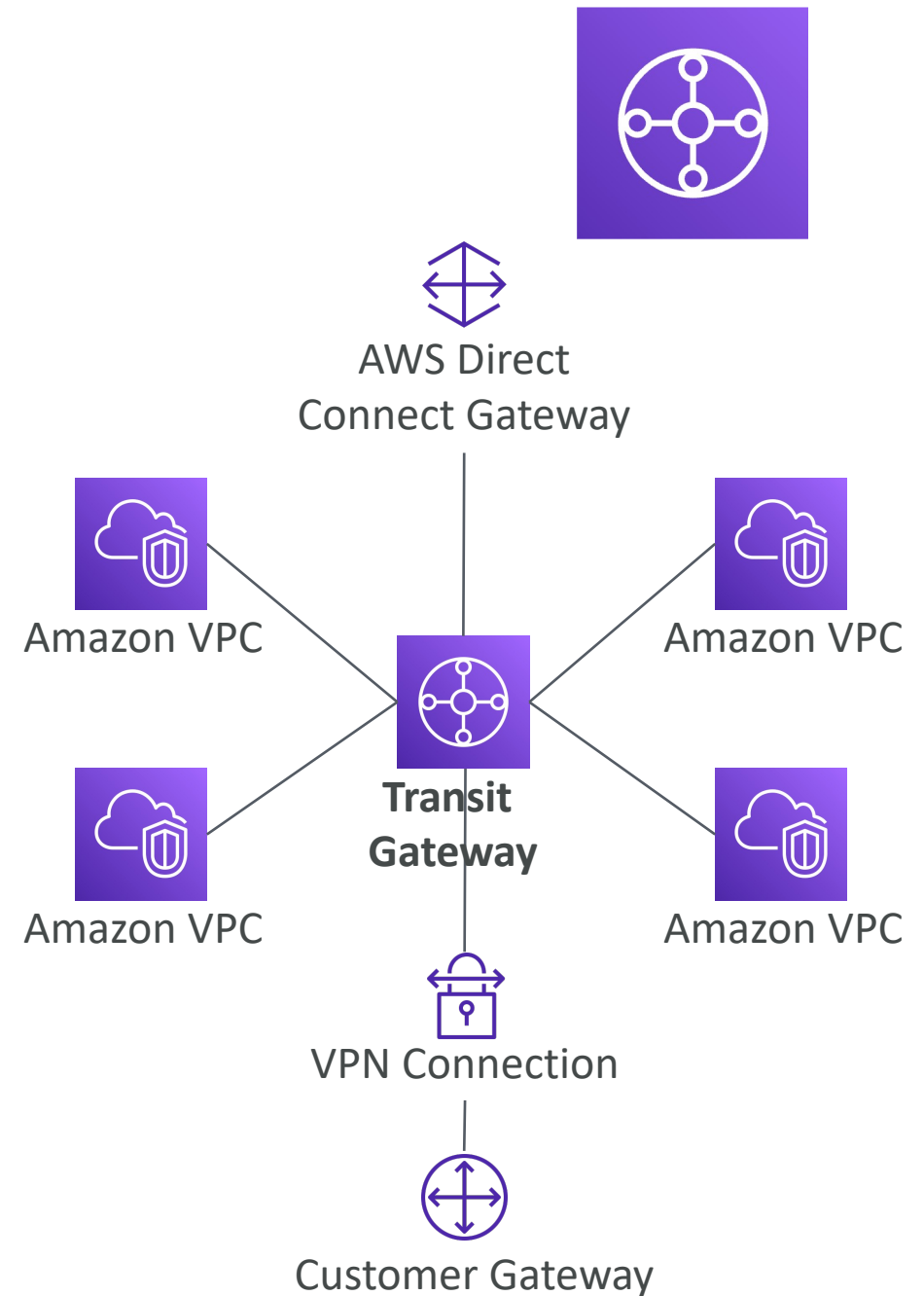
- **EC2-Classic:** instances run in a single network shared with other customers
- **Amazon VPC:** your instances run logically isolated to your AWS account
- **ClassicLink** allows you to link EC2-Classic instances to a VPC in your account
  - Must associate a security group
  - Enables communication using private IPv4 addresses
  - Removes the need to make use of public IPv4 addresses or Elastic IP addresses
- Likely to be distractors at the exam

# Network topologies can become complicated



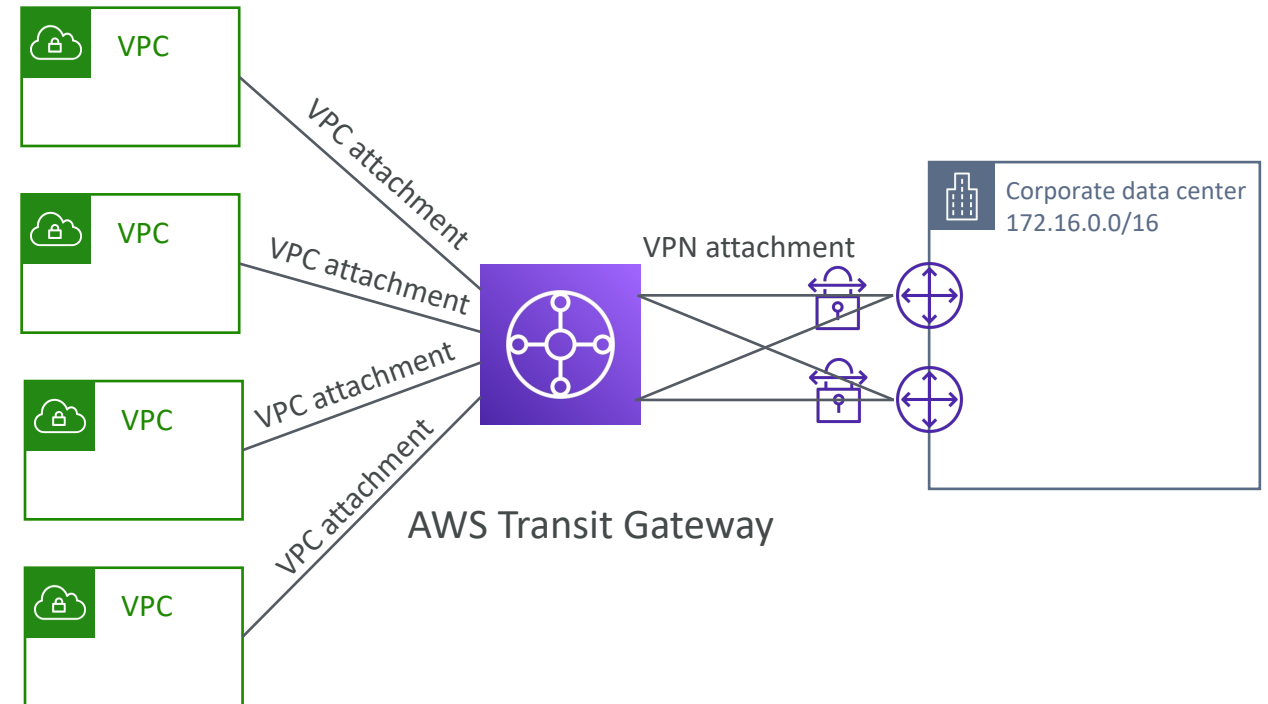
# Transit Gateway

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



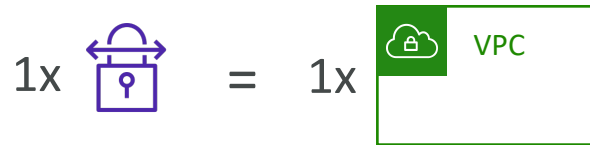
# Transit Gateway: Site-to-Site VPN ECMP


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





# Transit Gateway: throughput with ECMP

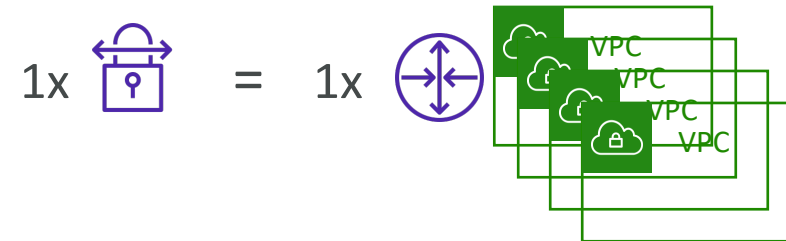
 VPN to virtual private gateway



1x  = 1.25 Gbps

 VPN connection  
(2 tunnels)

 VPN to transit gateway



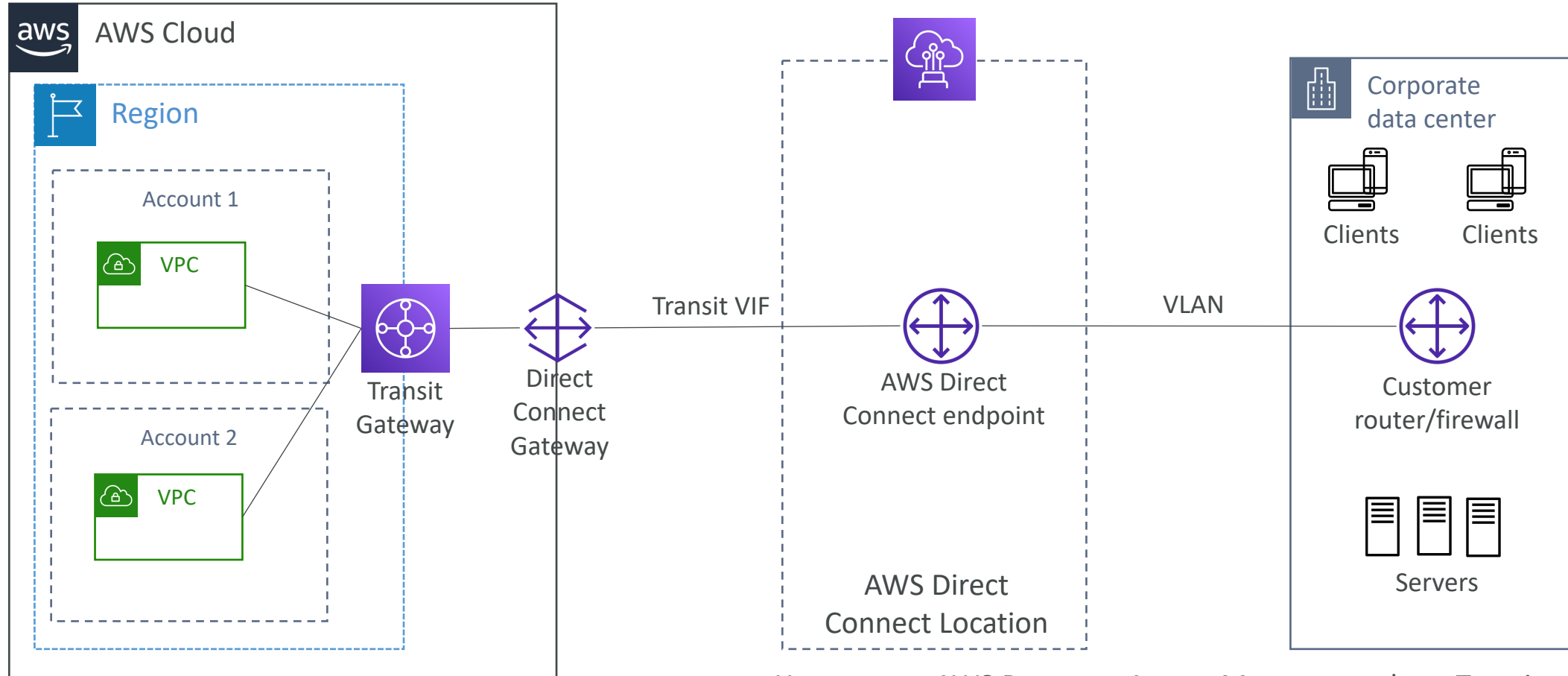
1x  = 2.5 Gbps (ECMP) – 2 tunnels used

2x  = 5.0 Gbps (ECMP)

3x  = 7.5 Gbps (ECMP)

+\$ per GB of TGW  
processed data

# Transit Gateway – Share Direct Connect between multiple accounts

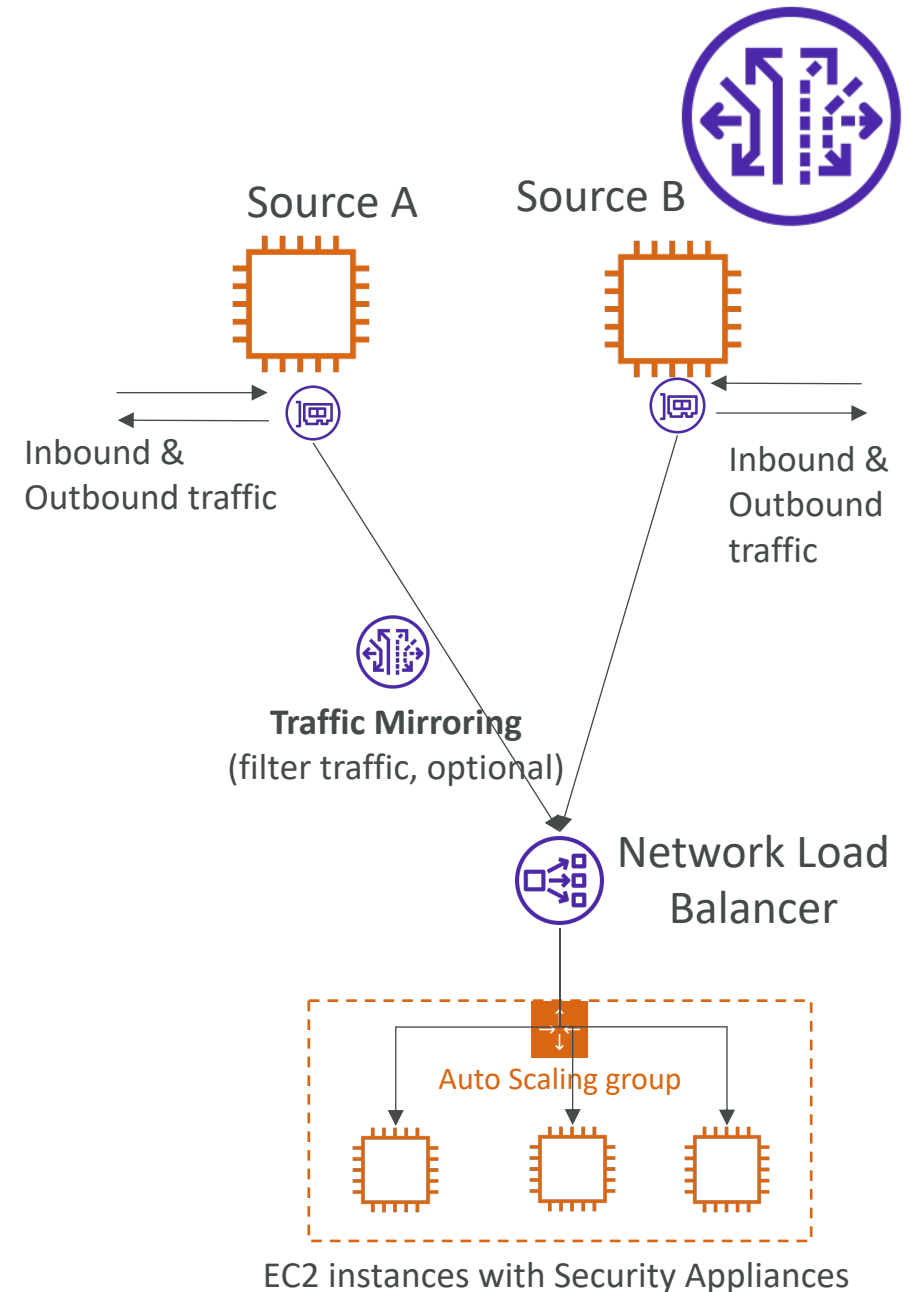


You can use AWS Resource Access Manager to share Transit Gateway with other accounts.



# VPC – Traffic Mirroring

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

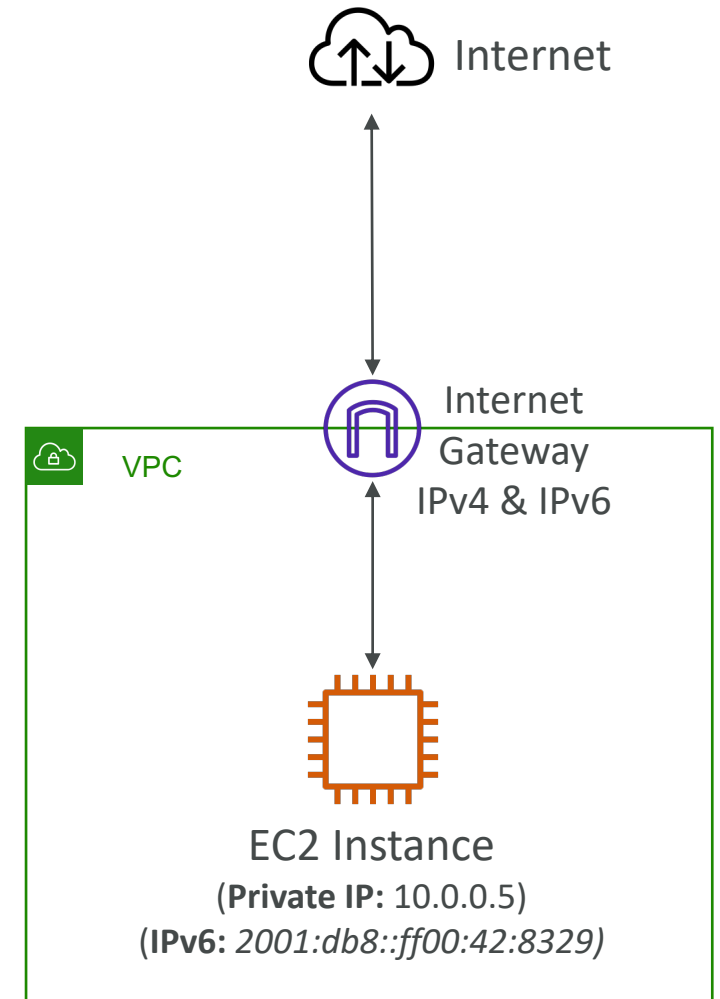


# What is IPv6?

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

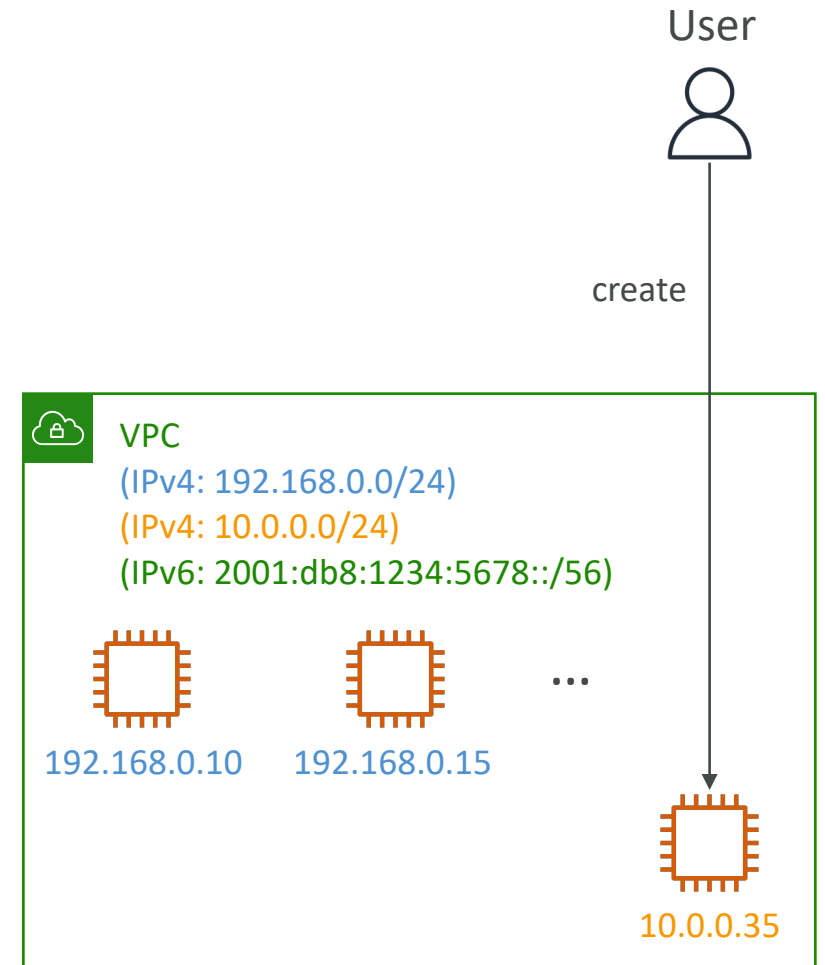
# IPv6 in VPC

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



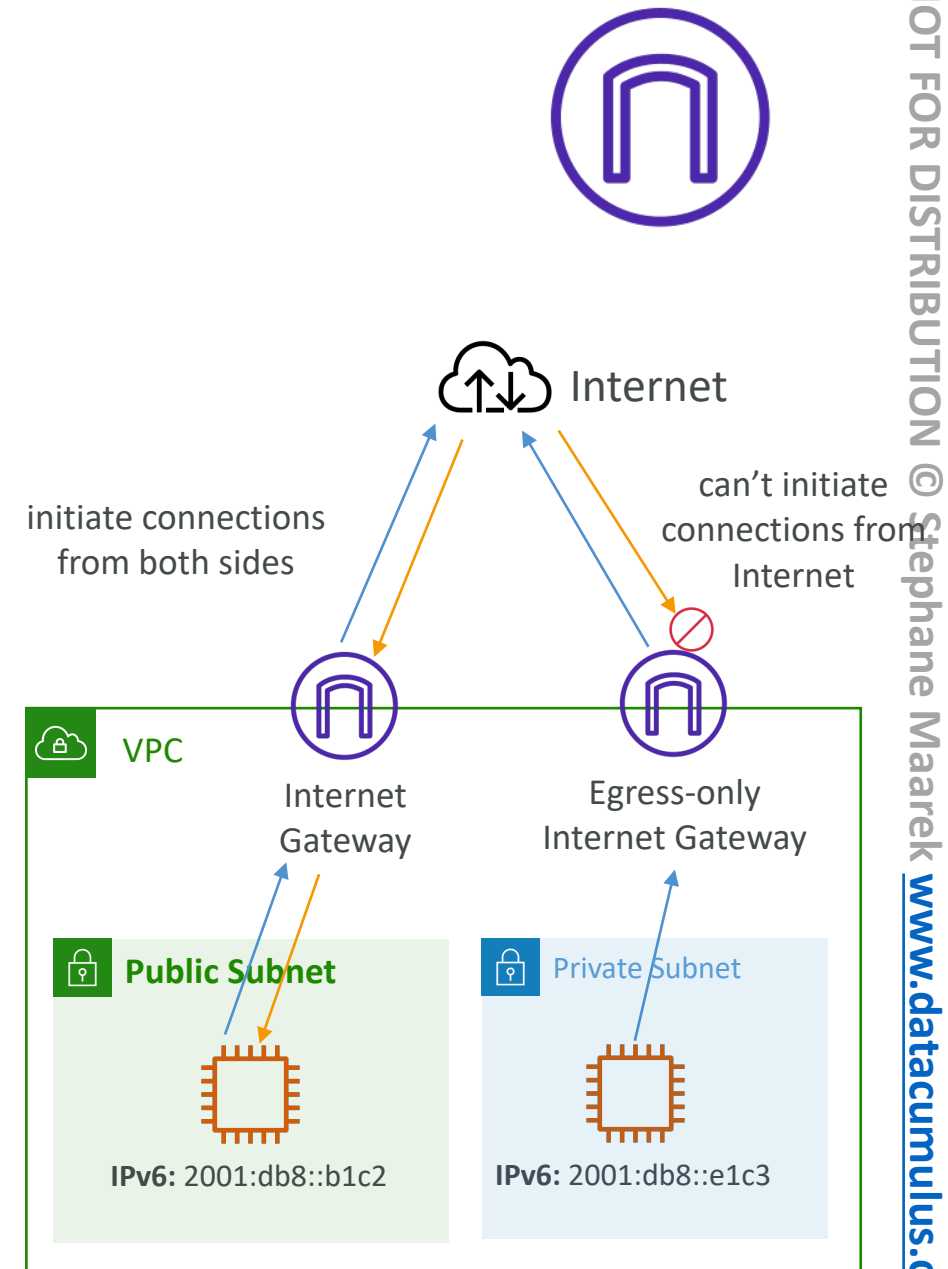
# IPv6 Troubleshooting

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

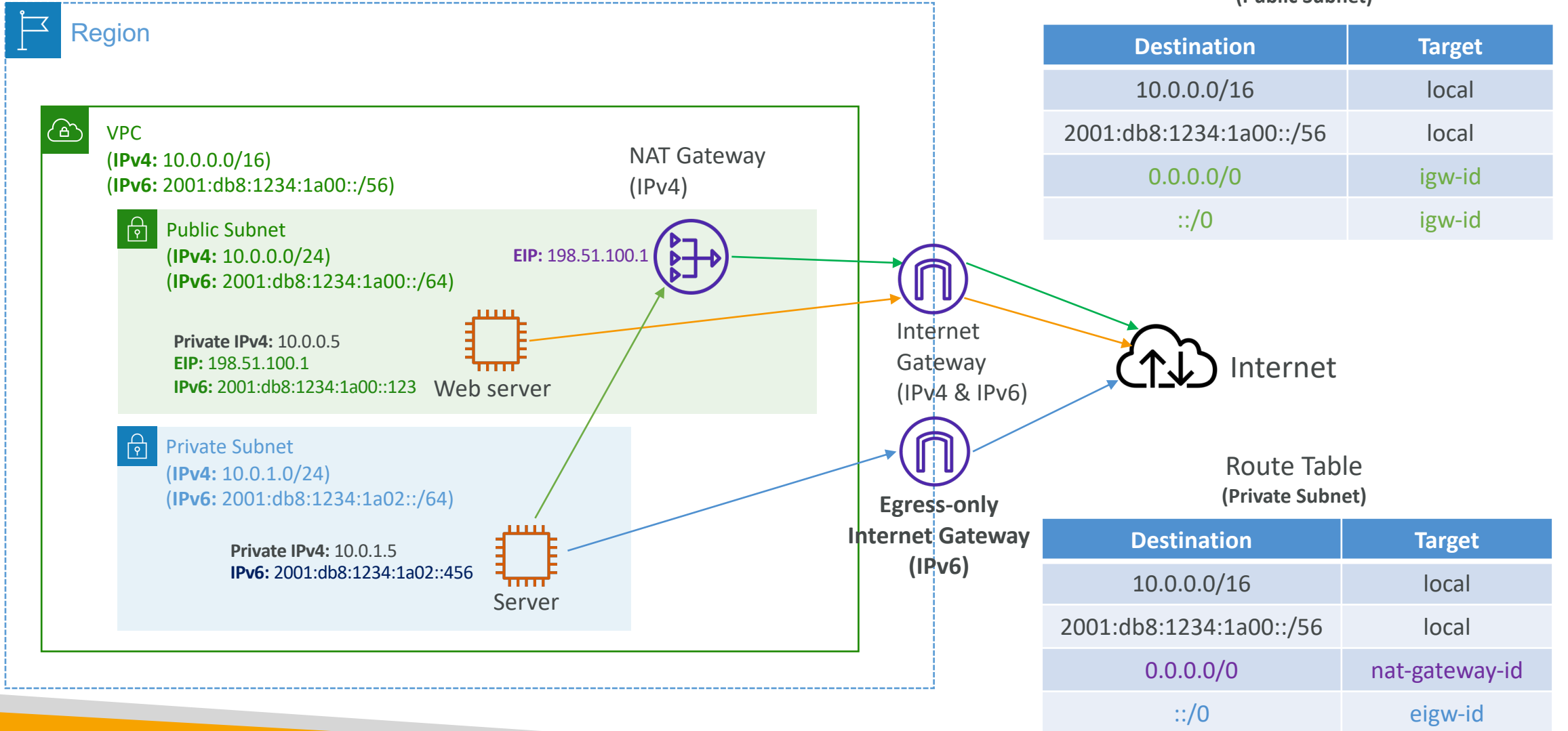


# Egress-only Internet Gateway

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



# IPv6 Routing



# VPC Section Summary (1/3)

- **CIDR** – IP Range
- **VPC** – Virtual Private Cloud => we define a list of IPv4 & IPv6 CIDR
- **Subnets** – tied to an AZ, we define a CIDR
- **Internet Gateway** – at the VPC level, provide IPv4 & IPv6 Internet Access
- **Route Tables** – must be edited to add routes from subnets to the IGW, VPC Peering Connections, VPC Endpoints, ...
- **Bastion Host** – public EC2 instance to SSH into, that has SSH connectivity to EC2 instances in private subnets
- **NAT Instances** – gives Internet access to EC2 instances in private subnets. Old, must be setup in a public subnet, disable Source / Destination check flag
- **NAT Gateway** – managed by AWS, provides scalable Internet access to private EC2 instances, IPv4 only
- **Private DNS + Route 53** – enable DNS Resolution + DNS Hostnames (VPC)

# VPC Section Summary (2/3)

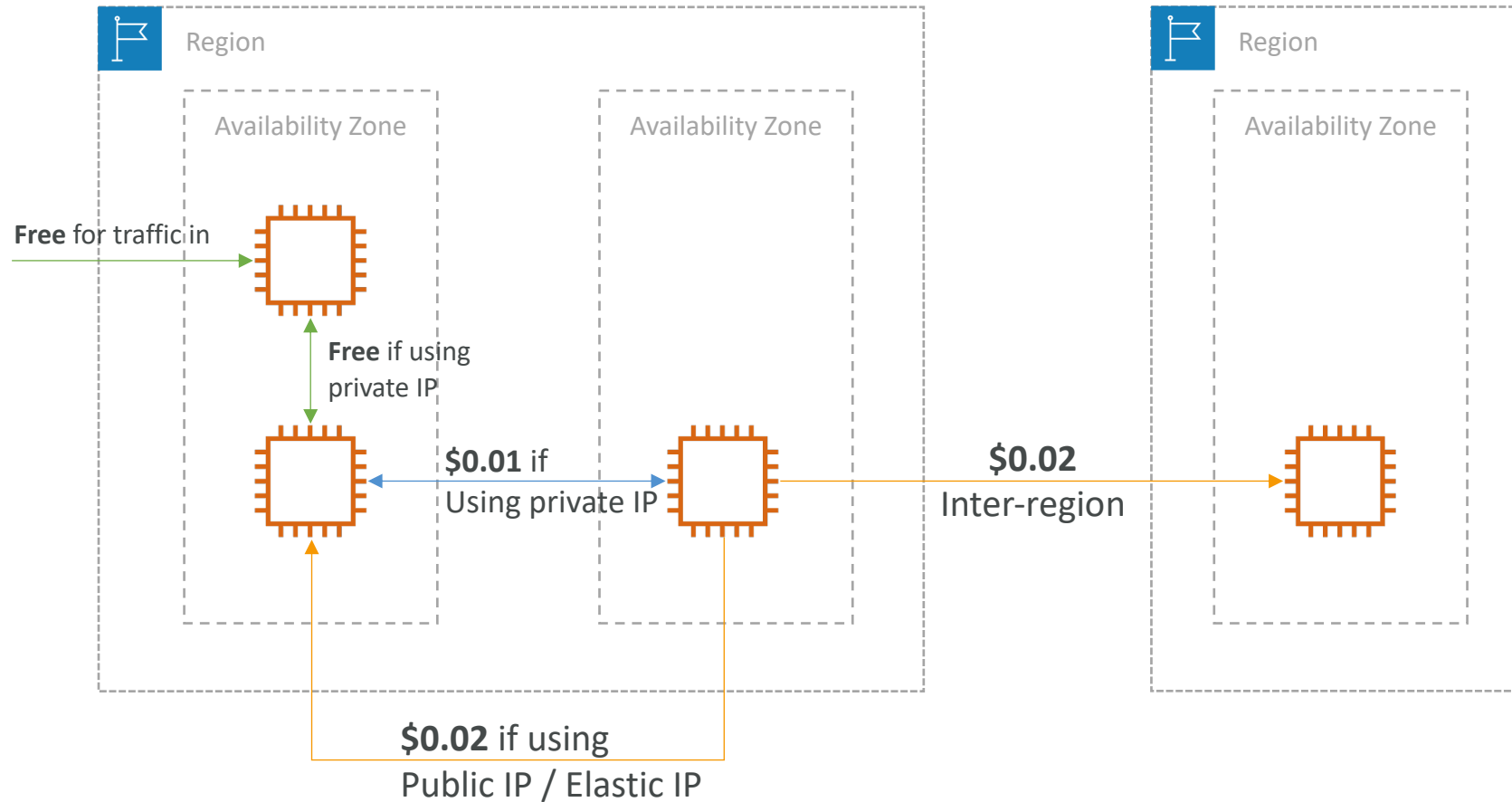
- **NACL** – stateless, subnet rules for inbound and outbound, don't forget Ephemeral Ports
- **Security Groups** – stateful, operate at the EC2 instance level
- **Reachability Analyzer** – perform network connectivity testing between AWS resources
- **VPC Peering** – connect two VPCs with non overlapping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



# VPC Section Summary (3/3)

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

# Networking Costs in AWS per GB - Simplified

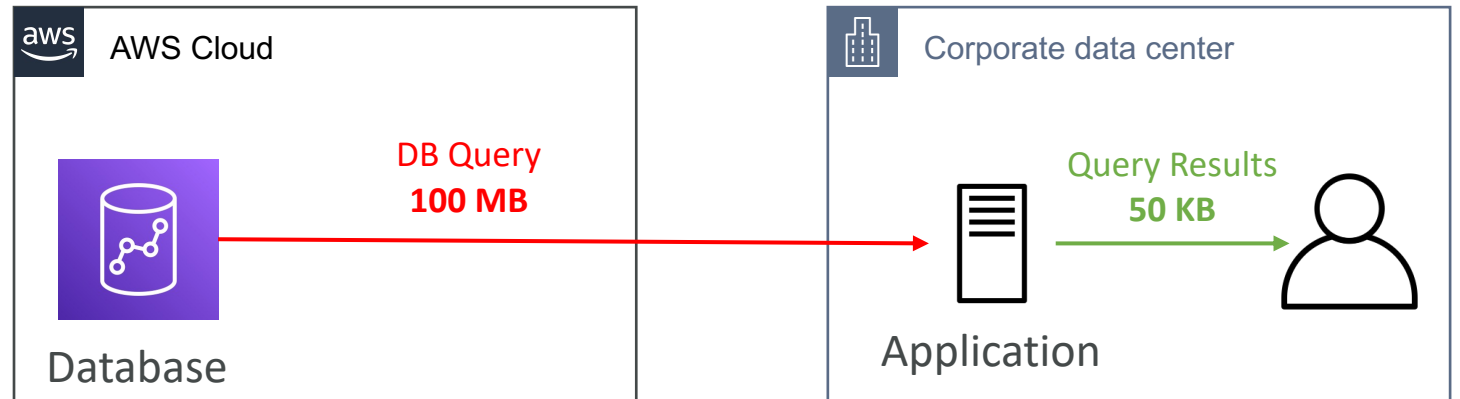


- Use Private IP instead of Public IP for good savings and better network performance
- Use same AZ for maximum savings (at the cost of high availability)

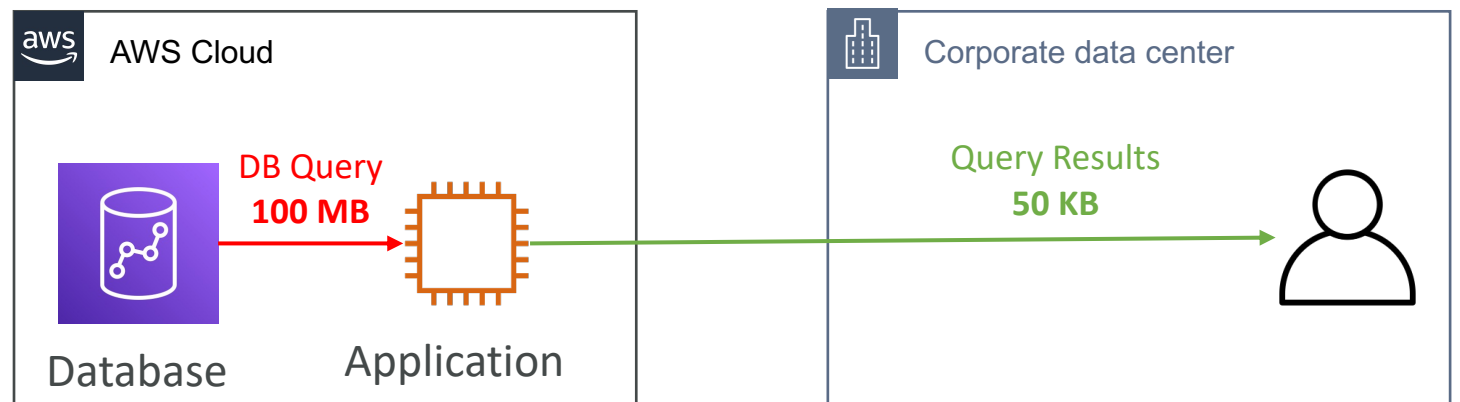
# Minimizing egress traffic network cost

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

## Egress cost is high

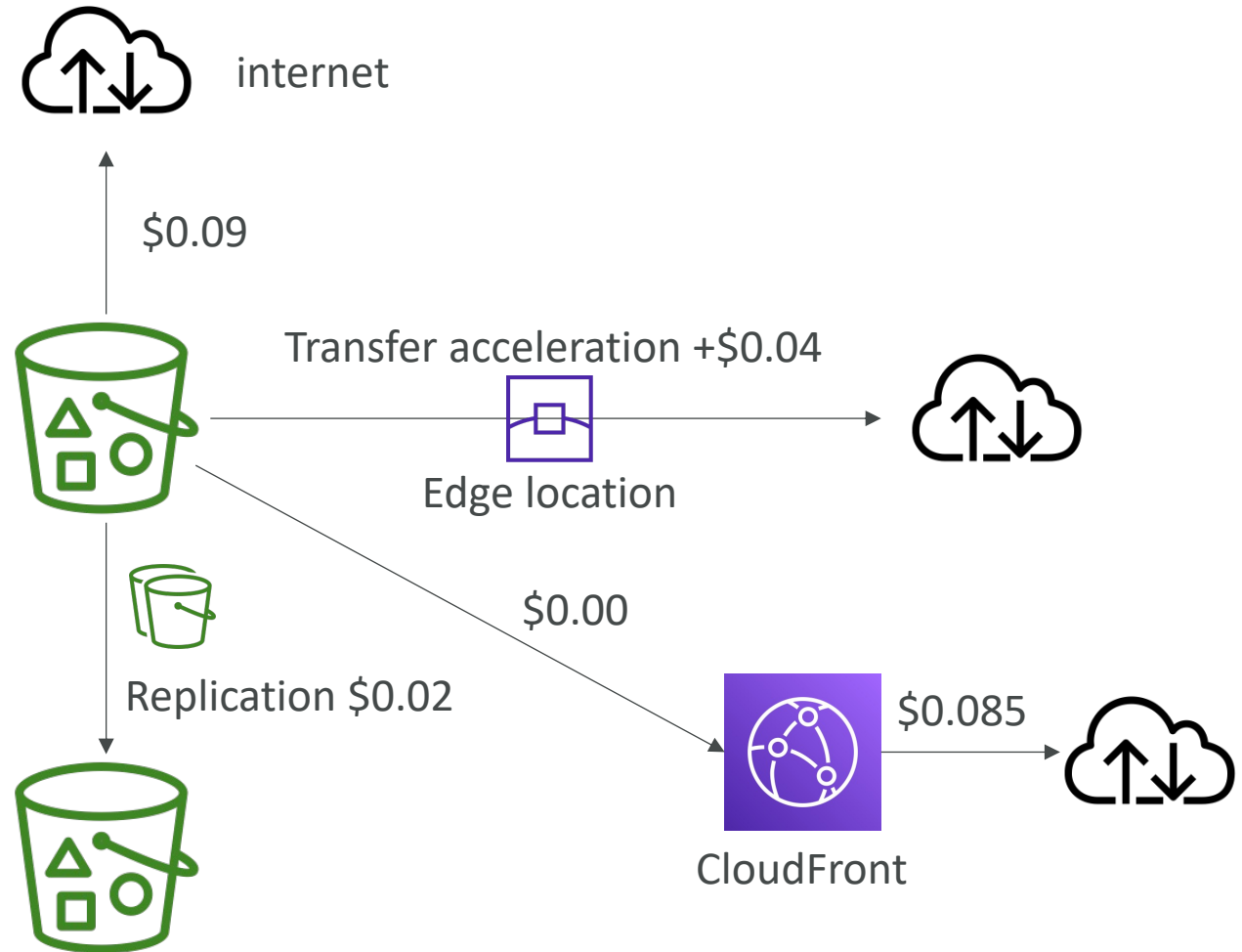


## Egress cost is minimized



# S3 Data Transfer Pricing – Analysis for USA

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



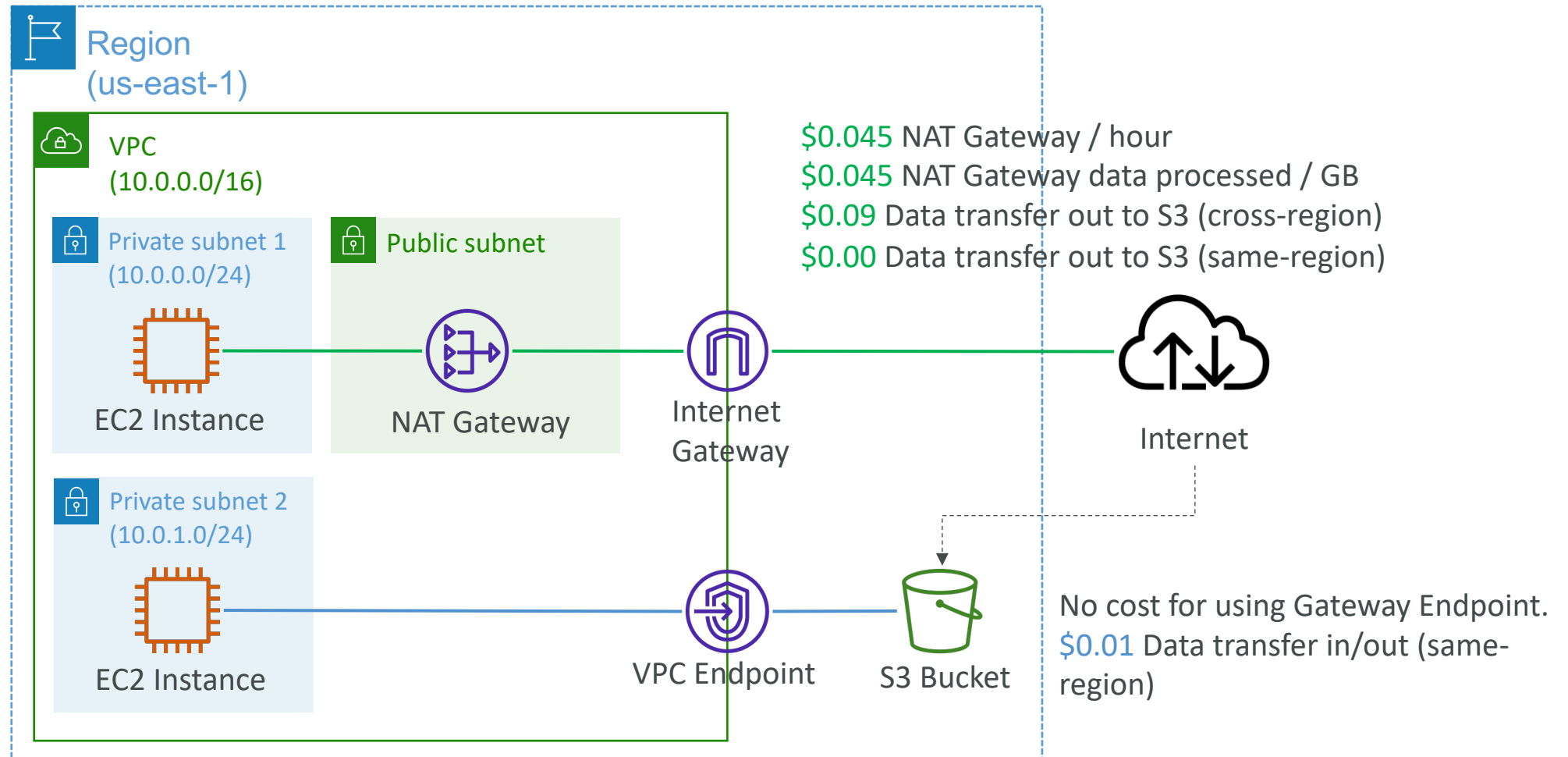
# Pricing: NAT Gateway vs Gateway VPC Endpoint

Subnet 1 route table

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

Subnet 2 route table

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

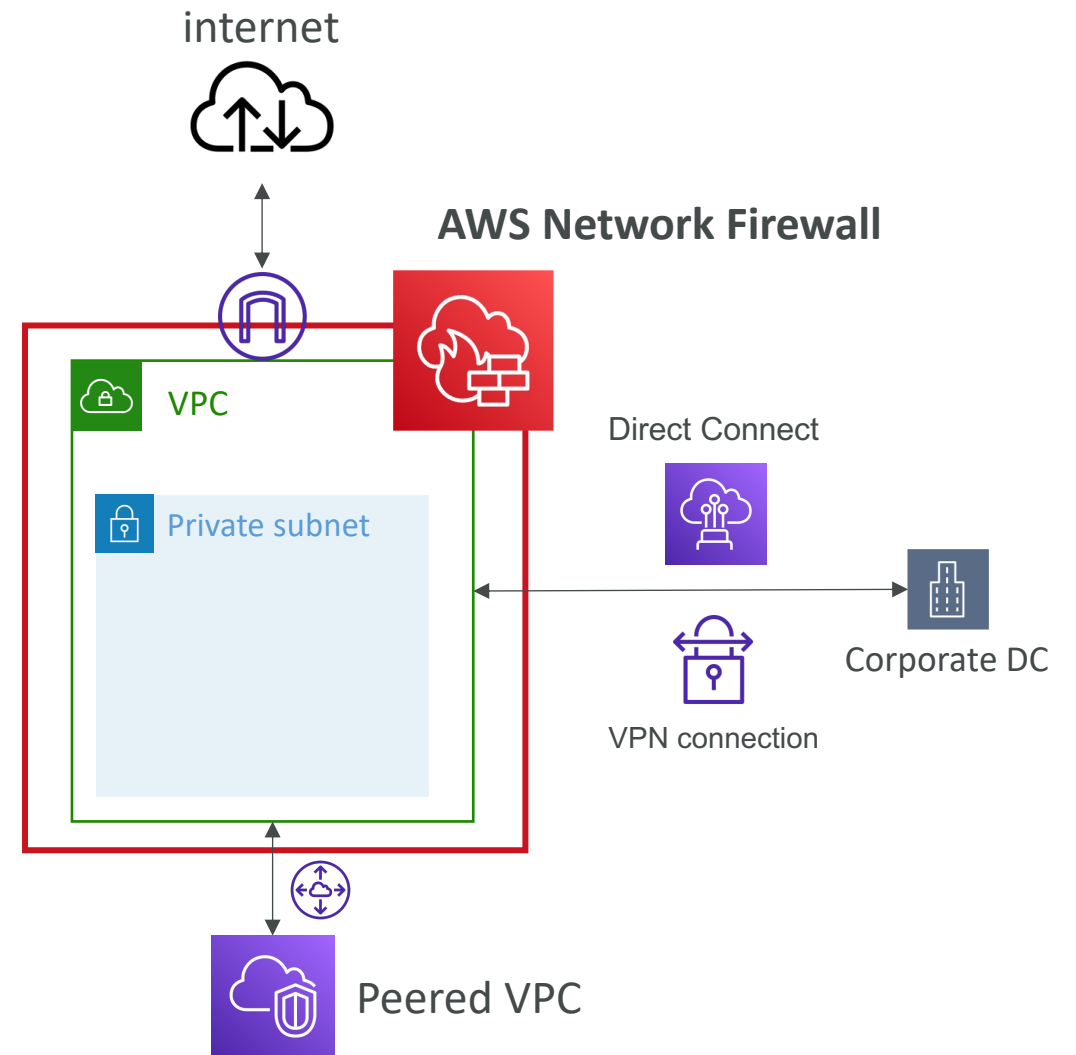


# Network Protection on AWS

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

# AWS Network Firewall

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



# Network Firewall – Fine Grained Controls



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

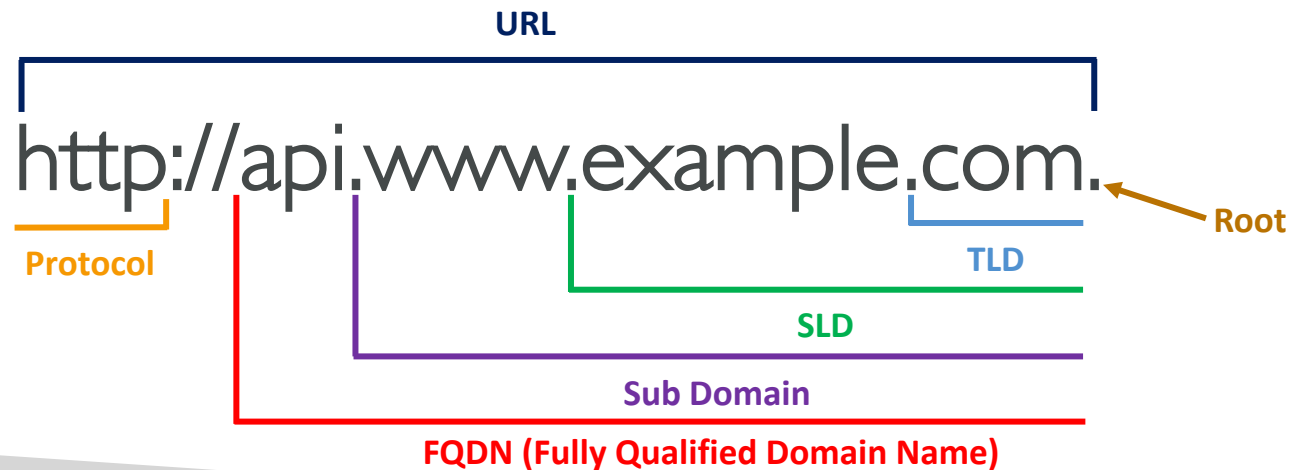


# Route 53

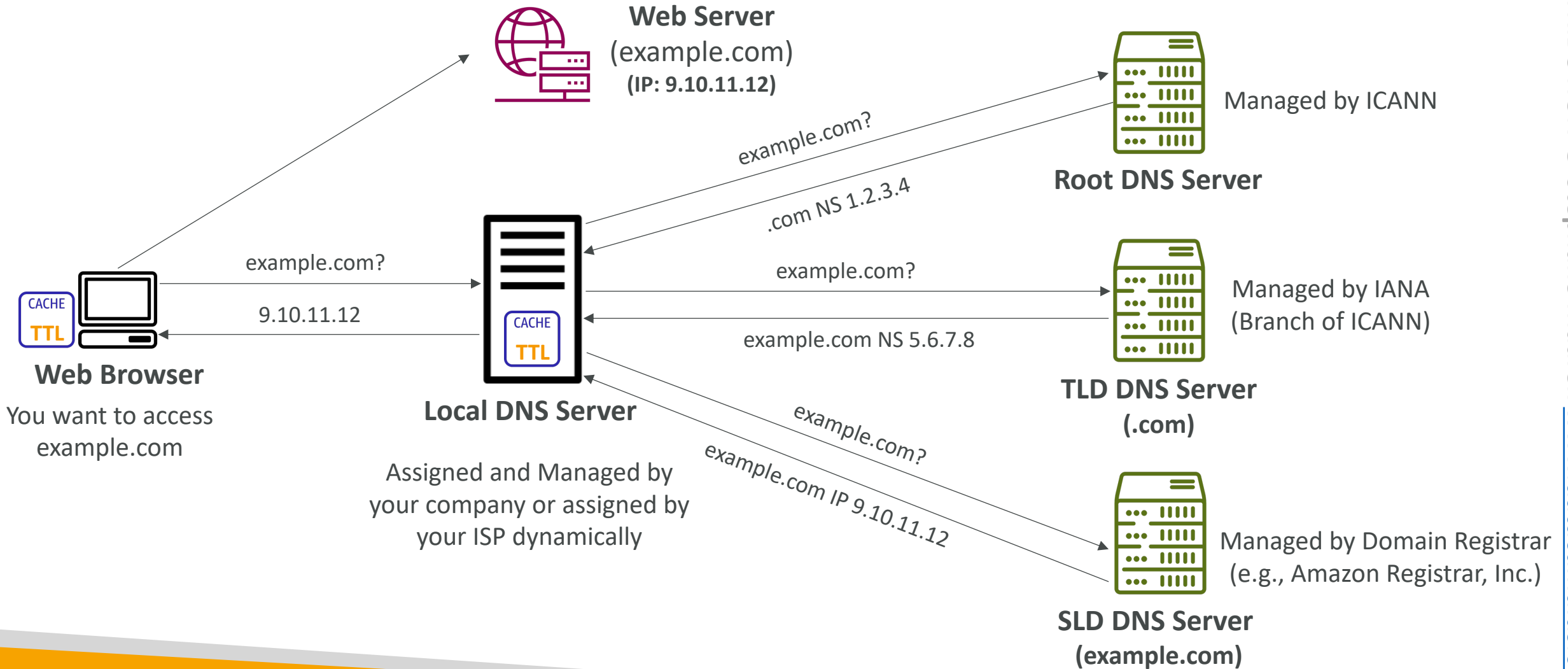


# DNS Terminologies

- Domain Registrar: Amazon Route 53, GoDaddy, ...
- DNS Records: A, AAAA, CNAME, NS, ...
- Zone File: contains DNS records
- Name Server: resolves DNS queries (Authoritative or Non-Authoritative)
- Top Level Domain (TLD): .com, .us, .in, .gov, .org, ...
- Second Level Domain (SLD): amazon.com, google.com, ...

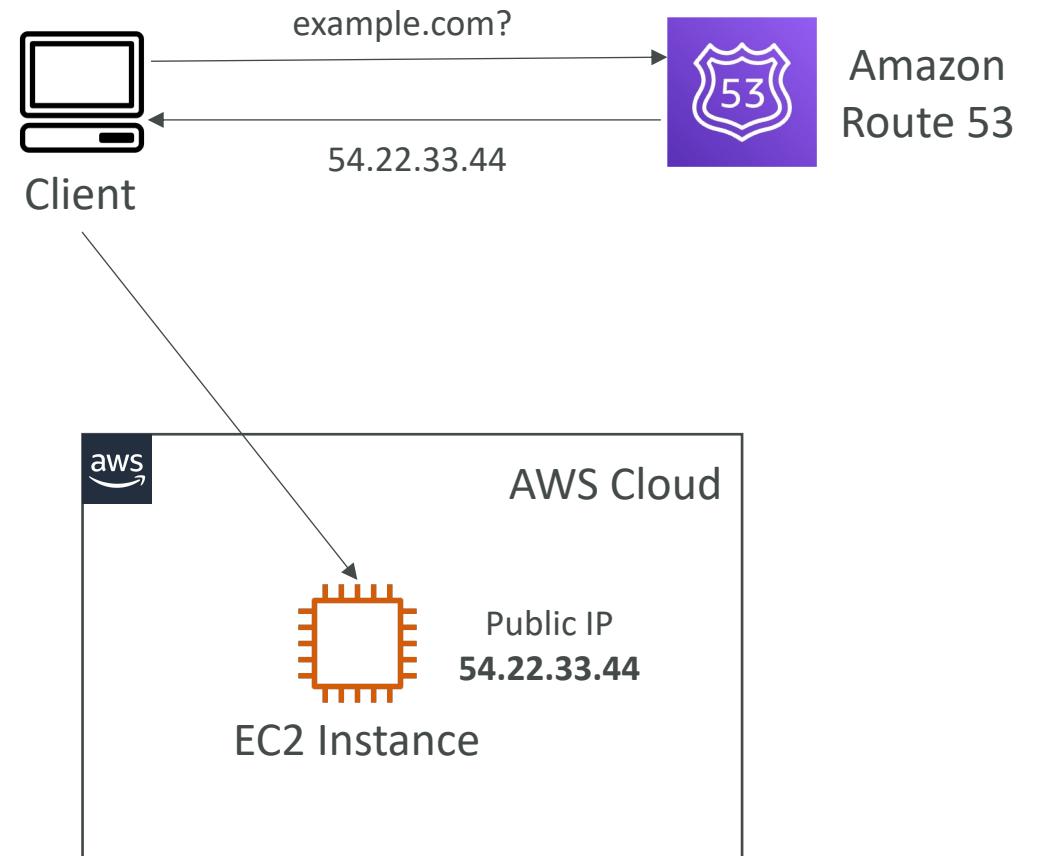


# How DNS Works



# Amazon Route 53

- A highly available, scalable, fully managed and *Authoritative* DNS
  - Authoritative = the customer (you) can update the DNS records
- Route 53 is also a Domain Registrar
- Ability to check the health of your resources
- The only AWS service which provides 100% availability SLA
- Why Route 53? 53 is a reference to the traditional DNS port



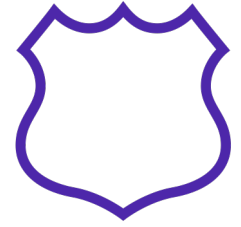
# Route 53 – Records

- How you want to route traffic for a domain
- Each record contains:
  - **Domain/subdomain Name** – e.g., example.com
  - **Record Type** – e.g., A or AAAA
  - **Value** – e.g., 12.34.56.78
  - **Routing Policy** – how Route 53 responds to queries
  - **TTL** – amount of time the record cached at DNS Resolvers
- Route 53 supports the following DNS record types:
  - (must know) A / AAAA / CNAME / NS
  - (advanced) CAA / DS / MX / NAPTR / PTR / SOA / TXT / SPF / SRV

# Route 53 – Record Types

- **A** – maps a hostname to IPv4
- **AAAA** – maps a hostname to IPv6
- **CNAME** – maps a hostname to another hostname
  - The target is a domain name which must have an A or AAAA record
  - Can't create a CNAME record for the top node of a DNS namespace (Zone Apex)
  - Example: you can't create for example.com, but you can create for www.example.com
- **NS** – Name Servers for the Hosted Zone
  - Control how traffic is routed for a domain

# Route 53 – Hosted Zones



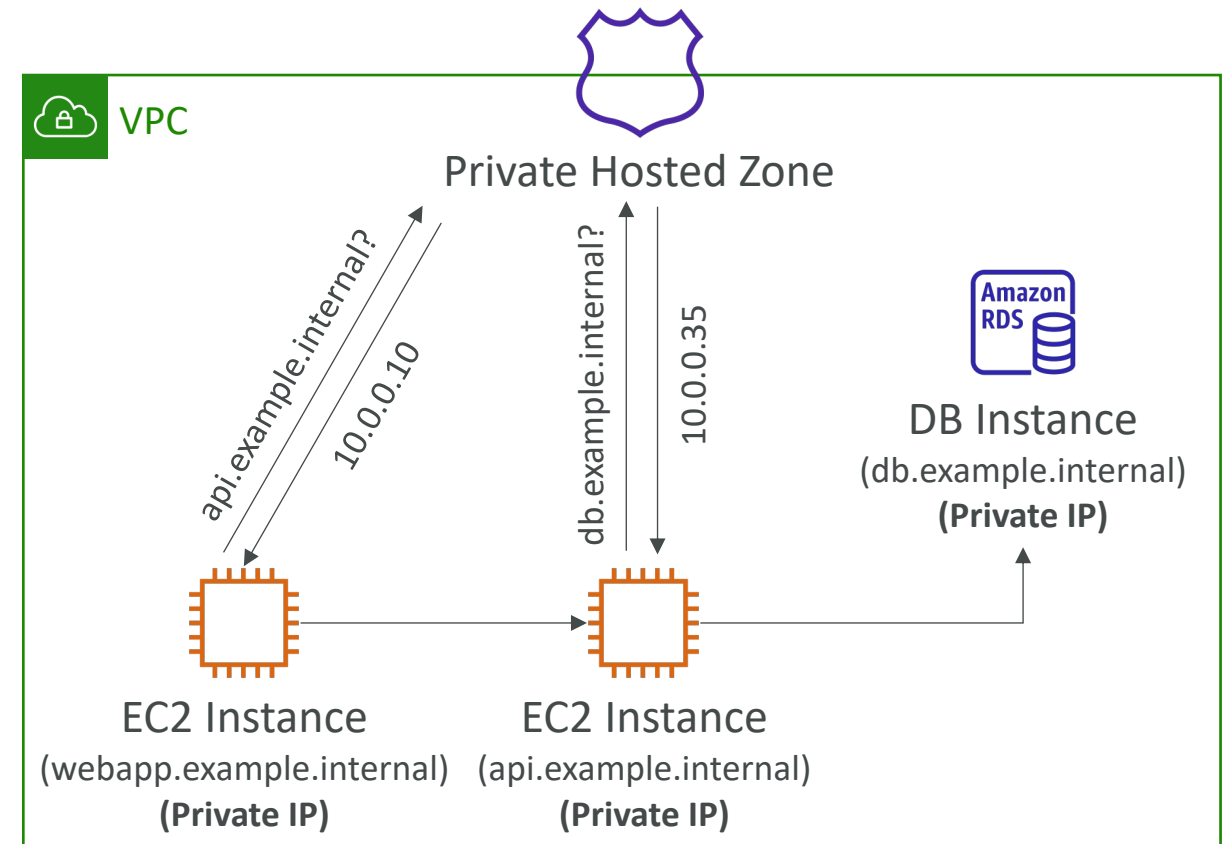
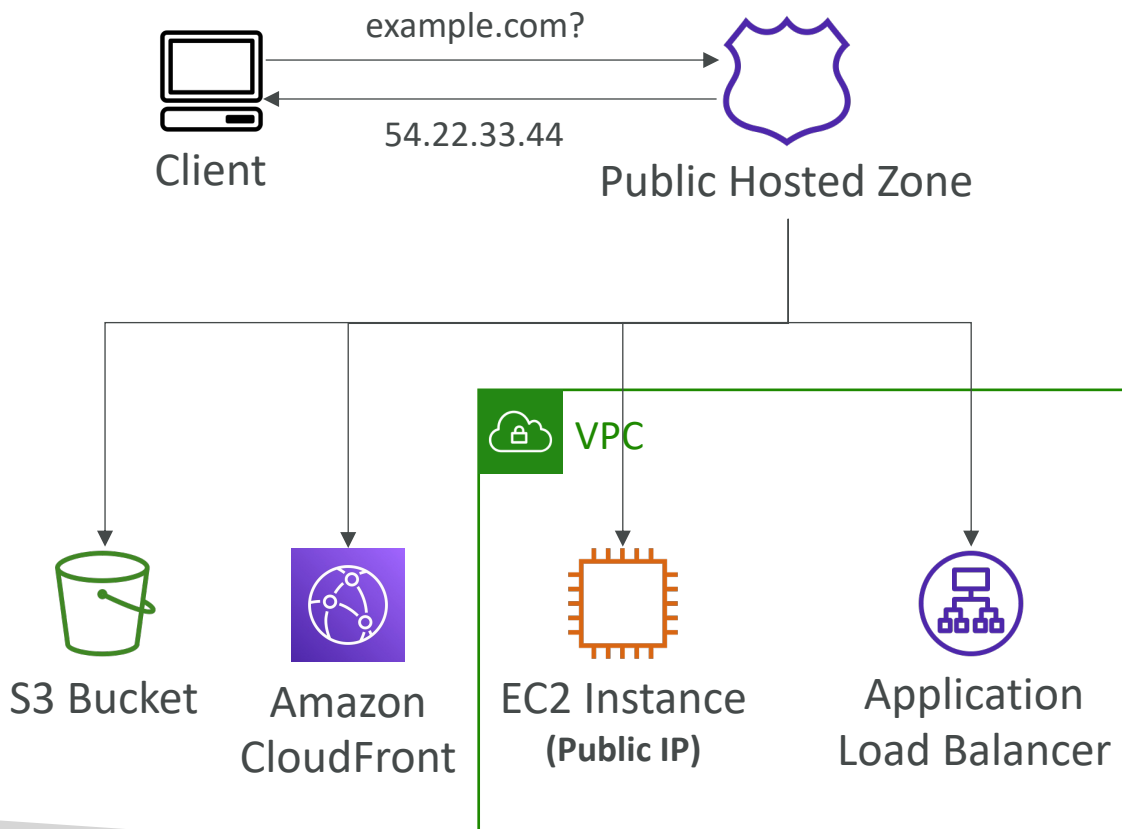
- A container for records that define how to route traffic to a domain and its subdomains
- **Public Hosted Zones** – contains records that specify how to route traffic on the Internet (public domain names)  
[application | .mypublicdomain.com](#)
- **Private Hosted Zones** – contain records that specify how you route traffic within one or more VPCs (private domain names)  
[application | .company.internal](#)
- You pay \$0.50 per month per hosted zone



# Route 53 – Public vs. Private Hosted Zones

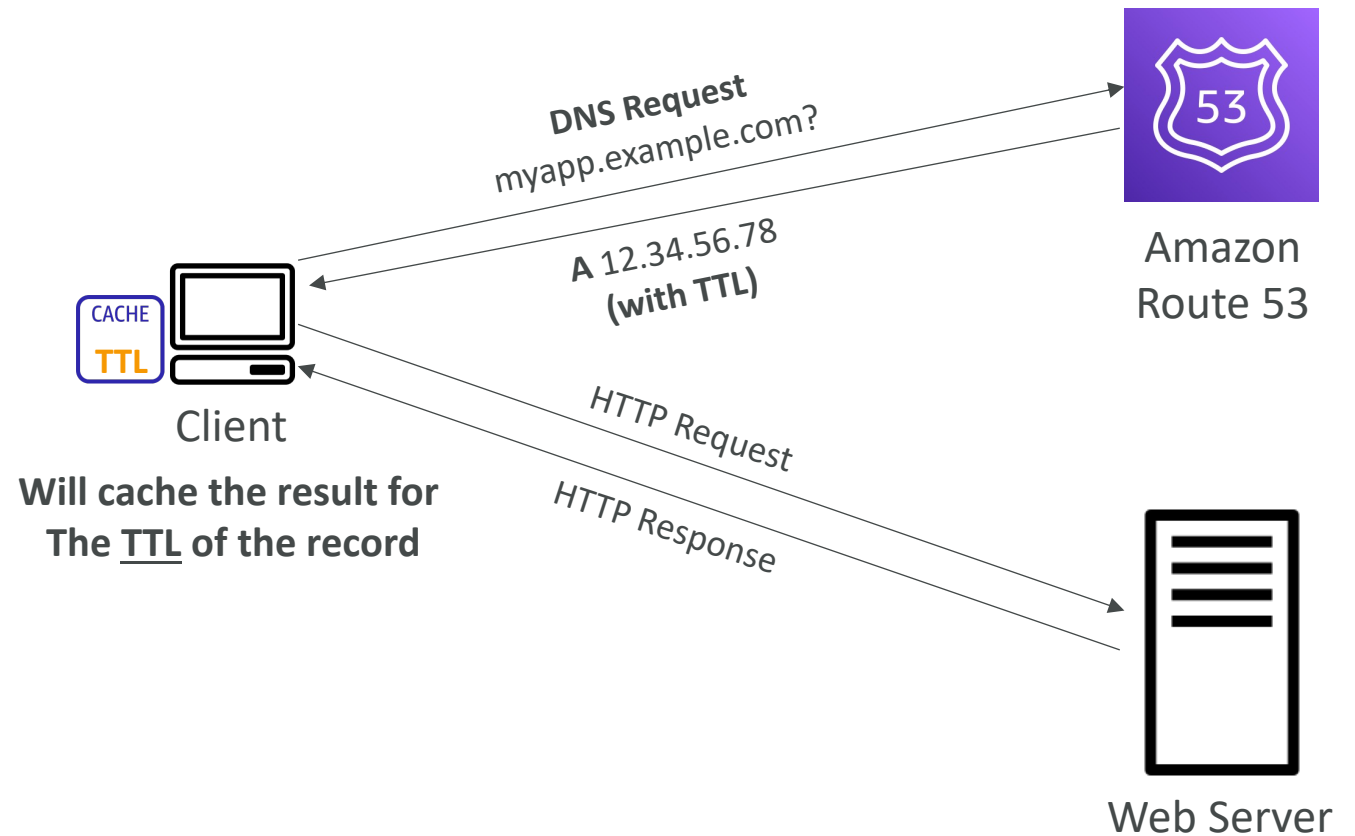
## Public Hosted Zone

## Private Hosted Zone



# Route 53 – Records TTL (Time To Live)

- High TTL – e.g., 24 hr
  - Less traffic on Route 53
  - Possibly outdated records
- Low TTL – e.g., 60 sec.
  - 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

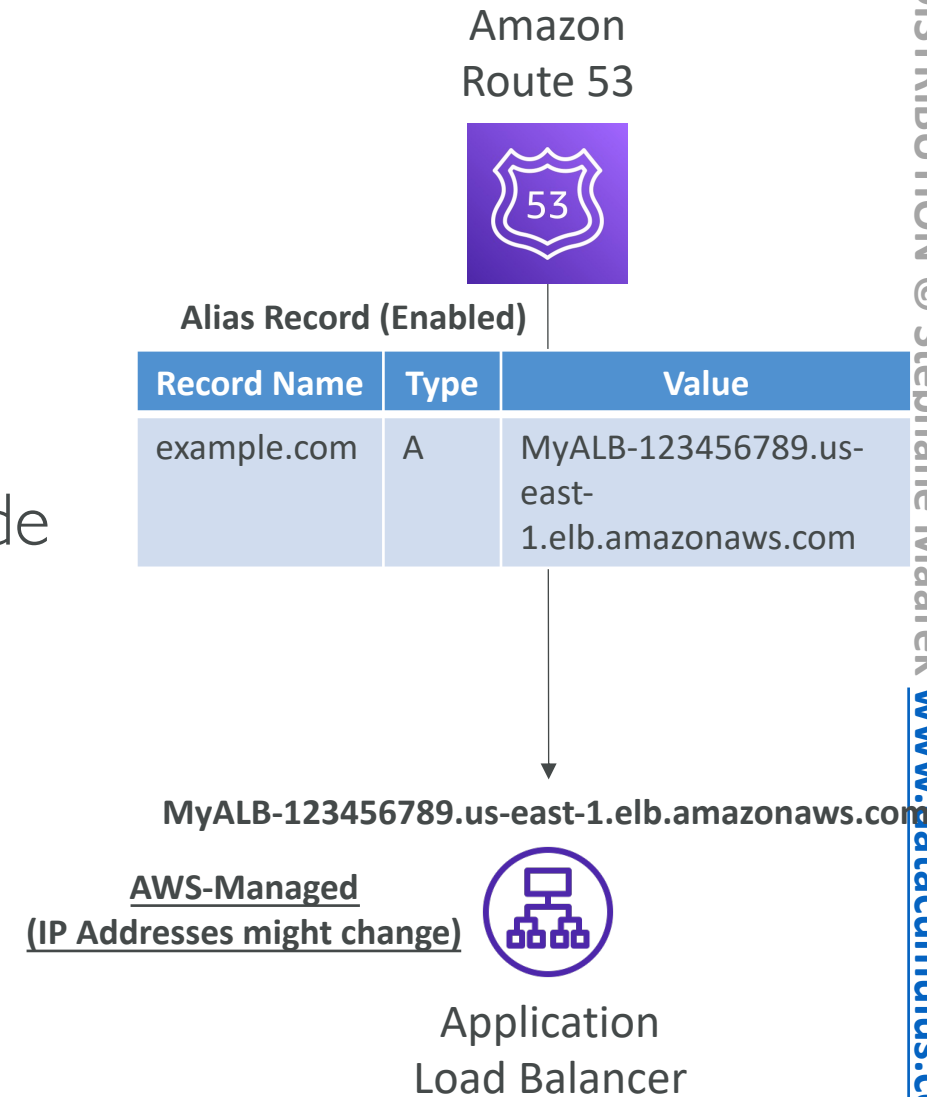


# CNAME vs Alias

- AWS Resources (Load Balancer, CloudFront...) expose an AWS hostname:
  - [lb1-1234.us-east-2.elb.amazonaws.com](#) and you want [myapp.mydomain.com](#)
- CNAME:
  - Points a hostname to any other hostname. (app.mydomain.com => blabla.anything.com)
  - ONLY FOR NON ROOT DOMAIN (aka. something.mydomain.com)
- Alias:
  - Points a hostname to an AWS Resource (app.mydomain.com => blabla.amazonaws.com)
  - Works for ROOT DOMAIN and NON ROOT DOMAIN (aka mydomain.com)
  - Free of charge
  - Native health check

# Route 53 – Alias Records

- Maps a hostname to an AWS resource
- An extension to DNS functionality
- Automatically recognizes changes in the resource's IP addresses
- Unlike CNAME, it can be used for the top node of a DNS namespace (Zone Apex), e.g.: example.com
- Alias Record is always of type A/AAAA for AWS resources (IPv4 / IPv6)
- You can't set the TTL



# Route 53 – Alias Records Targets

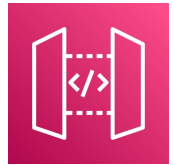
- Elastic Load Balancers
- CloudFront Distributions
- API Gateway
- Elastic Beanstalk environments
- S3 Websites
- VPC Interface Endpoints
- Global Accelerator accelerator
- Route 53 record in the same hosted zone



**Elastic  
Load Balancer**



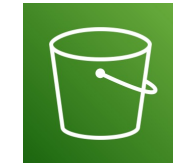
**Amazon  
CloudFront**



**Amazon  
API Gateway**



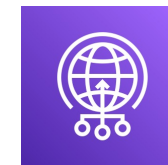
**Elastic Beanstalk**



**S3 Websites**



**VPC Interface  
Endpoints**



**Global Accelerator**



**Route 53 Record  
(same Hosted Zone)**

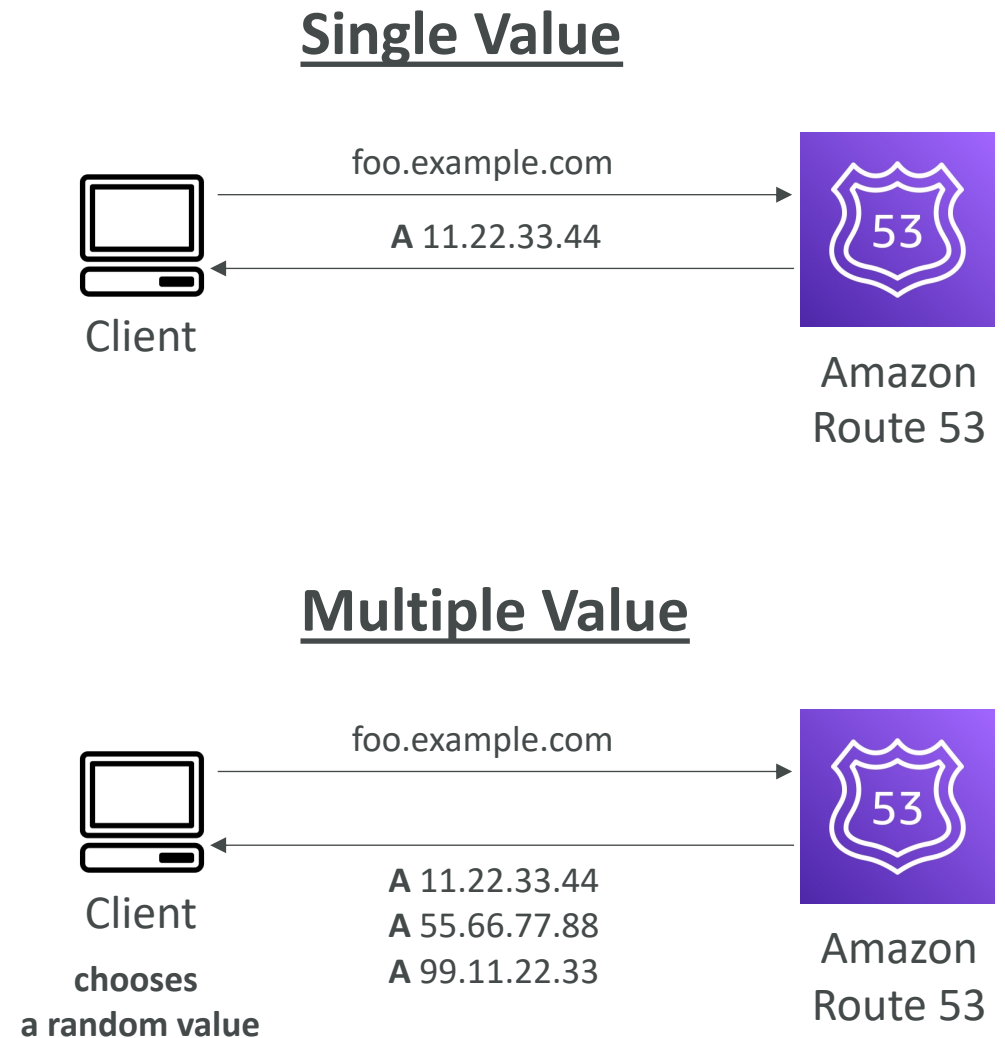
- You cannot set an ALIAS record for an EC2 DNS name

# Route 53 – Routing Policies

- Define how Route 53 responds to DNS queries
- Don't get confused by the word "*Routing*"
  - It's not the same as Load balancer routing which routes the traffic
  - DNS does not route any traffic, it only responds to the DNS queries
- Route 53 Supports the following Routing Policies
  - Simple
  - Weighted
  - Failover
  - Latency based
  - Geolocation
  - Multi-Value Answer
  - Geoproximity (using Route 53 Traffic Flow feature)

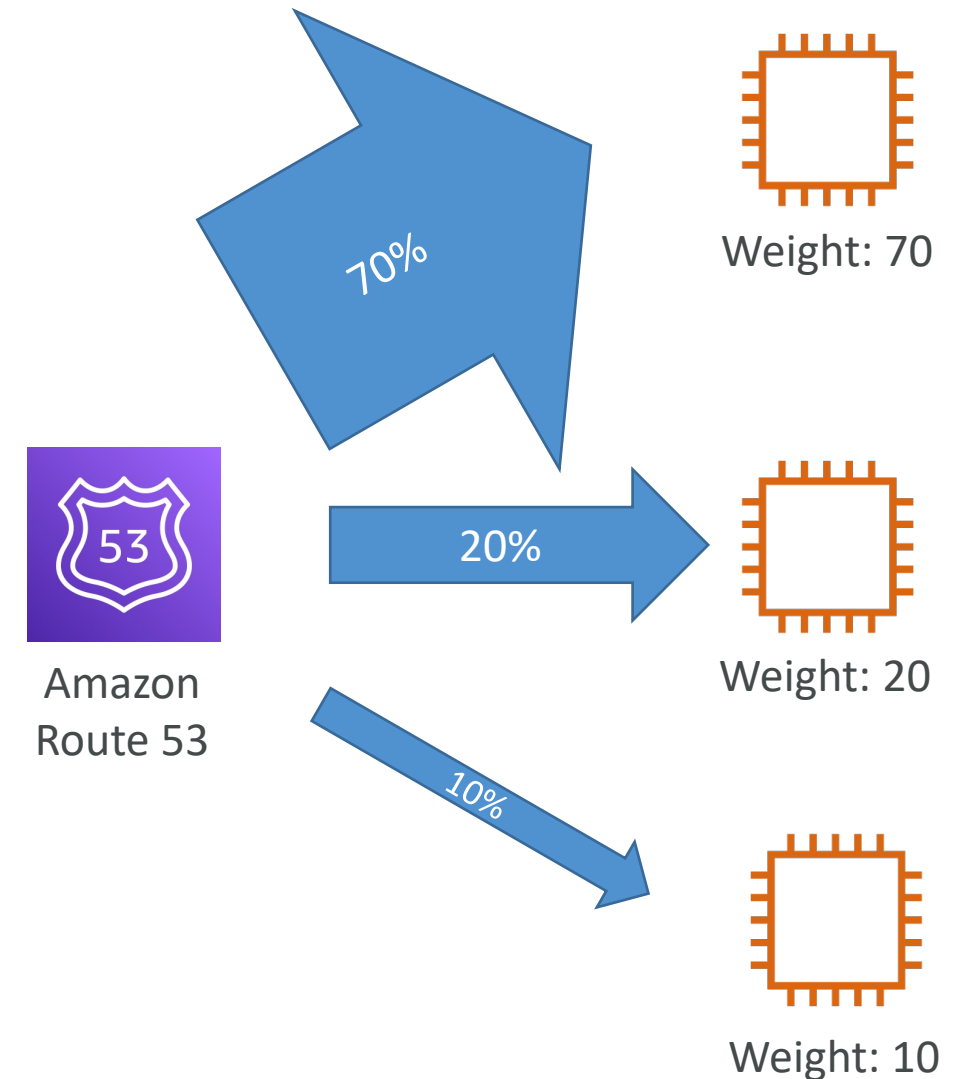
# Routing Policies – Simple

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



# Routing Policies – Weighted

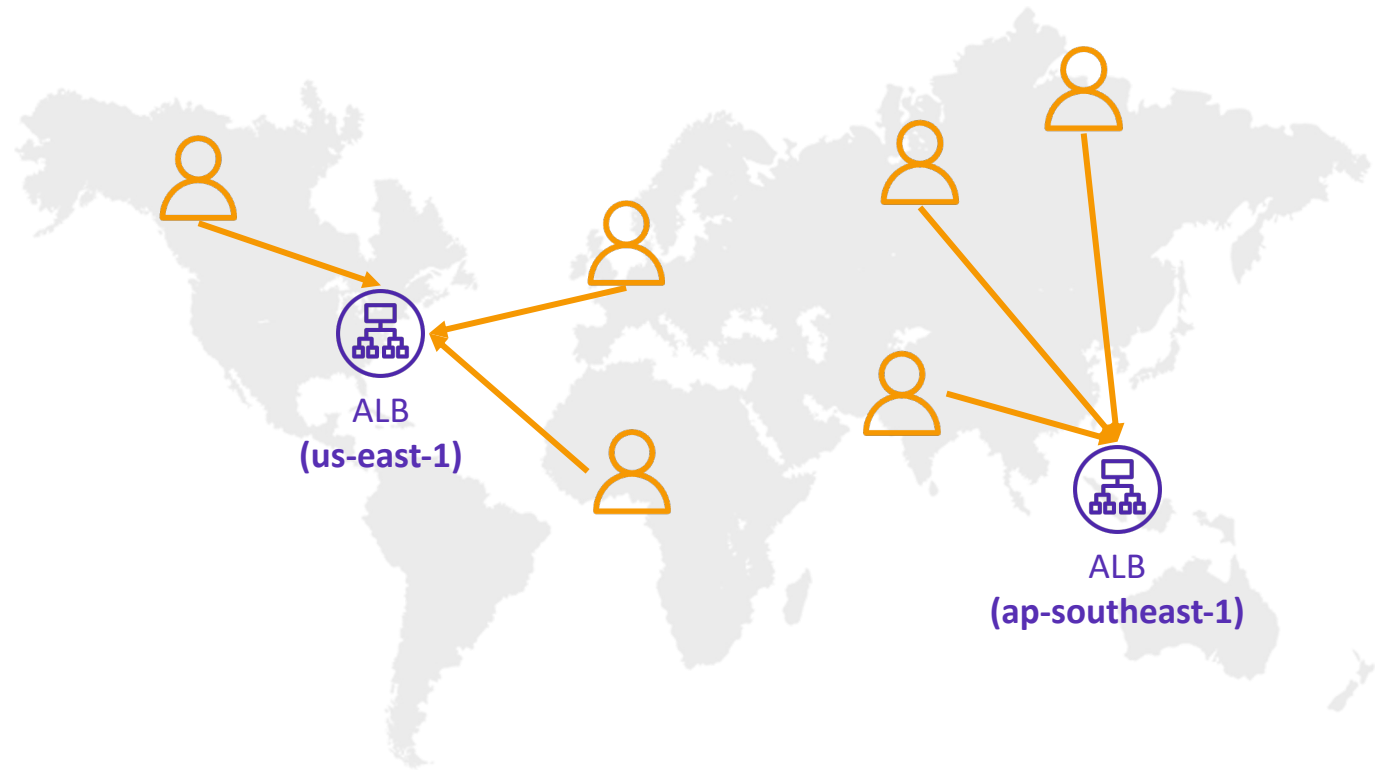
- Control the % of the requests that go to each specific resource
- Assign each record a relative weight:
  - $traffic\ (\%) = \frac{Weight\ for\ a\ specific\ record}{Sum\ of\ all\ the\ weights\ for\ all\ records}$
  - Weights don't need to sum up to 100
- DNS records must have the same name and type
- Can be associated with Health Checks
- Use cases: load balancing between regions, testing new application versions...
- Assign a weight of 0 to a record to stop sending traffic to a resource
- If all records have weight of 0, then all records will be returned equally





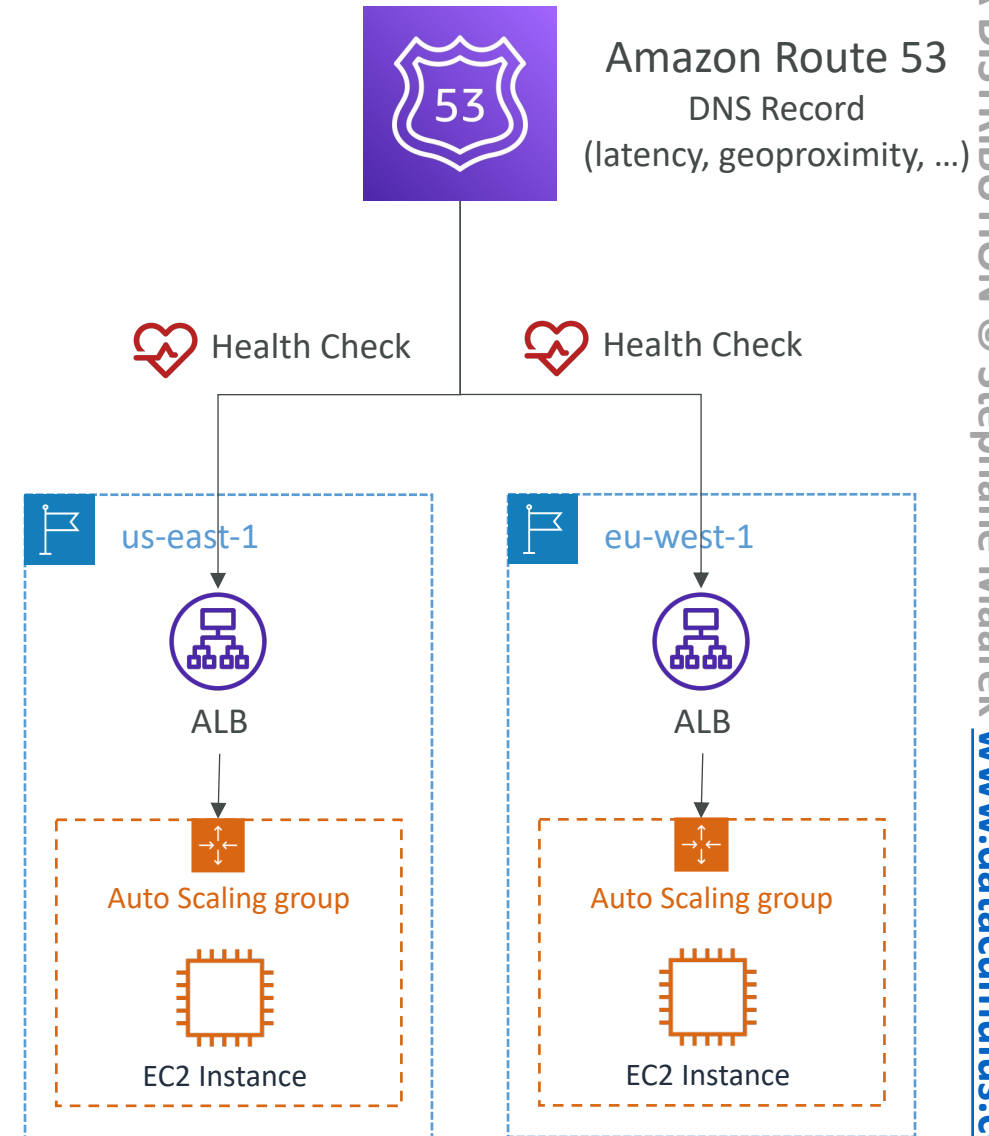
# Routing Policies – Latency-based

- Redirect to the resource that has the least latency close to us
- Super helpful when latency for users is a priority
- Latency is based on traffic between users and AWS Regions
- Germany users may be directed to the US (if that's the lowest latency)
- Can be associated with Health Checks (has a failover capability)



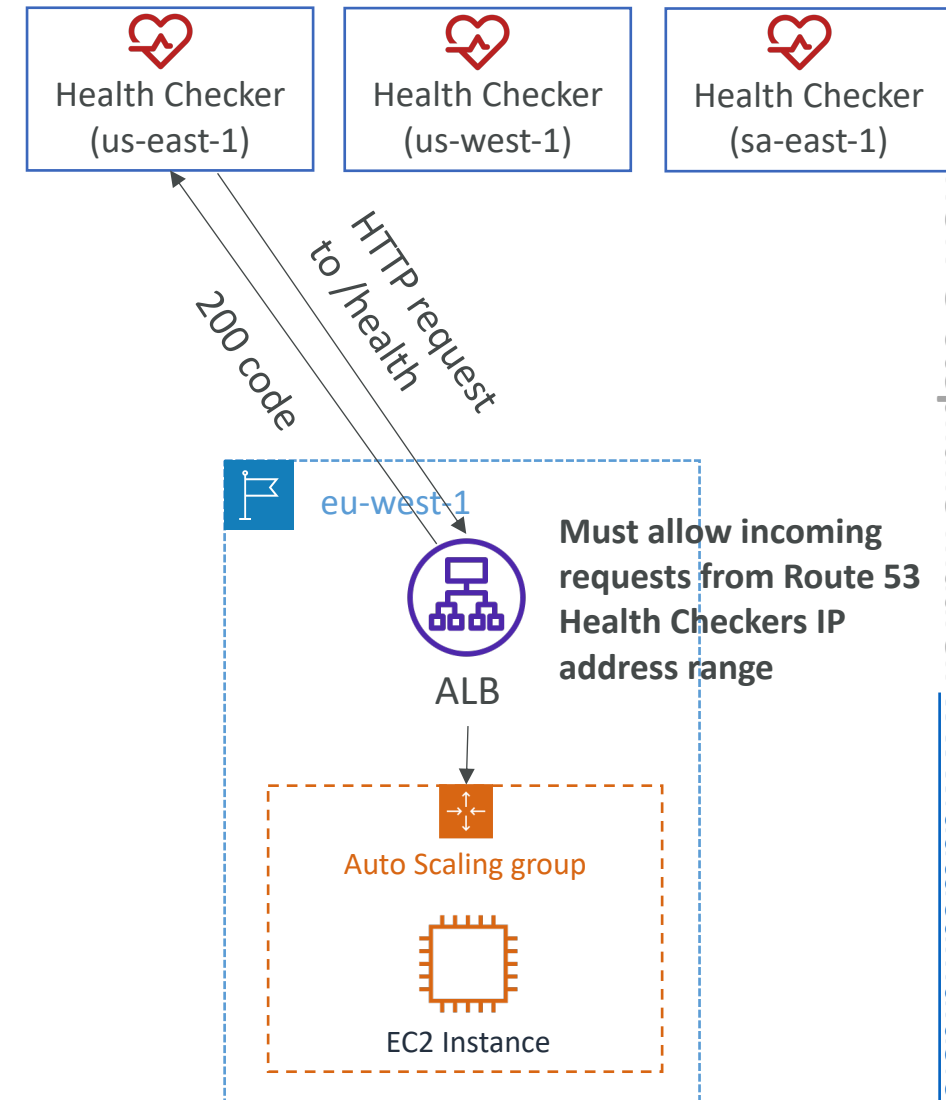
# Route 53 – Health Checks

- HTTP Health Checks are only for **public resources**
- Health Check => Automated DNS Failover:
  1. Health checks that monitor an endpoint (application, server, other AWS resource)
  2. Health checks that monitor other health checks (Calculated Health Checks)
  3. Health checks that monitor CloudWatch Alarms (full control !!) – e.g., throttles of DynamoDB, alarms on RDS, custom metrics, ... (helpful for private resources)
- Health Checks are integrated with CW metrics



# Health Checks – Monitor an Endpoint

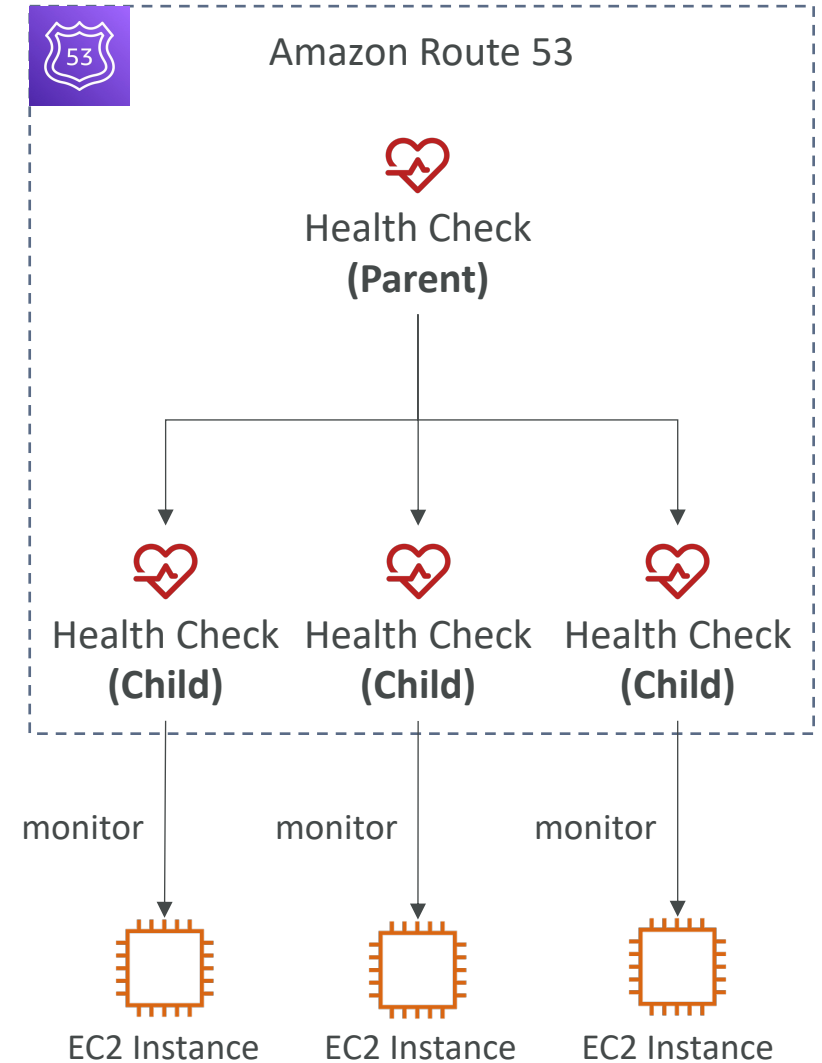
- About 15 global health checkers will check the endpoint health
  - Healthy/Unhealthy Threshold – 3 (default)
  - Interval – 30 sec (can set to 10 sec – higher cost)
  - Supported protocol: HTTP, HTTPS and TCP
  - If > 18% of health checkers report the endpoint is healthy, Route 53 considers it **Healthy**. Otherwise, it's **Unhealthy**
  - Ability to choose which locations you want Route 53 to use
- Health Checks pass only when the endpoint responds with the 2xx and 3xx status codes
- Health Checks can be setup to pass / fail based on the text in the first **5120 bytes** of the response
- Configure your router/firewall to allow incoming requests from Route 53 Health Checkers



<https://ip-ranges.amazonaws.com/ip-ranges.json>

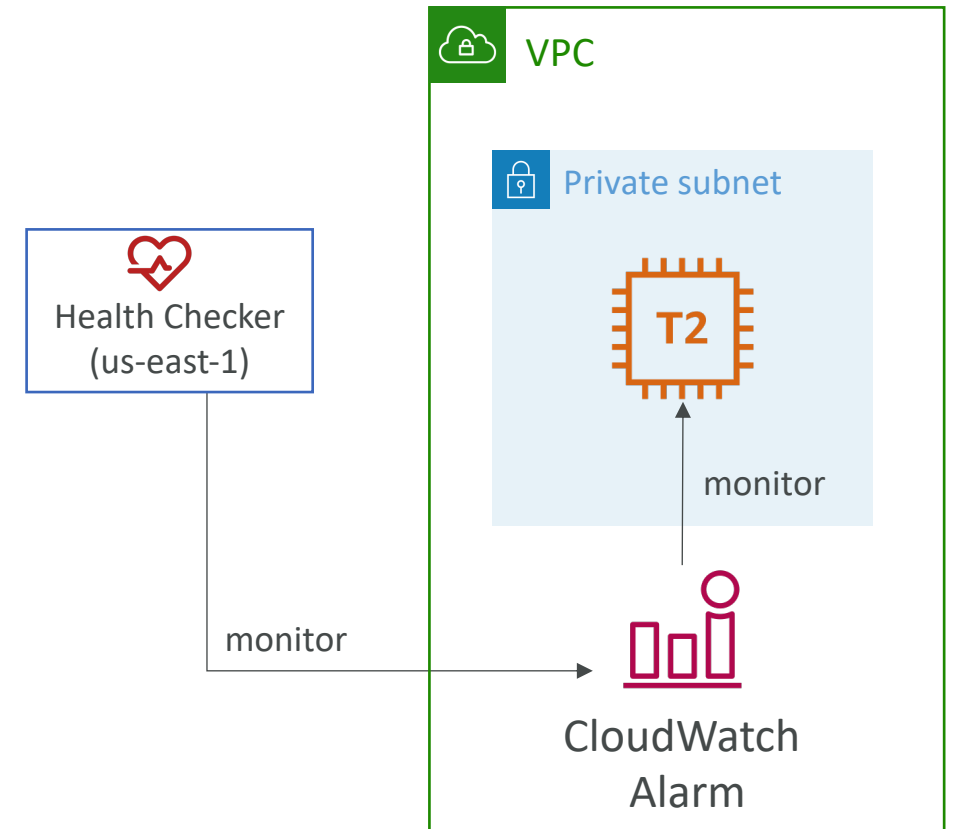
# Route 53 – Calculated Health Checks

- Combine the results of multiple Health Checks into a single Health Check
- You can use **OR**, **AND**, or **NOT**
- Can monitor up to 256 Child Health Checks
- Specify how many of the health checks need to pass to make the parent pass
- Usage: perform maintenance to your website without causing all health checks to fail

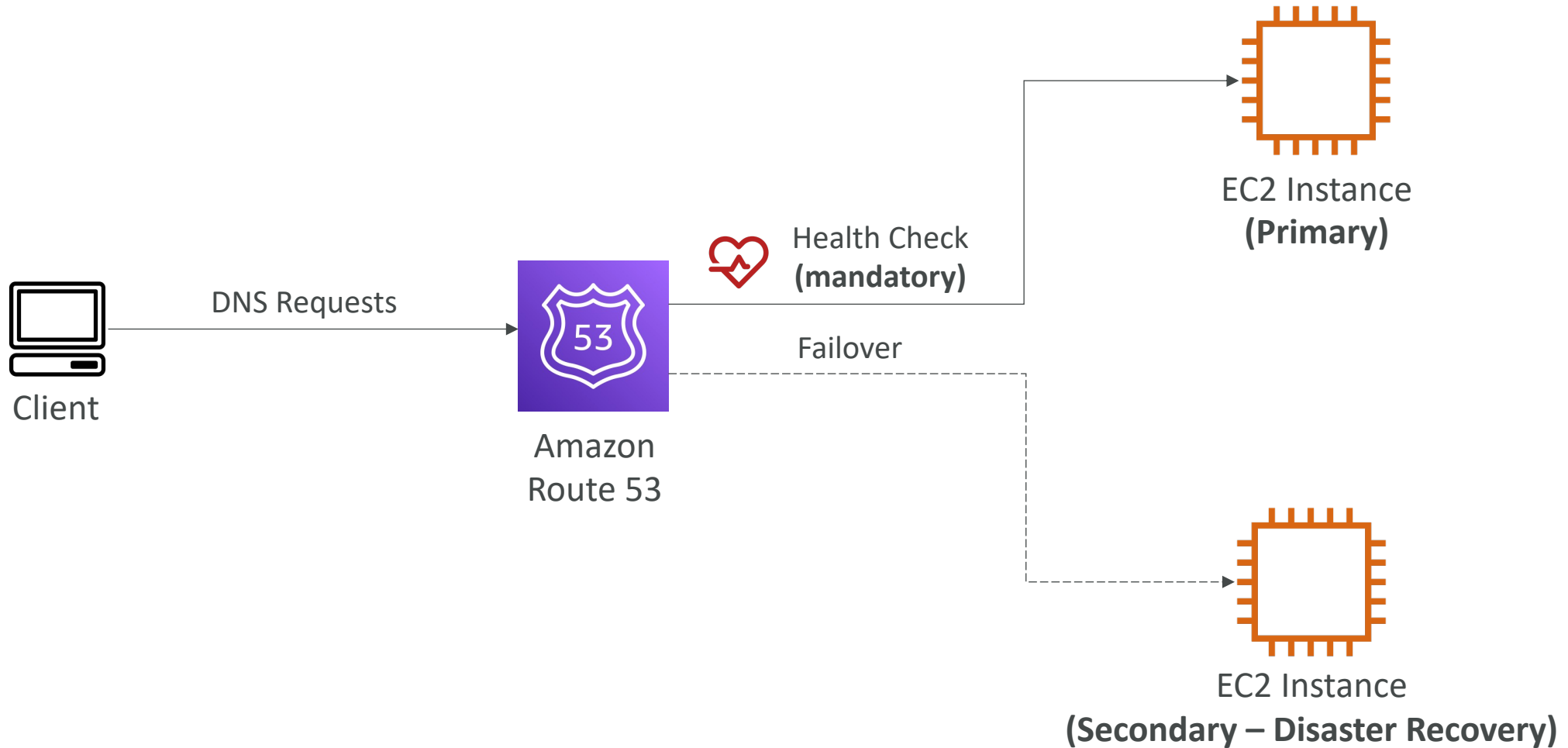


# Health Checks – Private Hosted Zones

- Route 53 health checkers are outside the VPC
- They can't access **private** endpoints (private VPC or on-premises resource)
- You can create a **CloudWatch Metric** and associate a **CloudWatch Alarm**, then create a Health Check that checks the alarm itself

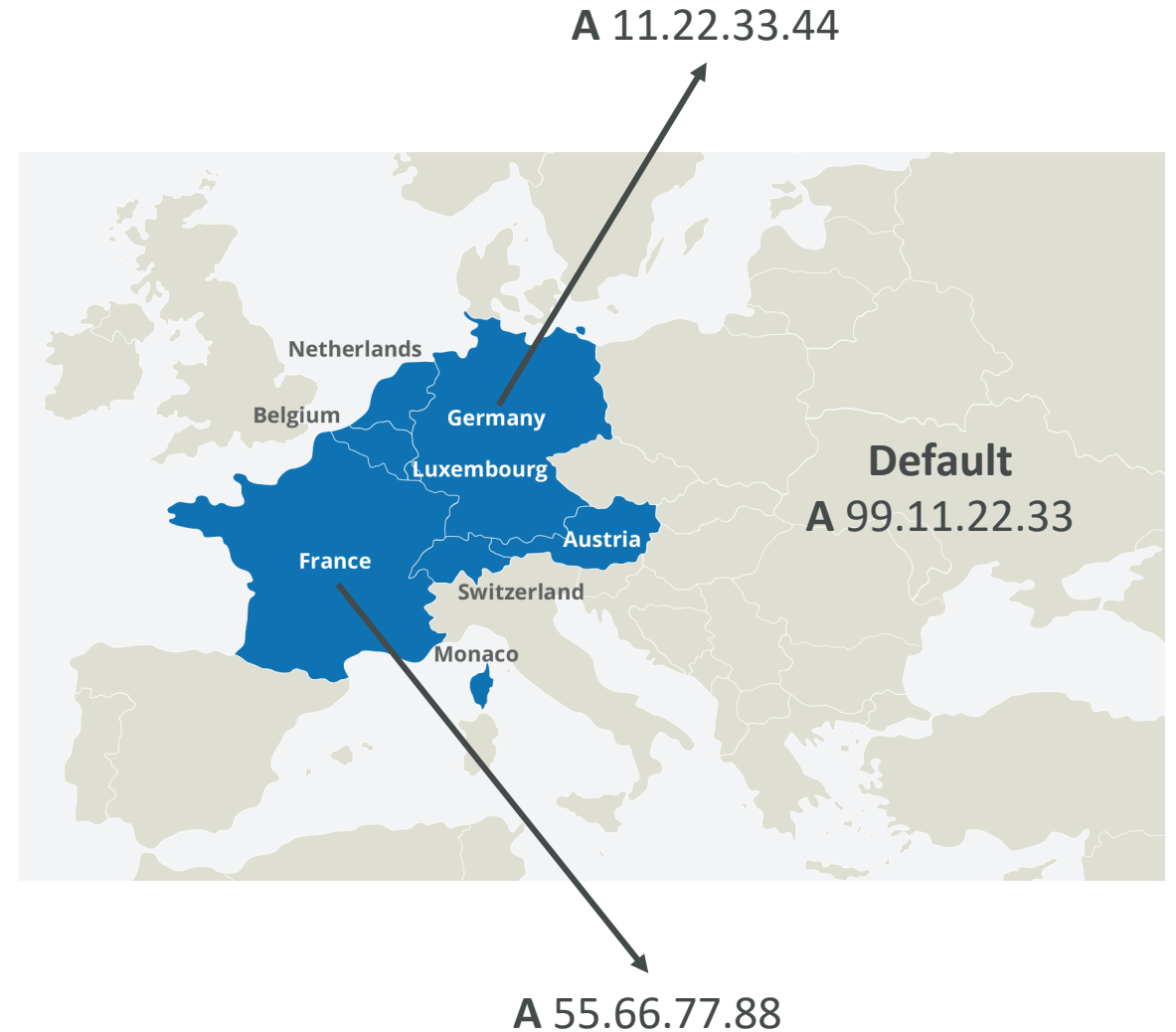


# Routing Policies – Failover (Active-Passive)



# Routing Policies – Geolocation

- Different from Latency-based!
- This routing is based on user location
- Specify location by Continent, Country or by US State (if there's overlapping, most precise location selected)
- Should create a “**Default**” record (in case there's no match on location)
- Use cases: website localization, restrict content distribution, load balancing, ...
- Can be associated with Health Checks

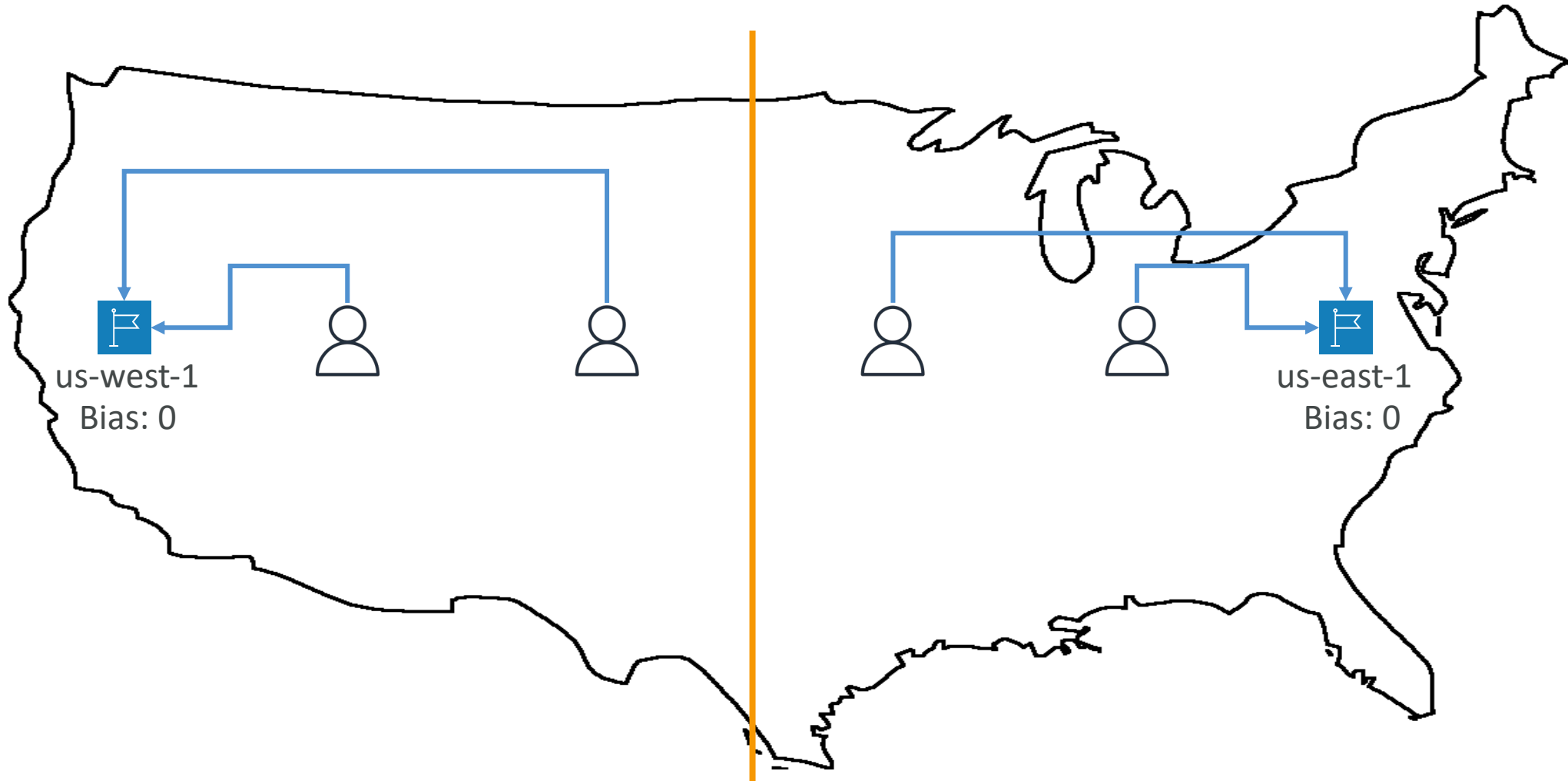


# Routing Policies – Geoproximity

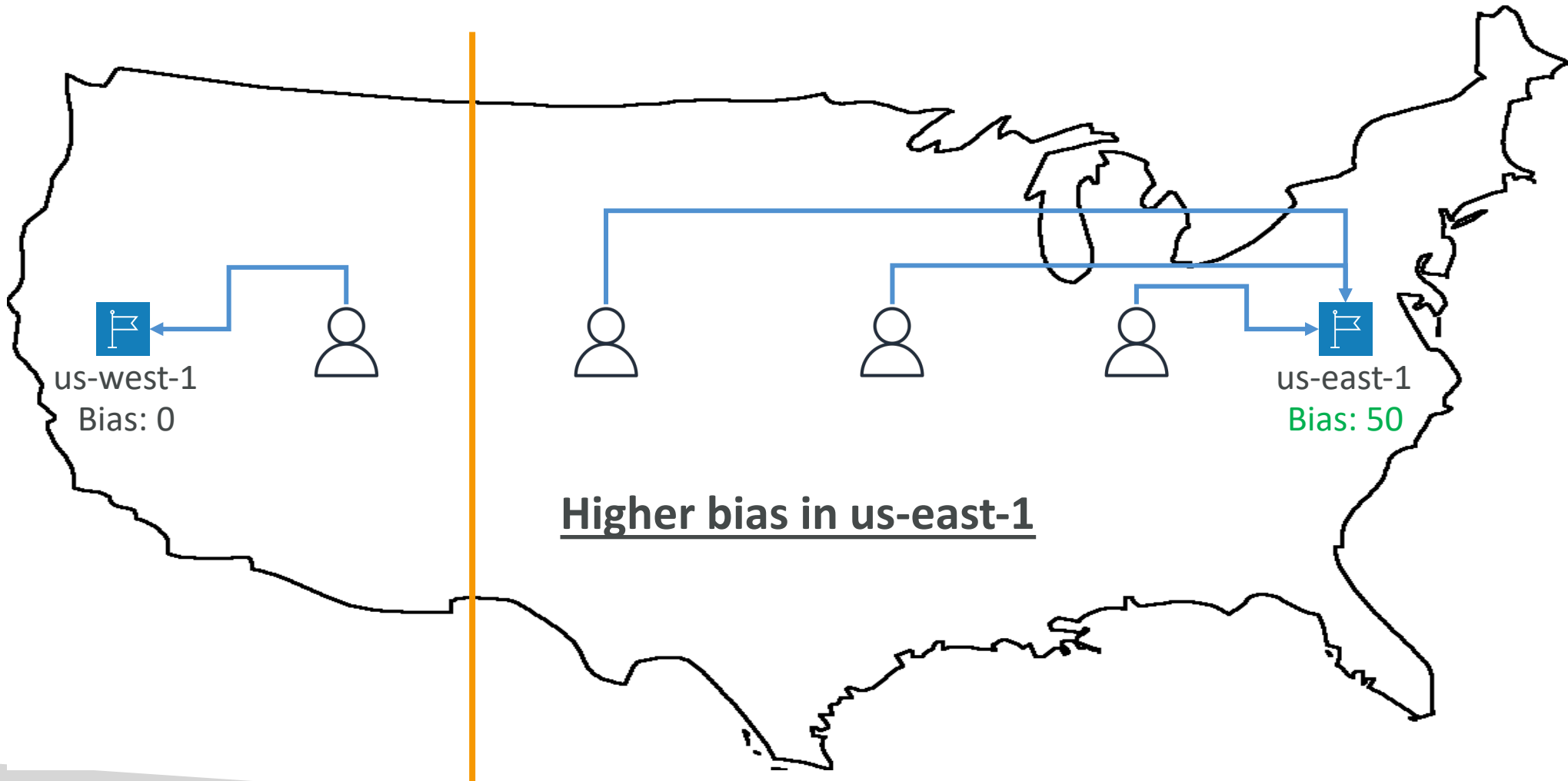
- Route traffic to your resources based on the geographic location of users and resources
- Ability to shift more traffic to resources based on the defined bias
- To change the size of the geographic region, specify **bias** values:
  - To expand (1 to 99) – more traffic to the resource
  - To shrink (-1 to -99) – less traffic to the resource
- Resources can be:
  - AWS resources (specify AWS region)
  - Non-AWS resources (specify Latitude and Longitude)
- You must use Route 53 **Traffic Flow** to use this feature



# Routing Policies – Geoproximity

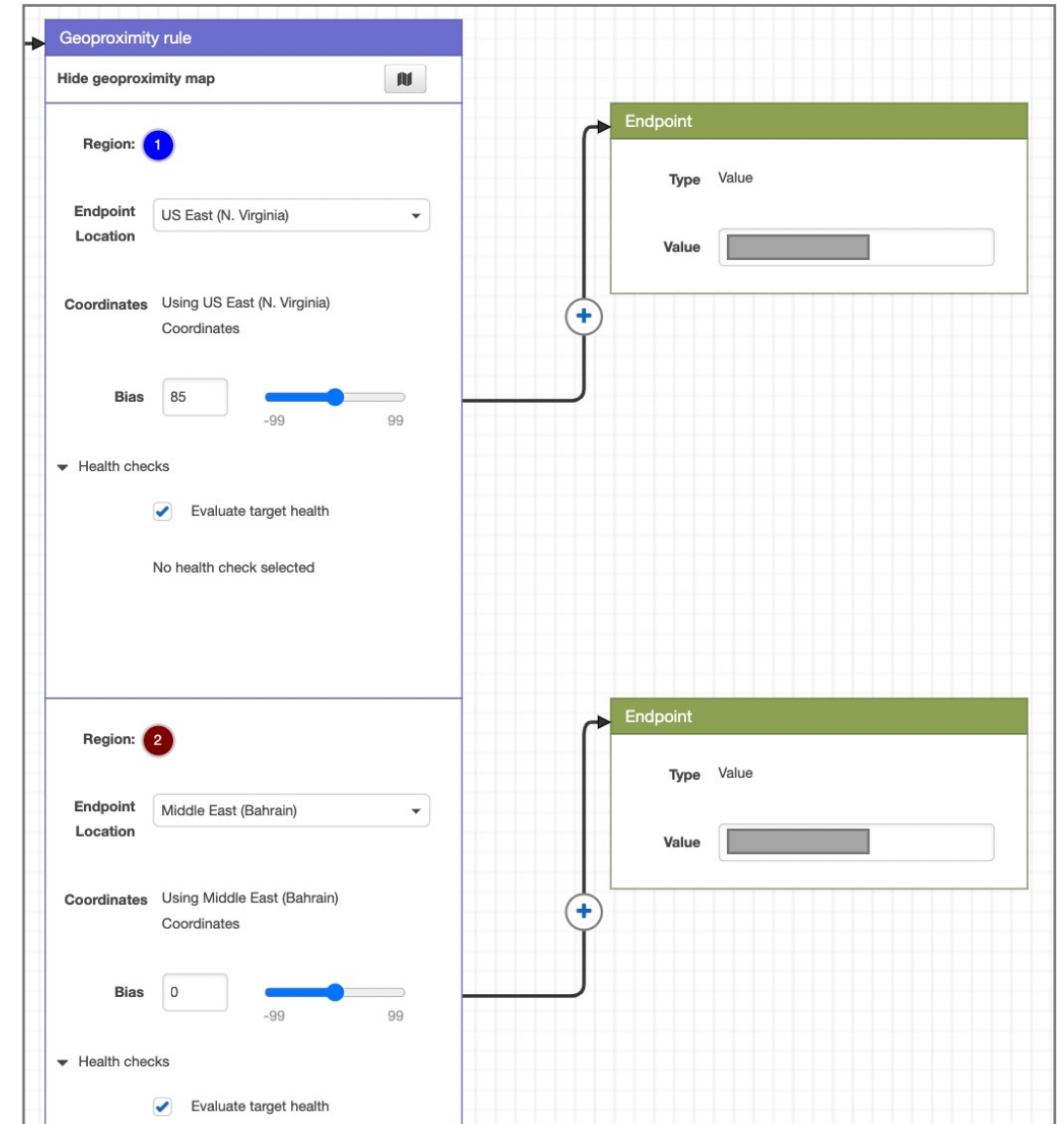


# Routing Policies – Geoproximity



# Route 53 – Traffic flow

- Simplify the process of creating and maintaining records in large and complex configurations
- Visual editor to manage complex routing decision trees
- Configurations can be saved as **Traffic Flow Policy**
  - Can be applied to different Route 53 Hosted Zones (different domain names)
  - Supports versioning



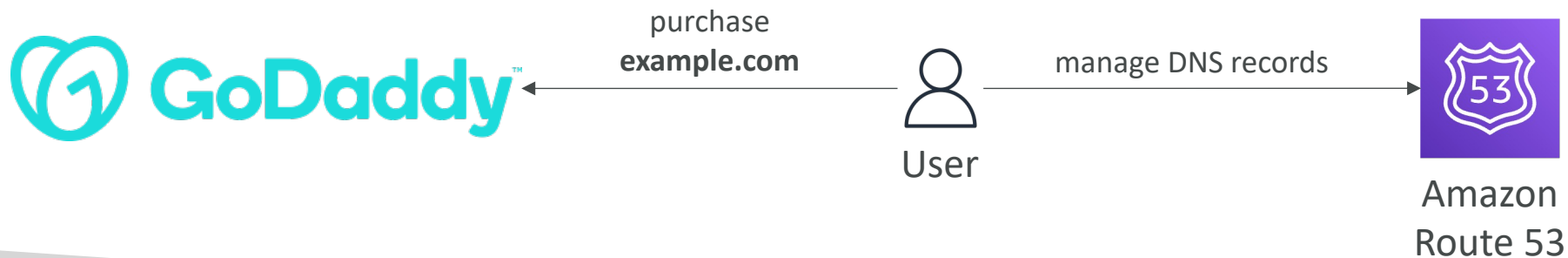
# Routing Policies – Multi-Value

- Use when routing traffic to multiple resources
- Route 53 return multiple values/resources
- Can be associated with Health Checks (return only values for healthy resources)
- Up to 8 healthy records are returned for each Multi-Value query
- Multi-Value is not a substitute for having an ELB

Name	Type	Value	TTL	Set ID	Health Check
www.example.com	A Record	192.0.2.2	60	Web1	A
www.example.com	A Record	198.51.100.2	60	Web2	B
www.example.com	A Record	203.0.113.2	60	Web3	C

# Domain Registrar vs. DNS Service

- You buy or register your domain name with a Domain Registrar typically by paying annual charges (e.g., GoDaddy, Amazon Registrar Inc., ...)
- The Domain Registrar usually provides you with a DNS service to manage your DNS records
- But you can use another DNS service to manage your DNS records
- Example: purchase the domain from GoDaddy and use Route 53 to manage your DNS records



# GoDaddy as Registrar & Route 53 as DNS Service



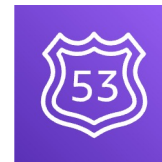
## Records

We can't display your DNS information because your nameservers aren't managed by us.

## Nameservers

Using custom nameservers [Change](#)

Nameserver
ns-1083.awsdns-07.org
ns-932.awsdns-52.net
ns-1911.awsdns-46.co.uk
ns-481.awsdns-60.com



Amazon  
Route 53

Public Hosted Zone  
stephanetheteacher.com

▼ Hosted zone details [Edit hosted zone](#)

Hosted zone ID Z30IUJCCWPKZUV	Type Public hosted zone
Description HostedZone created by Route53 Registrar	Record count 22
Query log	Name servers ns-252.awsdns-31.com ns-1468.awsdns-55.org ns-633.awsdns-15.net ns-1800.awsdns-33.co.uk

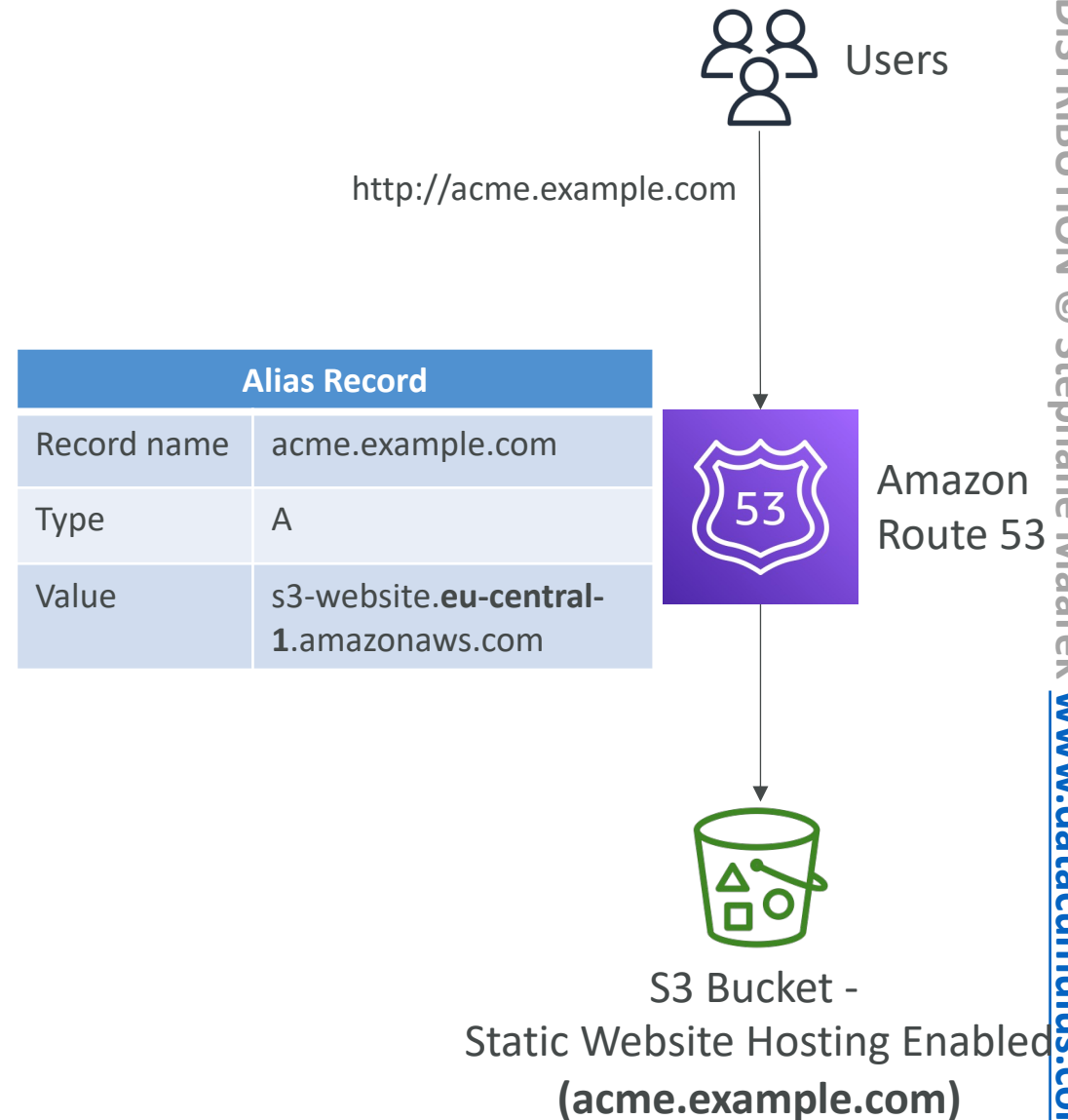
# 3<sup>rd</sup> Party Registrar with Amazon Route 53

- If you buy your domain on a 3<sup>rd</sup> party registrar, you can still use Route 53 as the DNS Service provider
  1. Create a Hosted Zone in Route 53
  2. Update NS Records on 3<sup>rd</sup> party website to use Route 53 **Name Servers**
- Domain Registrar  $\neq$  DNS Service
- But every Domain Registrar usually comes with some DNS features

# S3 Website with Route 53

For acme.example.com:

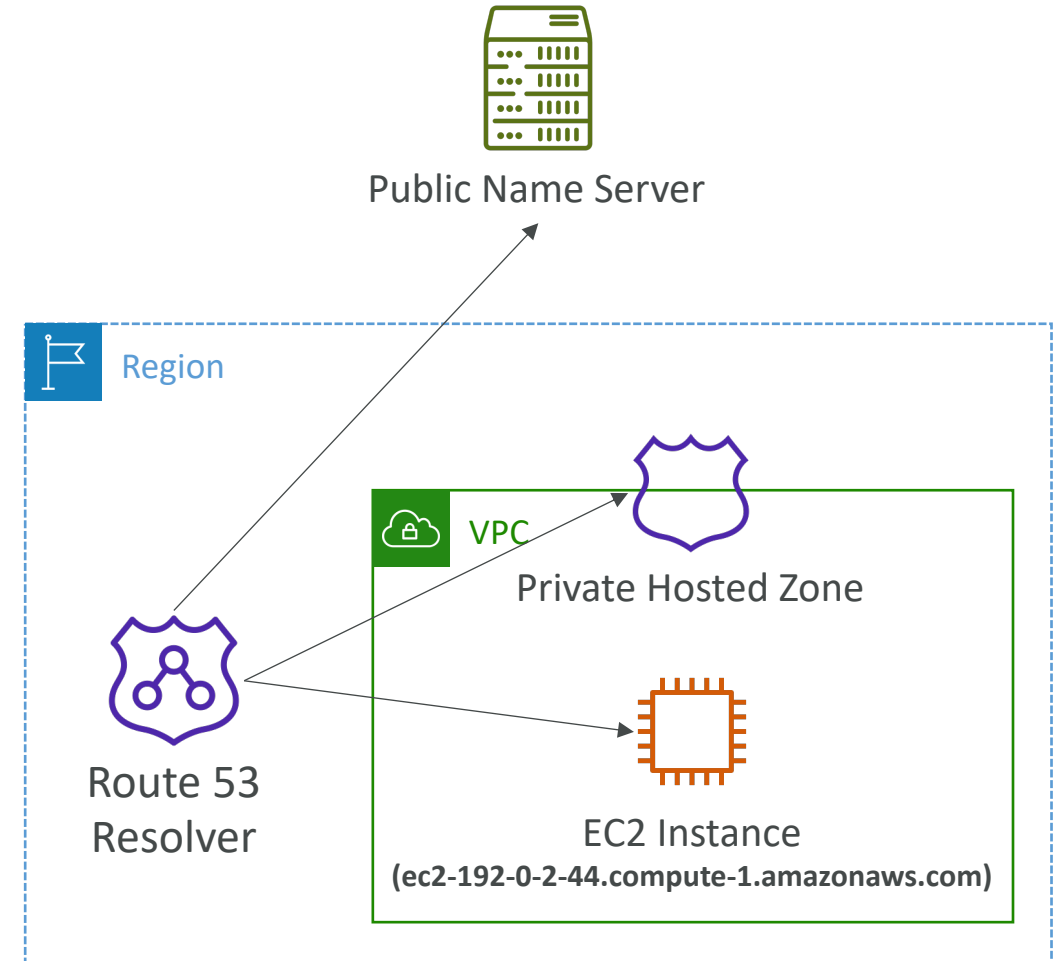
- Create an S3 bucket with the **same name** as the target record (acme.example.com)
- Enable S3 website on the bucket (and enable S3 bucket public settings)
- Create a Route 53 **Alias record** to the S3 website endpoint or type A – IPv4 address
- This only works for HTTP traffic (for HTTPS, use CloudFront)





# Route 53 – Hybrid DNS

- By default, Route 53 Resolver automatically answers DNS queries for:
  - Local domain names for EC2 instances
  - Records in Private Hosted Zones
  - Records in public Name Servers
- **Hybrid DNS** – resolving DNS queries between VPC (Route 53 Resolver) and your networks (other DNS Resolvers)
- Networks can be:
  - VPC itself / Peered VPC
  - On-premises Network (connected through Direct Connect or AWS VPN)



# Route 53 – Resolver Endpoints

- **Inbound Endpoint**

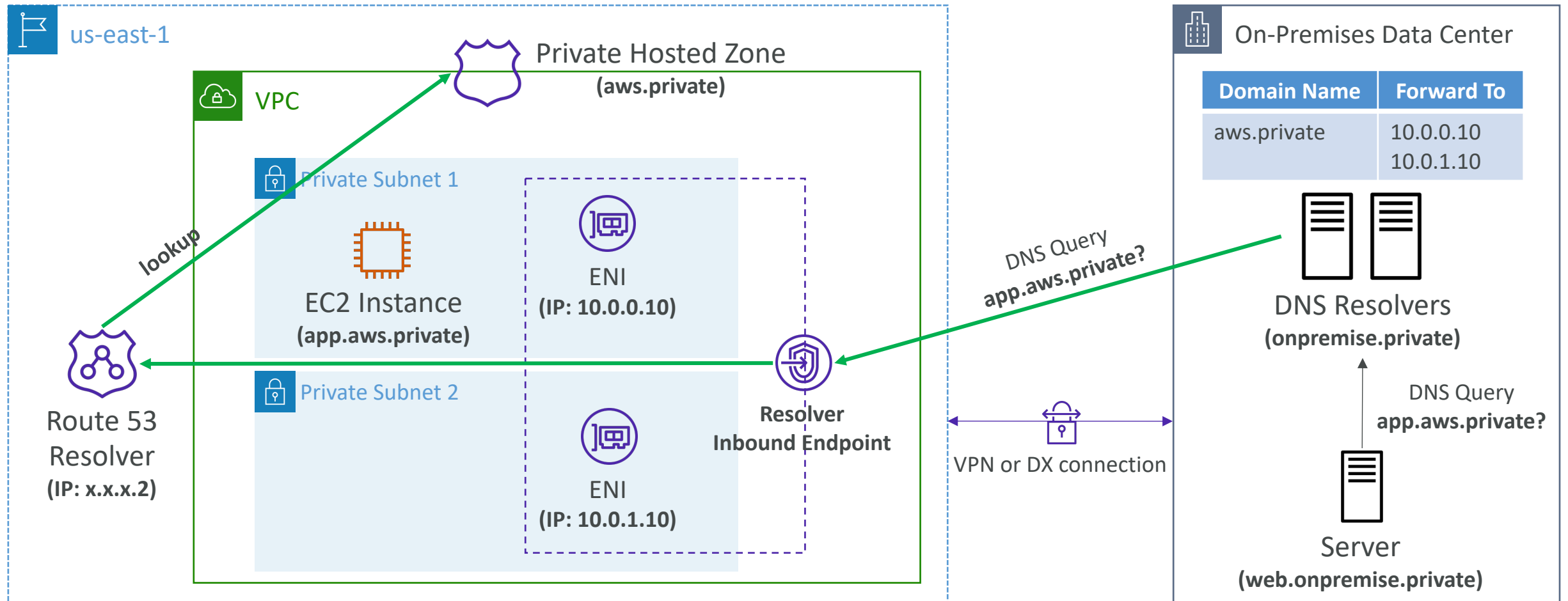
- DNS Resolvers on your network can forward DNS queries to Route 53 Resolver
- Allows your DNS Resolvers to resolve domain names for AWS resources (e.g., EC2 instances) and records in Route 53 Private Hosted Zones

- **Outbound Endpoint**

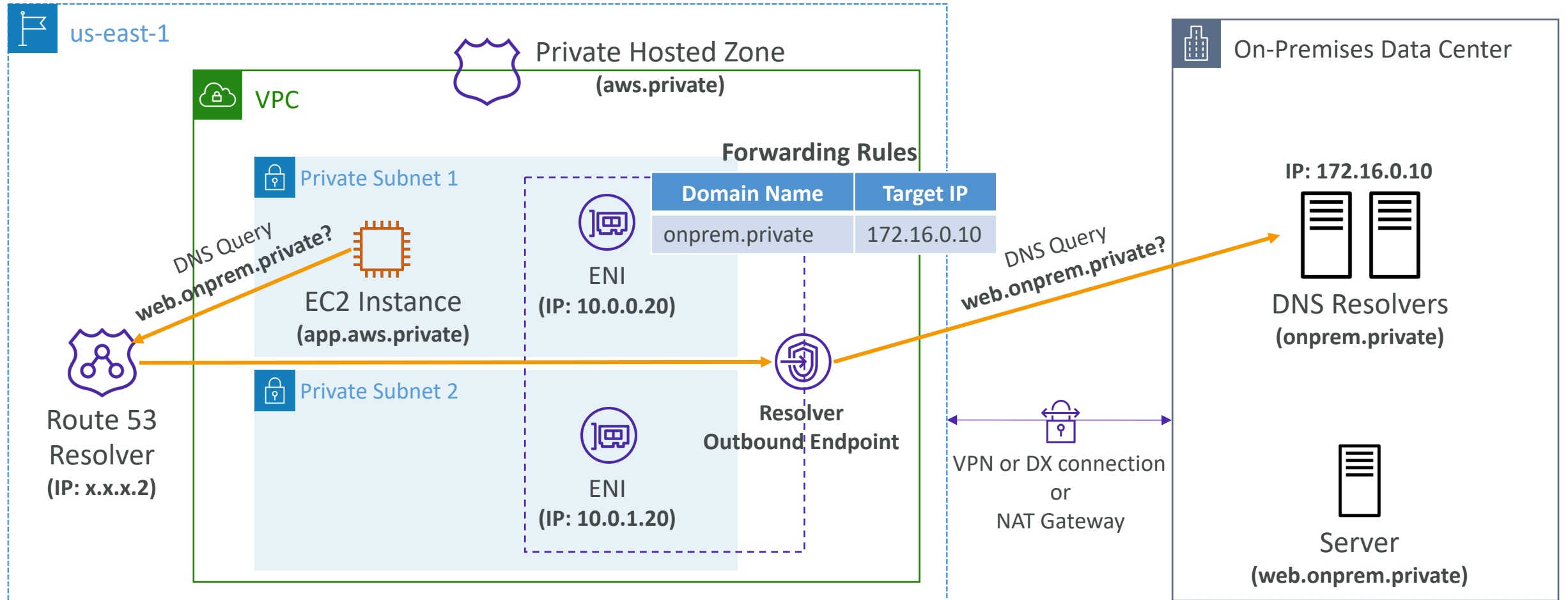
- Route 53 Resolver conditionally forwards DNS queries to your DNS Resolvers
- Use **Resolver Rules** to forward DNS queries to your DNS Resolvers

- Associated with one or more VPCs in the same AWS Region
- Create in two AZs for high availability
- Each Endpoint supports 10,000 queries per second per IP address

# Route 53 – Resolver Inbound Endpoints

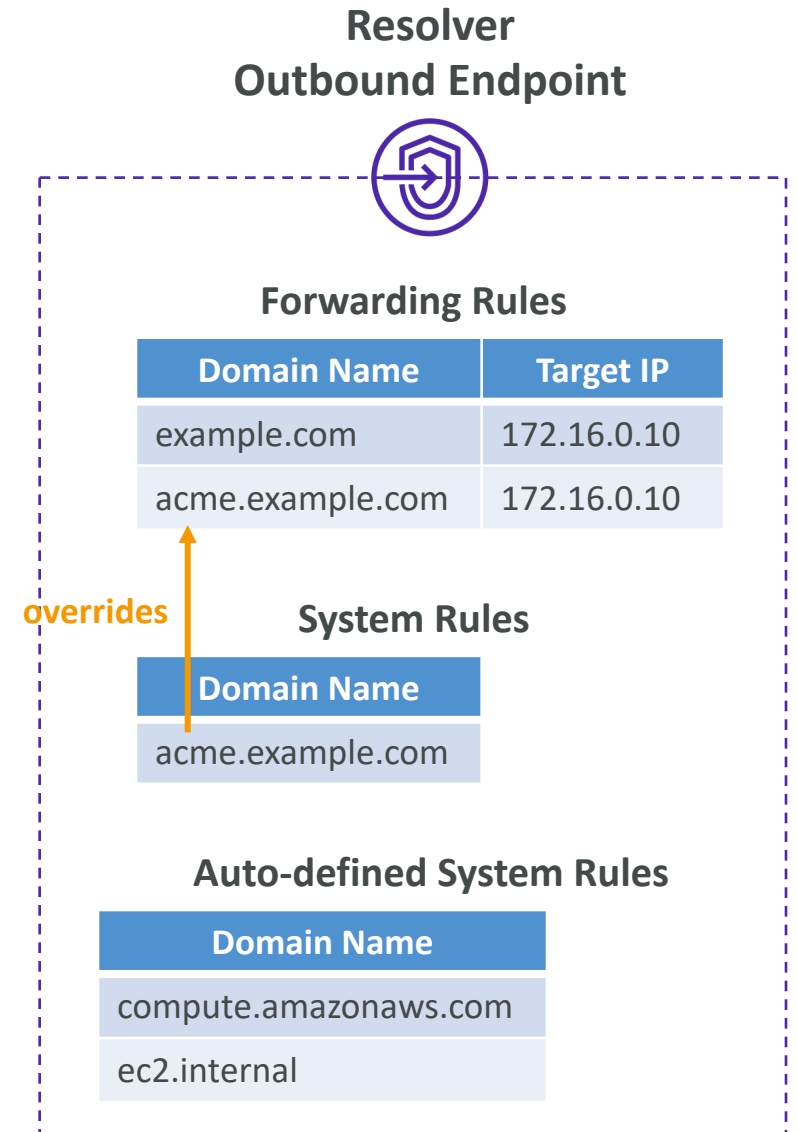


# Route 53 – Resolver Outbound Endpoints



# Route 53 – Resolver Rules

- Control which DNS queries are forwarded to DNS Resolvers on your network
- **Conditional Forwarding Rules (Forwarding Rules)**
  - Forward DNS queries for a specified domain and all its subdomains to **target IP addresses**
- **System Rules**
  - Selectively overriding the behavior defined in Forwarding Rules (e.g., don't forward DNS queries for a subdomain acme.example.com)
- **Auto-defined System Rules**
  - Defines how DNS queries for selected domains are resolved (e.g., AWS internal domain names, Privated Hosted Zones)
- If multiple rules matched, Route 53 Resolver chooses the most specific match



# Other Services

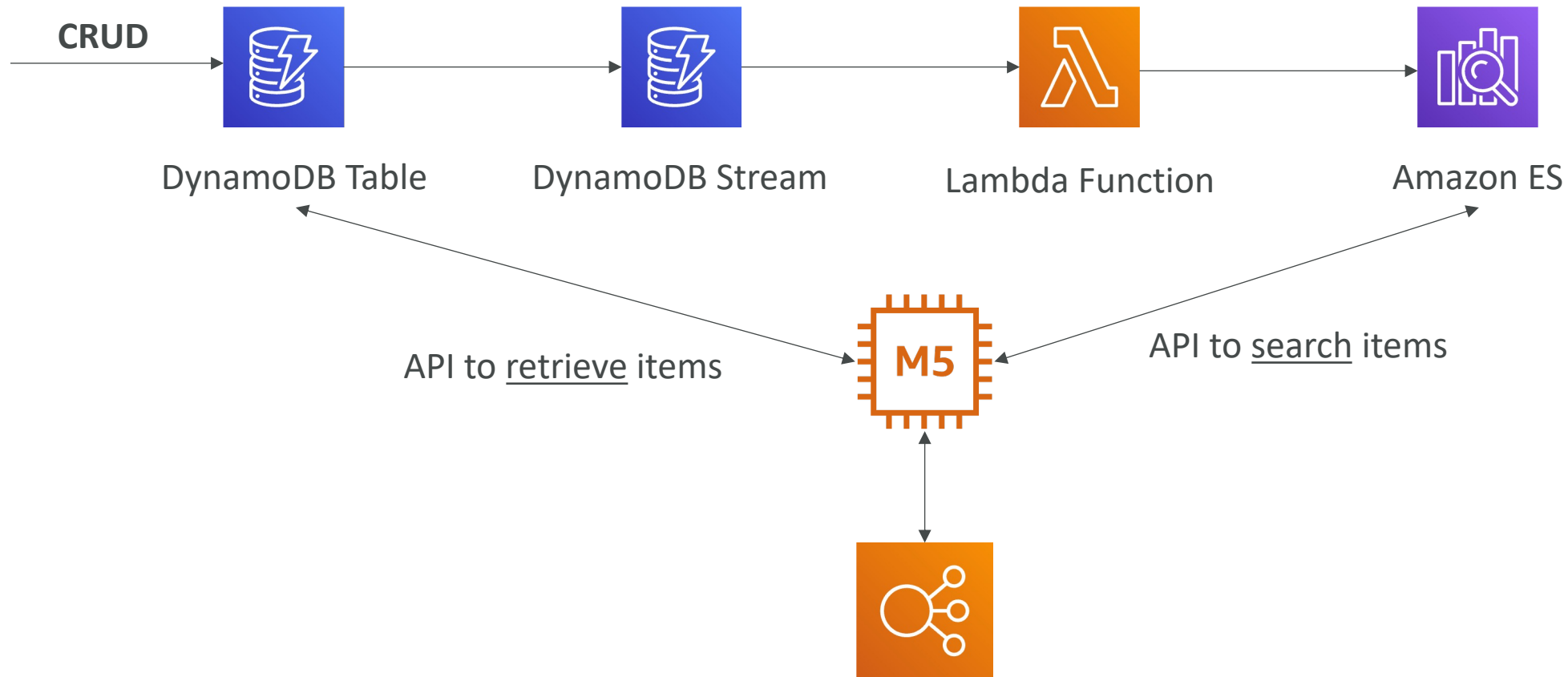
# Amazon ElasticSearch



- May be called Amazon ES at the exam
- Managed version of ElasticSearch (open source project)
- Needs to run on servers (not a serverless offering)
- Use cases:
  - Log Analytics
  - Real Time application monitoring
  - Security Analytics
  - Full Text Search
  - Clickstream Analytics
  - Indexing

# Elastic Search patterns

## DynamoDB



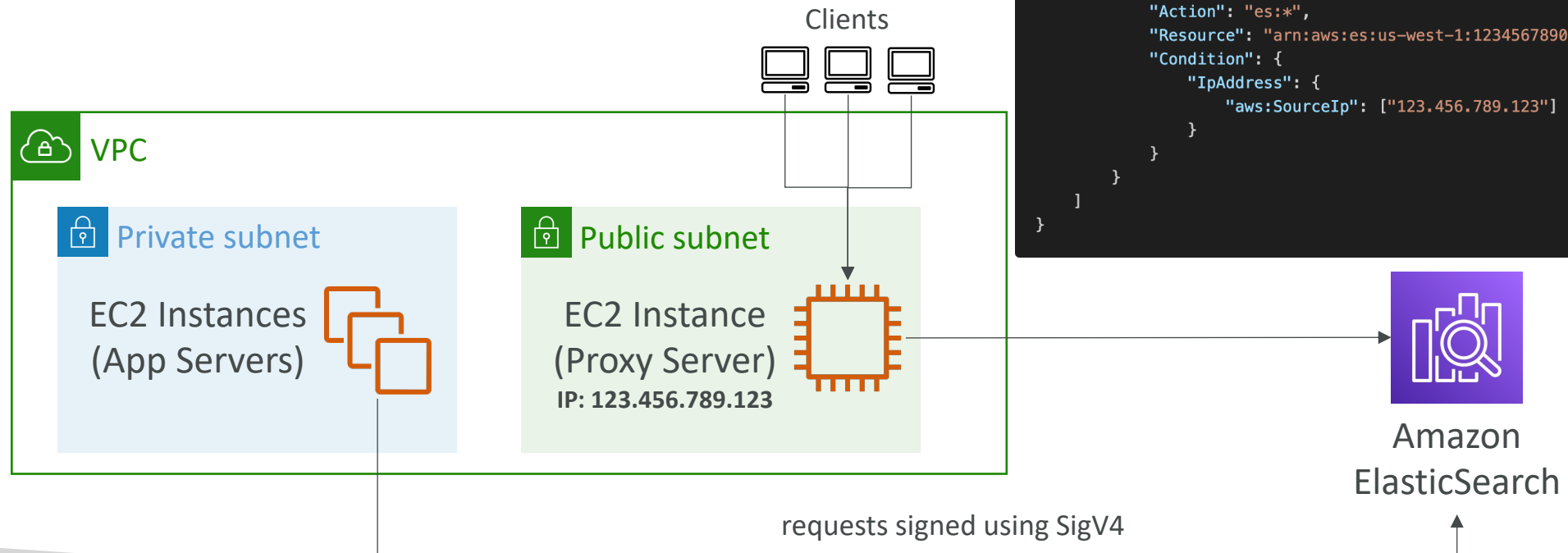


# ElasticSearch Access Policy

- IP-based Policies

- Resource-based policies used to restrict access to an ES domain to IP address(es) or CIDR blocks
- Allows unsigned requests to an ES domain (e.g., curl, Kibana, ...)

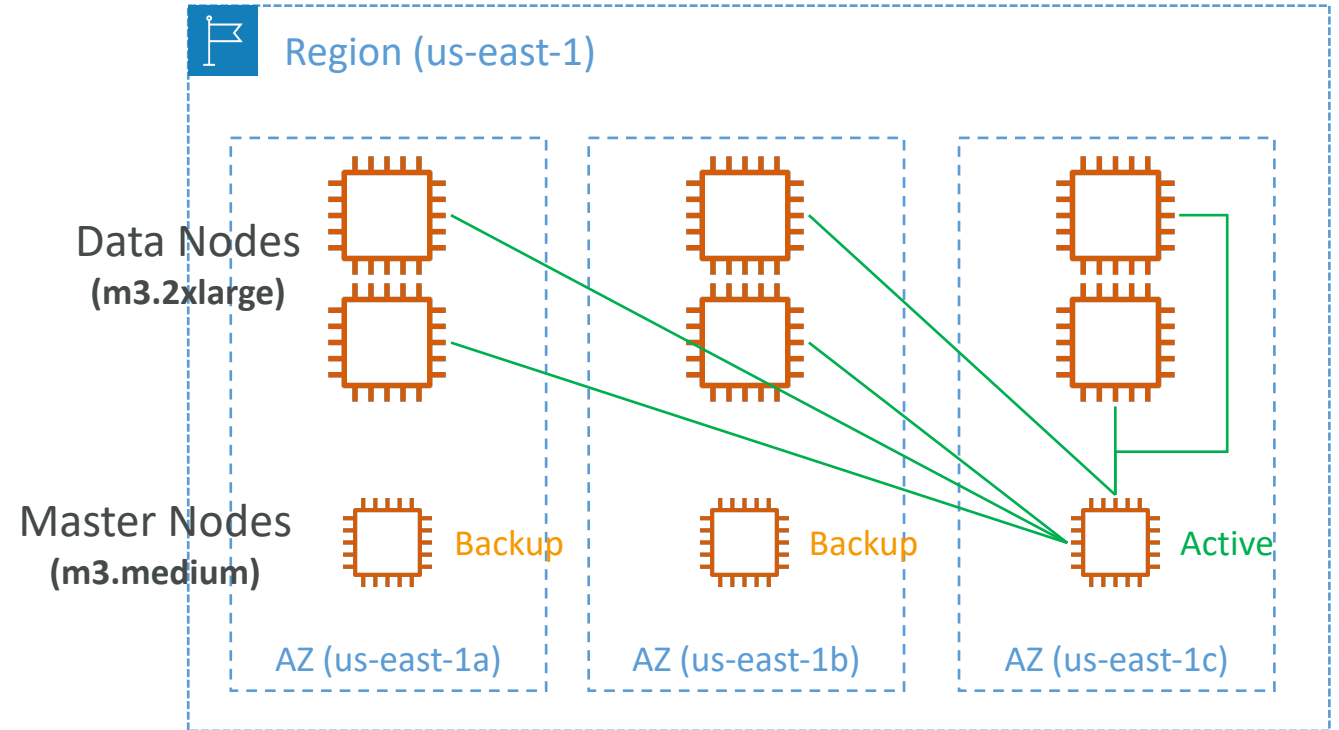
IP-based Policy  
(ES configured to allow proxy server IP)



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:*",
      "Resource": "arn:aws:es:us-west-1:123456789012:domain/test-domain/*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": ["123.456.789.123"]
        }
      }
    }
  ]
}
```

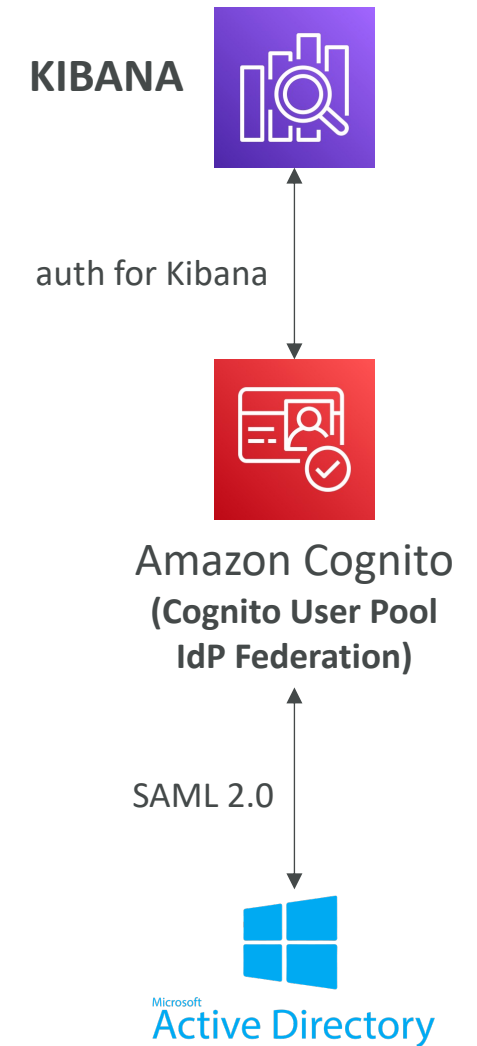
# ElasticSearch – Production Setup

- It's recommended to:
  - Use 3 dedicated Master nodes
  - Use at least 2 Data nodes per AZ (for replication)
  - Deploy the domain across 3 Availability Zones
  - Create at least one replica for each index in the cluster



# Kibana Authentication

- Kibana doesn't natively support IAM users and roles
- You can control access to Kibana using:
  - **HTTP Basic Authentication**
    - Internal user database stored in an ElasticSearch Index
  - **SAML**
    - Use existing 3<sup>rd</sup> party identity providers to log into Kibana
    - Supports SAML 2.0 (e.g., Active Directory Federation Services – ADFS, Auth0, ...)
  - **Amazon Cognito**
    - Easy for MS Active Directory (AD) integration



# AWS X-Ray



- Debugging in Production, the good old way:
  - Test locally
  - Add log statements everywhere
  - Re-deploy in production
- Log formats differ across applications and log analysis is hard.
- Debugging: one big monolith “easy”, distributed services “hard”
- No common views of your entire architecture
  
- Enter... AWS X-Ray!

# AWS X-Ray

## Visual analysis of our applications



# AWS X-Ray advantages

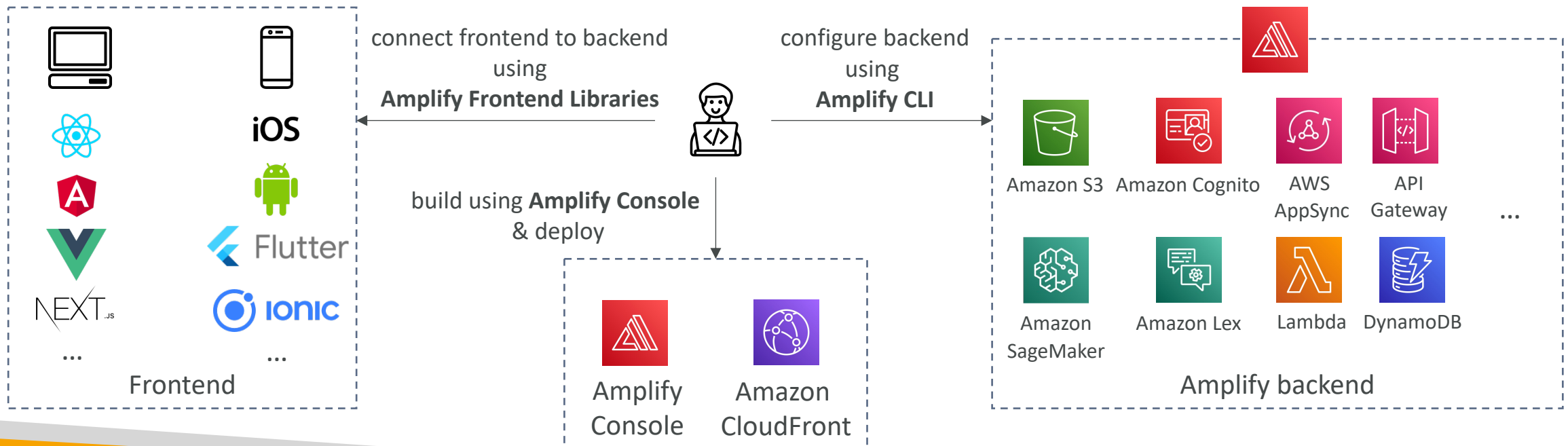


- Troubleshooting performance (bottlenecks)
- Understand dependencies in a microservice architecture
- Pinpoint service issues
- Review request behavior
- Find errors and exceptions
- Are we meeting time SLA?
- Where I am throttled?
- Identify users that are impacted

# AWS Amplify - web and mobile applications



- A set of tools and services that helps you develop and deploy scalable full stack web and mobile applications
- Authentication, Storage, API (REST, GraphQL), CI/CD, PubSub, Analytics, AI/ML Predictions, Monitoring, ...
- Connect your source code from GitHub, AWS CodeCommit, Bitbucket, GitLab, or upload directly



# Exam Review & Tips



# State of learning checkpoint

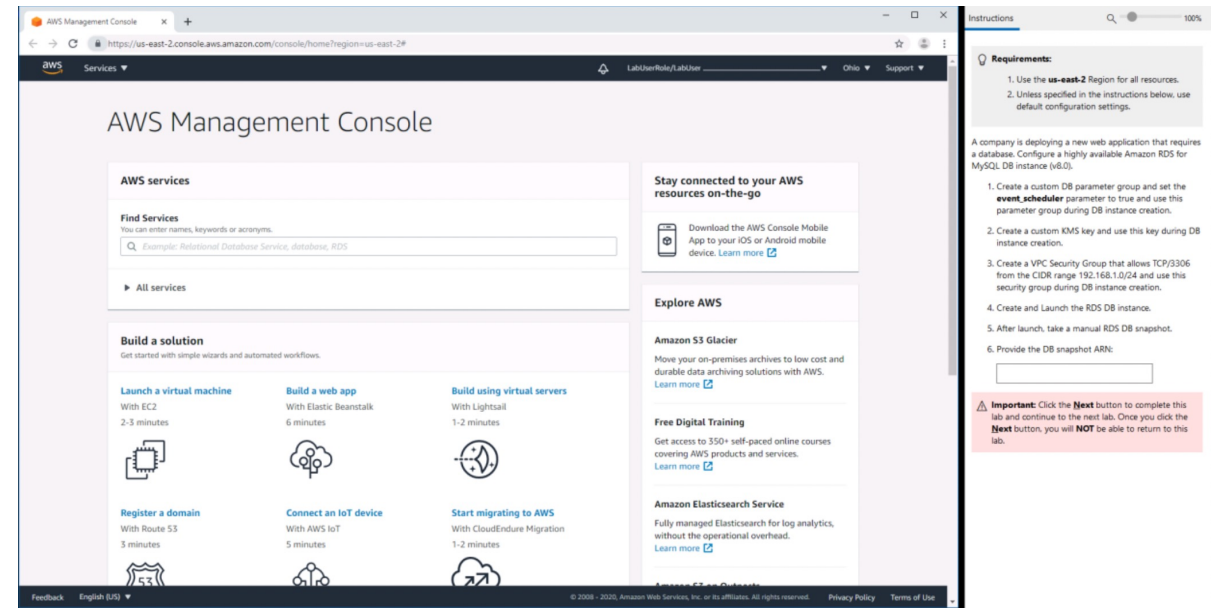
- Let's look how far we've gone on our learning journey
- <https://aws.amazon.com/certification/certified-sysops-admin-associate/>

# AWS Certified SysOps Exam Labs

- Starting with SOA-C02, the exam will contain **3 labs**
- Each exam lab consists of several different tasks
- AWS recommends to **allocate 20 minutes for each exam lab**
- First, you'll need to answer the multiple-choice/multiple-responses questions (you can't go back once you have completed this section)
- Second, you'll need to answer the exam labs
- You must complete all the work on the exam lab, before you move to the next one (you won't be able to go back to a lab once you have completed it)

# AWS Certified SysOps Exam Sample Lab

- A company is deploying a new web application. Configure a highly available MySQL 8.0 database with the following:
  1. Create a custom DB parametergroup and set the `event_scheduler` parameter to `true` and use this parameter during DB instance creation.
  2. Create a custom AWS Key Management Service (AWS KMS) key and use this key during DB instance creation.
  3. Create a VPC security group that allows TCP port 3306 from the CIDR block 192.168.1.0/24. Use this security group during DB instance creation.
  4. Launch the Amazon RDS DB instance.
  5. After launch, take a manual RDS DB snapshot.



# Congratulations!

# Congratulations!

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