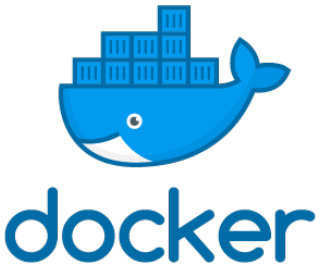
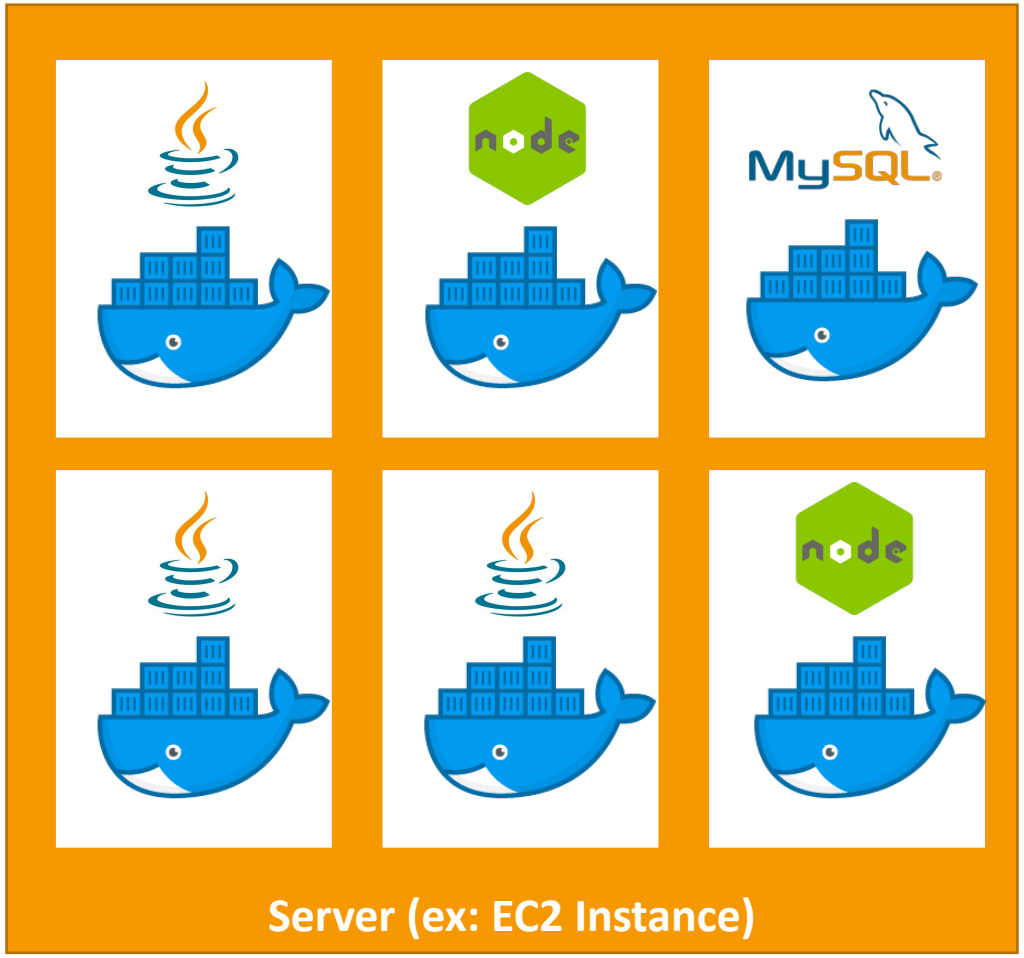
Other Compute Section

What is Docker?



- Docker is a software development platform to deploy apps
- Apps are packaged in **containers** that can be run on any OS
- **Apps run the same, regardless of where they're run**
  - Any machine
  - No compatibility issues
  - Predictable behavior
  - Less work
  - Easier to maintain and deploy
  - Works with any language, any OS, any technology
- Scale containers up and down very quickly (seconds)

Docker on an OS



Where Docker images are stored?

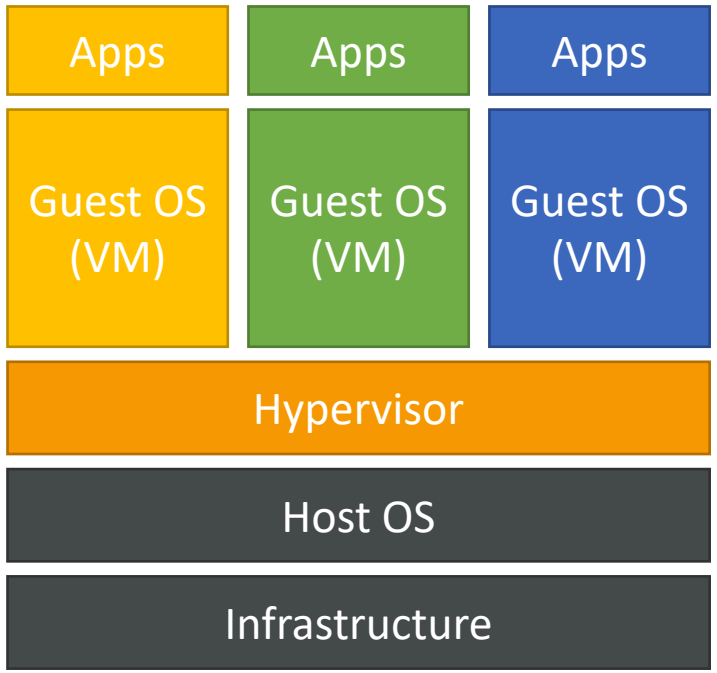
- Docker images are stored in Docker Repositories

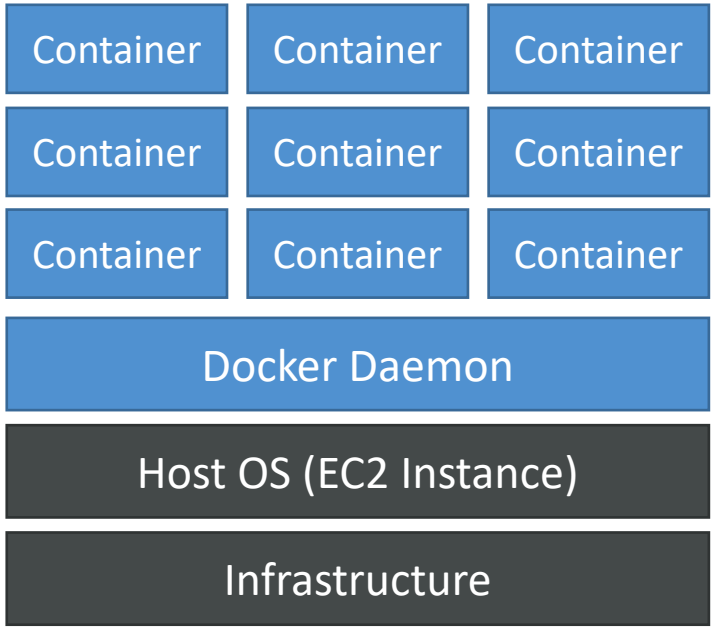
- Public: Docker Hub https://hub.docker.com/
  - Find base images for many technologies or OS:
  - Ubuntu
  - MySQL
  - NodeJS, Java…

- Private: Amazon ECR (Elastic Container Registry)

Docker versus Virtual Machines

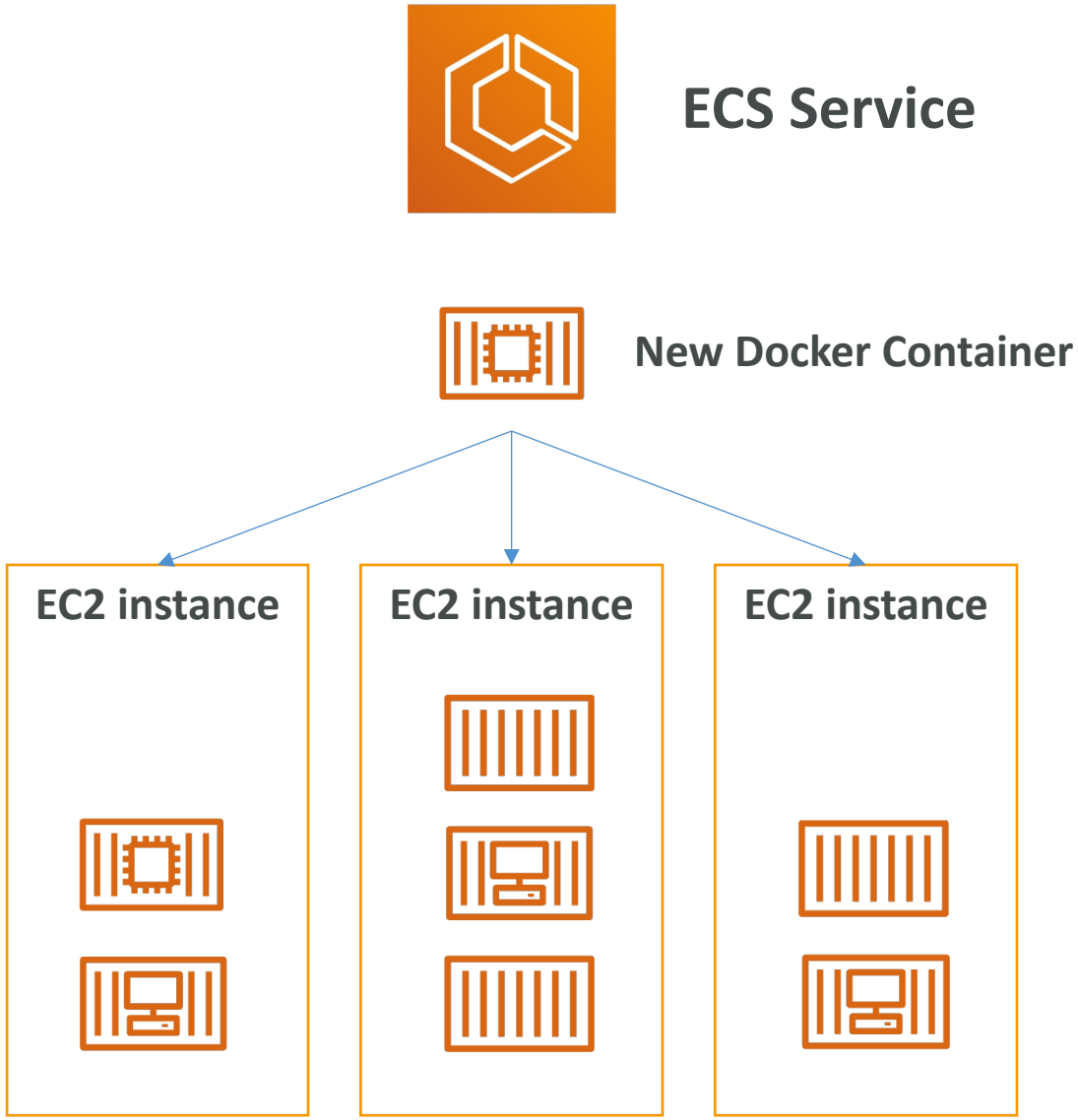
- Docker is "sort of" a virtualization technology, but not exactly
- Resources are shared with the host => many containers on one server





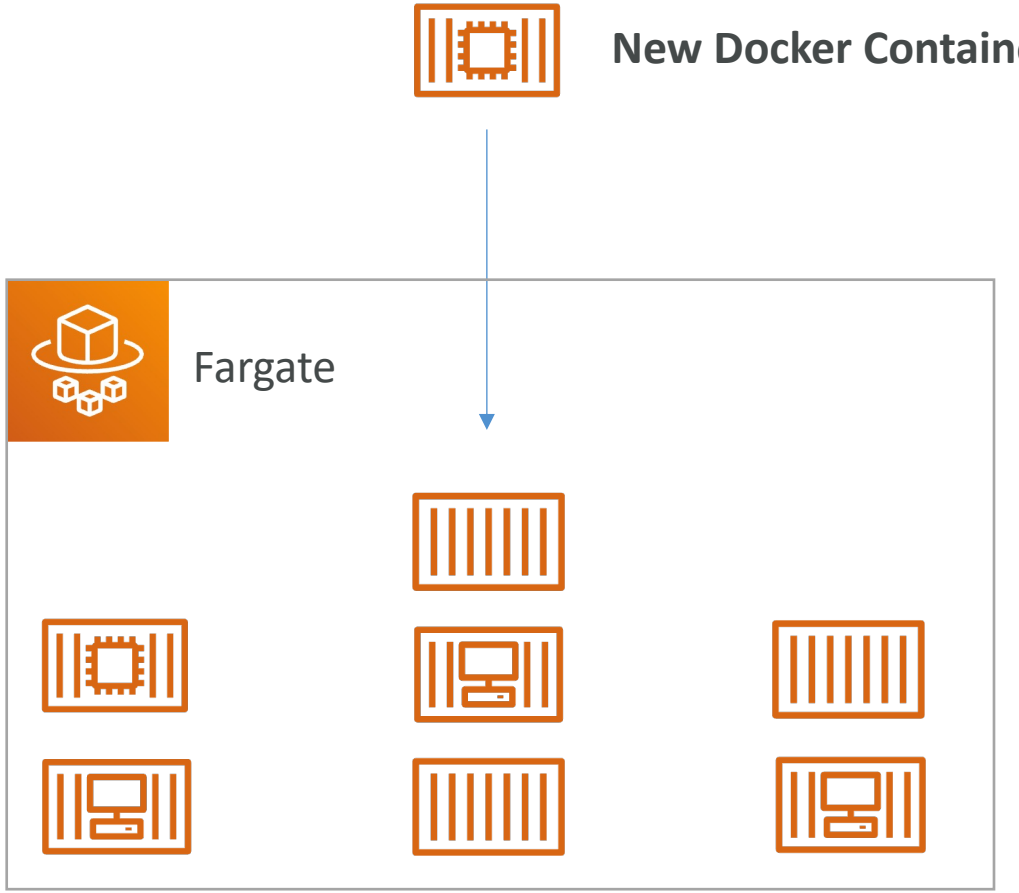
ECS

- ECS = Elastic Container Service
- Launch Docker containers on AWS
- You must provision & maintain the infrastructure (the EC2 instances)
- AWS takes care of starting / stopping containers
- Has integrations with the Application Load Balancer



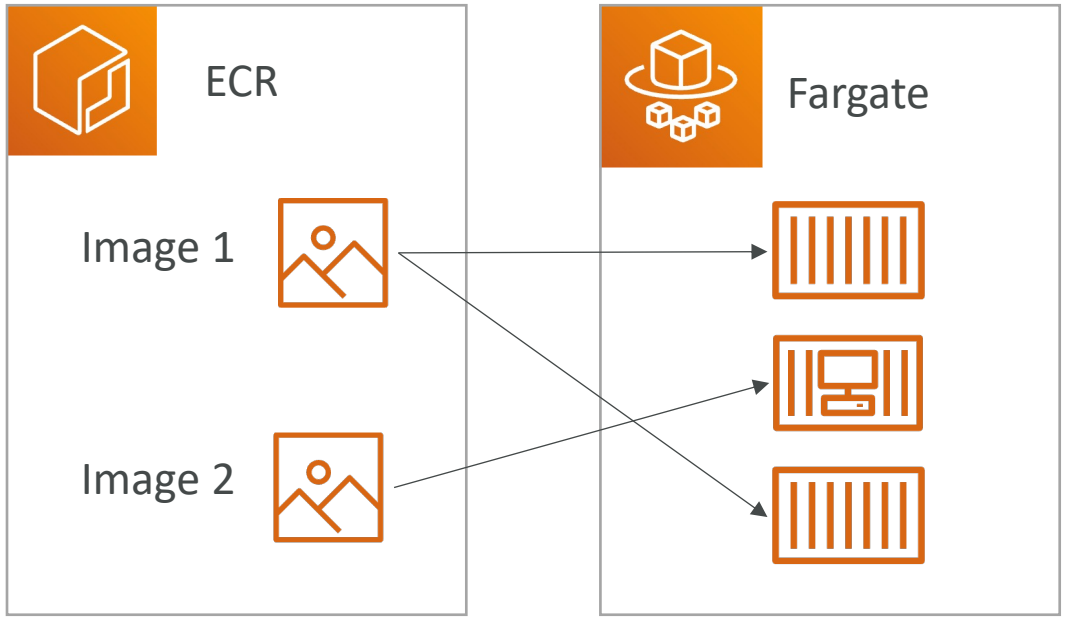
Fargate

- Launch Docker containers on AWS
- You do not provision the infrastructure (no EC2 instances to manage) – simpler!
- Serverless offering
- AWS just runs containers for you based on the CPU / RAM you need



ECR

- Elastic Container Registry
- Private Docker Registry on AWS
- This is where you **store your Docker images** so they can be run by ECS or Fargate



What's serverless?

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

So far in this course…

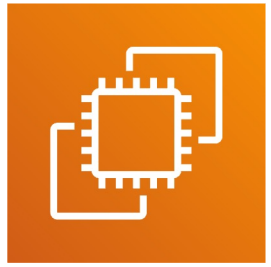
Amazon S3

DynamoDB

Fargate

Lambda

Why AWS Lambda



Amazon EC2

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

Amazon Lambda

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

Benefits of AWS Lambda

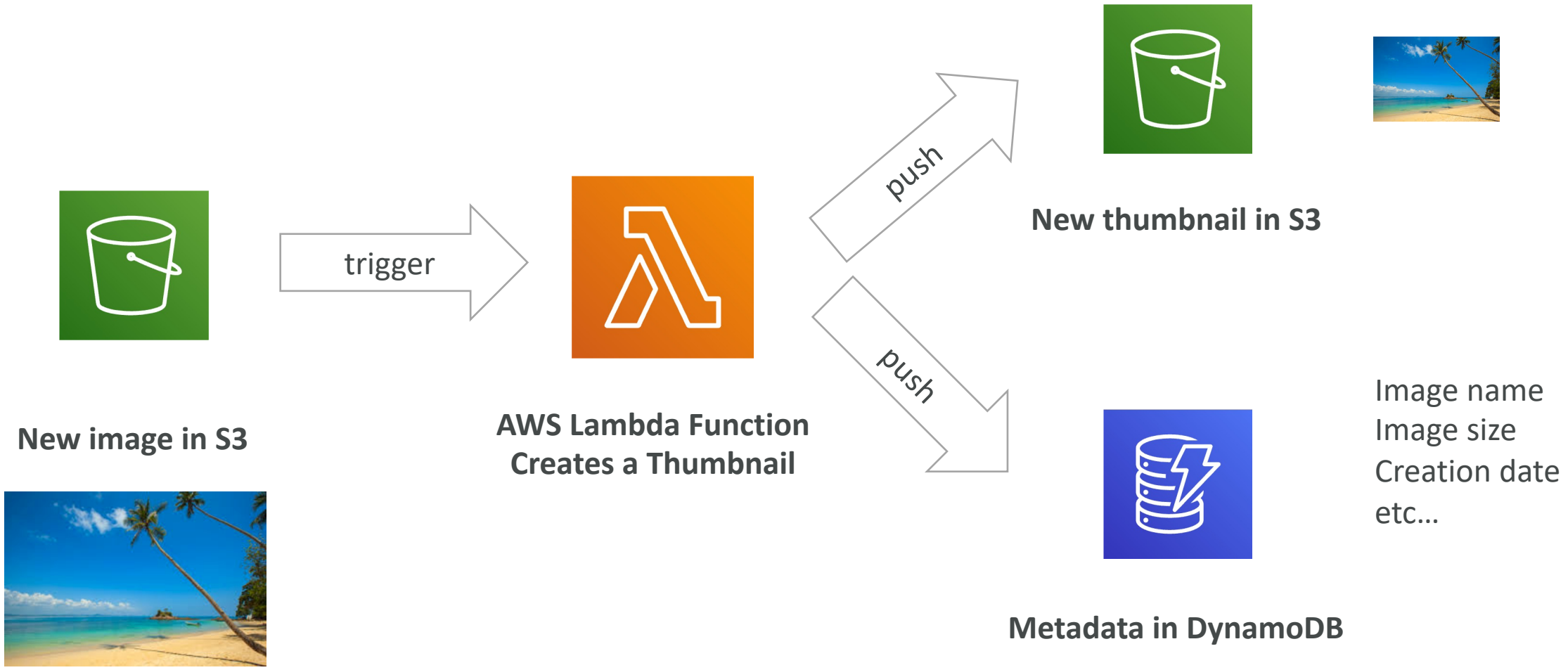
- Easy Pricing:
  - Pay per request and compute time
  - Free tier of 1,000,000 AWS Lambda requests and 400,000 GBs of compute time
- Integrated with the whole AWS suite of services
- **Event-Driven:** functions get invoked by AWS when needed
- Integrated with many programming languages
- Easy monitoring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

AWS Lambda language support

- Node.js (JavaScript)
- Python
- Java (Java 8 compatible)
- C# (.NET Core)
- Golang
- C# / Powershell
- Ruby
- Custom Runtime API (community supported, example Rust)

- Lambda Container Image
  - The container image must implement the Lambda Runtime API
  - ECS / Fargate is preferred for running arbitrary Docker images

Example: Serverless Thumbnail creation

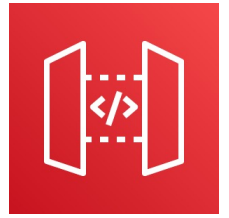


Example: Serverless CRON Job

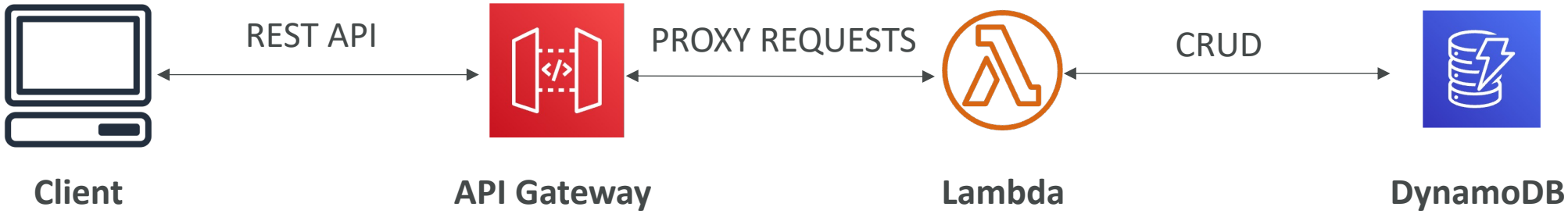
AWS Lambda Pricing: example

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

Amazon API Gateway



- Example: building a serverless API

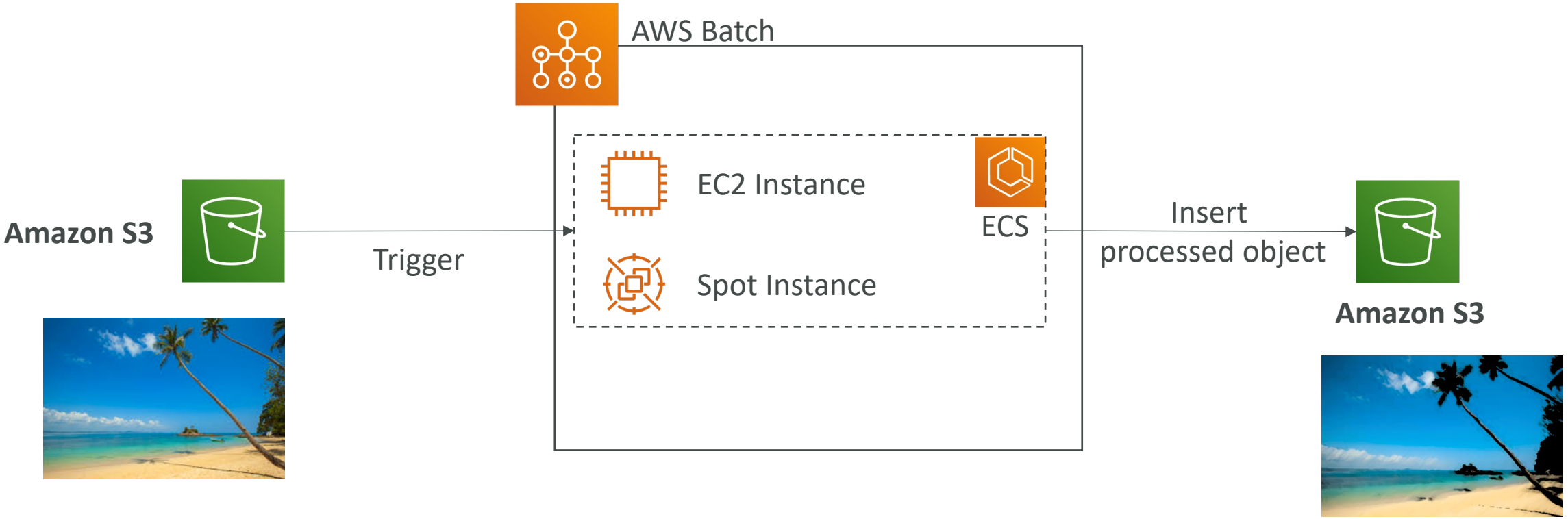


- Fully managed service for developers to easily create, publish, maintain, monitor, and secure APIs
- **Serverless** and scalable
- Supports RESTful APIs and WebSocket APIs
- Support for security, user authentication, API throttling, API keys, monitoring...

AWS Batch

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

AWS Batch – Simplified Example



Batch vs Lambda

- Lambda:
  - Time limit
  - Limited runtimes
  - Limited temporary disk space
  - Serverless

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

Amazon Lightsail

- Virtual servers, storage, databases, and networking
- Low & predictable pricing
- Simpler alternative to using EC2, RDS, ELB, EBS, Route 53…
- Great for people **with little cloud experience**!
- Can setup notifications and monitoring of your Lightsail resources
- Use cases:
  - Simple web applications (has templates for LAMP, Nginx, MEAN, Node.js…)
  - Websites (templates for WordPress, Magento, Plesk, Joomla)
  - Dev / Test environment
- Has high availability but no auto-scaling, limited AWS integrations

Other Compute - Summary

- **Docker:** container technology to run applications
- **ECS:** run Docker containers on EC2 instances
- **Fargate:**
  - Run Docker containers without provisioning the infrastructure
  - Serverless offering (no EC2 instances)
- **ECR:** Private Docker Images Repository
- **Batch:** run batch jobs on AWS across managed EC2 instances
- **Lightsail:** predictable & low pricing for simple application & DB stacks

Lambda Summary

- Lambda is Serverless, Function as a Service, seamless scaling, reactive
- **Lambda Billing:**
  - By the time run x by the RAM provisioned
  - By the number of invocations
- **Language Support:** many programming languages except (arbitrary) Docker
- **Invocation time:** up to 15 minutes
- **Use cases:**
  - Create Thumbnails for images uploaded onto S3
  - Run a Serverless cron job
- **API Gateway**: expose Lambda functions as HTTP API

Deploying and Managing Infrastructure at Scale Section

What is CloudFormation

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

Benefits of AWS CloudFormation (1/2)

- Infrastructure as code
  - No resources are manually created, which is excellent for control
  - Changes to the infrastructure are reviewed through code

- Cost
  - Each resources within the stack is tagged with an identifier so you can easily see how much a stack costs you
  - You can estimate the costs of your resources using the CloudFormation template
  - Savings strategy: In Dev, you could automation deletion of templates at 5 PM and recreated at 8 AM, safely

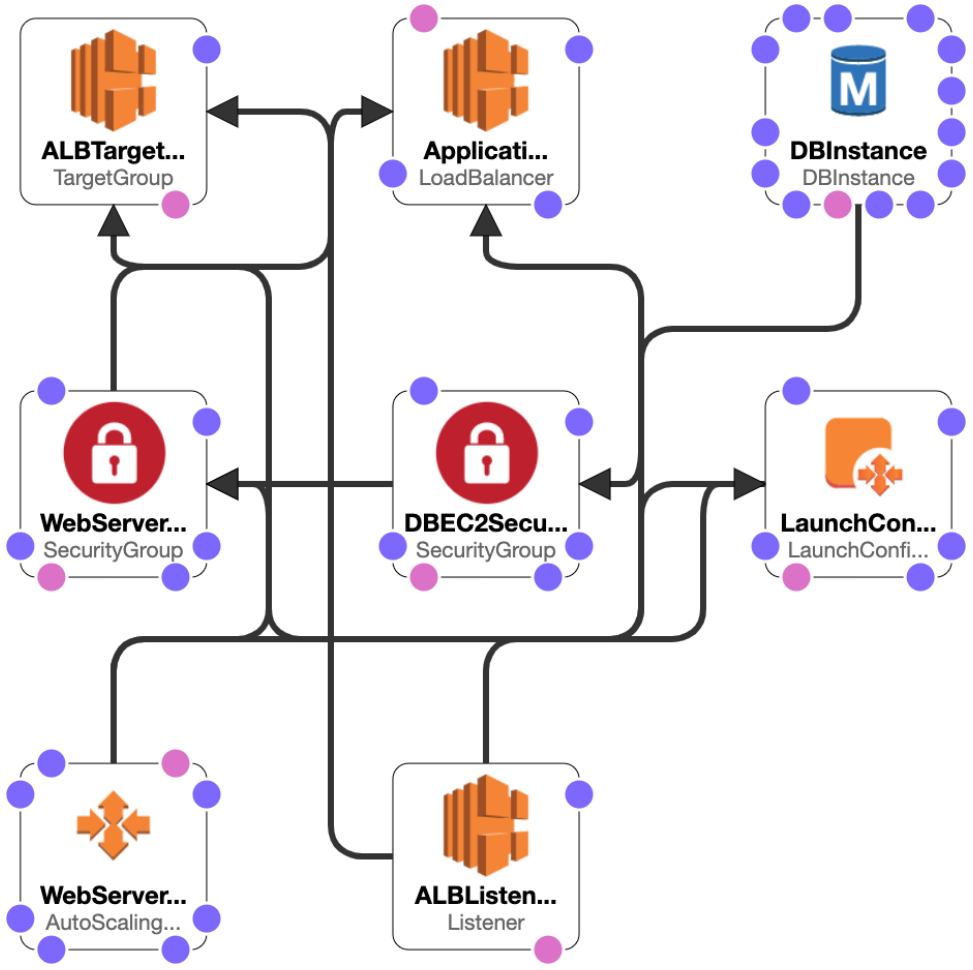
Benefits of AWS CloudFormation (2/2)

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

CloudFormation Stack Designer

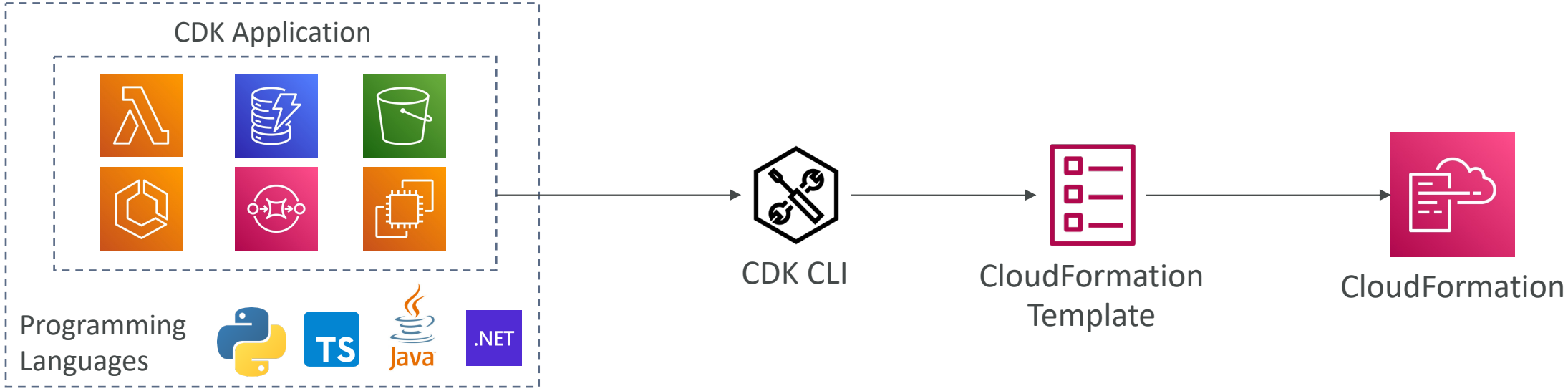
- Example: WordPress CloudFormation Stack

- We can see all the **resources**
- We can see the **relations** between the components



AWS Cloud Development Kit (CDK)

- Define your cloud infrastructure using a familiar language:
  - JavaScript/TypeScript, Python, Java, and .NET
- The code is "compiled" into a CloudFormation template (JSON/YAML)
- You can therefore deploy infrastructure and application runtime code together
  - Great for Lambda functions
  - Great for Docker containers in ECS / EKS



CDK Example

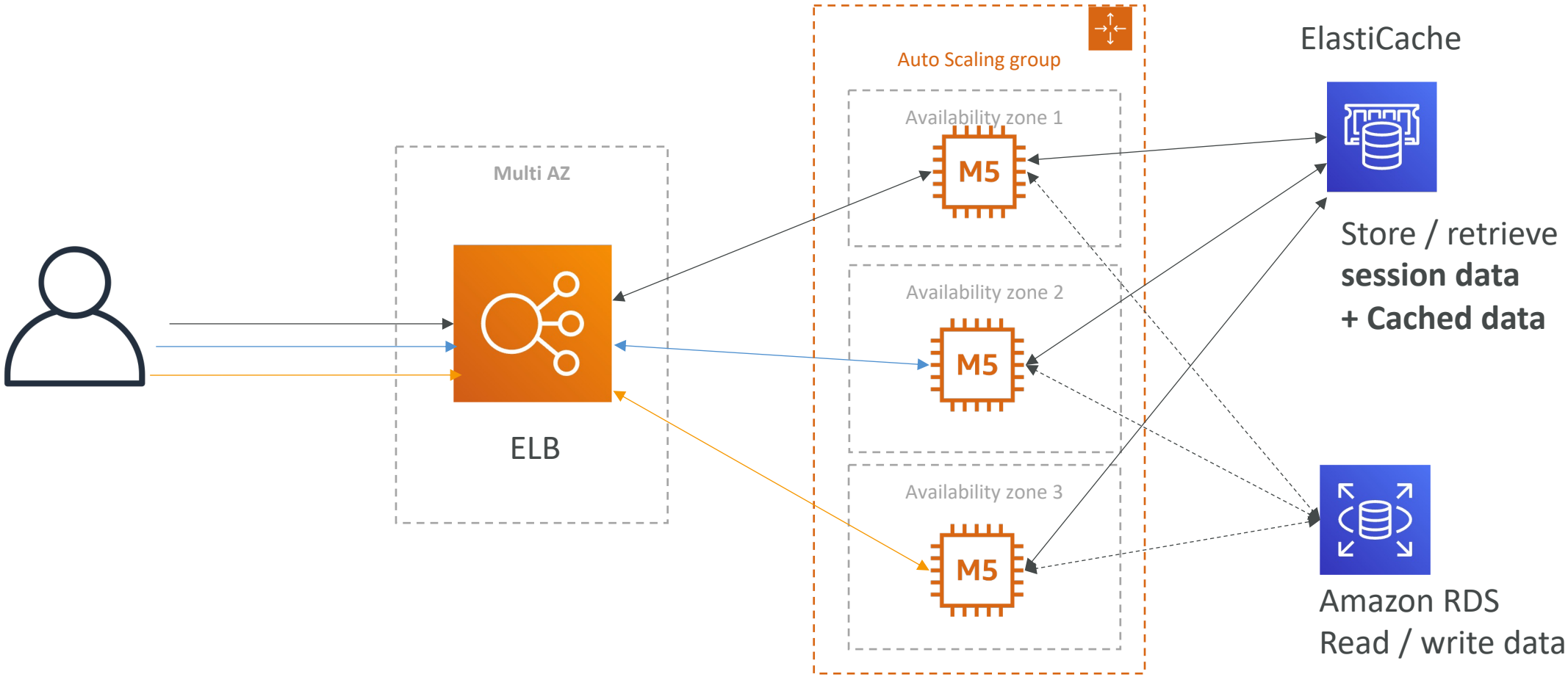
```
export class MyEcsConstructStack extends core.Stack {
  constructor(scope: core.App, id: string, props?: core.StackProps)
    super(scope, id, props);

    const vpc = new ec2.Vpc(this, "MyVpc", {
      maxAzs: 3 // Default is all AZs in region
    });

    const cluster = new ecs.Cluster(this, "MyCluster", {
      vpc: vpc
    });

    // Create a load-balanced Fargate service and make it public
    new ecs_patterns.ApplicationLoadBalancedFargateService(this, "My
      cluster: cluster, // Required
      cpu: 512, // Default is 256
      desiredCount: 6, // Default is 1
      taskImageOptions: { image: ecs.ContainerImage.fromRegistry("am
      memoryLimitMiB: 2048, // Default is 512
      publicLoadBalancer: true // Default is false
    });
  }
}
```

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

AWS 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, etc…
- But it's all in one view that's easy to make sense of!
- We still have full control over the configuration

- **Beanstalk = Platform as a Service (PaaS)**
- Beanstalk is free but you pay for the underlying instances

Elastic Beanstalk

- Managed service
  - Instance configuration / OS is handled by Beanstalk
  - Deployment strategy is configurable but performed by Elastic Beanstalk
  - Capacity provisioning
  - Load balancing & auto-scaling
  - Application health-monitoring & responsiveness

- Just the application code is the responsibility of the developer

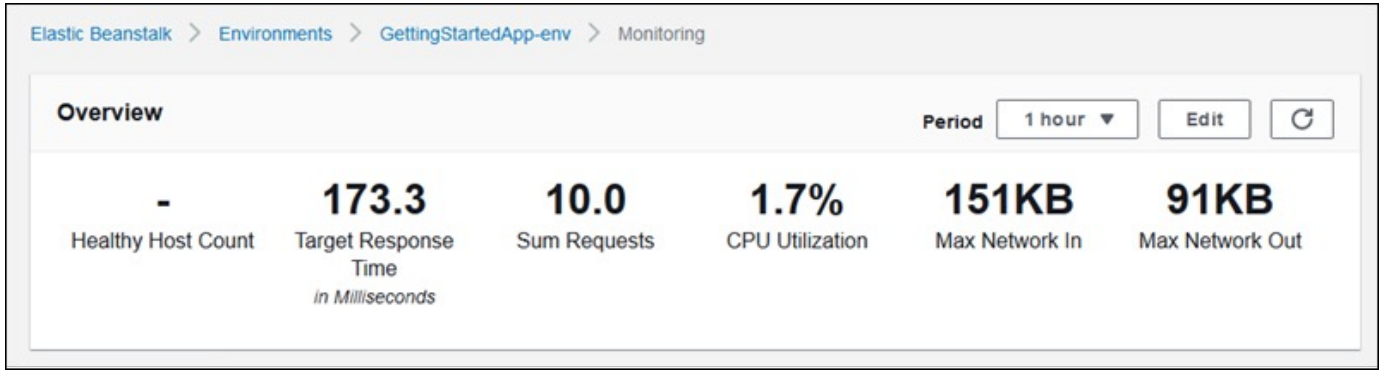
- Three architecture models:
  - Single Instance deployment: good for dev
  - LB + ASG: great for production or pre-production web applications
  - ASG only: great for non-web apps in production (workers, etc..)

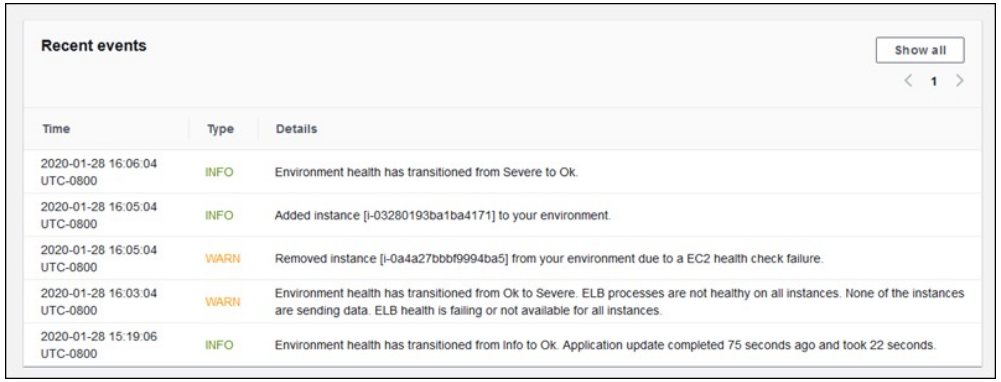
Elastic Beanstalk

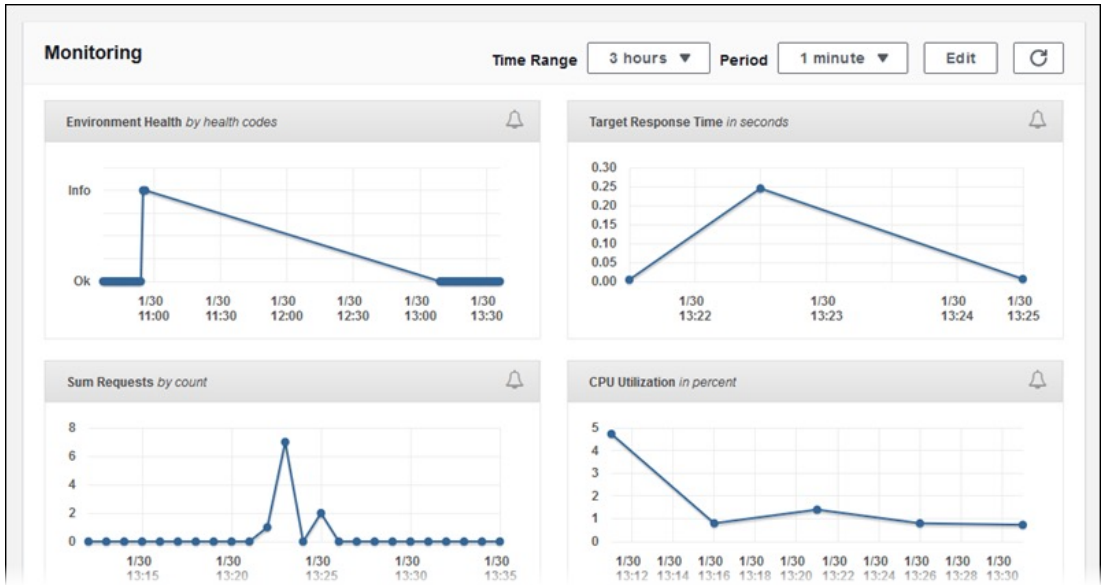
- Support for many platforms:
  - Go
  - Java SE
  - Java with Tomcat
  - .NET on Windows Server with IIS
  - Node.js
  - PHP
  - Python
  - Ruby
  - Packer Builder
  - Single Container Docker
  - Multi-Container Docker
  - Preconfigured Docker
- If not supported, you can write your custom platform (advanced)

Elastic Beanstalk – Health Monitoring

- Health agent pushes metrics to CloudWatch
- Checks for app health, publishes health events





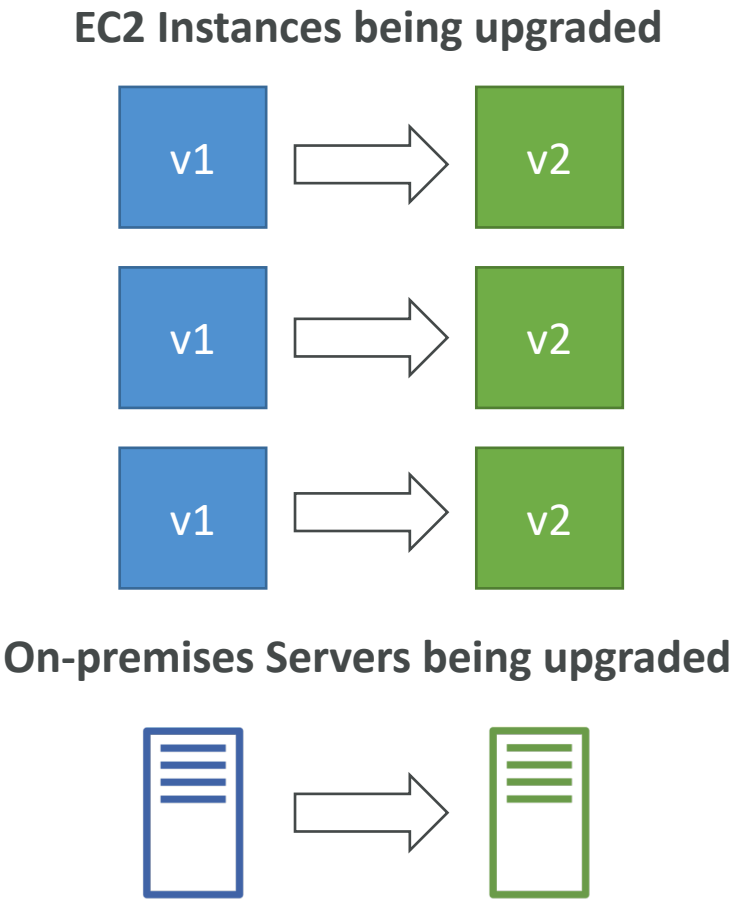


AWS CodeDeploy

- We want to deploy our application **automatically**

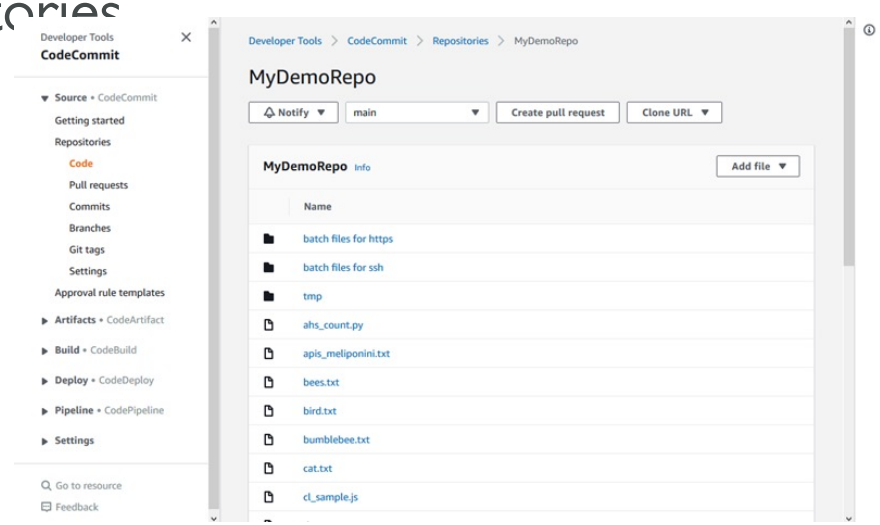
- Works with EC2 Instances
- Works with On-Premises Servers
- Hybrid service

- Servers / Instances must be provisioned and configured ahead of time with the CodeDeploy Agent



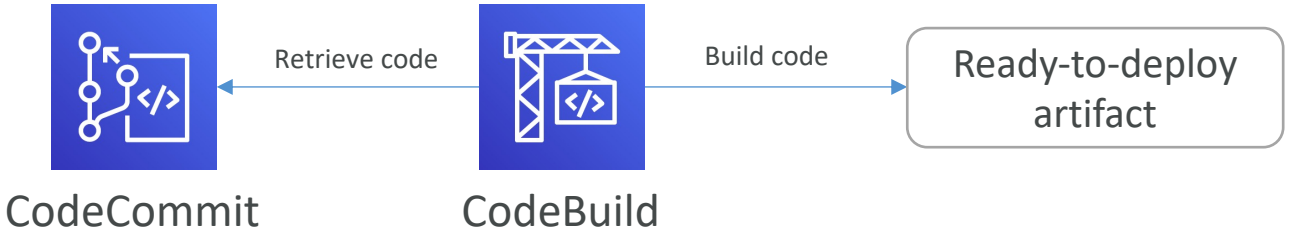
AWS CodeCommit

- Before pushing the application code to servers, it needs to be stored somewhere
- Developers usually store **code in a repository, using the Git technology**
- A famous public offering is GitHub, AWS' competing product is **CodeCommit**
- CodeCommit:
  - Source-control service that **hosts Git-based repositories**
  - Makes it easy to **collaborate with others on code**
  - The code changes are automatically **versioned**
- Benefits:
  - Fully managed
  - Scalable & highly available
  - Private, Secured, Integrated with AWS



AWS CodeBuild

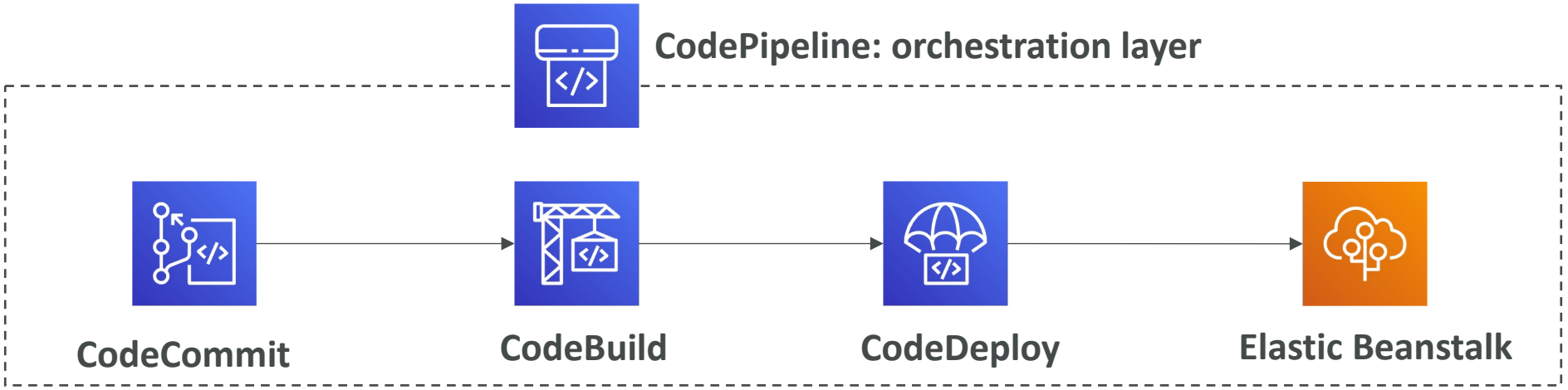
- Code building service in the cloud (name is obvious)
- Compiles source code, run tests, and produces packages that are ready to be deployed (by CodeDeploy for example)



- Benefits:
  - Fully managed, serverless
  - Continuously scalable & highly available
  - Secure
  - Pay-as-you-go pricing – only pay for the build time

AWS CodePipeline

- Orchestrate the different steps to have the code automatically pushed to production
  - Code => Build => Test => Provision => Deploy
  - Basis for CICD (Continuous Integration & Continuous Delivery)
- Benefits:
  - Fully managed, compatible with CodeCommit, CodeBuild, CodeDeploy, Elastic Beanstalk, CloudFormation, GitHub, 3rd-party services (GitHub…) & custom plugins…
  - Fast delivery & rapid updates

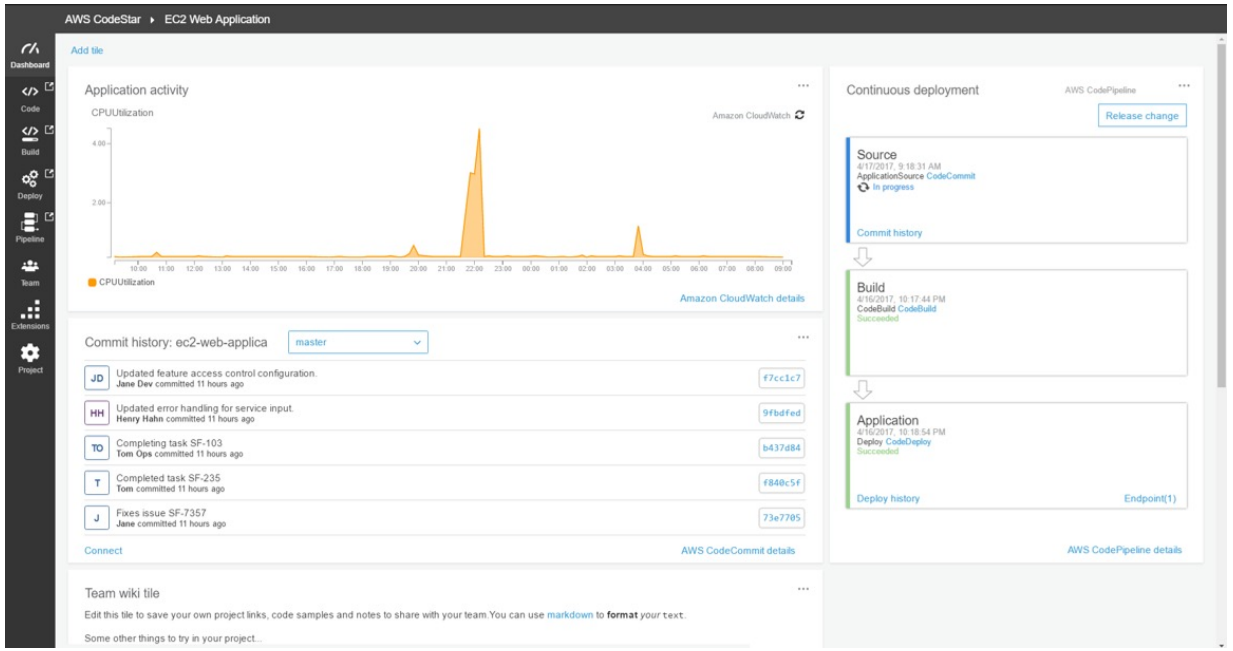


AWS CodeArtifact

- Software packages depend on each other to be built (also called code dependencies), and new ones are created
- Storing and retrieving these dependencies is called **artifact management**
- Traditionally you need to setup your own artifact management system
- **CodeArtifact** is a secure, scalable, and cost-effective **artifact management** for software development
- Works with common dependency management tools such as Maven, Gradle, npm, yarn, twine, pip, and NuGet
- **Developers and CodeBuild can then retrieve dependencies straight from CodeArtifact**

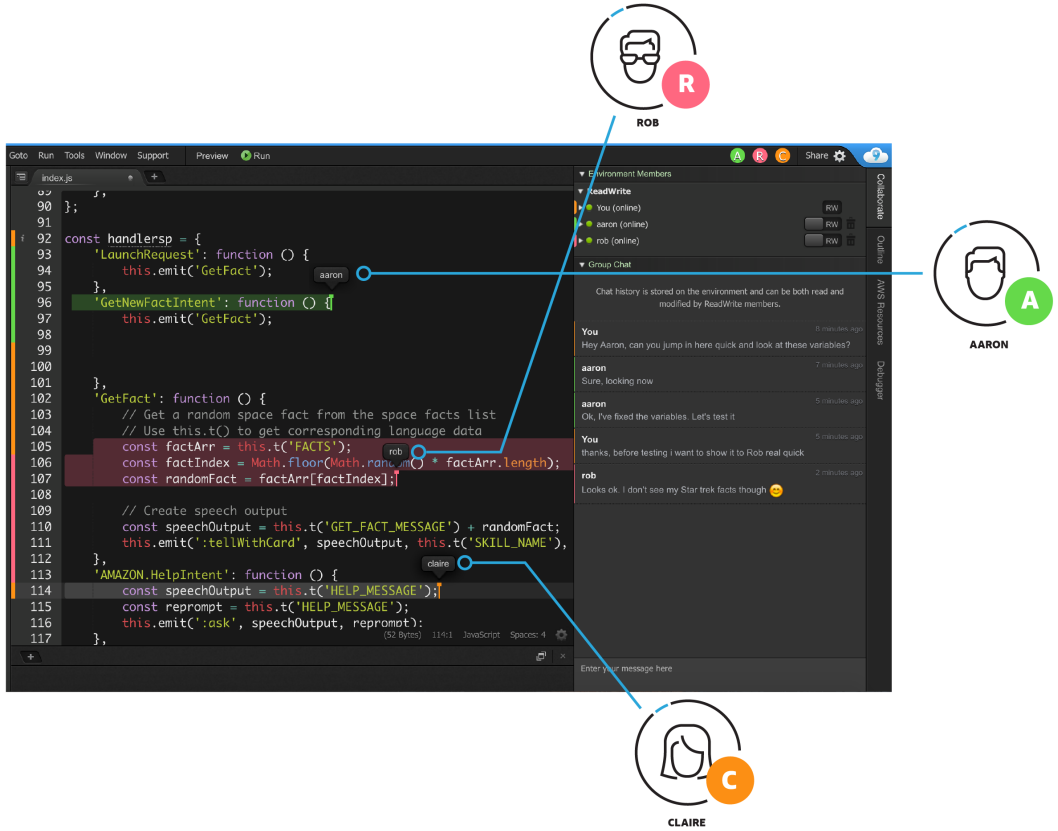
AWS CodeStar

- **Unified UI** to easily manage software development activities **in one place**
- "Quick way" to get started to correctly set-up CodeCommit, CodePipeline, CodeBuild, CodeDeploy, Elastic Beanstalk, EC2, etc…
- Can edit the code "in-the-cloud" using **AWS Cloud9**



AWS Cloud9

- AWS Cloud9 is a cloud IDE (Integrated Development Environment) for writing, running and debugging code
- "Classic" IDE (like IntelliJ, Visual Studio Code…) are downloaded on a computer before being used
- A cloud IDE can be used within a web browser, meaning you can work on your projects from your office, home, or anywhere with internet with no setup necessary
- AWS Cloud9 also allows for code collaboration in real-time (pair programming)

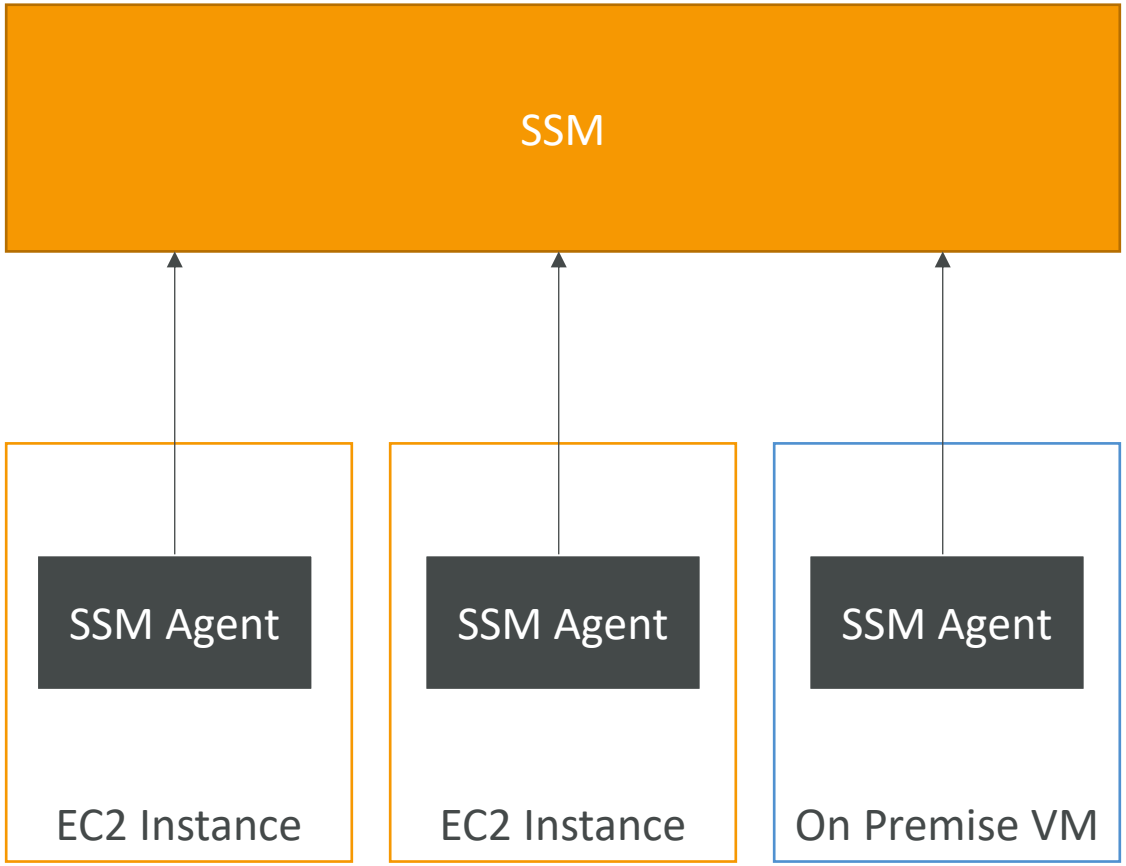


AWS Systems Manager (SSM)

- Helps you manage your **EC2 and On-Premises** systems at scale
- Another **Hybrid** AWS service
- Get operational insights about the state of your infrastructure
- Suite of 10+ products
- Most important features are:
  - **Patching automation for enhanced compliance**
  - **Run commands across an entire fleet of servers**
  - Store parameter configuration with the SSM Parameter Store
- Works for Linux, Windows, MacOS, and Raspberry Pi OS (Raspbian)

How Systems Manager works

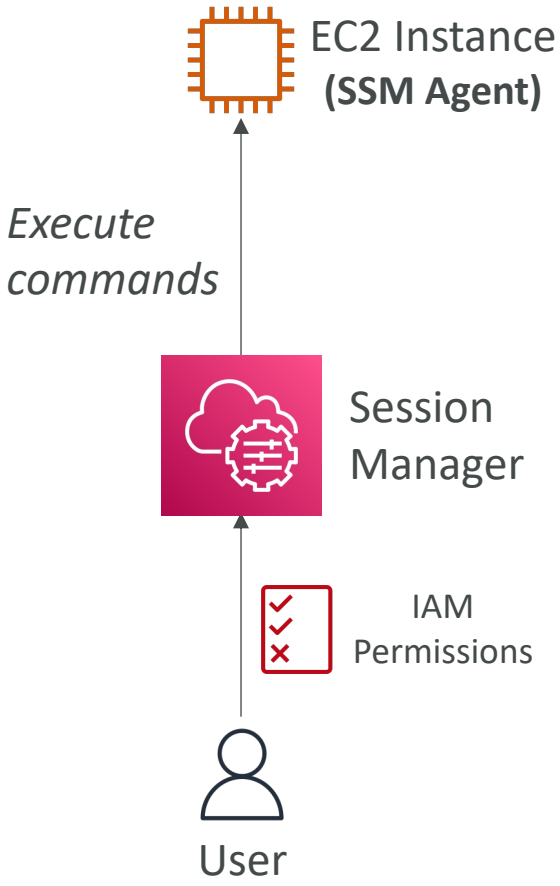
- We need to install the SSM agent onto the systems we control
- Installed by default on Amazon Linux AMI & some Ubuntu AMI
- If an instance can't be controlled with SSM, it's probably an issue with the SSM agent!
- Thanks to the SSM agent, we can **run commands**, **patch** & **configure** our servers



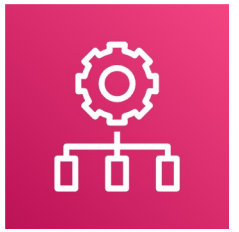
Systems Manager – SSM Session Manager

- Allows you to start a secure shell on your EC2 and on-premises servers

- No SSH access, bastion hosts, or SSH keys needed
- No port 22 needed (better security)
- Supports Linux, macOS, and Windows
- Send session log data to S3 or CloudWatch Logs



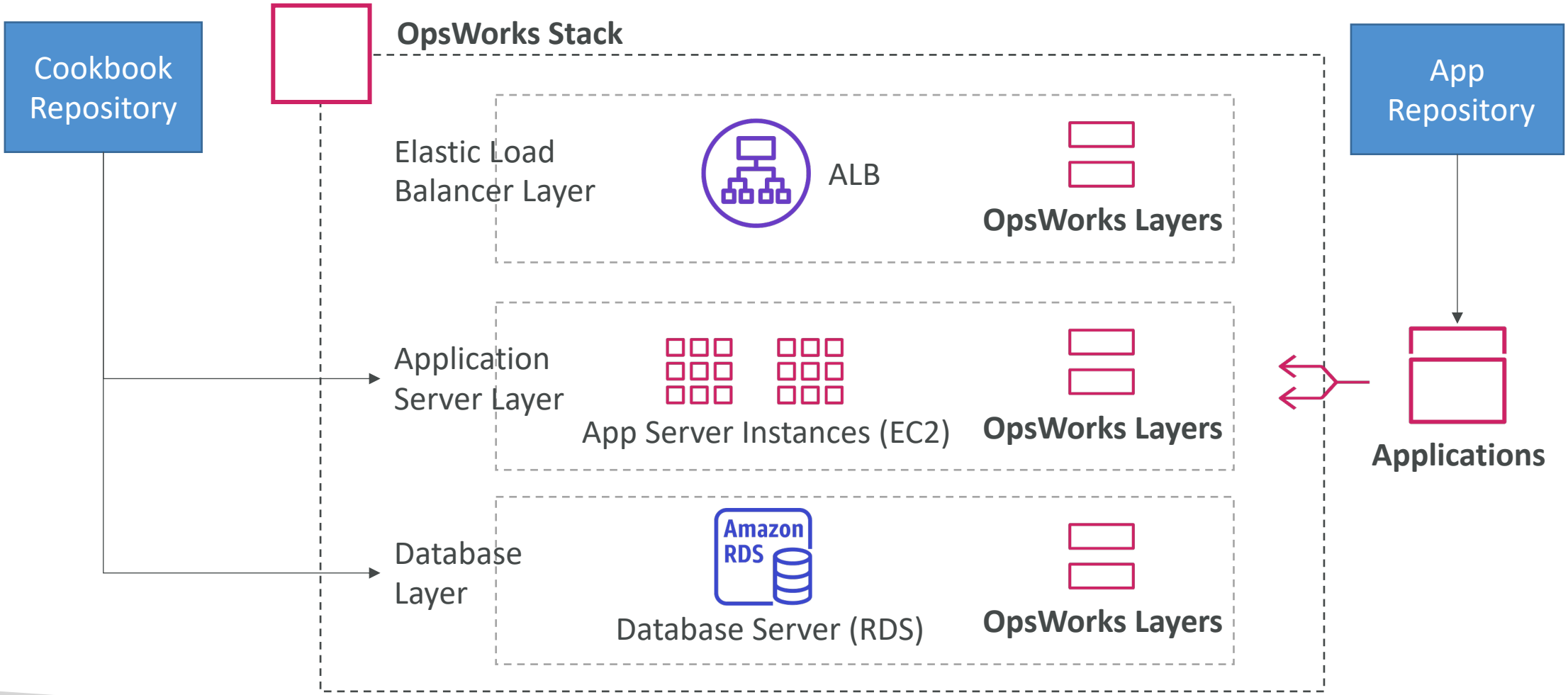
AWS OpsWorks



- 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
- Only provision **standard AWS resources**:
  - EC2 Instances, Databases, Load Balancers, EBS volumes…

- In the exam: Chef or Puppet needed => AWS OpsWorks

OpsWorks Architecture



Deployment - Summary

- CloudFormation: (AWS only)
  - Infrastructure as Code, works with almost all of AWS resources
  - Repeat across Regions & Accounts
- Beanstalk: (AWS only)
  - Platform as a Service (PaaS), limited to certain programming languages or Docker
  - Deploy code consistently with a known architecture: ex, ALB + EC2 + RDS
- CodeDeploy (hybrid): deploy & upgrade any application onto servers
- Systems Manager (hybrid): patch, configure and run commands at scale
- OpsWorks (hybrid): managed Chef and Puppet in AWS

Developer Services - Summary

- **CodeCommit**: Store code in private git repository (version controlled)
- **CodeBuild**: Build & test code in AWS
- **CodeDeploy**: Deploy code onto servers
- **CodePipeline**: Orchestration of pipeline (from code to build to deploy)
- **CodeArtifact**: Store software packages / dependencies on AWS
- **CodeStar**: Unified view for allowing developers to do CICD and code
- **Cloud9**: Cloud IDE (Integrated Development Environment) with collab
- **AWS CDK:** Define your cloud infrastructure using a programming language

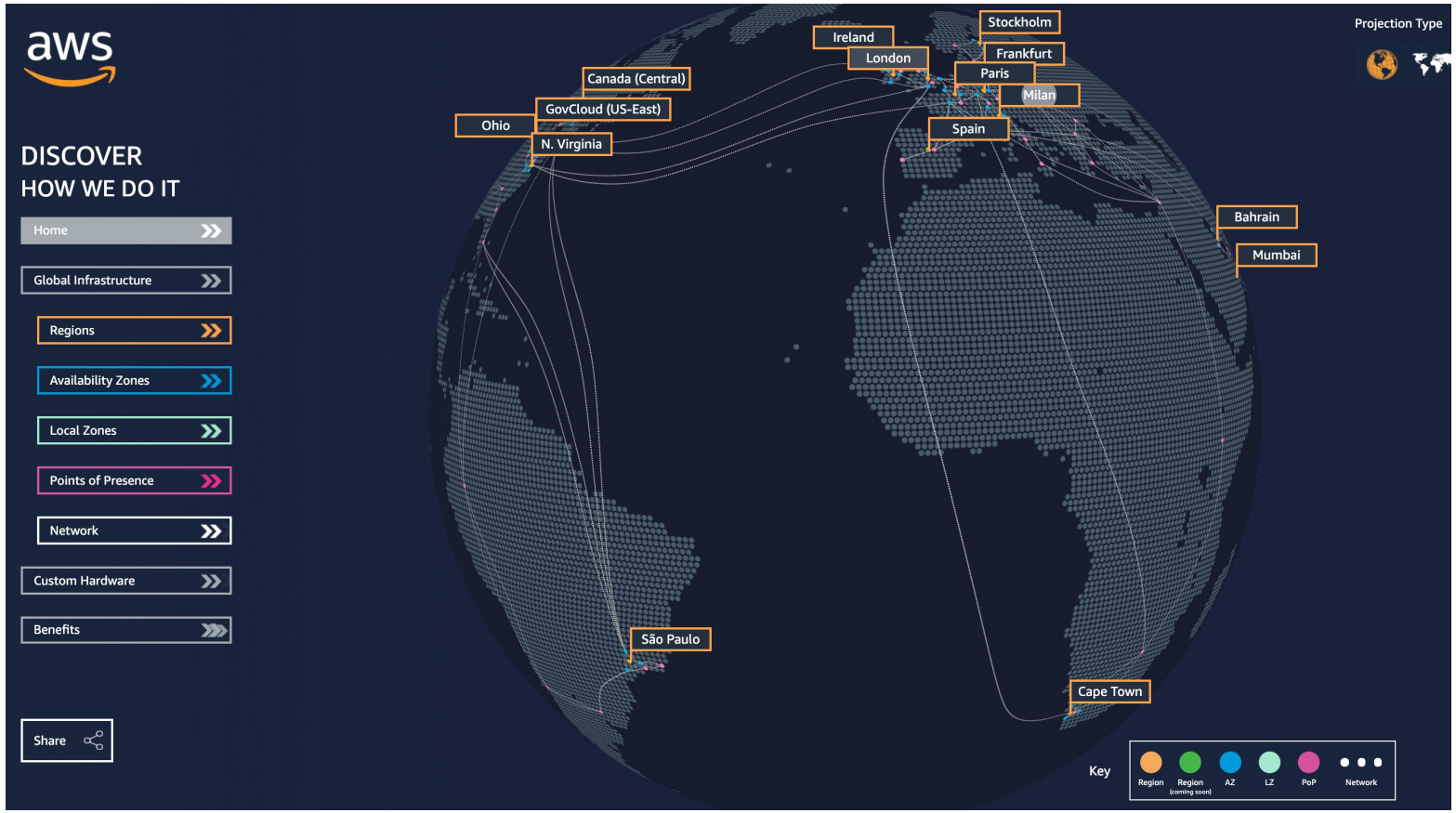
Global Infrastructure Section

Why make a global application?

- A **global application** is an application deployed in **multiple geographies**
- On AWS: this could be **Regions** and / or **Edge Locations**
- Decreased Latency
  - Latency is the time it takes for a network packet to reach a server
  - It takes time for a packet from Asia to reach the US
  - Deploy your applications closer to your users to decrease latency, better experience
- Disaster Recovery (DR)
  - If an AWS region goes down (earthquake, storms, power shutdown, politics)…
  - You can fail-over to another region and have your application still working
  - A DR plan is important to increase the availability of your application
- **Attack protection:** distributed global infrastructure is harder to attack

Global AWS Infrastructure

- **Regions:** For deploying applications and infrastructure
- **Availability Zones:** Made of multiple data centers
- **Edge Locations (Points of Presence):** for content delivery as close as possible to users
- More at: https://infrastructure.aws/



Global Applications in AWS

- Global DNS: Route 53
  - Great to route users to the closest deployment with least latency
  - Great for disaster recovery strategies

- Global Content Delivery Network (CDN): CloudFront
  - Replicate part of your application to AWS Edge Locations – decrease latency
  - Cache common requests – improved user experience and decreased latency

- S3 Transfer Acceleration
  - Accelerate global uploads & downloads into Amazon S3

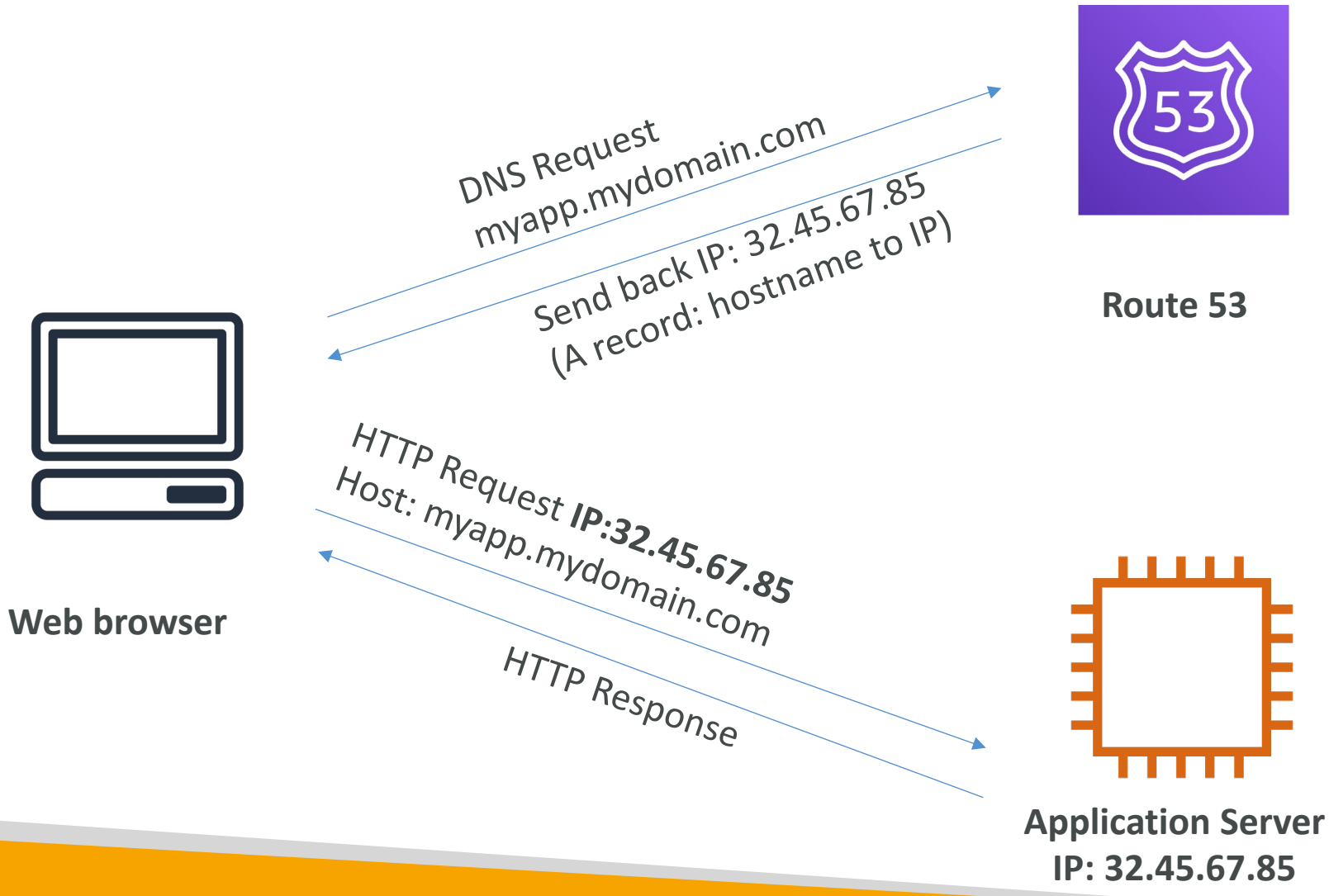
- AWS Global Accelerator:
  - Improve global application availability and performance using the AWS global network

Amazon Route 53 Overview

- Route53 is a Managed DNS (Domain Name System)
- DNS is a collection of rules and records which helps clients understand how to reach a server through URLs.

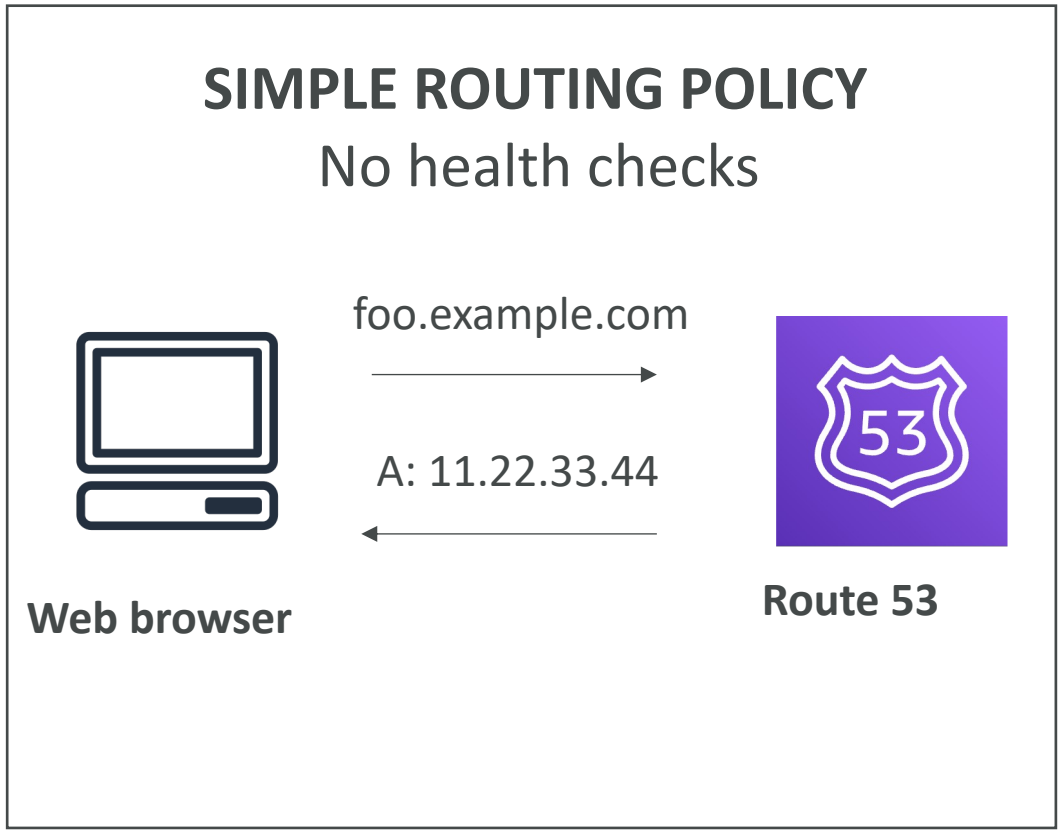
- In AWS, the most common records are:
  - www.google.com => 12.34.56.78 == A record (IPv4)
  - www.google.com => 2001:0db8:85a3:0000:0000:8a2e:0370:7334 == AAAA IPv6
  - search.google.com => www.google.com == CNAME: hostname to hostname
  - example.com => AWS resource == Alias (ex: ELB, CloudFront, S3, RDS, etc…)

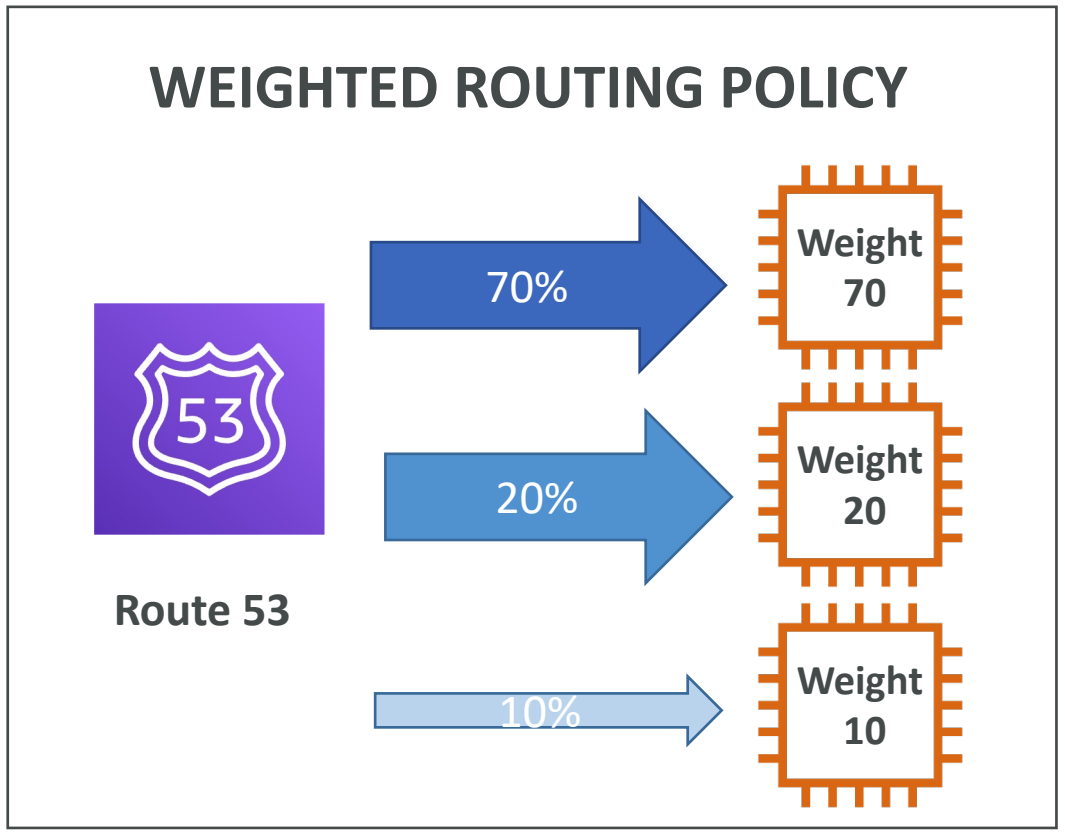
Route 53 – Diagram for A Record



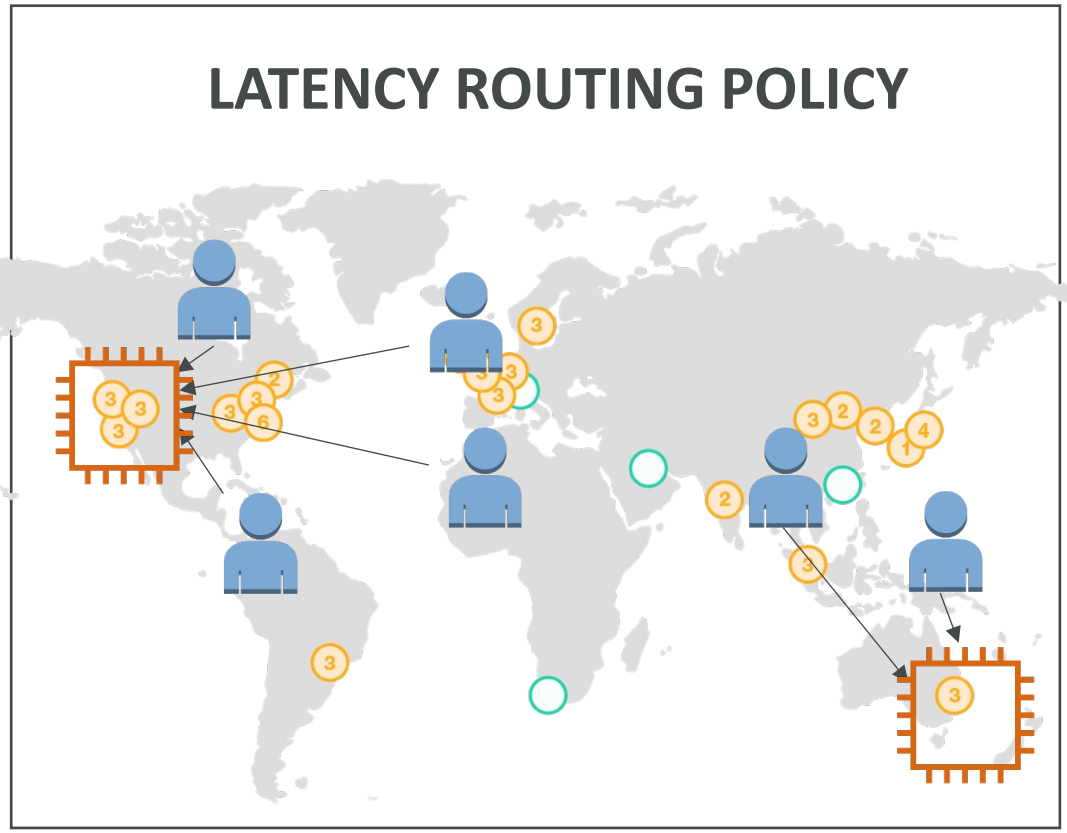
Route 53 Routing Policies

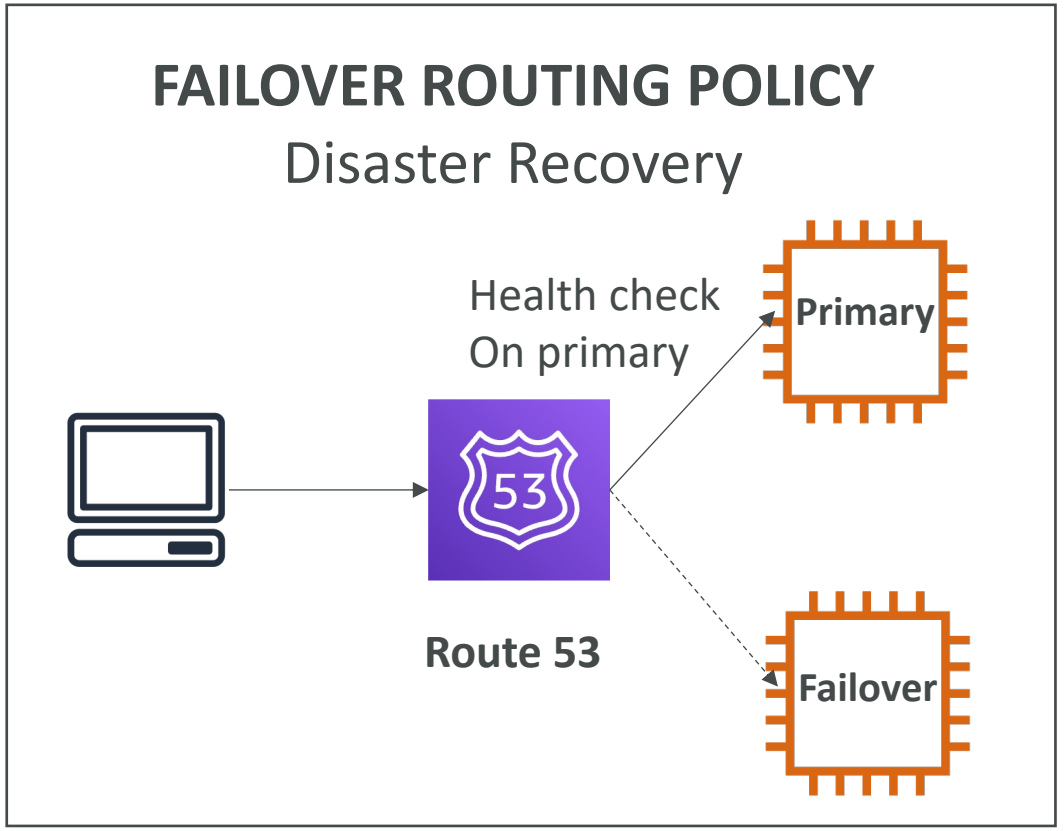
- Need to know them at a high-level for the Cloud Practitioner Exam





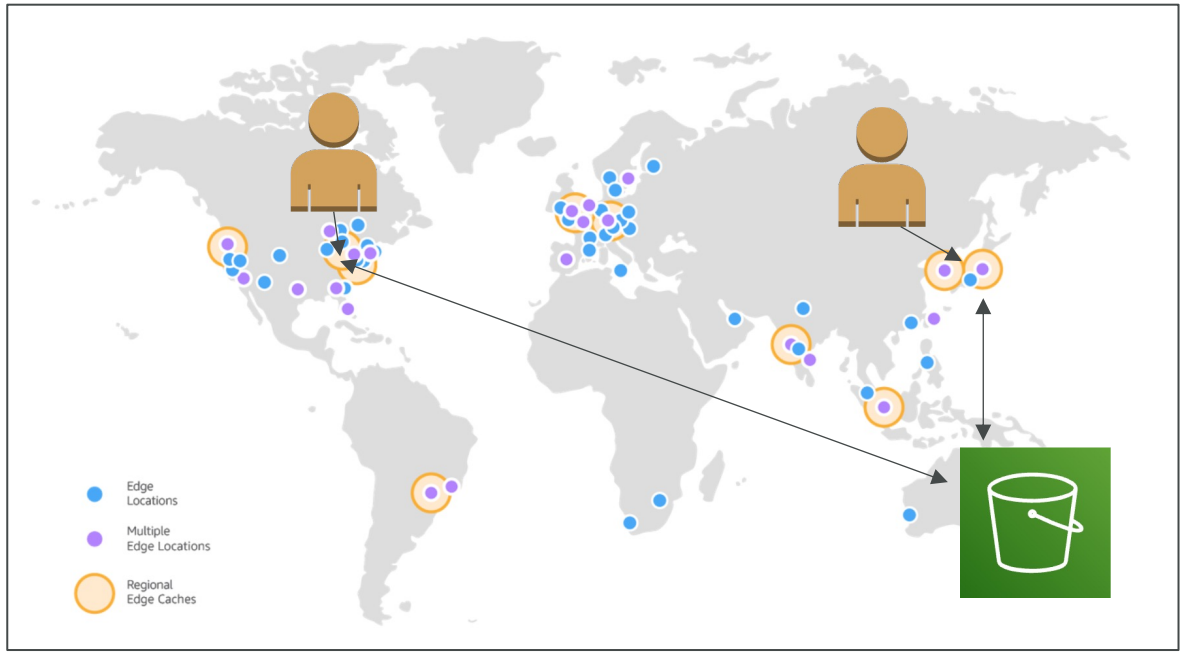
Route 53 Routing Policies





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

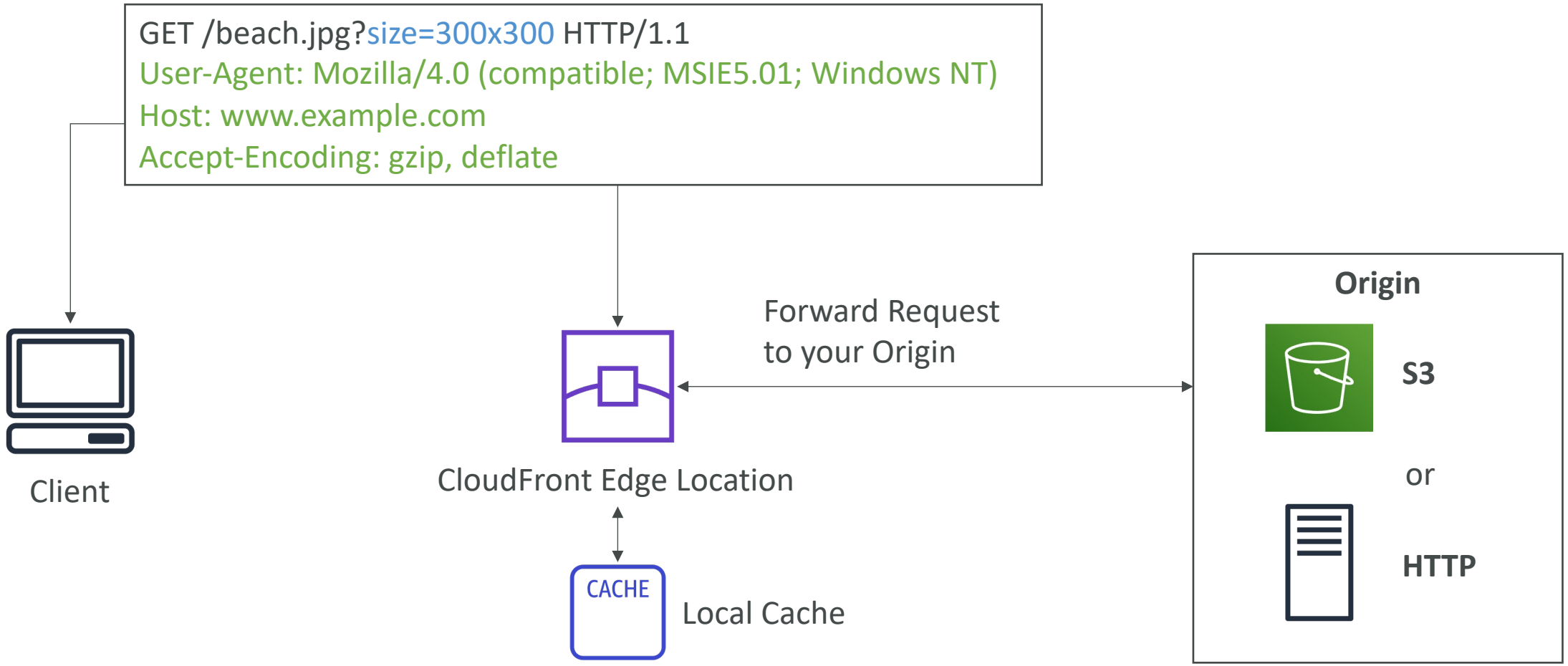


Source: https://aws.amazon.com/cloudfront/features/?nc=sn&loc=2

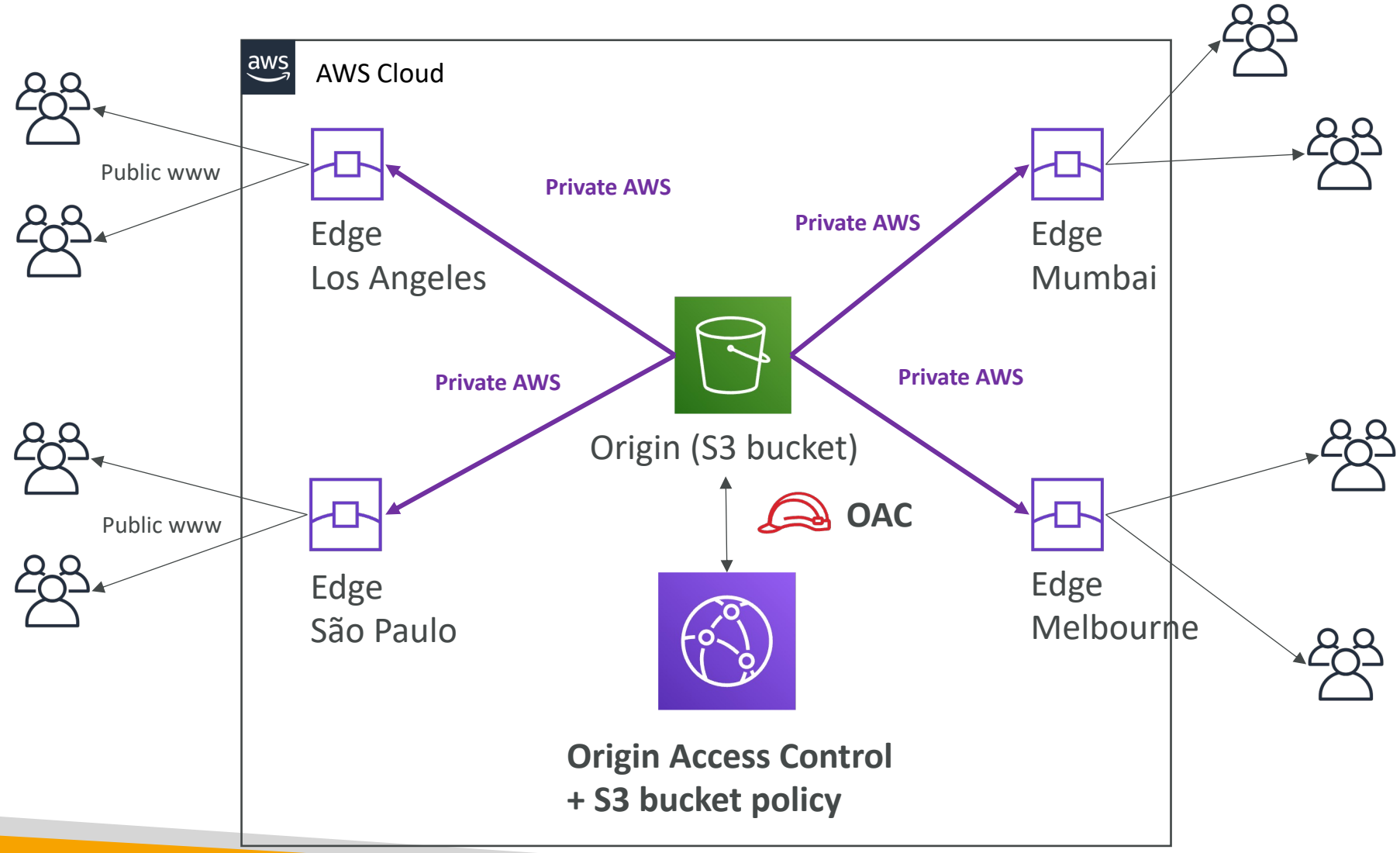
CloudFront – Origins

- 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



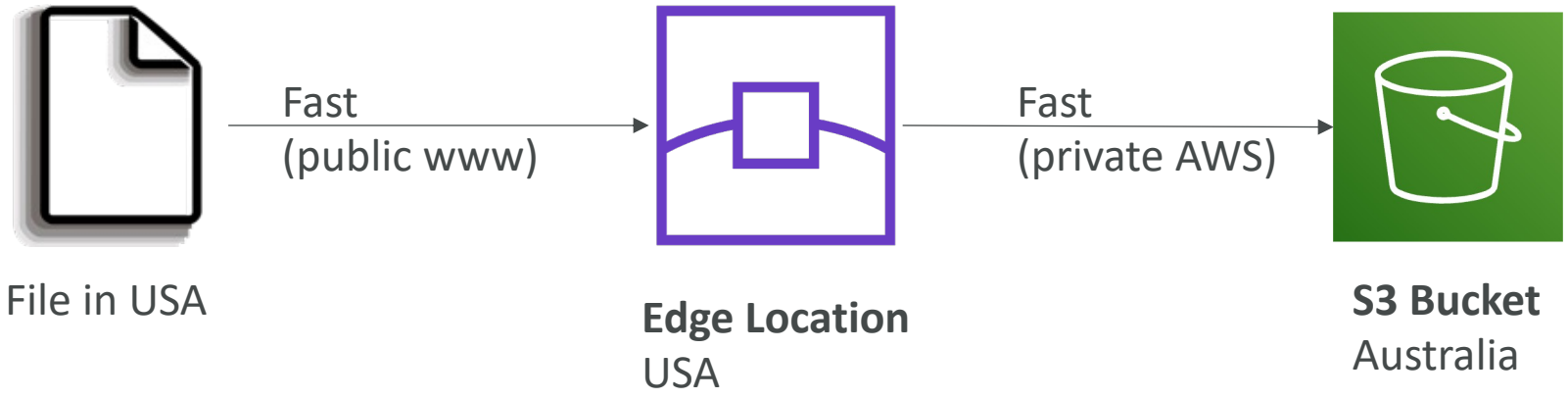
CloudFront vs S3 Cross Region Replication

- 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**

S3 Transfer Acceleration

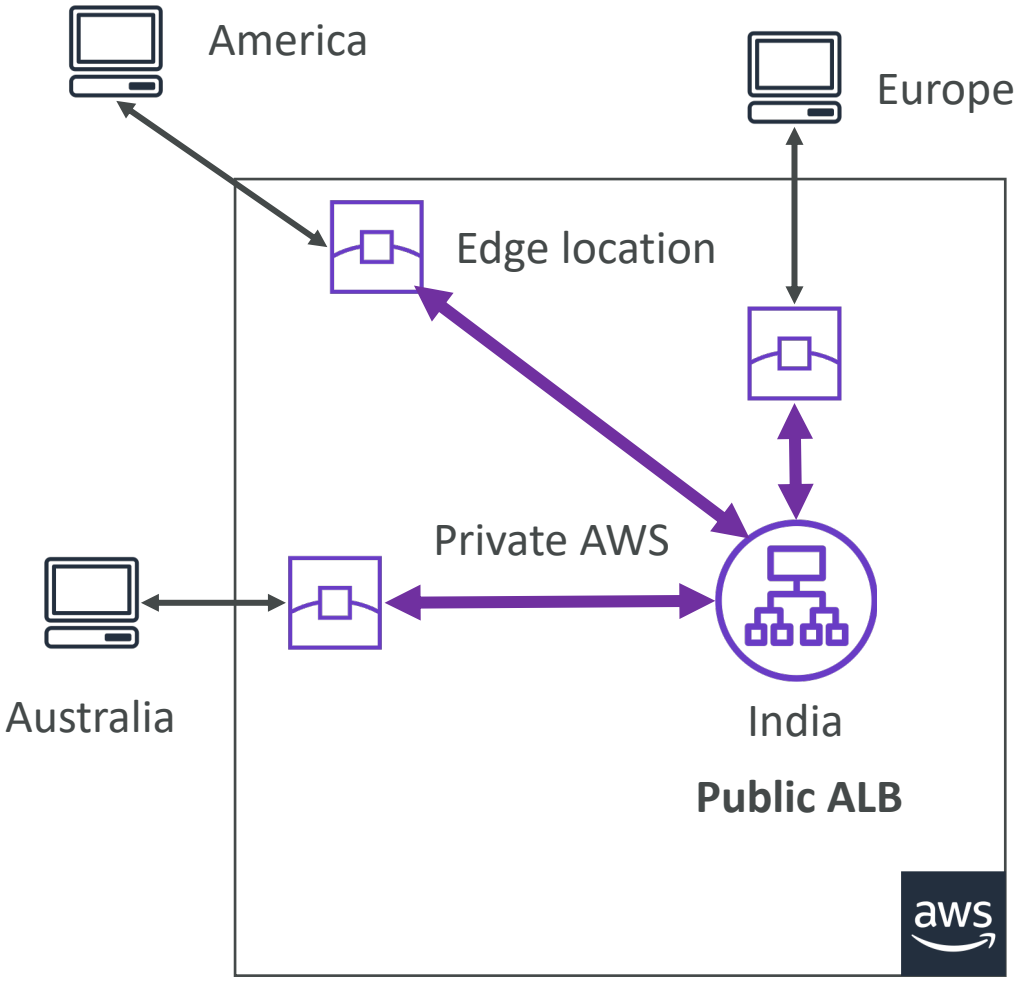
- Increase transfer speed by transferring file to an AWS edge location which will forward the data to the S3 bucket in the target region



Test the tool at: https://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/accelerate-speed-comparsion.html

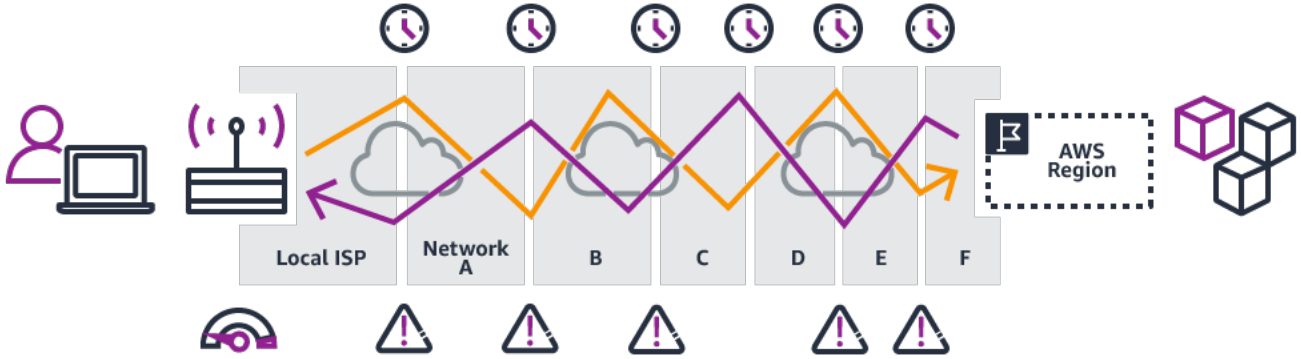
AWS Global Accelerator

- Improve global application availability and performance using the AWS global network
- Leverage the AWS internal network to optimize the route to your application (60% improvement)
- 2 Anycast IP are created for your application and traffic is sent through Edge Locations
- The Edge locations send the traffic to your application

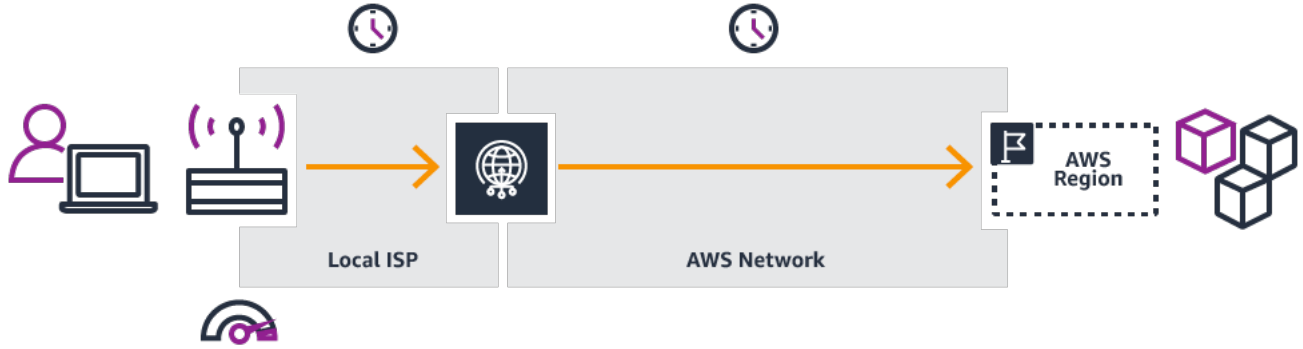


AWS Global Accelerator

Without Global Accelerator



With Global Accelerator



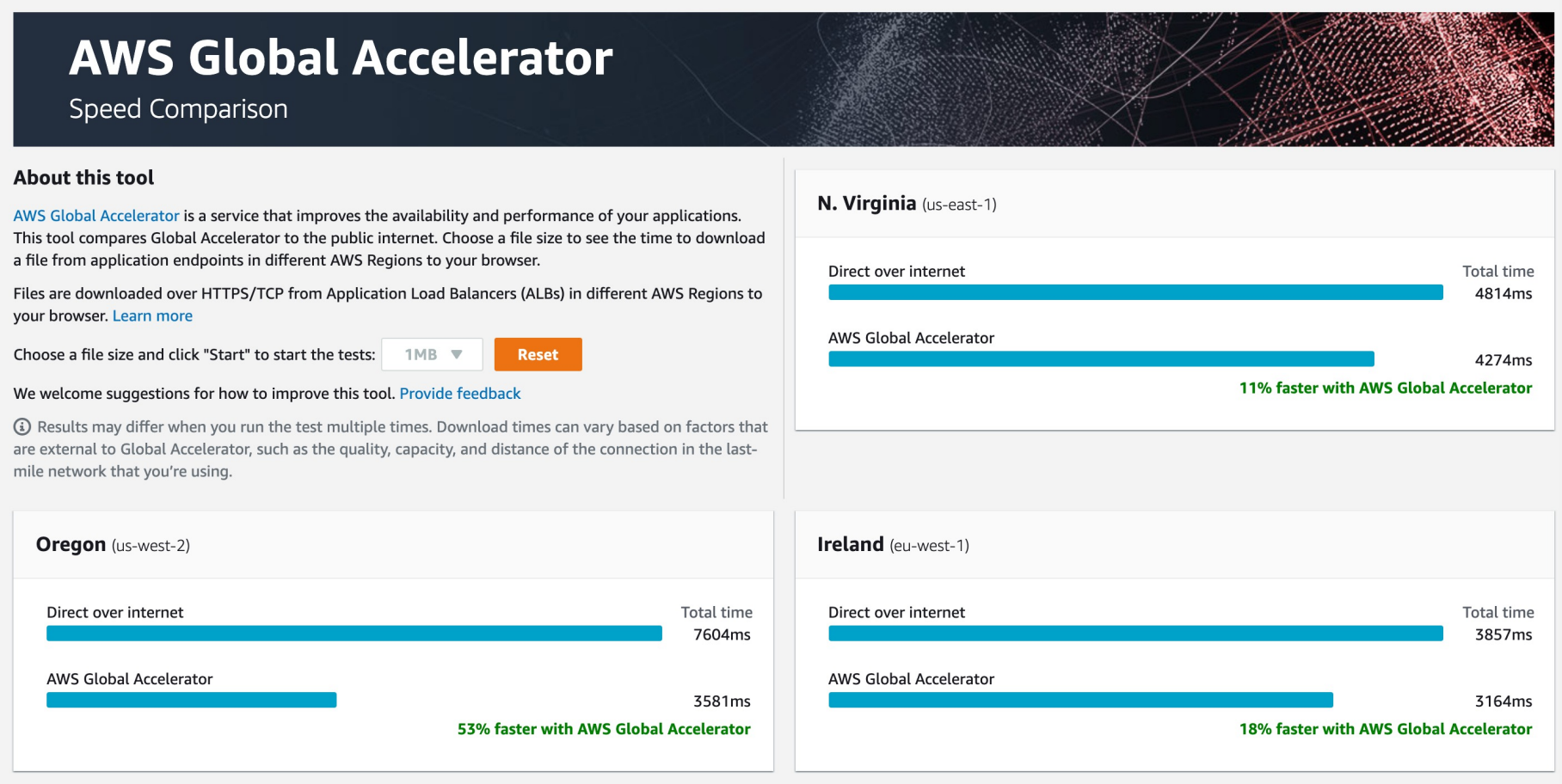
https://aws.amazon.com/global-accelerator

AWS Global Accelerator vs CloudFront

- They both use the AWS global network and its edge locations around the world
- Both services integrate with AWS Shield for DDoS protection.
- CloudFront – Content Delivery Network
  - Improves performance for your cacheable content (such as images and videos)
  - Content is served at the edge

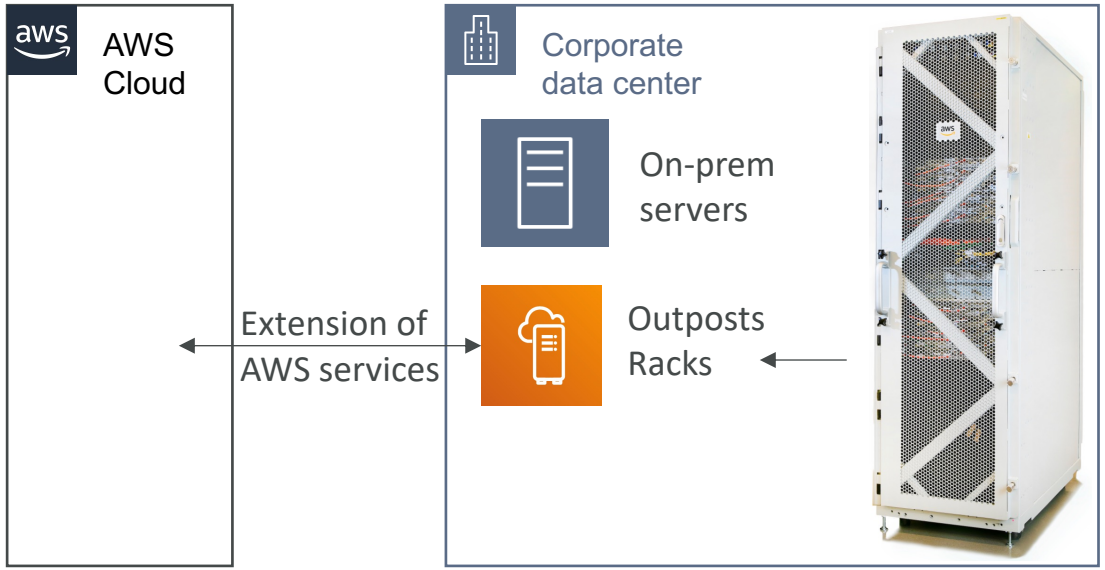
- Global Accelerator
  - No caching, proxying packets at the edge to applications running in one or more AWS Regions.
  - Improves performance for a wide range of applications over TCP or UDP
  - Good for HTTP use cases that require static IP addresses
  - Good for HTTP use cases that required deterministic, fast regional failover

https://speedtest.globalaccelerator.aws/#/



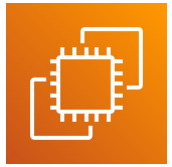
AWS Outposts

- **Hybrid Cloud**: businesses that keep an on-premises infrastructure alongside a cloud infrastructure
- Therefore, two ways of dealing with IT systems:
  - One for the AWS cloud (using the AWS console, CLI, and AWS APIs)
  - One for their on-premises infrastructure
- **AWS Outposts are "server racks"** that offers the same AWS infrastructure, services, APIs & tools to build your own applications on-premises just as in the cloud
- **AWS will setup and manage "Outposts Racks"** within your on-premises infrastructure and you can start leveraging AWS services on-premises
- You are responsible for the Outposts Rack physical security



AWS Outposts

- Benefits:
  - Low-latency access to on-premises systems
  - Local data processing
  - Data residency
  - Easier migration from on-premises to the cloud
  - Fully managed service
- Some services that work on Outposts:


Amazon EC2


Amazon EBS


Amazon S3


Amazon EKS

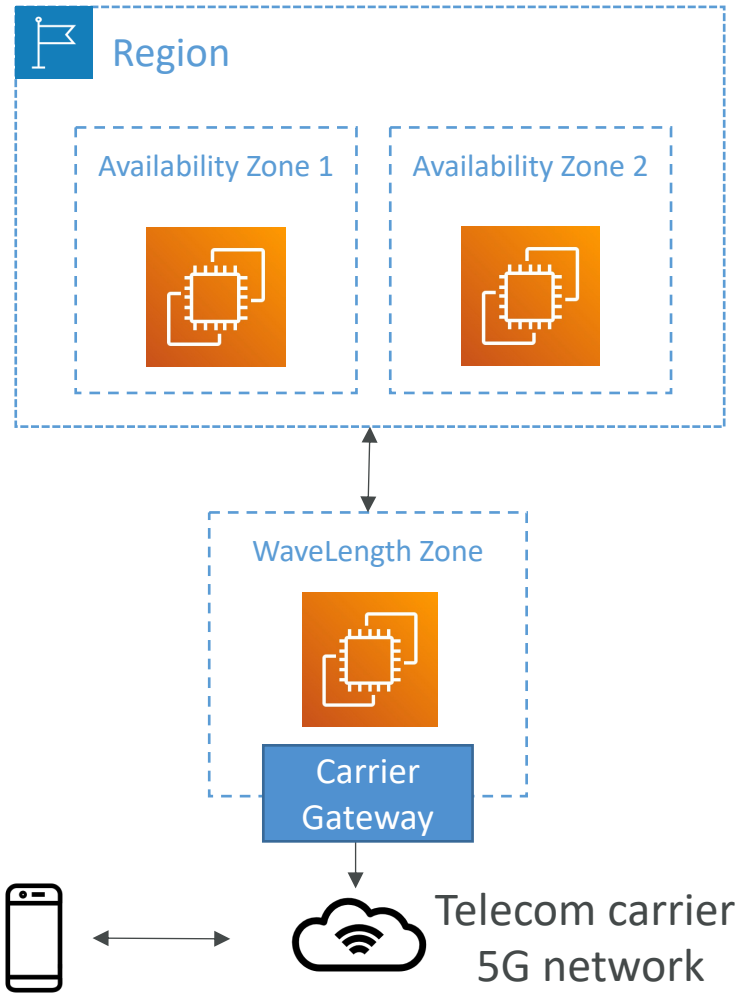

Amazon ECS


Amazon RDS


Amazon EMR

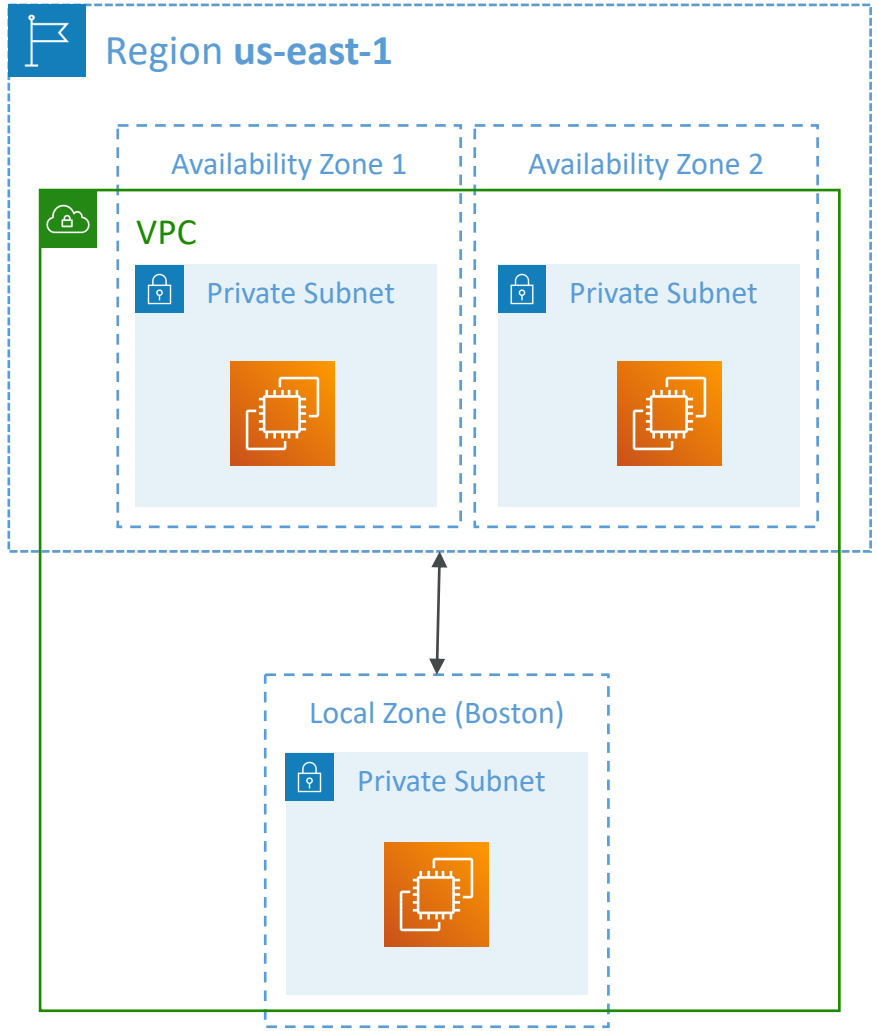
AWS WaveLength

- **WaveLength Zones** are infrastructure deployments embedded within the telecommunications providers' datacenters at the edge of the 5G networks
- Brings AWS services to the edge of the 5G networks
- Example: EC2, EBS, VPC…
- Ultra-low latency applications through 5G networks
- Traffic doesn't leave the Communication Service Provider's (CSP) network
- High-bandwidth and secure connection to the parent AWS Region
- No additional charges or service agreements
- Use cases: Smart Cities, ML-assisted diagnostics, Connected Vehicles, Interactive Live Video Streams, AR/VR, Real-time Gaming, …



AWS Local Zones

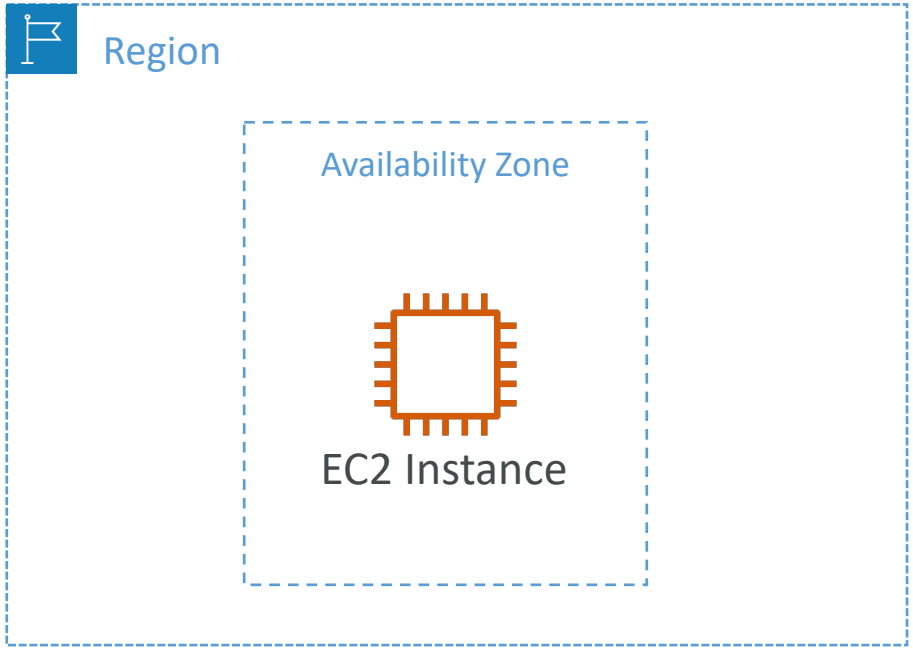
- Places AWS compute, storage, database, and other selected AWS services **closer to end users to run latency-sensitive applications**
- Extend your VPC to more locations – "**Extension of an AWS Region**"
- Compatible with EC2, RDS, ECS, EBS, ElastiCache, Direct Connect …
- Example:
  - **AWS Region:** N.Virginia (us-east-1)
  - **AWS Local Zones:** Boston, Chicago, Dallas, Houston, Miami, …



Global Applications Architecture

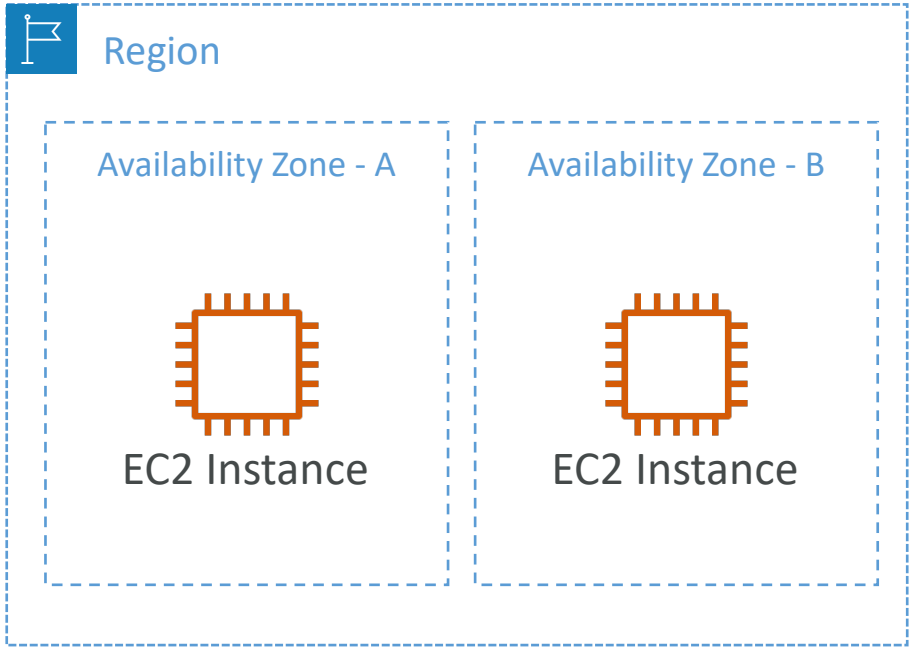
Single Region, **Single** AZ

- ✗ High Availability
- ✗ Global Latency
- Difficulty

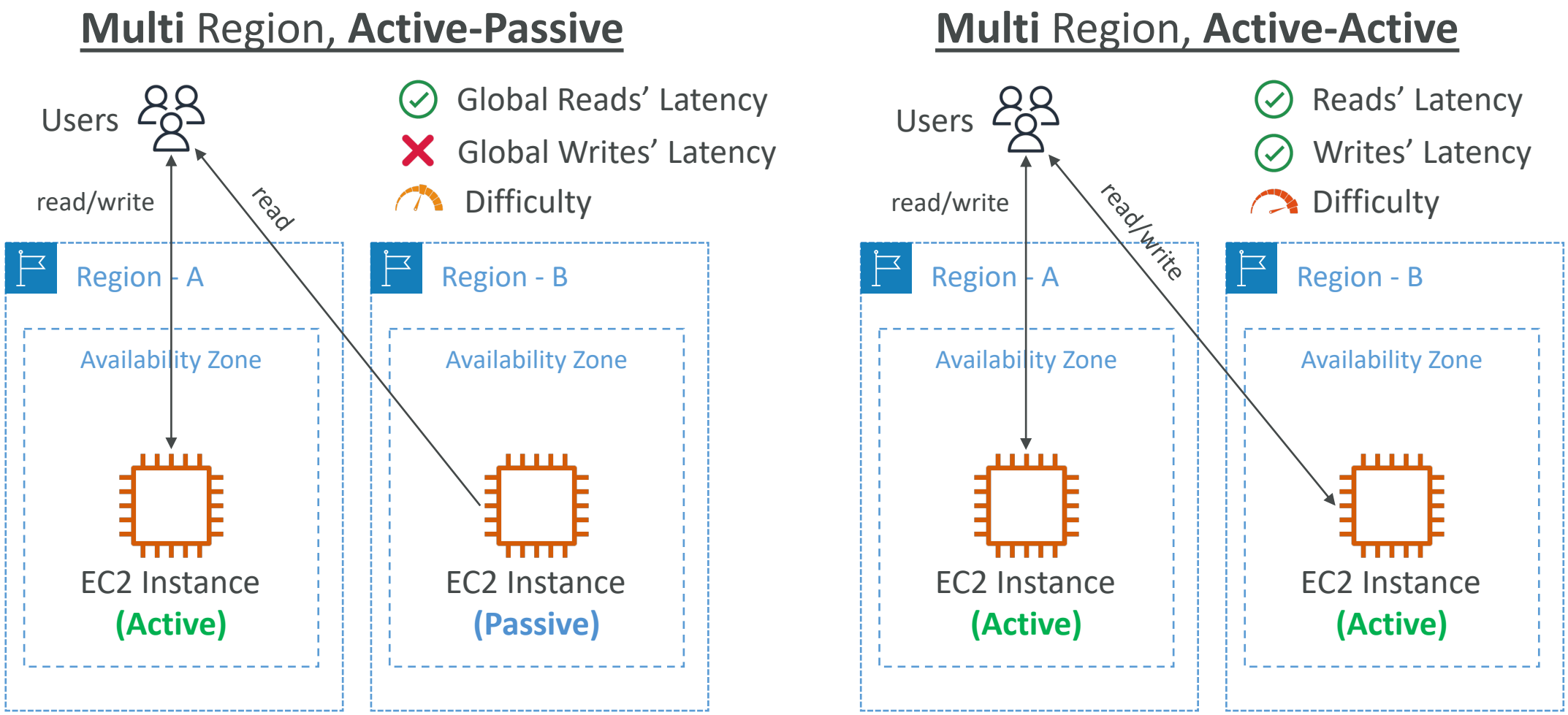


Single Region, **Multi** AZ

- ✓ High Availability
- ✗ Global Latency
- Difficulty

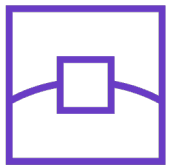


Global Applications Architecture



Global Applications in AWS - Summary






- Global DNS: Route 53
  - Great to route users to the closest deployment with least latency
  - Great for disaster recovery strategies
- Global Content Delivery Network (CDN): CloudFront
  - Replicate part of your application to AWS Edge Locations – decrease latency
  - Cache common requests – improved user experience and decreased latency
- S3 Transfer Acceleration
  - Accelerate global uploads & downloads into Amazon S3
- AWS Global Accelerator
  - Improve global application availability and performance using the AWS global network

Global Applications in AWS - Summary

- AWS Outposts
  - Deploy Outposts Racks in your own Data Centers to extend AWS services

- AWS WaveLength
  - Brings AWS services to the edge of the 5G networks
  - Ultra-low latency applications

- AWS Local Zones
  - Bring AWS resources (compute, database, storage, …) closer to your users
  - Good for latency-sensitive applications

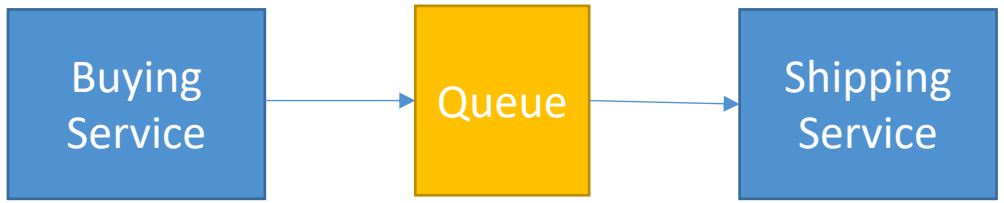
Cloud Integration Section

Section Introduction

- When we start deploying multiple applications, they will inevitably need to communicate with one another
- There are two patterns of application communication

1) Synchronous communications (application to application)

2) Asynchronous / Event based (application to queue to application)

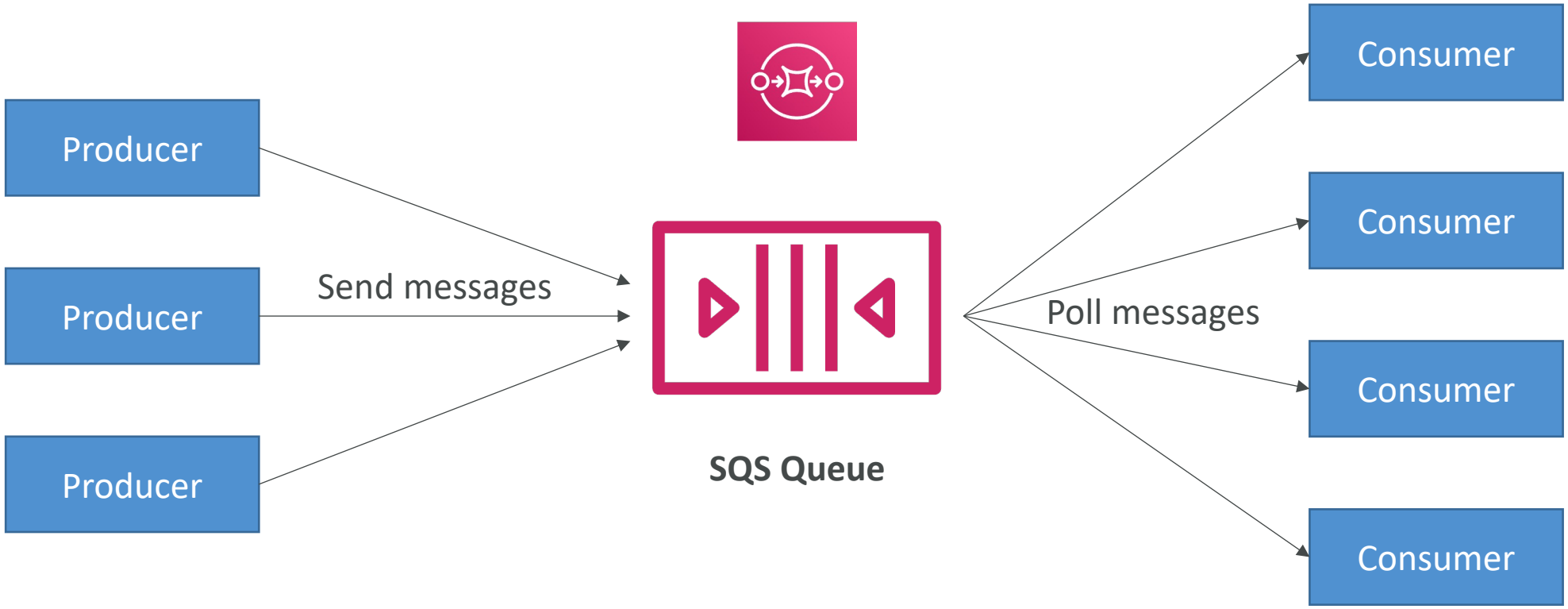


Section Introduction

- Synchronous between applications can be problematic if there are sudden spikes of traffic
- What if you need to suddenly encode 1000 videos but usually it's 10?

- In that case, it's better to **decouple** your applications:
  - using SQS: queue model
  - using SNS: pub/sub model
  - using Kinesis: real-time data streaming model
- These services can scale independently from our application!

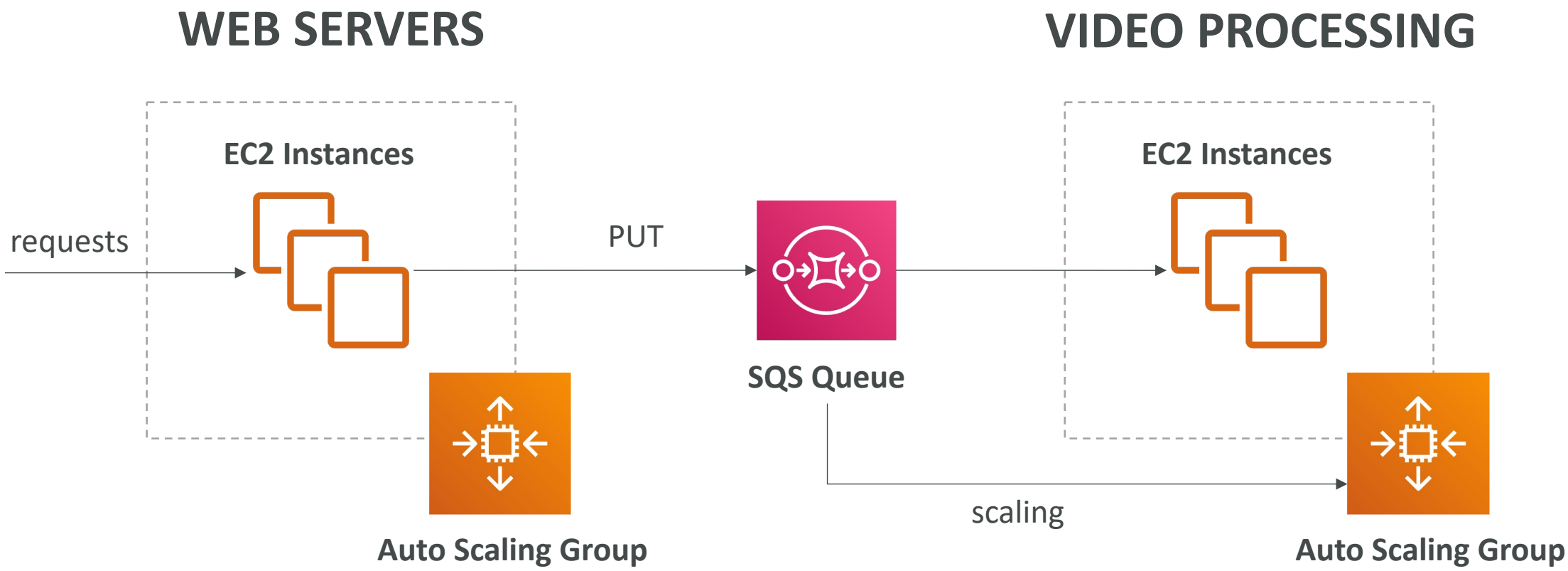
Amazon SQS – Simple Queue Service
What's a queue?



Amazon SQS – Standard Queue

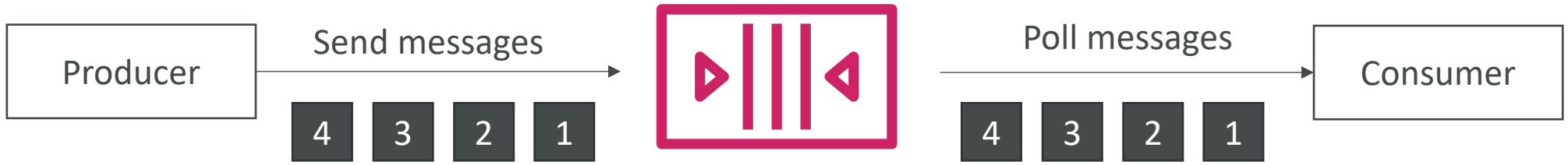
- Oldest AWS offering (over 10 years old)
- Fully managed service (~serverless), use to **decouple** applications
- Scales from 1 message per second to 10,000s per second
- Default retention of messages: 4 days, maximum of 14 days
- No limit to how many messages can be in the queue
- **Messages are deleted after they're read by consumers**
- Low latency (<10 ms on publish and receive)
- **Consumers share the work to read messages & scale horizontally**

SQS to decouple between application tiers



Amazon SQS – FIFO Queue

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



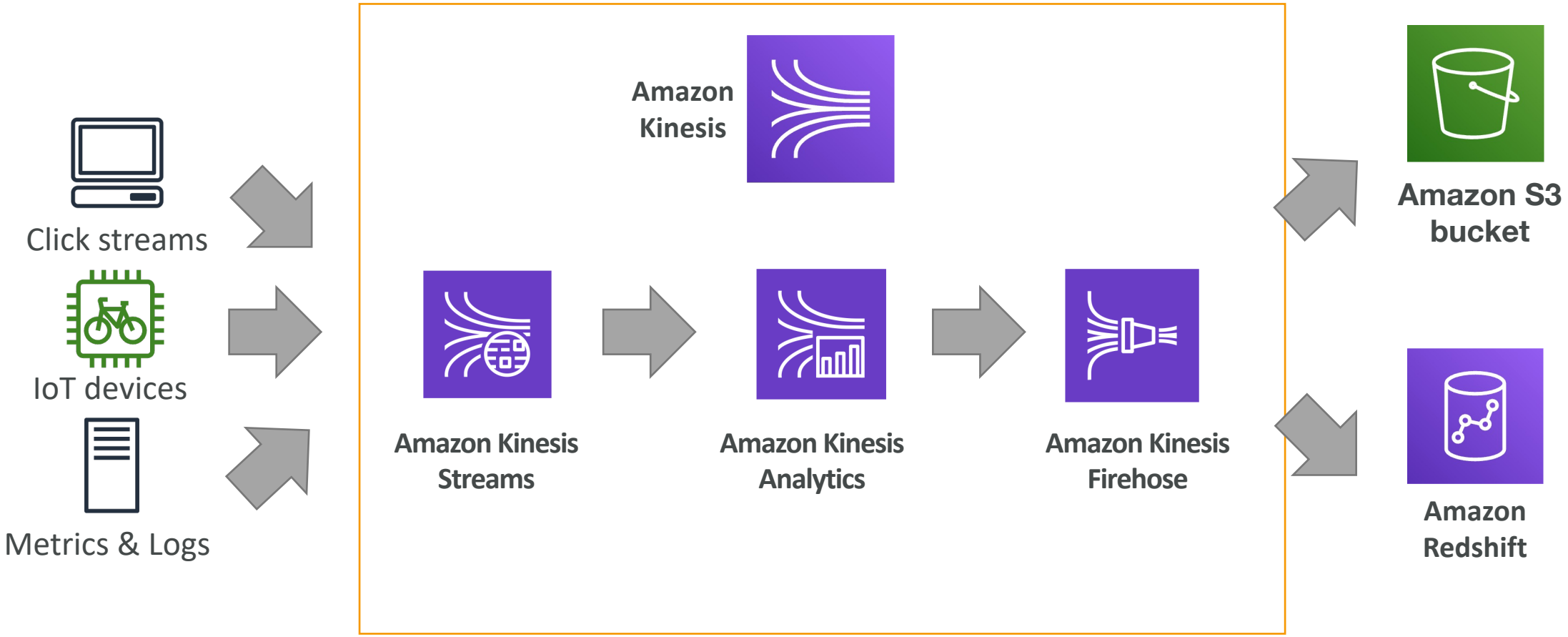
- Messages are processed in order by the consumer

Amazon Kinesis

- For the exam: **Kinesis = real-time big data streaming**
- **Managed service to collect, process, and analyze real-time streaming data at any scale**

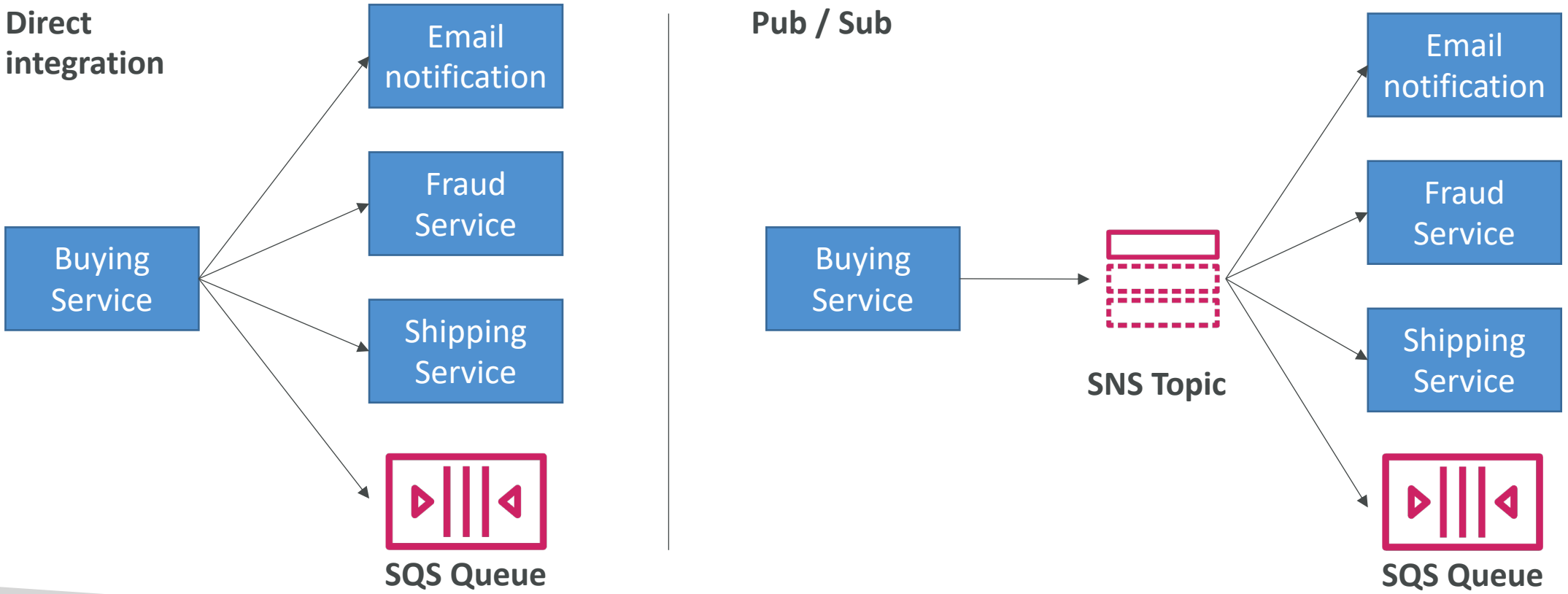
- Too detailed for the Cloud Practitioner exam but good to know:
  - **Kinesis Data Streams:** low latency streaming to ingest data at scale from hundreds of thousands of sources
  - **Kinesis Data Firehose:** load streams into S3, Redshift, ElasticSearch, etc…
  - **Kinesis Data Analytics:** perform real-time analytics on streams using SQL
  - **Kinesis Video Streams:** monitor real-time video streams for analytics or ML

Kinesis (high level overview)



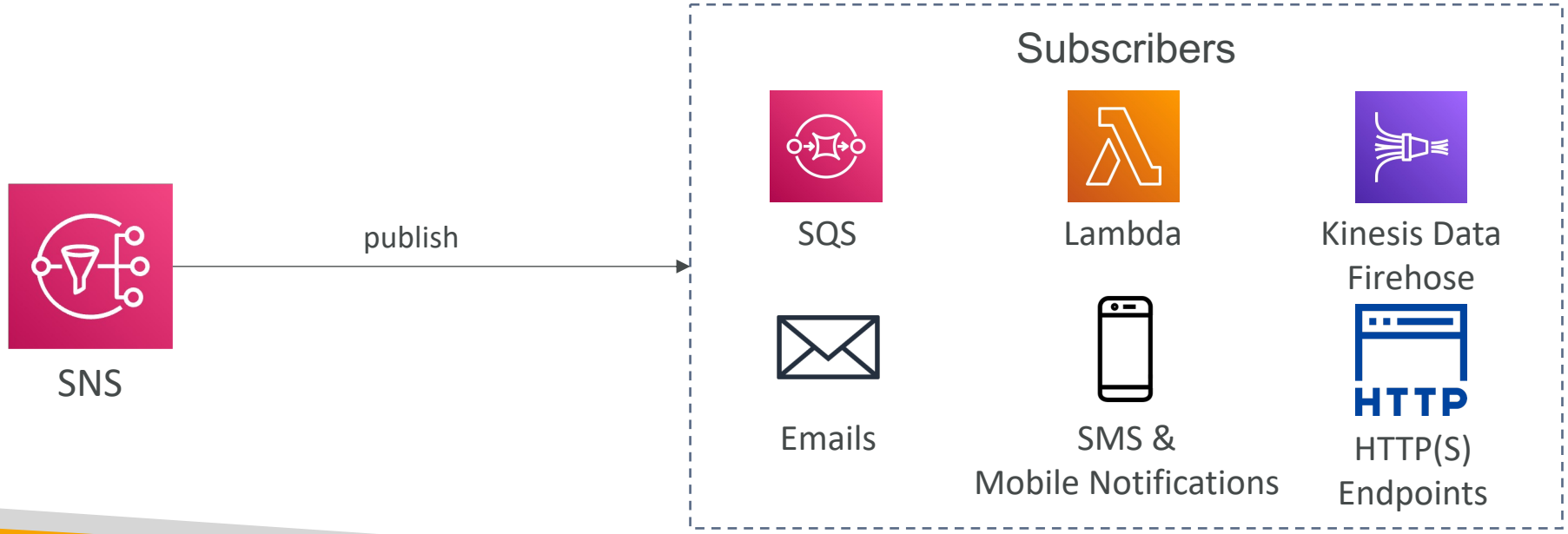
Amazon SNS

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



Amazon SNS

- The "event **publishers**" only sends message to one SNS topic
- As many "event **subscribers**" as we want to listen to the SNS topic notifications
- Each subscriber to the topic **will get all the messages**
- Up to 12,500,000 subscriptions per topic, 100,000 topics limit



Amazon MQ

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

RabbitMQ™

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

Integration Section – Summary

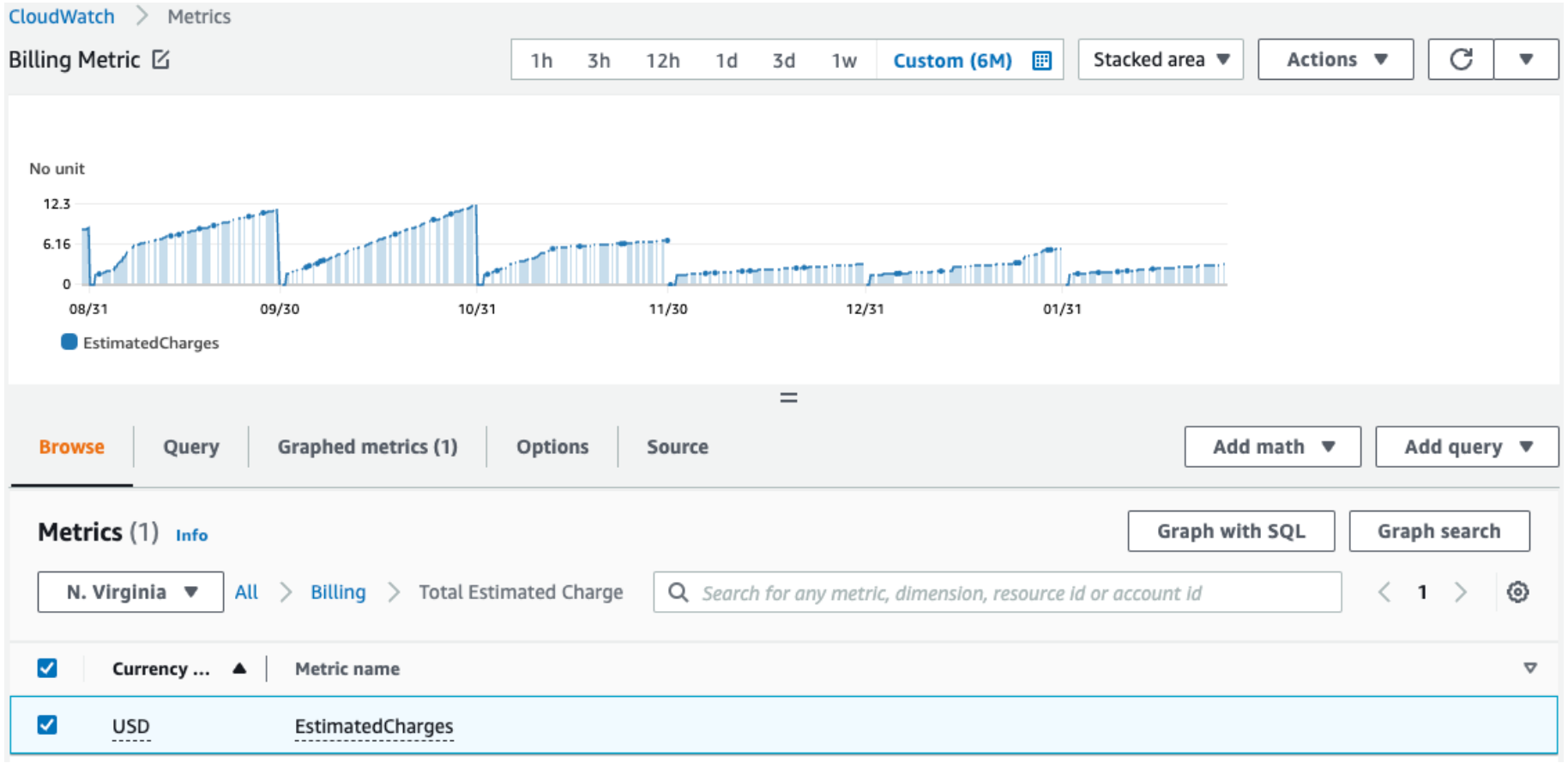
- SQS:
  - Queue service in AWS
  - Multiple Producers, messages are kept up to 14 days
  - Multiple Consumers share the read and delete messages when done
  - Used to **decouple** applications in AWS
- SNS:
  - Notification service in AWS
  - Subscribers: Email, Lambda, SQS, HTTP, Mobile…
  - Multiple Subscribers, send all messages to all of them
  - No message retention
- **Kinesis**: real-time data streaming, persistence and analysis
- **Amazon MQ**: managed message broker for ActiveMQ and RabbitMQ in the cloud (MQTT, AMQP.. protocols)

Cloud Monitoring Section

Amazon CloudWatch Metrics

- CloudWatch provides metrics for *every* services in AWS
- **Metric** is a variable to monitor (CPUUtilization, NetworkIn…)
- Metrics have **timestamps**
- Can create **CloudWatch dashboards** of metrics

Example: CloudWatch Billing metric



Important Metrics

- **EC2 instances:** CPU Utilization, Status Checks, Network (not RAM)
  - Default metrics every 5 minutes
  - Option for Detailed Monitoring ($$$): metrics every 1 minute
- **EBS volumes:** Disk Read/Writes
- **S3 buckets:** BucketSizeBytes, NumberOfObjects, AllRequests
- **Billing:** Total Estimated Charge (only in us-east-1)
- **Service Limits:** how much you've been using a service API
- **Custom metrics:** push your own metrics

Amazon CloudWatch Alarms

- Alarms are used to trigger notifications for any metric
- Alarms actions…
  - **Auto Scaling:** increase or decrease EC2 instances "desired" count
  - **EC2 Actions:** stop, terminate, reboot or **recover an EC2 instance**
  - **SNS notifications:** send a notification into an SNS topic
- Various options (sampling, %, max, min, etc…)
- Can choose the period on which to evaluate an alarm
- Example: create **a billing alarm** on the CloudWatch Billing metric
- Alarm States: OK. INSUFFICIENT_DATA, ALARM

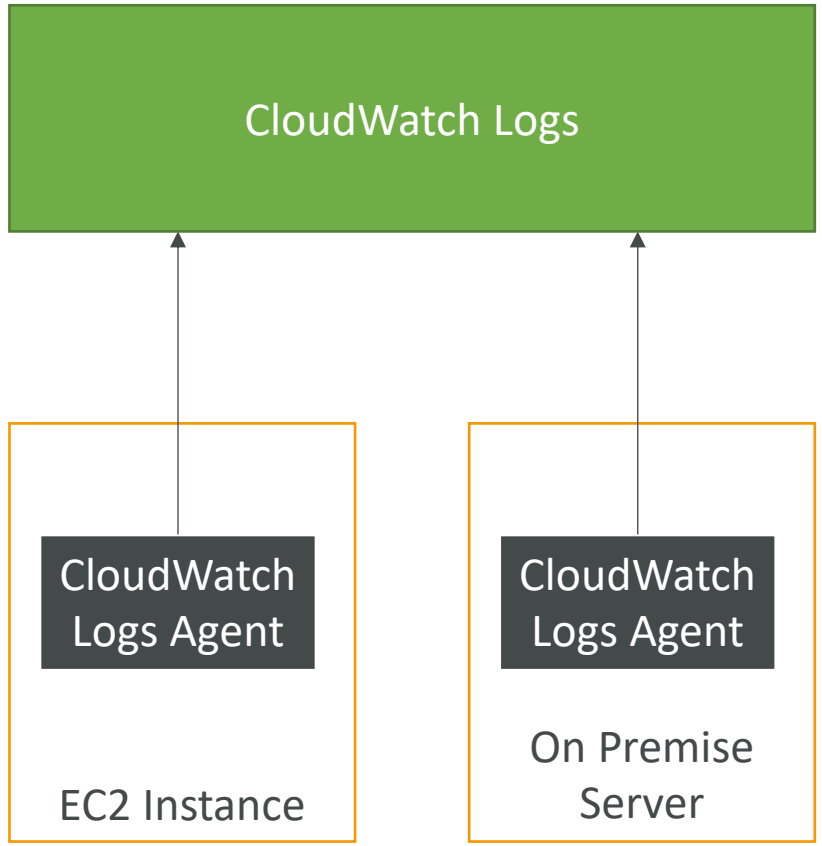
Amazon CloudWatch Logs

- CloudWatch Logs can collect log from:
  - Elastic Beanstalk: collection of logs from application
  - ECS: collection from containers
  - AWS Lambda: collection from function logs
  - CloudTrail based on filter
  - **CloudWatch log agents: on EC2 machines or on-premises servers**
  - Route53: Log DNS queries

- Enables **real-time monitoring** of logs
- Adjustable CloudWatch Logs retention

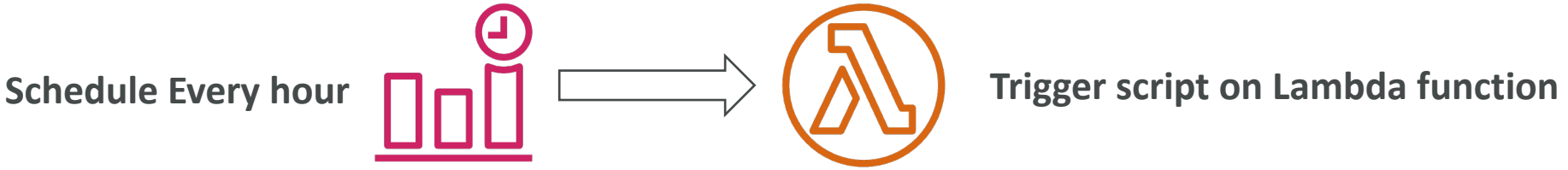
CloudWatch Logs for EC2

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

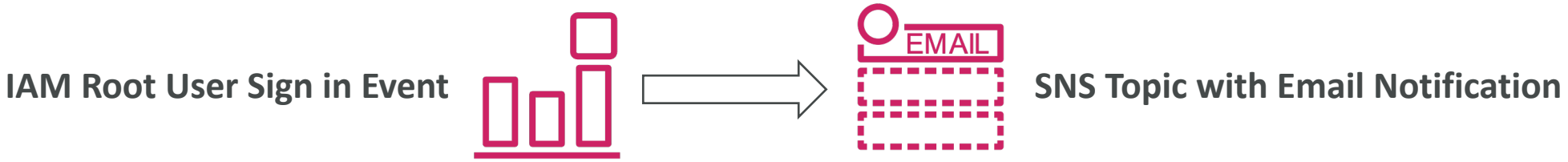


Amazon EventBridge (formerly CloudWatch Events)

- Schedule: Cron jobs (scheduled scripts)

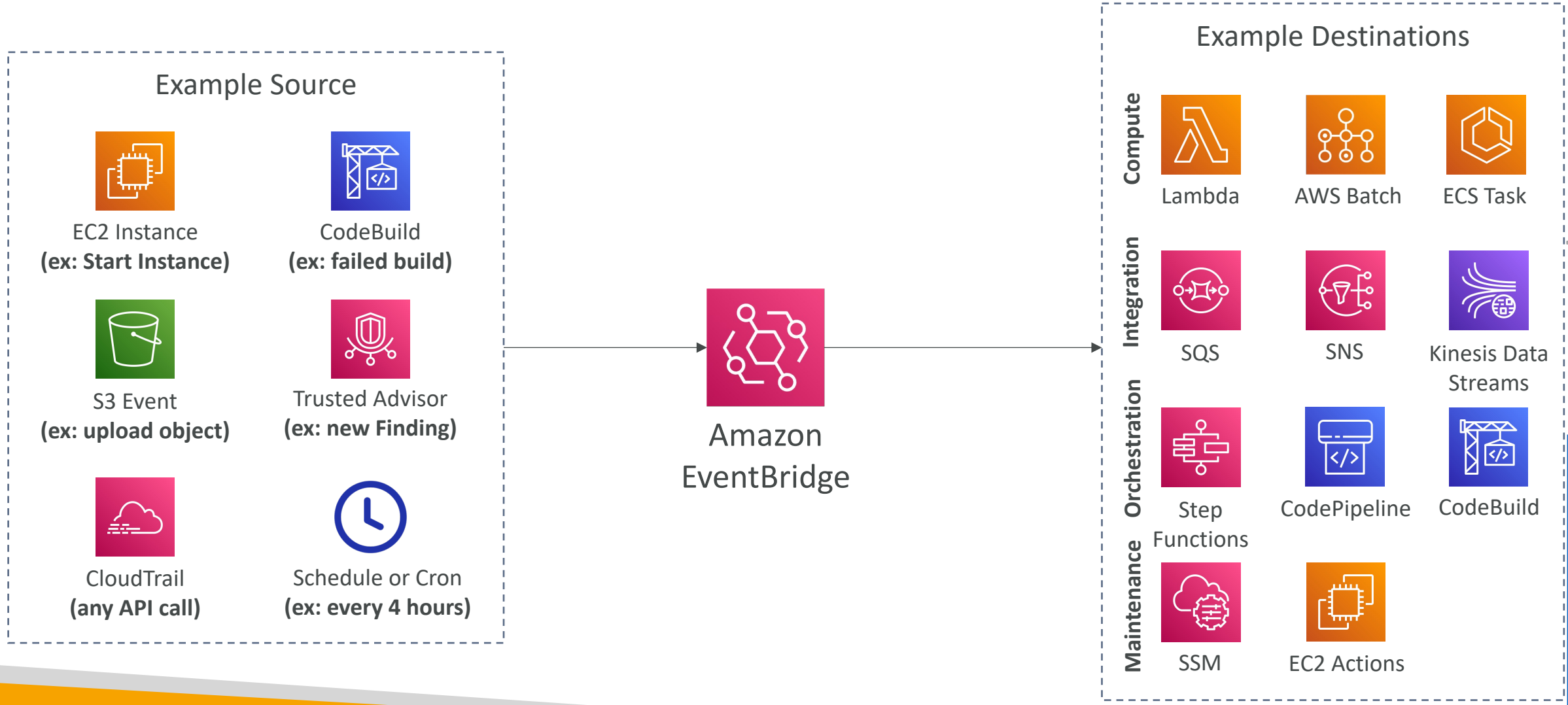


- Event Pattern: Event rules to react to a service doing something

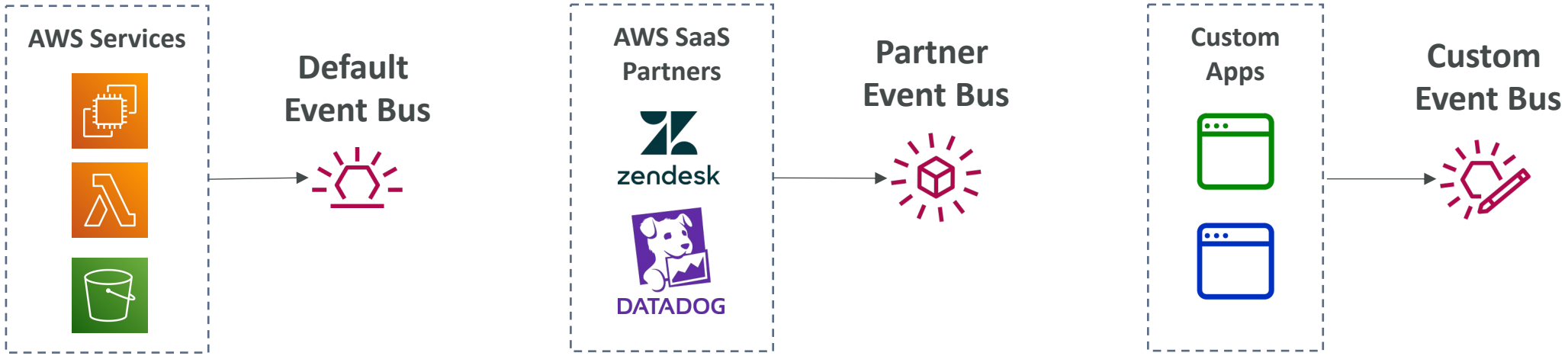


- Trigger Lambda functions, send SQS/SNS messages…

Amazon EventBridge Rules



Amazon EventBridge

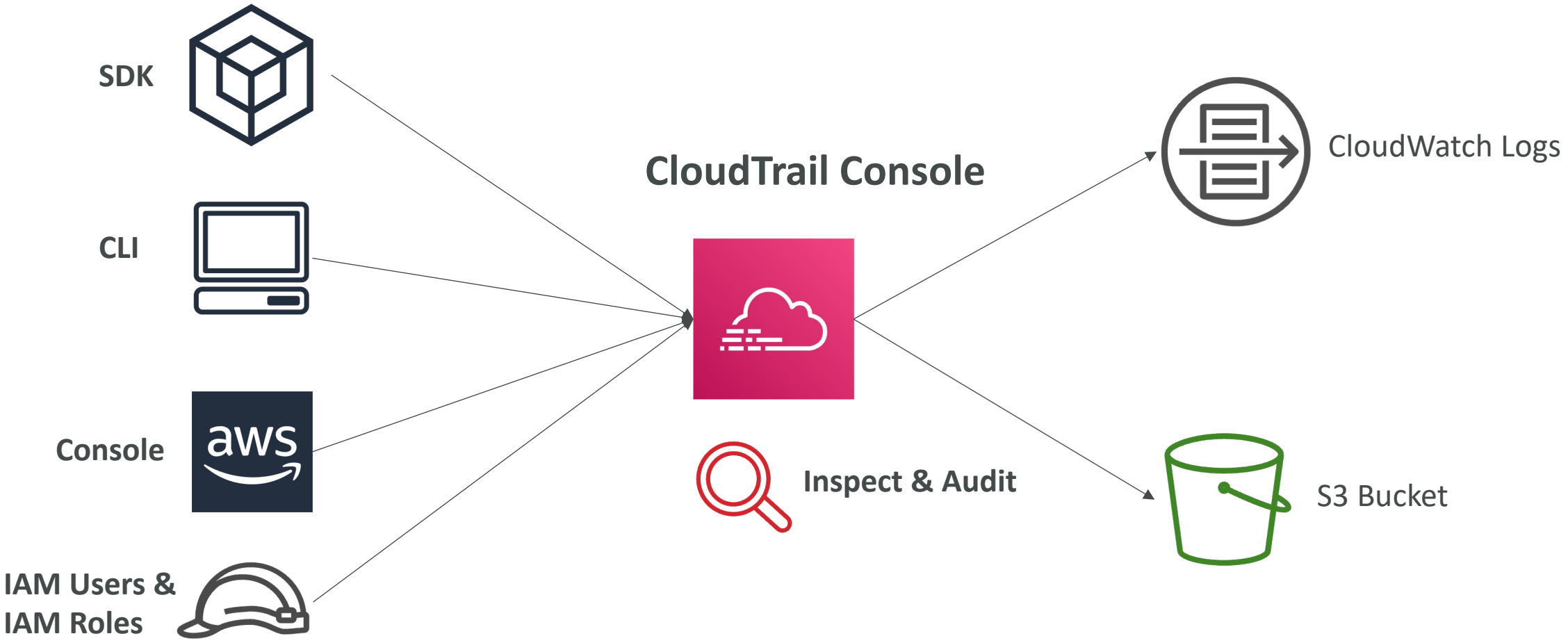


- **Schema Registry**: model event schema
- You can **archive events** (all/filter) sent to an event bus (indefinitely or set period)
- Ability to **replay archived events**

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

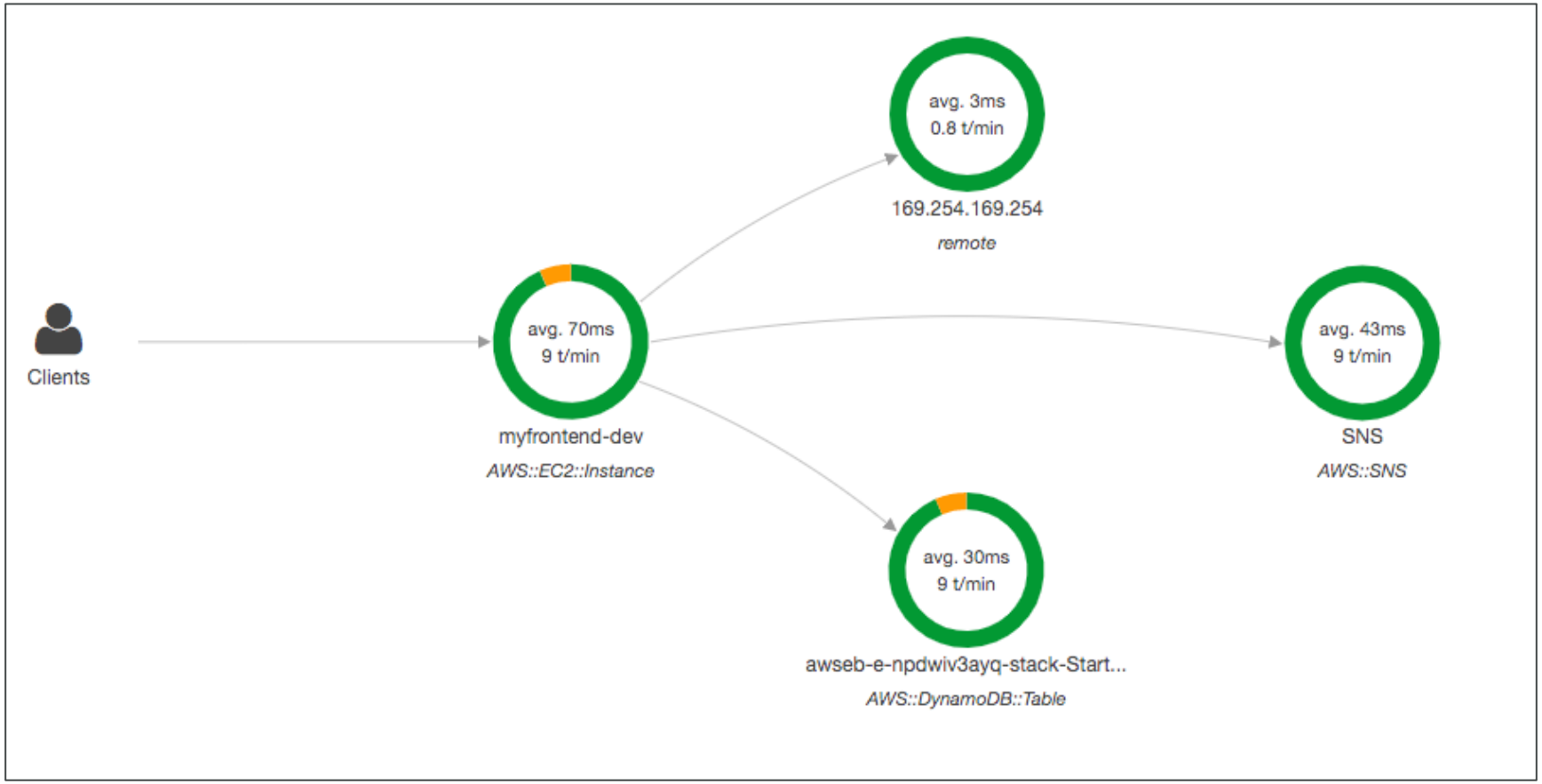


AWS X-Ray

- 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

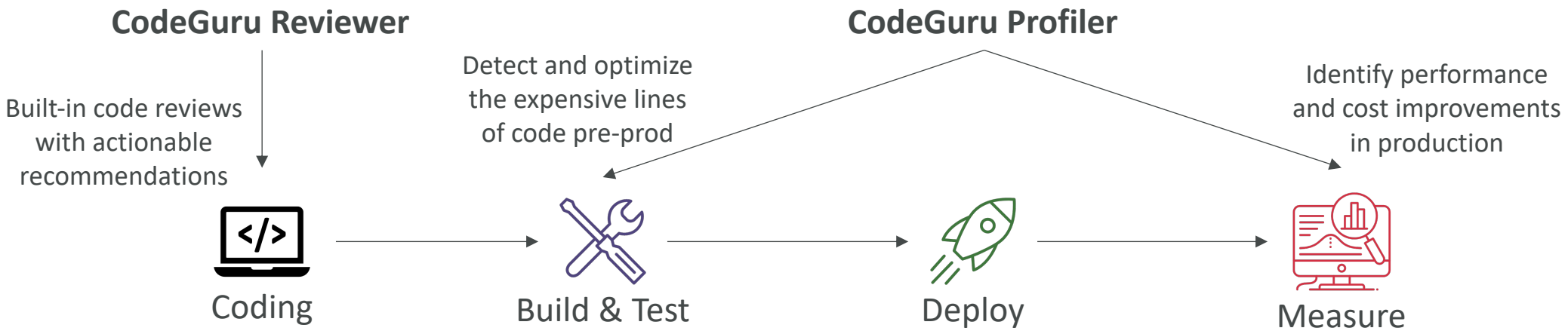


AWS X-Ray advantages

- 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

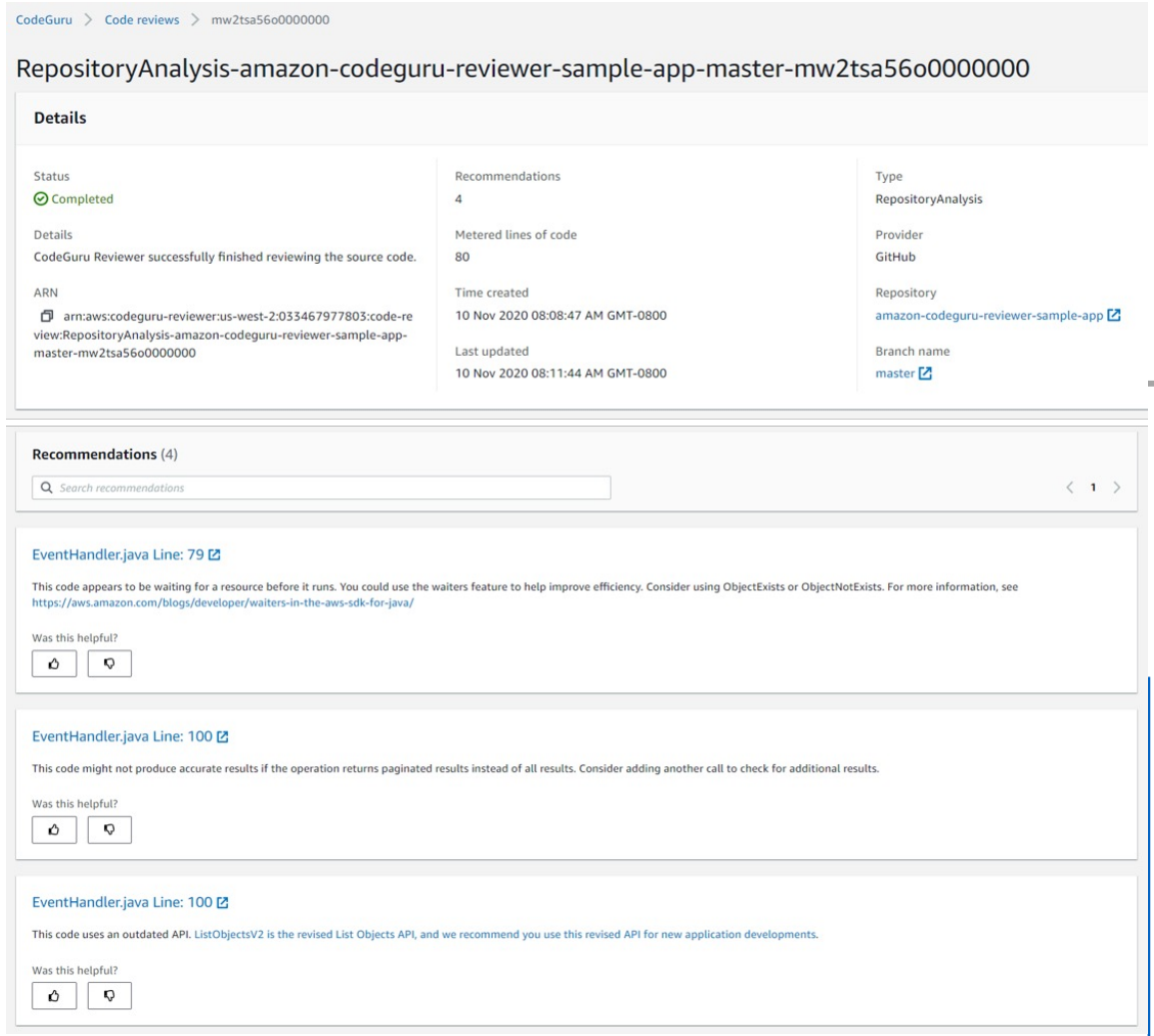
Amazon CodeGuru

- An ML-powered service for **automated code reviews** and **application performance recommendations**
- Provides two functionalities
  - **CodeGuru Reviewer:** automated code reviews for static code analysis (development)
  - **CodeGuru Profiler:** visibility/recommendations about application performance during runtime (production)



Amazon CodeGuru Reviewer

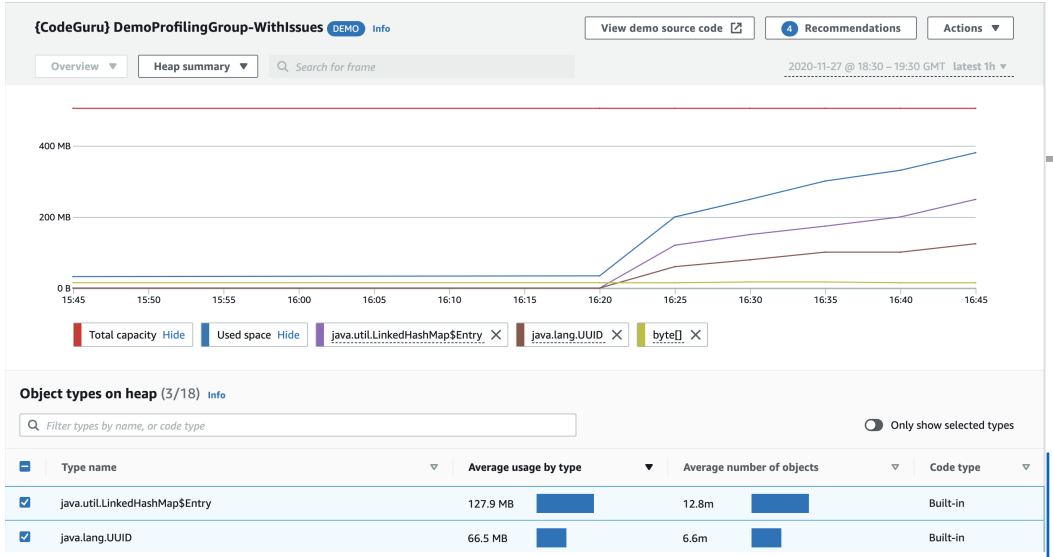
- Identify critical issues, security vulnerabilities, and hard-to-find bugs
- Example: common coding best practices, resource leaks, security detection, input validation
- Uses Machine Learning and automated reasoning
- Hard-learned lessons across millions of code reviews on 1000s of open-source and Amazon repositories
- Supports Java and Python
- Integrates with GitHub, Bitbucket, and AWS CodeCommit



https://aws.amazon.com/codeguru/features/

Amazon CodeGuru Profiler

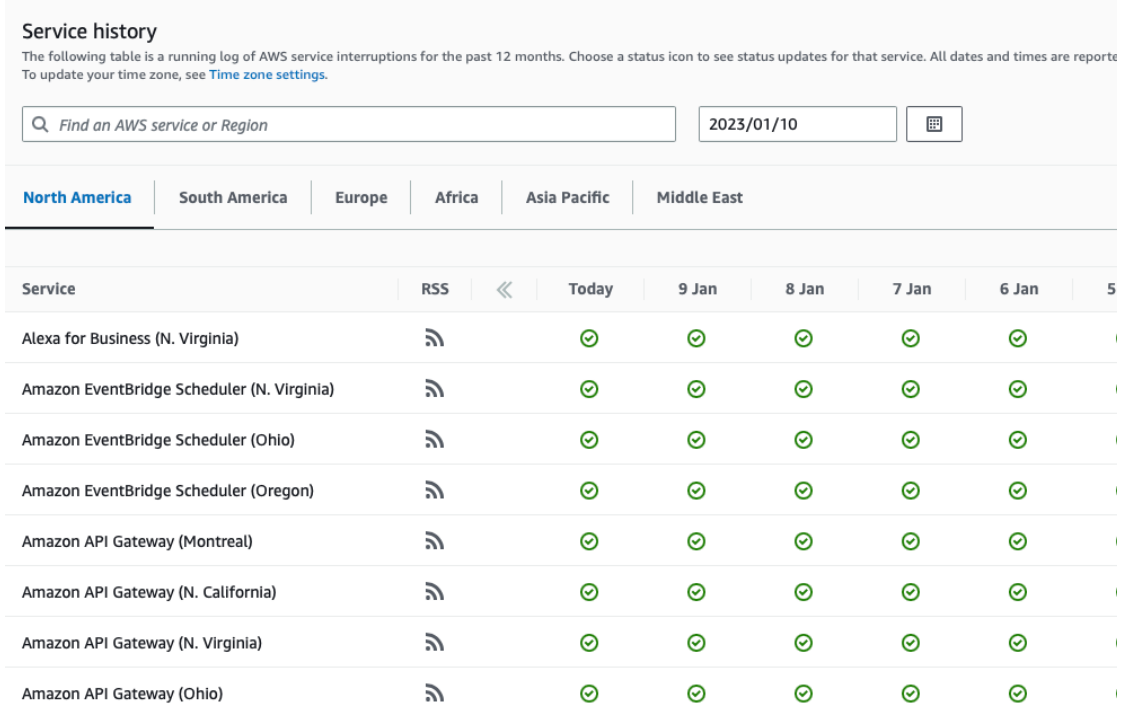
- Helps understand the runtime behavior of your application
- Example: identify if your application is consuming excessive CPU capacity on a logging routine
- Features:
  - Identify and remove code inefficiencies
  - Improve application performance (e.g., reduce CPU utilization)
  - Decrease compute costs
  - Provides heap summary (identify which objects using up memory)
  - Anomaly Detection
- Support applications running on AWS or on-premise
- Minimal overhead on application



https://aws.amazon.com/codeguru/features/

AWS Health Dashboard - Service History

- Shows all regions, all services health
- Shows historical information for each day
- Has an RSS feed you can subscribe to
- Previously called AWS Service Health Dashboard

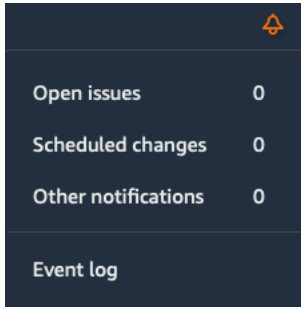


AWS Health Dashboard – Your Account

- Previously called AWS Personal Health Dashboard (PHD)
- AWS Account Health Dashboard provides **alerts and remediation guidance** when AWS is experiencing **events that may impact you.**
- While the Service Health Dashboard displays the general status of AWS services, Account Health Dashboard gives you a **personalized view into the performance and availability of the AWS services underlying your AWS resources.**
- The dashboard displays **relevant and timely information** to help you manage events in progress and provides **proactive notification** to help you plan for **scheduled activities.**
- **Can aggregate data from an entire AWS Organization**

AWS Health Dashboard – Your Account

- Global service
- Shows how AWS outages directly impact you & your AWS resources
- Alert, remediation, proactive, scheduled activities



Open and recent issues (0) | Scheduled changes (0) | Other notifications (0) | Event log

Event log

Add filter

1

Event	Status	Event category	Region / Zone Info	Start time	Last update time	Affected resources
Operational issue - EC2 (Ohio)	Closed	Issue	us-east-2	December 24, 2022 at 2:25:00 AM UTC	December 24, 2022 at 2:38:53 AM UTC	-
Operational issue - Codecatalyst (Oregon)	Closed	Issue	us-west-2	December 21, 2022 at 3:03:57 PM UTC	December 21, 2022 at 4:50:47 PM UTC	-
Operational issue - Amplify (N. Virginia)	Closed	Issue	us-east-1	December 17, 2022 at 2:24:17 PM UTC	December 17, 2022 at 2:43:21 PM UTC	-
Operational issue - Multiple services (Singapore)	Closed	Issue	ap-southeast-1	December 13, 2022 at 10:00:55 PM UTC	December 14, 2022 at 1:01:16 AM UTC	-

Monitoring Summary

- CloudWatch:
  - Metrics: monitor the performance of AWS services and billing metrics
  - Alarms: automate notification, perform EC2 action, notify to SNS based on metric
  - Logs: collect log files from EC2 instances, servers, Lambda functions…
  - Events (or EventBridge): react to events in AWS, or trigger a rule on a schedule
- CloudTrail: audit API calls made within your AWS account
- CloudTrail Insights: automated analysis of your CloudTrail Events
- X-Ray: trace requests made through your distributed applications
- AWS Health Dashboard: status of all AWS services across all regions
- AWS Account Health Dashboard: AWS events that impact your infrastructure
- Amazon CodeGuru: automated code reviews and application performance recommendations

VPC Section

VPC – Crash Course

- VPC is something you should know in depth for the AWS Certified Solutions Architect Associate & AWS Certified SysOps Administrator

- At the AWS Certified Cloud Practitioner Level, you should know about:
  - VPC, Subnets, Internet Gateways & NAT Gateways
  - Security Groups, Network ACL (NACL), VPC Flow Logs
  - VPC Peering, VPC Endpoints
  - Site to Site VPN & Direct Connect
  - Transit Gateway

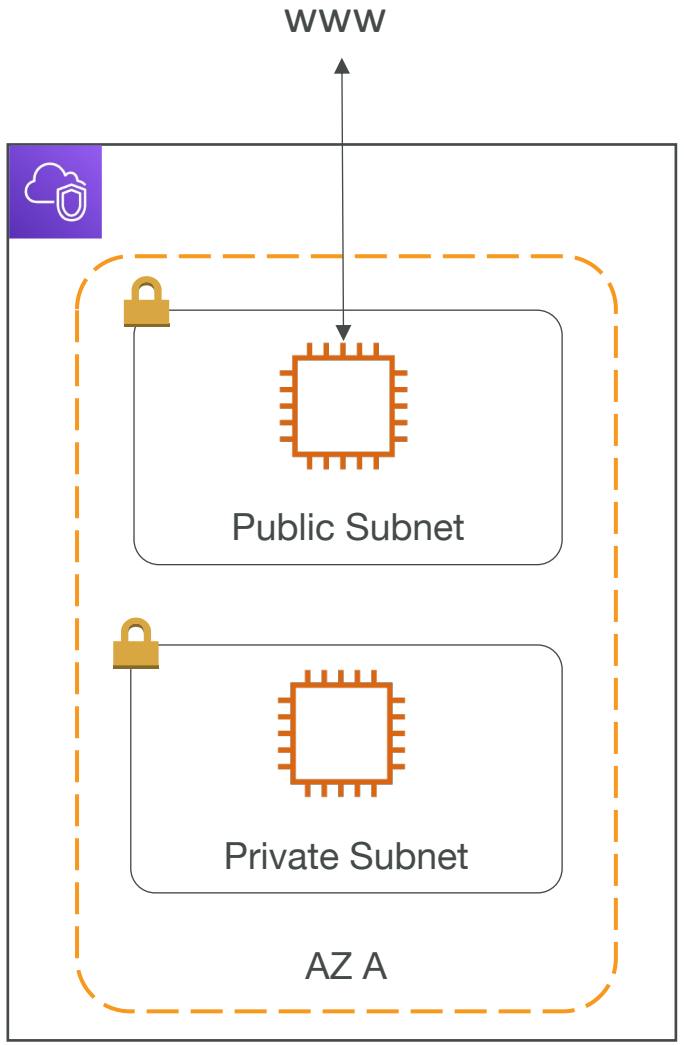
- I will just give you an overview, less than 1 or 2 questions at your exam.
- We'll have a look at the "default VPC" (created by default by AWS for you)
- There is a summary lecture at the end. It's okay if you don't understand it all

IP Addresses in AWS

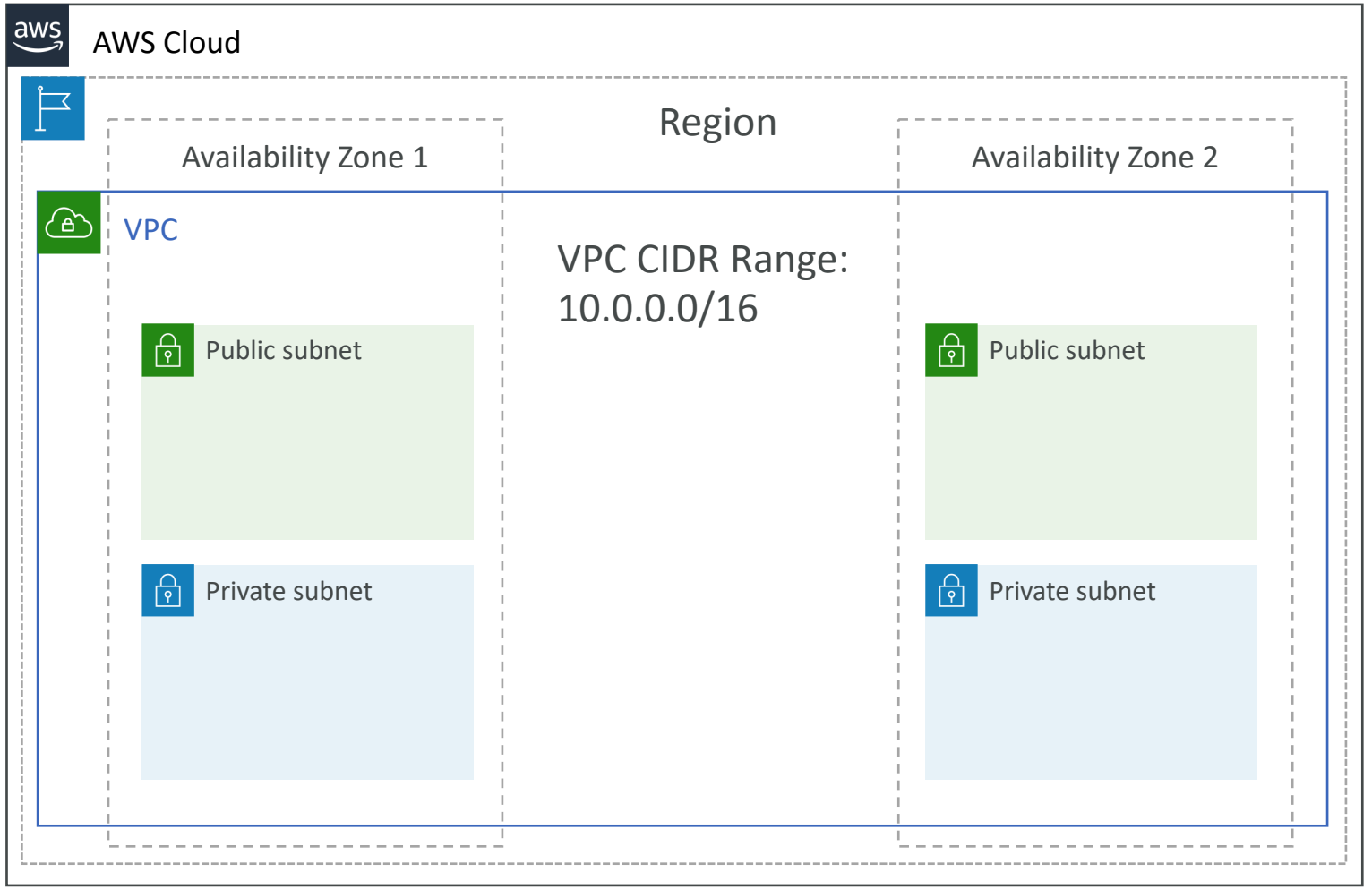
- IPv4 – Internet Protocol version 4 (4.3 Billion Addresses)
  - **Public IPv4** – can be used on the Internet
  - EC2 instance gets a new a public IP address every time you stop then start it **(default)**
  - **Private IPv4** – can be used on private networks (LAN) such as internal AWS networking (e.g., 192.168.1.1)
  - Private IPv4 is fixed for EC2 Instances even if you start/stop them
- Elastic IP – allows you to attach a fixed public IPv4 address to EC2 instance
  - Note: has ongoing cost if not attached to EC2 instance or if the EC2 instance is stopped
- IPv6 – Internet Protocol version 6 (3.4×10^{38} Addresses)
  - Every IP address is public (no private range)
  - Example: 2001:db8:3333:4444:cccc:dddd:eeee:ffff

VPC & Subnets Primer

- **VPC - Virtual Private Cloud:** private network to deploy your resources (regional resource)
- **Subnets** allow you to partition your network inside your VPC (Availability Zone resource)
- A **public subnet** is a subnet that is accessible from the internet
- A **private subnet** is a subnet that is not accessible from the internet
- To define access to the internet and between subnets, we use **Route Tables**.



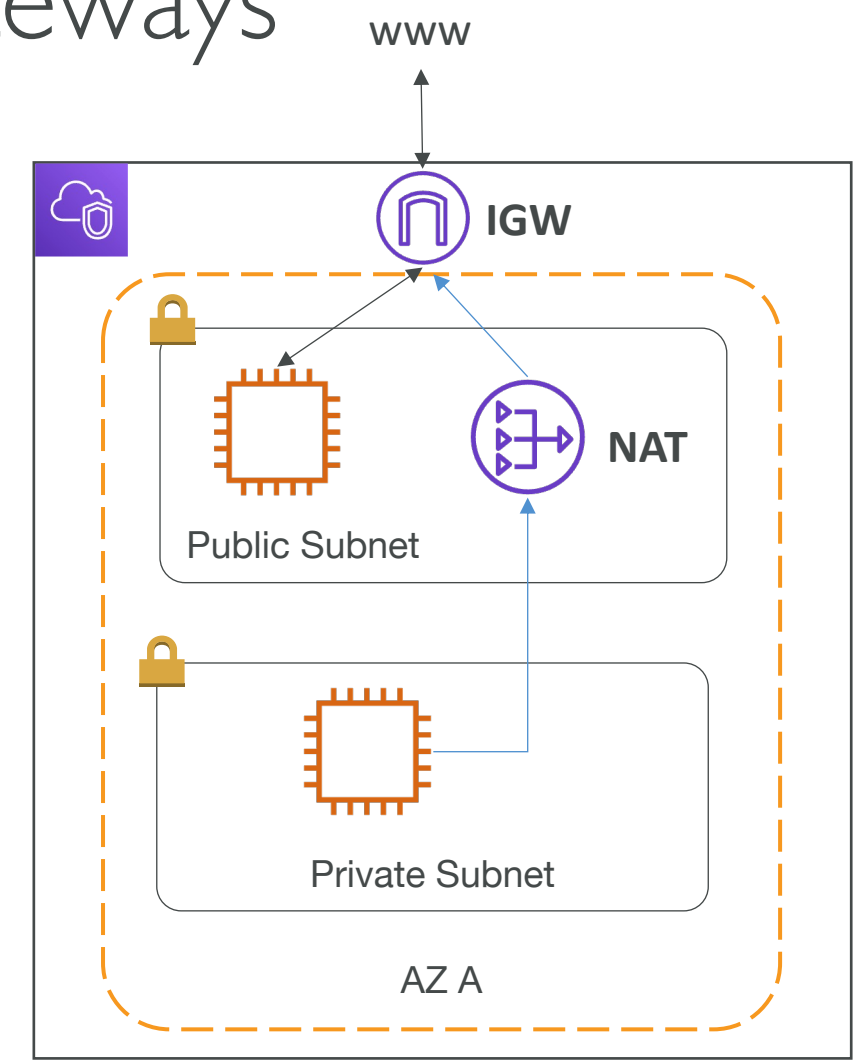
VPC Diagram



Internet Gateway & NAT Gateways

- **Internet Gateways** helps our VPC instances connect with the internet
- Public Subnets have a route to the internet gateway.

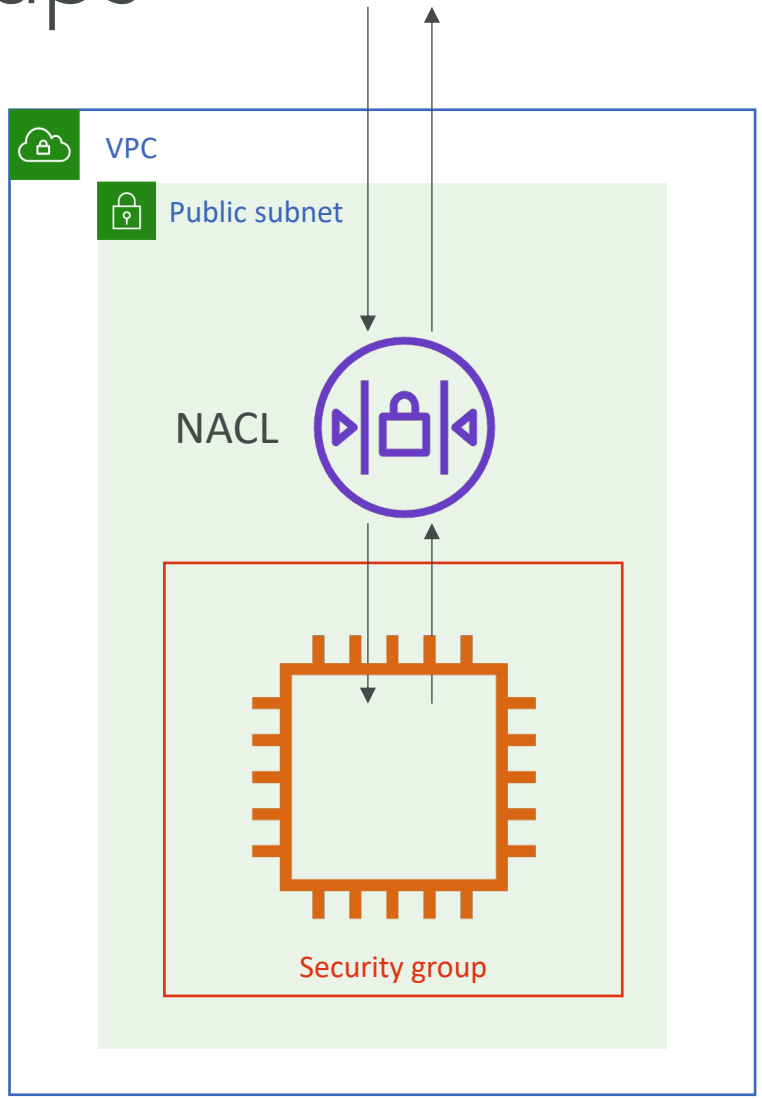
- **NAT Gateways** (AWS-managed) & **NAT Instances** (self-managed) allow your instances in your **Private Subnets** to access the internet while remaining private



Network ACL & Security Groups

- NACL (Network ACL)
  - A firewall which controls traffic from and to subnet
  - Can have ALLOW and DENY rules
  - Are attached at the **Subnet** level
  - Rules only include IP addresses

- Security Groups
  - A firewall that controls traffic to and from **an ENI / an EC2 Instance**
  - Can have only ALLOW rules
  - Rules include IP addresses and other security groups



Network ACLs vs Security Groups

Security Group	Network ACL
Operates at the instance level	Operates at the subnet level
Supports allow rules only	Supports allow rules and deny rules
Is stateful: Return traffic is automatically allowed, regardless of any rules	Is stateless: Return traffic must be explicitly allowed by rules
We evaluate all rules before deciding whether to allow traffic	We process rules in number order when deciding whether to allow traffic
Applies to an instance only if someone specifies the security group when launching the instance, or associates the security group with the instance later on	Automatically applies to all instances in the subnets it's associated with (therefore, you don't have to rely on users to specify the security group)

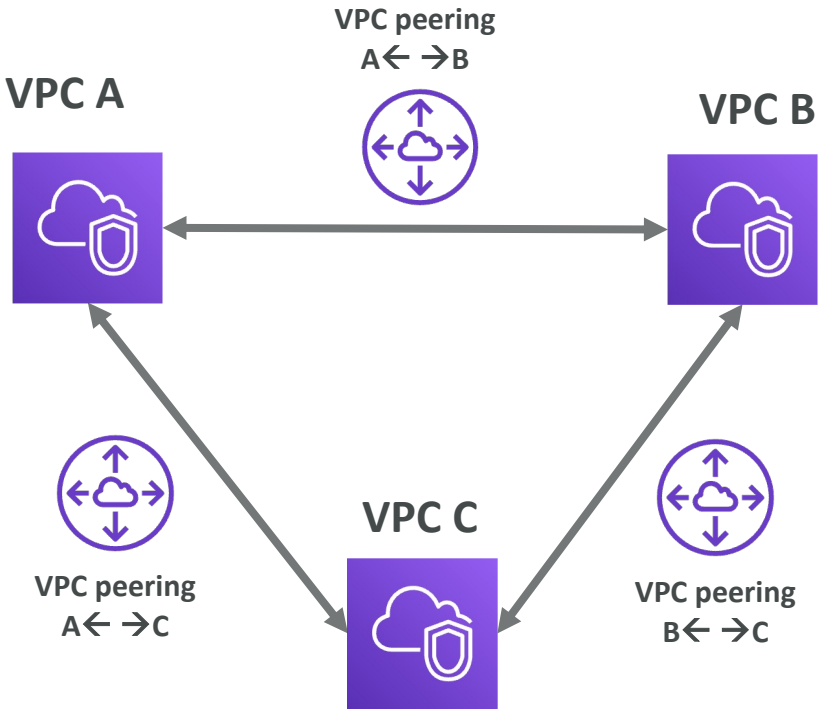
https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Security.html#VPC_Security_Comparison

VPC Flow Logs

- Capture information about IP traffic going into your interfaces:
  - **VPC** Flow Logs
  - **Subnet** Flow Logs
  - **Elastic Network Interface** Flow Logs
- Helps to monitor & troubleshoot connectivity issues. Example:
  - Subnets to internet
  - Subnets to subnets
  - Internet to subnets
- Captures network information from AWS managed interfaces too: Elastic Load Balancers, ElastiCache, RDS, Aurora, etc…
- VPC Flow logs data can go to S3, CloudWatch Logs, and Kinesis Data Firehose

VPC Peering

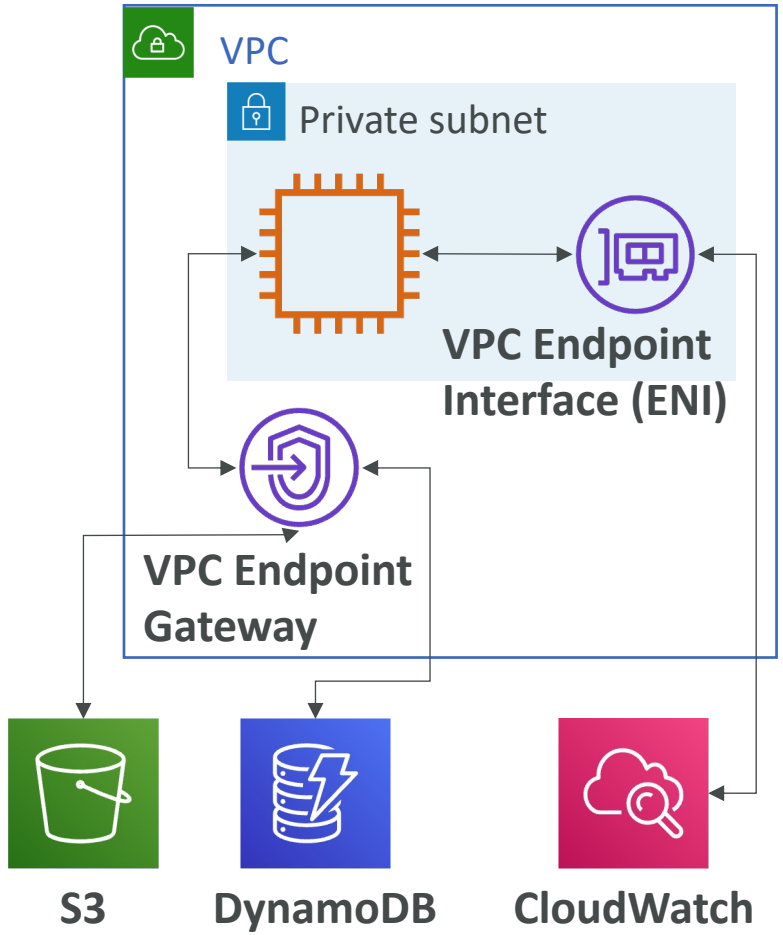
- Connect two VPC, privately using AWS' network
- Make them behave as if they were in the same network
- Must not have overlapping CIDR (IP address range)
- VPC Peering connection is **not transitive** (must be established for each VPC that need to communicate with one another)



VPC Endpoints

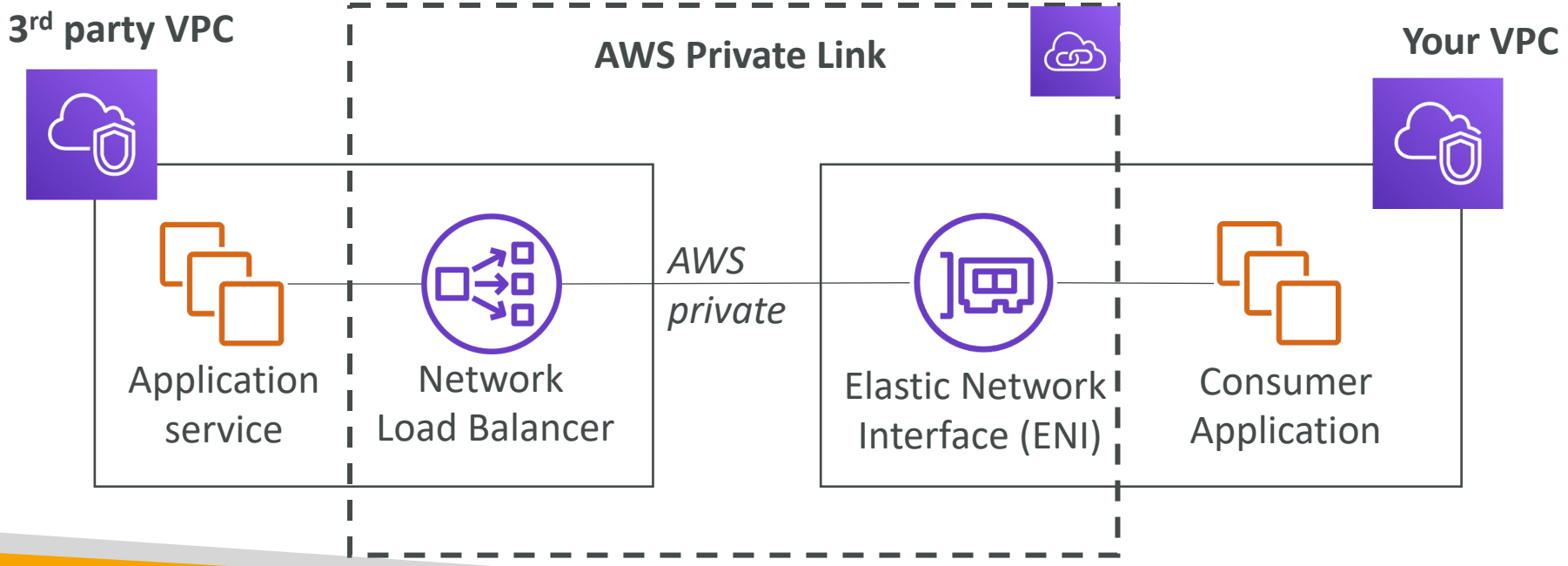
- Endpoints allow you to connect to AWS Services **using a private network** instead of the public www network
- This gives you enhanced security and lower latency to access AWS services

- VPC Endpoint **Gateway:** S3 & DynamoDB
- VPC Endpoint **Interface:** the rest



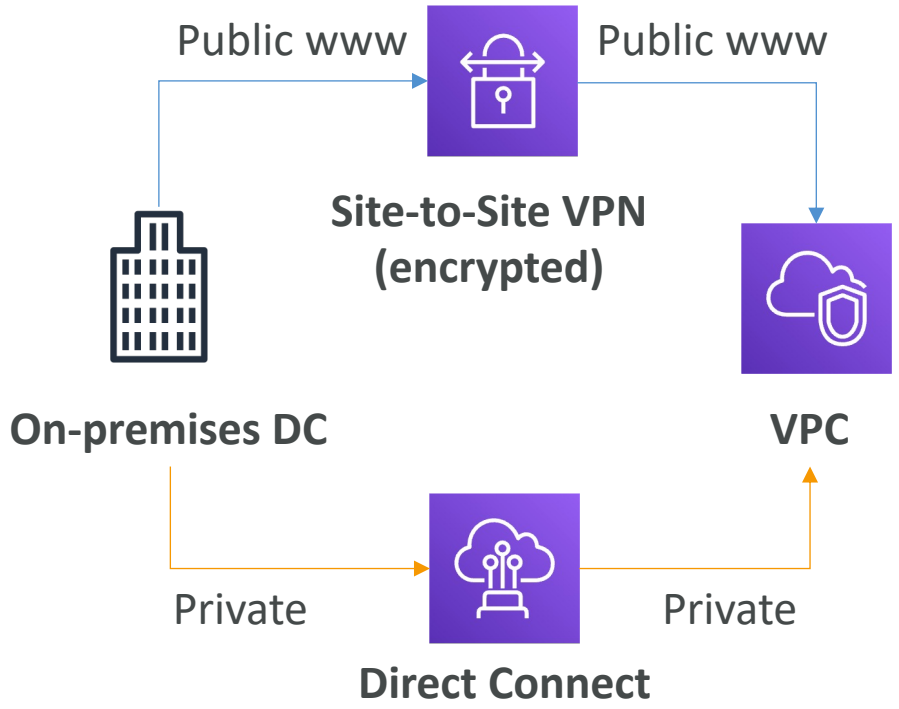
AWS PrivateLink (VPC Endpoint Services)

- Most secure & scalable way to expose a service to 1000s of VPCs
- Does not require VPC peering, internet gateway, NAT, route tables…
- Requires a network load balancer (Service VPC) and ENI (Customer VPC)



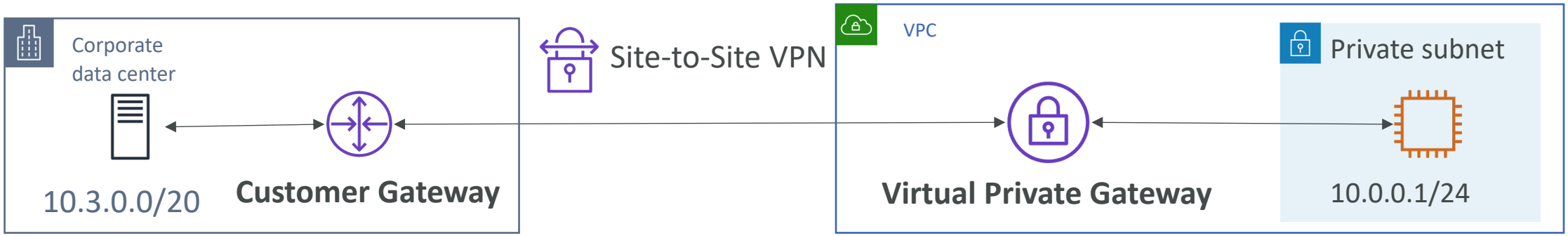
Site to Site VPN & Direct Connect

- Site to Site VPN
  - Connect an on-premises VPN to AWS
  - The connection is automatically encrypted
  - Goes over the public internet
- Direct Connect (DX)
  - Establish a physical connection between on-premises and AWS
  - The connection is private, secure and fast
  - Goes over a private network
  - Takes at least a month to establish



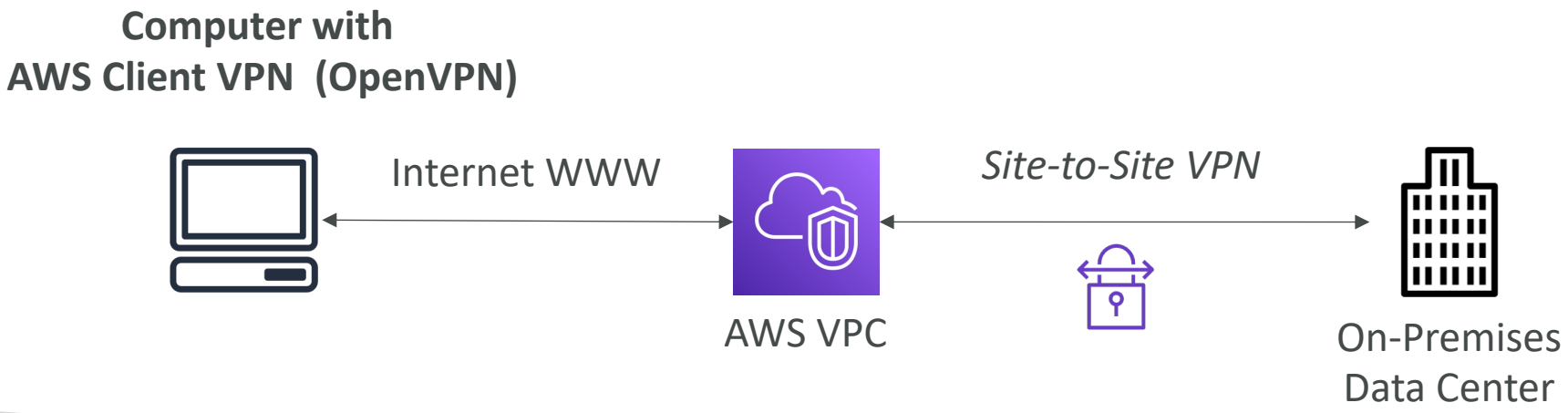
Site-to-Site VPN

- On-premises: must use a **Customer Gateway** (CGW)
- AWS: must use a **Virtual Private Gateway** (VGW)

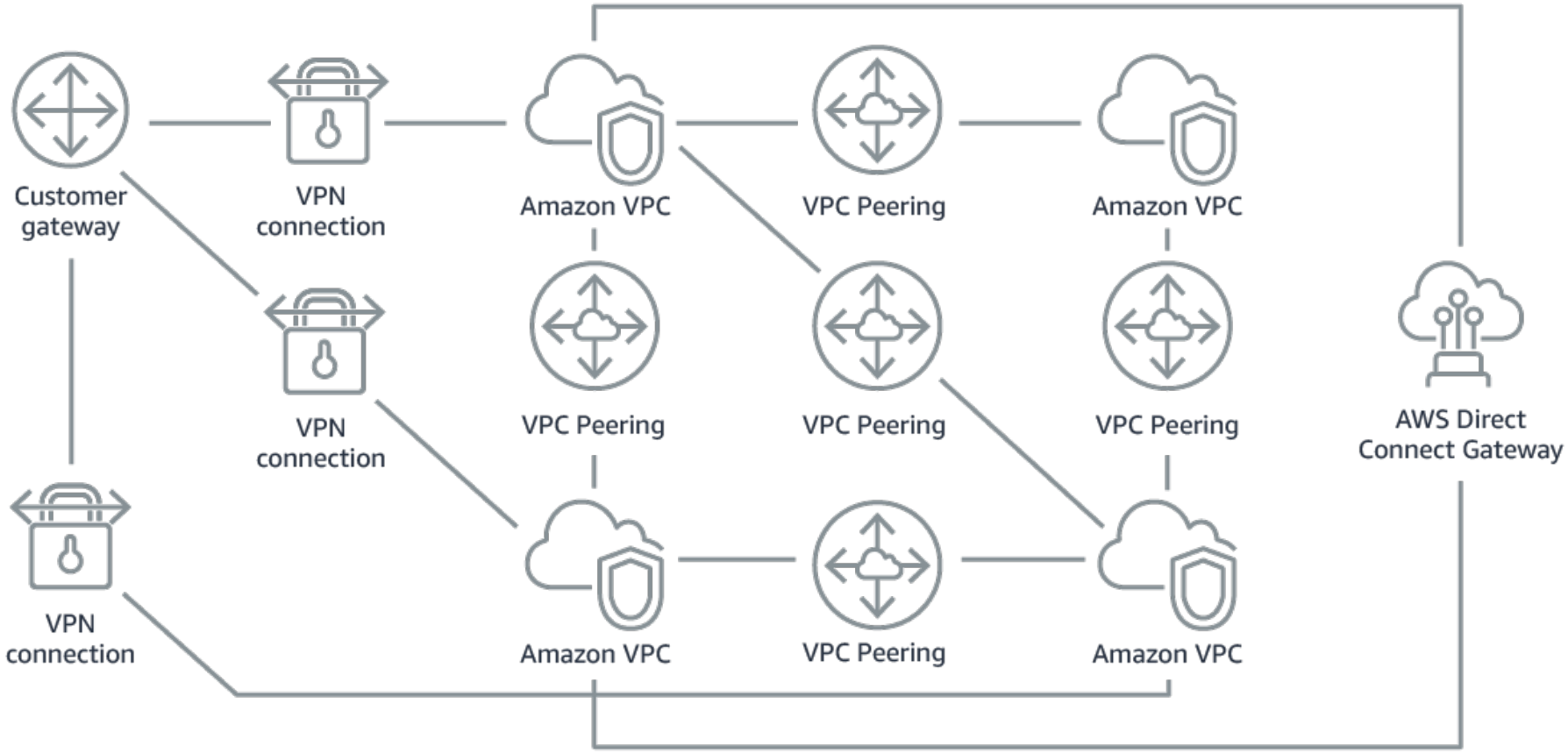


AWS Client VPN

- Connect from your computer using OpenVPN to your private network in AWS and on-premises
- Allow you to connect to your EC2 instances over a private IP (just as if you were in the private VPC network)
- Goes over public Internet

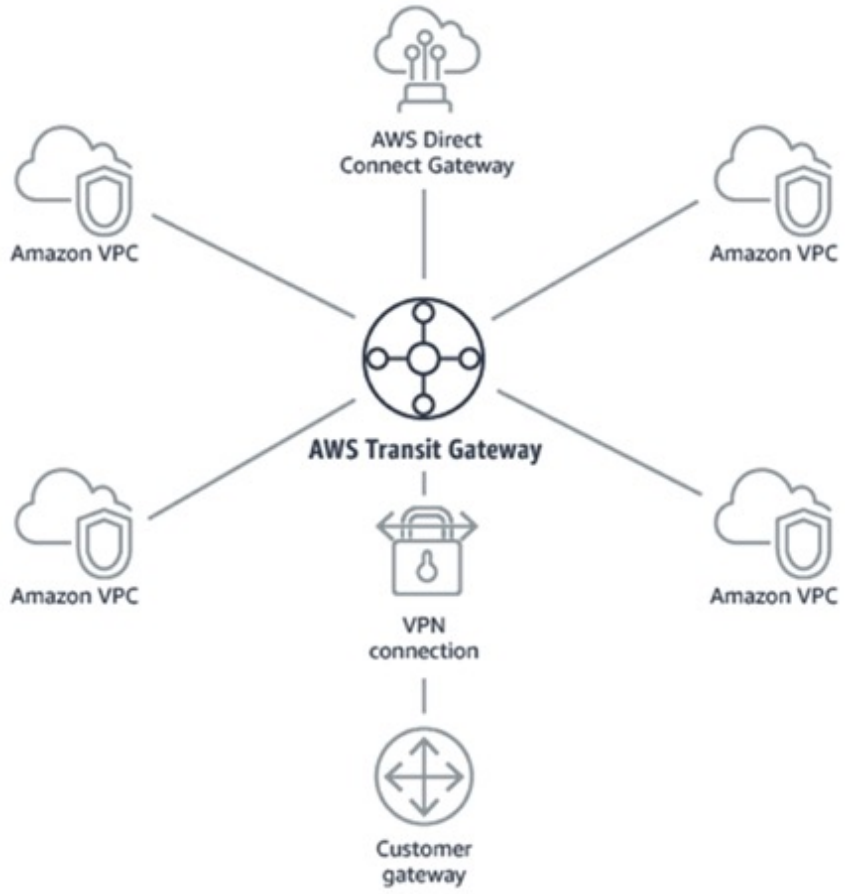


Network topologies can become complicated



Transit Gateway

- For having transitive peering between thousands of VPC and on-premises, hub-and-spoke (star) connection
- One single Gateway to provide this functionality
- Works with Direct Connect Gateway, VPN connections



VPC Closing Comments

- **VPC** – Virtual Private Cloud
- **Subnets** – Tied to an AZ, network partition of the VPC
- **Internet Gateway** – at the VPC level, provide Internet Access
- **NAT Gateway / Instances** – give internet access to private subnets
- **NACL** – Stateless, subnet rules for inbound and outbound
- **Security Groups** – Stateful, operate at the EC2 instance level or ENI
- **VPC Peering** – Connect two VPC with non overlapping IP ranges, nontransitive
- **Elastic IP** –fixed public IPv4, ongoing cost if not in-use

VPC Closing Comments

- **VPC Endpoints** – Provide private access to AWS Services within VPC
- **PrivateLink** – Privately connect to a service in a 3rd party VPC
- **VPC Flow Logs** – network traffic logs
- **Site to Site VPN** – VPN over public internet between on-premises DC and AWS
- **Client VPN** – OpenVPN connection from your computer into your VPC
- **Direct Connect** – direct private connection to AWS
- **Transit Gateway** – Connect thousands of VPC and on-premises networks together

Security & Compliance Section

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

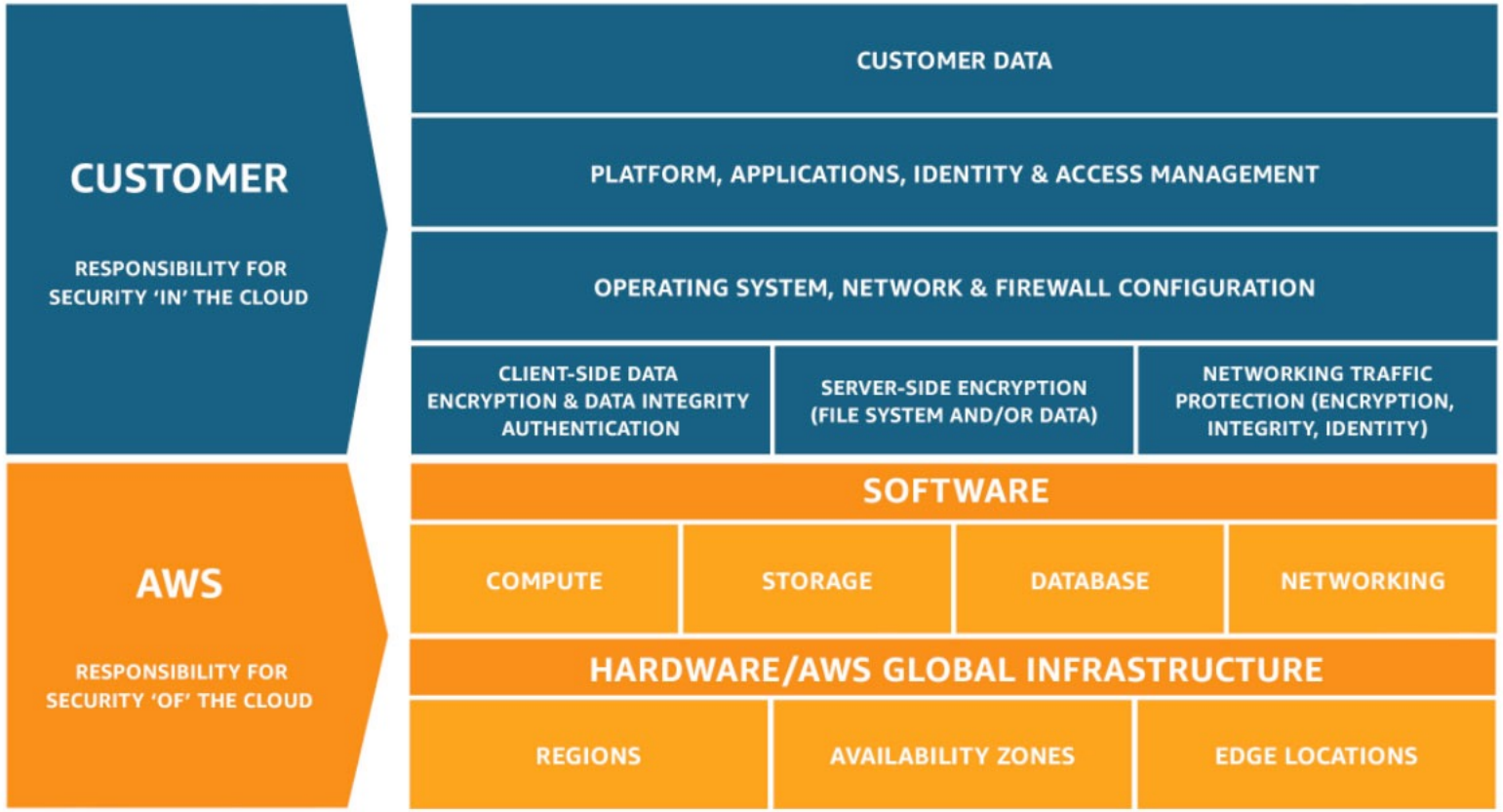
Example, for RDS

- 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

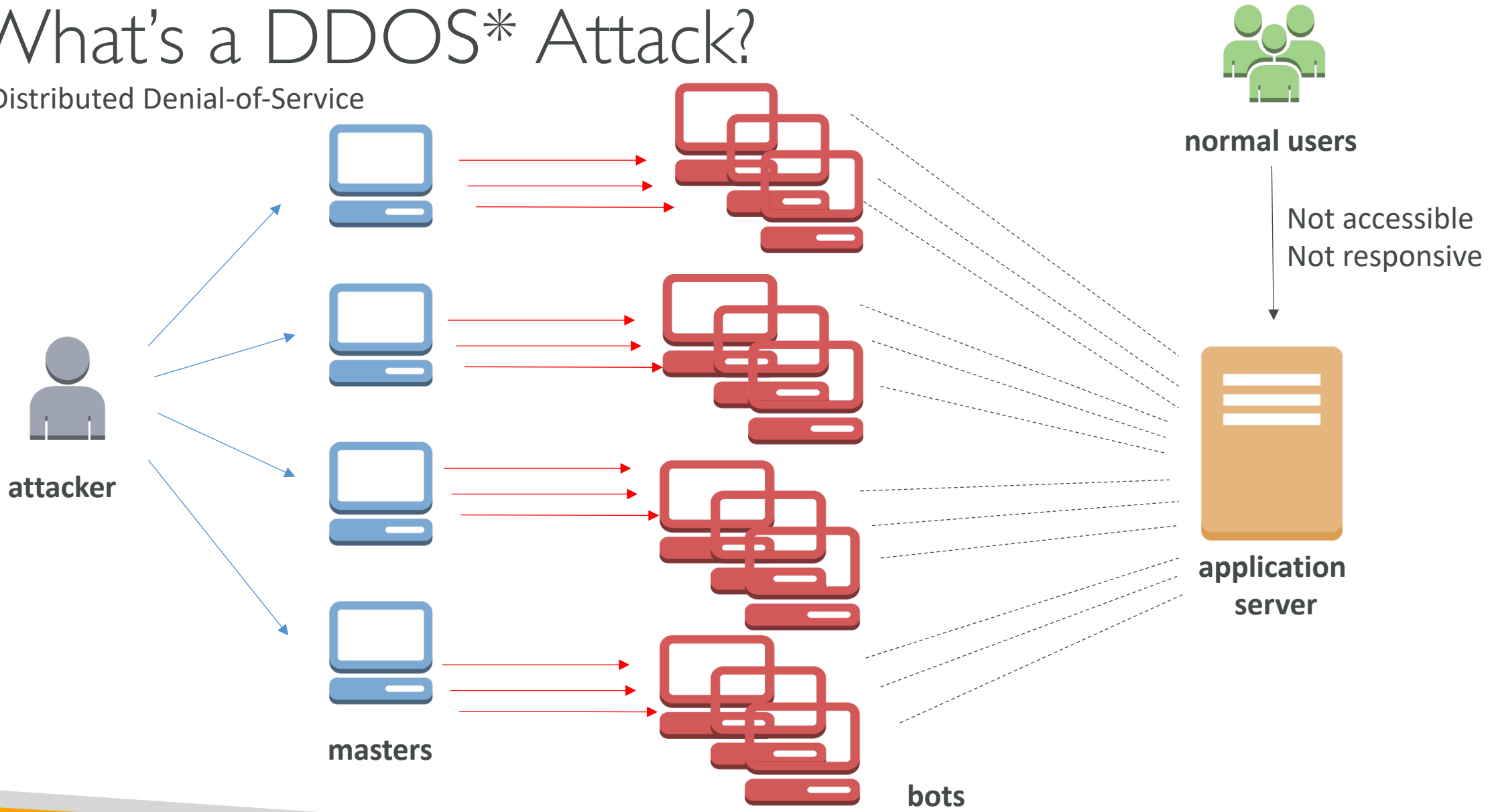
Shared Responsibility Model diagram



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

What's a DDOS* Attack?

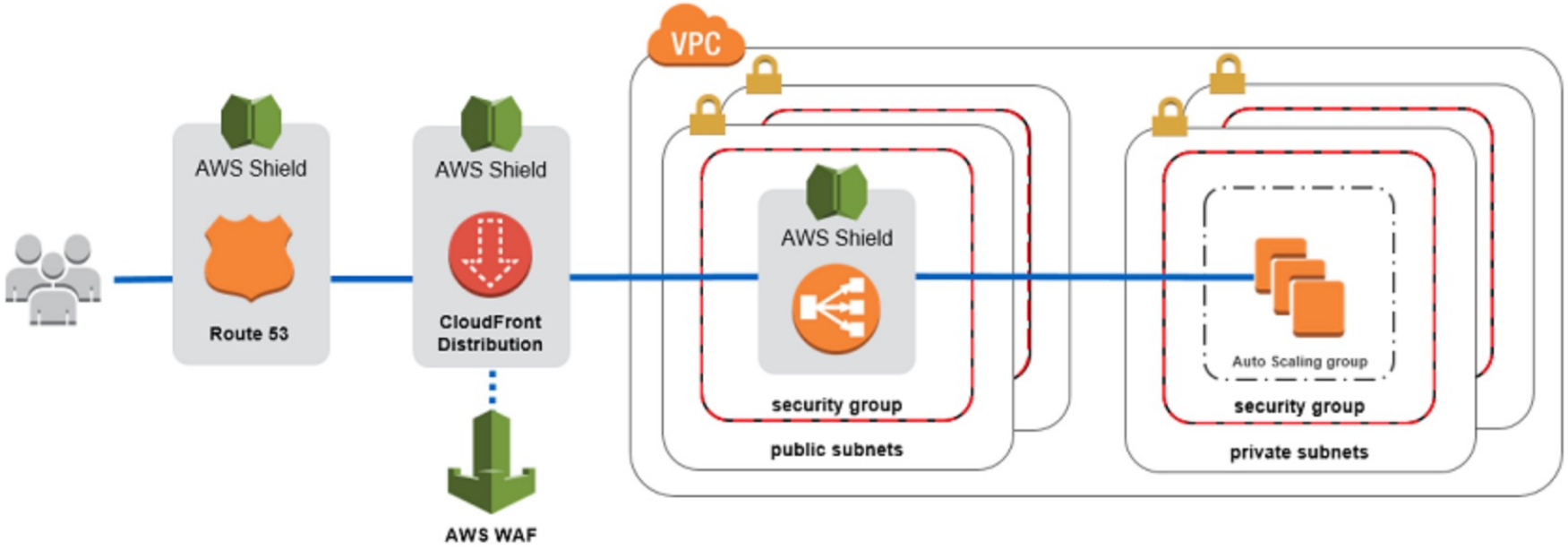
*Distributed Denial-of-Service



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 for DDoS Protection



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

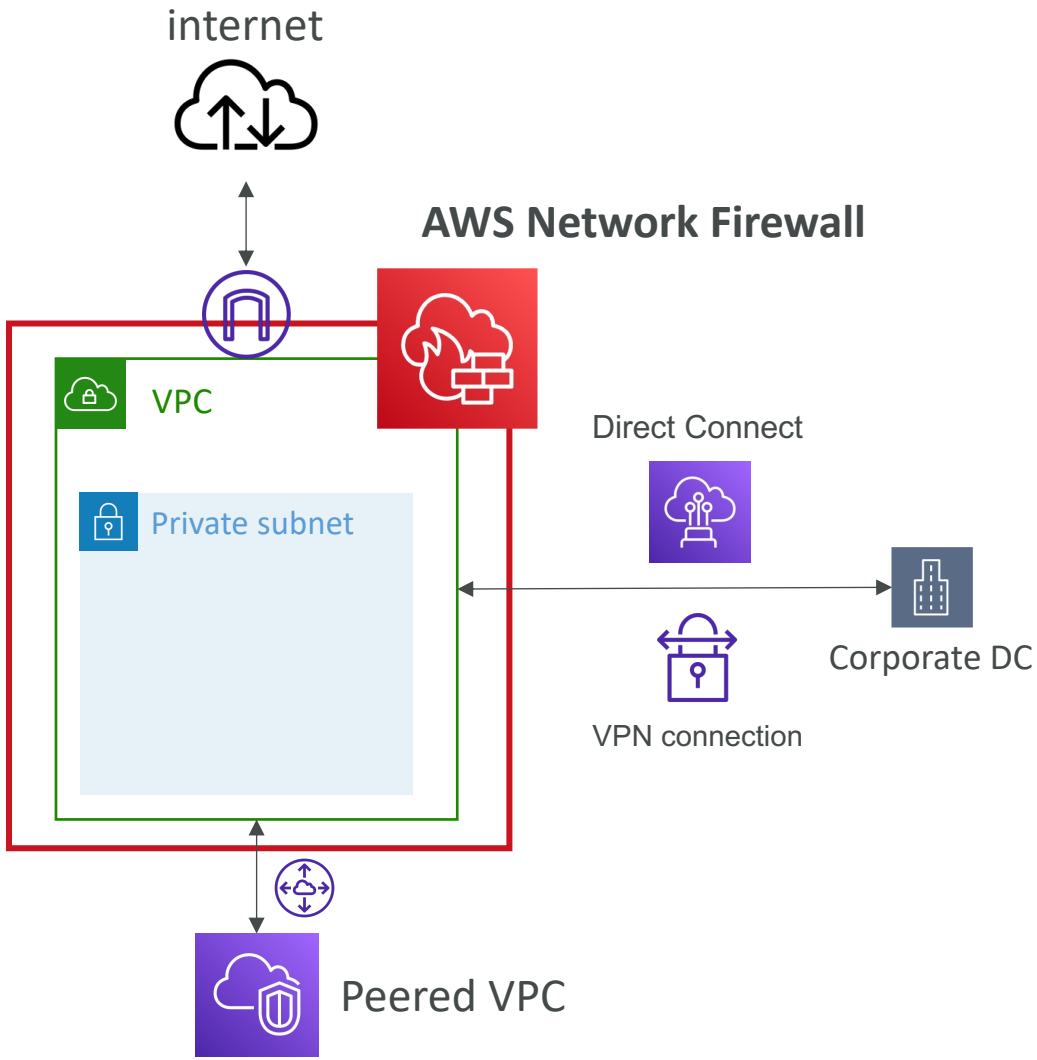
AWS WAF – Web Application Firewall

- 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**

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



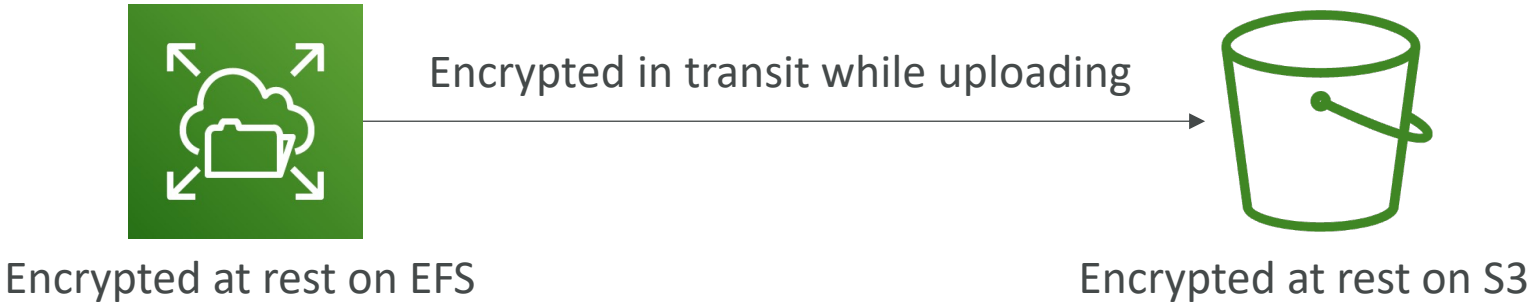
Penetration Testing on AWS Cloud

- 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
- Read more: https://aws.amazon.com/security/penetration-testing/

Data at rest vs. Data in transit



- **At rest:** data stored or archived on a device
  - On a hard disk, on a RDS instance, in S3 Glacier Deep Archive, etc.
- **In transit (in motion):** data being moved from one location to another
  - Transfer from on-premises to AWS, EC2 to DynamoDB, etc.
  - **Means data transferred on the network**
- We want to encrypt data in both states to protect it!
- For this we leverage **encryption keys**

AWS KMS (Key Management Service)

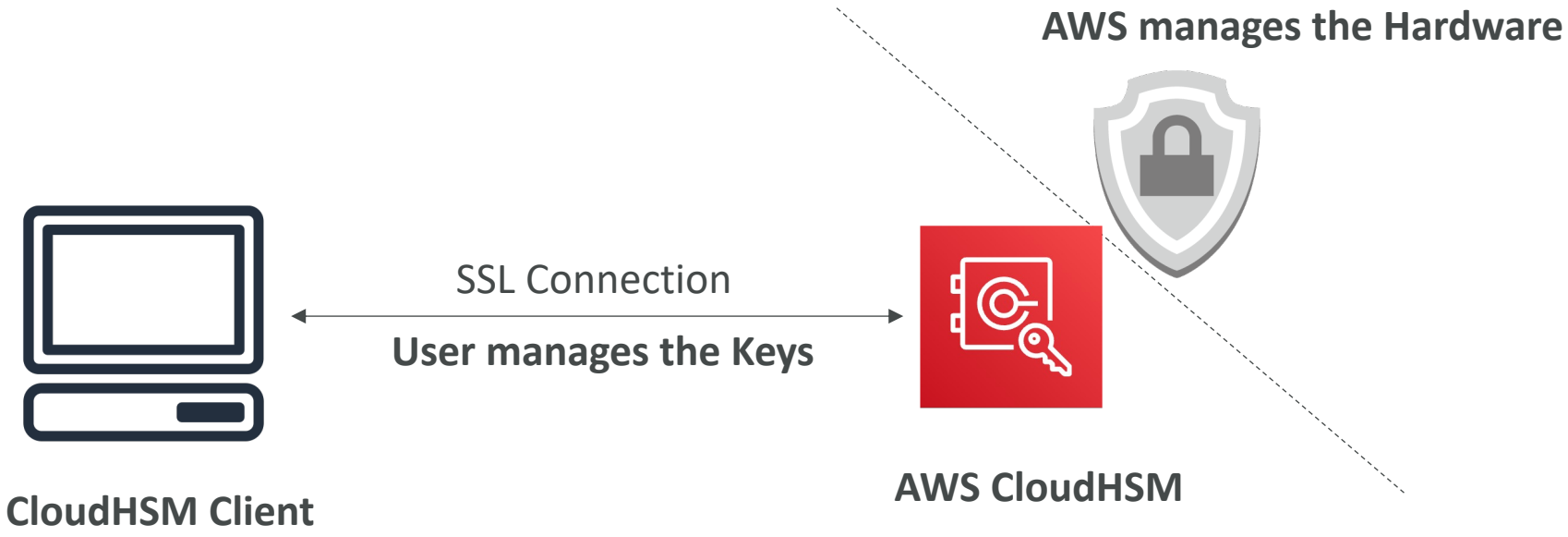
- Anytime you hear "encryption" for an AWS service, it's most likely KMS
- KMS = **AWS manages the encryption keys for us**
- **Encryption Opt-in:**
  - EBS volumes: encrypt volumes
  - S3 buckets: Server-side encryption of objects
  - Redshift database: encryption of data
  - RDS database: encryption of data
  - EFS drives: encryption of data
- **Encryption Automatically enabled:**
  - CloudTrail Logs
  - S3 Glacier
  - Storage Gateway

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

Sample HSM device

CloudHSM Diagram

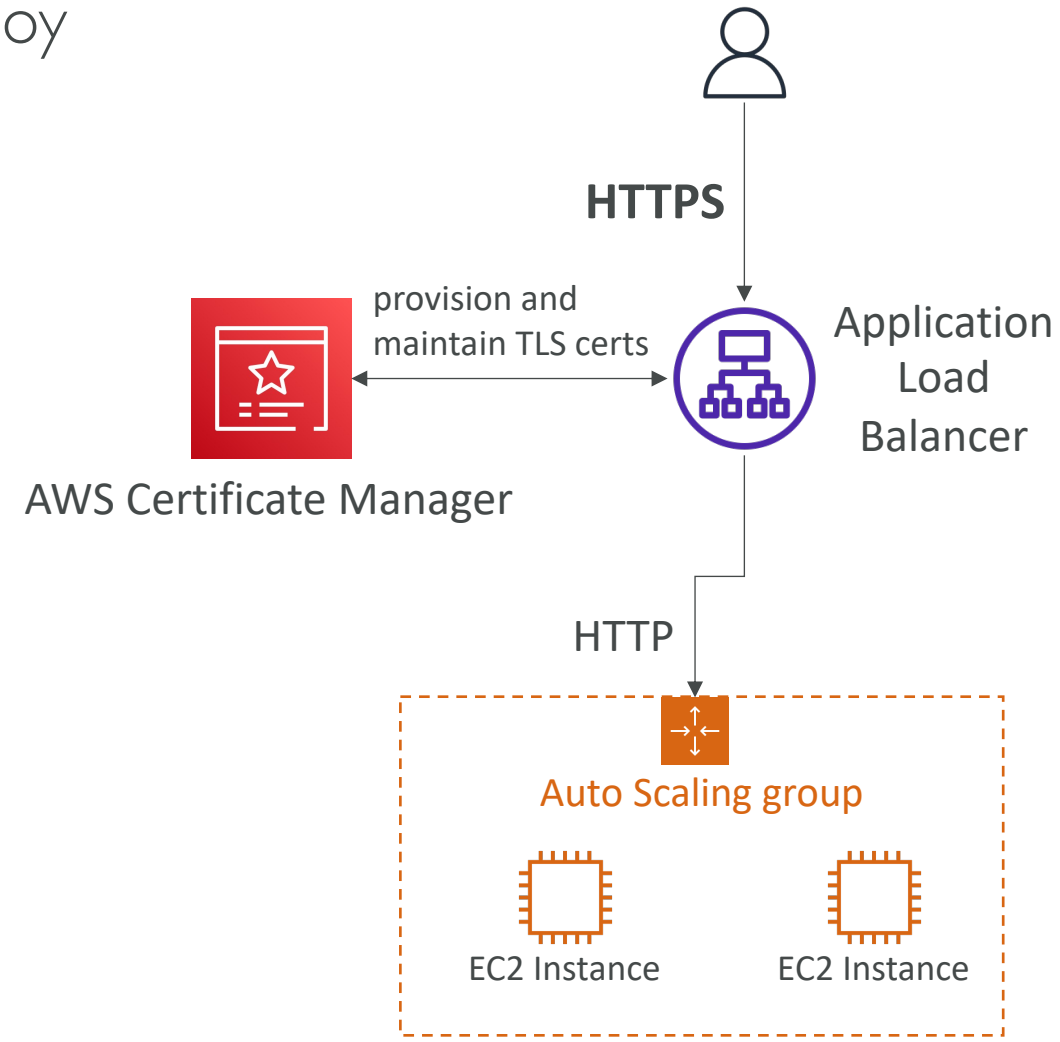


Types of Customer Master Keys: CMK

- Customer Managed CMK:
  - Create, manage and used by the customer, can enable or disable
  - Possibility of rotation policy (new key generated every year, old key preserved)
  - Possibility to bring-your-own-key
- AWS managed CMK:
  - Created, managed and used on the customer's behalf by AWS
  - Used by AWS services (aws/s3, aws/ebs, aws/redshift)
- AWS owned CMK:
  - Collection of CMKs that an AWS service owns and manages to use in multiple accounts
  - AWS can use those to protect resources in your account (but you can't view the keys)
- CloudHSM Keys (custom keystore):
  - Keys generated from your own CloudHSM hardware device
  - Cryptographic operations are performed within the CloudHSM cluster

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



AWS Secrets Manager

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

- Mostly meant for RDS integration

AWS Artifact (not really a service)

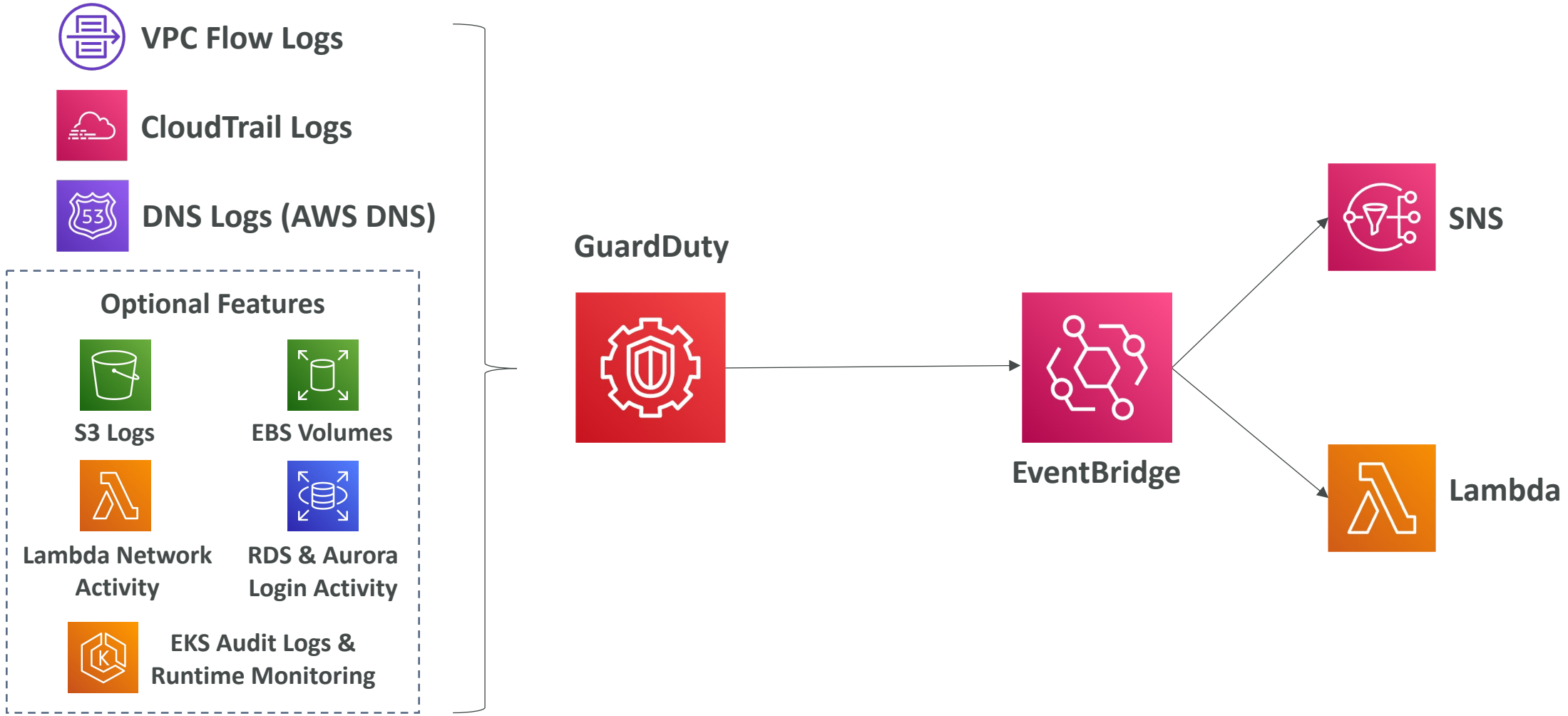
- Portal that provides customers with on-demand access to AWS compliance documentation and AWS agreements
- Artifact Reports - Allows you to download AWS security and compliance documents from third-party auditors, like AWS ISO certifications, Payment Card Industry (PCI), and System and Organization 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

Amazon GuardDuty

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

Amazon GuardDuty

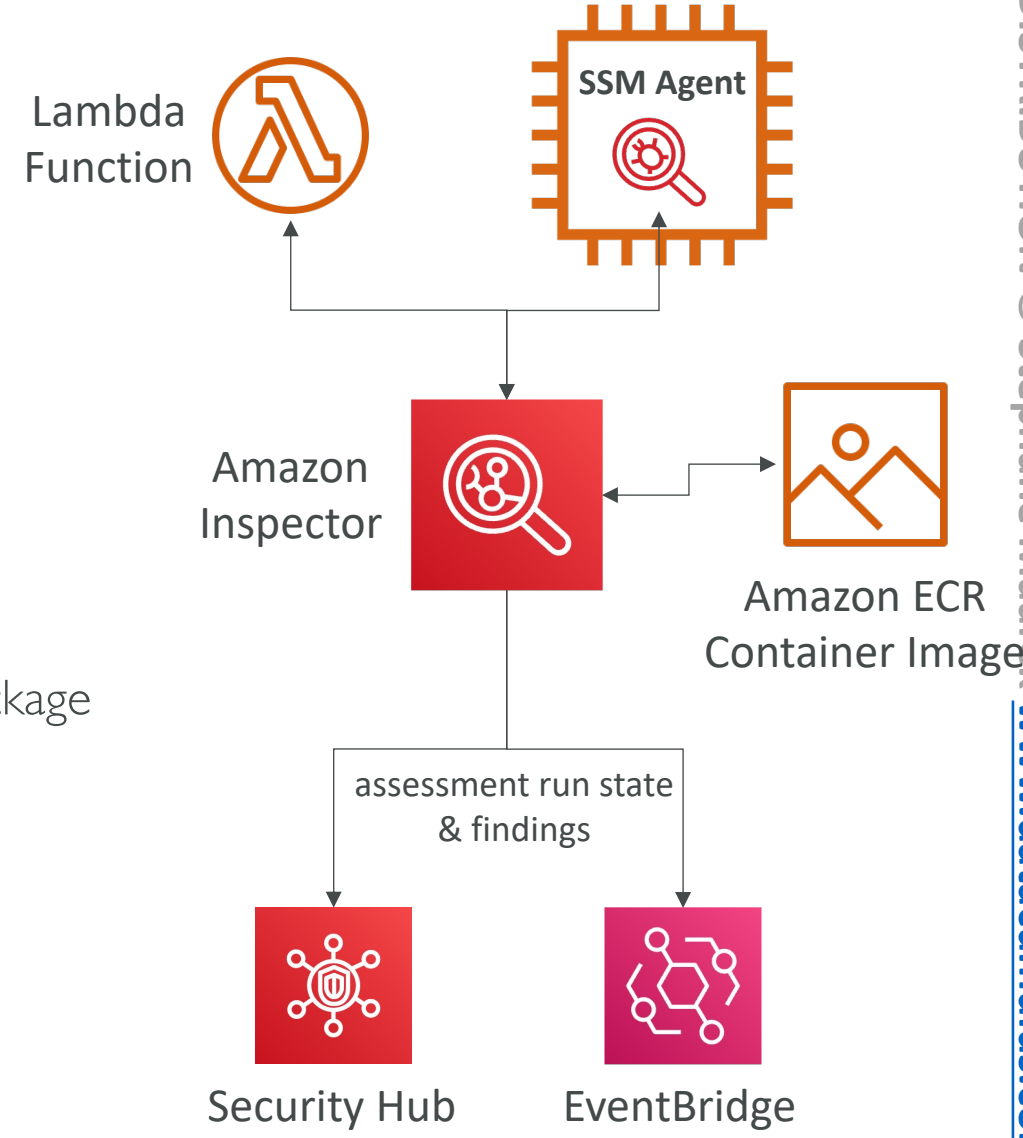


Amazon Inspector

- Automated Security Assessments

- For EC2 instances
  - Leveraging the AWS System Manager (SSM) agent
  - Analyze against unintended network accessibility
  - Analyze the running OS against known vulnerabilities
- For Container Images push to Amazon ECR
  - Assessment of Container Images as they are pushed
- For Lambda Functions
  - Identifies software vulnerabilities in function code and package dependencies
  - Assessment of functions as they are deployed

- Reporting & integration with AWS Security Hub
- Send findings to Amazon Event Bridge



What does Amazon Inspector evaluate?

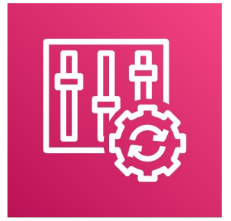
- Remember: only for EC2 instances, Container Images & Lambda functions

- Continuous scanning of the infrastructure, only when needed

- Package vulnerabilities (EC2, ECR & Lambda) – database of CVE
- Network reachability (EC2)

- A risk score is associated with all vulnerabilities for prioritization

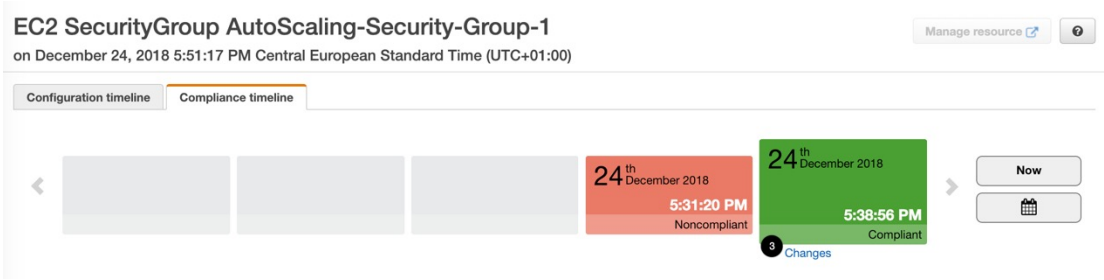
AWS Config



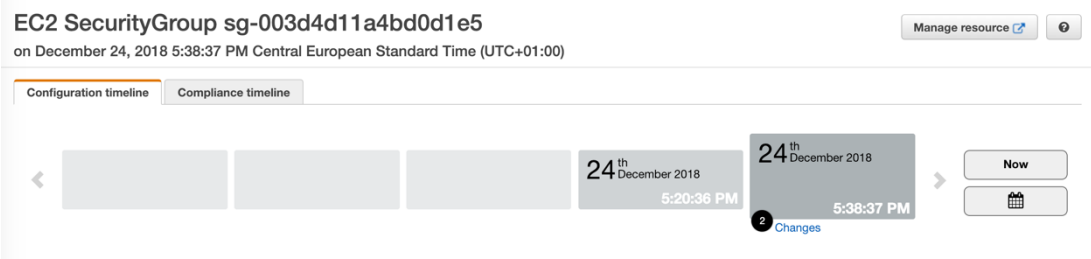
- Helps with **auditing and recording compliance of your AWS resources**
- Helps **record configurations and changes over time**
- Possibility of storing the configuration data into S3 (analyzed by Athena)
- Questions that can be solved by AWS Config:
  - Is there unrestricted SSH access to my security 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

AWS Config Resource

- View compliance of a resource over time



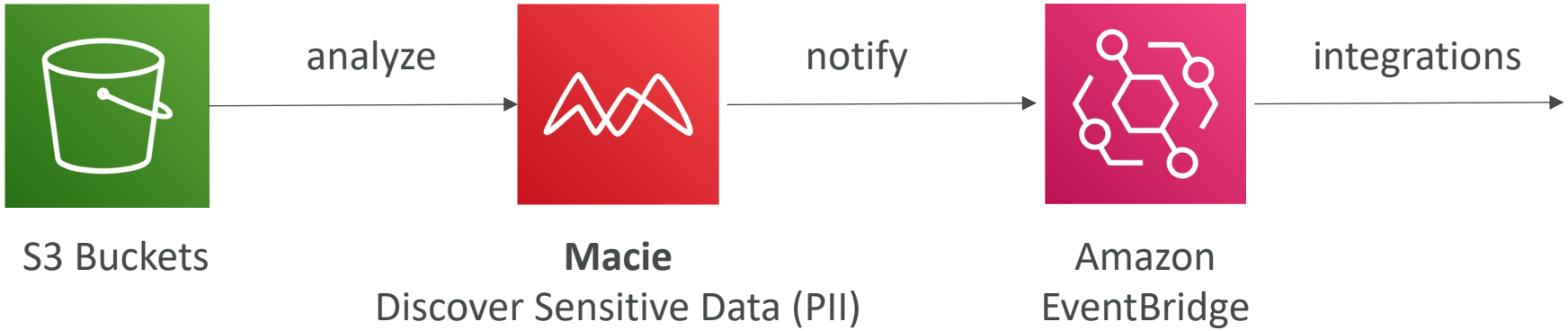
- View configuration of a resource over time



- View CloudTrail API calls if enabled

AWS Macie

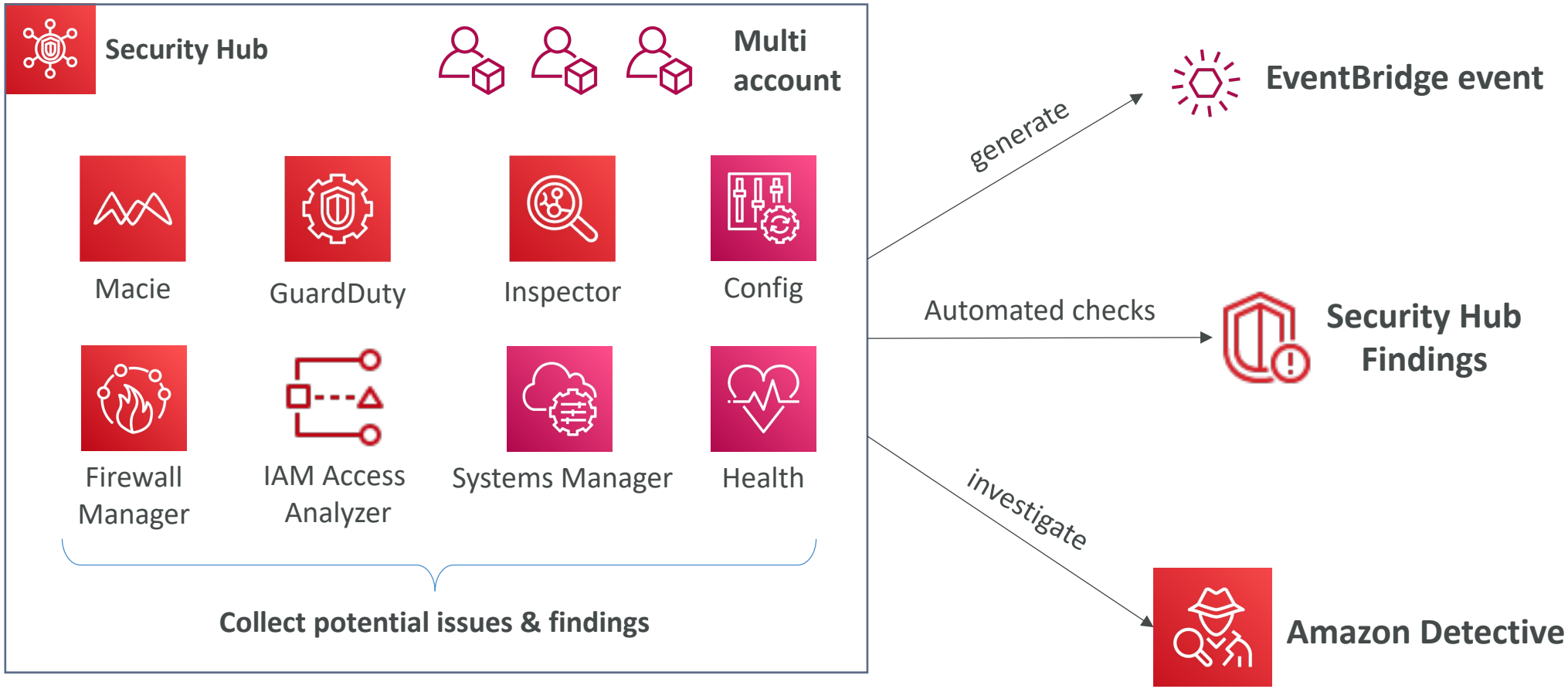
- Amazon Macie is a fully managed data security and data privacy service that uses **machine learning and pattern matching to discover and protect your sensitive data in AWS.**
- Macie helps identify and alert you to **sensitive data, such as personally identifiable information (PII)**



AWS Security Hub

- **Central security tool** to manage security **across several AWS accounts** and **automate security checks**
- Integrated dashboards showing current security and compliance status to quickly take actions
- Automatically aggregates alerts in predefined or personal findings formats from various AWS services & AWS partner tools:
  - Config
  - GuardDuty
  - Inspector
  - Macie
  - IAM Access Analyzer
  - AWS Systems Manager
  - AWS Firewall Manager
  - AWS Health
  - AWS Partner Network Solutions
- Must first enable the AWS Config Service

AWS Security Hub



Amazon Detective

- GuardDuty, Macie, and Security Hub are used to identify potential security issues, or findings
- Sometimes security findings require deeper analysis to isolate the root cause and take action – it's a complex process
- Amazon Detective **analyzes, investigates, and quickly identifies the root cause of security issues or suspicious activities (using ML and graphs)**
- **Automatically collects and processes events** from VPC Flow Logs, CloudTrail, GuardDuty and create a unified view
- Produces visualizations with details and context to get to the root cause

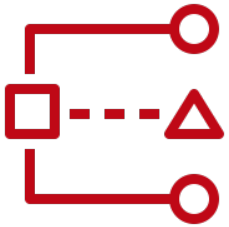
AWS Abuse

- Report suspected AWS resources used for abusive or illegal purposes
- Abusive & prohibited behaviors are:
  - **Spam** – receving undesired emails from AWS-owned IP address, websites & forums spammed by AWS resources
  - **Port scanning** – sending packets to your ports to discover the unsecured ones
  - **DoS or DDoS attacks** – AWS-owned IP addresses attempting to overwhlem or crash your servers/softwares
  - **Intrusion attempts** – logging in on your resources
  - **Hosting objectionable or copyrighted content** – distributing illegal or copyrighted content without consent
  - **Distributing malware** – AWS resources distributing softwares to harm computers or machines
- Contact the AWS Abuse team: AWS abuse form, or abuse@amazonaws.com

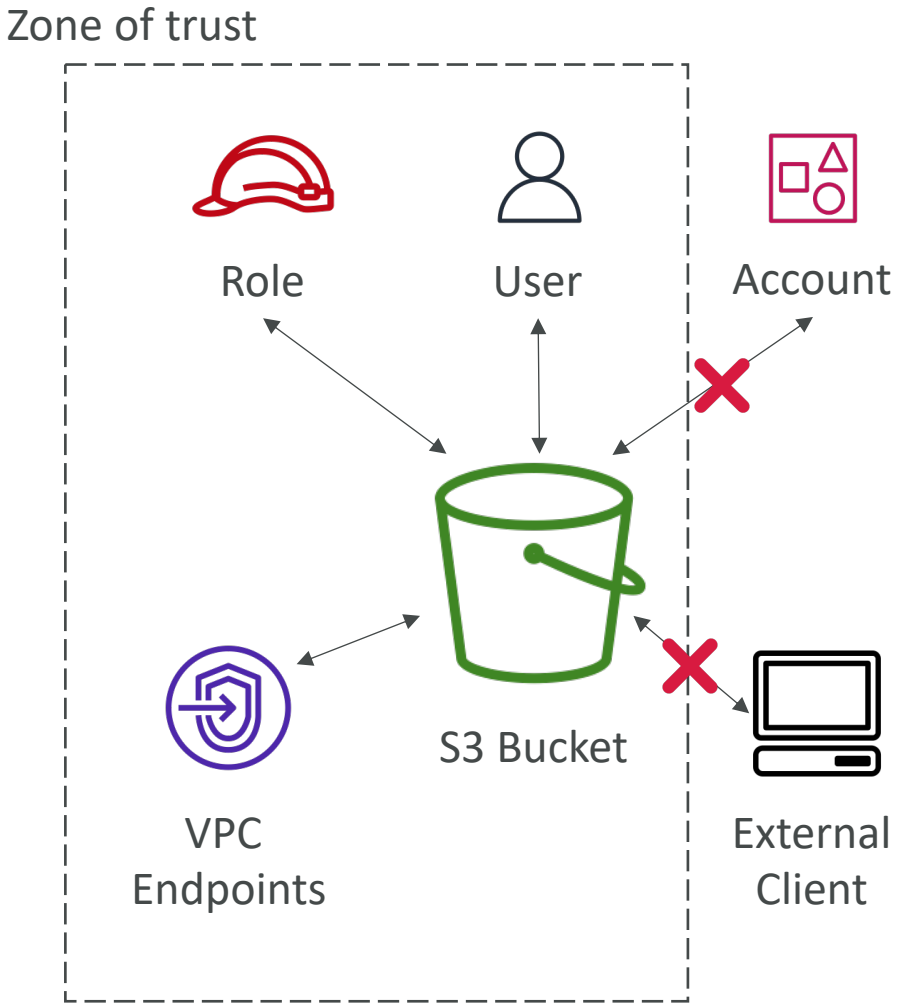
Root user privileges

- Root user = Account Owner (created when the account is created)
- Has complete access to all AWS services and resources
- **Lock away your AWS account root user access keys!**
- Do not use the root account for everyday tasks, even administrative tasks
- Actions that can be performed only by the root user:
  - **Change account settings** (account name, email address, root user password, root user access keys)
  - View certain tax invoices
  - **Close your AWS account**
  - Restore IAM user permissions
  - **Change or cancel your AWS Support plan**
  - **Register as a seller in the Reserved Instance Marketplace**
  - Configure an Amazon S3 bucket to enable MFA
  - Edit or delete an Amazon S3 bucket policy that includes an invalid VPC ID or VPC endpoint ID
  - Sign up for GovCloud

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



Section Summary: Security & Compliance

- **Shared Responsibility on AWS**
- **Shield:** Automatic DDoS Protection + 24/7 support for advanced
- **WAF:** Firewall to filter incoming requests based on rules
- **KMS:** Encryption keys managed by AWS
- **CloudHSM:** Hardware encryption, we manage encryption keys
- **AWS Certificate Manager:** provision, manage, and deploy SSL/TLS Certificates
- **Artifact:** Get access to compliance reports such as PCI, ISO, etc…
- **GuardDuty:** Find malicious behavior with VPC, DNS & CloudTrail Logs
- **Inspector:** find software vulnerabilities in EC2, ECR Images, and Lambda functions
- **Network Firewall:** Protect VPC against network attacks

Section Summary: Security & Compliance

- **Config:** Track config changes and compliance against rules
- **Macie:** Find sensitive data (ex: PII data) in Amazon S3 buckets
- **CloudTrail:** Track API calls made by users within account
- **AWS Security Hub:** gather security findings from multiple AWS accounts
- **Amazon Detective:** find the root cause of security issues or suspicious activities
- **AWS Abuse:** Report AWS resources used for abusive or illegal purposes
- **Root user privileges:**
  - Change account settings
  - Close your AWS account
  - Change or cancel your AWS Support plan
  - Register as a seller in the Reserved Instance Marketplace
- **IAM Access Analyzer:** identify which resources are shared externally

Machine Learning Section

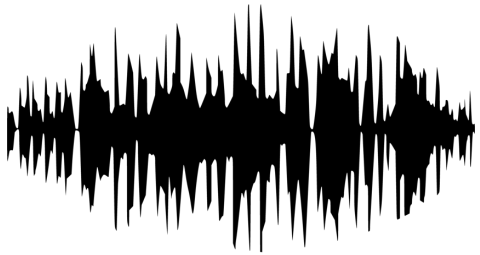
Amazon Rekognition

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

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

Amazon Transcribe

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

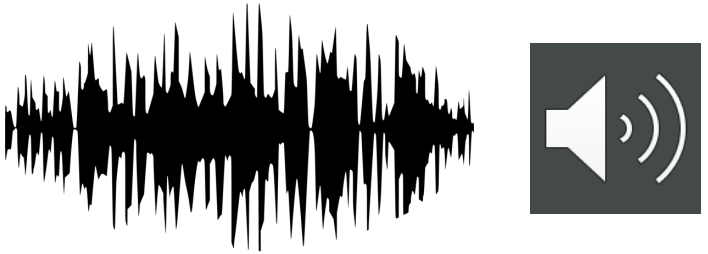


"Hello my name is Stéphane.
I hope you're enjoying the course!

Amazon Polly

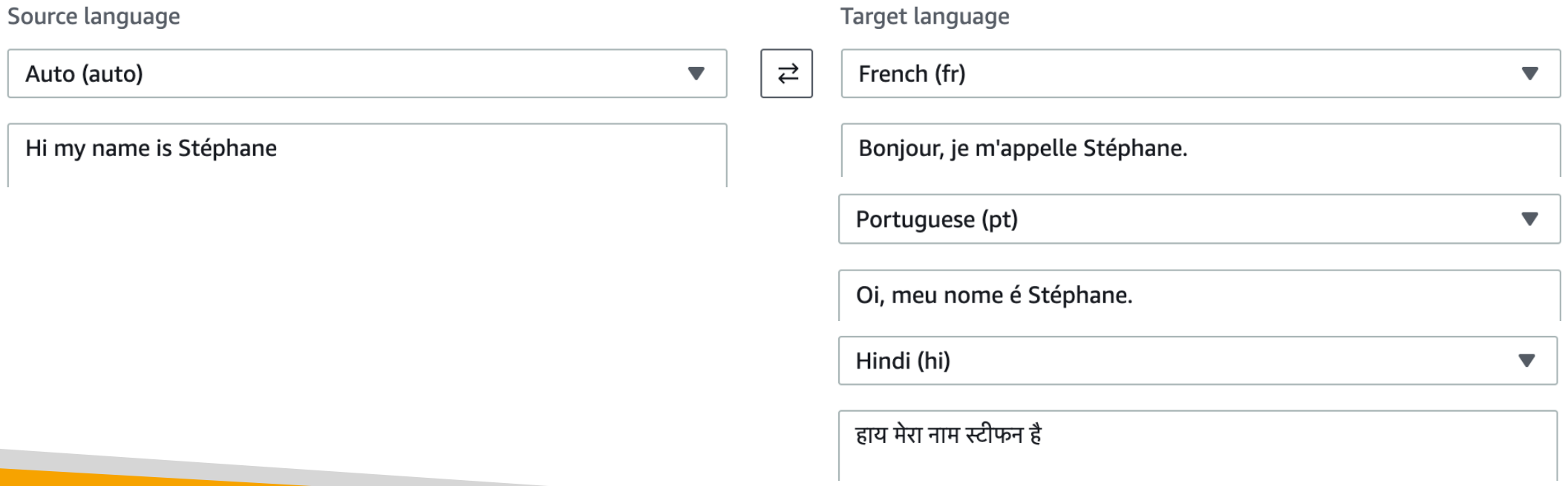
- Turn text into lifelike speech using deep learning
- Allowing you to create applications that talk

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



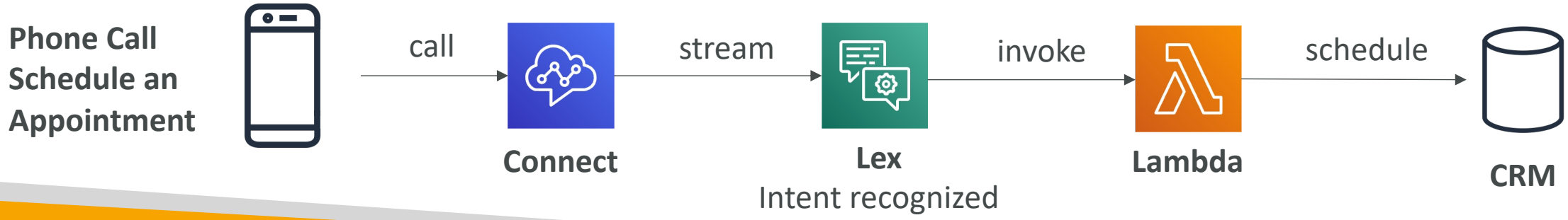
Amazon Translate

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



Amazon Lex & Connect

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

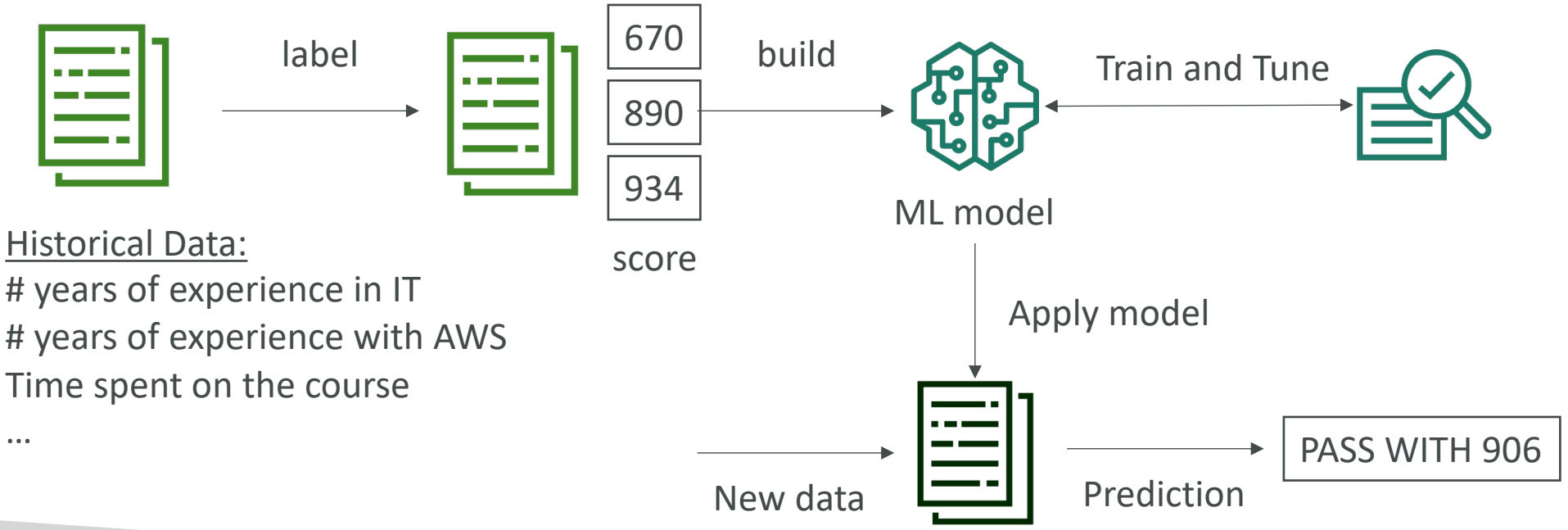


Amazon Comprehend

- For **Natural Language Processing – NLP**
- Fully managed and serverless service
- Uses machine learning to find insights and relationships in text
  - Language of the text
  - Extracts key phrases, places, people, brands, or events
  - Understands how positive or negative the text is
  - Analyzes text using tokenization and parts of speech
  - Automatically organizes a collection of text files by topic
- Sample use cases:
  - analyze customer interactions (emails) to find what leads to a positive or negative experience
  - Create and groups articles by topics that Comprehend will uncover

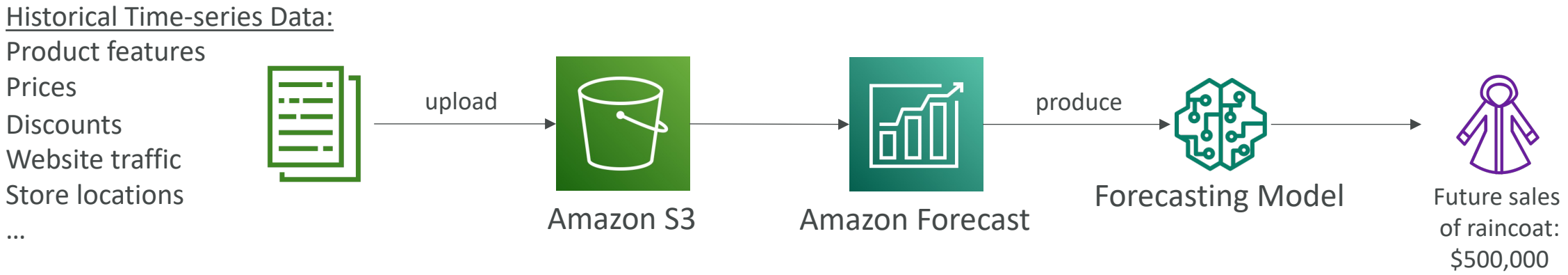
Amazon SageMaker

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



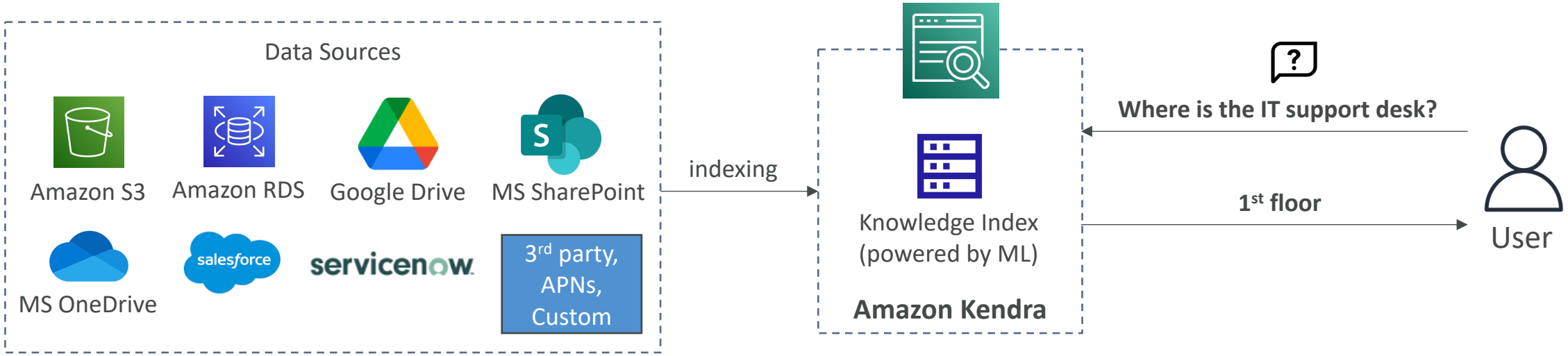
Amazon Forecast

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



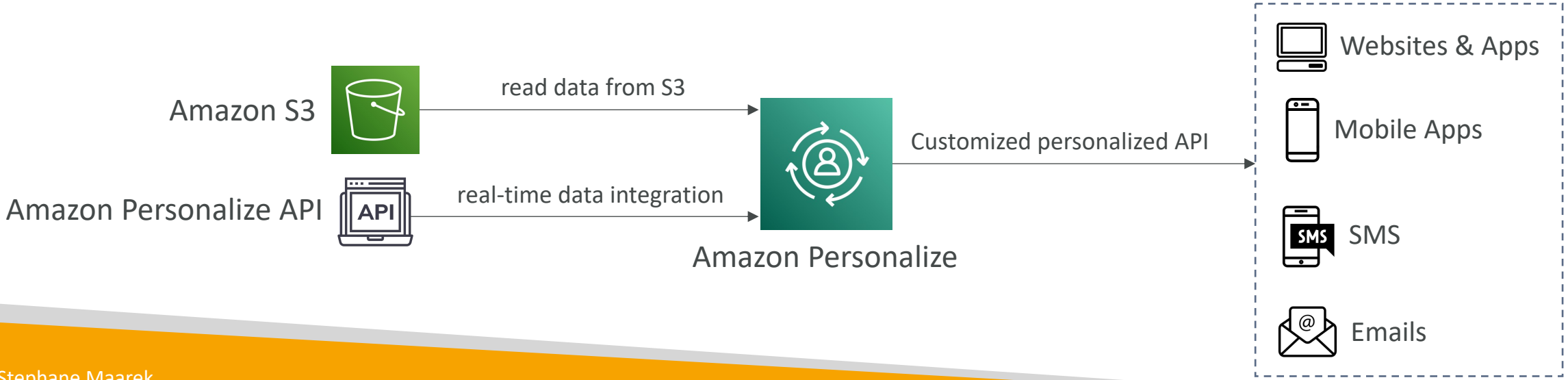
Amazon Kendra

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



Amazon Personalize

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



Amazon Textract

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

analyze

Amazon Textract

result

```
{
   "Document ID": "123456789-005",
   "Name": "",
   "SEX": "F",
   "DOB": "23.05.1997",
   ...
}
```

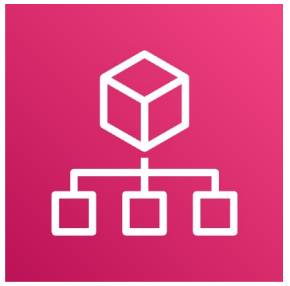
- Extract data from forms and tables
- Read and process any type of document (PDFs, images, …)
- Use cases:
  - Financial Services (e.g., invoices, financial reports)
  - Healthcare (e.g., medical records, insurance claims)
  - Public Sector (e.g., tax forms, ID documents, passports)

AWS Machine Learning - Summary

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

Account Management, Billing & Support Section

AWS Organizations



- Global service
- Allows to manage **multiple AWS accounts**
- The main account is the master account
- Cost Benefits:
  - **Consolidated Billing** across all accounts - single payment method
  - Pricing benefits from **aggregated usage** (volume discount for EC2, S3…)
  - **Pooling of Reserved EC2 instances** for optimal savings
- API is available to **automate AWS account creation**
- Restrict account privileges using Service Control Policies (SCP)

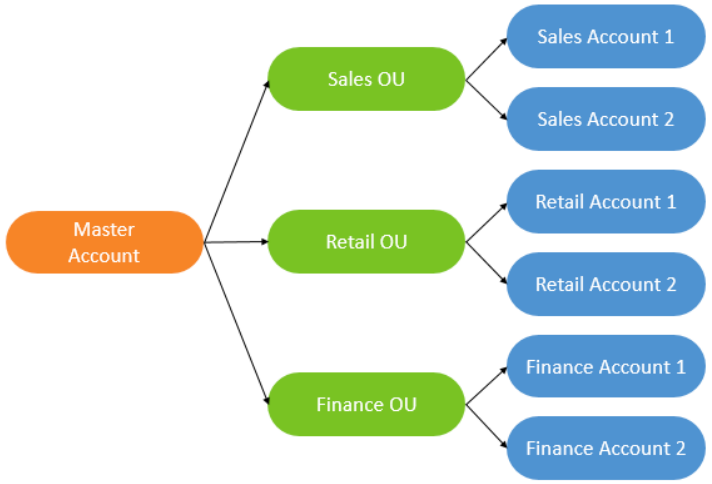
Multi Account Strategies

- Create accounts per **department**, per **cost center,** per **dev / test / prod**, based on **regulatory restrictions** (using SCP), for **better resource isolation** (ex: VPC), to have **separate per-account service limits**, isolated account for **logging**

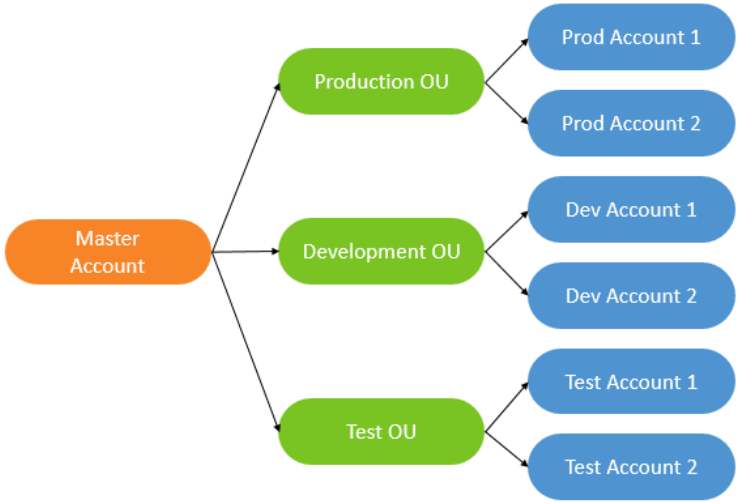
- Multi Account vs One Account Multi VPC
- Use tagging standards for billing purposes
- Enable CloudTrail on all accounts, send logs to central S3 account
- Send CloudWatch Logs to central logging account

Organizational Units (OU) - Examples

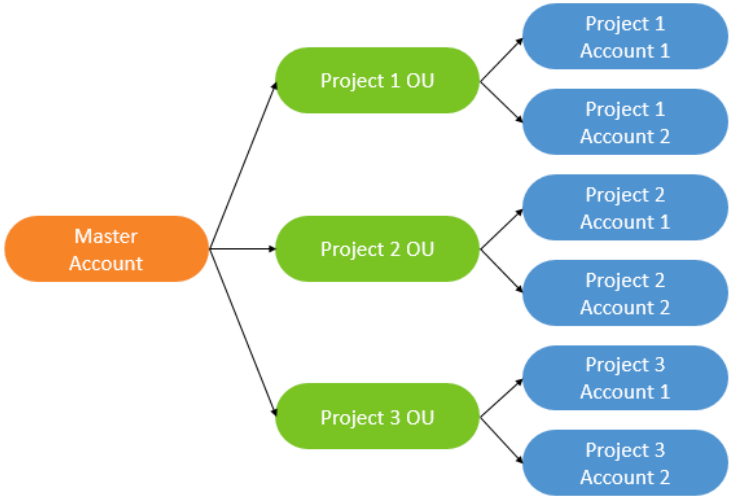
Business Unit



Environmental Lifecycle

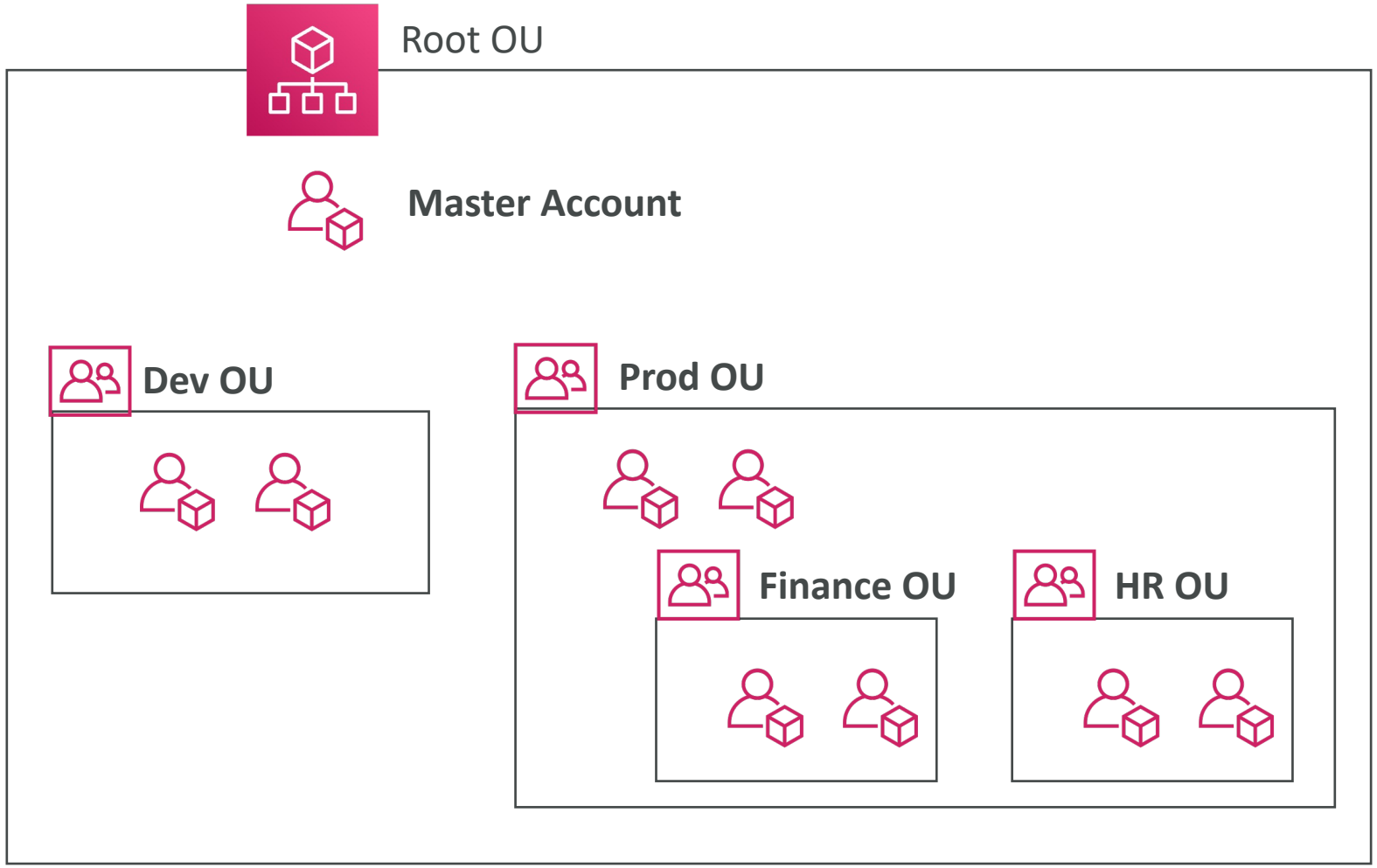


Project-based



https://aws.amazon.com/answers/account-management/aws-multi-account-billing-strategy/

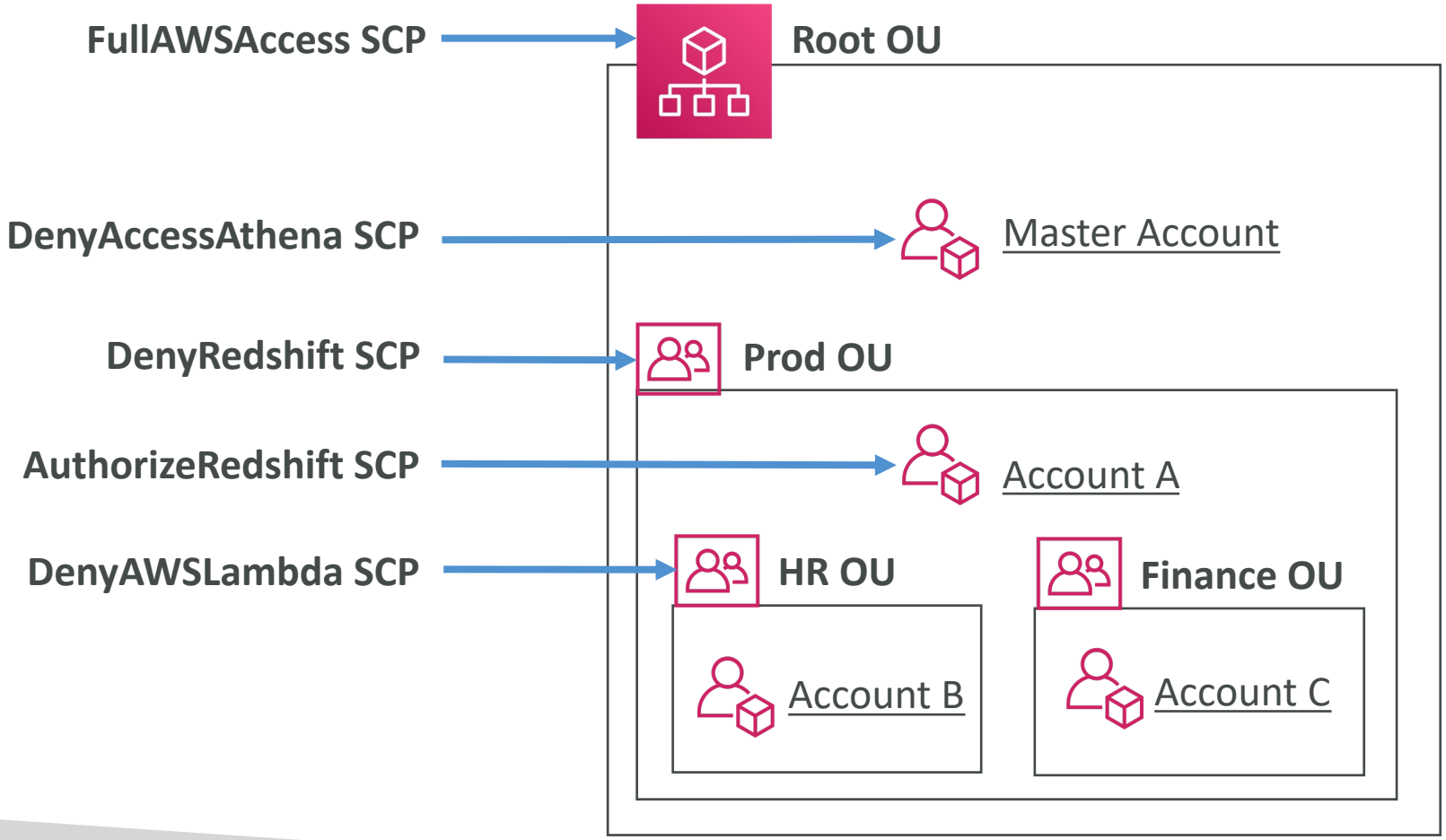
AWS Organization



Service Control Policies (SCP)

- Whitelist or blacklist IAM actions
- Applied at the **OU** or **Account** level
- Does not apply to the Master Account
- SCP is applied to all the **Users and Roles** of the Account, including Root user
- The SCP does not affect service-linked roles
  - Service-linked roles enable other AWS services to integrate with AWS Organizations and can't be restricted by SCPs.
- SCP must have an explicit Allow (does not allow anything by default)
- Use cases:
  - Restrict access to certain services (for example: can't use EMR)
  - Enforce PCI compliance by explicitly disabling services

SCP Hierarchy



- Master Account
  - Can do anything
  - (no SCP apply)
- Account A
  - Can do anything
  - EXCEPT access Redshift (explicit Deny from Prod 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
Blacklist and Whitelist strategies

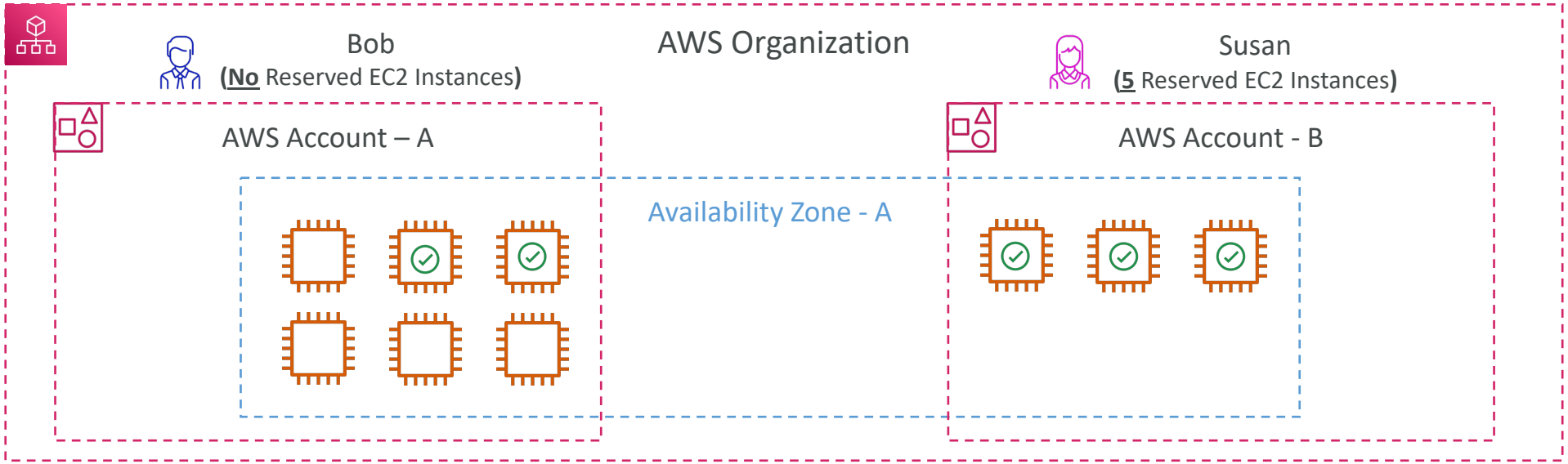
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAllActions",
            "Effect": "Allow",
            "Action": "*",
            "Resource": "*"
        },
        {
            "Sid": "DenyDynamoDB",
            "Effect": "Deny",
            "Action": "dynamodb:*",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

AWS Organization – Consolidated Billing

- When enabled, provides you with:
  - **Combined Usage** – combine the usage across all AWS accounts in the AWS Organization **to share the volume pricing, Reserved Instances and Savings Plans discounts**
  - **One Bill** – get one bill for all AWS Accounts in the AWS Organization
- The management account can turn off Reserved Instances discount sharing for any account in the AWS Organization, including itself

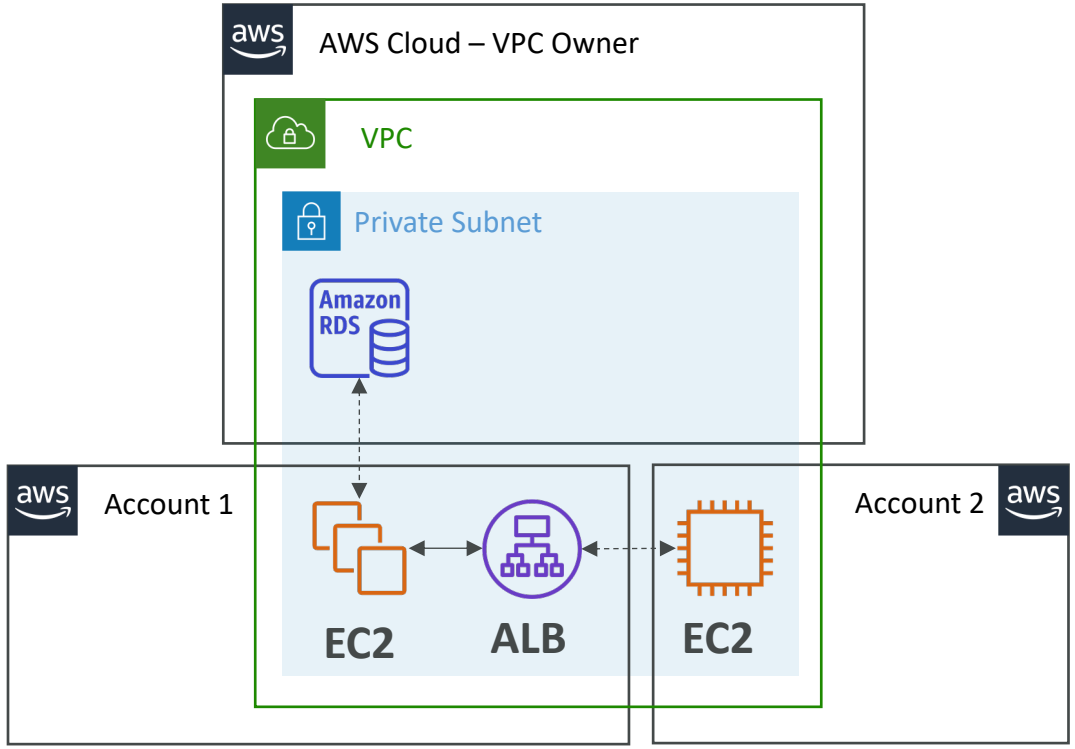


AWS Control Tower

- 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 Resource Access Manager (AWS RAM)

- Share AWS resources that you own with other AWS accounts
- Share with any account or within your Organization
- Avoid resource duplication!
- Supported resources include Aurora, VPC Subnets, Transit Gateway, Route 53, EC2 Dedicated Hosts, License Manager Configurations…



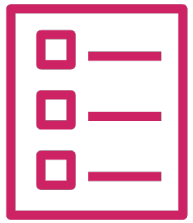
AWS Service Catalog

- 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

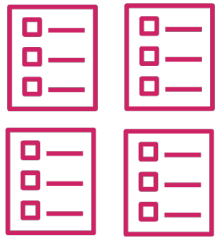
ADMIN TASKS

Product



CloudFormation Templates

Portfolio



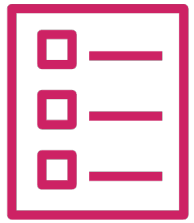
Collection of Products

Control

IAM Permissions to Access Portfolios

USER TASKS

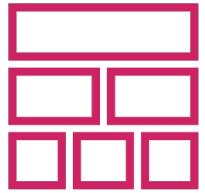
Product List



Authorized by IAM

launch

Provisioned Products



Ready to use
Properly Configured
Properly Tagged

Pricing Models in AWS

- AWS has 4 pricing models:
- **Pay as you go:** pay for what you use, remain agile, responsive, meet scale demands
- **Save when you reserve:** minimize risks, predictably manage budgets, comply with long-terms requirements
  - Reservations are available for EC2 Reserved Instances, DynamoDB Reserved Capacity, ElastiCache Reserved Nodes, RDS Reserved Instance, Redshift Reserved Nodes
- **Pay less by using more:** volume-based discounts
- **Pay less as AWS grows**

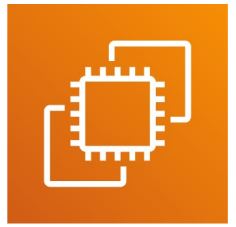
Free services & free tier in AWS

- IAM
- VPC
- Consolidated Billing
- Elastic Beanstalk
- CloudFormation
- Auto Scaling Groups

⚠ **You do pay for the resources created**

- Free Tier: https://aws.amazon.com/free/
  - EC2 t2.micro instance for a year
  - S3, EBS, ELB, AWS Data transfer

Compute Pricing – EC2



- Only charged for what you use
- Number of instances
- Instance configuration:
  - Physical capacity
  - Region
  - OS and software
  - Instance type
  - Instance size
- ELB running time and amount of data processed
- Detailed monitoring

Compute Pricing – EC2

- On-demand instances:
  - Minimum of 60s
  - Pay per second (Linux/Windows) or per hour (other)
- Reserved instances:
  - Up to 75% discount compared to On-demand on hourly rate
  - 1- or 3-years commitment
  - All upfront, partial upfront, no upfront
- Spot instances:
  - Up to 90% discount compared to On-demand on hourly rate
  - Bid for unused capacity
- Dedicated Host:
  - On-demand
  - Reservation for 1 year or 3 years commitment
- Savings plans as an alternative to save on sustained usage

Compute Pricing – Lambda & ECS

- Lambda:
  - Pay per call
  - Pay per duration

- ECS:
  - EC2 Launch Type Model: No additional fees, you pay for AWS resources stored and created in your application

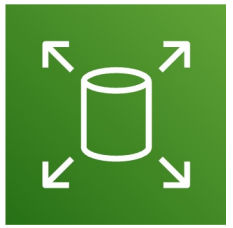
- Fargate:
  - Fargate Launch Type Model: Pay for vCPU and memory resources allocated to your applications in your containers

Storage Pricing – S3

- **Storage class:** S3 Standard, S3 Infrequent Access, S3 One-Zone IA, S3 Intelligent Tiering, S3 Glacier and S3 Glacier Deep Archive
- Number and size of objects: Price can be tiered (based on volume)
- Number and type of requests
- Data transfer OUT of the S3 region
- S3 Transfer Acceleration
- Lifecycle transitions

- Similar service: EFS (pay per use, has infrequent access & lifecycle rules)

Storage Pricing - EBS



- Volume type (based on performance)
- Storage volume in GB per month **provisionned**
- IOPS:
  - General Purpose SSD: Included
  - Provisioned IOPS SSD: Provisionned amount in IOPS
  - Magnetic: Number of requests
- Snapshots:
  - Added data cost per GB per month
- Data transfer:
  - Outbound data transfer are tiered for volume discounts
  - Inbound is free

Database Pricing - RDS

- Per hour billing
- Database characteristics:
  - Engine
  - Size
  - Memory class
- Purchase type:
  - On-demand
  - Reserved instances (1 or 3 years) with required up-front
- Backup Storage: There is no additional charge for backup storage up to 100% of your total database storage for a region.