Practical Cloud Computing with AWS

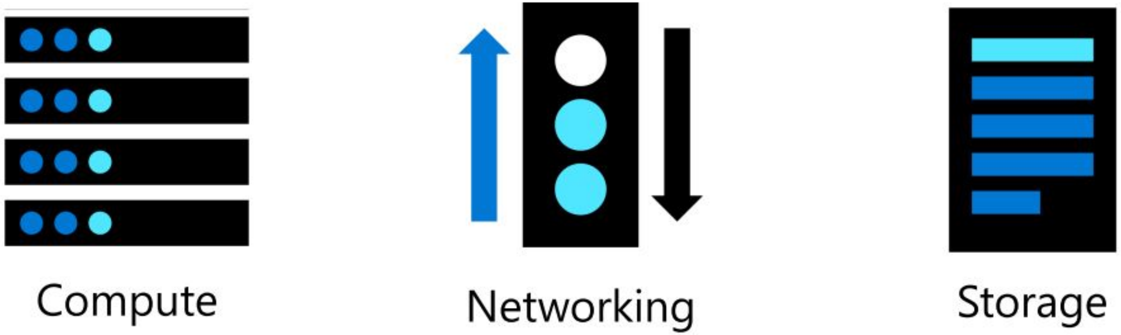
Week One

UNIVERSITY OF MARYLAND

FEARLESSLY FORWARD

What is Cloud Computing?

Cloud Computing is the delivery of computing services over the internet, enabling faster innovation, flexible resources, and economies of scale.



Benefits of Cloud Computing

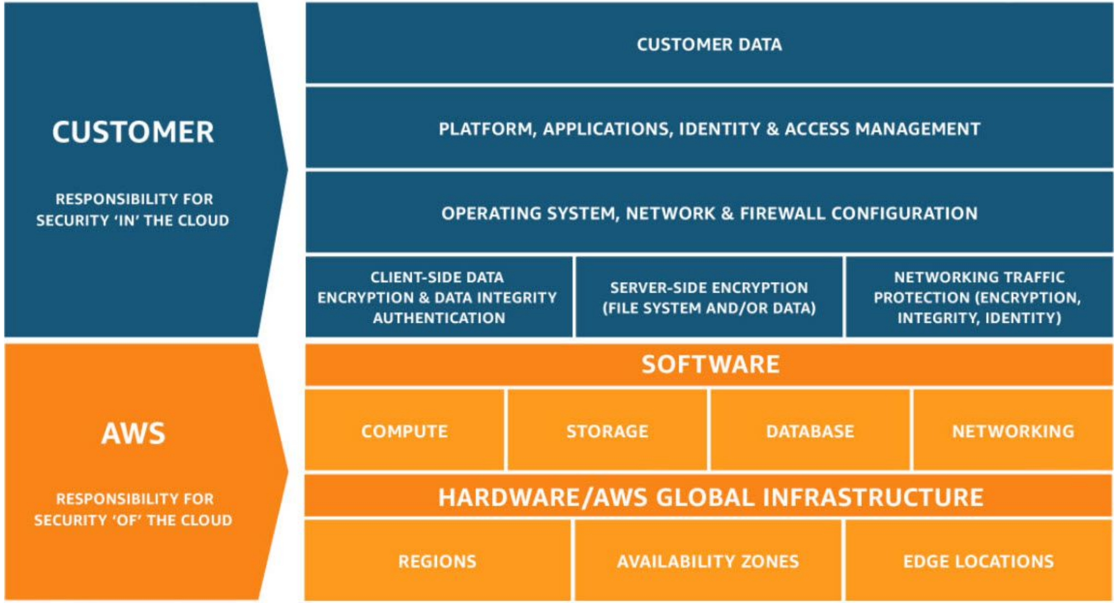
- On Demand self-service: users can provision computing services at will interacting with cloud providers

- Elasticity: Computing power and services can be automatically provisioned to scale proportionality with demand

- Reliability: The services that live on “the cloud” are designed to have an uptime of 99% to 99.9%

- Less maintenance: Sounds corny but only pay for what you need without needing to manage the physical infrastructure

Shared Responsibility Model



CUSTOMER

RESPONSIBILITY FOR SECURITY 'IN' THE CLOUD

CUSTOMER DATA

PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT

OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION

CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION

SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA)

NETWORKING TRAFFIC PROTECTION (ENCRYPTION, INTEGRITY, IDENTITY)

AWS

RESPONSIBILITY FOR SECURITY 'OF' THE CLOUD

SOFTWARE

COMPUTE

STORAGE

DATABASE

NETWORKING

HARDWARE/AWS GLOBAL INFRASTRUCTURE

REGIONS

AVAILABILITY ZONES

EDGE LOCATIONS

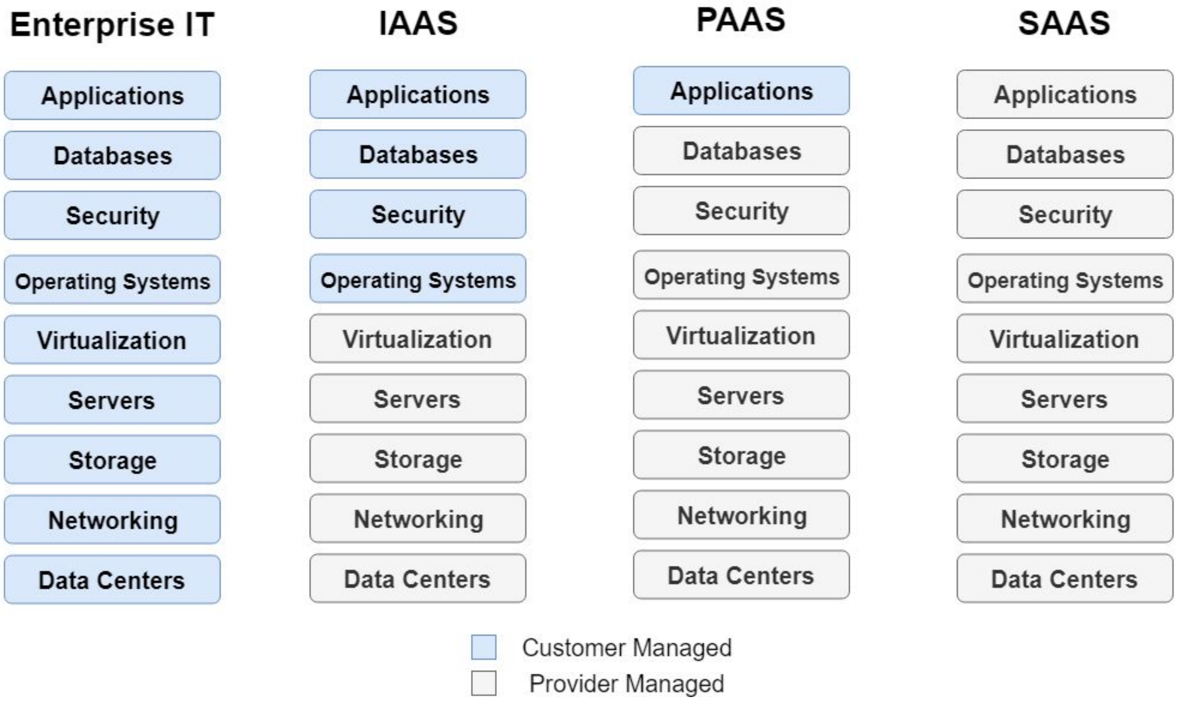
What are IaaS, SaaS, and PaaS?

- Infrastructure as a Service (IaaS): Renting servers and virtual machines (VM), storage, networks, etc from a cloud provider

- Platform as a Service (PaaS): Allows developers to build applications and deploy them without worrying about configuring infrastructure. I.e. Firebase, Vercel, Heroku.

- Software as a Service (SaaS): Packages of software that function for a specific purpose such as Gmail or ChatGPT

Breakdown of Responsibility



Enterprise IT	IAAS	PAAS	SAAS
🟦 Applications	🟦 Applications	🟦 Applications	⬜ Applications
🟦 Databases	🟦 Databases	⬜ Databases	⬜ Databases
🟦 Security	🟦 Security	⬜ Security	⬜ Security
🟦 Operating Systems	🟦 Operating Systems	⬜ Operating Systems	⬜ Operating Systems
🟦 Virtualization	⬜ Virtualization	⬜ Virtualization	⬜ Virtualization
🟦 Servers	⬜ Servers	⬜ Servers	⬜ Servers
🟦 Storage	⬜ Storage	⬜ Storage	⬜ Storage
🟦 Networking	⬜ Networking	⬜ Networking	⬜ Networking
🟦 Data Centers	⬜ Data Centers	⬜ Data Centers	⬜ Data Centers

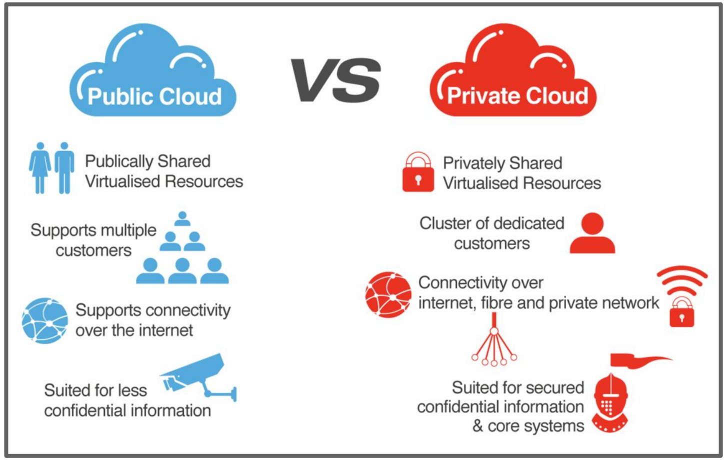
🟦 Customer Managed

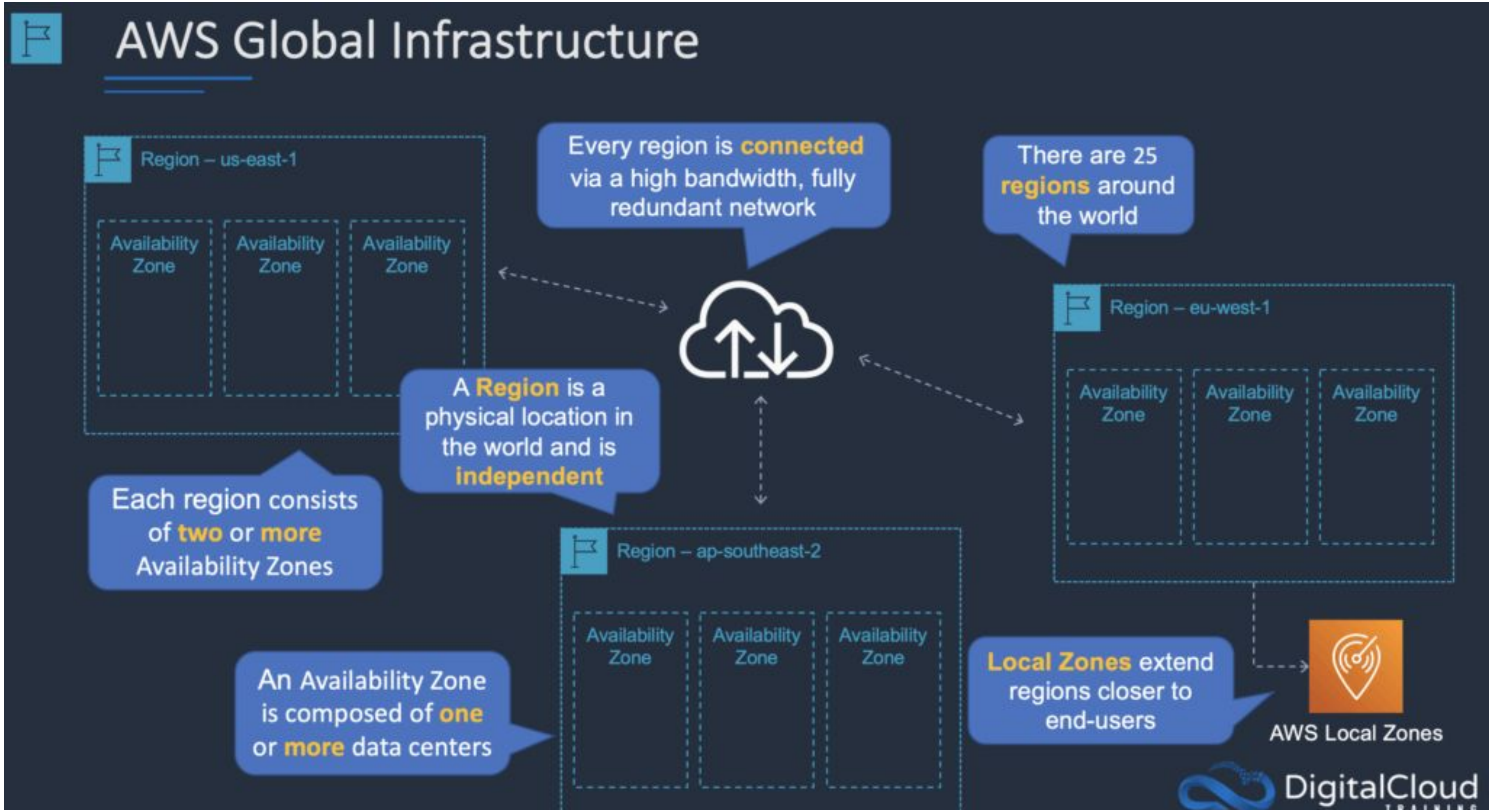
⬜ Provider Managed

Public Cloud vs Private Cloud

Public Cloud: cloud infrastructure provisioned for public use by third parties such as AWS, GCP, and Azure

Private Cloud: infrastructure provisioned for the use of one dedicated customer. They may manage or operate it with the help of a third party.


Public Cloud
VS
Private Cloud
Publically Shared Virtualised Resources
Supports multiple customers
Supports connectivity over the internet
Suited for less confidential information
Privately Shared Virtualised Resources
Cluster of dedicated customers
Connectivity over internet, fibre and private network
Suited for secured confidential information & core systems



AWS Global Infrastructure

Every region is connected via a high bandwidth, fully redundant network

There are 25 regions around the world

A Region is a physical location in the world and is independent

Each region consists of two or more Availability Zones

An Availability Zone is composed of one or more data centers

Local Zones extend regions closer to end-users

AWS Global Infrastructure

Regions: the largest measure of a geographical area for a cloud provider. Specific area such as the US northeast or even an entire country such as the UK. (us-east-1…)

Availability Zones: they specifically perform fault tolerance within regions (az-1, az-2, etc)

Edge Locations: Points of Presence (PoPs) that cache data in densely populated locations to improve end-user experience

Let's watch this video!

Intro to AWS Global Infrastructure

Next Class

- IAM users, groups, roles, and policies
- Multi-Factor Authentication (MFA)
- Best practices for securing AWS accounts
- AWS Organizations

Practical Cloud Computing with AWS

Week Two

UNIVERSITY OF MARYLAND

FEARLESSLY FORWARD

IAM - Identity Access Management

Imagine your AWS account is a university and you are the dean…

You

You are the AWS account root user which in our scenario would be the "Dean"

AWS Services

The different AWS services like S3, EC2, and VPCs would be the different buildings on campus

IAM

The IAM would be the security for the buildings which in this case would be the swipe machines and dorm security

IAM allows you to granularly control the access to your application through delegation of roles, groups, and policies.

IAM Users - The Students

IAM users are almost exactly like the unique ID cards everyone on campus has. They represent a **person or application** that needs access to interact with your app.

Each IAM user has its own set of username/pwd and the root credentials are never given to anyone.

! Critical Idea: Least Privilege Principle
Give each user only the permissions that they need to perform their job, no more!!

IAM Groups - The Departments

IAM Groups are like the departments/clubs at a university. Attach permissions to a group, then all the users within inherit the permissions.

Simplifying Management - create group called ‘Developers’ or ‘Teachers’ that can hold many users within them and assign permissions to groups

→ However, groups cannot contain other groups.

→ Groups do not have credentials (username/pwd), they are just a container of users.

IAM Policies - The Rulebook

IAM policies are like the rulebook or handbook that **define exactly what actions are allowed**. If it is not allowed, then it is implicitly denied.

Generally written in JSON format, they specify the **“who” (Principal), “what” (Action), “on what” (Resource), and under “what conditions” (Optional)**

They define whether or not to allow or deny an action which can range from any service on AWS. Resource refers to Amazon Resource Name (ARN) which identifies groups, objects, etc within the cloud environment.

IAM Policies - Cont.

AWS Managed Policies

Pre-defined policies by AWS

Inline Policies

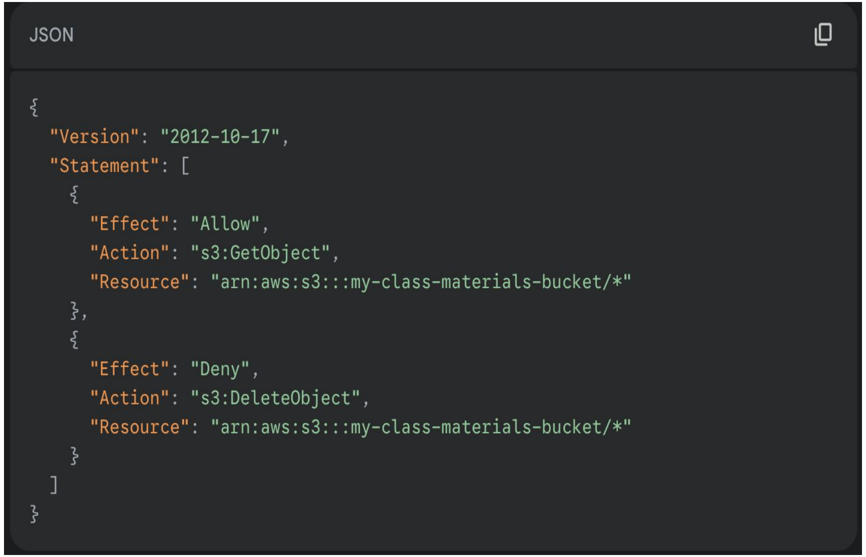
Policies embedded into a single user, group, or role

Identity Based Policies

Directly assigned to users, groups, or roles

Customer Managed Policy

Policies that you create



```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::my-class-materials-bucket/*"
    },
    {
      "Effect": "Deny",
      "Action": "s3:DeleteObject",
      "Resource": "arn:aws:s3:::my-class-materials-bucket/*"
    }
  ]
}
```

IAM Best Practices

1. **Never Use the Root User**
2. **Apply the Principle of Least Privilege:** Only grant the necessary permissions. Avoid `*` (all actions) and `*` (all resources) unless absolutely required for administrative roles, and even then, be cautious.
3. **Use IAM Groups:** Simplify permission management for multiple users.
4. **Enable MFA for Everyone:** Especially for users with elevated privileges.
5. **Use Strong Password Policies:** Configure a password policy for your IAM users (e.g., minimum length, complexity requirements, password rotation).
6. **Regularly Review Permissions:** Periodically check who has access to what. AWS provides tools like IAM Access Analyzer for this.
7. **Understand Policy Evaluation Logic:** Remember that an explicit `Deny` always overrides an `Allow`. If multiple policies apply, AWS evaluates them to determine the final permission.

AWS Organizations: The Fleet

Overview

If one AWS account was a university, then AWS Organizations is alike to a university system (UMD, UMBC, Shady Grove, etc.) The Organization (university system administration) acts as the central body, setting overall rules, budgeting, and maintaining consistency.

Purpose

Helps with isolating account-level issues and managing separate teams for large corporations. Sometimes AWS may even give a discount for large orgs with many heavy-usage accounts.

Service Control Policies

System-wide guardrails/regulations. Specifies the maximum available permissions for any IAM user or role in the accounts to which the SCP is attached. Think of it as a "filter" or an "upper limit" on what can be done.

Intro to EC2: Virtual Computers

Essentially: An EC2 Instance is a virtual server/machine that lives on the ‘cloud.’

Advantages

01 On Demand availability: launch one whenever you want

02 Scalability: Automatically add more with high traffic and terminate

03 Flexibility: choice of machine (Linux, Windows, Mac, etc.), RAM, CPU, storage

04 Full admin access to the instance, download anything you want

Intro to EC2: Instance Types

On Demand Instances

Highest per hour cost, no long-term commitment, pay per usage.

Best for: Unpredictable workloads, new applications, dev environments.

Reserved Instances

Significantly discounted (up to 75% off On-Demand instances), 1 or 3 year commitment, guaranteed server space.

Best for: steady, predictable workloads, long-term work.

Spot Instances

Low cost (up to 90% off On-Demand), no commitment, server space may be interrupted.

Best for: flexible, fault-tolerant, and interruptible workloads.

AMI: Amazon Machine Image

AMIs are pre-packaged template configurations of EC2 instances that have everything needed to deploy as is. Especially used to help assist with quick deployment of instances.

Public AMIs: These are the ones provided by AWS (Amazon Linux 3, Ubuntu, Windows Machine)

AWS Marketplace: Third-party vendors may offer special AMIs with configured databases, firewalls, etc.

Custom AMIs: Templates that you’ve created.

Practical Cloud Computing with AWS

Week Three

UNIVERSITY OF MARYLAND

FEARLESSLY FORWARD

EC2 Instances Types Cont.

Reserved Instances ☆

- → Upto 77% discount compared to On-Demand instances. This can increase as the commitment (1 year or 3 years) increases. Payment is flexible.

Spot Instances ⊗

- → Upto 90% discount compared to On-Demand instances

Dedicated Instances ☆

- → These are instances run on your hardware that you may share with other accounts. No control over where instances are placed.

Dedicated Hosts

- → Rent a physical server with EC2 instance capacity fully dedicated to you (very expensive)

Savings Plan ⚠

- → Can get a discount on long-term usage. Limited to use of one type and one region at a time. Usage beyond the prepaid amount is billed at on-demand rates

EC2 Purpose Types

General Purpose

Instances are for a wide range or workloads. (t2, t3, t4g: burstable instances good for web server or small dev environment) (m5, m6i, m7g: general workhorse instances)

Compute Optimized

Great for CPU intensive tasks such as high-perf web servers, batch processing, or scientific modeling (c5, c6a, c7g)

Memory Optimized

Designed for tasks that process large datasets (r5, r6i, r7g: high-perf relational and NoSQL databases and real-time big data)

Storage Optimized

Ideal for workloads that require high speed access to massive datasets on local storage (i3, i4g, i4i: focused on high I/O perf)

Security Groups for EC2

Bottomline: a gate that controls the type of traffic allowed in and out of the EC2 instance per region

Configuration Options

➔ **IP Address:** Port # or security group(s)

➔ **Security Group:** can be attached to multiple instances and an instance can be part of multiple groups

➔ **Note:** timeout error often means that traffic never reached and usually can be tracked down to the security group.

Common Port Numbers

Port	Use	Purpose
22	Secure Shell	Allows the user to control the ec2 remotely through the command line
22	SFTP	Uses the same as SSH to provide encrypted file transfer
21	FTP	Dedicated for file transfers
80	HTTP	For unsecured websites
443	HTTPS	For secured websites
3389	RDP	Specifically meant for windows instances

Elastic Load Balancing

Essentially: In distributed systems of two or more instances, AWS can manage traffic and algorithmically distribute data to specific instances

In order to achieve this, we use Elastic Load Balancers, which forward incoming internet traffic to multiple EC2 instances, while only exposing a single point of access (one DNS).

Main advantages include:

- ➔ Regular health checks on instances
- ➔ Provide end to end SSL (HTTPS) connection for sites
- ➔ High Availability across different zones within a region
- ➔ Fully-managed by AWS so no managed infrastructure overhead

Types of Load Balancers

Application Load Balancer

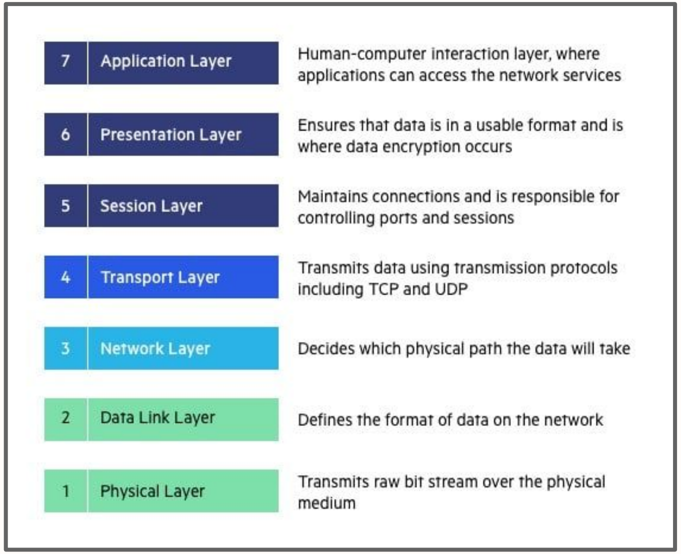
(HTTP/HTTPS only): Based on Layer 7 routing for static DNS

Network Load Balancer

(TCP & UDP only): Based on Layer 4 and designed for high performance with ultra-low latencies. Uses elastic IP address to mask static IP in the backend

Gateway Load Balancer

Based on Layer 3, routes traffic to firewalls that are managed by YOU on EC2 instances



7	Application Layer	Human-computer interaction layer, where applications can access the network services
6	Presentation Layer	Ensures that data is in a usable format and is where data encryption occurs
5	Session Layer	Maintains connections and is responsible for controlling ports and sessions
4	Transport Layer	Transmits data using transmission protocols including TCP and UDP
3	Network Layer	Decides which physical path the data will take
2	Data Link Layer	Defines the format of data on the network
1	Physical Layer	Transmits raw bit stream over the physical medium

Auto Scaling

Load Balancers distribute traffic amongst existing instances, but auto scalers increase the number of instances when traffic is congested even with load balancers.

Advantages

- → Ensures there is always an instance running
- → Auto scales in and out to match the minimum number required
- → Automatically deploys new instances to replace unhealthy instances through its own health checks

Auto Scaling Cont.

Manual Scaling: you can choose to update the size of the scaling group manually

Dynamic Scaling:

- ➔ Simple/Step scaling: add 2 instances if a CPU capacity (ex. > 70%) is achieved.
- ➔ Target Tracking Scaling: the average CPU of the group must be ~ 50%. The group will provision resources or terminate resources to match the target.
- ➔ Scheduled Scaling: set a predetermined scaling plan based on existing usage patterns. Can be scheduled manually as well.
- ➔ Predictive Scaling: use real-time existing usage data to have ML determine resources ahead of time and automatically provision resources. Useful for predictable usage patterns.

Serverless Intro

- ➔ Serverless is relatively new term that allows developers to abstract away the server aspect.

- ➔ Allows developers to focus on deploying code and functions.

- ➔ Main point of confusion: “Serverless” doesn’t mean that there are no servers, it just means that you aren’t managing them.

- ➔ A main example of this concept is AWS Lambda.

AWS Lambda

The Big Idea: EC2 has physical limitations for RAM and CPU. Running it for long periods of time can be expensive.

Advantages

01 Virtual servers with no overhead to manage

02 Running on-demand saves costs and scaling is automated on usage

03 Relatively inexpensive: based on pay per request and compute time (calls and duration) with a very generous free tier (1,000,000 requests & 400,000 GB)

04 Event-Driven: functions get invoked by AWS when needed so it is reactive

Practical Cloud Computing with AWS

Week Four

UNIVERSITY OF MARYLAND

FEARLESSLY FORWARD

EC2 Instance Storage

Elastic Block Storage

- ➔ A type of network drive that you can attach to EC2 instances
- ➔ Pay for GB of volume storage per month
- ➔ The main advantage is that it allows your data to persist beyond instance termination
- ➔ Can be detached and attached to new instances restoring the data to a new instance
- ➔ Can configure settings to automatically take periodic snapshots of volume and store them in archive

EC2 Instance Store

- ➔ Data can be deleted if instance is terminated
- ➔ But is more powerful and faster than EBS
- ➔ Good for buffer, cache, or temporary data

Elastic File Storage (EFS)

Elastic File Storage (EFS)

A shared network file system that can be mounted on many **MANY** instances, Works across multiple Availability Zones. High availability, scalable, but expensive.

Note: Good alternative to EBS since EBS is not cross AZ

Amazon FSx

Derivative of EFS designed for third party file solutions such as Windows File Server. Can be accessed from AWS or on-premises infrastructure.

FSx for Lustre

Fully-managed meant for High-Performance Computing use cases.

Amazon S3

The Big Idea: Allows people to store objects (files) in ‘buckets’ (directories)

Advantages

01 Used for disaster recovery, archive, hybrid cloud storage, application and media hosting

02 S3 can host static websites and have them accessible on the Internet with the URL

03 High durability (99.999%) of objects across multiple AZ

04 Ex. if you store 10,000,000 objects with Amazon S3, you can on average expect to incur a loss of a single object once every 10,000 years

Security

Policies

IAM Policies	Which API calls should be allowed for a specific user from IAM
Bucket Policies	Which can be used to grant public access to the bucket, force objects to be encrypted at upload, and grant access to another account (cross account)

Encryption

Server-Side Encryption	User uploads a file as normal to an S3 bucket and the server encrypts the file. Can be either with S3 or KMS (SSE-S3 or SSE-KMS).
Client-Side Encryption	User encrypts the file before uploading it and the S3 bucket stores it as normal

Replication and Storage Gateway

Buckets can be in different AWS accounts and copying is asynchronous.

- ➔ Must enable versioning in source and destination buckets and must give proper IAM perms to S3
- ➔ Cross-Region Replication (CRR) – used for compliance, lower latency access, replication across accounts
- ➔ Same-Region Replication (SRR) – used for log aggregation, live replication between production and test accounts

AWS Storage Gateway: a hybrid storage service to allow on-premises infra. to seamlessly use the AWS Cloud

S3 Storage Classes

S3 Standard ☆

→ (99.99% availability)
→ Used for frequently accessed data with low latency and high throughput use cases
→ Big Data analytics, mobile & gaming apps, etc.

S3 Infrequent Access ⊗

→ To store data that is less frequently accessed
→ But requires rapid access when needed; lower cost than S3 Standard

S3 One Zone IA

→ (99.50% Availability)
→ High durability in a single AZ, but data lost when AZ is destroyed
→ Use cases: Secondary backup copies of on-premise data

S3 Standard IA ⚠

→ (99.90% Availability)
→ Use cases: Disaster Recovery, backups, etc.

S3 Storage Classes Continued

Glacier Tiers

Low-cost object storage meant for archiving/backup

- ➔ S3 Glacier Instant Retrieval (90 day storage minimum)
- ➔ S3 Glacier Flexible Retrieval (90 day storage minimum)
- ➔ S3 Glacier Deep Archive (180 day storage minimum)

Intelligent Tiering

Small monthly monitoring and auto-tiering fee (but no retrieval charges) to move objects automatically between access tiers based on usage.

Express One Zone

High performance, single AZ storage class with objects stored in a Directory Bucket (bucket in a single AZ)
Purpose: co-locate your storage and compute resources in the same AZ (reduces latency)

AWS Snowball

Essentially (highly secure and portable) offline devices to perform data migrations. If it takes more than a week to transfer over the network, then use snowball devices.

Edge Storage Optimized: has much more storage capacity and mean for big data transfers.

Edge Computing Optimized: process data while it's being created on an edge location (ex. A truck on the road, a ship on the sea, etc.)

Note: These locations may have limited internet and no access to computing power so by setting up a snowball edge device we are able to run EC2 instances or Lambda functions.

Practical Cloud Computing with AWS

Week Five - AWS Databases

UNIVERSITY OF MARYLAND

FEARLESSLY FORWARD

Intro to Relational Database Service

Bottomline: RDS is a managed database service that uses SQL. You can choose between various database engines of choice.

- ✓ AWS fully managed RDS database engine OS
- ✓ Disaster recovery options for restoration
- ✓ Read replicas in multi AZ setups and scaling capability

RDS cont.

Read Replicas ☆

→ Scale the "read" workload of your DB by creating up to 15 replicas

Multi-AZ ⊗

→ Automatic failover in case of AZ outage. But can only have 1 AZ as failover

Multi Region (Read replicas)

→ Data will be read across regions. Meant for global cases with less latency

Side Note ⚠

→ Can purchase on-demand or reserved instances (1 or 3 years) with optional payment up front

AWS Aurora

Purpose: supports compatibility between PostgreSQL and MySQL.

➔ It stands out because AWS claims a **5x performance boost over MySQL and 3x over PostgreSQL**.

➔ Storage grows in steps of 10 GB all the way to 128 TB.

➔ However, Aurora costs 20% more than RDS but much more efficient.

➔ Serverless: automated database instantiation and auto-scaling; very low management overhead. Great for infrequent or unpredictable workloads.

AWS DynamoDB

DynamoDB is the **NoSQL brother to the RDS SQL options**.

➔ Fully managed, highly scalable, and available across 3 different AZs.

➔ It is mainly serverless with the ability to scale to **100s of TB of storage**.

➔ Performance is consistently in **single-digit millisecond latency** even with thousands of JSON objects as rows of data.

➔ Runs on a key/value partition to store data for quick retrieval.

AWS Dynamo Accelerator (DAX)

Purpose: fully managed, highly available caching service built for Amazon DynamoDB

Advantages:

✓ Secure

✓ Highly scalable and available

✓ 10 times performance improvement from milliseconds to microseconds

✓ Adds in-memory acceleration

AWS Elasticache

Purpose: get managed Redis or Memcached databases

Cache: in-memory databases with high performance, low latency

➔ Helps reduce load off databases for read intensive workloads by storing some queries in memory for faster retrieval

AWS Database Migration Service

Purpose: Migrate legacy applications and databases to AWS DBs in data centers.

- ➔ Homogenous Migrations – migrations where the same database technology is used for the source and target databases (ex. Oracle to Oracle)

- ➔ Heterogenous Migrations – migrations where different database technology is used for the source and target databases

AWS Redshift

Purpose: fully managed data warehouse service.

➔ Mainly designed for Online Analytical Processing (OLAP) workloads for large datasets and business intelligence tools.

Advantages:

✓ 10x better performance than other warehouses

✓ Scale to PBs of data with Massive Parallel Query Execution (MPP)

✓ Pay as you go based on the instances provided

✓ SQL interface for the queries

✓ Has a serverless option as well

AWS Athena

Purpose: Serverless query service to perform analytics against S3 objects

- ➔ S3 data is loaded → Athena can query and analyze it → B.I. workflow for reporting or analytics purpose.
- ➔ Supports CSV, JSON, Avro, etc through SQL querying.

Practical Cloud Computing with AWS

Week Six - Monitoring

UNIVERSITY OF MARYLAND

FEARLESSLY FORWARD

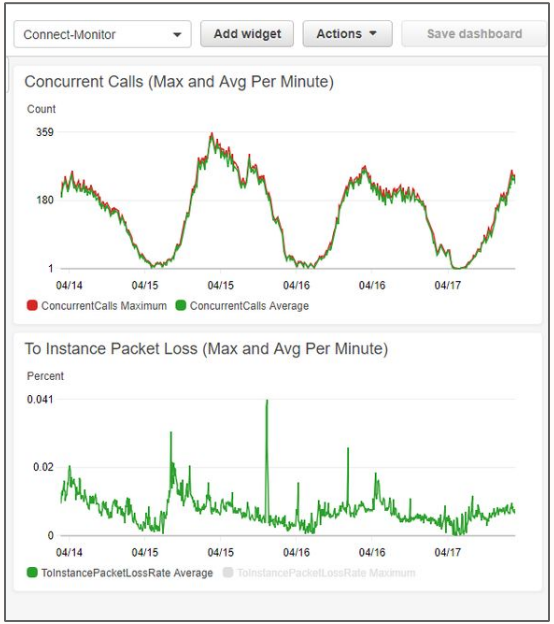
AWS CloudWatch

CloudWatch provides metrics for every services in AWS

- ➔ Metric is a variable with timestamps to monitor various services

Metrics ex:

- ➔ EC2: CPU Utilization
- ➔ S3 bucket: NumberOfObjects, BucketSizeBytes
- ➔ Create custom metrics



Concurrent Calls (Max and Avg Per Minute)

To Instance Packet Loss (Max and Avg Per Minute)

Percent
0.041
0.02
0
04/14
04/15
04/15
04/16
04/16
04/17
🟩 ToInstancePacketLossRate Average
ToInstancePacketLossRate Maximum

AWS CloudWatch

Alarms - used to trigger notifications for any metric

- ➔ States: OK, INSUFFICIENT_DATA, ALARM
- ➔ If metric crosses threshold → trigger some alarm action
  - ◆ Ex: Disk utilization is too high → auto scaling: increase number of EC2 instance

Logs - Centralize logs from all systems, applications, and services in one place

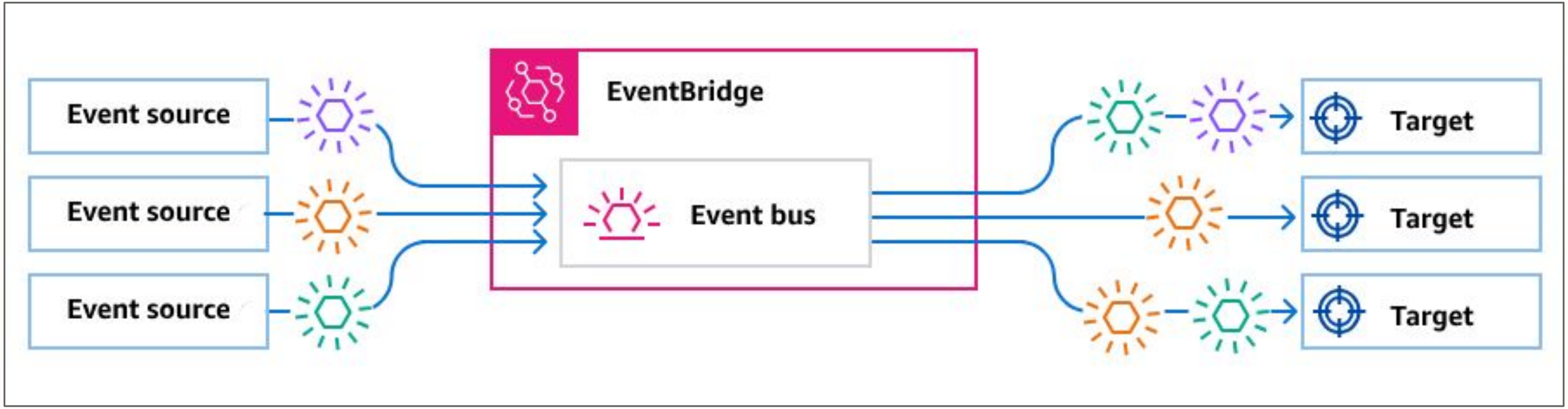
- ➔ Easily view, search, filter, archive, and query them
- ➔ Two types: Logs Standard and Logs Infrequent Access
- ➔ Log agents: exist on EC2 instances or on prem servers to push log files when needed

Amazon EventBridge

EventBridge - serverless scheduler, schedules based on a response

- ➔ Cron jobs (based on time) or recurring timer (every 10 minutes)
- ➔ “Schedule millions of tasks that can invoke more than 270 AWS services and over 6,000 API operations”
- ➔ Rules: react to a service doing something
- ➔ Event bus: router that receives events and delivers them to targets
  - ◆ Default event bus → triggered for AWS services
  - ◆ Partner event bus → triggered for any AWS SaaS partners like Datadog
  - ◆ Custom event bus → triggered for custom applications

Amazon EventBridge



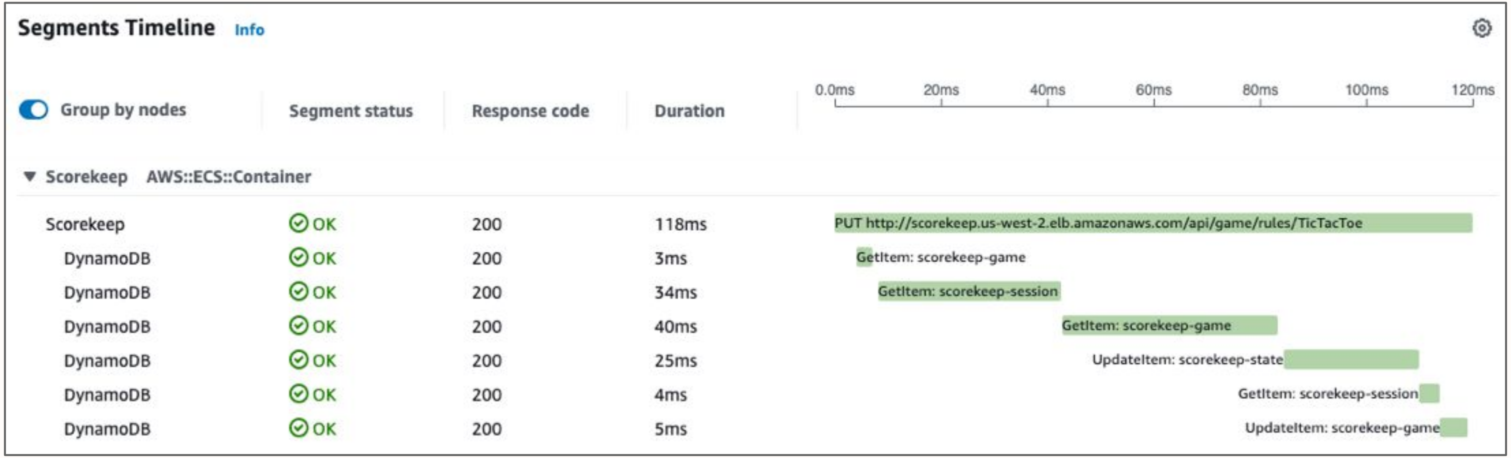
AWS CloudTrail

CloudTrail - provides governance, compliance, and audit for your AWS Account, default

- ➔ Trails – Trails capture a record of AWS activities, delivering and storing these events in an Amazon S3 bucket
- ➔ AWS CloudTrail Lake is a managed data lake for capturing, storing, accessing, and analyzing user and API activity
- ➔ Event history – The Event history provides a viewable, searchable, downloadable, and immutable record of the past 90 days of management events in an AWS Region
- ➔ Insights - automated analysis of the CloudTrail events

AWS X-Ray

X-Ray - collects data about requests that your application serves and provides tools to simplify log analysis and debugging



Segments Timeline Info

▼ Scorekeep AWS::ECS::Container

Group by nodes	Segment status	Response code	Duration	
Scorekeep	✓ OK	200	118ms	PUT http://scorekeep.us-west-2.elb.amazonaws.com/api/game/rules/TicTacToe
DynamoDB	✓ OK	200	3ms	GetItem: scorekeep-game
DynamoDB	✓ OK	200	34ms	GetItem: scorekeep-session
DynamoDB	✓ OK	200	40ms	GetItem: scorekeep-game
DynamoDB	✓ OK	200	25ms	UpdateItem: scorekeep-state
DynamoDB	✓ OK	200	4ms	GetItem: scorekeep-session
DynamoDB	✓ OK	200	5ms	UpdateItem: scorekeep-game

AWS Health

Health Dashboard – console to show resource performance and the availability of all services

- ✓ Configurable to a more personalized view
- ✓ Provides alerts and remediation guidance
- ✓ Available for all AWS customers at no additional cost

Health events – events to learn how service and resource changes might affect your applications running on AWS

Before we start the lesson, let's talk about

Why did AWS go down?

Source: https://aws.amazon.com/message/101925/

UNIVERSITY OF MARYLAND

FEARLESSLY FORWARD

What happened?

There were three distinct periods of impact to customer applications

1. Between 11:48 PM on October 19 and 2:40 AM on October 20, Amazon DynamoDB experienced increased API error rates in the N. Virginia (us-east-1) Region
2. Between 2:25 AM and 10:36 AM on October 20, new EC2 instance launches failed
3. Between 5:30 AM and 2:09 PM on October 20, Network Load Balancer (NLB) experienced increased connection errors for some load balancers in the N. Virginia (us-east-1) Region

Issue 1 - DynamoDB

The root cause of this issue was a latent race condition in the DynamoDB DNS management system that resulted in an incorrect empty DNS record for the service's regional endpoint (dynamodb.us-east-1.amazonaws.com) that the automation failed to repair.

Race condition with DNS Enactors (looks for new plans and attempts to update Route53 by replacing the current plan with a new plan)

First enactor experienced extreme delays causing it to run after the second enactor and overwrite the newer plan with an older plan, function to check if plan is newer than current was stale, then the second enactor deleted the older plan in its cleanup process deleting all IP addresses for the regional endpoint

Manual fix required

Issue 2 - EC2

Uses DropletWorkflow Manager (DWFM), which is responsible for the management of all the underlying physical servers that are used by EC2 for the hosting of EC2 instances (droplets)

DWFM state checks began to fail as the process depends on DynamoDB and was unable to complete

DWFM maintains a lease for each droplet allowing it to track the droplet state, once DynamoDB came back DWFM began to re-establish leases with droplets across the EC2 fleet but due to the large number of re-established leases, DWFM went into congestive collapse

Fixed by engineers, still some throttling, but once all requests processed, recovered

Issue 3 - Network Load Balancers

NLB health checking subsystem began to experience increased health check failures because of the health checking subsystem bringing new EC2 instances into service while the network state for those instances had not yet fully propagated

This resulted in health checks alternating between failing and healthy, alternating health check results increased the load on the health check subsystem, causing it to degrade, resulting in delays in health checks and triggering automatic AZ DNS failover to occur

Engineers disabled automatic health check failovers for NLB, allowing all available healthy NLB nodes and backend targets to be brought back into service

After EC2 recovered, automatic checks were enabled again

Practical Cloud Computing with AWS

Week Eight - Containerization

UNIVERSITY OF MARYLAND

FEARLESSLY FORWARD

Docker

Docker - open source platform allowing you to separate your applications from your infrastructure

- ➔ Packages software into standardized units called containers
- ➔ Operating system for containers similar to VMs
- ➔ Easy to build and run distributed microservice architectures

Amazon ECS

Elastic Container Service (ECS) - fully managed container orchestration service

- Capacity - infrastructure where your containers run
  - EC2 Instances, Fargate, On-premises compute
- Controller - deploy and manage your applications that run on containers
  - ECS Scheduler
- Provisioning - tools that you can use to interface with scheduler
- You must provision and maintain the infrastructure
- Integrations with Application Load Balancer to allocate enough EC2 instances to run your containers

Amazon ECR

Elastic Container Registry (ECR) - managed container image registry on AWS where you can store Docker images so they can be run by ECS or Fargate

- ➔ Supports public container image repositories as well
- ➔ Supports private repositories with resources based permissions in IAM

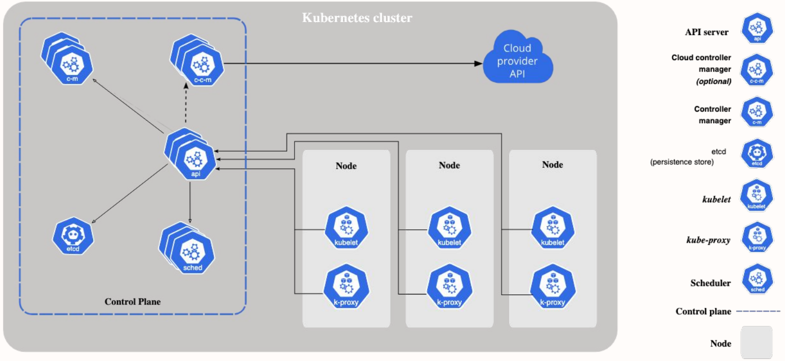
Amazon EKS

Elastic Kubernetes Service (EKS) – fully managed Kubernetes service

➔ EKS Standard – manages Kubernetes control plane

➔ EKS Auto Mode – also manages nodes (Kubernetes data plane)

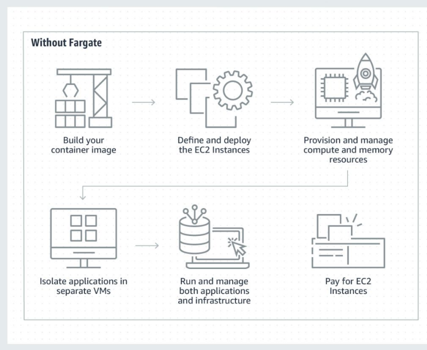
Kubernetes – open source system for management, deployment, and scaling of containerized apps and cloud agnostic

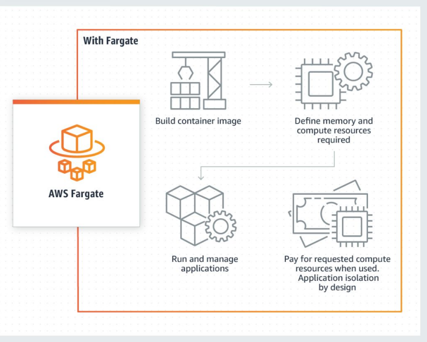


AWS Fargate

Fargate - technology to run containers without managing servers or EC2 clusters

- ➔ If new docker container, Fargate will automatically run that container for us
- ➔ Pay for vCPU and memory resources allocated to your applications
- ➔ Fargate Spot - run interruption tolerant ECS tasks at a discounted rate

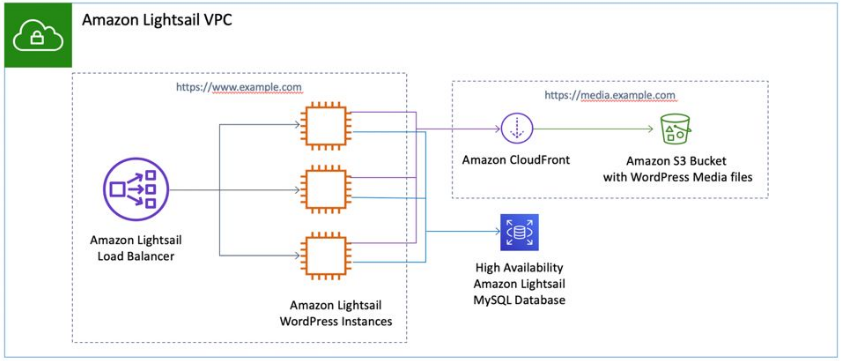

Without Fargate
Build your container image
Define and deploy the EC2 Instances
Provision and manage compute and memory resources
Isolate applications in separate VMs
Run and manage both applications and infrastructure
Pay for EC2 Instances


With Fargate
AWS Fargate
Build container image
Define memory and compute resources required
Run and manage applications
Pay for requested compute resources when used. Application isolation by design

Amazon Lightsail

Lightsail - easy-to-use virtual private server (VPS) that offers simple management of cloud resources such as containers

- ➔ High availability but no auto scaling, limited AWS integrations
- ➔ Great for people with little cloud experience with low and predictable pricing


Amazon Lightsail VPC
https://www.example.com
Amazon Lightsail Load Balancer
Amazon Lightsail WordPress Instances
https://media.example.com
Amazon CloudFront
Amazon S3 Bucket with WordPress Media files
High Availability Amazon Lightsail MySQL Database

AWS Batch

Batch - fully managed batch processing at any scale that can efficiently run 100,000s of computing batch jobs

- ➔ Batch jobs - job with a start and an end, can be defined as Docker images and run on ECS
- ➔ Can dynamically launch EC2 instances or Spot instances
- ➔ No time limit, not serverless, can have any runtime

Practical Cloud Computing with AWS

Week Nine - Networking

UNIVERSITY OF MARYLAND

FEARLESSLY FORWARD

IP Addresses in AWS

- The most common is IPv4: Internet Protocol version 4 (4.3 Billion Addresses)
- Public IPv4 – can be used on the Internet
- Private IPv4 – can be used on private networks (LAN) such as internal AWS networking
  Example Private IP: (e.g., 192.168.1.1)

Elastic IP – allows you to attach a fixed public IPv4 address to EC2 instance
IPv6 – Internet Protocol version 6 (3.4 × 10!" Addresses)

- Every IP address is **<u>public</u>** in AWS (no private range)
- Example: 2001:db8:3333:4444:cccc:dddd:eeee:ffff
- Free

<u>Note</u>: all public IPv4 on AWS will be charged $0.005 per hour (including EIP)
<u>Note</u>: EC2 instance gets a new a public IP address every time you stop then start it (default) Unless you use an Elastic IP address

Amazon VPC

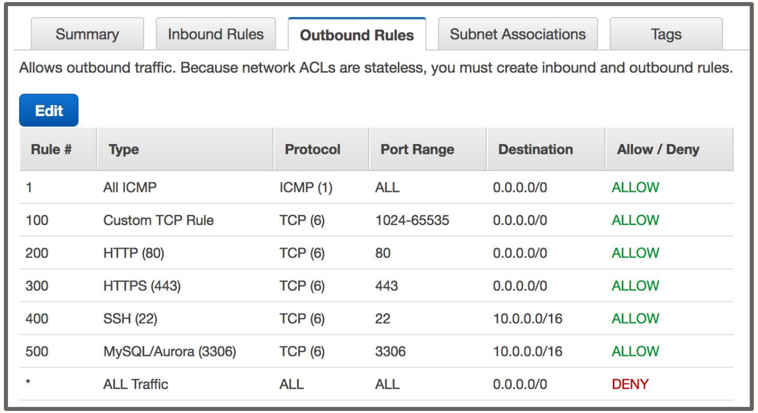
- ➔ Bottom line: Think of a VPC as your own customizable networking solution surrounding your cloud infrastructure.
- ➔ A VPC gives you complete control over the range of IP addresses and ports that can access your network. Specifically allow/block which destinations data can go into or out of.
- ➔ Some key components include:
  - ◆ Network Allow Control Lists
  - ◆ Security Groups
  - ◆ Subnets
  - ◆ Route Tables

VPC Subnets

- ➔ Subnets allow you to partition your network inside your VPC
- ➔ An Availability Zone (AZ) is connected to exactly One Subnet resource (there should always be a one-to-one ratio between AZs and Subnets)
- ➔ Subnets break up the CiDR block given to a VPC into sections
- ➔ 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

Amazon NACLs

- A firewall attached at the Subnet level which controls traffic from and to subnet
- Can have ALLOW and DENY rules
- Rules only include IP addresses
- Stateless: return traffic must be explicitly allowed by rules
- Allows for setting **priority rules**, such as an explicit, low-numbered DENY rule to block a known bad IP, followed by broader ALLOW rules.

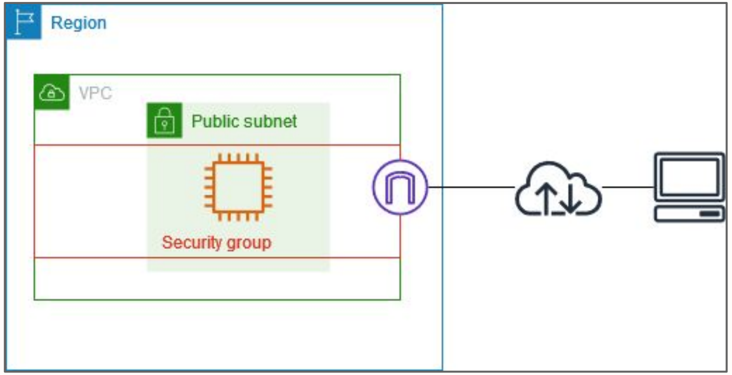


Allows outbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.

Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
1	All ICMP	ICMP (1)	ALL	0.0.0.0/0	ALLOW
100	Custom TCP Rule	TCP (6)	1024-65535	0.0.0.0/0	ALLOW
200	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW
300	HTTPS (443)	TCP (6)	443	0.0.0.0/0	ALLOW
400	SSH (22)	TCP (6)	22	10.0.0.0/16	ALLOW
500	MySQL/Aurora (3306)	TCP (6)	3306	10.0.0.0/16	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Security Groups

- ➔ A firewall attached at the instance level that controls traffic to and from an EC2 Instance
- ➔ Generally meant for individual resources
- ➔ Can have only ALLOW rules
- ➔ Rules include IP addresses and other security groups
- ➔ Stateful: return traffic is automatically allowed, regardless of any rules



VPC NAT Gateway

- ➔ There are two resources: Network Access Translation **Gateways & Instances**
- ➔ Main Idea: allow your instances in your Private Subnets to access the internet while remaining private. Compatible with NACLs.

NAT Gateway:

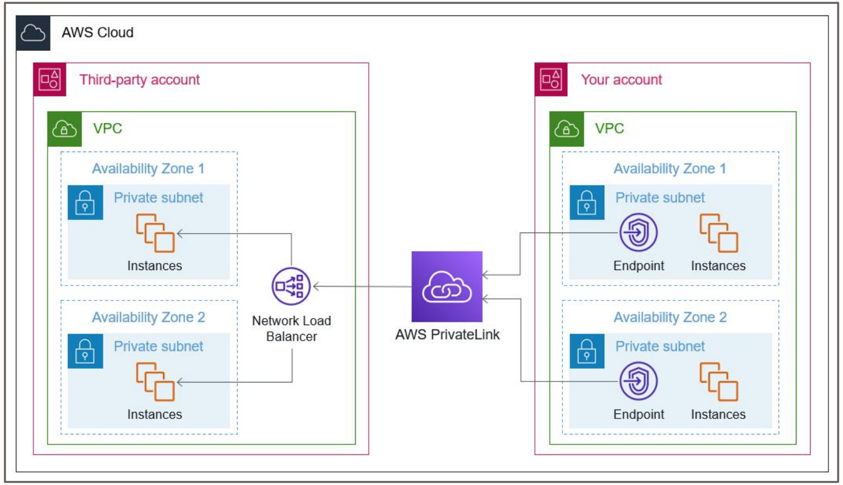
- Fully Managed by AWS; no maintenance required
- Auto scaling, but minimal customization. Not an instance
- Highly available within a single AZ

NAT Instance

- Requires manual management: OS, patches, software, scaling
- Single point of failure, but customizable instance with ability to add security groups

Amazon PrivateLink

- ➔ Most secure & scalable way to expose a (private) service to 1000s of VPCs
- ➔ Does not require VPC peering, internet gateway, NAT, route tables, etc. since it is on a private network



Amazon Route 53

➔ A fully-managed DNS service that is part of the AWS Global Architecture

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

Routing Policies:

Simple Routing Policy

No health checks

Latency Routing Policy

Route users to geographically closest servers to reduce latency

Weighted Routing Policy

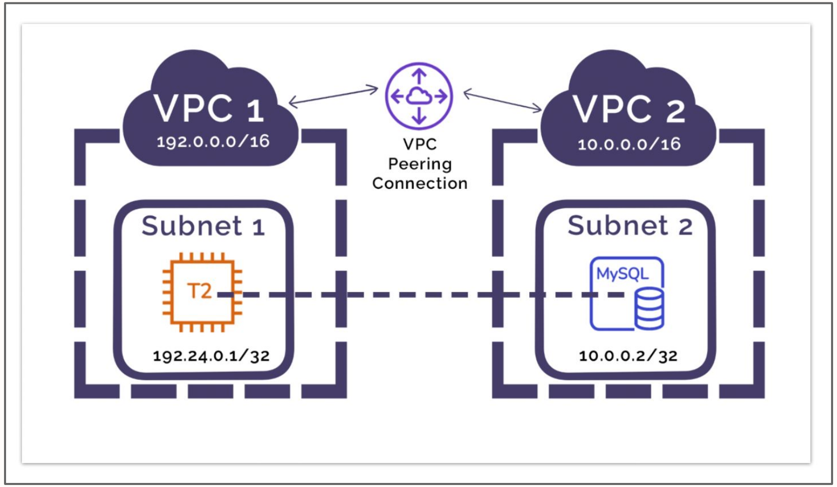
Distribute traffic across multiple EC2 in proportions you specify

Failover Routing Policy

Health check on primary instance and redirect if fail

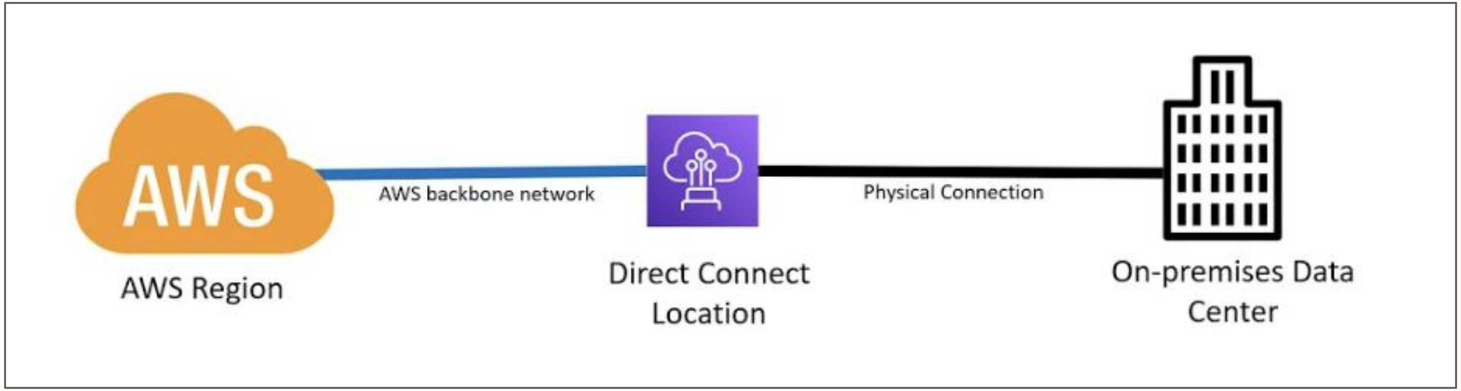
VPC Peering

- ➔ Connect two VPC, privately using AWS networks
- ➔ Must not have overlapping CIDR ranges
- ➔ VPC Peering connection is not transitive



Direct Connect (DX)

- ➔ Establish a physical, private connection between on-premises and AWS
- ➔ The connection is private, secure and fast
- ➔ Takes at least a month to establish



Transit Gateway

- ➔ For having transitive peering between thousands of VPC and on-premises, hub-and-spoke model
- ➔ No need to peer the VPC with one another, or create a bunch of connections and routes, all of this is done through one gateway
- ➔ Works with Direct Connect Gateway, VPN connections

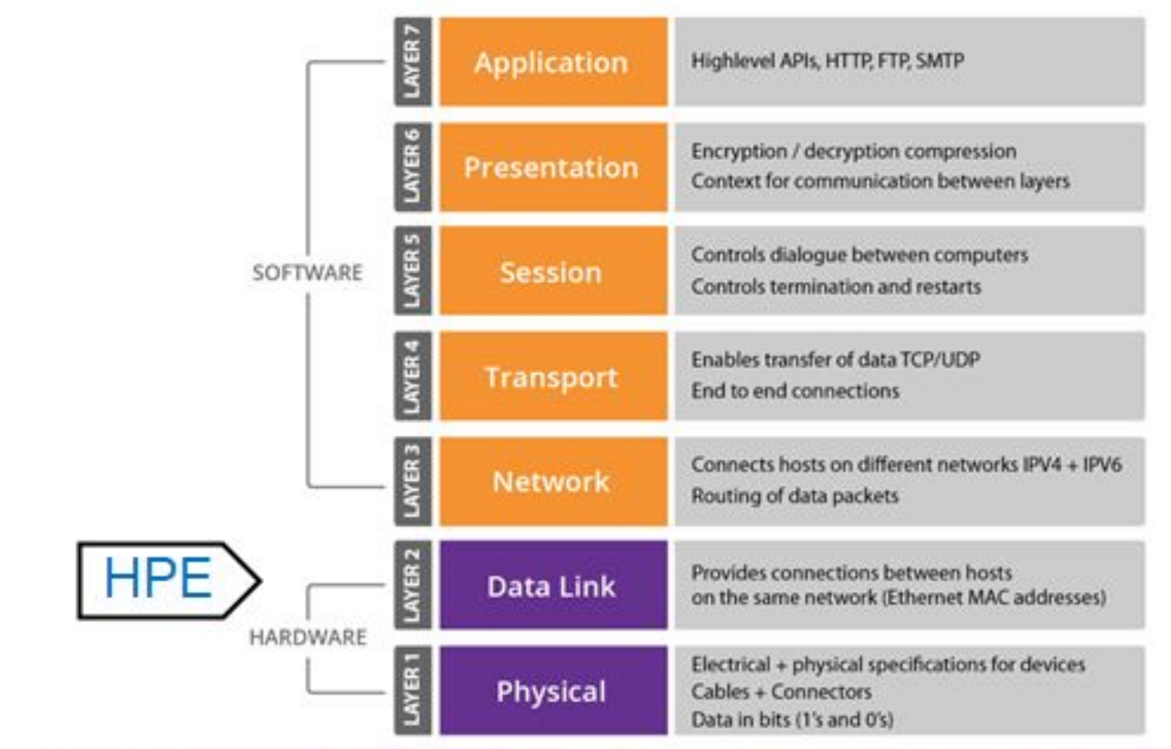
Practical Cloud Computing with AWS

Week Ten - Security Tools

Overview of the Tools

- ➔ AWS Shield
- ➔ AWS WAF (Web Application Firewall)
- ➔ AWS Firewall Manager
- ➔ AWS KMS
- ➔ Cloud HSM
- ➔ AWS Certificate Manager
- ➔ AWS Secrets Manager
- ➔ AWS Guard Duty
- ➔ Amazon Inspector
- ➔ Amazon Macie
- ➔ AWS Security Hub
- ➔ Amazon Detective
- ➔ AWS Config
- ➔ AWS Artifact

Layers



SOFTWARE	LAYER 7	Application	Highlevel APIs, HTTP, FTP, SMTP
	LAYER 6	Presentation	Encryption / decryption compression Context for communication between layers
	LAYER 5	Session	Controls dialogue between computers Controls termination and restarts
	LAYER 4	Transport	Enables transfer of data TCP/UDP End to end connections
	LAYER 3	Network	Connects hosts on different networks IPV4 + IPV6 Routing of data packets
HARDWARE	LAYER 2	Data Link	Provides connections between hosts on the same network (Ethernet MAC addresses)
	LAYER 1	Physical	Electrical + physical specifications for devices Cables + Connectors Data in bits (1's and 0's)

AWS Shield

- ➔ An essential layer of security that (by default) blocks all layer 3/layer 4 attacks on your infrastructure
- ➔ Provides protection from SYN/UDP Floods (DoS) and Reflection attacks
- ➔ Two Tiers: Shield Standard and Shield Advanced
- ➔ Shield Advanced ($3,000/month)
    - ◆ Enabled 24/7 and is a premium DDoS protection tool
    - ◆ Can protect against more sophisticated attacks on EC2, ELB, Route 53, etc
    - ◆ 24/7 access to AWS DDoS response team (DRP)
    - ◆ Makes sure you don't incur costs for usage spikes during attacks

AWS Web Application Firewall

➔ AWS WAF is in charge of layer 7 – HTTP

- ✓ Deploys on Application Load balancers, API Gateway, CloudFront. Essentially any service that has it's traffic run through HTTP
- ✓ Allows you to define Web ACLs (Web Access Control List): granular control over IP Addresses, HTTP headers, HTTP body, URL st
- ✓ Usually is setup with AWS Shield to cover multi-layer security

AWS Firewall Manager

- Works at an AWS Organization level & controls security rules across all the accounts
- Rules are applied to all past, present, and future resources created in any account within the Organization
- Security Policy: the common set of rules that allows you to tie together VPC Security Groups, WAF rules, Shield Advanced, and AWS Network Firewall
- AWS Network Firewall: First line of defense for your VPC, AWS managed

AWS Key Management Service

- Deals with any encryption in AWS or any storage of encryption keys

Customer Managed Key

Created, managed, used by the customer and can be disabled at any point

AWS Managed Key

Created, managed, used on the customer's behalf by AWS

AWS Owned Keys

Exclusively controlled and only viewable by the AWS service that encrypts your data, not viewable to you

Cloud HSM (physical device)

Keys generated from your own custom hardware device, you manage the keys entirely

AWS GuardDuty

- ➔ Uses Machine Learning to detect anomalies in using 3rd party data and your infrastructure monitoring data
- ➔ Can set up AWS EventBridge to notify you of any findings
- ➔ Can protect against Crypto attacks (has a specific mode for it)

Example Inputs of data:

- ➔ CloudTrail Events Logs: unusual API calls, unauthorized deployments
- ➔ VPC Flow Logs: unusual internal traffic, unusual IP addresses
- ➔ DNS Logs: compromised EC2 instances sending embedded data within DNS queries
- ➔ Others: Lambda, S3 Data Events, etc

AWS Inspector & Detective

Inspector: Allows for automated security assessments of EC2 instances, container images, and lambda functions

- ➔ Can also do continuous scanning of the infrastructure only when needed for package vulnerabilities
- ➔ Attaches a risk score for prioritization of issues within the security, sends data to Event Bridge

Detective: analyzes data to determine root cause of security issue or suspicious activity (using ML and graphs)

- ➔ Automatically takes data from VPC flow logs, CloudTrail, GuardDuty
- ➔ Produces visualizations with context to help you trace the root issue

AWS Security Hub & Macie

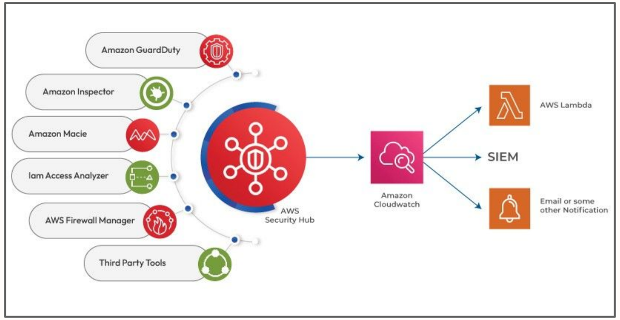
Security Hub

Central hub to manage security across several AWS accounts and automate security checks

Has integrated dashboards showing current security compliance status across all the services

Macie

Similar to GuardDuty and Detective, AWS Macie uses ML to discover and protect sensitive data stored in S3 buckets (data security)



AWS Certificate and Secrets Manager

- ➔ **Certificate Manager**: Used to easily provision, manage, and deploy (public & private) SSL/TLS certs
- ➔ Used to provide encryption for data in-transit or (in-flight) for websites (HTTPS)

- ➔ **Secrets Manager**: Automatically generates secrets (passwords) on rotation on a set schedule (uses lambda to rotate)
- ➔ Can be integrated with AWS RDS (Aurora, PostgreSQL, Oracle)

AWS Artifact and Config

- AWS Config: Records and audits your AWS infrastructure for compliance by recording configurations and changes to resources over time
- AWS Artifact: Portal that provides AWS customers with on-demand access to AWS compliance documentation and agreements
- Artifact Reports: allows you to download security & compliance documentation from 3rd party auditors. For instance: AWS ISO certs, Payment Card Industry (PCI), System and Organization Control (SOC) reports
- Artifact Agreements: allows you to track status of AWS agreements such as Business Associate Addendum (BAA) or Health Insurance Portability and Accountability Act (HIPAA) for an individual account

Practical Cloud Computing with AWS

Week Eleven - Global Architecture

UNIVERSITY OF MARYLAND

FEARLESSLY FORWARD

Overview of the Tools

- AWS Cloudformation
- AWS Beanstalk
- AWS CloudFront
- AWS Amplify
- AWS Global Accelerator
- AWS Outposts
- AWS SQS & SNS
- AWS Kinesis
- AWS Data Firehose

AWS CloudFront

- ➔ Taking the concept of caching, CloudFront improves data read performance by **caching** content at **edge locations**
- ➔ **100s** of Points of Presence (edge locations)
- ➔ Automatic connection to **AWS Shield & WAF**
- ➔ Popular backend links: ***S3 Storage***, Load Balancers, EC2 Instances, Custom HTTP site
- ➔ Great for static content that needs to be low latency for regions global
- ➔ **S3 Transfer Acceleration**: data to S3 can be faster through edge locations (uses CloudFront network distribution)
- ➔ Generally for images, videos, websites (reason for using S3)

AWS Global Accelerator

- ➔ Uses the AWS Global Network to improve application availability and performance
- ➔ Optimizes networking path to application (60% improvement)
- ➔ Similar to CloudFront, uses edge locations to make ***small efficient "jumps"***
- ➔ Unlike CloudFront however, it doesn't use caching
- ➔ Uses **TCP or UDP** for the underlying layer 4 protocol
- ➔ Good for HTTP use cases that require **static IP addresses or fast regional failover**

AWS Elastic Beanstalk

- Example of **Platform as a Service (PaaS) in AWS**
- Abstract the underlying infrastructure and only worry code, similar to AWS Lambda
- Have control of configuration of which type of cloud deployment you want
- **Three Architectural Models:**
  - Single Instance: good for development environments
  - Load Balancer + Auto Scaling Groups: great for production/pre-production web apps
  - Auto Scaling Groups: great for non-web apps in production
- Health Monitoring can be enabled in CloudWatch

AWS CloudFormation

- ➔ **A Structured document** of outlining the resources you want deployed in your infrastructure
- ➔ In a template, you can choose to specify EC2 instances, VPC policies, S3 buckets, etc
- ➔ Executed in top-bottom order exactly as written in the template configuration
- ➔ Bottomline: CloudFormation is an example of ***Infrastructure as a Code (IaaC)***
- ➔ Very similar to products such as Terraform and Pulumi
- ➔ Automation can be enabled to see certain stacks deployed on a schedule
- ➔ Resources created immediately from a configuration can be deleted from the CloudFormation all together at once
- ➔ **Infrastructure Composer** can be used to see all the resources and the relations between the components

AWS SQS & SNS

- **Simple Queue Service:** Fully managed (serverless) queue meant to decouple applications
  - Consumers (can be AWS Lambda, EC2, any compute) read messages in FIFO queue
  - Scaled Horizontally: from 1 message per second to 10,000s per second & Low latency (<10 ms on publish and receive)
  - Default retention of messages: 4 days, maximum of 14 days
  - No limit to how many messages can be in the queue

- **Simple Notification Service**:
  - “Event Publishers” send messages to one SNS topic
  - Each SNS topic has subscribers which will receive any new notification
  - Up to 12,500,000 subscriptions per topic, 100,000 topics limit

AWS Amplify

- ➔ **Essentially**: It is Amazon's version of Firebase
- ➔ It provides Backend as a Service (BaaS): a suite of services to handle all things backend
- ➔ For user authentication: AWS Amplify Cognito
- ➔ For databases: AWS AppSync with DynamoDB for Amplify
- ➔ Handles API procurement & **Hosting for web and mobile applications**
- ➔ Handles AI/ML, Monitoring, CI/CD, and anything else you would possible need

AWS Kinesis

- Ingest, buffer, and **process streaming data in real time** to derive insights in minutes
- Fully managed by AWS and runs on **serverless architecture**
- Handles streaming data from thousands of sources and processes it with low latency
- Video Streams:
  - Can ingest streaming data from millions of smartphones, security cameras, any other sensors
  - Easily links with ML video recognition apps in AWS to process data
- Data Streams:
  - Accumulates data from millions of application logs: clicks, touch sensors, in-app events
  - Combined with analytical tools, get results in seconds and can export results to S3

AWS Outposts

- Some companies maintain on-premises infrastructure along with cloud infrastructure (hybrid)
- **Outposts are “server racks”** that offer AWS infrastructure, services, APIs to build apps on premises just as you would in the cloud
- AWS will do the manual setup and manage the “racks”
- Essentially like having a mini piece of an AWS data center on premises
- However, **you are responsible for physical security** of the Outpost “rack”