# Section 3

## Five Characteristics of Cloud Computing

### Rapid Elasticity and Scalability

### Multi-tenancy and resource pooling

### On demand self service

### Broad network access

### Measured service

## Six Advantages of Cloud Computing

### Trade capital expense for operational expense

### Benefit from massive economies of scale

### Stop guessing capacity

### Increase speed and agility

### Stop spending money running and maintaining data centers

## Three Pricing Fundamentals of AWS Cloud (pay-as-you-go pricing)

### Compute

### Storage

### Data Transfer out of the AWS Cloud (data transfer in is free)

## How to choose an AWS Region

### Compliance

#### with data governance and legal requirements: data never leaves a region without your explicit permission

### Proximity

#### to customers with reduced latency

### Available services

#### within a region. some regions Don’t have the newest services and features

### Pricing

#### varies region to region and is transparent on the pricing page

## Availability Zones

### Each region has many availability zones

#### Usually 3 (min 2, max 6)

### Each availability zone is one or more discrete data centers with redundant power, networking and connectivity

### Each AZ is separate from each other so that they're isolated from disasters

### They're connected with high bandwidth, ultralow latency networking

## Points of Presence

#### 216 points of presences (covered later in the global section)

### Shared Responsibility

#### Customers - responsible for security IN the cloud

#### AWS - responsible for the security OF the cloud

#### diagram with a further breakdown - https://aws.amazon.com/compliance/shared-responsibility-model/

# Section 4 - IAM - Users and Groups

## Overview

## IAM = Identity Access Management (Global Service)

## Root Account created by default, shouldn't be used or shared

## Users are people within your organizational and can be grouped

## Groups only contain users, not other groups

## Users don't have to belong to group or can belong to multiple groups

## Permissions

### Users or Groups can be assigned JSON documents called policies

### These policies define the permissions of the users

### Least Privilege Principle: Don't give more permissions than a user needs

## Policies

### Policies can be attached at the group level applying to all users within the group

### Policies can be attached to a single user called 'inline' policies

### Premade Ones (such as...)

#### Allow user Access to all actions and resources

#### Allow user ReadOnly

### Can make custom one as well

## Policy Structure

### Consists of a Version, Id and Statement

### Statement

#### Sid: identifier for the statement (optional)

#### Effect: whether the statement allows or denies access (allow, deny)

#### Principal: Account/user/role to which this policy applies to

#### Action: list of actions this policy allows or denies

#### Resource: list of resources to which the actions applies to

#### Condition: conditions for when this policy is in effect (optional)

## Password Policy

### Can set a password policy for all users that may be created

### In AWS you can

#### Set a minimum password length

#### require specific character types

##### Uppercase letters, lowercase letters, numbers, non-alphabetical

#### Allow IAM users to change their own passwords

#### Requires user to change their passwords after some time (password expiration)

#### Prevent password re-use

## Multi Factor Authentication

### You want to protect your Root Accounts and IAM users with MFA

### MFA = password you know + security device one owns

### Main benefit is if a password is stolen or hacked, the account is not compromised

### MFA options in AWS

#### Virtual MFA device - Google Authenticator, Authy (Support for multiple tokens on a single device)

#### Universal 2nd Factor (U2f) security key - YubiKey (Support for multiple root and IAM users using a single security key) (physical key)

#### Hardware Key Fob MFA Device (Gemalto)

#### Hardware Key Fob MFA Device for AWS GovCloud (SurePass ID)

## How Can Users Access AWS

### AWS Management Console (protected by password + MFA)

### AWS Command Line Interface (CLI) (protected by access keys)

### AWS Software Development Kit (SDK) for code (protected by access keys)

#### Access Keys are secret just like a password

## AWS CLI

### Tool that enables you to interact with AWS services using commands in your command-line shell

### Direct access to the public APIs of AWS services

## AWS SDK

### Provides language specific APIs (set of libraries)

### Enables you to access and manage AWS services programmatically

### Embedded within your application

## AWS CloudShell

## IAM Roles for Services

### Some AWS services will need to perform actions on your behalf

### To do so, we will assign permissions to AWS services with IAM Roles

### Common Roles

#### EC2 Instance Roles

#### Lambda Function Roles

#### Roles for CloudFormation

### Can Create Roles specifically for AWS services within IAM with specified policies

#### Such that a service like EC2 can have Read Only Access in IAM to perform actions on its own

## IAM Security Tools

### IAM Credentials Report (account level)

#### A report that lists all your account's users and the status of their various credentials

### IAM Access Advisor (user-level)

#### Access advisor shows the service permissions granted to a user and when those services were last accessed

#### You can use this information to revise your policies

## IAM Best Practices

### Don't use the root account except for AWS account setup

### One physical user = One AWS user

### Assign users to groups and assign permissions to groups

### Create a strong password policy

### Use and enforce the use of MFA

### Create and use Roles for giving permissions to AWS services

### Use Access Keys for Programmatic Access (CLI / SDK)

### Audit Permissions of your account with the IAM Credentials Report

### Never share IAM users and Access Keys

## Shared Responsibility Model for IAM

### AWS

#### Infrastructure

#### Configuration

#### Compliance Validation

### You

#### Users, groups, roles, policies management and monitoring

#### Enable MFA on all accounts

#### Rotate all your keys often

#### Use IAM tools to apply appropriate permissions

#### Analyze Access Paterns and review permissions

# SECTION 5 - EC2 - Elastic Compute Cloud

## EC2 Basics

### One of the most popular services

### EC2 = Elastic Cloud Compute = Infrastructure as a Service

### Mainly consists in the capability of:

#### Renting virtual machines (EC2)

#### Storing data on virtual drives (EBS)

#### Distributing load across machine (ELB)

#### Scaling the services using an auto-scaling group (ASG)

## EC2 Sizing and Configuration options

### Operating Systems: Linux, Windows, or Mac OS

### How much...

#### Compute power and cores (CPU)

#### Random access memory (RAM)

#### Storage Space

##### Network Attached (EBS & EFS)

##### Hardware (EC2 Instance Store)

### Network Car; Speed of the card, public IP address

### Firewall rules: security group

### Bootstrap Script (configured at first launch): EC2 User Data

## EC2 User Data

### It is possible to bootstrap our instances using an EC2 User Data Script

### Bootstrapping means launching commands when a machine starts

### That script is only run once at the instance first start

### EC2 User Data is used to automate boot tasks such as

#### Installing updates, and software, downloading common files from the internet

### EC2 User Data Script runs through the root user

## EC2 Instance Types - Broad Chart with Instance Differences at Section 5.35 @ 4 minute mark

## EC2 Instance Types - Overview

### Has the following naming convention

#### m5.2xlarge

#### m: instance class

#### 5: generation

#### 2xlarge: size within the instance class

### General Purpose

#### Great for a diversity of workloads such as web servers or code repositories

#### Balanced between Compute - Memory - Network

#### Example is the t2.micro

### Compute Optimized (C class instances)

#### Great for compute-intensive tasks that require high performance processors

#### Batch processing workflows

#### Media transcoding

#### High performance web servers

#### High performance computing (HPC)

#### Scientific modeling & machine learning

#### Dedicated gaming servers

### Memory Optimized (r class instances)

#### Fast performance for workloads that process large data sets in memory

#### Use Cases

##### High Performance, relational/non-relational databases

##### Distributed web scale cache stores

##### In-memory databases optimized for BI

##### Applications performing real-time processing of big unstructured data

### Storage Optimized (I d or h1 class instances)

#### Great for storage intensive tasks that require high, sequential read and write access to large data sets on local storage

#### Use Cases

##### High frequency online transaction processing (OLTP) systems

##### Relational & NoSQL databases

##### Cache for in-memory databases

##### Data warehousing applications

##### Distributed file systems

## Security Groups and Classic Ports Overview

### Introduction

#### Security groups are the fundamental of network security in AWS

#### They control how traffic is allowed into or out of our EC2 Instances

#### Security group only contain allow rules which can reference by IP or by security group

#### They act as a firewall on EC2 instances

#### They regulate

##### Access to ports (i.e. only users using certain ports can make it through the security group)

##### Authorized IP ranges - IPv4 and IPv6

##### Control of inbound network (from other to the instance)

##### Control of outbound network (from the instance to other)

### Classic Ports to Know (for exam)

#### 22 = SSH (secure shell) - log into a Linux instance

#### 21 = FTP (File Transfer Protocol) - upload files into a file share

#### 22 = SFTP (Secure File Transfer Protocol) - upload files using SSH

#### 80 = HTTP - access unsecured websites

#### 443 = HTTP - access secured websites

#### 3389 = RDP (remote desktop protocol) - log into a windows instance

### EC2 Instances Purchasing Options

#### On-Demand Instances - short workload, predictable pricing, pay by second

#### Reserved (1 and 3 years)

##### Reserved Instances - long workloads

##### Convertible Reserved Instances - long workloads with flexible instances

#### Savings Plans (1 and 3 years) - commitment to an amount of usage, long workload

#### Spot instances - short workloads, cheap, can lose instances (less reliable)

#### Dedicated hosts - book an entire physical server, control instance placement

#### Dedicated instances - no other customers will share your hardware

#### Capacity Reservations - reserve capacity in a specific AZ for any duration

### EC2 On Demand

#### Pay for what you use

##### Linux or Windows - billing per second, after the first minute

##### All other OS - billing per hour

#### Has the highest cost but no upfront payment

#### No long-term commitment

#### Recommended for short-term and un-interrupted workloads, where you can't predict how the application will behave

### EC2 Reserved Instances

#### Up to 72% discount compared to On-demand

#### You reserve a specific instance attributes (instance type, region, tenancy, OS)

#### Specified Reservation Period, bigger discount for three-year reservation vs 1 year

#### Payment options - no upfront, partial upfront, all upfront

#### Reserved Instance's Scope - Regional or Zonal (reserve capacity in an AZ)

#### Recommended for steady-state usage applications (think database)

#### You can buy and sell in the Reserved Instance Marketplace

#### Convertible Reserved Instance

##### Can change the Ec2 Instance type and other things

##### Can still get a good discount on theses

### EC2 Savings Plans

#### Get a discount based on long term usage - up to 72%

#### Commit to a certain type of usage

#### Usage beyond EC2 savings plan is billed at the on-demand plan

#### Locked to a specific instance family and AWS region

#### Flexible across, instance size, OS, tenancy

### EC2 Spot Instances

#### Can get a discount up to 90% compared to On-demand

#### Instances that you can "lose" at any point of time if your max price is less than the current spot price

#### Useful for workloads that are resilient to failure

##### Batch jobs, data analysis, image processing, any distributed workloads, workloads with a flexible start and end time

#### **\*\*Not Suitable for Critical Jobs or Databases\*\*** - will be tested on this

### EC2 Dedicated Hosts

#### A physical server with EC2 instance capacity fully dedicated to your use

#### Allows you to address compliance requirements and use your existing server bound software licenses (per-socket, per-core, per-VM software licenses)

#### Purchasing Options

##### On-Demand - pay per second for active dedicated host

##### Reserved - 1 to 3 years (no upfront, partial upfront, all upfront)

#### The most expensive option

#### Useful for software that have complicated licensing model (BYOL - bring your own license)

#### Or for companies that have strong regulatory or compliance needs

#### (more specific than dedicated instances on hardware options)

### EC2 Dedicated Instances

#### Instances run on hardware that are dedicated to you

#### May share hardware with other instances in same account

#### No control over instance placement (can move hardware after Stop / Start)

### EC2 Capacity Reservations

#### Reserve On-Demand instances capacity in a specific AZ for any duration

#### You always have access to EC2 capacity when you need it

#### No time commitment (create/cancel anytime), no billing discounts

#### Combine with regional reserved instances and Savings Plans to benefit from billing discounts

#### You’re charged at On-Demand rate whether you run instances or not

#### Suitable for Short-term, uninterrupted workloads that needs to be in a specific AZ

## Shared Responsibility Model for EC2

### Amazon

#### Infrastructure (global network security)

#### Isolation on physical hosts

#### Replacing faulty hardware

#### Compliance Validation

### You

#### Security Group Rules

#### Operating-system patches and updates

#### Software and utilities installed on the EC2 instance

#### IAM Roles assigned to EC2 and IAM user access management

#### Data Security on your instance

# SECTION 6 - EC2 Instance Storage

## EBS Overview

### EBS Volume

#### An EBS (Elastic Block Store) Volume is a network drive you can attach to your instances while they run

#### It Allows your instances to persist data, even after their termination

#### They can only be mounted to one instance at a time (at the CCP level)

#### They are bound to a specific availability zone

#### Analogy: Think of them as a network USB stick

#### It’s a network drive (not a physical drive)

##### Uses the network to communicate the instance which means there might be a little bit of latency

##### It can be detached from an EC2 instance and attached to another one quickly

#### Its Locked to an Availability Zone

##### An EBS volume in us-east-la cannot be attached to us-east-lb

##### To move a volume across, you first need to snapshot it

#### Have a provisioned capacity (size in GBs, and IOPS)

##### You get billed for all the provisioned capacity

##### You can increase the capacity of the drive over time

### Delete on Termination Attribute

#### Controls the EBS behavior when an EC2 instance terminates

##### By default, the root EBS volume is deleted (attribute enabled)

##### By default, any other attached EBS volume is not deleted (attribute disabled)

#### This can be controlled by the AWS console / AWS CLI

#### USE CASE: preserve root volume when instance is terminated

### EBS Snapshots

#### Make a backup of your EBS volume at a point in time

#### Not necessary to detach volume to do snapshot, but recommended

#### Can copy snapshots across AZ or Region

#### Features

##### EBS Snapshot Archive

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

##### Recycle Bin for EBS snapshots

###### Setup rules to retain deleted snapshots so you can recover them after an accidental deletion

###### Specify retention from 1 day to 1 year

## AMI Overview

### Amazon Machine Image

### AMI are a customization of an EC2 instance

#### You add your own software, configuration, operating system, monitoring

#### Faster boot / config time because all your software is pre-packaged

### AMI are built for a specific region (can be copied across regions)

### You can launch EC2 instances from:

#### A public AMI: AWS provided

#### Your own AMI: you make and maintain them yourself

#### An AWS Marketplace AMI: an AMI someone else made (and potentially sells)

### AMI Process

#### Start an EC2 instance and customize it

#### Stop the instance

#### Build an AMI - this will also create EBS snapshots

#### Launch instances from other AMIs

## EC2 Image Builder

### Used to automate the creation of Virtual Machines or container images

### Automate the creation, maintain, validate, and test EC2 AMIs

### Can be run on a schedule (weekly, whenever packages are updates, etc)

### Free Service (only pay for underlying resources)

## EC2 Instance Store

### EBS volumes are network drives with good but "limited" performance

### If you need high-performance hardware disk, use EC2 Instance Store

### Better I/O performance

### EC2 Instance Store lose their storage if they're stopped (ephemeral)

### Good for buffer/cache/scratch data/temporary card (NOT good for long term storage)

### Risk of data loss if hardware fails

### Backups and replication are your responsibility

### Good for when an instance has very high IOPS (input output operations?)

## EFS - Elastic File System

### Managed NFS (network file system) that can be mounted on 100s of EC2s

### EFS works with Linux EC2 instances in multi-AZ

### Highly available, scalable, expensive, pay per use, no capacity planning

## EFS-IA EFS-(Infrequent Access)

### Storage class that is cost-optimized for files not accessed every day

### Up to 92% lower cost compared to EFS standard

### EFS will automatically move your files to EFS-IA based on the last time they were accessed

### Enable EFS-IA with a lifecycle policy

### Ex: Move files that are not accessed for 60 days to EFS-IA

### Transparent to the applications accessing EFS

## Shared Responsibility Model for EC2 Storage

### Amazon

#### Infrastructure

#### Replication for data for EBS volumes and EFS drives

#### Replacing faulty hardware

#### Ensuring their employees cannot access your data

### You

#### Setting up backup / snapshot procedures

#### Setting up data encryption

#### Responsibility of any data on the drives

#### Understanding the risk of using EC2 Instance Store

## Amazon FSx

### Overview

#### Launch 3rd party high-performance file systems on AWS

#### Fully managed service

#### FSx for...

##### Lustre

##### Windows File Server

##### NetApp ONTAPP

#### FSx for Windows File Server

##### fully managed, highly reliable, and scalable Windows native shared file system

##### Built on Windows File Server

##### Supports SMB protocol & Windows NTFS

##### Integrated with Microsoft Active Directory

##### Can be accessed from AWS or your on-premise infrastructure

#### FSx for Lustre

##### Fully managed, high-performance, scalable File Storage for high performance computing HPC

##### Name derived from "Linux" and "cluster"

##### Machine Learning, Analytics, Video processing, Financial modeling...

##### Scales up to 100s GB/s, millions of IOPs, sub-ms latencies

# SECTION 7 - ELB and ASG - Elastic Load Balancing and Auto Scaling Groups

## Scalability and High Availability

### Scalability means that an application / system can handle greater loads by adapting

### Two Kinds of Scalability

#### Horizontal (= elasticity)

#### Vertical

### Vertical Scalability

#### Means increasing the size of the instance

#### Ex: Scaling a t2.micro vertically to instead running a t2.large

#### Very common for non-distributed systems, such as a database

#### Usually a limit to how much you can vertically scale because of the hardware

### Horizontal Scalability

#### Means increasing the number of instances / systems for your application

#### Horizontal scaling implies distributed systems

#### This very common for web applications / modern applications

#### Super easy to scale on AWS

### High Availability

#### Usually goes hand in hand with horizontal scaling

#### Means running your application / system in at least 2 availability zones

##### If one AZ goes down, the instance in the other AZ is still available

#### Goal is to survive a data center loss (disaster)

### More Definitions

#### Scalability:

#### Ability to accommodate a larger load by making the hardware stronger (scale up) or by adding nodes (scale out)

#### Elasticity:

#### Once a system is scalable, elasticity means that there will be some "auto-scaling" so that the system can scale based on the load. This is "cloud-friendly": pay per use, match demand, optimize costs

#### Agility (a distractor on the exam) (means services are only a click away)

## Load Balancing (ELB)

### Load Balancers are servers that forward internet traffic to multiple servers (EC2 instances) downstream

### Why use one?

#### Spread load across multiple downstream instances

#### Expose a single point of access (DNS) to your application

#### Seamlessly handle failures of downstream instances

#### Do regular health checks to your instances

#### Provide SSL termination (HTTPS) for your websites

#### High availability across zones

### Why use an Elastic Load Balancer (ELB)

#### An ELB is a managed load balancer

##### AWS guarantees that it will be working

##### AWS takes care of upgrades, maintenance, high availability

##### AWS provides only a few configurations knobs

#### It costs less to setup your own load balancer but it will be a lot more effort on your end (maintenance, integrations)

#### 3 kinds of load balancers offered by AWS:

##### Application Load Balancer (HTTP / HTTPS only) - layer 7

##### Network Load Balancer (ultra-high performance, allows for TCP) - Layer 4

##### Classic Load Balancer (slowly retiring) - layer 4 and 7

## Auto Scaling Groups

### In real life, the load on your websites and application can change

### In the cloud, you can create and get rid of servers very quickly

### The goal of an Auto Scaling Group (ASG) is to:

#### Scale out (Add EC2 instances) to match an increased load

#### Scale in (remove EC2 instances) to match a decreased load

#### Ensure we have a minimum and a maximum number of machines running

#### Automatically register new instances to a load balancer

#### Replace unhealthy instances

### Huge cost savings, only running at an optimal capacity

### In AWS...

#### Minimum Size

#### Actual Size/Desired Capacity

#### Maximum Size

## Auto Scaling Group - Scaling Strategies

### Manual Scaling: update the size of an ASG manually

### Dynamic Scaling: Respond to changing demand

#### Simple / Step Scaling

##### When a CloudWatch alarm is triggered (ex CPU > 70% then add 2 units) and vice versa

#### TargetTracking Scaling

##### Example: I want the average ASG CPU to stay at around 40%

#### Scheduled Scaling

##### Anticipate a scaling based on known usage patterns at certain times

### Predictive Scaling

#### Use machine learning to predict future traffic ahead of time

#### Automatically provisions the right number of EC2 instances in advance

#### Useful when your load has predictable time-based patterns

# SECTION 8 - Amazon S3

## Overview

### One of the main building blocks of AWS

### Advertised as "infitely scaling" storage

## Use Cases

### Backup and Storage

### Disaster Recovery

### Archives

### Hybrid Cloud Storage

### Application hosting

### Media hosting

### Data lakes and big data analytics

### Software Delivery

### Static Websites

## Buckets

### S3 allows people to store objects (files) in "buckets" (directories)

### Buckets must have a globally unique name (across all regions and all accounts)

### Buckets are defined at the region level

### S3 looks like a global service but buckets are created in a region

### Naming convention:

#### No uppercase or underscore

#### 3-63 characters long

#### Not an ip

#### More conventions on S8.73 at 2:50 mark

## Objects

### Objects (files) have a key

### The key is the full path to that file

### Key is composed of a prefix + object name

#### Where prefix is the path and the object name is the file name

### Object values are the content of the body

#### Max object size is 5TB (5000GB)

#### If uploading more than 5GB, you must use "multi-part upload"

### Metadata: list of text key /value pairs - system or user metadata)

### Tags (Unicode key / value pair - up to 10) - useful for security/lifecycle

### Version Id

## S3 Security

### User-based: IAM policies - which API calls should be allowed for a specific user from IAM

### Resource-based:

#### bucket wide rules assigned from the S3 console which allows cross account

#### Object Access Control List (ACL) - finer grain

#### Bucket Access Control List (ACL) - less common

### Encryption - server-side bucket encryption

## S3 Bucket Policies

### JSON Based Policies

### Use S3 bucket for policy to:

#### Grant public access to the bucket

#### Force objects to be encrypted at upload

## S3 Versioning

### You can version your files in S3

### It is enabled at the bucket level

### Same key overwrite will change the "version", 1,2,3...

### It is best practice to version your buckets

#### Protect against unintended deletes

#### Easy roll back to previous version

## S3 Replication (CRR and SRR)

### Must enable versioning in source and destination buckets

### Cross-Region Replication (CRR)

### Same-Region Replication (SRR)

### Buckets can be in different AWS accounts

### Copying is asynchronous

### Must give proper IAM permissions to S3

### Use Cases:

#### CRR - Compliance, lower latency access, replication across accounts

#### SRR - log aggregation, live replication between production and test accounts

## S3 Storage Classes

### S3 Standard - General Purpose

#### 99.99 percent availability

#### Used for frequently accessed data

#### Low latency and high throughput

#### Sustain 2 concurrent facility failures

#### Use Cases: big data and analytics, mobile and gaming apps, content distribution

### S3 Standard Infrequent Access

#### For data that is accessed less frequently but requires rapid access when needed

#### lower cost than standard

#### 99.9% available

#### Use cases: Disaster recovery, backups

### S3 One Zone-Infrequent Access

#### Same as Standard-AI

#### But higher durability in a single AZ but data is lost if AZ is destroyed

#### 99.5 available

#### Use Cases: Storing secondary backup copies of on-premise data, or data you can recreate

### S3 Glacier Instant Retrieval

#### Low-cost object storage means for archiving /backup (all of glacier)

#### Pricing: price for storage + object retrieval cost (all of glacier)

#### Millisecond retrieval for data accessed once a quarter

#### Minimum storage duration of 90 days

### S3 Glacier Flexible Retrieval

#### Retrieval times: expedited (1 - 5 minutes), standard (3 to 5 hours) Bulk (5 to 12 hours) - free

#### Minimum storage duration of 90 days

### S3 Glacier Deep Archive

#### Timing: Standard (12 hours), bulk (48 hours)

#### Minimum storage duration of 180 days

#### least expensive but for long term storage

### S3 Intelligent Tiering

#### Small monthly monitoring and auto-tiering fee

#### Moves objects automatically between access tiers based on usage

#### No retrieval charges in inteli tier

### Can move between classes manually or using S3 LifeCycle configs

### Durability

#### High Durability - means you barely lose objects at all

#### On S3 if you store 10,000,000 objects with S3, you can on avg expect to incur a loss of a single object once every 10,000 years

#### Durability is the same for all storage classes

### Availability

#### Measures how readily available a service is

#### Varies depending on storage class

## S3 Encryption

### Three Options

#### No encryption

#### Server-Side Encryption

##### Server encrypts the file after receiving it

#### Client-Side Encryption

##### Client encrypts the file before uploading it

## Shared Responsibility Model for S3

### Amazon - Infrastructure, Configuration and vulnerability analysis, compliance validation

### Customer - S3 Versioning, S3 Bucket policies, S3 Replication setup, logging and monitoring, s3 storage class choice, Data encryption at rest and in transition

## AWS Snow Family

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

### Sometimes transferring data over the network can take way too long if there is too much data to be moved and the network is too small (i.e. small bandwidth, high network costs or network stability)

### If it takes more than a week to transfer over the network, it is preferable to use hardware within the Snow family to physically transfer the data.

### Snowball Edge (data transfers)

#### physical box: moves TBs or Pbs of data in or out of AWS

#### Pay per data transfer job

#### use cases: large data cloud migrations, DC decommission, disaster recovery

### Snowcone

#### Much smaller physical device, very light

#### 8 Terabytes of useable storage, use snowcones where snowballs don't fit

#### Must provide your own battery and cables

### Snowmobile

#### An actual truck that will transfer exabytes of data (1 Eb - 1,000 PB - 1,000,000 TBs)

#### Each snowmobile has 100PB worth of capacity - so to transfer an EB you need 10 snowmobiles

#### High Security, Temperature controlled, GPS, 24/7 surveillance

#### better than snowball if transferring more than 10Pb of data

### Use AWS OpsHub to manage Snow Family Devices

## AWS Storage Gateway - Cloud Native Options

### S3 is a proprietary technology so to use it on-premise for a hybrid cloud solution you need AWS Storage Gateway

### Bridges between on-premise data and cloud data in S3

### Hybrid storage service to allow on-premises to seamlessly use the AWS cloud

### Use cases: disaster recovery, backup and restore, tiered storage

# Section 9 - Databases and Analytics

## Intro

### Storing data on disk (EFS, EBS, EC2 Instance Store, S3) can have its limits

### sometimes, you want to store data in a database...

#### Data can be structured

#### Build indexes to efficiently query the data

#### Define relations between datasets

## Relational Databases

### OLTP

## NoSQL - Non-Relational Databases

### Benefits...

#### Flexibility

#### Scalability: designed to scale-out by using distributed clusters

#### High-performance: optimized for a specific data model

#### Highly-functional

### NoSQL Data Example: JSON

## Shared Responsibility for Databases on AWS

### AWS offers use to manage diff databases

### AWS Benefits include:

#### Quick provisioning, high availability, vertical and horizontal scaling

#### Automated backup, restore, operations, upgrades

#### Operating system patching

#### Monitoring and Alerting

### Many databases could be run on EC2 but you must handle yourself the resiliency, backups, patching and fault tolerance

#### Whereas with Managed Databases, AWS takes care of these things for you

## AWS RDS Overview

### RDS (Relational Database Service)

### Managed DB service for DB using SQL

### Allows you to create databases in the cloud that are managed by AWS

#### Postgres, MySQL, MariaDB, Oracle, Microsoft SQL server, Aurora (AWS proprietary DB)

### Benefits over EC2

#### RDS is a managed service

##### Autonomous provisioning and OS patching

##### Continuous backups and restores

##### Monitoring dashboards

##### Read replicas for improved read performance

##### maintenance windows for upgrades

##### Scaling capability

##### Storage backed by EBS

#### BUT you can't SSH into your instances

### Amazon Aurora

#### PostgreSQL and MySQL are both supported as Aurora DB

#### Aurora is "AWS cloud optimized: and claims 5x performance improvement over MySQL on RDS and Over 3x performance improvement over PostgreSQL on RDS

#### Aurora storage automatically grows in increments of 10GB up to 64 TB.

#### Aurora costs more than RDS (20% more) but is more efficient

#### Not included in free tier

## RDS Deployment Options

### Read Replicas

#### Scale the read workload of your DB

#### Created Replicas of the RDs that are copies dedicated to being read from by the application

#### Can create up to 5 read replicas

#### All writes are done to the main RDS

#### Used to scale Read operations (improves scalability)

### Multi-AZ

#### Failover in case of AZ outage (increases availability)

#### Replication across to a different AZ

#### If main RDS fails, AWS triggers a failover to the replica DB in a diff AZ

### Multi-Region

#### Same type of failover but instead it replicates to a different region

#### Disaster recovery strategy in case of regional issues

## Amazon ElastiCache Overview

#### Managed Database service for Redis or MemCached

#### Caches are in-memory databases with high-performance, low latency

#### Helps reduce load of databases for read intensive workloads

#### I think it’s generally used on top of an RDS, so it can take pressure off the RDS for intensive read purposes while the EC2 uses the RDB for slower reads and writes

## DynamoDB Overview

### Overview

#### Fully managed highly available with replication across 3 AZs

#### NoSQL database

#### Scales to massive workloads, distributed "serverless" database

#### Millions of requests per seconds, trillions of rows, 100 TBs of storage

#### Fast and consistent in performance

#### Single digit millisecond latency

#### Integrated with IAM for security, auth, and administration

#### Low cost and autoscaling capabilities

#### Standard and Infrequent Access Table class,

### Type of Data

#### DynamoDB is a key/value database

### DynamoDB Accelerator - DAX

#### Fully managed in-memory cache for DynamoDB

#### 10x performance improvement - single digit millisecond to microseconds latency when accessing DynamoDB tables

#### Secure, highly scalable, highly available

#### Difference with ElastiCache is at the CCP level

##### DAX is only used for and with DynamoDB

##### Whereas ElastiCache can be used with many other DBs

### Global Tables

#### Make a DynamoDB table accessible with low latency in multiple-regions

### Active-Active replication (read/write to any AWS region)

## RedShift Overview

### Is based on PostgreSQL but it’s not used for OLTP

### Its OLAP - Online analytical processing (analytics and data warehousing)

### Load data once every hour, not every second

### 10x better performance than other data warehouses, scale to PBs of data

### Columnar storage of data (instead of row based)

### Massively parallel query execution (MPP), highly available

### Pay as you go based on the instances provisioned

### Has a SQL interface for performing the queries

### BI tools such as AWS quicksight or Tableau integrate with it

## Amazon EMR

### EMR stands for "Elastic MapReduce:

### EMR helps creating Hadoop clusters (big data) to analyze and process vast amount of data

### The clusters can be made of hundreds of EC2 instances

### Also supports Apache Spark, HBAse, Presto, Flink

### EMR takes care of all the provisioning and configuration

### Auto-scaling and integrated with Spot instances

### Use cases: data processing, machine learning, web indexing, big data

## Amazon Athena

### Serverless query service to perform analytics against S3 objects

### Uses standard SQL language to query the files

### Supports CSV, JSON, ORC, AVRO, and Parquet (built on presto)

### Pricing: $5 per TB of Data scanned

### Use compressed or columnar data for cast-savings (less scan)

### Use Cases: BI / Analytics, Reporting, analyze & query VPC Flow logs, ELB lobs, CloudTrail trails, etc...

### EXAM TIP: Analyze data in S3 using serverless SQL, use Athena

## Amazon QuickSight

### Serverless machine learning-powered BI service to create interactive dashboards

### Fast, auto scalable, embeddable, with per-session pricing

### Use cases: Business Analytics, building visualizations, perform ad-hoc analysis, get business insights using data

### Integrate with RDS, Aurora, Athena, Redshift, S3...

## DocumentDB

### Aurora is an "AWS-implementation" of PostgreSQL / MySQl

#### as

### DocumentDB is an "AWS-implementation" for MongoDB (which is NoSQL)

#### Used to store, query, and index JSON data

### Similar deployment as Aurora

### Fully managed, highly available with replication across 2 AZs

### DocumentDB storage automatically grows in increments of 10GB, up to 64 TB

### Automatically scales to workloads with millions of requests per second

## Amazon Neptune

### Fully managed graph database

### A popular graph dataset would be a social network

### Highly available across 3 AZs with up to 15 read replicas

### Build and run applications working with highly connected datasets - optimized for these complex and hard queries

### Can store up to billions of relations and query the graph with milliseconds latency

### Great for knowledge graphs (Wikipedia), fraud detection, recommendation engines, social networking

## QLDB

### Stands for Quantum Ledger Database

### Ledger is a book for recording financial transactions

### Fully managed, serverless, high available, replication across 3 AZs

### Used to review history of all the changes made toy your application data over time

### Immutable system: no entry can be removed or modified, cryptographically verifiable

### 2-3x better performance than common ledger blockchain frameworks, manipulate data using SQL

### Difference with Amazon Managed Blockchain, no decentralization component, in accordance with financial regulation rules

## Amazon Managed Blockchain

### Blockchain makes it possible to build applications where multiple parties can execute transactions without the need for ae trusted, central authority

### Managed Blockchain is a managed service to...

#### Join public blockchain networks

#### Create your own scalable private network

### Compatible with frameworks Hyperledger Fabric and Ethereum

## AWS Glue

### Managed ETL service (extract, transform, load)

### Useful to perform and transform data for analytics

### Fully serverless service

### Example:

#### Can extract data from an S3 Bucket and RDS, transform the data, and then Load it into Redshift

### Glue Data Catalog

#### Catalog of datasets

#### Can be used by Athena, Redshift, EMR

## DMS - Database Migration Service

### EC2 runs the DMS

### DMS quickly and securely migrates databases from a source DB to a target DB

### resilient and self-healing

### Supports:

#### Homogeneous migrations (Ex: oracle to oracle)

#### Heterogenous migrations (Ex: SQL server to Aurora)

## SUMMARY

### Relational Databases - OLTP: RDS & Aurora (SQL)

### Differences between Multi-AZ, Read Replicas, Multi-Region

### In-memory Database: ElastiCache

### Key/Value Database: DynamoDB (serverless) & DAX (cache for DynamoDB)

### Warehouse - OLAP: Redshift (SQL)

### Hadoop Cluster: EMR

### Athena: query data on Amazon S3 (serverless and SQL)

### Quicksight: dashboards on your data

### Managed Blockchains: QLDB, Amazon Managed Blockchain (AMB)

### DocumentDB: Amazon's JSON - NoSQL Database

### Amazon QLDB: Financial Transactions Ledger (immutable journal, cryptographically verifiable) (centralized)

### Amazon Managed Blockchain: managed Hyperledger fabric and Ethereum blockchains

### Glue: Managed ETL and Data Catalog service

### Database Migrations: DMS

### Neptune: graph database

# Section 10 - Other Compute Services

## Docker Overview

### 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 machines, predictable behavior, less work, no compatibility issues

### Docker Images are stored in Docker Repositories

## 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 - Elastic Container Service

### Used to launch Docker containers on AWS

### You must provision and maintain the infrastructure (the EC2 instances)

### AWS takes care of starting and stopping the containers

### Has integrations with the Application Load Balancer

## Fargate

### Launch Docker container on AWS

### However, you don't 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

## Serverless Introduction

### Serverless is a new paradigm in which the developers don't have to manage servers anymore...

### They just deploy code / functions

### Initially serverless was 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 just means that you don't manage, provision, see them

### Examples: S3, DynamoDB, Fargate, Lambda

#### All of these we didn't provision any servers, they ran and scale as you use them.

## AWS Lambda

### Overview

#### Virtual Functions - no servers to manage!

#### Limited by time - short executions

#### Run on demand

#### Scaling is automated

### Benefits

#### 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 RAM1)

#### Increasing RAM will also improve CPU and network!

### Supports

#### Node.js (JavaScript)

#### Python

#### Java

#### C# (.NET core & PowerShell)

#### Lambda Container Images

##### Although container image must implement the Lambda Runtime API

#### Few others...

### Pricing

#### Pay per calls

##### First million requests are free, $.20 per million requests thereafter

#### Pay per duration

##### 400,000 GB seconds of compute time per month is free

##### after that $1 per 600,000 GB-seconds

#### Generally, very cheap to run AWS Lambda

## Amazon API GateWay

#### Fully managed service for developers to easily create, publish, maintain, monitor and secure APIs

#### Serverless and scalable

#### Supports RESTful APIs and WebSocket APIs

#### Expose Lambda functions as HTTP API

## 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 (as 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

## 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 and 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

##### Websites

##### Dev/test Environment

#### Has high availability but no auto-scaling, limited AWS integrations

# SECTION 11 - Deployments & Managing Infrastructure at Scale

## CloudFormation

### Declarative way of outlining your AWS Infrastructure for any resources

### Example:

#### I want a security group, two EC2 instances using this security group, an S3 bucket, and I want a load balancer in front of these machines

#### Then CloudFormation creates those for you, in the right order, with the exact configuration that you specify

### CloudFormation Templates

#### The declaration of the AWS resources that make up a stack

### Benefits of AWS CloudFormation

#### Infrastructure as code

##### No resources are manually created, which is excellent for control

##### Changes to the infrastructure are reviewed through code

#### Cost

##### each resource within the stack is tagged with an identifier so you can see the pricing involved

##### you can estimate the cost of resources using the CloudFormation template

##### savings strategy: In Dev, you could automate the deletion of templates at 5pm and recreated at 8am, safely

#### Productivity

##### Ability to destroy and re-create an infrastructure on the cloud on the fly

##### Automated generated of diagram for your templates

##### Declarative programming

### **Exam Prep**

#### Use when your recreating cloud infrastructure

## AWS Cloud Development Kit

### Define you cloud infrastructure using a familiar language

#### JavaScript, python, java, .NET

### The code is 'compiled' into a CloudFormation template

### You can therefore deploy infrastructure and application runtime code together

#### Great for Lambda functions

#### great for Docker containers in ECS / EKS

## Beanstalk Overview

### Elastic Beanstalk is a developer centric view of deploying an application on AWS

### It uses all the components we've seen before (ec2, asg, elb, rds...)

### But it's all in one view that’s easy to manage while still having full control of configuration

### Beanstalk = Platform as a Service

### Managed service

#### Install configuration / OS is handled by beanstalk

### Responsibility is only the application code for the developer

### Three architecture models

#### Single instance deployment: good for dev

#### LB + ASG: great for production or pre-prod web apps

#### ASG only: great for non-web apps in production

### Health Monitoring

#### Health agents pushes metrics to CloudWatch, checks for app health and publishes health events

## AWS CodeDeploy

### Used when you want to deploy your application automatically

#### Used to upgrade instances and servers from v1 to v2?

### Works with Ec2 instances or On-Premise / Hybrid servers

### But servers and 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

### CodeCommit is AWS's competing product with GitHub

### CodeCommit...

#### Git based source control service

#### Allows collaboration

#### Automatic versioning

#### private, secured and integrated with AWS

## AWS CodeBuild

### Code building server in the cloud

### Compiles source code, run tests, and produces packages that are ready to be deployed

### Pay as you go pricing - only pay for build time

## AWS CodePipeline

### Orchestrates the different steps to have the code auto pushed to production

#### Code -> build -> test -> Provision -> deploy

### AWS's basis for CICD

## AWS CodeArtifact

### A service for artifact management which is the process of storing and retrieving software package dependencies

### Developers and CodeBuild can retrieve dependencies straight from CodeArtifact just like NuGet

## AWS CodeStar

### Unified UI to easily managed software development activities in one place

### Quick way to correctly setup CodeCommit, CodePipeline, CodeBuild and so on

## AWS Cloud9

### AWS's cloud based IDE for writing code

### Used within a web browser

## AWS System 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

## AWS OpsWorks

### Chef & Puppet help perform server configuration automatically, or repetitive actions

### AWS OpsWorks = Managed Chef & Puppet

### It’s an alternative to AWS SSM

# Section 12: Global Infrastructure

## Overview

### A global application is an application deployed in multiple geographies

#### In AWS this could be Regions and or Edge Locations

### Aims to Decrease Latency

#### Deploy application closer to global users

### Helps with Disaster Recovery

#### If an AWS region goes down you can failover to another region

### Attack protection

#### distributed global infrastructure is harder to attack

## AWS 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

### Routing Policies

#### Simple Routing Policy

##### No health checks

#### Weighted Routing Policy

##### DNS routes percentage of traffic respective to the weights assigned with each server

#### Latency Routing Policy

##### Looks at where the user is located and then redirects the user's work to the closest server

#### Failover Routing Policy

##### Route53 redirects the user to a failover server if the primary server fails

## AWS CloudFront

### Overview

#### AWS's Content Delivery Network (CDN)

#### Improves read performance, content is cached at the edge, improving the user's experience

#### 216 Points of Presence globally (edge locations)

#### DDoS protection

### Origins

#### S3 Bucket

##### For distributing files and caching them at the edge

##### Enhanced security with CloudFront Origin Access Control (OAC)

#### Custom Origin (HTTP)

##### Application Load Balancer

##### EC2 Instance

##### ...

## AWS S3 Transfer Acceleration

### Overview

#### Increase transfer speed by transferring file to an AWS edge location which will

#### forward the data to the s3 bucket in the target location

## 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

### If trying to access an application in India through AWS Global Accelerator

#### Your America server will direct to AWS's nearest Edge Location, enter the private AWS network

#### and use their private network infrastructure to reach the India server

## Global Accelerator Vs CloudFront

### Both serves use AWS global private network and its edge locations as well as provide DDoS protection

### CloudFront

#### Improve performance for cacheable content (such as images and videos)

#### Content is served at the edge

### Global Accelerator

#### No caching, proxying packets at 3the edge to application running in one or more AWS regions

## AWS Outposts

### Overview

#### 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 "Outpost Racks" within your on-premise infrastructure and you can start leveraging AWS services on premises

#### But you on are responsible for the physical security of the racks

### Benefits

#### Low latency access to on-premise systems

#### Local data processing

#### Data residency

#### Fully managed service

## 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 apps thru 5G networks

### Traffic doesn't leave the Communication Service Providers (CSP) network

### Use cases: smart cities, connected vehicles, ML-assisted diagnostics, 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"

### Example: AWS Region: N.Virginia(us-east-1)

#### AWS Local Zones: Boston, Chicago, Dallas, Houston, Miami

## AWS Global Applications Architecture

# SECTION 13: Cloud Integrations

## Amazon SQS - Simple Queue Service

### Overview

#### Allows a service sends work to a queue instead of it going directly to another service

#### Decouples the services and allows the services to scale separately

#### Fully managed service (serverless)

#### **\*\*Used to Decouple applications\*\***

#### Scales from 1 message per second to 10,000s per second

#### Default retention of messages: 4 days, maximum of 14days

##### Messages (unit of work) needs to be processed within this time (time allowed in queue)

##### No limit to how many messages can be in the queue though

## Amazon Kinesis

### Overview

#### Kinesis = real-time big data streaming

#### Managed service to collect, process and analyze real-time streaming at any scale

## Amazon SNS - Simple Notification Server

### Overview

#### Used to send a message from one service to multiple different services

##### Decouples a service from multiple services

#### Using a pub/sub, publisher/subscriber system

##### 'event publishers' send the message to one SNS topic

##### Many 'event subscribers' listen to the SNS topic notification

##### Each subscriber to the topic will get all the messages

## Amazon MQ

### Managed message broker service for RabbitMQ and ActiveMQ

### Used when migrating on-premise to the cloud, instead of re-engineering the application's open protocols to use SQS or SNS

# Section 14 - Cloud Monitoring

## CloudWatch

### Overview

#### CloudWatch provides metrics for every service in AWS

#### Metrics examples: Ec2- CPU utilization, status checks, Network (not ram)

###### EBS volumes: disk read/writes

###### S3 buckets

###### Billing

###### Service Limits

###### Custom Metrics

### Alarms

#### Used to trigger notifications for any metric

### CloudWatch Logs

#### Used to collect log files from applications (logs of actions performed)

##### Beanstalk, ECS, Lambda

#### Enable real-time monitoring of logs

#### CloudWatch Logs for EC2

##### Need a CloudWatch agent on EC2 to collect log files, EC2 doesn't do it by default

##### Can be set up on AWS or On-Premise

## Amazon EventBridge

### Allows you to react to events happening in AWS accounts

### Example

#### Cron Jobs (scheduled scripts)

#### Event Patterns

##### Event rules to react to a service doing something

#### Security

##### Email Security Team whenever someone logs into the root user

## AWS CloudTrail

### Provides governance, compliance, and audit for your AWS Account

#### CloudTrail is enabled by default

### Gets a 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

### Example: a resource is deleted in AWS, one can investigate it through CloudTrail

## AWS X-Ray

### Used for getting a visual analysis of your applications / integrated services

### Helps troubleshoot performance

### Understand dependencies in a microservice arch

### Pinpoint service issues

### Review request behavior

### Find errors and exceptions

### Identify affected users

## 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

##### Identify critical issues, security vulnerabilities

##### Finds common coding best practices, resource leaks, security detection

#### CodeGuru Profiler: visibility/recommendation about application performance during runtime(production)

##### Helps understand the runtime behavior of your application

##### Identify and remove code inefficiencies, decrease compute costs...

## Service Help Dashboard

### Shows health of all regions and all services

### Shows historical info for each day

### Has an RSS feed so you can subscribe to statuses

## Personal Health Dashboard

### provides alerts and remediation guidance when AWS is experiencing events that may impact you

### displays relevant and timely info

# Section 15: VPC & Networking

## VPC - Virtual Private Cloud

### Overview

#### Private network to deploy your resources

### Subnets

#### Allow you to partition your network inside your VPC

#### 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

#### Use Route Tables to define access to the internet and between subnets

### 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

##### allow your instances in your private subnets to access the internet while remaining private

### Network ACL & Security Groups

#### NACL

##### A firewall which controls traffic from and to subnet

##### Can have allow or deny rules

##### Are attached at the subnet level

##### Rules only include IP addresses

##### Stateless

#### 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

##### Stateful

### VPC Flow Logs

#### Capture information about IP traffic going into your interfaces

#### Helps to monitor and troubleshoot connectivity issues

#### Capture network information from AWS managed interfaces too

### 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

### Site to Site VPN & Direct Connect

#### Site to Site VPN

##### Use to connect an on-premises VPN to AWS

##### The connection is automatically encrypted

##### Goes over the public network

##### On-premises side:

###### must use a Customer Gateway (CGW)

##### AWS side:

###### must use a Virtual Private Gateway (VGW)

#### 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

### AWS Client VPN

#### Use OpenVPN to connect to your private network in AWS from your computer

#### Allows to 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

### 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

# Section 16: Security and Compliance

## Shared Responsibility Model

### AWS responsibility

#### Security of the Cloud

##### Protecting the infrastructure that runs the services

#### Managing services

### Customer Responsibility

#### Security in the cloud

##### meaning customer is responsible for management of the guest OS, firewall, network config, IAM

#### Encrypting application data

### Shared controls

#### patch management, configuration management, awareness & training

## DDoS Protection

### DDoS attacks

#### "Distributed Denial of Service"

#### An attacker points their servers to direct many bots to overload the target servers with tons of work

#### preventing normal users from receiving responsive functionality from the server effectively downing the server

### AWS Shield Protection:

#### protects against DDoS attacks for your website and apps, for all customers at no additional cost

#### Provides protection from attacks such as SYN/UDP Floods, reflection attacks and other layer 3/layer 4 attacks

### AWS Shield Advanced:

#### 24/7 premium DDoS protection

#### Protects against more sophisticated attacks

#### Protects against higher fees during usage spikes due to DDoS

#### 24/7 access to an AWS help team

### AWS WAF: (Web application firewall)

#### Filter specific requests based on rules

#### Protects web applications from common web exploits (layer 7)

#### Deploy on Application Load Balancer, API Gateway, CloudFront

#### Define Web ACL

##### Rules can include IP addresses, Http headers, body, or URI strings

##### Protects from common attacks - SQL injection, cross-site scripting (XSS)

##### Block certain countries with Geo-match

### CloudFront and Route53

#### Availability protection using global edge network

#### Combined with AWS Shield, provides attack mitigation at the edge

### Be read to scale - leverage AWS Auto Scaling

## Penetration Testing

### Overview

#### A test where you attack your own infrastructure to test the security

#### Customers are welcome to carry out these tests without prior approval for 8 services

#### There are prohibited activities though for testing

##### DNS zone walking

##### DDoS

##### Port Flooding

##### ...

#### Need to get approval for these

## Encryption

### Types of Encryption

#### Data At Rest

##### data stored or archived on a device

#### Data In Transit

##### (in motion) data being moved from one location to another

#### Need to leverage encryption keys to encrypt both of these

### AWS KMS (key management service)

#### KMS manages the encryption keys for us and enables encryption for many other services

#### AWS manages the software for encryption

### CloudHSM

#### AWS provisions encryption hardware but the customer manages the encryption keys entirely

### Types of CMKs (Customer Master Keys)

#### Customer Managed CMK

##### Customer created, managed and used keys

#### AWS managed CMK

##### created, managed, and used by AWS on the customer's behalf

#### AWS owned CMK

##### collection of CMKs that an AWS service owns and manages to use in multiple accounts

#### CloudHSM Keys

##### Key generated from your our CloudHSM hardware device

## AWS Certificate Manager

### Lets you easily provision, manage and deploy SSL/TLS certificates

### Used to provide in-flight encryption for website (HTTPS)

## AWS Secrets Manager

### 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

## AWS Artifact

### Portal that provides customers with on-demand access to AWS compliance documentation and AWS agreements

### Artifact Reports

#### Find security and compliance documents from third-part auditors

### Artifact Agreements

#### Allows you to review, accept, and track status of AWS agreements such as BAA or HIPAA

## Amazon GuardDuty

### Intelligent Threat discovery to protect AWS account

### Uses ML algorithms, anomaly detection, 3rd party data

### One click to enable

### Can protect against Cryptocurrency Attacks

## Amazon Inspector

### Automated Security Assessments

### Only for...

#### Used with EC2 instances

#### Used with Container Images push to Amazon ECR

#### For Lambda Function

##### Assessments are done as they are deployed

### Reports it's finding with AWS Security Hub and Amazon Event Bridge

## AWS Config

### Helps with auditing and recording compliance of your AWS resources

### Help record configurations and changes over time

### Possibility of storing the config data into S3

## AWS Macie

### Fully managed data security and data privacy service that use ML and pattern matching to discover and protect sensitive data in AWS

### Macie helps identify and alter 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 format from various AWS services and AWS partner tools

## Amazon Detective

### Analyzes, investigates and quickly identifies the root cause of security issues or suspicious activity

#### 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

#### Spam - receiving undesired emails from AWS-owned IP addresses, website, forums

#### Port scanning

#### DDoS attacks coming from AWS servers

#### Intrusion attempts

#### Hosting objectionable or copyrighted content

#### Distributing malware - can't host malware on AWS

## Root User Privileges

### Lock away your AWS account root user access keys

### Do not use the root account for everyday tasks

### Actions that can be performed only by the root user:

#### Change account settings (account name, email...)

#### Close AWS account

#### Restore IAM user perms

#### Change or cancel your AWS support plan

#### Register as a seller in the Reserved Instance Marketplace

#### Configure an S3 bucket to use MFA

# Section 17: Machine Learning

## 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

### Other use cases: Labeling, content moderation

## Amazon Transcribe

### Automatically convert speech to text

### uses a deep learning process called automatic speech recognition (ASR) to convert speech to text quickly

### 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

## Polly

### Turn text into lifelike speech using deep learning

### Allows you to create applications that talk

## Translate

### Natural and accurate language translation

### Allows you to localize content - such as websites and applications for international users, and to easily translate large volumes of text efficiently

## Lex & Connect

### Lex - the same tech that powers Alexa

#### Automatic Speech Recognition to convert speech to text

#### Natural Language Understanding to recognize the intent of text and callers

#### Helps build chatbots, call center bots

### Connect

#### Receive calls, create contact flows, cloud based virtual contact center

#### can integrate with other CRM systems or AWS

#### No upfront payments, 80% cheaper than other traditional systems

## Comprehend

### Used for NLP - Natural Language Processing

### 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

#### Analyses text using tokenization and parts of speech

#### Automatically organizes a collection of text files by topic

### 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(?)

## SageMaker

### Fully managed service for developers / data scientists to build ML models

### Typically, difficult to do all the processes in one place + you need to provision the servers

### Machine learning process (simplified)

## Forecast

### Fully managed service that uses ML to deliver highly accurate forecasts

### Predict the future sales of a raincoat

### 50% more accurate than looking at the data itself

### Reduce forecasting time from month to hours

### Use case: Product demand planning, financial planning, resource planning

## Kendra

### Fully managed document service powered by machine learning

### Extract answers from within a document (text, pdf, HTML, PowerPoint, MS Word, FAQs)

### Natural Language search capabilities

### Ability to manually fine-tune search results

### Learn from interactions/feedback to promote preferred results (incremental learning)

## Personalize

### Fully managed ML-service to build apps with real-time personalized recommendations

### Personalized product recommendations/re-ranking, customized reactive marketing

### Same tech used by Amazon.com

### Integrates into existing website, applications, SMS, email marketing systems...

### Implement in days, not months, (you don’t need to build, train, and deploy ML solutions)

## Textract

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

### Extract data from forms and tables

### Read and process any type of document (pdfs, images)

### Use cases: invoices, financial reports, medical records, insurance claims, tax forms, IDs, passports

# Section 18 - Account Management

## AWS Organization

### Allows you to manage multiple AWS accounts

### Main account is the Master Account

### Benefits…

#### Consolidated billing across all account - 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 creating

#### Restrict account privileges us Service Control Policies

### Multi Account Strategies

#### Create account per department,

##### per cost center

##### per dev / test/ prod

##### based on regulatory restrictions

##### for better resource isolation

##### to have separate per-account service limits

##### isolated for logging

### SCP Service Control Policies

#### 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

## AWS Organization - Consolidated Billing

### When enabled, it provides you with...

#### Combined Usage - saving benefits for volume pricing and saving plans

#### One Bill - for all accounts

## 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 setup of your environment in 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

## 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 risk, predictably manage budgets, comply with long-term requirements

#### Reservations are available for EC2 Reserved Instances and many more...

### Pay Less By Using More: Volume-based discounts

### Pay less as AWS grows

## Compute Pricing - EC2

### Only charged for what you use

#### Number of Instances + Configuration for Instance

#### ELB running time and amount of data process

### On-demand Instances

### Reserved Instances

### Spot Instances

#### Bid for unused capacity

### Dedicated hosts

### Savings Plan

## Compute Pricing - Lambda & ECS

### Lambda

#### Pay per call

#### Pay per duration

### ECS

#### EC2 Launch Type Model: no additional fees, you pay for AWS resources store and created in your app

### Fargate

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

## Storage Pricing - S3

### Storage Class

### Number and size of objects

### Number and types of requests

### Pay for data transfer out of the S3 region

### Transfer acceleration

## Storage Pricing - EBS

### Volume type (based on performance)

### Storage column in GB per month provisioned

### IOPS:

### Snapshots

### Data transfer

## Database Pricing

### Per hour billing

### Database characteristics

#### Engine, size, memory class

### Purchase Type

#### On demand

#### Reserved Instances

### Backup... something

### Number of input/output requests

## Content Delivery Pricing - CloudFront

### pricing is different across different, geographic regions

### Aggregated for each edge location, then applied to your bill

## Networking Costs in AWS per GB -

### Use Private IP instead of Public IP for good savings and better network performance

### Use same AZ for maximum savings (at the cost of high availability)

### AWS charges more when using Public IP instead of Private IP

## Savings Plan Overview

### Commit a certain $ amount per hour for 1 or 3 years

### EC2 Savings Plan

#### Up to 72% discount compared to On-demand

#### Commit to usage of individual instance families in a region (e.g. C5 or M5)

### Compute Savings Plan

#### Up to 66% discount compared to On-Demand

#### Regardless of Family, Regio, size, or \*\*compute options\*\* (applicable to EC2, Fargate, Lambda)

### Machine Learning Savings Plan

#### For SageMaker

## AWS Compute Optimizer

### Reduce costs and improve performance by recommending optimal AWS resources for your workloads

### Helps you choose optimal configurations and right-size your workloads (over/under provisioned)

### Uses ML to analyze and CloudWatch metrics

## Billing and Costing Tools

### For Estimating Costs in the Cloud

#### Pricing Calculator

##### TCO calculator

##### Simple Monthly Calculator / Pricing calculator

### For Tracking Costs in the Cloud

#### Billing dashboard

#### Cost allocation tags

#### Cost and Usage Reports

#### Cost Explorer

### Monitoring Again Costs Plans

#### Billing Alarms

#### Budgets

## Cost Allocation Tags

### Use these tags to track your AWS costs on a detailed level

### AWS generated tags

#### Automatically applied to the resource you create

#### Starts with Prefix aws

### User Defined Tags

#### Starts with prefix User

### Can also use tags for organizing resources and creating Resource Groups

## Cost and Usage Reports

### Dive deeper into your AWS costs and usage

### Contains the most comprehensive set of AWS cost and usage data available

## Cost Explorer

### Visualize, understand and manager your AWS costs and usage over time

### Create custom reports that analyze cost and usage data

### Analyze your data at a high level: totals costs and usage across all accounts

### Helps choose an optimal Savings Plan

### Forecasts usage up to 12 months based on previous usage

## Billing Alarms in CloudWatch

### Billing data metric is stored in CloudWatch us-east-1

### Billing data are for overall worldwide AWS costs

### It’s for actual costs, not projected

## AWS Budgets

### Create budgets and send alarms when costs exceed the budget

### 3 types of budgets: Usage, Cost, Reservation

### For Reserved Instances (RI)

#### Track utilization

#### Supports EC2, ElastiCache, RDS, Redshift

### ...

## AWS Trusted Advisor

### A high-level AWS account assessment

### Analyzes your AWS account and provides recommendations on 5 categories

#### Cost Optimization

#### Performance

#### Security

#### Fault Tolerance

#### Service Limits

### Support Plans (7 core checks) (basic and developer support plan)

#### S3 Buckets Permissions

#### Security Groups

#### IAM Use

#### MFA on Root Account

#### EBS Public Snapshots

#### RDS Public Snapshots

#### Service Limits

### Support Plans (Full Checks) (Business and Enterprise Support Plan)

#### Full Checks on the 5 categories

#### Ability to set CloudWatch Alarms when reaching limits

#### Programmatic access using AWS Support API

## AWS Support Plans Pricing

### Basic Support: Free

#### Customer Service and Communities (24/7 access) (not 24/7 access to phone, email and chat though)

#### AWS Trusted Advisor (Access to the 7 core Trusted advisor checks and provision guidance)

#### AWS Personal Health Dashboard

### Developer Support Plan

#### All Basic Support Plan items

#### \*Business hours email access to Cloud Support Associates\*

#### Unlimited cases/ 1 primary contact

#### Case Severity / response times

##### General guidance < 24 hours

##### System impaired < 12 hours

### Business Support Plan

#### Intended to be used if you have production workloads

#### Trusted Advisor - full set of checks + API access

#### 24x7 phone, email, and chat access to Cloud Support Engineers

#### Unlimited cases / unlimited contacts

#### Access to infrastructure event management for an additional fee

#### Case Severity / Response Times

##### Much better than Dev Support...

### Enterprise On-Ramp Support Plan

#### Intended to be used if you have production or business critical workloads

#### All of business support plan included

#### Access to a pool of technical account managers (TAM)

#### Concierge Support Team

#### Infrastructure Event management, Well-Architected & Operations Reviews

#### Case Severity / Response Times

##### Even better then Business...

### Enterprise Support Pan

#### Intended to be used if you have mission critical workloads

#### All of everything before

#### Access to a \*designated\* Technical Account Manager

#### Access to online training with self-paced labs

#### Case Severity / Response Times

##### Better than Enterprise On-Ramp

##### Business-critical system down < 15 minutes

# Section 19 - Advanced Identity

## Security Token Service

### Enables you to create temporary, limited-privileges credentials to access your AWS resources

### Short-term credentials: you configure expiration period

### Use Cases

#### Identity federation: manage user identities in external systems and provide them with STS tokens to access AWS resources

#### IAM Roles for cross/same account access

#### IAM Roles for EC2 instances

## Amazon Cognito

### Database of users for you mobile and web apps

### Identity manager for your Web and Mobile application uses (potentially millions)

### Instead of creating them an IAM user, you create a user in Cognito

## Amazon Single Sign-On (SSO)

#### Centrally manage access to multiple AWS accounts and business applications

#### Creates a central user portal to in into third-party business apps

#### Assists with Logins for Facebook, google, twitter...(?)

## AWS Directory Services

### AWS Managed Microsoft AD

#### create your own Active Directory in AWS, manage users locally, supports MFA

#### Establish "trust" connections with your on-premise AD

### AD Connector

#### Directory Gateway (proxy) to redirect to on-premised AD

#### Users are managed on the on-premise AD

### Simple AD

#### AD-compatible managed directory on AWS

#### Cannot be joined with on-premise AD

## AWS IAM Identity Center

### (successor to AWS single sign-on)

### One Login for all your...

#### AWS accounts in AWS Organizations

#### Business cloud apps

#### SAML2.0 enabled apps

#### EC2 Windows Instances

### Identity Providers

#### Built in identity store in IAM identity center

#### 3rd party...

# Section 20 - Other Services (sometimes, but rarely appear on the AWS exam)

## Amazon WorkSpaces

### Managed desktop as a service (DaaS) solution to easily provision Windows or Linux Desktops

### Great to eliminate management of on-premise VDI (virtual desktop infrastructure)

### Multiple Regions

#### Reduce Latency by having WorkSpaces in multiple regions

## Amazon AppStream 2.0

### Desktop Application Streaming Service

### Deliver to any computer, without acquiring, provisioning infrastructure

### The application is delivered from within a web browser

### Simply streams a desktop application to web browsers

## Amazon Sumerian

### Create and run virtual reality (VR) and augmented reality (AR) and 3D applications

### Can be used to quickly create 3d models with animations

## AWS IoT Core

### Allows you to easily connect IoT devices to the AWS Cloud

#### Cars, fridges and other things like that

### Serverless, secure, and scalable to billions of devices and trillions of messages

## Amazon Elastic Transcoder

### Used to convert media files store in S3 into media files in the formats required by consumer playback devices (phones etc...)

## AWS AppSync

### Store and sync data across mobile and web apps in real-time

### Makes use of GraphQL (mobile tech from Facebook)

## AWS Amplify

### A set of tools and services that helps you develop and deploy scalable full stack web and mobile applications

### Authentication, Storage, API, CI/CD, PubSub... bunch more

## AWS Device Farm

### Fully managed service that tests your web and mobile apps against desktop browsers, real mobile devices, and tablets

### Runs tests concurrently on multiple devices (speed up execution)

### Ability to configure device settings (gps, languange, wifi, bluetooth)

## AWS Backup

### Fully-managed service to centrally manage and automate backups across AWS services

### On-demand and scheduled back ups

## Disaster Recovery Strategies

### Backup and Restore (cheapest to ...)

### Pilot Light

### Warm Standby

### Multi-Site / Hot-Site (...most expensive)

## AWS Elastic Disaster Recovery (DRS)

### Quickly and easily recover your physical, virtual, and cloud-based servers into AWS

### continuous block-level replication for your servers from corporate data centers into AWS Cloud

## AWS DataSync

### Move large amount of data from on-premises to AWS

### Can synchronize to: Amazon S3

### Replication tasks are \*\*incremental\*\* after the first full load

#### Can be scheduled hourly, daily, weekly

## AWS Application Discovery Service

### Plan migration projects by gathering information about on-premises data centers

### Server utilization data and dependency mapping are important for migrations

### Agentless Discovery

#### VM inventory, configuration, and performance history such as CPU, memory, and disk usage

### Agent-based Discovery

#### System config, sys performance, running processes, network details, sys connections

## AWS Application Migration Service (MGN)

### Lift-and-shift solution which simplifies migrating applications to AWS

### converts your physical, virtual, and cloud-based servers to run natively on AWS

## AWS Fault Injection Simulator (FIS)

### Fully managed service for running fault injection experiments on AWS workloads

### Based on \*\*Chaos Engineering\*\* - stressing an application by creating disruptive events, and observing how the system responds

## AWS Step Functions

### Build serverless virtual workflow to orchestrate your Lambda functions

### Features: sequence, parallel, conditions, timeouts, error handling...

### Use Cases: order fulfillment, data processing, web applications, any workflowp

## AWS Ground Station

### Fully managed service that lets you control satellite communications, process data, and scale your satellite operations

### Provides a global network of satellite ground stations near AWS regions

### Allows you to download satellite data to AWS VPC within seconds

### Use cases: weather forecasting, surface imaging, communications, video broadcasting

## Amazon Pinpoint

### Scalable 2-way (outbound/inbound) marketing communications service

### Supports email, SMS, push, voice, and in-app messaging

### Scales to billions of messages per day

### Use cases: run campaigns by sending marketing bulk, transactional SMS messages

### Vs Aws SNS or SES

#### In SNS & SES you managed each message's audience, content, and delivery schedule

#### In Pinpoint, you create message templates, delivery schedules, highly-targeted segments, and full campaigns

# Section 21: AWS Architecting & Ecosystem

## AWS Well Architected Framework General Guiding Principles

### Stop guessing your capacity

### Test systems at production scale

### Automate to make architectural experimentation easier

### Allow for evolutionary architectures

#### Design based on changing requirements

### Drive architectures using data

### Improve through game days

#### Simulate applications for flash sale days

## AWS Cloud Best Practices - Design Principles

### Scalability

### Disposable Resources - servers should be disposable and easily configured

### Automation: Serverless, infrastructure as a Service, Auto Scaling

### Loose Coupling

#### Monolithic apps can be too tightly coupled

#### Break it down into smaller, loosely coupled components

#### A change or failure in one component should not cascade to other components

### Think in services, not servers

#### Don’t just use EC2s, use managed services, databases, and serverless stuff

## Well Architected Framework 6 Pillars

### Operational Excellence

### Security

### Reliability

### Performance Efficiency

### Cost Optimization

### Sustainability

## Operational Excellence

### Includes the ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures

### Design Principles

#### Perform operations as code (Infrastructure as Code)

#### Annotate documentation

#### Make frequent, small, reversible changes

#### Refine operations procedures frequently

#### Anticipate failure

#### Learn from all operational failures

## Security

### Protecting information, systems and assets while delivering business value through risk assessments and mitigations strategies

### Design Principles

#### Implement a strong identity foundation

#### Enable traceability

#### Apply security at all layers

#### Automate security best practices

#### Protect data in transit and at rest

#### Keep people away from data - reduce or eliminate the need for direct access or manual processing of data

#### Prepare for security events

## Reliability

### Ability of a system to recover from infrastructure or service disruptions, dynamically acquire computing resources to meet demand.

### Design Principles

#### Test recovery procedures

#### Automatically recover from failure

#### Scale horizontally to increase aggregate system availability

#### Stop guessing capacity - maintain optimal levels for demand by using Auto Scaling

### Manage change in automation

## Performance Efficiency

### Use computing resources efficiently to meet sys requirements as demand changes

### Design Principles

#### Democratize advanced technologies - advanced technologies become services and hence you can focus more on product deployment

#### Go Global in Minutes

#### Use serverless architectures

#### Experiment more often - easy to carry out comparative testing

#### Mechanical sympathy - be aware of all AWS services

## Cost Optimization

### Ability to run systems to deliver business value at the lowest price point

### Design Principles

#### Adopt a consumption mode - pay only for what you use

#### Measure overall efficiency - use CloudWatch

#### Stop spending money on DataCenter operations - AWS does the infrastructure and the customer focuses on projects

#### Analyze and attribute expenditure

#### Use managed and application level services to reduce cost of ownership

## Sustainability

### Focuses on minimizing the environmental impacts of running cloud workloads

### Design Principles

#### Understand your impact

#### Establish sustainability goals

#### Maximize utilization

#### Anticipate and adopt new, more efficient hardware and software offerings

#### Use managed services 0 shared services reduce the amount of infrastructure; managed services help automate sustainability best practices

#### Reduce the downstream impact of your cloud workloads

## AWS Well-Architected Tool

### Free tool to review your architecture against the 6 Pillars and adopt best practices

## AWS Right Sizing

### EC2 has many instance types, but choosing the most powerful instance type isn't the best choice, because the cloud is \*Elastic\*

### Right sizing is the process of matching instance types and size to your workload performance and capacity requirements at the lowest possible cost

### Scaling up is easy so always start small

### Also, the process of finding where you can downsize / eliminating instances

## AWS IQ

### Quickly find professional help for your AWS projects

### Engage and pay certified 3rd party experts for on-demand project work

### Video conferencing, and secure collaboration

### Like fiverr for AWS projects

## AWS re:Post

### Forum website for AWS questions specifically

### Just like stack overflow and reddit with upvotes and stuff