**Shared Responsibility**

AWS is responsible for security ‘**OF’** the cloud, Infrastructure

User is responsible for security **‘IN**’ the cloud

**AWS Acceptable Use Policy** – Policy of AWS to be followed

**Types of Cloud Computing**

IaaS – data storage, networking (building blocks), physical Ex. EC2

PaaS – no infrastructure, managing and deploying applications focus (no physical management) Ex. Elastic Beanstalk

SaaS – end user (product is already made) Ex. Rekognition

**Difference between Edge Locations, AZ, Regions, Data Centers**

Regions – Regions are connected with a network, all around the world, cluster of data centers, use to run services.

AZ – Each region has different AZ (eu-east-1a, eu-east-1b), one or more discrete data centers with redundant power, networking and connectivity (one fails, other can be up, physically separated and isolated with disaster)

Edge Locations – Points of Presence, used for low latency, near the users to connect with AWS

Global Services – IAM, DNS (Route 53), CloudFront (Content Delivery Network), WAF (Web Application Firewall)

Region Scoped – AWS EC2, Beanstalk, Lambda, Rekogition

**Pricing Model**

1. **Pay as you go**

Compute, Storage, Data Transfer OUT of the cloud

**IAM (Identity and Access Management) –**

Global Service

Root Account, Users can be grouped

Groups can get IAM Permission in JSON called Policies

Least Privilege Principle

**MFA – Multi Factor Authentication**

Password you know + Security device you own

Options – Virtual MFA Device (Authy, Google Authenticator), U2F Security Key (USB Key) Physical Device,

Hardware Key Fob MFA Device, Key Fob for government.

IAM Roles for Services – Roles for Services to access the resource

IAM Security Roles (for auditing) – IAM Credentials Report (account level), IAM Access Advisor (user level)

Best Practices:

1. Do not use root account
2. User to groups and permissions to group
3. Use MFA
4. Each user is a physical user
5. Create roles
6. Access key for CLI or SDK

Policy is for User, Role is for Service

**Billing Dashboard –**

**Budgets –** alerts and create monthly budgets if we go over the budget

**EC2 –** Elastic Compute Cloud, IaaS (AMI: OS, Instance Size (CPU+RAM) + Storage, Security Group (firewall) and EC2 User Data), virtual server

**Security Group** – Controls traffic in and out of EC2 Instances

**EC2 User Data** – Bootstrap Script

**EC2 Purchasing Options:**

1. **On-Demand**

Pay for what you use, Linux per sec, Windows per hour

High Cost no upfront payment

Uninterrupted Workload

1. **Reserved (min 1 year or 3y) – Reserved, Convertible, Scheduled Reserved**

No Upfront, partial, all up front

Reserve an instance type

Change the EC2 Instance Type

Tue between 5pm to 9pm

1. **Spot Instances**

Can loose them over time

Biding auction

Workload resilient to failure

Batch Jobs, Data Analysis, Image Processing, distributed workload, flexible start and end time

1. **Dedicated Hosts**

Physical Server, compliance requirements and use existing server bound software licenses. 3y period reservation, per host billing, hardware access, hardware not shared with anyone

1. **EC2 Dedicated Instances**

Hardware to you but not control over hardware, per instance billing

**EC2 Instance Storage Options**

1. **EBS (Elastic Block Store) – AZ level, Data persists even if stopped**

Connected through network (network drives), experiences latency

Snapshots taken to move volumes

**One EBS cannot be attached to more than one instance at a time**

**Two EBS volume can be attached into 1 instance**

**EBS Snapshots – Back up**

1. **AMI – Amazon Machine Image (customization of EC2 as a whole, ready-to-use EC2)**

Increases boot and configuration time

Region Level and copied across regions

1. **EC2 Instance Store – Physical Device attached to the Server to increase performance, Data lost when terminated, very high IOPS**
2. **EFS (Elastic File System) – mount of 100s of EC2 only Linux on different AZ**

**ELB & ASG (Elastic Load Balancing and Auto Scaling Group)**

Vertical Scalability: Increase size of the Instance (Database), has hardware limit, scaling up and scaling down.

Horizontal Scalability (Elasticity): increase number of EC2 Instances, distributed systems, scaling in and out. (ELB, ASG)

High Availability System – Runs application on at least two AZ

**ELB:** Forward internet traffic to multiple servers (down streaming), single DNS, health checks, multiple AZs

**3 Kinds of ELB**: Application (Layer 7), Network (Ultra High Performance, TCP: Layer 4), Classic ELB (Layer 4,7)

**Auto Scaling Group:** Add or remove EC2 Instances depending on Load, registered with load balancer, removes unhealthy instances, running at optimal capacity.

Configured: Minimum Size, Desired Size, Maximum Size

**AWS S3 Storage: “infinitely scaling storage”, store objects in buckets, global storage**

Buckets are defined and create in regions

FACT: EBS Snapshots stored in AWS S3

Backup and Storage, Disaster Recovery, Archive, Analytics…

Objects have Key – Full Path (prefix + object name (file name))

Objects greater than 5GB Stored as multi part and max cap is 5000GB or 5TB

**Security in S3**

1. User Based: IAM Roles
2. Resource Based: Bucket Policies, Object ACL (more common), Bucket ACL (less common), Cross Account Policy
3. Encryption

Anonymous User can use the S3 Because of S3 Policy attached.

AWS User uses because of IAM User

EC2 Instance can access using EC2 Roles using IAM Policy

S3 Websites – Hosting Static Websites

**FACT: 403 IS FORBIDDEN ACCESS DENIED ERROR!**

S3 Versioning – Enabled at bucket level, previous version is stored, unintended deletes and can be rolled back.

S3 Access Logs – Auditing into another S3 Bucket, finding root cause of issues.

S3 Replication – Cross Region or Same Region (CRR & SRR), enabling version, happens asynchronously

Durability – how often you loose (11 9’s)

Availability – How ready available

**S3 Storage Classes**

1. Amazon S3 Standard: Frequency Accessed, 2 concurrent failures, 99.99%
2. Amazon S3 Standard – Infrequent Access (IA): rapid access, lower cost, retrieval fee, 99.9%
3. Amazon S3 One Zone – Infrequent Access: 99.9%, cost optimized between 1 and 2, one zone, thumbnail
4. Amazon S3 Intelligent Tiering – all zones and same as 3
5. Amazon Glacier – low cost, frozen for years, retrieval fees (Expediated, Standard, Bulk)
6. Amazon Glacier Deep Archive – 12 Hours retrieval time

Possible to move them into different storage classes using lifecycle rules.

Snowball: Physical TB or PBs in or out AWS (used to transport data into S3), has import/export function

1 week over network then use Snowball device

Snowball Edge: AMI can be loaded while it is moving, pre-process data

Snowmobile: TRUCK!! 1EB of data (more than 10PB)

AWS Store Gateway: Used to do Hybrid Storage on premise and cloud

BLOCK: AMAZON EBS, EC2 Instance Store

FILE: EFS

OBJECT: AMAZON S3, Glacier

**Databases:**

Structured way using indexes since others have limits, relationships between datasets

1. **Relational Database – SQL Language for querying, vertical scalability**

RDS

Aurora (Proprietary AWS DB), not open source, supports postgres SQL and MySQL, cloud optimized, good for OLTP (Online transactions)

1. **No SQL DB – flexible schema, horizontal scalability, data stored in JSON Format**

DynamoDB – replicated availability through 3 AZ, “**serverless**”, low latency, minimum latency, integrated with IAM

Key/value DB, items stored row by row

**ElastiCache –** Redis and Memcached “in-memory”, read intensive workload, reduce workload off databases, “server” needed.

**RedShift - Postgres SQL,** OLAP (Analytics in warehousing), load data every hour, Columnar storage of data, Massive Parallel Query Execution (MPP)

**EMR – Elastic MapReduce,** create HADOOP Clusters for bigdata, to analyze data, autoscaling with spot instances

**Athena – “Serverless DB” with SQL, S3, Pay for every query you run**

**DMS – DB Migration Service – migration of data into different DB, Homogenous, Heterogenous**

**Glue –** ETL Service ( Extract, transform, load) for analytics, transform data if not in correct format, “**serverless**”

**Glue Data Catalog –** **Catalog of services of datasets**

**­­­­­­­­­­Docker –** uses containers to deploy apps, works with any environment, light weight

Docker runs different container, images are stored in docker repos

ECR – Elastic Container Repo (Private AWS Repo to store images)

VM has Hypervisor

**ECS Elastic Container Service (launch docker containers on AWS):** Need to Manage infrastructure

**Fargate:** Launch Docker containers without infrastructure “serverless”

**ECR :** **Elastic Container Registry:** Store docker images, container is created by ECS or Fargate

**Serverless:** Deploy code, user doesn’t manage the server

**Serverless:** S3, DynamoDB, Fargate, Athena, Glue, Lambda

**EC2 Limitation –** Bound by RAM and CPU

**Lambda Service:** no server, short executions, on-demand, pay per request (call) and compute time, event-driven, reactive, 15 mins, thumbnails to upload on s3

**Limitation:** Doesn’t support Docker!!

Server CRON Job – Monotonous Job

**AWS Batch –** batch processing, start and end, not continuous, docker images running on ECS, dynamic provisioning of resources

**Lambda vs Batch**

Lambda - Serverless, time limit, limited runtime

Batch – No time limit, EBS or instance store, relies on EC2

**LightSail –** VMS, storage, database, network together, little cloud experience, notifications and monitoring can be done, template are present to be used, development and testing, no auto scaling with limited integrations but has high availability, low pricing

**Deployment -**

1. **CloudFormation (AWS only) –** template is provided, and they are created with the configuration given, no manual need to create. “IaaS”, tags are used to see the cost of the stack, generates diagrams on which to create first, declarative programming (no need to know what to create first), repeat architecture in different ways.
2. **Beanstalk (AWS only) –** “PaaS”, developer centric view, one view of all the services, managed service. Types: Single Instance deployment, LB+ASG, ASG only (non-web)
3. **Code Deploy (Hybrid) –** upgrading from version 1, works with EC2 and on-premises servers (hybrid service), deploy application automatically
4. **AWS System Manager (SSM, Hybrid) –** manage EC2 and on-premises service, operational insights of the state of infrastructure, works both on windows and Linux, patching automation for enhanced compliance, run command on all servers.
5. **OpsWork (Hybrid) –** **Chef & Puppet, Cookbook (Server configuration – third party),** alternative to SSM, no EC2, DBs, etc.

**Global Applications-**

1. **Global DNS: Route 53 -** Managed DNS, least latency, closest deployment

**Domain Registration, DNS, Health Check, Routing Policy**

Simple Routing Policy: No Health Checks

Weighted Routing Policy (20% - 80%): Load Balancing, health checks

Latency Routing Policy: find the closet server, minimize latency, health check

Failover Routing Policy – health check on primary and shift to secondary

1. **Global Content Delivery Network: Cloud Front (at the Edge Location)**

Improves latency by caching and improves read performance at edge locations, protects against DDoS, integrated with shield and AWS Web Application Firewall.

Stores: S3 Bucket, Custom HTTP, EC2

Cloud Front vs S3 CRR

CloudFront – Static, global network, cached for a day

CRR – Dynamic, setup for each region, read only, not cached but updated in real time

1. **S3 Transfer Acceleration**

File to Edge location (public), Edge Location to S3 (Private) – decrease latency

1. **AWS Global Accelerator**

Increase availability and performance, connect with edge location: decrease the traffic, no caching, uses internal AWS Network

**How Applications Communicate? (Cloud Integration)**

Async: Put in a queue, SQS, SNS

Sync: Lot of tragic

**Amazon SQS: Simple Queue Service –** “Serverless”, messages are stored in queues, independent scaling and decoupling (one component failure shouldn’t not affect other components), pull base service

**Amazon SNS:** PUB/SUB (SNS Topic) Simple Notification Service, no message retention

**Cloud Monitoring**

**CloudWatch (Hybrid) Metrics –** monitors every service on the cloud, timestamps, billing metric (us-east-1),

**EC2**  - CPU Utilization, Status Check, Network (not RAM)

**EBS –** Disk Read/Writes

**S3 Buckets –** Bucket Size in Bytes, Number of Objects, All Requests

**Billing –** Total Estimated Charge

**Service Limit – APIs called**

**CloudWatch Alarms –** If the metric goes above the threshold. **If happens then? Auto Scaling, EC2, SNS**

**Amazon Cloud Watch Logs –** used to collect log files (Beanstalk, ECS, AWS Lambda, CloudTrail, CloudWatch Agents, Route 53), real time monitoring of logs

**Configuration required on EC2 for enabling Cloud Watch (by downloading CloudWatch Log Agents)**

**CloudWatch Events –** Scheduling CRON Jobs, create event patterns

**Amazon Event Bridge –** evolution of Cloud watch, has default event bus, partner event bus, custom event bus, schema bridge.

**AWS Cloud Trail –** governance, compliance and audit of aws account, enabled by default, stores the history of events/ API Calls by all AWS Services. (90 Days)

CloudTrail can be put into CloudWatch Logs or S3 and can be applied to all regions (default) or just one region.

**AWS X-Ray –** debugging in production, tracing and get visual analysis and see what is happening in each service. Troubleshooting, understand dependencies, pinpoint service issues, find errors and exceptions, SLA (Service Level Agreements), “distributed services”.

**AWS Service Health Dashboard –** shows health of all regions and all services

**Personal Health Dashboard –** alerts when the services might affect you, provides remediation and guidance.

**VPC (Virtual Private Cloud)**

1. VPC: private cloud where resources are deployed, linked to a region, dedicated to your aws account, logically isolated section.
2. Subnet: Part of VPC linked with AZ
3. A public Subnet – direct connectivity through internet, EC2, Load Balancer
4. Private Subnet – no direct connectivity to internet, DB
5. Internet Gateway – VPC connect to Internet (public subnet)
6. NAT **Gateway**– establishing way to connect private to public and public to internet with the internet gateway.
7. Network ACL (NACL) Stateless– firewall that controls traffic in Subnet, Allow or deny, only IP Address (***used in network security***) NACL – Network access control lists
8. Security Group Stateful – Allow rules Used in EC2 or Elastic Network Interface (ENI), can have IP Addresses or Security Groups
9. VPC Flow logs – IP traffic into the interfaces, VPC Flow logs, Subnet Flow logs and ENI Flow logs, helps in monitoring and troubleshooting connectivity issues, exported to S3 or Cloud Watch.
10. VPC Peering – Connecting two VPCs using private network to combine as 1, IP must not overlaps, not transitive
11. VPC Endpoint – allows to connect with AWS Service using a private network for enhanced security. To connect with S3 and DynamoDB use “VPC Endpoint Gateway”. “VPC Endpoint Interface” – to connect with rest services.
12. Site to Site VPN – on premises VPN (Customer Gateway) to AWS (Virtual Private Gateway), encrypted automatically, public internet, limited bandwidth, quick (5 mins)
13. Direct Connect (DX) – Private connection, physical connection, expensive, takes a month at least.
14. Transit Gateway – hub and spoke star connection when everything gets complicated, works with everything, 100s and 1000s of VPCs or on-premises systems to be connected.

**Security and Compliance –**

DDOS Attack – Distributed denial of service: creates many requests from different hosts

1. **AWS Shield Standard** – free, common attacks, layer 3/4
2. **AWS Shield Advanced** – 24/7 protection, optional, response team aid
3. **WAF – Web application firewall**, filters requests based on rules, layer 7, can be deployed on application load balancers, CloudFront and API Gateway (since layer 7)

* Will have Web ACL: Filtering on IP Address, protects against SQL Injections, Cross-site scripting, geo match (block countries), rate based rules (5 requests per min)

1. **CloudFront and Route 53** protected by AWS Shield

**Architecture**

1. Route 53 protected with Shield
2. CloudFront protected with Shied
3. WAF to filter requests
4. Load Balancers
5. Autoscaling groups

**Penetration Testing:** Attacking own infrastructure to check security without prior approval (8 services)

**Encryption with KMS:** keys managed by AWS, auto-enabled for CloudTrail Logs, S3 Glacier, Storage Gateway.

1. Customer Manager CMK: bring own key, rotation of keys, create manage and use
2. AWS Managed Key: managed by AWS
3. **CloudHSM** – Encryption Hardware, Hardware Security Model is provided, security keys managed by user.

**Secrets Manager:** stores secrets, secrets need to change, automated using lambda, encrypted using kms, used in rds to generate password.

**Artifact:** On-demand access to AWS Documentation and AWS Agreements

**Guard Duty:** used to protect from threats outside and inside AWS Account, uses ML to detect anomalies. Sees CloudTrail Logs, DNS Logs, VPC Flow Logs.

**Inspector: *Automated Security assessments***, analyses the running OS against vulnerabilities, installed on the OS.

**Config:** Records the configuration for auditing, can be stored in s3, ssh access, buckets public, alb config

**Macie:** Uses ML and Pattern Matching, PPI Personally Identifiable Information, protects sensitive data in AWS, notifies using CloudWatch, finds sensitive data from s3 and then notifies.