AWS Certified Solutions Architect – Associate

https://www.youtube.com/watch?v=Ia-UEYYR44s

Simple Storage Service S3

Introduction

- Serverless object storage
- Unlimited storage
- Objects consist of Key, Value, Version ID and Metadata
- 0 bytes to 5TB objects
- Buckets
    - can also have folders containing objects
    - universal namespace

Storage Classes

	Standard	Intelligent Tiering	Standard IA	One-Zone IA	Glacier	Glacier Deep Archive
Durability	11 9s	11 9s	11 9s	11 9s	11 9s	11 9s
Availability	99.99%	99.99%	99.99%	99.5%	N/A	N/A
Availability SLA	99.99%	99%	99%	99%	N/A	N/A
AZs	>3	>3	>3	1	>3	>3
Min Capacity charge per object	N/A	N/A	128kb	128kb	40kb	40kb
Min storage duration charge	N/A	30 days	30 days	30 days	90 days	180 days
Retrieval fee	N/A	N/A	Per GB	Per GB	Per GB	Per GB
First byte latency	ms	ms	ms	ms	mins to hours	hours

Security

- private by default
- Logging can be turned on
- Access control with Bucket Policies(JSON) and ACL

Encryption

- In Transit: SSL/TLS
- At Rest:
    - Server Side Encryption:
        - S3 Managed Keys
        - SSE AES, SSSE-KMS, SSE-C

    - Client Side Encryption:
        - CSE-KMS, CSE-C

Data Consistency

- New Objects (PUTS): Read After Write Consistency
- Overwrite or Delete: Eventual Consitency (Replication to AZs takes time)

Cross-Region Replication

- Object will be automatically replicated to another regions if turned on
- Can be replicated to another AWS account

Versioning

- Store all versions of an object in S3
- Integrates with Lifecycle policies

Lifecycle Management

- Automate process of moving objects through storage classes or delete them

Transfer Acceleration

- Fast and secure transfer over long distance between end user and bucket
- Utilizes Edge Locations
    - Distinct URL for edge location

Presigned URLs

- Generate URL providing temporary access to upload or download object data to provide access to private objects
- Expiring time can be set

MFA Delete

- Ensures users cannot delete objects without MFA code
- Versioning is required

Snowball

Snowball

- Petabyte scale data transfer service
- Lower cost and faster than over network
- Tamper and weather proof, Data encryption, transfer must be completed in 90 days
- Can import and export from S3
- 50 TB or 80TB

Snowball Edge

- More storage and on-site computing capabilities
- LCD display, can undertake local processing and edge-computing workloads
- Can use in a cluster of 5 to 10 devices
- Storage optimized, compute optimized, GPU optimized

Snowmobile

- Shipping container on a truck
- Transfer up to 100 PB
- A lot of security features

Virtual Private Cloud VPC

- Provision logically isolated section of AWS Cloud in a virtual network
- Own personal data center
- Complete control over your virtual networking environment

Core Components

- Internet Gateway
- Virtual Private Gateway
- Routing Tables
- NACL
- Security Group
- Private Subnet
- Public Subnet
- NAT Gateway
- Customer Gateway
- VPC Endpoints
- VPC Peering

Key Features

- Region Specific, 5 per Region
- 200 Subnets per VPC
- IPv4 Cidr Block or IPv6 Cidr Block
- Free: VPC's, Route Tables, NACLs, Internet Gateway, Security Groups, Subnets, VPC Peering
- Not Free: NAT Gateway, VPC Endpoints, VPN Gateway, Customer Gateway
- DNS hostnames disabled by default

Default VPC

- One default VPC in every region, to immediately deploy instances
- Size 16 IPv4 CIDR block
- Size 20 default subnet in each AZ
- Internet Gateway connected to it
- Default security group associated to it
- Default NACL associated to it
- Default DHCP option set
- Created with main route table

Default Everywhere IP (0.0.0.0/0)

- Represents all possible IP addresses
- When in route table for IGW we allow internet access
- When in security group inbound we allow all traffic from internet or public resources

VPC Peering

- Allows to connect one VPC with another over direct network route using private IPs

- Instances behave just like they are in the same network
- Connect across AWS accounts and regions
- Star Configuration: 1 Central VPC – 4 other VPCs
- No transitive Peering (needs one to one connection)
- No overlapping CIDR Blocks

Route Tables

- Used to determine where network traffic is directed
- Each subnet must be associated with a route table
    - Subnet can only be associated with one route table at a time

Internet Gateway

- Allows internet access for VPC
- Provides a target in your VPC route tables for internet traffic
- Performs NAT for instances that have been assigned public IPv4 addresses
- Route with Destination 0.0.0.0/0 has to be added in Route table

Bastions / Jumpbox

- Security hardened EC2 instances to help gain access via SSH or RCP in a private subnet
- NATS should not be used as Bastions
- System Managers Sessions Manager replaces need for Bastions

Direct Connect

- Establishes dedicated network connections from on-prem to AWS with very fast network
- Helps reducing network cost and increase bandwidth throughput
- Provides more consistent network experience

VPC Endpoints

- Privately connect VPC to other AWS services
- Eliminates need for Internet Gateway, NAT device, VPN connection or AWS Direct Connect
- Instances do not require public IP address to communicate with service resources
- Traffic does not leave AWS network
- Horizontally scaled, redundant and highly available
- Secure communication between instances and services without adding risks or bandwidth constraints
- Interface Endpoints
    - Elastic Network Interfaces (ENI) with private IP
    - Entry point for traffic going to a supported service
    - Powered by AWS PrivateLink
- Gateway Endpoints
    - Free to use (only S3 and Dynamo DB supported atm.)

    - Gateway that is targeted for specific route in route table, used for traffic destined for supported AWS service

VPC Flow Logs

- Allow to capture IP traffic information within VPC
- Useable for: VPC, Subnets, Network Interface
- Stored using CloudWatch Logs
- Structure:

<version> <account-id> <interface-id> <srcaddr> <dstaddr> <srcport> <destport> <protocol> <packets> <bytes> <start-time> <end-time> <action> <log-status>

Network Access Control List (NACL)

- Optional layer of security acting as firewall for controlling in and out traffic of subnets
- Subnets can only belong to a single NACL
- Each NACL contains set of rules to allow or deny traffic
    - Rules have ordered evaluation

Security Groups

- Virtual firewall controlling inbound and outbound traffic at instance level
- Associated with EC2 instance
- Set of rules, deny all by default and can only allow traffic (Stateful)
- Limits: Up to 10k Security Groups in Region, 60 inbound and 60 outbound per SG, 16 SG per Elastic Network Interface

Network Address Translation (NAT)

- Re-mapping one IP address space into another running in Public Subnet
- NAT instances (legacy): individual EC2 instances
- NAT Gateways: managed service to launch redundant instances within AZ

Identity Access Management (IAM)

- Manages access of users and resources

Core Components

- IAM identities:
    - IAM Users: End users logging in console, using CLI/SDK
    - IAM Groups: Group of users with shared permissions
    - IAM Roles: Associate Permissions to role to assign to users or groups or to AWS resources
- IAM Policies: JSON document granting permissions, attached to IAM identities

Types of Policies

- Managed Policies: managed by AWS, not editable, labeled with orange box
- Customer Managed Policies: created by customer, editable, no symbol
- Inline Policies: Policy attached to a user

Policy Structure

- Version: Policy language version
- Statement: Container for policy element
- Sid: label
- Effect: Allow or deny
- Principal: account, user, role
- Action: List of actions to allow or deny
- Resource: resource the policy applies to
- Condition: Circumstances policy applies to

Password Policy

- To set minimum requirements, pw rotation, etc.

Access Keys

- Allowing access via CLI or SDK
- Two access keys per user allowed

Multi Factor Authorization

- Can be turned on per user
- Admin can request MFA to access certain resources

Cognito

- Decentralized managed Authentication

Web Identity federation

- To exchange identity, security information between identity provider (IdP) and application
- IdP: trusted Provider of user identity like Facebook, LinkedIn, Google, Amazon, etc.
- Protocols:
    - OAuth with OpenID Connect (OIDC)
    - Security Assertion Markup Language (SAML) for Single Sign On (SSO)

User Pools

- User directory with authentication to IpD to grant access to apps
- Actions: Sign-up, Sign-in, Account recovery, Account confirmation
- AWS Cognito as identity broker between AWS and IdP
- Successful authentication generates JSON Web Token
- On AWS: attributes, pw requirements, MFA, restrictions, etc.

Identity Pools

- Provide temporary credentials for users to access AWS services
- Actual Mechanism authorizing access to AWS resources

Sync

- Syncs user data and preferences across devices with push synchronizations using SNS

AWS CLI & SDK

- Control multiple AWS services from command line and automate with scripts

Command Line Interface (CLI)

- Let's you interact with AWS using a command line
- Python Script to install CLI
- Switch between accounts using **–profile**
- Change **--output** between JSON, table and text

Software Development Kit (SDK)

- Set of tools and libraries to control AWS services using programming languages
- C++, Go, Java, JS, .NET, NodeJs, Python, PHP, Ruby

Programmatic Access

- Needed for CLI and SDK usage
- Ability to create Access Key ID and Secret Access Key → AWS Credentials

Domain Name System DNS

- Translates domain names to IP addresses

Internet Protocol (IP)

- Uniquely identify computers on a network allowing communication
- IPv4: 32-bits, like 10.100.102.1
- IPv6: 128 bits

Domain Registrars

- Authorities with ability to assign domain names under top-level domains
- Registered through InterNIC

Top-Level Domains

- Last word within domain name
- Second word = second level domain
- Controlled by Internet Assigned Numbers Authority (IANA)

Start of Authority (SOA)

- Every domain must have an SOA record
- Contains various information about domain

Adress(A) Records

- Allows to convert name of a domain directly into IP address

Canonical Names (CNAME) Records

- Resolve one domain name to another, rather than IP address

Name Server (NS) Records

- Used by top-level domain servers to direct traffic to DNS servers

Time To Live (TTL)

- Length of time a DNS record gets cached

Route53

- HA and scalable DNS
- Register and manage domains, create DNS routing rules, resolve VPC's outside of AWS

Use Case

- To get your custom domains to point to your AWS resource

Record Sets

- Allowing to point naked domain and subdomains via Domain Records
- AWS has own special Alias Record to extend DNS functionality
    - Detect change on AI and continuously keep endpoint pointed to correct resource
    - In most cases you want to be using Alias

Routing Policies

- Traffic flow: visual editor to create routing configurations, supporting versioning
- 7 Policies:
- Simple Routing Policies
    - Default Policy
    - 1 record and provide multiple IP addresses (random one will be picked by Route53)
- Weighted Routing Policies
    - Split up traffic based on assigned weights (send percentage of request to endpoints)
- Latency Based Routing
    - Direct traffic based on lowest latency possible for end-user, based on region
- Failover Routing Policies
    - Create active/passive setups for primary site and secondary recovery site
    - Route 53 monitors health checks and redirects traffic if necessary
- Geolocation Routing Policies
    - Direct traffic based on geographic location
- Geoproximity Routing Policies
    - Direct traffic based on location of users and AWS resources
    - Assigning Bias for Regions to increase proximity boundaries
- Multi-Value Answer Policies
    - Like simple Routing Policies, using additional Health Checks to pick next IP

Health Checks

- Checks every 30s by default, can be reduced to 10s
- Can initiate failover if status unhealthy
- Cloud Watch Alarm can be created for alerts

Resolver

- Regional service to connect route DNS between VPC and on-premise network

Elastic Compute Cloud (EC2)

- Highly Configurable Server (Choose OS, Storage, Memory, Network Throughput)
- launch in minutes
- Choose OS via Amazon Machine Image (AMI)
- Choose Instance Type
- Add Storage (EBS, EFS)
- Configure Instance

Instance Types

- General Purpose: balance of compute, memory, network (web servers, code repos)
- Compute Optimized: higher CPU (modeling, gaming servers, ad server engines)
- Memory Optimized: process large data in memory (databases, big data analysis)
- Accelerated Optimized: hardware accelerators (ML, speech recognition)
- Storage Optimized: high sequential read and write on local storage (NoSQL, transactional DB, data warehousing)

Instance Sizes

- EC2 instances normally double in price and key attributes from nano to xlarge

Instance Profiles

- Attach a IAM role to instance via Instance Profile for permissions
- Always avoid embedding AWS credentials

Placement Groups

- Choose logical placement of instance to optimize performance, communication, durability
- Cluster:
    - packs instances close together inside AZ
    - low-latency → for High Performance Computing applications
- Partition:
    - Spreads across logical partitions, not sharing hardware
    - For large distributed and replicated workloads (Hadoop, Kafka, Cassandra)
- Spread:
    - Each instance in different rack
    - To spread critical instances from each other, can be multi AZ

Userdata

- Customer provided Script which runs automatically when launching EC2 instances

Metadata

- Access via endpoint at http://169.254.169.254/latest/metadata

On-Demand Instances

- Default model
- No up-front payment and no long-term commitment
- Charged by hour or minute
- For short-term, spikey or unpredictable workload

Reserved Instances (RI)

- Reduced pricing based on Term x Class Offering x Payment Option
    - Term: 1 or 3 Year contract
    - Class Offering:
        - Standard, up to 75% (cannot change RI attributes)
        - Convertible, up to 54% (allows to change RI attributes)
        - Scheduled (reserve for time periods, eg. Once a week for few hours)
    - Payment Options: All Upfront, Partial Upfront, No Upfront
- Can be shared between multiple accounts in organization
- Unused RI can be sold in marketplace
- For steady-state, predictable usage or when requiring reserved capacity

Spot Instances

- Use unused AWS compute capacity of idle servers
- Discount of 90%
- Can be terminated by AWS if compute is needed
- AWS Batch for easy and convenient way
- For applications with flexible start and end times

Dedicated Host Instances

- Single-Tenant instances
- Offered in On-demand or Reserved
- For regulatory requirements with strict server-bound licensing

Amazon Machine Image (AMI)

- Template to configure new instances
- You can turn EC2 instances into AMIs
    - create AMI from running or stopped existing EC2 instance
- Contains information: Root volume, Launch permissions, block device mapping
- AMIs have AMI ID which is region specific
- If you want to use AMI from other Region you need to Copy the AMI

Use Case

- Keep incremental changes to OS, apps, packages
- Using Systems Manager Automation, patch AMIs and back those AMIs
- Launch Configurations use AMIs to roll out updates to multiple instances

AMI Marketplace

- Purchase subscriptions to vendor maintained AMIs (like security hardened or special AMIs)
- Community AMI are free and maintained by community

Choosing an AMI

- AMI can be selected based on:
    - Region
    - OS
    - Architecture (32/64 bit)
    - Launch permissions, Root Device Volume (Instance Store or EB)

Auto Scaling Groups (ASG)

- Set scaling rules to automatically launch/shut down EC2 instances
- Contains collection of EC2 instances treated as group
- Scaling can occur via: Capacity Settings, Health Check Replacements, Scaling Policies
- Scale in: Removing Instances
- Scale out: Adding instances

Capacity Settings

- Based on Min, Max and desired Capacity
- ASG will always launch instances to meet min capacity

Health Check Replacements

- EC2 Health Check:
    - health checks on EC2 instances to detect software or hardware issues (status checks)
    - Unhealthy instances will be terminated and new instance will be launched
- ELB Health Check:
    - Health checks by pinging HTTP(S) endpoint with expected response

Scaling Policies

- Target Tracking Scaling Policy
    - Maintains specific metric at target value (e.g. average CPU)
- Simple Scaling Policy
    - Scales when alarm is breached (legacy scaling policy)
- Scaling policies with steps
    - Scales when alarm is breached
    - can escalate based on alarm value changing

ELB integration

- ASG can be associated with ELB for richer health checks
- Classic Load Balancers are associated directly to ASG
- Application and Network Load Balancers are associated indirectly via target groups

Launch Configuration

- Instance configuration template to launch EC2 instances
- Same process as Launching EC2 instance except it's executed when needed for later
- Launch configurations cannot be edited

Elastic Load Balancer (ELB)

- Distributes traffic across multiple targets
- Can be physical hardware or virtual software
- Balancing Load via different rules based on type of Load Balancer

Rules of Traffic

- Listeners:
  - Incoming traffic is evaluated against listeners
  - Evaluate any traffic for matching Port
- Rules: Listeners invoke rules to decide what to to with traffic
- Target Groups: EC2 instances as targets in groups
- For ALB and NLB:
  - traffic sent to listeners → Port matches → Rules forward traffic to target group → target group distributes traffic to instances
- For CLB:
  - traffic sent to listeners → Port matches → distributes traffic to instances

Application Load Balancer (ALB)

- Designed to balance HTTP(S) Traffic
- Operate at Application Layer (7)
- Request Routing: feature allowing to add rules to listeners based on HTTP protocol
- WAF can be attached
- Great for web applications

Network Load Balancer (NLB)

- Designed to balance TCP/UDP Traffic
- Operate at Transport Layer (4)
- Can handle millions of requests per second
- Can perform Cross-Zone Load Balancing
- Great for Multiplayer Video Games or critical network performance required

Classic Load Balancer (CLB)

- Balance HTTP(S) or TCP traffic
- Can use Layer 7-specific features, like sticky sessions
- Can use strict Layer 4 balancing
- Can perform Cross-Zone Load Balancing
- Not recommended to use (legacy)

Sticky Sessions

- Advanced load balancing method allowing to bin users session to specific EC2 instance
- Ensures all request are sent to same instance
- Typically with CLB, also available for target groups of ALB
- Useful when information stored locally on single isntance

X-Forwarded-For Header (XFF)

- Command method for identifying originating IP address of client through proxy or Load balancer

Health Checks

- Report if Instances are InService or OutofService
- Communicate directly with instances
- ELB does not terminate unhealthy instances, will just redirect traffic

Cross-Zone Load Balancing

- Only for CLB and NLB
- Request are distributed evenly across the instances in all enabled AZs

Request Routing

- Apply rules to incoming requests and then forward or redirect
- Host header, Http header, Source IP, Http header method, Path, Query string

Elastic File System (EFS)

- Scalable, elastic NFS file system
- Multiple EC2 instances can mount single EFS volume in same VPC
- Storage capacity grows and shrinks automatically
- 0.30$ GB/month

Elastic Block Store (EBS)

- Highly Available for persistent block storage: Virtual Hard Drive
- Volumes are replicated within AZs
- IOPS: Input/Output per Second → speed of non-contiguous reads and writes on storage
- Throughput: data transfer rate to and from the storage
- Bandwidth: measurement of total possible speed of data movement on the network

Types

- General Purpose (SSD):
  - Balance of prize and performance (Most workloads)
  - 1 GiB -16 TiB, IOPS: 16,000
- Provisioned IOPS (SSD):
  - Highest performance for critical workloads (Large DBs)
  - 4GB – 16 TiB
  - IOPS: 64,000
- Throughput Optimized HDD:
  - Low cost magnetic drive for quick throughput (Data Warehouse, Big Data)
  - 500 GiB – 15 TiB
  - IOPS: 500
- Cold HDD:
  - Lowest cost for infrequent access (File storage)
  - 500 GiB -15 TiB
  - IOPS: 250

Hard Disk Drive HDD

- Magnetic storage with rotating platters and magnetic head as reader
- Good at consistent writing
- Better for Throughput but has physical Parts

Solid State Drive (SSD)

- Integrated circuit assemblies as memory for persistent storage
- Quicker access times and lower latency → Very good I/O
- Physically more resistant

Moving Volumes

- From one AZ to another: Snapshot → AMI → Launch EC2 in desired AZ
- From one Region to another: Snapshot → AMI → copy AMI to desired Region → Launch EC2

Encrypted Root volumes

- Encryption on creation is easy
- Encrypt existing: Snapshot → Copy Snapshot with Encryption Option → Create AMI from encrypted Snapshot → Launch EC2

EBS vs Instance Store Volumes

- EBS:
    - Durable, block level storage that can be attached to single EC2
    - Can start and stop instances, data will persist on reboot
- Instance Store Volumes:
    - Temporary storage on physically attached disks to host machine
    - Created from template stored in S3
    - Cannot stop instances, only terminate
    - Data will be lost when host is shutdown

CloudFront

- Content Distribution Network (CDN) that creates cached copies at Edge Locations
- Delivers web pages/content to users based on location, webpage origin and content delivery server
- Can be used to deliver entire website to nearest possible Edge Location

Core Components

- Origin: Original Files are located here in EC2, S3, ELB, Route53
- Distribution: Collection of Edge locations, defines how content should behave
- Edge Location: Location where content will be cached, different than Region or AZ

Distributions

- Replicates copies based on Price Class
- Behaviors: Redirected to HTTPS, Restrict HTTP Methods, Access, TTLs
- Invalidations: Manually invalidate cache on specific files
- Error Pages: serve custom error pages
- Restrictions: Geo Restriction to blacklist/whitelist countries
- Web distribution or RTMP (for streaming)

Lambda@Edge

- To override behavior of requests and responses
- 4 Available Functions:
    - Viewer request: when CloudFront receives request from a viewer
    - Origin request: before CloudFront forwards to the origin
    - Origin response: when CloudFront receives request from the origin
    - Viewer response: before CloudFront returns response to viewer

Protection

- By Default allows everyone to have access
- Origin Identity Access (OAI): virtual user identity to give Distribution permission to fetch private objects
    - Needed to use Signed URLs or Signed Cookies
    - Signed URLs: provides temp access to cached objects
    - Signed Cookies: cookie passing along with requests

Relational Database Service (RDS)

- Managed relational database service
- Supports: Amazon Aurora, MySQL, MariaDB, PostgreSQL, Oracle, MSSQL

Encryption

- Encryption at rest possible for all engines, handled by KMS

Backups

- Automated Backups:
  - Choose Retention Period between 1 and 35 days
  - Stores transaction logs in S3
  - Enabled by default
- Manual Snapshots: Taken manually by user, persist even if deleting RDS instance

Restoring Backups

- When recovering AWS takes most recent backup
- Never restored overtop of existing instance → creates new instance with new DNS

Multi-AZ

- Ensures DB remains available if AZ becomes unavailable
- Makes exact copy of DB in another AZ and automatically synchronizes changes
- Automatic Failover protection if AZ goes down and slave will be promoted

Read Replicas

- Allow to run multiple copies of DB, only allowing reads
- Asynchronous replication between primary RDS and replicas
- Up to 5 Replicas with own DNS
- Can be in other AZs or regions too
- No automatic failover if primary RDS fails

Multi-AZ vs. Read Replicas

Multi-AZ Deployments	**Read Replicas**
Synchronous replication – highly durable	Asynchronous replication – highly scalable
Only database engine on primary instance is active	All read replicas are accessible and can be used for read scaling
Automated backups from standby	No backups by default
Always span two AZs in a region	Can be within AZ, Cross-AZs or Cross-Region
DB engine version upgrades on primary	DB engine version upgrade independent
Automatic failover	Manually promoting to standalone instance

Aurora

- Fully managed Postgres or MySQL compatible DB designed to default scaling
- Combines speed and availability of high-end-db with simplicity and cost-effectiveness of open source databases (1/10 cost of similar)

Scaling

- Start with 10GB storage and scale in up to 64TB (auto scaling)
- Compute scales up to 32 vCPUs and 248 GB memory

Availability

- Minimum of 3 AZs, each with 2 copies → 6 copies

Fault Tolerance & Durability

- Backup and Failover handled automatically
- Snapshots can be shared
- Self-healing data blocks and disks

Replicas

- Amazon Aurora Replicas:
  - Up to 15, asynchronous
  - low performance impact
  - no data loss as failover (automatic failover)
- MySQL Read Replicas:
  - Up to 5, asynchronous
  - high performance impact
  - data loss as failover (no automatic failover)
  - supports user-defined replication delay and different data or schema vs primary

Serverless

- DB will automatically start up, shut down and scale up or down capacity
- Pay for db storage, capacity, and I/O your database consumes
- For: low volume apps

Redshift

- Fully-Managed Petabyte-size Data Warehouse
- Data Warehouse:
    - Built to store large quantities of historical data for fast complex queries across all the data
- Pricing from 0.25 per hour up to petabytes for 1000$ per terabyte per year
- 1/10 of cost of similar services
- Used for Business Intelligence
- Uses Online Analytics Processing System (OLAP)
- Is Columnar Storage Database:
    - Stores data together data s columns instead of rows
    - Reduces overall disk I/O and amount of data needed to be loaded from disk
- Connect with JDBC or ODBC
- Single Node: Node in size of 160 GB
- Multi Node: Leader Nodes and up to 128 compute nodes
- Dense Compute Node: high performance, less storage
- Dense Storage Node: clusters which you have lot of data
- Uses multiple compression techniques
- Similar data stored sequential on disk, not requiring indexes
- Uses Massively Parallel Processing (MPP)
- Automatically distributes data
- Backups enabled by default with 1 day retention (up to 35 days) stored in S3
- Attempts to maintain 3 copies
- Compute Node Hours only for leader node hours
- Data-intransit and Data-at-rest Encryption
- Single-AZ Service(Snapshots can be restored to different AZ)

Dynamo DB

- Key-value and document DB (NoSQL)
  - Key value: Key with value, nothing more
  - Document store: Nested data structure
- Fully managed, Multiregion, Durable, Built-in security, Backup and restore, caching
- Eventual Consistent Reads (default) or Strongly Consistent Reads
- Just specify read and write capacity at whatever you need and want to pay
- Data Stored on SSD spread across 3 Regions

Table Structure

- Items with attributes and Primary Key

Consistent Reads

- Eventual Consistency Reads:
  - When copies are updated its possible to read an inconsistent copy
  - All data become generally consistent within a second
- Strongly Consistent Reads
  - Only Read until all copies are updated
  - Slower Reads but always correct data

CloudFormation

- Templating Language to define and provision AWS resources
- Infrastructure as Code: process of managing and provisioning infrastructure through definition files (YAML, JSON)

Template Formats

- JSON
- YAML

Template Anatomy

- MetaData
- Description
- Parameters (Values to pass to template at runtime)
- Mappings (Maps Keys to values)
- Conditions (If-Else)
- Transform (Applies Macros)
- Resources (Resource you want to create, requires at least one)
- Outputs (Return values)

AWS QuickStarts

- Collection of pre-built CloudFormation templates

CloudWatch

- Collection of monitoring services

CloudWatch Logs

- Monitor, store and access log files
- Log Group: collection of logs
- Log Stream: Log in Log Group
- By default logs are kept indefinitely and never expire

CloudWatch Metrics

- Time-ordered set of data points, variable to monitor
- Many predefined metrics (CPU, DiskRead/Write,NetworkIn/Out, ….)

CloudWatch Events

- Trigger and event based on a condition or schedule(like crontab)

Custom Metrics

- Using AWS CLI or SDK you can create and publish own metrics
- High Resolution Metrics: lets you track under 1 minute down to 1 second (costs more)

CloudWatch Alarms

- Triggers notification based when metric threshold is breached

CloudWatch Dashboards

- Create custom dashboards from CloudWatch metrics

Availability

- Collecting of data varies on services
    - EC2: 5 minute interval, with detailed monitoring 1 min
    - Most services are 1 minute by default

Agent & Host Level Metrics

- Script that can be installed with Systems Manager
- Gathers more detailed metrics on EC2 instances (default installed on Amazon Linux)

CloudTrail

- Logs API calls between AWS services
- Enables governance, compliance, operational auditing and risk auditing
- Easily identify which users and accounts made the call to AWS
- Where: Source IP
- When: Event Time
- Who: User, UserAgent
- What: Region, Resource, Action

Event History

- Logging by default and will collect for last 90 days
    - For more you need to create a Trail
    - Trails are output to S3
    - Amazon Athena to analyze Trails

Trail Options

- Can be set to log to all regions or not
- Can be set to across all accounts of Organization
- Logs can be Encrypted with SSE-KMS
- Log File Validation to ensure Integrity of Logs

CloudTrail to CloudWatch

- Can be set to deliver events to CloudWatch Logs

Management vs Data Events

- Management Events:
    - Turned on by default and cannot be turned of
    - Configuring security
    - Registering devices
    - Configuring routing rules
    - Setting up logging
    - Etc.
- Data events:
    - Turned off by default (high volume)
    - S3 or Lambda can be tracked

Lambda

- Run Code without provisioning or managing servers → Serverless Functions
- Executes code only when needed
- Pay for compute/invocation
- Ruby, Python, Java, Go, Powershell, NodeJS, C#

Use Cases

- Glue different services together
- Processing Thumbnails, Contact Email Form

Triggers

- Can be invoked via AWS SDK or from different AWS Services
    - API Gateway, AWS IoT, Alexa, Dynamo, S3 …
    - Also with Partner event sources through EventBridge

Pricing

- First 1 million requests per month are free, 0.20$ per 1 million requests afterwards
- 400,000 GB seconds free per month, thereafter very small price per GB second
    - Varies on amount of memory you allocate

Defaults and Limits

- 1000 Lambda running concurrently (Ask AWS Support for increase)
- /tmp directory contains up to 500MB
- Run in no VPC, you can set one but lambda will lose internet access
- Timeout to be maximum of 15 minutes
- Memory can be set between 128MB to 3008 MB

Cold Starts

- Delay which happens if AWS triggers lambda on server which is shut down
- Warm Server: Server which was already running → no delay
- Cold starts can cause delays in User Experience
- There a strategies like Pre Warming which keeps servers continuously running

Simple Queue Service (SQS)

- Fully managed queuing services
- Type of messaging system providing asynchronous communication
- Messages will be deleted once they are consumed
- Messages have to be pulled, not reactive
- For Application Integration
- Queue is temporary repository for messages that are waiting to be processed

Limits & Retention

- Message size: 1byte – 256 KB
- Amazon SQS Extended Client Library for Java: send messages 256KB to 2GB
- Message Retention: default 4 days, can be adjusted from 60sec to 40 days

Queue Types

- Standard Queue: nearly-unlimited number of transactions per second
    - Guarantees that message will be delivered AT LEAST once
    - More than one copy could be possibly delivered out of order
    - Provides best-effort ordering
- SQS First-In-First-Out: support multiple ordered message groups
    - Limited to 300 transactions per secon

Visibility Timeout

- Period of time that messages are invisible in SQS queue after reader picks up message
- Messages will be deleted after job has processed
- If NOT processed and visibility timeout is over, message will be delivered twice

Short vs Long Polling

- Short polling (default):
    - returns messages immediately even if queue is pulled empty
    - when you need message right away
- Long polling:
    - Waits until message arrives in queue or long poll timeout expires
    - Makes it inexpensive to retrieve messages as soon as they are available
    - Will reduce cost
    - Most use-cases should use it

Simple Notification Service (SNS)

- Subscribe and send notifications via email, webhooks, lambda, mobile, etc.
- To decouple microservices, distributed systems and serverless applications
- Publish-subscribe pattern
- Publisher send messages to event bus, categorizing into groups, receivers subscribe to these groups
- New messages are delivered immediately
- Publishers have no knowledge of subscribers

Topics

- Allow to group subscriptions together
- Able to deliver multiple protocols at once (mail, https. Mobile, etc.)
    - Messages will automatically formatted
- Can be encrypted with KMS

Subscriptions

- Can only subscribe to one protocol and one topic
- Protocols: HTTP(s), Email, Email-JSON, Amazon SQS, AWS Lambda, SNS, Platform application endpoints

Application As Subscriber

- Send push notifications directly to mobile apps

ElastiCache

- Managed caching service
- Caching: storing data in cache (tempory storage area), optimized for fast retrieval
- In-Memory Data Store: data stored in RAM → high volatility (fast, can be lost)
- Deploy run and scale popular open source in-memory data stores (Redis, Memcache)
- Only accessible to resource in same VPC

Cache Comparison

- Memcached:
  - Preffered for HTML fragments
  - Simple key-value store
  - Really simple and fast
- Redis:
  - Perform different operations on data
  - Good for leaderboards, keeping track of notification data
  - Fast but not as fast as Memcached

High Availability (HA)

- Run instances in Multi-AZ with help of ELB
- Run instances in other Regions with help of Route53
- Use Auto Scaling Groups to in-/decrease amount of instances
- Use Auto Scaling Groups to increase amount of instances
- Use CloudFront to cache static content for faster delivery on EdgeLocations

Scale Up and Scale Out

- Scale Up (Vertical):
  - Increase size of instances
  - Simpler to manage
  - Lower availability
- Scale Out (Horizontal)
  - Adding more instances
  - More complexity to manage
  - Higher Availability

Elastic Beanstalk

- Quickly deploy and manage web-apps without worrying about infrastructure
- Heroku of AWS
- Choose Platform, upload code and it runs without infrastructure knowledge
- Not Recommended for Production applications
- Powered by a CloudFormation template which setups ELB, AutoscalingGroups, RDS, EC2 instances, Monitoring, Security, Dockerized

API Gateway

- Fully managed service to create, publish, monitor, maintain and secure APIs at any scale
- APIs act as front door for applications

Key Features

- Handles all tasks involving in accepting and processing hundreds of thousands API calls
- Track and control any usage
- Expose HTTPS endpoints for RESTful APIs
- Highly scalable (automatically)
- Maintain multiple API versions

Configuration Part

- Resources: are the urls you define and can have child resources
- Methods: needed on Resources (eg. GET, POST)
- Stages: Versions of your API
- Invoke URL: For each stage AWS provides invoke URL for API calls
- Deploy API: For every change you need to deploy the API in chosen stage
- Integration Type: Lambda, HHTP, Mock, AWS Service, VPC Link

Caching

- Caching can be enabled to cache endpoint response to API for specified TTL
- API Gateway looks and responds from cache
- Reduce number of calls and latency

Cross-Origin Resource Sharing (CORS)

- Way that the server at the other end can relax a same-origin policy
- Allows restricted resources on webpage to be requested from different domain
- Should be enabled if using JS/AJAX
- Disabled by default

Same Origin Policy

- Concept in application security model where browser permits scripts contained in a first web page to access data in a second webpage
- Same origin Policies are used to help prevent Cross-Site Scripting

Kinesis

- Scalable and durable real-time data streaming service
- Collecting, processing, analyzing streaming data

Data Streams

- Pay per running shard
- Multiple consumers(Redshift, Dynamo, S3, EMR) possible
- Data can persist 24 hours(default) to 168 hours

Firehose

- Pay only for ingested data
- Choose one consumer from predefined list
- Data immediately disappers once consumed
- Convert, compress, secure incoming data

Video Streams

- Ingest video and audio encoded data
- Output data to ML or video processing services

Data Analytics

- Specific Data Streams or Firehose as input and output
- Data passing through can be analyzed in real-time through custom SQL

Storage Gateway

- Extending, backing up on-premise storage to the cloud
- Connect on-prem software with cloud-based storage
- Available as VM image (ESXi or Hyper-V)

File Gateway

- Uses NFS or SMB
- Ownership, permissions, timestamps stored in S3 metadata
- Once transferred, files can be managed as native S3 objects

Volume Gateway

- Uses iSCSI
- Data can be backed up as point-in-time snapshots and stored in AWS EBS Snapshots
    - Incremental and compressed snapshots
- Stored Volumes:
    - Primary data stored locally, backup on AWS
    - 1GB to 16 TB size
- Cached Volumes:
    - Primary Data on AWS S3, local data is used as cache
    - 1GB to 32 GB size

Tape Gateway (VTL)

- To archive data in AWS
- Store virtual tape cartridges on tape gateway