EVERYTHING YOU NEED TO KNOW

for the

AWS CERTIFIED SOLUTIONS ARCHITECT ASSOCIATE EXAM

By

Rishie Bothra

AWS Certified Solutions Architect Associate

https://www.linkedin.com/in/rishie-bothra/

http://rishie.ueuo.com/

STORAGE

1. **Simple Storage Service (S3)**
    a. Storage Type: Object Storage; No actual folder hierarchy, uses prefixes and delimiters
    b. Storage Classes
        i. **Standard**
            1. Min 3 AZ Availability (99.99%)
            2. For immediate data retrieval
        ii. **Reduced Redundancy**
            1. Availability 99.9%
            2. Least tolerable to failure
            3. For noncritical, reproducible data at lower levels of redundancy
        iii. **Standard – Infrequent Access** (3AZ)
        iv. **One zone - Infrequent Access** (1AZ 99.5% Availability)
        v. **Intelligent Tiering**
            1. Optimize costs by automatically moving data to the most cost-effective access tier, without performance impact or operational overhead.
            2. Store objects in two access tiers: one tier that is optimized for frequent access and another lower-cost tier that is optimized for infrequent access
            3. Monitors access patterns
                a. Object that is not accessed for 30 consecutive days moved to IA.
                b. Once accessed, object is immediately moved from IA to FA.
        vi. **Glacier** (Vault = Bucket, Archive = Object, Vault Lock = Encryption)
            1. Two ways to use:
                a. Set up vaults and build Glacier yourself (1 account can have a max of 1000 Vaults)
                b. Use it as a storage class for S3
            2. Performance
                a. Durability of 11 9s
                b. Availability of 99.99%
                c. Data resilient in one AZ destruction
            3. Upload
                a. You specify the region
                b. Data is stored with AES-256 Encryption
                c. Contents of the archive cannot be modified after uploading
                d. Uploading archives is synchronous and downloading is asynchronous
                e. Glacier must complete a job before you can get its output
            4. Retrieval
                a. Glacier objects are visible through S3 only.
                b. Retrieved data is available for 24 hours by default (can be changed)
                c. Settings:

i. Free tier: Up to 5% data/month (no rollover).
ii. Maximum retrieval limit can be set
iii. No retrieval limit option

d. There is a charge if you delete data within 90 days
e. Retrieved data will not be encrypted if uploaded unencrypted

5. Retrieval times
    a. **Expedited retrieval** is 1 – 5 minutes
    b. **Standard retrieval** is 3-5 hours
    c. **Bulk retrievals** for petabytes of data in approx 5 – 12 hours

vii. **Glacier Deep Archive**
1. Lowest cost of all S3 classes
2. Min 3 AZ availability
3. Minimum storage duration of 180 days.
4. Data can be restored within 12 hours

c. Features:

i. **Object Lifecycle Management**
1. Used for changing storage classes and saving costs.
2. Just specify the storage class to move the object to after *n* days.

ii. **Versioning**
1. Maintain multiple versions of object in bucket
2. By default, it is not turned on.

iii. **Multi-factor Authentication delete**

iv. **Cross-region Replication**
1. Once enabled, does not replicate what you already have in the bucket
2. Only replicates new stuff that goes into the bucket.

v. **Range GETs (Byte-range fetches)**
1. Use Range HTTP header in a GET Object request to fetch a byte-range from an object, transferring only the specified portion.
2. Accepts concurrent connections to S3 to fetch different byte ranges from within the same object.
3. For higher aggregate throughput versus a single whole-object request.
4. Improve retry times and handle service interruptions better.

vi. **Logging with Event Notifications**
1. Log actions on S3 bucket and use event notifications to be notified.
2. Can use event notifications to invoke Lambda functions

vii. **Object Level Logging using CloudTrail**

viii. **Requester Pay:** User requesting the file pays and not the company.

ix. Object and bucket level permissions.

x. **CORS** (Cross-origin Resource Sharing)
1. Allows client web application in one domain to interact with S3 bucket in another domain.

d. Valid bucket URLs

i. https://<bucket>.s3.amazonaws.com/<object> (virtual host style addressing)

ii. **https://s3.<region>.amazonaws.com/<bucket>/<object>**

e. S3 querying
    1. **Redshift Spectrum:** complex queries with large number of data lake users
    2. **Amazon Athena**: ad hoc SQL querying
f. Consistency: Only eventual consistency.
g. S3 Access
    a. In IAM policy, you can grant programmatic access or Management Console access to Amazon S3 resources
    b. To grant access to users in a specific folder in S3 bucket and no one else's
        a. Create an IAM Policy that applies folder level permissions
        b. Create an IAM Group, attach the IAM Policy and add users to the group
    c. To allow another AWS account to upload objects to your bucket
        a. Use **resource based S3 bucket policy**
        b. Use a conditional statement to ensure that full control permissions are granted to a specific account identified by an ID (e.g. email address)
h. S3 Content Upload & Delivery
    i. **S3 REST API**
        1. Representational State Transfer
        2. Used for uploading objects programmatically
        3. Maps HTTP methods to CRUD (create – PUT/POST, read - GET, update – POST/PUT, delete - DELETE) operations
    ii. **Pre-signed URL**
        1. Upload the object without having any AWS security credentials/permissions.
        2. Bypasses the web server avoiding any performance bottlenecks
        3. Can be generated programmatically
        4. Anyone with a valid pre-signed URL can then programmatically upload an object.
    iii. **Direct Connect**
        1. Use a private connection from enterprise network into AWS
    iv. **Storage Gateway**
    v. **Kinesis Firehose**
        1. To store large amount of analytical data in S3.
    vi. **CloudFront distribution:**
        1. To serve static content using the **HTTPS** (HTTP not allowed) protocol.
        2. Uses an S3 bucket as the origin.
    vii. **S3 Transfer Acceleration**
        1. For long distance file transfer between your client and S3 bucket.
        2. Uses **Amazon CloudFront's** globally distributed **AWS Edge Locations**.
        3. User uploads data at edge location that has an optimized route back to a central S3 bucket.

viii. **Multi part upload**
    1. For objects larger than 100 megabytes
    2. Large files broken down in chunks and assembled once uploaded
    3. Can also use **multipart uploads** with transfer acceleration.
ix. S3 with **Route 53**
    1. **Alias records** (Route 53 specific record type) map resource record sets in your hosted zone to ELBs, CloudFront distributions, Elastic Beanstalk environments, or S3 buckets that are configured as websites.
    2. Create a custom domain name with S3 using a Route 53 Alias record (uses HTTP)
    3. Bucket name must = custom domain name
x. **Snow Family**
    1. For physical transport of data to S3
    2. Types:
        a. **Snowball:** For Petabyte Scale data
        b. **Snowball Edge:** 100 TB local Storage that connects to your facility and has its own instance managing uploads to S3.
        c. **Snowmobile:** For Exabyte scale data

i. <u>S3 Encryption</u>
    i. At rest (objects encrypted after upload, auto decryption when retrieving):
        1. **SSE-S3**: Default. Keys managed by AWS using AES-256
        2. **SSE-KMS** (Server-Side Encryption – Key Management Service)
            a. Auditable master keys can be created, rotated, and disabled from the IAM console.
            b. Keys are managed by AWS.
            c. **Encryption key types:**
                i. AWS **KMS managed keys**
                ii. Customer created - **Customer Master Keys (CMK)**
        3. **SSE-C** (Customer)
            a. Customers keep the encryption keys on premises.
            b. Data is encrypted and decrypted in AWS (server-side) but customers manage the keys outside of AWS.
    ii. In transit
        1. Client-side encryption.
        2. Before uploading data to S3 over HTTPS, encrypt data locally with your own encryption keys.

Situation: Application receives and processes files of size 4GB. Application extracts metadata from files which takes a few seconds. The pattern of updates is highly dynamic with times of little activity and then multiple uploads within a short period of time.

- *Store file in S3 bucket and S3 event notifications to invoke a Lambda function to extract the metadata*
- *The former is more cost efficient that using a Kinesis Data Streams which runs on EC2. Therefore, you will have to provision capacity even when you do not need it.*
- *SQS queues only support max message size of 256KB. You can use the extended client library for Java to use pointers to a payload in S3 but the max payload size is 2GB.*

Situation: Public facing web application hosted on EC2 serves videos directly from an S3 bucket. Need to restrict third parties from directly accessing the videos.

- *Use a bucket policy to only allow referrals from main website URL.*
- *s3:GetObject with a condition with aws:referrer key that the get request must originate from the website only.*
- *Can also use condition statements in bucket policy to restrict access via IP addresses, but it is best practice to use DNS names and URLs instead.*
- *Only giving EC2 instance an IAM role that is authorized to access to S3 bucket won't do it.*
- *You will also have to create a bucket policy giving access to the IAM role.*

Situation: The security team requires an audit trail for operations on the S3 buckets that includes the requester, bucket name, request time, request action, and response status.

- ***Server Access Logging** provides detailed records for the requests that are made to a bucket.*
- *CloudTrail won't work here because it does not audit the bucket operations and only captures IAM/user identity information in logs.*

Situation: You have a private CloudFront distribution that serves files from S3 Bucket and is accessed using signed URLs. User must not be able to bypass the controls provided by CloudFront and access files directly.

- *Create an **origin access identity (OAI)** and associate it with your CloudFront distribution.*
- *Modify the permissions on S3 bucket so that only the origin access identity has read and download permissions.*

2. Elastic Block Storage (EBS)

a. Type: Block level durable storage in EC2 instance; Attached to one instance at a time

b. Options

i. HDD (Magnetic) (Can't be used as Boot Volumes)

1. **Cold HDD (sc1)**
    a. For throughput-oriented storage for large volumes of data that is infrequently accessed.
    b. Lowest cost HDD volume type.
    c. Performance: Up to 250 IOPS/volume (no SLA for IOPS)
2. **Throughput Optimized HDD (st1)**
    a. For streaming workloads requiring consistent, fast throughput at a low price.
    b. Large datasets and high I/O sizes
    c. Examples include MapReduce, Kafka, Big Data warehouses and Log Processing.
    d. Performance: Up to 500 IOPS/volume (no SLAS for IOPS)

ii. SSD (Must use EBS optimized EC2 Instance)

1. **EBS General Purpose SSD (gp2)**
    d. For most workloads including use as system boot volumes, virtual desktops, low-latency interactive apps, and development and test environments.
    e. Performance: 3 IOPS per GB up to 16,000 IOPS.
    f. Volume size: 1 GB to 16 TB
2. **EBS Provisioned IOPS SSD (io1)**
    g. For critical business applications that require sustained IOPS performance such as database workloads.
    h. Performance:
        i. up to 50 IOPS per GiB; 16,000 IOPS – 64,000 IPOS
        ii. >250 MiB/s of throughput per volume
    i. Volume size: 4 GB to 16TB
    j. EBS delivers within 10% of the provisioned IOPS performance 99.9 percent of the time over a given year.

c. Encryption

i. Snapshots of encrypted volumes are encrypted automatically
ii. Data in transit between an instance and an encrypted volume is also encrypted
iii. Not all instance types support encryption but All EBS types and all instance families support encryption.
iv. Use **KMS-managed or customer-managed encryption keys**

d. EBS Backup

i. Snapshots:

1. Only the most recent snapshot needed to create an EBS volume.
2. Snapshots are incremental, but the deletion process will ensure that no data is lost
3. **Snapshots** are manually created**,** or you can use **Amazon Data Lifecycle Manager (Amazon DLM)** to automate the creation, retention, and deletion of snapshots

ii. Facts:

1. Data is replicated across multiple servers within AZ and NOT across AZs
2. EBS volumes are not automatically backed.
3. You need to create manual screenshots that reside in S3.
4. If you terminate an instance, the default **DeleteOnTermination** value on an EBS root volume is set to True. Change to False and attached to volume to another instance to recover it.

Situation: Legacy database on EC2 instance. Data is stored on a 2000GB Amazon EBS (gp2 volume) (SSD). At peak load times, logs show excessive wait time. Persistent storage needed.

- *Migrate the data to EBS provisioned IOPS SSD (io1)*
- *Using an instance store volume will increase performance but the data is not persistent.*

Situation: How to improve database performance while using EBS?

- *Use **EBS Optimized Instances***
- *Use Provisioned IOPS EBS volumes*
- *Create a **Raid 0 array (striping)** where data is written across multiple disks and increased performance but no redundancy.*
- *Raid 1 array (mirroring) is the opposite where we create 2 copies of data to increase redundancy, not performance.*

Situation: Multiple AWS accounts for Dev, Prod and Test. You want to copy an EBS snapshot from Dev to Prod. The snapshot is from EBS volume encrypted with a custom key. How to share?

- *Share the custom key with prod account*
- *Modify permissions on the snapshot to share it with the prod account*
- *Ask prod account to copy the snapshot before creating volumes.*

Situation: Application on EC2 instance that saves user data to an EBS volume. EBS volume was attached to the instance after it was launched and is unencrypted. You would like to encrypt the data however you cannot shut down the instance. What to do?

- *There is no direct way to change the encryption state of an EBS volume*
- *You cannot restore a snapshot of a root volume without incurring downtime.*
- *Create a new encrypted volume, move the data to new volume and delete the old one or,*
- *Take a snapshot of the volume, encrypt it and create a new encrypted volume from the snapshot*

3. **Elastic File System (EFS)**
    a. Type: Hierarchical directory structure via NFSv4 protocol with strong data consistency (can be shared between multiple EC2 instances across AZs, regions, VPCs and on-prem servers)
    b. Modes:
        i. Performance:
            1. **General**
            2. **Max I/O:** Can scale to higher levels of aggregate throughput and operations per second with a tradeoff of slightly higher latencies for file operations.
        ii. Throughput:
            1. Provisioned: You set the limit
            2. Bursting: Scales as needed
    c. Mounting methods:
        i. From local VPC
        ii. From remote VPC via VPC Peering Connection.
        iii. From on-premises servers via **AWS Direct Connect** or **AWS VPN.**
    d. EFS is not supported on EFS instances because windows is a hierarchical directory in itself. EBS + Windows instance is the right combination.

Situation: You are using a file system on EFS which holds home directories for users. Users need to be able to save files to the system.

- *Create a **subdirectory in EFS** for each user, grant read-write-execute permissions.*
- *Mount the subdirectory to the user's home directory.*

Situation: You are using EFS to store sensitive data that will be accessed by multiple EC2 instances. Need to ensure that network traffic is restricted correctly based on firewall rules and access from hosts is restricted by user or group

- ***EFS Security Groups** to control traffic to EFS. Act like firewall*
- *You cannot use IAM to control access to files, IAM is only used for administration.*
- *You can control access to files and directories with POSIX-compliant user and group-level permissions. POSIX permissions allows you to restrict access from hosts by user and group.*

4. **Other Storage Options**
    a. **FSx for Lustre**
        i. Access your S3 objects as files on FSx to run analyses for hours to months.
        ii. Provides a high-performance file system optimized for fast processing of workloads such as ML, high performance computing (HPC), video processing, financial modeling, and electronic design automation (EDA).
    b. **FSx for Windows File Server**
        i. Provides a fully managed native Microsoft Windows file system so you can easily move your Windows-based applications that require shared file storage to AWS.
        ii. This solution integrates with Windows file shares, not with Amazon S3.

5. **Storage Gateway**
    a. To transfer data from on-prem network into cloud or vice-versa
    b. It is a software based Virtual Machine Interface (VMI) that creates a gateway on customer's on prem location.
    c. Downloadable from AWS and installed on local network.
    d. Types:
        i. **File Gateway**
            1. Offload the on-prem data set (most of it still on prem) into S3 whilst retaining a local cache for frequently accessed content on S3
            2. Provides a virtual on-premises file server, which enables you to store and retrieve files as objects in Amazon S3.
            3. Applications that need file storage in S3 for object-based workloads
            4. Offers SMB or NFS-based access to data in Amazon S3 with local caching
        ii. **Volume Gateway in cached volume mode**
            1. The entire dataset is stored on S3 and a cache of the most frequently accessed data is cached on-site
            2. Block-based (not file-based) iSCSI-based solution
            3. You cannot mount the storage with the SMB or NFS protocols
        iii. **Tape Gateway**:
            1. Used for backup with popular backup software like NetBackup, Backup Exec, Veeam etc. (Software + Network Tape Drive)
            2. Each gateway is preconfigured with a media changer and tape drives.
6. **Snowball for S3**
    a. A petabyte scale data transport solution for transferring data in or out of AWS.
    b. It uses a secure storage device for physical transportation
    c. The AWS Snowball Client is software that is installed on a local computer and is used to identify, compress, encrypt, and transfer data.
    d. It uses 256-bit encryption (managed with the AWS KMS) and tamper-resistant enclosures with TPM.
    e. Snowball can import to S3 or export from S3
    f. Snowball cannot be used with multipart upload
    g. You cannot use Snowball for migration between on-premise data center

7. **DataSync**
    a. Used to move large amounts of data online between on-premises storage and S3/EFS.
    b. Eliminates or automatically handle many tasks such as scripting copy jobs, scheduling and monitoring transfers, validating data, and optimizing network utilization.

DATABASE

1. DynamoDB

a. Features:
   i. NoSQL DB - change the schema easily.
   ii. Local secondary index maintains an alternate sort key for a given partition key value, it does not record item level changes
   iii. Good for clickstream data and read-heavy loads.

b. Tools:
   i. **DynamoDB global tables**: a fully managed solution for deploying a multi-region, multi-master database.
   ii. Supports **cross-region replication.**
   iii. When reading data, users can specify whether the read should be eventually consistent or strongly consistent (cost twice than former).
   iv. **DynamoDB Streams:** To keep a list of item level changes that have taken place in the last 24hrs.

c. Performance:
   i. Consistent single-digit millisecond latency.
   ii. DynamoDB + **DAX** further increases performance with response times in microseconds for millions of requests per second for read-heavy workloads.

d. Cost:
   i. More cost effective for read-heavy workloads.
   ii. Read capacity units (RCU) are half the price of write capacity units (WCU).
   iii. Charged based on the provisioned throughput you assign (RCU/WCU) regardless of whether you use it or not.
   iv. Strongly consistent reads are more expensive than eventually consistent reads.

e. Scaling
   i. Auto Scaling: Dynamic provisioned throughput adjustment for actual traffic patterns.
   ii. Push Button Scaling - Only DB that can scale without any downtime.
   iii. Scales horizontally and the mechanism is transparent to consumers.
   iv. Does not scale vertically by adding nodes.
   v. Throttle requests that exceed the provisioned throughput. Failed request is throttled with HTTP 400 code (Bad Request) and ProvisionedThroughputExceeded exception.
   vi. Provisioned capacity pricing model does not automatically scale. Automatically scale only when using the new on-demand capacity mode.

f. Best practices:
   i. Keep item sizes small
   ii. If you are storing serial data in DynamoDB that will require actions based on data/time use separate tables for days, weeks, months
   iii. Store more frequently and less frequently accessed data in separate tables
   iv. If possible, compress larger attribute values

v. Store objects larger than 400KB in S3 and use pointers (S3 Object ID) in DynamoDB

Situation: Application that will read and write data to a database. Deploy the application in 3 different AWS Regions in an active-active configuration. The databases need to replicate to keep information in sync.

- *DynamoDB global tables provide a fully managed solution for deploying a multi-region, multi-master database.*
- *Only solution that provides an active-active configuration*
- *Reads and writes can take place in multiple regions with full bi-directional synchronization.*

Situation: Application that writes data to a DynamoDB table. The client has asked how they can implement a function that runs code in response to item level changes that take place in the DynamoDB table. What would you suggest to the client?

- *If you enable DynamoDB Streams on a table, you can associate the stream ARN with a Lambda function that you write.*
- *Immediately after an item in the table is modified, a new record appears in the table's stream.*
- *Lambda polls the stream and invokes your Lambda function synchronously when it detects new stream records*
- *Event source mapping identifies a poll-based event source for a Lambda function. (Kinesis or DynamoDB stream)*

2. Elasticache

a. In-memory caching for database
b. An Elasticache cluster of caching servers
c. For Online Analytical Processing (OLAP) and not Online Transactional Processing (OLTP)
d. Types:
    i. Memcached:
        1. Only caching, not a data store
        2. Doesn't store data persistently
        3. Doesn't support multi-AZ failover or replication
        4. It is multi-threaded (multiple cores) and scales by adding/removing nodes
    ii. Redis
        1. Can be used as a data store
        2. Stores data persistently
        3. Supports multi AZ failover using read replicas in another AZs (cluster mode)
        4. Scales by shards
e. Networking
    i. Elasticache EC2 nodes can't be accessed from the Internet or by EC2 instances in other VPCs
    ii. A Subnet Group is used by ElastiCache
f. Security: Use **Redis Auth** command to improve data security by requiring the user to enter a password before they are granted to execute Redis commands on a password protected Redis server.
g. You cannot mix Memcached and Redis in a cluster.

Situation: An application to retain information about each user session and have decided to implement a layer within the application architecture to store this information.

- *In-memory key/values store such as Elasticache Redis.*
- ***Sticky Sessions** on an Elastic Load Balancer (ELB) which allow you to route a site user to the particular web server that is managing that individual user's session.*
- *Sticky sessions are enabled at target group level*
- *Session stickiness uses cookies and ensures a client is bound to an individual back-end instance for the duration of the cookie lifetime.*
- *ALB supports load balancer generated cookies only. Cookie name is always AWSALB*

Situation: An application is hosted on the U.S west coast. Users there have no problems, but users on the east coast are experiencing performance issues. The users have reported slow response times with the search bar autocomplete and display of account listings.

- *Elasticache Redis Database*

3. Amazon RDS

a. For OLTP workloads, not OLAP
b. Uses EC2 Instances but you cannot use Auto Scaling Groups on RDS
c. Always returns up-to-date data. Read-after-write consistency
d. **Read Replicas**
    i. **Read replicas** are for workload offloading only and do not provide the ability to write to the database. (Not for Disaster Recovery)
    ii. Min 6 and max 15 replicas
    iii. Can have read replicas of read replicas for MySQL & MariaDB but not PostgreSQL
    iv. You cannot have more than four instances involved in a replication chain
    v. A read replica can be in another region
e. **Multi-AZ RDS**
    i. Replica in another AZ within the same region and synchronously replicates to it. (For DR) (2 copies of DB in each AZ with min 3 AZ)
    ii. During failover RDS automatically updates configuration (including DNS endpoint) to use the second node
    iii. Cannot choose which AZ will be chosen to create the standby DB instance
    iv. In multi-AZ configuration system upgrades are applied first on the standby, before failing over and modifying the other DB Instance. The database is always available with minimal disruption.
    v. Failover times are typically 60-120 seconds
    vi. Read Replicas + Multi-AZ for MySQL and MariaDB. However, PostgreSQL is not currently supported.
    vii. Encryption:
        1. Cannot encrypt an existing RDS and only enable encryption for the master DB by creating a new DB from a snapshot with encryption enabled.
        2. A Read Replica of an Amazon RDS encrypted instance is also encrypted using the same key as the master instance when both are in the same region.
        3. If the master and Read Replica are in different regions, you encrypt using the encryption key for that region.
        4. You can't have an encrypted Read Replica of an unencrypted DB instance or vice versa.
    viii. Maintenance:
        1. Maintenance windows to allow DB instance modifications such as scaling and software patching. Some operations like security patching or OS patching require the DB instance to be taken offline briefly.
        2. Enabling Multi-AZ, promoting a Read Replica and updating DB parameter groups are not events that take place during a maintenance window.

ix. Restoration

1. Restored DBs will always be a new RDS instance with a new DNS endpoint
2. You can restore up to the last 5 minutes
3. You cannot restore from a DB snapshot to an existing DB – a new instance is created when you restore
3. Only default DB parameters and security groups are restored – you must manually associate all other DB parameters and SGs
4. RDS features such as Point-In-Time restore, and snapshot restore require a recoverable storage engine and are supported for the InnoDB storage engine only

Situation: The website currently runs on EC2 instances with one web server instance and one DB instance running (not Amazon RDS) MySQL. You are concerned about the lack of high availability in the current architecture.

- *You cannot use the Amazon RDS features such as read replicas and multi AZ in this case*
- *Migrating to RDS would entail a major change so it will be easier to use the native HA features of MySQL rather than to migrate to RDS.*
- *Install MySQL on an EC2 instance in another AZ and enable replication*

Situation: Requires a highly available database that can deliver an extremely low RPO. Which of the following configurations uses synchronous replication?

- *A Recovery Point Objective (RPO) relates to the amount of data loss that can be allowed, in this case a low RPO means that you need to minimize the amount of data lost so synchronous replication is required*
- *ONLY Amazon RDS in a multi-AZ configuration uses synchronization replication.*
- *RDS Read Replicas use asynchronous replication and are not used for DR.*

Situation: During an application load testing exercise, the Amazon RDS database was seen to cause a performance bottleneck. How to improve performance?

- *Scale up to a larger RDS instance type with more CPU/RAM*
- *Use RDS read replicas to offload read traffic from the master database instance.*
- *Using multi-AZ will not increase performance, only availability*

f. **Amazon Aurora DB (RDS)**
    i. Scale up to 64 TB and Aurora replicas feature millisecond latency.
    ii. **Aurora DB Cluster:**
        1. Consists of a DB instance, compatible with either MySQL or PostgreSQL, and a cluster volume that represents the data for the DB cluster copied across 3 Availability Zones as a single, virtual volume.
        2. The DB cluster contains a primary instance and, *optionally*, up to 15 Aurora Replicas.
    iii. **Cluster volume** manages the data for DB instances in a DB cluster.
    iv. **Aurora Replicas:**
        1. Independent endpoints in an Aurora DB cluster
        2. Best used for scaling read operations and increasing availability.
        3. Up to 15 Aurora Replicas can be distributed across the Availability Zones that a DB cluster spans within an AWS Region.
        4. To increase availability, you can use Aurora Replicas as failover targets. If the primary instance fails, an Aurora Replica is promoted to the primary instance.
    v. **Amazon Aurora Global Database:**
        1. Not suitable for scaling read operations within a region.
        2. It is a new feature in the MySQL-compatible edition of Amazon Aurora, designed for applications with a global footprint.
        3. It allows a single Aurora database to span multiple AWS regions, with fast replication to enable low-latency global reads and disaster recovery from region-wide outages
        4. Uses storage-based replication with typical latency of less than 1 second, using dedicated infrastructure that leaves your database fully available to serve application workloads. In the unlikely event of a regional degradation or outage, one of the secondary regions can be promoted to full read/write capabilities in less than 1 minute
    vi. **Aurora Serverless**:
        1. Does not require you to make capacity decisions upfront as you do not select an instance type.
        2. As a serverless service it will automatically scale as needed.

Situation: You need to scale read operations for your Amazon Aurora DB within a region. To increase availability, you also need to be able to failover if the primary instance fails.

- *Aurora Replicas*

Situation: A new application you are designing will store data in an Amazon Aurora MySQL DB. You are looking for a way to enable inter-region disaster recovery capabilities with fast replication and fast failover. Which of the following options is the BEST solution?

- *Amazon Aurora Global Database*
- *Aurora Replicas would not provide the fast storage replication and fast failover capabilities of the Aurora Global Database and is therefore not the best option*

4. **RedShift**
    a. Uses EC2 Instances
    b. A columnar data warehouse DB that is ideal for running long complex queries.
    c. Improve performance for repeat queries by caching the result and returning the cached result when queries are re-run.
    d. Dashboard, visualization, and business intelligence (BI) tools that execute repeat queries see a significant boost in performance due to result caching.
    e. When provisioning a cluster, you provide the **cluster subnet group** and Amazon Redshift creates the cluster on one of the subnets in the group
    f. Always keeps **three** copies of your data and provides continuous/incremental backups
    g. Single-node clusters do not support data replication
    h. Manual backups are not automatically deleted when you delete a cluster

Situation: An application you manage exports data from a relational database into an S3 bucket. The data analytics team wants to import this data into a RedShift cluster in a VPC in the same account. Due to the data being sensitive the security team has instructed you to ensure that the data traverses the VPC without being routed via the public Internet.

- *Amazon RedShift Enhanced VPC routing forces all COPY and UNLOAD traffic between clusters and data repositories through a VPC.*
- *Implementing an S3 VPC endpoint will allow S3 to be accessed from other AWS services without traversing the public network. Amazon S3 uses the Gateway Endpoint type of VPC endpoint with which a target for a specified route is entered into the VPC route table and used for traffic destined to a supported AWS service.*

5. **Database Migration Service**

Situation: To migrate an Oracle database running on RDS onto Amazon RedShift to improve performance and reduce cost.

- *Convert the data warehouse schema and code from the Oracle database running on RDS using the **AWS Schema Conversion Tool (AWS SCT)***
- *Then migrate data from the Oracle database to Amazon Redshift using the **AWS Database Migration Service (AWS DMS)***

Elastic Cloud Compute (EC2)

1. **Instance Classes** (can be changed on a running instance)
    a. General Purpose (T2, M5, M4 and M3):
        i. Balance of memory and network resources
        ii. Applications:
            1. Basic web hosting
            2. Small database hosting
        iii. T2:
            1. For actual use
            2. Provides burst performance: credits accrue during idle times and can be used when you there is a sudden load
        iv. M5, M4 and M3
            1. For development and staging
            2. Does not support bursting
    b. Compute Optimized (C5, C4 and C3)
        i. For CPU intensive applications
        ii. Applications:
            1. Media coding
            2. Intensive batch jobs
            3. Many concurrent users
            4. Gaming servers
    c. Memory Optimized (X1e, X1, R4 and R3)
        i. For high memory requirements
        ii. Applications:
            1. Large dataset processing
            2. In-memory databases
            3. Big data processing
    d. Storage Optimized (H1, I3 and D2)
        i. For high sequential read/writes to local storage
        ii. Applications:
            1. Relational databases
            2. Data warehousing
            3. Image storage and processing
    e. Advanced Computing (P3, P2, G3 and F1)
        i. For special hardware compute requirements
        ii. Applications:
            1. Graphics processing unit (GPU)
            2. Field programmable gate array (FPGA)
            3. Penetration testing

2. Pricing

a. On-demand
   i. Charge for usage time at a flat rate
   ii. Billed in 60 second increments rounded up. (Example use for 1m 20s, pay for 2m)

b. Reserved
   i. For a long-period usage
   ii. Minimum usage is 1 year
   iii. Fixed contract term of 1 year or 3 years
   iv. Accurate prediction of usage is required because you are charged whether you use it or not
   v. Can be less expensive than on-demand
   vi. You change the instance size within the same instance type
   vii. Instance type modifications are supported for Linux only
   viii. Cannot change the instance size of Windows RIs
   ix. You can sell reservations on the AWS marketplace
   x. Scheduled RI: Reserved for specific periods of time, hourly charges, billed monthly over 1-year term

c. Spot Pricing
   i. You bid on extra compute time for idle AWS instances
   ii. 90% savings as compared to on-demand
   iii. Can only use when instances are made available by AWS
   iv. For non-critical workloads
   v. You can Hibernate, Stop and Terminate a spot instance

d. Service limits per region:
   i. 20 on-demand instances
   ii. 20 reserved instances
   iii. Region specific dynamic spot instances
   iv. 300 TiB of aggregate PIOPS volume storage per region

3. Tenancy Models

a. Shared tenancy
   i. Multiple organizations run their instances on the same physical machine
   ii. Time and space sharing
   iii. You do not get to pick the physical server
   iv. Pros: Reduced cost, simpler deployment
   v. Cons: Lower performance, less control, not ideal for specific government policy and host level compliance

b. Dedicated Hosts
   i. Physical machines dedicated only to you
   ii. Must be explicitly configured
   iii. Not available in free tier
   iv. Pros:
      1. More accurate licensing management
      2. More detailed reporting
      3. Compliance management and server-bound software licenses.
      4. You can determine host placement during instance restarts
   v. Cons: Costs more
   vi. You can Bring Your Own License (BYOL) on these hosts by importing your VM (with license) into EC2 using ImportImageCLI command
   vii. Ideal for migrating from existing windows infrastructure to AWS windows infrastructure
   viii. Price: Per host

c. Dedicated Instances
   i. Hardware dedicated to a single customer
   ii. Auto instance placement so you do not have control over where exactly in the hardware your instance is launched
   iii. Must be explicitly configured
   iv. Not available in free tier
   v. Pros:
      1. Hardware dedicated to the customer
      2. Performance advantage
   vi. Con: Less accurate licensing management than dedicated hosts

4. Amazon Machine Image (AMI)

a. It is like a blueprint with server configuration details
b. Sources
    i. Amazon (free)
    ii. Amazon Marketplace (Other people can sell their AMIs)
    iii. Community (free)
c. AMI Launch permission types
    i. Public: Anyone can launch an instance using the AMI
    ii. Explicit: Only specified individuals are allowed
    iii. Implicit: Only the AMI owner is allowed
d. Creating an AMI
    i. Using existing AMI and customize it
    ii. Build a VM offline using VMWare and import the VM into AWS and save its AMI
    iii. Other public sources
e. Types
    i. Hardware Virtual Machine (HVM AMI)
        1. Virtualizes hardware fully
        2. Requires hardware assisted virtualization
    ii. Paravirtual Machine (PM AMI)
        1. Runs on hosts without specific support for virtualization
        2. Does not perform as good as HVM AMI

5. EC2 Instance Root Volume

a. It is the volume every instance comes with
b. Contains the boot sector than contains the boot loader which launces the OS
c. Types
    i. **Instance-Store Root Volume**:
        1. Cannot be stopped, detached, or reattached to other EC2 instances.
        2. Default deleted upon termination.
        3. Rebooting results in no loss.
        4. On failure, data is lost
        5. Can be only specified during instance launch.
        6. Can't be attached to running instances.
    ii. **EBS Root Volume**
        1. Can be detached and reattached to other EC2 instances.
        2. Default deleted upon termination.
        3. Rebooting results in no loss.
        4. **Block Device Mapping** to specify additional EBS volumes during launch and for running instances
        5. On failure, data is not lost

6. Instance Management

a. **Bootstrapping**:
   i. **User Data**: You specify a script to be run on an instance at launch
   ii. Limited to a size of 16KB
   iii. Can run shell scripts and cloud-init directives
   iv. User data is NOT encrypted
   v. Example: During Linux instance launch, you can run update commands

b. **Run Command** in Systems Manager:
   i. Manage the configuration of existing instances by using remotely executed commands.
   ii. For installing software, running ad hoc scripts or Microsoft PowerShell commands, configuring Windows Update settings.

c. VM Import/Export: Import VM to EC2, save its AMI and launch an EC2 instance from it

d. **Metadata**:
   i. Data that indicates the Instance ID, names of security groups and IAM roles assigned to the instance.
   ii. Metadata is NOT encrypted.
   iii. Ways to extract metadata**:**
      1. **Instance Metadata Query** tool to query the instance metadata without typing out the full URI or category names.
      2. Use Command curl http://169.254.169.254/latest/meta-data

e. You can change the instance type by STOP – CHANGE – RUN

f. Activate **Termination Protection** that helps the instances from accidental deletion

g. Use Instances with **Enhanced Networking** for low latency and high network throughput

Situation: EC2 generating very high packets-per-second and application performance is impacted.

- *Use Enhanced Networking.*
- *Use SR-IOV (single root input/output virtualization) that provides direct access to network adapters for higher performance and low latency.*
- *For this you must launch an HVM AMI.*
- *For enhanced networking EC2 instance must be in a VPC.*

7. Auto Scaling Groups (ASG)

a. Definition
   i. It is a collection of instances with similar characteristics
   ii. Can be scaled based on criteria like network communication or CPU utilization

b. Launch Configuration
   i. A template used to launch an ASG
   ii. It cannot be edited once defined but you can edit the ASGs and attach running instances to ASG.
   iii. You specify:
      1. Relevant metrics like CPU, network throughput, free memory
      2. No of instances
      3. Specify what AZ should the ASG span
      4. Scale up or scale down?
      5. Specify minimum and maximum no. of instances always running

c. Termination Policy (scale out: add instances, scale in: terminate instances)
   i. Check if there are multiple AZs. If yes, select the AZ with the most instances
   ii. Select the instance with the oldest launch configuration and terminate it
   iii. If you have multiple instances using the same launch configuration, you select the instance that is closest to the next billing hour and terminate it
   iv. If there are multiple instances closest to the next billing hour, select a random instance and terminate it
   v. You can always create a custom termination policy.

d. **ASG Health check**
   i. Checks if an instance is unhealthy so it can be terminated
   ii. **Health Check Grace Period** allows a period for a new instance to warm up before performing a health check. (300 seconds by default)
   iii. Termination process:
      1. ASG waits for connection draining to finish. If **connection draining** is enabled, Auto Scaling waits for in-flight requests to complete or timeout before terminating instances.
      1. Unhealthy instance is terminated
      2. New one is launched as a replacement

e. **AZ Rebalancing**
   i. To maintain equal number of instances across AZs in a region
   ii. Process:
      1. Launch new EC2 instances in AZs with fewer instances first
      2. Start terminating instances in AZs that had more instances.
      3. ASGs do not wait for any scaling events, they automatically perform rebalancing.

f. **Placement Groups**
   i. Control how instances are launched.
   ii. Types:
      1. Spread: Instances across underlying hardware across AZs for availability
      2. Cluster: clusters instances for low latency in single AZ

Situation: EC2 instances are not being terminated from an Auto Scaling Group behind an Elastic Load Balancer when traffic volumes are low.

- *Modify lower threshold setting on Auto Scaling Group (ASG).*

Situation: Instances with low proximity, low latency and high network throughput and cost effective:

- *EC2 instances with enhanced networking*
- *Use cluster placement groups*

Situation: App uses ASG with several EC2 instances. App has changed and new AMI must be used for launching new instances.

- *Create a new launch configuration that uses the AMI*
- *Update the ASG to use the new launch configuration.*
- *Launch configurations can't be edited once defined.*

Situation: Attempted to restart an EC2 instance and it went form "pending" to "terminated". Why?

- *Reached your EBS Volume limit or EBS Snapshot is corrupt*
- *Unsupported operation if instance type is not supported by AMI*
- *InsufficientInstanceCapacityError if AWS doesn't have capacity left*
- *InstanceLimitExceededError when you have reached the limit on number of instances per region*

Situation: What does ASG do when EC2 health check is unhealthy or impaired.

- *ASG waits for connection draining, terminates the instance and launches and replacement.*

Situation: What happens when you manually terminate instances in an ASG?

- *It will allow you, but it will launch additional instances to compensate for the ones that were terminated.*

Situation: You need to make sure that one particular EC2 instance is not affected by ASG what will you do?

- *Use Instance Protection Feature or Termination Protection*

Situation: An EC2 instances in ASG is causing ASG to launch new instances based on dynamic scaling policy. Temporary fix?

- *EC2 in Standby State and suspend scaling processes responsible for launching new instances.*

Situation: Multiple scaling events within an hour. How to stabilize?

- *Default value of cooldown period is 300 seconds*
- *Modify the cooldown timers*
- *Modify the CloudWatch alarm evaluation period that triggers your Auto Scaling based on the data points*

Situation: You have to scale up without terminating existing instances

- *Stop each instance, change type and start again*
- *Suspend all auto scaling processes on the ASG until changes are made*

Situation: Want disaster Recovery capability on ASG. Applications are stateless and read and write data into S3 buckets. Current AMI needs to be used as it has some modifications to it. Steps?

- *Copy the AMI to the DR region and create a new launch configuration for the ASG that uses the AMI*
- *Enable Cross Region Replication (CRR) specify a destination bucket in the DR region*

g. Scaling for workloads:
   i. Predictable: **Scheduled Scaling**
   ii. Unpredictable:
      1. **Step Scaling**:
         a. Based on size of alarm breach.
         b. Does not wait for scaling activity or health check replacement or cooldown period to complete.
      2. **Simple Scaling**: Waits for scaling activity or health check replacement or cooldown period to complete.
   iii. Event based:
      1. **Dynamic Scaling**
      2. **Target Tracking Scaling:** Select a scaling metric (such as aggregate CPU utilization %) and set a target value

Situation: Large number of users at the same time and system becomes very slow at 9:00 am?

- *Use Auto Scaling Scheduled Action to scale out resources at 8:30 a.m.*

Situation: Already have 16 instances (ASG1: 8 web servers and ASG2: 6 app servers and 2 database servers) serves in a subnet. Simulated increase of 50% and failed. Why?

- *50% increase is 24*
- *Soft limit of 20 on-demand instances*
- *32 possible hosts in a /27 subnet. AWS uses first 4 and last 1. ELB needs 8 within the subnet. And we are left with only 19.*

8. Instance Security

i. Protection against DDoS attacks:
   1. Use CloudFront for both static and dynamic content. Supports valid HTTPS requests. Also used geo-blocking.
   2. Configure Auto Scaling with a high max number of instances.

ii. Best practices for securing instances:
   1. Disable root API access keys and secret key
   2. Restrict access to instances from limited IP ranges using Security Groups
   3. Password protect the .pem file on user machines
   4. Delete keys from the authorized_keys file on your instances when someone leaves your organization or no longer requires access
   5. Rotate credentials (DB, Access Keys)
   6. Regularly run least privilege checks using IAM user Access Advisor and IAM user Last Used Access Keys
   7. Use bastion hosts to enforce control and visibility
      a. **Bastion hosts**: Used as a server to provide access to a private subnet from internet.
      b. Best practices for setting up Bastion Hosts:
         i. Configure EC2 instances as bastion hosts
         ii. Deployed in public subnets
         iii. Use SSH or RDP protocols to connect to bastion hosts
         iv. Configure a Security Group with relevant permissions to allow SSH or RDP protocols and to restrict IPs and CIDRs that can access the bastion host
         v. Bastion hosts can use auto-assigned public or Elastic IPs.
         vi. Deploy in 2 AZs and use an ASG to make sure that number of hosts match the desired capacity specified

Other Compute Services

1. **Elastic Container Service (ECS)** (Docker service within AWS)
    a. Launch Types
        i. **EC2 Launch type**
            1. Utilizes you own EC2 instances (must use ECS optimized AMI)
            2. You can access OS
        ii. **Fargate Launch type:**
            1. AWS Handles provisioning, configuring and managing the EC2 instances for you
            2. You cannot access OS
            3. To use this launch type, you must have your container images in the **Elastic Container Registry (ECR)** or Docker Hub.
    b. **Task Definition**
        i. Parameters:
            1. Which Docker images to use with the containers in your task
            2. How much CPU and memory to use with each container
            3. Whether containers are linked together in a task
            4. The Docker networking mode to use for the containers in your task
            5. What (if any) ports from the container are mapped to the host container instances
            6. Whether the task should continue if the container finished or fails
            7. The commands the container should run when it is started
            8. Environment variables that should be passed to the container when it starts
            9. Data volumes that should be used with the containers in the task
            10. IAM role the task should use for permissions
        ii. Remember:
            1. You can grant additional permissions to an individual container on ECS cluster without disturbing other container permissions only by creating a new Task Definition for that application container and assigning a new IAM role.
            2. You cannot use existing Task Definition because only one IAM role can be associated with one Task Definition.
    c. Scheduling containers in ECS:
        i. **Service Scheduler**:
            1. Ideally suited for long-running processes.
            2. You create your own schedulers and have single or batch jobs.
            3. This is not 3rd party.
        ii. **Custom Scheduler:**
            1. Use the **StarTask API** operation to place tasks on specific container instances.
            2. Only compatible with **EC2 Launch type**. Not with **Fargate Launch type**

3. You need a custom scheduler to integrate with Blox which is a container management platform.

Situation: Various ECS container instances in your ECS cluster are displaying as disconnected.

- *Create ECS instances using ECS optimized AMI*
- *Verify container agent is running on container instances*
- *Verify they have container agent if using any other compatible AMI*
- *Verify that the IAM instance profile(container for an IAM role that can be assigned to instances at launch) has necessary permissions*
- *Verify Docker daemon (listens for requests) is running on container instances*
- *Verify the Docker Container daemon is running on container instances*

Situation: Currently use IAM roles for assigning permissions to containerized application in ECS. You need more granular permissions.

- *IAM roles for Amazon ECS tasks, you can specify an IAM role that can be used by the containers in a task.*
- *Used with both EC2 and Fargate Launch Types.*

2. **Elastic Beanstalk**
    a. Used to run an application on AWS resources and it will handle the provisioning and management of the underlying resources i.e. creates the entire environment for you.
    b. You still maintain full control of the underlying resources.
    c. You cannot change the environment tier after creating the environment.

3. **Lambda**
    a. Max execution time in 900 seconds (15 minutes)
    b. Stream-based Services: **DynamoDB** and **Kinesis**. Underlying process is Lambda

4. **AWS Batch**: Regular batch jobs with auto provisioning, managing, monitoring, and scaling the computing jobs which utilize a large fleet of EC2 instances.

Situation: Application that uses lambda. Need to store sensitive information like credentials as environment variables within lambda. It has to be secure how?

- *Env variables are nor encrypted during the deployment process, only after the deployment process they are encrypted*
- *We need to encrypt this information before it is invoked by lambda function*
- *Use AWS Key Management Service to store sensitive information as Ciphertext.*

Networking

1. Things to know

a. Ports and Protocols

i. **Port**: Docking point that allows information flow from one device to another.

ii. **Protocol:**

1. A set of rules/procedures for transmitting data between devices.
2. Preexisting agreement as to how the information will be structured and how each side will send and receive it.
3. Only one protocol and one port can be defined per target group

iii. Types of protocols:

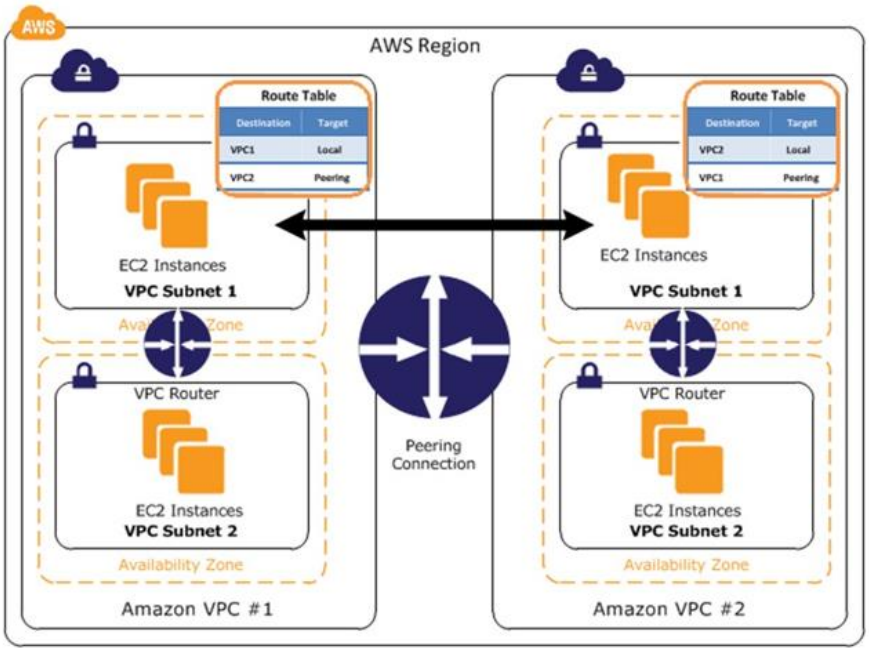
Protocol	Purpose	Port
Transmission Control Protocol (TCP) (L4)	1. Facilitates network messages b/w devices 2. TCP is slower than UDP, but more resilient 3. Retransmission of lost packets is possible	8080
User Datagram Protocol (UDP) (L4)	1. Facilitates network messages b/w devices 2. Faster than TCP, but less resilient 3. Retransmission of lost packets is not possible	1030
Internet Protocol V6 (IPv6)	1. Most recent internet communications protocol. 2. Provides an identification and location system for devices 3. Routes traffic across the Internet. 4. Used by Internet Gateway.	0 - 65535
Internet Protocol V4 (IPv4)	1. Used by NAT Gateway	0 - 65535
HyperText Transfer Protocol (HTTP)	1. Browsers send requests to web servers for web content 2. A stateless protocol which doesn't keep any data. 3. TCP sets up a reliable connection 4. HTTP uses this connection to transfer data	80
Secure Socket Layer (SSL) or Transport Layer Security (TLS)	1. Encrypted link between a server and a browser/mail 2. **HTTPS**: TLS/SSL + HTTP Protocol 3. Any website using HTTPS is employing TLS/SSL encryption 4. **SSL certificate** is a data file hosted in a web server that contains website's identity and public key. 5. Browser will reference to this file to obtain the public key and verify server's identity.	443
File Transfer Protocol (FTP)	1. For file transfer between devices	21
MySQL Protocol	1. Between MySQL clients and MySQL server	3306
Real-time Messaging Protocol (RTMP)	1. For streaming audio, video and data over internet between a flash player and a server	1935
Internet Control Message Protocol (ICMP)	1. Determine if data reaches its intended destination in a timely manner.	-
Proxy Protocol	1. Carries source IP/port info to the destination for which the connection was requested. 2. Only applies to Level 4 Networking Layer. 3. Front end listener can be TCP or SSL and Back end listener must be TCP	8080

Secure Shell Protocol (SSH)	1. For network communications that take place via an unsecured network	22
IPsec VPN Protocol	1. To establish a VPN connection. 2. Modes: a. **Tunnel mode** encrypts an outgoing packet. b. **Transport mode** encrypts end to end communication.	500
SSL VPN Protocol	1. Segmented access for users 2. Only access to a couple services in the entire network. 3. Modes: a. **Portal mode:** Authorization required for access b. **Tunnel mode** Secure web access but also non-web apps	443
Dynamic Host Configuration Protocol (DHCP)	1. Dynamically assigns an IP Address and other network configuration parameters to each device on network	67

2. Virtual Private Cloud (VPC)

a. Definition
   i. A personal data center in cloud.
   ii. Public cloud with security around it.

b. Features
   i. When an AWS account is created, you get a default VPC
   ii. You get only one VPC in one region
   iii. **VPC Peering:**
      1. Used to interconnect VPCs
      2. It is not transitive. VPC A is connected to VPC B and VPC B to VPC C. This does not mean VPC A is connected to VPC C.
      3. Needs a defined route between VPCs (**Route tables**)



c. Components
   i. **Subnets**
      1. Private networks inside a VPC
      2. Types
         a. Public subnet: for public facing applications
         b. Private subnet: for database or backend applications

Situation: The VPC will be used to host a two-tier application that will include Internet facing web servers, and internal-only DB servers. Zonal redundancy is required. How many subnets are required to support this requirement?

- *A public subnet should be used for the Internet-facing web servers and a separate private subnet should be used for the internal-only DB servers.*
- *Therefore, you need 4 subnets – 2 (for redundancy) per public/private subnet*

Situation: A VPC with a public subnet and a VPN-only subnet. The public subnet is associated with a custom route table that has a route to an Internet Gateway. The VPN-only subnet is associated with the main route table and has a route to a virtual private gateway. The Architect has created a new subnet in the VPC and launched an EC2 instance in it. However, the instance cannot connect to the Internet.

- *When you create a new subnet, it is automatically associated with the main route table (automatically comes with your VPC. It controls the routing for all subnets that are not explicitly associated with any other route table). Therefore, the EC2 instance will not have a route to the Internet.*
- *The Architect should associate the new subnet with the custom route table*
- *Subnets are always associated with a route table when created*

ii. **Elastic IP Addresses**
    1. Public IPv4 address (user's numerical identifier), reachable from the internet.
    2. You can mask the failure of an instance by rapidly remapping the address to another instance in your account.
    3. They can be moved between instances in the same region only.
    4. Can be requested from the Amazon pool or provide it yourself.
    5. Once you request it, you are charged for it until you release them even if it is not connected to an instance.

iii. **Elastic Network Interface (ENI)**
    1. An ENIs is where Elastic IP addresses can be used from.
    2. Types:
        a. Standard ENI
        b. **Elastic Network Adapter (ENA)**
            i. For enhanced networking
            ii. High bandwidth and low instance latency
        c. **Elastic Fiber Adapter (EFA)**
            i. ENA + Added capabilities
            ii. For tightly coupled High Performance Computing (HPC) applications
            iii. Used Message Parsing Interface (MPI)
    3. **Dual Homing**: Multiple ENIs can be connected associated with an instance. This allows for offline networking.

Situation: You create second ENI (eth1) when launching an EC2 instance. You would like to terminate the instance. What will happen to the attached ENI?

- *eth1 will persist and eth0 will be terminated. Because eth0 is attached to the instance and is ephemeral.*

iv. **Endpoints**:
    1. Definition:
        a. Used to create a private connection between your VPC and another AWS service without requiring access over the Internet
        b. Can be done through:
            i. NAT device
            ii. VPN connection
            iii. AWS Direct Connect
    2. Facts:
        a. It does not provide VPC to VPC connectivity
        b. One VPC can have multiple VPC Endpoints
    3. Types:
        a. **Interface Endpoint**: Uses an Elastic Network Interface (ENI) in the VPC
            i. **Public VIF:**
                1. Connect to AWS resources that are reachable by a public IP address (such as S3 bucket) or AWS public endpoints
                2. Can reach all AWS public IP addresses globally
                3. You can establish IPSec connections over **Public Virtual Interface (VIF)** to remote regions.
            ii. **Private VIF:**
                1. Connect to your resources hosted in a VPC using their private IP addresses
        b. **Gateway Endpoint**: Configure your route table to point to the endpoint (Used by S3 and DynamoDB)

Situation: Like to implement connectivity between all of the VPCs around the world so that you can provide full access to each other's resources. As you are security conscious you would like to ensure the traffic is encrypted and does not traverse the public Internet. The topology should be many-to-many to enable all VPCs to access the resources in all other VPCs.

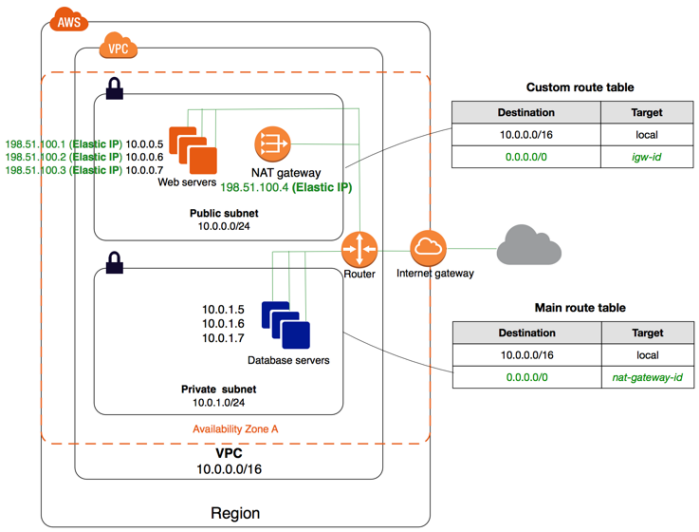
- ***Inter-region VPC peering**:*
    - *Data sent between VPCs in different regions is encrypted (traffic charges apply).*
    - *You must update route tables to configure routing.*
    - *You must also update the inbound and outbound rules for VPC security group to reference security groups in the peered VPC.*
- *Fully meshed topology*
- *A VPC endpoint enables you to privately connect your VPC to supported AWS services and VPC endpoint services, it does not provide full VPC to VPC connectivity.*

v. **Network Address Translation (NAT) (VPC to Internet Connection)**

1. **NAT Instance:**
    a. Allows private instances *outgoing* connectivity to the internet while at the same time *blocking* inbound traffic from the internet.
    b. Not redundant and limited in bandwidth
    c. Scaled manually
    d. Used for multi-homing by implementing a NAT instance on private and public subnet at the same time.
2. **NAT Gateway**:
    a. AWS managed NAT service that uses IPv4 protocol.
    b. Used to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.
    c. Deployed in public subnet
    d. Route tables of the private subnets are configured to forward internet-bound traffic to the NAT Gateway.
    e. Better than NAT Instance as it offers greater availability and bandwidth and requires less configuration and administration.
    f. Cost based on hourly usage and data processing.
    g. Scales automatically up to 45Gbps

vi. **Internet Gateway (VPC to Internet and Vice Versa)**

1. Uses IPv6 protocol.
2. Allows resources within your VPC to access the internet, and vice versa.
3. There needs to be a routing table entry allowing a subnet to access the internet gateway.
4. NAT Gateway only works one way. The internet at large cannot get through your NAT to your private resources unless you explicitly allow it.



Situation: A data processing application pulls large amounts of data from object storage system via the internet. What is the best way to provision internet connectivity? We need redundancy and should not have any constraints on bandwidth.

- *Internet Gateway*

Situation: A VPC has a fleet of EC2 instances running in a private subnet that need to connect to Internet-based hosts using the IPv6 protocol. What needs to be configured to enable this connectivity?

- *Egress-only Internet Gateway. NAT Gateway is used if IPv4 protocol.*

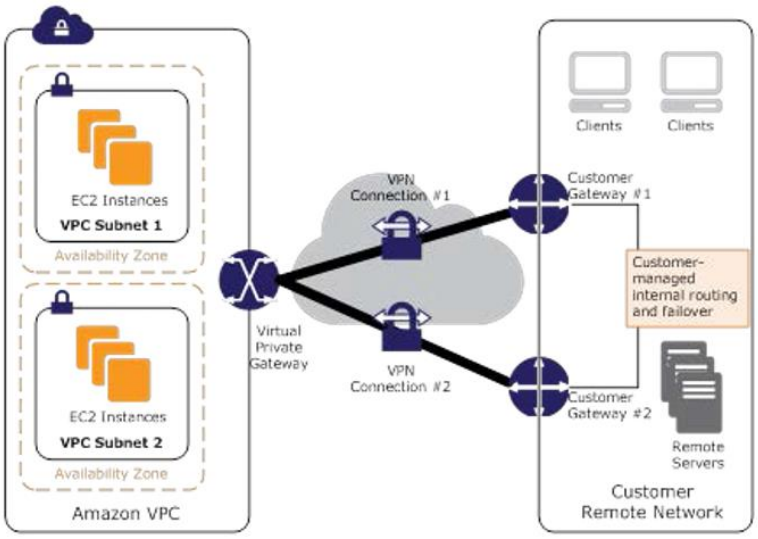
Situation: Need to connect form your office to instance running in public subnet in your VPC using internet (remote connection).

- *Need public or elastic IP on the instance*
- *Need an Internet Gateway (resides in public subnet) attached to the VPC and route table attached to the public subnet pointing to the internet gateway.*
- *Configure SGs and Network ACLs to allow SSH traffic.*

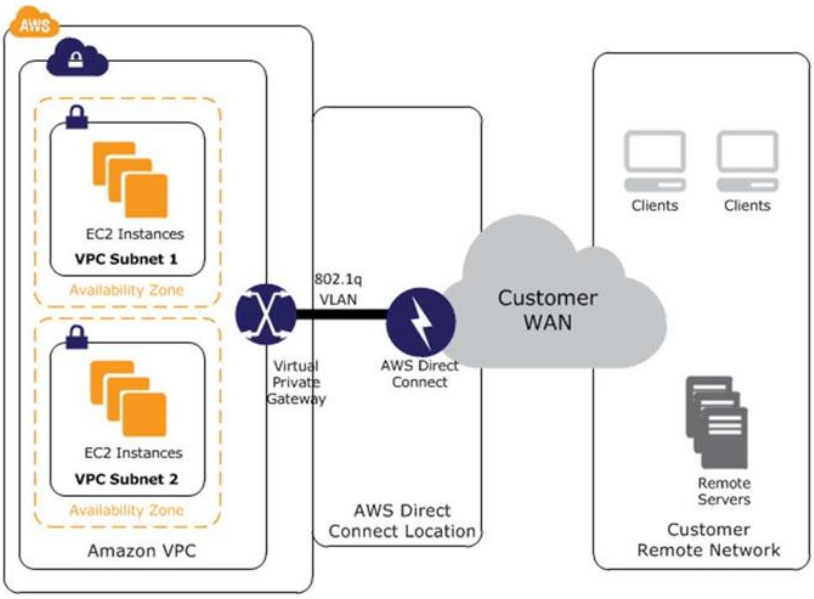
Situation: Cannot connect to EC2 instance in public subnet in your VPC from the internet.
Make sure for public subnet:

- *"Auto-assign public IPv4 address" set to "Yes" which will assign a public IP*
- *The subnet route table has an attached Internet Gateway (entry)*
- *Instance with a security group with an inbound rule allowing the traffic*

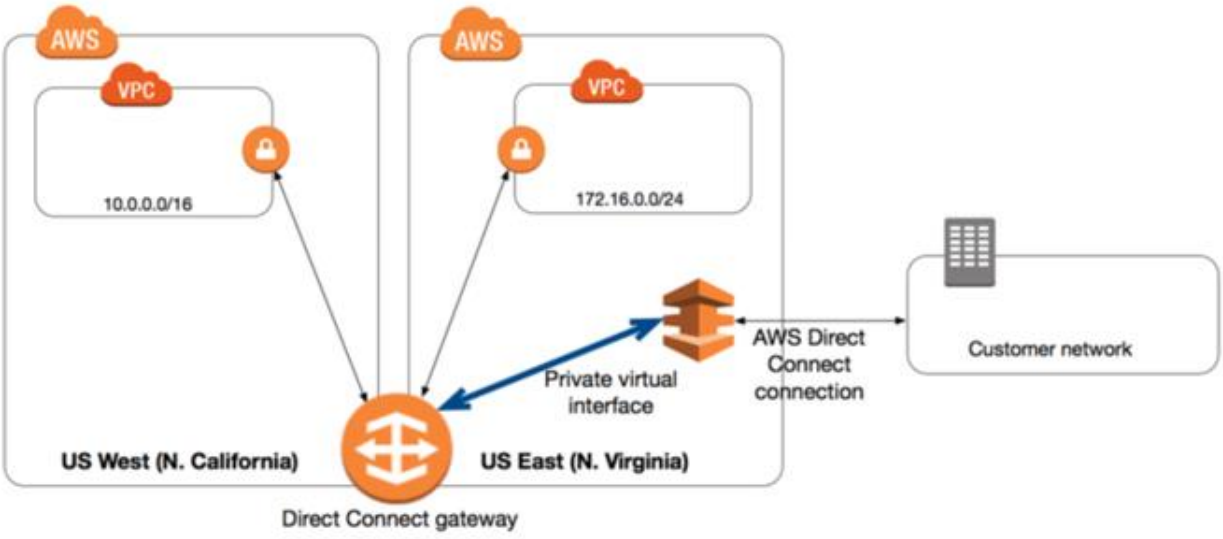
vii. **VPC Connections**
    1. **Virtual Private Network (VPN) Connection (Uses internet) (On-prem to VPC and vice-versa)**
        a. Enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network.
        b. It is like a public network connection but encrypted
        c. Components:
            i. **Virtual Private Gateway:**
                1. Anchor on AWS Side i.e. your VPC
                2. Provides a one-point access for multiple VPN connections
            ii. **Customer Gateway:**
                1. Anchor on your local on-prem network
                2. Can be a physical device or a software



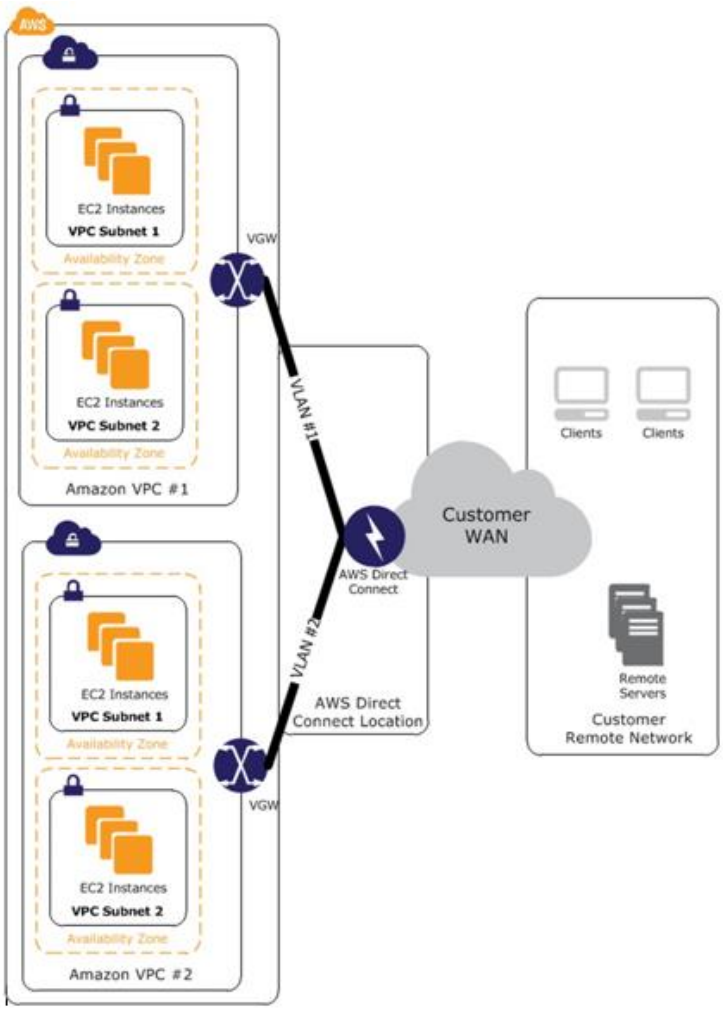
2. **Direct Connect (Does not use internet)**
    a. Once you have connected to an AWS region using AWS Direct Connect you can connect to all AZs within that region
    b. Local on-prem network to VPC
        i. Secure, reliable, and private connection
        ii. Bypasses internet and removes network congestion
        iii. Lead times are often longer than 1 month so it cannot be used to migrate data within a smaller timeframe



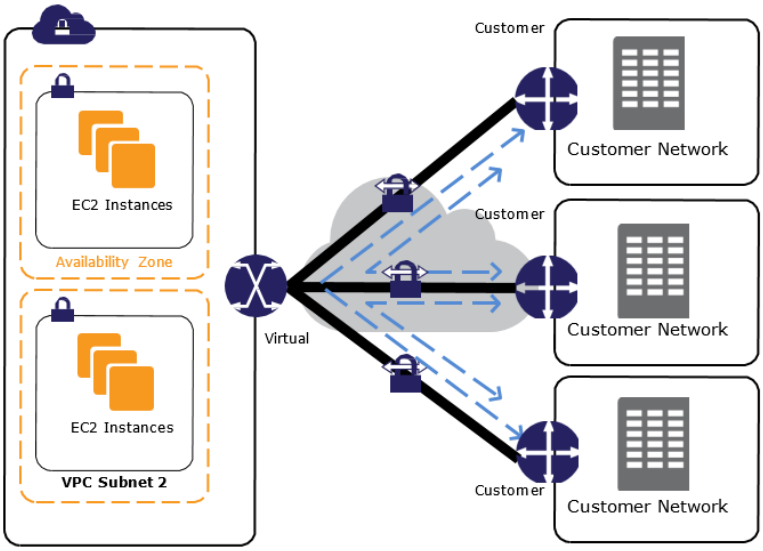
c. On-prem network to multiple VPC in same or different regions
   i. Use **Direct Connect Gateway** by placing it in any public region
   ii. Each Direct Connect connection can be configured with one or more **Virtual Interfaces** (VIFs).
   iii. Direct Connect Gateway provides a grouping of Virtual Private Gateways (on VPC) and Private Virtual Interfaces (VIFs) (Direct Connect) that belong to the same AWS account and enables you to interface with VPCs in any AWS Region.



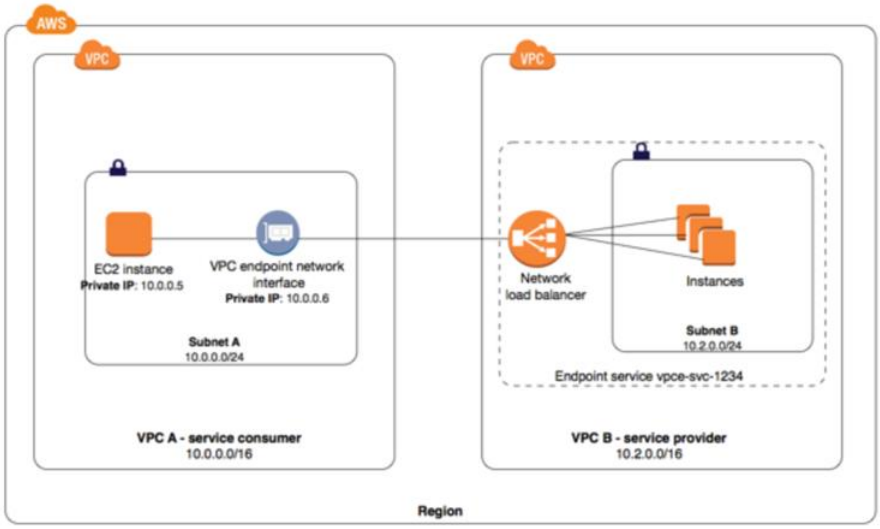
d. To connect two VPCs together
    i. Divide a Direct Connect connection into multiple logical connections, one for each VPC
    ii. Use these logical connections for routing traffic between VPCs



3. **VPN CloudHub (VPC to multiple on-prem networks)**
    a. Operates on a simple hub-and-spoke model that you can use with or without a VPC.
    b. For when you have multiple branch offices and existing internet connections



4. **PrivateLink (Private link VPC to VPC)**
    a. To use services offered by another VPC securely over private connection.
    b. You can create an interface endpoint to keep all traffic within AWS network.
    c. You have a service provider and service consumer
    d. VPC Security Groups used to manage access to the endpoints.



A Virtual Private Gateway is used to setup a VPN. You can use this in combination with Direct Connect to encrypt all data that traverses the Direct Connect link. This combination provides an IPsec-encrypted private connection that also reduces network costs, increases bandwidth throughput, and provides a more consistent network experience than internet-based VPN connections.

Situation: The company has a large presence in AWS in multiple regions. You have established a new office and need to implement a high-bandwidth, low-latency connection to multiple VPCs in multiple regions within the same account. The VPCs each have unique CIDR ranges.

- *AWS Direct Connect connection to the closest region.*
- *Use Direct Connect Gateway to VIFs to each AWS region. It provides a grouping of Virtual Private Gateways (on VPC) (VGWs) and Private Virtual Interfaces (VIFs) (Direct Connect) that belong to the same AWS account and enables you to interface with VPCs in any AWS Region.*

Situation: The company you work for is currently transitioning their infrastructure and applications into the AWS cloud. You are planning to deploy an Elastic Load Balancer (ELB) that distributes traffic for a web application running on EC2 instances. You still have some application servers running on-premise and you would like to distribute application traffic across both your AWS and on-premises resources.

- *Provision a Direct Connect connection between your on-premises location and AWS and create a target group on an ALB to use IP based targets for both your EC2 instances and on-premises servers*
- *You must have a VPN or Direct Connect connection to enable this configuration to work*

d. VPC Security

i. **Security Group**

1. Acts like a firewall
2. It is assigned to an instance (not subnet) in a VPC
3. One instance only a maximum of 5 security groups
4. 1 security group can be associated to many instances
5. Only permit rules allowed, cannot assign deny rules
6. Security group membership can be changed whilst instances are running
7. Any changes to security groups will take effect immediately
8. Stateful processing: Outgoing requests can be answered by incoming requests
9. Types
    a. **Default Security Groups** default permissions
        i. Inbound: Allow traffic from within the group
        ii. Outbound: Allow all traffic.
    b. **Custom Security Groups** default permissions
        i. Inbound: All denied
        ii. Outbound: All allowed

ii. **Network Access Control List (NACL)**

1. Acts like a network level firewall
2. It is assigned on a subnet in VPC
3. A VPC comes with a default NACL
4. Has both allow and deny rules
5. Lowest numbered rules first, first match overrules others
6. Stateless processing
7. Types:
    a. **Default NACL** default permissions
        i. Inbound: All traffic allowed
        ii. Outbound: All traffic allowed
    b. **Custom NACL** default permission
        i. Inbound: All denied
        ii. Outbound: All denied

Situation: To enable your Lambda function to access resources inside your private VPC, you must provide additional VPC-specific configuration information.

- *VPC subnet IDs and security group IDs.*
- *AWS Lambda uses this information to set up elastic network interfaces (ENIs) that enable your function to access resources inside your private VPC*

Elastic Load Balancer (ELB)

1. Concept
    a. Role:
        i. It is a process that checks for connection requests, using the protocol and port that you configure.
        ii. The rules that you define for a load balancer listener determine how the load balancer routes requests to its registered targets.
    b. Categories:
        i. Receiver Initiated:
            1. Receiver selects the targets
            2. For common use-cases
        ii. Sender Initiated:
            1. Sender locates the best target
            2. Example: Domain Name System (DNS) load balancing
    c. Load balancing algorithms:
        i. Round Robin:
            1. Node 1 handles 1st, 4th, and 7th request
            2. Node 2 handles 2nd, 5th, and 8th request
            3. Node 3 handles 3rd, 6th, and 9th
        ii. Randomized
        iii. Centrally managed: Intelligent load balancer that works on complex conditions
        iv. Threshold based: If a request threshold is crossed on one node, all traffic is routed to the second one
    d. Types:
        i. **Application Load Balancer (ALB)**
            1. Operates on level 7
            2. Only supports health checks on HTTP and HTTPS traffic
            3. Commonly used for web apps, web load balancing, microservices and containers
            4. ASG would automatically register new instances with the ALB.
            5. **Cross-zone load balancing** ensures traffic is distributed evenly between available instances in all AZs. It is enabled on the ALB by default.
        ii. **Network Load Balancer (NLB)**
            1. Operates on level 4
            2. Only supports TCP traffic.
            3. Commonly used for high performance, low latency, sudden/volatile traffic.
        iii. **Classic Load Balancer (CLB)**
            1. Oldest and not recommended by AWS
            2. Operates on both level 4 and 7
            3. Supports health checks on HTTP, TCP, HTTPS and SSL

e. ELB features:
    i. You can select only one subnet per availability zone for each ELB
    ii. Does not work between multiple subnets.
    iii. **Internet facing ELBs**:
        1. Have public IPs and route traffic to the private IP addresses of the EC2 instances.
        2. Placed in a public subnet.
    iv. Restricting access on IP addresses using ELB:
        1. **ELB Security Group** rules
        2. Configure the ELB to send the X-Forwarded For headers to the web servers. **X-forwarded-for for HTTP/HTTPS** carries the source IP/port information. Web servers can then filter traffic using a local firewall such as iptables. X-forwarded-for only applies to L7.
    v. ELB Health Check:
        1. ELB will stop sending traffic to the instance that failed that check
        2. It does not instruct Auto Scaling to terminate the instance. Auto Scaling has its own health checks.
    vi. **Perfect Forward Secrecy (PFS)** provides additional safeguards against the eavesdropping of encrypted data, through the use of a unique random session key.
    vii. With ALB and NLB IP addresses can be used to register:
        1. Instances in a peered VPC
        2. AWS resources that are addressable by IP address and port
        3. On-premises resources linked to AWS through Direct Connect or a VPN connection

Situation: Running an ASG with ELB. EC2 status health checks are configured on ASG. ELB has determined that an EC2 instance is unhealthy but the ASG has not terminated it.

- *ELB health check type is not selected for the ASG so it does not know. Therefore, always select and use ELB health check with ASG.*

Situation: In a public subnet, EC2 based reverse proxy behind an ELB Classic Load Balancer and performs content-based routing to 2 back end instances in private subnet. We are facing increased load. What to do?

- *Have ALB on front end and reverse proxy layer, and add auto scaling to back end EC2 fleet*

Situation: Expecting large bursts of traffic during a planned event?

- *Pre warm your Application Load balancer (ALB) by contacting AWS prior to the event.*

Situation: Using Auto Scaling for EC2 instances behind an ELB across two subnets. Only one subnet is running EC2.

- *You need to configure multiple subnets in the ASG.*

Situation: A company is deploying a new two-tier web application that uses EC2 web servers and a DynamoDB database backend. An Internet facing ELB distributes connections between the web servers. The Solutions Architect has created a security group for the web servers and needs to create a security group for the ELB. What rules should be added?

- *On ELB Security Group: Add an Inbound rule that allows HTTP/HTTPS, and specify the source as 0.0.0.0/0*
- *On web server security group: Add an Inbound rule allowing HTTP/HTTPS from the ELB security group*
- *On ELB Security Group: Add an Outbound rule that allows HTTP/HTTPS, and specify the destination as the web server security group*

Virtual Network Services

1. **Domain Name System (DNS)**
    a. Domain: Specified boundaries of activity
    b. Domain Name
        i. Human friendly name to access the actual IP address i.e Name to IP address Mapping
        ii. Fully Qualified Domain Name (FQDN) example services.aws.amazon.com
            1. aws – Host name
            2. amazon – Management/Organization Domain
            3. .com – Top level Domain
            4. services – Sub Domain
        iii. It can identify more than one IP address
        iv. More than one domain names can point to a single IP address
    c. Resolving process:
        i. Requestor (configures with a DNS server address) requests to connect to the Root DNS server (.com)
        ii. The request is sent down to the organization DNS (amazon.com)
        iii. The request is directed to the host (aws.amazon.com)
        iv. The IP address of the host it resolved
        v. To speed up this process the DNS hosting service can cache recurring requests in-memory
    d. Alias: Different names used for target destination
    e. DNS Records (record sets): Different entries in your DNS zone for different hosts or aliases that are used to resolve IP to host name or vice versa
    f. DNS zone transfer is used to offload name resolution processing
    g. Lookups:
        i. Forward: Request IP address of a host name
        ii. Reverse: Request host name of an IP address
    h. Server Name Indication (SNI):
        i. A user indicates the hostname to connect to
        ii. SNI supports multiple secure websites on a single secure listener

Situation: A solution for distributing load across a number of EC2 instances across multiple AZs within a region. Customers will connect to several different applications running on the client's servers through their browser using multiple domain names and SSL certificates. The certificates are stored in AWS Certificate Manager (ACM). What is the optimal architecture to ensure high availability, cost effectiveness, and performance?

- *A single ALB and bind multiple SSL certificates to the same listener. Clients use the Server Name Identification (SNI).*

Situation: How are DNS hostnames defined by default when launching an EC2 instance in VPC?

- *Default VPC: Both public and private DNS Domain name provided*
- *Non-default VPC: Private DNS Domain name provided but public DNS domain name is not provided, and you have to create one.*

2. **Route 53**
    a. A highly available and scalable cloud Domain Name System (DNS) web service.
    b. Extremely reliable and cost-effective way to route end users to Internet applications by translating names like www.example.com into the numeric IP addresses like 192.0.
    c. Services:
        i. Domain registration,
        ii. DNS service
        iii. Health checking.
        iv. Traffic management
    d. Route 53 **alias records** provide a Route 53–specific extension to DNS functionality. Alias records let you route traffic to selected AWS resources, such as:
        i. CloudFront distributions
        ii. Elastic BeanStalk Environment
        iii. ELB Load balancer
        iv. VPC interface endpoint
        v. API Gateway
        vi. Amazon S3 buckets.
        vii. Global Accelerator
        viii. Another Route 53 record in the same hosted zone
    e. Routing Policies:
        i. **Simple routing policy**
        ii. **Weighted routing policy**
        iii. **Latency routing policy:** Sends requests to lowest latency servers for that user.
        iv. **Failover routing policy** is used for active/passive configurations.
        v. **Geolocation routing policy** that directs users based on their geographical location. Different local content delivery
        vi. Health checking: Use **Route 53 Multivalue answers** to return up to 8 values (such as IP addresses) with each DNS query.
    f. To use Route 53 for an existing domain the architect needs to change the NS records (server that contains the actual DNS records) to point to the Amazon Route 53 name servers. This will direct name resolution to Route 53 for the domain name.

Situation: The client would like 80% of the traffic to hit the AWS-based web servers and 20% to be directed to the on-premises web servers.

- *Use **Route 53 with a weighted routing pol**icy and configure the respective weights. For active-active configurations.*
- *Application Load Balancer can distribute traffic to AWS and on-premise resources using IP addresses but cannot be used to distribute traffic in a weighted manner.*

3. **Flow Logs with S3 buckets**
    a. Capture information about traffic that is moving in your environment
    b. Can be created on Network interfaces, VPCs and Subnets

4. **AWS Global Accelerator:**
    a. Uses the vast, congestion-free AWS global network to route TCP and UDP traffic to a healthy application endpoint in the closest AWS Region to the user.
    b. Intelligently route traffic to the closest point of presence (reducing latency).
    c. Seamless failover is ensured as AWS Global Accelerator uses anycast IP address which means the IP does not change when failing over between regions so there are no issues with client caches having incorrect entries that need to expire.
    d. This is the only solution that provides deterministic failover.

5. **API Gateway**
    a. Decouples the client application from the back-end application-layer services by providing a single endpoint for API requests.
    b. Features:
        i. **Throttling** enables you to throttle the number of requests to your API which in turn means less traffic will be forwarded to your application server.
        ii. **Result caching** is possible to cache frequent queries
        iii. Caching features include customizable keys and **time-to-live (TTL) in seconds** for your API data which enhances response times and reduces load on back-end services.
        iv. If **Cross-origin Resource Sharing (CORS)** is not enabled and an API resource receives requests from another domain, the request will be blocked.
    c. Scaling:
        i. API Gateway scales up to the default throttling limit of 10,000 requests per second and can burst past that up to 5,000 RPS.
        ii. When request submissions exceed the steady-state request rate and burst limits, API Gateway fails the limit-exceeding requests and returns 429 Too Many Requests error
    d. Access:
        i. Use resource policies, standard IAM roles and policies, Lambda authorizers, and Amazon Cognito user pools.
        ii. IAM roles and policies can be used for controlling who can create and manage your APIs as well as who can invoke them.

6. **CloudFront**
    a. CloudFront is used as the public endpoint for API Gateway and provides reduced latency and distributed denial of service protection
    b. Query string parameters cause CloudFront to forward query strings to the origin and to cache based on the language parameter
    c. Origins can be either an S3 bucket, an EC2 instance, and Elastic Load Balancer, or Route53 – can also be external (non-AWS)
    d. You can use an OAI to restrict access to content in Amazon S3 but not on EC2 or ELB.
    e. **Regional Edge Caches** have larger cache-width than any individual edge location, so your objects remain in cache longer at these locations. Enabled by default for CloudFront Distributions.
    f. **Perfect Forward Secrecy (PFS)** provides additional safeguards against the eavesdropping of encrypted data, using a unique random session key (also works on ELB).

Situation: Your company shares some HR videos stored in an Amazon S3 bucket via CloudFront. You need to restrict access to the private content so users coming from specific IP addresses can access the videos and ensure direct access via the Amazon S3 bucket is not possible.

- *Create an **origin access identity (OAI)**, which is a special CloudFront user, and associate the OAI with your distribution.*
- *You can then change the permissions either on your Amazon S3 bucket or on the files in your bucket so that only the origin access identity has read permission*
- *You can also specify IP addresses of the users who can access your content.*
- ***Field-level encryption** in CloudFront adds an additional layer on top of HTTPS that lets you protect specific data so that it is only visible to specific applications.*

Situation: On-demand and live streaming video to your customers. The plan is to provide the users with both the media player and the media files from the AWS cloud. One of the features you need is for the content of the media files to begin playing while the file is still being downloaded.

- *Store media files in S3 bucket*
- *Use CloudFront with a Web (media player) and RTMP Distribution (for streaming media files)*
- *Allows an end user to begin playing a media file before the file has finished downloading from a CloudFront edge location*
- *For RTMP, files must be stored in S3 buckets.*

7. **WAF:**
    a. Helps detect and block malicious web requests targeted at your web applications.
    b. Allows you to create rules that can help protect against common web exploits like SQL injection and cross-site scripting.
    c. You first identify the resource (either an Amazon CloudFront distribution or an ALB) that you need to protect. You then deploy the rules and filters that will best protect your applications.

Identity Access and Management (IAM)

1. Concepts:
    a. Resources: Things on which actions can be taken
    b. Principals:
        i. Things that can take actions on resources.
        ii. They are
            1. Users:
                a. Individual people/entities in AWS
                b. Could be a person or service with permissions
                c. Can be created through Management Console or AWS CLI
                d. Needs credentials:
                    i. Name & Password
                    ii. Can be given a maximum of 2 access keys (access key ID + secret access key) that are used for (API/CLI) programmatic access to resources
                e. Can be members of IAM Groups
            2. Groups
                a. Collection of IAM users
                b. Permissions managed at the group level
                c. Users can be added or removed
                d. Groups are not used to login
                e. You cannot nest groups
            3. Roles
                a. Identities that are granted permissions
                b. Roles are not permanently assigned to an entity, therefore no permanent credentials
                c. They can be assumed by any identity
                d. They use keys to access AWS resources
                e. Roles are compatible with Federated Users (3rd party identification system) and Single Sign-On (SSO) that allows users to login to the AWS console without assigning IAM credentials
    c. Policies:
        i. Rights, permissions, authorizations usually in the form of a JSON file
        ii. Types:
            1. Identity based policies for principals
            2. Resource based policies for cross-account access
        iii. Policy process:
            1. By default all requests are denied
            2. Explicit allow overrides the default
            3. **Permission boundaries** can override explicit allow
    d. Actions: A request is first authenticated; it is then authorized and only then actions are allowed on resources

2. **Security Token Service**
    a. Used when you are using a custom identity broker
    b. Enables you to request temporary, limited-privilege credentials for IAM users or for users that you authenticate (federated users).
    c. Federations can come through 3 sources:
        i. Federation (typically Active Directory)
        ii. Federation with Mobile Apps (e.g. Facebook, Amazon, Google or other Open ID providers)
        iii. Cross account access (another AWS account)

3. **Multi-factor Authentication (MFA)**
    a. Activation:
        i. Management Console: user prompted for username, password and authentication code
        ii. AWS API: restrictions are added to IAM policies and developers can request temporary security credentials and pass MFA parameters in their **AWS STS API requests**
        iii. AWS CLI by obtaining temporary security credentials from STS (aws sts get-session-token)
    b. If your AWS root account MFA device is lost, you can sign in using alternative methods of authentication (email and phone that are registered with your account)

4. **AWS Directory**
    a. **Amazon Simple AD**
        i. Active Directory-compatible service with common directory features.
        ii. It is a standalone, fully managed, directory on the AWS cloud and is generally the least expensive option.
        iii. It is the best choice for less than 5000 users and when you don't need advanced AD features.
    b. **AD Connector**
        i. Directory gateway with which you can redirect directory requests to your on-premises Microsoft Active Directory without caching any information in the cloud.
        ii. Requires a VPN or Direct Connect connection
        iii. Only supports up to 5000 users
    c. **Active Directory Service for Microsoft AD**
        i. Best choice if you have more than 5000 users and/or need a trust relationship set up.
        ii. It provides advanced AD features that you do not get with SimpleAD.
        iii. Requires a VPN or Direct Connect connection
        iv. Does not support replication mode where you replicate your AD between on-premise and AWS

5. **Cognito**
    a. Can be used to define roles and maps users to those roles without IAM
    b. Authentication service for web and mobile apps through an external identity provider (FB, Google etc.) using a **user pool** (user directory in Cognito)
    c. Can scale to millions of users
    d. With an **identity pool,** users can obtain temporary AWS credentials to access AWS services, such as Amazon S3 and DynamoDB
    e. You can use MFA with a Cognito user pools

6. **AWS Organizations**
    a. Collection of AWS accounts
    b. Centralized management interface for billing and account management
    c. Free to use
    d. Organizational Units:
        i. Hierarchical account management
        ii. Nest OUs up to 5 levels deep
    e. To apply the restrictions across multiple member accounts you must use a **Service Control Policy (SCP)** in the AWS Organization
    f. Explicit deny
    g. **AWS Resource Access Manager:**
        i. Enables you to share AWS resources easily and securely with any AWS account or within your AWS Organization.
        ii. It is not used for restricting access or permissions.
    h. To move an account to AWS organization, use AWS Organizations Management Console
    i. Use Organizations API or CLI to migrate many accounts to an organization
    j. The source datastore can be Server Message Block (SMB) file servers.

7. **AWS Single Sign-On:** Cloud SSO service that makes it easy to centrally manage SSO access to multiple AWS accounts and business applications

8. **IAM Query API**
    a. To make direct calls to the IAM web service.
    b. An access key ID and secret access key must be used for authentication when using the Query API

9. IAM Best Practices
    a. Root user:
        i. It is the first account that is created using email and password
        ii. Has unlimited capabilities
        iii. Should be only used to:
            1. Modify support plan
            2. Close the account
            3. Create a CloudFront key pair
            4. Enable Multi-Factor Authentication on S3 bucket
            5. Restore permission to IAM users
        iv. Not recommended for everyday use
        v. Create an IAM admin user instead and rarely use the user account
        vi. Set up permission boundaries
    b. Password Policies
        i. Set up password policies to make it mandatory for users to create strong passwords
        ii. Prompt users to change passwords periodically by expiration, preventing reuse or doing an admin reset
    c. Principle of Lease Privilege: Grant users only the access that is needed to them
    d. Never give an IAM role admin privileges

10. **AWS CloudHSM**
    a. Cloud-based hardware security module (HSM) that allows you to easily add secure key storage and high-performance crypto operations to your AWS applications.
    b. Has no upfront costs
    c. Provides the ability to start and stop HSMs on-demand, allowing you to provision capacity when and where it is needed quickly and cost-effectively.
    d. A managed service that automates time-consuming administrative tasks, such as hardware provisioning, software patching, high availability, and backups.
    e. CloudHSM is a part of a Public Key Infrastructure (PKI) but a PKI is a broader term that does not specifically describe its function.
    f. There is no way to recover your keys if you lose your credential

Management and Governance

1. **CloudWatch**
    a. There is so standard metric for memory usage on EC2 Instances
    b. **CloudWatch Logs**
        i. Monitor applications and systems using log data and send you a notification whenever the rate of errors exceeds a threshold you specify.
        ii. Options for storing logs:
            1. CloudWatch Logs
            2. Centralized Logging System (Splunk)
            3. Custom Script and store on S3
    c. **CloudWatch Events** delivers a near real-time stream of system events that describe changes in Amazon Web Services (AWS) resources.
    d. **Alarms:**
        i. **System status checks** detect (StatusCheckFailed_System) problems with your instance that require AWS involvement to repair
        ii. **Instance status checks** (StatusCheckFailed_Instance) detect problems that require your involvement to repair

Situation: You have associated a new launch configuration to your Auto Scaling Group (ASG) which runs a fleet of EC2 instances. The new launch configuration changes monitoring from detailed to basic. There are a couple of CloudWatch alarms configured on the ASG which monitor every 60 seconds. There is a mismatch in frequency of metric reporting between these configuration settings, what will be the result?

- *If you do not update your alarms to match the five-minute period, they continue to check for statistics every minute and might find no data available for as many as four out of every five periods*

2. **CloudTrail**
    a. It is a logging service used more auditing and governance
    b. Process:
        i. Activity occurs in an account
        ii. AWS captures it and records it as an event
        iii. You can view/download the activity in the event history
        iv. You set up the Cloud Trail and define S3 bucket to store the event logs
        v. If you do not create a CloudTrail the event history only lasts for 90 days
    c. CloudTrail is used for recording API calls (auditing) whereas CloudWatch is used for recording metrics (performance monitoring).
    d. A single trail can be applied to all regions with a single KMS key used to encrypt log files for trails applied to all regions.
    e. CloudTrail log files are encrypted using S3 Server-Side Encryption (SSE) and you can also enable encryption SSE KMS for additional security.
    f. Trails can be configured to log Data events and management events:
        i. Data events (data plane operations):
            1. Resource operations performed on or within a resource.
            2. Disabled by default
        ii. Management events (control plane operations):
            1. Management operations performed on resources.
            2. Can also include non-API events that occur in your account.
            3. Last 90 days free of charge
3. **Systems Manager**
    a. Gives you visibility and control of your infrastructure on AWS.
    b. Allows you to automate operational tasks across your AWS resources.

4. **AWS Trusted Advisor**
    a. Help you reduce cost, increase performance, and improve security by optimizing your AWS environment.
    b. Provides real time guidance to help you provision your resources following AWS best practices.
    c. **Service Limits check** (in the Performance category) that displays your usage and limits for some aspects of some services

5. **Amazon Inspector**
    a. Automated security assessment service that helps improve the security and compliance of applications *deployed* on AWS.
    b. It is not used to secure the actual deployment of resources, only to assess the deployed state of the resources

6. **AWS X-Ray**
    a. It lets you analyze and debug serverless applications by providing distributed tracing and service maps to easily identify performance bottlenecks by visualizing a request end-to-end

Application Integration

1. **CloudFormation**
    a. **Logical IDs** are used to reference resources within the template
    b. **Physical IDs** identify resources outside of AWS CloudFormation templates, but only after the resources have been created
    c. **TemplateURL parameter**, lets you specify where the CloudFormation template for a stack action resides and enforce that it be used
    d. Two methods for updating stacks:
        i. **Direct update**: Directly update a stack, you submit changes and CloudFormation immediately deploys them. Used when you want to quickly deploy your updates.
        ii. **Change sets**: you can preview the changes CloudFormation will make to your stack, and then decide whether to apply those changes
    e. **StackSets:**
        i. Extends functionality of stacks by enabling you to create, update, delete multiple stacks across multiple target accounts and regions with a single operation
        ii. An administrator account is the AWS account in which you create stack sets
        iii. Before you can use a stack set to create stacks in a target account, you must set up a trust relationship between the administrator and target accounts
    f. **AWS SAM (Serverless Application Model)** extension AWS CloudFormation that is used to package, test, and deploy serverless applications
    g. Use a version control system with your templates so that you know exactly what changes were made, who made them, and when.

2. **Simple Queue Service (SQS)**
    a. Gives you access to message queues that store messages waiting to be processed.
    b. Offers a reliable, highly scalable, hosted queue for storing messages in transit between computers and is used for distributed/decoupled applications.
    c. Can be used with RedShift, DynamoDB, EC2, ECS, RDS, S3 and Lambda
    d. Messages can be stored in a queue for a maximum amount of time

The application will use a number of front-end EC2 instances that pick-up orders and place them in a queue for processing by another set of back-end EC2 instances. The client will have multiple options for customers to choose the level of service they want to pay for. The client has asked how he can design the application to process the orders in a prioritized way based on the level of service the customer has chosen.

- *Create multiple queues and configure the application to place orders onto a specific queue based on the level of service. You then configure the back-end instances to poll these queues in order or priority so they pick up the higher priority jobs first.*
- *Using a single queue won't work as there is no guarantee that the messages would be picked up in the correct order.*

3. **Amazon Simple Workflow Service (SWF)**
    a. Makes it easy to coordinate work across distributed application components.
    b. SWF enables applications for a range of use cases, including media processing, web application back-ends, business process workflows, and analytics pipelines, to be designed as a coordination of tasks.
4. **Amazon MQ**
    a. Supports industry-standard APIs and protocols so you can migrate from your existing message broker without rewriting application code
5. **Amazon Step Functions**
    a. Used for coordinating multiple AWS services into serverless workflows, it is not a message broker

ANALYTICS

1. **Kinesis Data Streams**
    a. Build custom applications that process or analyze streaming data for specialized needs
    b. Read throttling is enabled by default
    c. If total reads exceeding the per-shard limits, you need to increase the number of shards in the Kinesis data stream.
    d. When you have enabled **extended data retention** you can store data up to 7 days in Amazon Kinesis Data Streams – you cannot store it for 1 year
    e. **Kinesis Data Streams producer** for ingestion of the real-time data and then configuring a **Kinesis Data Streams consumer** to write the data to DynamoDB
    f. Scaling by shards
2. **Kinesis Data Firehose**
    a. Does not provide storage
    b. Can use Kinesis Data Streams as a source and can capture, transform, and load streaming data into a RedShift cluster.
    c. Can invoke a Lambda function to transform data before delivering it to destinations
    d. You cannot configure DynamoDB as a destination in Amazon Kinesis Firehose. The options are S3, RedShift, Elasticsearch and Splunk
    e. RDS is a transactional database and is not a supported Kinesis Firehose destination
    f. Auto scaling
3. **Amazon Kinesis Data Analytics**
    a. The easiest way to process and analyze real-time, streaming data.
    b. Can use standard SQL queries to process Kinesis data streams and can ingest data from Kinesis Streams and **Kinesis Firehose** but Firehose cannot be used for running SQL queries.
4. **Athena:**
    a. Interactive query service that makes it easy to analyze data in S3 using standard SQL.
    b. Athena is serverless, so there is no infrastructure to manage, and you pay only for the queries that you run.

5. **AWS Glue** is a fully managed extract, transform, and load (ETL) service that makes it easy for customers to prepare and load their data for analytics from S3.

6. **Redshift**
    a. RedShift is a SQL based data warehouse used for analytics applications
    b. Cannot analyze data in S3.
    c. **Amazon RedShift Spectrum** is a feature of Amazon Redshift that enables you to run queries against exabytes of unstructured data in Amazon S3, with no loading or ETL required.
    d. RedShift nodes run on EC2 instances, so for infrequent queries this will not minimize infrastructure cost.

7. **Amazon EMR**
    a. Provides a managed Hadoop framework that makes it easy, fast, and cost-effective to process vast amounts of data across dynamically scalable Amazon EC2 instance and S3
    b. EMR launches all nodes for a given cluster in the same Amazon EC2 Availability Zone.
8. **Amazon S3 Select**
    a. Amazon S3 Select is designed to help analyze and process data within an object in Amazon S3 buckets, faster and cheaper.
    b. It works by providing the ability to retrieve a subset of data from an object in Amazon S3 using simple SQL expressions

DEVELOPER TOOLS

1. The following AWS services can be used to fully automate the deployment process:
    a. **CodePipeline**: model, visualize, and automate the steps required to release your serverless application
    b. **CodeDeploy** to gradually deploy updates to your serverless applications
    c. **CodeBuild** to build, locally test, and package your serverless application
    d. **AWS CloudFormation** to deploy your application
2. **Code Commit**: Software version control
3. **CodeStar**: Quickly develop, build and deploy apps
4. **Step Functions**: Coordinate multiple AWS services into serverless workflows

MEDIA SERVICES

1. **Amazon Elastic Transcoder**:
    a. A highly scalable, easy to use and cost-effective way for developers and businesses to convert (or “transcode”) video and audio files from their source format into versions that will playback on devices like smartphones, tablets and PCs
2. **Amazon Personalize** machine learning service that makes it easy for developers to create individualized recommendations for customers using their applications
3. **Data Pipeline** helps you move, integrate, and process data across AWS compute and storage resources, as well as on your on-premises resources
4. **Rekognition** is a deep learning-based visual analysis service