Aws Platform.

→ Messaging
→ Security & Identity
→ Management Tools
→ Storage
→ Databases
→ N/w & Content Delivery
→ Compute
→ Aws Global Infrastructure.

Aws Global Infrastructure (Exam)

Region - A Geographical area. Each Region consists of 2 or more Availability Zones.
* An Availability Zone (AZ) is simply a Data Center.

Edge Location - CDN End Points for Cloud Front.

* Vpc - Virtual Private Cloud.
This is a virtual Data Center Just like a Availability Zone.

* Route 53 - Amazon's DNS lookup Service. You can register domain names using Route 53
(53 is the DNS Port. Hence they named Route 53)

* Cloud Front - Part of CDN. Consists of Edge Location

* Direct Connect : Use of dedicated line into AWS infrastructure. for reliability.

* EC2 – Elastic Compute Cloud. (Virtualization Technology)
This is a virtual machine on the cloud. like VMware.
ECCS → Highly Scalable, High Performance Container Management Service. Cluster management Infrastructure.

* Elastic BeanStalk – Infrastructure Provisioning.

* Lambda – This is Serverless unlike EC2 which has an Operating System. Just Upload code and code will respond to events

Storage (Exam)

(Simple Storage Service)

1) S3 – Virtual Disk on the Cloud to Store any objects.
Not used for installations.
Dropbox uses S3 for Object Storage.

2) Glacier – Archival block. Data Archival @ low cost.
[Can't access immediately, Typically 3 to 4 hours)

3) EFS – Elastic File Service.
DB installation, Application installation.

4) Storage Gateway – Virtual machine to communicate with S3.

Databases. (Exam)

① RDS – Relational Database Service
MySQL, Postgress, SQL Server, Oracle and Aurora DB's.

② Dynamo DB – NoSQL DB. Highly Scalable.

③ RedShift – Amazon Data Warehouse Solution. Used to Run Reports and things.

④ ElastiCache – Caching Data in the Cloud. To take load of the DB.
Ex: Top 10 Selling items on a website.

Migration:

① Snowball – Enterprise level Data Migration. Allows to transfer Storage to S3.

② Dms – Database Migration Services. On Premise DB to Cloud.
Oracle on Premise to Aurora Cloud DB.
[Dms will take care of everything.]

③ Sms (Server Migration Services)
Virtual machine Migration. Ex: VMware to AWS Cloud.

Analytics.

① Athena – Allows to run Query on S3 buckets.
i.e Flat Files into DB Query.

② EMR – Elastic Map Reduce. Big Data Related.

③ Cloud Search & Elastic Search. :- Search capabilities in websites.
④ Kinesis - Streaming and real time Analysis.
Ex: Financial Market Analysis.

Security & Identity. (Exam)

① IAM - Sign Authentication, User management.
Identity & Access management
② Inspector - VM inspector.
③ Certificate Manager - SSL Certificates for domain names.
④ Directory Service - Active Directory Connection to Aws.
⑤ WAF - Web Application Firewall.
To stop SQL injection / Cross Site Scripting.

Management Tools. (Exam)

① Cloud watch - Monitor Performance of EC2 e.t.c.
Ex: Disk Utilization.
② Cloud Formation - Turn infrastructure into code.
kind of template to Provision Aws's.
③ Cloud Trail - Auditing Aws Resources.
i.e Changes to Aws environment.
④ OpsWorks - Automating Deployments.

Application Services.

① Swf - Simple workflow Service.
⇒ Used in Amazon fulfilment centers for manual & Automated tasks workflow.

(2) API Gateway – API for backend Code Access.
Door way to access backend Services.

Messaging (Exam)

(1) SNS – Simple Notification Service. Notify via email, Text message or http / https endPoints.
(2) SQS – Decoupling applications. Post Jobs to Queue.
(3) SES – Simple Email Service.

IAM – Identity Access Management.

* Allows you to manage Users and their level of Access to AWS Console
* Centralised control to AWS account
* Shared access to AWS account
* Granular Permissions.
* Web Identity Federation (Active Directory, Facebook, Linkedin)
* Multifactor Authentication.
* Temporary Access to Users/Services where necessary.
* Allows Setting up Password rotation Policy.

Policies → A document that defines one or more Permissions
Users → End user i.e People.
Roles → Create role & then assign them to AWS resources, like For Ex: S3AdminAccess to EC2.

* Access key ID & Secret access key } Programatic Only.

* SAML – Security Assertion Markup Language.

Active Directory Federation.

(1) User types a URL in the browser inside his domain
(2) Sign in into Active Directory environment.
(3) Browser receives a SAML assertion in the form of an authentication from ADFS.
(4) Browser Posts the SAML assertion to the AWS Sigin endPoint
* (Sign in uses AssumeRole with SAML API to request temporary Security Credentials)
(5) Browser receives the Signin URL and is redirected to the Console.

Web Identity Federation.

* From the Console – Web Identity Federation PlayGround.
* Allows to login using FaceBook, linkedin etc.

Steps:

(1) Authenticate with Identity Provider. Ex: FaceBook
(2) Obtain temporary Security Credentials
   * By AssumeRole with Web Identity request
(3) Access AWS Resource.

* Role Arn – Amazon resource name.

** Cannot change IAM role on a running EC2 Instance.
You can only change the Permission on the IAM role
* This can be done now.

EC2 - Elastic Compute Cloud.

* Resizable compute capacity in the cloud. Allowing you to quickly scale capacity, both up & down, as your requirements change.

EC2 Options.

(1) On Demand - Allow you to pay a fixed rate by the hour with no commitments. (Test & Dev Env) (No up front payments) ★
→ Black Friday Sale Situation.

(2) Reserved - Provide with a reserved capacity with significant discount on the hourly charge of an instance.
1 Year or 3 Year Terms.

(3) Spot - Supply vs demand kind of model usually got by bid. For applications with flexible start and end times.
Example: Spot for large compute requirement. Based on certain time of best price, get the spot server and do your computation. like Hadoop, Big Data etc.
★ Commercially feasible.
★★ Amazon wont charge for partial hour usage if they terminate EC2 instance. If you terminate, you pay for the hour.

DIRT MCG
D - Density
I - IOPS
R - RAM
T - T2 Micro
M - main choice for General Purpose
C - Compute
G - Graphics.

(4) Dedicated Hosts: Used for regulatory requirements that may not support Multi-tenant virtualization
→ Great for licensing.

EBS (Elastic Block Storage)

* Disk in the Cloud attached to EC2 instances and You can install OS / DB's or any applications.
* Multiple EBS volumes for a single EC2 instance.
* Can't share between multiple EC2 instance.
* Placed in Specific AZ & automatically replicated to protect failure i.e avoid Single Point failure.

EBS volume Types.

① General Purpose SSD (GP2) – upto 10,000 IOPS.
- Designed for 99.999% availability

② Provisioned IOPS SSD (IO1)
- For large RDBMS or NOSQL
- More than 10,000 IOPS. Can go till 20,000 IOPS.

③ Magnetic (Standard)
- low Storage cost
- Infrequently accessed Storage. File Storage.

ELB – Elastic load Balancer.

→ Not free, Charged by the hour & on a Per Gb basis of usage
→ free Services – Cloud Formation, Elastic Bean Stalk, OPS Works Auto Scaling.
* But resources created by this is not free.

* Configure one or more listeners for ELB. A listener is a Process that checks for connection requests to your LB.

* It is configured with a Protocol and a Port Number.
  Ex: SSH 22

ELB Protocols.

HTTP
HTTPS
TCP
SSL (Secure TCP)

Ports: EC2-VPC 1-65535.

HTTP Codes.

* 200 - Request Success
* 3xx - Redirection
* 4xx - Client Error i.e @ browser 404 not found
* 5xx - Server Error i.e Server not running.

Available SDKs

- Android, iOS, Java, C++
- Java, .Net
- Node.js, PHP, Python, Ruby, GO, C++.

SDK default Regions - US-East-1
* Some do not (Node.js)

* In order to enable encryption at rest using EC2 and Elastic block Store, You need to → Configure encryption when creating the EBS volume.

* Default Encryption on S3 - AES 256.

11 9's.

[Durability = 99.9.... %.]

S3 - Simple Storage Service. (99.9% availability by Amazon)

* Secure highly-scalable Object Storage.
* Object based Storage - videos, photos, PDF docs etc
  i.e NO installables. ✱✱✱✱
* Files can be from 0 Bytes to 5TB. (Single Put Operation size is 5GB, so use Multi-Part Upload API)
* Files are stored in Buckets.
* S3 names should be unique globally.
* Upload success will give HTTP 200 code.

Data Consistency Model for S3.

⇒ & for all regions.

* Read after write for new objects is immediate
* Eventual consistency: Updates to existing object is going to take some time.
  → Updates are Atomic. i.e either new version or old version, no corrupt Data.
  * S3 = (key, value) - key is stored in Alphabetical Order
  - Version ID (which version of the Object) i.e "lexicographic order"
  - Metadata (Data about Data)
  Sub resources
  - Access Control lists - who can access this object
  - Torrent Support.

Exam Tip: Adding randomness to key names is good so that data is distributed across AZ.

* Tiered Storage Available
* Lifecycle Management
* Versioning
* Encryption
* Bucket Policies

S3 URL - https://s3-region.amazonaws.com/bucketname.

Ex: https://s3-eu-west-1.amazonaws.com/acloudguru

* S3 - Infrequently Accessed - For Data that is Accessed less frequently, but requires rapid access when needed. Lower fee than S3, but charged a retrival fee.
  Ex: Employee payroll data Accessed once a Year.

* Reduced redundancy Storage (RRS) - Same availability but lesser durability. Ex: Use thumbnails in this bucket as its cheaper and easily reproducable.

* Glacier - Archival only. Very cheap. Takes 3-5 hours to restore from Glacier. Cost as low as $0.01 Per GB Per month.

S3 Charges.

Charged for
1. Storage
2. Requests
3. Storage Mgmt Pricing i.e Dev Tag vs QA Tag
4. Data Transfer Pricing
5. Transfer Acceleration

S3 Transfer Acceleration. (Uses AWS CloudFront CDN)

→ Enables fast, easy & secure transfers of files over long distance b/w end users and S3 bucket.
→ Data arrives at edge locations, data routed over a Optimized Path. CloudFront.

S3 - Versioning.

* Stores all versions of an Object (including all writes and even if you delete an Object)
* Great backup tool
* Once enabled, Versioning cannot be disabled, only Suspended. (i.e bucket has to be deleted)
* Integrates with lifecycle rules.
* Multi factor delete capability - Additional layer of security for deletes. like MFA token or security code.

S3 - life cycle Mgmt.

Exam:

* Can be used in Conjunction with Versioning
* Can be applied to Current and Previous versions
* Following actions can be done
  1. Transition to Standard - Infrequent Access storage (30 days)
  2. Archive to Glacier (30 days after IA, if relevant)
  3. Permanently delete (will delete from Glacier) ↓ i.e 60 days ★ Total.

* S3 delete uses a delete marker. Not exactly hard delete.

* Cross region replication - Version should be enabled on Source as well as destination buckets.
  - → Files in an existing bucket are not replicated automatically. All subsequent updated files will be replicated.
  - → Delete markers are replicated.
  - → Deleting individual versions or delete markers will not be replicated.

Cloud Front.

→ CDN is a system of distributed servers that deliver web pages and other web content to a user based on geographical location of the user, the origin of webpage & content delivery server.

→ Edge location - Where the content will be cached.
→ Origin - Can be S3 bucket, EC2 instances, ELB or Route 53
** Can also be external i.e where is your original files.
→ Distribution (Origin) - Collection of Edge Locations.
↳ (1) Web Distribution (2) RTMP.

* Cloud Front uses Edge location to deliver a user's content. It can be a website, streaming and interactive content.

→ Cloud Front can also work with any non-AWS Origin Server.
→ Web distribution - Used for websites.
→ RTMP - Used for media streaming. Adobe Flash.

* Edge locations is not just Read only, You can write to them too.
* Objects are cached for the life of TTL. (Time to live)
* User can clear cached objects at a Edge location, but will be charged.

Types of distributions :-

(1) Web — Web Site related.
(2) RTMP. — Media Files

* Restrict Viewer Access – Use Signed URL's or Signed Cookies.
  ie for Private URL. Ex: A Cloud Guru Paid Videos.

AWS WAF – Web Application Fire Wall. To stop SQL injections.

* Geo-Restrictions – Black list / White list Countries.

* Invalidations – Remove Objects from Cloud Front edge Caches
  But this is Chargeable.

Snow Ball. (Device to move Data to AWS)
Import / Export.

Types.
1. Snow ball
2. Snow ball Edge
3. Snow mobile

Import to S3
Export from S3

Snow ball → • Peta byte Scale data transfer Solution.
• Simple, fast, Secure and One-fifth the cost of high Speed Internet.
• 80 TB Snowball in all regions.

* Basically a hardware appliance where you can Order it from Amazon and Upload all your data and Send device back to Amazon and they Upload data to S3.
– 256-bit encryption.

Snow ball Edge — On board Storage + Compute Capacity.
(100 TB)

* Can run lambda functions on it

Ex: Airlines can use Snow ball edge to collect data on a flight and send it back to AWS data center.

Snow mobile : Peta / Hexa bytes worth of data.

* Semi trailer truck. For moving extremly large amount of data to AWS.

S3 Transfer Acceleration → Using the edge location to upload data to S3 in a optimized way ie Faster.

* Comes with an additional fee.

Cross Origin Resource Sharing (CORS)

* Way of allowing JS in one S3 bucket to reference code in another S3 bucket.

CORS Configuration → Edit Allowed Origin and give the correct origin S3 website link.

Use case : Have image files in one region,
All JS files in another region etc.

** Always use the S3 website URL for CORS. Don't use the S3 URL.

* Cross Region Replication.

⇒ Transfer / Replicate Objects from one bucket in one region to another bucket in a completly different region.

Console

S3 → Bucket → Properties → Cross region replication

* Bucket must have versioning enabled. (Both source & destination)
* Cross region replication only works with new / update Objects. Already existing data won't be replicated.
* Deletes also get Replicated.
* Delete markers or individual delete won't get replicated.
* Regions must be unique. replication can't be done in same region.
* Chaining buckets not Possible. Ex: London bucket → Sydney bucket → California bucket.

S3 - Security and Encryption.

* Default → All new buckets are PRIVATE

Policies.

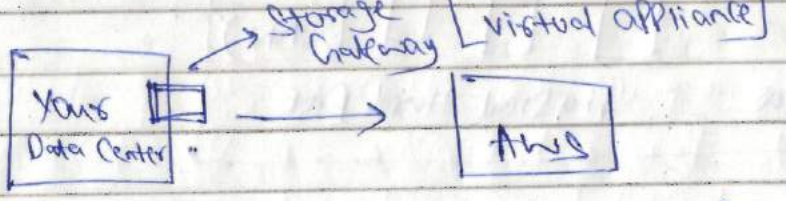
1. Bucket Policies - Bucket level
2. Access Control Lists - Individual S3 Object level.

* S3 buckets can have Access logs. i.e all requests to S3 bucket could be captured.

Encryption.

(1) In-Transit – Pc to bucket. SSL/TLS encryption (i.e HTTPS)
(2) At Rest
  i) Server Side encryption
    (a) S3 Managed keys – SSE S3
    (b) AWS key management Services – SSE KMS → * Gives Audit Trail
    (c) With Customer Provided keys – SSE-C
  ii) Client Side encryption – User encrypts data at Client Side and uploads to S3.

Storage Gateway (Popular Exam Q)



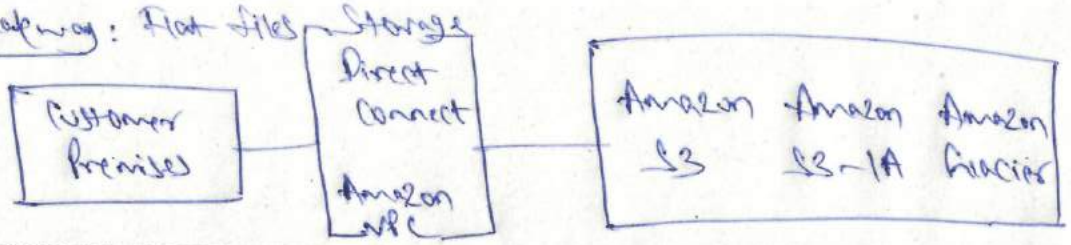
→ Asynchronously replicate data to AWS S3 / Glacier.
→ AWS Storage Gateway S/w is available for download as VM image. i.e VMware ESXi or Microsoft Hyper-V.

Four Types of Storage Gateway.

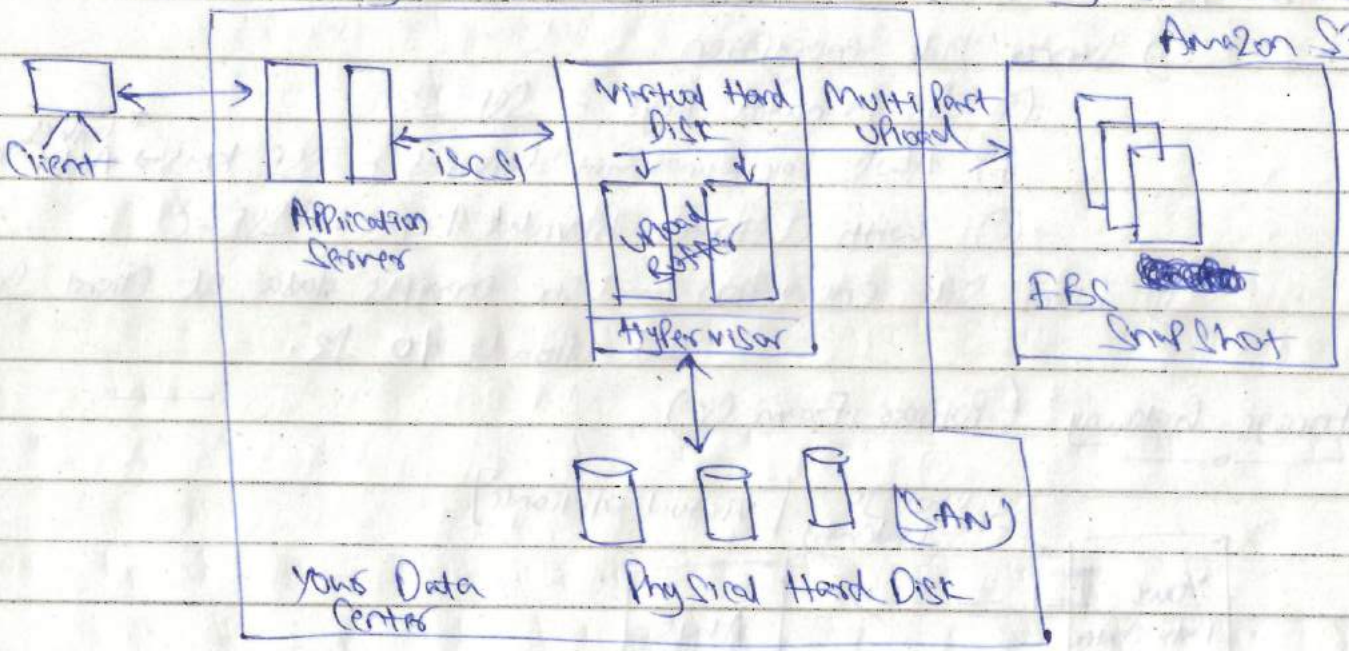
(1) File Gateway (NFS) – Store Flat files in S3. i.e PDF's, word, video etc
(2) Volumes Gateway (iSCSI) – This is block Storage. i.e Virtual Hard disk for installables. [OS, DB image etc]
  (i) Stored volume – Store entire data set on Premise.
  (ii) Cached volume – Recently accessed files. (On Site)
(3) Tape Gateway (VTL) – Back up & Archival for Glacier.
  Ex: Net Backup Application.

* Virtual Tapes → S3 → (to life cycle rules) Glacier for Archival.

File Gateway: Flat files → Storage



* Volume Gateway: Data written to this can be asynchronously backed up as point-in-time Snapshots & stored in cloud as EBS Snapshots.
→ Uses iSCSI block Protocol.
→ Complete copy of Data on Premise incrementally uploaded to S3



* Cached Volumes → Entire Data Set is Stored on S3 and the most frequently accessed data is cached on site.
1GB - 32 TB in size. Because most data in S3.

Exam:
** S3 link formats.

① Bucket Format
https://s3-region.amazonaws.com/bucket-name/cloudyr-website/index.html

② Website Format
http://domain name.s3-website-region.amazonaws.com

Dynamo DB (NO SQL in the cloud) ⟹ * fully managed Service.
i.e Can't SSh to this Service.

→ Fast, flexible NO SQL DB
→ Single-digit millisecond latency at any scale
→ Document and key-value data models.
→ for web, gaming, ad-tech, IoT etc.

* Stored on SSD Storage
* Spread across 3 geographically distinct data centers.
↳ (One main location + 2 replicated ones)
→ Eventual Consistent Reads (Default)
If app can wait { i.e Consistency across all copies of data is usually reached within a second.
→ Strongly Consistent Reads (Data in Sync all the time)
i.e returns a result that reflects all writes that received a successful response prior to the read.

Basics.
* Tables (max 256 Tables per Region) **
* Items (like a row) → max size limit is 10 GB.
* Attributes (like a column) → No limit on max attributes.

```
{
  "UniqueID" : 1978,        ← Attributes.
  "First Name" : "Syril",
  "Last Name" : "Sadasivan",
  "Address" : {
      "Number" : "15"
      "Street" : "River Road" }
}
```

* Nesting upto 35 levels.

Provisional Throughput capacity.

- Write Throughput
- Read Throughput

First 25 GB per month is free.
$0.25 per GB per month.

Query vs Scan.

* Query uses only primary key to find items in the table.
* Can also use a sort key attribute to refine the search results
* By default, Query returns all attributes in a table with the specified primary key, you can use "Projection Expression" parameter to get limited / some attributes.

** ScanIndexForward - Query results sorted by the key (default - true) Sort order is Ascending.
* Set false to reverse the order i.e descending.

* Query → Default → Eventually Consistent.

Scan.

* Examines every item in the table. Can use Projection Expression to refine.

Query VS Scan?

→ Obviously Query, as Scan always Scan the entire table. Scan cannot be used with large data sets because it will be slower.

* For Quicker response times, use the Query, Get or BatchGetItem API's.

DynamoDB Indexes & Streams.

Primary keys.

① Single Attribute (think Uniqueness ID)
- Partition key (Hash key) composed of one attribute
  Ex: Id field.

② Composite (Ex: Uniqueness ID & date range)
- Composed of two attributes.
  * Partition key + Sort key

* DynamoDB uses the Partition key's value as input to an internal hash function. The hash function determines the Partition i.e Physical location in which data is stored.

* Two items can have the same Partition key, but they must have a different Sort key. i.e for Composite PK.
"NO"

* All items with the same Partition key are stored together in Sorted Order by Sort key value.

Indexes. (Imp)

→ This limit can't be increased.

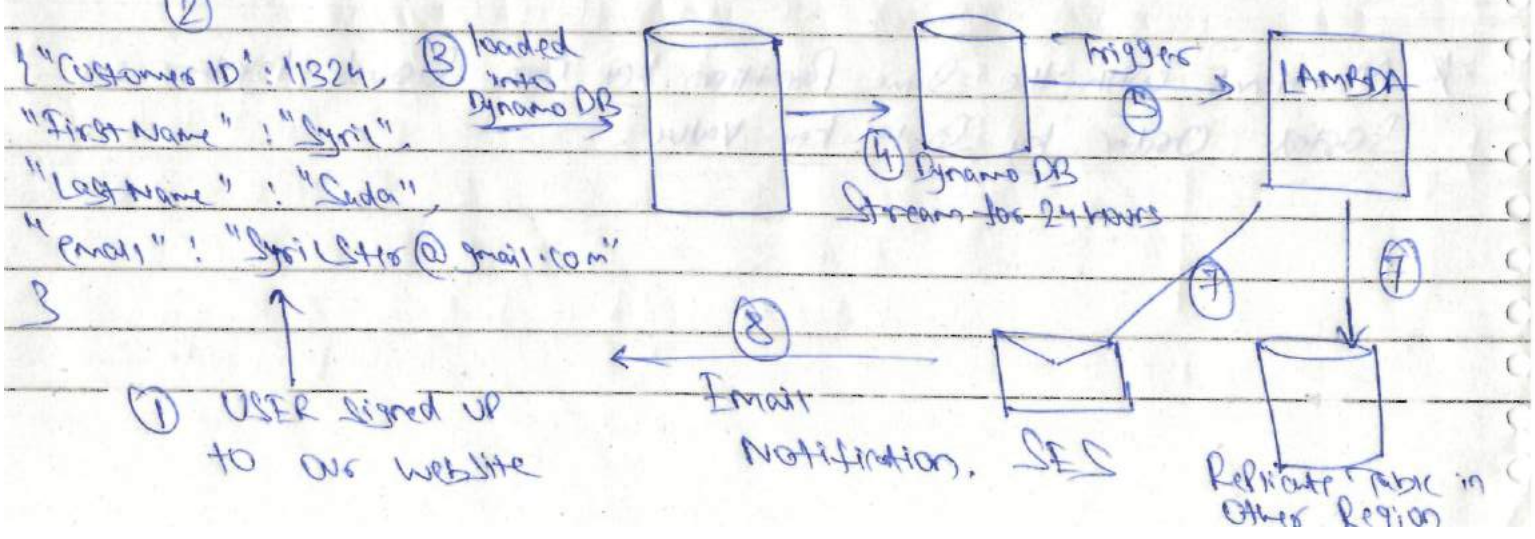
* Upto 5 LSI & 5 GSI per table. So max is 10.

① Local Secondary Index
- HAS the SAME Partition key, different Sort key.
- CAN ONLY be created when creating a table. They cannot be removed or modified later.

② Global Secondary Index
- HAS DIFFERENT Partition key & different Sort key.
- Can be created at table creation or added LATER.

DynamoDB Streams.

* Used to capture any kind of modification of the DynamoDB tables.
  - i) If item added → Stream captures an image of the entire item including all attributes.
  - ii) If item updated → Stream captures "before" and "after" image of any attributes modified in the item.
  - iii) If item deleted → Stream captures an image of the entire item before it was deleted.



Exam: Provisioned Throughput Calculations

① Unit of Read Provisioned throughput
* All reads rounded up to increments of 4KB
* Eventual Consistency reads - 2 reads Per Second.
* Strongly Consistent reads - 1 read Per Second.

② Unit of Write Provisioned throughput
* All writes are 1KB.
* All writes consist of 1 write Per Second.

Formula.

$$\left(\frac{\text{Size of read rounded to nearest 4KB (chunk)}}{4KB}\right) \times \text{no of items} = \text{Read throughput}$$

* Divide by 2 if eventually Consistent.

Question. You have an application that requires to read 10 items of 1KB Per Second using Eventual Consistency. What is read thput?

$$\frac{4}{4KB} \times 10 = 10.$$

Using Eventual Consistency we get 10/2 = 5.
So 5 units of read throughput.

Example 2: Application that requires 10 items of 6KB per second using eventual consistency. What should be the throughput?

6KB rounded to nearest 4KB chunk = 8KB

$$\frac{8KB}{4KB} \times 10 = 20$$

Eventual consistency so $\frac{20}{2} = 10$ units.

WRITE THROUGHPUT. (Simple multiplication)

Ex: You have an app that requires to write 5 items, with each item being 10KB in size per second. What should you set the write throughput to?

5 x 10 KB = 50 write units.

i.e write throughput = 50 units.

Error codes:

400 HTTP Status code – Provisioned Throughput Exceeded Exception

i.e You exceeded your maximum allowed Provisioned throughput for a table or for one or more global secondary indexes.

Web Identity Providers with DynamoDB.

→ Authenticate Using FaceBook, Google, Amazon etc Using AssumeRole with WebIdentity API

→ You will need to create a Role first.

Steps to Authenticate.

① User Authenticates with ID Providers (Such as FaceBook)
② They are passed a token by their ID Provider.
③ Your code calls AssumeRole with WebIdentity API and Provides the Providers token and Specifies the ARN for the IAM Role.
④ App can now access DynamoDB from between 15 minutes to 1 hour (default is 1 hour).

* i.e ① To Set up, Go to a table & then Access Control Tab.
② mention identity Provider & allowed attributes. Copy the Policy.
③ Create a new IAM role, Select Role for Identity Provider Access.
④ Attach a Custom Policy i.e the one which we Copied earlier to the new IAM Role.

DynamoDB – Key Facts.

* Conditional Writes.

i.e If Item = $10 then Update to $12.

Conditional writes are idempotent. Example, Suppose You issue a request to Update the Price of a book Item by 10%, with the expectation that the Price is currently 20$. However, before You

get a response, a network error occurs and you don't know whether your request was success or not. Because a conditional write is idempotent operation, you can send the same request again and DynamoDB will update the price only if the current price is still $20.

* Atomic counters: They are not idempotent i.e values are redundant.
  → Supports atomic counters, where you use the UpdateItem operation to increment or decrement the value of an existing attribute.
  * Atomic counters are not idempotent.
  * Don't use with critical applications which needs accurate counts / data.
  * Use it for cases like website visitor counter.
    i.e regardless of current count, increment the value.

Batch Operations :-

* BatchGetItem request to get multiple items.
* Can retrieve upto 1MB of data, which can contain as many as 100 items.

**

(SQS now supports FIFO)

SQS (Simple Queue Service)

→ It's a web service that gives you access to a message queue that can be used to store messages.
→ Amazon SQS is a distributed queue system.
→ Fail safe queue
** → 256 KB of text in any format.
→ Queue resolves issues that arise if the producer is producing work faster than the consumer can process it, or if the producer or consumer are only intermittently connected to n/w. i.e. Auto Scaling or failover.
* Supports multiple readers & writers.
* Does not guarantee first in, first out delivery of messages.
** → If order is required, you can place sequencing information now in each message, so that it can help in re-ordering.
⇒ Not relevant
→ Always polls.
* Asynchronously polls the task message from the queue.
* Visibility Time Out Check. → 12 hours window.

SQS AutoScaling

* Monitor when the queue grows rapidly fast, then set a AutoScaling threshold so that Auto Scaling group can come in and spin up multiple App services to bring down the messages in the queue.

Ex: Used in websites having to do image Encoding
Netflix.

* Max Retention Period for SQS message is 14 days.
* Create 2 SQS Queues - one for Premium members and one for free. Program your EC2 fleet to Poll Premium Queue first, if empty then Poll Free members Queue.
* SQS - Atleast once delivery of all messages in its Queue. Design for exactly once delivery.
* Billed at 64 KB Chunks. (i.e 4 bytes per message Chunk)
  $0.50 per 1 million Amazon SQS Requests per month.
* Single request can have from 1 to 10 messages, up to a maximum total Payload of 256 KB.

Exam Tips.

* SQS messages can be delivered multiple times and in any order.
* Default Visibility Time Out is 30 Seconds, Max Time Out is 12 hours.
* Extend visibility time out by using the "ChangeMessageVisibility" action to specify a new time out value.

SQS long Polling :-

→ Doesn't return a response until a message arrives in the Queue, or the long Poll times out.
→ Maximum long Poll time out = "20 Seconds".

Example Question : Polling in a tight loops is burning CPU cycles and costing the company money. How to fix this?
Ans : Enable long Polling.

SQS - Fanning Out : Using SNS Topic

* SNS will deliver the message to all the SQS Queues that are subscribed to the topic.
  i.e Pass msgs to multiple Queues. Message distribution.

(SQS + SNS) → Could be Combined.

Simple Notification Service. (Push-based Delivery - No Polling)

* Web Service to Send notifications from the Cloud.
* Pub-Sub messaging Paradigm.
* Notifications delivered to Clients using Push mechanism. As opposed to SQS which is Pull mechanism.

Exam
- Push messages out → SNS
- Poll a Queue → SQS.

* Notifications via Sms, Email, Amazon SQS or to any HTTP EndPoint.
* Messages Published to SNS Stored redundantly across multiple Availability Zones. (Fault Tolerance)

SNS Topic. :- Group multiple recepients using topic. One topic can support deliveries to multiple endPoint types.
Ex: Group IOS, Android and Sms recepients.
SNS will deliver appropriate formatted copies to each subscribers

*** (Topic name limited to 256 chars max)
* Formatting is JSON not xml.
* Subscription expries in 3 days if not confirmed.
* SNS message TTL for undelivered messages.

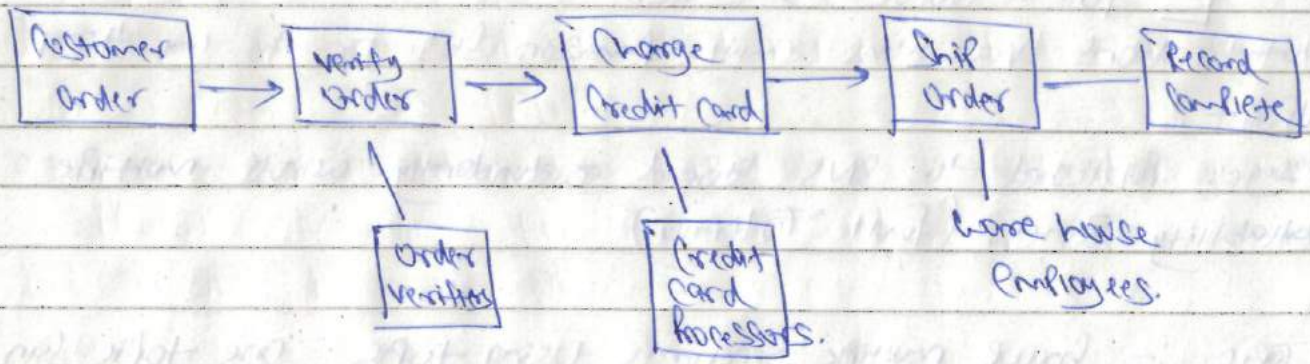
Message Email JSON.
- Topic ARN
- MessageId
- Message
- Subject
- Signature
- Signature Version
- TimeStamp
- Signing Cert URL

* Amazon SNS TTL - 4 weeks.
* Protocols → HTTP, HTTPS, Email, Email-JSON, SQS, Lambda.

Simple Workflow Service.

* Web Service to co-ordinate work across distributed application components.
  Ex: Media processing, business process workflow etc.

Example: (Used by Amazon offices)



* SWF Workers → Program that interact with SWF to get tasks, process received tasks and return results.
* SWF Deciders → Control co-ordination of tasks.
  i.e Ordering, Concurrency and Scheduling.

Main Diff b/w SQS.

* Workers and Deciders run on Cloud infra like Amazon EC2 or on machines behind fire walls. It ensures that task is assigned only once and is never duplicated.

SWF → maintains application state durably so workers and deciders do not have to keep track of execution state.
SQS → Task can be duplicated / assigned multiple times.

→ max 100 domains allowed.

* SWF Domains → Isolate a set of types, executions & task list from others within the same account.
  * Register Domain action in SWF API.
* Parameters Specified in JSON.
* Maximum workflow can be "1 year". value is in seconds.

* SWF vs SQS.

SWF	SQS
* Task Oriented API	* Message Oriented API
* Task assigned only once & not duplicated.	* Handles duplicated messages and message Processed only once.
* keeps track of all tasks & events in an application	* Implement own application level tracking.
* Human interaction types	* Application / message Delivery.
* Max workflow value - 1 year	* Max Time out - 12 hours.

* SNS Provides a guranteed one-time (atleast once) message delivery to SQS.

Navigate to last few pages.

Cloud Formation. (Management Tools related)

* Templated AWS Resource Creation.
* Cloud Formation is free, but based on the Stack selected, you can be charged. i.e For the EC2 Compute instance etc.
* Cloud Formation Template - JSON Script.
  {KeyName, DBName, DBUser, DBRootPassword ....}
* Output DNS name of your load Balancer Programatically
  In: GetAtt.
* Default (Auto) → CloudFormation will roll back and destroy other resources like EC2, Security Group etc.

Elastic Bean Stalk. (Infrastructure Provisioning)

* EBS is free, but Pay for the resources it creates.
* Pre Configured Env:
  Docker, Go, Java, IIS, NodeJS, PHP, Python, Ruby and Tomcat.

Virtual Private Cloud (VPC)

* VPC is a logical data center. This can span across availability zone.
* Have a virtual networking environment, including selection of own IP address range, creation of subnets etc.
* N/w configuration can be customized.
  Ex: Create a public-facing subnet for web servers that can access Internet. Place DB or app servers in private-facing subnet with no internet access.
* Hybrid cloud possible. ie Hardware VPN b/w your corporate data centers and your VPC to leverage AWS cloud.

Subnets — logical division of IP addresses.
Ex: 10.0.0.0 – 10.255.255.255 (10/8 Prefix) → Enterprise Internal
172.16.0.0 – 172.31.255.255
192.168.0.0 – 192.168.255.255 → Home N/w.
VPC – 10.0.0.0/16. Largest N/w size is /16.
Public Subnet → Internet accessible. Ex web servers
Private Subnet → not accessible by Internet. Ex: DB Servers

** One Subnet is directly mapped to an Availability Zone, "can't" span across multiple AZs.

** One Subnet per Network ACL.

What can you do with a VPC ?

* Launch instances into a subnet of your choice.
* Assign custom IP address ranges in each subnet.
* Configure route table b/w subnets.
  * route table defines if a subnet is public or private.
* Create a internet gateway and attach to VPC.
* ** → only one internet gateway per VPC.
* Better security control over AWS resources. i.e subnets to host specific IP address. (ACL's)
* Stateful security groups.
* Subnet ACL's — They are stateless. i.e open both inbound & outbound ports.

Default VPC vs Custom VPC.

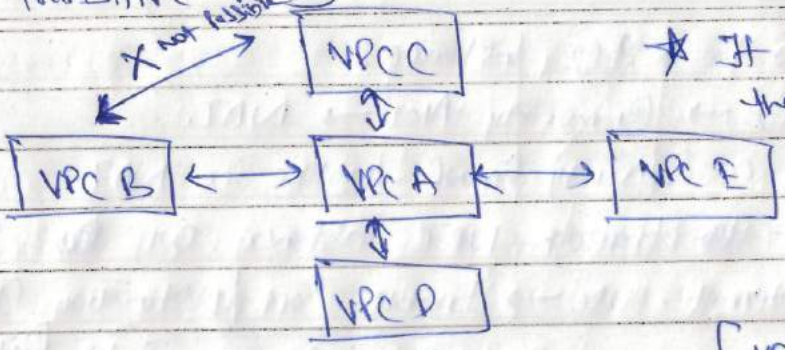
→ Default VPC by amazon to help deploy EC2 instances.
→ All subnets in default VPC is public. i.e Access to internet.
→ EC2 — Public & Private IP address.
→ Delete — Default VPC, then contact AWS to get it back.

VPC Peering.

* Connect one VPC to another via direct N/W.
* Peer VPC with other AWS account as well as with other VPC's in same account
* Peer is in a star configuration. NO TRANSITIVE PEERING.

Ex: i.e 1 central VPC Peers with 4 others.

* 'Transitive Peering'



* If VPC B needs to talk to VPC C then only way is to have a direct N/w. It can never communicate via other VPC i.e here VPC A.
[VPC B → VPC A → VPC C]

** When a VPC is created :-
- A main Route table is created automatically
- Default Security Group
- Default N/w ACL

Internet Gateway :- To make a Subnet either Public or Private.
* Cannot attach multiple Internet Gateway to a VPC.

Route Table → Create a route out to the internet using a Internet Gateway and then attach Subnets.

* Subnet Auto assign Public IP → Then only the EC2 instance will have a auto IP assigned
(Or) Create an 'Elastic IP' and assign to EC2 instance

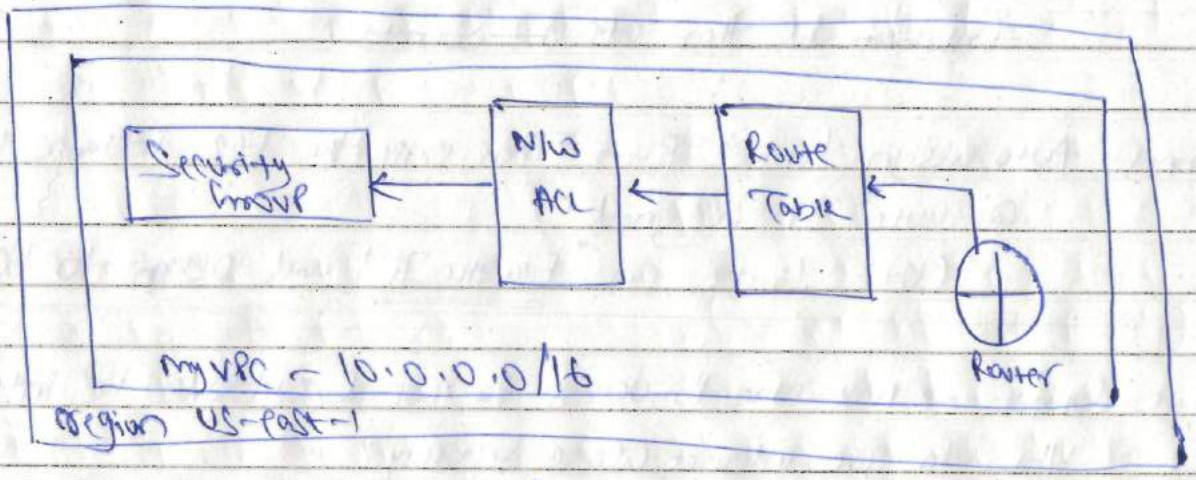
* Private Subnet → Not able to access the world outside i.e internet
How to give internet access securely?
(1) NAT Gateway - N/w Address Translation
(2) NAT Instance - But Single Point of Failure.

* NAT instance is going to be a EC2 instance.
  i.e Console → EC2 → Community AMI → NAT.
  → NAT instance needs a security group unlike a NAT.
  i.e Routing traffic through NAT instance out to the internet. use → "Enable Src/Destination Check" on NAT instance. i.e Disable this option
* NAT instance allow to use HTTP & HTTPS i.e configured in Security Group.

* Route Table → Allow NAT instance in the routes tab.

NAT Gateway → Specify a Subnet. It should be a Public one. Also make Route Table config.
  * Auto Security Group.
  * Amazon maintains for you.
  * Better For Production APPS.

* Default Diagram when you Just Create a VPC.



→ After this we Created 2 Sub nets.
  i) 10.0.1.0 - US-East-1a
  ii) 10.0.2.0 - US-East-1b
  } Named as per availability zone.

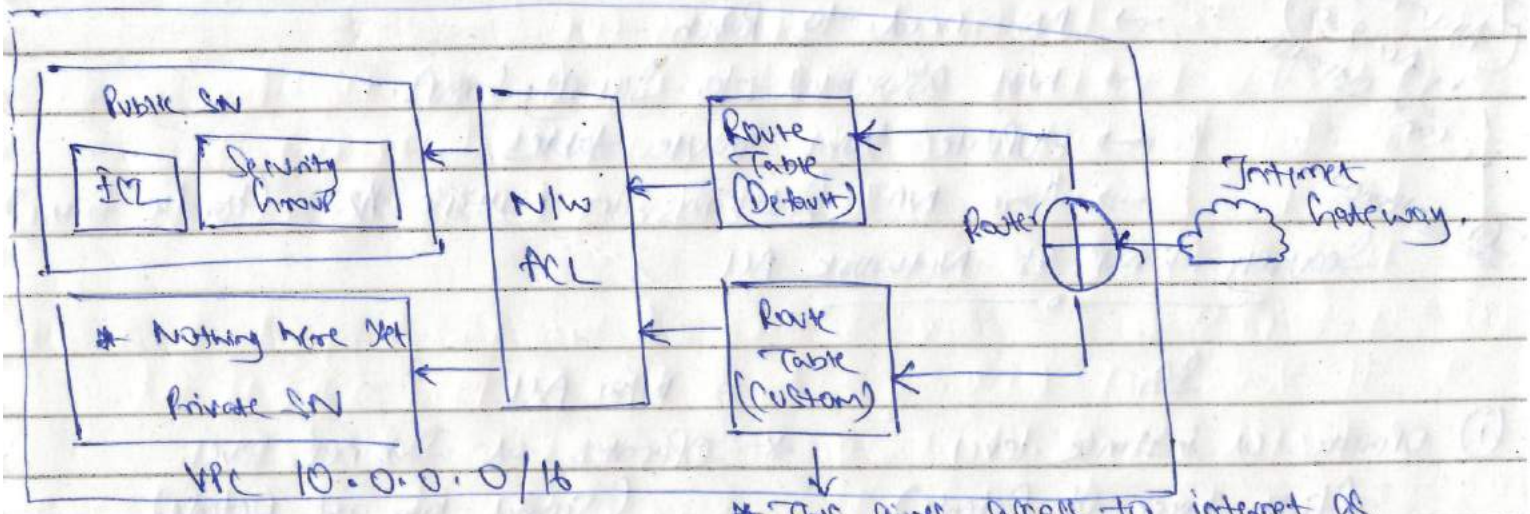
→ Out of these 2, we will make one as Public SN and Others as Private SN.

(1) Create a Internet Gateway and attach it to Your VPC.
ie One IG Per VPC.

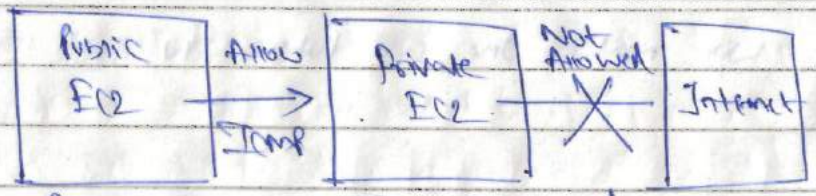
(2) Create a route table and associate 0.0.0.0/0 ie all traffic to the Internet Gateway.

(3) Associate a Subnet to the route table which You want to make a Public Subnet.

(4) Go to the Subnet and Enable Auto Assign Public IP's so that when we create a EC2 instance on this Subnet, we get a Public IP assigned.



* Icmp → Allow this in the Security group so that You can access the Private EC2 machine from your Public one in the same VPC.



Public IP — SSH to Private IP

Private IP → For this to happen we need a NAT Gateway.

* NAT instance – Auto Scaling Group to avoid Single Point of failure.
  - → NAT instance should be in a Public Subnet.
  - → Disable Source/Dest Check on the instance.
  - → Behind a Security Group.
* NAT Gateway :- Preferred by the Enterprise.
  - → Scale Automatically upto 10Gbps.
  - → No need to Patch
  - → Not Associated to Security Groups.
  - → Update your route tables.
  - → Have NAT Gateways in multiple AZ's. (Handle failure)

(All managed by Amazon) ie can't customize.

Job interview.

Security Group vs Network ACL

	SG	N/w ACL
(1)	Operates at instance level (first layer of Defense)	* Operates at Subnet level. (Second layer of Defense)
(2)	Supports allow rules only	* Allow and deny rules.
(3)	Is Stateful : Return traffic is automatically allowed	* Stateless : Return traffic is explicitly allowed by rules
(4)	Per Instance	* Apply to all instances in this Subnet.

Required on outbound rules only - on NACLs.

* Ephemeral Ports: To cover different types of clients that might initiate traffic to public-facing instances in your VPC, you can open ephemeral ports 1024 - 65535.
  i.e ACL → Outbound rule → Custom TCP Rule → Port Range 1024 - 65535.

* N/w ACL rules are evaluated according to the order.
  i.e 99, 100, 200 etc. (lowest numbered rule first)
* VPC comes with a default N/w ACL and default it allows all outbound & inbound traffic.
* Each subnet in a VPC should have a network ACL or else it will have a default N/w ACL.
* Network ACL has separate inbound and outbound rules, each rule can either allow or deny traffic.
* Custom i.e User defined NACL's denies all inbound & outbound traffic.
* Block IP Address using Network ACL's and Security Group's.

ELB. [Elastic load Balancer]

* If you want high availability → Have 2 public subnets.
  i.e one subnet in an Availability Zone.

NAT vs Bastion.

* Bastion → open up a public route to internet for a private subnet using a Bastion like firewall.
* Have a Bastion in each public subnet, with atleast 2 subnets across AZ for high availability.

* NAT is used to provide internet traffic to EC2 instances in private subnets. (Ex: MySQL)
* Bastion is used to securely administer EC2 instance (SSH or RDP) in private subnets. (Remember the lab where we stored the .pem file & then did the SSH to private subnet? bastion instance should be used instead of that.)

VPC Summary.

* VPC - logical data center in AWS. VPC can't be in different regions its only different AZ.
* 1 Subnet - 1 Availability Zone.
* When creating NAT instances - Disable Source / Destination Checks.
* Amount of traffic NAT instances support, depends on the instance size. If you are bottlenecking, increase instance size.
  Ex: t2 micro to something else.
* Create high availability using Autoscaling Groups, multiple subnets in diff AZ's and a script to automate failovers.

Resilient Architecture.

* If you want resiliency, always have 2 public subnets and 2 private subnets. Make sure each subnet in diff AZ's.
* With ELB's make sure they are in 2 public subnets.
* With Bastion hosts, put them behind an Autoscaling Group with a min size of 2. Route 53 for automatic failovers. (ie health checks)

* Giving Internet Access to a Private Subnet.

(1) Create a NAT instance. Launch instance → Community AMI's.
(2) for NAT instance, Actions → Networking → Disable Source/Dest Check.
(3) If the above step is not enabled, then you wont be able to see this NAT in the route table routes.
(4) Go to the main/default route table. Routes tab and create a route out to the internet i.e 0.0.0.0/0 for the NAT.
(5) Go to Security Groups and add HTTPS in the inbound rules. This is recommended by Amazon.

After above steps, SSH to Public Subnet, have your .Pem Key Pair there and then,

SSH ec2-user@Private-IP -i KeyPair.Pem
Then Yum Update -y should work.

* VPC Flow logs in the next notes.

IAM Continued (STS - Security Token Service)

→ Grants users limited & temporary access to AWS resources. Users can come from 3 sources:
i) Federation (typically Active Directory)
ii) Federation with mobile Apps
iii) Cross Account access - ie users from one AWS account access resources in another.

Key Terms:

① Federation: Combine or Join a list of users in one domain (such as IAM) with a list of users in another domain (Active Directory, Face book etc)

② Identity Broker: Service that allows you to take an identity from Point A and join it (federate it) to Point B. ✱ This need to be developed by ourselves

③ Identity Store - Services like Active Directory, Face book Google etc

④ Identities - A user of a service like face book etc.

Steps
① Develop an identity broker to communicate with LDAP and AWS STS. ie take user/Pass from front end & send to
② Identity Broker always authenticates with LDAP first, THEN with AWS STS. STS
③ App then gets temporary access to AWS resources.

Cloud Formation. (Infrastructure Templating)

→ Allows you to take traditional h/w infrastructure and convert it into code. (Think of Ansible Playbooks here)
→ Gives developers & Sys administrators an easy way to create and manage a collection of related AWS resources i.e Provisioning & Updating
* No need to worry about AWS services dependencies. Cloud Formation takes care of this for you.
* Version control to AWS infrastructure. Same like S/w applications.
* Cloud Formation template → JSON or YAML format

Ex: JSON

```
{ "Resources" : {
    "HelloBucket" : {
      "Type" : "AWS::S3::Bucket"
    }
  }
}
```

⁂ This is the only mandatory field.

* Fn::GetAtt to output data. Ex: O/P a Public IP.
* Cloud Formation → Create Stack. (Follow the intuitive UI)
* ListStackResources – Describe all resources of a specified stack.

Exam Tips:

* Automatic rollback on error → Enabled by default.
* You are charged for errors. (Ex: EC2 instances UP before er
* Cloud Formation is Free. Only resources used are charged.
* Stacks can wait for applications to be provisioned using the "Wait Condition".

* Fn: GetAtt → output Data
* Route 53 is completely supported. A Records, Aliases etc
* IAM role is also supported.
* Parameters section of CF template for intrinsic reference of function.
* Cloud Formation also provides a set of application bootstrapping scripts that helps in installing packages, files & services on EC2 instance.
* Deletion policy can be set for the resources.
  Ex: Preserve S3 buckets when stack is deleted.

Elastic Bean Stalk (Abstract focus towards infrastructure)

→ GUI for AWS. AWS will provision the infrastructure for you. For people who don't know AWS services / configurations i.e. Simplify infra mgmt. Easy application deployments.

→ Resources provisioned by EBS is charged. EBS as such is a free service.

* You can have multiple versions of your applications
* Your applications can be split into tiers (Web Tier / DB Tier)
* You can update your configuration. i.e. change EC2 instance later etc.
* If EBS creates your RDS DB then it will delete it when you delete your applications. If not then RDS stays.

Apache Tomcat for Java Apps
Apache HTTP Server for PHP applications
——" —— for Python applications
Nginx or Apache HTTP Server for Node.js
Passenger or Puma for Ruby applications.
Microsoft IIS 7.5, 8.5 & above.
Java SE
Docker
Go.