**Software Development Kits (SDK) -**

IOS, Android, Browser (Java scripts), Java, .NET, Node.js, PHP, Python, Ruby, Go, C++

**Simple Queue Service (SQS)**

 - message oriented API

SQS - Message can contain upto 256KB of text, billed at 64KB chunks,

Single request can have 1 to 10 messages unto maximum of 256KB payload

Even though there is one message of 256Kb its basically 4 request for billing since (4 \* 64KB)

NO ORDER - SQS messages can be delivered multiple times in any order

Design - you can have 2 priority queues for priority based message one for higher and other for lower priority

EC2 instances always poll for messages from the queue (pull from the queue and not push)

Visibility timeout always start from when the application instance polled the message.

Great design - Visibility timeout expires that means there is a failure somewhere since that message was polled but not processed and hence not deleted so other some other process will poll the message again and visibility timeout starts again.

Visibility timeout by default is 30 Seconds up to 12 hour maximum (ChangeMessageVisibility) / maximum visibility

Maximum long polling timeout 20 seconds (http://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-long-polling.html) —ReceiveMessageWaitTimeSeconds

Messages in the Queue can be retained for up to 14 days

First 1 million request ares free, then $0.50 PER EVERY MILLION REQUESTS

**Simple Notification Service (SNS)**

works on a publish - subscribe model, SNS notifies the message, and hence push based approach. Inexpensive pay as you go

CloudWatch or Autoscaling triggers SNS

SNS can notify to Email, Text / SMS, SQS or any HTTP end point.

protocols: HTTP, HTTPS, EMAIL, EMAIL-JSON, SQS or Application - messages can be customized for each protocol

SNS messages are stored redundantly to multiple AZs

SNS Dataformat - JSON (Subject, Message, TopicArn, MessageId, unsubscribeURL etc..)

$0.50 per 1 million SNS request

Different price for different recipient types

to HTTP: $0.06 / 100,000 notifications deliveries

to EMAIL: $2 / 100,000 notifications deliveries

to SMS: $0.75 / 100 notifications deliveries

http://docs.aws.amazon.com/sns/latest/dg/mobile-push-send-devicetoken.html (CreatePlatformEndpoint API)

Message Example:

**SWF** - task oriented API

Simple Work flow - human interaction to complete order or collection of services to complete a work order.

Workers - interact with SWF to get task, process received task and return the result

Deciders - program that co-ordinates the tasks, i.e. - ordering, concurrency and scheduling

Workers and Deciders can run independently

TASK is only assigned ONCE and NEVER DUPLICATED (key difference from SQS where messages can be processed multiple times)

SWF Domain - think of it as a container for the work flow. you can register a domain by Console or API

Maximum workflow processing time can be 1 year (equivalent seconds) - SQS is 12 hours processing time

**CloudFormation**

Use of CFT, Beanstalk and Autoscaling are free but you pay for the AWS resources that these services create.

Fn::GetAtt - values that you can use to return result for an AWS created resource or used to display in output

By Default - rollback everything on error

Infrastructure as a code, Version controlled, declarative and flexible

**Intrinsic Functions**

Fn::FindInMap - returns the value corresponding to keys in a two-level map that is declared in the Mappings section.

Fn::GetAtt - returns the value of an attribute from a resource in the template.

Fn::ImportValue - returns the value of an output exported by another stack.

Fn::Join - appends a set of values into a single value, separated by the specified delimiter. If a delimiter is the empty string, the set of values are concatenated with no delimiter.

Fn::Select - returns a single object from a list of objects by index.

Fn::Split – used to split a string into a list of string values so that you can select an element from the resulting string list

Fn::Sub - substitutes variables in an input string with values that you specify

Ref - returns the value of the specified parameter or resource.

* When you specify a parameter's logical name, it returns the value of the parameter.
* When you specify a resource's logical name, it returns a value that you can typically use to refer to that resource, such as a physical ID.

**Cloud Formation Templates**

{

"AWSTemplateFormatVersion" : "*version date*",

"Description" : "*JSON string*",

"Metadata" : {

*template metadata*

},

"Parameters" : {

*set of parameters*

},

"Mappings" : {

*set of mappings*

},

"Conditions" : {

*set of conditions*

},

"Transform" : {

*set of transforms*

},

"Resources" : {

*set of resources*

},

"Outputs" : {

*set of outputs*

}

}

Description (optional) - A text string that describes the template. This section must always follow the template format version section.

Metadata (optional) - Objects that provide additional information about the template.

Parameters (optional) - Values to pass to your template at runtime (when you create or update a stack). You can refer to parameters from the Resources and Outputs sections of the template.

Mappings (optional) - A mapping of keys and associated values that you can use to specify conditional parameter values, similar to a lookup table. You can match a key to a corresponding value by using the Fn::FindInMap intrinsic function in the Resources and Outputs section.

Conditions (optional) - Conditions that control whether certain resources are created or whether certain resource properties are assigned a value during stack creation or update. For example, you could conditionally create a resource that depends on whether the stack is for a production or test environment.

Transform (optional) - For serverless applications (also referred to as Lambda-based applications), specifies the version of the AWS Serverless Application Model (AWS SAM) to use. When you specify a transform, you can use AWS SAM syntax to declare resources in your template. The model defines the syntax that you can use and how it is processed.

Resources (required) - Specifies the stack resources and their properties, such as an Amazon Elastic Compute Cloud instance or an Amazon Simple Storage Service bucket. You can refer to resources in the Resources and Outputs sections of the template.

Outputs (optional) - Describes the values that are returned whenever you view your stack's properties. For example, you can declare an output for an S3 bucket name and then call the aws cloudformation describe-stacks AWS CLI command to view the name.

**ElasticBeanstalk**

Its uses ASG,ELB,EC2,RDS,SNS and S3 to provision things.

Environment Tier - Webserver, Worker

Predefined Configurations - IIS, Node.JS, PHP, Python, Ruby, Tomcat, Go, .NET,

preconfigured docker: Glassfish, Python or generic docker

Environment URL - has to be unique

Dashboard - Recent events, Monitor, Logs, Alarms, Upload and Deploy and Configurations

Configuration - Scaling, Instances (DIRTMCG instance types, key pair), Notifications, Software configuration (e.g. PHP.ini), Networking tier (ELB, VPC config), Data tier(RDS)

Environment properties (Access key and secret key as parameters)

**DynamoDB**

fast - flexible No sql database - single digit ms latency, fully managed, supports document and key-value (web, gaming, ad-tech, IOT)..

Table, Item (row), attribute (key - value)

Eventual Consistent Reads vs Strongly Consistent Reads

Read Capacity Units, Write Capacity Units (can be scaled up) - push button scalability

Writes are written to 3 different location / facilities/ datacenter (synchronous) - Amazon DynamoDB synchronously replicates data across three facilities in an AWS Region, giving you high availability and data durability.

Two types of primary key -

(1) Single Attribute (think unique id) - Partition Key (Hash Key) composted of 1 attribute (no nesting allowed here) - Partition key will help determine the physical location of data.

(2) Composite key (think unique id and range) - Partition Key(Hash Key) & Sort Key (Range key - e.g date) - composed of 2 attributes - if two data have same partition key (same location) it must have a different sort key, and they will be stored together on single location.

Secondary Indexes

(1) Local Secondary Index - Same Partion Key + Different Sort Key ( can only be created while creating the table, cannot be added/removed or modified later)

(2) Global Secondary Index - Different Partition Key + Different Sort Key ( can be created during the table creation or can be added later or removed / modified later)

DynamoDB Streams

use to capture any kinda modification to the dynamo db table, Lambda can capture events and push notifications thru SES

Table can be exported to csv (either select all items )

Query vs Scan

Query operation finds item in a table using only primary key attribute values , must provide partition attribute name and the value to search for, you can optionally provide a sort key attribute name and value to refine search results (e.g. all the forums with this ID since last 7 days). By default Query returns all the data attributes for those items with specified primary keys. You can further use ProjectionExpression parameter to only return a selected attributes.

Query results are always sorted by the sort key (ascending for both numbers and string by default). To reverse the sort order set the ScanIndexForward parameter to false

By Default Queries are going to be Eventually consistent but can be changed to StronglyConsistent.

Scan operation is basically examines every item - e.g. dumping the entire table, by default Scan returns all the data attributes but we could use ProjectionExpression parameter to only return a selected attributes.

Query operation is more efficient than scan operation

For quick response time, design your table in a way that you can use Query Get or BatchGetItem API (read multiple items - can get up to 100 items or up to 1MB of data) ,

Alternatively design your application to use scan operation in a way that minimize impact of your table’s request rate since it can use up the entire table’s provisioned throughput in a single scan operation

**DynamoDB Provisioned Throughput calculations**

Items == rows

Read Provisioned Throughput

* All units are rounded up to 4KB increments
* Eventual Consistent reads (default) consist of 2 reads per second
* Strongly Consistent reads consist of 1 read per second

( Size of Read Rounded to nearest 4KB Chunk / 4 KB \* no of items ) / 2 <— if eventual consistency

( Size of Read Rounded to nearest 4KB Chunk / 4 KB \* no of items ) / 1 <— if strongly consistency

Write Provisioned Throughput

* All units are rounded up to 1KB increments
* All writes consist of 1 write per second

( Size of write in KB \* no of items ) / 1

When you exceed your maximum allowed provisioned throughput for a table or one or more global secondary index you will get 400 HTTP Status code - ProvisionedThroughputExceededException

AssumeRolewithWebIdentity role

Idempotent = conditional write

Atomic counters - always need to increment so its not idempotent

if data is critical and no margin of error, then must use Idempotent conditional write.

http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Limits.html#limits-tables

Only Tables (256 table per region) and ProvisionedThroughput (80 K read, 80K write per account for US east, 20K for other regions) limits can be increased

http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/QueryAndScanGuidelines.html (Reduce Page Size for Scan operation and Isolate Scan Operation)

**Simple Storage Service (S3)**

* secure, durable, highly scalable object store (0 bytes to 5TB), universal namespace (must be unique bucket - regardless of regions),object based key value store, VersionID, Metadata, ACL
* The total volume of data and number of objects you can store are unlimited. Individual Amazon S3 objects can range in size from 0 bytes to 5 terabytes. The largest object that can be uploaded in a single PUT is 5 gigabytes. For objects larger than 100 megabytes, customers should consider using the Multipart Upload capability. it mean the largest single file into S3 is 5G, but after the 5G files are in S3, they can be assembled into a 5T file,
* You can use a Multipart Upload for objects from 5 MB to 5 TB in size (Exam question, scenario where more than 5GB file needs to be uploaded)
* object based storage vs block based Storage (EFS)
* data is spread out in multiple facilities, you can loose two facilities and still have access to files
* For PUTS of New Objects (Read after Write Consistency), For Overwrite PUTS and DELETE (Eventual Consistency)
* http://docs.aws.amazon.com/general/latest/gr/aws*service*limits.html#limits\_s3 ( Number of S3 bucket limit per account — 100)

Storage Tiers/ Class

S3 Standard - Durability (11 9s), Availability (99.99 %) - reliable regular for just about everything

S3 IA (Infrequent Access) - Durability (11 9s), Availability (99.9 %) - accessed every 1 month to 6 months or so (infrequent) but rapid access and low retrieval time (few ms)

S3 RRS(Reduced Redundant Storage)- Durability (99.99%), Availability (99.99 %) - less durability (data that can easily be regenerated - e.g thumbnails) - cheapest of all s3, less fault tolerant then the other two since you are willing to loose the data, reproducible data

Glacier - for archival only (3 to 5 hours restore time)

S3 price - charged for Storage, number of requests, data transfer (tiered so more you use less charge)

bucket name has to be all lowercase letters

S3 for static website hosting (Static Website Hosting > Enable website hosting) - no dynamic

Any time you create a bucket nothing is publicly accessible / Any time you add an object to a bucket its private by default (you will get 403) > Make the files public (even for public hosting)

every object inside the bucket can have different storage class (S3 standard, S3-IA, S3-RRS) and you can turn on server side encryption (AES - 256)

regular bucket link: https://s3-eu-west-1.amazonaws.com/ankittest <— https

bucket with Static website hosting: http://ankittestsite.s3-website-eu-west-1.amazonaws.com <— http (has to be for static hosting), you can turn it into SSL / https with cloudfront though

CORS (Cross Origin Resource Sharing) - to avoid the use of proxy

Versioning - once enable you cannot disable versioning / although it can be suspend it , if you want to turn it off delete the bucket and recreate (version id)

Once you delete the delete marker, you can get the file back that you have deleted while versioning on

every version is stored separately in the bucket for each version / might not be a good choice for cost perspective for large media files., multiple updates use case also not ideal for versioning.

Versioning’s MFA Delete Capability can be used to provide additional layer of security.

Cross Region Replication - (requires versioning enabled on source and destination buckets)

you can enable - need source and destination bucket (create a new bucket, source bucket will not show up on drop down of destination)

Existing objects will not be replicated, only new objects will be replicated across the region

Lifecycle management in S3

(1) when versioning is disabled

Transition to IA S3 - min 30 days and has a 128KB minimum of object size

Archive to Glacier - min 1 day if IA is not checked, min 60 day if Transition to IA S3 is checked

Permanently Delete - min 2 day if IA is not checked and 1 is selected for Glacier, min 61 day if IA is selected 30, Glacier is selected 60.

(2) when versioning is enabled you have lifecycle management options to take action on previous version as well as current version.

Security and Encryption in S3

* by default newly created buckets are private
* Access control using Bucket Policies (entire bucket) and ACL(individual objects and folders)
* access logs - all the request made to S3 buckets, to another bucket or another account’s S3 bucket

Encryption

(1) In Transit - SSL / TLS

(2) Data at rest

Server Side Encryption

* SSE- S3 Server Side Encryption with S3 managed keys, (amazon AES 256 handled for you) - click on the object and encrypt
* SSE - KMS - AWS Key management services , managed keys - additional charges / audit trail of keys, amazon manage keys
* SSE - C - Server side encryption with Customer provided keys - you manage encryption keys

Client Side Encryption

* you encrypt the data on client side and upload to s3

Every non-anonymous request to S3 must contain authentication information to establish the identity of the principal making the request. In REST, this is done by first putting the headers in a canonical format, then signing the headers using your AWS Secret Access Key.

You can insert a presigned url into a webpage to download private data directly from S3.

The object creation REST APIs (see Specifying Server-Side Encryption Using the REST API) provide a request header, x-amz-server-side-encryption that you can use to request server-side encryption.

S3 Transfer Acceleration

Utilize local edge locations to upload content to S3 - incur extra cost

further away you are the more benefit you get (faster)

GateWay

(1) Gateway stored volumes - entire dataset is stored onsite and asynchronously backed up to S3

(2) Gateway cached volumes - Most frequently used data is stored onsite and entire dataset is stored on S3

(3) Gateway Virtual Taped library - Used for backup if you don’t want to use Tapes, like Netbackups etc..

Import Export

Import / Export Disk

* Import to S3, EBS, Glacier
* export from S3

Import / Export Snowball

* Import to S3
* Export to S3

S3 stored data in alphabetical / lexigraphical order. so if you want to spread the load across S3, filename should not be similar (Optimize performance)

**CloudFront**

Content Delivery Network - edge locations, reduced latency, traffic serves from the closest nodes

Edge locations - content will be cached (over 50), different from region / AZ. TTL (speed of image // media is quicker - first user suffers the performance), can be not only read only (you can write it)

Origin can be - S3, EC2, ELB, Route53 also NON AWS origin server ,

Distribution - name given to the CDN consist of collection of Edge locations

(1) Web Distribution -

(2) RTMP (media streaming / flash) Distribution - for Adobe flash files only

you can have multiple origins of a distribution

Path Pattern (\*)

Restrict viewer access by signed URL or Signed Cookies

Restrict content based on geo location (whitelist and blacklist)

Create invalidate - invalidate TTL (you pay for it) like purge in Akamai

**VPC** - logical datacenters in AWS

Can span multiple AZ, but can’t span multiple regions, PEER VPC, but no Transitive Peering

Custom VPC has to be /16 can’t go higher then that /8 is not allowed

When you create Custom VPC it creates default security group, default network ACL and default route table., it doesn’t create default Subnet

One Subnet == one AZ, you can have security group spanning multiple AZ, ACL’s span across AZ (assign sg and ACL to two different subnets)

any CIDR block 5 reserved IPs (.0, .1, .2, .3, .255)

so for CIRD block /24: 2^8 - 5 = 256 - 5 = 251 available IP address space

when you create internet gateway, by default its detached, attach it to VPC then, only 1 IGW per VPC

When you create a VPC Default Routetable(Main Routable) is created where the default Routes are,

10.0.0.0/16 Local <— all subnets inside VPC will be able to talk to each other

Don’t touch Main route table

Create another routetable for route out to internet (0.0.0.0/0 IGW) <— route out to the internet

Last thing you associate this new route table to one of the subnet which will make it public. (you can enable auto assign public IP for the public subnet)

1 subnet can have 1 routetable

ICMP is for ping / monitor

NAT instance and NAT gateway

NAT Instance - disable source / destination check., always behind security group, must be in public subnet, must have an EIP, ,must be a route out of the private subnet to NAT

Increase the instance size if bottleneck

Change the main route table - add a route (0.0.0.0/0 NAT Instance target)

NAT Instance is a single point of failover (put it behind a ASG),

NAT gateway - released in 2016 - amazon handled

* Amazon maintains it for you, no need to handle yourself. (security patches applied by AWS)
* You can just create the gateway and assign EIP (put it in public subnet) (automatically assigned)
* Change the main route table - add a route (0.0.0.0/0 NAT gateway target)
* No need for disable source/destination check or no need to put it behind a security group - it handles it for you.
* Highly available / redundancy no need for ASG.NAT gateways are little bit costly - always use it in production scale automatically up to 10Gbps

**NACL vs SG**

Security groups are stateful - any inbound rule , applies to outbound as well (Only Allow rules)

* by default all inbound deny, all outbound allow
* can span across AZ

ACL are stateless -

* For default ACL, all inbound and outbound rules are allowed by default - associated with all subnets in VPC by default
* for Custom ACL, all inbound and outbound traffic is denied by default - not associated with any subnet
* 1 subnet is only associated with ACL. granular rules for ACLs, numbered rules (recommended steps of 100)
* rule number 99 takes precedence over rule number 100 (if 99 is blocked and 100 is allowed) 99 will be executed.
* Can SPAN across AZ

Ephemeral port - 1024 - 65535 should be allowed to take traffic.

If you want to BLOCK IP address then must use ACL, because security group doesn’t have deny

Bastion - keep it in public subnet to allow SSH / RDP into instances into private subnets (High availability - Bation in two public subnets and also ASG - Route 53 running Health checks on those Bastion)

VPC Flowlogs: to capture all the traffic information into logs - logs everything (create IAM role and create cloud watch log group - and log stream)

VPC Cleanup: can’t delete VPC if you have active running instance or ELB is running

**Error Codes**

HTTP 3XX – Redirection

HTTP 4XX - Client Side Error

HTTP 403 – Access Denied

HTTP 404 – NoSuchBucket, NoSuchKey, NoSuchLifecycleConfiguraion, NoSuchUpload, NoSuchVersion

HTTP 409 – BucketNotEmpty (The bucket you tried to delete is not empty)

HTTP 5XX - Server Side Error