***AWS Introduction***

Global Infrastructure

* Region: Geographical area that has two or more Availability Zones. There are 11 regions and 30 availability zones. An independent collection of AWS computing resources in a defined geography.
* Availability Zone: Data Center. Distinct locations from within an AWS region that are engineered to be isolated from failures.
* Edge Location: Are Content Distribution Networks (CDN) endpoints for Cloud Front. There are over 50 edge locations.

Networking

* VPC: Virtual Private Cloud. A virtual data center that exists within the AWS account. It is a separate and isolated set of resources from a logical perspective.
* Direct Connect: Way of connecting to the AWS environment without using Internet.
* Route53: Amazon DNS service.

Compute

* EC2: Virtual server.
* EC2 Container service: EC2 with Docker.
* Elastic Beanstalk: Deploy and scale web applications.
* Lambda: Let you run code without provisioning servers.

Storage

* S3: Object based storage.
* Cloud Front: CDN service.
* Glacier: Offline storage. Archaic storage.
* EFS: Elastic File System. Block level instead of Object level.
* Snowball: Import and export hard disk service.
* Storage Gateway: It connects a cloud storage software device in your organization with AWS storage infrastructure.

Database

* RDS: Relational Database Service.
* DynamoDB: No SQL database service.
* Elasticache: Releaving the stress on RDS.
* Redshift: Amazon data warehousing service.
* DMS: Database migration service.

Analytics

* EMR: Elastic Map Reduce
* Data Pipeline: Moving data from one server to another
* Elastic Search: Deploy, operate, and scale elastic search on the AWS cloud.
* Kinesis: Streaming data on AWS.
* Machine Learning: Predictive analytics
* Quick Sight: Business intelligence service

Security and Identity

* IAM: Identity Access Management.
* Directory Services: Connect AWS resources with Microsoft Active Directory or with an independent new directory in the cloud.
* Inspector: Install agents to inspect security vulnerabilities.
* WAF: Web Application Firewall
* Cloud HSM: Hub way Security Module. Securing your cloud based infrastructure using HSM devices.
* KMS: Key Management Service.

Management Tools

* Cloud Watch.
* Cloud Formation.
* Cloud Trail.
* Opsworks.
* Config.
* Service Catalog.
* Trusted Advisor.

Application Services

* API Gateway.
* AppStream.
* CloudSearch.
* Elastic Transcoder.
* SES: Simple Email Service.
* SQS: Simple Queue Service.
* SWS: Simple Work Flow service.

Developer Tools

* CodeCommit
* CodeDeploy
* CodePipeline

Mobile Services

* Mobile Hub.
* Cognito.
* Device Farm.
* Mobile Analytics.
* SNS: Simple Notification Service.

Enterprise Applications

* WorkSpaces.
* WorkDocs.
* WorkMail.

Internet of Things

***Identity Access Management - IAM***

What does IAM give you?

* Centralized control of your AWS account.
* Shared access to your AWS account.
* Granular permissions.
* Identity federation (including Active Directory, Facebook, LinkedIn).
* Multifactor Authentication.
* Provide temporary access to users/devices and services where necessary.
* Allows you to set your own password rotation policy.
* Integrates with many different AWS services.
* Supports PCI DSS Compliance.

Critical Terms

* Users
* Groups
* Roles
* Policies
* Not region specific

Root account: email that you used to set the AWS account. Complete administrator access. Activate multifactor authentication for the root account.

<https://osantamaria.signin.aws.amazon.com/console>

By default when a user is created, they do not have permissions to do anything.

Can I change the IAM role on a running EC2 instance?

* No, at this time you cannot change the IAM role on a running EC2 instance. You can change the permissions on the IAM role associated with a running instance and the updated permissions will take effect almost immediately.

Can I associate an IAM role with an already running EC2 instance?

* No, you can associate only one IAM role with an EC2 instance.

***Active Directory Federation***

1. The flow is initiated when a user (let's call him Bob) browses to the ADFS (Active Directory Federation Services) sample site

(https://Fully.Qualified.Domain.Name.Here/adfs/ls/IdpInitiatedSignOn.aspx) inside his domain. When you install ADFS, you get a new virtual directory named adfs for your default website, which includes this page.

1. The sign-on page authenticates Bob against AD. Depending on the browser Bob is using, he might be prompted for his AD user name and password.
2. Bob's browser receives a SAML (Security Assertive Markup Language) assertion in the form of an authentication response from ADFS.
3. Bob's browser posts the SAML assertion to the AWS sign-in endpoint for SAML (https://signin.aws.amazon.com/saml). Behind the scenes, sign-in uses the AssumeRoleWithSAML API to request a temporary security credentials and then constructs a sign-in URL for the AWS Management Console.
4. Bob's browser receives the sign-in URL and it is redirected to the console.

From Bob's perspective, the process happens transparently; He starts at an internal website and ends up at the AWS Management Console, without ever having to supply any AWS credentials.

Can you authenticate with Active Directory? Yes using SAML

Do you get security credentials first? No, Authenticate against Active Directory first and then you get credentials.

***Web Identity Federation with Mobile Applications***

1. Authenticate first with your identity provider
2. Get temporary security credentials
3. Call AssumeRoleWithWebIdentity
4. Access your AWS resource (arn: Amazon Resource Name)

***EC2***

1. Create an EC2 instance
2. The AWS-CLI using credentials
3. The AWS-CLI using roles
4. Installing PHP & Composer
5. Using the PHP SDK to access S3

EC2 Instance Meta-data

* <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html>
* Remember this for the exam:

curl <http://169.254.169.254/latest/meta-data/>

* get public ip address:

curl <http://169.254.169.254/latest/meta-data/public-ipv4>

Elastic Load Balancers - Exam Tips

* Multiple SSL Certificates can be terminated on an ELB
* ELB's are not free you are charged by the hour and on a per GB basis of usage
* Services which are free include CloudFormation, Elastic Beanstalk and Autoescaling. The resources that they provision are not free.
* Ports include: SMTP 25, HTTP 80, HTTPS 443 and Custom ports 1024-65535
* HTTP Codes

1. 200 The request has succeeded
2. 3xx Redirection
3. 4xx Client side error
4. 5xx Server side error

SDK Exam tips

Available SDK's:

* Android, IOS, JavaScript(Browser)
* Java
* .Net
* Node.js
* PHP
* Python
* Ruby

Default region: US-EAST-1

* Some have default regions (Java)
* 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

***S3***

S3 provides developers and IT Teams with secure, durable, high-scalable object storage. Amazon S3 is easy to use, with a simple web services interface to store and retrieve any amount of data from anywhere on the web.

* Object based i.e allows you to upload files
* Files can be from 1 byte to 5TB
* There is unlimited storage
* Files are stored in buckets
* Buckets have a unique namespace for each region

eg <https://s3-eu-west-1.amazonaws.com/ACLOUDGURU/>

* Amazon guarantees 99.99% availability for the S3 platform.
* Amazon guarantees 99.999999999% durability for S3 information
* Can have metadata (key value pairs) on each storage
* Lifecycle management
* Versioning
* Encryption

Storage Types

* Standard S3, 99.99% availability, 99.999999999% durability.
* Reduce Redundancy Storage (RRS) - Still has 99.99% availability but only 99.99% durability over a given year.
* Only use RRS for replaceable data. For example if you have 10,0000 files you could expect to lose 100 files over a year as opposed to 0.00001 file with standard

Glacier

* Extremely low cost storage service for data archival. Amazon Glacier stores data for as little as $0.01 per gigabyte per month, and is optimized for data that is infrequently accessed and for which retrieval times of 3 to 5 hours are suitable.

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.
* You pay for each version of the file that you are storing.
* The only way to disable it is deleting the bucket and all versions within the bucket.
* Versioning allows you to retrieve, preserve and restore every version of every object stored in the bucket.
* To restore a file you need to delete the "delete marker"
* To delete a file permanently you have to delete the "delete marker" and all versions of the file

S3 Lifecycle Management

* Can be used in conjunction with versioning
* Can be applied to current versions and previous versions
* Following actions are allowed in conjunction with or without versioning;
* Archive only
* Permanent delete only
* Archive and then permanently delete

S3 Encryption

* You can upload/download your data to S3 via SSL Encrypted End Points and S3 can automatically encrypt your data at rest.
* S3 gives you the choice of managing your keys through AWS Key Management Service (KMS), having Amazon S3 manage them for you, or providing your own keys.

S3 Encryption Exam Tips

* Advanced Encryption Standard (AES) 256 Encryption
* Encryption can be either Server Side or Client Side

Server side

* + - Amazon S3 Managed Keys
    - AWS KMS-Managed Keys
    - Customer Provided Keys

Client Side

S3 Security

* All buckets are private by default
* Allows access control lists (an individual user, can only have access to 1 buckets and only have read only access)
* Integrates with IAM (using roles for example allows users to have access S3 buckets by roles)
* All endpoints are encrypted by SSL (Secure Sockets Layer)

S3 Functionality

* Static Websites can be hosted on S3. No need for webservers, you can just upload a static .html to an s3 bucket and take advantage of AWS S3's durability and High Availability
* Integrates with Cloud Front CDN
* Multipart uploads, allows you to upload parts of a file concurrently.
* Suggested for files a 100MB over. It is required for any file over 5GBs
* Allows us to resume a stopped file upload
* Eventual consistency. All AZ's (Availability Zones) will eventually be consistent. Put/Write/Delete requests will eventually be consistent across AZ's.

S3 Use Cases

* File shares for networks
* Backup/Archiving
* Origin for CloudFront CDN's (Content Distribution Networks)
* Hosting static files
* Hosting static websites

S3 Static websites

<http://osv-website.s3-website-us-west-2.amazonaws.com/>

* osv-website. bucket name
* s3-website- public bucket vs static website (key is s3-website-)
* us-west-2 region
* .amazonaws.com

S3 CORS (Cross Origin Resource Sharing) If you need to enable an EC2 instance or a S3 website to cross referencing javascript or html5 to another s3 bucket.

* Get the endpoint of the origin bucket
* Go to the destination bucket. Click on permissions then add CORS configuration

Lifecycle Management and Glacier

* Lifecycle management policies can be enabled on buckets with or without versioning

CDN Content Delivery Network

* It is a system of distributed servers (network) that deliver webpages and other web content to a user based on the geographic locations of the user, the origin of the webpage and a content deliver server.

CloudFront

* It is the Amazon CDN. It can be used to deliver your entire website, including dynamic, static, streaming, and interactive content using a global network of edge locations.
* Requests for your content are automatically routed to the nearest edge location, so content is delivered with the best possible performance.
* Is optimized to work with other Amazon Web services like S3, EC2, ELB and Route53. It also works seamlessly with any non-AWS origin server, which stores the original, definitive versions of your files.

Terminology

* Origin: This is the origin of all the files that the CDN will distribute. This can be either an S3 bucket, an EC2 instance, an Elastic Load Balancer or Route53
* Distribution: This is the name given the CDN which consists of a collection of Edge Locations. You can have one distribution with multiple origins.

Distribution Types

* Web Distribution - Typically used for websites
* RTMP - Used for Media Streaming

Exam Tips

* S3 Encryption is AES 256
* Minimum file size 1 byte, maximum 5TB
* No limits on how much can be stored in S3
* 100 buckets per region
* Website format

https://s3-eu-west-1.amazonaws.com/acloudguru-website/index.html ---> this one just point to an index.html file

https://acloudguru-website.s3-website-eu-west-1.amazonaws.com ---> this one is a website

* HTTP codes
* 200: Successful
* 300: Redirection
* 400: Client side errors
* 500: Server side errors
* X.509 certificates to secure data
* Eventual consistency

Quiz

*1. The minimum file size allowed on S3 is 1 byte?*

True

*2. If you encrypt a bucket on S3 what encryption does AWS use?*

AES 256

*3. You create a static hosting website in a bucket called "acloudguru" in Japan using S3. What would the new URL End Point be?*

http://acloudguru.s3-website-ap-northeast-1.amazonaws.com (Check the region)

*4. You are hosting a static website in an S3 bucket which uses Java script to reference assets in another S3 bucket. For some reason however these assets are not displaying when users browse to the site. What could be the problem?*

You haven't enabled Cross Origin Resource Sharing (CORS) on the bucket where the assets are stored.

*5. What is the HTTP code you would see if once you successfully place a file in an S3 bucket?*

200

*6. S3 provides unlimited storage.*

True

*7. What is the maximum file size that can be stored on S3?*

5TB

*8. If you want to enable a user to download your private data directly from S3, you can insert a pre-signed URL into a web page before giving it to your user.*

True

*9. When you first create an S3 bucket, this bucket is publicly accessible by default.*

False

*10. What is the largest size file you can transfer to S3 using a PUT operation?*

5Gb

***Databases Overview & Concepts***

Introduction

* Relational (OLTP - Online Transaction Processing)

RDS: MySql, SQL, Postgres, Oracle, Aurora

* Non Relational Databases (NoSQL)

DynamoDB

* Data Warehousing Databases (OLAP - Online Analytics Programming)

RedShift

ElastiCache

Web service that makes it easy to deploy, operate and scale an in memory cache in the cloud.

The service improves the performance of web applications by allowing you to retrieve information from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases. ElastiCache supports two open-source in-memory caching engines:

* Memcached
* Redis

A very common use is to use ElastiCache is in sessions variables

NoSQL - Relational

* Collection = Table
* Document = Row
* Key Value Pair = Fields

DynamoDB vs MongoDB

* DynamoDB does not allow you to have embedded data structures, MongoDB does.

Data Warehousing

* Used for business intelligence. Tools like Cognos, Jaspersoft, SQL Server Reporting Services, Oracle Hyperion, SAP NetWeaver. Data Warehousing databases use different type of architecture both from a database perspective and infrastructure layer.

OLTP vs OLAP

* OLAP differs from OLAP in terms of the types of queries run.
* OLTP is for small amount of data.
* OLAP is more used for management, is not used frequently run at the end of every day, every month, every quarter. Large amount of data.

Summary

* RDS is for OLTP
* DynamoDB is a NoSQL
* RedShift is for OLAP
* ElastiCache is for in memory caching

***Introduction to DynamoDB***

DynamoDB: Is a fast and flexible NoSQL database service for all applications that need consistent, single digit millisecond latency at any scale. It is a fully managed database and supports both document and key value data models. Its flexible data model and reliable performance make it a great fit for mobile, web, gaming, ad-tech, loT, and many other apps.

Configuration

* Stored on SSD Storage (Solid state drive)
* Spread across 3 geographical distinct data centres.
* Eventual Consistent Reads (default): Consistency across all copies of data is usually reached within a second. Repeating a read after a short time should return the updated data. (Best read performance)
* Strongly Consistent Reads: A strongly consistent read returns a result that reflects all writes that received a successful response prior to the read.

Pricing

* Provisioned Throughput capacity
* Throughput: (is the rate of production or the rate at which something can be processed)
* Write throughput $0.0065 per hour for every 10 units
* Read throughput $0.0065 per hour for every 50 units
* Storage costs of $0.25Gb per month

Example: Let's assume that your application needs to perform 1 million writes and 1 million reads per day while storing 3gb of data. First you need to calculate how many writes and reads per second you need. 1 million evenly spread writes per day is equivalent to 1,000,000 (writes) / 24 (hours) / 60 (minutes) / 60 (seconds) = 11.6 writes per second.

A DynamoDB Write Capacity Unit can handle 1 write per second, so you need 12 Write capacity units. Similary, to handle 1 million strongly consistent reads per day you need 12 Read Capacity Units.

Using on demand pricing in the US East Region, 12 Write Capacity Units would cost $0.1872 per day and 12 Read Capacity

Units would cost $0.0374 per day. So your total cost of provisioned throughput capacity is $0.1872 + $0.0374 = $0.2246 per day. Storage costs $0.25Gb per month

Assuming a 30 day month your 3GB would cost you 3 x $0.25 / 30 = $0.025 per day. Combining these numbers, the total cost of your DynamoDB table would be $0.2246 (for provisioned throughput capacity) + $0.025 (for storage) = $0.2496 per day or $7.50 per month.

Primary Keys

* Hash: Unordered index on Hash Primary Key
* Hash and range: Unordered index on Hash Primary Key and sorted index on range primary key

Secondary Index Types

* Local secondary index

Has the same hash key, different range key

Can only be created when creating a table. They cannot be removed or modified later.

* Global secondary index

Has different hash key and different range key

Can be created at table creation or added later

\* You can have 5 local secondary indexes per table and 5 global secondary indexes per table

\* 64k of data per attribute

Query vs Scan

Query: A query operation finds items in a table using only primary key attribute values.

* You must provide a hash key attribute name and a distinct value to search for.
* You can optionally provide a range key attribute name and value and use a comparison operator to refine the search results
* By default a query returns all of the data attributes for items with the specified primary keys; however, you can use the ProyectionExpression parameter so that the query only returns some of the attributes, rather than all of them.
* Query results are always sorted by the range key. If the data type of the range key is Number, the results are returned in numeric order; otherwise, the results are returned in order of ASCII character code values. By default, the sort order is ascending. To reverse the order, set the ScanIndexFordward parameter to false.
* By default is eventually consistent but can be changed to be strongly consistent.

Scan: Examines every item in the table. By default a scan returns all of the data attributes for every item; however, you can use the ProyectionExpression parameter so that the Scan only returns some of the attributes, rather than all of them.

* Generally a query operation is more efficient than a Scan operation.
* A Scan operation always scans the entire table, then filters out values to provide the desired result essentially adding the extra step of removing data from the result set. Avoiding using a Scan operation on a large table with a filter that removes many results, if possible. Also, as a table grows, the Scan operation slows. The scan operation examines every item for the requested values, and can use up the provisioned throughput for a large table in a single operation. For a quicker response times, design your tables in a way that can use the Query, Get, or BatchGetItem APIs, instead. Alternatively, design your application to use Scan operations in a way that minimizes the impact on your table's request rate.

Provisioned Throughput

Unit of Read provisioned throughput

* All reads are 4KB
* Eventually Consistent Reads (default) consist of 2 reads per second.
* Strongly Consistent Reads consist of 1 read per second.

Unit of Write Provisioned throughput

* All writes are 1 kb
* All writes consist of 1 write per second

Examples

You have an application that requires to read 10 items of 1kb per second using eventual consistency. What should you set the read throughput?

- First we calculate how many read units we need

- 1kb / 4kb (divide by 4 because all reads are 4KB) = 0.25

- 0.25 Rounded up to nearest whole number = 1 read unit per item

- 1 X 10 read items = 10

- Using eventual consistency 10 / 2 (divide by 2 since Eventually Consistent Reads (default) consist of 2 reads per second) = 5

- Answer is 5 units of read throughput

Formula:

- (Size of read / 4kb) Rounded up x no of items = read throughput

- Divide by 2 if eventually consistent

You have an application that requires to read 10 items of 6kb per second using eventual consistency. What should you set the read throughput?

Read units = 6kb / 4kb = 1.5

Rounded up: 2

Rounded up x number of items: 2 x 10 = 20

Using eventual consistency: 20 / 2 = 10

Answer is 10 units of read throughput

You have an application that requires to read 5 items of 10kb per second using eventual consistency. What should you set the read throughput?

Read units: 10kb / 4kb = 2.5

Rounded up: 3

Rounded up x number of items: 3 x 5 = 15

Using eventual consistency: 15 / 2 = 7.5

Answer is 8 (rounded up) units of read throughput

You have an application that requires to read 5 items of 10kb per second using strongly consistent. What should you set the read throughput?

Read units: 10kb / 4kb = 2.5

Rounded up: 3

Rounded up x number of items: 3 x 5 = 15

Using eventual consistency: 15 (no need to divide by 2 since Strongly Consistent Reads consist of 1 read per second)

Answer is 15 units of read throughput

You have an application that requires to write 5 items of 10kb per second. What should you set the write throughput?

Each write consist of 1kb of data, you need to write 5 items per second using 10kb of data per write.

5 x 10 = 50

You need a write throughput of 50 units.

You have an application that requires to write 5 items of 10.5kb per second. What should you set the write throughput?

Each write consist of 1kb of data, you need to write 5 items per second using 10.5kb of data per write.

Rounded up: 10.5 = 11

5 x 11 = 55

You need a write throughput of 55 units.

API Error code

What happens when you exceed your write or read throughput

400 HTTP Status Code: ProvisionedThroughputExceededException

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

Web Identity Providers

You can authenticate users using web identity providers (such as Facebook, Google, Amazon or any other Open-ID Connect-compatible Identity Provider)

This is done using AssumeRoleWithWebIdentity API

* User authenticates with id provider (such as facebook)
* They are passed a token by their id provider
* Your code calls AssumeRoleWithWebIdentity API and provides the providers token and specifies the ARN for the IAM role
* APP can now access Dynamodb from between 15 minutes to 1 hour (default is one hour)

Conditional writes

* Conditional writes are idempotent. This means that you can send the same conditional write request multiple times, but it will have no further effect on the item after the first time DynamoDB performs the specified update.
* Example: 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 dont know whether your request was successful or not. Because a conditional update is an 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

* DynamoDB supports atomic counters, where you use the UpdateItem operation to increment or decrement the value of an existing attribute without interfering with other write requests. (All write requests are applied in the order in which they were received)
* Example: A web application might want to maintain a counter per visitor to their site. In this case, the application would need to increment this counter regardless of its current value.
* Atomic counter updates are not idempotent. This means that the counter will increment each time you call UpdateItem. If you suspect that a previous request was unsuccessful, your application could retry the UpdateItem operation; however, this would risk updating the counter twice. This might be acceptable for a web site counter, because you can tolerate with sightly over - or under - counting the visitors. However, in a banking application, it would be safer to use a conditional update rather than an atomic counter.

Batch Operations

* If your application needs to read multiple items, you can use the BatchGetItem API. A single BatchGetItem request can retrieve up to 1 MB of data, which can contain as many as 100 items. In addition, a single BatchGetItem request can retrieve items for multiple tables.

Export data

* You can export data from DynamoDB from the management console.

Summary

Configuration

* Stored on SSD Storage
* Spread across 3 geographical distinct data centres

Eventual Consistent Reads (default)

* Consistency across all copies of data is usually reached within a second.
* Repeating a read after a short time should return the updated data. (Best read performance)

Strongly Consistent Reads

* A strongly consistent read returns a result that reflects all writes that received a successful response prior to the read.

Query

* Finds items in a table using only primary key attribute values.
* You must provide a hash key attribute name and a distinct value to search for.
* You can optionally provide a range key attribute name and value, and use a comparison operator to refine the search results.
* By default a query returns all of the data attributes for items with the specified primary key(s) however, you can use the
* ProjectionExpression parameter so that the query only returns some of the attributes, rather than all of them.
* Query results are always sorted by the range key. If the data type of the key range key is Number, the results are returned in numeric order; otherwise, the results are returned in order of ASCII character code values. By default, the sort order is ascending. To reverse the order, set the ScanIndexFordward parameter to false.
* By default is eventually consistent but can be changed to be strongly consistent.

Scan

* Examines every item in the table. By default, a scan returns all of the data attributes for every item; however, you can use the ProjectionExpression parameter so that the Scan only returns some of the attributes, rather than all of them.

Provisioned Throughput

Unit of read provisioned throughput

* All reads are 4kb
* Eventually Consistent Reads (default) consist of 2 reads per second.
* Strongly Consistent Reads consist of 1 read per second.

Unit of Write provisioned throughput

* All writes are 1KB
* All writes consist of 1 write per second.

Conditional Writes

Atomic Counters

Batch Operations

Steps for Web Identity Providers

Quiz

*You have a motion sensor which writes 600 items of data every minute. Each item consists of 5kb. Your application uses eventually consistent reads.*

*What should you set the read throughput to?*

600 / 60 = 10 items per second. 5kb rounded to nearest 4kb chunk is 8kb. 8 / 4 = 2.

This means you need 2 reads per item. 2 x 10 = 20 reads per second. As the reads are EVENTUALLY consistent, 20 / 2 = 10.

*You have a motion sensor which writes 600 items of data every minute. Each item consists of 5kb. What should you set the write throughput to?*

600 / 60 = 10 items per second. You are using 5kb per write and writes are measured in 1kb per write.

This means that you need 5 writes per item, 5 x 10 = 50 writes per second.

*What is the API call to retrieve multiple items from a DynamoDB table?*

BatchGetItem

*You have a motion sensor which writes 600 items of data every minute. Each item consists of 5kb.*

*Your application uses strongly consistent reads. What should you set the read throughput to?*

600 / 60 = 10 items per second. 5kb rounded to nearest 4kb chunk is 8kb. 8 / 4 = 2. This means you need 2 reads per item. 2 x 10 = 20 reads per second.

As the reads are Strongly consistent, you do not need to divide by 2. So the answer is 20.

*Using the AWS portal, you are trying to Scale DynamoDB past its preconfigured maximums. Which service can you increase by raising a ticket to AWS support?*

Provisioned throughput limits

*You have an application that needs to read 25 items of 13kb in size per second. Your application uses eventually consistent reads.*

*What should you set the read throughput to?*

Your application reads 13kb per item. Round up to the nearest 4kb chunk = 16kb. 16kb / 4kb = 4 reads per item.

You have 25 items so 25 x 4 = 100. Your application requires eventually consistent reads which means you divide by 2. 100/2 = 50.

*You have an application that needs to read 25 items of 13kb in size per second. Your application uses strongly consistent reads.*

*What should you set the read throughput to?*

Your application reads 13kb per item. Round up to the nearest 4kb chunk = 16kb. 16kb / 4kb = 4 reads per item. You have 25 items so 25 x 4 = 100.

Your application requires strongly consistent reads which means you do not need to divide by 2. Answer = 100

***SQS Simple Queue Service***

Is a web service that gives you access to a message queue that can be used to store messages while waiting for a computer to process them.

Amazon SQS is a distributed queue system that enables web service applications to quickly and reliably queue messages that one component in the application generates to be consumed by another component. A queue is a temporary repository for messages that are awaiting processing.

Using Amazon SQS you can decouple the components of an application so they run independently, with SQS easing message management between components.

Any component of a distributed application can store messages in a fail-safe queue. Messages can contain up to 256 kb of text in any format. Any component can later retrieve the messages programmatically using the Amazon SQS API.

The queue acts as a buffer between the component producing and saving data for processing. This means the 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 the network.

SQS ensures delivery of each message at least once, and supports multiple readers and writers interacting with the same queue. A single queue can be used simultaneously by many distributed application components, with no need for those components to coordinate with each other to share the queue.

SQS is engineered to always be available and deliver messages. One of the resulting tradeoffs is that SQS does not guarantee first in, first out delivery of messages. For many distributed applications, each message can stand on its own, and as long as all messages are delivered the order is not important.

If your system requires that order to be preserved you can place sequencing information in each message, so that you can reorder the messages when the queue returns them.

To illustrate, suppose you have a number of image files to encode. In an Amazon SQS worker queue, you create an Amazon SQS message for each file specifying the command (jpeg-encode) and the location of the file in S3. A pool of EC2 instances running the needed image processing software does the following:

1. Asynchronously pulls the task messages from the queue.

2. Retrieves the named file

3. Processes the conversion

4. Writes the image back to S3.

5. Writes a "task complete" message to another queue.

6. Deletes the original task message.

Visibility timeout period only starts when the application server has picked up the message.

- Does not offer FIFO

- 12 hours visibility time out.

- SQS is engineered to provide at least once delivery of all messages in its queues.

Although most of the time each message will be delivered to your application

exactly once, you should desing your system so that processing a message

more than once does not create any erros or inconsistencies.

- 256kb message size now available.

- Billed at 64kb chunks

Pricing

* First 1 million SQS Requests per month are free.
* $0.50 per 1 million SQS Requests per month thereafter ($0.00000050 per SQS Request)
* A single request can have from 1 to 10 messages up to a maximum total payload of 256kb
* Each 64kb chunk of payload is billed as 1 request. For examplea single API call with a 256kb payload will be billed as four requests.

\*decouple think about SQS

Exam tips

SQS Messages can be delivered multiple times and in any order.

Default visibility timeout

Default is 30 seconds

Max is 12 hours

When you receive a message from a queue and begin processing it, you may find the

visibility timeout for the queue is insufficient to fully process and delete that

message. To give yourself more time to process the message you can extend its visibility

timeout by using the ChangeMessageVisibility action to specify a new timeout value.

Amazon SQS restarts the timeout period using the new value.

Long Polling

Is a way to retrieve messages from your SQS queues. While the traditional SQS short polling returns immediately, even if the queue being polled is empty, SQS long polling does not return a response until a message arrives in the queue, or the long poll times out. SQS long polling makes it easy and inexpensive to retrieve messages from your SQS queue as soon as they are available.

Max long poll time out = 20 seconds.

Example:

Polling in a tight loops is burning CPU cycles and costing the company money.

How would you fix this? Enabling Long Polling

Fanning out

Create an SNS topic first using SNS. Then create and subscribe multiple SQS queues to the SNS topic. Now whenever a message is sent to the SNS topic, the message will be fanned out to the SQS queues.

SNS will deliver the message to all the SQS queues that are subscribed to the topic.

Question:

*You are designing a new application which involves processing payments and delivering promotional emails to customers. You plan to use SQS to help facilitate this. You need to ensure that the payment process takes priority over the creation and delivery of emails. What is the best way to achieve this?*

Use 2 SQS queues for the platform. Have the EC2 fleet poll the payment SQS queue first. If this queue is empty, then poll the promotional emails queue.

***SNS Simple Notification Service***

Is a web service that makes it easy to setup, operate and send notifications from the cloud. It provides developers with a highly scalable, flexible and cost effective capability to publish messages from an application and immediately deliver them to subscribers or other applications.

SNS follows the publish - subscribe (pub-sub) messaging paradigm, with notifications being delivered to clients using a push mechanism that eliminates the need to periodically check or poll for new information and updates. With simple APIs requiring minimal up front development effort, no maintenance or management overhead and pay as you go pricing, Amazon SNS gives developers an easy mechanism to incorporate a powerful notification system with their applications.

It is opposed to SQS because SNS "pushes" and SQS "pulls".

Besides pushing cloud notifications directly to mobile devices, Amazon SNS can also deliver notifications by SMS text messages or email, to Amazon Simple Queue Service (SQS) queues or to any HTTP endpoint.

To prevent messages from being lost all messages published to Amazon SNS are stored redundantly across multiple availability zones.

Data type is JSON.

SNS Topics

SNS allows you to group multiple recipients using topics. A topic is an access point for allowing recipients to dynamically subscribe for identical copies of the same notification. One topic can support deliveries to multiple endpoint types -- for example, you can group together iOS, Android and SMS recipients. When you publish once to a topic, SNS delivers appropriately formatted copies of your message to each subscriber.

Benefits

* Instantaneous, push - based delivery (no polling)
* Simple API and easy integration with applications
* Flexible message delivery over multiple transport protocols
* Inexpensive, pay as you go model with no upfront costs
* Web based AWS Management Console offers the simplicity of a point and click interface

SNS vs SQS

* Both messaging services in AWS
* SNS - Push
* SQS - Polls (Pulls)

SNS Pricing

* Users pays $0.50 per 1 million Amazon SNS requests.
* $0.06 per 100,000 notifications delivered over HTTP.
* $0.75 per 100 notification deliveries over SMS.
* $2.00 per 100,000 notification deliveries over Email.

Summary

* Instantaneous, push - based delivery (no polling)
* Protocols include

HTTP

HTTPS

Email

Email - JSON

Amazon SQS

Application

* Messages can be customized for each protocol

***SWF Simple Workflow Service***

It's a web service that makes it easy to coordinate work across distributed application components. Amazon 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. Tasks represent invocations of various processing steps in an application which can be performed by executable code, web service calls, human actions, and scripts.

SWF Workers

* Workers are programs that interact with Amazon SWF to get tasks, process received tasks and returns the results.

SWF Decider

* The decider is a program that controls the coordination of tasks ie their ordering, concurrency and scheduling according to the application logic.

The workers and the decider can run on cloud infrastructure, such as Amazon EC2, or on machines behind firewalls. Amazon SWF brokers the interactions between workers and the decider. It allows the decider to get consistent views into the progress of tasks and to initiate new tasks in an ongoing manner.

At the same time, Amazon SWF stores tasks, assigns them to workers when they are ready, and monitors their progress. It ensures that a task is assigned only once and is never duplicated. Since Amazon SWF maintains the application's state durably, workers and deciders do not have to keep track of execution state. They can run independenly and scale quickly.

Difference between SQS ans SWF

* With SWF a task is assigned only once and it is never duplicated. With SQS you got your message visibility timeout a task can be assigned multiple times.

SWF Domains

* Your workflow and activity types and the workflow execution it self are all scoped to a domain. Domains isolate a set of types, executions and tasks list from others within the same account. You can register a domain by using the AWS Management Console or by using the RegisterDomain action in the Amazon SWF API.
* The parameters are specified in JavaScript Object Notation (JSON format).

https://swf.us-east-1.amazonaws.com

RegisterDomain

{

"name":"867530901",

"description":"music",

"workflowExecutionRetentionPeriodInDays":"60"

}

* How long for a workflow?

Maximum Workflow can be 1 year and the value is always measured in seconds.

SWF vs SQS

* SWF presents a task oriented API, whereas SQS offers a message-oriented API.
* SWF ensures that a task is assigned only once and is never duplicated. With SQS, you need to handle duplicated messages and may also need to ensure that a message is processed only once.
* SWF keeps track of all the tasks and events in an application. With SQS you need to implement your own application level tracking especially if your application uses multiple queues.

***Using Cloud Formation***

* Allows you to create scripts to deploy infrastructure.
* Cloud Formation is free but you pay for the resources that it creates.
* GetAtt: The value that you use to return a result.
* The default scripting language for CloudFormation is: JSON
* Cloud Formation itself is free, however the resources it provisions will be charged at the usual rates: true
* What happens if Cloud Formation encounters an error by default? It will terminate and rollback all resources created on failure.

*You are creating a virtual data centre using cloud formation and you need to output the DNS name of your load balancer. What command would you use to achieve this?*

FN::GetAtt

*What language are cloud formation templates written in?*

JSON

***Using Elastic Beanstalk***

With Elastic Beanstalk, you can quickly deploy and manage applications in the AWS cloud without worrying about the infrastructure that runs those applications. AWS Elastic Beanstalk reduces management complexity without restricting choice or control. You simply upload your application, and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring. Elastic Beanstalk uses highly reliable and scalable services that are available in the AWS Free Usage Tier.

Preconfiguration

Preconfigured

* IIS
* Node.js
* PHP
* Python
* Ruby
* Tomcat

Preconfigured - Docker

* GlassFish
* Python

Generic

* Docker

Elastic Beanstalk is free but you pay for the resources it uses.

*Elastic Beanstalk is object based storage. False*

*What languages and development stacks is NOT supported by AWS Elastic Beanstalk? Jetty for Jbos applications*

*Unlike Cloud Formation, Elastic Beanstalk itself is not free free AND you must also pay for the resources it provisions. False*

***VPC Virtual Private Cloud***

What is a VPC? Think of a VPC as a virtual data centre in the cloud.

Amazon VPC lets you provision a logically isolated section of the Amazon Web Services (AWS) Cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.

You can easily customize the network configuration for your Amazon Virtual Private Cloud. For example, you can create a public-facing subnet for your webservers that has access to the internet and place your backend systems such a databases or application servers in a private facing subnet with no Internet access.

You can leverage multiple layers of security, including security groups and network access control lists to help control access to Amazon EC2 instances in each subnet.

Additionally, you can create a Hardware Virtual Private Network (VPN) connection between your corporate datacenter and your VPC and leverage the AWS Cloud as an extension of your corporate datacenter.

What can I do with a VPC ?

* Launch instances into a subnet of your choosing.
* Assign custom IP address ranges in each subnet.
* Configure route tables between subnets
* Create internet gateways and attach them to subnets (or not)
* Much better security control over your AWS resources.
* Instance security groups.
* Subnet network access control lists (ACLS)

Default VPC vs Custom VPC

* Default VPC is user friendly, allowing you to immediately deploy instances.
* All subnets in default VPC have an internet gateway attached.
* Each EC2 instance has both a public and private IP address.
* If you delete the default VPC the only way to get it back is to contact AWS.

VPC Peering

* Allows you to connect one VPC with another via a direct network route using private IP addresses.
* Instances behave as if they were on the same private network.
* You can peer VPC's with other AWS accounts as well as with other VPCs in the same account.
* Peering is a star configuration, ie 1 central VPC peers with 4 others.

VPC Restrictions

* 5 Elastic IP addresses
* 5 Internet Gateways
* 5 VPCs per region (can be increased upon request)
* 50 VPN connections per region
* 50 Customer gateways per region
* 200 Route tables per region
* 100 Security Groups per VPC
* 50 Rules per Security group
* How many internet gateways can I attach to my custom VPC? 1
* CIDR block: Classless Inter-Domain Routing
* Even if you have ec2 instances inside your public subnet it does not mean that they have Internet access by default. You need to attach a public ip address or an elastic load balancer.
* Each ec2 instance performs source/destination checks by default. That means the instance must be the source/destination of any traffic it sends or receives.
* The NAT instance must be able to send or receive traffic when the source/destination is not itself. You need to disable source/destination checks on the NAT instance.

VPC - ACLs

* They act like a firewall, allowing you to put down network rules across entire subnets rather than doing it through security groups. Overrules what the rules are in a security group.
* When you create your VPC is comes with a modifiable ACL that by default allows all inbound/outbound.
* Each subnet must be associated with a Network Access Control List (ACL). If you do not explicitly associate a subnet with an ACL it is going to be associated with the default network ACL.
* When you add rules they are evaluated the "lowest number first"
* Can you have multiple ACL for a subnet? No

***Shared Responsability Model***

* http://media.amazonwebservices.com/AWS\_Security\_Best\_Practices.pdf
* Shared Responsability Model for Infrastructure Services
* Shared Responsibility Model for Container Services
* Shared Responsibility Model for Abstracted Services

*You are required to patch OS and Applications in RDS?* False

*In the shared responsibility model, what is AWS's responsibility?*

Restricting access to the data centres, proper destruction of decommissioned disks, patching of firmware for the hardware on which your AWS resources reside.