Renato Gama

Software Architecture Blog

AWS Certified Developer Associate Study Notes

For content that overlaps with CSAA exam please refer to AWS Certified Solutions Architect Exam Study Notes.

Elastic Beanstalk

- Deploy, monitor and scale an application quickly.
- It provides developers with the ability to provision application infrastructure in an almost transparent way.
- Intended to abstract infrastructure.
- Applications can have multiple environments (dev, test, prod, etc) and different versions.
- Can place application behind load balancers and auto scaling groups.
- Application is uploaded as an application bundle or a zip file.

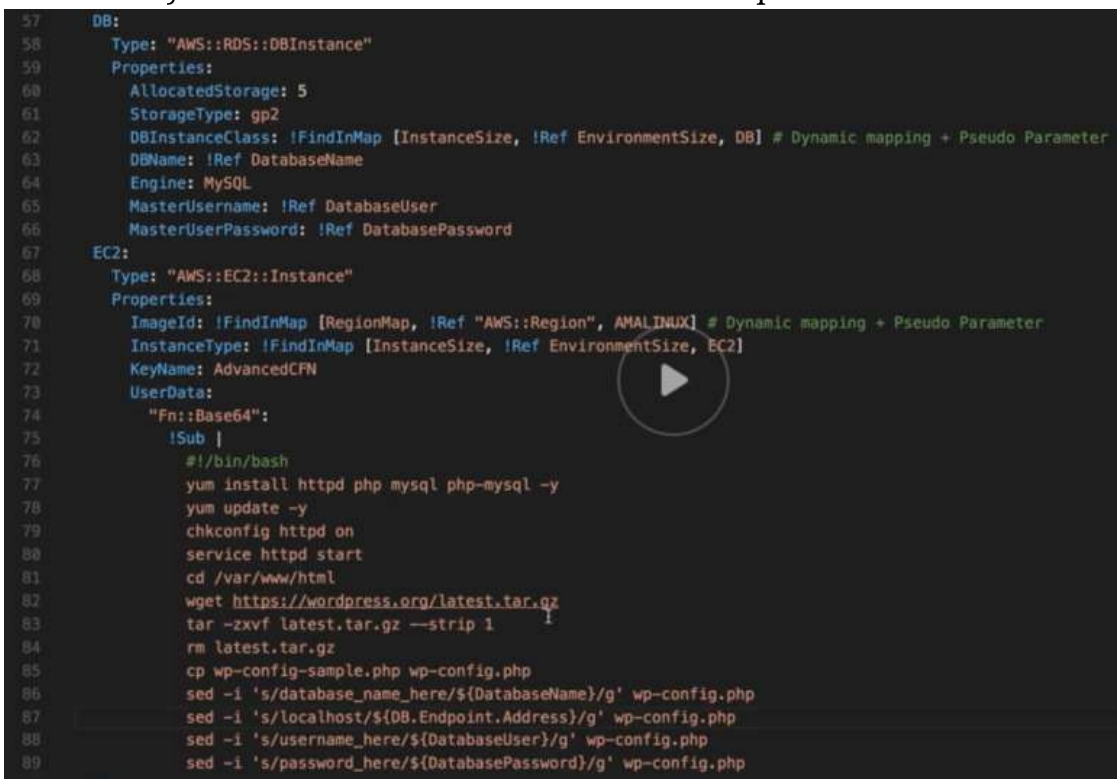
- Pre-configured environments for:
  - Node.js
  - Php
  - Python
  - Ruby
  - Tomcat
  - .NET (Windows/IIS)
  - Java
  - Go
  - Packer
- Elastic Beanstalk is free, AWS will charge for the underlying resources.
- If EB creates an RDS instance for you then it will delete the instance when you delete the application.
- You can SSH and RDP into instances created by Elastic Beanstalk.

CloudFormation

- JSON/YAML templates for infrastructure, called “Stack”.
- Enables you to version you infrastructure, just like you do with code.
- Takes care of provisioning all resources described in the file.
- Template elements
  - List of AWS resources and their associated configuration values (mandatory)
  - Templates file format and version number
  - Input values; template parameters (limit of 60)
  - Output values: the output values required once the stack has finished building, such as public IP address, ELB address, etc (limit of 60).
  - List of data tables: used to lookup static configuration values such as AMI ids, etc.
- Example:

- StackSet – A StackSet is a container for AWS CloudFormation stacks that lets you provision stacks across AWS accounts and regions by using a single AWS CloudFormation template.
- Automatic rollback on error feature is enabled by default

- You are charged even if the creation of your stack fails
- CloudFormation is free, you are charged by the underlying resources.
- Stacks can wait for applications to be provisioned using the “WaitCondition”.
- Route53 is completely supported. This is includes creating a new hosted zone or updating existing ones. Also you can create A records, ALIASES, etc.
- IAM role creation and assignment is also supported.
- You can SSH and RDP into instances created by CloudFormation.
- Invalid JSON syntax will cause an error message during template validation. Since the stack will never start creating, there is nothing to roll back.
- Dependency resolution between resources:
  - If you are creating a resource that depends on another, CloudFormation will have to work sequentially and wait for the needed resources. CloudFormation CAN detect the dependency chain when one resource is explicitly referenced by another. If there is no references explicitly defined then you can use the attribute “DependsOn” and pass the logical name of the resource wanted.
  - Example: because of the line 87, CloudFormation can detect that the DB resource is needed by EC2. It will then wait for DB creation to proceed to EC2.



```
57    DB:
58      Type: "AWS::RDS::DBInstance"
59      Properties:
60        AllocatedStorage: 5
61        StorageType: gp2
62        DBInstanceClass: !FindInMap [InstanceSize, !Ref EnvironmentSize, DB] # Dynamic mapping + Pseudo Parameter
63        DBName: !Ref DatabaseName
64        Engine: MySQL
65        MasterUsername: !Ref DatabaseUser
66        MasterUserPassword: !Ref DatabasePassword
67    EC2:
68      Type: "AWS::EC2::Instance"
69      Properties:
70        ImageId: !FindInMap [RegionMap, !Ref "AWS::Region", AMALINUX] # Dynamic mapping + Pseudo Parameter
71        InstanceType: !FindInMap [InstanceSize, !Ref EnvironmentSize, EC2]
72        KeyName: AdvancedCFN
73        UserData:
74          "Fn::Base64":
75            !Sub |
76              #!/bin/bash
77              yum install httpd php mysql php-mysql -y
78              yum update -y
79              chkconfig httpd on
80              service httpd start
81              cd /var/www/html
82              wget https://wordpress.org/latest.tar.gz
83              tar -zxvf latest.tar.gz --strip 1
84              rm latest.tar.gz
85              cp wp-config-sample.php wp-config.php
86              sed -i 's/database_name_here/${DatabaseName}/g' wp-config.php
87              sed -i 's/localhost/${DB.Endpoint.Address}/g' wp-config.php
88              sed -i 's/username_here/${DatabaseUser}/g' wp-config.php
89              sed -i 's/password_here/${DatabasePassword}/g' wp-config.php
```

- CloudFormation Init – Is a more powerful and feature rich way to provide configuration and perform instance bootstrapping from within CloudFormation.
  - Scales better/more portable than providing UserData attribute to EC2 resources;
    - Tries to be OS independent (yum wouldn’t work on Ubuntu for example)
    - Its a desired state engine (not procedural)
    - Supports multiple sets of configuration
    - Allows authentication (to perform downloads from GitHub for example)

    - It is idempotent (running multiple times provides the same results)
- CloudFormation can be used with **Chef** and **Puppet** to configure bootstrapped EC2 instances.
- Intrinsic Functions – Are functions that you can use within you template. Some common functions:
    - **fn::GetAttr** – The `GetAtt` intrinsic function returns the value of an attribute from a resource in the template.
    - **fn::Ref** – The intrinsic function `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. An AWS::EC2::Instance for example, returns its instanceId
    - **fn::Join** – The intrinsic function `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. the delimiter is passed as the first parameter, for example: `{ "Fn::Join" : [ "-", [ "value1", "value2" ] ] }` will produce the string `value1-value2`.
    - **fn::GetAZs** – Returns an array of availability zone strings
    - **fn::Select** – Receives an array and an index to return a single element
- Limits
    - Templates – There are no limits to the number of templates
    - Stacks – Limited to 200 stacks per account
- Creation Policies
    - CloudFormation reports the stack as CREATE_COMPLETE when all resources have been created and the appropriate services report creation completed. It doesn't have visibility on whats is running on EC2 instances – it doesn't see what is going on inside the instance. So resources might not be ready when CREATE_COMPLETE status is reached, configuration or automated tests may still be going on.
    - Creation Policies is a way to signal CloudFormation when the resources are ready for service.
    - ResourceSignal
        - Count – Number of signals needed for considering a SUCCESS (useful when creating multiple instances).
        - Timeout – Time during which CloudFormation will wait for the a FAILURE/SUCCESS signal. If not received within this time, CF assumes a FAILURE.
- Resource Deletion Policies
    - Defines what happens to a resource when it is deleted

- Three options: DELETE, RETAIN or SNAPSHOT
  - DELETE – Default policy
  - RETAIN – Supported by all resources. Simply leaves the resource as is after the deletion of the stack.
  - SNAPSHOT – Only supported in a few resources types; EC2 instances and RDS instances

SQS (Simple Queue Service)

- In Amazon SQS, you can use the API or the console to configure dead letter queues, which are queues that receive messages from other source queues.If you make a queue into a dead letter queue, it receives messages after a maximum number of processing attempts cannot be completed. You can use dead letter queues to isolate messages that can't be processed for later analysis.
- Supports TLS v1.0, v1.1 and v1.2
- When receiving messages, you can also set a special visibility timeout for the returned messages without changing the overall queue timeout.
- You can extend the visibility timeout even after receiving the message for processing.

SNS (Simple Notification Service)

- After a message has been published to a topic it can't be deleted (recalled).
- Message example

```
POST / HTTP/1.1
x-amz-sns-message-type: Notification
x-amz-sns-message-id: 22b80b92-fdea-4c2c-8f9d-bdfb0c7bf324
x-amz-sns-topic-arn: arn:aws:sns:us-west-2:123456789012:MyTopic
x-amz-sns-subscription-arn: arn:aws:sns:us-west-2:123456789012:MyTopic:c9135db0-26c4-47e
Content-Length: 773
Content-Type: text/plain; charset=UTF-8
Host: myhost.example.com
Connection: Keep-Alive
User-Agent: Amazon Simple Notification Service Agent

{
  "Type" : "Notification",
  "MessageId" : "22b80b92-fdea-4c2c-8f9d-bdfb0c7bf324",
  "TopicArn" : "arn:aws:sns:us-west-2:123456789012:MyTopic",
  "Subject" : "My First Message",
  "Message" : "Hello world!",
  "Timestamp" : "2012-05-02T00:54:06.655Z",
  "SignatureVersion" : "1",
  "Signature" : "EXAMPLEw6JRNwm1LFQL4ICB0bnXrdB8ClRMTQFGBqwLpGbM78tJ4etTwC5zU7O3tS6tGpey
  "SigningCertURL" : "https://sns.us-west-2.amazonaws.com/SimpleNotificationService-f3ec
  "UnsubscribeURL" : "https://sns.us-west-2.amazonaws.com/?Action=Unsubscribe&Subscripti
}
```

- URL atributes of an SNS message are;
  - SigningCertURL
  - SubscribeURL
  - UnsubscribeURL

- Maximum length of SNS Topic Name is 256 characters
- Maximum of
  - 100,000 topics per account
  - 10 million subscriptions per topic

Elastic Load Balacer

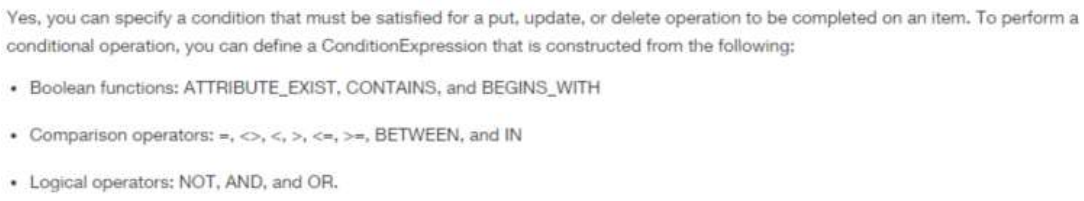
- Favor ElastiCache over sticky sessions
  - Sticky session causes non even distribution of traffic.
  - Auto-scaling cannot terminate web servers some user's session state.
  - ElastiCache is the best method for maintaining application session state.
- When routing requests to web servers when they are in private subnets, you will have to choose the public subnets when configuring the ELB. Internet-facing ELB should be created on public subnets.
- When you use ELB you are given a DNS host name. Route 53 is amazon's DNS service that handles DNS on the backend.
- When you are load testing an ELB you must make sure to re-resolve the DNS so that clients don't cache the instance IP and keep sending requests to the same instance.
  - Best practices for testing and evaluating ELB: ELB Best Practices Whitepaper

DynamoDB

- By default, you are allowed to have 256 tables per account per region.
- Control/Data Planes
  - Control Plane – Let you create and manage DynamoDB tables. Also let you work with indexes, streams, and other objects that are dependent on tables.
  - Data Plane – Let you perform create, read, update, and delete (also called CRUD) actions on data in a table.
- Unless Amazon explicitly mentions an **eventually** consistent read, they are referring to a strongly consistent read.
- Secondary indexes
  - Global secondary index — an index with a partition key and a sort key that can be different from those on the base table. A global secondary index is considered "global" because queries on the index can span all of the data in the base table, across all partitions. Limited to 5. Can be created any time after table creation.
  - Local secondary index — an index that has the same partition key as the base table, but a different sort key. A local secondary index is "local" in the sense that every partition of a local secondary index is scoped to a base table partition that

has the same partition key value. Limited to 5. Can only be defined during table creation.

- Read/Write Capacity
  - One read capacity unit represents one strongly consistent read per second, for an item up to 4KB in size. Eventually consistent read required only half unit to read 4KB. Read capacity units always round up. Eg.: For a strongly consistent read of 5Kb you will need 2 units
  - One write capacity unit represents 1KB write per second.
- Partitions
  - Size limitation – There is a size limitation of 10GB (considering item's size and any local secondary index size) that will cause DynamoDB to split your data into a new partition.
  - Performance limitation – Whenever you provision more than 1000 units of read/write capacity, DynamoDB will split your data as well.
  - Partitions are not shrunk after data has been split into multiple partitions. Neither by reducing table size nor by decreasing read/write capacity units.
  - Read/write capacity is evenly divided among partitions, so that you may end up exceeding the partitions capacity limit and thus facing "ProvisionedThroughputExceededException" even if you are not exceeding the table capacity.
- If you can predict your need for DynamoDB read and write throughput, Reserved Capacity offers significant savings over the normal price of DynamoDB.
  - Minimum amount of reserved capacity that can be purchases is 100 units.
- DynamoDB uses optimistic locking and conditional writes for consistency
  - Optimistic locking is a strategy to ensure that the client-side item that you are updating (or deleting) is the same as the item in DynamoDB. If you use this strategy, then your database writes are protected from being overwritten by the writes of others — and vice-versa.
  - With optimistic locking, each item has an attribute that acts as a version number. If you retrieve an item from a table, the application records the version number of that item. You can update the item, but only if the version number on the server side has not changed. If there is a version mismatch, it means that someone else has modified the item before you did; the update attempt fails, because you have a stale version of the item. If this happens, you simply try again by retrieving the item and then attempting to update it. Optimistic locking prevents you from accidentally overwriting changes that were made by others; it also prevents others from accidentally overwriting your changes.
- Conditional operations can be write, update or delete.


Yes, you can specify a condition that must be satisfied for a put, update, or delete operation to be completed on an item. To perform a conditional operation, you can define a ConditionExpression that is constructed from the following:

- Boolean functions: ATTRIBUTE_EXIST, CONTAINS, and BEGINS_WITH
- Comparison operators: =, <>, <, >, <=, >=, BETWEEN, and IN
- Logical operators: NOT, AND, and OR.

- The cumulative number of tables and indexes in the `CREATING`, `DELETING` or `UPDATING` state cannot exceed 10, otherwise you will get a **LimitExceededException.**
- You can only create one secondary index at a time (either local or global). Create the second table and its accompanying index after the first table has status = 'active."
- 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.
- The maximum limit of data that can be retrieved by a scan operation is 1MB. You can use LastEvaluatedKey api call to retrieve more items. Do not confuse this limit with the limit for BatchGetItem which is 16MB.
- ***Expressions*** are used as part of the query API call to filter results based o the values of primary keys.

S3

- Amazon recommends using multipart uploads for objects larger than 100MB.
- Benefits of multipart uploads
  - **Improved throughtput** – You can upload parts in parallel
  - **Quick recovery from network issues** – Smaller part sizes minimize the impact of network issues, once you won't have to start the upload from the first part.
  - **Pause and resume uploads**
  - **Begin an upload before you know the final object size** – You can upload an object as you are creating it.
- Static website hosting – You don't need to use Route53 if you don't care about the website URL.
  - Default url schema is: *http://domain.com.s3-website-sa-east-1.amazonaws.com* (domain + service + region + amazon AWS domain). Think of it as an ascending order, domain is smaller than service that is smaller than the region that is smaller than the full AWS.
- S3 bucket names may only contain only **lower case letters**, **periods**, **numbers**, and **dashes**. Bucket names **must not be formatted as an IP address**, and they may not begin with a period.
- Each account can have a maximum of 100 buckets by default

- Keys can use UTF-8 encoding but must not be longer than 1024 bytes
- An object may have up to 10 tags
  - Object tagging is a charged service
- S3 supports publishing the following events:
  - **ReducedRedundacyLostObject**
  - **ObjectCreated** – You can specify notifications for PUT, POST or both; ObjectCreated:Put, ObjectCreated:Post or ObjectCreated:*
  - **ObjectRemoved** – You can specify notifications for regular DELETEs or or for DeleteMarkersCreated
- Amazon S3 resources are not transferable. The resource owner can optionally grant access permissions to others by writing an access policy.
- Trying to delete a bucket that is not empty will result in a 409 Conclict error; "BucketNotEmpty".
- Server-side encryption
  - When using server side encryption you must provide the header "*x-amz-server-side-encryption*".
  - Amazon S3 supports bucket policies that you can use if you require server-side encryption for all objects that are stored in your bucket. For example, the following bucket policy denies upload object (s3:PutObject) permission to everyone if the request does not include the x-amz-server-side-encryption header requesting server-side encryption.

```
{
  "Version": "2012-10-17",
  "Id": "PutObjPolicy",
  "Statement": [
    {
      "Sid": "DenyIncorrectEncryptionHeader",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::YourBucket/*",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-server-side-encryption": "AES256"
        }
      }
    },
    {
      "Sid": "DenyUnEncryptedObjectUploads",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::YourBucket/*",
      "Condition": {
        "Null": {
          "s3:x-amz-server-side-encryption": "true"
        }
      }
    }
  ]
}
```

- **Server-side encryption encrypts only the object data. Any object metadata is not encrypted.**
- 

	Standard	Infrequent Access	Reduced Redundancy	Glacier
Durability	99.999999999%	99.999999999%	99.99%	99.999999999%
Availability	99.99%	99.9%	99.99%	N/A
Availability SLA	99.9%	99%	99.9%	N/A
Min Object Size	N/A	128KB	N/A	N/A
Min Storage Duration	N/A	30 days	N/A	90 days
Retrieval Fee	N/A	per GB	N/A	per GB
First Byte Latency	ms	ms	ms	Select mins or hours
Storage Class	Object Level	Object Level	Object Level	Object Level
Lifecycle Transitions	Yes	Yes	Yes	Yes
SSL Support	Yes	Yes	Yes	Yes

- You can attach metadata to objects. Metadata is in the form of key/value pairs. Custom metadata must start with `x-amx-meta-` and are retrieved as HTTP headers.
  - Metadata can also be used to store regular http headers associated with the object, for example, `Content-Disposition; inline;`

SES

- Amazon SES can reliably deliver merchandising, subscription, transactional, and notification email messages.
- Can send and receive email
- You can process received email in two forms
  - Receive an SNS notification and download email data from S3
  - Write a Lambda function
- You can send email via SMTP or API calls.
- After validating SES you must request limit upgrade to send production emails (outside the sandbox).
- Sends email messages up to 10MB in size
- Maximum of 50 recipients per email (TO + CC + BCC)
- Sending limits
  - Sending quota – maximum number of emails per 24 hours
  - Sending rate – maximum number of emails per second
  - ***Sending limits are bases on recipients rather than on messages***
- Supports
  - DKIM
  - SPF

SWF

- Guarantees the task is delivered only once.
- Domains are used to segregate application resources in SWF.
- There two types of tasks;
  - **Decision Tasks** – Computes and decides the flow of the tasks based on the event history for that execution. Can be implemented as a pooling program in EC2, or as a lambda function. The decider is invoked every time the state of the workflow changes; a signal is received, a worker task is done, etc. ***Do not*** support human interaction.
  - **Worker Tasks** – Performs the actual tasks. Also implemented as a pooling program. ***Can*** support human interaction.
- Long pooling timeout for tasks are 60 seconds, differently from SQS which is 20 seconds maximum.
- Maximums;
  - 100 domains
  - 10000 workflow/activity types (either registered or deprecated)
  - 25000 workflow history events
- Use cases;
  - Media processing
  - Business process automation
  - Data analytics
  - Migration to the cloud
  - Batch processing

VPC

- A route table contains a set of rules, called routes, that are used to determine where network traffic is directed.Each subnet in your VPC must be associated with a route table; the table controls the routing for the subnet. A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same route table.
  - 1 route table → N subnets
  - 1 subnet → 1 route table

EC2

- Maximum volume sizes

  - EBS Backed – 16TiB
  - Instance store – 10 Gib
- Differences between EBS backed and instance stored volumes includes:

Characteristic	Amazon EBS-Backed	Amazon Instance Store-Backed
Boot time	Usually less than 1 minute	Usually less than 5 minutes
Size limit	16 TiB	10 GiB
Root device volume	Amazon EBS volume	Instance store volume
Data persistence	By default, the root volume is deleted when the instance terminates.* Data on any other Amazon EBS volumes persists after instance termination by default. Data on any instance store volumes persists only during the life of the instance.	Data on any instance store volumes persists only during the life of the instance. Data on any Amazon EBS volumes persists after instance termination by default.
Upgrading	The instance type, kernel, RAM disk, and user data can be changed while the instance is stopped.	Instance attributes are fixed for the life of an instance.
Charges	You're charged for instance usage, Amazon EBS volume usage, and storing your AMI as an Amazon EBS snapshot.	You're charged for instance usage and storing your AMI in Amazon S3
AMI creation/bundling	Uses a single command/call	Requires installation and use of AMI tools
Stopped state	Can be placed in stopped state where instance is not running, but the root volume is persisted in Amazon EBS	Cannot be in stopped state; instances are running or terminated

-

SDKs

- AWS currently offers the following SDKs
  - Android
  - Browser
  - iOS
  - Java
  - .NET
  - Node.js
  - PHP
  - Python
  - Ruby
  - Go
  - C++
  - AWS Mobile SDK

APIs

- S3
  - **GetObject** – Get an object data by its key.
- EC2
  - **AttachVolume** – Used to attach an EBS volume to an EC2 instance (although it mounts a volume on the OS, the wording used by the API is "attach").
  - **DescribeImages –** Describes one or more of the images available to you (public images, your private images, others private image in which your account has explicit access).
  - **BundleInstance –** Used to create an AMI and stored data on volumes other than the root volume. *Not applicable for EBS backed instances.*
- SQS
  - **ChangeMessageVisibility** – Change visibility timeout of a message (not the whole queue)
- SNS
  - **ListTopics**
  - **ListSubscriptionsByTopic**
  - **AddPermission**
  - **RemovePermission**
- CloudFormation
  - **ListStackResources** – Lists resources created for a specific stack (Can list resources for deleted stacks for the next 90 days after deletion)
- CloudFront
  - **CreateDistributions**
  - **CreateInvalidation**
- DynamoDB
  - **GetItem** – Query an item by its primary key
  - **BatchGetItem** – Retrieves multiple items at once. Can retrieve a maximum of 100 items. The total size of all items retrieved cannot exceed 16MB
  - **ListTables** – A single ListTables call can return a maximum of 100 table names; if you have more than 100 tables, you can request that ListTables return paginated results, so that you can retrieve all of the table names.

API Tips

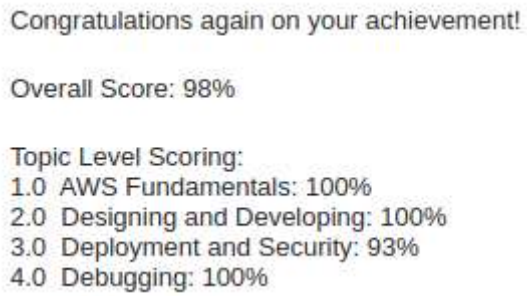
- APIs that use the "DESCRIBE" prefix
  - EC2 – ***DescribeInstances/DescribeImages***
- APIS that use "LIST" prefix
  - S3 – ***ListBuckets***
  - SQS – ***ListQueues***

- APIS that use both “LIST” and “DESCRIBE”
  - SWF – ***DescribeActivityType***/***DescribeDomain*** (describe in the sense of “get only one”, or “get by id”) and ***listActivityTypes***/***listDomains.***
  - CloudFormation – ***ListStacks*** (filter necessary), ***DescribeStacks*** (returns one or more stacks)
  - DynamoDB – **DescribeTable**/**ListTables**
  - *My perception is when an API uses both describe and list, describe is in the sense of “gimme details of this specific resource”, as opposed to the EC2 API, for example, which uses describe to list all instances.*

Exam History

- 01/03/2018 – Whizlabs – Practice Exam – 84%
- 03/03/2018 – ACloudGuru – Practice Exam – 78%
- 04/03/2018 – Whizlabs – Practice Exam – 85%
- 06/03/2018 – Whizlabs – Practice Exam – 88%
- 08/03/2018 – Whizlabs – Practice Exam – 97%
- 08/03/2018 – ACloudGuru – Practice Exam – 93%
- 10/03/2018 – Whizlabs – Practice Exam – 95%
- 12/03/2018 – PSI – Practice Exam – 100%
- 12/03/2018 – Whizlabs – Practice Exam – 88%
- 13/03/2018 – Whizlabs – Practice Exam – 94%
- 14/03/2018 – ACloudGuru – Practice Exam – 90%
- 14/03/2018 – PSI – Actual Exam – 98%

Exam Score Breakdown


Congratulations again on your achievement!

Overall Score: 98%

Topic Level Scoring:
1.0 AWS Fundamentals: 100%
2.0 Designing and Developing: 100%
3.0 Deployment and Security: 93%
4.0 Debugging: 100%

To pass this certification exam I studied for 17 hours and 50 minutes, from 20/02/2018 to 14/03/2018.