Sunday, September 22, 2019

AWS Certified Developer Exam Study Notes[1]

[1] Note: Contains cut-n-pasted descriptions from Amazon's website(s).

IAM - Identity and Access Management

- centralized control of your AWS account shared access
- Global
- users, groups, and roles
- apply roles to both users and AWS services
- apply role to EC2 instance at runtime
- modify EC2 role anytime
- identity federation - use corporate password to authenticate in AWS
- user’s access key id and secret access key used for accessing AWS via APIs and Command-line
- policies
    - written using JSON
    - attached to users
    - determines resources they may access
    - must have a Statement element in JSON document
    - user credential
        - two types
            - programmatic access - key
            - management console access - password
    - max 10 policies attached to role
    - max 10 policies attached to a user
    - policy document structure: effect, service, resource, action, condition
        - effect – ALLOW or DENY
        - resource – where to apply
        - action – the api to apply (or grouping of APIs etc)
    - three types of policies - managed, customer managed, inline
    - managed policy - IAM policy created and administered by AWS
    - inline policy - embedded policy within user, group, role it applies
    - customer policy - managed by user

- user
  - can be a member of up to 10 groups
  - assigned two access keys
  - can be assigned one MFA device
  - max 5000 per account
- group
  - way to group users and apply policies to them collectively
  - max 300 groups per account
- temporary security token
  - Use AWS Security Token Service (STS) - see that section
  - lifetime range: 15 min to 36 hours

AWS STS - AWS Security Token Service

- create and provide trusted users with temporary security credentials
- short term
- not stored, generated dynamically and provided to user when requested
- do not distribute nor embed security credentials in application
- do not have to define an identity, just give temporary credentials
- consist of:
  - access key ID
  - secret access key
  - security token
- Scenarios to use:
  - Identity Federation - manage user identities in an external system outside of AWS and grant users who sign in from those systems access to perform AWS tasks and access your AWS resources
  - Web Identity Federation - let users sign in using a well-known third party provider such as Amazon, Facebook, Google, and OpenID Connect (OIDC)
- AWS STS supports following APIs
  - AssumeRole
  - AssumeRoleWithSAML
  - AssumeRoleWithWebIdentity
  - DecodeAuthorizationMessage

  - GetCallerIdentity
  - GetFederationToken
  - GetSessionToken

Web Identity Federation

- Federation consist of identity provider and identity consumer
- Consumer – stores references to identity and provides authorization locally
- Producer – stores identities, provides mechanisms for authentication
- Five mechanisms AWS federation can facilitate:
  - Custom IDP
  - Cross-account access
  - SAML
  - OIDC
  - Microsoft Active Directory
- gives users access to AWS resource after successfully authenticated with a web-based identity provider. facebook, amazon, google, OIDC
- cross account access – multiple accounts using one set of credentials
  - switching roles
  - two accounts – source account and target account
    - target account has a permissions policy and trust policy
    - issued short-term credentials using AssumeRole
- after success authentication, user receives authentication code from web id provider, which trade for AWS security credentials
- AWS SSO – AWS Service to manage access
- note: Amazon recommends Cognito because it does work for you

AWS Cognito

- an identity broker that handles interactions between applications and web id provider
- provides Web Identity Federation with following features
  - sign up and sign in
  - guest user access

  - acts as identity broker
  - synchronize user data for multiple devices
  - recommended for all mobile applications
- uses user pools to manage user signup, signing, directly or via web identity providers
- uses push notification to devices associated with a user id
- preferred way to use web identity federation
- how it works:
  - start using app
  - login with provider
  - exchange token for Cognito token
  - exchange Cognito token for temporary AWS credentials
  - use temporary credentials to access AWS services
- cognito streams - receives events as data is updated and synchronized. can use this to analyze info on the users in Cognito

EBS - Elastic Block Storage

- Amazon Elastic Block Store (EBS) is an easy to use, high performance block storage service designed for use with Amazon Elastic Compute Cloud (EC2) for both throughput and transaction intensive workloads at any scale. A broad range of workloads, such as relational and non-relational databases, enterprise applications, containerized applications, big data analytics engines, file systems, and media workflows are widely deployed on Amazon EBS[2]
- persistent block-based disk storage attached to EC2 instances
- can attach multiple EBS to an EC2 instance
- cannot attach an EBS to multiple EC2 instances
- AttachVolume API call to attach a volume to a region
- incur costs even when not attached
- note: NOT elastic file system, this is not was beanstalk uses
- volume types: Solid State Drive, IOPS, Magnetic
  - SSD - general purpose and < 10,000 IOPS
  - IOPS - faster > 10,000 IOPS
  - Magnetic - cheap but slow, rotational disk storage

[2] AWS Website

- limits
  - maximum 5000 EBS volumes
  - maximum 10000 EBS snapshots
  - maximum total volume per disk type 20TiB
  - maximum total provisioned IOPS is 40,000

EC2 - Elastic Cloud Compute

- Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.[3]
- reserved - allows user a capacity reservation
- spot - instances start when price is below a threshold, flexible
- on demand - allows you to pay a fixed rate by the hour for EC2
- GPU instance type - G2, G3, P2, P3, F1
- Compute optimized instance type - C3, C4, C5
- General purpose instance type - T2, M3, M4, M5
- Memory Optimized instance type - R3, R3, X1, X1E
- Storage Optimized instance type - I2, I3, D2, H1
- EC2 instance limit per region - 20
- when EC2 instance is stopped, container instance status remains Active but ECS container agent status changes to FALSE

S3 - Simple Storage Service

- Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data, at any time, from anywhere on the web. It gives any developer access to the same highly scalable, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of web sites. The service aims to maximize benefits of scale and to pass those benefits on to developers.[4]
- object-based storage for static files
- total volume and number of objects can store are unlimited

[3] AWS Website

[4] AWS Website

- objects stored in buckets
- objects larger than 100 MB should use Multipart Upload
- bucket names are universal and must be unique globally
- max file size 5TB
- max PUT operation: 5GB
- url format: https://s3-<region>.amazonaws.com/<bucket>/<object>
- static website url format: https://<bucket-name>.s3-website-<region>.amazonaws.com
- encryption
  - in transit - SSL/TLS
  - at rest server side encryption
    - S3 managed keys
    - AWS Key management system
    - Server Side encryption with customer keys
- read after write consistency - upload new file, available immediately
- eventual consistency - overwrite or delete, takes time to propagate
- aws:MultiFactorAuthPresent denies users without MFA authentication
- before deleting a bucket it must be empty
- use multi-part uploads for files greater than 5 KB
  - can start a multi-part upload as file is being created
- Amazon S3 Inventory tool generates a report of objects in a bucket, can use to see versions etc. if you notice a slowdown.
- Increase GET throughput
  - Use cloudfront distribution in front of bucket
  - Use sequential data-based naming prefix

AWS CloudFront

- Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment.[5]
- content delivery system so nodes are closer to user for images etc.

[5] AWS Website

- high-performance content delivery w/out contracts
- supports live/on-demand streaming
- edge location - endpoint for cloud front, over 60

AWS CloudFormation

- AWS CloudFormation provides a common language for you to describe and provision all the infrastructure resources in your cloud environment. CloudFormation allows you to use a simple text file to model and provision, in an automated and secure manner, all the resources needed for your applications across all regions and accounts. This file serves as the single source of truth for your cloud environment.[6]
- deploy infrastructure using scripts, allows consistent setup of resources repeatedly
- free, but resources created are billed
- CloudFormation template - declaration of AWS resources that make up a stack
- uses templates (JSON or YAML)
- create a stack.
- template sections: description, metadata, parameters, mappings, conditions, transforms, resources, outputs
    - conditions - used for reusing templates based on scenarios
    - parameters can define input in conditions section evaluate params
- modify cloudformation at runtime using parameters combined with conditions section of template
- max 200 resources in a stack
- 60 parameters and 60 outputs max allowed in template
- if stack formation fails, rolled back
- required section of a template - Resources
- data types in template: String, Number, List<Number>

SNS - Simple Notification Service

- Amazon Simple Notification Service (SNS) is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and serverless applications[7]

[6] AWS Website

[7] AWS Website

- send messages based on actions and triggers
- push
- supported formats: HTTP/HTTPS, Email, Email-JSON, SQS, SMS
- can use to fan-out SQS messages
- message contains: messageid, timestamp, topicarn, type, unsubscribeURL, message, subject, signature, signature version
- topic name <= 256 characters
- max 10 million subscribers per account
- max 100,000 topics per account
- message can contain max of 256KB or text data

SQS - AWS Simple Queue Service

- web service that provides a message queue for applications
- pull based
- FIFO - first in first out (message sent once)
- messages sent one or more times and in any order
- short polling and long polling
    - long polling waits, short polling does not
- VisibilityTimeout - parameter controls how long message is not visible once pulled from queue
- ChangeMessageVisibility - parameter can prolong VisibilityTimeout period
- ReceiveMessageWaitTimeSeconds - parameter controls long polling on/off
- DelaySeconds - parameter hide messages from clients to the queue
- Dead Letter Queue - used to hold messages that cannot be processed
- MessageRetentionPeriod - parameter how long SQS holds message in queue
- minimum message size is 1KB
- maximum message size is 256KB
- text based messages
- can contain up to 10 metadata attributes
- queue name up to 80 characters long
- queue name is case-sensitive
- object key names are stored lexicographically (alphabetical order)

- default visibility timeout is 30 seconds
- SQS allowed including structured metadata with message

DynamoDB

- AWS NoSQL database
- key/value storage
- automatically replicated on 3 azs within region
- eventually and strongly consistent supported
- streams - like transaction logs
- triggers - use AWS lambda
- scan vs query: query finds selected items, scan selects all items
- query is more efficient then scan
- query - uses primary key attribute values
  - provide partition key attribute and value to search for
  - returned ascending order unless set ScanIndexForward is true
- query can retrieve up to 1 MB of data
- scan
  - use ProjectionExpression parameter to remove attributes unwanted, otherwise all
- read provisioned throughput
  - units 4KB increments
  - eventually consistent - 2 reads per second
  - strongly consistent - 1 read per second
- write provisioned throughput
  - 1KB with 1 write per second
- local and global secondary indexes
  - global - can be added after table creation
    - supports eventual reads not consistent
  - local - only created at table creation
- CreateTable, UpdateTable, and DeleteTable limit - 10
- Max DeleteItem - 25

- Max PutItem - 25
- Max BatchGetItem and BatchWriteItem size < 16 MB
- max item size 400 KB
- read operations - Query, Scan, GetItem, BatchGetItem
- BatchGetItem max items - 100
- AmazonDynamoDB Accelerator - used to optimize frequent reads
- global tables - used for multi-region, multi-master database
- DynamoDB Accelerator (DAX) offers fastest reads, faster than elastic cache, allows fast in-memory performance cache
- need to encrypt global secondary index - use table key with AWS owned CMK
- table grows and scans cause throttling errors - reduced page size would avoid
- global tables are a good solution for reducing request latency from different regions
- condition expression - used to return specific attributes during put, update, delete operations
- DAX does not cache for strongly consistent reads - requests are forwarded to DynamoDB and results are not cached
- to have a search criteria for a query need to specify key condition expression and specify partition key name and value in equality condition
- DynamoDB encryption is mandatory - two types
  - default - AWS owned CMK
  - KMS - AWS managed CMK
- projection expression - identifies the attributes you want for read. available for scan and query
- if you enable TTL you can use any name attribute to store expiry timestamp
- if you need to pass multiple partition and/or sort keys to query, use BatchGetItem

Elastic Beanstalk

- With Elastic Beanstalk, you can quickly deploy and manage applications in the AWS Cloud without having to learn about the infrastructure that runs those applications. 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.[8]
- To use Elastic Beanstalk, you create an application, upload an application version in the form of an application source bundle (for example, a Java .war file) to Elastic Beanstalk, and then provide some information about the application. Elastic Beanstalk automatically launches an environment and

[8] AWS Website

creates and configures the AWS resources needed to run your code. After your environment is launched, you can then manage your environment and deploy new application versions.[9]

- servers - Apache, Nginx, Passenger, IIS
- languages - Java, .NET, PHP, Node.js, Python, Ruby, Go, Docker
- deployment options: all at once, rolling, rolling with additional batch, immutable, blue/green
  - all at once - deploys to all instances simultaneously
  - rolling - standard rolling deployment
  - rolling with additional batch - creates extra batch, allows full capacity
  - immutable and blue/green - create new instances
  - deploys to existing: all at once, rolling, rolling with additional batch
  - blue/green - deploy new version to separate environment then swap CNAMEs of the two environments to redirect traffic to new instance
- configuration files go in a top level folder named .ebextensions
- max applications - 75
- max versions - 1000
- max environments - 200
- max size of uploaded file - < 512 MB
- rollout modes that create new instance and deploy code to new instances
  - immutable deployment and blue/green deployment
- rollout mode that ensures infrastructure maintain full capacity
  - immutable, rolling with additional batch
- application version lifecycle policy can be added/used to prevent hitting application version limit
- say a worker application is going to run periodic tasks - use a cron.yaml file
- web application and worker application
- cron.yaml – configure a periodic task

AWS Continuous Integration & Continuous Deployment

- phases:
  - source phase
  - build phase
  - test phase

[9] AWS Website

- deployment phase
- monitor phase

- CI/CD continuous integration, continuous delivery
- code commit and code repository
- code build - build management system
- code deploy - deploying
- can tie all together using Code Pipeline

Code Commit

- AWS CodeCommit is a fully-managed source control service that hosts secure Git-based repositories. It makes it easy for teams to collaborate on code in a secure and highly scalable ecosystem. CodeCommit eliminates the need to operate your own source control system or worry about scaling its infrastructure. You can use CodeCommit to securely store anything from source code to binaries, and it works seamlessly with your existing Git tools.[10]
- fully managed source control service
- is a Git repository
- data encrypted in transit and at rest
- centralized repository for code, binaries, images, libraries
- maintains version history
- manages, updates, enables collaboration

Code Deploy

- AWS CodeDeploy is a fully managed deployment service that automates software deployments to a variety of compute services such as Amazon EC2, AWS Fargate, AWS Lambda, and your on-premises servers. AWS CodeDeploy makes it easier for you to rapidly release new features, helps you avoid downtime during application deployment, and handles the complexity of updating your applications. You can use AWS CodeDeploy to automate software deployments, eliminating the need for error-prone manual operations. The service scales to match your deployment needs.[11]
- fully managed automated deployment service and can be used as part of the continuous delivery or continuous deployment
- deployment options

[10] AWS Website
[11] AWS Website

- in place or rolling - stop application on each host and deploy latest code
- blue/green - new instance provisioned and new application deployed to new instances
  - blue is active
  - green is new release
  - blue/green deployment, performing
    - open beanstalk console
    - clone current environment
    - deploy new version
    - test
    - swap environment urls
  - deployed to existing
    - all at once
    - in place
    - rolling
    - rolling w/ additional batch
  - deployed to new instances
    - blue/green
    - immutable
  - Rollback mechanism
    - Redeploy
      - All at once
      - In place
      - Rolling
      - Rolling w/ additional batch
      - immutable
    - Swap url
      - Blue/green

- run-order hooks
  - BeforeBlockTraffic -> BlockTraffic -> AfterBlockTraffic
  - ApplicationStop
  - BeforeInstall

    - Install
    - AfterInstall
    - ApplicationStart
    - ValidateService
    - BeforeAllowTraffic -> AllowTraffic -> AfterAllowTraffic
- AppSpec.json - specify how application is deployed to underlying instance

CodeBuild

- AWS CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy. With CodeBuild, you don't need to provision, manage, and scale your own build servers. CodeBuild scales continuously and processes multiple builds concurrently, so your builds are not left waiting in a queue. You can get started quickly by using prepackaged build environments, or you can create custom build environments that use your own build tools[12]
- compile source code, run tests, and package code

Code Pipeline

- AWS CodePipeline is a fully managed continuous delivery service that helps you automate your release pipelines for fast and reliable application and infrastructure updates. CodePipeline automates the build, test, and deploy phases of your release process every time there is a code change, based on the release model you define. This enables you to rapidly and reliably deliver features and updates. You can easily integrate AWS CodePipeline with third-party services such as GitHub or with your own custom plugin.[13]
- fully managed continuous integration and continuous delivery service, fully automates build, test, deployment process
- orchestrate build, test, and application deployment
- automates end-to-end software release process based on user defined workflow
- automatically trigger pipeline as soon as change detected in source code repository
- integrates with other AWS services like CodeBuild and CodeDeploy

RDS - Relational Database Service

[12] AWS Website
[13] AWS Website

- a web service that makes it easier to setup, operate, and scale relational DB in the cloud
- dbengines: MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server
- db storage types: magnetic, solid state drive, iops
- can reduce load on DB by routing read queries from applications to the read replica

AWS X-Ray

- allows you to see traces in your AWS Lambda function which can allow you to see detailed tracing to your downstream services
- collects data about requests application serves, tools to view, filter, and gain insights into that data to identify issues and opportunities for optimization
- default sampling - one request per second
- there is an X-Ray API
    - interceptors to add to code to trace incoming HTTP requests
    - client handlers to instrumenting AWS SDK clients that your app can use to call other AWS services
    - http client to use instrument calls to other internal and external HTTP web services
- three environment variables might use
    - X_AMZN_TRACE_ID
    - AWS_XRAY_CONTEXT_MISSING
    - AWS_XRAY_DAEMON_ADDRESS
- setting up X-ray with lambda
    - get execution role for lambda function
    - attach following managed policy example: AWSXrayWriteOnlyAccess
    - to deploy X-Ray daemon to an ECS cluster you create a docker image with the X-Ray daemon then deploy docker image to the cluster
- default sampling rule - one request per second & 5 percent of any additional request per host

AWS Route53

- Amazon's domain name system (DNS) web service
- three main functions:
    - domain registration

    - dns routing
    - health checking
- say your using Route53 and you want to route portion of users to beta version, use route 53 failover routing policies

Kinesis Data Firehose

- fully managed service for delivering real-time streaming data to destinations
- destinations: S3, RedShift, ElasticSearch, Splunk
- don’t need to write applications or manage resources - configure data producers to send data to kinesiss datafirehose and it delivers data to the destination
- create a delivery system & then sending data to it
- record <= 1000 KB

AWS SAM - Serverless Application Model

- an extension to CloudFormation used to define server less applications (Lambda)
- to use options: via templates or via AWS SAM CLI
- can nest applications, max 200
- deployment preference types - Canary10Percent30Minutes, Canary10Percent15Minutes, Canary10Percent10Minutes, Canary10Percent5Minutes, Linear10PercentEvery10Minutes
    - Canary10Percent15Minutes - send 10% traffic to new version and 15 minutes later complete by sending remainder
    - Canary10Percent5Minutes - send 10% traffic to new version and 5 minutes later complete by sending remainder
    - Linear10PercentEvery10Minutes - send 10% traffic to never version every 10 minutes
- AWS::Serverless::Application - used to embed application using SAM template
- AWS::Serverless::Function - describes configuration for creating a lambda function
- AWS::Serverless::LayerVersion - creates Lambda layered function
-

AWS CloudWatch

- monitors infrastructure and applications that run on AWS

- tracks metrics and generates alarms from the metrics
- cloud watch namespaces - containers in which metrics for different applications are stored. ensures application metrics not mixed up
- by default stores data at 5 minute intervals
- advanced monitoring can store at 1 minute intervals
- retains data 15 months before discarding
- lower level granularity discarded sooner
  - 1 min - 15 days
  - 5 min - 63 days
- can establish a threshold that is constantly monitored and triggers an action when threshold is breached
- alarm states: OK, ALARM, INSUFFICIENT_DATA
- monitoring types
  - basic monitoring - free 5 min intervals
  - detailed monitoring - chargeable 1 minute intervals
- monitors: cpu, network, disk, status check
- note ram is a custom metric
- to use cloud watch locally you must download and install SSM Agent and CloudWatch Agent
- maximum of 5 alarms per region
- if you wanted to create an alarm with monitoring frequency of every 10 seconds you would create a high-resolution custom Amazon CloudWatch metric

AWS KMS - Key Management Service

- Customer Master Key (CMK) generates data keys
- GenerateDataKey command - generates encrypted key and plain text
- GenerateDataKeyWithoutPlainText - generates encrypted key only
- AWS Managed CMK is chargable while AWS Owned CMK is free

ELB - Elastic Load Balancing

- automatically distributes incoming application traffic across multiple Amazon EC2 instances

- allows fault tolerance in applications
- types: classic load balancer, application load balancer
    - classic load balancer
        - network level
        - does not look at packets
        - no access to HTTP and HTTPS headers
        - “dumb”
    - application load balancer
        - sophisticated, powerful
        - inspects packets
        - access to HTTP and HTTPS headers
    - listeners configured for load balancer
        - front-end protocol and port (client —> lb)
        - back-end protocol and port (lb —> instance)
    - supported protocols: HTTP, HTTPS, TCP, SSL
- max of 20 load balancers per region
- max of 100 listeners per load balancer

SWF - Simple Workflow Service

- Amazon SWF helps developers build, run, and scale background jobs that have parallel or sequential steps. You can think of Amazon SWF as a fully-managed state tracker and task coordinator in the Cloud.[14]
- makes it easy to build applications that coordinate work across distributed components
- domain, worker, decider
- domain - container that isolates workflow
- worker - (activity) - receives tasks, executes, returns results
- decider - flow logic
- maximum workflow length is one year
- task is assigned once and only once

[14] AWS Website

API Gateway

- Resource Policies allows denying or permitting specific IP addresses to access API Gateway
- API Caching - way to minimize request latency to requests of the API gateway service
- AWS::Serverless::API - creates gateway resources and methods that can be invoked through HTTPS endpoints
- Integration Request is how API Gateway interacts with back-end services such as DynamoDB
- what can API use as authentication mechanism - lambda authorizers, AWS Cognito, API keys

Lambda

- allows running code without a server. Triggered by events or called via API Gateway
- authoring options: from scratch, blueprints, serverless application repository
- provide 3 details when creating: name, runtime, role
- advantages – no servers to manage, do not have to worry about scaling
- languages: C#, Go, Java, Node.js, python
- A function can specify up to 5 layers
- Execution Role - the function's policy that grants it permission to access AWS services and resources
- AppSpec file - used in AWS Lambda to specify the function version to deploy and functions to be used as validation tests
- handler - name of the method that Lambda calls to execute function
- lambda function default timeout - 3 seconds.
- lambda function timeout can be set - between 1 and 300 seconds
- the routing-config parameter when updating an alias allows specifying % of incoming traffic goes to each version
- CPU assigned to function is automatic, you choose the amount or memory for a function
- outbound: TCP/IP and UDP/IP sockets supported
- inbound: network connections blocked
- for sensitive information Amazon recommends client-side encryption using AWS KMS and store values as environment variables
- Lambda Layer is how you share and manage frameworks, SDKs, libraries, across functions
- respond to dynamodb change, subscribe Lambda function to DynamoDB Stream associated with table.
- can invoke a Lambda function from code directly using Lambda's invoke API
- lambda functions being invoked asynchronously are retried at least 3 times

- push model and pull model options
- Two parts of the AWS Lambda function
  - Function package
  - Function handler
- Event object and context object are passed to function
- lambda execution permissions – enable AWS lambda fundtions to access other AWS resources in your account
  - LambdaBasicExecutionRole
  - LambdaKinesisExecutionRole
  - LambdaDynamoDBExecutionRole
  - LambdaVPCAccessExecutionRole

Elasticache

- Amazon ElastiCache offers fully managed Redis and Memcached. Seamlessly deploy, run, and scale popular open source compatible in-memory data stores. Build data-intensive apps or improve the performance of your existing apps by retrieving data from high throughput and low latency in-memory data stores. Amazon ElastiCache is a popular choice for Gaming, Ad-Tech, Financial Services, Healthcare, and IoT apps.[15]
- Redis, Memcached
- Redis - Manage and analyze fast moving data with a versatile in-memory data store.[16]
- Memcached - Build a scalable Caching Tier for data-intensive apps.[17]
- Memcached Only
  - Multithreaded Architecture
- Redis Only
  - Advanced data structures, snapshots, replication, transactions, pub/sub, lua scripting, geospatial support
- sub-millisecond response times - in memory so quicker than databases
- distribute data among multiple nodes
- choose memcached if (but remember limitations - Redis only)
  - simplest possible, large nodes with multicores

[15] AWS Website
[16] AWS Website
[17] AWS Website

- scale in and out as nodes needed
- cache objects

Elastic Container Service

- fully managed orchestration service available in AWS
-

AWS Kinesis Streams & Kinesis Data Firehouse

- if you need to encrypt kinesis stream then enable server-side encryption for Kinesis streams
-

Miscellaneous

- Web Application Firewall (WAF) - allows configuring rules to allow, block, monitor web requests, protects web applications from external attack
- service catalog - allows organizations to centrally manage commonly deployed IT services and achieve consistent governance
- ARN - Amazon Resource Name
- CodeStar - a cloud-based service for creating, managing, and working with software development projects on AWS
- OpsWorks - configuration management service that provides managed instances of Chef and Puppet to automate server config, deployment, and management
  - configuration management service that uses Chef, an automation platform that treats server configuartion as code. uses Chef to automate how servers are configured, deployed, and managed across your AWS EC2 instances or on-premise compute environments
- Storage Gateway - connect on-premise applications or services with AWS cloud storage
- Data Pipeline - data movement and processing within AWS and also between on-premise and AWS cloud
- redshift - fully managed petabyte scalable columnar data warehouse
- Elastic File System - fully managed, scalable, sharable storage system
- exponential backoff algorithm - use progressively longer waits between tries
- System Manager Parameter Store - use this to store secret database credentials etc.

Sunday, September 22, 2019

- Access-Control-Allow-Headers, Access-Control-Allow-Orgin are headers required by Web browser for CORS